ഹയർ സെക്കന്ററി കോഴ്സ്

കമ്പ്യൂട്ടർ ആപ്ലിക്കേഷൻസ്
(കോമേഴ്സ്)

ക്ലാസ് - XI

കേരളസർക്കാർ
പൊതുവിദ്യാഭ്യാസ വകുപ്പ്
സംസ്ഥാന വിദ്യാഭ്യാസ ഗവേഷണ പരിശീലന സമിതി (SCERT); കേരളം
2019

ദേശീയഗാനം

ജനഗണമന അധിനായക ജയഹേ
ഭാരത ഭാഗ്യവിധാതാ,
പഞ്ചാബസിന്ധു ഗുജറാത്ത മറാഠാ
ദ്രാവിഡ ഉത്ക്കല ബംഗാ,
വിന്ധ്യഹിമാചല യമുനാഗംഗാ,
ഉച്ഛ്വല ജലധിതരംഗാ,
തവശുഭനാമേ ജാഗേ,
തവശുഭ ആശിഷ മാഗേ,
ഗാഹേ തവ ജയ ഗാഥാ
ജനഗണമംഗലദായക ജയഹേ
ഭാരത ഭാഗ്യവിധാതാ
ജയഹേ, ജയഹേ, ജയഹേ,
ജയ ജയ ജയ ജയഹേ!

പ്രതിജ്ഞ

ഇന്ത്യ എന്റെ രാജ്യമാണ്. എല്ലാ ഇന്ത്യക്കാരും എന്റെ സഹോദരീ സഹോദരന്മാരാണ്.

ഞാൻ എന്റെ രാജ്യത്തെ സ്നേഹിക്കുന്നു; സമ്പൂർണവും വൈവിധ്യപൂർണവുമായ അതിന്റെ പാരമ്പര്യത്തിൽ ഞാൻ അഭിമാനം കൊള്ളുന്നു.

ഞാൻ എന്റെ മാതാപിതാക്കളെയും ഗുരുക്കന്മാരെയും മുതിർന്നവരെയും ബഹുമാനിക്കും.

ഞാൻ എന്റെ രാജ്യത്തിന്റെയും എന്റെ നാട്ടുകാരുടെയും ക്ഷേമത്തിനും ഐശ്വര്യത്തിനും വേണ്ടി പ്രയത്നിക്കും.

Prepared by :

State Council of Educational Research and Training (SCERT)
Poojappura, Thiruvananthapuram 695012, Kerala

Website : www.scertkerala.gov.in *e-mail* : scertkerala@gmail.com
Phone : 0471 - 2341883, Fax : 0471 - 2341869
Typesetting and Layout : SCERT
© Department of Education, Government of Kerala

To be printed in quality paper - 80gsm map litho (snow-white)

Foreword
ആമുഖം

ഏതു വിജ്ഞാനവും മാതൃഭാഷയിൽ പഠിക്കാനും പ്രകാശനം ചെയ്യാനും സാധിക്കും. അതിനുള്ള അവസരം പഠിതാക്കൾക്ക് ഒരുക്കേണ്ടത്, ഏതൊരു പഠന സമ്പ്രദായത്തിന്റെയും അനിവാര്യതയാണ്. അതിന്റെ തുടക്കമെന്ന നിലയ്ക്കാണ് ഹയർസെക്കന്ററി തലത്തിൽ ഭാഷേതര വിഷയങ്ങളിലെ പാഠപുസ്തകങ്ങൾ മലയാളത്തിൽ പ്രസിദ്ധീകരിക്കുന്നത്.

മാതൃഭാഷയിലൂടെയുള്ള വിദ്യാഭ്യാസം, ജ്ഞാനസമ്പാദനത്തിനുള്ള സുഗമമാർഗം എന്നതിനോടൊപ്പം സാംസ്കാരികത്തനിമയുടെ തിരിച്ചറിയൽ കൂടിയാണ്. അതുകൊണ്ടാണ് വികസിതരാജ്യങ്ങൾ മാതൃഭാഷയെ മുഖ്യബോധന മാധ്യമമായി സ്വീകരിച്ചിരിക്കുന്നത്. ഇന്ത്യയിലാകട്ടെ, ദേശീയതലത്തിലുള്ള പ്രധാന പരീക്ഷകളെല്ലാം പ്രാദേശിക ഭാഷകളിൽക്കൂടി നടത്തുന്നതിനുള്ള സംവിധാനവും ഉണ്ടായി വരികയാണ്. ഈയൊരു സാഹചര്യത്തിൽ നമ്മുടെ കുട്ടികളും മാതൃഭാഷയുടെ ശക്തിസൗന്ദര്യങ്ങൾ തിരിച്ചറിഞ്ഞ് വിവിധ വിഷയങ്ങളിൽ ജ്ഞാനനിർമിതിയിൽ ഏർപ്പെടേണ്ടതുണ്ട്. അതിന് അവരെ സജ്ജരാക്കുകയാണ് ഈ പാഠപുസ്തകങ്ങളുടെ മുഖ്യ ലക്ഷ്യം.

പരിഭാഷപ്പെടുത്തിയ പുസ്തകങ്ങളിൽ അതത് വിഷയങ്ങളിലെ സാങ്കേതിക പദങ്ങൾ പരമാവധി മലയാളത്തിലാക്കിയിട്ടുണ്ട്. നമ്മുടെ ഭാഷയിൽ ചിരപരിചിതമായ ഇംഗ്ലീഷ് പദങ്ങളെ അതേപടി സ്വീകരിച്ചിട്ടുമുണ്ട്. വിവർത്തനത്തിന് തീർത്തും വഴങ്ങാത്ത പദങ്ങളെ അതേരീതിയിൽ തന്നെ ഉപയോഗിച്ചിരിക്കുന്നു. മാതൃഭാഷയിൽ പഠിക്കുന്നവർക്ക് ആശയഗ്രഹണം സുഗമമാക്കുന്ന വിധത്തിലാണ് പാഠപുസ്തകരചന നടത്തിയിരിക്കുന്നത്. അതോടൊപ്പം മലയാളഭാഷയുടെ വളർച്ചയ്ക്കും ഈ പ്രവർത്തനം സഹായകമാകുമെന്ന് കരുതുന്നു.

പാഠപുസ്തകവിവർത്തന രംഗത്ത് നമ്മുടെ രാജ്യത്ത് നടന്ന വലിയൊരു കാൽവെപ്പാണ് ഇത്. പ്രഥമ സംരംഭമെന്നനിലയിൽ പല പരിമിതികളും പരിഭാഷയിൽ വന്നിട്ടുണ്ടാകാം. ക്ലാസ്മുറിയിൽ പ്രയോഗത്തിൽ വരുമ്പോഴാണ് അവയെല്ലാം കൂടുതൽ ബോധ്യപ്പെടുക. തുടർന്ന് വരുന്ന ഘട്ടങ്ങളിൽ അവയൊക്കെ പരിഹരിക്കുന്നതിന് എല്ലാ അഭ്യുദയകാംക്ഷികളിൽ നിന്നും വിശിഷ്യ അധ്യാപകർ, വിദ്യാർത്ഥികൾ എന്നിവരിൽ നിന്നും അഭിപ്രായങ്ങളും നിർദേശങ്ങളും പ്രതീക്ഷിക്കുന്നു.

ഡോ.ജെ. പ്രസാദ്

ഡയറക്ടർ,

എസ്.സി.ഇ.ആർ.ടി. കേരളം

പാഠപുസ്തക നിർമാണ സമിതി
കംപ്യൂട്ടർ ആപ്ലിക്കേഷൻസ് - കോമേഴ്സ്

ശ്രീ. ജോയ് ജോൺ
എച്ച്.എസ്.എസ്.ടി.,
സെന്റ് ജോസഫ്സ് എച്ച്.എസ്.എസ്.,
തിരുവനന്തപുരം

ശ്രീ. വിനോദ്.വി
എച്ച്.എസ്.എസ്.ടി., എൻ.എസ്.എസ്
എച്ച്.എസ്.എസ്., പ്രാക്കുളം, കൊല്ലം

ശ്രീ. റോയ് ജോൺ
എച്ച്.എസ്.എസ്.ടി.,
സെന്റ് അലോഷ്യസ് എച്ച്.എസ്.എസ്.,
എൽതുരുത്ത്, തുരുത്ത്

ശ്രീ. അബൂബക്കർ.പി
എച്ച്.എസ്.എസ്.ടി., ഗവ.ജി എച്ച്.എസ്.എസ്.,
ചാലപ്പുറം, കോഴിക്കോട്

ശ്രീ. മുസ്തഫ ഷംസുൽ ഹക്ക് കെ.കെ
എച്ച്.എസ്.എസ്.ടി., ജി.എച്ച്.എസ്.എസ്,
കൂത്തുപറമ്പ്, കണ്ണൂർ

ശ്രീ. പ്രശാന്ത് പി.എം
എച്ച്.എസ്.എസ്.ടി., സെന്റ് ജോസഫ്സ്
ബോയ്സ് എച്ച്.എസ്.എസ്., കോഴിക്കോട്

ശ്രീ. അസീസ് വി.
എച്ച്.എസ്.എസ്.ടി.,
ജി.എച്ച്.എസ്.എസ്., വെള്ളിയോട്,
കോഴിക്കോട്

ശ്രീ. രാജ് മോഹൻ
എച്ച്.എസ്.എസ്.ടി.,
നവമുകുന്ദ എച്ച്.എസ്.എസ്.,
തിരുനാവായ, മലപ്പുറം

ശ്രീ. ഷാജൻ ജോസ് എൻ.
എച്ച്.എസ്.എസ്.ടി.,
സെന്റ്. ജോസഫ് എച്ച്.എസ്.എസ്.,
പവറട്ടി, തൃശൂർ

ശ്രീ. സായിപ്രകാശ്.എസ്
എച്ച്.എസ്.എസ്.ടി., സെന്റ് തോമസ്
എച്ച്.എസ്.എസ്, പൂന്തുറ, തിരുവനന്തപുരം

വിദഗ്ധർ

ഡോ. ലജീഷ് വി.എൽ.
അസ്സി. പ്രൊഫ. ഡിപ്പാർട്ട്മെന്റ് ഓഫ് കമ്പ്യൂട്ടർ സയൻസ് , കോഴിക്കോട് സർവകലാശാല

ഡോ. മധു എസ്സ്. നായർ
അസ്സി. പ്രൊഫ. ഡിപ്പാർട്ട്മെന്റ് ഓഫ് കമ്പ്യൂട്ടർ സയൻസ് , കേരള സർവകലാശാല

മധു വി.ടി.
ഡയറക്ടർ, കമ്പ്യൂട്ടർ സെന്റർ, കോഴിക്കോട് സർവകലാശാല

ഡോ. ബിനു.പി.ചാക്കോ
അസ്സി. പ്രൊഫ. ഡിപ്പാർട്ട്മെന്റ് ഓഫ് കമ്പ്യൂട്ടർ സയൻസ് , പ്രജ്യോതി നികേതൻ കോളജ്, പുതുക്കാട്, തൃശ്ശൂർ

ഡോ. സുശീൽ കുമാർ
അസ്സി. പ്രൊഫ. ഡിപ്പാർട്ട്മെന്റ് ഓഫ് ഇംഗ്ലീഷ്, ഡി.ബി.കോളജ്, ശാസ്താംകോട്ട

ഡോ. വിനീത് കെ. പാലേരി
പ്രൊഫസർ, ഡിപാർട്ടമെന്റ് ഓഫ് കമ്പ്യൂട്ടർ സയൻസ് ആന്റ് എഞ്ചിനിയറിംഗ്, എൻ.ഐ.ടി. കോഴിക്കോട്

ഡോ. മഹേശ്വരൻ നായർ വി.
സബ് ഡിവിഷണൽ എഞ്ചിനീയർ, റീജിയണൽ ടെലികോം ട്രെയിനിംഗ് സെന്റർ, കൈമനം, തിരുവനന്തപുരം

ആർട്ടിസ്റ്റ്

സുധീർ വൈ **വിനീത് വി**

അക്കാദമിക് കോർഡിനേറ്റർ

ഡോ. മീന എസ്.
റിസർച്ച് ഓഫീസർ, എസ്.സി.ഇ.ആർ.ടി

പാഠപുസ്തക പരിഭാഷ സമിതി
(മലയാളം)

ഡോ. ബിനു പി ചാക്കോ
അസോസിയേറ്റ് പ്രൊഫസർ, ഡിപ്പാർട്ട്മെന്റ് ഓഫ് കമ്പ്യൂട്ടർ സയൻസ്
പ്രജ്യോതി നികേതൻ കോളേജ്, പുതുക്കാട്

ഡോ. ഗ്ലാഡ്സ്റ്റൺ എസ്.രാജ്
അസിസ്റ്റന്റ് പ്രൊഫസർ & ഹെഡ്, ഡിപ്പാർട്ട്മെന്റ് ഓഫ് കമ്പ്യൂട്ടർ സയൻസ്
ഗവ. കോളേജ്, നെടുമങ്ങാട്

ഡോ. എസ്. ഗിരീഷ് കുമാർ
അസി. പ്രൊഫസർ, മലയാള വിഭാഗം,
ശ്രീ കേരളവർമ്മ കോളേജ്, തൃശൂർ

ഡോ. പ്രിയ ആർ
അസിസ്റ്റന്റ് പ്രൊഫസർ & ഹെഡ് ഡിപ്പാർട്ട്മെന്റ് ഓഫ് കമ്പ്യൂട്ടർ സയൻസ്
ഗവ. കോളേജ് കാര്യവട്ടം, തിരുവനന്തപുരം

ആദർശ്.വി.കെ
സീനിയർ മാനേജർ (ടെക്നിക്കൽ), യൂണിയൻ ബാങ്ക് ഓഫ് ഇന്ത്യ

ശ്രീ. അംബരീഷ് ജി.എസ്
എച്ച്.എസ്.എസ്.ടി.,
സി.കെ.ജി.എം.എച്ച്.എസ്.എസ്., ചിങ്ങപ്പുരം,
കോഴിക്കോട്

ശ്രീമതി. ദിവ്യ പി.എസ്.
ഗവ. നളന്ദ എച്ച്.എസ്.എസ്.,
കിഴുപ്പിള്ളിക്കര,
തൃശൂർ

ശ്രീ. മടവൂർ ശശി എ.
എച്ച്.എസ്.എസ്. (റിട്ട.)
തിരുവനന്തപുരം

ശ്രീ. മുഹമ്മദ് സർഫറാസ്
എച്ച്.എസ്.എസ്.ടി., ഡി.എച്ച്.എസ്.എസ്.
നെല്ലിപ്പുഴ

ശ്രീ. സി.വിനയചന്ദ്രൻ
എച്ച്.എസ്.എസ്.ടി., വി.ജി.എസ്.എസ്.,
അംബികോദയം എച്ച്.എസ്.എസ്.,
നെടിയവിള, കൊല്ലം

ശ്രീ. സജിൻ.ടി
എച്ച്.എസ്.എസ്.ടി., വേളം, എച്ച്.എസ്.എസ്.,
ചേരാപുറം, കോഴിക്കോട്

ശ്രീ. സുനിൽ കുമാർ
എച്ച്.എസ്.എസ്.ടി., ജി.എച്ച്.എസ്.എസ്.
തേമ്പാംമൂട്, തിരുവനന്തപുരം

ശ്രീ. സുധീർ പി.എസ്
എച്ച്.എ്.എ്.ടി., എ.കെ.എൻ.എം.എ,
മെമ്മോറിയൽ എച്ച്.എസ്.എസ്., കാട്ടുകുളം,
പാലക്കാട്

ശ്രീ. ശശി എ.
എച്ച്.എസ്.എസ്.ടി., ജി.വി.എച്ച്.എസ്.എസ്
വേങ്ങര, മലപ്പുറം.

ശ്രീ. ശ്രീജിത്ത് പി.
എച്ച്.എസ്.എസ്.ടി., സി.ജെ. എച്ച്.എസ്.എസ്.,
ചെമ്മനാട്, കാസർഗോഡ്

ശ്രീ. ഹരി.കെ
എച്ച്.എസ്.എസ്.ടി., ജി.വി.എച്ച്.എസ്.എസ്
തിരുവനന്തപുരം

അക്കാദമിക് കോർഡിനേറ്റർ

ശ്രീമതി റിയാന അൻസാരി
റിസർച്ച് ഓഫീസർ, എസ്.സി.ഇ.ആർ.ടി

CONTENTS

ഉള്ളടക്കം

1.	കമ്പ്യൂട്ടറിലെ അടിസ്ഥാന ഘടകങ്ങൾ	09
2.	കമ്പ്യൂട്ടർ സിസ്റ്റത്തിന്റെ ഭാഗങ്ങൾ	55
3.	പ്രോഗ്രാമിങ് തത്വങ്ങളും പ്രശ്നപരിഹരണവും	101
4.	C++ പ്രോഗ്രാമിങ് - ഒരു ആമുഖം	130
5.	ഡാറ്റയുടെ ഇനങ്ങളും ഓപ്പറേറ്ററുകളും	145
6.	പ്രോഗ്രാമിങ് - ഒരു ആമുഖം	163
7.	നിയന്ത്രണ പ്രസ്താവനകൾ	185
8.	കമ്പ്യൂട്ടർ ശൃംഖലകൾ	221
9.	ഇന്റർനെറ്റ്	263
10.	വിവര സാങ്കേതിക വിദ്യയുടെ ഉപയോഗം	295

G9Q2U6

പാഠപുസ്തകത്തിൽ ഉപയോഗിച്ചിരിക്കുന്ന സൂചനകൾ

നമുക്ക് ചെയ്യാം

നിങ്ങളുടെ പുരോഗതി അറിയുക

ഇൻഫർമേഷൻ ബോക്സ്

നമുക്ക് പരിശീലിക്കാം

പഠനനേട്ടങ്ങൾ

1

പ്രധാന ആശയങ്ങൾ

- ഡാറ്റയും വിവരവും
- ഡാറ്റ പ്രോസസിംഗ്
- കമ്പ്യൂട്ടറിലെ അടിസ്ഥാന പ്രവർത്തന ഘടകങ്ങൾ
- കമ്പ്യൂട്ടർ ഒരു ഡാറ്റ പ്രോസസർ എന്ന നിലയിൽ
  കമ്പ്യൂട്ടറിന്റെ സ്വഭാവ സവിശേഷതകൾ
- സംഖ്യാന സമ്പ്രദായങ്ങൾ
  - ദശസംഖ്യാന സമ്പ്രദായം
  - ദ്വയസംഖ്യാന (ബൈനറി സംഖ്യാന സമ്പ്രദായം)
  - അഷ്ടസംഖ്യാന സമ്പ്രദായം (ഒക്ടൽ സംഖ്യാന സമ്പ്രദായം)
  - ഷോഡശസംഖ്യാന സമ്പ്രദായം (ഹെക്സാഡെസിമൽ സംഖ്യാന സമ്പ്രദായം)
- സംഖ്യ പരിവർത്തനങ്ങൾ
  - ദശസംഖ്യയിൽ നിന്നും ദ്വയ സംഖ്യയിലേക്ക്
  - ദശസംഖ്യയിൽ നിന്നും അഷ്ട സംഖ്യയിലേക്ക്
  - ദശസംഖ്യയിൽ നിന്നും ഹെക്സാഡെസിമൽ സംഖ്യയിലേക്ക്
  - ദ്വയസംഖ്യയിൽ നിന്നും ദശ സംഖ്യയിലേക്ക്
  - അഷ്ടസംഖ്യയിൽ നിന്നും ദശ സംഖ്യയിലേക്ക്
  - ഹെക്സാഡെസിമൽ സംഖ്യയിൽ നിന്നും ദശ സംഖ്യയിലേക്ക്
  - അഷ്ടസംഖ്യയിൽ നിന്നും ദ്വയ സംഖ്യയിലേക്ക്
  - ഹെക്സാഡെസിമൽ സംഖ്യയിൽ നിന്നും ദ്വയ സംഖ്യയിലേക്ക്
  - അഷ്ട സംഖ്യയിൽ നിന്നും ഹെക്സാഡെസിമൽ സംഖ്യയിലേക്ക്
- ഡാറ്റ പ്രതിനിധാനം
  - സംഖ്യകളുടെ പ്രതിനിധാനം
  - അക്ഷരങ്ങളുടെ പ്രതിനിധാനം
  - ഓഡിയോ, ചിത്രം, വീഡിയോ ഇവയുടെ പ്രതിനിധാനം

I2Y3K8

കമ്പ്യൂട്ടറിലെ അടിസ്ഥാന ഘടകങ്ങൾ

കമ്പ്യൂട്ടറുകൾ നമ്മുടെ നിത്യജീവിതത്തിന്റെ ഭാഗമായി മാറിയിട്ടുണ്ട്. വ്യത്യസ്ത ആവശ്യങ്ങൾക്കും ഉദ്ദേശ്യങ്ങൾക്കുമായി ജനങ്ങൾ കമ്പ്യൂട്ടറുകൾ ഉപയോഗിക്കുന്നു. വിദ്യാഭ്യാസം, വ്യാപാരം, വിനോദം, ആശയവിനിമയം, സർക്കാർ സേവനങ്ങൾ, ഗതാഗതം എന്നിങ്ങനെ എല്ലാ മേഖലയിലും കമ്പ്യൂട്ടറുകൾ ഇന്ന് ഒഴിച്ചുകൂടാനാവാത്തവയാണ്. വിദ്യാർഥികളെ സംബന്ധിച്ചിടത്തോളം ഗണിതത്തിലെ പ്രാഥമിക ക്രിയകൾക്കു പുറമേ വിവിധങ്ങളായ വിഷയങ്ങൾ ഫലപ്രദമായി പഠിക്കുന്നതിനും പഠനപ്രവർത്തനങ്ങൾ കാര്യക്ഷമമായി ചെയ്യുന്നതിനും കമ്പ്യൂട്ടറുകൾ ഉപയോഗിക്കുന്നു. നിലവിൽ കമ്പ്യൂട്ടർ ഉപയോഗിക്കുന്ന സാഹചര്യങ്ങൾ മനസ്സിലാക്കി അവ മുഖാന്തിരം ഉണ്ടായ നേട്ടങ്ങൾ വിശകലനം ചെയ്യുന്നതിലൂടെ അവയുടെ അനന്ത സാധ്യതകൾ കൂടുതൽ മനസ്സിലാക്കാൻ കഴിയും. ആയതിനാൽ കമ്പ്യൂട്ടറുകളെയും അതിന്റെ പ്രയോഗ സാധ്യതകളെയും കുറിച്ച് കൂടുതൽ അറിയേണ്ടത് അത്യാവശ്യമാണ്. കമ്പ്യൂട്ടറിന്റെ പ്രവർത്തന ഘടകങ്ങൾ, ഡാറ്റ പ്രോസസിംഗിന്റെ (Data Processing) അടിസ്ഥാനാശയങ്ങൾ എന്നിവ ഈ അധ്യായത്തിൽ പ്രതിപാദിക്കുന്നു. കമ്പ്യൂട്ടറിൽ ഉപയോഗിക്കുന്ന വിവിധ ഡാറ്റ പ്രതിനിധാന രീതികളും ഇവിടെ ചർച്ച ചെയ്യുന്നു.

1.1 ഡാറ്റയും വിവരവും (Data and Information)

ഡാറ്റയും വിവരവും നമ്മളിൽ പലർക്കും സുപരിചിതമായ പദങ്ങളാണ്. ദൈനംദിന ജീവിതത്തിൽ ഈ പദങ്ങൾ നാം മിക്കപ്പോഴും പരസ്പരം മാറ്റി ഉപയോഗിക്കാറുണ്ട്. പക്ഷേ ഇവ തമ്മിൽ അടിസ്ഥാനപരമായ വ്യത്യാസങ്ങൾ നിലനിൽക്കുന്നുണ്ട്. കമ്പ്യൂട്ടർ മേഖലയിൽ സമഗ്ര പഠനം നടത്താനുള്ള ശ്രമത്തിന്റെ ഭാഗമായി ഈ പദങ്ങൾ തമ്മിലുള്ള വ്യത്യാസം മനസ്സിലാക്കേണ്ടത് വളരെ അത്യാവശ്യമാണ്.

ചിത്രം 1.1: ഡാറ്റയ്ക്ക് ഉദാഹരണം

ചിത്രം 1.1 ൽ ഒരു അധ്യാപകന്റെ ക്ലാസ് ഡയറിയുടെ ഒരു ഭാഗമാണ് കൊടുത്തിരിക്കുന്നത്. നിങ്ങൾക്കതിലെ വാക്കുകളും സംഖ്യകളും മനസ്സിലാകുന്നുണ്ടോ? ഏതൊരു അധ്യാപകന്റെ ക്ലാസ് ഡയറി ആയതിനാൽ വാക്കുകൾ ചില കുട്ടികളുടെ പേരുകളായിരിക്കാം. സംഖ്യകൾ എന്തിനെയാണ് സൂചിപ്പിക്കുന്നത്? അത് കുട്ടികൾ പരീക്ഷകളിൽ നേടിയ മാർക്കുകളാവാം, അവരുടെ ചില മാസങ്ങളിലെ ഹാജരാവാം, അല്ലെങ്കിൽ അതുപോലെ മറ്റെന്തെങ്കിലുമാവാം. നമ്മൾ ഈ വസ്തുതകളെയും കണക്കുകളെയും ഡാറ്റ എന്നു വിളിക്കുന്നു. കാരണം, അവ പൂർണമായ ആശയം നൽകുന്നില്ല. പ്രോസസ് ചെയ്യുവാനും കൈകാര്യം ചെയ്യുവാനും കഴിയുന്ന അക്കങ്ങൾ, വാക്കുകൾ, തുക, അളവ് മുതലായ അസംസ്കൃത വസ്തുതകളെയും കണക്കുകളെയും ഡാറ്റ എന്ന് സൂചിപ്പിക്കുന്നു .

ഈ വസ്തുതകളെയും കണക്കുകളെയും ചിത്രം 1.2 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ എഴുതിയിരുന്നെങ്കിൽ അവ എന്താണ് സൂചിപ്പിക്കുന്നതെന്ന് ആശയകുഴപ്പം ഉണ്ടാകുമായിരുന്നില്ല. നിരന്തര മൂല്യനിർണയ പ്രവർത്തനങ്ങളിൽ (CE) കുട്ടികൾ നേടിയ സ്കോറുകളാണ് (Scores) ഈ കണക്കുകൾ കാണിക്കുന്നതെന്ന് വ്യക്തമാണ്. അർഥപൂർണമായ രീതിയിൽ ഡാറ്റ ക്രമീകരിക്കുമ്പോൾ ഈ വസ്തുതകളെയും കണക്കുകളെയും കുറിച്ച് വളരെ വ്യക്തമായ ആശയം നമുക്ക് ലഭിക്കുന്നു. ഇത് **വിവരം (Infor-mation)** എന്നറിയപ്പെടുന്നു. അത് അർഥപൂർണമായതും പ്രോസസ് ചെയ്യപ്പെട്ടതുമായ ഡാറ്റയുടെ രൂപമാകുന്നു.

റോൾ നമ്പർ	പേര്	20- ലുള്ള സ്കോറുകൾ			
		അസൈൻമെന്റ്	പരീക്ഷ	സെമിനാർ	പ്രോജക്ട്
1	അനിത	19	19	20	19
2	ആദർശ്	20	18	18	19

ചിത്രം 1.2: വിവരത്തിന്റെ മാതൃക

വിവരം ഡാറ്റയായി വർത്തിക്കുന്ന മറ്റ് സന്ദർഭങ്ങളുമുണ്ട്. കുട്ടികളുടെ CE സ്കോറുകൾ തയാറാക്കുന്ന ഉദാഹരണത്തിൽ അധ്യാപകൻ ഈ സംഖ്യകളെ 10 എന്ന ഏകീകൃത സ്കോറിലേക്ക് മാറ്റുന്നു. അതുപോലെ പൊതു പരീക്ഷയുടെ ഉത്തരകടലാസുകൾ മൂല്യനിർണ്ണയം നടത്തിയതിനുശേഷം ഓരോ കുട്ടിക്കും 40 ലുള്ള സ്കോറുകൾ നൽകുന്നത്. പരീക്ഷാഫലം തയാറാക്കുന്ന സമയത്ത് എല്ലാ വിഷയങ്ങളുടെയും സ്കോറുകൾ ശേഖരിക്കുകയും അനുബന്ധമായ ഗ്രേഡുകൾ നൽകുകയും ചെയ്യുന്നു. വ്യക്തിഗതവിവരങ്ങളും ഗ്രേഡുകളും അനുയോജ്യമായ ലാബലുകളോടു കൂടി ഉചിതമായി രൂപത്തിൽ രേഖപ്പെടുത്തിയാൽ അത് ഒരു വിദ്യാർഥിയുടെ സ്കോർ ഷീറ്റായി മാറുന്നു. അത് വീണ്ടും വിവരമായി മാറുന്നു.

പൊതു പരീക്ഷയിൽ ഒരു വിദ്യാർഥിക്കു നൽകിയ സ്കോർഷീറ്റാണ് ചിത്രം 1.3 ൽ കാണിച്ചിരിക്കുന്നത്. വിദ്യാർഥിയുടെ വ്യക്തിഗത വിവരങ്ങളും ഓരോ വിഷയത്തിലും നേടിയ ഗ്രേഡുകളും അതിൽ അടങ്ങിയിരിക്കുന്നു. വ്യക്തിഗത വിവരങ്ങളായ അനിത മോഹൻ, സ്ത്രീ, 13/04/1997, എന്നിവ യഥാക്രമം പേര്, ലിംഗം, ജനന തീയ്യതി എന്നി ലേബലുകൾക്കു നേരെ അച്ചടിച്ചിരിക്കുന്നു. ഇവിടെ വസ്തുതകളും കണക്കുകളുമായി പ്രതിനിധീകരിക്കുന്ന വ്യക്തിഗത വിവരങ്ങളും ഗ്രേഡുകളും ഡാറ്റയാകുന്നു. ഈ ഡാറ്റ അനുയോജ്യമായ ലേബലുകളോടുകൂടി വ്യക്തമാക്കുമ്പോൾ അത് വിദ്യാർഥിയെ സംബന്ധിച്ച വിവരമായി മാറുന്നു. അപ്രകാരം ഒരു വിദ്യാർഥിയുടെ പരീക്ഷയിലെ പ്രകടനത്തെ സംബന്ധിച്ചുള്ള വിവരമാണ് സ്കോർ ഷീറ്റിൽ അടങ്ങിയിരിക്കുന്നതെന്ന് നമുക്കു പറയാം. അതിൽനിന്ന് വിദ്യാർഥിയുടെ വിവിധ വിഷയങ്ങളിലെ നിലവാര ത്തെപ്പറ്റിയുള്ള അറിവ് നമുക്കു ലഭിക്കുന്നു. ഉന്നത വിദ്യാഭ്യാസത്തെ സംബന്ധിച്ച് തീരുമാനങ്ങൾ എടുക്കുന്നതിനോ ഭാവി പദ്ധതികൾ തയാറാക്കുന്നതിനോ ഇത് അവരെ സഹായിക്കുന്നു.

No. N 389066

9HIH6G9GH9EEE9332583

GOVERNMENT OF KERALA
GENERAL EDUCATION DEPARTMENT
SECONDARY SCHOOL LEAVING CERTIFICATE

Register Number: 121367 Month & Year: MARCH 2013 No. of Chances: 1

1. Name of Candidate : ANITHA MOHAN
2. Name in Regional Language : അനിത മോഹൻ
3. Name of School St. Joseph's H S S
4. Admission No. : 1377
5. Date of Birth (in figures) : 13/04/1997
   (in words) : THIRTEENTH APRIL NINETEEN NINETY SEVEN
6. Sex : FEMALE
7. Religion & Caste : HINDU, NAIR
8. Whether SC/ST/OEC/OBC/GL : GENERAL
9. Nationality : INDIAN
10. Name of Mother : JAYANTHI L
11. Name of Father : RADHAKRISHNAN K S
12. Name of Guardian : RADHAKRISHNAN K S
13. Home Address : ANITHA BHAVAN, MUTTATHARA VALLAKKADAVU P O, THIRUVANANTHAPURAM
14. Identification Marks
1. A BLACK MOLE ON THE RIGHT SIDE OF THE CHEEK
2. A BLACK MOLE ON THE RIGHT EYEBROW.

This is to certify that the candidate herein has appeared for the SSLC Examination and secured the following grades

Subject	Grade	Grade in words
First Language Paper - I (MALAYALAM)	A+	A Plus
First Language Paper - II (MALAYALAM)	A+	A Plus
English	A+	A Plus
Hindi	A	A Only
Social Science	A	A Only
Physics	A+	A Plus
Chemistry	A	A Only
Biology	A	A Only
Mathematics	A	A Only
Information Technology	A	A Only

RANGE OF GRADES

A+ 90% and above : Outstanding	B 60% - 69% : Good	D+ 30% - 39% : Marginal
A 80% - 89% : Excellent	C+ 50% - 59% : Above Average	D 20% - 29% : Need Improvement
B+ 70% - 79% : Very Good	C 40% - 49% : Average	E Below 20% : Need Improvement

Eligibility for higher studies - Minimum D+ grade for each paper

ELIGIBLE FOR HIGHER STUDIES

KERALA
School Seal

Name & Signature of the Head of School

Date of Publication of Result : 24/04/2013

JOHNS V. JOHN
SECRETARY
Board of Public Examinations, Kerala

ചിത്രം 1.3 : എസ്.എസ്.എൽ.സി. സ്കോർ ഷീറ്റ്

ഏതെങ്കിലും തരത്തിലുള്ള പ്രവർത്തനങ്ങൾക്ക് ഡാറ്റ വിധേയമാവുമ്പോഴാണ് വിവരം ഉളവാകുന്നത്. മറ്റൊരു തരത്തിൽ പറഞ്ഞാൽ ഡാറ്റ എന്നത് വിവരം ഉണ്ടാക്കുന്നതിനുള്ള അസംസ്കൃത വസ്തുവാണ്. നമുക്കിവിടെ ഈ രണ്ടു പദങ്ങളെയും വേർതിരിക്കാം (പട്ടിക1.1).

ഡാറ്റ	വിവരം
• അസംസ്കൃത വസ്തുതകളും കണക്കുകളും	• പ്രോസസ് ചെയ്ത ഡാറ്റ
• അസംസ്കൃത വസ്തുവിന് സമാനം	• പൂർത്തിയായ ഉൽപ്പന്നത്തിനു സമാനം
• നേരിട്ട് ഉപയോഗിക്കുവാൻ സാധിക്കാത്തത്	• അറിവു വർധിപ്പിക്കുന്നതിനും തീരുമാനങ്ങളെടുക്കുന്നതിനും സഹായിക്കുന്നത്
• കൃത്യമായ ധാരണയും വ്യക്തതയും നൽകുന്നില്ല.	• വ്യക്തവും അർഥ പൂർണവുമാണ്

പട്ടിക 1.1: ഡാറ്റയും വിവരവും തമ്മിലുള്ള താരതമ്യം

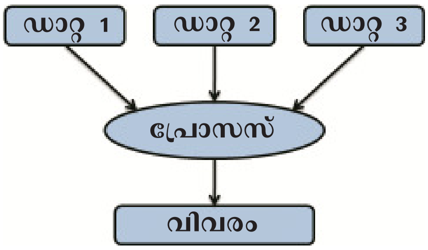
വിവരം എല്ലായിപ്പോഴും അറിവ് വർധിപ്പിക്കുമെന്ന് നമുക്ക് അറിയാമല്ലോ. പ്രശ്നങ്ങൾ പരിഹരിക്കുന്നതിനോ തീരുമാനങ്ങൾ എടുക്കുന്നതിനോ ഈ അറിവ് ഒരാൾക്ക് ഉപയോഗിക്കാം. നേടിയ അറിവിൽ നിന്ന് ഉപയോഗപ്രദമായ അനുമാനത്തിലേക്ക് എത്തിച്ചേരുന്നതിനുള്ള കഴിവിനെ പൊതുവേ ബുദ്ധിവൈഭവം (Intelligence) എന്നു പറയുന്നു. ഇത് എങ്ങനെ നാം അറിവ് പ്രോസസ് ചെയ്ത് വിവിധ സാഹചര്യങ്ങളിൽ പ്രയോഗിക്കുന്നു എന്നതിനെ ആശ്രയിച്ചിരിക്കുന്നു. അറിവിനും ബുദ്ധിക്കും ഇണങ്ങുന്ന വിധത്തിൽ ഈ കാലഘട്ടത്തിൽ മനുഷ്യർ ചെയ്യുന്നതുപോലെ കമ്പ്യൂട്ടറുകളെയും ആക്കിത്തീർക്കുന്നതിനുള്ള ശ്രമങ്ങളിൽ കമ്പ്യൂട്ടർ സയൻസിലും സാങ്കേതികവിദ്യയിലും സമീപകാലത്ത് വൻ പുരോഗതി ഉണ്ടായിട്ടുണ്ട്. ഇതിനെ നിർമ്മിത ബുദ്ധി (Artificial Intelligence) എന്ന് വിളിക്കുന്നു.

നമുക്കു ചെയ്യാം

- ഒരു ടെലിഫോൺ ബിൽ, വൈദ്യുതി ബിൽ, അല്ലെങ്കിൽ ജല ബിൽ പരിശോധിച്ച് അതിലടങ്ങിയിരിക്കുന്ന ഡാറ്റ തിരിച്ചറിയുക
- ഒരു കടയിൽ നിന്ന് ചില സാധനങ്ങൾ വാങ്ങുന്നു എന്നു കരുതുക. അതിൽ ഉൾപ്പെട്ട ഡാറ്റ തിരിച്ചറിഞ്ഞ് അവയെ എങ്ങനെ വിവരമാക്കി മാറ്റിയിരിക്കുന്നു എന്ന് മനസ്സിലാക്കുക.
- നിത്യ ജീവിതവുമായി ബന്ധപ്പെട്ട ഏതെങ്കിലും ഡാറ്റയും വിവരവും തിരച്ചറിയുക, നിങ്ങൾക്കവ വ്യക്തമായി വേർതിരിക്കാൻ കഴിയുമെന്ന് ഉറപ്പുവരുത്തുക

1.2 ഡാറ്റ പ്രോസസിങ് (Data processing)

സ്കോർഷീറ്റ് തയാറാക്കുന്നതിന്റെ വിശദാംശങ്ങൾ മുൻപ് സൂചിപ്പിച്ചുവല്ലോ. നിരന്തരമൂല്യ നിർണയത്തിന്റെയും (CE) പാദവാർഷിക മൂല്യ നിർണയത്തിന്റെയും (TE) ഭാഗമായി ഓരോ വിഷയത്തിനും നൽകിയ സ്കോർ ഒരുമിച്ചു കൂട്ടുകയും, ഗ്രേഡുകൾ തീരുമാനിക്കുകയും ചെയ്യുന്നത് മുൻകൂട്ടി നിശ്ചയിച്ച ചില മാനദണ്ഡങ്ങൾ അടിസ്ഥാനമാക്കിയാണ്. വിവരങ്ങൾ ലഭ്യമാക്കുന്നതിനുള്ള ഇത്തരം പ്രവർത്തനങ്ങളെ മൊത്തമായി പ്രോസസ് എന്നു പറയുന്നു. വിവരം ലഭ്യമാ



ചിത്രം 1.4 :ഡാറ്റ പ്രോസസിങ്

ക്കുന്നതിന് ഡാറ്റയിൽ നടത്തുന്ന പ്രവർത്തനങ്ങളെ ഡാറ്റ പ്രോസസിങ് എന്നു പറയുന്നു. അതുകൊണ്ടു തന്നെ ഡാറ്റ പ്രോസസിങ്ങിന്റെ അനന്തര ഫലമാണ് വിവരം എന്ന് നമുക്ക് പറയാം.

പ്രോസസിങ്ങിനായി ഡാറ്റ നൽകിയിരിക്കുന്നതും പ്രോസസിംഗിനുശേഷം വിവരം ലഭിക്കുന്നതും ചിത്രം 1.4ൽ കാണിച്ചിരിക്കുന്നു. മറ്റൊരു വിധത്തിൽ പറഞ്ഞാൽ ഡാറ്റ പ്രോസസിന്റെ ഇൻപുട്ടും വിവരം പ്രോസസിൽ നിന്നുള്ള ഔട്ട്പുട്ടുമാണ്.

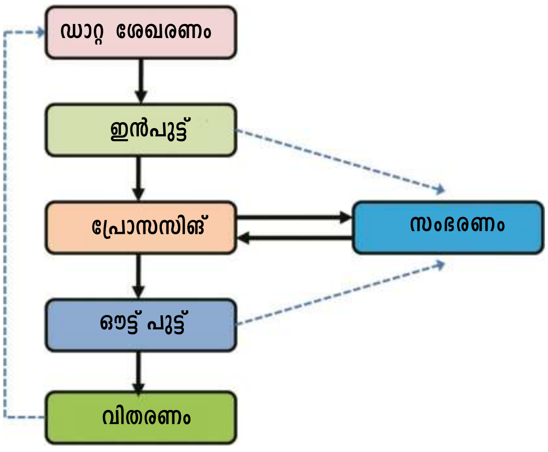
കേരളത്തിലെ ഹയർസെക്കന്ററി കോഴ്സുകളിലേക്കുള്ള ഏകജാലക പ്രവേശനരീതി പരിഗണിക്കുക. അതിന്റെ പ്രവർത്തനങ്ങൾ ചുവടെ കൊടുത്തിരിക്കുന്നതു പോലെ നമുക്ക് ചുരുക്കി പറയാം.

1. അപേക്ഷകരിൽ നിന്ന് പ്രിൻസിപ്പാൾ അപേക്ഷാഫോറത്തിലൂടെ ഡാറ്റ ശേഖരിക്കുന്നു. അതിനോടൊപ്പം ആവശ്യമായ വിശദാംശങ്ങൾക്ക് 10-ാം ക്ലാസ് പരീക്ഷയുടെ സ്കോർഷീറ്റ് നൽകുന്നു. ഈ സന്ദർഭത്തിൽ സ്കോർഷീറ്റിലെ വസ്തുതകളും കണക്കുകളും ഡാറ്റയായി മാറുന്നു എന്ന് ശ്രദ്ധിക്കേണ്ടതാണ്.
2. ശേഖരിച്ച ഡാറ്റ കമ്പ്യൂട്ടറിലേക്ക് നൽകുന്നു.
3. നൽകിയ ഡാറ്റ സംഭരിച്ചുവയ്ക്കുകയും പിന്നീട് പ്രോസസിംഗിനായി തിരികെയെടുക്കുകയും ചെയ്യുന്നു.
4. കണക്ക്കൂട്ടലുകൾ, താരതമ്യങ്ങൾ, ഇനംതിരിക്കൽ, ക്രമീകരിക്കൽ, വേർതിരിക്കൽ തുടങ്ങിയ പ്രവർത്തനങ്ങൾക്ക് കമ്പ്യൂട്ടറിലെ ഡാറ്റ ഉപയോഗിക്കുന്നു.
5. വിദ്യാർഥികൾക്കുള്ള അലോട്ട്മെന്റ് സ്ലിപ്പുകളും സ്കൂളുകൾക്കുള്ള അലോട്ട്മെന്റ് പട്ടികകളും തയാറാക്കുന്നു. ഈ സ്ലിപ്പുകളും പട്ടികകളും പ്രിന്റ് ചെയ്യുകയോ പിന്നീട് മറ്റ് ഉപയോഗങ്ങൾക്കായി സംഭരിക്കുകയോ ചെയ്യാം. മറ്റു സന്ദർഭങ്ങളിൽ വിവരം ലഭ്യമാക്കാൻ ഡാറ്റയായി ഇത് ഉപയോഗിക്കാം.
6. അലോട്ട്മെന്റ് സ്ലിപ്പുകൾ അപേക്ഷകർക്ക് വിതരണം ചെയ്യുകയും അലോട്ട്മെന്റ് പട്ടികകൾ സ്കൂളിലേക്ക് അയച്ചു കൊടുക്കുകയും ചെയ്യുന്നു.

അപ്രകാരം താഴെപ്പറയുന്ന 6 വിവിധ ഘട്ടങ്ങളിലൂടെയാണ് ഡാറ്റ പ്രോസസിംഗ് കടന്നുപോകുന്നതെന്ന് ഇത് വ്യക്തമാകുന്നു.

(എ) ഡാറ്റ ശേഖരണം.
(ബി) ഡാറ്റ ഇൻപുട്ട് ചെയ്യുന്നു.
(സി) ഡാറ്റ സംഭരിക്കുന്നു.
(ഡി) ഡാറ്റ പ്രോസസിങ്/ കൈകാര്യം ചെയ്യൽ.
(ഇ) വിവരം ഔട്ട്പുട്ട് ചെയ്യുന്നു.
(എഫ്) വിവരം വിതരണം ചെയ്യുന്നു.

ചിത്രം 1.5 ലെ കട്ടിയായ അമ്പടയാളം ഡാറ്റ പ്രോസസിങ്ങിലെ ഒഴുക്കിനെ



ചിത്രം 1.5: ഡാറ്റ പ്രോസസിംഗിന്റെ വിവധ ഘട്ടങ്ങൾ

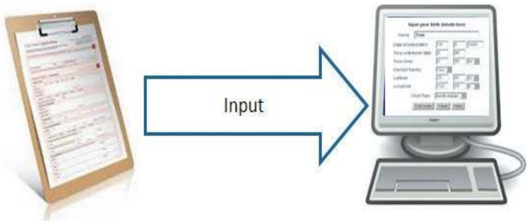
സൂചിപ്പിക്കുന്നു. ആവശ്യമെങ്കിൽ മാത്രം നടക്കുന്ന പ്രവൃത്തികൾ കുത്തിട്ടവരകൾ കൊണ്ട് സൂചിപ്പിക്കുന്നു. ഡാറ്റാ പ്രോസസിംഗിന്റെ വിവിധ ഘട്ടങ്ങൾ നമുക്ക് സൂക്ഷ്മമായി പരിശോധിക്കാം.

എ. ഡാറ്റ ശേഖരണം (Data Capturing)

ഹയർ സെക്കന്ററി പ്രവേശനത്തിന് നാം അപേക്ഷിക്കുമ്പോൾ സാധാരണയായി നിർദിഷ്ട അപേക്ഷ ഫോറത്തിലൂടെ വിശദാംശങ്ങൾ നൽകാറുണ്ട്. വാസ്തവത്തിൽ അഡ്മിഷൻ പ്രവർത്തനങ്ങൾക്കു വേണ്ടി ആവശ്യമായ ഡാറ്റകൾ ശേഖരിക്കുകയാണ് ഇതിലൂടെ പ്രിൻസിപ്പാൾ ചെയ്യുന്നത്. ഡാറ്റ പ്രോസസിംഗിന്റെ ആദ്യഘട്ടമാണിത്. ഉറവിട പ്രമാണം (Source document) എന്നറിയപ്പെടുന്ന ഈ ഫോറം രൂപകൽപന ചെയ്തിരിക്കുന്നത് ഉചിതമായ ഡാറ്റ അനുയോജ്യമായ ക്രമത്തിലും രൂപത്തിലും രേഖപ്പെടുത്താനുതകും വിധമാണ്. ഇപ്രകാരം ഉറവിട പ്രമാണത്തിന്റെ ഹാർഡ് കോപ്പി തയാറാക്കലും ഡാറ്റ ശേഖരണവുമാണ് ഈ ഘട്ടത്തിൽ നടക്കുന്ന പ്രവർത്തനങ്ങൾ. ഡാറ്റ ശേഖരണത്തിന് ഇപ്പോൾ നിർദിഷ്ട അപേക്ഷാഫോറങ്ങളുടെ ഹാർഡ് കോപ്പി ഉപയോഗിക്കുന്നില്ല പകരം ഓൺലൈൻ സൗകര്യത്തിലൂടെ ഡാറ്റ നേരിട്ട് രേഖപ്പെടുത്തുകയാണ് ചെയ്യുന്നത്.

ബി. ഇൻപുട്ട് (Input)

പ്രവേശനം തേടുന്ന സമയത്ത് നാം പൂരിപ്പിച്ച അപേക്ഷാഫോറം സ്കൂളിൽ സമർപ്പിക്കുന്നു. അതിൽനിന്ന് ഡാറ്റ വേർതിരിച്ച് കമ്പ്യൂട്ടറിലേക്ക് നൽകുന്നു. ചില അവസരങ്ങളിൽ ഈ ഡാറ്റയെ കമ്പ്യൂട്ടറിലേക്ക് നേരിട്ടും നൽകാറുണ്ട്. പ്രോസസിങ്ങിനായി കമ്പ്യൂട്ടറിലേക്ക് ഡാറ്റ നൽകുന്നതിനെ ഇൻപുട്ട് എന്ന് അറിയപ്പെടുന്നു. ഇൻപുട്ടായി നൽകിയ ഡാറ്റ സാധാരണയായി പ്രോസസിങ്ങിനു മുമ്പ് കമ്പ്യൂട്ടറിൽ സംഭരിക്കുന്നു



സി. സംഭരണം (Storage)

പല അവസരങ്ങളിലും കമ്പ്യൂട്ടറിലേക്ക് നൽകപ്പെടുന്ന ഡാറ്റയുടെ അളവ് വളരെ കൂടുതലായിരിക്കും. കൂടാതെ ഒറ്റഘട്ടമായോ ഒരു ദിവസം കൊണ്ടോ ഡാറ്റ നൽകൽ പൂർത്തിയാകണമെന്നില്ല. പ്രവേശന കാര്യത്തിൽ ലക്ഷക്കണക്കിന് അപേക്ഷകരുടെ ഡാറ്റയാണ് ഇൻപുട്ടായി കമ്പ്യൂട്ടറിലേക്ക് നൽകുന്നത്. ഡാറ്റ ഇൻപുട്ട് ചെയ്യുന്നത് പൂർത്തിയാക്കുവാൻ സാധാരണയായി കുറച്ച് ആഴ്ചകൾ എടുക്കാറുണ്ട്. അതുകൊണ്ട് വിവിധ സമയങ്ങളിൽ ഇൻപുട്ട് ചെയ്യുന്ന ഡാറ്റ തദവസരത്തിൽ തന്നെ സംഭരിക്കേണ്ടതായി വരുന്നു. മുഴുവൻ ഡാറ്റയും സംഭരിച്ചതിനുശേഷം മാത്രമേ പ്രോസസിങ് ആരംഭിക്കുകയുള്ളൂ. പ്രോസസിംഗിന്റെ ഫലമായി ലഭിക്കുന്ന വിവരവും കൂടി കമ്പ്യൂട്ടറിൽ സംഭരിക്കപ്പെടുന്നു. ഇങ്ങനെ സംഭരിക്കപ്പെട്ട ഡാറ്റയും വിവരവും ഭാവിയിൽ വിവിധ ആവശ്യങ്ങൾക്കായി ഉപയോഗിക്കാവുന്നതാണ്.

ഡി. പ്രോസസ് (Process)

കമ്പ്യൂട്ടറിൽ സംഭരിക്കപ്പെട്ട ഡാറ്റ പ്രോസസിങ്ങിനായി തിരികെ എടുക്കുന്നു. പ്രോസസിങ്ങിന്റെ ഭാഗമായി ഗണിത ക്രിയകൾ, തരംതിരിക്കൽ താരതമ്യം, ക്രമീകരിക്കൽ, വേർതിരിക്കൽ, സംഗ്രഹിക്കൽ തുടങ്ങിയ പ്രവർത്തനങ്ങൾ നടക്കുന്നു. ഹയർ സെക്കന്ററി കോഴ്സിന്റെ പ്രവേശനകാര്യത്തിൽ ഓരോ അപേക്ഷകന്റെയും WGPA (Weighted Grade Point Average) കണക്കാക്കുന്നു. പിന്നീട് WGPA യുടെ അവരോഹണ ക്രമത്തിൽ അപേക്ഷകരെ വിവിധ വിഭാഗങ്ങളിലാക്കി പട്ടികപ്പെടുത്തുന്നു. ഇവിടെ തിരഞ്ഞെടുത്ത സ്കൂൾ, കോഴ്സ്, പാഠ്യേതര പ്രവർത്തനങ്ങളിലെ മികവ് എല്ലാം പരിഗണിക്കപ്പെടുന്നു. അവസാനം സ്കൂളുകളിലേക്കുള്ള അലോട്ട്മെന്റ് പട്ടികയും അപേക്ഷകർക്കുള്ള അലോട്ട്മെന്റ് സ്ലിപ്പുകളും തയാറാക്കുന്നു.

ഇ. ഔട്ട്പുട്ട് (Output)

പ്രോസസിങ്ങിനു ശേഷമുള്ള വിവരം ഈ ഘട്ടത്തിൽ ലഭ്യമാകുന്നു. ഗുണഭോക്താവിന് ഉചിതമായ തീരുമാനങ്ങൾ എടുക്കുന്നതിനോ പ്രശ്നപരിഹാരത്തിനോ ഉതകുന്ന രീതിയിലോ ആയിരിക്കണം ഔട്ട് പുട്ട് ഘട്ടത്തിൽ വിവരം നൽകേണ്ടത്. ഹയർ സെക്കന്ററി പ്രവേശനത്തിന്റെ ഭാഗമായി അപേക്ഷകനുള്ള അലോട്ട്മെന്റ് സ്ലിപ്പും സ്കൂളുകൾക്കുള്ള അലോട്ടമെന്റ് പട്ടികയും ആവശ്യമായ മാതൃകയിൽ ഔട്ട് പുട്ടായി തയാറാക്കുന്നു.

എഫ്. വിവരത്തിന്റെ വിതരണം (Distribution of Information)

ഔട്ട്പുട്ട് ഘട്ടത്തിൽ ലഭിച്ച വിവരം ഗുണഭോക്താക്കൾക്ക് വിതരണം ചെയ്യുന്നു. വിവരത്തിനനുസരിച്ച് അവർ തീരുമാനങ്ങൾ എടുക്കുകയോ പ്രശ്നങ്ങൾ പരിഹരിക്കുകയോ ചെയ്യുന്നു. ഉദാഹരണമായി ഹയർസെക്കന്ററി സ്കൂൾ പ്രവേശനത്തിനുള്ള അലോട്ട്മെന്റ് സ്ലിപ്പ് അപേക്ഷകർക്ക് അനുവദിച്ച സ്കൂളിൽ ചേരുന്നതിനും സ്കൂളുകൾക്ക് യോഗ്യരായ അപേക്ഷകരെ പ്രവേശിപ്പിക്കുന്നതിനുമായി അലോട്ട്മെന്റ് പട്ടിക വിതരണം ചെയ്യുന്നു. അഡ്മിഷൻ രജിസ്റ്ററോ (admission register), ഹാജർ പട്ടികയോ (Class Register) തയാറാക്കുന്നതിന് അലോട്ട്മെന്റ് സ്ലിപ്പുകൾ ഉപയോഗിക്കാവുന്നതാണ്. പൊതുപരീക്ഷക്ക് വിദ്യാർഥികളെ രജിസ്റ്റർ ചെയ്യുന്നതിനുള്ള നോമിനൽ റോൾ (Nominal Roll) തയ്യാറാക്കുന്നതിനായി അലോട്ട്മെന്റ് പട്ടികകൾ ഉപയോഗിക്കാവുന്നതാണ്.

നമുക്കു ചെയ്യാം

- താഴെ കൊടുത്തിരിക്കുന്നവയിലെ ഡാറ്റ പ്രോസസിംഗ് പ്രവർത്തനങ്ങൾ തിരിച്ചറിഞ്ഞ് എഴുതുക. (i) ബാങ്കിൽ ഒരു അക്കൗണ്ട് തുടങ്ങുന്നു, (ii) സ്കോളർഷിപ്പുകൾക്ക് അപേക്ഷിക്കുന്നു.
- നിത്യജീവിതത്തിൽ ഏതെങ്കിലും സാഹചര്യങ്ങളിലെ പ്രപവർത്തനങ്ങളിൽ ഡാറ്റ പ്രോസസിങ് പ്രവർത്തനം തിരിച്ചറിഞ്ഞ് അതിൽ ഓരോ ഘട്ടത്തിലും അനുവർത്തിക്കുന്ന പ്രവർത്തനങ്ങൾ എഴുതുക

സ്വയം വിലയിരുത്താം

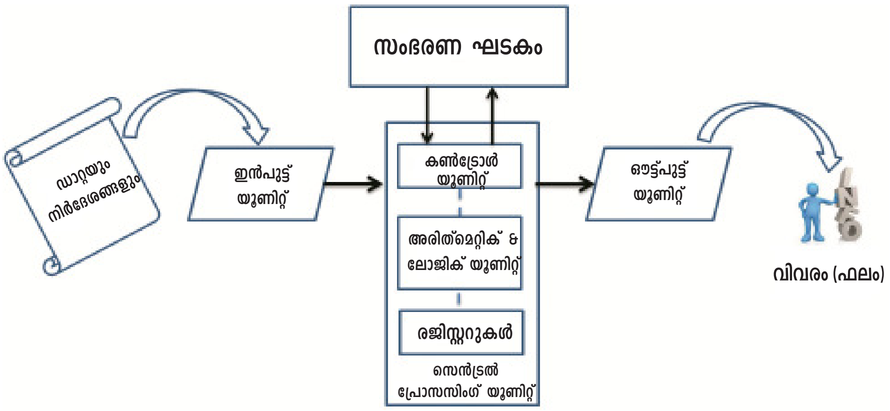
1. അസംസ്കൃത വസ്തുതകളെയും കണക്കുകളെയും .... എന്നറിയപ്പെടുന്നു.
2. പ്രോസസ് ചെയ്ത ഡാറ്റയെ.....................എന്നറിയപ്പെടുന്നു.
3. താഴെ പറയുന്നവയിൽ ഏതാണ് തീരുമാനങ്ങളെടുക്കുവാൻ നമ്മെ സഹായിക്കുന്നത് ?
   എ). ഡാറ്റ ബി).വിവരം സി). അറിവ്. ഡി).ബുദ്ധി.
4. വിവരം ലഭിക്കുന്നതിന് ഡാറ്റയെ കൈകാര്യം ചെയ്യുന്നതിനെ .... എന്നറിയപ്പെടുന്നു.
5. താഴെ കൊടുത്തിരിക്കുന്നവയെ ശരിയായ രീതിയിൽ ക്രമപ്പെടുത്തുക.
   പ്രോസസ്, ഔട്ട്പുട്ട്, സംഭരണം, വിതരണം, ഡാറ്റാശേഖരണം, ഇൻപുട്ട്.
6. കൂട്ടത്തിൽപ്പെടാത്തത് കണ്ടെത്തുക, കാരണം നൽകുക
   എ).കണക്ക്കൂട്ടൽ ബി). സംഭരണം സി). താരതമ്യം ഡി). ഇനംതിരിക്കൽ
7. നാം വിവരം സംഭരിക്കുന്നത് എന്തിനാണ്?
8. വിവരം ഒരു ഡാറ്റയായി പ്രവർത്തിക്കാം. ശരിയോ തെറ്റോ എന്ന് പ്രസ്താവിക്കുക.
9. ഡാറ്റ പ്രോസസിംഗിന്റെ അവസാന ഘട്ടമേത്?
10. ഉറവിട പ്രമാണം എന്നാൽ എന്ത്?

1.3 കമ്പ്യൂട്ടറിന്റെ പ്രവർത്തന ഘടകങ്ങൾ (Functional units of a computer)

കമ്പ്യൂട്ടറുകൾ അവയുടെ വലുപ്പത്തിലും ആകൃതിയിലും പ്രവർത്തനമികവിലും വിലയിലും വ്യത്യസ്തമാണെങ്കിലും അവയുടെ അടിസ്ഥാനപരമായ ഘടന ഒരേ പോലെയാണ്. ഗണിതജ്ഞനും കമ്പ്യൂട്ടർ ശാസ്ത്രജ്ഞനുമായ ജോൺവോൺ ന്യൂമൻ നിർദേശിച്ച മാതൃകയെ അടിസ്ഥാനപ്പെടുത്തിയാണ് ഇതിന്റെ ഘടന. ഇൻപുട്ട് യൂണിറ്റ്, സെൻട്രൽ പ്രോസസിങ് യൂണിറ്റ് (CPU), സംഭരണ യൂണിറ്റ് (Storage Unit), ഔട്ട്പുട്ട് യൂണിറ്റ് എന്നിവയാണ് ഇതിലടങ്ങിയിട്ടുള്ള പ്രവർത്തന ഘടകങ്ങൾ. ഇവയിൽ ഓരോ ഘടകത്തിനും പ്രത്യേക ദൗത്യമാണ് നിർവഹിക്കാനുള്ളത്.

ചിത്രം 1.6 ജോൺ വോൺ ന്യൂമാൻ (1903 - 1957)

ഈ ഘടകങ്ങളുടെ പ്രവർത്തനങ്ങൾ നമുക്കിവിടെ ചർച്ച ചെയ്യാം. കമ്പ്യൂട്ടറിന്റെ അടിസ്ഥാന ഘടന ചിത്രം 1.7 ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 1.7: കമ്പ്യൂട്ടറിന്റെ അടിസ്ഥാന ഘടന

1. ഇൻപുട്ട് യൂണിറ്റ്

ശേഖരിക്കപ്പെട്ടിട്ടുള്ള ഡാറ്റയും നിർദേശങ്ങളും പ്രോസസിംഗിനായി കമ്പ്യൂട്ടറിലേക്ക് നൽകുന്നത് ഇൻപുട്ട് യൂണിറ്റിലൂടെയാണ്. അവ മെമ്മറിയിൽ സംഭരിക്കുന്നു (സംഭരണ ഘടകം). അക്കങ്ങൾ, അക്ഷരങ്ങൾ, ചിത്രങ്ങൾ, ശബ്ദം, വീഡിയോ മുതലായ വ്യത്യസ്ത രൂപങ്ങളിലായിരിക്കും ഡാറ്റ. ഡാറ്റ ഇൻപുട്ട് ചെയ്യുന്നതിനായി അതിന്റെ സ്വഭാവമനുസരിച്ച് പലതരത്തിലുള്ള ഇൻപുട്ട് ഉപകരണങ്ങൾ ലഭ്യമാണ്. സാധാരണയായി ഉപയോഗിക്കുന്ന ഇൻപുട്ട് ഉപകരണങ്ങൾ കീബോർഡ്, മൗസ്, സ്കാനർ, മൈക്ക്, ഡിജിറ്റൽ ക്യാമറ മുതലായവയാണ്. ചുരുക്കത്തിൽ ഇൻപുട്ട് യൂണിറ്റ് ചെയ്യുന്ന പ്രവർത്തനങ്ങൾ താഴെ കൊടുത്തിരിക്കുന്നവയാണ്.

1. നിർദേശങ്ങളും ഡാറ്റയും പുറമേ നിന്നും സ്വീകരിക്കുക.
2. ഈ നിർദേശങ്ങളും ഡാറ്റയും കമ്പ്യൂട്ടറിലേക്ക് സ്വീകരിക്കാവുന്ന രീതിയിലേക്ക് മാറ്റുക.
3. മാറ്റിയ നിർദേശങ്ങളും ഡാറ്റയും പ്രോസസിംഗിനായി കമ്പ്യൂട്ടറിലേക്ക് നൽകുക.

2. സെൻട്രൽ പ്രോസസിംഗ് യൂണിറ്റ് (Central Processing Unit) (CPU)

കമ്പ്യൂട്ടറിന്റെ മസ്തിഷ്കമാണ് സി പി യു. മനുഷ്യൻ പ്രധാന തീരുമാനങ്ങൾ എടുക്കുന്നതും ശരീരത്തിന്റെ മറ്റു ഭാഗങ്ങളുടെ പ്രവർത്തനങ്ങൾ നിയന്ത്രിക്കുന്നതും മസ്തിഷ്കം നിർദേശിക്കുന്നതുപോലെയാണ്. അതുപോലെ ഒരു കമ്പ്യൂട്ടർ സംവിധാനത്തിൽ എല്ലാ പ്രധാന ഗണിത ക്രിയകളും താരതമ്യങ്ങളും നടത്തുന്നത് സി പി യു വിലാണ്. കമ്പ്യൂട്ടറിലെ മറ്റ് ഘടകങ്ങളുടെ പ്രവർത്തനങ്ങൾ നിയന്ത്രിക്കുന്നതിനും പ്രവർത്തനക്ഷമമാക്കുന്നതിനുമുള്ള ചുമതലയും ഇതിനുണ്ട്. സി പി യു വിന്റെ പ്രവർത്തനങ്ങൾ നിർവഹിക്കുന്നത് അതിലെ മൂന്ന് ഭാഗങ്ങളാണ് - അരിത്മെറ്റിക് & ലോജിക്ക് യൂണിറ്റ് (Arithmetic & Logic Unit) (ALU), കൺട്രോൾ യൂണിറ്റ് (Cotnrol Unit) (CU), രജിസ്റ്ററുകൾ (Registers).

എ. അരിത്മെറ്റിക് & ലോജിക്ക് യൂണിറ്റ് (ALU)

നിർദേശങ്ങളിൽ സൂചിപ്പിച്ചിട്ടുള്ള യഥാർഥ പ്രവർത്തനങ്ങൾ നിർവഹിക്കപ്പെടുന്നത് അരിത്മെറ്റിക് & ലോജിക് യൂണിറ്റിലാണ് (ALU). ഗണിതക്രിയകളും, താരതമ്യം ചെയ്യൽ, തീരുമാനമെടുക്കൽ എന്നീ യുക്തി സഹമായ പ്രവർത്തനങ്ങളും നടത്തുന്നത് ഇതാണ്. ഡാറ്റയും നിർദേശങ്ങളും സംഭരണ ഘടകത്തിൽ സൂക്ഷിക്കുകയും ALU വിലേക്ക് കൈമാറുകയും ചെയ്യുന്നു. അതിനുശേഷം ALU വിൽ പ്രോസസിംഗ് നടക്കുന്നു. ALU വിൽ ഉണ്ടായ ഇടക്കാല ഫലങ്ങൾ സംഭരണ ഘടകത്തിലേക്ക് കൈമാറുകയും പിന്നീട് പ്രോസസിംഗിന് ആവശ്യമാകുമ്പോൾ അവ തിരിച്ചെടുക്കുകയും ചെയ്യുന്നു. ഇപ്രകാരം മുഴുവൻ പ്രവർത്തനങ്ങളും പൂർത്തിയാകുന്നതിനു മുൻപ് സംഭരണ ഘടകത്തിനും ALU വിനുമിടയ്ക്ക് പല തവണ ഡാറ്റയുടെ ഒഴുക്ക് ഉണ്ടാകുന്നു.

ബി. കൺട്രോൾ യൂണിറ്റ് (CU)

കമ്പ്യൂട്ടറിലെ ഓരോ ഘടകത്തിനും അതിന്റേതായ പ്രവർത്തനമുണ്ട്. എന്നാൽ ഈ ഘടകങ്ങൾ വ്യക്തമായ നിർദേശങ്ങൾക്ക് അനുസൃതമായേ പ്രവർത്തിക്കൂ. ഇത്തരം നിർദേശങ്ങൾ പുറപ്പെടുവിക്കുന്നത് കൺട്രോൾ യൂണിറ്റ് (CU) ആണ്. ഇത് മറ്റ് ഘടകങ്ങളോട് ബന്ധപ്പെട്ട പ്രവർത്തനങ്ങൾ ഏറ്റെടുക്കാൻ നിർദേശിക്കുന്നു. ഇത് കമ്പ്യൂട്ടറിന്റെ മറ്റെല്ലാ ഘടകങ്ങളെയും പരസ്പരം കൂട്ടിയണക്കുകയും നിയന്ത്രിക്കുകയും ചെയ്യുന്ന കേന്ദ്ര നാഡീവ്യവസ്ഥയാണ്. മെമ്മറിയിൽ സംഭരിച്ചിട്ടുള്ള പ്രോഗ്രാമുകളിൽ (Programmes) നിന്ന് ഇത് നിർദേശങ്ങൾ സ്വീകരിക്കുകയും അവയിലടങ്ങിയിട്ടുള്ള പ്രവർത്തനങ്ങളെ വ്യാഖ്യാനിക്കുകയും ബന്ധപ്പെട്ട ഘടകങ്ങൾക്ക് അവ നിർവഹിക്കുന്നതിനു വേണ്ട സൂചനകൾ നൽകയും ചെയ്യുന്നു.

സി. രജിസ്റ്ററുകൾ (Registers)

C P U വിന്റെ പ്രവർത്തനങ്ങൾ ത്വരിതപ്പെടുത്തുന്നതിനുള്ള താൽക്കാലിക സംഭരണ ഘടകങ്ങളാണിവ. ഡാറ്റ, നിർദേശം, മെമ്മറി അഡ്രസ്, ഫലങ്ങൾ മുതലായവ സംഭരിക്കുന്നതിനായി വ്യത്യസ്ത തരം രജിസ്റ്ററുകൾ രൂപകല്പന ചെയ്തിട്ടുണ്ട്.

3. സംഭരണ ഘടകം (Storage Unit)

യഥാർഥ പ്രോസസിംഗ് ആരംഭിക്കും മുൻപ് ഇൻപുട്ട് ഘടകത്തിലൂടെ കമ്പ്യൂട്ടറിലേക്ക് നൽകിയ ഡാറ്റയും നിർദേശങ്ങളും കമ്പ്യൂട്ടറിനുള്ളിൽ സംഭരിച്ചു വയ്ക്കുന്നു. അതുപോലെ പ്രോസസിംഗിനുശേഷം ഉൽപാദിപ്പിക്കുന്ന വിവരങ്ങളും ഫലങ്ങളും കൂടി ഔട്ട്പുട്ട് ഘടകത്തിലേയ്ക്ക് നൽകുന്നതിനു മുൻപായി കമ്പ്യൂട്ടറിനുള്ളിൽ സംഭരിക്കുന്നു. കൂടാതെ ഇടക്കാല ഫലങ്ങൾ (Intermediate Results) എന്തെങ്കിലുമുണ്ടെങ്കിൽ പിന്നീടുള്ള പ്രോസസിംഗിനായി അവയും സംഭരിക്കുന്നു. കമ്പ്യൂട്ടറിലെ സംഭരണ ഘടകത്തെ ഇങ്ങനെയുള്ള എല്ലാ ലക്ഷ്യങ്ങൾക്കും പ്രയോജനപ്പെടുത്തുന്നു. ചുരുക്കത്തിൽ കമ്പ്യൂട്ടറിലെ സംഭരണഘടകം താഴെ പറയുന്നവ സൂക്ഷിക്കുന്നതിനായി ഉപയോഗിക്കുന്നു.

1. പ്രോസസിംഗിന് ആവശ്യമായ ഡാറ്റയും നിർദേശങ്ങളും
2. നടന്നു കൊണ്ടിരിക്കുന്ന പ്രവർത്തനങ്ങൾക്കുവേണ്ട ഇടക്കാല ഫലങ്ങൾ (Intermediate Results)

3. ഔട്ട്പുട്ട് യൂണിറ്റിലേക്ക് നൽകുന്നതിനു മുൻപുള്ള പ്രോസസിംഗിന്റെ അവസാന ഫലങ്ങൾ.

സംഭരണ ഘടകം രണ്ട് തരമുണ്ട് പ്രാഥമിക സംഭരണം, ദ്വിതീയ സംഭരണം.

പ്രാഥമിക സംഭരണം (Primary Storage)

ഇത് പ്രധാന മെമ്മറി എന്നും അറിയപ്പെടുന്നു. ഇതിനെ റാൻഡം ആക്സസ് മെമ്മറി (RAM) എന്നും റീഡ് ഓൺലി മെമ്മറി (ROM) എന്ന് വീണ്ടും രണ്ടായി തിരിച്ചിരിക്കുന്നു. ഡാറ്റയും നിർദേശങ്ങളും ഡാറ്റാ പ്രോസസിംഗിന്റെ ഇടക്കാല ഫലങ്ങളും റാം (RAM) സൂക്ഷിക്കുന്നു. കമ്പ്യൂട്ടർ തൊട്ടു മുൻപ് ചെയ്ത പ്രവൃത്തിയുടെ ഫലങ്ങളും ഇത് സൂക്ഷിക്കുന്നു. കമ്പ്യൂട്ടറിന്റെ ആരംഭ പ്രവർത്തനങ്ങൾക്കുള്ള നിർദേശങ്ങളാണ് റോമിൽ (ROM) അടങ്ങിയിരിക്കുന്നത്. സെൻട്രൽ പ്രോസസിംഗ് യൂണിറ്റിന് (CPU) പ്രധാന മെമ്മറിയെ വളരെ വേഗത്തിൽ നേരിട്ട് ഉപയോഗിക്കാം. എന്നാൽ പ്രാഥമിക സംഭരണഘടകം വില കൂടിയതും പരിമിതമായ സംഭരണ ശേഷിയുള്ളതുമാകുന്നു.

ദ്വിതീയ സംഭരണം (Secondary Storage)

ഇത് സഹായക (auxilary) സംഭരണ ഘടകം എന്നും അറിയപ്പെടുന്നു. സെക്കന്ററി സംഭരണ ഘടകത്തിന് പ്രാഥമിക സംഭരണ ഘടകത്തിന്റെ ന്യൂനതകൾ പരിഹരിക്കാനാകും. വലിയ സംഭരണശേഷിയുള്ള ഇവയിൽ ഡാറ്റ, പ്രോഗ്രാമുകൾ, വിവരങ്ങൾ എന്നിവ സ്ഥിരമായി സൂക്ഷിച്ചു വയ്ക്കുന്നു. പക്ഷേ നാം ഇതിനായി പ്രത്യേക നിർദേശങ്ങൾ നൽകേണ്ടതുണ്ട്. ഹാർഡ്ഡിസ്ക്, സി ഡി, ഡി വി ഡി, മെമ്മറി സ്റ്റിക് മുതലായവ ദ്വിതീയ സംഭരണ ഘടകത്തിന് ചില ഉദാഹരണങ്ങളാണ്.

4 ഔട്ട്പുട്ട് ഘടകം (Output Unit)

ഡാറ്റ പ്രോസസിംഗിനു ശേഷം ലഭിക്കുന്ന വിവരങ്ങൾ മനുഷ്യനു വായിക്കുവാൻ കഴിയുന്ന രൂപത്തിൽ പുറം ലോകത്തിലേക്ക് ഔട്ട്പുട്ട് ഘടകത്തിലൂടെ നൽകുന്നു. മോണിറ്ററും, പ്രിന്ററുമാണ് സാധാരണയായി ഉപയോഗിക്കുന്ന ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ. ഔട്ട്പുട്ട് ഘടകം നടത്തുന്ന പ്രവർത്തനങ്ങൾ താഴെ പറയുന്ന രീതിയിൽ സംഗ്രഹിക്കാം.

1. സി പി യു ഉൽപ്പാദിപ്പിക്കുന്ന ഫലങ്ങൾ കോഡ് രൂപത്തിൽ സ്വീകരിക്കുന്നു.
2. കോഡ് രൂപത്തിലുള്ള ഫലങ്ങൾ മനുഷ്യന് വായിക്കുവാൻ കഴിയുന്ന രൂപത്തിലേക്കു മാറ്റുന്നു.
3. ഫലങ്ങൾ പുറം ലോകത്തിനു നൽകുന്നു.

1.4 കമ്പ്യൂട്ടർ ഡാറ്റ പ്രോസസർ എന്ന നിലയിൽ

ഡാറ്റ പ്രോസസിംഗും അതിന്റെ വിവിധ ഘട്ടങ്ങളിലുള്ള പ്രവർത്തനങ്ങളും നാം കണ്ടു വല്ലോ. ഈ ഘട്ടങ്ങളിലും പ്രവർത്തനങ്ങളിലും മനുഷ്യരാണ് ഏർപ്പെടുന്നത് എന്ന് സങ്കല്പിക്കൂ, എല്ലായ്പ്പോഴും തെറ്റ് കൂടാതെയും കൃത്യസമയത്തും വിവരങ്ങൾ ലഭിക്കില്ല എന്ന് നമുക്ക് ഉറപ്പാണ്. കൃത്യവും സമഗ്രവും വിശ്വസനീയവും അതേസമയം യോജിച്ച ഘടനയിലും മാധ്യമത്തിലും ഉള്ള അറിവ് നമുക്ക് എല്ലായ്പ്പോഴും ആവശ്യമാണ്. എങ്കിലേ അറിവ് വ്യക്തമായി പ്രകടിപ്പിക്കാനുള്ള സന്ദർഭത്തിൽ വിവരം പ്രായോഗികമാക്കാൻ കഴിയൂ. എങ്കിൽ മാത്രമേ ബുദ്ധി ഉപയോഗിച്ച് പ്രശ്നങ്ങൾ പരിഹരിക്കാനും തീരുമാ

നങ്ങൾ എടുക്കുവാനും കഴിയൂ. നമ്മൾ ഇതു വരെ നടത്തിയ ചർച്ചകളുടെ അടിസ്ഥാനത്തിൽ കമ്പ്യൂട്ടറിനെ ഏറ്റവും നല്ല ഡാറ്റ പ്രോസസിംഗ് യന്ത്രമായി കരുതാം. ചുരുക്കത്തിൽ ഡാറ്റയും നിർദേശങ്ങളും സ്വീകരിക്കുകയും, ആ നിർദേശങ്ങൾ പ്രകാരം ഡാറ്റയിൽ ഗണിത ക്രിയകൾ, യുക്തി പരമായ പ്രവർത്തനങ്ങൾ എന്നിവ നടത്തുകയും അതിന്റെ ഫലങ്ങളോ വിവരങ്ങളോ പുറത്ത് വിടുകയും ചെയ്യുവാനായി രൂപകല്പന ചെയ്തിട്ടുള്ള ഇലക്ട്രോണിക് യന്ത്രമാണ് കമ്പ്യൂട്ടർ.

നമുക്കു ചെയ്യാം

ഡാറ്റ പ്രോസസിംഗുമായി ബന്ധപ്പെടുത്തി മനുഷ്യനെയും കമ്പ്യൂട്ടറിനെയും താരതമ്യം ചെയ്ത് താഴെ കൊടുത്തിരിക്കുന്ന പട്ടിക പൂരിപ്പിക്കുക. പ്രവർത്തനങ്ങളുടെ കാര്യത്തിൽ അവയവങ്ങളെ അല്ലെങ്കിൽ ഘടകങ്ങളെ എടുത്തുപറയാം. സ്വഭാവ വിശേഷങ്ങൾക്കു വേണ്ടി പ്രവർത്തന മികവ് സൂചകമാക്കാം. നിങ്ങൾക്ക് കൂടുതൽ സവിശേഷതകൾ കൂട്ടിച്ചേർക്കാം.

സവിശേഷതകൾ	മനുഷ്യൻ	കമ്പ്യൂട്ടർ
പ്രവർത്തനങ്ങൾ		
ഇൻപുട്ട്	*കണ്ണുകൾ,ചെവികൾ*	*കീബോർഡ്,മൗസ്*
ഔട്ട്പുട്ട്		
ഗണിത ക്രിയകളും താരതമ്യവും		
താൽക്കാലിക സംഭരണം		
സ്ഥിര സംഭരണം		
നിയന്ത്രണം		
സവിശേഷതകൾ		
വേഗത		
കൃത്യത		
വിശ്വാസ്യത		

1.4.1 കമ്പ്യൂട്ടറുകളുടെ സവിശേഷതകൾ (Characteristics of computers)

നമുക്കു ചെയ്യാം എന്ന താരതമ്യപട്ടിക പൂർത്തിയാക്കുന്നതിലൂടെ കമ്പ്യൂട്ടറിന്റെ ചില സ്വഭാവ സവിശേഷതകൾ നിങ്ങളൾ തിരിച്ചറിഞ്ഞിട്ടുണ്ടാകും. നമുക്കറിയാവുന്നതു പോലെ കമ്പ്യൂട്ടറുകൾക്ക് ഒരു സെക്കന്റിൽ ദശലക്ഷകക്കണക്കിന് നിർദേശങ്ങൾ നിർവഹിക്കുവാൻ കഴിയും. ഡാറ്റ പ്രോസസിംഗിനു ശേഷം ലഭിക്കുന്ന ഫലങ്ങൾ വളരെ കൃത്യമാണ്. പക്ഷെ കമ്പ്യൂട്ടറുകൾക്ക് ആ ഫലങ്ങൾ വിശകലനം ചെയ്യാൻ മതിയായ അറിവോ ബുദ്ധിയോ ഇല്ല. അവ അനുസരണയുള്ള ഒരു സേവകനെപ്പോലെ നിർദേശങ്ങൾ നിർവഹിക്കുക മാത്രമാണ് ചെയ്യുന്നത്. ശരിയായ ഡാറ്റയും നിർദേശങ്ങളും നൽകിയാൽ മാത്രമേ

കമ്പ്യൂട്ടറുകൾ കൃത്യമായ ഫലം നൽകൂ. ഈ വിശേഷ ഗുണത്തെ Garbage in Garbage out (GIGO) എന്ന പദം ഉപയോഗിച്ച് സൂചിപ്പിക്കാം. അതായത് തെറ്റായ ഇൻപുട്ട് കമ്പ്യൂട്ടറിനു നൽകിയാൽ തെറ്റായ ഔട്ട്പുട്ട് കമ്പ്യൂട്ടറും നൽകും. പട്ടിക 1.2 നോക്കി കമ്പ്യൂട്ടറിന്റെ മേന്മകളും പരിമിതികളും തിരിച്ചറിയുക.

കമ്പ്യൂട്ടറുകൾ	
മേന്മകൾ	പരിമിതികൾ
വേഗത: ഒരു സെക്കന്റ് അല്ലെങ്കിൽ അതിന്റെ ഒരംശം കൊണ്ട് ദശലക്ഷക്കണക്കിന് പ്രവർത്തനങ്ങൾ നിർവഹിക്കുവാൻ കമ്പ്യൂട്ടറുകൾക്ക് കഴിയും. ഒരു മനുഷ്യൻ മാസങ്ങളോ വർഷങ്ങളോ എടുത്ത് ചെയ്യുന്ന ജോലികൾ കമ്പ്യൂട്ടറിന് ഒരു മിനിട്ടു കൊണ്ട് ചെയ്യുവാൻ കഴിയും. **കൃത്യത**: കമ്പ്യൂട്ടറിന് ഗണിതക്രിയകൾ വളരെ ഉയർന്ന കൃത്യതയോടു കൂടി നിർവഹിക്കാൻ കഴിയും. ഫലങ്ങളിലും ഗണിത ക്രിയകളുടെ സൂക്ഷ്മതയിലും യാതൊരുവിധ തെറ്റുകളും ഉണ്ടാകില്ല എന്നുള്ളതാണ് കൃത്യത എന്നതു കൊണ്ട് ഉദ്ദേശിക്കുന്നത്. **സ്ഥിരോത്സാഹം**: കമ്പ്യൂട്ടർ ഒരു യന്ത്രമായതുകൊണ്ട് അതിന് മണിക്കൂറുകളോളം മുഷിയാതെ പ്രവർത്തിക്കാൻ കഴിയും. മനുഷ്യരിൽ നിന്നും വിത്യസ്തമായി അത് നമ്മോട് അനുസരണക്കേടോ മറ്റ് വികാരങ്ങളോ പ്രകടിപ്പിക്കില്ല. അതുകൊണ്ട് കമ്പ്യൂട്ടറുകൾ പതിവ് ജോലികൾക്ക് ഏറ്റവും യോജിച്ചതാണ്. **ബഹുമുഖ വൈദഗ്ധ്യം**: ധാരാളം വ്യത്യസ്ത തരത്തിലുള്ള പ്രോസസിംഗ് ദൗത്യങ്ങൾ നിർവഹിക്കാൻ കമ്പ്യൂട്ടറുകളെ ഉപയോഗിക്കാം. ഇത് പൊതു ഉപയോഗത്തിനുള്ള ഡാറ്റ പ്രോസസിംഗ് യന്ത്രമാണ്. **വളരെ വലിയ മെമ്മറി**: കമ്പ്യൂട്ടറുകൾക്ക് വർധിച്ച തോതിലുള്ള സംഭരണശേഷിയുണ്ട്. പ്രോസസിംഗിനായി വലിയ അളവിൽ ഡാറ്റ മെമ്മറിയിൽ സംഭരിക്കാനാകും. സംഭരണ അളവ് ആവശ്യത്തിനനുസരിച്ച് വർധിപ്പിക്കാനുമാകും.	**ഐ ക്യു (IQ) വിന്റെ അഭാവം**: കമ്പ്യൂട്ടറിന് അമാനുഷിക കഴിവുകൾ ഉണ്ടെന്നാണ് മിക്ക വരും കരുതുന്നത്. എന്നാൽ അത് വാസ്തവമല്ല. കമ്പ്യൂട്ടറിന് മനുഷ്യർക്കുള്ളതുപോലെ സ്വതസ്സിദ്ധമായ ബുദ്ധിയില്ല. **തീരുമാനമെടുക്കാനുള്ള കഴിവിന്റെ അഭാവം**: കമ്പ്യൂട്ടറുകൾക്ക് സ്വന്തം നിലയ്ക്ക് തീരുമാനങ്ങൾ എടുക്കാൻ കഴിയില്ല. മനുഷ്യർക്കുള്ളതുപോലെ അന്തർജ്ഞാനപരമായ കഴിവുകൾ അതിനില്ല.

പട്ടിക 1.2 കമ്പ്യൂട്ടറിന്റെ മെച്ചങ്ങളും പരിമിതികളും

സ്വയം വിലയിരുത്താം

1. ആധുനിക കമ്പ്യൂട്ടറിന്റെ മാതൃക മുന്നോട്ട് വച്ചതാര്?
2. സി പി യു (CPU) വിന്റെ ഘടകങ്ങളുടെ പേരെഴുതുക?
3. കമ്പ്യൂട്ടറിന്റെ അടിസ്ഥാന ഘടകങ്ങളിൽ ഏതാണ് ഡാറ്റ പ്രോസസിംഗ് പ്രവർത്തനങ്ങളിൽ നേരിട്ട് ഉൾപ്പെടാത്തത്?
4. ഒരു നിർദേശത്തിന്റെ നിർവഹണം എന്ന് ഉദ്ദേശിക്കുന്നതെന്താണ്?
5. കമ്പ്യൂട്ടറിന്റെ ഏതു ഭാഗമാണ് മനുഷ്യ മസ്തിഷ്കത്തോട് താരതമ്യപ്പെടുത്താവുന്നത്.

1.5 സംഖ്യാ സമ്പ്രദായം (Number systems)

എണ്ണുന്നതിനും, അടയാളപ്പെടുത്തുന്നതിനും, അളക്കുന്നതിനും ഉള്ള ഗണിതശാസ്ത്രപരമായ ഉപാധിയാണ് സംഖ്യ. ചിട്ടയോടെ സംഖ്യകളെ പ്രതിനിധാനം ചെയ്യുന്ന രീതിയാണ് സംഖ്യാന സമ്പ്രദായം. പത്ത് അക്കങ്ങൾ ഉപയോഗിച്ച് കൊണ്ടുള്ള ദശസംഖ്യാ സമ്പ്രദായമാണ് (Decimal Number System) നമ്മൾ നിത്യജീവിതത്തിൽ ഉപയോഗിച്ച് വരുന്നത്. 289 എന്ന സംഖ്യയെ ഇരുന്നൂറ്റി എൺപത്തി ഒൻപത് എന്നാണ് ഉച്ചരിക്കുന്നത്. ഇതിൽ 2, 8, 9 എന്നീ അക്കങ്ങൾ അടങ്ങിയിട്ടുണ്ട്. അതുപോലെ മറ്റ് സംഖ്യാന സമ്പ്രദായങ്ങളും നിലവിലുണ്ട്. ഓരോന്നിനും അതിന്റേതായ ചിഹ്നങ്ങളും രീതികളുമാണ് അവയിലെ സംഖ്യ രൂപകൽപന ചെയ്യുന്നതിന് ഉപയോഗിക്കുന്നത്. ഓരോ സംഖ്യാ സമ്പ്രദായത്തിനും തനതായ ആധാരം ഉണ്ട്. ഇത് ആ സംഖ്യാ സമ്പ്രദായത്തിലെ ചിഹ്നങ്ങളുടെ എണ്ണത്തെ ആശ്രയിച്ചിരിക്കുന്നു. ഒരു സംഖ്യാ സമ്പ്രദായത്തിൽ ഉപയോഗിക്കുന്ന അക്കങ്ങളുടെ അല്ലെങ്കിൽ ചിഹ്നങ്ങളുടെ എണ്ണത്തെ ആ സംഖ്യാ സമ്പ്രദായത്തിലെ ആധാരം (Base) അല്ലെങ്കിൽ മൂലസംഖ്യ (Radix) എന്ന് പറയുന്നു.

ചില സംഖ്യാ സമ്പ്രദായങ്ങളെ കുറിച്ച് നമുക്ക് ചർച്ച ചെയ്യാം.

1.5.1 ദശസംഖ്യാ സമ്പ്രദായം (Decimal number system)

ദശസംഖ്യാ സമ്പ്രദായത്തിൽ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 തുടങ്ങിയ പത്ത് അക്കങ്ങളാണ് സംഖ്യാ രൂപീകരണത്തിന് ഉപയോഗിക്കുന്നത്. ദശസംഖ്യാ സമ്പ്രദായത്തിൽ പത്ത് അക്കങ്ങൾ ഉപയോഗിക്കുന്നതുകൊണ്ട് അതിന്റെ ആധാരം (Base) 10 ആകുന്നു. അതുകൊണ്ടു ദശസംഖ്യാ സമ്പ്രദായത്തെ 10 ആധാരമാക്കിയ സംഖ്യാ സമ്പ്രദായം എന്നു കൂടി വിളിക്കുന്നു.

743, 347 എന്നീ രണ്ട് ദശസംഖ്യകൾ പരിഗണിക്കുക.

743 = ഏഴ് നൂറുകൾ + നാലു പത്തുകൾ + മൂന്ന് ഒന്നുകൾ ($7\times10^2 + 4\times10^1 + 3\times10^0$)

347 = മൂന്ന് നൂറുകൾ + നാലു പത്തുകൾ + ഏഴ് ഒന്നുകൾ ($3\times10^2 + 4\times10^1 + 7\times10^0$)

ഇവിടെ ഒന്നാമത്തെ സംഖ്യയായ 743 ൽ 7 ന്റെ സ്ഥാനവില (Weight) $10^2 = 100$ ആകുന്നു. എന്നാൽ രണ്ടാമത്തെ സംഖ്യയായ 347 ൽ 7 ന്റെ സ്ഥാനവില $10^0 = 1$ ആകുന്നു. ഒരു സംഖ്യയുടെ സ്ഥാനവില അതിന്റെ ആപേക്ഷിക സ്ഥാനത്തെ ആശ്രയിച്ചിരിക്കുന്നു. അത്തരം സംഖ്യാ സമ്പ്രദായത്തെ സ്ഥാനീയ സംഖ്യാ സമ്പ്രദായം (Positional Number System) എന്നു പറയുന്നു. എല്ലാ സ്ഥാനീയ സംഖ്യാ സമ്പ്രദായത്തിനും ഒരു ആധാരം (Base) ഉണ്ടായിരിക്കും. ഒരു അക്കത്തിന്റെ സ്ഥാനവില ആധാരത്തിന്റെ ചില കൃത്യങ്കം (Power) ആയിരിക്കും. ഓരോ ദശസംഖ്യ അക്കത്തിന്റെ സ്ഥാന വില 10 ന്റെ കൃത്യങ്കം ആയിരിക്കും (10^0, 10^1, 10^2, ……). 5876 എന്ന ദശസംഖ്യ പരിഗണിക്കുക. ഈ സംഖ്യ താഴെ കാണിച്ചിരിക്കുന്നത് പോലെ വിപുലീകരിച്ചു എഴുതാം.

സ്ഥാനവില (Weight)	10^3	10^2	10^1	10^0
ദശസംഖ്യ	5	8	7	6

$= 5\times10^3 + 8\times10^2 + 7\times10^1 + 6\times10^0$

$= 5\times1000 + 8\times100 + 7\times10 + 6\times1$

$= 5000 + 800 + 70 + 6$

$= 5876$

മുകളിൽ കൊടുത്തിരിക്കുന്ന ഉദാഹരണത്തിൽ 5 എന്ന അക്കത്തിന് ഏറ്റവും കൂടിയ സ്ഥാനവിലയായ 10^3=1000 ഉം 6 എന്ന അക്കത്തിന് ഏറ്റവും കുറഞ്ഞ സ്ഥാനവിലയായ 10^0=1 ഉം ആണ്. ഏറ്റവും കൂടിയ സ്ഥാനവിലയുള്ള അക്കത്തെ കൂടിയ പ്രബലതയുള്ള അക്കം (Most Significant Digit - MSD) എന്നും ഏറ്റവും കുറഞ്ഞ സ്ഥാനവിലയുള്ള അക്കത്തെ കുറഞ്ഞ പ്രബലതയുള്ള അക്കം (Least Significant Digit - LSD) എന്നും വിളിക്കുന്നു. അതിനാൽ മുകളിൽ കൊടുത്തിരിക്കുന്ന സംഖ്യയിൽ MSD എന്നത് 5 ഉം LSD എന്നത് 6 ഉം ആകുന്നു.

ഒരു സംഖ്യയുടെ ഏറ്റവും ഇടതു വശത്തുള്ള അക്കം MSD ഉം ഏറ്റവും വലതു വശത്തുള്ള അക്കം LSD ഉം ആകുന്നു.

ദശാംശ സംഖ്യകളിൽ ദശാംശ ബിന്ദുവിന് വലുത് ഭാഗത്തുള്ള സംഖ്യകളുടെ സ്ഥാനവില 10ന്റെ നെഗറ്റിവ് കൃത്യങ്കങ്ങൾ ആണ് (10^{-1}, 10^{-2}, 10^{-3}, ...). 249.367 എന്ന സംഖ്യ ഉദാഹരണമായി എടുക്കാം.

സ്ഥാനവില (Weight)	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}
ദശസംഖ്യ	2	4	9	3	6	7

MSD (.) LSD

$= 2\times10^2 + 4\times10^1 + 9\times10^0 + 3\times10^{-1} + 6\times10^{-2} + 7\times10^{-3}$

$= 2\times100 + 4\times10 + 9\times1 + 3\times0.1 + 6\times0.01 + 7\times0.001$

$= 200 + 40 + 9 + 0.3 + 0.06 + 0.007$

$= 249.367$

ഇതുവരെ 10 അക്കങ്ങൾ ഉപയോഗിച്ചുകൊണ്ടുള്ള ഒരു സംഖ്യാന സമ്പ്രദായത്തെക്കുറിച്ചാണ് നമ്മൾ ചർച്ചചെയ്തത്. ഇനി നമുക്ക് വ്യത്യസ്ത ആധാരങ്ങളിലുള്ള മറ്റ് സംഖ്യാന സമ്പ്രദായങ്ങൾ രൂപീകരിക്കുന്നതെങ്ങനെ എന്ന് പഠിക്കാം.

1.5.2 ദ്വയസംഖ്യാ സമ്പ്രദായം (Binary number system)

ഒരു സംഖ്യ രൂപീകരിക്കാൻ 0, 1 എന്നീ രണ്ടക്കങ്ങൾ മാത്രം ഉപയോഗിക്കുന്ന സംഖ്യാ സമ്പ്രദായത്തെയാണ് ദ്വയസംഖ്യാ സമ്പ്രദായം (Binary Number System) എന്ന് പറയുന്നത്. ഇംഗ്ലീഷിൽ bi (ബൈ) എന്ന വാക്ക് കൊണ്ട് അർഥമാക്കുന്നത് 2 എന്നാണ്. അതിനാൽ ഈ സംഖ്യാ സമ്പ്രദായത്തിന്റെ ആധാരം 2 ആകുന്നു. അതുകൊണ്ട് ഇതിനെ 2 ആധാരമാക്കിയുള്ള സംഖ്യാ സമ്പ്രദായം എന്ന് കൂടി വിളിക്കുന്നു. ഒരു സംഖ്യ ദ്വയസംഖ്യയാണെന്ന് സൂചിപ്പിക്കാൻ ആ സംഖ്യയോടു കൂടി 2 കീഴ്ക്കുറിപ്പ് (Subscript) ആയി ഉപയോഗിക്കുന്നു.

ഉദാഹരണങ്ങൾ $(1101)_2$, $(101010)_2$, $(1101.11)_2$

ഒരു ദ്വയസംഖ്യയിലെ ഓരോ അക്കത്തെയും ബിറ്റ് (bit) എന്നാണ് വിളിക്കുന്നത്. ഇംഗ്ലീഷിൽ bit ന്റെ പൂർണരൂപം binary digit എന്നാകുന്നു. ദ്വയസംഖ്യാന സമ്പ്രദായവും ഒരു സ്ഥാനീയ സംഖ്യാന സമ്പ്രദായമാണ്. ഓരോ ദ്വയസംഖ്യ അക്കത്തിന്റെയും സ്ഥാന വില 2 ന്റെ കൃത്യങ്കം (Power) ആണ്. $(1101)_2$ എന്ന ദ്വയസംഖ്യ ഉദാഹരണമായി പരിഗണിക്കുക. ഈ ദ്വയസംഖ്യ താഴെ കാണിച്ചിരിക്കുന്നത് പോലെ വിപുലീകരിച്ച് എഴുതാം.

സ്ഥാനവില (Weight)	2^3	2^2	2^1	2^0
Binary Number	1	1	0	1
	MSB			LSB

$= 1\times2^3 + 1\times 2^2 + 0\times2^1 + 1\times2^0$
$= 1\times8 + 1\times4 + 0\times2 + 1\times1$
$= 8 + 4 + 0 + 1$
$= 13$

ദ്വയസംഖ്യയിലെ ഏറ്റവും വലതുവശത്തു നിൽക്കുന്ന അക്കത്തിനെ കുറഞ്ഞ പ്രബലതയുള്ള ബിറ്റ് (Least Significant Bit - LSB) എന്നും ഏറ്റവും ഇടതുവശത്തു നിൽക്കുന്ന അക്കത്തിനെ കൂടുതൽ പ്രബലതയുള്ള ബിറ്റ് (Most Significant Bit - MSB) എന്നും വിളിക്കുന്നു.

1101 എന്ന ദ്വയസംഖ്യ 13 എന്ന ദശസംഖ്യയ്ക്ക് തുല്യമാണ്. എന്നാൽ 1101 എന്ന സംഖ്യ ദശസംഖ്യാന സമ്പ്രദായത്തിലും ഉണ്ട്. പക്ഷെ അതിനെ വ്യാഖ്യാനിക്കുന്നത് ആയിരത്തി ഒരുന്നൂറ്റി ഒന്ന് എന്നാണ്. ഈ ആശയക്കുഴപ്പം ഒഴിവാക്കുവാൻ വേണ്ടി ദശസംഖ്യാ സമ്പ്രദായം ഒഴികെയുള്ള എല്ലാ സംഖ്യാന സമ്പ്രദായങ്ങളിലും ആധാരം വ്യക്തമായി സൂചിപ്പിക്കണം. അതിന്റെ പൊതുവായ ഘടന താഴെ കൊടുത്തിരിക്കുന്നു.

$$(\text{സംഖ്യ})_{\text{ആധാരം}}$$

വ്യത്യസ്ത ആധാരത്തിലുള്ള സംഖ്യകളെ തരം തിരിച്ചറിയുവാൻ ഈ അടയാളപ്പെടുത്തൽ സഹായിക്കുന്നു. ഉദാഹരണമായി 1101 എന്ന ദ്വയസംഖ്യയെ $(1101)_2$ എന്ന് എഴുതുകയും അതിനെ 'ഒന്ന് ഒന്ന് പൂജ്യം ഒന്ന് ആധാരം രണ്ട് ' എന്ന് വായിക്കുകയും ചെയ്യണം. ഒരു സംഖ്യയ്ക്ക് ആധാരം നൽകിയിട്ടില്ലെങ്കിൽ അതിനെ ദശസംഖ്യയായി പരിഗണിക്കണം. അതായത് ദശസംഖ്യക്ക് ആധാരം സൂചിപ്പിക്കണമെന്ന് നിർബന്ധമില്ല.

ഭിന്നകമായ ഒരു ദ്വയസംഖ്യയുടെ Binary point വലതുഭാഗത്തുള്ള അക്കങ്ങളുടെ സ്ഥാനവില 2 ന്റെ നെഗറ്റീവ് കൃത്യങ്കം ആയിരിക്കും. (2^{-1}, 2^{-2}, 2^{-3},....). $(111.011)_2$ എന്ന സംഖ്യ ഉദാഹരണമായി എടുക്കാം.

സ്ഥാനവില (Weight)	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
ദ്വയസംഖ്യ (Binary Numbers)	1	1	1	0	1	1
	MSB			(.)		LSB

$= 1\times2^2 + 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} + 1\times2^{-3}$

$= 1\times4 + 1\times2 + 1\times1 + 0\times\frac{1}{2} + 1\times\frac{1}{4} + 1\times\frac{1}{8}$

$= 4 + 2 + 1 + 0 + 0.25 + 0.125$

$= 7.375$

കമ്പ്യൂട്ടറിൽ ദ്വയസംഖ്യയുടെ പ്രധാന്യം

ദ്വയസംഖ്യാ സമ്പ്രദായം 1, 0 എന്നീ അക്കങ്ങൾ അടിസ്ഥാനമാക്കിയാണെന്നു നമ്മൾ കണ്ടല്ലോ. ചിത്രം 1.8 ൽ വൈദ്യുതിയുടെ ഓൺ (ON) ആയിരിക്കുന്ന അവസ്ഥ 1 കൊണ്ടും ഓഫ് (OFF) ആയിരിക്കുന്ന അവസ്ഥ 0 കൊണ്ടും സൂചിപ്പിക്കുന്നു. ഇക്കാരണത്താൽ, കമ്പ്യൂട്ടറിൽ ഡാറ്റയെ പ്രതിനിധാനം ചെയ്യുന്നതിന് അടിസ്ഥാന സംഖ്യാ സമ്പ്രദായമായി ദ്വയസംഖ്യാ സമ്പ്രദായം ഉപയോഗിക്കുന്നു.

ചിത്രം 1.8: ON ഉം OFF ന്റെയും ഡിജിറ്റൽ രൂപത്തിന്റെ പ്രതിനിധാനം

1.5.3 അഷ്ടസംഖ്യാ സമ്പ്രദായം (Octal number system)

എട്ട് അക്കങ്ങളായ 0, 1, 2, 3, 4, 5, 6, 7 എന്നിവ ഉപയോഗിച്ചുണ്ടാക്കുന്ന സംഖ്യാ സമ്പ്രദായത്തെ അഷ്ടസംഖ്യാ സമ്പ്രദായം (Octal Number System) എന്ന് പറയുന്നു. ഇംഗ്ലീഷിൽ Octa (ഒക്ട) എന്ന വാക്ക് കൊണ്ട് അർഥമാക്കുന്നത് 8 എന്നാണ്. അതിനാലാണ് ഈ സംഖ്യാ സമ്പ്രദായത്തെ ഒക്ടൽ സംഖ്യാന സമ്പ്രദായം എന്ന് പറയുന്നത്. ഈ സംഖ്യാ സമ്പ്രദായത്തിന്റെ ആധാരം 8 ആകുന്നു. അതുകൊണ്ട് ഇതിനെ 8 ആധാരമായ സംഖ്യാ സമ്പ്രദായം എന്നും വിളിക്കുന്നു. ഉദാഹരണമായി $(236)_8$ പരിഗണിക്കുക. ഓരോ ഒക്ടൽ അക്കത്തിന്റെയും സ്ഥാനവില 8 ന്റെ കൃത്യങ്കം (Power) ആയിരിക്കും (8^0, 8^1, 8^2, 8^3, …..). $(236)_8$ എന്ന സംഖ്യ താഴെ കാണിച്ചിരിക്കുന്നത് പോലെ വിപുലീകരിച്ചു എഴുതാം.

സ്ഥാനവില (Weight)	8^2	8^1	8^0
ഒക്ടൽ സംഖ്യ	2	3	6

$= 2\times8^2 + 3\times8^1 + 6\times8^0$

$= 2\times64 + 3\times8 + 6\times1$

$= 128 + 24 + 6$

$= 158$

ഭിന്നകമായ ഒരു അഷ്ടസംഖ്യയുടെ അംശബിന്ദുവിന് വലതുഭാഗത്തുള്ള അക്കങ്ങളുടെ

സ്ഥാനവില 8 ന്റെ നെഗറ്റീവ് കൃത്യങ്കം ആയിരിക്കും (8^{-1}, 8^{-2}, 8^{-3}, …….). $(172.4)_8$ എന്ന സംഖ്യ ഉദാഹരണമായി എടുക്കാം.

സ്ഥാനവില (Weight)	8^2	8^1	8^0	8^{-1}
ഒക്ടൽ സംഖ്യ	1	7	2	4

$$= 1\times8^2 + 7\times8^1 + 2\times8^0 + 4\times8^{-1}$$
$$= 64 + 56 + 2 + 4\times\frac{1}{8}$$
$$= 122 + 0.5$$
$$= 122.5$$

1.5.4 ഷോഡശ (ഹെക്സാഡെസിമൽ) സംഖ്യാ സമ്പ്രദായം (Hexadecimal number system)

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F എന്നീ 16 ചിഹ്നങ്ങൾ ഉപയോഗിച്ചുണ്ടാക്കുന്ന സംഖ്യാന സമ്പ്രദായത്തെ ഹെക്സാഡെസിമൽ സംഖ്യാന സമ്പ്രദായം എന്ന് പറയുന്നു. ഈ സംഖ്യാന സമ്പ്രദായത്തിൽ 16 ചിഹ്നങ്ങൾ ഉപയോഗിക്കുന്നതുകൊണ്ട് ഇതിന്റെ ആധാരം 16 ആകുന്നു. ആയതിനാൽ ഇതിനെ 16 ആധാരമായ സംഖ്യാന സമ്പ്രദായം എന്നും വിളിക്കുന്നു. ഈ സംഖ്യാന സമ്പ്രദായത്തിലെ A, B, C, D, E, F എന്നീ ചിഹ്നങ്ങൾ ഉപയോഗിക്കുന്നത് യഥാക്രമം ദശസംഖ്യാന സമ്പ്രദായത്തിലെ 10, 11, 12, 13, 14, 15 എന്ന സംഖ്യകളെ സൂചിപ്പിക്കുന്നതിനാണ്. ഹെക്സാഡെസിമൽ അക്കങ്ങളും അവയ്ക്ക് തുല്യമായ ദശസംഖ്യകളും ചുവടെ കാണിച്ചിരിക്കുന്നു.

ഹെക്സാഡെസിമൽ	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ദശസംഖ്യ	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

ഉദാഹരണമായി $(12AF)_{16}$ എന്ന ഹെക്സാഡെസിമൽ സംഖ്യ പരിഗണിക്കുക. ഓരോ ഹെക്സാഡെസിമൽ അക്കത്തിന്റെയും സ്ഥാനവില 16 ന്റെ കൃത്യങ്കം (Power) ആയിരിക്കും (16^0, 16^1, 16^2, 16^3, …..). ഈ സംഖ്യയെ താഴെ കാണിച്ചിരിക്കുന്നത് പോലെ വിപുലീകരിച്ചു എഴുതാം.

സ്ഥാനവില (Weight)	16^3	16^2	16^1	16^0
ഹെക്സാഡെസിമൽ അക്കം	1	2	A	F

$$= 1\times16^3 + 2\times16^2 + 10\times16^1 + 15\times16^0$$
$$= 1\times4096 + 2\times256 + 10\times16 + 15\times1$$
$$= 4096 + 512 + 160 + 15$$
$$= 4783$$

ഭിന്നകമായ ഒരു ഹെക്സാഡെസിമൽ സംഖ്യയുടെ അംശബിന്ദുവിന് വലതുഭാഗത്തുള്ള അക്കങ്ങളുടെ സ്ഥാനവില 16 ന്റെ നെഗറ്റീവ് കൃത്യങ്കം ആയിരിക്കും (16^{-1}, 16^{-2}, 16^{-3}, ...) $(2D.4)_{16}$ എന്ന സംഖ്യാ ഉദാഹരണമായി എടുക്കാം.

സ്ഥാനവില (Weight)	16^1	16^0	16^{-1}
ഹെക്സാഡെസിമൽ അക്കം	2	D	4

$= 2\times16^1 + 13\times16^0 + 4\times$

$= 32 + 13 + 0.25$

$= 45.25$

പട്ടിക 1.3 ൽ വിവിധ സംഖ്യാന സമ്പ്രദായത്തിൽ ഉപയോഗിക്കുന്ന ആധാരവും ചിഹ്നങ്ങളും കാണിച്ചിരിക്കുന്നു.

സംഖ്യാ സമ്പ്രദായം	ആധാരം	ഉപയോഗിക്കുന്ന ചിഹ്നങ്ങൾ
ബൈനറി	2	0, 1
ഒക്ടൽ	8	0, 1, 2, 3, 4, 5, 6, 7
ഡെസിമൽ	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
ഹെക്സാ ഡെസിമൽ	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

പട്ടിക 1.3: വിവിധ സംഖ്യാ സമ്പ്രദായത്തിലുമുള്ള ആധാരവും ചിഹ്നങ്ങളും

ഒക്ടൽ, ഹെക്സാഡെസിമൽ സംഖ്യാ സമ്പ്രദായങ്ങളുടെ പ്രാധാന്യം

$\frac{1}{16}$

കമ്പ്യൂട്ടറിൽ ഡാറ്റ പ്രോസസ്സ് ചെയ്യുന്നതിനും അതിനെ പ്രതിനിധാനം ചെയ്യുന്നതിനും ബൈനറി സംഖ്യാ സമ്പ്രദായമാണ് ഉപയോഗിക്കുന്നത് എന്ന് നമ്മൾ മനസ്സിലാക്കിക്കഴിഞ്ഞു. ബൈനറി സംവിധാനത്തിൽ സംഖ്യകൾ പ്രതിനിധാനം ചെയ്യുന്നതിനും അവയുടെ പ്രവർത്തനങ്ങൾക്കും കൂടുതൽ ബിറ്റുകളും പ്രയത്നങ്ങളും ആവശ്യമാണ്. മൂന്നു ബിറ്റുകളുടെ ഗ്രൂപ്പിനെ ഒരു ഒക്ടൽ അക്കമായും (കാരണം $2^3 = 8$) നാലു ബിറ്റുകളുടെ ഗ്രൂപ്പിനെ ഒരു ഹെക്സാഡെസിമൽ അക്കമായും (കാരണം $2^4 = 16$) മാറ്റാവുന്നതും ഇത്തരം ഗ്രൂപ്പുകളെ അവയുടെ തത്തുല്യമായ ഒക്ടൽ, ഹെക്സാഡെസിമൽ ചിഹ്നങ്ങളിലേക്കു മാറ്റാവുന്നതാണ്. ബൈനറി സംഖ്യകളുടെ ഒക്ടൽ, ഹെക്സാഡെസിമൽ സംഖ്യാ സമ്പ്രദായത്തിലേക്കുള്ള ഇത്തരം മാറ്റവും തിരിച്ചുള്ള മാറ്റവും വളരെ എളുപ്പമാണ്. ഇലക്ട്രോണിക് സർക്യൂട്ടുകളുടെ രൂപകൽപ്പനയിലും പ്രവർത്തനത്തിലും ഈ പരിവർത്തന പ്രക്രിയ വലിയ തോതിൽ ഉപയോഗിക്കുന്നു.

സ്വയം വിലയിരുത്താം

1. ഒരു സംഖ്യാ സമ്പ്രദായത്തിൽ ഉപയോഗിച്ചിരിക്കുന്ന ചിഹ്നങ്ങളുടെ എണ്ണത്തെ .... എന്ന് വിളിക്കുന്നു.
2. താഴെ കൊടുത്തിരിക്കുന്നതിൽ നിന്ന് അസാധുവായ സംഖ്യകൾ തിരഞ്ഞെടുക്കുക.
   i) $(10101)_8$ ii) $(123)_4$ iii) $(768)_8$ iv) $(ABC)_{16}$
3. ബിറ്റ് എന്ന പദം നിർവചിക്കുക.
4. 7854.25. എന്ന ദശസംഖ്യയുടെ എം.എസ്.ഡി (MSD) കണ്ടുപിടിക്കുക.
5. ഹെക്സാഡെസിമൽ സംഖ്യാ സമ്പ്രദായത്തിന്റെ ആധാരം .... ആകുന്നു.

1.6 സംഖ്യകളുടെ പരിവർത്തനങ്ങൾ (Number Conversions)

വിവിധ സംഖ്യാന സമ്പ്രദായങ്ങളെക്കുറിച്ച് നമ്മൾ പഠിച്ചു കഴിഞ്ഞു, ഒരാധാരത്തിലുള്ള സംഖ്യകളെ മറ്റൊരാധാരത്തിലുള്ള തത്തുല്യ സംഖ്യകളാക്കി പരിവർത്തനം ചെയ്യുന്ന തെങ്ങനെയാണെന്നു നമുക്ക് ചർച്ച ചെയ്യാം. ദശസംഖ്യയിൽ നിന്ന് ബൈനറി, ബൈനറിയിൽ നിന്ന് ദശസംഖ്യ, ദശസംഖ്യയിൽ നിന്ന് ഒക്ടൽ എന്നിങ്ങനെ പല വിധത്തിലുള്ള സംഖ്യാന സമ്പ്രദായത്തിലേക്കു പരിവർത്തനം ചെയ്യാം. ഒരു സംഖ്യാന സമ്പ്രദായത്തിൽ നിന്ന് മറ്റൊരു സംഖ്യാന സമ്പ്രദായത്തിലേക്ക് എങ്ങനെ പരിവർത്തനം ചെയ്യാമെന്ന് നമുക്ക് നോക്കാം.

1.6.1 ദശസംഖ്യയിൽ നിന്ന് ബൈനറിസംഖ്യയിലേക്കുള്ള പരിവർത്തനം (Decimal to binary conversion)

ആവർത്തിച്ചുള്ള ഹരണം വഴിയാണ് ദശസംഖ്യയെ ബൈനറി സംഖ്യയിലേക്കു പരിവർത്തനം ചെയ്യുന്നത്. ഈ രീതിയിൽ ദശസംഖ്യയെ 2 കൊണ്ട് തുടർച്ചയായി ഹരിക്കുകയും (സംഖ്യ 0 ആകുന്നത് വരെ), അതിന്റെ ശിഷ്ടങ്ങൾ രേഖപ്പെടുത്തുകയും ചെയ്യുന്നു. MSB അവസാന ശിഷ്ടമായും LSB ആദ്യത്തെ ശിഷ്ടമായും എടുത്ത് ശിഷ്ടങ്ങളെ കൂട്ടമായി എഴുതിയാൽ ദ്വയസംഖ്യക്ക് തുല്യമായ സംഖ്യ ലഭിക്കുന്നു. ഓരോ ഘട്ടത്തിലും ഹരിക്കുമ്പോൾ കിട്ടുന്ന ശിഷ്ടങ്ങൾ ഒന്നുകിൽ 0 അല്ലെങ്കിൽ 1 എന്നീ ബൈനറി അക്കങ്ങൾ ആയിരിക്കും.

ഉദാഹരണങ്ങൾ

25 എന്ന ദശസംഖ്യയുടെ ബൈനറിക്ക് തുല്യമായ സംഖ്യ കണ്ടുപിടിക്കുക.

		ശിഷ്ടങ്ങൾ	
2	25		
2	12	1	LSB
2	6	0	
2	3	0	
2	1	1	
	0	1	MSB

$(25)_{10} = (11001)_2$

$(80)_{10}$ ന് തുല്യമായ ബൈനറി സംഖ്യ കണ്ടുപിടിക്കുക.

		ശിഷ്ടങ്ങൾ	
2	80		
2	40	0	LSB
2	20	0	
2	10	0	
2	5	0	
2	2	1	
2	1	0	
	0	1	MSB

$(80)_{10} = (1010000)_2$

സൂചന: ഒറ്റ സംഖ്യയായ ദശസംഖ്യയ്ക്ക് തുല്യമായ ബൈനറി സംഖ്യ 1ൽ അവസാനിക്കുകയും ഇരട്ട സംഖ്യയായ ദശസംഖ്യക്ക് തുല്യമായ ബൈനറി സംഖ്യ 0 ൽ അവസാനിക്കുകയും ചെയ്യുന്നു.

ദശാംശ സംഖ്യകൾ ബൈനറിയിലേക്ക് പരിവർത്തനം ചെയ്യൽ (Converting decimal fraction to binary)

ദശാംശ സംഖ്യകൾ ബൈനറിയിലേക്ക് മാറ്റാൻ അതിനെ തുടർച്ചയായി 2 കൊണ്ട് ഗുണിക്കുന്ന രീതിയാണ് നാം ഉപയോഗിക്കുന്നത്. ഉത്തരത്തിന്റെ പൂർണസംഖ്യഭാഗം ബൈനറി ഭിന്നകത്തിലെ MSB ആയിരിക്കും. അടുത്ത ബൈനറി ഭിന്നകത്തിന്റെ പ്രബലതയുള്ള ബിറ്റ് കിട്ടുന്നതിന് വീണ്ടും ഭിന്നക ഭാഗത്തിന്റെ ഉത്തരത്തെ 2 കൊണ്ട് ഗുണിക്കുന്നു. ഭിന്നക ഭാഗം പൂജ്യം ആകുന്നതു വരെയോ അല്ലെങ്കിൽ ആവശ്യമുള്ളത്ര കൃത്യത (Precision) ലഭിക്കുന്നത് വരെയോ ഈ നടപടിക്രമം തുടരുന്നു.

ഉദാഹരണങ്ങൾ:

0.75 നെ ബൈനറിയിലേക്ക് മാറ്റുക.

	0.75 × 2 = 1.50
1	.50 × 2 = 1.00
1	.00

$(0.75)_{10} = (0.11)_2$

0.625 നെ ബൈനറിയിലേക്ക് മാറ്റുക.

	0.625 × 2 = 1.25
1	.25 × 2 = 0.50
0	.50 × 2 = 1.00
1	.00

$(0.625)_{10} = (0.101)_2$

15.25 നെ ബൈനറിയിലേക്ക് മാറ്റുക.

15 നെ ബൈനറിയിലേക്കു മാറ്റുക.

		ശിഷ്ടങ്ങൾ
2	15	
2	7	1
2	3	1
2	1	1
	0	1

0.25നെ ബൈനറിയിലേക്കു മാറ്റുക

	0.25 × 2 = 0.50
0	.50 × 2 = 1.00
1	.00

$(15.25)_{10} = (1111.01)_2$

1.6.2 ദശസംഖ്യയിൽ നിന്ന് ഒക്ടലിലേക്കുള്ള പരിവർത്തനം (Decimal to octal conversion)

ആവർത്തിച്ചുള്ള ഹരണം വഴിയാണ് ദശസംഖ്യയെ ഒക്ടൽ സംഖ്യയിലേക്കു പരിവർത്തനം ചെയ്യുന്നത്. ദശസംഖ്യയെ 8 കൊണ്ട് തുടർച്ചയായി ഹരിക്കുകയും (സംഖ്യ 0 ആകുന്നത് വരെ),

അതിന്റെ ശിഷ്ടങ്ങൾ രേഖപ്പെടുത്തുകയും ചെയ്യുന്നു. MSD അവസാന ശിഷ്ടമായും LSD ആദ്യത്തെ ശിഷ്ടമായും എടുത്ത് ശിഷ്ടങ്ങളെ കൂട്ടമായി എഴുതിയാൽ ഒക്ടൽസംഖ്യക്ക് തുല്യമായ സംഖ്യ ലഭിക്കുന്നു. ഓരോ ഘട്ടത്തിലും ഹരിക്കുമ്പോൾ കിട്ടുന്ന ശിഷ്ടങ്ങൾ 0, 1, 2, 3, 4, 5, 6 , 7 എന്നിവയിൽ ഏതെങ്കിലും ആയിരിക്കും.

ഉദാഹരണങ്ങൾ:

125 എന്ന ദശസംഖ്യക്ക് തുല്യമായ ഒക്ടൽ സംഖ്യ കണ്ടുപിടിക്കുക.

8	125	ശിഷ്ടങ്ങൾ	
8	15	5	↑ LSD
8	1	7	
	0	1	MSD

$(125)_{10} = (175)_8$

$(400)_{10}$ ന് തുല്യമായ ഒക്ടൽ സംഖ്യ കണ്ടുപിടിക്കുക.

8	400	ശിഷ്ടങ്ങൾ
8	50	0 ↑
8	6	2
	0	6

$(400)_{10} = (620)_8$

1.6.3 ദശസംഖ്യയിൽ നിന്ന് ഹെക്സാഡെസിമലിലേക്കുള്ള പരിവർത്തനം (Decimal to hexadecimal conversion)

ആവർത്തിച്ചുള്ള ഹരണം വഴിയാണ് ദശസംഖ്യയെ ഹെക്സാഡെസിമൽ സംഖ്യയിലേക്കു പരിവർത്തനം ചെയ്യുന്നത്. ദശസംഖ്യയെ 16 കൊണ്ട് തുടർച്ചയായി ഹരിക്കുകയും (സംഖ്യ 0 ആകുന്നത് വരെ), അതിന്റെ ശിഷ്ടങ്ങൾ രേഖപ്പെടുത്തുകയും ചെയ്യുന്നു. MSD അവസാന ശിഷ്ടമായും LSD ആദ്യത്തെ ശിഷ്ടമായും എടുത്ത് ശിഷ്ടങ്ങളെ കൂട്ടമായി എഴുതിയാൽ ഹെക്സാഡെസിമൽ സംഖ്യക്ക് തുല്യമായ സംഖ്യ ലഭിക്കുന്നു. ഓരോ ഘട്ടത്തിലും ഹരിക്കുമ്പോൾ കിട്ടുന്ന ശിഷ്ടങ്ങൾ 0, 1, 2, 3, 4, 5, 6,7 , 8, 9, 10, 11, 12, 13, 14, 15 എന്നിവയിൽ ഏതെങ്കിലും ആയിരിക്കും. കിട്ടുന്ന ശിഷ്ടങ്ങൾ 10, 11, 12, 13, 14, 15 ആണെങ്കിൽ അതിനെ യഥാക്രമം A, B, C, D, E, F എന്നിങ്ങനെ രേഖപ്പെടുത്തണം.

ഉദാഹരണങ്ങൾ:

155 എന്ന ദശസംഖ്യക്ക് തുല്യമായ ഹെക്സാഡെസിമൽ സംഖ്യ കണ്ടുപിടിക്കുക.

16	155	ശിഷ്ടങ്ങൾ	
16	9	11 (B) ↑	→ LSD
	0	9	→ MSD

$(155)_{10} = (9B)_{16}$

ഉദാഹരണങ്ങൾ: 380_{10} തുല്യമായ ഹെക്സാഡെസിമൽ സംഖ്യ കണ്ടുപിടിക്കുക.

		ശിഷ്ടങ്ങൾ
16	380	
16	23	12 (C)
16	1	7
	0	1

$(380)_{10} = (17C)_{16}$

1.6.4 ബൈനറിയിൽ നിന്ന് ദശസംഖ്യയിലേക്കുള്ള പരിവർത്തനം (Binary to decimal conversion)

ബൈനറി സംഖ്യകൾക്ക് തുല്യമായ ദശസംഖ്യ കാണുന്നതിന്, ബൈനറി സംഖ്യയിലെ ഓരോ അക്കത്തിനെയും, അതിന്റെ സ്ഥാനവില കൊണ്ടു ക്രമമായി ഗുണിച്ച് തുക കണ്ടാൽ മതി. സ്ഥാനവില 2 ന്റെ കൃത്യങ്കം ആയിരിക്കും $(2^0, 2^1, 2^2, 2^3, ...)$..

ഉദാഹരണങ്ങൾ:

$(10110)_2$ നെ ദശസംഖ്യയിലേക്കു മാറ്റുക.

സ്ഥാനവില (Weight)	2^4	2^3	2^2	2^1	2^0
ബൈനറി അക്കം	1	0	1	1	0

$$(10110)_2 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$$
$$= 16 + 0 + 4 + 2 + 0$$
$$= 22$$

$(10110)_2 = (22)_{10}$

$(11011)_2$ നെ ദശസംഖ്യയിലേക്കു മാറ്റുക.

സ്ഥാനവില (Weight)	2^4	2^3	2^2	2^1	2^0
ബൈനറി അക്കം	1	1	0	1	1

$$(11011)_2 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$$
$$= 16 + 8 + 2 + 1$$
$$= 27$$

$(11011)_2 = (27)_{10}$

$(1100010)_2$ നെ ദശസംഖ്യയിലേക്കു മാറ്റുക.

സ്ഥാനവില (Weight)	2^6	2^5	2^4	2^3	2^2	2^1	2^0
ബൈനറി അക്കം	1	1	0	0	0	1	0

$(1100010)_2 = 1\times2^6 + 1\times2^5 + 0\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$

$= 64 + 32 + 2$

$= 98$

$(1100010)_2=(98)_{10}$

പട്ടിക 1.4 ൽ രണ്ടിന്റെ 10 വരെയുള്ള കൃത്യങ്കങ്ങൾ കൊടുത്തിരിക്കുന്നു.

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1024	512	256	128	64	32	16	8	4	2	1

പട്ടിക 1.4: രണ്ടിന്റെ കൃത്യങ്കങ്ങൾ

ബൈനറി ഭിന്നകങ്ങൾ ദശസംഖ്യയിലേക്ക് പരിവർത്തനം ചെയ്യൽ (Binary fraction to decimal)

ഒരു ബൈനറി ഭിന്നസംഖ്യ ദശസംഖ്യയിലേക്ക് മാറ്റുന്നതിന്, ഓരോ അക്കത്തിനെയും അതിന്റെ സ്ഥാനവില കൊണ്ടു ക്രമമായി ഗുണിച്ച് തുക കണ്ടാൽ മതി. ബൈനറി അംശബിന്ദുവിന് ശേഷമുള്ള അക്കത്തിന്റെ സ്ഥാനവില 2 ന്റെ നെഗറ്റീവ് കൃത്യങ്കം ആയിരിക്കും ($2^{-1}, 2^{-2}, 2^{-3}, \ldots$)

ഉദാഹരണങ്ങൾ:

$(0.1011)_2$ നെ ദശസംഖ്യയിലേക്ക് മാറ്റുക.

സ്ഥാനവില (Weight)	2^{-1}	2^{-2}	2^{-3}	2^{-4}
ബൈനറി അക്കം	1	0	1	1

$(0.1011)_2 = 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + 1\times2^{-4}$

$= 0.5 + 0 + 0.125 + 0.0625$

$= 0.6875$

$(0.1011)_2 = (0.6875)_{10}$

$(0.101)_2$ നെ ദശസംഖ്യയിലേക്ക് മാറ്റുക.

സ്ഥാനവില (Weight)	2^{-1}	2^{-2}	2^{-3}
ബൈനറി അക്കം	1	0	1

$(0.101)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$

$= 0.5 + 0 + 0.125$

$= 0.625$

$(0.101)_2 = (0.625)_{10}$

$(1010.11)_2$ നെ ദശസംഖ്യയിലേക്ക് മാറ്റുക.

സ്ഥാനവില (Weight)	2^3	2^2	2^1	2^0
ബൈനറി അക്കം	1	0	1	1

$(1010)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$

$= 8 + 0+2+0$

$= 10$ $\quad (1010)_2 = (10)_{10}$

$(0.11)_2 = 1\times2^{-1} + 1\times2^{-2}$

$= 0.5 + 0.25$

$= 0.75$ $\quad (0.11)_2 =(0.75)_{10}$

സ്ഥാനവില (Weight)	2^{-1}	2^{-2}
ബൈനറി അക്കം	1	1

$(1010.11)_2 = (10.75)_{10}$

പട്ടിക 1.5 ൽ രണ്ടിന്റെ നെഗറ്റീവ് കൃത്യങ്കങ്ങൾ കാണിച്ചിരിക്കുന്നു.

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
0.5	0.25	0.125	0.0625	0.03125

പട്ടിക 1.5: രണ്ടിന്റെ നെഗറ്റീവ് കൃത്യങ്കങ്ങൾ

1.6.5 ഒക്ടൽ സംഖ്യയിൽ നിന്ന് ദശസംഖ്യയിലേക്കുള്ള പരിവർത്തനം (Octal to decimal conversion)

ഒക്ടൽ സംഖ്യയെ ദശസംഖ്യയിലേക്കു മാറ്റുന്നതിന്, ഒക്ടൽ സംഖ്യയിലെ ഓരോ അക്കത്തിനെയും, അതിന്റെ സ്ഥാനവില കൊണ്ടു ക്രമമായി ഗുണിച്ച് തുക കണ്ടാൽ മതി. സ്ഥാനവില 8 ന്റെ കൃത്യങ്കം ആയിരിക്കും ($8^0, 8^1, 8^2, 8^3, \ldots$).

ഉദാഹരണങ്ങൾ:

$(257)_8$ നെ ദശസംഖ്യയിലേക്കു മാറ്റുക.

സ്ഥാനവില (Weight)	8^2	8^1	8^0
ഒക്ടൽ അക്കം	2	5	7

$(257)_8 = 2\times8^2 + 5\times8^1 + 7\times8^0$

$= 128 + 40 +7$

$= 175$

$(257)_8 = (175)_{10}$

$(157)_8$ നെ ദശസംഖ്യയിലേക്കു മാറ്റുക.

സ്ഥാനവില (Weight)	8^2	8^1	8^0
ഒക്ടൽ അക്കം	1	5	7

$(157)_8 = 1\times8^2 + 5\times8^1 + 7\times8^0$

$= 64 + 40 + 7$

$= 111$

$(157)_8 = (111)_{10}$

$(1005)_8$ നെ ദശസംഖ്യയിലേക്കു മാറ്റുക.

സ്ഥാനവില Weight	8^3	8^2	8^1	8^0
ഒക്റ്റൽ അക്കം	1	0	0	5

$(1005)_8 = 1\times8^3 + 0\times8^2 + 0\times8^1 + 5\times8^0$

$= 512+5$

$= 517$

$(1005)_8 = (517)_{10}$

1.6.6 ഹെക്സാഡെസിമൽ സംഖ്യയിൽ നിന്ന് ദശസംഖ്യയിലേക്കുള്ള പരിവർത്തനം (Hexadecimal to decimal conversion)

ഹെക്സാഡെസിമൽ സംഖ്യയെ ദശസംഖ്യയിലേക്കു മാറ്റുന്നതിന്, ഹെക്സാഡെസിമൽ സംഖ്യയിലെ ഓരോ അക്കത്തിനെയും, അതിന്റെ സ്ഥാനവില കൊണ്ടു ക്രമമായി ഗുണിച്ച് തുക കണ്ടാൽ മതി. സ്ഥാനവില 16 ന്റെ കൃത്യങ്കം ആയിരിക്കും (16^0, 16^1, 16^2, ...). ഹെക്സാഡെസിമൽ അക്കങ്ങൾ A, B, C, D, E, F ആണെങ്കിൽ അത് യഥാക്രമം 10, 11, 12, 13, 14, 15 എന്നിങ്ങനെ മാറ്റി എഴുതണം.

ഉദാഹരണങ്ങൾ:

$(AB)_{16}$ നെ ദശസംഖ്യയിലേക്കു മാറ്റുക.

സ്ഥാന വില (Weight)	16^1	16^0
ഹെക്സാഡെസിമൽ അക്കം	A	B

A = 10 B = 11

$(AB)_{16} = 10\times16^1+11\times16^0$

$= 160 + 11$

$= 171$

$(AB)_{16} = (171)_{10}$

ഉദാഹരണങ്ങൾ: $(2D5)_{16}$ നെ ദശസംഖ്യയിലേക്കു മാറ്റുക.

സ്ഥാന വില (Weight)	16^1	16^1	16^0
ഹെക്സാഡെസിമൽ അക്കം	2	D	5

D = 13

$(2D5)_{16} = 2\times16^2+13\times16^1+5\times16^0$

$= 512+208+5$

$= 725$

$(AB)_{16} = (171)_{10}$

1.6.7 ഒക്ടലിൽ നിന്ന് ബൈനറിയിലേക്കുള്ള പരിവർത്തനം (Octal to binary conversion)

ഓരോ ഒക്ടൽ അക്കവും തത്തുല്യമായ 3 ബിറ്റ് ബൈനറി അക്കത്തിലേക്ക് മാറ്റി എഴുതിയാൽ ഒക്ടൽ സംഖ്യ ബൈനറി സംഖ്യയായി പരിവർത്തനം ചെയ്യാനാകും. സാധ്യമായ എട്ട് ഒക്ടൽ അക്കങ്ങളും അവയുടെ തത്തുല്യ ബൈനറി അക്കങ്ങളും പട്ടിക 1.6 ൽ നൽകിയിരിക്കുന്നു.

ഒക്ടൽ അക്കം	0	1	2	3	4	5	6	7
തുല്യമായ ബൈനറി	000	001	010	011	100	101	110	111

പട്ടിക 1.6: ഒക്ടൽ അക്കങ്ങളുടെ തത്തുല്യമായ ബൈനറി സംഖ്യകൾ.

ഉദാഹരണങ്ങൾ:

$(437)_8$ നെ ബൈനറിയിലേക്കു മാറ്റുക.

ഓരോ ഒക്ടൽ അക്കത്തിനും തുല്യമായ 3 ബിറ്റ് ബൈനറി അക്കങ്ങൾ താഴെക്കൊടുത്തിരിക്കുന്നു.

4	3	7
↓	↓	↓
100	011	111

$(437)_8=(100011111)_2$

$(7201)_8$ നെ ബൈനറിയിലേക്കു മാറ്റുക.

ഓരോ ഒക്ടൽ അക്കത്തിനും തുല്യമായ 3 ബിറ്റ് ബൈനറി അക്കങ്ങൾ താഴെക്കൊടുത്തിരിക്കുന്നു.

7	2	0	1
↓	↓	↓	↓
111	010	000	001

$(7201)_8 = (111010000001)_2$

1.6.8 ഹെക്സാഡെസിമലിൽ നിന്ന് ബൈനറിയിലേക്കുള്ള പരിവർത്തനം (Hexadecimal to binary conversion)

ഓരോ ഹെക്സാഡെസിമൽ അക്കവും തത്തുല്യമായ 4 ബിറ്റ് ബൈനറി ആക്കി മാറ്റി എഴുതിയാൽ ഹെക്സാഡെസിമൽ സംഖ്യ ബൈനറിയായി പരിവർത്തനം ചെയ്യാനാകും. ഹെക്സാഡെസിമൽ അക്കങ്ങളും അവയ്ക്കു തുല്യമായ ബൈനറി അക്കങ്ങളും പട്ടിക 1.7 ൽ നൽകിയിരിക്കുന്നു.

ഉദാഹരണങ്ങൾ:

$(AB)_{16}$ നെ ബൈനറിയിലേക്കു മാറ്റുക.

ഓരോ ഹെക്സാഡെസിമൽ അക്കത്തിനും തുല്യമായ 4 ബിറ്റ് ബൈനറി അക്കങ്ങൾ താഴെക്കൊടുത്തിരിക്കുന്നു.

A	B
↓	↓
1010	1011

$$(AB)_{16} = (10101011)_2$$

$(2F15)_{16}$ നെ ബൈനറിയിലേക്കു മാറ്റുക

ഓരോ ഹെക്സാഡെസിമൽ അക്കത്തിനും തുല്യമായ 4 ബിറ്റ് ബൈനറി അക്കങ്ങൾ താഴെക്കൊടുത്തിരിക്കുന്നു.

2	F	1	5
↓	↓	↓	↓
0010	1111	0001	0101

$$(2F15)_{16} = (10111100010101)_2$$

ഹെക്സാ ഡെസിമൽ അക്കം	തുല്യമായ ബൈനറി
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

പട്ടിക 1.7: ഹെക്സാഡെസിമൽ അക്കങ്ങളുടെ തത്തുല്യമായ ബൈനറി അക്കങ്ങൾ.

1.6.9 ബൈനറിയിൽ നിന്നും ഒക്ടലിലേക്കുള്ള പരിവർത്തനം (Binary to octal conversion)

തന്നിരിക്കുന്ന ബൈനറി സംഖ്യ വലത്തു നിന്ന് ഇടത്തേക്ക് 3 ബൈനറി ബിറ്റുകളുടെ കൂട്ടങ്ങളാക്കി അതിന്റെ തത്തുല്യമായ ഒക്ടൽ അക്കം എഴുതിയാൽ ഒരു ബൈനറി സംഖ്യ ഒക്ടൽ സംഖ്യയിലേക്കു പരിവർത്തനം ചെയ്യാം. മൂന്നിന്റെ കൂട്ടങ്ങൾ ആക്കുമ്പോൾ ഏറ്റവും ഇടത് വശത്തെ കൂട്ടത്തിൽ 3 ബിറ്റുകൾ തികയുന്നില്ലെങ്കിൽ ഇടത് വശത്ത് ആവശ്യമായ പൂജ്യങ്ങൾ കൊടുത്ത് 3 ബിറ്റ് രൂപത്തിൽ ആക്കണം.

ഉദാഹരണങ്ങൾ:

$(101100111)_2$ നെ ഒക്ടലിലേക്കു മാറ്റുക.

ബൈനറി സംഖ്യ 101100111 ന്റെ വലതുഭാഗത്ത് നിന്ന് ചുവടെ കാണിച്ചിരിക്കുന്നതുപോലെ കൂട്ടങ്ങളാക്കാം.

101	100	111
↓	↓	↓
5	4	7

$$(101100111)_2 = (547)_8$$

$(10011000011)_2$ നെ ഒക്ടലിലേക്കു മാറ്റുക.

ബൈനറി സംഖ്യ 10011000011 ന്റെ വലതുഭാഗത്ത് നിന്ന് ചുവടെ കാണിച്ചിരിക്കുന്നതു പോലെ കൂട്ടങ്ങളാക്കാം.

കൂട്ടങ്ങളാക്കിയശേഷം ഏറ്റവും ഇടത് ഭാഗത്തെ കൂട്ടത്തിൽ 3 ബിറ്റുകൾ ഇല്ലെങ്കിൽ ആവശ്യമായ 0 ചേർത്ത് 3 ബിറ്റുകൾ ആക്കുക.

010	011	000	011
↓	↓	↓	↓
2	3	0	3

$$(10011000011)_2 = (2303)_8$$

1.6.10 ബൈനറിയിൽ നിന്ന് ഹെക്സാഡെസിമലിലേക്കുള്ള പരിവർത്തനം (Binary to Hexadecimal conversion)

തന്നിരിക്കുന്ന ബൈനറി സംഖ്യ വലത്തു നിന്ന് ഇടത്തേക്ക് 4 ബൈനറി ബിറ്റുകളുടെ കൂട്ടങ്ങളാക്കി അതിന്റെ തത്തുല്യമായ ഹെക്സാഡെസിമൽ അക്കം എഴുതിയാൽ ഒരു ബൈനറി സംഖ്യയെ ഹെക്സാഡെസിമൽ സംഖ്യയിലേക്കു പരിവർത്തനം ചെയ്യാം. നാലിന്റെ കൂട്ടങ്ങൾ ആക്കുമ്പോൾ ഏറ്റവും ഇടത് വശത്തെ കൂട്ടത്തിൽ 4 ബിറ്റുകൾ തികയുന്നില്ലെങ്കിൽ ഇടത് വശത്ത് ആവശ്യമായ പൂജ്യങ്ങൾ കൊടുത്ത് 4 ബിറ്റ് രൂപത്തിൽ ആക്കണം.

ഉദാഹരണങ്ങൾ:

$(101100111010)_2$ നെ ഹെക്സാഡെസിമലിലേക്കു മാറ്റുക.

ബൈനറി സംഖ്യ 101100111010 ന്റെ വലതുഭാഗത്ത് നിന്ന് ചുവടെ കാണിച്ചിരിക്കുന്നതുപ പോലെ കൂട്ടങ്ങളാക്കാം.

1011	0011	1010
↓	↓	↓
B	3	A

$$(101100111010)_2 = (B3A)_{16}$$

$(110111100001100)_2$ നെ ഹെക്സാഡെസിമലിലേക്കു മാറ്റുക.

ബൈനറി സംഖ്യ 110111100001100 ന്റെ വലതുഭാഗത്ത് നിന്ന് ചുവടെ കാണിച്ചിരിക്കുന്നതുപോലെ കൂട്ടങ്ങളാക്കാം.

കൂട്ടങ്ങളാക്കിയശേഷം ഏറ്റവും ഇടത് ഭാഗത്തെ കൂട്ടത്തിൽ 4 ബിറ്റുകൾ ഇല്ലെങ്കിൽ ആവശ്യമായ 0 ചേർത്ത് 4 ബിറ്റുകൾ ആക്കുക.

0110	1111	0000	1100
↓	↓	↓	↓
6	F	0	C

$$(110111100001100)_2 = (6F0C)_{16}$$

1.6.11 ഒക്ടലിൽ നിന്ന് ഹെക്സാഡെസിമലിലേക്കുള്ള പരിവർത്തനം (Octal to hexadecimal conversion)

ഒക്ടൽ സംഖ്യയിൽ നിന്ന് ഹെക്സാഡെസിമൽ സംഖ്യയിലേക്ക് മാറ്റുന്നതിന് രണ്ട് ഘട്ടങ്ങൾ ഉണ്ട്.

ആദ്യം ഒക്ടൽ സംഖ്യ ബൈനറിയായി പരിവർത്തനം ചെയ്യുക. ഈ ബൈനറി സംഖ്യ തത്തുല്യമായ ഹെക്സഡെസിമൽ സംഖ്യയിലേക്കു മാറ്റുക.

ഉദാഹരണം:

$(457)_8$ നെ ഹെക്സാഡെസിമലിലേക്കു മാറ്റുക.

ഘട്ടം 1. $(457)_8$ നെ ബൈനറിയിലേക്കു മാറ്റുക.

$$(457)_8 = \begin{array}{ccc} 4 & 5 & 7 \\ \downarrow & \downarrow & \downarrow \\ 100 & 101 & 111 \end{array}$$

$$= (100101111)_2$$

ഘട്ടം 2. $(100101111)_2$ നെ ഹെക്സാഡെസിമലിലേക്കു മാറ്റുക.

$(100101111)_2$ നെ 4 ബിറ്റുകളുടെ കൂട്ടങ്ങളാക്കി മാറ്റുക.

$$(100101111)_2 = \begin{array}{ccc} 0001 & 0010 & 1111 \\ \downarrow & \downarrow & \downarrow \\ 1 & 2 & F \end{array}$$

$$= (12F)_{16}$$

$$\mathbf{(457)_8 = (12F)_{16}}$$

1.6.12 ഹെക്സാഡെസിമലിൽ നിന്ന് ഒക്ടലിലേക്കുള്ള പരിവർത്തനം (Hexadecimal to Octal conversion)

ഹെക്സാഡെസിമൽ സംഖ്യയിൽ നിന്ന് ഒക്ടൽ സംഖ്യയിലേക്ക് മാറ്റുന്നതിന് രണ്ട് ഘട്ടങ്ങൾ ഉണ്ട്. ആദ്യം ഹെക്സാഡെസിമൽ സംഖ്യ ബൈനറിയായി പരിവർത്തനം ചെയ്യുക. ഈ ബൈനറി സംഖ്യ തത്തുല്യമായ ഒക്ടൽ സംഖ്യയിലേക്കു മാറ്റുക..

ഉദാഹരണം:

$(A2D)_{16}$ നെ ഒക്ടലിലേക്കു മാറ്റുക.

ഘട്ടം 1. $(A2D)_{16}$ നെ ബൈനറിയിലേക്കു മാറ്റുക.

$$(A2D)_{16} = \begin{array}{ccc} A & 2 & D \\ \downarrow & \downarrow & \downarrow \\ 1010 & 0010 & 1101 \end{array}$$

$$= (101000101101)_2$$

ഘട്ടം 2. $(101000101101)_2$ നെ ഒക്ടലിലേക്കു മാറ്റുക.

$(101000101101)_2$ നെ 3 ബിറ്റുകളുടെ കൂട്ടങ്ങളാക്കി മാറ്റുക.

$$(101000101101)_2 = \begin{array}{cccc} 101 & 000 & 101 & 101 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 5 & 0 & 5 & 5 \end{array}$$

$$= (5055)_8$$

$$\mathbf{(A2D)_{16} = (5055)_8}$$

പട്ടിക 1.8: ൽ വിവിധ സംഖ്യ പരിവർത്തനങ്ങളുടെ നടപടിക്രമങ്ങൾ കാണിച്ചിക്കുന്നു.

സംഖ്യ പരിവർത്തനം	നടപടിക്രമം
ദശസംഖ്യയിൽ നിന്ന് ബൈനറിയിലേക്ക്	തുടർച്ചയായി 2 കൊണ്ട് ഹരിച്ച് ശിഷ്ടങ്ങൾ കൂട്ടങ്ങളാക്കുക.
ദശസംഖ്യയിൽ നിന്ന് ഒക്ടലിലേക്ക്	തുടർച്ചയായി 8 കൊണ്ട് ഹരിച്ച് ശിഷ്ടങ്ങൾ കൂട്ടങ്ങളാക്കുക.
ദശസംഖ്യയിൽ നിന്ന് ഹെക്സാഡെസിമലിലേക്ക്	തുടർച്ചയായി 16 കൊണ്ട് ഹരിച്ച് ശിഷ്ടങ്ങൾ കൂട്ടങ്ങളാക്കുക.
ബൈനറിയിൽ നിന്ന് ദശസംഖ്യയിലേക്ക്	ബൈനറി സംഖ്യയിലെ ഓരോ അക്കത്തിന്റെയും സ്ഥാനവില (2 ന്റെ കൃത്യങ്കം) കൊണ്ടു ക്രമമായി ഗുണിച്ച് തുക കാണുക.
ഒക്ടലിൽ നിന്ന് ദശസംഖ്യയിൽലേക്ക്	ഒക്ടൽ സംഖ്യയിലെ ഓരോ അക്കത്തിന്റെയും സ്ഥാനവില (8 ന്റെ കൃത്യങ്കം) കൊണ്ടു ക്രമമായി ഗുണിച്ച് തുക കാണുക.
ഹെക്സാഡെസിമലിൽ നിന്ന് ദശസംഖ്യയിലേക്ക്	ഹെക്സാഡെസിമൽ സംഖ്യയിലെ ഓരോ അക്കത്തിന്റെയും സ്ഥാനവില (16 ന്റെ കൃത്യങ്കം) കൊണ്ടു ക്രമമായി ഗുണിച്ച് തുക കാണുക.
ഒക്ടലിൽ നിന്ന് ബൈനറിയിലേക്ക്	ഓരോ ഒക്ടൽ അക്കവും 3 ബിറ്റ് ബൈനറി സംഖ്യ ആയി പരിവർത്തനം ചെയ്യുക.
ഹെക്സാഡെസിമലിൽ നിന്ന് ബൈനറിയിലേക്ക്	ഓരോ ഹെക്സാഡെസിമൽ അക്കവും 4 ബിറ്റ് ബൈനറി സംഖ്യ ആയി പരിവർത്തനം ചെയ്യുക.
ബൈനറിയിൽ നിന്ന് ഒക്ടലിലേക്ക്	ബൈനറി സംഖ്യ വലത്തു നിന്ന് ഇടത്തേക്ക് 3 ബൈനറി ബിറ്റുകളുടെ കൂട്ടങ്ങളാക്കി അതിന്റെ തുല്യമായ ഒക്ടൽ അക്കം എഴുതുക.
ബൈനറിയിൽ നിന്ന്	ബൈനറി സംഖ്യ വലത്തു നിന്ന് ഇടത്തേക്ക് 4 ബൈനറി ബിറ്റുകളുടെ
ഹെക്സാഡെസിമലിലേക്ക്	കൂട്ടങ്ങളാക്കി അതിന്റെ തുല്യമായ ഹെക്സാ ഡെസിമൽ അക്കം എഴുതുക.
ഒക്ടലിൽ നിന്ന്	ഒക്ടലിനെ ബൈനറിയിലേക്കും തുടർന്ന് ബൈനറിയിൽ നിന്ന് ഹെക്സാ
ഹെക്സാഡെസിമലിലേക്ക്	ഡെസിമലിലേക്കും മാറ്റുക.
ഹെക്സാഡെസിമലിൽ നിന്ന് ഒക്ടലിലേക്ക്	ഹെക്സാഡെസിമലിനെ ബൈനറിയിലേക്കും തുടർന്ന് ബൈനറിയിൽ നിന്ന് ഒക്ടലിലേക്കും മാറ്റുക.

പട്ടിക 1.8: വിവിധ സംഖ്യാ പരിവർത്തനങ്ങളുടെ നടപടിക്രമങ്ങൾ

സ്വയം വിലയിരുത്താം

1. 31 എന്ന ദശസംഖ്യ ബൈനറിയിലേക്കു മാറ്റുക.
2. $(10001)_2$ നു തത്തുല്യമായ ദശസംഖ്യ കണ്ടുപിടിക്കുക.
3. $(x)_8 = (101011)_2$, ആയാൽ x ന്റെ വില കാണുക.
4. വിട്ട ഭാഗം പൂരിപ്പിക്കുക.
   a) $(______)_2 = (AB)_{16}$
   b) $(__D__)_{16} = (1010____1000)_2$
   c) $0.25_{10} = (______)_2$
5. താഴെ കൊടുത്തിരിക്കുന്ന സംഖ്യകളിൽ ഏറ്റവും വലിയ സംഖ്യ കണ്ടുപിടിക്കുക.
   (i) $(1001)_2$ (ii) $(A)_{16}$ (iii) $(10)_8$ (iv) $(11)_{10}$

1.7 ബൈനറി സംഖ്യകളുടെ സങ്കലനം (Binary addition)

ദശസംഖ്യാന സമ്പ്രദായത്തിലുള്ളത് പോലെ ദ്വയസംഖ്യാന സമ്പ്രദായത്തിലും ഗണിത ക്രിയകൾ ചെയ്യാം. നമ്മൾ രണ്ട് ദശസംഖ്യകൾ കമ്പ്യൂട്ടറിൽ സങ്കലനം ചെയ്യാൻ നിർദേശം നൽകുമ്പോൾ, കമ്പ്യൂട്ടർ അതിന്റെ തുല്യമായ ബൈനറി സംഖ്യകൾ ആണ് കൂട്ടുന്നത്. ബൈനറി സംഖ്യകളുടെ സങ്കലനവും വ്യവകലനവും എങ്ങനെയാണ് ചെയ്യുന്നത് എന്ന് നമുക്ക് നോക്കാം.

രണ്ട് ബിറ്റുകൾ കൂട്ടുവാനുള്ള നിയമങ്ങൾ താഴെ കൊടുത്തിരിക്കുന്നു.

A	B	തുക	ശിഷ്ടം
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

പട്ടിക 1.9 ബൈനറി സംഖ്യകളുടെ സങ്കലന നിയമങ്ങൾ

ഒന്നും ഒന്നും കൂട്ടുമ്പോൾ മാത്രമാണ് ശിഷ്ട (ക്യാരി) ബിറ്റ് 1 ഉണ്ടാകുന്നത് എന്ന് ശ്രദ്ധിക്കുക. മൂന്നു ഒന്നുകൾ കൂട്ടുമ്പോൾ (1+1+1) തുക 1 ഉം ശിഷ്ടം (ക്യാരി) ബിറ്റ് 1 ഉം കിട്ടുന്നു.

ഉദാഹരണങ്ങൾ:

ബൈനറി സംഖ്യകളായ 1011 ന്റെയും 1001 ന്റെയും തുക കണ്ടുപിടിക്കുക.

```
  1 0 1 1 +
  1 0 0 1
1 0 1 0 0
```

ബൈനറി സംഖ്യകളായ 110111 ന്റെയും 100110 ന്റെയും തുക കണ്ടുപിടിക്കുക.

```
  1 1 0 1 1 1 +
  1 0 0 1 1 0
1 0 1 1 1 0 1
```

1.8 ഡാറ്റയുടെ പ്രതിനിധാനം (Data representation)

സംഖ്യകൾ അക്ഷരങ്ങൾ ചിത്രങ്ങൾ ശബ്ദങ്ങൾ വീഡിയോകൾ എന്നിങ്ങനെ വ്യത്യസ്ത തരത്തിലുള്ള ഡാറ്റയെ കമ്പ്യൂട്ടർ പ്രോസസ് ചെയ്യുന്നു. വൈദ്യുതിയുടെ രണ്ട് അവസ്ഥകളായ ഓൺ (ON), ഓഫ് (OFF) എന്നിവയെ അടിസ്ഥാനമാക്കി പ്രവർത്തിക്കുന്ന ഒരു ഇലക്ട്രോണിക് ഉപകരണമാണ് കമ്പ്യൂട്ടർ എന്ന് നമുക്ക് അറിയാം. എല്ലാ ഇലക്ട്രോണിക് സർക്യൂട്ടുകൾക്കും തുറന്നിരിക്കുന്നതും അടഞ്ഞിരിക്കുന്നതുമായ രണ്ട് അവസ്ഥകളാണ് ഉള്ളത്. തുറന്നിരിക്കുന്ന അവസ്ഥയെ സൂചിപ്പിക്കാനായി ഓഫ് (OFF) അല്ലെങ്കിൽ പൂജ്യവും അടഞ്ഞിരിക്കുന്ന അവസ്ഥയെ സൂചിപ്പിക്കാനായി ഓൺ (ON) അല്ലങ്കിൽ ഒന്നും ഉപയോഗിക്കുന്നു. ഈ രണ്ട് അവസ്ഥയിലുള്ള പ്രവർത്തനത്തെ ബൈനറി ഓപ്പറേഷൻ എന്ന് വിളിക്കുന്നു. അതുകൊണ്ടു കമ്പ്യൂട്ടറിലേക്ക് നൽകുന്ന ഡാറ്റയും ബൈനറി രൂപത്തിലായിരിക്കണം. കമ്പ്യൂട്ടറിൽ ആന്തരികമായി ഒരു ഡാറ്റയെ പ്രതിനിധാനം ചെയ്യാൻ ഉപയോഗിക്കുന്ന രീതിയെ ഡാറ്റ പ്രതിനിധാനം എന്നു പറയുന്നു. കമ്പ്യൂട്ടർ ഡാറ്റയുടെ ഒരു ഭാഗം പ്രതിനിധാനം ചെയ്യാൻ നിശ്ചിത എണ്ണം ബിറ്റുകൾ ഉപയോഗിക്കുന്നു. അവ ഒരു സംഖ്യ, ഒരക്ഷരം, ചിത്രം, ശബ്ദം, വീഡിയോ മുതലായവയാകാം. കമ്പ്യൂട്ടർ മെമ്മറിയിൽ എങ്ങനെയാണ് വ്യത്യസ്ത ഡേറ്റകളെ പ്രതിനിധാനം ചെയ്യുക എന്നു നമുക്ക് നോക്കാം.

ഡാറ്റയുടെ ബാഹ്യവും ആന്തരികവുമായ രൂപങ്ങൾ

1.8.1 സംഖ്യകളുടെ പ്രതിനിധാനം (Representation of numbers)

സംഖ്യകളെ പൂർണസംഖ്യകൾ, ദശാംശസംഖ്യകൾ എന്നിങ്ങനെ രണ്ടായി തിരിക്കാം. പൂർണസംഖ്യകൾ ഭിന്നസംഖ്യാ ഭാഗം ഇല്ലാത്ത സംഖ്യകൾ ആകുന്നു. ദശാംശസംഖ്യ (Floating Point Number) അല്ലെങ്കിൽ രേഖീയസംഖ്യ ഭിന്നകഭാഗത്തോട് കൂടിയ സംഖ്യ ആകുന്നു. ഈ രണ്ടു സംഖ്യകളെയും കമ്പ്യൂട്ടറിന്റെ മെമ്മറിയിൽ വ്യത്യസ്തമായിട്ടാണ് കൈകാര്യം ചെയ്യുന്നത്. പൂർണസംഖ്യകൾ എങ്ങനെയാണ് മെമ്മറിയിൽ പ്രതിനിധാനം ചെയ്യുന്നത് എന്ന് നമുക്ക് നോക്കാം.

എ. പൂർണസംഖ്യകളുടെ പ്രതിനിധാനം (Representation of integers)

ഒരു പൂർണ സംഖ്യ കമ്പ്യൂട്ടറിന്റെ മെമ്മറിയിൽ പ്രതിനിധീകരിക്കുന്നത് മൂന്ന് രീതിയിലാണ്.

i) ചിഹ്നവും മൂല്യവും കൊണ്ടുള്ള പ്രതിനിധാനം (Sign and magnitude representation)

ii) 1 ന്റെ പൂരകം കൊണ്ടുള്ള പ്രതിനിധാനം (1's complement representation)

iii) 2 ന്റെ പൂരകം കൊണ്ടുള്ള പ്രതിനിധാനം (2's complement representation)

കമ്പ്യൂട്ടർ പ്രോസസ്സർ ഒരു യൂണിറ്റായി കൈകാര്യം ചെയ്യുന്ന നിശ്ചിത വ്യാപ്തിയിലുള്ള ഒരു കൂട്ടം ബിറ്റുകളെയാണ് പദം (Word) എന്ന് പറയുന്നത്. ഒരു പദത്തിലെ ബിറ്റുകളുടെ എണ്ണത്തെ പദദൈർഘ്യം (Word length) എന്ന് പറയുന്നു. കമ്പ്യൂട്ടർ രൂപകൽപ്പന ചെയ്യുന്ന വിദഗ്ധരാണ് അതിന്റെ പദദൈർഘ്യം തീരുമാനിക്കുന്നത്. 8, 16, 32, 64 എന്നിവ സാധാരണയായി നിലവിലുള്ള ചില പദദൈർഘ്യങ്ങളാണ്. പദങ്ങൾ ബിറ്റുകളുടെ കൂട്ടമായതുകൊണ്ട് പദദൈർഘ്യം രണ്ടിന്റെ കൃത്യങ്കങ്ങൾ ആയിരിക്കും.

ഇനി ഡാറ്റയെ പ്രതിനിധാനം ചെയ്യുന്ന രീതികൾ (8 ബിറ്റ് പദദൈർഘ്യത്തെ അടിസ്ഥാനമാക്കി) വിശദമായി പരിശോധിക്കാം.

i. ചിഹ്നവും മൂല്യവും കൊണ്ടുള്ള പ്രതിനിധാനം (Sign and magnitude representation)

ഈ രീതിയിൽ, ഇടതുഭാഗത്തെ ആദ്യത്തെ ബിറ്റ് (MSB) പൂർണസംഖ്യയുടെ ചിഹ്നത്തെയും ബാക്കിയുള്ള 7 ബിറ്റുകൾ സംഖ്യയുടെ മൂല്യത്തെയും പ്രതിനിധാനം ചെയ്യുന്നു. ചിഹ്നത്തെ പ്രതിനിധാനം ചെയ്യുന്ന ബിറ്റ് 1 ആണെങ്കിൽ അത് നെഗറ്റീവ് പൂർണസംഖ്യയും 0 ആണെങ്കിൽ പോസിറ്റീവ് പൂർണസംഖ്യയുമായിരിയ്ക്കും.

ഉദാഹരണങ്ങൾ:

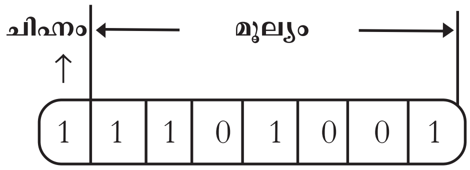
+ 23 നെ ചിഹ്നവും മൂല്യവും ഉപയോഗിച്ച് പ്രതിനിധാനം ചെയ്യുക.

സംഖ്യ പോസിറ്റീവ് ആയതിനാൽ ഒന്നാമത്തെ ബിറ്റ് (MSB) 0 ആകുന്നു.

23 ന് തുല്യമായ 7 ബിറ്റ് ബൈനറി സംഖ്യ= $(0010111)_2$

അതുകൊണ്ട് +23 നെ $(00010111)_2$ കൊണ്ട് പ്രതിനിധീകരിക്കാം.

-105 നെ ചിഹ്നവും മൂല്യവും രൂപത്തിൽ പ്രതിനിധാനം ചെയ്യുക



സംഖ്യ നെഗറ്റീവ് ആയതിനാൽ ഒന്നാമത്തെ ബിറ്റ് (MSB) 1 ആകുന്നു.

7 ബിറ്റ് ബൈനറി സംഖ്യ 105 = $(1101001)_2$

-105 ന് തുല്യമായ 7 ബിറ്റ് ബൈനറി സംഖ്യ = $(11101001)_2$

അതിനാൽ -105 നെ $(11101001)_2$ കൊണ്ട് പ്രതിനിധീകരിക്കാം

കുറിപ്പ് : ഈ രീതിയിൽ 8 ബിറ്റ് പദം കൊണ്ട് 2^8-1 =255 സംഖ്യകൾ പ്രതിനിധാനം ചെയ്യാൻ കഴിയുന്നു. സംഖ്യകൾ $-(2^7 - 1)$ മുതൽ $+(2^7 - 1)$ വരെ ആയിരിക്കും. (അതായത് -127 മുതൽ +127 വരെ). അതുപോലെ 16 ബിറ്റ് പദം കൊണ്ട് $2^{16} - 1 = 65535$ സംഖ്യകൾ പ്രതിനിധാനം ചെയ്യാനും കഴിയുന്നു (അതായത് -32767 മുതൽ +32767 വരെ). പൊതുവായി, *n* ബിറ്റ് പദം കൊണ്ട് 2^n-1 സംഖ്യകൾ പ്രതിനിധാനം ചെയ്യാൻ കഴിയും (അതായത് $-(2^{n-1}-1)$ മുതൽ $+(2^{n-1}-1)$. വരെ). പൂർണസംഖ്യയായ പൂജ്യത്തെ +0 = 00000000 എന്നും 0 = 10000000 എന്നും രണ്ട് രീതിയിൽ പ്രതിനിധാനം ചെയ്യാം.

ii. 1 ന്റെ പൂരകം കൊണ്ടുള്ള പ്രതിനിധാനം (1's complement representation)

ഈ രീതിയിൽ, പൂർണസംഖ്യയുടെ കേവല വിലയ്ക്ക് തത്തുല്യമായ 8 ബിറ്റ് ബൈനറി സംഖ്യ കണ്ടുപിടിക്കുന്നു. ബൈനറി സംഖ്യയ്ക്ക് 8 ബിറ്റുകൾ ഇല്ലെങ്കിൽ ഇടതുവശത്ത് ആവശ്യമായ പൂജ്യം ചേർത്ത് 8 ബിറ്റ് സംഖ്യ ആക്കുക. സംഖ്യയിലെ ഓരോ പൂജ്യത്തിനു പകരം ഒന്ന് എന്നും ഓരോ ഒന്നിന് പകരം പൂജ്യം എന്നും മാറ്റി എഴുതിയാൽ ആ സംഖ്യയുടെ 1 ന്റെ പൂരകം ലഭിക്കും. ചില ബൈനറി സംഖ്യകളും അവയുടെ 1ന്റെ പൂരക പ്രതിനിധാനങ്ങളും താഴെ കൊടുത്തിരിക്കുന്നു.

പൂർണസംഖ്യ	ബൈനറി സംഖ്യ	1 ന്റെ പൂരക പ്രതിനിധാനം
+25	00011001	00011001
- 25	00011001	11100110

സംഖ്യ നെഗറ്റീവ് ആണെങ്കിൽ അതിന്റെ തത്തുല്യമായ 8 ബിറ്റ് ബൈനറി രൂപത്തെ 1 ന്റെ പൂരകമായി പ്രതിനിധീകരിക്കുന്നു. എന്നാൽ സംഖ്യ പോസിറ്റീവ് ആണെങ്കിൽ സംഖ്യയുടെ 8 ബിറ്റ് പ്രതിനിധാനവും 1ന്റെ പൂരക പ്രതിനിധാനവും ഒരു പോലെയായിരിക്കും.

ഉദാഹരണങ്ങൾ: -

119 നെ 1ന്റെ പൂരക രൂപത്തിൽ പ്രതിനിധാനം ചെയ്യുക.

119 ന്റെ 8 ബിറ്റ് ബൈനറി രൂപം = $(01110111)_2$

-119 ന്റെ 1ന്റെ പൂരക പ്രതിനിധാന രൂപം = $(10001000)_2$

+119 നെ 1ന്റെ പൂരക രൂപത്തിൽ പ്രതിനിധാനം ചെയ്യുക

119 ന്റെ 8 ബിറ്റിൽ ഉള്ള ബൈനറി രൂപം = $(01110111)_2$

+119 ന്റെ 1ന്റെ പൂരക പ്രതിനിധാന രൂപം = $(01110111)_2$

(സംഖ്യ പോസിറ്റീവ് ആയതിനാൽ 1 ന്റെ പൂരക പ്രതിനിധാനം കണ്ടുപിടിക്കേണ്ടതില്ല)

കുറിപ്പ് : ഇത്തരം പ്രതിനിധീകരണത്തിൽ ഒന്നാമത്തെ ബിറ്റ് (MSB) 0 ആണെങ്കിൽ സംഖ്യ പോസിറ്റീവും MSB 1 ആണെങ്കിൽ സംഖ്യ നെഗറ്റീവും ആയിരിക്കും. 8 ബിറ്റ് പദദൈർഘ്യം കൊണ്ട് -127 (10000000) മുതൽ +127 (01111111) വരെ പ്രതിനിധാനം ചെയ്യാൻ കഴിയുന്നു. ഈ സംവിധാനത്തിലൂടെ പൂജ്യത്തിനെ +0 = 00000000 എന്നും -0 = 11111111 എന്നും രണ്ട് രീതിയിൽ പ്രതിനിധാനം ചെയ്യാം. പൊതുവായി, *n* ബിറ്റ് പദം കൊണ്ട് 2^n-1 സംഖ്യകൾ പ്രതിനിധാനം ചെയ്യാൻ കഴിയും (അതായത് - $(2^{n-1} - 1)$ മുതൽ $+(2^{n-1} - 1)$ വരെ.

iii. 2 ന്റെ പൂരകം കൊണ്ടുള്ള പ്രതിനിധാനം (2’s complement representation)

ഈ രീതിയിൽ, പൂർണസംഖ്യയുടെ കേവലവിലയ്ക്ക് തത്തുല്യമായ 8 ബിറ്റ് ബൈനറി സംഖ്യ കണ്ടുപിടിക്കുന്നു. സംഖ്യ നെഗറ്റീവ് ആണെങ്കിൽ 8 ബിറ്റ് ബൈനറിയുടെ 2 ന്റെ പൂരകരൂപത്തിൽ അതിനെ പ്രതിനിധാനം ചെയ്യുന്നു. എന്നാൽ സംഖ്യ പോസിറ്റീവ് ആണെങ്കിൽ 8 ബിറ്റ് ബൈനറി സംഖ്യ തന്നെയാണ് അതിന്റെ 2 ന്റെ പൂരക പ്രതിനിധാനം. ഒരു ബൈനറി സംഖ്യയുടെ 1 ന്റെ പൂരകത്തോട് 1 കൂട്ടിയാൽ അതിന്റെ 2 ന്റെ പൂരകം കിട്ടുന്നു.

ഉദാഹരണമായി നമുക്ക് $(10101)_2$ ന്റെ 2 ന്റെ പൂരകം കണ്ടുപിടിക്കാം.

$(00010101)_2$ ന്റെ 1 ന്റെ പൂരകം = $(11101010)_2$

$(10101)_2$ ന്റെ 2 ന്റെ പൂരകം = 11101010 +

1

= $(11011010)_2$

ഉദാഹരണങ്ങൾ:

-38 നെ 2ന്റെ പൂരക രൂപത്തിൽ പ്രതിനിധാനം ചെയ്യുക.

38 ന്റെ 8 ബിറ്റിലുള്ള ബൈനറി രൂപം = $(00100110)_2$

-38 ന്റെ 2 ന്റെ പൂരക പ്രതിനിധാനം = 11011001+
1
= $(11011010)_2$

+38 നെ 2ന്റെ പൂരക രൂപത്തിൽ പ്രതിനിധാനം ചെയ്യുക..

38 ന്റെ 8 ബിറ്റിലുള്ള ബൈനറി രൂപം = $(00100110)_2$

+38 ന്റെ 2 ന്റെ പൂരക പ്രതിനിധാനം = $(00100110)_2$ (സംഖ്യ പോസിറ്റീവ് ആയതിനാൽ 2ന്റെ പൂരക പ്രതിനിധാനം കണ്ടുപിടിക്കേണ്ടതില്ല)

കുറിപ്പ് : ഈ രീതിയിൽ ഒന്നാമത്തെ ബിറ്റ് (MSB) 0 ആണെങ്കിൽ സംഖ്യ പോസിറ്റീവും MSB 1 ആണെങ്കിൽ സംഖ്യ നെഗറ്റീവും ആയിരിക്കും. ഇവിടെ പൂജ്യം എന്ന പൂർണസംഖ്യ 00000000 എന്ന രീതിയിൽ മാത്രമേ പ്രതിനിധീകരിക്കുവാൻ കഴിയൂ. 8 ബിറ്റ് പദം കൊണ്ട് -128 (10000000) മുതൽ +127 (01111111) വരെ പ്രതിനിധാനം ചെയ്യാൻ കഴിയുന്നു. പൊതുവായി, *n* ബിറ്റ് പദം കൊണ്ട് 2^n സംഖ്യകൾ പ്രതിനിധാനം ചെയ്യുവാൻ കഴിയും. സംഖ്യകൾ $-(2^{n-1})$ മുതൽ $+(2^{n-1}-1)$. വരെ ആകുന്നു. ഈ രീതിയാണ് പൂർണസംഖ്യ പ്രതിനിധീകരിക്കുന്നതിന് സർവസാധാരണമായി ഉപയോഗിക്കുന്നത്. പട്ടിക 1.9 ൽ പൂർണസംഖ്യകളെ 8 ബിറ്റ് പദദൈർഘ്യത്തിൽ പ്രതിനിധീകരിക്കുന്നതിനുള്ള വിവിധ രീതികൾ താരതമ്യം ചെയ്യുന്നു.

സവിശേഷത	ചിഹ്നവും മൂല്യവും	1 ന്റെ പൂരകം	2 ന്റെ പൂരകം	കുറിപ്പ്
പരിധി	-127 മുതൽ +127 വരെ	-127 മുതൽ +127 വരെ	-128 മുതൽ +127 വരെ	2 ന്റെ പൂരകത്തിൽ പരിധി കൂടുതലാണ്
ആകെ സംഖ്യകൾ	255	255	256	
0 ന്റെ പ്രതിനിധാനം	2 രീതിയിലുള്ള പ്രതിനിധാനം	2 രീതിയിലുള്ള പ്രതിനിധാനം	ഒരേയൊരു രീതിയിലു ള്ള പ്രതിനിധാനം	പൂജ്യത്തെ 2 ന്റെ പൂരകത്തിൽ പ്രതിനിധീകരിക്കുന്നതിന് ഒരു അവ്യക്തതയും ഇല്ല.
പോസിറ്റീവ് സംഖ്യകളുടെ പ്രതിനിധാനം	സംഖ്യയ്ക്ക് തുല്യമായ 8 ബിറ്റ് ബൈനറി രൂപം	സംഖ്യയ്ക്ക് തുല്യമായ 8 ബിറ്റ് ബൈനറി രൂപം	സംഖ്യയ്ക്ക് തുല്യമായ 8 ബിറ്റ് ബൈനറി രൂപം	മൂന്നു രീതിയിലും ഒരേ പോലെയാണ്
നെഗറ്റീവ് സംഖ്യകളുടെ പ്രതിനിധാനം	ചിഹ്നം 1 ബിറ്റിലും മൂല്യം 7 ബിറ്റ് ബൈനറി രൂപത്തിലും പ്രതിനിധീകരിക്കുന്നു	8 ബിറ്റ് ബൈനറി രൂപത്തിലാക്കിയ ശേഷം അതിന്റെ 1 ന്റെ പൂരകം കാണുന്നു.	8 ബിറ്റ് ബൈനറി രൂപത്തിലാക്കിയശേഷം അതിന്റെ 2 ന്റെ പൂരകം കാണുന്നു.	എല്ലാ നെഗറ്റീവ് സംഖ്യകുളുടെയും MSB 1 ആകുന്നു

പട്ടിക 1.10 ൽ പൂർണ സംഖ്യകളുടെ 8 ബിറ്റ് പദദൈർഘ്യത്തിലുള്ള വിവിധ പ്രതിനിധാനങ്ങളുടെ താരതമ്യം

താഴെ കൊടുത്തിരിക്കുന്ന പട്ടികയിൽ 4 ബിറ്റ് പദദൈർഘ്യം ഉപയോഗിച്ച് പൂർണ സംഖ്യകളുടെ 3 രീതിയിലുള്ള പ്രതിനിധാനങ്ങൾ വിശദീകരിച്ചിരിക്കുന്നു.

സംഖ്യ	ചിഹ്നവും മൂല്യവും	1 ന്റെ പൂരകം	2 ന്റെ പൂരകം
-8	സാധ്യമല്ല	സാധ്യമല്ല	1000
-7	1111	1000	1001
-6	1110	1001	1010
-5	1101	1010	1011
-4	1100	1011	1100
-3	1011	1100	1101
-2	1010	1101	1110
-1	1001	1110	1111
0	1000 അല്ലെങ്കിൽ 0000	0000 അല്ലെങ്കിൽ 1111	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	0101
6	0110	0110	0110
7	0111	0111	0111

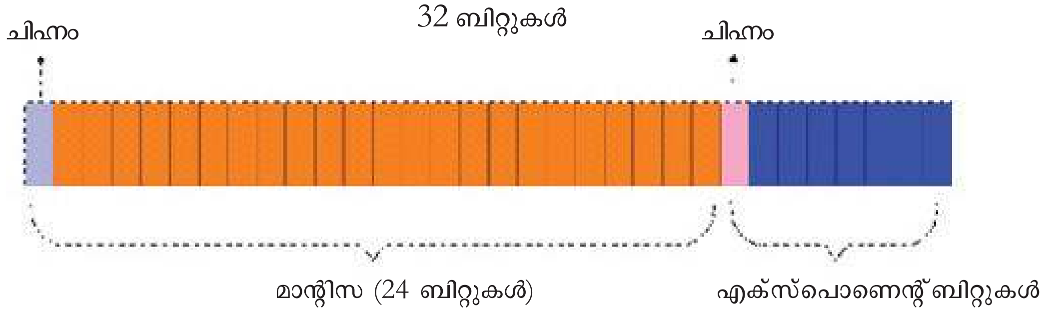
പൂർണ സംഖ്യകളുടെ മൂന്നു രീതികളിലുള്ള പ്രതിനിധാനത്തിലും MSB സംഖ്യയുടെ ചിഹ്നം സൂചിപ്പിക്കുന്നു. ബിറ്റ് 1 ആണെങ്കിൽ സംഖ്യ നെഗറ്റീവും ബിറ്റ് 0 ആണെങ്കിൽ സംഖ്യ പോസിറ്റീവും ആണ്. തന്നിരിക്കുന്ന പദദൈർഘ്യം കൊണ്ട് സംഖ്യകളെ ഏറ്റവും കൂടുതൽ പ്രതിനിധീകരിക്കുവാൻ സാധിക്കുന്നത് 2 ന്റെ പൂരക രീതിയിലാണെന്ന് പട്ടികയിൽ കാണുന്നു. 4 പദദൈർഘ്യം ഉപയോഗിച്ചാൽ 7 നെക്കാൾ ചെറുതും +7 നെക്കാൾ വലുതും ആയ സംഖ്യകൾ പ്രതിനിധീകരിക്കാൻ ചിഹ്നവും മൂല്യവും രീതിയിലും 1 ന്റെ പൂരക രീതിയിലും സാധ്യമല്ല. അതുകൊണ്ട് 8 ബിറ്റ് പദദൈർഘ്യമുള്ള പ്രതിനിധാനം ഉപയോഗിക്കുന്നു. അതുപോലെ 2 ന്റെ പൂരക പ്രതിനിധാന രീതിയിൽ - 8 മുതൽ +7 പരിധിക്ക് പുറത്തുള്ള സംഖ്യകൾ കൈകാര്യം ചെയ്യുന്നതിനായി 8 ബിറ്റ് ആവശ്യമാണ്.

8 ബിറ്റ് പദദൈർഘ്യം ഉപയോഗരീതിയിൽ - 128 മുതൽ +127 വരെയുള്ള സംഖ്യകൾ 2 ന്റെ പൂരക രീതിയിൽ പ്രതിനിധീകരിക്കാം. എന്നാൽ മറ്റു രണ്ടു രീതികളായ 1 ന്റെ പൂരകത്തിലും, ചിഹ്നവും മൂല്യത്തിലും - 127 മുതൽ +127 വരെ പരിധിയുള്ള സംഖ്യകൾ പ്രതിനിധാനം ചെയ്യാൻ സാധിക്കുകയുള്ളൂ. മേൽപ്പറഞ്ഞ പരിധിക്ക് പുറത്തുള്ള സംഖ്യകൾ പ്രതിനിധാനം ചെയ്യാൻ നമ്മൾ 16 ബിറ്റ് ഉപയോഗിക്കുന്നു.

ബി. ഫ്ളോട്ടിംഗ് പോയിന്റ് സംഖ്യകളുടെ പ്രതിനിധാനം (Representation of floating point numbers)

ഒരു ഫ്ളോട്ടിംഗ് പോയിന്റ് സംഖ്യ അല്ലെങ്കിൽ രേഖീയ സംഖ്യയിൽ പൂർണസംഖ്യാഭാഗവും ഭിന്നക ഭാഗവും അടങ്ങിയിട്ടുണ്ട്. ഒരു രേഖീയ സംഖ്യയെ ഫ്ളോട്ടിംഗ് പോയിന്റ് എന്ന സവിശേഷമായ ചിഹ്നസമ്പ്രദായം ഉപയോഗിച്ച് എഴുതാവുന്നതാണ്. ഈ ചിഹ്നസമ്പ്രദായം ഉപയോഗിച്ച് എഴുതുമ്പോൾ ഏത് സംഖ്യയ്ക്കും മാന്റിസ, എക്സ്പൊണെന്റ് എന്നീ രണ്ട് ഭാഗങ്ങൾ ഉണ്ടാകും. ഉദാഹരണമായി 25.45 നെ $0.2545x10^2$ എന്നെഴുതാം. ഇതിൽ 0.2545 എന്നത് മാന്റിസയും കൃത്യങ്കം 2 എന്നത് എക്സ്പൊണെന്റുമാണ്. (ക്രമാനുസൃതമായ (Normalised) ഫ്ളോട്ടിംഗ് പോയിന്റ് പ്രതിനിധാനത്തിൽ മാന്റിസ 0.1നും 1നും ഇടയിലായിരിക്കും). അതുപോലെ 0.0035 എന്ന സംഖ്യ $-0.35x10^{-2}$ എന്ന് എഴുതാം. ഇവിടെ -0.35 എന്നത് മാന്റിസയും കൃത്യങ്കം -2 എന്നത് എക്സ്പൊണെന്റുമാണ്.

32 ബിറ്റ് പദദൈർഘ്യമുള്ള കമ്പ്യൂട്ടറിൽ ഒരു രേഖീയ സംഖ്യ എങ്ങനെയാണ് പ്രതിനിധാനം ചെയ്യുന്നതെന്ന് നോക്കാം. ചിത്രം 1.10 ൽ കാണുന്നതുപോലെ, ഇതിൽ 24 ബിറ്റുകൾ മാന്റിസ രേഖപ്പെടുത്താനും (അതിൽ ആദ്യത്തെ ബിറ്റ് മാന്റിസയുടെ ചിഹ്നത്തിനുവേണ്ടിയാണ്), 8



ചിത്രം 1.10 ഫ്ളോട്ടിങ് പോയിന്റ് സംഖ്യകളുടെ പ്രതിനിധാനം

ബിറ്റുകൾ എക്സ്പൊണെന്റ് രേഖപ്പെടുത്താനും (അതിൽ ആദ്യത്തെ ബിറ്റ് എക്സ്പൊണെന്റിന്റെ ചിഹ്നത്തിനുവേണ്ടി) ഉപയോഗിക്കുന്നു. ദശാംശബിന്ദു മാന്റിസയുടെ ചിഹ്നം സൂചിപ്പിക്കുന്ന ബിറ്റിന്റെ വലത് ഭാഗത്താണെന്ന് അനുമാനിക്കുക. ദശാംശസ്ഥാനം സാങ്കല്പികമായതിനാൽ അത് രേഖപ്പെടുത്താൻ പ്രത്യേകമായി ബിറ്റുകൾ ആവശ്യമില്ല.

രേഖീയസംഖ്യകളിൽ ബൈനറി അംശബിന്ദു മാന്റിസ, എക്സ്പൊണെന്റ് ഭാഗങ്ങളുടെ വിവരങ്ങൾ സൂക്ഷിക്കുന്നു. ബൈനറി അംശബിന്ദുവിന്റെ സ്ഥാനം സ്ഥിരമല്ലാത്തതിനാൽ മാന്റിസ എക്സ്പൊണെന്റ് എന്നിവയുടെ വിലകൾ സംഖ്യകൾ തോറും മാറുന്നു. മറ്റൊരു വിധത്തിൽപറഞ്ഞാൽ അത് ഫ്ളോട്ട് ചെയ്യുകയാണ് (വെള്ളത്തിൽ പൊങ്ങിക്കിടക്കുന്നതു പോലെ) അതിനാൽ ഈ പ്രതിനിധാനത്തെ ഫ്ളോട്ടിംഗ് പോയിന്റ് പ്രതിനിധാനം എന്ന് അറിയപ്പെടുന്നു.

ഉദാഹരണമായി 25.45 എന്ന രേഖീയ സംഖ്യ മാന്റിസ എക്സ്പൊണെന്റ് രീതിയിൽ 0.2545 X 10^2 എന്ന് എഴുതാം. ഇവിടെ മാന്റിസയായ 0.2545 നെയും എക്സ്പൊണെന്റായ 2 നെയും ബൈനറി രൂപത്തിലേക്കു മാറ്റി അവയെ അതാതു സ്ഥാനങ്ങളിൽ രേഖപ്പെടുത്തുന്നു. മാന്റിസയും എക്സ്പൊണെന്റും രേഖപ്പെടുത്താൻ വ്യത്യസ്തങ്ങളായ മാനദണ്ഡങ്ങൾ ഉപയോഗിക്കുന്നു. പദദൈർഘ്യം മാറുന്നതിനനുസരിച്ച് മാന്റിസയും എക്സ്പൊണെന്റും രേഖപ്പെടുത്താൻ ഉപയോഗിക്കുന്ന ബിറ്റുകളുടെ ।എണ്ണത്തിലും മാറ്റം ഉണ്ടാകും.

1.8.2 അക്ഷരങ്ങളുടെ പ്രതിനിധാനം (Representation of characters)

കംപ്യൂട്ടറിന്റെ മെമ്മറിയിൽ സംഖ്യകൾ പ്രതിനിധാനം ചെയ്യുന്നത് എങ്ങനെയെന്ന് നമ്മൾ കണ്ടു. അതുപോലെ അക്ഷരങ്ങളെ (Characters) പ്രതിനിധാനം ചെയ്യുന്നതിന് വ്യത്യസ്തങ്ങളായ സമ്പ്രദായങ്ങളുണ്ട്. അവയിൽ ചിലതിനെക്കുറിച്ച് ചുവടെ പ്രതിപാദിക്കുന്നു.

എ. ആസ്കി (ASCII)

കമ്പ്യൂട്ടറിന്റെ മെമ്മറിയിൽ 7 ബിറ്റുകൾ ഉപയോഗിച്ച് ഓരോ അക്ഷരവും പ്രതിനിധാനം ചെയ്യാൻ ഉപയോഗിക്കുന്ന ASCII (ആസ്കി) കോഡ് Amercican Standard Code for Information Interchange (അമേരിക്കൻ സ്റ്റാൻഡേർഡ് കോഡ് ഫോർ ഇൻഫർമേഷൻ ഇന്റർചേഞ്ച്) എന്നതിന്റെ ചുരുക്കരൂപമാണ്. അമേരിക്കൻ സർക്കാർ അംഗീകരിച്ച ആസ്കികോഡ് വ്യാപകമായി സ്വീകരിക്കപ്പെട്ടു കഴിഞ്ഞു. ഇതിൽ ഓരോ അക്ഷരത്തിനും വത്യസ്ത പൂർണസംഖ്യ നിശ്ചയിച്ചിരിക്കുന്നു. ആസ്കി കോഡ് എന്ന് വിളിക്കുന്ന ഈ പൂർണസംഖ്യ മെമ്മറിയിൽ സൂക്ഷിക്കുന്നതിനായി ബൈനറി സംഖ്യയിലേക്ക് പരിവർത്തനം ചെയ്യുന്നു. ഉദാഹരണത്തിന് A എന്ന അക്ഷരത്തിന്റെ ആസ്കി കോഡ് 65 ആകുന്നു. ഇതിന് തുല്യമായ 7 ബിറ്റ് ബൈനറി 1000001 ആണ്. 7 ബിറ്റുകൾ കൊണ്ട് വത്യസ്തങ്ങളായ 128 സംയോഗങ്ങൾ സൃഷ്ടിക്കാനാകും. ആയതിനാൽ 7 ബിറ്റ് ആസ്കി ഉപയോഗിച്ച് 128 അക്ഷരങ്ങളുടെ കോഡുകൾ ഉണ്ടാക്കാം.

ഓരോ അക്ഷരത്തിനും 8 ബിറ്റ് ഉപയോഗിക്കുന്ന ഇതിന്റെ മറ്റൊരു പതിപ്പിനെ ആസ്കി 8 അഥവാ എക്സ്റ്റന്റഡ് ആസ്കി (Extended ASCII) എന്ന് വിളിക്കുന്നു. 8 ബിറ്റ് ആസ്കി കൊണ്ട് 256 വ്യത്യസ്താക്ഷരങ്ങളുടെ കോഡുകൾ ഉണ്ടാക്കാം. ഉദാഹരണമായി A എന്ന അക്ഷരത്തെ 01000001 എന്നും B എന്ന അക്ഷരത്തെ 01000010 എന്നും കോഡ് ചെയ്യപ്പെടുന്നു. സാധാരണ കീബോർഡിലെ മുഴുവൻ അക്ഷരങ്ങൾക്കും കോഡ് നൽകുവാൻ ആസ്കി 8 ന് കഴിയുന്നു.

ബി. എബ്സിഡിക് (EBCDIC)

എക്സ്റ്റന്റഡ് ബൈനറി കോഡഡ് ഡെസിമൽ ഇന്റർചേഞ്ച് കോഡ് (Extended Binary Coded Decimal Interchange Code) എന്നതിന്റെ ചുരുക്ക രൂപമാണിത്. ഇന്റർനാഷണൽ ബിസിനസ് മെഷീൻ (ഐ.ബി.എം) നിർമ്മിക്കുന്ന കമ്പ്യൂട്ടറുകളിൽ, ആസ്കിയെ പോലെ ഇതിലും 8 ബിറ്റ് കോഡ് ഉപയോഗിക്കുന്നു. ഇതുപയോഗിച്ച് 256 അക്ഷരങ്ങൾക്ക് കോഡ് നൽകാനാവും. ആസ്കിയിൽ കോഡ് ചെയ്യപ്പെട്ട ഡാറ്റ എബ്സിഡിക് കോഡ് ഉപയോഗിക്കുന്ന കമ്പ്യൂട്ടറിൽ ഉപയോഗിക്കണമെങ്കിൽ ആസ്കി കോഡിൽ നിന്ന് എബ്സിഡിക് കോഡിലേക്ക് മാറ്റേണ്ടതുണ്ട്. അതുപോലെ, എബ്സിഡിക് കോഡ് ഉപയോഗിച്ചുണ്ടാക്കിയ ഡാറ്റ ആസ്കി കമ്പ്യൂട്ടറിൽ ഉപയോഗിക്കണമെങ്കിൽ, ആസ്കിയിലേക്കും മാറ്റേണ്ടതുണ്ട്.

സി. ഇസ്കി (ISCII)

ഇന്ത്യൻ സ്റ്റാൻഡേർഡ് കോഡ് ഫോർ ഇൻഫർമേഷൻ ഇന്റർചേഞ്ച് (Indian Standard Code for Information Interchage) അല്ലെങ്കിൽ ഇന്ത്യൻ സ്ക്രിപ്റ്റ് കോഡ് ഫോർ ഇൻഫർമേഷൻ ഇന്റർചേഞ്ച് (Indian Script Code for Information Interchage) എന്നതിന്റെ ചുരുക്കരൂപമാണിത്. വിവിധ ഇന്ത്യൻഭാഷകളിലെ അക്ഷരങ്ങളുടെ എൻകോഡിംഗ് (Encoding) വ്യവസ്ഥയാണിത്. 8 ബിറ്റ് ഉപയോഗിച്ചാണ് ഇസ്കി ഡാറ്റ പ്രതിനിധാനം ചെയ്യുന്നത്. 1986 ൽ ഇലക്ട്രോണിക് വകുപ്പിന് കീഴിലുള്ള നിലവാരം നിശ്ചയിക്കൽ സമിതി ചിട്ടപ്പെടുത്തിയ ഈ വ്യവസ്ഥ ബ്യൂറോ ഓഫ് ഇന്ത്യൻ സ്റ്റാൻഡേർഡ്സ് (BIS) അംഗീകരിച്ചതാണ്. ഇസ്കിക്ക് പകരം യൂനിക്കോഡാണ് ഇപ്പോൾ ഉപയോഗിക്കുന്നത്.

ഡി. യൂണികോഡ് (Unicode)

8 ബിറ്റുകൾ ഉപയോഗിക്കുന്ന ആസ്കിക്ക് 256 അക്ഷരങ്ങൾ മാത്രമേ പ്രതിനിധാനം ചെയ്യാനാകൂ. ലോകം മുഴുവനുമുള്ള ലിഖിതഭാഷകളിലെ അക്ഷരങ്ങളെയും ചിഹ്നങ്ങളേയും പ്രതിനിധാനം ചെയ്യാൻ ഇത് മതിയാകില്ല. ഈ പ്രശ്നം പരിഹരിക്കാനാണ് യൂണിക്കോഡ് വികസിപ്പിച്ചെടുത്തത്. ആഗോളവും കാര്യക്ഷമവും നിലവാരമുള്ളതും ആയ അക്ഷരങ്ങളുടെ എൻകോഡിങ് രീതിയാണ് അതിന്റെ ലക്ഷ്യം. ഏത് ഭാഷയായാലും ഏത് പ്ലാറ്റ്ഫോമായാലും (Platform) അവയ്ക്കെല്ലാം വ്യത്യസ്തമായ ഒരക്കം ഇത് നൽകുന്നു.

യൂണിക്കോഡിൽ മൗലികമായി 16 ബിറ്റുകളാണ് ഉപയോഗിക്കുന്നത്. അതിന് 65,536 അക്ഷരങ്ങൾ പ്രതിനിധീകരിക്കാൻ കഴിയും. യൂണിക്കോഡ് കൺസോർഷ്യം എന്ന ലാഭേച്ഛയില്ലാത്ത സംഘടനയാണ് ഇത് ചിട്ടപ്പെടുത്തുന്നത്. കൺസോർഷ്യം 1991 ൽ ആദ്യപതിപ്പായ 1.0.0 പ്രസിദ്ധീകരിച്ചു. അതിനെ അടിസ്ഥാനമാക്കി നിലവാരം മെച്ചപ്പെടുത്തുന്നതിനുള്ള ശ്രമം തുടരുകയാണ്. ഈ കാലയളവിൽ യൂണികോഡ് ഉപയോഗിക്കുന്നത് 16ൽ അധികം ബിറ്റുകളാണ്. അതിനാൽ ധാരാളം അക്ഷരങ്ങളെ പ്രതിനിധാനം ചെയ്യാൻ അതിന് സാധിക്കും. ലോകത്തിലെ എല്ലാ ലിഖിത ഭാഷകളുടെയും അക്ഷരങ്ങളെ പ്രതിനിധാനം ചെയ്യുവാൻ യൂണിക്കോഡിന് സാധിക്കുന്നു.

1.8.3 ശബ്ദം, ചിത്രം, വീഡിയോ എന്നിവയുടെ പ്രതിനിധാനം (Representation audio, image & video)

ഇതിന് മുമ്പുള്ള ഭാഗത്തിൽ അക്കങ്ങളും അക്ഷരങ്ങളും ഉൾപ്പെട്ട വിവരങ്ങൾ കമ്പ്യൂട്ടറിൽ പ്രതിനിധാനം ചെയ്യുന്ന വിധവും അവയുടെ വ്യത്യസ്ത മാനദണ്ഡങ്ങളും നാം പരിചയപ്പെട്ടു.

ഡിജിറ്റൽ കമ്പ്യൂട്ടറുകളുടെ സഹായത്തോടെ നിത്യജീവിതത്തിലെ പ്രശ്നങ്ങൾ കൈകാര്യം ചെയ്യുമ്പോൾ മിക്കപ്പോഴും അക്കങ്ങളോ അക്ഷരങ്ങളോ അല്ലാത്ത വിവരങ്ങൾ രേഖപ്പെടുത്തുകയോ പ്രോസസ്സ് ചെയ്യേണ്ടതായോ വരാം. അക്കങ്ങളെയും അക്ഷരങ്ങളെയും പോലെ ശബ്ദം, ചിത്രം, വീഡിയോ എന്നിവയിലും ധാരാളം വിവരങ്ങൾ അടങ്ങിയിട്ടുണ്ട്. ഇവ സംഭരിക്കുന്നതിനുള്ള വിവിധ ഫയൽ ഘടനകളെക്കുറിച്ച് നമുക്ക് ചർച്ച ചെയ്യാം.

ഡിജിറ്റൽ ശബ്ദം, ചിത്രം, വീഡിയോ എന്നിവയുടെ ഫയൽ ഘടനകൾ (Digital audio, image & video file formats)

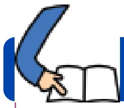
ശബ്ദം, ചിത്രം, വീഡിയോ എന്നിങ്ങനെയുള്ള മൾട്ടിമീഡിയ ഡാറ്റ വ്യത്യസ്ത ഫയൽ ഘടനകളിലാണ് സംഭരിക്കുന്നത്. ഡാറ്റയുടെ വലുപ്പം കുറയ്ക്കുന്നതിനും ചുരുക്കുന്നതിനും വിവിധ കെട്ടുകളാക്കുന്നതിനും വിവിധ സമീപനരീതികൾ ഉപയോഗിക്കുമ്പോൾ അവ വ്യത്യസ്ത ഫയൽഘടനയ്ക്ക് കാരണമാകുന്നു. ഉദാഹരണത്തിന് ഒരു ചിത്രം സാധാരണനിലയിൽ ജോയിന്റ് പിക്ചർ എക്സ്പേർട്ട്സ് ഗ്രൂപ്പ് (ജെപെഗ് - JPEG) ഫയൽ ഘടനയിലാണ് സംഭരിക്കുന്നത്. ഈ ചിത്രത്തിന്റെ ഫയലിൽ തലക്കെട്ട് (Header) വിവരങ്ങളും ചിത്രത്തിന്റെ (Image) ഡാറ്റയും അടങ്ങിയിരിക്കുന്നു. ഫയലിന്റെ പേര്, വലുപ്പം, പരിഷ്കരിച്ച ഡാറ്റ, ഫയൽഘടന മുതലായ വിവരങ്ങൾ തലക്കെട്ട് ഭാഗത്താണ് സംഭരിക്കുന്നത്. പിക്സലുകളുടെ തീവ്രതയെക്കുറിച്ചുള്ള വിവരങ്ങൾ ഡാറ്റ ഭാഗത്തും ശേഖരിക്കുന്നു.

ഫയലിന്റെ വലുപ്പം കുറയ്ക്കുന്നതിന് ഡാറ്റ ചുരുക്കിയോ അല്ലാതെയോ സംഭരിക്കാം. സാധാരണനിലയിൽ ചിത്രം ഡാറ്റയെ ചുരുക്കിയാണ് സംഭരിക്കുന്നത്. എന്താണ് ചുരുക്കൽ (Compression) എന്ന് നോക്കാം. 400x400 പിക്സൽ വലുപ്പമുള്ള, കറുപ്പ് നിറമുള്ള ഒരു ചിത്രം ഉദാഹരണമായി എടുക്കാം. 1,60,000 (400x400) പിക്സലിലും കറുപ്പ്, കറുപ്പ്, ...കറുപ്പ് എന്നിങ്ങനെ ആവർത്തിച്ച് സംഭരിക്കാം. ഇത് ചുരുക്കാതെയുള്ള രൂപമാണ്. അതേസമയം, കറുപ്പ് എന്ന് ഒരു തവണ രേഖപ്പെടുത്തുകയും 1,60,000 തവണ ആവർത്തനം എന്നും രേഖപ്പെടുത്തുന്നതാണ് ചുരുക്കി സംഭരിക്കൽ. ചുരുക്കലിനായി ഇത്തരം നിരവധി രീതികൾ ഉപയോഗിക്കാറുണ്ട്. ബിറ്റ്മാപ് (BMP), ടാഗ്ഡ് ഇമേജ് ഫയൽ ഫോർമാറ്റ് (TIFF), ഗ്രാഫിക്സ് ഇന്റർചേഞ്ച് ഫോർമാറ്റ് (GIF), പോർട്ടബിൾ പബ്ലിക് നെറ്റ്‌വർക്ക് ഗ്രാഫ് (PNG) തുടങ്ങി വിവിധ തരത്തിലുള്ള ഫയൽ ഘടനകളിൽ ചിത്രങ്ങൾ ഉപയോഗത്തിനനുസരിച്ച് സംഭരിക്കപ്പെടുന്നു.

ചിത്രത്തിന്റെ കാര്യത്തിൽപ്പറഞ്ഞ തലക്കെട്ട് ഫയൽ വിവരങ്ങൾ, ശബ്ദം, വീഡിയോ എന്നീ ഫയലുകൾക്കും ബാധകമാണ്. WAV, MP3, MIDI, AIFF മുതലായ വ്യത്യസ്ത ഫയൽ ഘടനകളിൽ ഡിജിറ്റൽ ശബ്ദ ഡാറ്റ സംഭരിക്കാൻ കഴിയും. ഡിജിറ്റൽ ശബ്ദഡാറ്റ സംഭരിക്കുന്നതിന് ഒരു ശബ്ദ ഫയൽ ഘടന വിവരിക്കുന്നുണ്ട്. ചില സമയങ്ങളിൽ ഇത് കണ്ടെയ്നർ ഫോർമാറ്റ് (Container Format) എന്ന് സൂചിപ്പിക്കാറുണ്ട്. ഉദാഹരണമായി WAV ചുരുക്കാത്ത ശബ്ദവും, MP3 ഫയലുകളിൽ ചുരുക്കിയ ശബ്ദവുമാണ് ഉൾക്കൊള്ളുക. സംശ്ലേഷണം ചെയ്ത സംഗീത ഡാറ്റ ശേഖരിച്ചുവയ്ക്കുന്ന സംവിധാനമാണ് MIDI (Musical Instrument Digital Interface). അതുപോലെ AVI (Audio Video Interleave) എന്നത് വീഡിയോഫയൽ ശേഖരിച്ചുവയ്ക്കുന്ന മറ്റൊരു സംവിധാനമാണ്. MP3, JPEG-2, WMV എന്നീ ഫയൽ ഘടനകൾ ശബ്ദം, വീഡിയോ സംഭരിച്ചുവയ്ക്കുന്നതിനും ഒരേ സമയം പ്രവർത്തിപ്പിക്കുന്നതിനും ഉപയോഗിക്കുന്നു.

സ്വയം വിലയിരുത്താം

1. -80 നെ ചിഹ്നവും മൂല്യവും രൂപത്തിൽ പ്രതിനിധാനം ചെയ്താൽ അതിന്റെ MSB ഏതാണ്?
2. 28.756 നെ മാന്റിസ എക്സ്പൊണെന്റ് രൂപത്തിൽ എഴുതുക.
3. ASCII യുടെ പൂർണരൂപം എഴുതുക.
4. 60 നെ 1 ന്റെ പൂരകമായി പ്രതിനിധാനം ചെയ്യുക.
5. യൂണികോഡ് നിർവചിക്കുക.
6. ഏതെങ്കിലും രണ്ട് ചിത്രഫയൽ ഘടനകൾ എഴുതുക.



നമുക്ക് സംഗ്രഹിക്കാം

ഡാറ്റ പ്രോസസിംഗ് തുടർച്ചയായ ഒരു പ്രവർത്തനമാകുന്നു അതിൽ ഡാറ്റയെ വിവരമാക്കി മാറ്റുന്നു. യാന്ത്രികമല്ലാതെ ചെയ്യുന്ന ഡാറ്റ പ്രോസസിംഗിന്റെ പരിമിതികൾ ഇലക്ട്രോണിക് ഡാറ്റ പ്രോസസിംഗ് കൊണ്ട് തരണം ചെയ്യുന്നു. ഏറ്റവും നല്ല ഡാറ്റ പ്രോസസറാണ് കമ്പ്യൂട്ടർ. ഒരു കമ്പ്യൂട്ടറിന് 5 അടിസ്ഥാന ഘടകങ്ങളുണ്ട് അവ ഇൻപുട്ട് യൂണിറ്റ്, സംഭരണ യൂണിറ്റ്, അരിത്തമെറ്റിക് & ലോജിക്ക് യൂണിറ്റ്, കൺട്രോൾ യൂണിറ്റ്, ഔട്ട്പുട്ട് യൂണിറ്റ് എന്നിവയാകുന്നു.

വിവിധ രൂപങ്ങളിലാണ് കമ്പ്യൂട്ടറിലേക്ക് ഡേറ്റകൾ നല്കുന്നതെങ്കിലും ആന്തരികമായി അവ പ്രതിനിധീകരിക്കുന്നത് ബിറ്റുകൾ ഉപയോഗിച്ചാണ്. കമ്പ്യൂട്ടറുമായി ബന്ധപ്പെട്ട വിത്യസ്ത സംഖ്യാന സമ്പ്രദായങ്ങളുണ്ട്. ഒരു സമ്പ്രദായത്തിലെ ഏതൊരു സംഖ്യക്കും തത്തുല്യമായ സംഖ്യ മറ്റ് സമ്പ്രദായ ങ്ങളിലുമുണ്ട്. അക്ഷരങ്ങളെ സൂചിപ്പിക്കാൻ വ്യത്യസ്തമായ കോഡിംഗ് രീതികൾ കമ്പ്യൂട്ടറുകളിൽ ലഭ്യമാണ്. ഓഡിയോ (Audio) ചിത്രം (Image) വീഡിയോ (Video) ഇവയും ബൈനറി രൂപത്തിലാണ് സംഭരിക്കുന്നത്. വ്യത്യസ്ത ഫയൽ ഘടനകളും ഇവയെ പ്രതിനിധാനം ചെയ്യാൻ ഉപയോഗിക്കുന്നു.

പഠന നേട്ടങ്ങൾ

ഈ അധ്യായം പൂർത്തത്തിയാക്കിയ പഠിതാവ്

- ഡാറ്റയും വിവരവും വേർതിരിച്ചറിയൽ
- ഡാറ്റ പ്രോസസിംഗിന്റെ വിവിധ ഘട്ടങ്ങൾ തിരിച്ചറിയൽ
- കമ്പ്യൂട്ടറിന്റെ പ്രവർത്തന ഘടകങ്ങൾ പട്ടികപ്പെടുത്തുകയും ഒരോന്നിന്റേയും പ്രവർത്തനങ്ങൾ വിശദീകരിക്കുകയും ചെയ്യുക
- എന്തുകൊണ്ട് കമ്പ്യൂട്ടർ ഏറ്റവും നല്ല ഡാറ്റ പ്രോസസിംഗ് യന്ത്രമാകുന്നു എന്ന് വിശദീകരിക്കൽ
- കമ്പ്യൂട്ടറിനുള്ളിലെ ഡാറ്റ പ്രതിനിധാനം എന്ന ആശയം ഗ്രഹിക്കുക
- സംഖ്യയെ ഒരു സമ്പ്രദായത്തിൽ നിന്നും മറ്റൊന്നിലേക്ക് മാറ്റുന്നത്
- അക്ഷരങ്ങളെ പ്രതിനിധാനം ചെയ്യാൻ ഉപയോഗിക്കുന്ന വ്യത്യസ്ത കോഡിംഗ് സമ്പ്രദായങ്ങളുടെ സവിശേഷതകൾ പട്ടികപ്പെടുത്തുക.

മാതൃക ചോദ്യങ്ങൾ

ഹ്രസ്വോത്തര ചോദ്യങ്ങൾ

1. ഡാറ്റ എന്നാൽ എന്ത്?
2. പ്രോസസ് ചെയ്ത ഡാറ്റയെ .... എന്ന് വിളിക്കുന്നു.
3. 29610 ൽ 9ന്റെ സ്ഥാനവില .... ആകുന്നു.
4. ഹെക്സഡെസിമൽ സംഖ്യാന സമ്പ്രദായത്തിൽ............. എണ്ണം ചിഹ്നങ്ങളും ഒക്ടൽ സംഖ്യാന സമ്പ്രദായത്തിൽ ....എണ്ണം ചിഹ്നങ്ങളും ഉപയോഗിക്കുന്നു.
5. 55 എന്ന ദശസംഖ്യയ്ക്ക് തുല്യമായ ഒക്ടൽ സംഖ്യ കാണുക.
6. EBCDIC യുടെ പൂർണ്ണരൂപം .... ആകുന്നു.
7. ലോകത്തിലെ എല്ലാ ഭാഷകളിലേയും അക്ഷരങ്ങളെ പ്രതിനിധാനം ചെയ്യാൻ കഴിയുന്ന കോഡിംഗ് സമ്പ്രദായത്തിന്റെ പേര് പറയുക.
8. ചിഹ്നവും മൂല്യവും രൂപത്തിൽ ഒരു സംഖ്യയുടെ നെഗറ്റീവ് ചിഹ്നത്തിന്റെ ബിറ്റ് .... ആകുന്നു. സംഖ്യ പോസിറ്റീവ് ആണെങ്കിൽ ചിഹ്നത്തിന്റെ ബിറ്റ് .... ആകുന്നു.

ലഘു ഉപന്യാസ ചോദ്യങ്ങൾ

1. ഡാറ്റയും വിവരവും തമ്മിലുള്ള വ്യത്യാസങ്ങളെഴുതുക?
2. പ്ലസ് വൺ അഡ്മിഷന്റെ അപേക്ഷഫോറത്തിൽ നിങ്ങളുടെ വ്യക്തിഗതവിവരങ്ങളും സ്കൂളുകളുടേയും ഗ്രൂപ്പുകളുടേയും തെരഞ്ഞെടുപ്പും അടങ്ങിയിരിക്കുന്നു.

   എ. പ്രവേശന പ്രവർത്തനങ്ങളിലെ ഡാറ്റയും വിവരവും തിരച്ചറിയുക.

   ബി. ലഭിക്കുന്ന വിവരം അപേക്ഷകരേയും സ്കൂൾ അധികാരകളേയും സഹായിക്കുന്നത് എങ്ങനെയാണ്?

   സി. ഡാറ്റ പ്രോസസിംഗിൽ ഉൾപ്പെട്ട പ്രവർത്തനങ്ങൾ എഴുതുക.
3. ഡാറ്റ പ്രോസസിംഗിൽ കമ്പ്യൂട്ടർ മനുഷ്യനേക്കാൾ മികവ് കാട്ടുന്നത് എങ്ങനെ?
4. ഡാറ്റ പ്രോസസിംഗിൽ സംഭരണ ഘടകത്തിന്റെ പ്രാധാന്യം വിശദീകരിക്കുക.
5. കമ്പ്യൂട്ടറിലെ ഇൻപുട്ട് യൂണിറ്റിന്റെ പ്രവർത്തനങ്ങൾ പട്ടികപ്പെടുത്തുക.
6. ദ്വിതീയ സംഭരണം കമ്പ്യൂട്ടറിന് ആവശ്യമുണ്ടോ? നിങ്ങളുടെ ഉത്തരം ന്യായീകരിക്കുക.
7. കമ്പ്യൂട്ടറിൽ കൺട്രോൾ യൂണിറ്റിന്റെ പ്രാധാന്യം എഴുതുക?
8. സി പി യു (CPU) വിന്റെ പ്രവർത്തനത്തിൽ മെമ്മറി യൂണിറ്റ് എങ്ങനെ സഹായിക്കുന്നു?

9. 'കമ്പ്യൂട്ടർ അടിമകളും, മനുഷ്യർ ഉടമകളുമാകുന്നു'. നിങ്ങൾ ഇതുമായി യോജിക്കുന്നുണ്ടോ? കാരണം വ്യക്തമാക്കുക.

10. കമ്പ്യൂട്ടറിന്റെ സവിശേഷതകൾ പട്ടികപ്പെടുത്തുക.

11. കമ്പ്യൂട്ടർ ബഹുമുഖ യന്ത്രമാകുന്നു എങ്ങനെ?

12. കമ്പ്യൂട്ടറിന്റെ സവിശേഷതകളിൽ സ്ഥിരോത്സാഹം എന്ന പദം കൊണ്ട് അർത്ഥമാക്കുന്നതെന്ത്?

13. ഡാറ്റ പ്രതിനിധാനം നിർവചിക്കുക.

14. സംഖ്യാന സമ്പ്രദായം എന്നാൽ എന്ത്? ഏതെങ്കിലും നാല് സംഖ്യാന സമ്പ്രദായം പട്ടികപ്പെടുത്തുക.

15. താഴെ കൊടുത്തിരിക്കുന്ന ശ്രേണികളിലെ വിട്ടുപോയ പദങ്ങൾ കണ്ടെത്തുക.

എ. $(101)_2$, $(110)_2$, $(111)_2$, ..., ...

ബി. $(15)_8$, $(16)_8$, $(17)_8$, ..., ...

സി. $(18)_{16}$, $(1A)_{16}$, $(1C)_{16}$, ..., ...

16. കമ്പ്യൂട്ടർ മെമ്മറിയിൽ ഡാറ്റ പ്രതിനിധാനം ചെയ്യുന്നതിന് ബൈനറി സംഖ്യാന സമ്പ്രദായമാണ് ഉപയോഗിക്കുന്നത് എന്തുകൊണ്ട്?

17. താഴെകൊടുത്തിരിക്കുന്ന ദശസംഖ്യകളെ തത്തുല്യ ബൈനറി സംഖ്യയിലേക്ക് മാറ്റുക.

എ. 25   ബി. 128   സി. 255   ഡി. 19.875   ഇ. 89.25

18. താഴെകൊടുത്തിരിക്കുന്ന ബൈനറിസംഖ്യകളെ തത്തുല്യ ദശസംഖ്യയിലേക്ക് മാറ്റുക.

എ. $(1011)_2$ ബി. $(111001)_2$ സി. $(1000001)_2$ ഡി. $(110001110)_2$

ഇ. $(1111.111)_2$

19. താഴെകൊടുത്തിരിക്കുന്ന ദശസംഖ്യകളെ ഒക്ടലിലേയ്ക്കും ഹെക്സാഡെസിമലിലേയ്ക്കും മാറ്റുക.

എ. 17   ബി. 75   സി. 100   ഡി. 199   ഇ. 256

20. താഴെകൊടുത്തിരിക്കുന്ന ബൈനറിസംഖ്യകളെ ഒക്ടലിലേക്കും ഹെക്സാഡെസിമലിലേയ്ക്കും മാറ്റുക.

എ. $(1011)_2$ ബി. $(101001)_2$ സി. $(11100011)_2$ ഡി. $(110001110)_2$

ഇ. $(10000010001)_2$

21. താഴെകൊടുത്തിരിക്കുന്ന ഒക്ടൽ സംഖ്യകളെ ദശസംഖ്യയിലേക്ക് മാറ്റുക.

എ. $(57)_8$ ബി. $(101)_8$ സി. $(77)_8$ ഡി. $(245)_8$ ഇ. $(1205)_8$

22. താഴെകൊടുത്തിരിക്കുന്ന ഹെക്സാ ഡെസിമൽ സംഖ്യകളെ ദശസംഖ്യയിലേക്ക് മാറ്റുക.

എ. $(2A)_{16}$, ബി. $(101)_{16}$ സി. $(AB)_{16}$,   ഡി. $(1F8)_{16}$,   ഇ. $(ABC)_{16}$,

23. താഴെകൊടുത്തിരിക്കുന്ന ഒക്ടൽ സംഖ്യകളെ ബൈനറിയിലേക്കും ഹെക്സാഡെസിമലിലേയ്ക്കും മാറ്റുക.

എ. $(67)_8$ ബി. $(123)_8$ സി. $(167)_8$ ഡി. $(745)_8$ ഇ. $(1054)_8$

24. താഴെകൊടുത്തിരിക്കുന്ന ഹെക്സാഡെസിമൽ സംഖ്യകളെ ബൈനറിയിലേക്കും ഒക്ടലിലേയ്ക്കും മാറ്റുക.

എ. $(7F)_{16}$, ബി. $(207)_{16}$ സി. $(AB)_{16}$, ഡി. $(9F)_{16}$, ഇ. $(ABC)_{16}$,

25. $(X)_2=(Y)_8=(Z)_{16}=(28)_{10}$ എങ്കിൽ X, Y, Z എന്നിവ കണ്ടുപിടിക്കുക.

26. താഴെകൊടുത്തിരിക്കുന്ന സംഖ്യകളെ അവരോഹണ ക്രമത്തിൽ എഴുതുക.

എ. $(101)_{16}$, ബി. $(110)_{10}$ സി. $(111000)_2$, ഡി. $(251)_8$

27. കമ്പ്യൂട്ടർ മെമ്മറിയിൽ പൂർണസംഖ്യകളെ പ്രതിനിധാനം ചെയ്യുന്ന രീതികൾ എന്തെല്ലാം.

28. താഴെകൊടുത്തിരിക്കുന്ന സംഖ്യകളെ ചിഹ്നവും മൂല്യവും രീതിയിൽ പ്രിതനിധാനം ചെയ്യുക.

എ. –19, ബി. +69, സി. –97 ഡി. –127

29. താഴെകൊടുത്തിരിക്കുന്ന സംഖ്യകളെ ഒന്നിന്റെ പൂരക രീതിയിൽ പ്രിതനിധാനം ചെയ്യുക.

എ. –24, ബി. 69, സി. –100 ഡി. –127

30. താഴെകൊടുത്തിരിക്കുന്ന സംഖ്യകളെ രണ്ടിന്റെ പൂരക രീതിയിൽ പ്രിതനിധാനം ചെയ്യുക.

എ. –33, ബി. +71, സി. –111 ഡി. –127

31. –83നെ മറ്റ് മൂന്ന് സംഖ്യാന സമ്പ്രദായങ്ങളും പ്രതിനിധാനം ചെയ്യുക.

32. ചിഹ്നവും അളവും രീതിയിൽ പ്രിതിനിധാനം ചെയ്ത $(10011001)_2$ ന്റെ ദശസംഖ്യ കാണുക.

33. ഒരു ദശസംഖ്യയെ 32 ബിറ്റ് കമ്പ്യൂട്ടറിൽ എങ്ങനെ പ്രിതനിധാനം ചെയ്യുന്നു. വിശദീകരിക്കുക.

34. കമ്പ്യൂട്ടർ മെമ്മറിയിൽ അക്ഷരങ്ങളെ പ്രതിനിധാനം ചെയ്യാനുള്ള രീതികൾ ഏതെല്ലാം?

35. ആസ്കി (ASCII), ഇസ്കി (ISCII) എന്നിവയെ കുറിച്ച് ലഘു കുറിപ്പെഴുതുക.

36. അക്ഷരങ്ങളെ പ്രതിനിധാനം ചെയ്യുന്നതിൽ യൂനികോഡ് (unicode) ന്റെ പ്രാധാന്യം ചുരുക്കി വിവരിക്കുക.

ഉപന്യാസ ചോദ്യങ്ങൾ

1. ബാങ്കിന്റെ പ്രവർത്തനങ്ങൾ ഉദാഹരണമായെടുത്ത് ഡാറ്റ പ്രോസസിംഗിന്റെ വിവിധ ഘട്ടങ്ങളിൽ അടങ്ങിയിട്ടുള്ള പ്രവർത്തനങ്ങൾ ചുരുക്കി വിവരിക്കുക.
2. ചിത്രത്തിന്റെ സഹായത്തോടുകൂടി കമ്പ്യൂട്ടറിന്റെ പ്രവർത്തന ഘടകങ്ങളെ കുറിച്ച് വിശദീകരിക്കുക.
3. താഴെ പറയുന്ന സംഖ്യകളെ മാന്റിസ എക്സപൊണെന്റ് രീതിയിൽ എഴുതുക.

   എ. $(1011.101)_2$ ബി. $(65356)_{10}$ സി. $(A5F)_{16}$ ഡി. $(67.4)_8$

   ഇ. $(763.452)_{10}$
4. കമ്പ്യൂട്ടർ മെമ്മറിയിൽ സംഖ്യകളെ പ്രതിനിധാനം ചെയ്യുന്ന രീതികളെ കുറിച്ച് ചുരുക്കി വിവരിക്കുക.
5. കമ്പ്യൂട്ടർ മെമ്മറിയിൽ അക്ഷരങ്ങളെ പ്രതിനിധാനം ചെയ്യുന്ന രീതികളെ കുറിച്ച് ചുരുക്കി വിവരിക്കുക.

2

F5D6P5

പ്രധാന ആശയങ്ങൾ

- ഹാർഡ്‌വെയർ
  - പ്രോസസ്സർ
  - മദർബോർഡ്
  - പെരിഫെറലുകളും പോർട്ടുകളും
  - മെമ്മറി – പ്രാഥമിക മെമ്മറി, ദ്വിതീയ മെമ്മറി
  - ഇൻപുട്ട് ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ
- ഇ–വേസ്റ്റ്
- ഹരിത കമ്പ്യൂട്ടിംഗ്
- സോഫ്റ്റ്‌വെയർ
  - സിസ്റ്റം സോഫ്റ്റ്‌വെയർ (ഓപ്പറേറ്റിങ്സിസ്റ്റം, ഭാഷ പ്രോസസ്സറുകൾ, യൂട്ടിലിറ്റി സോഫ്റ്റ്‌വെയർ)
  - ആപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയർ (പൊതു ഉപയോഗം, പ്രത്യേക ഉപയോഗം)
  - സ്വതന്ത്ര ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയർ ആശയങ്ങളും
  - ഫ്രീവെയറും ഷെയർവെയറും
  - ഉടമസ്ഥാവകാശമുള്ള സോഫ്റ്റ്‌വെയർ (Proprietary software)
- ഹ്യൂമൺവെയർ/ലൈവ്‌വെയർ

കമ്പ്യൂട്ടർ സിസ്റ്റത്തിന്റെ ഭാഗങ്ങൾ

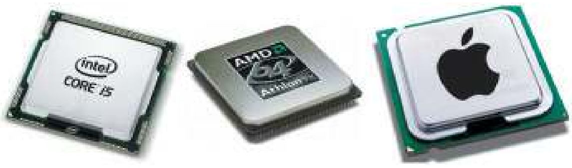
ഇന്നത്തെ ലോകത്തു കമ്പ്യൂട്ടറുകളും അതിന്റെ ഉപയോഗങ്ങളും നമുക്കു പരിചിതമാണ്. നൽകുന്ന നിർദേശങ്ങൾക്കനുസരിച്ച് ഡാറ്റ സ്വീകരിക്കുകയും പ്രോസസ്സ് ചെയ്ത് ഔട്ട്പുട്ട് പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്ന വേഗതയേറിയ ഒരു ഇലക്ട്രോണിക് ഉപകരണമാണ് കമ്പ്യൂട്ടർ. ഒരു കമ്പ്യൂട്ടർ സിസ്റ്റത്തിന്റെ അടിസ്ഥാന രൂപകൽപ്പനയുടെ അവലോകനം ഈ പാഠഭാഗത്ത് അവതരിപ്പിക്കുന്നു. ഒരു കമ്പ്യൂട്ടർ സിസ്റ്റത്തിന്റെ വിവിധ ഘടകങ്ങൾ എങ്ങനെ ക്രമീകരിക്കപ്പെട്ടിരിക്കുന്നു എന്നും, ഒരു പ്രത്യേക ചുമതല നിർവ്വഹിക്കാൻ ഏതെല്ലാം പ്രവർത്തനങ്ങൾ നടത്തുന്നു എന്നും ഇവിടെ ചർച്ച ചെയ്യുന്നു. ഒരു കമ്പ്യൂട്ടറിന് പ്രധാനമായും രണ്ട് ഘടകങ്ങളാണുള്ളത് – ഹാർഡ്‌വെയറും സോഫ്റ്റ്‌വെയറും ഒരു കമ്പ്യൂട്ടർ സംവിധാനവുമായി ബന്ധപ്പെട്ട എല്ലാ ഭൗതിക ഘടകങ്ങളേയും ഹാർഡ്‌വെയർ എന്ന് സൂചിപ്പിക്കുന്നു. ഒരു നിർദ്ദിഷ്ട പ്രവൃത്തി ചെയ്യുന്ന ഹാർഡ്‌വെയറിനുള്ള ഒരു കൂട്ടം നിർദ്ദേശങ്ങളാണ് സോഫ്റ്റ്‌വെയർ. നിത്യജീവിതത്തിലെ ഏതെങ്കിലും പ്രശ്നം കമ്പ്യൂട്ടർ ഉപയോഗിച്ച് പരിഹരിക്കേണ്ടി വരുമ്പോൾ, വിവരങ്ങൾ ഉല്പാദിപ്പിക്കുന്നതിനാവശ്യമായ ഡാറ്റയെ പ്രോസസ് ചെയ്യുന്നതിനുള്ള പ്രവർത്തനങ്ങൾ ചിട്ടപ്പെടുത്തേണ്ടതുണ്ട്. വിവിധ ഹാർഡ്‌വെയർ ഘടകങ്ങൾ അവതരിപ്പിക്കുന്നു. അതിനു ശേഷം ഇലക്ട്രോണിക് വേസ്റ്റ്, അവയുടെ നിർമാർജ്ജന രീതികൾ, ഹരിത കമ്പ്യൂട്ടിങ്ങിന്റെ ആശയം എന്നിവ വിവരിക്കുന്നു. പിന്നീട് വിവിധതരത്തിലുള്ള

സോഫ്റ്റ്‌വെയറുകളുടെ വിശദമായ തരംതിരിവും നൽകിയിരിക്കുന്നു. ഓപ്പൺ സോഴ്സ്, ഫ്രീവെയർ, ഷെയർ വെയർ, ഉടമസ്ഥാവകാശമുള്ള സോഫ്റ്റ്‌വെയർ എന്നിവയുടെ ആശയങ്ങളെക്കുറിച്ചും നമ്മൾ ചർച്ച ചെയ്യുന്നു.

2.1 ഹാർഡ്‌വെയർ (Hardware)

ഹാർഡ്‌വെയറും സോഫ്ട്‌വെയറും അടങ്ങുന്നതാണ് കമ്പ്യൂട്ടർ സംവിധാനം എന്ന് നമുക്കറിയാം. തൊട്ടറിയാൻ കഴിയുന്നതും കാണാൻ സാധിക്കുന്നതുമായ കമ്പ്യൂട്ടറിന്റെ ഭാഗങ്ങളാണ് ഹാർഡ്‌വെയർ എന്ന പദം കൊണ്ടുദ്ദേശിക്കുന്നത്. മാത്രമല്ല ഇലക്ട്രോ മെക്കാനിക്കൽ ഘടകങ്ങളും ഇതിൽ ഉൾപ്പെടുന്നു. കമ്പ്യൂട്ടറിന്റെ പ്രവർത്തന ഭാഗങ്ങൾ ഹാർഡ്‌വെയർ ഘടകങ്ങളുമായി ബന്ധപ്പെട്ടിരിക്കുന്നു. നമുക്ക് ഈ പ്രവർത്തന ഘടകങ്ങളെ പരിചയപ്പെടാം.

2.1.1. പ്രോസസ്സറുകൾ (Processors)



ചിത്രം 2.1: വിവിധതരം പ്രോസസ്സറുകൾ

കമ്പ്യൂട്ടറിൽ എല്ലാ കണക്കുകൂട്ടലുകൾ നടത്തുന്നതും യുക്തിപരമായ തീരുമാനങ്ങൾ എടുക്കുന്നതും മറ്റു പ്രവർത്തനങ്ങൾ ഏകോപിപ്പിക്കുന്നതും CPU ആണെന്ന് നാം കഴിഞ്ഞ അധ്യായത്തിൽ പഠിച്ചുവല്ലോ. CPU വിന്റെ പ്രവർത്തനം കമ്പ്യൂട്ടറിന്റെ മൊത്തത്തിലുള്ള പ്രകടനത്തെ നിർണ്ണയിക്കുന്നു. CPU എന്നത് ഒരു ഇന്റഗ്രേറ്റഡ് സർക്യൂട്ട് (IC) പാക്കേജ് ആണ്. ഇത് ദശലക്ഷക്കണക്കിന് ട്രാൻസിസ്റ്ററുകളും അനുബന്ധഘടകങ്ങളും കൂടിചേർന്ന ഒരു സിലിക്കൺ ചിപ്പാണ്. ഇതിനെ മൈക്രോപ്രൊസസ്സർ എന്ന്‌വിളിക്കുന്നു. ചിത്രം 2.1 ൽ ചില കമ്പനികളുടെ പ്രോസസ്സറുകൾ കാണിച്ചിരിക്കുന്നു. കമ്പ്യൂട്ടറിലെ പ്രധാന ബോർഡായ മദർ ബോർഡിലെ വലിയ സോക്കറ്റുമായാണ് CPU സാധാരണയായി ബന്ധിപ്പിച്ചിരിക്കുന്നത്. CPU പ്രവർത്തിക്കുമ്പോൾ ധാരാളം താപം പുറംതള്ളുന്നതുകൊണ്ട് ഫാനും ഹീറ്റ് സിങ്കും ഉൾപ്പെട്ട കൂളിംഗ് സിസ്റ്റം ഇതിൽ ഘടിപ്പിച്ചിരിക്കുന്നു. സാധാരണമായി ഉപയോഗിക്കുന്ന പ്രോസസ്സറുകളാണ് ഇന്റൽ കോർ i3, കോർ i5, കോർ i7, AMD Quadcore തുടങ്ങിയവ.

എല്ലാ കമ്പ്യൂട്ടറുകളിലും ഒരു ക്ലോക്ക് ഉണ്ട്. അത് നിർദ്ദേശങ്ങൾ നിർവ്വഹിക്കുന്ന നിരക്ക് ക്രമീകരിക്കുന്നു. ഓരോ നിർദ്ദേശവും നിർവ്വഹിക്കാൻ CPU വിന് ഒരു നിശ്ചിത എണ്ണം ക്ലോക്ക് ടിക്കുകൾ (ക്ലോക്ക് ആവൃത്തി) ആവശ്യമാണ്. ക്ലോക്കിന്റെ വേഗത കൂടുമ്പോൾ, സി.പി.യു. വിന് ഒരു സെക്കന്റിൽ കൂടുതൽ നിർദേശങ്ങൾ നിർവ്വഹിക്കുവാൻ സാധിക്കും. മറ്റൊരു ഘടകം ചിപ്പിന്റെ രൂപഘടനയാണ്. ഒരു സമയം പ്രോസസ്സറിന് പ്രോസസ്സ് ചെയ്യുവാൻ കഴിയുന്ന ബിറ്റുകളുടെ സംഖ്യയെ word size എന്ന് വിളിക്കുന്നു. വിവിധ word size ഉള്ള പ്രോസസ്സറുകൾ ഉപയോഗിക്കപ്പെടുന്നു. (ഉദാഹരണം 8-bit, 16-bit, 32-bit, 64-bit തുടങ്ങിയവ)

CPU വിന് ഉള്ളിലെ സംഭരണ സ്ഥലങ്ങളാണ് രജിസ്റ്ററുകൾ, മറ്റ് മെമ്മറി ഭാഗങ്ങളേക്കാൾ വേഗത്തിൽ അതിലെ ഉള്ളടക്കത്തെ CPU വിന് ഉപയോഗിക്കാൻ കഴിയും. നിർദ്ദേശങ്ങളും ഡാറ്റയും താൽക്കാലികമായി സംഭരിക്കാനുള്ള സ്ഥലമാണ് രജിസ്റ്ററുകൾ. ഇവ മെമ്മറിയുടെ ഭാഗമല്ല. എന്നാൽ കമ്പ്യൂട്ടറിന് വേഗത പ്രദാനം ചെയ്യുന്ന പ്രത്യേക സംഭരണ സ്ഥലങ്ങളാണ്. രജിസ്റ്ററുകൾ പ്രവർത്തിക്കുന്നത് കൺട്രോൾ യൂണിറ്റിന്റെ നിർദ്ദേശാനുസരണമാണ്. ഡാറ്റയും നിർദേശങ്ങളും ശേഖരിച്ച് വളരെ വേഗത്തിൽ അരിത്തമെറ്റിക് അഥവാ ലോജിക്കൽ പ്രവർത്തനങ്ങൾ നിർവ്വഹിക്കാൻ ഇവ സഹായിക്കുന്നു. പ്രോഗ്രാമിന്റെ കൃത്യനിർവ്വഹണം ഇത് വേഗത്തിലാക്കുന്നു. CPU വിനുള്ളിലെ പ്രധാനപ്പെട്ട രജിസ്റ്ററുകൾ താഴെ കൊടുക്കുന്നു.

i. **അക്യുമുലേറ്റർ (Accumulator)**: അരിത്തമെറ്റിക് ആന്റ് ലോജിക് യൂണിറ്റിന്റെ (ALU) ഒരു ഭാഗമാണ് അക്യുമുലേറ്റർ. അരിത്തമെറ്റിക് ലോജിക്കൽ പ്രവർത്തനങ്ങൾ നിർവ്വഹിക്കുമ്പോൾ അതിന്റെ ഫലങ്ങൾ സൂക്ഷിക്കാൻ ഈ രജിസ്റ്റർ ഉപയോഗിക്കുന്നു. ഇതിനെ രജിസ്റ്റർ A എന്നും വിളിക്കാറുണ്ട്.

ii. **മെമ്മറി അഡ്രസ്സ് രജിസ്റ്റർ (MAR):** ഡാറ്റ സംഭരിക്കപ്പെടേണ്ടതോ അല്ലെങ്കിൽ എവിടെ നിന്നാണോ വീണ്ടെടുക്കപ്പെടേണ്ടത് ആ മെമ്മറി ലൊക്കേഷന്റെ വിലാസം സൂക്ഷിക്കുന്ന രജിസ്റ്ററാണ് മെമ്മറി അഡ്രസ് രജിസ്റ്റർ.

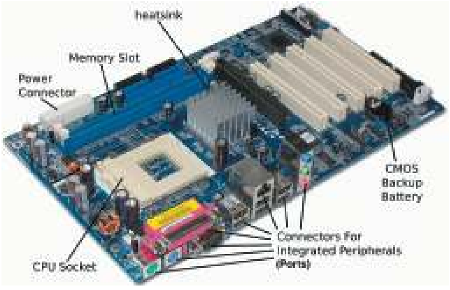
iii. **മെമ്മറി ബഫർ രജിസ്റ്റർ (MBR):** ഡാറ്റാ പ്രോസസ്സിങ്ങിനുവേണ്ടി പ്രോസസ്സർ എടുക്കുന്നതോ പ്രോസസിങ്ങിനുശേഷം കൊടുക്കേണ്ടതോ ആയ ഡാറ്റ താൽക്കാലികമായി സൂക്ഷിക്കുന്ന രജിസ്റ്ററാണ് മെമ്മറി ബഫർ രജിസ്റ്റർ.

iv. **ഇൻസ്ട്രക്ഷൻ രജിസ്റ്റർ (IR):** ഏത് നിർദ്ദേശമാണോ പ്രൊസസ്സർ നിർവ്വഹിക്കേണ്ടത്, ആ നിർദ്ദേശം സൂക്ഷിച്ചുവെക്കുന്ന രജിസ്റ്ററാണ് ഇൻസ്ട്രക്ഷൻ രജിസ്റ്റർ.

v. **പ്രോഗ്രാം കൗണ്ടർ (PC):** പ്രോസസ്സർ അടുത്തതായി നിർവ്വഹിക്കേണ്ട നിർദ്ദേശത്തിന്റെ മെമ്മറി വിലാസം സൂക്ഷിക്കുന്ന രജിസ്റ്ററാണ് പ്രോഗ്രാം കൗണ്ടർ.

2.1.2 മദർബോർഡ് (Motherboard)

പ്രൊസസ്സറും അതിനോട് അനുബന്ധിച്ചുള്ള പ്രധാനപ്പെട്ട ഘടകങ്ങളും അടങ്ങിയിട്ടുള്ള വലിയ പ്രിന്റഡ് സർക്യൂട്ട് ബോർഡാണ് മദർബോർഡ് (PCB). മെമ്മറി, ഗ്രാഫിക്സ് കാർഡ്, സൗണ്ട് കാർഡ് തുടങ്ങിയുള്ള സർക്യൂട്ട് ബോർഡുകൾ ആവശ്യാനുസരണം വേണ്ടിവന്നാൽ ഉൾപ്പെടുത്താനുള്ള എക്സ്പാൻഷൻ സ്ലോട്ടുകൾ ഇതിലുണ്ട്. (ചിത്രം 2.2 നോക്കുക) മദർബോർഡ് നിർബ്ബന്ധമായും പ്രോസസ്സറിന് അനുയോജ്യമായിരിക്കണം.

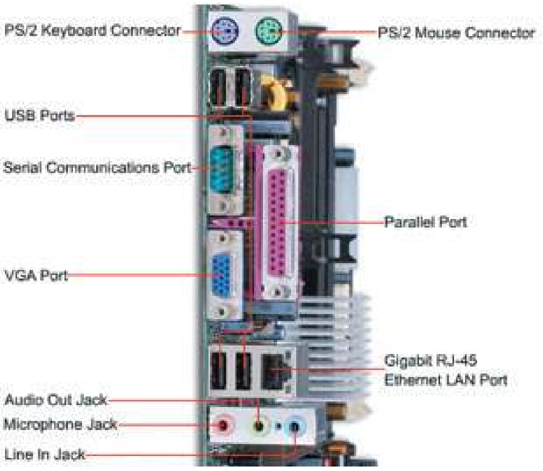


ചിത്രം 2.2: മദർബോർഡ്

2.1.3 പെരിഫറലുകളും പോർട്ടുകളും (Peripherals and ports)

ഒരു കമ്പ്യൂട്ടർ സിസ്റ്റത്തിന്റെ കഴിവുകൾ മെച്ചപ്പെടുത്തുന്നതിനു വേണ്ടി അതുമായി ബന്ധിപ്പിച്ചിട്ടുള്ള ഉപകരണങ്ങളാണ് പെരിഫറലുകൾ. ബാഹ്യ ഉപകരണങ്ങളെ കമ്പ്യൂട്ട

റുമായി ബന്ധിപ്പിക്കുന്നതിന് മദർബോർഡിലെ പോർട്ടുകൾ ഉപയോഗിക്കുന്നു. കമ്പ്യൂട്ടർ സിസ്റ്റത്തിലെ വിവിധ പോർട്ടുകൾ ചിത്രം 2.3 ൽ കാണിക്കുന്നു. ഇൻപുട്ട് ഉപകരണങ്ങൾ, ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ, ബാഹ്യ സംഭരണ ഉപകരണങ്ങൾ, ആശയവിനിമയ ഉപകരണങ്ങൾ എന്നിവ പെരിഫറലുകളിൽ ഉൾപ്പെടുന്നു. വീഡിയോ ഗ്രാഫിക്സ് അറേ (VGA), ഐ.ബി.എം. പേഴ്സണൽ സിസ്റ്റം/2 (PS/2), യൂണിവേഴ്സൽ സീരിയൽ ബസ് (USB), ഇതർനെറ്റ്, ഹൈ ഡെഫനിഷൻ മൾട്ടിമീഡിയ ഇന്റർഫേസ് (HDMI) എന്നിങ്ങനെ മദർബോർഡിൽ ലഭ്യമായ വിവിധ പോർട്ടുകളിലൂടെ ബാഹ്യ ഉപകരണങ്ങൾ മദർബോർഡുമായി ആശയവിനിമയം നടത്തുന്നു. പേഴ്സണൽ കമ്പ്യൂട്ടറുകളിൽ ഉപയോഗിക്കുന്ന ചില പോർട്ടുകൾ നമുക്ക് നോക്കാം.



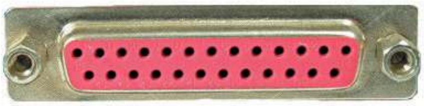
ചിത്രം 2.3: പോർട്ടുകൾ

a. സീരിയൽ പോർട്ട്

ചിത്രം 2.4: സീരിയൽ പോർട്ട്

ഒരു സീരിയൽ പോർട്ട്/സീരിയൽ കമ്മ്യൂണിക്കേഷൻ പോർട്ട് ഒരു സമയത്ത് ഒരു ബിറ്റ് ഡാറ്റ വീതം അയക്കുന്നു. പഴയ കമ്പ്യൂട്ടറുകളിൽ മോഡം, മൗസ് അല്ലെങ്കിൽ കീബോർഡ് പോലുള്ള ഉപകരണങ്ങൾ സീരിയൽ പോർട്ടുകളിലൂടെയാണ് ബന്ധിപ്പിച്ചിരുന്നത് (ചിത്രം 2.4 കാണുക). സീരിയൽ കേബിളുകൾ നിർമ്മിക്കുന്നതിന് ചെലവ് കുറവാണ്, അതോടൊപ്പം ഇന്റർഫെറൻസിൽ നിന്നും സംരക്ഷണം നൽകുകയും ചെയ്യുന്നു. എന്നാൽ വേഗത വളരെ കുറവായതിനാൽ PS/2, USB മുതലായ വേഗതയേറിയ പോർട്ടുകൾ സീരിയൽ പോർട്ടുകൾക്ക് പകരക്കാരാകുന്നു.

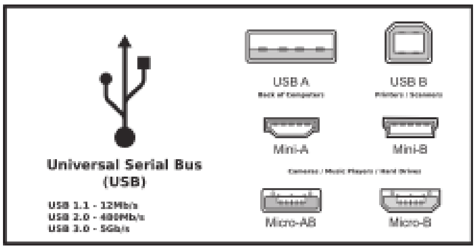
b. പാരലൽ പോർട്ട്


ചിത്രം 2.5: പാരലൽ പോർട്ട്

പാരലൽ പോർട്ടുകൾക്ക് ഒരേസമയം നിരവധി ബിറ്റ് ഡാറ്റ ഒരേ സമയം അയക്കുകയും സ്വീകരിക്കുകയും ചെയ്യാം. സീരിയൽ പോർട്ടിനേക്കാൾ വേഗതയുള്ളതാണ് പാരലൽ പോർട്ട്. കമ്പ്യൂട്ടറിനെ പ്രിന്റർ അല്ലെങ്കിൽ സ്കാനറുമായി ബന്ധിപ്പിക്കുവാൻ പാരലൽ പോർട്ട് ഉപയോഗിക്കുന്നു. ഒരു കമ്പ്യൂട്ടർ സിസ്റ്റത്തിലെ പാരലൽ പോർട്ട് ചിത്രം 2.5 ൽ കാണിക്കുന്നു.

c. USB പോർട്ട്

ഉപകരണങ്ങൾ തമ്മിൽ ഉയർന്ന വേഗതയിൽ ഡാറ്റ വിനിമയം സാധ്യമാക്കുന്ന ഒരു പോർട്ട് ആണ് യൂണിവേഴ്സൽ സീരിയൽ ബസ് (USB).

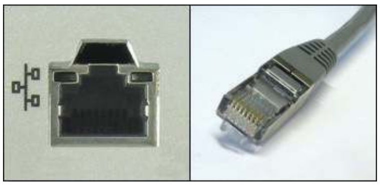


ചിത്രം 2.6: യു.എസ്.ബി പോർട്ടുകൾ

ഉയർന്ന ബാൻഡ്‌വിഡ്ത് ഡാറ്റ കൈമാറ്റത്തിന്റെ വേഗത വർധിപ്പിക്കുന്നു. ഹ്രസ്വ ദൂരത്തേക്കുള്ള ഡാറ്റ വിനിമയത്തിനായി ഇത് ഉപയോഗിക്കുന്നു. കീബോർഡ്, മൗസ്, പ്രിന്റർ, സ്കാനർ, ഫ്ളാഷ് ഡ്രൈവ്, ബാഹ്യ ഹാർഡ് ഡിസ്ക് മുതലായ ഉപകരണങ്ങളെ ബന്ധിപ്പിക്കുന്നതിനായി USB പോർട്ട് ഉപയോഗിക്കുന്നു. വിവിധ തരം USB പോർട്ടുകൾ ചിത്രം 2.6 ൽ കാണിക്കുന്നു. USB പോർട്ടുകളുടെ പ്രധാന ഗുണങ്ങൾ താഴെ പറയുന്നവയാണ്:

ബാഹ്യ ഉപകരണങ്ങളിലേക്ക് വൈദ്യുതോർജ്ജം നൽകുന്നതിന് USB പോർട്ടുകൾക്ക് കഴിയും. ഈ സാങ്കേതികവിദ്യ ബാഹ്യ ഹാർഡ് ഡിസ്ക്, ഫ്ളാഷ് ഡ്രൈവ്, ഡോങ്കിൾ തുടങ്ങിയവ വികസിപ്പിച്ചെടുക്കാൻ കാരണമായി. മൊബൈൽ ഫോണുകൾ, ടാബ്‌ലറ്റുകൾ, ലാപ്ടോപ്പുകൾ തുടങ്ങിയ ഉപകരണങ്ങളിൽ ഇത് ഉപയോഗിക്കുന്നു.

ഉപകരണങ്ങൾ പ്രവർത്തിക്കുമ്പോൾ പോലും USB ഉപകരണങ്ങൾ ബന്ധിപ്പിക്കുകയും വിച്ഛേദിക്കുകയും ചെയ്യാം.

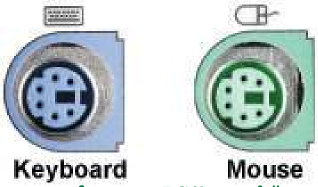


ചിത്രം 2.7: LAN പോർട്ടും RJ45

d. LAN പോർട്ട്

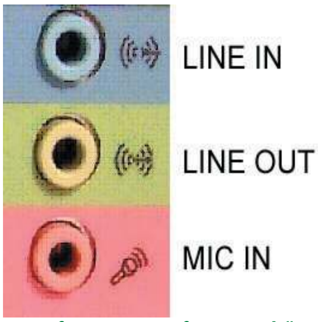
ഇതർനെറ്റ് പോർട്ട്, നെറ്റ്‌വർക്ക് കണക്ഷൻ, നെറ്റ്‌വർക്ക് പോർട്ട് എന്നിങ്ങനെയും, ലോക്കൽ ഏരിയ നെറ്റ്‌വർക്ക് (LAN) പോർട്ട് അറിയപ്പെടുന്നു. ഒരു വയർ ഉപയോഗിച്ച് ഒരു കമ്പ്യൂട്ടറിനെ നെറ്റ്‌വർക്കുമായി ബന്ധിപ്പിക്കുവാൻ LAN പോർട്ട് ഉപയോഗിക്കുന്നു. LAN പോർട്ടുകളിൽ കേബിളുകൾ ബന്ധിപ്പിക്കുന്നതിനുള്ള ഒരു അംഗീകൃത കണക്റ്റർ ആണ് രജിസ്റ്റേർഡ് ജാക്ക് RJ45 (ചിത്രം 2.7 കാണുക).

e. PS/2 പോർട്ട്



ചിത്രം 2.8: PS/2 പോർട്ട്

കീബോർഡ്, മൗസ് എന്നിവ ബന്ധിപ്പിക്കുന്നതിന് ഇന്റർനാഷണൽ ബിസിനസ് മെഷീൻസ് (IBM) കണ്ടുപിടിച്ച പ്രത്യേക തരം പോർട്ടുകളാണ് പേഴ്സണൽ സിസ്റ്റം/2 (PS/2) പോർട്ടുകൾ. (ചിത്രം 2.8 കാണുക). ഇത്തരത്തിലുള്ള പോർട്ടുകൾക്ക് വേഗത വളരെ കുറവാണ്. ഇപ്പോൾ ഇവയ്ക്ക് പകരം USB പോലുള്ള വേഗതയേറിയ പോർട്ടുകൾ ഉപയോഗിച്ചു വരുന്നു.



ചിത്രം 2.9: ഓഡിയോ പോർട്ട്

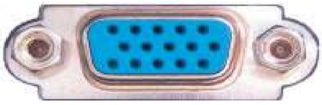
f. ഓഡിയോ പോർട്ട്

സ്പീക്കർ, മൈക്രോഫോൺ മുതലായവ ഓഡിയോ ഉപകരണങ്ങളെ ബന്ധിപ്പിക്കുവാൻ ഓഡിയോ പോർട്ടുകൾ ഉപയോഗിക്കുന്നു. ചിത്രം 2.9 ൽ കാണിച്ചിരിക്കുന്ന മൂന്ന് ചെറിയ കണക്ടറുകൾ ഇവയെ ബന്ധിപ്പിക്കുന്നതിനായി ഉപയോഗിക്കുന്നു:

i. ലൈൻ ഇൻ – ശബ്ദം ഇൻപുട്ട് (മൊബൈൽ ലൈൻ ഔട്ട്, ഐ പോഡ് തുടങ്ങിയ ഓഡിയോ ഉപകരണങ്ങളിൽ നിന്നും)

ii. ലൈൻ ഔട്ട് - പുറത്തേക്കുള്ള ശബ്ദം (ബാഹ്യ സ്പീക്കറുകളിലേക്ക് നിങ്ങളുടെ പിസിയിൽ നിന്നുള്ള ശബ്ദ ഔട്ട്പുട്ട് കണക്ട് ചെയ്യുന്നതിന്)

iii. മൈക്ക് ഇൻ - മൈക്രോഫോൺ ഇൻപുട്ട് പോർട്ട്



ചിത്രം 2.10: VGA പോർട്ട്

g. വീഡിയോ ഗ്രാഫിക്സ് അറേ (VGA) പോർട്ട്

ഐബിഎം വികസിപ്പിച്ചെടുത്ത ഒരു ജനപ്രിയ അംഗീകൃത പ്രദർശന പോർട്ടാണ് VGA. കമ്പ്യൂട്ടർ ഒരു മോണിറ്ററുമായോ പ്രൊജക്ടറുമായോ ബന്ധിപ്പിക്കുന്നതിന് VGA പോർട്ട് ഉപയോഗിക്കുന്നു. ചിത്രം 2.10 ൽ കാണുന്നതുപോലെ മൂന്നു വരികളിലായി 15 പിന്നുകൾ VGA കണക്ടറിനുണ്ട്. 800 x 600 റെസല്യൂഷനുള്ള സൂപ്പർ VGA (SVGA) ആദ്യം വികസിപ്പിച്ചെടുത്തു. പിന്നീട് 1024 x 768 റെസല്യൂഷനുള്ള എക്സ്റ്റെൻഡഡ് ഗ്രാഫിക്സ് അറേ (XGA) പുറത്തിറക്കി. ഒരു മോണിറ്റർ ഏത് റെസല്യൂഷൻ പിന്തുണക്കുന്നു എന്നത് പരിഗണിക്കാതെ, സാധാരണയായി അവയെ VGA എന്ന് വിളിക്കുന്നു.

h. ഹൈ ഡെഫനിഷൻ മൾട്ടിമീഡിയ ഇന്റർഫേസ് (HDMI) പോർട്ട്

ഒരു കേബിൾ മുഖേന ഉയർന്ന ഡെഫനിഷൻ വീഡിയോയും ഒന്നിലധികം ചാനൽ ഓഡിയോകളും പകർത്താൻ കഴിവുള്ള ഒരു ഡിജിറ്റൽ കണക്ഷൻ ആണ് HDMI (ചിത്രം 2.11). അനലോഗ് കേബിളുകൾ ഉപയോഗിച്ച് ഇതേ കാര്യം ചെയ്യാൻ, നമുക്ക് അനേകം വീഡിയോ, ഓഡിയോ കേബിളുകൾ ബന്ധിപ്പിക്കേണ്ടതുണ്ട്.

ചിത്രം 2.11: HDMI പോർട്ട്

2.1.4 മെമ്മറി (Memory)

ഡാറ്റയോ, നിർദ്ദേശങ്ങളോ, ഫലങ്ങളോ താൽക്കാലികമായോ സ്ഥിരമായോ സൂക്ഷിച്ചു വയ്ക്കാനുള്ള സ്ഥലമാണ് മെമ്മറി. മെമ്മറിയെ രണ്ടായി തരംതിരിച്ചിരിക്കുന്നു. പ്രാഥമിക മെമ്മറി, ദ്വിതീയ മെമ്മറി. മദർബോർഡിൽ സ്ഥിതി ചെയ്യുന്നതും പ്രോസസ്സറുമായി നേരിട്ട് ബന്ധപ്പെടുന്നതുമായ മെമ്മറിയാണ് പ്രാഥമിക മെമ്മറി. സ്ഥിരമായി വിവരങ്ങൾ സൂക്ഷിക്കാൻ ഉപയോഗിക്കുന്നതും പ്രോസസ്സറുമായി പ്രാഥമിക മെമ്മറിയിലൂടെ മാത്രം വിവരങ്ങൾ കൈമാറുന്നതുമായ മെമ്മറിയാണ് ദ്വിതീയ മെമ്മറി. മെമ്മറിയെ കുറിച്ച് കൂടുതൽ പഠിക്കുന്നതിന് മുമ്പായി മെമ്മറി അളക്കുന്ന യൂണിറ്റുകളെ മനസിലാക്കാം. താഴെ പറയുന്നവയാണ് ഈ അളവിന്റെ ഏകകങ്ങൾ.

ബൈനറി ഡിജിറ്റ് = 1 ബിറ്റ്	1 MB (മെഗാ ബൈറ്റ്)= 1024 KB
1 നിബ്ൾ = 4 ബിറ്റ്	1 GB (ജിഗാബൈറ്റ്) = 1024 MB
1 ബൈറ്റ് = 8 ബിറ്റ്സ്	1 TB (ടെറാ ബൈറ്റ്) = 1024 GB
1 KB (കിലോ ബൈറ്റ്) = 1024 ബൈറ്റ്സ്	1 PB (പെറ്റാ ബൈറ്റ്) = 1024 TB

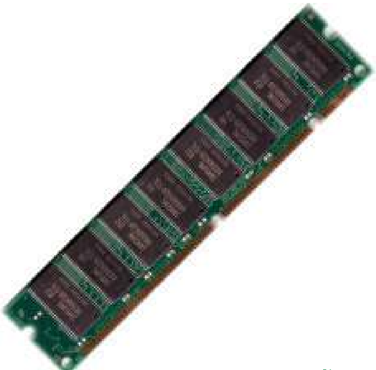
a. പ്രാഥമിക മെമ്മറി

പ്രാഥമിക മെമ്മറി എന്നത് സെമികണ്ടക്ടർ മെമ്മറിയാണ്. ഇതിനെ CPU നേരിട്ട് കൈകാര്യം ചെയ്യുന്നു. ഇതിന് ഡാറ്റ വളരെ വേഗത്തിൽ അയക്കുന്നതിനും സ്വീകരിക്കുന്ന

തിനും കഴിവുണ്ട്. 3 തരത്തിലുള്ള മെമ്മറിയാണ് ഇതിൽ ഉൾപ്പെട്ടിരിക്കുന്നത്. അവ റാം, റോം, ക്യാഷ് മെമ്മറി എന്നിവയാണ്.

i. റാൻഡം ആക്സസ് മെമ്മറി (RAM)

മൈക്രോ പ്രൊസസ്സറിന് ഡാറ്റ സംഭരിക്കാനും തിരിച്ചെടുക്കാനും സാധിക്കുന്ന RAM എന്ന പ്രാഥമിക മെമ്മറി ചിത്രം 3.6 ൽ കാണിച്ചിരിക്കുന്നു. RAM നുള്ളിൽ ഡാറ്റ എവിടെ നിന്ന് വേണമെങ്കിലും ശേഖരിക്കാനോ തിരിച്ചെടുക്കാനോ സാധിക്കും. CPU പ്രൊസസ്സ് ചെയ്യുന്ന ഡാറ്റയോ നിർദ്ദേശങ്ങളോ റാമിനുള്ളിൽ നിർബന്ധമായും ഉണ്ടായിരിക്കണം. വൈദ്യുത ബന്ധം നിലക്കുമ്പോൾ റാമിനുള്ളിൽ അടങ്ങിയിരിക്കുന്നതെല്ലാം നഷ്ടപ്പെടുന്നു. അതിനാൽ റാം ഒരു അസ്ഥിരമെമ്മറിയാണ് റാമിന്റെ സംഭരണശേഷി സാധാരണ ജിഗാബൈറ്റിലാണ് പറയാറുള്ളത്.



ചിത്രം 2.12 ഞഅങ പോർട്ട്

എത്ര മാത്രം വേഗത്തിൽ ഡാറ്റ മെമ്മറിയിൽ സംഭരിക്കുന്നു/തിരിച്ചെടുക്കുന്നു എന്നതാണ് റാമിന്റെ വേഗത കൊണ്ടുദ്ദേശിക്കുന്നത്. ഇത് അളക്കുന്നത് മെഗാ ഹെർട്സിൽ ആണ് (MHz). ഒരു കമ്പ്യൂട്ടർ പ്രവർത്തിച്ചുകൊണ്ടിരിക്കുമ്പോൾ അതിലെ റാമിൽ താഴെ പറയുന്നവ ഉണ്ടായിരിക്കും.

1. ഓപ്പറേറ്റിംഗ് സിസ്റ്റം
2. നിലവിൽ ഉപയോഗിച്ച് കൊണ്ടിരിക്കുന്ന ആപ്ലിക്കേഷൻ സോഫ്ട്വെയർ
3. പ്രൊസസ്സ് ചെയ്തുകൊണ്ടിരിക്കുന്ന ഡാറ്റ

ചിത്രം 2.13 ഞഛങ ചിപ്പ്

ii. റീഡ് ഓൺലി മെമ്മറി (ROM)

ROM എന്നത് സ്ഥിരമായ മെമ്മറിയാണ്. അതിൽ നിന്ന് ഡാറ്റ വീണ്ടെടുക്കുവാൻ മാത്രമേ കഴിയൂ. ഇതിന്റെ ഉള്ളടക്കം എളുപ്പത്തിൽ മാറ്റാൻ സാധിക്കില്ല. വൈദ്യുതി ബന്ധം നിലച്ചാലും ഇതിലെ ഉള്ളടക്കം മാഞ്ഞു പോകാതെ നിലനിൽക്കുന്നു. BIOS എന്ന പേരിൽ അറിയപ്പെടുന്ന ഒരു പ്രത്യേക ബൂട്ട് അപ്പ് (boot up) പ്രോഗ്രാം റോമിൽ അടങ്ങിയിരിക്കുന്നു. ROM ചിപ്പാണ് ചിത്രം 2.13 ൽ കാണിച്ചിരിക്കുന്നത്. കമ്പ്യൂട്ടർ ഓൺ ചെയ്യുമ്പോഴോ 'ബൂട്ട് അപ്പ്' ചെയ്യുമ്പോഴോ ഈ സോഫ്റ്റ്വെയർ പ്രവർത്തിക്കുന്നു. ഇത് കമ്പ്യൂട്ടർ ഹാർഡ്‌വെയറിനെ പരിശോധിക്കുകയും ഓപ്പറേറ്റിംങ് സിസ്റ്റത്തിനെ മെമ്മറിയിലേക്ക് കൊണ്ടുവരികയും ചെയ്യുന്നു. ROM ന്റെ ചില പരിഷ്കരിച്ച രൂപങ്ങൾ താഴെകൊടുക്കുന്നു.

1. പ്രോഗ്രാമബിൾ റീഡ് ഓൺലി മെമ്മറി (PROM) (ഒരിക്കൽ മാത്രം പ്രോഗ്രാം ചെയ്യപ്പെടുന്ന മെമ്മറിയാണിത്)

2. ഇറേസബിൾ പ്രോഗ്രാമബിൾ റീഡ് ഓൺലി മെമ്മറി (EPROM) (അൾട്രാവയലറ്റ് റേഡിയേഷൻ ഉപയോഗിച്ച് മായ്ച്ച് വീണ്ടും എഴുതാൻ കഴിയുന്ന മെമ്മറിയാണിത്)
3. ഇലക്ട്രിക്കലി ഇറേസബിൾ പ്രോഗ്രാമബിൾ റീഡ് ഓൺലി മെമ്മറി (EEPROM) (വൈദ്യുതി ഉപയോഗിച്ച് മായ്ച് വീണ്ടും എഴുതാൻ കഴിയുന്ന മെമ്മറിയാണിത്.)

പട്ടിക 2.1 ൽ റാമിന്റെയും റോമിന്റെയും താരതമ്യം കൊടുത്തിരിക്കുന്നു.

റാൻഡം ആക്സസ് മെമ്മറി (RAM)	റീഡ് ഓൺലി മെമ്മറി (ROM)
• ഇത് റോമിനേക്കാൾ വേഗത കൂടിയതാണ്	• ഇതിന് വേഗത കുറവാണ്.
• കമ്പ്യൂട്ടർ പ്രവർത്തിക്കുമ്പോൾ ഡാറ്റയും, അപ്ലിക്കേഷൻ പ്രോഗ്രാമും ഓപ്പറേറ്റിംഗ് സിസ്റ്റവും സൂക്ഷിക്കുന്നു.	• കമ്പ്യൂട്ടർ ബൂട്ട് ചെയ്യുവാനുള്ള പ്രോഗ്രാം ഇത് സൂക്ഷിക്കുന്നു.
• ഡാറ്റയുടെ സംഭരണവും വീണ്ടെടുക്കലും ഇത് അനുവദിക്കുന്നു.	• സാധാരണയായി ഇതിൽ നിന്ന് ഡാറ്റ തിരിച്ചെടുക്കാൻ മാത്രമേ സാധിക്കുകയുള്ളൂ.
• കമ്പ്യൂട്ടർ ഓഫാക്കിയാൽ ഇതിലെ ഉള്ളടക്കം നഷ്ടപ്പെടുന്നതിനാൽ ഇതൊരു അസ്ഥിരമെമ്മറിയാണ്.	• കമ്പ്യൂട്ടർ ഓഫ് ആക്കിയാലും ROM ലെ ഉള്ളടക്കം നഷ്ടപ്പെടാത്തതിനാൽ ഇതൊരു സ്ഥിര മെമ്മറിയാണ്.

പട്ടിക 2.1: RAM- ROM എന്നിവയുടെ താരതമ്യം

iii. ക്യാഷ് മെമ്മറി

പ്രൊസസ്സറിന്റെയും റാമിന്റെയും (അഥവാ മെയിൻ മെമ്മറിയുടെയും) ഇടയ്ക്ക് ഉള്ള ചെറുതും വേഗതയേറിയതുമായ മെമ്മറിയാണ് ക്യാഷ് മെമ്മറി. ഇടയ്ക്കിടെ ആവശ്യമായി വരുന്ന ഡാറ്റയും, നിർദ്ദേശങ്ങളും ഇടക്കാല ഫലങ്ങളും വേഗത്തിൽ എടുക്കുവാൻവേണ്ടി ക്യാഷ് മെമ്മറിയിൽ സൂക്ഷിക്കുന്നു. പ്രോസസ്സർ റാമിലെ ഒരു ലൊക്കേഷനിൽ സംഭരിക്കുകയോ അതിൽ നിന്ന് തിരിച്ചെടുക്കുകയോ ചെയ്യുമ്പോൾ ആദ്യം ക്യാഷ് മെമ്മറിയിൽ ഡാറ്റയുടെ ഒരു കോപ്പി ഉണ്ടോ എന്ന് പരിശോധിക്കുന്നു. അങ്ങിനെയാണെങ്കിൽ പ്രൊസസ്സർ ക്യാഷ് മെമ്മറിയിൽ നിന്നും ഇത് പെട്ടെന്ന് വീണ്ടെടുക്കുന്നു. റാമിനെക്കാൾ വേഗത്തിൽ ക്യാഷ് മെമ്മറിയിൽ നിന്നും ഡാറ്റ തിരിച്ചെടുക്കുന്നു. റാമിനേക്കാളും വിലയേറിയതാണ് ക്യാഷ് മെമ്മറി. CPU വിന്റെയും മദർബോർഡിന്റെയും ഉള്ളിലുള്ള ക്യാഷ് മെമ്മറി സിസ്റ്റത്തിന്റെ പ്രകടനം മെച്ചപ്പെടുത്തുന്നതിന് സഹായിക്കുന്നു.

b. ദ്വിതീയ സംഭരണി (Secondary/Auxilary storage)

ദ്വിതീയ മെമ്മറി സ്ഥിരമാണ്. റാമിൽ നിന്നും നിന്നും വ്യത്യസ്തമായി, കമ്പ്യൂട്ടറിന്റെ വൈദ്യുതബന്ധം വിച്ഛേദിച്ചാലും ഈ ഉപകരണങ്ങളിൽ സംഭരിച്ചിട്ടുള്ള ഡാറ്റ ഒരിക്കലും നഷ്ടപ്പെടുന്നില്ല. ദ്വിതീയ മെമ്മറി റാമിനേക്കാൾ സംഭരണശേഷി വളരെ കൂടുതലുള്ളതാണ്. എന്നാൽ ഇതിന് വേഗത കുറവാണ്. പ്രോഗ്രാമും ഡാറ്റയും ഇതിൽ സൂക്ഷിക്കുന്നുണ്ടെങ്കിലും പ്രോസസ്സറിന് അവയെ നേരിട്ട് ഉപയോഗിക്കാൻ സാധിക്കില്ല. ഒരു കമ്പ്യൂട്ടറിൽ നിന്നും വേറൊരു കമ്പ്യൂട്ടറിലേക്ക് ഡാറ്റയോ, പ്രോഗ്രാമുകളോ കൈമാറ്റം ചെയ്യാൻ

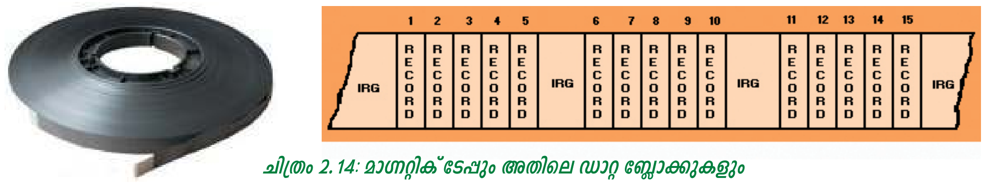
ദ്വിതീയ മെമ്മറി ഉപയോഗിക്കുന്നു. ഇത് ഒരു പിന്തുണാസംഭരണി (ബാക്ക് അപ്പ്) ആയും ഉപയോഗിക്കുന്നു. സംഭരണ ഉപകരണങ്ങളുടെ പ്രധാന തരം തിരിവുകളാണ് മാഗ്നറ്റിക്, ഒപ്റ്റിക്കൽ, സെമികണ്ടക്ടർ മെമ്മറി എന്നിവ.

i. കാന്തിക സംഭരണ ഉപകരണങ്ങൾ (Magnetic storage devices)

കാന്തിക വസ്തുക്കൾ ലേപനം ചെയ്ത പ്ലാസ്റ്റിക് ടേപ്പോ മെറ്റൽ/പ്ലാസ്റ്റിക് ഡിസ്കോ ആണ് കാന്തിക സംഭരണ ഉപകരണങ്ങളായി ഉപയോഗിക്കുന്നത്. ഈ ഉപകരണങ്ങളിൽ ഡാറ്റ കാന്തികമായി റെക്കോർഡ് ചെയ്യപ്പെടുന്നു. ഡാറ്റ ഈ ഉപകരണങ്ങളിൽ നിന്നും ഉപയോഗിക്കുന്നത് റീഡ്/റൈറ്റ് ഹെഡ് ഉപയോഗിച്ചാണ്. ചില പ്രശസ്തമായ കാന്തിക സംഭരണ ഉപകരണങ്ങളാണ് മാഗ്നറ്റിക് ടേപ്പുകൾ (Magnetic tapes), ഹാർഡ് ഡിസ്കുകൾ (Hard disks) തുടങ്ങിയവ.

കാന്തിക ടേപ്പ് (Magnetic tape)

ഭീമമായ അളവിൽ ഡാറ്റ സംഭരിക്കുന്നതിനുള്ള ഒരു പിന്തുണാ സംഭരണ (ബാക്ക് അപ്) ഉപകരണമാണ് മാഗ്നറ്റിക് ടേപ്പ്. മറ്റ് കാന്തിക സംഭരണ ഉപകരണങ്ങളെ അപേക്ഷിച്ച് ഒരു യൂണിറ്റ് അളവ് ടേപ്പിന്റെ വില താരതമ്യേന കുറവാണ്.



ചിത്രം 2.14: മാഗ്നറ്റിക് ടേപ്പും അതിലെ ഡാറ്റ ബ്ലോക്കുകളും

നേർത്ത കാന്തിക പദാർഥം പൂശിയ കനം കുറഞ്ഞ ഒരു ടേപ്പ് അടങ്ങുന്ന ആലേഖന മാധ്യമം ആണിത്. അനലോഗ് ഡാറ്റയോ ഡിജിറ്റൽ ഡാറ്റയോ ആലേഖനം ചെയ്യാൻ ഇതുപയോഗിക്കാം. ഒരു റീഡ്/റൈറ്റ് ഹെഡ് ഉപയോഗിച്ച് ടേപ്പിന്റെ വീതിയ്ക്ക് കുറുകെ ഫ്രെയിമുകളായാണ് ഡാറ്റാ സംഭരിക്കുന്നത്. ഇത്തരം ഫ്രെയിമുകളെ ഒരുമിച്ച് ചേർത്ത് ബ്ലോക്കുകൾ അഥവാ റെക്കോർഡുകളാക്കി മാറ്റുകയും ചിത്രം 2.14 ൽ കാണിച്ചിരിക്കുന്നതു പോലെ അവയെ ഗ്യാപ്പുകളിലൂടെ മറ്റ് ബ്ലോക്കുകളിൽ നിന്ന് വേർതിരിച്ച് നിർത്തുകയും ചെയ്യുന്നു.

മാഗ്നറ്റിക് ടേപ്പ് ഒരു ഓഡിയോ കാസറ്റ് പോലെ അനുവർത്തന സമീപന ശൈലിയിലുള്ള (sequential access mode) മാധ്യമം ആയതിനാൽ, അതിൽ ഒരു ഡാറ്റയുടെ സ്ഥാനം കണ്ടെത്താൻ കൂടുതൽ സമയമെടുക്കും. ഉദാഹരണമായി, ടേപ്പിലെ 100-ാമത്തെ റെക്കോർഡ് വേണമെങ്കിൽ അതിന് മുമ്പിലുള്ള 99 റെക്കോർഡുകളിലൂടെയും കടന്നു പോയേ മതിയാവൂ. ടേപ്പ് മാധ്യമത്തിന്റെ സംഭരണ ശേഷി ടെറാബൈറ്റുകളിലാണ് സൂചിപ്പിക്കുന്നത്.

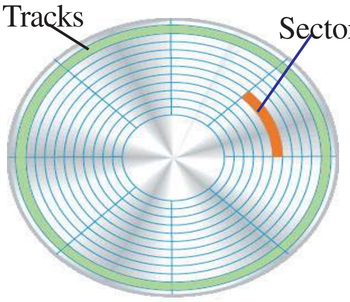
ഹാർഡ് ഡിസ്ക്

ഹാർഡ് ഡിസ്ക് എന്നത് പൊടി കടക്കാത്ത പെട്ടിക്കുള്ളിൽ ഉള്ളടക്കം ചെയ്തിട്ടുള്ള കാന്തിക പദാർഥം പൂശിയ ലോഹത്തകിടുകളാണ്. ഹാർഡ് ഡിസ്കുകൾക്ക് വളരെ

ഉയർന്ന സംഭരണ ശേഷിയും ഉയർന്ന ഡാറ്റാ വിനിമയ നിരക്കും കുറഞ്ഞ സമീപന സമയവും (access time) ആണുള്ളത്. ഇത് കൂടുതൽ ശാശ്വതവും എന്നാൽ പിഴവുകൾക്ക് സാധ്യതയില്ലാത്തതുമാണ്. കമ്പ്യൂട്ടറുകളിൽ സർവസാധാരണമായി ഉപയോഗിക്കുന്ന ദ്വിതീയ സംഭരണ ഉപകരണമാണിത്.

ചിത്രം 2.15: ഹാർഡ് ഡിസ്ക്

ഒരു ഹാർഡ് ഡിസ്കിൽ ഒന്നോ അതിലധികമോ താലങ്ങൾ (platters) അടങ്ങിയിരിക്കും. ഓരോ താലത്തിലും ഒരു പ്രതലത്തിന് ഒന്ന് എന്ന കണക്കിൽ രണ്ട് റീഡ്/ റൈറ്റ് ഹെഡുകളുണ്ടാകും. ഡാറ്റ സംഭരിക്കുന്നതിനായി ചിത്രം 2.15 ൽ കാണുന്നതുപോലെ ഒരു സമീപന കരത്തിൽ (access arm) ഘടിപ്പിച്ചിട്ടുള്ള ചെറിയൊരു വിദ്യുത് കാന്തിക (eletcromagnetic) റീഡ്/ റൈറ്റ് ഹെഡ് ഡിസ്കിന്റെ പ്രതലത്തിലെ ചെറിയ പുള്ളികളെ കാന്തികമാക്കുന്നു. ഇതേ വിദ്യുത് കാന്തിക ഹെഡ് തന്നെ പിന്നീട് ഈ പുള്ളികളിലെ കാന്തിക മണ്ഡലം തിരിച്ചറിയുകയും ഡിസ്കിൽ നിന്നും ഡാറ്റ തിരികെ വായിച്ചെടുക്കുകയും ചെയ്യുന്നു.



ചിത്രം 2.16: താലം (platter)

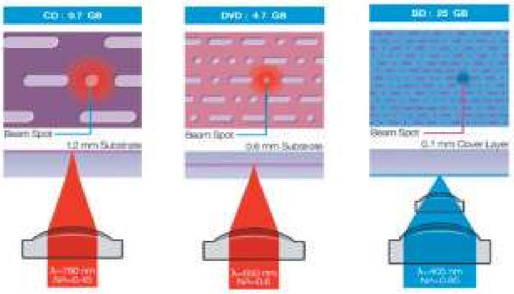
താലത്തിന്റെ പ്രതലത്തിൽ ട്രാക്കുകളിലും സെക്ടറുകളിലുമായാണ് ഡാറ്റ സംഭരിക്കുന്നത്. ചിത്രം 2.16 ൽ കാണുന്നതുപോലെ ഒരു പ്രതലത്തിലെ ഏകകേന്ദ്ര വൃത്തങ്ങളെ ട്രാക്കുകൾ എന്നും പ്രതലത്തിലെ പൈ-കഷണങ്ങളെപ്പോലെയുള്ള ഭാഗങ്ങളെ സെക്ടറുകൾ എന്നും പറയുന്നു. ഡിസ്കിൽ ട്രാക്കുകളും സെക്ടറുകളും സജ്ജമാക്കുന്ന പ്രവർത്തനത്തെ ഡിസ്ക് ഫോർമാറ്റിംഗ് എന്നു പറയുന്നു. ഇതിനു ശേഷം മാത്രമെ റീഡ്/റൈറ്റ് പ്രവർത്തനങ്ങൾ ഡിസ്കിൽ ചെയ്യാൻ കഴിയുകയുള്ളൂ. ഫോർമാറ്റ് ചെയ്യുന്ന ഡിസ്കിൽ ഡാറ്റ ഉണ്ടെങ്കിൽ അത് മുഴുവനും നഷ്ടപ്പെടും.

സംഭരണ ശേഷി, വേഗത, സമീപന സമയം എന്നിവയെ ആസ്പദമാക്കി ഹാർഡ് ഡിസ്ക് ഡ്രൈവുകളെ വിലയിരുത്തുന്നു അഥവാ റേറ്റിംഗ് നടത്തുന്നു.

സംഭരണ ശേഷി	:	500 GB, 1 TB യോ അതിലധികമോ
വേഗത	:	എത്ര വേഗത്തിലാണ് ഡിസ്ക് കറങ്ങുന്നത്, ഒരു മിനിട്ടിൽ എത്ര തവണ (rotations per minute - rpm), സാധാരണയായി 5400 rpm / 7200 rpm
സമീപന സമയം	:	ഡിസ്കിൽ നിന്നും ഡാറ്റ ബിറ്റുകൾ എടുക്കാൻ വേണ്ടി വരുന്ന സമയം (മില്ലി സെക്കൻഡിൽ)

വലിയ ഡിസ്കുകളെ യുക്താനുസൃതം (അഭൗതികമായി) വിവിധ ഭാഗങ്ങൾ അഥവാ വാല്യങ്ങളായി (volumes)വിഭജിക്കാവുന്നതാണ്. ഈ വാല്യങ്ങൾ സ്വതന്ത്രമായിരിക്കും, അതുകൊണ്ടു തന്നെ അവയെ വെവ്വേറെ ഫോർമാറ്റ് ചെയ്യാനും സാധിക്കും.

ii. ഒപ്റ്റിക്കൽ സംഭരണ ഉപകരണങ്ങൾ



ചിത്രം 2.17: CD, DVD, ബ്ലൂ - റേ ഡിസ്കുകൾ - കുഴികളും പരന്ന പ്രതലങ്ങളും

ഡാറ്റ വായിക്കുന്നതിനും എഴുതുന്നതിനും (read and write) ശക്തി കുറഞ്ഞ ലേസർ കിരണം ഉപയോഗിക്കുന്ന ഡാറ്റ സംഭരണ മാധ്യമമാണ് ഒപ്റ്റിക്കൽ ഡിസ്ക്. വൃത്താകൃതിയിലുള്ള രണ്ട് പ്ലാസ്റ്റിക് തകിടുകൾക്കിടയിൽ തിരുകിവച്ചിരിക്കുന്ന വളരെ കനം കുറഞ്ഞ ഒരു അലുമിനിയം ലോഹ പാളി ഇതിനുണ്ട്. ചിത്രം 2.17 ൽ കാണുന്നതുപോലെ തുടർച്ചയായ സർപ്പിളാകൃതിയിൽ (spiral) കുഴികളും നിരന്ന പ്രതലങ്ങളുമായാണ് (pits and lands) ഇതിൽ ഡാറ്റ രേഖപ്പെടുത്തുന്നത്. ഈ കുഴികളെയും പ്രതലങ്ങളെയും ലേസർ കിരണം പൂജ്യങ്ങളും (0) ഒന്നുകളും (1) ആയാണ് വായിക്കുന്നത്. വലിയ തോതിലുള്ള ഇവയുടെ ഉത്പാദനചെലവ് കുറവായതിനാൽ ഇവ പ്രചാരമേറിയ ദ്വിതീയ സംഭരണ മാധ്യമമാണ്. CD, DVD, ബ്ലൂ റെ എന്നിവ പ്രധാന ഒപ്റ്റിക്കൽ ഡിസ്കുകളാണ്.

ചിത്രം 2.18: CD

കോംപാക്റ്റ് ഡിസ്ക് (CD)

700 MB ഡാറ്റ വരെ ശേഖരിക്കാൻ ശേഷിയുള്ള ഒരു ഒപ്റ്റിക്കൽ സംഭരണ മാധ്യമമാണ് കോംപാക്റ്റ് ഡിസ്ക്. ഒരു CD ഡ്രൈവ് CDയിൽ നിന്നും ഡാറ്റ വായിക്കുന്നതിനും അതിലേയ്ക്ക് എഴുതുന്ന തിന്നും ചുവന്ന ലേസർ കിരണം ഉപയോഗിക്കുന്നു. രണ്ട് തരം CD കളാണുള്ളത് - CD-R ഉം CD-RW ഉം. CD-Rൽ (CD-റെക്കോർഡബിൾ) ഒരു തവണ ഡാറ്റ എഴുതാനും എത്ര തവണ വേണമെങ്കിലും വായിക്കാനും കഴിയും. എന്നാൽ CD-RW (CD-റീറൈറ്റബിൾ) ഡിസ്കിൽ സംഭരിച്ചിരിക്കുന്ന ഡാറ്റയെ എപ്പോൾ വേണമെങ്കിലും മായ്ച്ച് കളയാനും വീണ്ടും എഴുതാനും സാധിക്കും.

ഡിജിറ്റൽ വെർസറ്റയിൽ ഡിസ്ക് (DVD)

CD-ROMന് സമാനമായ, എന്നാൽ അതിനേക്കാൾ ഉയർന്ന സംഭരണശേഷിയുള്ള ഒരു ഒപ്റ്റിക്കൽ സംഭരണ മാധ്യമമാണ് ഡിജിറ്റൽ വെർസറ്റയിൽ ഡിസ്ക്. ഡാറ്റ ആലേഖനം ചെയ്യുന്നതിന് ചെറിയ സ്ഥാനങ്ങൾ ഉപയോഗപ്പെടുത്തിയാണ് ഈ ശേഷി കൈവരിച്ചിരിക്കുന്നത്. ഡാറ്റ എഴുതുന്നതും വായിക്കുന്നതും DVD ഡ്രൈവ് ഉപയോഗിച്ചാണ്. ഈ പ്രവർത്തനങ്ങൾക്കായി ഇവിടെയും ചുവന്ന ലേസർ കിരണങ്ങളാണ് ഉപയോഗിക്കുന്നത്. ഒരു DVD യുടെ സംഭരണ ശേഷി 4.37 GB മുതൽ 15.9 GB വരെയാണ്. പ്രധാനമായും മൂന്ന് തരം DVD കളാണുള്ളത് - DVD-ROM, DVD-RW, DVD-RAM.

DVD-ROM : DVD-റീഡ് ഓൺലി മെമ്മറി CD-ROMകൾ പോലെ പ്രവർത്തിക്കുന്നു.

DVD-RW : DVD റീ റൈറ്റബിൾ ഡിസ്ക്കുകളിൽ എപ്പോൾ വേണമെങ്കിലും ഡാറ്റ മായ്ച്ച് കളയാനും പുനരാലേഖനം ചെയ്യാനും സാധിക്കും.

DVD-RAM : DVD-റാൻഡം ആക്സസ് മെമ്മറി ഡിസ്ക്കുകളിൽ ഡാറ്റ ആവർത്തിച്ച് ആലേഖനം ചെയ്യാനും മായ്ച്ച് കളയാനും സാധിക്കും. DVD-RAM ഘടന പിന്തുണക്കുന്ന കമ്പനികൾ നിർമ്മിക്കുന്ന ഉപകരണങ്ങൾക്ക് മാത്രമെ ഇവ അനുഗുണമാവുകയുള്ളു. ഇതിനെ DVD-റീറൈറ്റബിൾ ഡിസ്ക്കുമായി സാമ്യപ്പെടുത്താവുന്നതാണ്, എന്നാൽ അവയെക്കാൾ ഉയർന്ന ഉപയോഗ കാലയളവുള്ളതും അടുത്തടുത്ത് മായ്ച്ച് ഉപയോഗിക്കാവുന്നതുമാണ്.

DVD- 5 എന്നറിയപ്പെടുന്ന ഒരു സാധാരണ DVD 4.37 GB ഡാറ്റ സംഭരിക്കും, എന്നാൽ DVD- 18 എന്നറിയപ്പെടുന്ന ഇരട്ട വശങ്ങളുള്ള ദ്വന്ദ ലെയറുകളോട് കൂടിയ DVD 15.9GB ഡാറ്റ സംഭരിക്കും

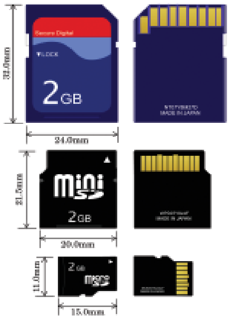
ബ്ലൂ-റേ DVD

ഹൈ ഡെഫനിഷൻ (HD) വീഡിയോകൾ ആലേഖനം ചെയ്യുന്നതിനും പുനരാലേഖനം ചെയ്യുന്നതിനും അവ വീണ്ടും കാണുന്നതിനുമായി വികസിപ്പിച്ചെടുത്ത ഒപ്റ്റിക്കൽ ഡിസ്ക് ഘടനയാണ് ബ്ലൂ-റേ. ഇതിൽ ഭീമമായ അളവിലുള്ള ഡാറ്റ സംഭരിക്കാനാവും. CD/DVD സാങ്കേതികവിദ്യകൾ എഴുതുന്നതിനും വായിക്കുന്നതിനും ചുവന്ന ലേസർ കിരണങ്ങളാണ് ഉപയോഗിക്കുന്നതെങ്കിൽ ബ്ലൂ-റേ ഫോർമാറ്റ് നീല-വയലറ്റ് ലേസറാണ് ഉപയോഗിക്കുന്നത്. അതുകൊണ്ടാണ് ബ്ലൂ-റേ എന്ന പേരുണ്ടായതും. ചുവന്ന ലേസറിനെ അപേക്ഷിച്ച് നീല-വയലറ്റ് ലേസറിന് കുറഞ്ഞ തരംഗ ദൈർഘ്യം (wave length) ഉള്ളതിനാൽ വളരെ കൃത്യതയോടെ ലേസർ സ്ഥാനങ്ങൾ കേന്ദ്രീകരിക്കാൻ കഴിയും എന്നതാണ് ഇത് ഉപയോഗിക്കുന്നതുകൊണ്ടുള്ള മെച്ചം. ഡാറ്റയെ കൂടുതൽ അടുക്കി ക്രമീകരിക്കാൻ ഇത് സഹായിക്കുന്നു. അതുകൊണ്ട് തന്നെ CD/DVD യുടെ അതേ വലുപ്പമെ ഉള്ളുവെങ്കിലും കൂടുതൽ ഡാറ്റ ഇതിൽ സംഭരിക്കാനാകും. ഒരു സാധാരണ DVD യുടെ അഞ്ച് മടങ്ങ് സംഭരണ ശേഷി ഇവയ്ക്ക് നൽകാനാകും. ഏക ലെയർ ഡിസ്കിന് 25 GB വരെയും ദ്വന്ദ ലെയർ ഡിസ്കിന് 50 GB വരെയും സംഭരിക്കാനാകും.

ചിത്രം 2.19 : ഫ്ളാഷ് ഡ്രൈവ്

iii. അർദ്ധചാലക സംഭരണി (ഫ്ളാഷ് മെമ്മറി)

ഡാറ്റ സംഭരണത്തിനായി ഫ്ളാഷ് ഡ്രൈവുകൾ EEPROM ചിപ്പുകൾ ഉപയോഗിക്കുന്നു. അവയിൽ ചലനാത്മക ഭാഗങ്ങൾ ഇല്ലാത്തതുകൊണ്ട് തന്നെ അവ ആഘാതത്തെ പ്രതിരോധിക്കുന്നു. മറ്റ് ദ്വിതീയ സംഭരണികളുമായി താരതമ്യപ്പെടുത്തുമ്പോൾ ഫ്ളാഷ് മെമ്മറി വേഗതയേറിയതും ശാശ്വതമായതുമാണ്. വിവിധ തരം ഫ്ളാഷ് മെമ്മറികൾ ഇന്ന് ലഭ്യമാണ്.



ചിത്രം 2.20: ഫ്ളാഷ് മെമ്മറി കാർഡുകൾ

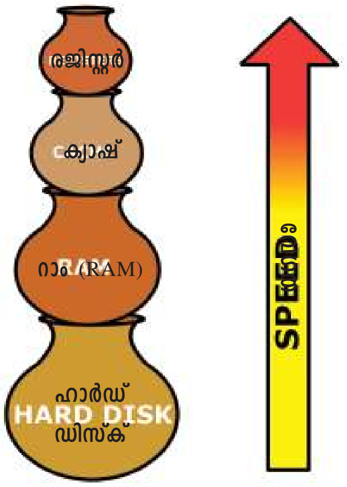
USB ഫ്ളാഷ് ഡ്രൈവ്

ഫ്ളാഷ് മെമ്മറി അടങ്ങിയിട്ടുള്ള, നമ്മുടെ തള്ളവിരലിന്റെ വലുപ്പമുള്ള ഒരു ബാഹ്യസംഭരണ ഉപകരണമാണ് ഫ്ളാഷ് ഡ്രൈവ്. ഫ്ളാഷ് ഡ്രൈവുകൾ കൊണ്ടുനടക്കാൻ എളുപ്പവും പുനരാലേഖനം സാധ്യമായവയുമാണ്. നിലവിലുള്ള USB ഡ്രൈവുകളുടെ സംഭരണ ശേഷി 2 GB മുതൽ 32 GB വരെയാണ്.

ഫ്ളാഷ് മെമ്മറി കാർഡുകൾ

മറ്റൊരുതരം ഫ്ളാഷ് മെമ്മറിയാണ് ഫ്ളാഷ് മെമ്മറി കാർഡ്. അവ പരന്നതും ഏതാണ്ട് 2 mm കനത്തിൽ 1 ഇഞ്ച് X 0.75 ഇഞ്ച് വലുപ്പമുള്ളതുമാണ്. നിലവിലുള്ള മെമ്മറി കാർഡുകൾക്ക് 1 GB മുതൽ 32 GB വരെ സംഭരണ ശേഷിയുണ്ട്. ഫ്ളാഷ് മെമ്മറിയുടെ ചെറു പതിപ്പുകൾ സെൽ ഫോണുകൾ, ടാബ്‌ലറ്റുകൾ മുതലായവയിലും ഉപയോഗിക്കുന്നു. (ചിത്രം 2.20 ശ്രദ്ധിക്കുക). 1 mm ൽ താഴെ കനമുള്ള ഈ ചെറിയ കാർഡുകളുടെ വലുപ്പം ഏതാണ്ട് 6 mm X 3 mm ആയിരിക്കും.

കമ്പ്യൂട്ടറിൽ മെമ്മറിയുടെ പ്രാധാന്യം



ചിത്രം 2.21: മെമ്മറി ശ്രേണി

തൊഴിലാളികളുടെ ശമ്പളം തയാറാക്കാൻ ഉപയോഗിക്കുന്ന ഒരു പേ റോൾ പ്രോഗ്രാം പരിഗണിക്കുക. എല്ലാ തൊഴിലാളികളുടെയും ഡാറ്റ ഹാർഡ് ഡിസ്കിൽ ലഭ്യമായിരിക്കും. ഓരോ തൊഴിലാളിയെയും കുറിച്ചുള്ള വിവരങ്ങൾ റാമിലേക്ക് എടുക്കുന്നു. അവിടെ നിന്നും ശമ്പളം കണക്കു കൂട്ടുന്നതിനാവശ്യമായ വിവരങ്ങൾ (ബോണസ്, കുറയ്ക്കേണ്ടവ എന്നിവ) ക്യാഷ് മെമ്മറിയിലേക്കും എടുക്കുന്നു. എത്ര മണിക്കൂർ ജോലി ചെയ്തുവെന്നും അതിനുള്ള ശമ്പളം എത്രയാണെന്നുമുള്ള ഡാറ്റ ബന്ധപ്പെട്ട രജിസ്റ്ററുകളിലേക്ക് മാറുന്നു. കൺട്രോൾ യൂണിറ്റിൽ നിന്നുള്ള നിർദ്ദേശത്തിനനുസരിച്ച് ജോലി ചെയ്ത സമയം പരിഗണിച്ച് ALU ശമ്പളം കണക്കാക്കുന്നു. (അധിക സമയം ജോലി ചെയ്തത്, ബോണസ് എന്നിവ) ക്യാഷ് മെമ്മറിയിൽ നിന്നും രജിസ്റ്ററിലേക്ക് മാറ്റുന്നു. ഒരാളുടെ ശമ്പളത്തിന്റെ കണക്കുകൂട്ടലുകൾ CPU പൂർത്തിയാക്കി കഴിയുമ്പോൾ അടുത്ത ആളുടെ വിവരങ്ങൾ ദ്വിതീയ മെമ്മറിയിൽ നിന്നും റാമിലേക്ക് കൊണ്ടു വരുന്നു. അവിടെനിന്ന് ക്യാഷ് മെമ്മറിയിലേക്കും തുടർന്ന് രജിസ്റ്ററിലേക്കും എത്തിക്കുന്നു.

സംഭരണശേഷിയുടെയും അതിന്റെ വേഗതയുടെയും അടിസ്ഥാനത്തിൽ വിവിധതരം മെമ്മറികളുടെ ഒരു ശ്രേണി ചിത്രം 2.21 ൽ കൊടുത്തിരിക്കുന്നു.

വിവിധ തരം ഡാറ്റ സംഭരണികളുടെ സവിശേഷതകൾ പട്ടിക 2.2 ൽ സംഗ്രഹിച്ചിരിക്കുന്നു.

സംഭരണം	വേഗത	സംഭരണശേഷി	അനുപാതിക മൂല്യം	അസ്ഥിരമായത് (Volatile)
രജിസ്റ്റർ	അതിവേഗം	വളരെ കുറവ്	ഏറ്റവും കൂടുതൽ	അതെ
ക്യാഷ്	കൂടിയ വേഗം	കുറവ്	വളരെ കൂടുതൽ	അതെ
റാം (RAM)	വളരെവേഗം	കുറവ്/മിതം	കൂടുതൽ	അതെ
ഹാർഡ് ഡിസ്ക്	മിത വേഗം	വളരെ കൂടുതൽ	വളരെ കുറവ്	അല്ല

പട്ടിക 2.2: വിവിധതരം മെമ്മറികളുടെ സവിശേഷതകളുടെ താരതമ്യം

എങ്ങനെയാണ് രജിസ്റ്ററുകളും പ്രാഥമിക മെമ്മറിയും ദ്വിതീയ മെമ്മറിയും ഒരുമിച്ച് ഒരു കമ്പ്യൂട്ടറിൽ പ്രവർത്തിക്കുന്നതെന്ന് മനസിലാക്കുന്നതിന് താഴെയുള്ള ഉദാഹരണം ശ്രദ്ധിക്കുക.

അടുക്കളയിൽ സാലഡ് ഉണ്ടാക്കുന്നത് എങ്ങനെയെന്ന് നോക്കാം.

- സാലഡ് ഉണ്ടാക്കുന്നതിന് ആവശ്യമായ പച്ചക്കറികൾ സൂക്ഷിക്കുന്നതിനുള്ള റഫ്രിജറേറ്റർ
- പച്ചക്കറികൾ നുറുക്കുവാൻ ആവശ്യമായ പലക വെക്കുവാനുള്ള മേശ
- പച്ചക്കറികൾ വെച്ച് അരിയുവാൻ മേശമേൽ വെക്കുന്ന പലക.
- ഏതൊക്കെ പച്ചക്കറികൾ അരിയണം എന്നതിനുള്ള പാചക കുറിപ്പ്.
- ഭാഗികമായി അരിഞ്ഞ പച്ചക്കറികൾ വീണ്ടും ചെറുതാക്കുന്നതിന് വേണ്ടിയോ ഭാഗികമായി അരിഞ്ഞ് മറ്റു പച്ചക്കറികളുമായി കൂട്ടികലർത്തുന്നതിന് വേണ്ടിയോ പലകയുടെ മൂലകൾ ശൂന്യമായി സൂക്ഷിക്കണം.
- സാലഡ് നന്നായി ഇളക്കുവാൻ ആവശ്യമായ പാത്രം.
- സാലഡ് ഉണ്ടാക്കി കഴിഞ്ഞാൽ അത് സൂക്ഷിച്ചുവെക്കുവാൻ ആവശ്യമായ ഫ്രിഡ്ജിലെ സ്ഥലം.

സാലഡ് ഉണ്ടാക്കുന്നവിധം: റഫ്രിജറേറ്റിൽ നിന്ന് പച്ചക്കറികൾ മേശക്കുമേൽ എത്തിക്കുന്നു. പാചക കുറിപ്പിന് അനുസരിച്ച് ചില പച്ചക്കറികൾ എടുത്ത് അരിയുന്ന ബോക്സിൽവെക്കുക, മുറിക്കുന്ന ബോർഡിന്റെ അരികിലേക്ക് ചില പച്ചക്കറികൾ താൽക്കാലികമായി മാറ്റിവയ്ക്കുന്നു. അവ ചെറിയ കഷണങ്ങളാക്കുന്നു. പകുതി മുറിച്ച കഷണങ്ങൾ താൽക്കാലികമായി പലകയുടെ മൂലയിലേക്ക് മാറ്റിവക്കാവുന്നതാണ്. മുറിച്ചെടുത്ത പച്ചക്കറികൾ പാത്രത്തിൽ വെക്കുക. തീൻ മേശയിൽ അത് അപ്പോൾ വിതരണം ചെയ്യുന്നില്ലെങ്കിൽ റഫ്രിജറേറ്ററിലേക്ക് സൂക്ഷിക്കുക.

ഇവിടെ റഫ്രിജറേറ്റർ ഒരു ദ്വിതീയ സംഭരണി അഥവാ ഹാർഡ് ഡിസ്ക് ആയി പ്രവർത്തിക്കുന്നു. കൂടുതൽ പച്ചക്കറികൾ കൂടുതൽ കാലത്തേക്ക് സൂക്ഷിച്ചു വെക്കുന്നു. പച്ചക്കറി അരിയാനുപയോഗിച്ച മേശ കമ്പ്യൂട്ടറിന്റെ മദർബോർഡിനെപോലെ പ്രവർത്തിക്കുന്നു. എല്ലാ പ്രവർത്തനങ്ങളും അവിടെവച്ചാണ് നടക്കുന്നത് (കമ്പ്യൂട്ടറിനുള്ളിലുള്ള). കഷണം നുറുക്കാനുള്ള ബോർഡാണ് ALU - പ്രവർത്തനങ്ങൾ നടക്കുന്നത് അവിടെയാണ്. പാചക കുറിപ്പാണ് കൺട്രോൾ യൂണിറ്റ് - മുറിക്കേണ്ട പലകയിൽ എന്താണ് ചെയ്യേണ്ടത് (ALU) എന്ന് ഇതിൽ പറയുന്നു. മേശപ്പുറത്തെ ഒഴിഞ്ഞ ഭാഗമാണ് റാം പെട്ടെന്ന് എടുക്കുന്നതിനുവേണ്ടി എല്ലാ പച്ചക്കറികളും റഫ്രിജറേറ്ററിൽ നിന്ന് എടുത്ത് (Counter top) മേശപ്പുറത്ത് വെക്കുന്നു. ഇവിടെ പച്ചക്കറികൾ റഫ്രിജറേറ്ററിൽ നിന്നും (disk) എടു

ക്കുന്നതിനേക്കാൾ വേഗത്തിൽ മേശപ്പുറത്തു നിന്ന് എടുക്കാൻ സാധിക്കും, എന്നാൽ കൂടുതൽ അളവിൽ, കുറെ നേരം വെച്ചിരിക്കാൻ സാധിക്കുകയില്ല. ഭാഗികമായി മുറിച്ച പച്ചക്കറികൾ താൽക്കാലികമായി വച്ചിരിക്കുന്ന പലകയുടെ മൂലകൾ രജിസ്റ്ററുകൾക്ക് തുല്യമാണ്. ഈ മൂലകളിൽ വച്ചിരിക്കുന്ന പച്ചക്കറികൾ വളരെ വേഗത്തിൽ എടുക്കാൻ സാധിക്കും, എന്നാൽ കൂടുതൽ നേരം വെച്ചിരിക്കാൻ സാധിക്കുകയില്ല. സാലഡ് വെച്ചിരിക്കുന്ന പാത്രം ക്യാഷ് മെമ്മറി പോലെയാണ്. ഇത് പലകയുടെ മൂലയിൽ താൽക്കാലികമായി മാറ്റപ്പെടുന്ന മുറിച്ച പച്ചക്കറികൾ സൂഷിക്കുന്നതിനോ (അവിടെ കൂടുതൽ ഉണ്ടെങ്കിൽ), സാലഡ് തിരിച്ച് റഫ്രിജറേറ്ററിലേക്ക് വെക്കുന്നതിനോ (ഡാറ്റ തിരിച്ച് ഡിസ്കിലേക്ക് വെക്കുന്നതുപോലെ) അല്ലെങ്കിൽ ഡിന്നർ ടേബിളിലേക്ക് വെക്കുന്നതിനോ ആണ്. (ഔട്ട്പുട്ട് ആയി ഡാറ്റ ഔട്ട്പുട്ട് ഉപകരണത്തിലേക്ക് നൽകുന്നതുപോലെ).

സ്വയം വിലയിരുത്താം

1. കമ്പ്യൂട്ടറിലെ അതിവേഗതയുള്ള മെമ്മറിയാണ് ____________
2. ഒരു സാധാരണ ഡി.വി.ഡി യുടെ സംഭരണ ശേഷി ____________ ആകുന്നു.
3. എന്താണ് ക്യാഷ് മെമ്മറി?
4. പ്രോഗ്രാം കൗണ്ടർ രജിസ്റ്ററിന്റെ ഉപയോഗമെന്താണ്?
5. HDMI എന്നാൽ എന്താണ്?

2.1.5 ഇൻപുട്ട്/ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ (Input/Output devices)

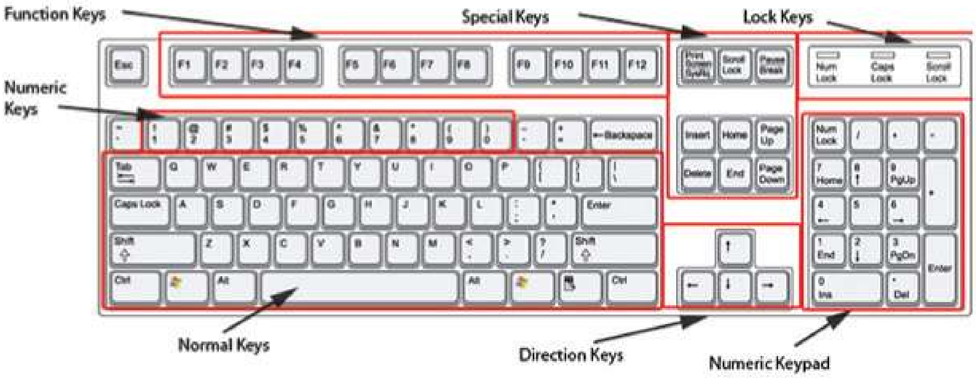
പുറം ലോകവുമായി ആശയവിനിമയം നടത്താൻ കഴിയില്ലെങ്കിൽ കമ്പ്യൂട്ടർ കൊണ്ട് ഒരു പ്രയോജനവുമില്ല. കമ്പ്യൂട്ടറുമായി ഉപയോക്താവിന് സംവദിക്കാൻ ഇൻപുട്ട്/ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ ആവശ്യമാണ്. ചുരുക്കിപ്പറഞ്ഞാൽ, ഇൻപുട്ട് ഉപകരണങ്ങൾ കമ്പ്യൂട്ടറിന് ഡാറ്റയും നിർദ്ദേശങ്ങളും നൽകുകയും, ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ കമ്പ്യൂട്ടറിൽ നിന്നും വിവരങ്ങൾ തിരികെ നൽകുകയും ചെയ്യുന്നു. ഈ ഉപകരണങ്ങളെ CPUവുമായി ബന്ധിപ്പിക്കുന്നത് വിവിധ പോർട്ടുകളിലൂടെയോ വയർലെസ് സാങ്കേതികവിദ്യയിലൂടെയോ ആയിരിക്കും. ഇവ CPUവിന് പുറത്ത് സ്ഥിതിചെയ്യുന്നതിനാൽ പെരിഫറലുകൾ എന്ന് അറിയപ്പെടുന്നു.

a. ഇൻപുട്ട് ഉപകരണങ്ങൾ

കമ്പ്യൂട്ടറിന് ഡാറ്റ നൽകാൻ ഇൻപുട്ട് ഉപകരണം ഉപയോഗിക്കുന്നു. കമ്പ്യൂട്ടറും ഉപയോക്താവും തമ്മിലുള്ള ആശയവിനിമയം സാധ്യമാക്കുന്ന ഉപകരണം എന്നും ഇതിനെ നിർവ്വചിക്കാവുന്നതാണ്. ചില ഇൻപുട്ട് ഉപകരണങ്ങളെ വിശദമായി പരിചയപ്പെടാം.

i. കീബോർഡ്

സർവസാധാരണമായ ഒരു ഇൻപുട്ട് ഉപകരണമാണ് കീബോർഡ്. അക്ഷരങ്ങളും സംഖ്യകളും വിവിധ ചിഹ്നങ്ങളും ഇൻപുട്ട് ചെയ്യുന്നതിന് ഇത് ഉപയോഗിക്കുന്നു. നാം ഒരു കീ അമർത്തുമ്പോൾ കീബോർഡ് അതിലെ അക്ഷരത്തിന് തത്തുല്യമായ ASCII കോഡ് ഉണ്ടാക്കി കമ്പ്യൂട്ടറിന് നൽകും. 1986ൽ യുഎസ് പുറത്തിറക്കിയ കീബോർഡിന് 101 കീകൾ ഉണ്ടായിരുന്നു. QWERTY രൂപരേഖ എന്നറിയപ്പെടുന്ന ഒരു ഘടനയാണ് അതിനുള്ളത്. കീബോർഡിന്റെ ഇടതുവശത്ത് മുകളിലായി കാണുന്ന ആദ്യത്തെ ആറ് അക്ഷരങ്ങളെ ഒരുമിച്ചെടുത്താണ് ഈ പേര് രൂപപ്പെടുത്തിയത് (ചിത്രം 2.22).



ചിത്രം 2.22: കീബോർഡ് ഘടന

കീബോർഡുകളെ വയേർഡ് എന്നും വയർലെസ് എന്നും രണ്ടായി തിരിക്കാം. വയേർഡ് കീബോർഡുകളെ സീരിയൽ, PS/2 പോർട്ട് ഉപയോഗിച്ചോ USB പോർട്ട് ഉപയോഗിച്ചോ ആണ് CPUവുമായി ബന്ധിപ്പിക്കുന്നത്. എന്നാൽ വയർലെസ് കീബോർഡുകളെ കമ്പ്യൂട്ടറുമായി ബന്ധിപ്പിക്കുന്നതിന് ഇൻഫ്രാറെഡ് തരംഗങ്ങളോ റേഡിയോ ഫ്രീക്വൻസി തരംഗങ്ങളോ ബ്ലൂടൂത്ത് സങ്കേതമോ ആയിരിക്കും ഉപയോഗിക്കുന്നത്. കൈപ്പിടിയിലൊതുങ്ങുന്നതും കൂടെ കൊണ്ടു നടക്കാവുന്നതുമായ കീബോർഡുകളും ഇന്ന് ലഭ്യമാണ്. ഏതെങ്കിലുമൊരു പ്രതലത്തിൽ കീബോർഡിന്റെ ഘടന പതിപ്പിക്കാവുന്ന ലേസർ കീബോർഡുകൾ പോലെയുള്ള നവയുഗ കീബോർഡുകൾ വികസനത്തിന്റെ പാതയിലാണ്.

ii. മൗസ്

കമ്പ്യൂട്ടർ സ്ക്രീനിലെ കർസറിന്റെ സ്ഥാനത്തെ സൂചിപ്പിക്കുന്ന കൈപ്പിടിയിലൊതുങ്ങുന്ന ഒരു ചെറിയ ഇൻപുട്ട് ഉപകരണമാണ് മൗസ്. ഒരു പരന്ന പ്രതലത്തിലോ മൗസ് പാഡിലോ നീക്കിക്കൊണ്ട് കർസറിന്റെ സ്ഥാനം മാറ്റാൻ ഇതിന് കഴിയും. മൗസിന് സാധാരാണയായി ഒന്നോ അതിലധികമോ ബട്ടണുകളും ഒരു ഉരുളൻ ചക്രവും (സ്ക്രോൾ വീലും) ഉണ്ട് സ്ക്രീൻ ദൃശ്യത്തെ ലംബമായോ തിരശ്ചീനമായോ നീക്കുന്നതിനാണ് സ്ക്രോൾ വീൽ ഉപയോഗിക്കുന്നത്. ബോൾ മൗസ്,

ചിത്രം 2.23: മൗസ്

ഒപ്റ്റിക്കൽ മൗസ്, ലേസർ മൗസ് എന്നിങ്ങനെ പലതരം മൗസുകളുണ്ട്. ബോൾ മൗസ് പ്രവർത്തിക്കുന്നത് അതിന്റെ ചുവട്ടിലുള്ള ഒരു ഗോളത്തിന്റെ ചലനതത്വത്തിന്റെ അടിസ്ഥാനത്തിലാണ്. എന്നാൽ ഒപ്റ്റിക്കൽ മൗസ് LED ഉപയോഗിച്ചും ലേസർ മൗസ് ലേസർ രശ്മികൾ ഉപയോഗിച്ചുമാണ് ചലനഗതി മനസിലാക്കുന്നത്. മറ്റ് മൗസുകളെ അപേക്ഷിച്ച് ലേസർ മൗസിന് സൂക്ഷ്മതല ചലനങ്ങൾ സാധ്യമാകും. വയേർഡ് മൗസ് കമ്പ്യൂട്ടറുമായുള്ള ആശയവിനിമയത്തിന് സീരിയൽ, PS/2, USB പോർട്ടുകൾ ഉപയോഗിക്കുമ്പോൾ വയർലെസ് മൗസുകൾ റേഡിയോ തരംഗങ്ങൾ പ്രയോജനപ്പെടുത്തുന്നു.

iii. ലൈറ്റ് പെൻ

പേനയുടെ രൂപത്തിലുള്ള ഒരു ചൂണ്ടുപകരണമാണ് ലൈറ്റ് പെൻ. ഇതിന്റെ അഗ്രത്തിൽ പ്രകാശ പ്രതിപ്രവർത്തനം നടത്തുന്ന ഒരു ഘടകമുണ്ട്. ഈ ഭാഗം സ്ക്രീനിൽ വയ്ക്കുമ്പോൾ അവിടെയുള്ള പ്രകാശം തിരിച്ചറിയുകയും അങ്ങനെ സ്ക്രീനിൽ പേനയുടെ സ്ഥാനം എവിടെയാണെന്ന് കമ്പ്യൂട്ടറിന് മനസിലാക്കാൻ സാധിക്കുകയും ചെയ്യുന്നു. സ്ക്രീനിൽ നേരിട്ട് വരയ്ക്കാൻ കഴിയുമെന്നത് ലൈറ്റ് പേനയുടെ ഒരു മേന്മയാണ്. കമ്പ്യൂട്ടറധിഷ്ഠിത രൂപകൽപനയ്ക്കും (കമ്പ്യൂട്ടർ എയിഡഡ് ഡിസൈനിംഗിനും - CAD) ചിത്രം വരയ്ക്കുന്നതിനും എഞ്ചിനീയർമാരും കലാകാരന്മാരും വസ്ത്രാലങ്കാരം ചെയ്യുന്നവരും ഇത് ഉപയോഗിക്കുന്നു.

ചിത്രം 2.24 : ലൈറ്റ്പെൻ

iv. ടച്ച് സ്ക്രീൻ

സ്ക്രീനിൽ സ്പർശിച്ചുകൊണ്ട് കമ്പ്യൂട്ടറിൽ വിവിധ പ്രവർത്തനങ്ങൾ നടത്താൻ ഉപയോക്താവിന് സൗകര്യം ഒരുക്കുന്ന ഇൻപുട്ട് ഉപകരണമാണ് ടച്ച് സ്ക്രീൻ. ചില കമ്പ്യൂട്ടറുകൾ, ടാബ്ലെറ്റുകൾ, സ്മാർട്ട് ഫോണുകൾ മുതലായവയിൽ സ്പർശിച്ചുകൊണ്ട് പ്രവർത്തനക്ഷമമാകുന്ന സ്ക്രീനുകളുണ്ട്. സ്റ്റൈലസ് ഉപയോഗിച്ചാൽ കുറേക്കൂടി സൂക്ഷ്മമായ സ്പർശനം സാധ്യമാകും. റെയിൽവേ സ്റ്റേഷനുകളിലെ ഇൻഫർമേഷൻ ബൂത്തുകളിലും (Information kiosk) എറ്റിഎം കൗണ്ടറുകളിലും ടച്ച് സ്ക്രീനുകളാണ് ഇൻപുട്ട് ഉപകരണമായി ഉപയോഗിക്കുന്നത്.

ചിത്രം 2.25 : ടച്ച് സ്ക്രീൻ

v. ഗ്രാഫിക് ടാബ്ലെറ്റ്

ഒരു ഇലക്ട്രോണിക് എഴുത്ത് പ്രതലവും അതിലെഴുതാൻ ഉപയോഗിക്കുന്ന ഒരു പ്രത്യേക തരം പേനയും അടങ്ങുന്നതാണ് ഗ്രാഫിക് ടാബ്ലെറ്റ്. സാമ്പ്രദായിക ഉപകരണങ്ങൾ ഉപയോഗിച്ച് വരയ്ക്കുന്നതുപോലെ ചലനങ്ങളും പ്രവർത്തനങ്ങളുമുള്ള ഗ്രാഫിക് ചിത്രങ്ങൾ നിർമ്മിക്കുന്നതിന് കലാകാരന്മാർ ഇത് ഉപയോഗിച്ചു വരുന്നു. ഇതിലെ പേന മർദ്ദ-പ്രതിപ്രവർത്തന ശേഷിയുള്ളതാണ്. ടാബ്ലെറ്റിന്മേലുള്ള ദൃഢമോ

ചിത്രം 2.26 : ഗ്രാഫിക് ടാബ്ലെറ്റ്

മൃദുവോ ആയ മർദ്ദത്തിന് ഗ്രാഫിക്സ് പ്രോഗ്രാമിൽ വ്യത്യസ്ത വീതിയിലുള്ള ബ്രഷ് വരകൾ സംജാതമാക്കാൻ കഴിയും.

vi. ടച്ച് പാഡ്

കൊണ്ടുനടക്കാൻ എളുപ്പമുള്ള (portable) കമ്പ്യൂട്ടറുകളിലും ബാഹ്യ (external) കീബോർഡുകളിലും ചൂണ്ടുപകരണമായി ടച്ച് പാഡുപയോഗിക്കുന്നു. ഒരു മൗസിന്റെ സാന്നിധ്യമില്ലാതെ തന്നെ മൗസ് പോയിന്റർ നീക്കുവാൻ ഇത് സഹായിക്കുന്നു. ടച്ച് പാഡിലെ പരന്ന പ്രതലത്തിലൂടെ വിരൽ നീക്കിയാണ് ഇത് പ്രവർത്തിപ്പിക്കുന്നത്. പ്രതലത്തിലെ വിരലിന്റെ നീക്കത്തിനനുസരിച്ച് സ്ക്രീനിൽ കർസറിന് സ്ഥാനമാറ്റം സംഭവിക്കും. സ്പർശനപ്രതലത്തിന് താഴെയായി ക്ലിക്കിംഗ് സാധ്യമാക്കുന്ന രണ്ട് ബട്ടണുകളും ടച്ച് പാഡിലുണ്ട്.

ചിത്രം 2.27 : ടച്ച്പാഡ്

vii. ജോയ്സ്റ്റിക്ക്

വീഡിയോ ഗെയിമുകൾ കളിക്കുന്നതിനും പരിശീലന അനുകരണങ്ങളും (training simulators) റോബോട്ടുകളെ നിയന്ത്രിക്കുന്നതിനും സാധാരണയായി ഉപയോഗിച്ചു വരുന്ന ഇൻപുട്ട് ഉപകരണമാണ് ജോയ് സ്റ്റിക്ക്. ചൂണ്ടുപകരണങ്ങളായും ജോയ്സ്റ്റിക്കുകളും മറ്റ് ഗെയിം നിയന്ത്രണോപകരണങ്ങളും ഉപയോഗിക്കാറുണ്ട്. ജോയ്സ്റ്റിക്കിന് ഏത് ദിശയിലേക്കും ചലിപ്പിക്കാൻ കഴിയുന്ന ഒരു ലംബദണ്ഡുണ്ട്. ഇതുപയോഗിച്ച് വീഡിയോ ഗെയിമിലെ വസ്തുക്കളെ നിയന്ത്രിക്കാനും സ്ക്രീനിലെ കർസർ ചലിപ്പിച്ച് മെനുവിൽ ലഭ്യമാകുന്ന ഓപ്ഷനുകളിൽ നമുക്ക് വേണ്ടത് അടയാളപ്പെടുത്താനും സാധിക്കും. ഇതിന്റെ മുകളിലുള്ള ബട്ടൺ ഉപയോഗിച്ച് അടയാളപ്പെടുത്തി വച്ചിരിക്കുന്ന ഓപ്ഷൻ തിരഞ്ഞെടുക്കാം.

ചിത്രം 2.28: ജോയ്സ്റ്റിക്ക്

viii. മൈക്രോഫോൺ

ചിത്രം 2.29 : മൈക്രോഫോൺ

കമ്പ്യൂട്ടറുമായി ഒരു മൈക്രോഫോൺ ബന്ധിപ്പിച്ചാൽ അതിലേയ്ക്ക് ശബ്ദം ഇൻപുട്ട് ചെയ്യാം. സാധാരണയായി ലഭിക്കുന്ന അനലോഗ് രൂപത്തിലുള്ള ശബ്ദം ഇത് സ്വീകരിച്ച് അതിനെ ഡിജിറ്റൽ രൂപത്തിലേയ്ക്ക് മാറ്റുന്നു. ഇങ്ങനെ മാറ്റിയ ശബ്ദത്തെ എന്തെങ്കിലും സംസ്കരണം നടത്തുന്നതിനോ തിരികെ കേൾപ്പിക്കുന്നതിനോ വേണ്ടി സംഭരിച്ചുവയ്ക്കാവുന്നതാണ്. കമ്പ്യൂട്ടറിലുള്ള സ്പീച്ച് റെക്കഗ്നിഷൻ സോഫ്റ്റ്‌വയർ ഉപയോഗിച്ച് ഇൻപുട്ട് ചെയ്യുന്ന ശബ്ദത്തിലെ ടെക്സ്റ്റ് സംഭരിച്ച് വേർഡ് പ്രോസസിംഗിനു വേണ്ടിയും ഉപയോഗിക്കാം. ഒരു വോയിസ് റെക്കഗ്നിഷൻ പ്രോഗ്രാമിന് ഇൻപുട്ട് ശബ്ദത്തെ കമ്പ്യൂട്ടറിന് മനസിലാകുന്ന നിർദ്ദേശങ്ങളാക്കി മാറ്റാനും സാധിക്കും.

ix. സ്കാനർ

ചിത്രങ്ങളോ ടെക്സ്റ്റോ പോലെയുള്ള വിവരങ്ങളെ കടലാസിൽ നിന്നും സ്വീകരിച്ച് ഡിജിറ്റൽ രൂപത്തിലേയ്ക്ക് മാറ്റി എഡിറ്റിംഗിനും മറ്റുമായി കമ്പ്യൂട്ടറിന് നൽകുന്ന ഉപകരണമാണ് സ്കാനർ. ചിത്രത്തിന്റെ ഗുണനിലവാരം സ്കാനറിന്റെ വിശ്ലേഷണത്തെ അഥവാ മിഴിവിനെ (resolution) ആശ്രയിച്ചിരിക്കുന്നു. സ്കാൻ ചെയ്യപ്പെടുന്ന ചിത്രത്തിന്റെ റെസല്യൂഷൻ രേഖപ്പെടുത്തുന്നത് ഡോട്ട്സ് പർ ഇഞ്ച് (Dots Per Inch - DPI) എന്ന അളവ്കോൽ ഉപയോഗിച്ചാണ്. DPI കൂടുന്തോറും റെസല്യുഷനും മെച്ചപ്പെട്ടതായിരിക്കും. ഫ്ളാറ്റ് ബെഡ്, ഷീറ്റ് ഫീഡ്, ഹാൻഡ്ഹെൽഡ് എന്നീ വകഭേദങ്ങളിൽ സ്കാനറുകൾ ലഭ്യമാണ്. ഷീറ്റ് ഫീഡ് സ്കാനറിന് ഒരു ഷീറ്റ് മാത്രമെ സ്കാൻ ചെയ്യാൻ കഴിയു; എന്നാൽ ഫ്ളാറ്റ് ബെഡിന് ഒരു പുസ്തകത്തിൽ നിന്നുള്ള പേജുകളെ സ്കാൻ ചെയ്യാൻ കഴിയും. പക്ഷെ ഫ്ളാറ്റ് ബെഡ് സ്കാനറുകൾ കൊണ്ടുനടക്കാൻ എളുപ്പമല്ല. ഹാൻഡ്ഹെൽഡ് സ്കാനറുകൾ എളുപ്പത്തിൽ കൊണ്ടുനടക്കാനാവുമെങ്കിലും സ്കാനിംഗ് നടത്താൻ ഇതിനെ ഉപയോക്താവ് തന്നെ പേജിലൂടെ നീക്കണം.

ചിത്രം . 2.30: സ്കാനർ

സ്കാൻ ചെയ്യപ്പെടുന്ന ടെക്സ്റ്റിനെ ഓപ്ടിക്കൽ ക്യാരക്ടർ റെക്കഗ്നിഷൻ (Optical Character Recognition - OCR) സോഫ്റ്റ്വെയർ ഉപയോഗിച്ച് തിരിച്ചറിയുകയും, ടെക്സ്റ്റ് എഡിറ്റർ ഉപയോഗിച്ച് ആവശ്യമായ ചിട്ടപ്പെടുത്തലുകൾ നടത്താൻ, അതിനെ ഉചിതമായ ടെക്സ്റ്റ് ഫോർമാറ്റിലേയ്ക്ക് മാറ്റുകയും ചെയ്യുന്നു. മികച്ച OCR സമ്പ്രദായങ്ങൾക്ക് വൈവിധ്യമാർന്ന അക്ഷര വടിവുകളിലുള്ളതും പ്രിന്റ് രൂപത്തിലുള്ളതും ആയ ടെക്സ്റ്റ് തിരിച്ചറിയാൻ കഴിയുമെങ്കിലും കൈയ്യെഴുത്തു പ്രതികൾ വായിക്കാൻ പലപ്പോഴും സാധിക്കാറുമില്ല. ആക്കുറേറ്റ് OCR SDK, ഹിന്ദി OCR സോഫ്റ്റ്വെയർ, അക്ഷര മലയാളം OCR മുതലായവ OCR സോഫ്റ്റ്വെയറുകൾക്ക് ഉദാഹരണങ്ങളാണ്.

x. ഒപ്റ്റിക്കൽ മാർക്ക് റീഡർ (OMR)

പ്രിന്റ് ചെയ്ത ഒരു ഫോറം സ്കാൻ ചെയ്ത് മുൻ നിശ്ചയ പ്രകാരമുള്ള സ്ഥാനങ്ങളിലെ അടയാളങ്ങൾ വായിച്ച് അവ കമ്പ്യൂട്ടറിൽ രേഖപ്പെടുത്തുന്നതാണ് OMR സാങ്കേതികവിദ്യ. മൾട്ടിപ്പിൾ ചോയിസ് ചോദ്യങ്ങളടങ്ങുന്ന ഒബ്ജക്റ്റീവ് ടൈപ്പ് പരീക്ഷകളിലും ചോദ്യാവലികളിലും രേഖപ്പെടുത്തുന്ന പ്രതികരണങ്ങൾ ഉയർന്ന കൃത്യതയോടെയും ഏറ്റവും വേഗത്തിലും പരിശോധിക്കുന്നതിന് ഇത് ഫലപ്രദമായി വിനിയോഗിക്കുന്നു. മൽസര പരീക്ഷകളുടെ മൂല്യനിർണ്ണയം നടത്താൻ സാധാരണ ഉപയോഗിച്ച് വരുന്ന OMR ഷീറ്റാണ് ചിത്രം 2.32 ൽ കൊടുത്തിരിക്കുന്നത്. ഉത്തരങ്ങൾ രേഖപ്പെടുത്തുന്നതിനായി കുമിളയുടെ ആകൃതിയിലുള്ള (bubble shaped) ഓപ്ഷനുകളായിരിക്കും അതിൽ നൽകിയിരിക്കുന്നത്. പരീക്ഷാർഥികൾ ശരിയായ

ചിത്രം. 2.31 : ഒപ്റ്റിക്കൽ മാർക്ക് റീഡർ

ചിത്രം 2.32: OMR ഷീറ്റ്

ഉത്തരത്തെ സൂചിപ്പിക്കുന്ന കുമിള (bubble) പേനയോ പെൻസിലോ ഉപയോഗിച്ച് കറുപ്പിക്കണം. റീഡർ (ചിത്രം 2.31) ഈ അടയാളം തിരിച്ചറിയുകയും അനുയോജ്യമായ സോഫ്റ്റ്‌വെയറുപയോഗിച്ച് മൂല്യനിർണ്ണയത്തിനായി കമ്പ്യൂട്ടറിലേയ്ക്ക് ഇൻപുട്ട് ചെയ്യുകയും ചെയ്യുന്നു. കൃത്യതയാർന്ന ഫലം ലഭിക്കാൻ ഗുണമേന്മയുള്ള പേപ്പർ ഉപയോഗിക്കുകയും അതിൽ കുമിളകൾ കൃത്യമായി വിന്യസിക്കുകയും വേണം.

xi. ബാർ കോഡ്/ക്വിക്ക് റെസ്പോൺസ് (QR)കോഡ് റീഡർ

ഒരു സംഖ്യയെ സൂചിപ്പിക്കാനായി വ്യത്യസ്ത കനത്തിലും അകലത്തിലും ലംബമായി അടുക്കിവച്ചിരിക്കുന്ന ഒരു കൂട്ടം വരകളാണ് ബാർകോഡ് എന്നറിയപ്പെടുന്നത്. ഇത്തരം ബാർ കോഡുകൾ പ്രതിനിധാനം ചെയ്യുന്ന ഡാറ്റയെ ഇൻപുട്ട് ചെയ്യാനുപയോഗിക്കുന്ന ഉപകരണമാണ് ബാർകോഡ് റീഡർ (ചിത്രം 2.33). കടകളിൽ എളുപ്പത്തിൽ ബിൽ തയ്യാറാക്കുന്നതിന് സാധനങ്ങളുടെ വിലവിവരം ലഭിക്കാൻ അവയിലെ കോഡുകൾ സ്കാൻ ചെയ്യുന്നത് നിങ്ങൾ കണ്ടിട്ടുണ്ടാവും. സാധാരണയായി ഇതിനുവേണ്ടി കൈയിൽ പിടിക്കാവുന്ന സ്കാനറുകൾ അഥവാ റീഡറുകൾ ഉപയോഗിക്കുന്നു. ക്യാമറയും പ്രത്യേകതരം സോഫ്റ്റ്‌വെയറുമുണ്ടെങ്കിൽ മൊബൈൽ ഫോണും ബാർകോഡ് റീഡറായി ഉപയോഗിക്കാം.

ചിത്രം 2.33 : ബാർകോഡ് റീഡർ

ചിത്രം 2.34: QR കോഡ്

ബാർകോഡുകൾക്ക് ഏതാണ്ട് സമാനമാണ് QR കോഡുകൾ. ബാർകോഡ് ഏകമാനമാണെങ്കിൽ QR കോഡ് ചിത്രം 2.34-ൽ കാണുന്നതുപോലെ ദ്വിമാനമാണ്. അതുകൊണ്ട് തന്നെ QR കോഡിന് ബാർകോഡിനെ അപേക്ഷിച്ച് കൂടുതൽ ഡാറ്റ സംഭരിക്കാനാകും. വെബ്സൈറ്റ് URLകൾ, സാധാരണ ടെക്സ്റ്റ്, ഫോൺ നമ്പർ, ഇ-മെയിൽ വിലാസം എന്നിവയൊക്കെ QR കോഡിൽ സംഭരിക്കാം. ഒരു ബാർകോഡ് റീഡർ അല്ലെങ്കിൽ ക്യാമറയും ആവശ്യമായ സോഫ്റ്റ്‌വെയറുമുള്ള മൊബൈൽ ഫോൺ ഉപയോഗിച്ച് QR കോഡിലെ വിവരം വായിച്ച് കമ്പ്യൂട്ടറിന് നൽകാവുന്നതാണ്.

xii. ബയോമെട്രിക് സെൻസർ

മനുഷ്യന്റെ അദ്വിതീയമായ ശാരീരിക സവിശേഷതകൾ വളരെ കൃത്യമായി തിരിച്ചറിയാൻ കഴിയുന്ന ഒരു ഉപകരണമാണ് ബയോമെട്രിക് സെൻസർ. ഇത് ഒരു വ്യക്തിയുടെ അനന്യതയെ തിരിച്ചറിയുന്നതിനും പരിശോധിക്കുന്നതിനും തെളിയിക്കുന്നതിനുമായി അയാളുടെ ശാരീരിക സവിശേഷതകളായ വിരലടയാളം, നേത്രപടലം, കൃഷ്ണമണി തുടങ്ങിയവയെ ഉപയോഗിക്കുന്ന ബയോമെട്രിക് വ്യവസ്ഥയുടെ ഒരു ഘടകമാണ്. അർദ്ധചാലക സെൻസർ, ഒപ്റ്റിക്കൽ സെൻസർ, അൾട്രാസൗണ്ട് സെൻസർ എന്നിങ്ങനെ പ്രധാനമായും മൂന്ന് തരം ബയോമെട്രിക് സെൻസറുകളുണ്ട്. ഒരു വിരലടയാള (Finger print) സെൻസറാണ് ചിത്രം 2.35-ൽ കാണുന്നത്.

ചിത്രം 2.35: ബയോമെട്രിക് സെൻസർ

xiii. സ്മാർട്ട് കാർഡ് റീഡർ

ചിത്രം. 2.36 : സ്മാർട്ട് കാർഡ് റീഡർ

ഡാറ്റ സംഭരിക്കാനും വിനിമയം ചെയ്യാനും കഴിയുന്ന ഒരു പ്ലാസ്റ്റിക് കാർഡാണ് സ്മാർട്ട് കാർഡ്. ഈ ഡാറ്റാ കാർഡിൽ ഒരു മെമ്മറിയും മൈക്രോപ്രോസസറും അടങ്ങിയിരിക്കുന്നു. വെറുമൊരു മെമ്മറി കാർഡാണെങ്കിൽ അത് ഡാറ്റ സംഭരിക്കാൻ മാത്രമെ ഉപയുക്തമാവുകയുള്ളു; എന്നാൽ അതിൽ മൈക്രോപ്രോസസറും കൂടിയുണ്ടെങ്കിൽ ആ മെമ്മറിയിൽ ഡാറ്റ കൂട്ടിച്ചേർക്കുവാനും ആവശ്യമില്ലാത്തവയെ നീക്കം ചെയ്യുവാനും, ഉള്ളവയിൽ എന്തെങ്കിലും ക്രിയകൾ ചെയ്യുവാനും കഴിയും. പ്രധാനമായും ബാങ്ക് ഇടപാടുകൾക്കും, ആരോഗ്യപരിപാലന മേഖലയിലും, ടെലിഫോൺ ഉപയോഗിക്കുന്നതിനും, പണമടയ്ക്കുന്നതിനുമൊക്കെയാണ് സ്മാർട്ട് കാർഡുകൾ ഉപയോഗിക്കുന്നത്.

സ്മാർട്ട് കാർഡിലുള്ള ഡാറ്റ എടുക്കുന്നതിന് സ്മാർട്ട് കാർഡ് റീഡർ ഉപയോഗിക്കുന്നു. സമ്പർക്ക രീതിയിൽ പ്രവർത്തിക്കുന്നതും സമ്പർക്കമില്ലാതെ പ്രവർത്തിക്കുന്നതുമായ റീഡറുകൾ പ്രചാരത്തിലുണ്ട്. ആദ്യത്തേതിൽ കാർഡിനെ റീഡറിനുള്ളിലേയ്ക്ക് കയറ്റി ഒരു ഭൗതിക സമ്പർക്കം സ്ഥാപിച്ചാണ് ഡാറ്റ വിനിമയം ചെയ്യുന്നത്. എന്നാൽ രണ്ടാമത്തെ വിഭാഗത്തിൽ റീഡറിന് സമീപം കാർഡ് കൊണ്ടുവരുമ്പോൾ അവ തമ്മിൽ ഒരു റേഡിയോ ഫ്രീക്വൻസിയിലൂടെ ഡാറ്റ വിനിമയം നടത്തുന്നു. ഗതാഗത രംഗത്തെ ടോൾ ഗേറ്റിലെ ഫീസ് ഒടുക്കുന്നതിനും, വ്യക്തികളുടെ തിരിച്ചറിയൽ സംവിധാനത്തിനുമൊക്കെയാണ് സമ്പർക്കമില്ലാത്ത റീഡറുകൾ ഉപയോഗിക്കുന്നത്.

xiv. ഡിജിറ്റൽ ക്യാമറ

ചിത്രം 2.37 : ഡിജിറ്റൽ ക്യാമറ

ചിത്രങ്ങളും വീഡിയോകളും പകർത്താനും അവയെ ഡിജിറ്റൽ രൂപത്തിലേയ്ക്ക് മാറ്റാനും ഡിജിറ്റൽ ക്യാമറ ഉപയോഗിക്കുന്നു. ഇതുപയോഗിച്ച് പകർത്തുന്ന ദൃശ്യങ്ങൾ ഇതിനുള്ളിലെ മെമ്മറിയിൽ സംഭരിക്കുകയും പിന്നീട് ഇതിനെ കമ്പ്യൂട്ടറുമായി ബന്ധിപ്പിച്ച് അതിലേയ്ക്ക് മാറ്റുകയും ചെയ്യുന്നു. ക്യാമറയിൽ ഉപയോഗിക്കുന്ന ലെൻസിന്റെ ഗുണമേന്മ, ചാർജ് കപ്പിൾ ഡിവൈസിന്റെ (Charge Couple Device - CCD) സാന്ദ്രത, മെഗാ പിക്സലിൽ നിർണ്ണയിക്കുന്ന റെസല്യൂഷൻ, ഒപ്റ്റിക്കൽ സൂം (Optical Zoom), ഇൻസ്റ്റാൾ ചെയ്തിരിക്കുന്ന സോഫ്റ്റ്വെയർ എന്നീ ഘടകങ്ങൾ ചിത്രത്തിന്റെ ഗുണനിലവാരം നിശ്ചയിക്കും. ഓരോ ചിത്രവും ആയിരക്കണക്കിന് ചെറു ചിത്രകണികകൾ അഥവാ പിക്സലുകൾ (picture element) കൊണ്ടാണ് നിർമ്മിക്കുന്നത്. ഓരോ പിക്സലിലേയും നിറത്തെ സംബന്ധിച്ച വിവരങ്ങൾ ക്യാമറയിൽ ശേഖരിച്ചു വയ്ക്കും. ഓരോ ചിത്രത്തിലെയും പിക്സലുകളുടെ എണ്ണമാണ് ആ ചിത്രത്തിന്റെ ഗുണമേന്മ നിശ്ചയിക്കുന്നത്. 2 മുതൽ 24 വരെ മെഗാ പിക്സൽ റെസല്യൂഷനും, 3x മുതൽ 60x വരെ ഒപ്റ്റിക്കൽ സൂമുമുള്ള ശ്രേണിയിൽ ഡിജിറ്റൽ ക്യാമറകൾ ലഭ്യമാണ്.

ചിത്രം 2.38 : വെബ് ക്യാമറ

ഡിജിറ്റൽ ക്യാമറയേക്കാൾ വിലക്കുറവും ഒതുങ്ങിയതുമായ ഒരു വകഭേദമാണ് വെബ് ക്യാമറ അഥവാ വെബ്ക്യാം. വീഡിയോ സംവാദം, വീഡിയോ സല്ലാപം തുടങ്ങിയ ആവശ്യങ്ങൾക്കായി ഇത് ഉപയോഗിച്ച് വരുന്നു. ഇതിനുള്ളിൽ മെമ്മറി ഉണ്ടായിരിക്കുകയില്ല. സ്കൈപ്, യാഹൂ മെസഞ്ചർ തുടങ്ങിയ ആപ്ലിക്കേഷനുകളിൽ ചിത്രം പകർത്തുന്നതിന് വെബ്ക്യാം ഉപയോഗിക്കുന്നു. ഇപ്പോൾ ലഭ്യമാകുന്ന മിക്കവാറും ലാപ്പ് ടോപ്പുകളിലും വെബ് ക്യാമറ ഉൾച്ചേർത്തിരിക്കും.

b. ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ

ഒരു കമ്പ്യൂട്ടറിൽ നിന്ന് പ്രിന്റ്/ഡിസ്പ്ലേ ചെയ്യുന്ന ഉപകരണങ്ങളാണ് ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ. ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ ഉൽപാദിപ്പിക്കുന്ന ഉൽപന്നങ്ങളെ ഹാർഡ്കോപ്പി ഔട്ട്പുട്ട് അല്ലെങ്കിൽ സോഫ്റ്റ്കോപ്പി ഔട്ട്പുട്ട് എന്നു വിളിക്കുന്നു. ഹാർഡ്കോപ്പി ഔട്ട്പുട്ടുകൾ ശാശ്വതമായ ഉൽപ്പന്നങ്ങളാണ്, അത് പിന്നീട് ആവശ്യമുള്ളപ്പോൾ ഉപയോഗിക്കാം. അവ പേപ്പറിൽ സ്ഥിരമായ ഒരു രേഖ ഉണ്ടാക്കുന്നു. ഹാർഡ്കോപ്പി ഔട്ട്പുട്ടുകൾ നിർമ്മിക്കുന്ന സാധാരണ ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ പ്രിന്ററുകളും പ്ലോട്ടറുകളുമാണ്. സോഫ്റ്റ്കോപ്പി ഔട്ട്പുട്ട് ഇലക്ട്രോണിക്സ് ഉപയോഗിച്ച് ഡിജിറ്റൽ രൂപത്തിൽ സ്ക്രീനിൽ ലഭ്യമാക്കുന്നു. അവ ഒരു സ്ഥിരമായ രേഖ ഉണ്ടാക്കുന്നില്ല. ഒരു സാധാരണ സോഫ്റ്റ്കോപ്പി ഔട്ട്പുട്ട് ഉപകരണമാണ് വിഷ്വൽ ഡിസ്പ്ലേ യൂണിറ്റ് (VDU).

i. വിഷ്വൽ ഡിസ്പ്ലേ യൂണിറ്റ് (VDU)

ഒരു വിഷ്വൽ ഡിസ്പ്ലേ യൂണിറ്റ് (VDU) ടെക്സ്റ്റ്, ഗ്രാഫിക്സ്, വീഡിയോ വിവരങ്ങൾ എന്നിവ ദൃശ്യമായി നൽകുന്ന ഒരു ഔട്ട്പുട്ട് ഉപകരണമാണ്. ഒരു ഡിസ്പ്ലേ ഉപകരണത്തിൽ കാണുന്ന വിവരങ്ങളെ സോഫ്റ്റ്കോപ്പി എന്നാണ് വിളിക്കുന്നത്. വിവരങ്ങൾ ഇലക്ട്രോണിക് ആയിരിക്കുകയും താൽക്കാലികമായി മാത്രം പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നത് കൊണ്ടാണ് അവയെ സോഫ്റ്റ്കോപ്പി എന്നറിയപ്പെടുന്നത്. കാതോഡ് റേ ട്യൂബ് (CRT) മോണിറ്ററുകൾ, ലിക്വിഡ് ക്രിസ്റ്റൽ ഡിസ്പ്ലേ (LCD) മോണിറ്ററുകൾ, തിൻ ഫിലിം ട്രാൻസിസ്റ്റർ (TFT) മോണിറ്ററുകൾ, ലൈറ്റ് എമിറ്റിങ് ഡയോഡ് (LED) മോണിറ്ററുകൾ, വാതക പ്ലാസ്മാ മോണിറ്ററുകൾ എന്നിവ വിഷ്വൽ ഡിസ്പ്ലേ യൂണിറ്റ് ഉപകരണങ്ങളിൽ ഉൾപ്പെടുന്നു.

വിഷ്വൽ ഡിസ്പ്ലേ യൂണിറ്റിന്റെ (VDU) ചില സവിശേഷതകൾ വലുപ്പം, മിഴിവ്, പിക്സൽ പിച്ച്, പ്രതികരണ സമയം എന്നിവയാണ്. VDU കൾ വ്യത്യസ്ത വലിപ്പത്തിൽ ലഭ്യമാണ്. സ്ക്രീനിനു കോണോടുകോണായ ഇഞ്ചുകളിലെ അളവ് ഒരു മോണിറ്ററിന്റെ വലിപ്പമായി കണക്കാക്കുന്നു. തിരശ്ചീനമായും ലംബമായും പ്രദർശിപ്പിക്കാൻ കഴിയുന്ന പരമാവധി പിക്സലുകളെ (800 x 600 അല്ലെങ്കിൽ 1024 x 768 അല്ലെങ്കിൽ 1600 x 1200 പോലുള്ളവ) മോണിറ്ററിന്റെ റെസല്യൂഷൻ എന്ന് വിളിക്കുന്നു. സ്ക്രീനിലെ പിക്സലുകൾ തമ്മിലുള്ള അന്തരം ഡോട്ട് പിച്ച് എന്നാണ് അറിയപ്പെടുന്നത്. ചെറിയ ഡോട്ട് പിച്ച് ഉള്ള ഒരു സ്ക്രീൻ കൂടുതൽ വ്യക്തമായ ചിത്രങ്ങൾ നൽകുന്നു. ഒരു പിക്സൽ തിളക്കമുള്ള അവസ്ഥയിൽ നിന്ന് തിളക്കം ഇല്ലാത്ത അവസ്ഥയിലേക്കും തിരിച്ച് തിളക്കമുള്ള അവസ്ഥയിലേക്കും മാറാനെടുക്കുന്ന സമയത്തെ പ്രതികരണ സമയം എന്ന് വിളിക്കുന്നു. പ്രതികരണ സമയം കുറവുള്ള മോണിറ്ററുകൾ മെച്ചപ്പെട്ട ദൃശ്യാനുഭവം നൽകുന്നു.

കാതോഡ് റേ ട്യൂബ് (CRT) മോണിറ്ററുകൾ

കാതോഡ് റേ ട്യൂബ് (CRT) മോണിറ്റർ കഴിഞ്ഞകാല ടെലിവിഷനുകൾക്ക് സമാനമാണ്. മോണോക്രോം, കളർ എന്നിങ്ങനെ രണ്ട് തരം CRT മോണിറ്ററുകൾ ലഭ്യമാണ്.

ചിത്രം 2.39: CRT മോണിറ്റർ

മോണോക്രോം മോണിറ്റർ കറുത്ത പശ്ചാത്തലത്തിൽ ഏതെങ്കിലും ഒരു നിറത്തിൽ അക്ഷരങ്ങളും ചിത്രങ്ങളും പ്രദർശിപ്പിക്കുന്നു. ചാരനിറത്തിലുള്ള വ്യത്യസ്ത ഷേഡുകൾ പ്രദർശിപ്പിക്കാൻ കഴിയുന്ന മോണോക്രോം മോണിറ്ററുകളുടെ മറ്റൊരു വകഭേദത്തെ ഗ്രേ സ്കെയിൽ മോണിറ്റർ എന്നു വിളിക്കുന്നു. കളർ മോണിറ്ററുകൾ ചുവപ്പ്, നീല, പച്ച എന്നീ അടിസ്ഥാന നിറങ്ങൾ ഉപയോഗിച്ച് 16 മുതൽ 1 മില്ല്യൺ വരെ വ്യത്യസ്ത നിറങ്ങളിൽ അക്ഷരങ്ങളും ചിത്രങ്ങളും പ്രദർശിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്നു. ചില ഗ്രാഫിക് ആർട്ടിസ്റ്റുകൾ അവരുടെ കൃത്യമായ വർണ്ണ വിവർത്തനത്തിനും, ഗെയിമർമാർ അവരുടെ വേഗതയുള്ള ഗ്രാഫിക്സിനോട് കൃത്യമായി പ്രതികരിക്കുന്നതു കൊണ്ടും ഈ മോണിറ്ററുകളെ ഇഷ്ടപ്പെടുന്നു.

ഫ്ളാറ്റ് പാനൽ മോണിറ്റർ

ഫ്ളാറ്റ് പാനൽ ഡിസ്പ്ലേകൾ കനം കുറഞ്ഞതും, ഭാരം കുറഞ്ഞതും, CRT മോണിറ്ററുകളുമായി താരതമ്യപ്പെടുത്തുമ്പോൾ കുറച്ച് ഊർജ്ജം ഉപയോഗിക്കുകയും, കുറച്ച് ചൂട് ഉണ്ടാക്കുന്നതുമാണ്. ഫ്ളാറ്റ് പാനൽ മോണിറ്ററുകൾ സാധാരണയായി ലാപ്ടോപ്പുകളിൽ ഉപയോഗിക്കുന്നു. LCD മോണിറ്ററുകൾ, LED മോണിറ്ററുകൾ, പ്ലാസ്മാ മോണിറ്ററുകൾ, OLED മോണിറ്ററുകൾ എന്നിവയാണ് വിവിധ തരം ഫ്ളാറ്റ് പാനൽ മോണിറ്ററുകൾ.

ലിക്വിഡ് ക്രിസ്ടൽ ഡിസ്പ്ലേ (LCD) മോണിറ്ററുകൾ

ചിത്രം 2.40: LCD

LCD ഡിസ്പ്ലേകൾ രണ്ടു പ്ലാസ്റ്റിക് പ്ലേറ്റുകളുടെ ഇടയിൽ ലിക്വിഡ് ക്രിസ്റ്റൽസ് വെച്ചാണ് നിർമ്മിച്ചിരിക്കുന്നത്. ഒരു വൈദ്യുതപ്രവാഹം കടന്നുപോകുമ്പോൾ ഈ സ്ഫടികങ്ങൾ/ക്രിസ്റ്റൽസ് ഒരു ചിത്രം ഉണ്ടാക്കുന്നതിനായി പുനർക്രമീകരണം നടത്തും. ഈ പ്ലേറ്റിന്റെ പിൻഭാഗത്ത് നിന്നുമുള്ള ഒരു പ്രകാശ സ്രോതസ്സ് ചിത്രത്തെ ദൃശ്യമാക്കുന്നു. ഈ പ്രകാശ സ്രോതസ്സ് ഫ്ളൂറസന്റ് അല്ലെങ്കിൽ LED ആകാം.

ലൈറ്റ് എമിറ്റിംഗ് ഡയോഡ് (LED) മോണിറ്ററുകൾ

LED മോണിറ്ററുകൾ സ്ക്രീനിൽ പ്രകാശിപ്പിക്കാനായി ലിക്വിഡ് ക്രിസ്റ്റൽ ഡിസ്പ്ലേക്ക് (LCD) പിന്നിൽ നേരിട്ട് LED ഉപയോഗിക്കുന്നു. സ്ക്രീനിന്റെ ഓരോ ഭാഗത്തിനും അതിന്റെ വെളിച്ചം ഓൺ അല്ലെങ്കിൽ ഓഫ് നൽകാൻ അതിന് കഴിയും എന്നുള്ളതുകൊണ്ട് ഈ രീതി വളരെ ഫലപ്രദമാണ്. LED സ്ക്രീനുകൾക്ക് വലിയ കോൺട്രാസ്റ്റ് അനുപാതങ്ങൾ ഉള്ളതുകൊണ്ട് കറുത്ത നിറവും മറ്റ് നിറങ്ങളും തമ്മിലുള്ള വ്യത്യാസം പൂർണതയോടെ അനുഭവവേദ്യമാകുന്നു. ഈ സാങ്കേതികവിദ്യ ചെലവേറിയതാണ്. മികച്ച നിറം നിലവാരം, വ്യക്തത, കോണുകളിൽ നിന്നുമുള്ള കാഴ്ച, വേഗത്തിലുള്ള റിഫ്രഷ് നിരക്ക്, പവർ സേവിംഗ്സ് എന്നിവയാണ് LED ഉപയോഗിക്കുന്നത് കൊണ്ടുള്ള മെച്ചങ്ങൾ.

പ്ലാസ്മ മോണിറ്ററുകൾ

സമാന്തര ഇലക്ട്രോഡുകൾ ഉറപ്പിച്ച രണ്ട് സീൽഡ് ഗ്ലാസ് പ്ലേറ്റുകളുടെ ഇടയിൽ നിയോൺ/സിനോൺ ഗ്യാസ് നിറച്ചാണ് പ്ലാസ്മ മോണിറ്ററുകൾ നിർമിക്കുന്നത്. ഈ രണ്ട് ഇലക്ട്രോ

ഡുകൾക്കിടയിൽ ഒരു വോൾട്ടേജ് പൾസ് കടന്നു പോകുമ്പോൾ, വാതകങ്ങൾ വ്യത്യസ്ത നിറങ്ങളായി തിളങ്ങുന്നു. അങ്ങനെ മോണിറ്ററിൽ ചിത്രങ്ങൾ സൃഷ്ടിക്കപ്പെടുന്നു. പ്ലാസ്മ മോണിറ്ററുകൾ ഉയർന്ന റെസല്യൂഷൻ നൽകുമെങ്കിലും ചിലവേറിയതാണ്.

ഓർഗാനിക് ലൈറ്റ് എമിറ്റിംഗ് ഡയോഡ് (OLED) മോണിറ്ററുകൾ

ദശലക്ഷക്കണക്കിന് ചെറിയ LED-കൾ ഉപയോഗിച്ചാണ് OLEDയുടെ പാനൽ നിർമ്മിച്ചിരിക്കുന്നത്. OLED യിലെ O ഓർഗാനിക്കിനെ സൂചിപ്പിക്കുന്നു. ആയതിനാൽ ഇതിന്റെ പാനലിൽ കാർബൺ ഉണ്ട് എന്ന് മനസ്സിലാക്കാം. LCD, LED എന്നിവയെക്കാൾ OLED സ്ക്രീനുകൾ കനം കുറഞ്ഞവയും ഭാരം കുറഞ്ഞവയുമാണ്. മെച്ചപ്പെട്ട ഗുണനിലവാരമുള്ള ഇമേജുകൾ നിർമ്മിക്കാനും ഏതു കോണിൽ നിന്നും മികച്ച കാഴ്ച നൽകാനും OLED ക്ക് കഴിയും. OLED-കൾ ഉപയോഗിക്കുന്ന ഊർജം കുറവാണ്, എന്നാൽ വളരെ ചെലവേറിയതാണ്.

LCD പ്രോജക്ടർ

ചിത്രം 2.41: LCD പ്രോജെക്ടർ

ഒരു വലിയ സ്ക്രീൻ അല്ലെങ്കിൽ പരന്ന പ്രതലത്തിൽ വീഡിയോ, ഇമേജുകൾ അല്ലെങ്കിൽ കമ്പ്യൂട്ടർ ഡാറ്റ പ്രദർശിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്ന വീഡിയോ ഉപകരണമാണ് LCD പ്രൊജക്ടർ. സ്ലൈഡ് പ്രൊജക്റ്ററിന്റെ അല്ലെങ്കിൽ ഓവർഹെഡ് പ്രൊജക്ടറിന്റെ ആധുനിക തുലനമാണിത്. LCD ഡിസ്പ്ലേയിൽ ഉയർന്ന തീവ്രതയുള്ള പ്രകാശത്തിന്റെ ഒരു ബീം ആയിരക്കണക്കിന് മാറുന്ന പിക്സലുകൾ വഴി യാത്ര ചെയ്യുന്നു. പ്രകാശത്തിന്റെ ഈ ബീം ഒരു ലെൻസിലൂടെ കടന്നാണ് സ്ക്രീനിൽ ചിത്രങ്ങൾ നിർമിക്കുന്നത്.

ii. പ്രിൻറർ

ഹാർഡ് കോപ്പി നിർമ്മാണത്തിന് പ്രിന്ററുകൾ ഉപയോഗിക്കുന്നു. ഉപയോഗിച്ച സാങ്കേതിക വിദ്യ അടിസ്ഥാനമാക്കി, അവ മർദ്ദം ഉപയോഗിക്കുന്ന (impact) അല്ലെങ്കിൽ മർദ്ദം ഉപയോഗിക്കാത്ത (non-impact) പ്രിന്ററുകൾ ആയി വർഗീകരിക്കാവുന്നതാണ്. ഇംപാക്റ്റ് പ്രിന്ററുകൾ ടൈപ്പ്റൈറ്റിംഗ് അല്ലെങ്കിൽ പ്രിന്റിംഗ് യാന്ത്രികവിദ്യ ഉപയോഗിക്കുന്നു. ഇവിടെ ഒരു ചുറ്റിക റിബൺ വഴി പേപ്പറിൽ അടിച്ചാണ് പ്രിന്റിംഗ് നടക്കുന്നത്. ഡോട്ട് മാട്രിക്സ് പ്രിന്ററുകൾ ഈ വിഭാഗത്തിൽപ്പെടുന്നു. നോൺ ഇംപാക്റ്റ് പ്രിന്ററുകൾ പേപ്പറിൽ തൊടാതെയാണ് പ്രിന്റ് ചെയ്യുന്നത്. പേപ്പറിലെ അക്ഷരങ്ങൾ പ്രിന്റ് ചെയ്യാൻ അവ വ്യത്യസ്ത സാങ്കേതികവിദ്യകൾ ഉപയോഗിക്കുന്നു. ഇങ്ക്ജെറ്റ്, ലേസർ, തെർമൽ പ്രിന്ററുകൾ എന്നിവ ഇത്തരം പ്രിന്ററുകളാണ്.

പ്രിന്ററിന്റെ ഗുണനിലവാരം നിർണ്ണയിക്കുന്ന രണ്ട് ഘടകങ്ങൾ അതിന്റെ മിഴിവ് (resolution), വേഗത എന്നിവയാണ്. മിഴിവ് അളക്കാൻ DPI (Dots Per Inch) ഉപയോഗിക്കുന്നു. ഒരു യൂണിറ്റ് സമയത്തിൽ അച്ചടിക്കുന്ന അക്ഷരങ്ങളുടെ എണ്ണം അനുസരിച്ചാണ് വേഗത അളക്കുന്നത്. വേഗത സൂചിപ്പിക്കാൻ Characters Per Second (CPS), Lines Per Minute (LPM) അല്ലെങ്കിൽ Pages Per Minute (PPM) എന്നിവ ഉപയോഗിക്കുന്നു.

ഡോട്ട്മാട്രിക്സ് പ്രിന്റർ (DMP)

ചിത്രം 2.42: DMP പ്രിന്റർ

പ്രിന്റ് ഹെഡിലെ ഇലക്ട്രോമാഗ്നെറ്റിക് ആയി ഉത്തേജനം ചെയ്ത ചെറിയ പിന്നുകൾ ഒരു മഷിയുള്ള റിബണിൽ പതിച്ച് ഇമേജുകൾ നിർമ്മിക്കുന്നു. സാധാരണയായി ഉപയോഗിക്കുന്ന പ്രിന്റ് ഹെഡുകളിൽ 9 പിന്നുകൾ അടങ്ങിയിരിക്കുന്നു. മികച്ച പ്രിന്റ് നിലവാരത്തിനായി ചില പ്രിന്ററുകൾ 24 പിന്നുകൾ വരെ ഉപയോഗിക്കുന്നു. ഈ പ്രിന്ററുകൾക്ക് വേഗത കുറവാണ്, അതുപോലെ പ്രിന്റിംഗ് ശബ്ദവുമുണ്ടാക്കും. അതുകൊണ്ടുതന്നെ അത് വ്യക്തിപരമായ ആവശ്യങ്ങൾക്ക് സാധാരണയായി ഉപയോഗിക്കുന്നില്ല. അച്ചടി ചെലവ് കുറവായതിനാലും കാർബൺ പകർപ്പുകൾ ലഭിക്കും എന്നുള്ളതിനാലും ഡോട്ട് മാട്രിക്സ് പ്രിന്ററുകൾ കടകളിലെ ക്യാഷ് കൗണ്ടറുകളിൽ വ്യാപകമായി ഉപയോഗിക്കുന്നു.

ഇങ്ക്ജെറ്റ് പ്രിന്റർ

ചിത്രം 2.43: ഇങ്ക്ജെറ്റ് പ്രിന്റർ

ഇങ്ക്ജെറ്റ് പ്രിന്ററുകൾ പ്രിന്റ് ഹെഡിൽ നിന്ന് മഷിയുടെ ചെറിയ തുള്ളികൾ സ്പ്രേ ചെയ്താണ് പേജ് പ്രിന്റ് ചെയ്യുന്നത്. വർണ ഇമേജുകൾ ഉണ്ടാക്കുന്നതിന് പ്രിന്ററിന് നിരവധി നിറങ്ങളിലുള്ള മഷി (സിയാൻ, മഞ്ഞ, മജന്ത, കറുപ്പ്) ആവശ്യമാണ്. ചില ഫോട്ടോ നിലവാരത്തിലുള്ള ഇങ്ക്ജെറ്റ് പ്രിന്ററുകളിൽ കൂടുതൽ നിറങ്ങളിലുള്ള മഷി ഉണ്ട്. ഇങ്ക്ജെറ്റ് പ്രിന്ററുകൾ വിലകുറഞ്ഞവയാണ്, പക്ഷേ മഷി കാറ്ററിഡ്ജുകളുടെ വില ദീർഘകാലാടിസ്ഥാനത്തിലുള്ള ഉപയോഗത്തെ ചെലവേറിയതാക്കുന്നു.

ലേസർ പ്രിന്റർ

ചിത്രം 2.44: ലേസർ പ്രിന്റർ

ഒരു ലേസർ പ്രിന്റർ മികച്ച ഗുണനിലവാരമുള്ള പ്രിന്റുകൾ നൽകുന്നു. പ്രിന്റ് ചെയ്യേണ്ട ചിത്രം ലേസർ ബീം ഉപയോഗിച്ച് ഒരു ഡ്രമ്മിലേക്ക് പകർത്തുന്നു. ടോണർ കാറ്ററിഡ്ജിൽ നിന്ന് ടോണർ പൗഡർ ഡ്രമ്മിലേക്ക് സ്പ്രേ ചെയ്യുന്നു. ലേസർ ബീം ഉപയോഗിച്ച് ഡ്രമ്മിൽ വരച്ച ഭാഗങ്ങളിൽ ടോണർ പൊടി പറ്റിപ്പിടിക്കുന്നു. ഡ്രം പേപ്പറിന് മുകളിൽ ഉരുട്ടുമ്പോൾ, ടോണർ പൊടി പേപ്പറിൽ പകർന്നു കിട്ടുന്നു. പേപ്പർ ചൂടാക്കുമ്പോൾ പൊടി പേപ്പറിൽ പ്രിന്റ് ആയി മാറുന്നു.

മോണോക്രോം, കളർ എന്നിങ്ങനെ ലേസർ പ്രിന്ററുകൾ ലഭ്യമാണ്. കളർ ലേസർ പ്രിന്ററുകൾ ഒന്നിലധികം വർണ്ണ ടോണർ കാറ്ററിഡ്ജുകൾ ഉപയോഗിക്കുന്നു. അതിനാൽ തന്നെ അത് ചിലവേറിയതാണ്. ലേസർ പ്രിന്ററുകൾ വളരെ വേഗതയുള്ളതാണ്, അവയുടെ വേഗത Pages Per Minute (PPM) –ൽ അളക്കാം.

തെർമൽ പ്രിന്റർ

ചിത്രം 2.45: തെർമൽ പ്രിന്റർ

തെർമൽ പ്രിന്ററിലെ പ്രിന്റിങ് ഹെഡിലൂടെ താപസംവേദിയായ (heat sensitive) തെർമൽ പേപ്പർ കടന്നുപോകുമ്പോൾ ആവശ്യാനുസരണം പേപ്പറിനെ ചൂടാക്കി ഒരു അച്ചടിച്ച ഇമേജ് ഉണ്ടാക്കുന്നു. ചൂടുപിടിച്ച ഭാഗങ്ങൾ കറുപ്പ് നിറത്തിലേക്കു മാറിയാണ് ഒരു ചിത്രം

നിർമ്മിക്കുക. ഡോട്ട്മാട്രിക്സ് പ്രിന്ററുകളേക്കാൾ മിഴിവും വേഗവും തെളിച്ചവുമുള്ള പ്രിന്റുകൾ തെർമൽ പ്രിന്ററുകളിൽ ലഭിക്കുന്നു. അവ ചെറുതും ഭാരം കുറഞ്ഞതും കുറഞ്ഞ ഊർജ്ജം ഉപയോഗിക്കുന്നവയും, ശബ്ദം കുറവുള്ളവയുമാണ്. അതിനാൽ അവയെ സുവഹനീയമായ പ്രിന്ററുകളായി ഉപയോഗിക്കുന്നു. തെർമൽ പ്രിന്ററുകൾ പോയിന്റ് ഓഫ് സെയിൽ ടെർമിനലുകളിലെ ഉപയോഗത്തിന് പ്രശസ്തമാണ്.

സവിശേഷതകൾ	ലേസർ പ്രിന്ററുകൾ	ഇങ്ക്ജെറ്റ് പ്രിന്ററുകൾ	തെർമൽ പ്രിന്ററുകൾ	ഡോട്ട്മെട്രിക്സ് പ്രിന്ററുകൾ
പ്രിന്റ് ചെയ്യാനുപയോഗിക്കുന്ന വസ്തു	പൊടി രൂപത്തിലുള്ള മഷി (ഇങ്ക് പൗഡർ)	ദ്രവരൂപത്തിലുള്ള മഷി	താപസംവേദിയായ പേപ്പർ	മഷി പുരട്ടിയ റിബൺ
എങ്ങിനെ ഇത് പ്രിന്റ് ചെയ്യുന്നു	പൊടി ചൂടാക്കി പേപ്പറിൽ ചേർക്കുന്നു.	അതിസൂക്ഷ്മമായ കുഴലുകളിലൂടെ ദ്രാവകരൂപത്തിലുള്ള മഷി പേപ്പറിൽ തളിക്കുന്നു.	താപസംവേദിയായ പേപ്പറിന് മുകളിലൂടെ തെർമൽ പ്രിന്റ് ഹെഡ് കടന്നു പോകുന്നു.	പേപ്പറിന് മുകളിലുള്ള റിബണിൽ പിന്നുകൾ അമർത്തുമ്പോൾ
പ്രിന്റിങ്ങ് വേഗത	മിനുട്ടിൽ 20 പേജുകൾ	മിനുട്ടിൽ 6 പേജുകൾ	സെക്കന്റിൽ 150 mm	ഒരു സെക്കന്റിൽ 30 മുതൽ 550 ക്യാരക്ടറുകൾ വരെ
ഗുണമേന്മ	നല്ല ഗുണനിലവാരമുള്ള പ്രിന്റിംഗ്. കറുപ്പ് വെളുപ്പ് നിറത്തിലുള്ള പ്രിന്റിങ്ങിന് ഏറ്റവും യോജിച്ചത്.	നല്ല ഗുണനിലവാരമുള്ള പ്രിന്റിംഗ്. ചെറിയ ഫോണ്ടുകൾക്ക് യോജിച്ചത്.	ചിത്രങ്ങൾ പ്രിന്റ് ചെയ്യുന്നതിന് യോജിച്ചതല്ല, ടെക്സ്റ്റുകൾ പ്രിന്റ് ചെയ്യാൻ നല്ലത്.	മോശം നിലവാരമുള്ള ചിത്രങ്ങളുടെ പ്രിന്റിംഗ്. നിലവാരമുള്ള അക്ഷരങ്ങളുടെ പ്രിന്റിംഗ്.
നേട്ടങ്ങൾ	നിശബ്ദമായി, വേഗത്തിൽ പ്രിന്റ് ചെയ്യുന്നു, ഉയർന്ന പ്രിന്റ് നിലവാരം.	നിശബ്ദമായി, ഉയർന്ന പ്രിന്റ് നിലവാരം, പ്രിന്റർ തയ്യാറാകാനുള്ള സമയം കുറവാണ്, പ്രിന്ററിന്റെ വില കുറവാണ്.	നിശബ്ദമായി, വേഗതയും, ചെറുതും, ഭാരം കുറഞ്ഞതും, കുറഞ്ഞ ഊർജ്ജം ഉപയോഗിക്കുന്നത്, കൊണ്ടുനടക്കാൻ എളുപ്പം.	റിബൺ ഉപയോഗിക്കുന്നത് കൊണ്ട് അച്ചടി ചെലവു കുറഞ്ഞതാണ്. കാർബൺ പകർപ്പ് സാധ്യമാണ്.
ദോഷങ്ങൾ	പേപ്പർ ജാമുകൾക്ക് സാധ്യത. ടോണറും പ്രിന്ററും ചെലവേറിയത്	മഷി വിലയേറിയതാണ്. വെള്ളം നനഞ്ഞാൽ മഷി പടരും. മഷി പുറത്തേക്ക് വരുന്ന കുഴലുകൾ അടഞ്ഞുപോകാൻ സാധ്യതയുണ്ട്.	പ്രത്യേക താപസംവേദിയായ പേപ്പർ ആവശ്യമാണ്. മോശം ഗുണനിലവാരമുള്ള അച്ചടി.	അറ്റകുറ്റപ്പണികൾ ചെലവേറിയതാണ്. അച്ചടി വേഗത്തിൽ അല്ല. ശബ്ദം ഉണ്ടാക്കുന്നു.

പട്ടിക 2.3: പ്രിന്ററുകളുടെ താരതമ്യം

iii. പ്ലോട്ടർ

ഗ്രാഫുകളും, ഡിസൈനുകളും ഹാർഡ്കോപ്പിയായി നിർമ്മിക്കുന്നതിനുള്ള ഒരു ഔട്ട്പുട്ട് ഉപകരണമാണ് പ്ലോട്ടർ. വലിയ ഫോർമാറ്റിലുള്ള ഗ്രാഫുകൾ, നിർമ്മാണ മാപ്പുകൾ, എഞ്ചിനീയറിങ് ഡ്രോയിങ്ങുകൾ, വലിയ പോസ്റ്ററുകൾ എന്നിവ പ്രിന്റ് ചെയ്യാൻ പ്ലോട്ടർ ഉപയോഗിക്കുന്നു. കാറുകൾ, കപ്പലുകൾ, വിമാനങ്ങൾ, കെട്ടിടങ്ങൾ, ഹൈവേകൾ എന്നിവയുടെ രൂപകൽപനയിൽ ഇത് ഉപയോഗിക്കുന്നു. പ്ലോട്ടറുകൾ രണ്ട് തരം ഉണ്ട്: ഡ്രം പ്ലോട്ടറുകൾ, ഫ്ളാറ്റ് ബെഡ് പ്ലോട്ടറുകൾ എന്നിവ.

ഡ്രം പ്ലോട്ടർ

ചിത്രം 2.46: ഡ്രം പ്ലോട്ടർ

ഡ്രം പ്ലോട്ടറിന്റെ മറ്റൊരു പേരാണ് റോളർ പ്ലോട്ടർ. പേപ്പർ ഉറപ്പിച്ച ഒരു ഡ്രം അല്ലെങ്കിൽ റോളർ ഇതിൽ ഉൾക്കൊള്ളുന്നു. ഈ ഡ്രം മുൻപിലേക്കും പുറകിലേക്കും തിരിച്ചാണ് പേപ്പറിൽ ഗ്രാഫ് നിർമ്മിക്കുന്നത്. നിറമുള്ള മഷി പേനകൾ അല്ലെങ്കിൽ പെൻസിലുകൾ അടങ്ങുന്ന ഒരു ഡ്രോയിംഗ് കൈ (drawing arm) ഇതിൽ അടങ്ങിയിരിക്കുന്നു. ഡ്രോയിംഗ് കൈ ഇരു വശത്തേക്കും നീങ്ങുകയും, റോളറിലൂടെ പേപ്പർ മുമ്പിലേക്കും പുറകിലേക്കും നീക്കുകയും ചെയ്താണ് ഒരു കൃത്യതയാർന്ന ഗ്രാഫ് അല്ലെങ്കിൽ മാപ്പ് പേപ്പറിൽ സൃഷ്ടിക്കുന്നത്.

ഫ്ളാറ്റ് ബെഡ് പ്ലോട്ടർ

ചിത്രം 2.47: ഫ്ലാറ്റ് ബെഡ് പ്ലോട്ടർ

ഫ്ളാറ്റ് ബെഡ് പ്ലോട്ടറിന്റെ മറ്റൊരു പേരാണ് ടേബിൾ പ്ലോട്ടർ. ചതുരാകൃതിയിലുള്ള ഫ്ളാറ്റ് ബെഡ് ടേബിളിൽ പേപ്പർ ഉറപ്പിച്ച് ഇത് പ്ലോട്ടുകൾ നൽകുന്നു. ഫ്ളാറ്റ് ബെഡ് പ്ലോട്ടർ രണ്ടു ഡ്രോയിംഗ് കൈകൾ (drawing arms) ഉപയോഗിക്കുന്നു. ഇതിൽ ഓരോന്നിനും നിറമുള്ള മഷി പേനകൾ അല്ലെങ്കിൽ പെൻസിലുകൾ ഉണ്ട്. സ്ഥിരമായി ഉറപ്പിച്ച പേപ്പറിനു മുകളിലൂടെ ഡ്രോയിംഗ് കൈകൾ നീങ്ങി ഫ്ളാറ്റ് ബെഡ് പ്ലോട്ടർ പേപ്പറിൽ ഗ്രാഫുകൾ വരയ്ക്കുന്നു. ഗ്രാഫുകൾ വരയ്ക്കുന്നതിലും പ്രിന്റ് ചെയ്യുന്നതിലും ഫ്ളാറ്റ് ബെഡ് പ്ലോട്ടർ വളരെ പതുക്കെയാണ്. വലുതും സങ്കീർണ്ണവുമായ ഡ്രോയിംഗ് പ്രിന്റ് ചെയ്യാനായി മണിക്കൂറുകളെടുക്കും.

iv. ത്രീ ഡൈമെൻഷനൽ/ത്രിമാന (3D) പ്രിന്റർ

ചിത്രം 2.48: 3D പ്രിന്റർ

3D വസ്തുക്കൾ അച്ചടിക്കാൻ ഉപയോഗിക്കുന്ന ഒരു പുതിയ തലമുറ ഔട്ട്പുട്ട് ഉപകരണമാണ് 3D പ്രിന്റർ. വ്യത്യസ്തമായ പദാർത്ഥങ്ങൾ ഉപയോഗിച്ച് വ്യത്യസ്ത തരത്തിലുള്ള വസ്തുക്കൾ പ്രിന്റ് ചെയ്യാൻ 3D പ്രിന്ററിന് കഴിയും. ഒരു 3D പ്രിന്ററിന് സെറാമിക് കപ്പ് മുതൽ പ്ലാസ്റ്റിക് കളിപ്പാട്ടങ്ങൾ, മെറ്റൽ മെഷീൻ

ഭാഗങ്ങൾ, കല്ല് കൊണ്ടുള്ള പൂച്ചട്ടികൾ, ഫാൻസി ചോക്ലേറ്റ് കേക്കുകൾ മുതലായവ അച്ചടിക്കാൻ കഴിയും.

3D പ്രിന്റിങ് പ്രക്രിയ ഒരു വസ്തുവിനെ ആയിരക്കണക്കിന് തിരശ്ചീനമായ ചെറിയ പാളികളായി അച്ചടിക്കാൻ ശ്രമിക്കുന്നു. അതിനു ശേഷം ഈ പാളികൾ താഴെ നിന്നും മുകളിലേക്ക്, ഓരോ ലെയറുകളായി പ്രിന്റ് ചെയ്യുന്നു. ഈ ചെറിയ പാളികൾ ഒട്ടിച്ചേർന്ന് ഒരു ഖര വസ്തുവിനെ രൂപപ്പെടുത്തുന്നു.

v. ഓഡിയോ ഔട്ട്പുട്ട് ഉപകരണം

കമ്പ്യൂട്ടർ ശബ്ദം സൃഷ്ടിക്കുന്നത് ഓഡിയോ ഔട്ട്പുട്ട് ഉപയോഗിച്ചാണ്. ശബ്ദം പുറപ്പെടുവിക്കുന്ന ഔട്ട്പുട്ട് ഉപകരണമാണ് സ്പീക്കർ. ഇത് ഓഡിയോ പോർട്ടുകളിലൂടെ കമ്പ്യൂട്ടറുമായി ബന്ധിപ്പിച്ചിരിക്കുന്നു. ഓഡിയോ പോർട്ടിൽ നിന്ന് വരുന്ന ഇലക്ട്രിക്കൽ സിഗ്നലുകൾക്കനുസരിച്ച് സ്പീക്കറിന്റെ ഡയഫ്രം മുന്നോട്ടും പിന്നോട്ടും നീങ്ങി ശബ്ദം പുറപ്പെടുവിക്കുന്നു. ഉയർന്ന നിലവാരത്തിലുള്ള ശബ്ദത്തിന് കമ്പ്യൂട്ടറുകളിൽ 2.1 (3 സ്പീക്കറുകൾ), 5.1 (5 സ്പീക്കറുകൾ), 7.1 (7 സ്പീക്കറുകൾ) എന്നിങ്ങനെയുള്ള സ്പീക്കർ സിസ്റ്റങ്ങൾ ഉപയോഗിക്കുന്നു.

ചിത്രം 2.49: സ്പീക്കർ

2.2 ഇ-മാലിന്യം (e-Waste)

ഇലക്ട്രോണിക് ഉപകരണങ്ങൾ ഉപയോഗശൂന്യമാകുമ്പോൾ അവ അറിയപ്പെടുന്നത് ഇ-മാലിന്യം എന്ന പേരിലാണ്. ഉപേക്ഷിക്കപ്പെട്ട കമ്പ്യൂട്ടറുകൾ, ഓഫീസ് ഇലക്ട്രോണിക് ഉപകരണങ്ങൾ, വിനോദത്തിനുപയോഗിക്കുന്ന ഉപകരണങ്ങൾ, മൊബൈൽ ഫോൺ, ടെലിവിഷൻ, റഫ്രിജറേറ്റർ ഇവയെല്ലാം ഇ-വേസ്റ്റ് എന്ന വിഭാഗത്തിൽപെടുന്നു. പുനരുപയോഗം, പുനർവിൽപന, വീണ്ടെടുക്കൽ, പുനരുൽപാദനം എന്നിങ്ങനെ നിർമാർജ്ജനം ചെയ്യാവുന്ന ഉപയോഗം കഴിഞ്ഞ ഇലക്ട്രോണിക് ഉപകരണങ്ങളെയും ഇ-വേസ്റ്റ് ആയി പരിഗണിക്കാം.

ആധുനിക ജീവിതത്തിൽ ഒഴിച്ചു കൂടാൻ കഴിയാത്ത ഇലക്ട്രോണിക് ഉപകരണങ്ങളാണ് ഡസ്ക്ടോപ്പ് കമ്പ്യൂട്ടർ, ലാപ്ടോപ്പ്, മൊബൈൽ ഫോൺ, റഫ്രിജറേറ്റർ, ടെലിവിഷൻ എന്നിവ. നമ്മുടെ ആവശ്യങ്ങൾക്കനുസരിച്ച് ഓരോ വർഷവും പുതിയ ഉപകരണങ്ങൾ നമ്മൾ വാങ്ങുന്നു. ഓരോ വർഷവും 300 ദശലക്ഷം കമ്പ്യൂട്ടറുകളും 1 ലക്ഷം കോടി സെൽഫോണുകളും ഉൽപാദിപ്പിക്കപ്പെടുന്നു. രണ്ടോ മൂന്നോ വർഷത്തിനുള്ളിൽ തന്നെ ഈ ഇലക്ട്രോണിക് ഉപകരണങ്ങൾ ഉപയോഗശൂന്യമായി തീരുന്നു. ഓരോ വർഷവും ലോകത്താകമാനമുള്ള വേസ്റ്റ് 8% നിരക്കിൽ വർദ്ധിക്കുന്നു എന്ന് കണക്കുകൾ സൂചിപ്പിക്കുന്നു.

മാറിക്കൊണ്ടിരിക്കുന്ന സാങ്കേതികവിദ്യയും, മാധ്യമങ്ങളിൽ വരുന്ന മാറ്റങ്ങളും, വിലയിടിവും, ആസൂത്രിതമായ അസ്ഥിരതയും ലോകത്താകമാനമുള്ള ഇ-വേസ്റ്റ് വർദ്ധിക്കുന്നതിന് കാരണമാകുന്നു. 50 ദശലക്ഷം ടൺ ഇ-മാലിന്യം ഓരോ വർഷവും സൃഷ്ടിക്കപ്പെടുന്നു എന്ന് കണക്കുകൾ സൂചിപ്പിക്കുന്നു. ഇതിൽ 15 മുതൽ 20% വരെ മാത്രമേ പുതുക്കി ഉപയോഗിക്കുന്നുള്ളൂ. ബാക്കിയെല്ലാം മണ്ണിൽ കുഴിച്ചു മൂടപ്പെടുകയോ കത്തിച്ചു കളയുകയോ, മറ്റെവിടെയെങ്കിലും ഉപക്ഷിക്കുകയോ ചെയ്യുന്നു. ഇന്ത്യ, ചൈന പോലുള്ള രാജ്യ

ങ്ങളിലും ആഫ്രിക്ക, ലാറ്റിൻ അമേരിക്ക തുടങ്ങിയ ഭൂഖണ്ഡങ്ങളിലും ഇലക്ട്രോണിക് ഉൽപന്നങ്ങളുടെ വിൽപ്പന അടുത്ത 10 വർഷത്തിനുള്ളിൽ കുത്തനെ ഉയരാൻ സാധ്യതയുണ്ട്.

2.2.1 ഇ-മാലിന്യത്തെ കുറിച്ച് നാം എന്തുകൊണ്ട് ഉത്കണ്ഠപ്പെടണം? (Why should we be concerned about e-Waste?)

ഇലക്ട്രോണിക് മാലിന്യം വെറുമൊരു മാലിന്യം അല്ല. മെർക്കുറി, ലെഡ്, കാഡ്മിയം, ബ്രോമിനേറ്റഡ് ഫ്ളൈം റിട്ടാർഡന്റ്സ് എന്നിങ്ങനെ ആരോഗ്യത്തിന് ഹാനികരമായേക്കാവുന്ന വിഷവസ്തുക്കൾ ഇതിൽ അടങ്ങിയിരിക്കുന്നു. വേണ്ടവിധം നിയന്ത്രിച്ചില്ലെങ്കിൽ ഈ വിഷവസ്തുക്കൾ കാൻസർ, പ്രത്യുൽപ്പാദനശേഷി കുറവ്, മറ്റ് ആരോഗ്യപ്രശ്നങ്ങൾ എന്നിവക്ക് കാരണമാകുന്നു. ഇ-മാലിന്യം കുഴിച്ചു മൂടുന്നതിനാൽ 40% വരെ ലെഡ് മണ്ണിൽ കലരുവാൻ ഇടയാക്കുന്നു.

പട്ടിക 2.4 ൽ അപകടകരമായ ചില രാസപദാർഥങ്ങൾ, അവയുടെ ഉറവിടം, പ്രത്യാഘാതം എന്നിവ കൊടുത്തിരിക്കുന്നു.

രാസപദാർത്ഥം	ഉറവിടം	പ്രത്യാഘാതം
ലെഡ്	കമ്പ്യൂട്ടർ മോണിറ്റർ ഗ്ലാസ്സിലും PCB സോൾഡറിലും ഇത് കാണുന്നു.	കേന്ദ്ര നാഡീവ്യൂഹത്തെ ബാധിക്കുന്നു. രക്തചംക്രമണത്തെയും കിഡ്നിയേയും ബാധിക്കുന്നു.
മെർക്കുറി	PCB, എൽ.സി.ഡി. സ്ക്രീനിന്റെ ഉള്ളിലെ ലൈറ്റുകളിലും കാണുന്നു.	ചെറിയ കുട്ടികളുടെ തലച്ചോറിനെയും നാഡീ വ്യൂഹത്തെയും ബാധിക്കുന്നു. മുതിർന്നവരിൽ അവയവ വൈകല്യങ്ങൾ, മാനസിക വൈകല്യങ്ങൾ, മറ്റനേകം രോഗ ലക്ഷണങ്ങൾ എന്നിവ പ്രകടമാകുന്നു.
കാഡ്മിയം	ചിപ് റസിസ്റ്ററുകളിലും സെമികണ്ടക്ടറുകളിലും കാണുന്നു	പലതരത്തിലുള്ള കാൻസറുകൾക്ക് കാരണമാകുന്നു. കാഡ്മിയം കിഡ്നിയിൽ കുമിഞ്ഞുകൂടി അതിന് ദോഷം ചെയ്യുന്നു.
ബ്രോമിനേറ്റഡ് ഫ്ളെയിം റിട്ടാർഡന്റ്സ് (BFRs)	PCB യിലും ചില പ്ലാസ്റ്റിക്കുകളിലും കാണപ്പെടുന്നു.	കാൻസർ സാധ്യത വർദ്ധിപ്പിക്കുന്നു.

പട്ടിക 2.4: അപകടകരമായ രാസപദാർഥങ്ങളും അതിന്റെ ഉറവിടവും പ്രത്യാഘാതവും.

2.2.2 ഇ-മാലിന്യത്തിന് എന്ത് സംഭവിക്കുന്നു? (What happens to the e-Waste?)

നിർഭാഗ്യവശാൽ ഒരു ചെറിയശതമാനം ഇ-മാലിന്യം മാത്രമേ പുതുക്കി ഉപയോഗിക്കുന്നുള്ളൂ. പുതുക്കൽ കേന്ദ്രങ്ങളിൽ കൊണ്ടുപോയാലും നാം പ്രതീക്ഷിക്കുന്നതുപോലെ മിക്കപ്പോഴും അവ യഥാർത്ഥത്തിൽ പുതുക്കപ്പെടുന്നില്ല. CRT മോണിറ്ററുകളിൽ ഡിസ്പ്ലേക്ക് സഹായകരമാകുന്ന ഫോസ്ഫറസും ലെഡും താരതമ്യേന കൂടിയ അളവിൽ കാണപ്പെടുന്നു. ഉപയോഗശൂന്യമായ CRT കളെ 'അപകടകരമായ മാലിന്യം' എന്ന ഗണത്തിലാണ്

അമേരിക്കൻ പരിസ്ഥിതി സംരക്ഷണ ഏജൻസി (United States Environmental Protection Agency- EPA) ഉൾപ്പെടുത്തിയിരിക്കുന്നത്.

ഭൂരിഭാഗം ഇ-മാലിന്യങ്ങളും മാലിന്യകൂമ്പാരങ്ങളിൽ തള്ളുകയോ ഇൻസിനറേറ്റർ ഉപയോഗിച്ച് കത്തിക്കുകയോ ആണ് ചെയ്യുന്നത്. ഇത്തരം വേസ്റ്റിന്റെ അനുചിതമായ നശീകരണ മാർഗ്ഗങ്ങൾ അവലംബിക്കുന്നതോടെ അവയിൽ നിന്നും വിലപിടിപ്പുള്ള വസ്തുക്കൾ ശേഖരിക്കുവാനോ അപകടകരമായ വിഷവസ്തുക്കളെ നിയന്ത്രിക്കാനോ സാധിക്കാതെ വരുന്നു. ഇതിന്റെ ഭാഗമായി ഇവ നമ്മുടെ മണ്ണിനെയും ജലത്തെയും വായുവിനെയും മലിനപ്പെടുത്തുന്നു.

ചിത്രം 2.50 : കേടുവന്നതും കാലഹരണപ്പെട്ടതുമായ ഇലക്ട്രോണിക് വസ്തുക്കൾ

ഇ-മാലിന്യം യാതൊരു കാരണവശാലും മറ്റു ഗൃഹമാലിന്യങ്ങൾക്കൊപ്പം ഉപേക്ഷിക്കാൻ പാടില്ല. ഇവ എവിടെയാണോ ഉള്ളത് അവിടെ വച്ചുതന്നെ വേർതിരിക്കപ്പെടേണ്ടതും വിവിധ സന്നദ്ധ സംഘടനകൾക്ക് കൈമാറ്റം ചെയ്യപ്പെടേണ്ടതുമാണ്. ഇ-മാലിന്യം എന്ന രൂക്ഷമായ പ്രശ്നം പരിഗണിക്കുമ്പോൾ ഗവൺമെന്റോ, വ്യാവസായിക സ്ഥാപനങ്ങളോ, പൊതുജനങ്ങളോ ഇവ നിയന്ത്രിക്കേണ്ടതിനാവശ്യമായ നടപടികൾ കൈക്കൊള്ളേണ്ടത് അത്യന്താപേക്ഷിതമാണ്.

ഇന്ത്യാ ഗവൺമെന്റിന്റെ കേന്ദ്ര മാലിന്യ നിയന്ത്രണ ബോർഡ് (CPCB -സെൻട്രൽ പൊല്യൂഷൻ കൺട്രോൾ ബോർഡ്) രൂപീകരിച്ച 'ഇ-മാലിന്യം നിയന്ത്രണ നിയമങ്ങൾ, 2011' 01—5–2012 മുതൽ നിലവിൽ വന്നു. ഈ നിയമങ്ങൾ ഇലക്ട്രിക്കൽ, ഇലക്ട്രോണിക് ഉപകരണങ്ങളുടെ നിർമ്മാണവും വിൽപനയും പ്രവർത്തനവുമായി ബന്ധപ്പെട്ടു കിടക്കുന്ന എല്ലാവർക്കും (നിർമ്മാതാക്കൾ, ഉപഭോക്താക്കൾ, ശേഖരണകേന്ദ്രം) ബാധകമാണ്. സംസ്ഥാനത്ത് ഇതിന്റെ നടത്തിപ്പും മേൽനോട്ടവും നിർവ്വഹിക്കുന്നത് സംസ്ഥാന മാലിന്യ നിയന്ത്രണ ബോർഡിന്റെ നേതൃത്വത്തിലാണ്.

ഇ-മാലിന്യത്തിന്റെ ശേഖരണത്തിനും നിർമ്മാർജ്ജനത്തിനുംവേണ്ടി കേരള ഗവൺമെന്റ് പ്രത്യേക നിർദ്ദേശം കൊടുത്തിട്ടുണ്ട്. നിർമ്മാതാക്കളുടെയും തദ്ദേശസ്വയംഭരണസ്ഥാപനങ്ങളുടെയും മാലിന്യ നിയന്ത്രണബോർഡിന്റെയും ചുമതലകൾ ഗവൺമെന്റ് വ്യക്തമായി നിർവ്വചിച്ചിട്ടുണ്ട്. നിർമ്മാതാക്കളുടെ തിരികെ വാങ്ങൽ പദ്ധതിയിലൂടെയോ തദ്ദേശസ്വയംഭരണ സ്ഥാപനങ്ങളുടെ നേതൃത്വത്തിലുള്ള തിരിച്ചെടുക്കൽ സംവിധാനങ്ങളിലൂടെയോ ഇലക്ട്രിക്കൽ, ഇലക്ട്രോണിക്സ് പോലുള്ള ഇ-മാലിന്യങ്ങൾ ശേഖരിക്കപ്പെടുന്നു. ഇവയെല്ലാം അംഗീകൃത പുതുക്കൽ കേന്ദ്രത്തിന് കൈമാറുന്നു. പ്രധാനപ്പെട്ട ബ്രാൻഡുകൾ, ഉപയോഗം കഴിഞ്ഞ ഉൽപന്നങ്ങളെല്ലാം തന്നെ നിർമാതാക്കൾക്കു തന്നെ തിരിച്ച് ഏൽപ്പിക്കാൻ നിർദ്ദേശം നൽകിയിട്ടുണ്ട്. അതുമല്ലെങ്കിൽ തദ്ദേശസ്വയംഭരണ സ്ഥാപനങ്ങൾ തയ്യാറാക്കിയിട്ടുള്ള ശേഖരണ കേന്ദ്രങ്ങളിലും തിരികെ ഏൽപ്പിക്കാൻ സംവിധാനമുണ്ട്. ഇ-മാലിന്യം നിർമാർജനത്തെക്കുറിച്ച് ബോധവൽക്കരണ പരിപാടികൾ സംഘടിപ്പിക്കുന്നതിനും പുതുക്കൽ അഥവാ മാലിന്യം നിർമാർജനം ചെയ്യുന്നതിനും ഉള്ള സംവിധാനങ്ങൾ പൊല്യൂഷൻ കൺട്രോൾ ബോർഡ് നിർവ്വഹിച്ചു പോരുന്നുണ്ട്.

2.2.3 ഇ-മാലിന്യം നിർമാർജന മാർഗ്ഗങ്ങൾ (e-waste disposal methods)

ഇ-മാലിന്യം നിർമാർജനത്തിനുവേണ്ടി താഴെ പറയുന്ന മാർഗ്ഗങ്ങൾ ഉപയോഗിക്കാം.

a) പുനരുപയോഗം (Reuse): സെക്കന്റ് ഹാൻഡ് ഉപയോഗം അഥവാ കേടുപാടുകൾ പരിഹരിച്ച് മെച്ചപ്പെടുത്തി ഉപയോഗിക്കുക എന്നതാണ് പുനരുപയോഗം കൊണ്ട് ഇവിടെ ഉദ്ദേശിക്കുന്നത്. മിക്ക പഴയ കമ്പ്യൂട്ടറുകളും ബന്ധുക്കൾക്കോ സുഹൃത്തുക്കൾക്കോ ചില്ല റക്കച്ചവടം നടത്തുന്നവർക്കോ പൈസക്കോ അല്ലാതെയോ കൈമാറാം. ചിലത് സന്നദ്ധ സംഘടനകൾ, വിദ്യാഭ്യാസ സ്ഥാപനങ്ങൾ എന്നിവയ്ക്കോ കൈമാറാം. ഇങ്ക്ജെറ്റ് കാട്രി ഡ്ജുകളും ലേസർ ടോണറുകളും പുനരുപയോഗം ചെയ്യാം. ഇത് ഇ-വേസ്റ്റിന്റെ തോത് കുറയ്ക്കാൻ സഹായിക്കും.

b) കത്തിച്ചു കളയൽ (Incineration): പ്രത്യേകം രൂപകൽപന ചെയ്ത ഇൻസിനറേറ്ററിൽ 900 മുതൽ 1000 ഡിഗ്രി സെൽഷ്യസ് വരെ ഉള്ള ഉയർന്ന ഊഷ്മാവിൽ നിയന്ത്രണവിധേ യമായി കത്തിച്ചു കളയുന്നു.

c) ഇ-മാലിന്യത്തിന്റെ പുനരുൽപാദനം (Recycling): ഉൽപന്നങ്ങളിൽ നിന്നും ഉപയോഗിക്കാൻ പറ്റുന്ന ഘടകങ്ങളുപയോഗിച്ച് പുതിയ ഉപകരണങ്ങൾ നിർമ്മിക്കുന്നതിനെ യാണ് പുനരുൽപാദനം എന്ന് പറയുന്നത്. മോണിറ്ററുകൾ, കീബോർഡുകൾ, ലാപ്ടോപ്പു കൾ, മോഡം, ടെലിഫോൺ ബോർഡുകൾ, ഹാർഡ് ഡ്രൈവുകൾ, സിഡി, മൊബൈലുകൾ, ഫാക്സ് മെഷീൻ, പ്രിന്റർ, സി.പി.യു, മെമ്മറി ചിപ്പുകൾ, കൂട്ടിയോജിപ്പിക്കുന്ന വയറുകൾ, കേബിളുകൾ എന്നിവയെല്ലാം പുതുക്കി ഉപയോഗിക്കാം.

d) മണ്ണിട്ടുമൂടൽ (Land Filling) : ഏറ്റവും കൂടുതൽ ഉപയോഗിക്കുന്നതും എന്നാൽ ശുപാർശ ചെയ്യപ്പെടാത്തതുമായ മാർഗ്ഗമാണ് മണ്ണിട്ടു മൂടൽ.

2.2.4 ഇ-മാലിന്യം നിർമ്മാർജ്ജനത്തിൽ വിദ്യാർത്ഥികളുടെ പങ്ക് (Students' role in e-Waste disposal)

- ആവശ്യമില്ലാതെ ഇലക്ട്രോണിക് ഉപകരണങ്ങൾ വാങ്ങുന്നത് നിർത്തുക.
- ഉപകരണങ്ങൾ കേടുവരുമ്പോൾ പുതിയത് വാങ്ങുന്നതിന് പകരം അവ നന്നാക്കി ഉപയോഗിക്കുക.
- ഉപകരണങ്ങൾ പാഴാക്കാതെ അവ മറ്റുള്ളവർക്ക് വിൽക്കുകയോ സംഭാവനയായി നൽകുകയോ ചെയ്ത് അവയുടെ പ്രവർത്തന കാലയളവ് ദീർഘിപ്പിക്കുക.
- പുതിയ ഇലക്ട്രോണിക് ഉപകരണങ്ങൾ വാങ്ങുമ്പോൾ അപകടകരമായ പദാർത്ഥ ങ്ങൾ അടങ്ങിയിട്ടില്ലെന്ന് ഉറപ്പു വരുത്തുകയും, പുതുക്കി ഉപയോഗിക്കാവുന്നവയാ ണെന്നും, ഊർജ്ജം കുറച്ച് മാത്രം ഉപയോഗിക്കുന്നവയാണെന്നും, കൂടുതൽ കാലം ഉപയോഗിക്കാവുന്നവയാണെന്നും, വളരെ കുറഞ്ഞ ദുർവ്യയം മാത്രം സൃഷ്ടിക്കുന്ന വയാണെന്നും ഉറപ്പു വരുത്തുക.
- ഉപയോഗ ശൂന്യമായാൽ ഉപകരണങ്ങൾ തിരികെ എടുക്കുന്ന പദ്ധതിയുണ്ടോ എന്ന് നിർമ്മാതാക്കളുടെ ഫോൺ നമ്പർ വഴിയോ വെബ്സൈറ്റ് വഴിയോ മനസ്സിലാക്കുക.
- ബാറ്ററിയിൽ പ്രവർത്തിക്കുന്ന ഉപകരണങ്ങളിൽ ഉപയോഗശേഷം കളയുന്ന ബാറ്ററിക്കു പകരം റീചാർജ്ജ് ചെയ്യാവുന്ന ബാറ്ററികൾ ഉപയോഗിക്കുക.
- ഗുണമേൻമ ഉത്തരവാദിത്വം (Warranty) ഉള്ളതും തിരികെ എടുക്കുന്നതുമായ ഉൽപ ന്നങ്ങൾ വാങ്ങുക.

2.3. ഹരിത കമ്പ്യൂട്ടിംഗ്/ഹരിത സാങ്കേതികവിദ്യ (Green computing or Green IT)

പരിസ്ഥിതിക്ക് നാശം സംഭവിക്കാതെയുള്ള സാങ്കേതികവിദ്യയുടെ പഠനവും പ്രയോഗവുമാണ് ഹരിത കമ്പ്യൂട്ടിംഗ് അഥവാ ഹരിത സാങ്കേതികവിദ്യ. കമ്പ്യൂട്ടറിന്റെയും അനുബന്ധ ഘടകങ്ങളുടെയും രൂപകൽപന, നിർമാണം, ഉപയോഗം, നിർമ്മാർജ്ജനം എന്നിവയുമായി ബന്ധപ്പെട്ട ഘടകങ്ങളായ മോണിറ്ററുകൾ, പ്രിന്ററുകൾ, സംഭരണ ഉപകരണങ്ങൾ എന്നിവ ഫലപ്രദമായി പരിസ്ഥിതിക്ക് യോജിക്കുന്നവിധം നടപ്പാക്കുന്നതിനെയാണ് ഹരിത കമ്പ്യൂട്ടിംഗ് എന്ന് വിളിക്കുന്നത്.

ചിത്രം 2.51 : ഇനർജി സ്റ്റാർ ലേബൽ

ഹരിത കമ്പ്യൂട്ടിംഗിന്റെ ആരംഭകാല 'എനർജി സ്റ്റാർ' എന്നറിയപ്പെടുന്ന സ്വമേധയാ ഉള്ള ലേബലിംഗ് പ്രോഗ്രാം ആയിട്ടായിരുന്നു. എല്ലാവിധ ഹാർഡ്‌വെയറുകളിലും ഊർജ്ജ കാര്യക്ഷമത വർദ്ധിപ്പിക്കുന്നതിനായി 1992-ൽ EPA ആണ് ഇത് നടപ്പിലാക്കിയത്. നോട്ട്ബുക്ക് കമ്പ്യൂട്ടറുകളിലും, ഡിസ്പ്ലേകളിലും എനർജി സ്റ്റാർ ലേബൽ സാധാരണ കാഴ്ചയാണ്. യൂറോപ്പിലും, ഏഷ്യയിലും ഈ പദ്ധതി നടപ്പാക്കപ്പെട്ടു. ചിത്രം 2.51 ൽ സാധാരണയായി ഉപയോഗിക്കുന്ന എനർജി സ്റ്റാർ അടയാളം കാണിച്ചിരിക്കുന്നു.

ഹരിത കമ്പ്യൂട്ടിംഗ് എന്ന ആശയത്തിന്മേലുള്ള ഗവൺമെന്റിന്റെ നിയന്ത്രണം ഭാഗികം മാത്രമാണ്. ആഗോള പരിസ്ഥിതിയെ ദോഷകരമായി ബാധിക്കാതെയുള്ള ഒരു തൊഴിൽ സംസ്കാരം വളർത്തിയെടുക്കാൻ കമ്പ്യൂട്ടർ ഉപയോഗിക്കുന്നവരും സ്ഥാപനങ്ങളും ശ്രദ്ധിക്കേണ്ടതായുണ്ട്. അതിനായി ചെയ്യേണ്ട ചില കാര്യങ്ങൾ താഴെ കൊടുത്തിരിക്കുന്നു.

- കമ്പ്യൂട്ടർ ഉപയോഗിക്കാത്തപ്പോൾ അത് ഓഫ് ചെയ്യുക.
- ലേസർ പ്രിന്റർ പോലുള്ള പെരിഫറലുകൾ ആവശ്യമുള്ളപ്പോൾ മാത്രം പ്രവർത്തിപ്പിക്കുക.
- ഊർജ്ജ സംരക്ഷണ സമ്പ്രദായം ഉപയോഗിക്കുക.
- ഡസ്ക്ടോപ്പ് കമ്പ്യൂട്ടറിനുപകരം ലാപ്പ്ടോപ്പ് സാധിക്കുന്നിടത്തോളം ഉപയോഗിക്കുക.
- ആവശ്യമാണെങ്കിൽ മാത്രം പ്രിന്റൗട്ട് എടുക്കുക.
- CRT മോണിറ്ററുകൾക്കു പകരം LCD മോണിറ്ററുകൾ ഉപയോഗിക്കുക.
- എനർജി സ്റ്റാർ അടയാളമുള്ള ഹാർഡ്‌വെയർ, സോഫ്റ്റ്‌വെയർ ഉപയോഗിക്കുക.
- കേന്ദ്ര, സംസ്ഥാന, പ്രാദേശിക നിയന്ത്രണങ്ങൾക്കനുസരിച്ച് ഇ-വേസ്റ്റ് നിർമാർജനം ചെയ്യുക.
- സൗരോർജം പോലുള്ള ബദൽ ഊർജ ഉറവിടങ്ങൾ ഉപയോഗിക്കുക.

കമ്പ്യൂട്ടറുകളെ എങ്ങിനെ ഹരിതാഭമാക്കാം (How to make computers Green)

കമ്പ്യൂട്ടറുകളെ ഹരിതാഭമാക്കി മാറ്റുന്നതിൽ അവയുടെ വലുപ്പം, കാര്യക്ഷമത, അതില ടങ്ങിയിരിക്കുന്ന വസ്തുക്കൾ എന്നിവ പ്രാധാന്യമർഹിക്കുന്നു. ചെറിയ കമ്പ്യൂട്ടറുകൾ കൂടുതൽ ഹരിതാഭമാണ്. എന്തുകൊണ്ടെന്നാൽ അവ കുറച്ച് വസ്തുക്കളെ ഉപയോഗി ക്കുന്നുള്ളൂ. കൂടാതെ അവയുടെ പ്രവർത്തനത്തിന് കുറഞ്ഞ വൈദ്യുതിയേ ആവശ്യമുള്ളൂ. ഹരിത കമ്പ്യൂട്ടിങ്ങിന്റെ കാര്യത്തിൽ ഊർജ്ജത്തിന്റെ കാര്യക്ഷമമായ വിനിയോഗം പ്രാധാ ന്യമർഹിക്കുന്നു. ലാപ്ടോപ്പുകൾ വലിയ കമ്പ്യൂട്ടറുകളെ അപേക്ഷിച്ച് ഊർജ്ജ വിനി യോഗം കുറവാണ്. അതുപോലെ തന്നെ LCD സ്ക്രീനുകൾ CRT മോഡലുകളെ അപേ ക്ഷിച്ച് വളരെ കുറച്ച് ഊർജ്ജം മാത്രമേ ഉപയോഗിക്കുന്നുള്ളൂ. ലെഡ്, മെർക്കുറി പോലുള്ള അപകടകരമായ വസ്തുക്കളുടെ ഉപയോഗം കുറക്കണം.

ഹരിത കമ്പ്യൂട്ടിംഗ് പ്രോത്സാഹിപ്പിക്കുന്നതിനുവേണ്ടി താഴെ പറയുന്ന 4 സമീപനങ്ങൾ ഉൾക്കൊള്ളിച്ചിരിക്കുന്നു.

ഹരിത രൂപകൽപന (Green design): കമ്പ്യൂട്ടറുകൾ, സെർവറുകൾ, പ്രിന്ററുകൾ, പ്രൊജക്ടറുകൾ, മറ്റ് ഡിജിറ്റൽ ഉപ കരണങ്ങൾ എന്നിവ രൂപകൽപ്പന ചെയ്യുമ്പോൾ അവ പരിസ്ഥി തിക്ക് അനുയോജ്യമായും ഊർജ്ജ കാര്യക്ഷമമായും ഉപയോ ഗിക്കാൻ തരത്തിൽ തയ്യാറാക്കുക.

ഹരിത നിർമാണം (Green manufacturing): കമ്പ്യൂട്ടറും മറ്റ് അനു ബന്ധ ഘടകങ്ങളും നിർമ്മിക്കുമ്പോൾ ദുർവ്യയം പരമാവധി കുറച്ചു കൊണ്ട് പരിസ്ഥിതിക്ക് ദോഷം ഒന്നും ഉണ്ടാക്കാത്ത രീതിയിൽ തയ്യാറാക്കുക.

ഹരിത ഉപയോഗം (Green use): കമ്പ്യൂട്ടറിന്റെയും അനുബന്ധ ഉപകരണങ്ങളുടെയും വൈദ്യുത ഉപഭോഗം കുറച്ചു കൊണ്ട് പരിസ്ഥിതി സൗഹാർദ്ദപരമായി ഉപയോഗിക്കുക.

ഹരിത നിർമാർജനം (Green disposal) : കമ്പ്യൂട്ടർ കേടുപാടുകൾ തീർത്ത് ഉപയോഗി ക്കുക, ഉചിതമായ രീതിയിൽ നിർമ്മാർജ്ജനം ചെയ്യുക, ആവശ്യമില്ലാത്ത ഇലക്ട്രോ ണിക് ഉപകരണങ്ങൾ പുതുക്കി ഉപയോഗിക്കുക.

സ്വയം വിലയിരുത്താം

1. പാരിസ്ഥിതിക ഉത്തരവാദിത്വത്തോടും പ്രകൃതിക്ക് യോജിച്ചതുമായ രീതിയിൽ കമ്പ്യൂട്ട റുകളും അവയുടെ വിഭവങ്ങളും ഉപയോഗിക്കുന്നതിനെ ____________ എന്നു പറ യുന്നു
2. ഉൽപന്നങ്ങളിൽ നിന്നും ഉപയോഗിക്കാൻ പറ്റുന്ന ഘടകങ്ങൾ ഉപയോഗിച്ച് പുതിയ ഉപ കരണങ്ങൾ നിർമിക്കുന്നതിനെ ____________ എന്നു വിളിക്കുന്നു.
3. കമ്പ്യൂട്ടറുകളിലും അവയുടെ വിഭവങ്ങളിലും ഊർജ്ജ കാര്യക്ഷമത വർദ്ധിപ്പിക്കുന്ന ലേബലിങ്ങ് പരിപാടിയെ ____________ എന്നു വിളിക്കുന്നു.
4. ഏതെങ്കിലും രണ്ട് ഇൻപുട്ട്/ ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ വീതം പട്ടികപ്പെടുത്തുക.

നമുക്കു ചെയ്യാം

1. *പരിസ്ഥിതിക്കും ജനങ്ങളുടെ ആരോഗ്യത്തിനും ഇ - വേസ്റ്റ് ഉണ്ടാക്കുന്ന പ്രത്യാഘാതത്തെക്കുറിച്ച് പഠിക്കുന്നതിനായി ഒരു സർവ്വെ നിങ്ങളുടെ പ്രദേശത്ത് നടത്തി റിപ്പോർട്ട് എഴുതുക.*
2. *ഹരിത കമ്പ്യൂട്ടിംഗിന്റെ പ്രധാന്യത്തെ കുറിച്ച് ചർച്ച ചെയ്യുക.*

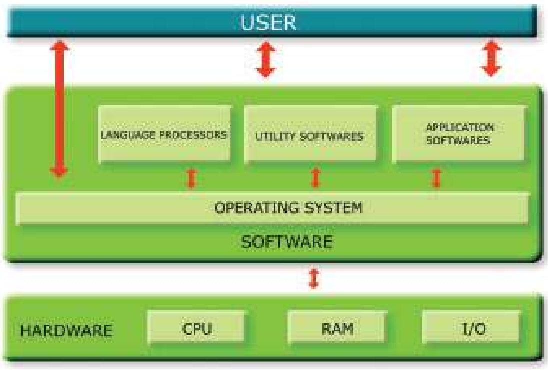
2.4 സോഫ്റ്റ്‌വെയർ (Software)

ഒരുകൂട്ടം പ്രോഗ്രാമുകൾ ഉപയോഗിച്ച് കമ്പ്യൂട്ടർ സിസ്റ്റത്തിന്റെയും മറ്റ് ഇലക്ട്രോണിക് ഉപകരണങ്ങളുടെയും പ്രവർത്തനം കാര്യക്ഷമവും ഫലപ്രദവുമായി നടത്താൻ സഹായിക്കുന്നവയാണ് സോഫ്റ്റ്‌വെയർ. ഹാർഡ്‌വെയർ കമ്പ്യൂട്ടർ സിസ്റ്റത്തിന്റെ ശരീരം രൂപപ്പെടുത്തുമെന്ന് പറയാമെങ്കിൽ സോഫ്റ്റ്‌വെയർ അതിന്റെ മനസ്സോ ആത്മാവോ ആകുന്നു. രണ്ടു തരത്തിലുള്ള സോഫ്റ്റ്‌വെയറാണുള്ളത്.

- സിസ്റ്റം സോഫ്റ്റ്‌വെയർ
- ആപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയർ

2.4.1. സിസ്റ്റം സോഫ്റ്റ്‌വെയർ (System software)

കമ്പ്യൂട്ടറിന്റെ പ്രവർത്തനങ്ങൾ നിയന്ത്രിക്കുന്നതിനായി രൂപകൽപന ചെയ്തിട്ടുള്ള ഒരു കൂട്ടം പ്രോഗ്രാമുകളെയാണ് സിസ്റ്റം സോഫ്റ്റ്‌വെയർ എന്ന് പറയുന്നത്. കമ്പ്യൂട്ടറിന്റെ പ്രവർത്തനങ്ങളെ നിയന്ത്രിച്ചുകൊണ്ടും, കമ്പ്യൂട്ടർ സംവിധാനത്തിന്റെ അകത്തേയ്ക്കും പുറത്തേയ്ക്കും ഡാറ്റ എത്തിച്ചുകൊണ്ടും, ആപ്ലിക്കേഷൻ പ്രോഗ്രാമുകളുടെ കൃത്യനിർവഹണത്തിന്റെ എല്ലാ ഘട്ടങ്ങളും ചെയ്തുകൊണ്ടും കമ്പ്യൂട്ടർ സിസ്റ്റത്തിന്റെ ഉപയോഗത്തിൽ മനുഷ്യനെ സഹായിക്കാൻ വേണ്ടി രൂപകൽപന ചെയ്ത പൊതു പ്രോഗ്രാമുകളാണ് അവ. ചുരുക്കത്തിൽ സിസ്റ്റം സോഫ്റ്റ്‌വെയർ മറ്റ് സോഫ്റ്റ്‌വെയറുകളുടെ പ്രവർത്തനത്തെ പിന്തുണക്കുകയും പെരിഫറൽ ഉപകരണങ്ങളുമായി ആശയവിനിമയം നടത്തുകയും ചെയ്യുന്നു. കമ്പ്യൂട്ടർ ഫലപ്രദമായി ഉപയോഗിക്കുന്നതിന് ഉപയോക്താക്കളെ ഇത് സഹായിക്കുന്നു. കമ്പ്യൂട്ടറിലെ വിഭവങ്ങൾ നിയന്ത്രിക്കുവാൻ സിസ്റ്റം സോഫ്റ്റ്‌വെയർ സഹായിക്കുന്നു എന്ന് ഇത് സൂചിപ്പിക്കുന്നു.



ചിത്രം 2.52 : സോഫ്റ്റ്‌വെയറിലൂടെ ഉപയോക്താവുംഹാർഡ്‌വെയറും തമ്മിലുള്ള സമ്പർക്കം

സിസ്റ്റം സോഫ്റ്റ്‌വെയർ, ഉപയോക്താവിനെയും ഹാർഡ്‌വെയറിനെയും എങ്ങനെ ബന്ധിപ്പിച്ചിരിക്കുന്നു എന്ന് ചിത്രം 2.52 ൽ കാണിക്കുന്നു.

സിസ്റ്റം സോഫ്റ്റ്‌വെയറിന്റെ ഘടകങ്ങൾ താഴെ കൊടുക്കുന്നു.

a) ഓപ്പറേറ്റിംഗ് സിസ്റ്റം

b) ഭാഷ പ്രൊസസ്സറുകൾ

c) യൂട്ടിലിറ്റി സോഫ്റ്റ്‌വെയർ

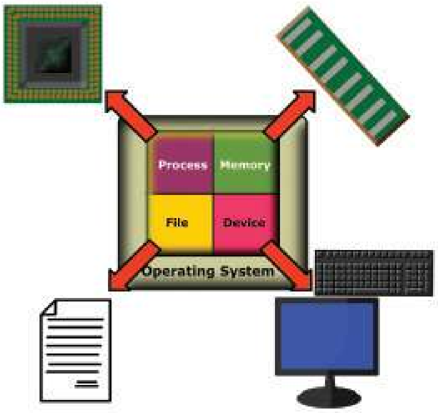
a) ഓപ്പറേറ്റിംഗ് സിസ്റ്റം (Operating system)

ഉപയോക്താവിനേയും കമ്പ്യൂട്ടർ ഹാർഡ്‌വെയറിനെയും ബന്ധിപ്പിക്കാനായുള്ള ഒരുകൂട്ടം പ്രോഗ്രാമുകളെയാണ് ഓപ്പറേറ്റിംഗ് സിസ്റ്റം എന്നുപറയുന്നത്. കമ്പ്യൂട്ടർ സിസ്റ്റത്തെ ഉപയോഗ്യമാക്കുക എന്നതാണ് ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന്റെ പ്രാഥമികലക്ഷ്യം. ഉപയോക്താവിന് പ്രോഗ്രാമുകൾ പ്രവർത്തിപ്പിക്കാൻ അനുയോജ്യമായ പരിസ്ഥിതി ഒരുക്കിക്കൊടുക്കുകയാണ് ഓപ്പറേറ്റിംഗ് സിസ്റ്റം ചെയ്യുന്നത്. കാര്യക്ഷമമായ രീതിയിൽ കമ്പ്യൂട്ടർ ഹാർഡ്‌വെയർ ഉപയോഗിക്കാൻ സഹായിക്കുക എന്നതും ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന്റെ ജോലിയാണ്.

കമ്പ്യൂട്ടറിലെ എല്ലാവിധ പ്രവർത്തനങ്ങളെയും നിയന്ത്രിക്കുകയും ഏകോപിപ്പിക്കുകയും ചെയ്യുന്നത് ഓപ്പറേറ്റിംഗ് സിസ്റ്റമാണ്. ചിത്രം 2.54 ൽ കമ്പ്യൂട്ടർ സിസ്റ്റത്തിലെ വിഭവങ്ങളുടെ മാനേജരായി (Resouce manager) ഓപ്പറേറ്റിംഗ് സിസ്റ്റം പ്രവർത്തിക്കുന്നത് കാണിക്കുന്നു. ഏറ്റവും പ്രധാനപ്പെട്ട സിസ്റ്റം സോഫ്റ്റ്‌വെയറാണ് ഓപ്പറേറ്റിംഗ് സിസ്റ്റം. കമ്പ്യൂട്ടറിലെ ഹാർഡ് ഡിസ്കിൽ നിന്നും ആദ്യം എടുക്കുന്ന പ്രോഗ്രാമും ഓഫാക്കുന്നതുവരെ മെമ്മറിയിൽ നിലനിൽക്കുന്ന പ്രോഗ്രാമുമാണിത്. കമ്പ്യൂട്ടറിന്റെ അനുചിതമായ ഉപയോഗവും തെറ്റുകൾ സംഭവിക്കുന്നത് തടയാനും ഇത് ശ്രമിക്കുന്നു.

ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന്റെ പ്രധാനധർമ്മങ്ങൾ (Functions of operating system)

ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന്റെ പ്രധാനപ്പെട്ട ധർമ്മങ്ങളാണ് പ്രോസസ്സ് കൈകാര്യംചെയ്യുക, മെമ്മറി കൈകാര്യം ചെയ്യുക, ഫയൽ കൈകാര്യചെയ്യുക, സെക്യൂരിറ്റി കൈകാര്യം ചെയ്യുക, നിർദ്ദേശങ്ങൾ വ്യാഖ്യാനിക്കുക തുടങ്ങിയവ.



ചിത്രം 2.53 ഓപ്പറേറ്റിംഗ് സിസ്റ്റം വിഭവങ്ങളുടെ മാനേജർ എന്ന നിലയിൽ

i പ്രോസസ്സ് കൈകാര്യം ചെയ്യുക

പ്രോസസ്സുകളുടെ വിന്യാസവും തിരിച്ചെടുക്കലും, വിവിധ പ്രൊസസ്സുകൾക്ക് വിഭവങ്ങൾ നൽകുന്നതിനുള്ള പദ്ധതി തയാറാക്കൽ എന്നിവയാണ് ഓപ്പറേറ്റിംങ് സിസ്റ്റത്തിലെ പ്രോസസ്സ് കൈകാര്യം ചെയ്യുന്ന ഭാഗം ശ്രദ്ധപുലർത്തുന്നത്.

ii മെമ്മറി കൈകാര്യം ചെയ്യുക

പ്രാഥമിക മെമ്മറിയെ കൈകാര്യം ചെയ്യുകയോ നിയന്ത്രിക്കുകയോ ചെയ്യുന്നത് ഓപ്പ റേറ്റിംഗ് സിസ്റ്റത്തിലെ മെമ്മറി കൈകാര്യം ചെയ്യുന്ന ഭാഗമാണ്. ഓരോ മെമ്മറി സ്ഥാന ത്തിന്റെയും അവസ്ഥ (ഒരു പ്രോസസ്സിന് നീക്കിവെച്ചിരിക്കുകയാണോ അതോ ഒഴിഞ്ഞു കിടക്കുകയാണോ) അത് സൂക്ഷിച്ചുവെക്കുന്നു. ഓരോ പ്രോസസ്സിനും എത്രമാത്രം മെമ്മറി നീക്കിവെക്കണമെന്ന് അത് കണക്കാക്കുകയും അതു പ്രകാരം നീക്കി വെക്കുകയും ചെയ്യു ന്നു. മെമ്മറി പിന്നീട് ആവശ്യമില്ലെങ്കിൽ അത് തിരിച്ചെടുക്കുകയും ചെയ്യുന്നു.

iii ഫയൽ കൈകാര്യം ചെയ്യുക

ഫയലുകളുമായി ബന്ധപ്പെട്ട പ്രവർത്തനങ്ങളായ ആസൂത്രണം ചെയ്യുക, പേര് കൊടു ക്കുക, സംഭരിക്കുക, തിരിച്ചെടുക്കുക, കൈമാറ്റം ചെയ്യുക, സംരക്ഷിക്കുക എന്നീ പ്രവർത്ത നങ്ങൾ ചെയ്യുന്നു.

iv ഡിവൈസ് കൈകാര്യം ചെയ്യുക

കമ്പ്യൂട്ടറുമായി ബന്ധിപ്പിച്ചിരിക്കുന്ന ഉപകരണങ്ങളുടെ നിയന്ത്രണമാണ് ഡിവൈസ് കൈ കാര്യംചെയ്യൽ. ഹാർഡ്‌വെയറും സോഫ്റ്റ്‌വെയറും സംയോജിപ്പിച്ചുകൊണ്ട് അത് ഉപക രണങ്ങളെ കൈകാര്യം ചെയ്യുന്നു. ഡിവൈസ് ഡ്രൈവർ സോഫ്റ്റ്‌വെയറിലൂടെ ഓപ്പറേ റ്റിംഗ് സിസ്റ്റം ഹാർഡ്‌വെയർ ഉപകരണങ്ങളുമായി സംവദിക്കുന്നു. ഡോസ് (DOS), വിൻ ഡോസ്, യുണിക്സ്, ലിനക്സ്, മാക് ഒ.എസ്. എന്നിവ വിവിധതരം ഓപ്പറേറ്റിംങ് സിസ്റ്റ ങ്ങൾക്കുള്ള ഉദാഹരണങ്ങളാണ്.

b. ഭാഷ പ്രോസസ്സറുകൾ (Language processors)

മനുഷ്യർ തമ്മിൽ ആശയവിനിമയം നടത്താൻ ഭാഷ ഉപയോഗിക്കുന്നു. കമ്പ്യൂട്ടറുമായി ആശയവിനിമയം നടത്താൻ കമ്പ്യൂട്ടറിന് അറിയാവുന്ന ഒരു ഭാഷ ഉപയോക്താവിന് ആവ ശ്യമായി വരുന്നു. കമ്പ്യൂട്ടർ ഭാഷകളെ ഉയർന്നതലത്തിലുള്ള കമ്പ്യൂട്ടർ ഭാഷ (HLL) എന്നും താഴ്ന്ന തലത്തിലുള്ള കമ്പ്യൂട്ടർ ഭാഷ (LLL) എന്നും രണ്ടായി തരംതിരിച്ചിരിക്കുന്നു.

താഴ്ന്നതലത്തിലുള്ള കമ്പ്യൂട്ടർ ഭാഷ യന്ത്രാധിഷ്ഠിതഭാഷ എന്നറിയപ്പെടുന്നു. കമ്പ്യൂട്ട റിൽ ലഭ്യമായ മെമ്മറിയും, രജിസ്റ്ററുകളും ഉപയോഗിച്ച് ഈ ഭാഷയിലുള്ള പ്രോഗ്രാമു കൾ എഴുതപ്പെടുന്നു. ഓരോ കമ്പ്യൂട്ടറിന്റെയും രൂപകൽപന വ്യത്യസ്തമായതു കൊണ്ട് ഓരോ കമ്പ്യൂട്ടറിനും പ്രത്യേക താഴ്ന്നതലത്തിലുള്ള യന്ത്രഭാഷ ഉപയോഗിക്കുന്നു. മെഷീൻ ലാംഗ്വേജും അസംബ്ലി ലാംഗ്വേജും താഴ്ന്നതലത്തിലുള്ള കമ്പ്യൂട്ടർ ഭാഷകളാണ്.

യന്ത്ര ഭാഷ (Machine language) : പ്രത്യേകതരം സൂചനകൾ മാത്രമെ കമ്പ്യൂട്ടറിന് മനസ്സി ലാക്കാൻ സാധിക്കുകയുള്ളൂ എന്ന് നമുക്കറിയാമല്ലോ. 1, 0 എന്നീ ബൈനറി സംഖ്യകളിലൂടെ പ്രതിനിധീകരിക്കപ്പെടുന്ന പ്രത്യേക അടയാളങ്ങൾ മാത്രമേ കമ്പ്യൂട്ടറിന് മനസിലാകൂ എന്ന്

നമുക്കറിയാം. ബൈനറി അക്കങ്ങൾ ഉപയോഗിക്കുന്ന ഭാഷകളെ യന്ത്രഭാഷ എന്നു വിളിക്കുന്നു. യന്ത്ര ഭാഷയിൽ പ്രോഗ്രാം എഴുതുന്നത് വളരെ ബുദ്ധിമുട്ടാണ്. എല്ലാ നിർദ്ദേശങ്ങൾക്കും, 0 ന്റെയും 1 ന്റെയും ദൈർഘ്യമേറിയ സിട്രിങ്ങ് ഓർത്തിരിക്കാൻ സാധ്യമല്ല.

അസംബ്ലി ഭാഷ (Assembly language): അസംബ്ലി ഭാഷ മധ്യവർത്തിയായ പ്രോഗ്രാമിംഗ് ഭാഷയാണ്. അസംബ്ലി ഭാഷകൾ ന്യുമോണിക്കുകൾ ഉപയോഗിക്കുന്നു. ഒരു പ്രവർത്തനത്തിന് കൊടുക്കുന്ന പ്രതീകാത്മകമായ പേരാണ് ന്യൂമോണിക്. ഉദാഹരണമായി സങ്കലനത്തിന് ADD, വ്യവകലനത്തിന് SUB, തുടങ്ങിയവ. യന്ത്രഭാഷയെ അപേക്ഷിച്ച് അസംബ്ലി ഭാഷയിൽ കമ്പ്യൂട്ടർ പ്രോഗ്രാം എഴുതാൻ എളുപ്പമാണ്. ഇത് യന്ത്രാധിഷ്ഠിത ഭാഷയായതിനാൽ പ്രോഗ്രാമർക്ക് കമ്പ്യൂട്ടർ രൂപഘടനയെക്കുറിച്ച് അറിവുണ്ടായിരിക്കേണ്ടതാണ്.

ഉയർന്നതലത്തിലുള്ള ഭാഷ (High level language): ഈ ഭാഷകൾ ഇംഗ്ലീഷ് ഭാഷയെപ്പോലെയുള്ളതും അസംബ്ലി ഭാഷയെക്കാളും യന്ത്രഭാഷയെക്കാളും ലളിതമായി മനസിലാക്കാൻ സാധിക്കുന്നതുമാണ്. ഉയർന്നതലത്തിലുള്ള കമ്പ്യൂട്ടർ ഭാഷകൾ (HLL) കമ്പ്യൂട്ടറിന് മനസ്സിലാക്കാൻ സാധിക്കുകയില്ല. ഉയർന്നതലത്തിലുള്ള ഭാഷകളിൽ എഴുതിയിരിക്കുന്ന കമ്പ്യൂട്ടർ പ്രോഗ്രാമിനെ അതിന് തുല്യമായ യന്ത്രഭാഷയിലേക്ക് മാറ്റുന്നു. ആയതിനാൽ ഇത്തരം ഭാഷകളെ തർജ്ജമ ചെയ്യുന്നതിന് ഒരു ഭാഷ വിവർത്തകനെ (കമ്പൈലറുകളോ ഇന്റർപ്രറ്ററുകളോ) ആവശ്യമായി വരുന്നു. BASIC, C, C++, Java എന്നിവ ഉയർന്നതലത്തിലുള്ള പ്രോഗ്രാമിംഗ് ഭാഷകൾക്ക് ഉദാഹരണങ്ങളാണ്.

ഭാഷ പ്രോസസ്സറിന്റെ ആവശ്യകത

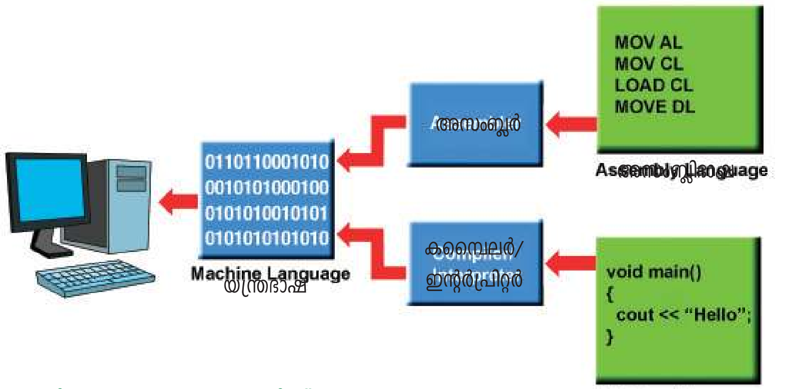
ഉയർന്നതലത്തിലുള്ള ഭാഷകളിലോ അസംബ്ലി ഭാഷകളിലോ എഴുതിയിരിക്കുന്ന പ്രോഗ്രാമുകൾ കമ്പ്യൂട്ടറിന് മനസ്സിലാക്കാൻ സാധിക്കുകയില്ല. ഇങ്ങനെയുള്ള പ്രോഗ്രാമുകളെ യന്ത്ര ഭാഷകളിലേക്ക് (കമ്പ്യൂട്ടറിന് മനസിലാകുന്ന ഭാഷ) മാറ്റുന്നതിന് ഭാഷ പ്രോസസ്സറുകൾ ഉപയോഗിക്കുന്നു. ഉയർന്നതലത്തിലുള്ള ഭാഷകളിലോ അസംബ്ലി ഭാഷകളിലോ എഴുതിയിരിക്കുന്ന പ്രോഗ്രാമുകളെ അതിന് സമാനമായ യന്ത്ര ഭാഷകളിലേക്ക് മാറ്റം ചെയ്യുന്ന സിസ്റ്റം പ്രോഗ്രാമുകളെയാണ് ഭാഷ പ്രൊസസ്സറുകൾ എന്നു വിളിക്കുന്നത്.

വിവിധതരത്തിലുള്ള ഭാഷ പ്രോസസ്സറുകൾ (Types of language processors)

- **അസംബ്ലർ (Assembler):** അസംബ്ലി ഭാഷയിൽ എഴുതിയ പ്രോഗ്രാമുകളെ യന്ത്ര ഭാഷയിലേക്ക് തർജ്ജമ ചെയ്യുന്നതിന് അസംബ്ലർ എന്നറിയപ്പെടുന്ന ഒരു വിവർത്തകൻ ആവശ്യമുണ്ട്. പ്രോഗ്രാമിലെ പ്രവൃത്തികൾ നടപ്പിലാക്കുന്നത് തർജ്ജമ ചെയ്യപ്പെട്ട ശേഷമാണ്. കാരണം കമ്പ്യൂട്ടറിന് യന്ത്ര കോഡ് നിർദ്ദേശം മാത്രമെ മനസിലാക്കാൻ കഴിയുകയുള്ളൂ. അസംബ്ലർ യന്ത്രാധിഷ്ഠിതമാണ്.

- **ഇന്റർപ്രറ്റർ (Interpreter)** : ഉയർന്നതലത്തിലെഴുതിയ പ്രോഗ്രാമുകളെ വരിവരിയായി യന്ത്രഭാഷയിലേക്ക് മൊഴിമാറ്റം നടത്തുന്ന ഭാഷ പ്രൊസസ്സറാണ് ഇന്റർപ്രറ്റർ. ഏതെങ്കിലും ഒരു വരിയിൽ തെറ്റുണ്ടെങ്കിൽ, തെറ്റുകൾ വെളിപ്പെടുത്തുകയും പ്രവർത്തനം അവിടെ വെച്ച് അവസാനിപ്പിക്കുകയും ചെയ്യുന്നു. തെറ്റു തിരുത്തിയതിനുശേഷം മാത്രമേ വിവർത്തനം തുടരുകയുള്ളൂ. BASIC ഒരു ഇന്റർപ്രറ്റഡ് ഭാഷയാണ്.

- **കമ്പൈലർ (Compiler):** ഉയർന്നതലത്തിലുള്ള ഭാഷയിലെഴുതിയ പ്രോഗ്രാമുകളെ യന്ത്ര ഭാഷയിലേക്ക് മൊഴിമാറ്റം നടത്തുന്ന ഭാഷ പ്രൊസസ്സറാണ് കമ്പൈലർ. ഒറ്റത്തവണ കൊണ്ടുതന്നെ അത് പ്രോഗ്രാം മുഴുവനായും വ്യാഖ്യാനിക്കുന്നു. ഇതിൽ എന്തെങ്കിലും തെറ്റുകൾ വന്നാൽ ആ തെറ്റുകൾ കമ്പൈലേഷന്റെ അവസാനം ക്രമനമ്പറോടു കൂടി സന്ദേശങ്ങളായി സ്ക്രീനിൽ തെളിയും. വാക്യഘടനയിൽ തെറ്റൊന്നുമില്ലെങ്കിൽ കമ്പൈലർ ഒരു ഒബ്ജക്റ്റ് ഫയൽ സൃഷ്ടിക്കും. കമ്പൈലർ ഉപയോഗിച്ചുള്ള തർജ്ജമയെ കമ്പൈലേഷൻ എന്നു പറയുന്നു. തർജ്ജമക്കുശേഷം പ്രോഗ്രാം റൺ ചെയ്യുന്നതിന് കമ്പൈലർ മെമ്മറിയിൽ ആവശ്യമില്ല. കമ്പൈലർ ഉപയോഗിക്കുന്ന പ്രോഗ്രാമിംഗ് ഭാഷകളാണ് C, C++, Pascal തുടങ്ങിയവ. അസംബ്ലി ഭാഷയിൽ നിന്നും, ഉയർന്നതലത്തിലുള്ള ഭാഷയിൽ നിന്നും പ്രോഗ്രാമുകൾ യന്ത്രഭാഷ പ്രോഗ്രാമുകളായി വിവർത്തനം ചെയ്യുന്ന പ്രവൃത്തികൾ ചിത്രം 2.54 ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 2.54: ഭാഷ പ്രൊസസ്സിംഗ്

c. യൂട്ടിലിറ്റി സോഫ്റ്റ്‌വെയർ (Utility software)

പതിവ് ജോലികളും സിസ്റ്റം പരിപാലന ജോലികളും നിർവ്വഹിക്കാൻ ഉപയോക്താക്കളെ സഹായിക്കുന്ന ഒരുകൂട്ടം പ്രോഗ്രാമുകളാണ് യൂട്ടിലിറ്റി സോഫ്ട്‌വെയർ. ചില യൂട്ടിലിറ്റി പ്രോഗ്രാമുകളും അവയുടെ ധർമ്മങ്ങളും താഴെ കൊടുത്തിരിക്കുന്നു

- **കംപ്രഷൻ ടൂൾസ് (Compression tools):** വലിയ ഫയലുകളെ ചുരുങ്ങിയ സംഭരണ സ്ഥലങ്ങളിലേക്ക് ചുരുക്കുക എന്നതാണ് കംപ്രഷൻ ടൂൾസിലൂടെ ചെയ്യുന്നത്. ആവശ്യാനുസരണം ഒരിക്കൽ കംപ്രഷൻ യൂട്ടിലിറ്റി ഉപയോഗിച്ച് വലിപ്പം കുറച്ച ഫയലുകളെ ഡീകമ്പ്രസ് ചെയ്ത് യഥാർഥ വലിപ്പത്തിലേക്ക് മാറ്റുവാൻ സാധിക്കും. ഫയലുകളുടെ കമ്പ്രഷനെ സിപ്പിംഗ് (Zipping) എന്നും ഡികമ്പ്രഷനെ അൺസിപ്പിംങ് (Unzipping) എന്നും വിളിക്കുന്നു. ഉദാഹരണം Winzip, WinRAR എന്നിവ.
- **ഡിസ്ക് ഡീഫ്രാഗ്മെന്റർ (Disk defragmenter):** കമ്പ്യൂട്ടർ ഹാർഡ് ഡിസ്കിലെ ഫയലുകളെ പുനക്രമീകരണം നടത്തുന്ന പ്രോഗ്രാമുകളെ ഡിസ്ക് ഡീഫ്രാഗ്മെന്റർ എന്നു പറയുന്നു. ഡിസ്കിന്റെ പലഭാഗങ്ങളിലായി ചിതറികിടക്കുന്ന ഫയലുകളെ ക്രമീകരിച്ചാൽ കമ്പ്യൂട്ടറിന് വേഗത കൂടുകയും പ്രവർത്തനം കാര്യക്ഷമമാവുകയും ചെയ്യും.

- **ബാക്ക്അപ്പ് സോഫ്റ്റ്‌വെയർ (Backup software)** : ഏതെങ്കിലും കാരണത്താൽ ഹാർഡ് ഡിസ്ക്കുകൾ പ്രവർത്തനരഹിതമാവുകയോ അബദ്ധത്തിൽ മാറ്റങ്ങൾക്ക് വിധേയമാവുകയോ ചെയ്താൽ ഡിസ്കിൽ സൂക്ഷിച്ചു വച്ചിരിക്കുന്ന വിവരങ്ങളുടെ പകർപ്പ് നമുക്ക് എടുക്കുവാൻ സാധിക്കുന്ന സോഫ്റ്റ്‌വെയറാണിത്. ഈ സൗകര്യം ഉപയോഗിച്ച് ഫയലുകളോ ഫോൾഡറുകളോ ഡ്രൈവുകളോ നമുക്ക് ബാക്ക് അപ്പിന് വിധേയമാക്കാം.
- **ആന്റി വൈറസ് സോഫ്ട്‌വെയർ (Antivirus software):** കമ്പ്യൂട്ടറിന്റെ പ്രവർത്തനത്തെ ദോഷകരമായി ബാധിക്കുന്ന പ്രോഗ്രാമുകളാണ് കമ്പ്യൂട്ടർ വൈറസുകൾ. ആന്റിവൈറസ് സോഫ്റ്റ്‌വെയർ എന്ന യൂട്ടിലിറ്റി പ്രോഗ്രാം ഉപയോഗിച്ച് കമ്പ്യൂട്ടറിനെ ബാധിച്ചിരിക്കുന്ന വൈറസിനെ കണ്ടെത്താനും അവയെ ഒഴിവാക്കാനും സാധിക്കും. പുതിയ വൈറസുകൾ സൃഷ്ടിക്കപ്പെട്ടു കൊണ്ടിരിക്കുന്നതിനനുസരിച്ച് ആന്റി വൈറസ് സോഫ്റ്റ്‌വെയറുകളും പുതുക്കേണ്ടതുണ്ട്. എല്ലാ ആന്റിവൈറസ് സോഫ്റ്റ്‌വെയറുകളിലും സ്വയം പുതുക്കുന്ന സംവിധാനങ്ങളുണ്ട്. നോർട്ടൺ ആന്റിവൈറസ്, കാസ്പേഴ്സ്കി, എ.വി.ജി എന്നിവ ഉദാഹരണങ്ങളാണ്.

2.4.2 ആപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയർ (Application software)

ഒരു പ്രത്യേക ആവശ്യത്തിനായി വികസിപ്പിച്ചെടുക്കുന്ന സോഫ്റ്റ്‌വെയറുകളെയാണ് ആപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയർ എന്നു വിളിക്കുന്നത്. പൊതുവായ ആവശ്യങ്ങൾക്കായുള്ള സോഫ്റ്റ്‌വെയർ പാക്കേജുകളും പ്രത്യേക ആവശ്യങ്ങൾക്കായുള്ള സോഫ്റ്റ്‌വെയറുകളും ഇതിൽപെടുന്നു. ആപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയറുകൾക്ക് ഉദാഹരണങ്ങളാണ് GIMP, Payroll system, Airline Reservation System, Tally എന്നിവ.

a. പൊതുവായ ആവശ്യങ്ങൾക്കുള്ള പാക്കേജുകൾ (General purpose software packages)

ഒരു പ്രത്യേക ആപ്ലിക്കേഷനിലെ പ്രവർത്തനങ്ങൾ നിർവ്വഹിക്കാൻ ഉപയോഗിക്കുന്ന സോഫ്റ്റ്‌വെയർ പാക്കേജാണിത്. ഉപയോക്താവിന്റെ ആവശ്യങ്ങൾക്കനുസരിച്ച് ധാരാളം സവിശേഷതകൾ പ്രത്യേക പാക്കേജുകളിൽ ഉൾപ്പെടുത്തിയിരിക്കുന്നു. വേഡ് പ്രോസസറുകൾ, സ്പ്രെഡ്ഷീറ്റ് സോഫ്റ്റ്‌വെയർ, പ്രസന്റേഷൻ സോഫ്റ്റ്‌വെയർ, ഡാറ്റാബേസ് സോഫ്റ്റ്‌വെയർ, മൾട്ടിമീഡിയ സോഫ്റ്റ്‌വെയർ എന്നിങ്ങനെ ഇവയെ തരംതിരിച്ചിരിക്കുന്നു.

- **വേഡ് പ്രൊസസ്സിങ്ങ് സോഫ്റ്റ്‌വെയർ (Word processing software):** ഡോക്യുമെന്റുകൾ നിർമ്മിക്കുന്നതിനും മാറ്റങ്ങൾ വരുത്തുന്നതിനും വേണ്ടി രൂപകൽപന ചെയ്തിരിക്കുന്ന സോഫ്റ്റ്‌വെയർ ആണ് വേഡ്പ്രൊസസ്സിംഗ്. വളരെ എളുപ്പത്തിൽ ലിഖിത ഉള്ളടക്കം രൂപകൽപന ചെയ്യുവാനും ചിട്ടപ്പെടുത്തുവാനും നിർമിക്കുവാനും പ്രിന്റ് ചെയ്യുവാനും ഇത് ഉപയോഗിക്കുന്നു, ഇതിന്റെ സഹായത്തോടെ ഫോണ്ടുകൾ സെറ്റ് ചെയ്യുവാനും, പാരഗ്രാഫ് സെറ്റ് ചെയ്യുവാനും, വ്യത്യസ്ത രീതിയിൽ അടയാളങ്ങൾ കൊടുക്കുവാനും, വരിയായി നിരത്തൽ ശരിയാക്കുവാനും, വ്യാകരണവും, അക്ഷര വിന്യാസവും പരിശോധിക്കുവാനും, ചിത്രങ്ങൾ ഉൾപ്പെടുത്തുവാനും, രേഖാചിത്രങ്ങളും, പട്ടികകളും നിർമ്മിക്കുവാനും സാധിക്കുന്നു. ഡോക്യുമെന്റിന്റെ ഓരോ പേജുകളിലും തലക്കെട്ടും അടിക്കുറിപ്പും സജ്ജീകരിക്കാനും സാധിക്കുന്നു. MS Word, Open Office Writer, Apple i Work Pages എന്നിവ ഉദാഹരണങ്ങളാണ്.

- **സ്പ്രെഡ്ഷീറ്റ് സോഫ്റ്റ്‌വെയർ (Spreadsheet software):** പട്ടിക രൂപത്തിലുള്ള ഈ സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് കണക്കുകൂട്ടലുകൾ എളുപ്പത്തിൽ നടത്താൻ സാധിക്കും. പേപ്പർ വർക്ക്ഷീറ്റിനെ അനുകരിച്ചുകൊണ്ട് സെല്ലുകൾ കൊണ്ട് ഒരു ഗ്രിഡ് നിർമ്മിക്കുന്നു. ചിത്രങ്ങൾ ഉൾപ്പെടുത്തുവാനും വിവിധതരത്തിലുള്ള ചാർട്ടുകൾ നിർമ്മിക്കുവാനും അത് അനുവദിക്കുന്നു. എംഎസ്എക്സൽ, ഓപ്പൺ ഓഫീസ് കാൽക്, ലോട്ടസ് 1-2-3, ആപ്പിൾ i Work നമ്പേഴ്സ്. (MS Excel, Open Office Calc, Lotus 1-2-3, Apple i Work numbers) എന്നിവ ഉദാഹരണങ്ങളാണ്.
- **പ്രസന്റേഷൻ സോഫ്റ്റ്‌വെയർ (Presentation software):** സ്ലൈഡ് ഷോവിൽ ചലിക്കുന്ന ചിത്രങ്ങളും ശബ്ദങ്ങളും ഉപയോഗിച്ച് വിവരങ്ങൾ തയ്യാറാക്കാൻ ഉപയോഗിക്കുന്ന സോഫ്റ്റ്‌വെയറാണിത്. ചിത്രങ്ങളും, ടെക്സ്റ്റുകളും, ആനിമേഷനും, വീഡിയോകളും, ശബ്ദങ്ങളും ഉൾപ്പെടുത്തി വ്യത്യസ്ത തരത്തിലുള്ള ആശയങ്ങൾ സ്ലൈഡുകളിലൂടെ നിർമ്മിക്കാൻ പ്രസന്റേഷൻ സോഫ്റ്റ്‌വെയർ സഹായിക്കുന്നു. മൈക്രോസോഫ്റ്റ് പവർപോയിന്റ്, ഓപ്പൺ ഓഫീസ് ഇംപ്രസ്, ആപ്പിൾ ഐവർക്ക് കീനോട്ട് (Apple i Work Keynote) എന്നിവ ഉദാഹരണങ്ങളാണ്.
- **ഡാറ്റാബേസ് സോഫ്റ്റ്‌വെയർ (Database software):** പരസ്പരം ബന്ധപ്പെട്ടു കിടക്കുന്ന ഒരുകൂട്ടം ഡാറ്റകളെ ശേഖരിച്ച് പട്ടികാ രീതിയിൽ സൂക്ഷിച്ചിരിക്കുന്നവയെയാണ് ഡാറ്റാബേസുകൾ എന്നുപറയുന്നത്. പരസ്പരം ബന്ധപ്പെട്ടു കിടക്കുന്ന ഡാറ്റയും അവ സ്വീകരിക്കാനുള്ള ഒരുകൂട്ടം പ്രോഗ്രാമുകളും ചേർന്നതാണ് ഡാറ്റാബേസ് മാനേജ്മെന്റ് സിസ്റ്റം (DBMS). ഡാറ്റാബേസിലുള്ള വിവരങ്ങൾ അതിനനുയോജ്യമായ രീതിയിലും കാര്യക്ഷമമായും വിനിയോഗിച്ച് അവ വേണ്ട രീതിയിൽ സൂക്ഷിക്കുകയും തിരിച്ചെടുക്കുകയും ചെയ്യലാണ് ഡാറ്റാബേസിന്റെ പ്രധാന ലക്ഷ്യം. സുരക്ഷിതത്വവും സ്വകാര്യതയും, പ്രത്യേക മാനദണ്ഡങ്ങളും ഇവ നൽകുന്ന പ്രത്യേകതകളാണ്. മൈക്രോസോഫ്റ്റ് ആക്സസ് (Microsoft access), ഒറാക്കിൾ (Oracle), പോസ്റ്റ്ഗ്രസ് എസ്.ക്യൂ.എൽ (Postgres SQL), മൈ എസ്.ക്യൂ.എൽ (My SQL) എന്നിവ ഉദാഹരണങ്ങളാണ്.
- **മൾട്ടിമീഡിയ സോഫ്റ്റ്‌വെയർ (Multimedia software):** വിവിധ രൂപങ്ങളിലുള്ള മാധ്യമങ്ങളുടെ ഏകീകൃത രൂപമാണ് മൾട്ടിമീഡിയ. അക്ഷരങ്ങളും, ചിത്രങ്ങളും, ഗ്രാഫിക്സുകളും, ഓഡിയോകളും, വീഡിയോകളും കൂടി ചേർന്നുള്ള രൂപമാണ് മൾട്ടിമീഡിയ. വിവിധതരത്തിലുള്ള വിവരങ്ങൾ പ്രൊസസ്സ് ചെയ്യാൻ മൾട്ടിമീഡിയ സോഫ്റ്റ്‌വെയറിൽ സാധിക്കും. ഓഡിയോ, വീഡിയോ ഫയലുകൾ നിർമ്മിക്കുന്നതിനും എഡിറ്റ് ചെയ്യുന്നതിനും ഇത് സഹായിക്കുന്നു. ഒരു രൂപത്തിൽ നിന്നും വേറൊരു രൂപത്തിലേക്ക് തർജ്ജമ ചെയ്യുവാനുള്ള (ഓഡിയോ വീഡിയോ ഫയലുകൾ) സോഫ്റ്റ്‌വെയറുകൾ ഉണ്ട്. വി.എൽ.സി. പ്ലയർ (VLC Player), അഡോബ് ഫ്ളാഷ് (Adobe Flash), റിയൽ പ്ലയർ (Real Player), മീഡിയ പ്ലയർ (Media Player) എന്നിവ ഉദാഹരണങ്ങളാണ്.

b. പ്രത്യേക ആവശ്യങ്ങൾക്കുള്ള സോഫ്റ്റ്‌വെയർ (Specific Purpose Software)

പ്രത്യേക ആവശ്യങ്ങൾക്കു മാത്രമായി തയ്യാറാക്കുന്ന സോഫ്റ്റ്‌വെയറുകളാണിത്. ഒരു സ്ഥാപനത്തിനുവേണ്ടി പ്രത്യേകം തയ്യാറാക്കിയിരിക്കുന്ന ഈ സോഫ്റ്റ്‌വെയറിനെ ടെയ്‌ലർ - മെയ്ഡ് സോഫ്റ്റ്‌വെയർ എന്നു വിളിക്കുന്നു. സാമ്പ്രദായികമായി ചിട്ടപ്പെടു

ത്തിയ സോഫ്റ്റ്‌വെയർ എന്നും ഇതിനെ പറയാറുണ്ട്. ഒരു ഉപയോക്താവിനെ മാത്രം ഉദ്ദേശിച്ച് അയാളുടെ മുൻഗണനകളും പ്രതീക്ഷകളും മാത്രം ഉൾപ്പെടുത്തി തയാറാക്കുന്ന സോഫ്റ്റ്‌വെയറാണ് സാമ്പ്രദായികമായി ചിട്ടപ്പെടുത്തിയ സോഫ്റ്റ്‌വെയർ. പട്ടിക 2.5 ൽ പ്രത്യേക ആവശ്യങ്ങൾക്കായുള്ള ആപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയർ വിശദീകരിച്ചിരിക്കുന്നു.

അപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയർ	ഉദ്ദേശ്യങ്ങൾ
പേ റോൾ സിസ്റ്റം	ഒരു സ്ഥാപനത്തിലെ തൊഴിലാളികളുടെ വേതനത്തെ കുറിച്ചും മറ്റ് എല്ലാവിധ വിവരങ്ങളെ കുറിച്ചും വിശദമായി പരിപാലിക്കുന്നത് പേ റോൾ സിസ്റ്റമാണ്.
ഇൻവെന്ററി മാനേജ്മെന്റ് സിസ്റ്റം	ഒരു വ്യാപാര സ്ഥാപനത്തിലെ ആസ്തി വിവര പട്ടികകൾ, ഓർഡറുകൾ, വിപണനം, വിതരണം എന്നിവയെ വേണ്ടവിധം പരിപാലിക്കുന്നു.
ഹ്യൂമൺ റിസോഴ്സ് മാനേജ്മെന്റ് സിസ്റ്റം	ഒരു സ്ഥാപനത്തിലെ മനുഷ്യവിഭവങ്ങളെ വേണ്ട വിധം പരിപാലിക്കുന്നു.

പട്ടിക. 2.5: ആപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയറിന് ഉദാഹരണം

സ്വയം വിലയിരുത്താം

1. ഓപ്പറേറ്റിങ്ങ് സിസ്റ്റം നിർവചിക്കുക
2. ഓപ്പറേറ്റിങ്ങ് സിസ്റ്റത്തിന് 2 ഉദാഹരണം കൊടുക്കുക.
3. ഒരു പ്രോഗ്രാം കൃത്യനിർവ്വഹണത്തിന് പറയുന്ന പേരാണ്________.
4. ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന്റെ ഏതെങ്കിലും രണ്ട് ധർമ്മങ്ങൾ എഴുതുക.
5. അസംബ്ലി ഭാഷയെ യന്ത്രഭാഷയിലേക്ക് തർജ്ജമ ചെയ്യുന്ന സോഫ്റ്റ്‌വെയറിന്റെ പേരെഴുതുക
6. കമ്പൈലറും ഇന്റർപ്രറ്ററും തമ്മിലുള്ള വ്യത്യാസം എഴുതുക.
7. DBMS എന്നാൽ ______________ആണ്.
8. സാമ്പ്രദായികമായി ചിട്ടപ്പെടുത്തിയ സോഫ്റ്റ്‌വെയറിന് രണ്ട് ഉദാഹരണം എഴുതുക.
9. ഡിസ്കിലെ വിവരങ്ങളുടെ തനിപ്പകർപ്പിനെ വിളിക്കുന്ന പേര് ________.

2.5 സ്വതന്ത്ര ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയർ (Free and open source software)

ഉപയോഗിക്കുന്നതിനും, പകർപ്പ് എടുക്കുന്നതിനും, വിതരണം ചെയ്യുന്നതിനും, പരിശോധിക്കുന്നതിനും, മാറ്റങ്ങൾ വരുത്തുന്നതിനും, മെച്ചപ്പെടുത്തുന്നതിനും ഉപയോക്താവിന് സ്വാതന്ത്ര്യം നൽകുന്ന സോഫ്റ്റ്‌വെയറാണ് സ്വതന്ത്ര ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെ

യർ. കുറഞ്ഞ ചിലവ്, സുരക്ഷിതത്വം, കുത്തക കമ്പനികളിൽ നിന്നുള്ള സ്വാതന്ത്ര്യം, കാര്യക്ഷമമായ പ്രവർത്തനം, പരസ്പര പ്രവർത്തനക്ഷമത തുടങ്ങിയവ നൽകുന്നത് കൊണ്ട് സ്വതന്ത്ര ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയർ ഇപ്പോൾ വ്യാപകമായി ഉപയോഗിക്കുന്നു.

നാല് തരത്തിലുള്ള സ്വാതന്ത്ര്യം സ്വതന്ത്ര സോഫ്റ്റ്‌വെയർ ഫൗണ്ടേഷൻ (Free Software Foundation - FSF) നിർവചിക്കുന്നു.

സ്വാതന്ത്ര്യം 0 (Freedom 0) : ഏത് ആവശ്യത്തിനും ഇഷ്ടപ്രകാരം ഉപയോഗിക്കുന്നതിനുള്ള സ്വാതന്ത്ര്യം.

സ്വാതന്ത്ര്യം 1 (Freedom 1) : സോഫ്റ്റ്‌വെയർ എങ്ങനെ പ്രവർത്തിക്കുന്നു എന്ന് വിശകലനം ചെയ്യുന്നതിനുള്ള സ്വാതന്ത്ര്യം.

സ്വാതന്ത്ര്യം 2 (Freedom 2) : പ്രോഗ്രാമിന്റെ പകർപ്പുകൾ പുനർവിതരണം ചെയ്യുവാനുള്ള സ്വാതന്ത്ര്യം.

സ്വാതന്ത്ര്യം 3 (Freedom 3) : പ്രോഗ്രാമിനെ നവീകരിക്കാനും നവീകരിച്ചവയെ പുറത്തിറക്കാനുമുള്ള സ്വാതന്ത്ര്യം.

ഫ്രീ ആന്റ് ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയറുകളിൽ ചിലത് താഴെ കൊടുക്കുന്നു

ഗ്നു/ലിനക്സ് (GNU/Linux): സ്വതന്ത്ര ഓപ്പൺ സോഫ്റ്റ്‌വെയർ മാതൃകയിലുള്ള ഓപ്പറേറ്റിംഗ് സിസ്റ്റം വിഭാഗത്തിൽപ്പെട്ട സോഫ്റ്റ്‌വെയറാണ് GNU/Linux. 1983 ൽ FSF ൽ റിച്ചാർഡ് സ്റ്റാൾമാൻ അവതരിപ്പിച്ച GNU പ്രൊജക്ടിൽ ചിട്ടപ്പെടുത്തിയ ഒന്നാണ് ഇത്.

ജിമ്പ് (GIMP): GNU ഇമേജ് മാനിപുലേഷൻ പ്രോഗ്രാം എന്നാണ് ഇത് അറിയപ്പെടുന്നത്. ചിത്രങ്ങളെ ചിട്ടപ്പെടുത്തുവാൻ ഉപയോഗിക്കുന്ന സോഫ്റ്റ്‌വെയറാണിത്. ചിത്രങ്ങൾ നിർമ്മിക്കുവാനും, ചിട്ടപ്പെടുത്തുവാനും വേണ്ട രീതിയിൽ കൈകാര്യം ചെയ്യുവാനും ഈ സോഫ്റ്റ്‌വെയർ ഉപയോഗിക്കുന്നു. വിവിധ ഫയൽ ഫോർമാറ്റുകളെ പിന്തുണക്കാനും ഒരു രൂപത്തിൽ നിന്ന് മറ്റൊന്നിലേക്ക് മാറ്റുന്നതിനും GIMP സഹായിക്കുന്നു.

മോസില്ല ഫയർഫോക്സ് (Mozilla Firefox) : മോസില്ല കോർപ്പറേഷൻ നിർമ്മിച്ച വളരെ പ്രശസ്തമായ വെബ് ബ്രൗസർ ആണിത്. സുരക്ഷിതമായ ബ്രൗസിങ്ങിന് ഇത് അനുയോജ്യമാണ്.

ഓപ്പൺ ഓഫീസ്.ഒആർജി (Open offfice .org) : ഒരു സമ്പൂർണ്ണ ഓഫീസ് പാക്കേജ് ആണ് ഓപ്പൺ ഓഫീസ്. ഇതിൽ ലിഖിത ഉള്ളടക്കങ്ങൾ തയ്യാറാക്കാനും രൂപമാറ്റം ചെയ്യുവാനും വേഡ് പ്രൊസസ്സറായ റൈറ്റർ, സ്പ്രെഡ്ഷീറ്റായ കാൽക്ക്, പ്രസന്റേഷൻ സോഫ്റ്റ്‌വെയറായ ഇംപ്രസ്സ് എന്നിവ ഉപയോഗിക്കുന്നു. ഇത് ലിനക്സ്, വിൻഡോസ് സംവിധാനങ്ങളിൽ പ്രവർത്തിക്കുന്നു.

2.6 ഫ്രീവെയറും ഷെയർവെയറും (Freeware and shareware)

പ്രത്യേകമായ വിലയൊന്നും കൂടാതെ, പരിധിയൊന്നുമില്ലാതെ ഉപയോഗിക്കാൻ സാധിക്കുന്ന പകർപ്പവകാശം ഉള്ള കമ്പ്യൂട്ടർ സോഫ്റ്റ്‌വെയർ ആണ് ഫ്രീവെയർ.

ഒരു ചുരുങ്ങിയ കാലയളവിലേക്ക് പരീക്ഷണ അടിസ്ഥാനത്തിൽ വിതരണം ചെയ്യപ്പെടുന്ന വാണിജ്യ പരമായ സോഫ്റ്റ്‌വെയർ ആണ് ഷെയർവെയർ. വിലയൊന്നും കൂടാതെ വളരെ പരിമിതമായ പ്രവർത്തനക്ഷമതയോടെ വിതരണം ചെയ്യപ്പെടുന്നതാണിത്. ഇന്റർനെറ്റിൽ നിന്ന് ഡൗൺലോഡ് ചെയ്യാൻ സാധിക്കുന്ന വിധത്തിലാണ് ഷെയർവെയറുകൾ ലഭ്യമാകുന്നത്. വിലകൊടുത്ത് വാങ്ങുന്നതിന് മുമ്പ് ഉപയോക്താക്കൾക്ക് ഇവയെ വിലയിരുത്തുവാനുള്ള അവസരം നൽകുന്നു എന്നതാണ് ഇതിന്റെ ലക്ഷ്യം. ചില ഷെയർവെയറുകൾ പരിമിതമായ കാലയളവിലെക്ക് മാത്രം പ്രവർത്തിപ്പിക്കുന്നവയാണ്. പട്ടിക 2.6 ൽ ഫ്രീവെയറിന്റെയും ഷെയർവെയറിന്റെയും താരതമ്യം കൊടുത്തിരിക്കുന്നു

ഫ്രീവെയർ	ഷെയർവെയർ
• ഇന്റർനെറ്റിൽ നിന്നും ആർക്കു വേണമെങ്കിലും സൗജന്യമായി ഡൗൺലോഡ് ചെയ്യുകയും ഉപയോഗിക്കുകയും ചെയ്യാം.	• വാങ്ങുന്നതിനും മുമ്പ് തന്നെ സോഫ്റ്റ്‌വെയറിനെ പരിചയപ്പെടാൻ സാധിക്കുന്നു.
• എല്ലാ സവിശേഷതകളും സൗജന്യമാണ്.	• എല്ലാ സവിശേഷതകളും ലഭ്യമാവുകയില്ല. എല്ലാ സവിശേഷതകളും ലഭിക്കണമെങ്കിൽ വില കൊടുത്തു വാങ്ങിക്കണം.
• ഫ്രീവെയർ പ്രോഗ്രാമുകൾ വിലയില്ലാതെയാണ് വിതരണം ചെയ്യുന്നത്.	• ഷെയർവെയർ വില കൊടുത്തും അല്ലാതെയും വിതരണം ചെയ്യുന്നു. പല സന്ദർഭങ്ങളിലും നിർമ്മിച്ച ആളുടെ അനുമതിയോടെ മാത്രമേ ഷെയർവെയർ വിതരണം ചെയ്യുകയുള്ളൂ.

പട്ടിക 2.6 : ഫ്രീവെയറും ഷെയർവെയറും തമ്മിലുള്ള താരതമ്യം

നമുക്കു ചെയ്യാം

- *"ഫ്രീ സോഫ്റ്റ്‌വെയറും ഉടമസ്ഥാവകാശമുള്ള സോഫ്റ്റ്‌വെയറും" എന്ന വിഷയത്തെ ആസ്പദമാക്കി ഒരു ചർച്ച സംഘടിപ്പിക്കുക*
- *ഫ്രീ സോഫ്റ്റ് വെയറുകൾ നൽകുന്ന നാലു സ്വാതന്ത്ര്യങ്ങൾ ചർച്ച ചെയ്യുക.*

2.7 ഉടമസ്ഥാവകാശമുള്ള സോഫ്റ്റ്‌വെയർ (Proprietary software)

സോഫ്റ്റ്‌വെയർ നിർമ്മാതാവ് അഥവാ പ്രസാധകന്റെ പൂർണ്ണ അധികാരപരിധിയിൽ വരുന്ന സോഫ്റ്റ്‌വെയറാണിത്. അനുവാദ ഉടമ്പടി ഇല്ലാതെ ഇത് പകർത്താനോ വിതരണം ചെയ്യുവാനോ പാടുള്ളതല്ല. പ്രോഗ്രാമിന്റെ സോഴ്സ് കോഡ് ലഭ്യമാകാത്തതിനാൽ ഇതിൽ മാറ്റം വരുത്തി മെച്ചപ്പെടുത്തുവാനൊന്നും ഉപയോക്താവിന് സാധിക്കുകയില്ല. മൈക്രോസോഫ്റ്റ് വിൻഡോസ് ഓപ്പറേറ്റിംഗ് സിസ്റ്റം, എംഎസ് ഓഫീസ്, മാക് ഒഎസ് എന്നിവ ഉദാഹരണങ്ങളാണ്.

2.8 ഹ്യൂമൻവെയർ/ലൈവ് വെയർ (Humanware or Liveware)

കമ്പ്യൂട്ടർ ഉപയോഗിക്കുന്ന ആളുകളെയാണ് ഹ്യൂമൻവെയർ/ലൈവ്വെയർ എന്നുപറയുന്നത്. പ്രോഗ്രാമർ, സിസ്റ്റം അനലിസ്റ്റ്, ഓപ്പറേറ്റിംങ്ങ് സ്റ്റാഫ്, കമ്പ്യൂട്ടർ സിസ്റ്റം ഉപയോഗിക്കുന്നവർ എന്നിവരെല്ലാം ഇതിൽ ഉൾപ്പെടുന്നു.

പട്ടിക 2.7 ൽ വിവിധതരം ഹ്യൂമൻവെയറുകളും അവരുടെ ജോലികളും വിശദീകരിക്കുന്നു.

ഹ്യൂമൺവെയർ	ജോലികൾ
സിസ്റ്റം അഡ്മിനിസ്ട്രേറ്റർ	കമ്പ്യൂട്ടർ സിസ്റ്റത്തെയും, സെർവറിനെയും പരിപാലിക്കുക. ക്രമീകരണം നടത്തുക, വിശ്വസനീയമായ പ്രവർത്തനങ്ങൾ നിർവ്വഹിക്കുക. പ്രത്യേകിച്ച് സെർവറുകളെയും ഒന്നിൽ കൂടുതൽ ഉപയോക്താക്കളുള്ള കമ്പ്യൂട്ടറുകളെയും പരിപാലിക്കുക.
സിസ്റ്റം മാനേജർ	ഉപഭോക്തൃ സേവനങ്ങൾ ഉത്തമമായി ഉറപ്പുവരുത്തുകയും വ്യാപാര സംവിധാനങ്ങളിലെ വൈദഗ്ദ്ധ്യം നിലനിർത്തി പോരുകയും വിൽപനക്കാരും കരാറുകാരും പോലെയുള്ളവരുമായുള്ള തൊഴിൽപരമായ ബന്ധം വളർത്തികൊണ്ടു വരികയും ചെയ്യുക.
സിസ്റ്റം അനലിസ്റ്റ്	പുതിയ ഐ.ടി. പരിഹാരങ്ങൾ രൂപകൽപന ചെയ്ത് വ്യാപാരത്തിന്റെ കാര്യക്ഷമതയും ഉൽപാദനക്ഷമതയും മെച്ചപ്പെടുത്തുക.
ഡാറ്റാബേസ് അഡ്മിനിസ്ട്രേറ്റർ	ഡാറ്റാബേസ് പരിഹാരങ്ങൾ രൂപകല്പന ചെയ്യുകയും നിരീക്ഷിക്കുകയും, അപഗ്രഥിക്കുകയും, നടപ്പിൽ വരുത്തുകയും ചെയ്യുക.
കമ്പ്യൂട്ടർ എഞ്ചിനീയർ	കമ്പ്യൂട്ടർ സിസ്റ്റത്തിലെ ഹാർഡ്വെയറിന്റെയും സോഫ്റ്റ്വെയറിന്റെയും രൂപകൽപന നടത്തുക.
കമ്പ്യൂട്ടർ പ്രോഗ്രാമർ	കമ്പ്യൂട്ടറുകളെ ശരിയായ രീതിയിൽ പ്രവർത്തിപ്പിക്കുവാൻ ആവശ്യമായ കോഡുകൾ എഴുതുന്നു.
കമ്പ്യൂട്ടർ ഓപ്പറേറ്റർ	കമ്പ്യൂട്ടർ സിസ്റ്റത്തിന്റെ മേൽനോട്ടം നിർവ്വഹിക്കുന്നു. ഇവ വേണ്ട വിധം പ്രവർത്തിക്കുന്നു എന്ന് ഉറപ്പുവരുത്തുന്നു. ഭൗതിക സുരക്ഷിതത്വം ഉറപ്പാക്കുന്നു. തെറ്റുകൾ വരുന്ന സാഹചര്യം ഒഴിവാക്കുന്നു.

പട്ടിക 2.7 : വിവിധതരം ഹ്യൂമൻവെയറുകളും അവരുടെ ജോലിവിവരങ്ങളും

സ്വയം വിലയിരുത്താം

1. സ്വതന്ത്ര ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയറിന് ഒരു ഉദാഹരണം നൽകുക.
2. വാങ്ങുന്നതിനുമുമ്പ് ഉപഭോക്താവിന് ഉപയോഗിക്കാനുള്ള സൗകര്യം ഒരുക്കുന്ന സോഫ്റ്റ്‌വെയർ ആണ് ______________
3. ഫ്രീ ആന്റ് ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയർ എന്നാലെന്ത്?
4. ഉടമസ്ഥാവകാശമുള്ള സോഫ്റ്റ്‌വെയറിന് ഉദാഹരണം നൽകുക.
5. ഹ്യൂമൻവെയറിന് രണ്ട് ഉദാഹരണം എഴുതുക.

നമുക്ക് സംഗ്രഹിക്കാം

ഒരു കമ്പ്യൂട്ടർ സംവിധാനത്തിൽ ഹാർഡ്‌വെയറും സോഫ്റ്റ്‌വെയറുമുണ്ട്. കമ്പ്യൂട്ടറിന്റെ ദൃശ്യവും സ്പർശനീയവുമായ ഭാഗങ്ങളാണ് ഹാർഡ്‌വെയർ. പ്രൊസസർ, മദർബോർഡ്, പെരിഫറലുകൾ, പോർട്ടുകൾ, മെമ്മറി ഉപകരണങ്ങൾ, ഇൻപുട്ട് / ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ മുതലായവ കമ്പ്യൂട്ടറിന്റെ ഹാർഡ്‌വെയർ ഘടകങ്ങളാണ്. പ്രാഥമികം, ദ്വിതീയം എന്നിങ്ങനെ രണ്ടായി മെമ്മറി തരം തിരിച്ചിരിക്കുന്നു. പ്രാഥമിക മെമ്മറിയിൽ RAM, ROM, കാഷ് എന്നിവ അടങ്ങിയിരിക്കുന്നു. ദ്വിതീയ സംഭരണ ഉപകരണങ്ങളെ കാന്തിക സംഭരണം, ഒപ്റ്റിക്കൽ സംഭരണം, അർദ്ധചാലക സംഭരണം എന്നിങ്ങനെ വേർതിരിച്ചിരിക്കുന്നു. കീബോർഡ്, മൗസ്, ലൈറ്റ് പെൻ, ടച്ച് സ്ക്രീൻ, ഗ്രാഫിക് ടാബ്‌ലെറ്റ്, ടച്ച്പാഡ്, ജോയ്സ്റ്റിക്, മൈക്രോഫോൺ, സ്കാനർ, OMR, ബാർകോഡ് റീഡർ, ബയോമെട്രിക്ക് സെൻസർ, സ്മാർട്ട് കാർഡ് റീഡർ, ഡിജിറ്റൽ ക്യാമറ മുതലായവ വിവിധ തരം ഇൻപുട്ട് ഉപകരണങ്ങളാണ്. വിഷ്വൽ ഡിസ്പ്ലെ യൂണിറ്റുകൾ, പ്രിന്ററുകൾ, പ്ലോട്ടറുകൾ, ഓഡിയോ ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ തുടങ്ങിയവ ഔട്ട്പുട്ട് ഉപകരണങ്ങളാണ്. പുനരുപയോഗം, പുനർവിൽപന, വീണ്ടെടുക്കൽ, പുനരുൽപാദനം അല്ലെങ്കിൽ ഒഴിവാക്കൽ എന്നിവയ്ക്കായി മാറ്റിവെക്കുന്ന ഉപയോഗിച്ച ഇലക്ട്രോണിക് ഉൽപന്നങ്ങളെ ഇവേസ്റ്റ് എന്ന് വിളിക്കുന്നു. പുനരുപയോഗം, ദഹനം, പുനരുൽപാദനം, മണ്ണിട്ട് മൂടൽ തുടങ്ങിയവയാണ് ഇവേസ്റ്റ് നിർമ്മാർജ്ജനം ചെയ്യുന്നതിനുള്ള വിവിധ മാർഗ്ഗങ്ങൾ. ഹരിത കമ്പ്യൂട്ടിംഗ് എന്നത് സുസ്ഥിര കമ്പ്യൂട്ടിംഗിന്റെയും വിവരസാങ്കേതികവിദ്യയുടെയും പരിസ്ഥിതി സൗഹൃദപരമായ ഉപയോഗമാണ്.

കമ്പ്യൂട്ടർ, മറ്റ് ഇലക്ട്രോണിക് ഉപകരണങ്ങൾ എന്നിവ കാര്യക്ഷമമായും ഫലപ്രദമായും ഉപയോഗിക്കാൻ നമ്മെ സഹായിക്കുന്ന അദൃശ്യവും അസ്പർശനീയവുമായ ഒരു കൂട്ടം പ്രോഗ്രാമുകളാണ് സോഫ്റ്റ്‌വെയർ. സോഫ്റ്റ്‌വെയറിനെ സിസ്റ്റം സോഫ്റ്റ്‌വെയർ, ആപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയർ എന്നിങ്ങനെ രണ്ടായി തരം തിരിച്ചിരിക്കുന്നു. ഓപ്പറേറ്റിംഗ് സിസ്റ്റം, ഭാഷ പ്രോസസ്സറുകൾ, യൂട്ടിലിറ്റി സോഫ്റ്റ്‌വെയർ എന്നിവയാണ് സിസ്റ്റം സോഫ്റ്റ്‌വെയറിൽ ഉൾപ്പെടുന്നത്. പൊതു ആവശ്യങ്ങൾക്കുള്ള സോഫ്റ്റ്‌വെയർ, പ്രത്യേക ആവശ്യങ്ങൾക്കുള്ള സോഫ്റ്റ്‌വെയർ എന്നിങ്ങനെ ആപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയർ വേർതിരിച്ചിരിക്കുന്നു. സോഫ്റ്റ്‌വെയർ ഉപയോഗിക്കാനും പകർത്താനും, വിതരണം ചെയ്യാനും, പരിശോധിക്കാനും, മാറ്റം വരുത്താനും, മെച്ചപ്പെടുത്താനുമുള്ള സ്വാതന്ത്ര്യം സ്വതന്ത്ര സോഫ്റ്റ്‌വെയർ നൽകുന്നു. ഇന്റർനെറ്റിൽ നിന്നും ഡൗൺലോഡ് ചെയ്ത് സൗജന്യമായി ഉപയോഗിക്കാവുന്ന സോഫ്റ്റ്‌വെയറാണ് ഫ്രീവെയർ. ഷെയർവെയർ ഉപയോക്താക്കൾക്ക് സോഫ്റ്റ്‌വെയർ വാങ്ങുന്നതിനുമുമ്പ് അത് പരീക്ഷിക്കാൻ അവസരം നൽകുന്നു. കമ്പ്യൂട്ടറുമായി ബന്ധപ്പെട്ട മറ്റൊരു ഘടകമാണ് ഹ്യൂമൻവെയർ. വിവിധ തലങ്ങളിൽ കമ്പ്യൂട്ടർ ഉപയോഗിക്കുന്ന മനുഷ്യരെ ഇത് സൂചിപ്പിക്കുന്നു.

പഠന നേട്ടങ്ങൾ

ഈ അധ്യായം പൂർത്തത്തിയാക്കിയ പഠിതാവ്

- പലതരത്തിലുള്ള ഇൻപുട്ട് ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ തിരിച്ചറിയും
- സിസ്റ്റം സോഫ്റ്റ്‌വെയറും ആപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയറും വേർതിരിക്കും
- ഇ - വേസ്റ്റ് നിർമാർജനത്തിന്റെ പ്രാധാന്യം തിരിച്ചറിയും
- ഹരിത കമ്പ്യൂട്ടിങ്ങിന്റെ ആശയം തിരിച്ചറിയും
- പലതരത്തിലുള്ള സോഫ്റ്റ്‌വെയറുകളെ തരം തിരിക്കും
- ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന്റെ ധർമ്മങ്ങൾ തിരിച്ചറിയും
- വേഡ് പ്രൊസസ്സറും, ഇലക്ട്രോണിക് സ്പ്രെഡ്ഷീറ്റും, പ്രസന്റേഷൻ സോഫ്റ്റ്‌വെയറും ഉപയോഗിക്കും
- വിവിധതരത്തിലുള്ള കമ്പ്യൂട്ടർ ഭാഷകൾ തരംതിരിക്കും
- വ്യത്യസ്ത തരത്തിലുള്ള യൂട്ടിലിറ്റി സോഫ്റ്റ്‌വെയറുകൾ പട്ടികപ്പെടുത്തും
- ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയർ പ്രോത്സാഹിപ്പിക്കും
- ഹ്യൂമൺവെയറും ഷെയർവെയറും വിശദീകരിക്കും

മാതൃക ചോദ്യങ്ങൾ

ഹ്രസ്വോത്തര ചോദ്യങ്ങൾ

1. വിവിധതരത്തിലുള്ള പ്രധാന മെമ്മറികൾ ഏതെല്ലാം?
2. EPROM ന് മീതെ EEPROM ന്റെ മേന്മകൾ എന്തെല്ലാം?
3. എപ്പോഴാണ് നമ്മൾ റോം (ROM) ഉപയോഗിക്കുന്നത്?
4. ഇൻപുട്ട് ഉപകരണം എന്നാൽ എന്ത്? സാധാരണയായി ഉപയോഗിക്കുന്ന ഇൻപുട്ട് ഉപകരണങ്ങൾ പട്ടികപ്പെടുത്തുക.
5. ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ എന്നാൽ എന്ത്? സാധാരണയായി ഉപയോഗിക്കുന്ന ഔട്ട്പുട്ട് ഉപകരണങ്ങൾ പട്ടികപ്പെടുത്തുക.
6. സംഭരണ ഉപകരണങ്ങൾ എന്താണ്? സാധാരണ ഉപയോഗിക്കുന്ന സംഭരണ ഉപകരണങ്ങൾ പട്ടികപ്പെടുത്തുക.
7. രജിസ്റ്റേഴ്സ് എന്താണ്? ഏതെങ്കിലും രണ്ട് എണ്ണം എഴുതുകയും വിശദീകരിക്കുകയും ചെയ്യുക
8. ഹാർഡ്കോപ്പിയും സോഫ്റ്റ്കോപ്പിയും താരതമ്യം ചെയ്യുക
9. ഇ-വേസ്റ്റ് എന്നാൽ എന്താണ്?

10. ഓപ്പറേറ്റിംങ്ങ് സിസ്റ്റം എന്നാൽ എന്താണ്?
11. ഭാഷ പ്രൊസസ്സർ എന്താണ്?
12. കമ്പ്യൂട്ടർ ഭാഷകളെ തരംതിരിക്കുക?
13. ഡിസ്ക് ഡിഫ്രാഗ്മെന്റർ എന്താണ്?
14. ഉടമസ്ഥാവകാശമുള്ള സോഫ്റ്റ്‌വെയർ എന്താണ്?
15. ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയർ കൊണ്ടുദ്ദേശിക്കുന്നതെന്താണ്?

ലഘു ഉപന്യാസ ചോദ്യങ്ങൾ

1. ഏതെങ്കിലും മൂന്ന് ഇൻപുട്ട് ഉപകരണങ്ങളെക്കുറിച്ച് ചുരുക്കി വിശദീകരിക്കുക.
2. സി.ആർ.ടി. (CRT) മോണിറ്ററും എൽ.ഇ.ഡി. (LED) മോണിറ്ററും താരതമ്യം ചെയ്യുക
3. RAM, ROM എന്നിവ തമ്മിലുള്ള വ്യത്യാസം എഴുതുക
4. ഇ-വേസ്റ്റ് നിർമാർജനം പട്ടികപ്പെടുത്തി വിശദീകരിക്കുക
5. ഹരിത കമ്പ്യൂട്ടിംഗ് നടപ്പിൽ വരുത്തുവാൻ ആവശ്യമായ ഘട്ടങ്ങളെക്കുറിച്ച് സംഗ്രഹിക്കുക
6. സാമ്പ്രദായികമായി ചിട്ടപ്പെടുത്തിയ സോഫ്റ്റ്‌വെയർ കൊണ്ടുദ്ദേശിക്കുന്നതെന്താണ്? ഉദാഹരണം നൽകുക.
7. താഴ്ന്നതലത്തിലുള്ള ഭാഷകളും ഉയർന്നതലത്തിലുള്ള ഭാഷകളും വേർതിരിച്ചെഴുതുക.
8. കംപൈലർ, ഇന്റർപ്രിറ്റർ എന്നിവ തമ്മിലുള്ള വ്യത്യാസം എഴുതുക.
9. ഇലക്ട്രോണിക് സ്പ്രെഡ്ഷീറ്റിന്റെ ഉപയോഗത്തെക്കുറിച്ച് വിശദീകരിക്കുക
10. യൂട്ടിലിറ്റി സോഫ്റ്റ്‌വെയർ എന്താണ്? രണ്ട് ഉദാഹരണങ്ങൾ നൽകുക.
11. താഴെ കൊടുത്തിരിക്കുന്ന സോഫ്റ്റ്‌വെയറുകളെ ഓപ്പറേറ്റിംഗ് സിസ്റ്റം, ആപ്ലിക്കേഷൻ പാക്കേജ്, യൂട്ടിലിറ്റി പ്രോഗ്രാമുകൾ എന്നിങ്ങനെ തരംതിരിക്കുക

    Linux, Tally, Winzip, MS-Word, Windows, MS-Excel
12. ഫ്രീവെയറും ഷെയർവെയറും തമ്മിലുള്ള വ്യത്യാസം എഴുതുക.
13. സോഫ്റ്റ്‌വെയറുകൾ സ്വതന്ത്രമായും ഓപ്പൺ സോഴ്സായും നിർമ്മിക്കുമ്പോൾ ഉണ്ടായിരിക്കേണ്ട നാല് സ്വാതന്ത്ര്യങ്ങൾ ഏതെല്ലാമാണ്.
14. ഹ്യൂമൻവെയർ കൊണ്ടുദ്ദേശിക്കുന്നതെന്താണ്? ഏതെങ്കിലും രണ്ട് ഉദാഹരണങ്ങൾ നൽകുക.

ഉപന്യാസ ചോദ്യങ്ങൾ

1. വിവിധതരത്തിലുള്ള മെമ്മറിയെ കുറിച്ച് ചുരുക്കി വിവരിക്കുക.

2. പ്രിന്ററിന്റെ വർഗ്ഗീകരണത്തെക്കുറിച്ച് വിശദീകരിക്കുക.
3. നമ്മുടെ ആരോഗ്യത്തിനും പരിസ്ഥിതിക്കും ഇ-വേസ്റ്റ് ആപൽക്കരമാണ്. പ്രസ്താവന ന്യായീകരിക്കുക. ഇ-വേസ്റ്റ് നിർമാർജനത്തിന് പൊതുവായി ഉപയോഗിക്കുന്ന രീതികൾ പട്ടികപ്പെടുത്തി വിശദമാക്കുക.
4. ഹരിത കമ്പ്യൂട്ടിംഗ് നിർവചിക്കുക. നിങ്ങൾക്ക് സാധ്യമാകുന്ന രീതിയിൽ ഹരിത കമ്പ്യൂട്ടിംങ്ങിന്റെ പ്രോത്സാഹനത്തിനുള്ള ആശയങ്ങൾ വിശദമാക്കുക.
5. സോഫ്റ്റ്‌വെയറിന്റെ വിവിധ വിഭാഗങ്ങൾ പട്ടികപ്പെടുത്തി വിശദീകരിക്കുക.
6. വിവിധ തരത്തിലുള്ള യൂട്ടിലിറ്റി സോഫ്റ്റ്‌വെയറുകളുടെ ഉപയോഗം വിശദീകരിക്കുക
7. ഓപ്പറേറ്റിംഗ് സിസ്റ്റം നിർവചിക്കുക. ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന്റെ പ്രധാന ധർമ്മങ്ങൾ പട്ടികപ്പെടുത്തി വിശദമാക്കുക.
8. പൊതുവായ പ്രവർത്തനങ്ങൾക്കുള്ള ആപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയർ ഉദാഹരണ സഹിതം വിശദമാക്കുക.

3

പ്രോഗ്രാമിങ് തത്വങ്ങളും പ്രശ്നപരിഹരണവും

പ്രധാന ആശയങ്ങൾ

- **കമ്പ്യൂട്ടറുകളുടെ സഹായത്തോടെയുള്ള പ്രശ്ന പരിഹാരം**
- **പ്രശ്ന പരിഹാരത്തിനുള്ള സമീപനങ്ങൾ**
  - o ടോപ് ഡൗൺ രൂപകല്പന
  - o ബോട്ടം അപ് രൂപകല്പന
- **പ്രോഗ്രാമിങ്ങിലെ വിവിധ ഘട്ടങ്ങൾ**
  - o പ്രശ്നം തിരിച്ചറിയൽ (Problem Identification)
  - o അൽഗോരിതങ്ങളും ഫ്ളോച്ചാർട്ടുകളും തയാറാക്കൽ (Preparing algorithms and Flowcharts)
  - o പ്രോഗ്രാം കോഡിങ് (Program Coding)
  - o പരിഭാഷ (Translation)
  - o ഡീബഗ്ഗിങ് (Debugging)
  - o പ്രവർത്തനവും പരീക്ഷണവും (Execution and Testing)
  - o വിവരണം തയാറാക്കൽ (Documentation)
- **അൽഗോരിതങ്ങളുടെ പ്രകടനം വിലയിരുത്തൽ**

Z7I2F4

ഡാറ്റാ പ്രോസസ്സിങ് എന്ന ആശയവും, അതിൽ കമ്പ്യൂട്ടറുകൾക്കുള്ള പങ്കും നാം പഠിച്ചിട്ടുണ്ട്. ഹാർഡ്‌വെയർ, സോഫ്റ്റ്‌വെയർ, ഉപയോക്താക്കൾ എന്നിവ അടങ്ങുന്ന ഒരു സംവിധാനം എന്ന നിലയിലും ഈ ഘടകങ്ങളെല്ലാം മുൻ അധ്യായത്തിൽ നാം വിശദമായി ചർച്ച ചെയ്തിട്ടുണ്ട്. സോഫ്ട്‌വെയറിന്റെ നിർവചനം നമുക്ക് ഓർത്തെടുത്ത് നോക്കാം. ലളിതമായ രീതിയിൽ പറഞ്ഞാൽ, കമ്പ്യൂട്ടർ ഉപയോഗിച്ച് പ്രശ്നങ്ങൾ പരിഹരിക്കുന്നതിനുള്ള പ്രോഗ്രാമുകളുടെ ശേഖരമാണ് സോഫ്റ്റ്‌വെയർ. കമ്പ്യൂട്ടറിന് സ്വന്തമായി ഒന്നും ചെയ്യാനുള്ള കഴിവില്ലെന്ന് നമുക്കറിയാവുന്നതാണ്. ഒരു ജോലി നിർവഹിക്കണമെങ്കിൽ വ്യക്തമായ നിർദേശങ്ങൾ കമ്പ്യൂട്ടറിന് ആവശ്യമാണ്. ആയതിനാൽ ഒരു പ്രശ്ന പരിഹാരത്തിന് കമ്പ്യൂട്ടർ നിർവഹിക്കേണ്ട നിർദേശങ്ങളുടെ ക്രമം വ്യക്തമാക്കേണ്ടത് അത്യാവശ്യമാണ്. ഇത്തരത്തിൽ കമ്പ്യൂട്ടറിന് മനസ്സിലാകുന്ന ഒരു ഭാഷയിൽ എഴുതിയിരിക്കുന്ന, ക്രമത്തിലുള്ള നിർദേശങ്ങൾ 'കമ്പ്യൂട്ടർ പ്രോഗ്രാം' എന്ന പേരിൽ അറിയപ്പെടുന്നു. കമ്പ്യൂട്ടർ പ്രോഗ്രാം എഴുതുക എന്നത് ഒരു വെല്ലുവിളിയാണ്. എന്നിരുന്നാലും പ്രോഗ്രാമിങ്ങിന്റെ വിവിധ ഘട്ടങ്ങളും പ്രശ്നപരിഹാരസാങ്കേതികതാ ആശയങ്ങളും ആർജിച്ചുകൊണ്ട് നമുക്ക് അതിനായി ശ്രമിക്കാം.

3.1 കമ്പ്യൂട്ടറുകൾ ഉപയോഗിച്ചുള്ള പ്രശ്ന പരിഹാരം (Problem solving using computers)

കമ്പ്യൂട്ടറിലേക്ക് നമ്മൾ നിർദേശങ്ങൾ നൽകിയാൽ മാത്രമെ അതിനു പ്രശ്നങ്ങൾ പരിഹരിക്കാൻ കഴിയുകയുള്ളൂ. നിർദേശങ്ങളിൽ അടങ്ങിയിരിക്കുന്ന ക്രിയകൾ മനസ്സിലായാൽ അതിനനുസരിച്ച് കമ്പ്യൂട്ടർ പ്രവർത്തിക്കും. നിർദേശം (Instruction) ഒരു ക്രിയാധിഷ്ഠിത പ്രസ്താവനയാണ്. ഏതു പ്രവൃത്തിയാണ് നിർവഹി

ക്കേണ്ടത് എന്ന് കംപ്യൂട്ടറിനോട് അത് പറയുന്നു. കൃത്യതയോടും സൂക്ഷ്മതയോടും കൂടി ചെയ്യേണ്ട പ്രവൃത്തി വ്യക്തമാക്കിയാൽ മാത്രമേ ഒരു നിർദേശത്തെ കമ്പ്യൂട്ടറിനു നിർവഹിക്കാൻ കഴിയുകയുള്ളൂ. പ്രശ്നപരിഹാരത്തിനായി പ്രോഗ്രാമർമാർ ഒരു പ്രത്യേക ക്രമത്തിൽ നിർദേശങ്ങൾ എഴുതുന്നു എന്ന് മുൻ അധ്യായത്തിൽ നാം പഠിച്ചിട്ടുണ്ട്. പ്രോഗ്രാം വികസിപ്പിക്കുകയും കമ്പ്യൂട്ടറിൽ സ്ഥിരമായി സൂക്ഷിക്കുകയും ചെയ്തു കഴിഞ്ഞാൽ നമുക്ക് ആവശ്യാനുസരണം അവ പ്രവർത്തിപ്പിക്കാൻ കമ്പ്യൂട്ടറിനോട് ആവശ്യപ്പെടാം

കമ്പ്യൂട്ടറിനു സാമാന്യബുദ്ധിയോ അന്തർജ്ഞാനമോ ഇല്ലാത്തതിനാൽ ഒരു പ്രോഗ്രാം രൂപകല്പന ചെയ്യുമ്പോൾ പ്രശ്ന പരിഹാരത്തിന്റെ യുക്തിയും നിർദേശങ്ങളുടെ ഘടനയും വ്യക്തമായി നാം മനസ്സിലാക്കിയിരിക്കണം. മനുഷ്യരായ നാം അനുഭവങ്ങളുടെ അടിസ്ഥാനത്തിൽ വിഷയാധിഷ്ഠിതവും വൈകാരികവുമായ പരിഗണനകൾക്കനുസരിച്ച്, തീരുമാനങ്ങൾ കൈക്കൊള്ളുന്നു. അത്തരം മൂല്യാധിഷ്ഠിത വിധിന്യായങ്ങൾ പലപ്പോഴും സാമാന്യബോധത്തെ ആശ്രയിച്ചിട്ടായിരിക്കും. ഇതിനു വിരുദ്ധമായി, ഒരു കമ്പ്യൂട്ടറിന് വികാരപ്രകടനമോ സാമാന്യബുദ്ധിയോ ഇല്ല. അതുകൊണ്ടാണ് കമ്പ്യൂട്ടറിന് സ്വന്തമായ ബുദ്ധിവൈഭവം ഇല്ല എന്നു നാം പറയുന്നത്.

ഒരു വിധത്തിൽ പറഞ്ഞാൽ, കമ്പ്യൂട്ടറിനെ ഒരു 'അനുസരണയുള്ള ഭൃത്യൻ' ആയി കണക്കാക്കാം. 'സാമാന്യബോധം' ഉപയോഗിക്കാതെയുള്ള അനുസരണശീലം പലപ്പോഴും അലോസരപ്പെടുത്തുന്നതും ഫലമില്ലാത്തതുമാകുന്നു.ഉദാഹരണത്തിന്, 'തപാൽ ഓഫീസിൽ പോയി പത്ത് 5 രൂപ സ്റ്റാമ്പുകൾ വാങ്ങുക'. എന്ന് നിർദേശിച്ച് തന്റെ അനുസരണയുള്ള ഭൃത്യനെ തപാൽ ഓഫീസിലേക്ക് യജമാനൻ അയച്ചു എന്ന് കരുതുക. ഭൃത്യൻ കാശുമായി തപാൽ ഓഫീസിൽ പോയിട്ട് ദീർഘ നേരമായും തിരികെ വരുന്നില്ല. ചിന്താകുലനായ യജമാനൻ ഭൃത്യനെ അന്വേഷിച്ചു തപാൽ ഓഫീസിലെത്തുമ്പോൾ കാണുന്നത് സ്റ്റാമ്പുകളുമായി നിൽക്കുന്ന ഭൃത്യനെയാണ്.കോപിഷ്ഠനായ യജമാനൻ ഭൃത്യനോട് കാരണം അന്വേഷിക്കുമ്പോൾ, തന്നോട് പത്തു 5 രൂപ സ്റ്റാമ്പുകൾ വാങ്ങുവാൻ മാത്രമേ കല്പിച്ചിട്ടുള്ളൂ എന്നും, അവയുമായി തിരികെ വരാൻ നിർദേശിച്ചിട്ടില്ല എന്നുമുള്ള മറുപടിയാണ് ഭൃത്യനിൽ നിന്ന് ലഭിക്കുന്നത്. .

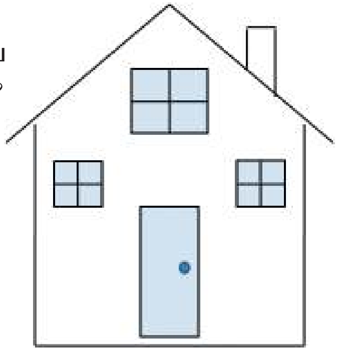
3.2 പ്രശ്നപരിഹാരത്തിലെ സമീപനങ്ങൾ (Approaches in problem solving)

ഒരു പ്രശ്നം വ്യത്യസ്ത രീതികളിലൂടെ പരിഹരിക്കാവുന്നതാണ്. സമീപനം പോലും വ്യത്യസ്തമായിരിക്കാം. നമ്മുടെ ദൈനംദിന ജീവിതത്തിൽ, അസുഖം ബാധിക്കുമ്പോൾ നാം ഒരു അലോപ്പതി, ആയുർവേദം അല്ലെങ്കിൽ ഹോമിയോപ്പതി ചികിത്സകനെ സമീപിച്ച് വൈദ്യചികിത്സ തേടുന്നു. ഒരേ രോഗമാണ് ചികിത്സിക്കുന്നതെങ്കിലും ഇവരിൽ ഓരോരുത്തരുടേയും സമീപനങ്ങൾ വ്യത്യസ്തമായിരിക്കും. അതു പോലെ പ്രശ്നപരിഹാരത്തിന് വ്യത്യസ്ത സമീപനങ്ങൾ ഉപയോഗിക്കുന്നു. പ്രശ്നപരിഹാരത്തിനുള്ള പ്രശസ്തമായ രണ്ടു രൂപകൽപ്പനാ രീതികൾ നമുക്ക് പരിചയപ്പെടാം ടോപ് ഡൗൺ (Top Down) രൂപകല്പനയും ബോട്ടം അപ് (Bottom Up) രൂപകല്പനയും

3.2.1 ടോപ് ഡൗൺ രൂപകല്പന (Top down design)

ചിത്രം 3.1 നോക്കുക. ഈ ചിത്രം വരയ്ക്കണമെന്ന് നിങ്ങളോട് ആവശ്യപ്പെടുകയാണെങ്കിൽ, എങ്ങനെയാണ് നിങ്ങൾ അത് വരയ്ക്കുക? ഒരു പക്ഷേ താഴെ പറയുന്ന പ്രകാരമായിരിക്കാം :

1. വീടിന്റെ രേഖാചിത്രം വരയ്ക്കുക.
2. ചിമ്മിനി വരയ്ക്കുക
3. വാതിൽ വരയ്ക്കുക
4. ജാലകങ്ങൾ വരയ്ക്കുക



ചിത്രം 3 .1 ഒരു വീടിന്റെ രേഖാചിത്രം

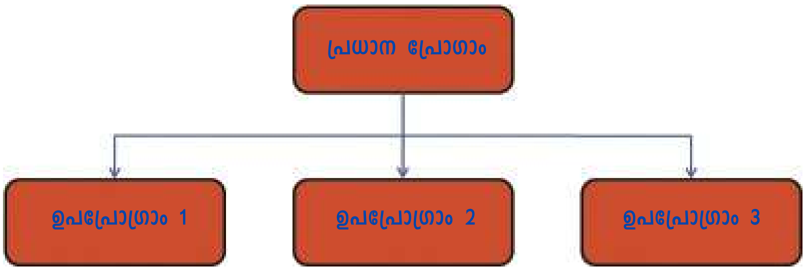
മുകളിൽ വിവരിച്ച നടപടിക്രമങ്ങളെ ഇപ്രകാരം സംഗ്രഹിക്കാം :

ഘട്ടം 3 ലെ വാതിൽ വരക്കുമ്പോഴുള്ള നടപടിക്രമം താഴെ കൊടുത്തിരിക്കുന്നു :
3.1 വാതിലിന്റെ ബാഹ്യരേഖ
3.2 ഷേഡിംഗ്
3.3 വാതിലിന്റെ പിടി

അത് പോലെ താഴെ പറയുന്ന പ്രകാരം ജാലകങ്ങൾ വരയ്ക്കാം
4.1 ജാലകത്തിന്റെ ബാഹ്യരേഖ
4.2 ഷേഡിംഗ്
4.3 തിരശ്ചീനവും ലംബവുമായ വരകൾ

തന്നിരിക്കുന്ന പ്രശ്നത്തെ (ഇവിടെ ചിത്രം വരയ്ക്കുക എന്നത്) ചെറിയ ക്രിയകളായി (Task) വിഭജിച്ചിരിക്കുന്നു. അതുപ്രകാരം ഈ പ്രശ്നം പരിഹരിക്കാൻ നാല് പ്രവർത്തനങ്ങൾ കണ്ടെത്തിയിട്ടുണ്ട്.ഇവയിൽ ചിലത് (ഇവിടെ വാതിലുകളും ജാലകങ്ങളും വരയ്ക്കുന്നത്) വീണ്ടും വിഭജിച്ചിട്ടുണ്ട്. അങ്ങനെ സങ്കീർണമായ ഒരു പ്രശ്നം,വിവിധ ക്രിയകളായി വിഭജിച്ച്, ഓരോ ക്രിയയെയും ലളിതമായ പ്രവർത്തനങ്ങളിലൂടെ പ്രാവർത്തികമാക്കാൻ കഴിയും. ഈ പ്രശ്ന പരിഹാര രീതി ടോപ് ഡൗൺ രൂപകല്പന (Top Down Design) എന്ന് അറിയപ്പെടുന്നു.

ഏറ്റവും ഫലവത്താണെന്നു തെളിയിക്കപ്പെട്ട പ്രോഗ്രാമിങ് സമീപനങ്ങളിലൊന്നാണിത്. ചിത്രം 3.2 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ, ടോപ് ഡൗൺ രൂപകല്പന എന്നത് തന്നിരിക്കുന്ന നടപടിക്രമത്തെ അല്ലെങ്കിൽ കൃത്യത്തെ ഘടകങ്ങൾ ആക്കുകയും ഏറ്റവും അടിസ്ഥാനപരമായ ക്രിയകൾ അടങ്ങിയ ഘടകം ലഭിക്കുന്നത് വരെ ഓരോ ഘടകത്തെയും വീണ്ടും വിഭജിക്കുകയും ചെയ്യുന്നു പ്രക്രിയയാണ്. ഒരു പൊതുവായ പ്രശ്നം മുകളിലെ തലം മുതൽ ആരംഭിക്കുകയും അതിലെ ഓരോ ഉപവിഭാഗത്തിനും പ്രത്യേക പരിഹാരങ്ങൾ രൂപകല്പന നടത്തുകയും ചെയ്യുന്നതിനാൽ ടോപ് ഡൗൺ വിഭജനം എന്ന പേരിലും ഇത് അറിയപ്പെടുന്നു. തന്നിരിക്കുന്ന പ്രധാന പ്രശ്നത്തിന് ഫലപ്രദമായ പരിഹാരം ലഭിക്കണമെങ്കിൽ ഓരോ ഉപപ്രശ്നവും മറ്റൊന്നിൽ നിന്ന് സ്വതന്ത്രമായിരിക്കണം. അപ്രകാരമായാൽ ഓരോ ഉപപ്രശ്നവും സ്വതന്ത്രമായി പരിഹരിക്കാനും പരിശോധിക്കാനും സാധിക്കും.



ചിത്രം 3.2: ഒരു പ്രശ്നത്തെ വിഭജിക്കുന്നു.

വിഭജനത്തിലൂടെ പ്രശ്നം പരിഹരിക്കുന്നത് കൊണ്ടുള്ള പ്രയോജനങ്ങൾ താഴെപ്പറയുന്നവയാണ്:

- പ്രശ്നത്തെ വിഭജിക്കുന്നതു കൊണ്ട് ഓരോ ഭാഗത്തും എന്ത് പ്രവൃത്തിയാണ് ചെയ്യേണ്ടതെന്നതിനെക്കുറിച്ച് ഒരു ധാരണ ലഭിക്കാൻ നമ്മെ സഹായിക്കുന്നു.

- ഓരോ ഘട്ടത്തിലും തിരിച്ചറിയുന്ന പുതിയ ഉപപ്രശ്നങ്ങളിൽ സങ്കീർണത കുറവായതിനാൽ അതിന്റെ പ്രശ്ന പരിഹാരം എളുപ്പത്തിൽ ലഭിക്കുന്നു.
- പ്രശ്ന പരിഹാരത്തിലെ ചില ഭാഗങ്ങൾ പുനരുപയോഗിക്കാൻ കഴിയുന്നതായിരിക്കാം.
- പ്രശ്നവിഭജനത്തിലൂടെ ഒന്നിലധികം ആളുകൾക്ക് ഒരേ സമയം പ്രശ്ന പരിഹാരത്തിൽ പങ്കാളികളാകാൻ സാധിക്കുന്നു.

3.2.2 ബോട്ടം അപ് രൂപകല്പന (Bottom Up Design)

ഒരു വീടിന്റെ നിർമ്മാണപ്രവർത്തനം പരിഗണിക്കുക. ടോപ് ഡൗൺ രൂപകല്പനയല്ല മറിച്ചു ബോട്ടം അപ് രൂപകല്പനയാണ് നമ്മൾ ഇവിടെ പിന്തുടരുന്നത്. അസ്ഥിവാരമിടുക എന്നത് ആദ്യത്തെ പ്രവൃത്തിയും മേൽക്കൂര പണിയുക എന്നത് അവസാനത്തെ പ്രവൃത്തിയുമാണ്. ഇവിടെയും പ്രധാന പ്രവർത്തനത്തെ ഉപപ്രവർത്തനങ്ങളായി വിഭജനം നടത്തുന്നു. ഇവയിൽ ത്തന്നെ ചില പ്രവർത്തനങ്ങൾ പൂർത്തിയായാൽ മാത്രമേ മറ്റു പ്രവർത്തനങ്ങൾ നടത്തുവാൻ കഴിയൂ. എന്നാൽ താഴെത്തട്ടിലുള്ള എല്ലാ പ്രവർത്തനങ്ങളും പൂർത്തിയായാൽ മാത്രമേ പ്രധാന പ്രവർത്തനമായ മേൽക്കൂര പണിയൽ സാധ്യമാവുകയുള്ളൂ.

ഇതുപോലെ പ്രോഗ്രാമിങ്ങിലും ആകെയുള്ള നടപടിക്രമങ്ങളെ വിവിധ ഘടകങ്ങളായി വിഭജിക്കുകയും, ഏറ്റവും താഴ്ന്ന തലത്തിലുള്ള ഖണ്ഡം ലഭിക്കുന്നത് വരെ ഈ ഘടകങ്ങൾ പുനർവിഭജിക്കുകയും ചെയ്യുന്നു. ഏറ്റവും താഴ്ന്ന ഘടകം മുതൽ പ്രശ്ന പരിഹാരം ആരംഭി ക്കുന്നു. പ്രധാന പ്രശ്നത്തിനുള്ള പരിഹാരം, ഉപവിഭാഗങ്ങളുടെ പരിഹാരത്തിനു ശേഷം മാത്രമേ സാധ്യമാകൂ. ഈ രീതിയിലുള്ള സമീപനത്തെ പ്രശ്ന പരിഹാരത്തിനുള്ള ബോട്ടം അപ്പ് രൂപക ല്പന എന്ന് പറയുന്നു. മുൻപ് പറഞ്ഞത് പോലെ ഇവിടെയും ഒരു ഉപപ്രശ്നം മറ്റൊരു ഉപപ്ര ശ്നത്തിൽ നിന്ന് സ്വതന്ത്രമായിരിക്കുന്നതാണ് അഭിലഷണീയം.

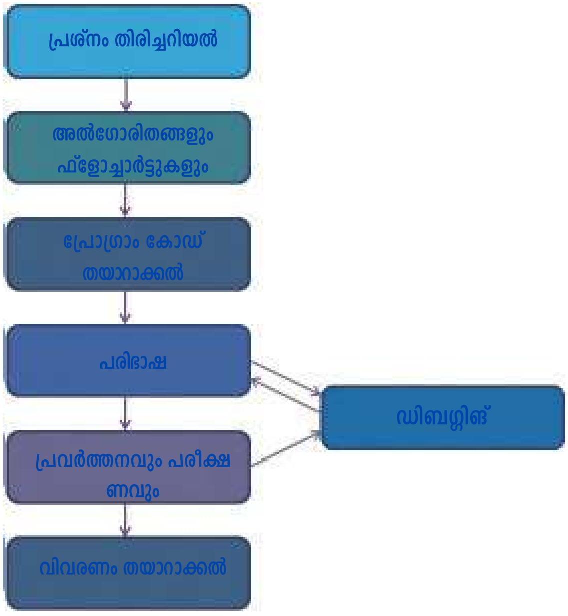
നമുക്കു ചെയ്യാം

എല്ലാ വർഷവും നമ്മുടെ സ്കൂളുകളിൽ യുവജനോത്സവം നടത്താറുണ്ട്. ഈ അവസരത്തിൽ, സാധാരണഗതിയിൽ ചുമതലകളും ഉത്തരവാദിത്വങ്ങളും വിഭജിച്ച് നൽകുന്നു. യുവജനോത്സവത്തിന്റെ വിജയകരമായ നടത്തിപ്പിന് എങ്ങനെയാണ് ഓരോ പ്രവർത്തനവും വിഭജിക്കുകയും പ്രാവർത്തികമാക്കുകയും ചെയ്യുന്നതെന്ന് ചർച്ച ചെയ്യുക.

3.3 പ്രോഗ്രാമിങ്ങിലെ വിവിധ ഘട്ടങ്ങൾ (Phases in programming)

കമ്പ്യൂട്ടർ ഉപയോഗിച്ച് പ്രശ്നം പരിഹരിക്കുക എന്നത് ഒരു വലിയ വെല്ലുവിളിയാണ് എന്ന് നാം കണ്ടുകഴിഞ്ഞു. ഇതിനായി ചിട്ടയായി ഒരു സമീപനം അത്യന്താപേക്ഷിതമാണ്. ആവശ്യമായ പ്രോഗ്രാമുകൾ നിർമ്മിക്കുന്നതിന് വിവിധ ഘട്ടങ്ങളിലൂടെ കടന്നുപോകേണ്ടതുണ്ട്. പ്രശ്ന പരിഹാ രത്തിനുള്ള ജൻമസിദ്ധമായ കഴിവ് നമുക്കുണ്ടെങ്കിലും അത് ഫലപ്രദമായി ഉപയോഗപ്പെടുത്ത ണമെങ്കിൽ പ്രശ്നം പരിഹരിക്കുന്നതിന് ഉതകുന്ന രീതിയിലുള്ള ചിന്തയും, ആസൂത്രണവും യുക്തിസഹമായ ന്യായവാദവും വളർത്തിയെടുക്കേണ്ടതുണ്ട്. താഴെപ്പറയുന്ന ഘട്ടങ്ങൾ പിന്തു ടർന്ന് നമുക്കിത് നേടിയെടുക്കാവുന്നതാണ്.

1. പ്രശ്നം തിരിച്ചറിയൽ (Problem Identification)
2. അൽഗോരിതങ്ങളും ഫ്ളോച്ചാർട്ടുകളും തയാറാക്കൽ (Preparing algorithms and Flowcharts)
3. പ്രോഗ്രാമിങ് ഭാഷ ഉപയോഗിച്ച് പ്രോഗ്രാം കോഡ് ചെയ്യൽ (Coding the Program using Programming Language)
4. പരിഭാഷ (Translation)
5. ഡീബഗ്ഗിങ് (Debugging)
6. പ്രവർത്തനവും പരീക്ഷണവും (Execution and Testing)
7. വിവരണം തയ്യാറാക്കുക (Documentation)



ചിത്രം 3.3 പ്രോഗ്രാമിങ്ങിലെ വിവിധ ഘട്ടങ്ങൾ

പ്രോഗ്രാമിങ്ങിന്റെ വിവിധ ഘട്ടങ്ങളിൽ ചെയ്യുന്ന പ്രവർത്തനങ്ങളുടെ ക്രമം ചിത്രം 3.3 ൽ കാണിച്ചിരിക്കുന്നു. ഡീബഗ്ഗിങ് ഘട്ടം പരിഭാഷയുമായും നിർവഹണവുമായും ബന്ധപ്പെട്ടിരിക്കുന്നു എന്നത് ശ്രദ്ധിക്കുക. മേൽപ്പറഞ്ഞ ഏഴ് ഘട്ടങ്ങളിൽ ഉൾപ്പെട്ടിരിക്കുന്ന പ്രവർത്തനങ്ങൾ താഴെ വിശദീകരിക്കുന്നു.

3.3.1 പ്രശ്നം തിരിച്ചറിയൽ (Problem Identification)

നിങ്ങൾക്കു വയറ് വേദന അനുഭവപ്പെടുന്നതായി കരുതുക. ഈ പ്രശ്നം ഒരു ഡോക്ടർക്കു പരിഹരിക്കാൻ കഴിയുമെന്ന് നിങ്ങൾക്കറിയാം. വേദന തുടങ്ങിയ സമയം,മുൻപ് ഇത് പോലെ വേദന അനുഭവപ്പെട്ട സാഹചര്യം, ഭക്ഷണരീതി എന്നിവയെക്കുറിച്ചുള്ള ചില ചോദ്യങ്ങൾ ഡോക്ടർ ചോദിക്കുകയും, സ്റ്റെതസ്കോപ്പ്, എക്സ്റേ അല്ലെങ്കിൽ സ്കാൻ എന്നിവ ഉപയോഗിച്ച് നിങ്ങളുടെ ശരീരത്തിന്റെ ചില ഭാഗങ്ങൾ പരിശോധിക്കുകയും ചെയ്യുന്നു.ഇവയെല്ലാം രോഗനിർണയത്തിന്റെ ഭാഗമാണ്. ഈ നടപടിക്രമങ്ങൾക്ക് ശേഷം, ഡോക്ടർ പ്രശ്നം തിരിച്ചറിഞ്ഞ് ചില വൈദ്യശാസ്ത്ര പദങ്ങൾ ഉപയോഗിച്ച് നിങ്ങളെ ബോധ്യപ്പെടുത്തുന്നു.അടുത്ത ഘട്ടം ഈ പ്രശ്നത്തിനുള്ള പരിഹാര നടപടികൾ തയാറാക്കലാണ്. അതിനെ മരുന്ന് കുറിപ്പ് എന്ന് പറയുന്നു.

ഇതിൽ നിന്നും പ്രശ്ന പരിഹാരത്തിനുള്ള നടപടികൾ സ്വീകരിക്കുന്നതിന് മുമ്പ് പ്രശ്നം വിശകലനം ചെയ്യേണ്ടത് അത്യാവശ്യമാണ് എന്ന് വ്യക്തമാകുന്നു. ഈ ഘട്ടത്തിൽ നിങ്ങൾക്ക് നടപ്പിലാക്കേണ്ട പ്രവർത്തനത്തിന് ആവശ്യമായ ഡാറ്റ, അതിന്റെ ഇനം, അളവ്, ഉപയോഗിക്കേണ്ട സൂത്രവാക്യം, ഉൾപ്പെട്ടിരിക്കുന്ന പ്രവർത്തനങ്ങൾ ലഭിക്കേണ്ട ഔട്ട്പുട്ട് എന്നിവ തിരിച്ചറിയാൻ സാധിക്കുന്നു. പ്രശ്നം വ്യക്തമായി പഠിക്കുകയും, പ്രശ്നപരിഹാരത്തിനാവശ്യമായ കൃത്യങ്ങളുടെ ക്രമം സംബന്ധിച്ച് നമുക്ക് ബോധ്യമാകുകയും ചെയ്താൽ, അടുത്ത ഘട്ടത്തിലേക്ക് പ്രവേശിക്കാം. പ്രോഗ്രാമറുടെ (പ്രശ്ന പരിഹാരകൻ) കാര്യക്ഷമത പരമാവധി ഉപയോഗപ്പെടുത്തേണ്ടതിനാൽ തീർച്ചയായും ഇത് ഒരു വെല്ലുവിളി നിറഞ്ഞ ഘട്ടമാണ്.

3.3.2 അൽഗോരിതങ്ങളും ഫ്ളോച്ചാർട്ടുകളും (Algorithms and Flowcharts)

പ്രശ്നം തിരിച്ചറിഞ്ഞു കഴിഞ്ഞാൽ അത് പരിഹരിക്കാൻ പടിപടിയായുള്ള നടപടിക്രമങ്ങൾ കൃത്യമായി വികസിപ്പിക്കേണ്ടത് അത്യാവശ്യമാണ്. ഈ നടപടിക്രമം പുതിയതോ കമ്പ്യൂട്ടർ മേഖലയ്ക്കുമാത്രം പരിമിതമായതോ അല്ല. കാലാകാലമായി ജീവിതത്തിന്റെ എല്ലാ മേഖലകളിലും തുടർന്ന് കൊണ്ടിരിക്കുന്ന ഒന്നാണിത്. നിത്യ ജീവിതത്തിൽ നിന്നെടുത്ത അത്തരത്തിലുള്ള ഒരു നടപടിക്രമം താഴെ വിവരിച്ചിരിക്കുന്നു. ഒരു മാസികയിൽ നിന്നും എടുത്തിട്ടുള്ള ഓംലെറ്റ് തയാറാക്കാനുള്ള പാചകക്കുറിപ്പാണിത്

ചേരുവകൾ

മുട്ട 2 എണ്ണം, ഉള്ളി 1 എണ്ണം (ചെറുതായി അരിഞ്ഞത്); പച്ച മുളക് 2 (ചെറുതായി അരിഞ്ഞത്); എണ്ണ 2 ടീ സ്പൂൺ, ഉപ്പ് ഒരു നുള്ള്.

രീതി

ഘട്ടം 1 : മുട്ടകൾ പൊട്ടിച്ച് ഒരു പാത്രത്തിലിട്ട് നന്നായി ഇളക്കുക.

ഘട്ടം 2 : അരിഞ്ഞു വച്ച ഉള്ളി, പച്ചമുളക്, ഉപ്പ് എന്നിവ മുട്ടയിൽ ചേർത്തിളക്കുക.

ഘട്ടം 3 : അടുപ്പിൽ ഒരു പാൻ വച്ച് അടുപ്പു കത്തിക്കുക.

ഘട്ടം 4 : പാനിൽ എണ്ണ ഒഴിച്ച്, ചൂടാകുന്നതുവരെ കാത്തിരിക്കുക.

ഘട്ടം 5 : ഘട്ടം 2 ൽ തയാറാക്കിയ മിശ്രിതം പാനിൽ ഒഴിച്ച് ഒരു ഭാഗം പൊരിയുന്നത് വരെ കാത്തിരിക്കുക.

ഘട്ടം 6 : മറിച്ചിട്ട് മറുവശം നന്നായി പൊരിക്കുക.

ഘട്ടം 7 : കുറച്ച് സെക്കൻഡുകൾക്ക് ശേഷം ഇത് എടുക്കുക.

ഫലം

കുരുമുളക് പൊടി ചേർത്ത് കഴിക്കാൻ പാകത്തിനുള്ള ഒരു ഓംലെറ്റ് തയാർ.

മുകളിൽ കൊടുത്ത പാചകകുറിപ്പിന് താഴെപ്പറയുന്ന സവിശേഷതകൾ ഉണ്ട്:

1. ഓംലെറ്റ് പാചകം ചെയ്യാൻ ആവശ്യമായ ചേരുവകളുടെ ഒരു പട്ടിക നിർമിക്കുന്നതിലൂടെയാണ് പ്രവർത്തനം ആരംഭിക്കുന്നത്. ഈ ചേരുവകളെ ഇൻപുട്ടുകൾ എന്ന് വിളിക്കാം.

2. ഇൻപുട്ടുകൾ ഉപയോഗിച്ചുള്ള പ്രവർത്തനത്തിന് ആവശ്യമായ, ക്രമത്തിലുള്ള നിർദേശങ്ങൾ നൽകുന്നു.

3. നിർദേശങ്ങൾ നടപ്പാക്കുന്നതിന്റെ ഫലമായി ചില ഔട്ട്പുട്ടുകൾ (ഇവിടെ, ഓംലെറ്റ്) ലഭിക്കുന്നു.

എന്നാൽ ഇൻപുട്ടുകൾ ഉപയോഗിച്ച് പ്രവർത്തനങ്ങൾക്കുള്ള നിർദേശങ്ങൾ കൃത്യമല്ല. അവ അവ്യക്തമാണ്. ഉദാഹരണത്തിന്, അഞ്ചാം ഘട്ടത്തിൽ "ഒരു ഭാഗം പൊരിക്കുന്നത്", ആറാം ഘട്ടത്തിൽ "നന്നായി പൊരിക്കുക" തുടങ്ങിയ നിർദേശങ്ങളുടെ വ്യാഖ്യാനം വ്യക്തികൾക്കനുസരിച്ചു വ്യത്യാസപ്പെട്ടിരിക്കും. തന്മൂലം കൃത്യമായ നിർദേശങ്ങളും, സമാന ഇൻപുട്ടുകളുമായി ഒരേ പാചകക്കുറിപ്പ് പിന്തുടരുന്ന വ്യത്യസ്ത വ്യക്തികൾക്ക്, വലുപ്പം, ആകൃതി, രുചി എന്നിവയ്ക്കനുസരിച്ച് വ്യത്യസ്ത ഓംലെറ്റുകൾ ലഭിക്കുന്നു.

കമ്പ്യൂട്ടർ ഉപയോഗിച്ചുള്ള പ്രശ്നങ്ങൾ പരിഹരിക്കുന്നതിനുള്ള നടപടികൾ എഴുതുമ്പോൾ മുകളിൽ പറഞ്ഞ അവ്യക്തതകൾ ഒഴിവാക്കേണ്ടതാണ്.

a. അൽഗോരിതം (Algorithm)

അബു ജഅഫർ മുഹമ്മദ് ഇബ്നു മുസാ അൽ ഖവാറീസ്മി എന്ന അറബി ഗണിതജ്ഞനാണ് അൽഗോരിതം എന്ന വാക്കിന്റെ ഉപജ്ഞാതാവായി അറിയപ്പെടുന്നത്. അദ്ദേഹത്തിന്റെ പേരിന്റെ 'അൽ ഖവാരിസിമി' എന്ന പദത്തിൽ നിന്നാണ് അൽഗോരിതം എന്ന പേര് ലഭിച്ചത്. കമ്പ്യൂട്ടർ പദാവലിയിൽ ഒരു പ്രശ്നം പരിഹരിക്കുന്നതിനുള്ള ക്രമത്തിലുള്ള നിശ്ചിത നിർദേശങ്ങളെ അൽഗോരിതം എന്നു നിർവചിക്കാവുന്നതാണ്. പ്രശ്നം പരിഹരിക്കാനുള്ള ഘട്ടംഘട്ടമായ നടപടിക്രമാണ് അൽഗോരിതം. ഇതിലെ ഓരോ ഘട്ടവും ചെയ്യപ്പെടേണ്ട നിശ്ചിതമായ കൃത്യത്തെ പ്രതിനിധീകരിക്കുന്നു. എന്നിരുന്നാലും, ഒരു അൽഗോരിതം ആകണമെങ്കിൽ, ക്രമത്തിലുള്ള നിർദേശങ്ങൾക്ക് താഴെപ്പറയുന്ന സവിശേഷതകൾ ഉണ്ടായിരിക്കേണ്ടതാണ് :

ചിത്രം 3.4 അബു ജഅഫർ മുഹമ്മദ് ഇബ്നു മുസാ അൽ ഖവാറീസ്മി (780 - 850)

(i) ഇൻപുട്ടുകൾ സ്വീകരിക്കുന്നതിനുള്ള നിർദേശം (നിർദേശങ്ങൾ) കൊണ്ടായിരിക്കണം അതിന്റെ തുടക്കം. തുടർന്നുള്ള നിർദേശങ്ങൾ വഴി ഈ ഇൻപുട്ടുകൾക്ക് മേൽ പ്രവർത്തനങ്ങൾ നടപ്പിലാക്കുന്നു. ചില സാഹചര്യങ്ങളിൽ, ഉപയോഗിക്കേണ്ട ഡാറ്റ പ്രശ്നത്തിനോടൊപ്പം തന്നെ നൽകിയിരിക്കും. അത്തരം സാഹചര്യങ്ങളിൽ, അൽഗോരിതത്തിന്റെ തുടക്കത്തിൽ ഇൻപുട്ട് സ്വീകരിക്കാനുള്ള നിർദേശങ്ങൾ ഉണ്ടായിരിക്കുകയില്ല .

(ii) ഡാറ്റയെ സൂചിപ്പിക്കാൻ വേരിയബിളുകൾ ഉപയോഗിക്കുക. ഇവിടെ വേരിയബിളുകൾ എന്നതു ഗണിതശാസ്ത്രത്തിലേതു പോലെ അക്ഷരങ്ങളും അക്കങ്ങളും അടങ്ങിയ ഉപയോക്തൃ നിർവചിത വാക്കുകളാണ്. ഡാറ്റ ഇൻപുട്ട് ചെയ്യുന്നതിനും വിലകൾ/ഫലങ്ങൾ സംഭരിക്കുന്നതിനും വേരിയബിളുകൾ തീർച്ചയായും ഉപയോഗിക്കേണ്ടതാണ്.

(iii) ഓരോ നിർദേശവും കൃത്യവും സ്പഷ്ടവും ആയിരിക്കണം. മറ്റൊരു വിധത്തിൽ പറഞ്ഞാൽ, നിർദേശങ്ങൾ അവ്യക്തമായിരിക്കരുത്. മാത്രമല്ല അവ നടപ്പിലാക്കാൻ സാധ്യമായതുമാകണം.

(iv) ഒരു വ്യക്തിക്ക് പേപ്പറും പെൻസിലും ഉപയോഗിച്ച് നിശ്ചിത സമയം കൊണ്ട് ചെയ്തു തീർക്കാവുന്ന തരത്തിൽ അടിസ്ഥാനപരമായതായിരിക്കണം ഓരോ നിർദേശവും.

(v) അൽഗോരിതത്തിൽ പറഞ്ഞിരിക്കുന്ന എല്ലാ നടപടികളും ചെയ്യുവാനുള്ള സമയം നിശ്ചിതമായിരിക്കണം. എന്തെന്നാൽ അൽഗോരിതത്തിലെ ചില നിർദേശങ്ങൾ ആവർത്തിച്ചു നിർവഹിക്കേണ്ടവയായിരിക്കും. അത്തരത്തിലുള്ള ആവർത്തനങ്ങളുടെ എണ്ണം നിശ്ചിതമായിരിക്കണം എന്നാണ് ഇത് സൂചിപ്പിക്കുന്നത്.

(vi) അൽഗോരിത്തിൽ നൽകിയിരിക്കുന്ന നിർദേശങ്ങൾ നിർവഹിച്ചാൽ പ്രതീക്ഷിച്ച ഫലം (ഔട്ട്പുട്ട്) ലഭിച്ചിരിക്കേണ്ടതാണ്.

അൽഗോരിതം സംബന്ധിച്ച ഉൾക്കാഴ്ച നേടാൻ നമുക്ക് ഒരു ലളിതമായ ഉദാഹരണം പരിഗണിക്കാം. തന്നിരിക്കുന്ന മൂന്ന് സംഖ്യകളുടെ ആകത്തുകയും (sum) ശരാശരിയും (average) നമുക്ക് കണ്ടുപിടിക്കണം.ഈ പ്രശ്നം പരിഹരിക്കാനുള്ള നടപടിക്രമം നമുക്ക് താഴെ പറയും പ്രകാരം എഴുതാം:

ഘട്ടം 1: മൂന്ന് സംഖ്യകൾ ഇൻപുട്ട് ചെയ്യുക.

ഘട്ടം 2: ആകത്തുക ലഭിക്കുന്നതിന് ഈ സംഖ്യകൾ കൂട്ടുക.

ഘട്ടം 3: ശരാശരി ലഭിക്കുന്നതിന് ആകത്തുകയെ 3 കൊണ്ട് ഹരിക്കുക.

ഘട്ടം 4: ആകത്തുകയും ശരാശരിയും പ്രിൻറ് ചെയ്യുക.

ഇവിടെ നടപടിക്രമം ശരിയാണെങ്കിലും, ഒരു അൽഗോരിതം തയാറാക്കുമ്പോൾ, ഒരു അംഗീകൃത ഘടന നമ്മൾ പിന്തുടരേണ്ടതുണ്ട്. മുകളിൽ പറഞ്ഞ പ്രക്രിയ ഒരു അൽഗോരിതം രൂപത്തിൽ എങ്ങനെ എഴുതാം എന്ന് നോക്കാം.

ഉദാഹരണം 3.1: മൂന്ന് സംഖ്യകളുടെ ആകത്തുക,ശരാശരി എന്നിവ കാണാനുള്ള അൽഗോരിതം.

ഇൻപുട്ട് ചെയ്യുന്ന സംഖ്യകൾ സ്വീകരിക്കാൻ A, B, C എന്ന വേരിയബിളുകൾ ഉപയോഗിക്കുക. അതുപോലെ S ആകത്തുകയ്ക്കു വേണ്ടിയും, Avg ശരാശരിക്കു വേണ്ടിയുമുള്ള വേരിയബിളുകൾ ആക്കുക.

ഘട്ടം 1: ആരംഭിക്കുക.
ഘട്ടം 2: A, B, C ഇൻപുട്ട് ചെയ്യുക.
ഘട്ടം 3: S = A+ B + C.
ഘട്ടം 4: Avg = S / 3.
ഘട്ടം 5: S, Avg പ്രിൻറ് ചെയ്യുക.
ഘട്ടം 6: അവസാനിപ്പിക്കുക

താഴെ പറയുന്ന കാരണങ്ങളാൽ മുകളിൽ പറഞ്ഞിരിക്കുന്ന നിർദേശങ്ങളുടെ കൂട്ടത്തെ അൽഗോരിതം ആയി കണക്കാക്കുന്നു:

- ഇതിന് ഇൻപുട്ട് ഉണ്ട് (ഇൻപുട്ട് ഡാറ്റ സംഭരിക്കാൻ വേരിയബിളുകൾ A, B, C ഉപയോഗിക്കുന്നു).
- ഒരു വ്യക്തിക്ക് പേപ്പറും പെൻസിലും ഉപയോഗിച്ച് കൃത്യമായി നിർവഹിക്കാവുന്ന രൂപത്തിൽ നടപടികൾ കൃത്യമായി പ്രസ്താവിച്ചിരിക്കുന്നു. (ഘട്ടം 3 ലും ഘട്ടം 4 ലും ഉചിതമായ ഓപ്പറേറ്ററുകൾ ഉപയോഗിച്ചുകൊണ്ട്)

- ഓരോ നിർദേശവും അടിസ്ഥാനപരവും അർഥവത്തുമാണ് (ഇൻപുട്ട്, പ്രിന്റ്, കൂട്ടുക, ഹരിക്കുക).
- ഇത് ആകെത്തുക (S), ശരാശരി (Avg) എന്നിങ്ങനെ രണ്ടു ഔട്ട്പുട്ടുകൾ സൃഷ്ടിക്കുന്നു.
- ആരംഭവും അവസാനവും സൂചിപ്പിക്കാനായി തുടങ്ങുക, നിർത്തുക എന്നീ നിർദേശങ്ങൾ ഉപയോഗിച്ചിരിക്കുന്നു.

നിർദേശങ്ങളുടെ തരങ്ങൾ

നമുക്കറിയാം ഒരു കമ്പ്യൂട്ടറിന് നിർവഹിക്കാൻ കഴിയുന്ന ക്രിയകളുടെ എണ്ണം പരിമിതമാണ്. അതിനാൽ പ്രശ്നപരിഹാരത്തിന് അത്രയും നിർദേശങ്ങൾ മാത്രമേ നമുക്ക് ഉപയോഗിക്കാൻ സാധിക്കുകയുള്ളൂ. കൂടുതൽ അൽഗോരിതങ്ങൾ നിർമിക്കുന്നതിന് മുമ്പ് അൽഗോരിതം നിർമിക്കാൻ ഉപയോഗിക്കുന്ന നിർദേശങ്ങളുടെ തരം ഏതൊക്കെയാണെന്ന് നമുക്ക് നോക്കാം.

- നമ്മൾ നൽകുന്ന ഡാറ്റ കമ്പ്യൂട്ടറിന് സ്വീകരിക്കാൻ സാധിക്കുന്നു. ആയതിനാൽ ഇൻപുട്ടിനുള്ള നിർദേശങ്ങൾ നമുക്ക് ഉപയോഗിക്കാവുന്നതാണ്. ഇൻപുട്ട്, സ്വീകരിക്കുക, വായിക്കുക മുതലായ പദങ്ങൾ നമുക്ക് ഇതിനായി ഉപയോഗിക്കാം.

- കമ്പ്യൂട്ടർ ഫലങ്ങൾ ഔട്ട്പുട്ടായി നൽകുന്നു.ആയതിനാൽ നമുക്ക് ഔട്ട്പുട്ടുമായി ബന്ധപ്പെട്ട നിർദേശങ്ങൾ ഉപയോഗിക്കാം. പ്രിന്റ്, പ്രദർശിപ്പിക്കുക, എഴുതുക മുതലായ പദങ്ങൾ നമുക്ക് ഇതിന് ഉപയോഗിക്കാം.

- ഒരു മെമ്മറി സ്ഥാനത്ത് ഡാറ്റ നേരിട്ട് ശേഖരിക്കാം അല്ലെങ്കിൽ ഡാറ്റ ഒരു സ്ഥാനത്ത് നിന്ന് മറ്റൊന്നിലേക്ക് പകർത്തുകയും ചെയ്യാം. അതുപോലെ, ഡാറ്റയുടെ മുകളിലുള്ള ഗണിത പ്രവർത്തനങ്ങളുടെ ഫലങ്ങൾ മെമ്മറി സ്ഥാനങ്ങളിൽ സംഭരിക്കാം. ഇതിനായി ഗണിത ശാസ്ത്രത്തിലേതിനു സമാനമായി വിലനൽകൽ (assignment) (അല്ലെങ്കിൽ സംഭരണം) നിർദേശം നമുക്ക് ഉപയോഗിക്കാം. മൂല്യങ്ങൾ സംഭരിക്കുന്നതിന് വേരിയബിളുകൾക്കു ശേഷം സമ ചിഹ്നം (=) ഉപയോഗിക്കുന്നു. ഇവിടെ വേരിയബിളുകൾ എന്നത് മെമ്മറി സ്ഥാനങ്ങളെ സൂചിപ്പിക്കുന്നു.

- കമ്പ്യൂട്ടറിന് ഡാറ്റ മൂല്യങ്ങൾ താരതമ്യം ചെയ്ത് (ലോജിക്കൽ ഓപ്പറേഷൻ എന്ന് വിളിക്കുന്നു), അതിന്റെ അടിസ്ഥാനത്തിലുള്ള തീരുമാനങ്ങൾ എടുക്കാൻ സാധിക്കും. ഇത്തരം തീരുമാനങ്ങൾ ഒന്നോ അതിലധികമോ പ്രസ്താവനകളുടെ തിരഞ്ഞെടുക്കൽ / ഒഴിവാക്കൽ അല്ലെങ്കിൽ ഒരുകൂട്ടം നിർദേശങ്ങളുടെ ആവർത്തിച്ചുള്ള പ്രവർത്തനം എന്ന രൂപത്തിലായിരിക്കും.

b. ഫ്ളോച്ചാർട്ടുകൾ (Flowchart)

ഒരു ചിത്രം അല്ലെങ്കിൽ രേഖാചിത്രത്തിന്റെ രൂപത്തിൽ ആവിഷ്കരിക്കപ്പെട്ടിരിക്കുന്ന ഒരു സങ്കല്പം ആണ് എഴുത്ത് രൂപത്തെക്കാൾ ആളുകൾക്ക് സ്വീകാര്യമാകുക. ചില സാഹചര്യങ്ങളിൽ അൽഗോരിതം മനസ്സിലാക്കുക എന്നത് ബുദ്ധിമുട്ടേറിയതായിരിക്കും. എന്തെന്നാൽ ചില സങ്കീർണമായ പ്രവർത്തനങ്ങളും ആവർത്തിച്ചുവരുന്ന ഘട്ടങ്ങളും അതിൽ ഉൾപ്പെട്ടേക്കാം. അതിനാൽ അൽഗോരിതം ചിത്ര രൂപത്തിൽ ആവിഷ്കരിക്കുക എന്നതായിരിക്കും മെച്ചപ്പെട്ട രീതി. നിർദേശങ്ങൾ സൂചിപ്പിക്കുന്നതിനുള്ള പ്രത്യേക ചിഹ്നങ്ങളും പ്രവർത്തനങ്ങളുടെ ക്രമം സൂചിപ്പിക്കുന്നതിന് ആരോകളും ഉപയോഗിച്ച് നിർമിക്കുന്ന അൽഗോരിതത്തിന്റെ ചിത്ര ആവിഷ്കരണമാണ് ഫ്ളോച്ചാർട്ട്. ഒരു അൽഗോരിതം ചിട്ടപ്പെടുത്തുന്നതിനും മനസ്സിലാക്കുന്നതിനുമുള്ള ഒരു സഹാ

യിയായിട്ടാണ് പ്രധാനമായിട്ടും ഇത് ഉപയോഗിക്കുന്നത്. വിവിധതരത്തിലുള്ള നിർദേശങ്ങൾ സൂചിപ്പിക്കുന്നതിനായി അടിസ്ഥാനപരമായ ജ്യാമിതീയ രൂപങ്ങൾ ഫ്ളോച്ചാർട്ടുകളിൽ സാധാരണമായി ഉപയോഗിക്കുന്നു. യഥാർഥ നിർദേശങ്ങൾ ഇത്തരം രൂപങ്ങൾക്കുള്ളിൽ കൃത്യവും വ്യക്തവുമായ പ്രസ്താവനകൾ ഉപയോഗിച്ച് എഴുതുന്നു. പ്രവർത്തനങ്ങളുടെ ക്രമത്തെ അതായത് നിർദേശങ്ങൾ പ്രാവർത്തികമാക്കേണ്ട ക്രമത്തെ സൂചിപ്പിക്കുന്ന തരത്തിൽ ഈ രൂപങ്ങൾ ആരോകളോട് കൂടിയ നേർരേഖകൾ വഴി പരസ്പരം ബന്ധപ്പെടുത്തിയിരിക്കുന്നു.

സാധാരണഗതിയിൽ ഒരു അൽഗോരിതത്തെ ഫ്ളോച്ചാർട്ടിലേക്ക് രൂപഭേദം വരുത്തിയ ശേഷമാണ് നിർദേശങ്ങൾ പ്രോഗ്രാമിങ് ഭാഷയിൽ ആവിഷ്കരിക്കുന്നത്. പ്രോഗ്രാം എഴുതുന്നതിൽ ഈ ദ്വിതല (Two step approach) സമീപനത്തിന്റെ മുഖ്യ ഗുണം എന്തെന്നാൽ ഫ്ളോച്ചാർട്ട് വരയ്ക്കുന്ന സമയത്ത് പ്രോഗ്രാം ഭാഷയുടെ വിവിധ ഘടകങ്ങളുടെ വിശദാംശങ്ങളെ പറ്റി വരയ്ക്കുന്നയാൾ ചിന്തിക്കേണ്ടതില്ല. അതിനാൽ അവന്/അവൾക്ക് നടപടിക്രമത്തിന്റെ യുക്തിയിൽ (ഘട്ടംഘട്ടമായുള്ള രീതി) കൂടുതൽ ശ്രദ്ധ പതിപ്പിക്കാൻ സാധിക്കുന്നു. മാത്രമല്ല, ഫ്ളോച്ചാർട്ടിൽ, പ്രവർത്തനങ്ങളുടെ ക്രമം ചിത്രരൂപത്തിൽ സൂചിപ്പിക്കുന്നതിനാൽ നടപടിക്രമത്തിന്റെ യുക്തിയിൽ വരുന്ന തെറ്റുകൾ, പ്രോഗ്രാമിൽ തെറ്റ് കണ്ടെത്തുന്നതിനെക്കാൾ എളുപ്പത്തിൽ കണ്ടെത്താൻ കഴിയുന്നു. അൽഗോരിതവും ഫ്ളോച്ചാർട്ടും എപ്പോഴും പ്രോഗ്രാമറിനുള്ള പ്രമാണം ആകുന്നു. ഒരിക്കൽ ഇവ തയാറാകുകയും പ്രശ്ന പരിഹാരത്തിന്റെ യുക്തി ശരിയാണെന്ന് ബോധ്യമാവുകയും ചെയ്തു കഴിഞ്ഞാൽ പ്രോഗ്രാമർക്ക് പ്രോഗ്രാമിങ് ഭാഷയിലുള്ള വിവിധ നിർമ്മിതികളുടെ സഹായത്തോടെ ചെയ്യേണ്ട പ്രവർത്തനങ്ങളെ കോഡ് ചെയ്യുന്നതിൽ പ്രോഗ്രാമർക്ക് ശ്രദ്ധ ചെലുത്താൻ സാധിക്കുന്നു. പ്രോഗ്രാം തെറ്റുകൾ ഇല്ലാത്തതാണെന്നു ഉറപ്പു വരുത്താൻ സാധാരണ ഗതിയിൽ ഇതിനു കഴിയുന്നു.

ഫ്ളോച്ചാർട്ടിലെ ചിഹ്നങ്ങൾ

അംഗീകരിക്കപ്പെട്ട അർഥവത്തായ ചിഹ്നങ്ങൾ ഉപയോഗിക്കുന്നതിലൂടെ ഫ്ളോചാർട്ടുകൾ വഴിയുള്ള പ്രോഗ്രാം യുക്തിയുടെ ആശയ വിനിമയം എളുപ്പമായി തീരുന്നു. അവശ്യമായ ചില പ്രവർത്തനങ്ങൾ സൂചിപ്പിക്കുന്നതിന് ആവശ്യമായ ചില ചിഹ്നങ്ങൾ മാത്രമാണ് നമ്മൾ ഇവിടെ പരിചയപ്പെടുന്നത്. ഈ ചിഹ്നങ്ങൾ അമേരിക്കൻ നാഷണൽ സ്റ്റാൻഡേർഡ്സ് ഇൻസ്റ്റിറ്റ്യൂട്ട് (ANSI) അംഗീകരിച്ചതാണ്.

1. ടെർമിനൽ (Terminal)

പേര് സൂചിപ്പിക്കുന്നതുപോലെ, പ്രോഗ്രാം യുക്തിയുടെ ആരംഭത്തെയും അവസാനത്തേയും ഈ ചിഹ്നം സൂചിപ്പിക്കുന്നു. ഒരു ഫ്ളോച്ചാർട്ടിന്റെ ആദ്യത്തെയും അവസാനത്തെയും ചിഹ്നമാണിത്.

ഒരു അണ്ഡവൃത്തത്തിന്റെ (ellipse) ആകൃതിയാണ് ഇതിന്. തുടക്കത്തിൽ ഉപയോഗിക്കുമ്പോൾ നിർഗമനത്തിന്റെ ആരോ ഇതിൽ ഘടിപ്പിച്ചിരിക്കും. പക്ഷേ അവസാനത്തിൽ ഉപയോഗിക്കുമ്പോൾ ആഗമനത്തിന്റെ ആരോ ഇതിൽ ഘടിപ്പിച്ചിരിക്കും.

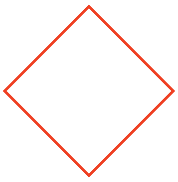
2. ഇൻപുട്ട്/ഔട്ട്പുട്ട് (input/output)

ഇൻപുട്ട്/ഔട്ട്പുട്ട് ചിഹ്നമായി ഉപയോഗിക്കുന്നത് ഒരു സമാന്തരഭുജ (Parallelogram) മാണ്. പ്രോഗ്രാമിൽ ഒരു ഇൻപുട്ട് /ഔട്ട്പുട്ട് ഉപകരണത്തിന്റെ ധർമ്മത്തെ ഇത് സൂചിപ്പിക്കുന്നു. എല്ലാ ഇൻപുട്ട്/ഔട്ട്പുട്ട് നിർദേശങ്ങളും ഈ ചിഹ്നം ഉപയോഗിച്ചാണ് ആവിഷ്കരിച്ചിരിക്കുന്നത്. ഇതിലേക്ക് ഒരു ആഗമന ആരോയും ഒരു നിർഗമന ആരോയും ഘടിപ്പിച്ചിരിക്കും.

3. പ്രവർത്തനം (process)

പ്രവർത്തന ഘട്ടത്തെ പ്രതിനിധീകരിക്കാൻ ദീർഘചതുരം ഉപയോഗിക്കുന്നു. സങ്കലനം, വ്യവകലനം ഗുണിനം ഹരണം തുടങ്ങിയ ഗണിത ക്രിയകൾ ചെയ്യാനും അതുപോലെ വേരിയബിളിയിലേക്ക് വില നൽകുവാനും ഈ ചിഹ്നം ഉപയോഗിക്കുന്നു. വേരിയബിളുകൾക്ക് വില നൽകുക (assignment) എന്നത് കൊണ്ടുദ്ദേശിക്കുന്നത്- ഒരു മെമ്മറി സ്ഥാനത്തു നിന്ന് മറ്റൊന്നിലേക്ക് ഡാറ്റ പകർത്തുന്നതോ (ഉദാ:a=b) അല്ലെങ്കിൽ എ.ൽ.യു. വിൽ (ALU) നിന്നും ഔട്ട്പുട്ടിനെ മെമ്മറി സ്ഥാനത്തേക്ക് പകർത്തുന്നതോ (ഉദാ:a=b+5) അല്ലെങ്കിൽ ഒരുപക്ഷേ മെമ്മറി സ്ഥാനത്തേക്ക് വില നേരിട്ട് സംഭരിക്കുന്നതോ (ഉദാ:a=2) ആകാം. പ്രോസസ്സ് ചിഹ്നത്തിനു ഒരു ആഗമന ആരോയും ഒരു നിർഗമന ആരോയും ഉണ്ടായിരിക്കും.

4. തീരുമാനം (Decision)



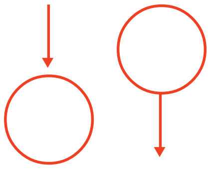
തീരുമാനങ്ങൾ സൂചിപ്പിക്കുന്നതിനായുള്ള ചിഹ്നമായി ചതുർഭുജം ഉപയോഗിക്കുന്നു. ഇത് തീരുമാനങ്ങൾ കൈക്കൊള്ളേണ്ട ഒരു ഘട്ടത്തെയാണ് സൂചിപ്പിക്കുന്നത്. ഈ ഘട്ടത്തിൽ നിന്ന് ഒന്നോ അതിലധികമോ ഇതര ഘട്ടങ്ങളിലേക്ക് വിഭജനം സാധ്യമാണ്. എല്ലാ നിർഗമന പാതകളും ഇവിടെ പരാമർശിച്ചിട്ടുണ്ടായിരിക്കും. എന്നിരുന്നാലും ഒരു വ്യവസ്ഥ (condition) പരിശോധിച്ച് അതിന്റെ ഫലത്തിന്റെ അടിസ്ഥാനത്തിൽ അടുത്ത ഒരു പാത മാത്രമേ തിരഞ്ഞെടുക്കപ്പെടുകയുള്ളൂ. സാധാരണഗതിയിൽ ഈ ചിഹ്നത്തിന് ഒരു ആഗമന മാർഗവും 2 നിർഗമന മാർഗങ്ങളും ഉണ്ടായിരിക്കും. ഒന്ന് വ്യവസ്ഥയുടെ ഫലം ശരിയാണെങ്കിൽ ചെയ്യേണ്ടുന്ന പ്രവൃത്തിയുടെ നേർക്കും, മറ്റേത് ഇതര മാർഗത്തിലേക്കും.

5. ഫ്ളോ ലൈനുകൾ (flow lines)

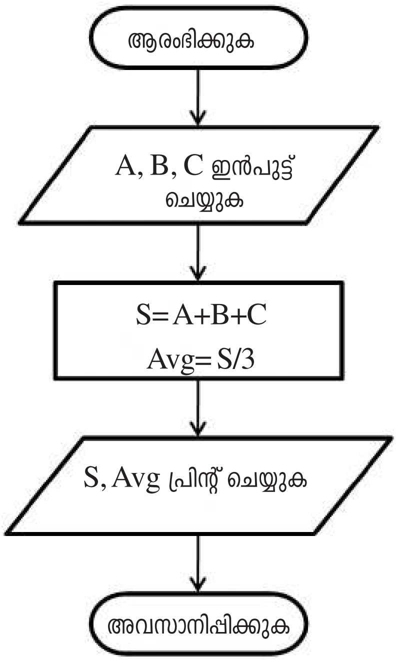
പ്രവർത്തന ക്രിയകളുടെ ഒഴുക്കിനെ സൂചിപ്പിക്കാൻ ആരോകളോട് കൂടിയ ഫ്ളോ ലൈനുകൾ ഉപയോഗിക്കുന്നു. അതായത് നിർദേശങ്ങൾ പ്രാവർത്തികമാക്കേണ്ട കൃത്യമായ ക്രമത്തെ ഇവ സൂചിപ്പിക്കുന്നു. സാധാരണഗതിയിൽ ഫ്ളോ ലൈനുകളുടെ ദിശ മുകളിൽ നിന്നും താഴേക്കും ഇടത്തുനിന്നും വലത്തേക്കും ആയിരിക്കും. എന്നാൽ ചില സാഹചര്യങ്ങളിൽ ഇത് വലതു നിന്ന് ഇടത്തേക്കും താഴെനിന്നും മുകളിലേക്കും ആവാം. ഫ്ളോ ലൈനുകൾ

പരസ്പരം ഛേദിക്കുന്നത് നല്ല പ്രവണതയല്ല. അത്തരം ഛേദനങ്ങൾ പരമാവധി ഒഴിവാക്കേണ്ടതാണ്.

6. കണക്ടർ (connector)



ഫ്ളോച്ചാർട്ടുകളുടെ വലിപ്പം കൂടുമ്പോൾ ഫ്ളോ ലൈനുകൾ പല ഭാഗങ്ങളിലും പരസ്പരം ഛേദിക്കുകയും പല സ്ഥലങ്ങളിലും ആശയക്കുഴപ്പം വരുത്തുകയും തൻമൂലം ഫ്ളോച്ചാർട്ടിന്റെ ഗ്രഹണം ബുദ്ധിമുട്ടാക്കുകയും ചെയ്യുന്നു. അതുപോലെ ഒരു പേജിൽ ഒതുങ്ങുന്നതിനെക്കാൾ വലിയ ഫ്ളോച്ചാർട്ട് ആകുമ്പോൾ ഫ്ളോ ലൈനുകളുടെ ഉപയോഗം അസാധ്യമായിത്തീരുന്നു. ഒരു ഫ്ളോച്ചാർട്ട് സങ്കീർണമാകുകയും ഫ്ളോ ലൈനുകളുടെ എണ്ണം, ദിശ എന്നിവയെക്കുറിച്ച് ആശയക്കുഴപ്പം സംഭവിക്കുകയോ അല്ലെങ്കിൽ ഫ്ളോച്ചാർട്ട് ഒന്നിൽ അധികം പേജുകളിലായി പടരുകയോ ചെയ്താൽ മുറിഞ്ഞുപോയ ഫ്ളോ ലൈനുകളെ തമ്മിൽ കൂട്ടിയോജിപ്പിക്കാൻ ഒരു ജോടി കണക്റ്റർ ചിഹ്നങ്ങൾ ഉപയോഗിക്കാം. ഫ്ളോച്ചാർട്ടിന്റെ ഒരു ഭാഗത്ത് നിന്നുമുള്ള ആഗമനത്തെ അല്ലെങ്കിൽ അതിന്റെ മറ്റൊരു ഭാഗത്തേക്കുള്ള നിർഗമനത്തെ ഈ ചിഹ്നം സൂചിപ്പിക്കുന്നു. ഒരു അക്കം അല്ലെങ്കിൽ അക്ഷരത്തോട് കൂടിയ ഒരു വൃത്തം ഉപയോഗിച്ചാണ് കണക്റ്റർ ചിഹ്നം സൂചിപ്പിക്കുന്നത്. ഒരേ പോലെ അടയാളപ്പെടുത്തിയ ഒരു ജോഡി കണക്റ്റർ ചിഹ്നങ്ങൾ അൽഗോരിതത്തിന്റെ തുടർച്ചയെ സൂചിപ്പിക്കുന്നു. ഒരു വലിയ ഫ്ളോ ലൈനിന് പകരം ഒരേ ചിഹ്നങ്ങൾ / അക്കങ്ങൾ ഉള്ള രണ്ടു കണക്റ്ററുകൾ ഉപയോഗിക്കാം. അതായത് ഒരേപോലെ അടയാളപ്പെടുത്തിയ ഒരു ജോടി കണക്ടറുകളിൽ ഒന്ന് ഫ്ളോച്ചാർട്ടിന്റെ മറ്റൊരു ഭാഗത്തേക്കുള്ള നിർഗമനത്തെയും, രണ്ടാമത്തേത് ഫ്ളോച്ചാർട്ടിന്റെ മറ്റൊരു ഭാഗത്തേക്കുള്ള ആഗമനത്തെയും സൂചിപ്പിക്കുന്നു.



ചിത്രം 4.5 ആകെതുകയും ശരാശരിയും കാണാനുള്ള ഫ്ളോച്ചാർട്ട്

ഉദാഹരണം 4.1 ൽ വിശദീകരിച്ച പ്രശ്നത്തിന്റെ ഫ്ളോച്ചാർട്ട് ചിത്രം 4.5 ൽ കാണിച്ചിരിക്കുന്നു.

അൽഗോരിതത്തിലെ ഓരോ ഘട്ടത്തിലെയും നിർദേശങ്ങൾ സൂചിപ്പിക്കാൻ ഉചിതമായ ചിഹ്നങ്ങൾ ഉപയോഗിച്ചിരിക്കുന്നു.മാത്രമല്ല ഓരോ ചിഹ്നത്തിലും അതിനനുസൃതമായ നിർദേശങ്ങൾ രേഖപ്പെടുത്തുന്നു. പ്രവർത്തനങ്ങളുടെ ക്രമം ഫ്ളോ ലൈനുകൾ ഉപയോഗിച്ച് കൃത്യമായി അടയാളപ്പെടുത്തുകയും ചെയ്യുന്നു.

ഫ്ളോച്ചാർട്ടിന്റെ ഗുണങ്ങൾ

പ്രോഗ്രാം ആസൂത്രണത്തിൽ പലരീതികളിലും ഫ്ളോച്ചാർട്ടുകൾ ഗുണപ്രദങ്ങളാണ്.

- **മികച്ച ആശയവിനിമയം :** ഫ്ളോച്ചാർട്ട് ഒരു പ്രോഗ്രാമിന്റെ ചിത്രരൂപത്തിലുള്ള സൂചകം, ആയതിനാൽ ഒരു പ്രോഗ്രാമർക്ക് മറ്റൊരു പ്രോഗ്രാമറിന് പ്രോഗ്രാമിന്റെ യുക്തി ഫ്ളോച്ചാർട്ട് ഉപയോഗിച്ച് വിശദീകരിച്ചു കൊടുക്കുന്നത് പ്രോഗ്രാം വിശദീരിക്കുന്നതിനേക്കാൾ എളുപ്പമാണ്.
- **ഫലപ്രദമായ വിശകലനം :** പ്രോഗ്രാമിലെ വിവിധ ഘട്ടങ്ങൾ കൃത്യമായി ഫ്ളോച്ചാർട്ടിൽ പ്രതിപാദിച്ചിരിക്കുന്നതിനാൽ, പ്രോഗ്രാമിനെ ഫലപ്രദമായി വിശകലനം ചെയ്യാൻ ഫ്ളോച്ചാർട്ട് സഹായിക്കുന്നു.
- **ഫലപ്രദമായ സമന്വയം :** പ്രോഗ്രാമിനെ വിവിധ ഘടകങ്ങളായി തിരിക്കുകയും അവ ഓരോന്നിന്റെയും പരിഹാരം ഫ്ളോച്ചാർട്ടുകളായി പ്രത്യേകം പ്രത്യേകമായി തയാറാക്കുകയും ചെയ്താൽ, അവയെ എല്ലാം കൂട്ടി യോജിപ്പിച്ചു മൊത്തത്തിലുള്ള സിസ്റ്റത്തിന്റെ രൂപരേഖ നമുക്ക് തയാറാക്കാവുന്നതാണ്.
- **ഫലപ്രദമായ കോഡിങ് :** ഫ്ളോച്ചാർട്ട് തയ്യാറാക്കി കഴിഞ്ഞാൽ പ്രോഗ്രാമർക്കു അനുബന്ധ പ്രോഗ്രാം തയ്യാറാക്കാൻ എളുപ്പമാണ്, എന്തെന്നാൽ ഫ്ളോച്ചാർട്ട് പ്രോഗ്രാമിന്റെ ഒരു രേഖാചിത്രമായി പ്രവർത്തിക്കുന്നു. പ്രോഗ്രാമിന്റെ തുടക്കം മുതലുള്ള എല്ലാ ഘട്ടങ്ങളിലൂടെയും കടന്നു പോയി, ഒന്ന് പോലും വിട്ടുപോകാതെ അവസാനം വരെയും എത്തിച്ചേരുവാനുള്ള ഒരു സഹായിയായി ഇത് വർത്തിക്കുന്നു.

ഫ്ളോച്ചാർട്ടിന്റെ പരിമിതികൾ

ഫ്ളോച്ചാർട്ടുകൾക്ക് ഇത്തരത്തിലുള്ള ഗുണങ്ങൾ എടുത്തു പറയാമെങ്കിലും ,ചില പരിമിതികളും അവയ്ക്കുണ്ട്.

- ഉചിതമായ ചിഹ്നങ്ങളും സ്പേസും നൽകിയുള്ള ഫ്ളോച്ചാർട്ട് നിർമാണം സമയം ചെലവഴിച്ചു ചെയ്യേണ്ടതും കഠിനാധ്വാനം ആവശ്യമായതുമാണ്, പ്രത്യേകിച്ചും സങ്കീർണ്ണമായ അൽഗോരിതങ്ങൾ ആണെങ്കിൽ.
- അൽഗോരിതത്തിന്റെ യുക്തിയിലുള്ള വളരെച്ചെറിയ മാറ്റത്തിനുപോലും പുതിയ ഫ്ളോച്ചാർട്ട് ആവശ്യമായി വരുന്നു.
- ഫ്ളോച്ചാർട്ടിൽ ഉൾപ്പെടുത്തേണ്ട വിശദശാംശങ്ങളെ പറ്റി വിശദീകരിക്കുന്ന ഒരു തരത്തിലുള്ള മാനദണ്ഡങ്ങളും നിലവിലില്ല.

വിവിധ പ്രശ്നങ്ങൾ പരിഹരിക്കുന്നതിനുള്ള അൽഗോരിതങ്ങളും ഫ്ളോച്ചാർട്ടുകളും നമുക്ക് വികസിപ്പിക്കാം.

ഉദാഹരണം 3.2: ഒരു ചതുരത്തിന്റെ വിസ്തീർണവും ചുറ്റളവും കണ്ടെത്തുക

ചതുരത്തിന്റെ നീളവും വീതിയും ലഭ്യമായാൽ ഈ പ്രശ്നം നമുക്കു പരിഹരിക്കാം. താഴെ പറയുന്ന സമവാക്യം ഇതിന് വേണ്ടി ഉപയോഗിക്കാവുന്നതാണ്.

ചുറ്റളവ് = 2 x (നീളം + വീതി), വിസ്തീർണം = (നീളം x വീതി).

L, B എന്നീ വേരിയബിളുകൾ നീളം വീതി എന്നി സൂചിപ്പിക്കാനും P, A എന്നീ വേരിയബിളുകൾ വിസ്തീർണം, ചുറ്റളവ് എന്നിവ സൂചിപ്പിക്കാനും ഉപയോഗിക്കുന്നു എന്നിരിക്കട്ടെ

ഘട്ടം 1 : തുടങ്ങുക

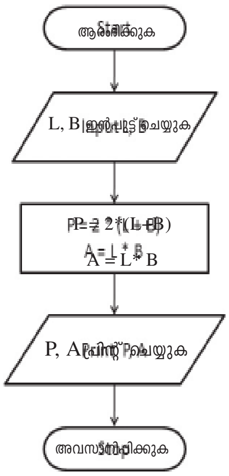
ഘട്ടം 2 : L , B ഇൻപുട്ടായി സ്വീകരിക്കുക

ഘട്ടം 3 : P = 2 *(L +B)

ഘട്ടം 4 : A = L *B

ഘട്ടം 5 : P ,A പ്രിൻറ് ചെയ്യുക.

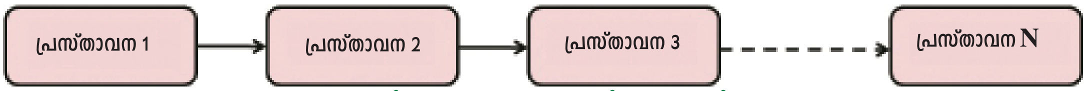
ഘട്ടം 6. അവസാനിപ്പിക്കുക.



ചിത്രം 3.6 വിസ്തീർണം ചുറ്റളവ് കാണാനുള്ള ഫ്ളോച്ചാർട്ട്

ചിത്രം 3 .6 ൽ ഇതിനുള്ള ഫ്ളോച്ചാർട്ട് നൽകിയിരിക്കുന്നു.

ഉദാഹരണം 4.1 ലും 4.2 ലും വികസിപ്പിച്ചെടുത്തിരിക്കുന്ന അൽഗോരിതങ്ങൾക്ക് ഓരോന്നിലും ആറു വീതം നിർദേശങ്ങളാണുള്ളത്. ചിത്രം 3.7 ൽ കാണിച്ചിരിക്കുന്നത് പോലെ ഈ രണ്ടു സന്ദർഭങ്ങളിലും നിർദേശങ്ങൾ ഓരോന്നും അനുക്രമമായ രീതിയിലാണ് പ്രവർത്തിക്കുക. നിർദേശങ്ങളുടെ പ്രവർത്തനക്രമത്തെ നിയന്ത്രണഗതി (Flow of Cotnrol) എന്ന് പറയുന്നു. അപ്രകാരം മുകളിൽപ്പറഞ്ഞ രണ്ട് അൽഗോരിതങ്ങളും അനുക്രമമായ നിയന്ത്രണ ഗതിയാണ് പിന്തുടരുന്നതെന്നു നമുക്ക് പറയാവുന്നതാണ്.



ചിത്രം 3.7 അനുക്രമമായ നിയന്ത്രണ ഗതി

നമുക്കു ചെയ്യാം

സെക്കന്റുകളായി സമയം ഇൻപുട്ട് ആയി സ്വീകരിച്ച് Hr:min:Sec രൂപത്തിൽ ലഭിക്കുവാനുള്ള അൽഗോരിതവും ഫ്ളോച്ചാർട്ടും വികസിപ്പിക്കുക. (ഉദാഹരണത്തിന് 3700 എന്ന് ഇൻപുട്ട് നൽകിയാൽ 1 Hr : 1 Min : 40 sec എന്ന ഔട്ട്പുട്ട് ലഭിക്കണം)

ഉദാഹരണം 3.3 രണ്ടു വിദ്യാർഥികളിൽ ഉയരം കൂടിയ ആളുടെ ഉയരം കണ്ടെത്തുക

ഇവിടെ,രണ്ടു വിദ്യാർത്ഥികളുടെ ഉയരത്തെ പ്രതിനിധീകരിക്കുന്ന രണ്ടു സംഖ്യകൾ ഇൻപുട്ട് ആയി സ്വീകരിക്കേണ്ടതാണ്. അവയിലെ വലിയ സംഖ്യയാണ് ഉത്തരമായിട്ടു പരിഗണിക്കുക. ഇതിനായി ഈ സംഖ്യകളെ താരതമ്യം ചെയ്യേണ്ടതുണ്ടെന്നു നമുക്കറിയാവുന്നതാണ്. അൽഗോരിതം താഴെ നൽകിയിരിക്കുന്നു.

ഘട്ടം 1 : തുടങ്ങുക

ഘട്ടം 2 : H1 ,H2 ഇൻപുട്ട് ആയി സ്വീകരിക്കുക

ഘട്ടം 3 : അഥവാ H1 >H2 ആണെങ്കിൽ

ഘട്ടം 4 : H1 പ്രിന്റ് ചെയ്യുക

ഘട്ടം 5 : അല്ലെങ്കിൽ

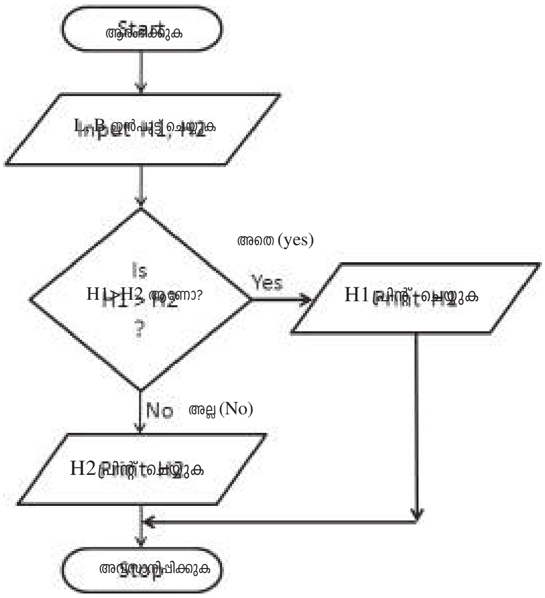
ഘട്ടം 6 : H2 പ്രിന്റ് ചെയ്യുക

ഘട്ടം 7 : പരിശോധന അവസാനിക്കുന്നു

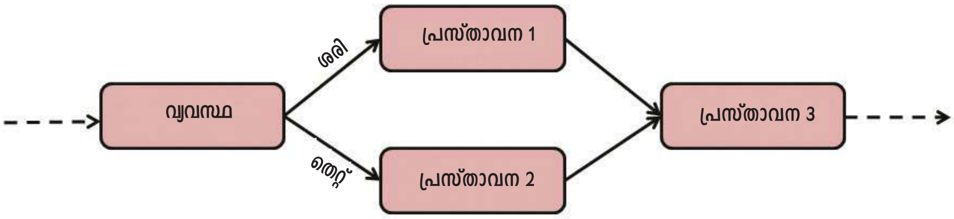
ഘട്ടം 8 : അവസാനിപ്പിക്കുക

ഈ അൽഗോരിതത്തിന്റെ ഫ്ളോച്ചാർട്ട് ചിത്രം 3.8 ൽ നൽകിയിരിക്കുന്നു. തീരുമാനങ്ങൾ കൈകൊള്ളാനുള്ള സങ്കേതം ഉപയോഗപെടുത്തുന്ന ഒരു അൽഗോരിതമാണിത്. ഘട്ടം 3 ൽ ഒരു

നിബന്ധന പരിശോധിക്കുന്നു. H1 ,H2 എന്നിവയുടെ വിലകളുടെ അടിസ്ഥാനത്തിൽ അതിന്റെ ഉത്തരം തീർച്ചയായും ശരി അല്ലെങ്കിൽ തെറ്റ് എന്നായിരിക്കും. തീരുമാനം കൈക്കൊള്ളുന്നത് ഈ നിബന്ധനയുടെ ഉത്തരത്തിനെ അടിസ്ഥാനമായിട്ടായിരിക്കും. ഉത്തരം ശരി എന്നാണെങ്കിൽ ഘട്ടം 4 പ്രവർത്തിക്കും അല്ലെങ്കിൽ ഘട്ടം 6 ആണ് പ്രവർത്തിക്കുക. ഇവിടെ ഒരു നിബന്ധനയുടെ അടിസ്ഥാനത്തിൽ രണ്ടു പ്രസ്താവനകളിൽ ഏതെങ്കിലും ഒന്നാണ് (ഘട്ടം 4 അല്ലെങ്കിൽ ഘട്ടം 6) പ്രവർത്തനത്തിനായി തിരഞ്ഞെടുക്കപ്പെടുക. ഘട്ടം 3 ൽ പ്രോഗ്രാം രണ്ടു ശാഖകളായി പിരിയുന്നു. അതായത് പ്രശ്നം പരിഹരിക്കുന്നതിന് വേണ്ടി ഈ അൽഗോരിതം ഒരു തിരഞ്ഞെടുക്കൽ ഘടന ഉപയോഗപ്പെടുത്തുന്നു. ചിത്രം 3.9 ൽ കാണിച്ചിരിക്കുന്നത് പോലെ നിബന്ധനയുടെ ഉത്തരത്തിനനുസരിച്ചു പ്രവർത്തനത്തിന്റെ ഗതി രണ്ടിൽ ഏതെങ്കിലും ഒരു പ്രസ്താവനയിലേക്ക് വിഘടിച്ചു പോകുന്നു.

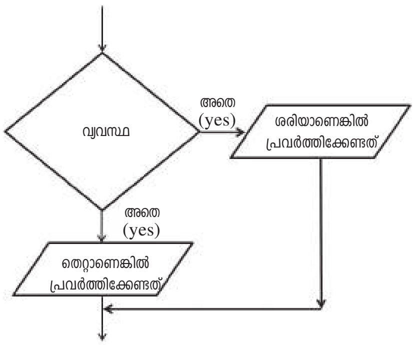


ചിത്രം 3.8 ഉയർന്ന വില കാണാനുള്ള ഫ്ളോച്ചാർട്ട്



ചിത്രം 3.9 തിരഞ്ഞെടുക്കൽ ഘടന

തിരഞ്ഞെടുക്കൽ നിർമ്മിതിയുടെ പ്രവർത്തനം ചിത്രം 3.10 ൽ കൊടുത്തിരിക്കുന്നു. നിയന്ത്രണ ഗതി നിബന്ധനയിലേക്കു വരികയും നിബന്ധന ശരി അല്ലെങ്കിൽ തെറ്റ് എന്ന് വിലയിരുത്തപ്പെടുകയും ചെയ്യുന്നു. നിബന്ധനയുടെ ഫലം ശരി എന്നാണെങ്കിൽ, ശരിയായാൽ പ്രവർത്തിക്കേണ്ട നിർദേശങ്ങളുടെ കൂട്ടം നടപ്പിലാക്കുകയും, തെറ്റായാൽ പ്രവർത്തിക്കേണ്ട നിർദേശങ്ങളുടെ കൂട്ടത്തെ ഒഴിവാക്കുകയും ചെയ്യുന്നു. നിബന്ധനയുടെ ഫലം തെറ്റ് ആണെങ്കിൽ തെറ്റായാൽ പ്രവർത്തിക്കേണ്ട നിർദേശങ്ങളുടെ കൂട്ടം നടപ്പിലാക്കുകയും ശരിയായാൽ പ്രവർത്തിക്കേണ്ട നിർദേശങ്ങളുടെ കൂട്ടാത്തെ ഒഴിവാക്കുകയും ചെയ്യുന്നു. ഇനി നമുക്ക് മറ്റൊരു പ്രശ്നം പരിഹരിക്കാം.



ചിത്രം 3.10 തിരഞ്ഞെടുക്കലിന്റെ ഫ്ളോച്ചാർട്ട്

ഉദാഹരണം 3.4 : 3 യൂണിറ്റ് ടെസ്റ്റുകളിൽ ലഭിച്ച സ്കോറുകൾ ഇൻപുട്ട് ചെയ്ത ഏറ്റവും ഉയർന്ന സ്കോർ കണ്ടെത്തുക

ഇവിടെ സ്കോറുകൾ സൂചിപ്പിക്കുന്നതിനു വേണ്ടി മൂന്നു സംഖ്യകൾ നൽകുകയും അവയിൽ ഏറ്റവും വലിയ സംഖ്യ കണ്ടുപിടിക്കുകയും ചെയ്യുന്നു. ഇതിന്റെ അൽഗോരിതം താഴെ കൊടുത്തിരിക്കുന്നു. ചിത്രം 3.11 ൽ ഫ്ളോച്ചാർട്ട് പ്രദർശിപ്പിച്ചിരിക്കുന്നു.

ഘട്ടം 1 : ആരംഭിക്കുക

ഘട്ടം 2 : M 1, M 2 , M 3 എന്നീ സംഖ്യകൾ ഇൻപുട്ട്

ആരംഭിക്കുക

ചെയ്യുക .

ഘട്ടം 3 : അഥവാ M 1 > M 2 ഉം M 1 > M 3 ആണെങ്കിൽ

ഘട്ടം 4 : M 1 പ്രിന്റ് ചെയ്യുക

ഘട്ടം 5 : അല്ലെങ്കിൽ M 2 >M 3 ആണെങ്കിൽ

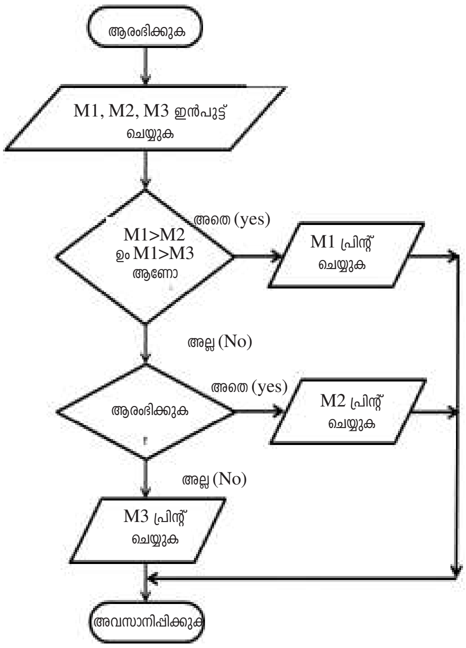
ഘട്ടം 6 : M 2 പ്രിന്റ് ചെയ്യുക

ഘട്ടം 7 : അല്ലെങ്കിൽ

ഘട്ടം 8 : M 3 പ്രിൻറ് ചെയ്യുക.

ഘട്ടം 9 : വ്യവസ്ഥാ പരിശോധനയുടെ അവസാനം

ഘട്ടം 10 : അവസാനിപ്പിക്കുക



ചിത്രം 3.11 മൂന്ന് സംഖ്യകളിൽ ഏറ്റവും വലിയ സംഖ്യ കാണാനുള്ള ഫ്ളോച്ചാർട്ട്

വ്യത്യസ്തമായ നിബന്ധനകളുടെ അടിസ്ഥാനത്തിൽ അൽഗോരിതത്തിൽ ഒന്നിലധികം തിരഞ്ഞെടുക്കൽ നിർമിതകൾ ഉപയോഗിക്കുന്നു. ഇവിടെ വ്യത്യസ്തമായ മൂന്നു പ്രവൃത്തികൾ നൽകിയിരിക്കുന്നു. എന്നാൽ അവയിൽ ഒന്ന് മാത്രമേ പ്രവർത്തിക മാകുകയുള്ളൂ. ശ്രദ്ധിക്കേണ്ട മറ്റൊരു വസ്തുത, ഇവിടെ ആദ്യത്തെ നിബന്ധനയിൽ രണ്ടു താര തമ്യങ്ങൾ അടങ്ങിയിരിക്കുന്നു എന്നതാണ്. ഇത്തരം നിബന്ധനകൾ സംയുക്ത നിബന്ധനകൾ (Compound conditions) എന്ന് പറയുന്നു.

നമുക്കു ചെയ്യാം

1. *തന്നിരിക്കുന്ന സംഖ്യ ഒറ്റയാണോ ഇരട്ടയാണോ എന്ന് പരിശോധിക്കാനുള്ള അൽഗോരിതം തയാറാക്കുക. ഫ്ളോച്ചാർട്ട് വരയ്ക്കുക.*
2. *ദിവസത്തെ സൂചിപ്പിക്കുന്ന സംഖ്യ ഇൻപുട്ട് ആയി നൽകിയാൽ ദിവസത്തിന്റെ പേര് പ്രദർശിപ്പിക്കാനുള്ള അൽഗോരിതവും ഫ്ളോച്ചാർട്ടും തയാറാക്കുക. (ഉദാഹരണത്തിന് 1 ഇൻപുട്ട് നൽകിയാൽ ഔട്ട്പുട്ട് Sunday എന്നായിരിക്കണം. അഥവാ 2 ആണ് ഇൻപുട്ടെങ്കിൽ ഔട്ട്പുട്ട് Monday എന്നായിരിക്കണം. 1 മുതൽ 7 വരെയുള്ള സംഖ്യ അല്ല ഇൻപുട്ടെങ്കിൽ 'INVALID DATA' എന്നായിരിക്കേണ്ടതാണ്.*
3. *പത്താം തരത്തിലെ മൂല്യനിർണയ വ്യവസ്ഥയുടെ അടിസ്ഥാനത്തിൽ ഒരു സ്കോർ (പരമാവധി 100) സ്വീകരിച്ചു ഗ്രേഡ് കാണാനുള്ള അൽഗോരിതം തയാറാക്കുക.*

ഒരു പ്രവൃത്തി തന്നെ ആവർത്തിച്ചു നിർവഹിക്കേണ്ട ഒരു സാഹചര്യം പരിഗണിക്കുക. ഉദാഹരണത്തിന് ആദ്യത്തെ 100 എണ്ണൽ സംഖ്യകൾ പ്രിൻറ് ചെയ്യണമെന്നിരിക്കട്ടെ എങ്ങനെ യാണ് നമുക്കത് ചെയ്യാൻ സാധിക്കുക? നമുക്കറിയാം ആദ്യത്തെ സംഖ്യ 1 ആണ് .അത് പ്രിന്റ് ചെയ്യേണ്ടതാണ്.ആദ്യത്തെ സംഖ്യയോട് 1 കൂട്ടിയാൽ അടുത്ത സംഖ്യ ലഭിക്കുന്നു. അതും പ്രിന്റ് ചെയ്യണം. ഇതിൽ നിന്ന് ഒരു കാര്യം വ്യക്തമാണ്, സംഖ്യ പ്രിൻറ് ചെയ്യുക സംഖ്യയോടു 1 കൂട്ടുക എന്നീ പ്രവൃത്തികൾ ആവർത്തിച്ച് ചെയ്യേണ്ടവയാണ്. അവസാനത്തെ സംഖ്യ പ്രിൻറ് ചെയ്തു കഴിഞ്ഞാൽ പ്രവർത്തനം അവസാനിപ്പിക്കേണ്ടതാണ് .ഇതിനു വേണ്ടിയുള്ള ഒരു അൽഗോരിതം നമുക്ക് തയാറാക്കാം.

ഉദാഹരണം 3.5: 1 മുതൽ 100 വരെയുള്ള സംഖ്യകൾ പ്രിന്റ് ചെയ്യാൻ

ഘട്ടം 1 : ആരംഭിക്കുക

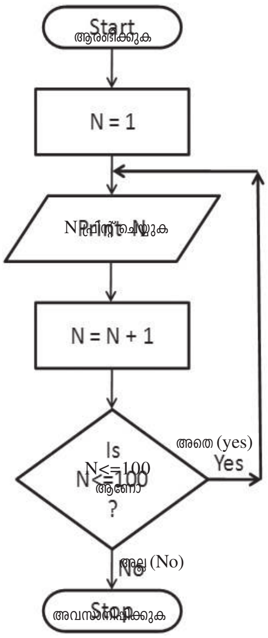
ഘട്ടം 2 : N =1

ഘട്ടം 3 : N പ്രിന്റ് ചെയ്യുക

ഘട്ടം 4 : N =N + 1

ഘട്ടം 5 : അഥവാ N <=100 ആണെങ്കിൽ ഘട്ടം 3 ലേക്ക് പോകുക

ഘട്ടം 6 : അവസാനിപ്പിക്കുക



ചിത്രം 3.12 : 1 മുതൽ 100 വരെയുള്ള സംഖ്യകൾ പ്രിന്റ് ചെയ്യാനുള്ള ഫ്ളോച്ചാർട്ട്

ഉദാഹരണം 3.5ൽ കൊടുത്തിരിക്കുന്ന അൽഗോരിതത്തിൽ ഘട്ടം 5 ൽ ഒരു വ്യവസ്ഥ പരിശോധിക്കുന്നു. അഥവാ വ്യവസ്ഥ ശരി യാണെങ്കിൽ നിയന്ത്രണ ഗതി ഘട്ടം 3 ലേക്ക് തിരിച്ചു പോകുന്നു. അതു കാരണം വ്യവസ്ഥ ശരിയായിരിക്കുന്നതു വരെ ഘട്ടം 3, ഘട്ടം 4, ഘട്ടം 5 എന്നിവ ആവർത്തിച്ചു പ്രവർത്തിച്ചു കൊണ്ടി രിക്കും. ഇവിടെ ഒരു ലൂപ്പ് രൂപം കൊണ്ടതായി നമുക്ക് പറയാം. ഘട്ടം 3,4,5 എന്നിവ ചേർന്നതാണ് ആ ലൂപ്പ്. വ്യവസ്ഥ തെറ്റാകു മ്പോൾ മാത്രമെ നിയന്ത്രണം ലൂപ്പിനു പുറത്തേക്കു വരുകയുള്ളൂ. ഈ അൽഗോരിതത്തിന്റെ ഫ്ളോച്ചാർട്ട് ചിത്രം 3.12 ൽ കാണിച്ചിരി ക്കുന്നു.

മുകളിൽപ്പറഞ്ഞ അൽഗോരിതത്തെ താഴെ പറയും പ്രകാരം ലഘൂകരിക്കാം

ഘട്ടം 1 : ആരംഭിക്കുക

ഘട്ടം 2 : N = 1

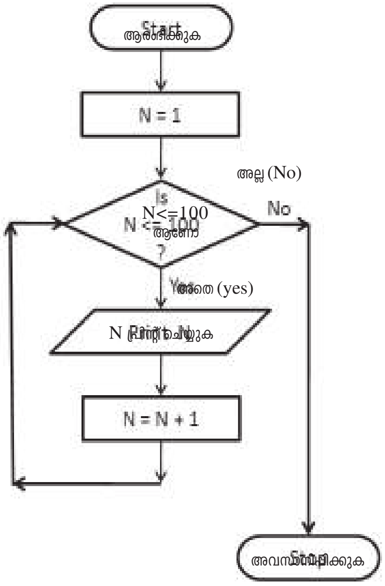
ഘട്ടം 3 : N <100 ആയിരിക്കുന്നത് വരെ ഘട്ടം 4 ഉം 5 ഉം ആവർത്തിക്കുക

ഘട്ടം 4 : N പ്രിന്റ് ചെയ്യുക

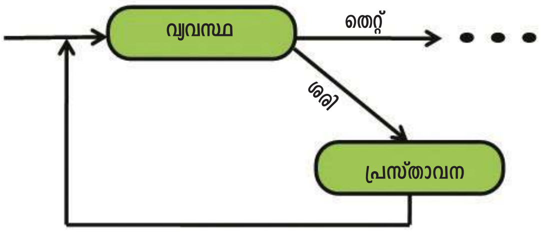
ഘട്ടം 5 : N = N + 1

ഘട്ടം 6 : അവസാനിപ്പിക്കുക

ഘട്ടം 3 ൽ 'ആവർത്തിക്കുക', 'ആയിരിക്കുന്നത് വരെ' മുതലായ വാക്കുകൾ ലൂപ്പ് നിർമ്മിക്കാൻ ഉപയോഗിക്കുന്നു. ആവർത്തിച്ചു പ്രവർത്തിക്കേണ്ട പ്രസ്താവനകൾ 'ആവർത്തിക്കുക' എന്ന വാക്കിന്റെ കൂടെ പ്രസ്താവിക്കുകയും പരിശോധിക്കേണ്ട വ്യവസ്ഥ 'ആയിരിക്കുന്നത് വരെ' എന്ന വാക്കിന്റെ കൂടെയും നൽകുന്നു. അൽഗോരിതം വ്യത്യസ്തമായിരിക്കുന്നത് പോലെ ചിത്രം 3.13 ൽ കാണിച്ചിരിക്കുന്ന ഫ്ളോച്ചാർട്ടും അല്പം വ്യത്യസ്തമായിരിക്കും. ചിത്രം 3.14 ൽ ലൂപ്പിന്റെ പ്രവർത്തന ശൈലി കാണിച്ചിരിക്കുന്നു.

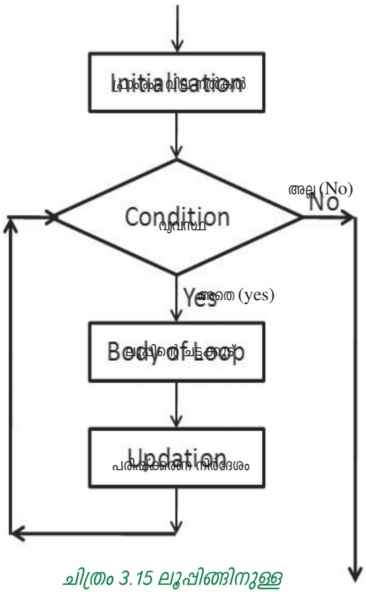


ചിത്രം 3.13: 1 മുതൽ 100 വരെയുള്ള സംഖ്യകൾ പ്രിന്റ് ചെയ്യാനുള്ള ഫ്ളോച്ചാർട്ട്



ചിത്രം 3 .14: ലൂപ്പിംഗ് നിർമിതി

ഒരു ലൂപ്പിനു നാല് ഘടകങ്ങൾ ഉണ്ട്. സ്വാഭാവികമായും അവയിൽ ഒന്ന് വ്യവസ്ഥ (Condition) തന്നെ. വ്യവസ്ഥ നൽകുന്നതിനു വേണ്ടി ഒരു വേരിയബിളെങ്കിലും ഉപയോഗിക്കണം എന്നു നമുക്കറിയാം. ഇതിനെ ലൂപ്പ് നിയന്ത്രണ വേരിയബിൾ എന്ന് വിളിക്കാം. വ്യവസ്ഥ പരിശോധിക്കുന്നതിന് മുമ്പ് ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിന് ഒരു വില ലഭ്യമാക്കേണ്ടതാണ്. ഇൻപുട്ട് അല്ലെങ്കിൽ വില നൽകൽ (Assignment) വഴി ഇത് സാധ്യമാകുന്നതാണ്. അത്തരത്തിലുള്ള നിർദേശങ്ങളെ ലൂപ്പിന്റെ പ്രാരംഭ വില നൽകൽ നിർദേശങ്ങൾ (Initialization Instructions) എന്ന് പറയുന്നു. പരിഷ്കരണ നിർദേശം (Update Instruction) എന്ന മൂന്നാമത്തെ ഘടകമാണ് ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിന്റെ വില മാറ്റുന്നത്. ഇത് വളരെ അത്യാവശ്യമാണ്. എന്തെന്നാൽ പരിഷ്കരണ നിർദേശം ഇല്ലെങ്കിൽ ലൂപ്പിന്റെ പ്രവർത്തനം ഒരിക്കലും അവസാനിക്കില്ല. ആവർത്തിച്ചു പ്രവർത്തിക്കേണ്ട ഒരു കൂട്ടം നിർദേശങ്ങളാണ് ലൂപ്പിന്റെ ചട്ടക്കൂട് (Body of the Loop) എന്ന നാലാമത്തെ ഘടകം. ചിത്രം 3.15 ൽ കാണിച്ചിരിക്കുന്ന ഫ്ളോച്ചാർട്ട് ലൂപ്പിന്റ ഘടന വ്യക്തമാക്കുന്നു.



ചിത്രം 3.15 ലൂപ്പിങ്ങിനുള്ള ഫ്ളോച്ചാർട്ട്

പ്രാരംഭ വില നൽകൽ നിർദേശമാണ് ആദ്യം പ്രവർത്തിക്കുക. ശേഷം വ്യവസ്ഥ പരിശോധിക്കുന്നു. വ്യവസ്ഥ ശരിയാണെങ്കിൽ ലൂപ്പിന്റെ ചട്ടക്കൂടും തുടർന്ന് പരിഷ്കരണ നിർദേശവും പ്രവർത്തിക്കുന്നു. പരിഷ്കരണ നിർദേശം പ്രവർത്തിച്ചു കഴിഞ്ഞാൽ വീണ്ടും വ്യവസ്ഥ പരിശോധിക്കുന്നു. വ്യവസ്ഥ തെറ്റാകുന്നത് വരെ ഈ പ്രക്രിയ തുടരുന്നു. ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രാവർത്തികമാകുന്നതിനു മുമ്പ് വ്യവസ്ഥ പരിശോധിക്കുന്ന ലൂപ്പിനെ ആഗമന നിയന്ത്രിത ലൂപ്പ് (entry cotnrolled Loop) എന്ന് വിളിക്കുന്നു. മറ്റൊരു രീതിയിലുള്ള ലൂപ്പിംഗ് നിർമ്മിതിയും നിലവിലുണ്ട്. അതിൽ ലൂപ്പിന്റെ ചട്ടക്കൂടും പരിഷ്കരണ നിർദേശവും പ്രവർത്തിച്ചു കഴിഞ്ഞ ശേഷം മാത്രമേ വ്യവസ്ഥ പരിശോധിക്കുകയുള്ളൂ. ഇത്തരത്തിലുള്ള ലൂപ്പിനെ നിർഗമന നിയന്ത്രിത ലൂപ്പ് (Exit Cotnrolled Loop) എന്ന് വിളിക്കുന്നു.

ഉദാഹരണം 3.6: ആദ്യത്തെ N എണ്ണൽ സംഖ്യകളുടെ തുക കണ്ടെത്തുക

ഇവിടെ N ന്റെ വില നമ്മൾ ഇൻപുട്ട് ആയി നൽകുന്നു. 1 മുതൽ N വരെയുള്ള സംഖ്യകളുടെ തുക കണ്ടുപിടിക്കേണ്ടതാണ്. തുക സംഭരിക്കുന്നതിനായി 'S' എന്ന വേരിയബിളാണെന്നിരിക്കട്ടെ, ചിത്രം 3.16 ൽ ഈ അൽഗോരിതത്തിൻറെ ഫ്ളോച്ചാർട്ട് കാണിച്ചിരിക്കുന്നു.

ഘട്ടം 1 : തുടങ്ങുക

ഘട്ടം 2 : N ലേക്ക് വില ഇൻപുട്ട് ആയി സ്വീകരിക്കുക

ഘട്ടം 3 : A=1 ,S=0

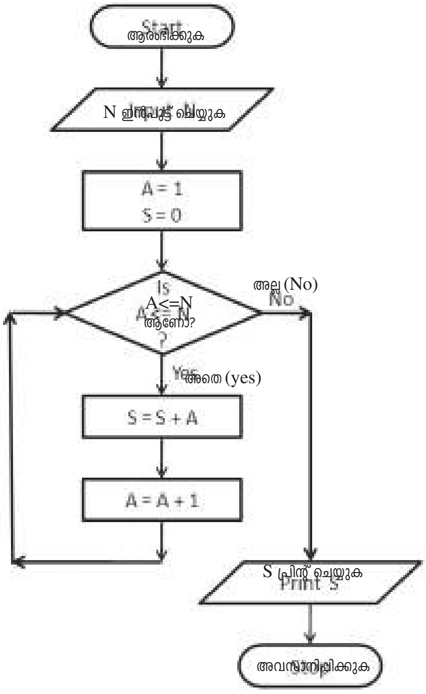
ഘട്ടം 4 : (A<=N) ആയിരിക്കുന്നത് വരെ ഘട്ടം 5 ഉം 6 ഉം ആവർത്തിക്കുക

ഘട്ടം 5 : S =S +A

ഘട്ടം 6 : A =A +1

ഘട്ടം 7 : S പ്രിന്റ് ചെയ്യുക.

ഘട്ടം 8 : അവസാനിപ്പിക്കുക



ചിത്രം 3.16 ആദ്യത്തെ N എണ്ണൽ സംഖ്യകളുടെ ആകെത്തുക കാണാനുള്ള ഫ്ളോച്ചാർട്ട്

ഈ അൽഗോരിതം ഒരു ആഗമന നിയന്ത്രിത ലൂപ്പ് ഉപയോഗിക്കുന്നു. അടുത്ത ഉദാഹരണത്തിൽ പ്രശ്നപരിഹാരത്തിനായി ഒരു നിർഗമന നിയന്ത്രിത ലൂപ്പ് ഉപയോഗിക്കുന്ന അൽഗോരിതം കാണാം.

ഉദാഹരണം 3.7 : തന്നിരിക്കുന്ന സംഖ്യയുടെ ആദ്യത്തെ 10 ഗുണിതങ്ങൾ കണ്ടെത്തുക

ഘട്ടം 1 : തുടങ്ങുക

ഘട്ടം 2 : N ഇൻപുട്ട് ആയി സ്വീകരിക്കുക

ഘട്ടം 3 : A=1

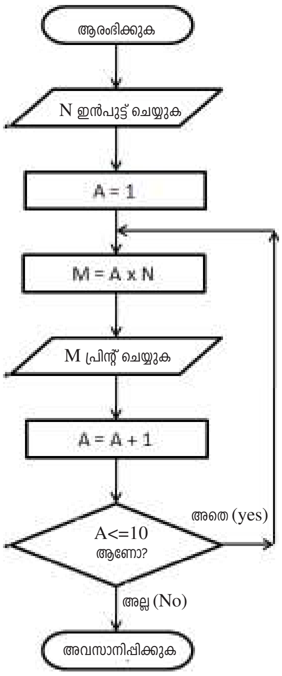
ഘട്ടം 4 : M =A x N

ഘട്ടം 5 : M പ്രിന്റ് ചെയ്യുക

ഘട്ടം 6 : A=A+1

ഘട്ടം 7 : (A<=10) ആയിരിക്കുന്നത് വരെ ഘട്ടം 4 മുതൽ 5 വരെ ആവർത്തിക്കുക

ഘട്ടം 8 : അവസാനിപ്പിക്കുക



ചിത്രം 3.17 തന്നിരിക്കുന്ന സംഖ്യയുടെ ആദ്യത്തെ 10 ഗുണിതങ്ങൾ കണ്ടുപിടിക്കാനുള്ള ഫ്ളോച്ചാർട്ട്

ഈ അൽഗോരിതത്തിന് അനുസൃതമായ ഫ്ളോച്ചാർട്ട് ചിത്രം 3.17 ൽ നൽകിയിരിക്കുന്നു. ഇവിടെ ലൂപ്പ് ചട്ടക്കൂട് പ്രവർത്തിച്ച ശേഷം മാത്രം പരിശോധിക്കപ്പെടുന്ന ഒരു വ്യവസ്ഥ അടങ്ങിയ ഒരു ലൂപ്പ് ഉപയോഗിച്ചിരിക്കുന്നു. ആഗമന നിയന്ത്രിത ലൂപ്പും നിർഗമന നിയന്ത്രിത ലൂപ്പും തമ്മിലുള്ള താരതമ്യം പട്ടിക 3.1 ൽ കാണിച്ചിരിക്കുന്നു.

ആഗമന നിയന്ത്രിത ലൂപ്	നിർഗമന നിയന്ത്രിത ലൂപ്
• ലൂപ് ചട്ടക്കൂട് പ്രാവർത്തികമാക്കുന്നതിനു മുമ്പ് വ്യവസ്ഥ പരിശോധിക്കുന്നു	• ലൂപ് ചട്ടക്കൂട് പ്രാവർത്തികമാക്കിയതിന് ശേഷം വ്യവസ്ഥ പരിശോധിക്കുന്നു
• ലൂപ് ചട്ടക്കൂട് ഒരിക്കൽ പോലും പ്രവർത്തിച്ചില്ല എന്ന് വരാം	• ലൂപ് ചട്ടക്കൂട് കുറഞ്ഞത് ഒരു തവണ എങ്കിലും നിർബന്ധമായും പ്രവർത്തിച്ചിരിക്കും
• ലൂപ് ചട്ടക്കൂട് പ്രാവർത്തികമാക്കുന്നതിൽ നിന്നും ഒഴിവാക്കണമെങ്കിൽ ഇത് ഉപയോഗിക്കുന്നു.	• ചട്ടക്കൂടിന്റെ സാധാരണ നിലയിലുള്ള പ്രവർത്തനം ഉറപ്പുവരുത്തണമെങ്കിൽ ഉപയോഗിക്കുന്നു

പട്ടിക 3.1: ലൂപ്പുകളുടെ താരതമ്യം

പഠന പ്രവർത്തനങ്ങളുടെ ഭാഗമായി നൽകിയിരിക്കുന്ന പ്രശ്നപരിഹാരത്തിനുള്ള അൽഗോരിതങ്ങളും ഫ്ളോച്ചാർട്ടും ഉപയോഗിച്ച് നമുക്ക് പരിശീലിക്കാം

നമുക്കു ചെയ്യാം

താഴെ പറയുന്ന പ്രശ്നങ്ങൾക്കു വേണ്ടി അൽഗോരിതവും ഫ്ളോച്ചാർട്ടും തയാറാക്കുക

1. 100 ൽ താഴെയുള്ള എല്ലാ ഇരട്ട സംഖ്യകളും അവരോഹണ ക്രമത്തിൽ പ്രിന്റ് ചെയ്യുക
2. 100 നും 200 നും ഇടയിലുള്ള ഒറ്റ സംഖ്യകളുടെ തുക കണ്ടുപിടിക്കുക
3. തന്നിരിക്കുന്ന സംഖ്യയുടെ ഗുണനപട്ടിക പ്രിൻറ് ചെയ്യുക
4. ഒരു സംഖ്യയുടെ ഫാക്ടോറിയൽ (FACTORIAL) കണ്ടുപിടിക്കുക
5. ഒരു സംഖ്യ ഇൻപുട്ട് ചെയ്ത് അവിഭാജ്യ സംഖ്യയാണോ എന്ന് പരിശോധിക്കുക

3.3.3 പ്രോഗ്രാം കോഡ് തയാറാക്കൽ (Coding the Program)

അൽഗോരിതവും ഫ്ളോച്ചാർട്ടും രൂപകൽപന ചെയ്തു കഴിഞ്ഞാൽ പ്രോഗ്രാമിങിന്റെ അടുത്ത പടി നിർദേശങ്ങൾ സൂക്ഷ്മവും സംക്ഷിപ്തവുമായ പ്രതീകങ്ങൾ ഉപയോഗിച്ച് ആവിഷ്കരിക്കുക എന്നതാണ്. അതായത് നിർദേശങ്ങൾ പ്രോഗ്രാമിങ് ഭാഷയിൽ ആവിഷ്കരിക്കുന്നു. പ്രശ്ന പരിഹാരത്തിന് വേണ്ടി ഇത്തരം പ്രോഗ്രാം നിർദേശങ്ങൾ എഴുതുന്നതിനെയാണ് കോഡിങ് എന്ന് പറയുന്നത്. കമ്പ്യൂട്ടറിൽ കോഡ് എഴുതുന്നതിനായി ടെക്സ്റ്റ് എഡിറ്റർ പ്രോഗ്രാമുകൾ ലഭ്യമാണ്.

ആശയവിനിമയത്തിനുള്ള ഒരു സംവിധാനമാണ് ഭാഷ. ഇംഗ്ലീഷ് മലയാളം മുതലായ സ്വാഭാവിക ഭാഷകൾ ഉപയോഗിച്ച് നാം നമ്മുടെ ആശയങ്ങളും വികാരങ്ങളും പരസ്പരം പങ്കുവയ്ക്കുന്നു .

അത് പോലെ ഉപയോക്താവിനും കമ്പ്യൂട്ടറിനും ഇടയിൽ ആശയവിനിമയം നടത്താൻ പ്രോഗ്രാമിങ് ഭാഷ ഉപയോഗിക്കുന്നു. കമ്പ്യൂട്ടർ പ്രോഗ്രാം എഴുതുന്ന വ്യക്തിക്ക് കംപ്യൂട്ടറിനു മനസ്സിലാകുന്ന ഭാഷയും പരിചിതമായിരിക്കണം. മനുഷ്യർക്ക് മനസ്സിലാക്കാനും ഉപയോഗിക്കാനും ബുദ്ധിമുട്ടുള്ള ദ്വയാങ്ക ഭാഷ (BInary Language) മാത്രമേ കമ്പ്യൂട്ടറിന് അറിയുകയുള്ളൂ എന്നത് നമ്മൾ നേരത്തെ കണ്ടതാണ്. അതിനാൽ അധ്യായം മൂന്നിൽ പഠിച്ചതു പോലെ ഇംഗ്ലീഷ് ഭാഷയ്ക്ക് സമാനമായതും മനുഷ്യർക്ക് സഹൃദയമായതുമായ ഉയർന്നതല ഭാഷ (High Level Language) (HLL) നമുക്ക് ഉപയോഗിക്കാവുന്നതാണ്. ഉയർന്നതല ഭാഷയിൽ എഴുതിയിരിക്കുന്ന പ്രോഗ്രാമുകളെ യന്ത്ര ഭാഷയിലേക്കു വിവർത്തനം അല്ലെങ്കിൽ മൊഴിമാറ്റം ചെയ്യാൻ ലാംഗ്വേജ് പ്രോസസ്സർ ഉപയോഗിക്കുന്നു. ഉയർന്നതല ഭാഷയിൽ എഴുതിയിരിക്കുന്ന പ്രോഗ്രാം, സോഴ്സ് കോഡ് (Source Code) എന്നറിയപ്പെടുന്നു.

ഒരു പ്രോഗ്രാമർ ആകണമെങ്കിൽ പ്രോഗ്രാമിലെ നിർദേശങ്ങൾ ആവിഷ്ക്കരിക്കുന്നതിനു വേണ്ടിയുള്ള ബേസിക് (BASIC), കോബോൾ (COBOL), പാസ്കൽ (PASCAL), C++ തുടങ്ങിയ ഉയർന്നതല ഭാഷകളിൽ ഏതിലെങ്കിലും ഒന്നിൽ നൈപുണ്യം കൈവരിക്കേണ്ടതാണ്. പ്രോഗ്രാമുകൾ തയാറാക്കുന്നതിന് വേണ്ടി ഓരോ പ്രോഗ്രാമിങ് ഭാഷയും അതിന്റേതായ അക്ഷരമാലകളും (Charecter Set), ശബ്ദകോശങ്ങളും (Voccabulary), വ്യാകരണങ്ങളും (Grammar) (നമുക്കതിനെ വാക്യ ഘടന (Syntax) എന്ന് വിളിക്കാം) ഉണ്ടായിരിക്കും.

ഈ ഭാഷ ഉപയോഗിച്ച് പ്രോഗ്രാം തയ്യാറാക്കി കഴിഞ്ഞാൽ അത് ഒരു ഫയലിൽ (സോഴ്സ് ഫയൽ) സൂക്ഷിക്കുകയും പ്രോഗ്രാമിങ്ങിന്റെ അടുത്ത ഘട്ടത്തിലേക്ക് കടക്കുകയും ചെയ്യുന്നു.

3.3.4 പരിഭാഷ (Translation)

സോഴ്സ് കോഡ് വികസിപ്പിക്കുന്നതിനായി ഭാഷ തിരഞ്ഞെടുക്കുമ്പോൾ ഉപയോഗിക്കുന്ന ഡാറ്റയുടെ അളവ്, പ്രവർത്തനത്തിന്റെ സങ്കീർണ്ണത, ഫയലുകളുടെ ഉപയോഗം മുതലായ ചില മാനദണ്ഡങ്ങൾ പരിഗണിക്കേണ്ടതുണ്ട്. പ്രോഗ്രാമിങ് ഭാഷ തിരഞ്ഞെടുക്കുകയും സോഴ്സ് കോഡ് തയ്യാറാക്കുകയും ചെയ്തു കഴിഞ്ഞാൽ അതിനെ ലാംഗ്വേജ് പ്രോസെസ്സറിൻറെ സഹായത്തോടെ പരിഭാഷ ചെയേണ്ടതാണ്. ഉയർന്നതല ഭാഷയിൽ എഴുതപ്പെട്ടിരിക്കുന്ന പ്രോഗ്രാമിനെ അതിനു തുല്യമായ യന്ത്രഭാഷാ പതിപ്പിലേക്കു രൂപഭേദം വരുത്തുന്ന പ്രക്രിയയാണ് പരിഭാഷ (Translation). കംപൈലർ അല്ലെങ്കിൽ ഇൻറെർപ്രെറ്റർ ഇതിനായി ഉപയോഗിക്കുന്നു. പ്രോഗ്രാമിലുള്ള സിന്റാക്സ് എററുകൾ (Syntax Error) ഈ ഘട്ടത്തിൽ ദൃശ്യമാകുന്നു. സോഴ്സ് കോഡ് അടങ്ങിയ ഫയൽ തുറന്നു ഈ തെറ്റുകൾ തിരുത്തേണ്ടതാണ്. അതിനു ശേഷം സോഴ്സ് കോഡ് വീണ്ടും കമ്പൈൽ ചെയ്യാനായി (പരിഭാഷ) നൽകുന്നു. "No Errors or Warnings" അല്ലെങ്കിൽ "Sucessfull Compilation" എന്ന സന്ദേശം ലഭിക്കുന്നത് വരെ ഈ പ്രക്രിയ തുടർന്ന് കൊണ്ടിരിക്കും. പൂർണമായും യന്ത്രഭാഷയിലെ നിർദേശങ്ങൾ അടങ്ങിയ പ്രോഗ്രാം നമുക്കിപ്പോൾ ലഭ്യമാകുന്നു.സോഴ്സ് കോഡിൻറെ ഈ പതിപ്പ് ഒബ്ജക്റ്റ് കോഡ് (Object Code) എന്ന് അറിയപ്പെടുന്നു. കംപൈലർ തന്നെ ഈ ഒബ്ജക്റ്റ് കോഡിനെ മറ്റൊരു ഒരു ഫയലിൽ സൂക്ഷിക്കുകയും ചെയ്യുന്നു

ചിത്രം 3.18 പരിഭാഷാ പ്രവർത്തനം

ഇങ്ങനെ ഒബ്ജക്റ്റ് കോഡ് ലഭിച്ചു കഴിഞ്ഞാൽ പ്രോഗ്രാം ഉപയോഗിക്കുന്നിടത്തോളം കാലം ഈ കോഡ് ആ കമ്പ്യൂട്ടറിൽ തന്നെ ഉണ്ടായിരിക്കേണ്ടതാണ്.

3.3.5 ഡീബഗ്ഗിങ് (Debugging)

പ്രോഗ്രാമിങ്ങിലുള്ള തെറ്റുകൾ കണ്ടെത്തുകയും അവ തിരുത്തുകയും ചെയ്യുന്ന ഘട്ടമാണ് ഡീബഗ്ഗിങ്.മനുഷ്യർ കമ്പ്യൂട്ടറുകളെ പ്രോഗ്രാം ചെയ്യുന്നിടത്തോളം കാലം പ്രോഗ്രാമുകളിൽ തെറ്റുകൾ സംഭവിക്കാം. പ്രോഗ്രാമിങ്ങിലുള്ള തെറ്റുകളെ 'ബഗ്ഗ്' എന്ന് പറയുന്നു.തെറ്റുകൾ കണ്ടുപിടിക്കുകയും തിരുത്തുകയും ചെയ്യുന്ന പ്രക്രിയയെ 'ഡീബഗ്ഗിങ്' എന്ന് വിളിക്കുന്നു. പൊതുവെ പ്രോഗ്രാമിങ്ങിൽ രണ്ടു തരത്തിലുള്ള തെറ്റുകളാണ് വന്നുകൂടുക സിന്റാക്സ് എററുകളും ലോജിക്കൽ തെറ്റുകളും. പ്രോഗ്രാമിങ് ഭാഷയുടെ നിയമങ്ങൾ അല്ലെങ്കിൽ വാക്യഘടന പാലിക്കാത്തതു കൊണ്ട് സംഭവിക്കുന്ന തെറ്റുകളെയാണ് സിന്റാക്സ് എറർ എന്ന് വിളിക്കുന്നത്. തെറ്റായ ചിഹ്നങ്ങൾ ഉപയോഗിക്കുക, തെറ്റായ വാക്യപ്രയോഗം, നിർവചിക്കപ്പെടാത്ത പദങ്ങളുടെ ഉപയോഗം, നിയമവിരുദ്ധമായ വാക്യരചന അല്ലെങ്കിൽ പദങ്ങളുടെ ഉപയോഗം മുതലായ കാരണങ്ങൾ

കൊണ്ടാണ് സാധാരണ അത്തരം തെറ്റുകൾ സംഭവിക്കുക. പരിഭാഷക്കു വേണ്ടി പ്രോഗ്രാം നൽകി കഴിഞ്ഞാൽ തന്നെ ലാംഗ്വേജ് പ്രൊസസ്സറുകൾ സിൻറ്റാക്സ് തെറ്റുകൾ കണ്ടെത്തുന്നു. തെറ്റുകൾ സംഭവിച്ചിരിക്കുന്ന പ്രസ്താവനകളുടെ ലൈൻ നമ്പറുകൾ അതിനോടൊപ്പം സംഭവിച്ചിരിക്കുന്ന തെറ്റിനെ കുറിച്ചുള്ള സൂചനകളും അവ നൽകുന്നു. എറർ മെസ്സേജുകളായി പ്രദർശിപ്പിക്കുന്നു. ഇൻറ്റർപ്രെറ്ററുകളുടെ കാര്യത്തിൽ, പ്രവർത്തന ഘട്ടത്തിലാണ് സിൻറ്റാക്സ് തെറ്റുകൾ കണ്ടു പിടിച്ച് പ്രദർശിപ്പിക്കുന്നത്. ഡീബഗ്ഗിങ് പ്രക്രിയക്കായി വേണ്ടി വരുന്ന സമയവും പരിശ്രമവുമാണ് ഒരു പ്രോഗ്രാമിങ് ഭാഷ ഉപയോഗിക്കുന്നതിൽ പ്രോഗ്രാമർക്കുള്ള പ്രാപ്തി എത്രത്തോളമുണ്ട് എന്ന് തീരുമാനിക്കുന്നത്. എല്ലാ സിൻറ്റാക്സ് തെറ്റുകളും തിരുത്തിക്കഴിഞ്ഞാൽ മാത്രമേ ഒബ്ജക്റ്റ് പ്രോഗ്രാം നിർമ്മിക്കപ്പെടുകയുള്ളു.

പ്രോഗ്രാമിങ് യുക്തിയുടെ ആസൂത്രണത്തിലുള്ള അപാകതകൾ കാരണമാണ് ലോജിക്കൽ എറർ എന്ന് പേരുള്ള രണ്ടാമത്തെ തരം തെറ്റുകൾ സംഭവിക്കുന്നത്.സിൻറ്റാക്സ് എററുകൾ ഇല്ലെങ്കിൽ ലാംഗ്വേജ് പ്രൊസസ്സറുകൾ വിജയകരമായി സോഴ്സ് കോഡിനെ പരിഭാഷപ്പെടുത്തുന്നു. പ്രോഗ്രാമിന്റെ പ്രവർത്തന ഘട്ടത്തിൽ കമ്പ്യൂട്ടർ, പ്രോഗ്രാം നിർദേശങ്ങൾ പിന്തുടരുകയും അവക്ക് അനുസൃതമായ ഔട്ട്പുട്ട് നൽകുകയും ചെയ്യുന്നു. പക്ഷെ ഈ ഔട്ട്പുട്ട് ശരിയായിരിക്കണമെന്നില്ല. ഇതിനെയാണ് ലോജിക്കൽ എറർ എന്ന് പറയുന്നത്. ലോജിക്കൽ എറർ സംഭവിക്കുമ്പോൾ പ്രോഗ്രാം തെറ്റായ ഔട്ട്പുട്ട് ആണ് തരുന്നത് എന്ന് മാത്രമേ നമുക്ക് ഗ്രഹിക്കാൻ സാധിക്കുകയുള്ളു.എന്താണ് തെറ്റ് എന്ന് കമ്പ്യൂട്ടർ നമ്മോടു പറയുന്നില്ല. പ്രോഗ്രാമർ അല്ലെങ്കിൽ ഉപയോക്താവാണ് അത് കണ്ടെത്തേണ്ടത്. ലോജിക്കൽ തെറ്റുകൾ ഉണ്ടോ ഇല്ലയോ എന്നറിയുന്നതിനായി പ്രോഗ്രാം പരീക്ഷിക്കപ്പെടേണ്ടതാണ്. അതിനായി പ്രോഗ്രാമിങ്ങിന്റെ അടുത്ത ഘട്ടത്തിലേക്ക് നമുക്ക് കടക്കാം.

3.3.6 പ്രവർത്തനവും പരീക്ഷണവും (Execution and Testing)

മുകളിൽ പറഞ്ഞത് പോലെ ലോജിക്കൽ തെറ്റുകൾ കൂടി തിരുത്തിയാൽ മാത്രമേ ഒരു പ്രോഗ്രാം തെറ്റുകളിൽ നിന്നും മുക്തമാണ് എന്ന് നമുക്ക് പറയാൻ കഴിയൂ. ആയതിനാൽ കംപൈൽ ചെയ്യപ്പെട്ട പ്രോഗ്രാമിന്റെ പതിപ്പ് പരീക്ഷണത്തിനായി പ്രവർത്തിപ്പിക്കേണ്ടതാണ്. ശരിയായ ഫലങ്ങൾ ലഭ്യമാകുന്നുണ്ടോ എന്ന് പരിശോധിക്കുകയാണ് പരീക്ഷണത്തിന്റെ ഉദ്ദേശ്യം. 'അറിയാവുന്ന ഫലങ്ങൾ' ലഭിക്കുന്നതിന് വേണ്ടിയുള്ള പരീക്ഷണ ഡാറ്റ നൽകി പ്രോഗ്രാം പ്രാവർത്തികമാക്കുക എന്ന പ്രക്രിയയാണ് പരീക്ഷണ ഘട്ടത്തിൽ ഉൾപ്പെട്ടിരിക്കുന്നത്. അതായത് പ്രോഗ്രാമിൽ ഉൾപ്പെട്ടിരിക്കുന്ന ക്രിയകൾ മനുഷ്യൻ തന്നെ ചെയ്യുകയും, ലഭ്യമാകുന്ന ഔട്ട്പുട്ട് കമ്പ്യൂട്ടർ മുഖേന ചെയ്യുമ്പോൾ ലഭിക്കുന്ന ഔപുട്ടുമായി താരതമ്യം ചെയ്യുകയും വേണം. പ്രോഗ്രാം യുക്തിയുടെ കൃത്യത ഈ പരീക്ഷണ ഘട്ടം കൊണ്ട് നിർണയിക്കാവുന്നതാണ്. പരീക്ഷണത്തിനായുള്ള ഡാറ്റ തിരഞ്ഞെടുക്കുമ്പോൾ പ്രോഗ്രാം യുക്തിയുടെ എല്ലാ വശങ്ങളും പരീക്ഷിക്കപ്പെടുന്നുണ്ട് എന്ന് ഉറപ്പാക്കേണ്ടതാണ്. അതിനാൽ തന്നെ ഉചിതമായ ഡാറ്റ തിരഞ്ഞെടുക്കുക എന്നത് പ്രോഗ്രാം പരീക്ഷണത്തിൽ വളരെ പ്രാധാന്യമുള്ളതാണ്.

തെറ്റായ യുക്തി മൂലം ലഭിക്കുന്ന തെറ്റായ ഔട്ട്പുട്ടുകളെ കുറിച്ചാണ് നാം ഇതുവരെ ചർച്ച ചെയ്തു കൊണ്ടിരുന്നത്. എന്നാൽ പ്രോഗ്രാം പ്രവർത്തനത്തെ തടസ്സപ്പെടുത്താവുന്ന മറ്റൊരു തരം തെറ്റ് സംഭവിക്കാൻ സാധ്യതയുണ്ട്. ഒരു ക്രിയയിൽ അനുചിതമായ ഡാറ്റ വരുന്നത് മൂലം സംഭവിക്കാവുന്ന ഒന്നാണത്.ഉദാഹരണത്തിന്, A =B / C എന്ന നിർദേശം പരിഗണിക്കുക. Cയുടെ വില പൂജ്യം ആയാൽ ഈ പ്രസ്താവന പ്രോഗ്രാമിന്റെ പ്രവർത്തനത്തെ തടസ്സപ്പെടുത്തുന്നു (പൂജ്യം കൊണ്ടുള്ള ഹരണം മൂലം). ഇത്തരം സാഹചര്യങ്ങളിൽ പ്രോഗ്രാമിംഗ് ഭാഷയിലുള്ള തെറ്റുകൾ കൈകാര്യം ചെയ്യുന്ന ഫങ്ക്ഷനുകൾ (Error handling function) എറർ മെസ്സേജുകൾ പ്രദർശിപ്പിക്കുന്നു.ഇത്തരം തെറ്റുകളെ റൺ ടൈം എറർ എന്ന് വിളിക്കുന്നു.ഡാറ്റ പ്രോസസ്സ് ചെയ്യപ്പെടുന്നതിനു മുമ്പ് ഡാറ്റയുടെ സാധുത പരിശോധിക്കാനുള്ള അനുബന്ധ നിർദേശങ്ങൾ പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തുക വഴി ഇത്തരം തെറ്റുകൾ തിരുത്താവുന്നതാണ്.

3.3.7 വിവരണം തയ്യാറാക്കൽ (Documentation)

ഉചിതമായ രീതിയിയിൽ വിവരണം തയ്യാറാക്കാത്ത ഒരു കംപ്യൂട്ടർവൽകൃത സംവിധാനം ഒരിക്കലും പൂർണമാണെന്നു പറയാൻ നമുക്ക് കഴിയില്ല. വാസ്തവത്തിൽ, പ്രശ്നപഠന ഘട്ടം മുതൽ അത് പ്രയോഗത്തിൽ വരുത്തുന്നത് വരെ തുടർച്ചയായി നടന്നുകൊണ്ടിരിക്കുന്ന ഒരു പ്രക്രിയയാണ് വിവരണം തയ്യാരാക്കൽ. ഇതിന്റെ ഭാഗമായി സോഴ്സ് കോഡിൽ നമുക്ക് കമെന്റുകൾ ഉൾപ്പെടുത്താവുന്നതാണ്. ഇത് ആന്തരിക വിവരണം (Internal Documentation) എന്നറിയപ്പെടുന്നു. ഡീബഗ്ഗിങ് ഘട്ടത്തിലും പിൽക്കാലത്ത് പ്രോഗ്രാമിൽ വരുന്ന മാറ്റങ്ങൾ ഉൾപ്പെടുത്താനും ഇത് സഹായിക്കുന്നു. പ്രോഗ്രാം നിർമാണ സമയത്ത് ഉപയോഗിച്ച യുക്തി പിന്നീട് നമ്മുടെ തന്നെ പ്രോഗ്രാമിലൂടെ കടന്നു പോകുമ്പോൾ നമുക്ക് ഓർമയുണ്ടായിരിക്കണമെന്നില്ല. മാത്രമല്ല ചില സാഹചര്യങ്ങളിൽ ഒരു വ്യക്തി എഴുതിയ പ്രോഗ്രാം മറ്റൊരു വ്യക്തിക്ക് ഭാവിയിൽ മാറ്റേണ്ടതായി വരാം. ഒരു പ്രോഗ്രാമിൽ കൃത്യമായി വിവരണം തയ്യാറാക്കിയാൽ നമ്മൾ ഉപയോഗിച്ച യുക്തി മനസ്സിലാക്കാനും പ്രോഗ്രാമിൽ ഒരു പ്രസ്താവന എന്ത് കൊണ്ടാണ് ഉപയോഗിച്ചിരിക്കുന്നത് എന്ന് മനസ്സിലാക്കാനും സാധിക്കുന്നു. എന്നിരുന്നാലും പ്രോഗ്രാം പരിഭാഷക്കായി നൽകുമ്പോൾ ലാംഗ്വേജ് പ്രൊസസ്സറുകൾ പ്രോഗ്രാമിന്റെ വിവരണ ഭാഗങ്ങൾ പരിഭാഷക്കായി പരിഗണിക്കുകയില്ല.

പ്രോഗ്രാമിന്റെ ഭാഗമായ കമെന്റുകൾ വിവരണം തയ്യാറാക്കുന്ന ഘട്ടത്തിന്റെ ഒരു ഭാഗം മാത്രമാണ്. സിസ്റ്റം മാനുവൽ, ഉപയോക്തൃ മാനുവൽ എന്നിവ തയ്യാറാക്കുക എന്നത് വിവരണം തയ്യാറാക്കുന്നതിന്റെ മറ്റൊരു പ്രക്രിയയാണ്. കമ്പ്യൂട്ടർ വ്യവസ്ഥയുടെ പ്രവർത്തനം, അവയുടെ ആവശ്യകത, പ്രോഗ്രാമുകൾ ഇൻസ്റ്റാൾ ചെയ്യുക, അവ ഉപയോഗിക്കുന്ന രീതികൾ എന്നിവ ഉൾപ്പെടുന്ന ഹാർഡ് കോപ്പികളാണിവ. വിവിധ ആവശ്യങ്ങൾക്ക് വേണ്ടിയുള്ള സോഫ്റ്റ്‌വെയറുകൾ തയ്യാറാക്കുമ്പോൾ ഇത്തരം മാനുവലുകൾ നിർബന്ധമാണ്. ഇത്തരം വിവരണം തയ്യാറാക്കുന്നതിനെയാണ് ബാഹ്യമായ വിവരണം (External Documentation) എന്ന് പറയുന്നത്.

ഇപ്പോൾ നാം ഒരു പ്രശ്നത്തെ വിശകലനം ചെയ്യുകയും പരിഹാരത്തിനുള്ള യുക്തി കണ്ടെത്തുകയും, ഫ്ളോ ചാർട്ട് രൂപത്തിൽ പ്രതിപാദിക്കുകയും, പ്രോഗ്രാമിങ് ഭാഷയിൽ കോഡ് തയ്യാറാക്കുകയും, സിന്റാക്സിലെ പിശകുകൾ നീക്കം ചെയ്ത ശേഷം പരിഭാഷപ്പെടുത്തുകയും ചെയ്തു. കൂടാതെ ലോജിക്കൽ തെറ്റുകളും റൺ ടൈം തെറ്റുകളും നീക്കം ചെയ്ത് ശേഷം ഔട്ട്പുട്ടിന്റെ കൃത്യത പരിശോധിക്കുകയും അവസാനമായി പ്രോഗ്രാമിന്റെ വിവരണം തയ്യാറാക്കുകയും ചെയ്തു.

സ്വയം വിലയിരുത്താം

1. അൽഗോരിതം എന്നാൽ എന്താണ് ?
2. അൽഗോരിതത്തിന്റെ ചിത്ര ആവിഷ്കരണമാണ് __________.
3. ഏതു ഫ്ളോച്ചാർട്ട് ചിഹ്നമാണ് എപ്പോഴും ജോഡികളായി ഉപയോഗിക്കുന്നത്?
4. ഏതു ഫ്ളോച്ചാർട്ട് ചിഹ്നത്തിനാണ് ഒരു ആഗമന മാർഗവും രണ്ടോ അതിലധികമോ നിർഗമന മാർഗങ്ങളും ഉള്ളത് ?
5. HLL ൽ എഴുതിയിരിക്കുന്ന പ്രോഗ്രാം _________ എന്ന് അറിയപ്പെടുന്നു.
6. ഡീബഗ്ഗിങ് എന്നാലെന്ത് ?
7. ഒബ്ജക്റ്റ് കോഡ് എന്നാലെന്ത് ?

3.4 അൽഗോരിതങ്ങളുടെ പ്രകടനം വിലയിരുത്തൽ (Performance evaluation of algorithms)

വിവിധ പ്രശ്നങ്ങൾ പരിഹരിക്കുന്നതിനായി നമ്മൾ അൽഗോരിതങ്ങൾ തയാറാക്കിയിട്ടുണ്ട്. ചിലപ്പോൾ ചില പ്രശ്നങ്ങൾ പരിഹരിക്കുന്നതിന് വ്യത്യസ്തമായ യുക്തി പ്രയോഗിക്കാമായിരുന്നു എന്ന് നമുക്ക് തോന്നാം. ഒരേ പ്രശ്നം തന്നെ വ്യത്യസ്തമായ ഒരു കൂട്ടം നിർദേശങ്ങൾ ഉപയോഗിച്ച് പരിഹരിക്കാവുന്നതാണ്. എന്നിരുന്നാലും വളരെ കുറഞ്ഞ കമ്പ്യൂട്ടർ വിഭവങ്ങൾ ഉപയോഗപ്പെടുത്തി, കുറഞ്ഞ സമയം കൊണ്ട് വളരെ കൃത്യമായ ഫലം നൽകുന്ന പ്രോഗ്രാം തയ്യാറാക്കുന്നയാളാണ് ഒരു സമർത്ഥനായ പ്രോഗ്രാമർ എന്ന് പറയുന്നത്. ഒരു അൽഗോരിതത്തിന്റെ പ്രകടനം അളക്കുന്നത് ടൈം കോംപ്ലക്സിറ്റി (Time complexity), സ്പേസ് കോംപ്ലക്സിറ്റി (Space complexity) എന്നിവയുടെ അടിസ്ഥാനത്തിലാണ്. ഏറ്റവും കുറവ് മെമ്മറി ഉപയോഗപ്പെടുത്തി ഏറ്റവും വേഗത്തിൽ പ്രവർത്തിക്കുന്ന അൽഗോരിതത്തെയാണ് പ്രശ്ന പരിഹാരത്തിനുള്ള ഏറ്റവും നല്ല അൽഗോരിതം ആയി കണക്കാക്കുന്നത്.

അൽഗോരിതം 1		അൽഗോരിതം 2	
ഘട്ടം 1 :	തുടങ്ങുക	ഘട്ടം 1 :	തുടങ്ങുക
ഘട്ടം 2 :	A ,B ,C ഇൻപുട്ട് ചെയ്യുക	ഘട്ടം 2 :	A ,B ,C ഇൻപുട്ട് ചെയ്യുക
ഘട്ടം 3 :	S =A +B +C	ഘട്ടം 3 :	S=A+B+C
ഘട്ടം 4 :	AVG =S / 3	ഘട്ടം 4 :	AVG =(A +B +C) / 3
ഘട്ടം 5 :	S, AVG പ്രിന്റ് ചെയ്യുക	ഘട്ടം 5 :	S, AVG പ്രിന്റ് ചെയ്യുക
ഘട്ടം 6 :	അവസാനിപ്പിക്കുക	ഘട്ടം 6 :	അവസാനിപ്പിക്കുക

പട്ടിക 3 .2 മൂന്നു സംഖ്യകളുടെ തുകയും ശരാശരിയും കാണാനുള്ള രണ്ടു അൽഗോരിതങ്ങൾ

മൂന്നു സംഖ്യകളുടെ തുകയും ശരാശരിയും കാണാനുള്ള രണ്ടു അൽഗോരിതങ്ങൾ പട്ടിക 3.2ൽ കൊടുത്തിരിക്കുന്നു. അവയെ നമുക്ക് താരതമ്യം ചെയ്യാം. രണ്ടു അൽഗോരിതങ്ങളും ഘട്ടം 4 ൽ വ്യത്യാസപ്പെട്ടിരിക്കുന്നു. അൽഗോരിതം2 ൽ ഒരേ ഡാറ്റയിൽ രണ്ടു തവണ സങ്കലനം ചെയ്യുന്നു (ഘട്ടം 3, ഘട്ടം 4).സ്വാഭാവികമായിട്ടും ഈ അൽഗോരിതം അൽഗോരിതം1 നെക്കാൾ കൂടുതൽ സമയം പ്രവർത്തനത്തിനായി ഉപയോഗിക്കുന്നു. അതിനാൽ കോഡിങ്ങിനു നല്ലതു അൽഗോരിതം 1 ആകുന്നു.

ഒരു പ്രസ്താവന തിരഞ്ഞെടുക്കുന്നതിന് വേണ്ടി താരതമ്യ ക്രിയകൾ ഉൾപ്പെടുന്ന മറ്റൊരു ഉദാഹരണം നമുക്കു പരിശോധിക്കാം. മൂന്നു സംഖ്യകളിൽ വലുത് ഏതാണെന്നു കണ്ടെത്താനുള്ള

അൽഗോരിതം ഉദാഹരണം 3.4 ൽ നമ്മൾ ചർച്ച ചെയ്യുകയുണ്ടായി. അതേ പ്രശ്നം പരിഹരിക്കുന്നതിനുള്ള രണ്ടു അൽഗോരിതങ്ങൾ പട്ടിക 4.3 ൽ നൽകിയിരിക്കുന്നു.

ഉദാഹരണം 3.3 ൽ മൂന്ന് താരതമ്യ ക്രിയകളും ഒരു ലോജിക്കൽ ക്രിയയും ഉപയോഗിച്ചിരിക്കുന്നു. ഏറ്റവും വലിയ വില M3 ൽ (മൂന്നാമത്തെ വേരിയബിൾ) വരുമ്പോളാണ് ഈ ക്രിയകൾ എല്ലാം പ്രവർത്തിക്കുക. പ്രവർത്തനത്തിന്റെ വേഗത അളക്കുന്നതിനായി താരതമ്യ ക്രിയക്ക് 1 സെക്കന്റ് സമയമെടുക്കുന്നു എന്ന് നമുക്ക് അനുമാനിക്കാം. അങ്ങനെയാണെങ്കിൽ ഏറ്റവും വേഗം കൂടിയ ഫലം 3 സെക്കന്റ് വേഗം കുറവുള്ള ഫലം 4 സെക്കന്റ് വേണ്ടി വരുന്നതായി കാണാം. ശരാശരി വേഗം 3.5 സെക്കന്റ് ആയിരിക്കും.

അൽഗോരിതം 1		അൽഗോരിതം 2	
ഘട്ടം 1 :	തുടങ്ങുക	ഘട്ടം 1 :	തുടങ്ങുക
ഘട്ടം 2 :	M 1 ,M 2 ,M 3 ഇൻപുട്ട് ആയി സ്വീകരിക്കുക	ഘട്ടം 2 :	M 1, M 2, M 3 ഇൻപുട്ട് ആയി സ്വീകരിക്കുക
ഘട്ടം 3 :	അഥവാ M 1 >M 2 ഉം M 1 > M 3 ഉം ആണെങ്കിൽ	ഘട്ടം 3 :	അഥവാ M 1 >M 2 ആണെങ്കിൽ
ഘട്ടം 4 :	M 1 പ്രിന്റ് ചെയ്യുക	ഘട്ടം 4 :	Big = M 1
ഘട്ടം 5 :	അഥവാ M 2 >M 1 ഉം M 2 > M 3 ഉം ആണെങ്കിൽ	ഘട്ടം 5 :	അല്ലെങ്കിൽ
ഘട്ടം 6 :	M 2 പ്രിന്റ് ചെയ്യുക	ഘട്ടം 6 :	Big = M 2
ഘട്ടം 7 :	അഥവാ M 3 >M 1 ഉം M 3 > M 2 ഉം ആണെങ്കിൽ	ഘട്ടം 7 :	അഥവാ M 3 >Big ആണെങ്കിൽ Big = M 3
ഘട്ടം 8 :	M3 പ്രിന്റ് ചെയ്യുക	ഘട്ടം 8 :	Big പ്രിന്റ് ചെയ്യുക
ഘട്ടം 9 :	അവസാനിപ്പിക്കുക	ഘട്ടം 9 :	അവസാനിപ്പിക്കുക

പട്ടിക 3.3 മൂന്നു സംഖ്യകളിൽ ഏറ്റവും വലുത് കണ്ടെത്താനുള്ള അൽഗോരിതങ്ങൾ

ഇനി നമുക്ക് പട്ടിക 3.3 ലുള്ള അൽഗോരിതം1 വിശകലനം ചെയ്യാം. അതിൽ മൂന്ന് താരതമ്യ ക്രിയകൾ അടങ്ങിയ മൂന്ന് If (അഥവാ) പ്രസ്താവനകൾ അടങ്ങിയിരിക്കുന്നു. മുകളിൽ പ്രസ്താവിച്ചിട്ടുള്ള അനുമാനം പിന്തുടരുകയാണെങ്കിൽ വേരിയബിലിന്റെ വില ഏതാണെങ്കിലും നമുക്ക് ഫലം 9 സെക്കന്റിൽ ലഭിക്കുന്നതാണ്. അപ്പോൾ ശരാശരി വേഗം എന്ന് പറയുന്നത്തു 9 സെക്കന്റ് ആണ്. പക്ഷെ പട്ടിക 3.3 ൽ പറയുന്ന അൽഗോരിതം2ൽ രണ്ടു If (അഥവാ) പ്രസ്താവനകൾ ഉപയോഗിച്ചിരിക്കുന്നു. അൽഗോരിതം2 ൽ വേരിയബിളുകളിൽ ഏതു വില വന്നാലും താരതമ്യം ചെയ്യാൻ വേണ്ട സമയം 2 സെക്കന്റ് ആയിരിക്കും. അപ്പോൾ ശരാശരി വേഗം എന്ന് പറയുന്നത് 2 സെക്കന്റ് ആയിരിക്കും.എന്നു പറഞ്ഞാൽ അൽഗോരിതം 2 മറ്റു രണ്ടു അൽഗോരിതങ്ങളെക്കാൾ മെച്ചപ്പെട്ടതാണ്.

ലൂപ്പുകൾ അടങ്ങിയ മറ്റൊരു ഉദാഹരണം കൂടി നമുക്ക് നോക്കാം. 100 നും 200 നും ഇടയിലുള്ള എല്ലാ ഒറ്റ സംഖ്യകളുടെ തുകയും എല്ലാ ഇരട്ട സംഖ്യകളുടെ തുകയും കണ്ടുപിടിക്കാനുള്ള രണ്ടു അൽഗോരിതങ്ങൾ പട്ടിക 3.4 ൽ കൊടുത്തിരിക്കുന്നു.

അൽഗോരിതം 1	അൽഗോരിതം 2
ഘട്ടം 1 : തുടങ്ങുക	ഘട്ടം 1 : തുടങ്ങുക
ഘട്ടം 2 : N =100, ES =0	ഘട്ടം 2 : N =100 ,ES =0,OS =0
ഘട്ടം 3 : (N <=200) ആയിരിക്കുന്നതു വരെ ഘട്ടം 4 മുതൽ 6 വരെ ആവർത്തിക്കുക	ഘട്ടം 3 : (N <=200) ആയിരിക്കുന്നതു വരെ ഘട്ടം 4 മുതൽ 8 വരെ ആവർത്തിക്കുക
ഘട്ടം 4 : അഥവാ N/2 ന്റെ ശിഷ്ടം = 0 ആണെങ്കിൽ	ഘട്ടം 4 : അഥവാ N/2 ന്റെ ശിഷ്ടം = 0 ആണെങ്കിൽ
ഘട്ടം 5 : ES =ES + N	ഘട്ടം 5 : ES =ES + N
ഘട്ടം 6 : N =N +1	ഘട്ടം 6 : അല്ലെങ്കിൽ
ഘട്ടം 7 : ES പ്രിന്റ് ചെയ്യുക	ഘട്ടം 7 : OS =OS + N
ഘട്ടം 8 : N =100, OS =0	ഘട്ടം 8 : N =N +1
ഘട്ടം 9 : (N <=200) ആയിരിക്കുന്നതു വരെ ഘട്ടം 10 മുതൽ 12 വരെ ആവർത്തിക്കുക	ഘട്ടം 9 : ES പ്രിന്റ് ചെയ്യുക
ഘട്ടം 10 : അഥവാ N/2 ന്റെ ശിഷ്ടം = 1 ആണെങ്കിൽ	ഘട്ടം 10 : OS പ്രിന്റ് ചെയ്യുക
ഘട്ടം 11 : OS =OS + N	ഘട്ടം 11 : അവസാനിപ്പിക്കുക
ഘട്ടം 12 : N =N +1	
ഘട്ടം 13 : OS പ്രിന്റ് ചെയ്യുക	
ഘട്ടം 14 : അവസാനിപ്പിക്കുക	

പട്ടിക 3.4 ഒറ്റ സംഖ്യകളുടെയും ഇരട്ട സംഖ്യകളുടെയും തുക കാണാനുള്ള അൽഗോരിതങ്ങൾ

അൽഗോരിതം1 രണ്ടു ലൂപ്പുകളും അൽഗോരിതം2 ഒരു ലൂപ്പും ഉപയോഗിക്കുന്നു. സ്വാഭാവികമായിട്ടും അൽഗോരിതം2 നെ അപേക്ഷിച്ച് അൽഗോരിതം1 ന് പ്രാരംഭ വിലനൽകാനും, പരിശോധനയ്ക്ക് വേണ്ടിയും ലൂപ്പ് വേരിയബിൾ പുതുക്കുന്നതിന് വേണ്ടിയും മറ്റും ഇരട്ടി സമയം ആവശ്യമായി വരുന്നു. അൽഗോരിതം2 മെച്ചപ്പെട്ടതും കാര്യക്ഷമവുമാണ് എന്ന് പട്ടികയിൽ നിന്ന് തന്നെ വ്യക്തമാണ്. അതിനാൽ തന്നെ പ്രശ്ന പരിഹാരത്തിനുള്ള യുക്തി വികസിപ്പിക്കുന്നതിനു മുമ്പ് വ്യത്യസ്തവും വിഭിന്നവുമായി ചിന്തിക്കേണ്ടതു അനിവാര്യമാണ്.

നമുക്ക് സംഗ്രഹിക്കാം

ഒരു കമ്പ്യൂട്ടർ ഭാഷയിൽ ക്രമത്തിൽ എഴുതിയിരിക്കുന്ന നിർദേശങ്ങളാണ് പ്രോഗ്രാം. പ്രോഗ്രാമിങ് പ്രക്രിയ ചില ഘട്ടങ്ങളിലൂടെ കടന്നു പോകുന്നു. അൽഗോരിതവും ഫ്ളോച്ചാർട്ടും തയ്യാറാക്കുന്നത് പ്രോഗ്രാമിങ്ങിന്റെ യുക്തി വികസിപ്പിക്കാൻ സഹായിക്കുന്നു. HLL ൽ തയ്യാറാക്കിയിരിക്കുന്ന പ്രോഗ്രാമിനെ സോഴ്സ് കോഡ് എന്ന് വിളിക്കുന്നു. അതിനെ യന്ത്ര ഭാഷയിലേക്കു പരിഭാഷപ്പെടുത്തേണ്ടതാണ്. അതിന്റെ ഫലമായി ലഭിക്കുന്ന കോഡ്, ഒബ്ജക്റ്റ് കോഡ് എന്ന് അറിയപ്പെടുന്നു. ഡീബഗ്ഗിങ് എന്ന് വിളിക്കുന്ന പ്രക്രിയയിലൂടെ പ്രോഗ്രാമിൽ സംഭവിച്ചിരിക്കുന്ന തെറ്റുകളെ നീക്കം ചെയ്യുന്നു. പരിഭാഷപ്പെടുത്തിയ പതിപ്പ് കമ്പ്യൂട്ടർ പ്രവർത്തിപ്പിക്കുന്നു. ഉചിതമായ വിവരണം തയ്യാറാക്കുന്നതിന് പിൽക്കാലത്തു പ്രോഗ്രാമിൽ മാറ്റം വരുത്തുന്നതിന് സഹായകമാകുന്നു. പ്രശ്ന പരിഹാരത്തിന് വ്യത്യസ്തങ്ങളായ യുക്തികൾ പ്രയോഗിക്കാമെങ്കിലും പ്രോഗ്രാമിന്റെ പ്രകടനം അളക്കുന്നത് ടൈം കോംപ്ലക്സിറ്റിയുടെയും സ്പേസ് കോംപ്ലക്സിറ്റിയുടെയും അടിസ്ഥാനത്തിലാണ്.

പഠന നേട്ടങ്ങൾ

ഈ അദ്ധ്യായം പൂർത്തിയാകുന്നതോടെ പഠിതാവിന്

- പ്രശ്ന പരിഹാരത്തിന്റെ വിവിധ വശങ്ങൾ വിശദീകരിക്കാൻ സാധിക്കുന്നു.
- പ്രശ്നങ്ങൾ പരിഹരിക്കാനുള്ള അൽഗോരിതങ്ങൾ വികസിപ്പിക്കാൻ സാധിക്കുന്നു.
- അൽഗോരിതത്തിൽ കൃത്യത ഉറപ്പു വരുത്താൻ ഫ്ളോച്ചാർട്ടുകൾ വരയ്ക്കാൻ സാധിക്കുന്നു.
- പ്രശ്നം പരിഹരിക്കാനുള്ള ഏറ്റവും നല്ല അൽഗോരിതം തിരഞ്ഞെടുക്കാൻ സാധിക്കുന്നു.

മാതൃക ചോദ്യങ്ങൾ

ഒറ്റ വാക്യത്തിൽ ഉത്തരം എഴുതുക

1. അൽഗോരിതം എന്നാലെന്ത് ?
2. പ്രശ്ന പരിഹാരത്തിൽ കമ്പ്യൂട്ടറിന്റെ പങ്ക് എന്താണ് ?
3. ഫ്ളോച്ചാർട്ടിൽ കണക്റ്ററിന്റെ ഉപയോഗമെന്ത് ?
4. പ്രോഗ്രാമിൽ ലോജിക്കൽ തെറ്റുകൾ എന്നാലെന്ത് ?

ലഘു വിവരണാത്മകം

1. കമ്പ്യൂട്ടർ പ്രോഗ്രാം എന്നാലെന്ത് ? പ്രോഗ്രാമുകൾ തയ്യാറാക്കുന്നതിന് അൽഗോരിതങ്ങൾ എങ്ങനെ സഹായിക്കുന്നു ?
2. 3 സംഖ്യകളുടെ തുകയും ശരാശരിയും കണ്ടുപിടിക്കാനുള്ള അൽഗോരിതം എഴുതുക
3. ആദ്യത്തെ 100 എണ്ണൽ സംഖ്യകൾ പ്രദർശിപ്പിക്കാനുള്ള ഫ്ളോച്ചാർട്ട് വരയ്ക്കുക
4. ഫ്ളോച്ചാർട്ടുകളുടെ പരിമിതികൾ എന്തെല്ലാം?
5. ഡീബഗ്ഗിങ് എന്നാലെന്ത്?
6. ഒരു പ്രോഗ്രാമിൽ വിവരണം തയ്യാറാക്കുന്നതിന്റെ ആവശ്യകത എന്ത്?

വിവരണാത്മകം

1. അൽഗോരിതത്തിന്റെ സവിശേഷതകൾ എന്തെല്ലാം?
2. ഫ്ളോച്ചാർട്ട് ഉപയോഗിക്കുന്നത് കൊണ്ടുള്ള ഗുണങ്ങൾ എന്തെല്ലാം?
3. പ്രോഗ്രാമിങിന്റെ വിവിധ ഘട്ടങ്ങളെ പറ്റി ചുരുക്കത്തിൽ വിവരിക്കുക

4

C++ പ്രോഗ്രാമിങ് - ഒരു ആമുഖം

പ്രധാന ആശയങ്ങൾ

- C++ -ലെ ക്യാരക്ടർ സെറ്റ്
- ടോക്കണുകൾ
  - കീവേഡുകൾ
  - ഐഡന്റിഫയറുകൾ
  - ലിറ്ററലുകൾ
  - പങ്ച്വേറ്ററുകൾ
  - ഓപ്പറേറ്ററുകൾ
- ഇന്റഗ്രേറ്റഡ് ഡവലപ്മെന്റ് എൻവയോൺമെന്റ് (IDE)
  - ജിനി (IDE)

W9X5J1

Bjarne Stroupstrup വികസിപ്പിച്ച ശക്തവും ജനപ്രിയവുമായ ഒരു ഒബ്ജക്ട് ഓറിയന്റഡ് പ്രോഗ്രാമിങ് ഭാഷയാണ് C++ (ഉച്ചരിക്കുന്നത് c പ്ലസ് പ്ലസ്). C++ എന്ന ആശയം വന്നത് C-യോടൊപ്പം + + ഓപ്പറേറ്റർ കൂടി ചേർന്നാണ്. അങ്ങനെ C++ എന്നത് C ഭാഷയുടെ ഒരു വിപുലീകരിച്ച രൂപമായിത്തീർന്നു.

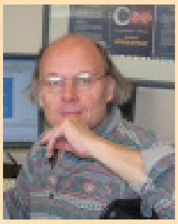
നാം കഴിഞ്ഞ അധ്യായത്തിൽ ചർച്ച ചെയ്ത ക്രമപ്പെടുത്തൽ, തിരഞ്ഞെടുക്കൽ, ആവർത്തനം തുടങ്ങിയ വിവിധ തരം പ്രോഗ്രാമിങ് ആശയങ്ങൾ പ്രാവർത്തികമാക്കാൻ C++ ഭാഷ ഉപയോഗിക്കുന്നു. ഈ അധ്യായത്തിൽ, C++ -ന്റെ അടിസ്ഥാന ആശയങ്ങളെക്കുറിച്ച് ഒരു അവലോകനം നടത്താം. C++ പ്രോഗ്രാം എഴുതുന്നതിനുള്ള വിവിധ ഭാഷാ പ്രൊസസ്സർ പാക്കേജുകളും നമുക്ക് പരിചയപ്പെടാം.

ഏതൊരു ഭാഷയേയും പോലെ, C++ ഭാഷാപഠനവും തുടങ്ങുന്നത് അതിലെ അടിസ്ഥാന ചിഹ്നങ്ങളായ അക്ഷരങ്ങൾ പരിചയപ്പെട്ടുകൊണ്ടാണ്. അതിനുശേഷം വാക്കുകൾ, വാക്യങ്ങൾ (എക്സ്പ്രഷനുകൾ), പ്രസ്താവനകൾ തുടങ്ങിയവയിലൂടെ പഠനക്രമം തുടരും. അക്ഷരങ്ങൾ പഠിച്ച് കൊണ്ട് നമുക്ക് തുടങ്ങാം.

4.1 ക്യാരക്ടർ സെറ്റ് (Character set)

നമുക്ക് അറിയാവുന്നത് പോലെ ഇംഗ്ലീഷ്, മലയാളം, ഹിന്ദി തുടങ്ങിയ ഏതൊരു ഭാഷാപഠനവും ആരംഭിക്കുന്നത് അക്ഷരമാലയിലാണ്. അതുപോലെ C++ ഭാഷയ്ക്കും അതിന്റേതായ അക്ഷരമാലയുണ്ട്. ഒരു പ്രോഗ്രാമിങ് ഭാഷയുടെ അക്ഷരമാലയെ അതിന്റെ ക്യാരക്ടർ സെറ്റ് എന്ന് വിളിക്കുന്നു. ഭാഷയിലെ അംഗീകരിക്കപ്പെട്ട ചിഹ്നങ്ങളുടെ ഗണമാണ് അത്. അതിൽ അക്ഷരങ്ങളും, അക്കങ്ങളും മറ്റ് ചിഹ്നങ്ങളും ഉൾപ്പെടുന്നു.

ന്യൂജഴ്സിയിലെ (USA) മുറേഹില്ലിലുള്ള, AT&T ബെൽ ലബോറട്ടറിയിലാണ് Bjarne Stroustrup C++ വികസിപ്പിച്ചത്. ഇപ്പോൾ അദ്ദേഹം കൊളംബിയ സർവകലാശാലയിലെ സന്ദർശക അധ്യാപകനും ടെക്സാസ് A&M സർവകലാശാലയിലുള്ള കോളേജ് ഓഫ് എഞ്ചിനിയറിംഗിലെ കമ്പ്യൂട്ടർ ശാസ്ത്ര വിഭാഗത്തിന്റെ ചുമതലയുള്ളയാളുമാണ്. അദ്ദേഹത്തിന് നിരവധി പുരസ്കാരങ്ങൾ ലഭിച്ചിട്ടുണ്ട്. C++ ന്റെ ആദ്യകാല നാമം, ക്ലാസ്സുകളോട് കൂടിയ C എന്നായിരുന്നു. പിന്നീട് 1983 - ൽ C++ എന്ന് പുനർനാമകരണം ചെയ്തു.



Bjarne Stroustrup

C++ അക്ഷരമാല ചുവടെ ചേർക്കുംവിധം തരം തിരിച്ചിരിക്കുന്നു.

(i)	അക്ഷരങ്ങൾ	-	A മുതൽ Z വരെ
			a മുതൽ z വരെ
(ii)	അക്കങ്ങൾ	-	0 മുതൽ 9 വരെ
(iii)	പ്രത്യേക ചിഹ്നങ്ങൾ	-	+ - * / ^ \ ( ) [ ] { } = < > . ‘ “ $, ; : % ? _ # @
(iv)	വൈറ്റ് സ്പെയ്സുകൾ	-	സ്പെയ്സ് ബാർ, ഹൊറിസോണ്ടൽ ടാബ് (→), ക്യാരിയേജ് റിട്ടേൺ ↵ ന്യൂ ലൈൻ, ഫോം ഫീഡ് തുടങ്ങിയവ കൊണ്ട് ഉണ്ടാക്കുന്ന വൈറ്റ് സ്പെയ്സുകൾ
(v)	മറ്റു ചിഹ്നങ്ങൾ	-	256 ASCII ചിഹ്നങ്ങളിൽ ഏതിനേയും C++ ലിറ്ററൽ ആയോ ഡാറ്റ ആയോ പ്രക്രിയയ്ക്കു വിധേയമാക്കാൻ സാധിക്കും.

അടുത്തടുത്ത വാക്കുകളും സംഖ്യകളും വേർതിരിക്കാൻ ഉപയോഗിക്കുന്ന അക്ഷരങ്ങളാണ് വൈറ്റ് സ്പെയ്സുകൾ.

4.2 ടോക്കണുകൾ (Tokens)

അക്ഷരമാല പഠിച്ചതിന് ശേഷം അക്ഷരങ്ങൾ ചേർത്തുണ്ടാകുന്ന വാക്കുകളെക്കുറിച്ചുള്ള പഠനമാണ് അടുത്തത്. സ്വാഭാവിക ഭാഷയിലെ വാക്കുകൾക്ക് സമാനമാണ് C++ ഭാഷയിലെ ‘ടോക്കൺ’. ഒരു പ്രോഗ്രാം വികസിപ്പിക്കുന്നതിനുള്ള അടിസ്ഥാന ഘടകങ്ങളാണ് ടോക്കണുകൾ. അവ ലെക്സിക്കൽ യൂണിറ്റുകൾ എന്നും അറിയപ്പെടുന്നു. C++ -ൽ താഴെപ്പറയും വിധം അഞ്ച് തരത്തിലുള്ള ടോക്കണുകളുണ്ട്.

1 കീവേർഡുകൾ
2 ഐഡന്റിഫയറുകൾ
3 ലിറ്ററലുകൾ
4 പഞ്ചുവേറ്ററുകൾ
5 ഓപ്പറേറ്ററുകൾ

4.2.1 കീവേഡുകൾ (Keywords)

ഭാഷാ കംബയിലറിന് ഒരു പ്രത്യേക അർഥം നൽകുന്ന വാക്കുകളാണ് (ടോക്കണുകളാണ്) കീവേഡുകൾ. പ്രത്യേക കാര്യങ്ങൾക്കായി ഭാഷ മാറ്റിവച്ച വാക്കുകളായതിനാലും മറ്റാവശ്യങ്ങൾക്ക് പുനർനിർവചിക്കാൻ സാധിക്കാത്തതിനാലും ഇവയെ കീവേഡുകൾ എന്നു വിളിക്കുന്നു. C++ -ലെ 48 കീവേഡുകൾ പട്ടിക 5.1-ൽ അക്ഷരമാലാക്രമത്തിൽ രേഖപ്പെടുത്തിയിട്ടുണ്ട്. അവയുടെ ഉപയോഗം പിന്നീട് വിശദീകരിക്കാം.

asm	continue	float	new	signed	try
auto	default	for	operator	sizeof	typedef
break	delete	friend	private	static	union
case	do	goto	protected	struct	unsigned
catch	double	if	public	switch	virtual
char	else	inline	register	template	void
class	enum	int	return	this	volatile
const	extern	long	short	throw	while

പട്ടിക 4.1 : C++ - ലെ കീവേഡുകൾ

4.2.2 ഐഡന്റിഫയറുകൾ (Identifiers)

സ്ഥലങ്ങൾ, വ്യക്തികൾ, വസ്തുക്കൾ എന്നിവ തിരിച്ചറിയുന്നതിനായി നാം പേരുകൾ നൽകാറുണ്ട്. ഇതിനു വേണ്ടിയാണ് C++ -ൽ ഐഡന്റിഫയറുകൾ ഉപയോഗിക്കുന്നത്. മെമ്മറി സ്ഥാനങ്ങൾ, വാചകങ്ങൾ, ഫങ്ഷനുകൾ, ഒബ്ജക്ടുകൾ, ക്ലാസ്സുകൾ തുടങ്ങിയ പ്രോഗ്രാമിലെ ഘടകങ്ങൾക്ക് പേര് നൽകുന്നതിന് ഉപയോക്താവ് നിർവചിക്കുന്ന വാക്കുകളാണ് ഐഡന്റിഫയറുകൾ. മെമ്മറി സ്ഥാനങ്ങളുടെ ഐഡന്റിഫയറുകളെ വേരിയബിളുകൾ എന്ന് വിളിക്കുന്നു. വാചകങ്ങൾക്ക് നൽകുന്ന ഐഡന്റിഫയറുകളെ ലേബലുകൾ എന്ന് വിളിക്കുന്നു. ഒരു കൂട്ടം പ്രസ്താവനകൾ പ്രതിനിധാനം ചെയ്യുന്ന ഐഡന്റിഫയറുകളാണ് ഫങ്ഷൻ നാമങ്ങൾ.

ഒരു പ്രോഗ്രാമിൽ ഐഡന്റിഫയറുകൾ നിർമ്മിക്കുമ്പോൾ, ചില നിബന്ധനകൾ കൃത്യമായി പാലിക്കേണ്ടതാവശ്യമാണ്. ആ നിയമങ്ങൾ താഴെപ്പറയും വിധമാണ്.

- അക്ഷരങ്ങൾ, അക്കങ്ങൾ, അണ്ടർ സ്കോർ (_) തുടങ്ങിയവയുടെ ഒരു ശ്രേണിയാണ് ഐഡന്റിഫയർ.
- ഐഡന്റിഫയർ തുടങ്ങുന്നത് അക്ഷരത്തിലോ, അണ്ടർ സ്കോറിലോ (_) ആയിരിക്കണം.
- വൈറ്റ് സ്പെയ്സോ പ്രത്യേക ചിഹ്നങ്ങളോ അനുവദനീയമല്ല.
- കീവേഡുകൾ ഐഡന്റിഫയറായി ഉപയോഗിക്കാൻ പാടില്ല.
- ചെറിയ അക്ഷരങ്ങളും വലിയ അക്ഷരങ്ങളും വ്യത്യസ്തമായാണ് C++-ൽ ഉപയോഗിക്കുന്നത്. അതായത് C++ കെയ്സ് സെൻസിറ്റീവാണ്.

സാധുവായ ചില ഐഡന്റിഫയറുകൾക്ക് ഉദാഹരണം: `Count, sumof2numbers, Average_Height, _1stRank, Main, FOR`

താഴെപ്പറയുന്നവ ചില അസാധുവായ ഐഡന്റിഫയറുകളാണ്. അതിനുള്ള കാരണങ്ങളും സൂചിപ്പിച്ചിരിക്കുന്നു.

Sum of digits	→	ശൂന്യസ്ഥലങ്ങൾ ഉപയോഗിച്ചിരിക്കുന്നു
1styear	→	അക്കം ആദ്യത്തെ അക്ഷരമായി ഉപയോഗിച്ചിരിക്കുന്നു
First . Jan	→	പ്രത്യേക ചിഹ്നം (.) ഉപയോഗിച്ചിരിക്കുന്നു
for	→	അതൊരു കീവേഡാണ്

നമുക്കു ചെയ്യാം

താഴെ തന്നിരിക്കുന്നവയിൽ നിന്ന് അസാധുവായ ഐഡന്റിഫയറുകൾ കണ്ടെത്തുക. അതിനുള്ള കാരണവും എഴുതുക.

`Data_rec, _data, Idata, data1, my.file, asm, switch, goto, break`

4.2.3 ലിറ്ററലുകൾ (Literals)

11-ാം ക്ലാസ് വിദ്യാർഥികൾക്കായുള്ള ഏകജാലക പ്രവേശന സംവിധാനം പരിഗണിക്കുക. പ്രവേശന ഫോറത്തിൽ നിങ്ങൾ ജനനത്തീയതി നൽകിയിട്ടുണ്ടാകും. ജനനത്തീയതി ഒരിക്കലും മാറുന്നില്ല. ഗണിത ശാസ്ത്രത്തിൽ π ഒരു സ്ഥിരാംഗവും ഗുരുത്വാകർഷണ സ്ഥിരാംഗം 'g' യുടെ വില (9.8 മീ./സെ.) ഒരിക്കലും മാറാത്തതുമാണെന്ന് നമുക്കറിയാം. അതുപോലെ, C++ -ൽ ഡാറ്റാ ഇനങ്ങളെ പ്രതിനിധാനം ചെയ്യാൻ നാം ഉപയോഗിക്കുന്ന ടോക്കണുകളായ ലിറ്ററലുകൾ, പ്രോഗ്രാം പ്രവർത്തിച്ച് തീരും വരെ വില മാറ്റം വരാതെ തുടരുന്നു. അവയെ പലപ്പോഴും സ്ഥിരാംഗങ്ങൾ എന്ന് വിളിക്കുന്നു. ലിറ്ററലുകൾ താഴെ പറയും വിധം നാലായി തരം തിരിക്കപ്പെട്ടിരിക്കുന്നു.

1. ഇന്റിജർ ലിറ്ററൽ
2. ഫ്ളോട്ടിംഗ് പോയിന്റ് ലിറ്ററൽ
3. ക്യാരക്ടർ ലിറ്ററൽ
4. സ്ട്രിങ് ലിറ്ററൽ

ഇന്റിജർ ലിറ്ററലുകൾ (പൂർണസംഖ്യാ സ്ഥിരാംഗങ്ങൾ)

1776, 707, –273 എന്നീ സംഖ്യകൾ പരിഗണിക്കുക. അവ പൂർണ ദശാങ്ക വിലകൾ സൂചിപ്പിക്കുന്ന ഇന്റിജർ സ്ഥിരാംഗങ്ങളാണ്. അക്കങ്ങൾ മാത്രം ചേർത്തുണ്ടായിട്ടുള്ള ടോക്കണുകളായ ഇന്റിജർ ലിറ്ററലുകൾ അവിഭാജ്യ ഘടകങ്ങളില്ലാത്ത പൂർണ സംഖ്യകളാണ്. ഇന്റിജർ ലിറ്ററലുകളുടെ സ്വഭാവ സവിശേഷതകൾ താഴെപ്പറയുന്നവയാണ്.

- ഒരു പൂർണ സംഖ്യാ സ്ഥിരാംഗത്തിന് ഒരു അക്കമെങ്കിലും ഉണ്ടായിരിക്കണം. ദശാംശം ഉണ്ടാകരുത്.
- സംഖ്യ പോസിറ്റീവ് എന്നോ നെഗറ്റീവ് എന്നോ കാണിക്കുന്ന + അല്ലെങ്കിൽ – ചിഹ്നം ആദ്യ അക്ഷരമായി വരാവുന്നതാണ്.
- ചിഹ്നം ഇല്ലാത്ത സംഖ്യ പോസിറ്റീവ് സംഖ്യയായി കണക്കാക്കുന്നു.
- മറ്റ് ചിഹ്നങ്ങൾ അനുവദിക്കില്ല.

നമുക്കു ചെയ്യാം

താഴെത്തന്നിട്ടുള്ളവ സാധുവോ അസാധുവോ ആയ പൂർണ്ണ സംഖ്യാ സ്ഥിരാംഗങ്ങളായി തരംതിരിക്കുകയും അവ അസാധുവായതിന്റെ കാരണം കണ്ടെത്തുകയും ചെയ്യുക.

77,000	70	314.	-5432	+15346
+23267	-.7563	-02281+0	1234E56	-9999

C++ - ൽ ദശാംശ സംഖ്യകൾ (ബേസ്- 10) കൂടാതെ അഷ്ടാംഗസംഖ്യകളും (ബേസ്- 8) ഷോഡശ സംഖ്യകളും (ബേസ്- 16) ലിറ്ററലുകളായി (സ്ഥിരാംഗങ്ങളായി) ഉപയോഗിക്കുന്നു. അഷ്ടാംഗ സംഖ്യയെ സൂചിപ്പിക്കുന്നതിന് 0 (പൂജ്യം എന്ന ചിഹ്നം) മുന്നിൽ നൽകുകയും ഷോഡശ സംഖ്യയെ സൂചിപ്പിക്കുന്നതിന് 0x (പൂജ്യം, x) മുന്നിൽ നൽകുകയും ചെയ്യുന്നു. ഉദാഹരണമായി പൂർണസംഖ്യാ സ്ഥിരാംഗങ്ങളായ 75 (ബേസ്- 10)- ഉം 0113 (ബേസ്- 8)- ഉം 0 x 4B (ബേസ്- 16)- ഉം എല്ലാം പരസ്പരം തുല്യ മാണ്.

ഫ്ളോട്ടിങ് പോയിന്റ് (അസ്ഥിര ദശാംശ സംഖ്യ) ലിറ്ററലുകൾ

3.1459, 3.0x10^8, 1.6x10^{-19}, 3.0 തുടങ്ങിയ സംഖ്യകൾ നിങ്ങൾക്ക് പരിചിതമാണല്ലോ. ഇവ സാധുവായ നാല് വിലകളാണ്. ഒന്നാമത്തേത് π (പൈ)-യുടെ വിലയും, രണ്ടാമത്തേത് പ്രകാശ വേഗത മീറ്റർ/സെക്കന്റിൽ ഉള്ളതും, മൂന്നാമത്തേത് ഇലക്ട്രോണിന്റെ വൈദ്യുത ചാർജും (വളരെ ചെറിയ സംഖ്യ) അവസാനത്തേത് 3 എന്ന സംഖ്യയെ അസ്ഥിര ദശാംശസംഖ്യ ലിറ്ററലായി പ്രസ്താവിച്ചിട്ടുള്ളതുമാണ്.

അസ്ഥിര ദശാംശസംഖ്യ ലിറ്ററലുകൾ അവിഭാജ്യ ഘടകങ്ങളുള്ള രേഖീയ സ്ഥിരാംഗങ്ങൾ എന്നും അറിയപ്പെടുന്നു. ഇവ അവിഭാജ്യ രൂപമായോ കൃത്യങ്കരൂപമായോ എഴുതാൻ സാധിക്കും.

രേഖീയ സ്ഥിരാംഗങ്ങൾ അവിഭാജ്യ രൂപത്തിൽ ചിഹ്നമുള്ളതോ ഇല്ലാത്തതോ ആയതും, അക്കങ്ങൾ ചേർന്നതും, അവയ്ക്കിടയിൽ ദശാംശമുള്ളവയുമായിരിക്കും. രേഖീയ സ്ഥിരാംഗങ്ങൾ അവിഭാജ്യ രൂപത്തിലെഴുതുന്നതിനുള്ള നിയമങ്ങൾ ചുവടെ ചേർക്കുന്നു.

- ഒരു രേഖീയ സ്ഥിരാംഗത്തിന് അതിന്റെ അവിഭാജ്യ രൂപത്തിൽ ഒരക്കവും ഒരു ദശാംശവും എങ്കിലും ഉണ്ടായിരിക്കണം.
- അതിന് ഒന്നുകിൽ + (അധികം) അല്ലെങ്കിൽ – (ന്യൂനം) ചിഹ്നം തുടക്കത്തിൽ ഉണ്ടാകാം.
- ചിഹ്നം ഇല്ലാത്ത ഒരു സ്ഥിരാംഗത്തെ അധിക ചിഹ്നമുള്ള സംഖ്യയായി കണക്കാക്കാം.

കൃത്യങ്കരൂപത്തിൽ ഒരു രേഖീയ സ്ഥിരാംഗത്തിന് രണ്ട് ഭാഗങ്ങളുണ്ടാകും – അപൂർണ സംഖ്യാ ഭാഗവും (മാന്റിസ്സ) കൃത്യങ്കവും. ഉദാഹരണമായി, 5.8 എന്നത് 0.58 x 10^1 = 0.58 E1 എന്നെഴു താനാകും. ഇതിൽ കൃത്യങ്കഭാഗം 1-ഉം (E ക്ക് ശേഷം വരുന്ന ഭാഗം) മാന്റിസ്സ ഭാഗം 0.58-ഉം (E യ്ക്ക് മുമ്പുള്ള ഭാഗം) ആണ്. E 1 എന്നത് 10^1-നെ സൂചിപ്പിക്കുന്നു. കൃത്യങ്ക രൂപത്തിൽ രേഖീയ സ്ഥിരാംഗം എഴുതുന്നതിനുള്ള നിയമങ്ങൾ താഴെപ്പറയും വിധമാണ്.

- കൃത്യങ്ക രൂപത്തിൽ രേഖീയ സ്ഥിരാംഗങ്ങൾക്ക് രണ്ട് ഭാഗങ്ങളുണ്ടായിരിക്കും. ഒരു പൂർണ സംഖ്യാ ഭാഗവും ഒരു കൃത്യങ്കവും.

- അപൂർണ സംഖ്യാ ഭാഗം ഒന്നുകിൽ ഒരു പൂർണ സംഖ്യാ രൂപത്തിലോ അല്ലെങ്കിൽ സാധുവായ അവിഭാജ്യ രൂപത്തിലോ ആയിരിക്കും.
- അപൂർണ സംഖ്യാ ഭാഗത്തെ തുടർന്ന് E അല്ലെങ്കിൽ e എന്ന അക്ഷരവും കൃത്യങ്കവുമുണ്ടാകും.
- കൃത്യങ്കം പൂർണസംഖ്യ ആയിരിക്കണം.

താഴെത്തന്നിട്ടുള്ളവ സാധുവായ രേഖീയ സ്ഥിരാംഗങ്ങളാണ്.

52.0	107.5	-713.8	-.00925
453.E-5	1.25E08	.212E04	562.0E09
152E+8	1520E04	-0.573E-7	-.097

ചില അസാധുവായ രേഖീയ സ്ഥിരാംഗങ്ങൾ കാരണസഹിതം ഇതോടൊപ്പം ചേർക്കുന്നു.

58,250.262 (കോമ ഉപയോഗിച്ചിരിക്കുന്നു), 5.8E (കൃത്യങ്ക ഭാഗം ഇല്ല), 0.58E2.3 (കൃത്യങ്കമായി അവിഭാജ്യ സംഖ്യ ഉപയോഗിച്ചിരിക്കുന്നു).

നമുക്കു ചെയ്യാം

താഴെ തന്നിട്ടുള്ളവയെ കാരണസഹിതം സാധുവോ അസാധുവോ ആയ രേഖീയ സ്ഥിരാംഗങ്ങളായി തരംതിരിക്കുക.

77,00,000	7.0	3.14	-5.0E5.4	+53.45E-6
+532.67.	.756E-3	-0.528E10	1234.56789	34,56.24
4353	+34/2	5.6E	4356	0

ക്യാരക്ടർ ലിറ്ററൽ

സാധാരണ ലിംഗഭേദത്തെ കാണിക്കുന്നതിനായി ആൺ (Male) എന്നതിന് 'M' അല്ലെങ്കിൽ 'm' എന്നും പെൺ (Female) എന്നതിന് 'F' അല്ലെങ്കിൽ 'f' എന്നും ഉപയോഗിക്കാറുണ്ട്. അതുപോലെ, അതെ (Yes) എന്നതിന് 'y' അല്ലങ്കിൽ 'Y' എന്നും അല്ല (No) എന്നതിന് 'n' അല്ലെങ്കിൽ 'N' എന്നും നാം ഉപയോഗിക്കുന്നു. ഇവയെല്ലാം ഒറ്റ അക്ഷരങ്ങളാണ്. പ്രോഗ്രാം പ്രവർത്തിച്ച് തീരും വരെ അവ വില മാറ്റമില്ലാതെ തുടരുന്നു. ഒരു ജോഡി ഏക സൂചകങ്ങൾക്കുള്ളിൽ (ഒറ്റ ഉദ്ധരണികൾക്ക് ഉള്ളിൽ) ഉപയോഗിച്ചിരിക്കുന്ന അക്ഷരത്തെ ക്യാരക്ടർ ലിറ്ററൽ അല്ലെങ്കിൽ ക്യാരക്ടർ സ്ഥിരാംഗം എന്നു വിളിക്കുന്നു.

ഉദ്ധരണി ഇല്ലാതെ ഉപയോഗിക്കുന്ന X ഐഡന്റിഫയറും ഉദ്ധരണിക്കുള്ളിൽ ഉപയോഗിക്കുന്ന 'X' ക്യാരക്ടർ ലിറ്ററലുമാണ്. ക്യാരക്ടർ സ്ഥിരാംഗത്തിന്റെ വില എന്നത് അതിന്റെ ASCII വിലയാണ്. 'c' -യുടെ വില എന്നത് 'c' -യുടെ ASCII വിലയായ 99-ഉം 'A'-യുടെ വില എന്നത് അതിന്റെ ASCII വിലയായ 65-ഉം ആണ്.

C++ ഭാഷയിൽ ചില ചിത്രീകരിക്കാനാകാത്ത ക്യാരക്ടർ സ്ഥിരാംഗങ്ങളുണ്ട്, അവയെ കീബോർഡിൽ നിന്ന് നേരിട്ട് ടൈപ്പ് ചെയ്യാൻ സാധിക്കില്ല. ഉദാഹരണമായി, ക്യാരിയേജ് റിട്ടേൺ അല്ലെങ്കിൽ എന്റർ കീ, ടാബ് കീ, ബാക്ക് സ്പെയ്സ് കീ എന്നിവ ഒരു രീതിയിലും പ്രകടിപ്പിക്കാൻ സാധ്യമല്ല. ഇത്തരം പ്രകടിപ്പിക്കാനാകാത്ത ചിഹ്നങ്ങളെ പ്രതിനിധീകരിക്കാനായി

എസ്കേപ്പ് സീക്വൻസുകൾ ഉപയോഗിക്കാം. അവയിൽ ബാക്ക് സ്ലാഷും തുടർന്ന് വരുന്ന ഒന്നോ അതിലധികമോ അക്ഷരങ്ങളും അടങ്ങിയിരിക്കും. എസ്കെയ്പ് സീക്വൻസുകൾ ഒരു ജോഡി ഏക സൂചകങ്ങൾക്കുള്ളിലുള്ള ഒന്നിലധികം അക്ഷരങ്ങൾ ചേർന്നതാണെങ്കിലും ആനുപാതികമായ ഒരു ASCII കോഡുപയോഗിച്ചാണ് അത് സൂചിപ്പിക്കുന്നത് എന്ന കാര്യം ശ്രദ്ധിക്കേണ്ടതാണ്. അതുകൊണ്ടാണ് അവയെ ക്യാരക്ടർ സ്ഥിരാംഗങ്ങളായി കണക്കാക്കുന്നത്. പട്ടിക 5.2-ൽ എസ്കേപ്പ് സീക്വൻസുകളും ആനുപാതികമായ ക്യാരക്ടറുകളും കുറിച്ചിട്ടുണ്ട്.

എസ്കേപ്പ് സീക്വൻസ്	ആനുപാതികമായ ചിത്രീകരിക്കാനാകാത്ത ചിഹ്നങ്ങൾ
\a	ഓഡിയബിൾ ബെൽ (അലെർട്ട്)
\b	ബാക്ക് സ്പെയ്സ്
\f	ഫോം ഫീഡ്
\n	ന്യൂലൈൻ അല്ലെങ്കിൽ ലൈൻഫീഡ്
\r	ക്യാരിയേജ് റിട്ടേൺ
\t	ഹൊറിസോണ്ടൽ ടാബ്
\v	വെർട്ടിക്കൽ ടാബ്
\\	ബാക്ക് സ്ലാഷ്
\'	സിംഗിൾ ക്വോട്ട് (ഏകസൂചകം)
\"	ഡബിൾ ക്വോട്ട് (ജോഡിയായ സൂചകം)
\?	ക്വസ്റ്റ്യൻ മാർക്ക് (ചോദ്യ ചിഹ്നം)
\0	നൾ ക്യാരക്ടർ

പട്ടിക 5.2: എസ്കേപ്പ് സീക്വൻസുകൾ

പട്ടിക 5.2-ൽ \', \", \? എന്നിവ നിങ്ങൾക്ക് കാണാൻ സാധിക്കും. ഈ ചിഹ്നങ്ങൾ (അക്ഷരങ്ങൾ) കീ ബോർഡിൽ നിന്ന് ടൈപ്പ് ചെയ്യാൻ സാധിക്കുമെങ്കിലും എസ്കേപ്പ് സീക്വൻസ് ഇല്ലാതെ ഉപയോഗിച്ചാൽ അവ പ്രത്യേക ഉദ്ദേശ്യവും കാര്യവുമായിരിക്കും നിർവഹിക്കുന്നത്. അവ അതുപോലെ തന്നെ പ്രദർശിപ്പിക്കുകയോ അച്ചടിക്കുകയോ ചെയ്യേണ്ടി വന്നാൽ എസ്കേപ്പ് സീക്വൻസ് ഉപയോഗിക്കണം. ചില സാധുവായ ക്യാരക്ടർ സ്ഥിരാംഗങ്ങൾക്ക് ഉദാഹരണങ്ങളാണ്:

's', 'S', '$', '\n', '+', '9'

ഒക്ടൽ സംഖ്യകളെയും ഹെക്സാ ഡെസിമൽ സംഖ്യകളെയും എസ്കേപ്പ് സീക്വൻസുകളുടെ സഹായത്താലാണ് C++ - ൽ സൂചിപ്പിക്കുന്നത്. \0n-ഉം \xHn-ഉം യഥാക്രമം ഒക്ടൽ സംഖ്യാ സമ്പ്രദായത്തിലും ഹെക്സാ ഡെസിമൽ സംഖ്യാന സമ്പ്രദായത്തിലും ഉള്ള സംഖ്യകളെ സൂചിപ്പിക്കുന്നു.

ചില അസാധുവായ ക്യാരക്ടർ സ്ഥിരാങ്കങ്ങൾ കാരണ സഹിതം തന്നിരിക്കുന്നു.

A (ഏക ഉദ്ധരണി ഇല്ല), '82' (ഒന്നിലധികം അക്ഷരങ്ങൾ), "K" (ഏക ഉദ്ധരണിക്ക് പകരം ജോഡിയായ ഉദ്ധരണികൾ), '\g' (അസാധുവായ എസ്കേപ്പ് സീക്വൻസ് അല്ലെങ്കിൽ ഒന്നിലധികം അക്ഷരങ്ങൾ).

സ്ട്രിങ് ലിറ്ററൽ

നന്ദന ഒരു വിദ്യാർഥിനിയാണ്. അവൾ ബാപ്പുജി നഗറിൽ താമസിക്കുന്നു. ഇവിടെ "Nandana" എന്നത് ഒരു പെൺകുട്ടിയുടെ പേരും "Bapuji Nagar" എന്നത് ഒരു സ്ഥല നാമവുമാണ്. ഈ

രീതിയിലുള്ള ഡാറ്റ, പ്രോഗ്രാമുകളുടെ സഹായത്താൽ പ്രോസസ്സ് ചെയ്യേണ്ടി വരും. അത്തരം ഡാറ്റയെ സ്ട്രിങ് സ്ഥിരാംഗങ്ങളായി പരിഗണിക്കുകയും ജോഡികളായ ഉദ്ധരണികൾക്കുള്ളിൽ ഉൾപ്പെടുത്തുകയും ചെയ്യുന്നു. തുടർച്ചയായ ഒന്നോ അതിലധികമോ അക്ഷരങ്ങൾ (ചിഹ്നങ്ങൾ), രണ്ട് ജോടിയായ സൂചകങ്ങൾക്കുള്ളിൽ (ഉദ്ധരണികൾക്കുള്ളിൽ) അടക്കം ചെയ്തതാണ് സ്ട്രിങ് സ്ഥിരാംഗങ്ങൾ. ഉദാഹരണമായി "Hello friends", "123", "C++", "Baby\'s Day Out" എന്നിവ സാധുവായ സ്ട്രിങ് സ്ഥിരാംഗങ്ങളാണ്.

നമുക്കു ചെയ്യാം

താഴെ തന്നിട്ടുള്ളവ പല വിഭാഗങ്ങളിലുള്ള ലിറ്ററലുകളായി (സ്ഥിരാംഗങ്ങളായി) തരംതിരിക്കുക.

'a'	'rita'	-124	12.5	-12e-1
"raju\'s pen"	0	-11.999	'\\'	32760

4.2.4 പഞ്ചുവേറ്ററുകൾ (പൂർണ വിരാമ ചിഹ്നങ്ങൾ) (Punctuators)

വാക്യങ്ങളിലെ വ്യാകരണത്തിന്റെ പൂർണതയ്ക്കായി പഞ്ചുവേഷൻ അടയാളങ്ങൾ ഇംഗ്ലീഷ്, മലയാളം തുടങ്ങിയ ഭാഷകളിലുപയോഗിക്കുന്നുണ്ട്. ഉദാഹരണമായി, 'ആരാണ് C++ വികസിപ്പിച്ചത്?' എന്ന വാക്യം ശ്രദ്ധിക്കുക. ഇവിടെ '?' എന്നത് പഞ്ചുവേഷൻ അടയാളവും അത് സൂചിപ്പിക്കുന്നത് ഒരു ചോദ്യ പ്രസ്താവനയെയുമാണ്. അതുപോലെ എല്ലാ വാക്യങ്ങളുടെയും അവസാനം നാം പൂർണവിരാമ ചിഹ്നം (.) ഇടുന്നു. അത്തരത്തിൽ അർഥ സംബന്ധമായോ, പദവിന്യാസപരമായോ ഉള്ള പൊരുൾ കംബയിലറിന് എത്തിക്കുവാനായി C++ -ലും ചില പ്രത്യേക ചിഹ്നങ്ങൾ ഉണ്ട്. അവയെ പഞ്ചുവേറ്ററുകൾ എന്ന് വിളിക്കുന്നു.

അവയ്ക്കുള്ള ഉദാഹരണങ്ങളാണ് # ; ' " ( ) { } [ ] എന്നിവ. ഓരോ പഞ്ചുവേറ്ററിന്റെയും ഉപയോഗം പിന്നീട് ചർച്ച ചെയ്യാം.

4.2.5 ഓപ്പറേറ്ററുകൾ (Operators)

5-ഉം 3-ഉം തമ്മിൽ കൂട്ടേണ്ടി വരുമ്പോൾ, അവ 5+3 എന്ന് നാം സൂചിപ്പിക്കുന്നു. ഇവിടെ + എന്നത് സങ്കലനം എന്ന പ്രവൃത്തിയെ കാണിക്കുന്ന ഓപ്പറേറ്റർ ആണ്. C++-ൽ ഇതുപോലെയുള്ള ധാരാളം ഓപ്പറേറ്ററുകൾ ഉണ്ട്. കംമ്പയിലറിനോട് ഒരു പ്രത്യേക പ്രവൃത്തിയെക്കുറിച്ച് പറയുന്നതിന് ഉപയോഗിക്കുന്ന ചിഹ്നമാണ് ഓപ്പറേറ്റർ. ഏതെങ്കിലും തരത്തിലുള്ള പ്രവൃത്തിയെ ഉത്തേജിപ്പിക്കുന്ന ടോക്കണുകളാണ് അവ. ഓപ്പറന്റുകൾ എന്ന് വിളിക്കുന്ന ഒരു കൂട്ടം ഡാറ്റാകളിൽ ഓപ്പറേറ്ററുകൾ പ്രയോഗിക്കുന്നു. അരിത്മെറ്റിക്, ലോജിക്കൽ, റിലേഷണൽ, കണ്ടീഷണൽ, അസൈൻമെന്റ് തുടങ്ങിയ വിവിധ തരം ഓപ്പറേറ്ററുകൾ C++ -ൽ ഉണ്ട്. ഓപ്പറേറ്ററുകളെക്കുറിച്ച് കൂടുതലായി നമുക്ക് അടുത്ത അധ്യായത്തിൽ ചർച്ച ചെയ്യാം.

നമുക്കു ചെയ്യാം

താഴെ തന്നിട്ടുള്ളവ വിവിധയിനം ടോക്കണുകളായി തരംതിരിക്കുക.

/	-24	+	-12e-1	'	"KL01"	
Sum	"raju\'s	pen"if	rita	'\\'	break	}

4.3 ഇന്റഗ്രേറ്റഡ് ഡവലപ്മെന്റ് എൻവയോൺമെന്റ് (Integrated Development Environment (IDE))

ഒരു C++ പ്രോഗ്രാമിലെ അടിസ്ഥാന ഘടകങ്ങൾ നാം ഇപ്പോൾ പഠിച്ചു കഴിഞ്ഞു. C++ പ്രോഗ്രാമുകൾ എഴുതിത്തുടങ്ങുന്നതിന് മുൻപ് എവിടെയാണ് പ്രോഗ്രാം ടൈപ്പ് ചെയ്യേണ്ടതെന്ന് നാം അറിഞ്ഞിരിക്കണം. മറ്റ് പ്രോഗ്രാമിങ് ഭാഷകളെപ്പോലെ ഒരു ടെക്സ്റ്റ് എഡിറ്റർ ഉപയോഗിച്ച് C++ പ്രോഗ്രാം നിർമ്മിക്കാം. C++ പ്രോഗ്രാമുകൾ സൃഷ്ടിക്കാനായി ടർബോ C++, ബോറോലാന്റ് C++, GCC തുടങ്ങിയ കമ്പയിലറുകൾ IDE ലഭ്യമാക്കുന്നു. ഇവയിൽ പല IDE-കളിലും ടൈപ്പിങ്, എഡിറ്റിങ്, സേർച്ചിങ്, ലിങ്കിങ്, കമ്പയിലിങ്, എക്സിക്യൂട്ടിങ് എന്നീ സൗകര്യങ്ങൾ ഉണ്ട്. ഇത്തരം പ്രവർത്തനങ്ങൾക്കായി നാം ഇവിടെ ജിനി IDE (ഐ.ടി.@ സ്കൂൾ ഉബുണ്ടു ലിനക്സ് 14.04) ഉപയോഗിക്കുന്നു.

GCC യോടൊപ്പമുള്ള ജിനി IDE

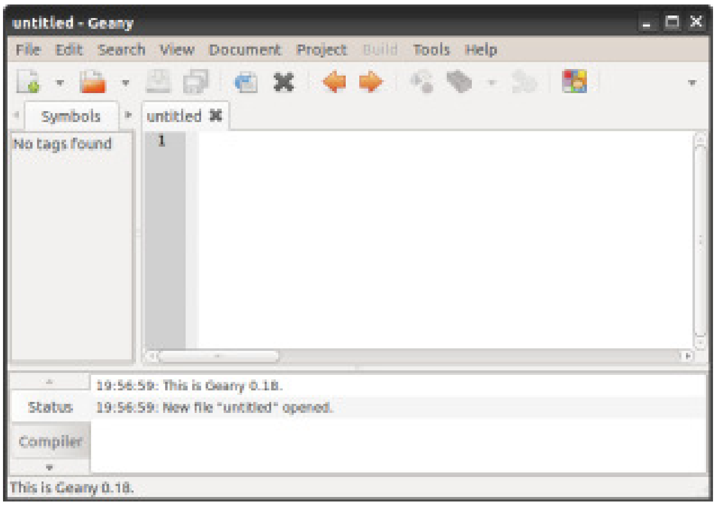
ലിനക്സ് ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തോടൊപ്പം ലഭ്യമാകുന്ന ഒരു സ്വതന്ത്ര സോഫ്റ്റ്‌വെയറാണ് GCC കമ്പയിലർ. GCC എന്നത് GNU കമ്പയിലർ കളക്ഷനെ പ്രതിനിധാനം ചെയ്യുന്നതും ISO C++ മാനദണ്ഡങ്ങൾ അനുസരിച്ച് പ്രവർത്തിക്കുന്നതുമായ ഒരു ജനപ്രിയ കമ്പയിലറാണ്. C++ പ്രോഗ്രാമുകൾ എഴുതുന്നതിനും, കമ്പയിൽ ചെയ്യുന്നതിനും എക്സിക്യൂട്ട് ചെയ്യുന്നതിനുമുള്ള ഒരു ക്രോസ് - പ്ലാറ്റ്ഫോം IDE ആണ് ജിനി.

a. എഡിറ്റ് വിൻഡോ തുറക്കുന്ന വിധം

ഉബുണ്ടു ലിനക്സിന്റെ ആപ്ലിക്കേഷൻസ് മെനുവിൽ നിന്നുമാണ് ജിനി IDE-യുടെ എഡിറ്റ് വിൻഡോ തുറക്കുന്നത്.

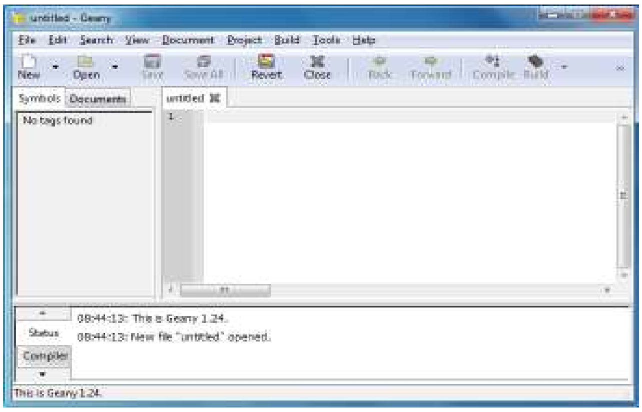
അപ്ലിക്കേഷനുകൾ → പ്രോഗ്രാമിങ് →ജിനി

ജിനി IDE അതിന്റെ വിൻഡോ തുറക്കുന്നത് ചിത്രം 5.1-ൽ കാണിച്ചിരിക്കും വിധമാണ്. അതിന് ഒരു ടൈറ്റിൽ ബാർ, മെനു ബാർ, ടൂൾ ബാർ, കോഡ് എഡിറ്റ് ചെയ്യാനുള്ള സ്ഥലം എന്നിവ ഉണ്ടാകും. അൺ ടൈറ്റിൽഡ് എന്ന പേരിലുള്ള ഒരു ടാബും അവിടെ കാണാൻ സാധിക്കും. വിൻഡോസ് ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിൽ ജിനി 1.24 ആണ് നാം ഉപയോഗിക്കുന്നതെങ്കിൽ, തുറക്കുന്ന ജാലകം (വിൻഡോ) ചിത്രം 5.2-ൽ കാണിച്ചിരിക്കും വിധമായിരിക്കും. മുകളിൽ പരാമർശിച്ച രണ്ട് ജാലകങ്ങളും (വിൻഡോകളും) സമാനമാണെന്ന് നമുക്ക് കാണാം.



ചിത്രം 4.1: ഉബുണ്ടു ലിനക്സിലെ ജിനി IDE- - യുടെ പ്രാരംഭ സ്ക്രീൻ

അൺടൈറ്റിൽഡ് എന്ന തനത് പേരുള്ള ഫയലിൽ നമുക്ക് പ്രോഗ്രാം എഡിറ്റ് ചെയ്യാം. ഒരു പുതിയ ഫയൽ തുറക്കാനായി ഫയൽ മെനുവിലെ ന്യൂ ഓപ്ഷൻ തിരഞ്ഞെടുക്കുകയോ ടൂൾ ബാറിലുള്ള ന്യൂ ബട്ടൺ ക്ലിക്ക് ചെയ്യുകയോ ആകാം. ഇതേ ആവശ്യത്തിനായി Ctrl+N എന്നീ കീ-കൾ ഒരുമിച്ചും ഉപയോഗിക്കാം.



ചിത്രം 4.2: വിൻഡോസ് ഓപ്പറേറ്റിങ് സിസ്റ്റത്തിൽ

b. പ്രോഗ്രാം സേവ് ചെയ്യുന്നു

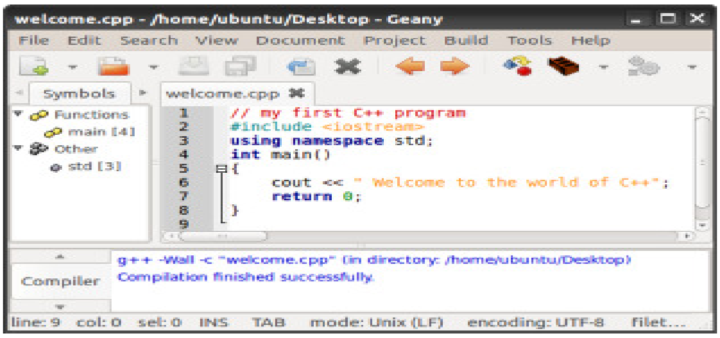
ഒരിക്കൽ ഫയൽ തുറന്നാൽ C++ പ്രോഗ്രാം നൽകുകയും അനുയോജ്യമായ ഫയൽ നാമത്തോടൊപ്പം .cpp എന്ന എക്സ്റ്റൻഷൻ കൊടുത്ത് അത് സേവ് ചെയ്യുകയും ആകാം. GCC എന്നത് ഒരു കൂട്ടം കമ്പയിലറുകൾ ആയതിനാൽ കോഡിന്റെ കമ്പയിലേഷന് വേണ്ടി ഏത് കമ്പയിലർ തിരഞ്ഞെടുക്കണമെന്ന് തീരുമാനിക്കുന്നത് അതിന്റെ എക്സ്റ്റൻഷൻ നോക്കിയാണ്. അതുകൊണ്ട് ഫയൽ എക്സ്റ്റൻഷൻ കൃത്യമായി നൽകിയിരിക്കണം. പ്രോഗ്രാം ടൈപ്പ് ചെയ്യുന്നതിന് മുൻപായി നാം ഫയൽ നാമം നൽകുകയാണെങ്കിൽ, പ്രോഗ്രാമിലുള്ള വിവിധയിനം ടോക്കണുകളെ വേർതിരിച്ച് കാണിക്കാനായി GCC സ്വയം പല നിറങ്ങൾ ലഭ്യമാക്കും. സോഴ്സ് കോഡിലെ വാചകങ്ങളുടെ സ്ഥാനം തിരിച്ചറിയുന്നതിനുള്ള ഇൻഡന്റേഷനും ഇത് ഉപയോഗിക്കുന്നു. ഇൻഡന്റേഷൻ എന്ന ആശയം നമുക്ക് പിന്നീട് ചർച്ച ചെയ്യാം.

പ്രോഗ്രാം 5.1-ൽ കാണിച്ചിരിക്കുന്നത് പോലെ ലളിതമായ ഒരു പ്രോഗ്രാം നമുക്ക് എഴുതുകയും welcome.cpp എന്ന പേരിൽ അത് സേവ് ചെയ്യുകയും ആകാം.

പ്രോഗ്രാം 5.1: IDE മനസ്സിലാക്കുന്നതിനുള്ള ഒരു പ്രോഗ്രാം.

```
// my first C++ program
#include<iostream>
using namespace std;
int main()
{
    cout << "Welcome to the world of C++";
    return 0;
}  //end of program
```

പ്രോഗ്രാം 5.1 കൊടുത്തശേഷമുള്ള IDE വിൻഡോ ചിത്രം 5.3-ൽ കാണിച്ചിരിക്കുന്നു. ടോക്കണുകളിൽ ഉപയോഗിച്ചിട്ടുള്ള വിവധ നിറങ്ങൾ നിരീക്ഷിക്കുക.



ചിത്രം 4.3: ജിനി IDE- യിൽ ഒരു പേര് നൽകി പ്രോഗ്രാം സേവ് ചെയ്യുന്നു.

പ്രോഗ്രാം സേവ് ചെയ്യുന്നതിനായി, ഫയൽ മെനുവിലെ സേവ് ഓപ്ഷൻ എടുക്കുകയോ കീബോർഡിലെ Ctrl+S എന്ന കുറുക്കുവഴി ഉപയോഗിക്കുകയോ ചെയ്യുക. ടൂൾ ബാറിലെ സേവ് ബട്ടൺ ക്ലിക്ക് ചെയ്ത് കൊണ്ടും ഫയൽ സേവ് ചെയ്യാവുന്നതാണ്. ഇടയ്ക്കിടെ Ctrl+S അമർത്തിക്കൊണ്ട് പ്രോഗ്രാം സേവ് ചെയ്യുന്നതാണ് ഉചിതമായ രീതി .

അപ്രതീക്ഷിതമായ സിസ്റ്റത്തിന്റെ പിഴവോ, വൈദ്യുതിത്തകരാറോ മൂലം ഡാറ്റ നഷ്ടപ്പെടുന്നത് ഒഴിവാക്കാൻ ഇത് സഹായിക്കുന്നു. ഒരിക്കൽ, പ്രോഗ്രാം ടൈപ്പ് ചെയ്യുന്നത് പൂർണമായാൽ കമ്പയിൻ ചെയ്യുകയോ വ്യത്യാസം വരുത്തുകയോ ചെയ്യുന്നതിന് മുൻപായി ഫയൽ സേവ് ചെയ്യുന്നതാണ് ഏറ്റവും ഉചിതം. താൽക്കാലികമായ അസ്ഥിര പ്രാഥമിക മെമ്മറിയിൽ (വോളട്ടൈൽ പ്രൈമറി മെമ്മറിയിൽ) നിന്ന് ഒരു ഫയൽ ദീർഘകാലത്തേക്കുള്ള സുസ്ഥിരമായ ദ്വിതീയ മെമ്മറിയിലേക്ക് (നോൺ വോളട്ടൈൽ സെക്കന്ററി മെമ്മറിയിലേക്ക്) പകർത്തുന്നതിനെയാണ് പ്രോഗ്രാം സേവ് ചെയ്യുക എന്ന് പറയുന്നത്.

വിവിധ C++ കമ്പയിലറുകൾക്കനുസരിച്ച് പല എക്സ്റ്റൻഷനുകളാണ് പ്രോഗ്രാം ഫയലുകൾക്ക് നൽകുന്നത്. അതായത് വിവിധ കമ്പയിലറുകൾ വ്യത്യസ്തങ്ങളായ ഫയൽ എക്സ്റ്റൻഷനുകൾ പിൻതുടരുന്നു. ഉദാഹരണമായി .cpp, .cxx, .cc, .c++ തുടങ്ങിയ എക്സ്റ്റൻഷനുകൾ ശ്രദ്ധിക്കുക.

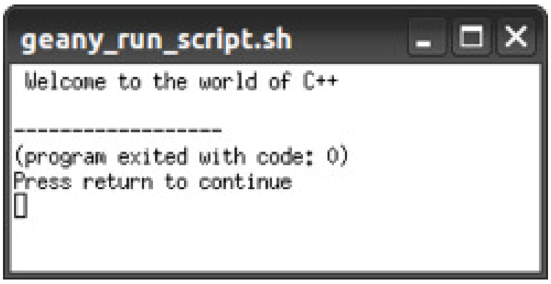
c. പ്രോഗ്രാമിന്റെ കമ്പയിലിങ്ങും ലിങ്കിങ്ങും

പ്രോഗ്രാം കമ്പയിൽ ചെയ്യുകയും, തെറ്റ് കണ്ടെത്തിയാൽ തിരുത്തുകയും ചെയ്യുന്ന ഘട്ടമാണ് അടുത്തത്. ഇതിന് ബിൽഡ് മെനുവിലെ കമ്പയിൽ ഓപ്ഷൻ എടുക്കുകയോ കമ്പയിൽ ബട്ടൺ ഉപയോഗിക്കുകയോ ചെയ്യാം. തെറ്റുകളുണ്ടായാൽ, ആ തെറ്റുകൾ താഴെ ഭാഗത്തുള്ള കമ്പയിലറിന്റെ സ്ഥിതി കാണിക്കുന്ന വിൻഡോയിൽ പ്രദർശിപ്പിക്കുകയും മറിച്ചാണെങ്കിൽ, കമ്പയിലേഷൻ വിജയകരമായി പൂർത്തീകരിച്ചു എന്ന സന്ദേശം പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു (ചിത്രം 5.3 പരിശോധിക്കുക). വിജയകരമായ കമ്പയിലേഷന് ശേഷം ലിങ്ക് ചെയ്യാനായി ടൂൾ ബാറിലെ ബിൽഡ് ബട്ടൺ ക്ലിക്ക് ചെയ്യുകയോ ബിൽഡ് മെനുവിലെ ബിൽഡ് ഓപ്ഷൻ തിരഞ്ഞെടുക്കുകയോ ചെയ്യുക. ഇപ്പോൾ പ്രോഗ്രാം എക്സിക്യൂഷന് തയാറായിക്കഴിഞ്ഞിരിക്കുന്നു.

d. പ്രോഗ്രാം റൺ/എക്സിക്യൂട്ട് ചെയ്യുക.

ഒരു കമ്പ്യൂട്ടർ പ്രോഗ്രാമിലെ നിർദേശങ്ങൾ കമ്പ്യൂട്ടർ പ്രാവർത്തികമാക്കുന്ന പ്രക്രിയയാണ് പ്രോഗ്രാം റണ്ണിംഗ്. പ്രോഗ്രാം റൺ ചെയ്യാനായി ബിൽഡ് മെനുവിലെ എക്സിക്യൂട്ട് ഓപ്ഷൻ

തിരഞ്ഞെടുക്കുക. ടൂൾ ബാറിലുള്ള എക്സിക്യൂട്ട് ബട്ടൺ ക്ലിക്ക് ചെയ്തുകൊണ്ടും പ്രോഗ്രാം എക്സിക്യൂട്ട് ചെയ്യാവുന്നതാണ്. ഔട്ട്പുട്ട് ചിത്രം 5.4-ൽ കാണുന്നത് പോലെ ഒരു പുതിയ വിൻഡോയിൽ പ്രദർശിപ്പിക്കുന്നതാണ്.



ചിത്രം 4.4: ഔട്ട് പുട്ട് വിൻഡോ

e. IDE യുടെ പ്രവർത്തനം അവസാനിപ്പിക്കുക

ഒരിക്കൽ നാം പ്രോഗ്രാം എക്സിക്യൂട്ട് ചെയ്യുകയും പ്രതീക്ഷിച്ച ഔട്ട്പുട്ട് ലഭിക്കുകയും ചെയ്ത് കഴിഞ്ഞാൽ, ഫയൽ മെനുവിൽ നിന്ന് ക്ലോസ് ഓപ്ഷൻ തിരഞ്ഞെടുത്തോ സജീവമായ ടാബിലെ ക്ലോസ് ബട്ടൺ (X) ക്ലിക്ക് ചെയ്തോ പ്രോഗ്രാം അവസാനിപ്പിക്കാം. പ്രോഗ്രാമിങ് തുടരുന്നുവെങ്കിൽ മുമ്പ് വിവരിച്ചത് പോലെ പുതിയ ഫയൽ തുറന്ന് അതിൽ പ്രോഗ്രാം ടൈപ്പ് ചെയ്യാവുന്നതാണ്. അല്ലെങ്കിൽ ടൈറ്റിൽ ബാറിലെ ക്ലോസ് ബട്ടൺ (X) ക്ലിക്ക് ചെയ്ത് IDE- യുടെ പ്രവർത്തനം അവസാനിപ്പിക്കാം. ഇതിനു പുറമേ Ctrl+Q കീ-കൾ ഒരുമിച്ചു ഉപയോഗിച്ചോ ഫയൽ മെനുവിൽ നിന്ന് ക്വിറ്റ് ഓപ്ഷൻ തിരഞ്ഞെടുത്തോ IDE-ൽ നിന്ന് പുറത്തുവരാം.

ഉബുണ്ടു ലിനക്സിലെ ജിനി IDE-യിൽ .cpp എന്ന എക്സ്റ്റൻഷൻ നൽകി സേവ് ചെയ്ത ഒരു സോഴ്സ് പ്രോഗ്രാം കമ്പയിൽ ചെയ്യുമ്പോൾ ലഭിക്കുന്ന ഫയലിന്റെ എക്സ്റ്റൻഷൻ .o (ഒബ്ജക്ട് ഫയൽ) എന്നായിരിക്കും. പ്രോഗ്രാം ലിങ്ക് ചെയ്യുമ്പോൾ ഒബ്ജക്ട് ഫയലിൽ നിന്ന് പ്രൊസസ്സറിന് എക്സിക്യൂട്ട് ചെയ്യാൻ വേണ്ട .out എന്ന എക്സ്റ്റൻഷനോട് കൂടിയ എക്സിക്യൂട്ടബിൾ ഫയൽ ലഭിക്കും. ഈ ഫയലാണ് പ്രൊസസ്സർ എക്സിക്യൂട്ട് ചെയ്യുന്നത്.

നമുക്കു ചെയ്യാം

1. *"SMOKING IS INJURIOUS TO HEALTH" എന്ന സന്ദേശം സ്ക്രീനിൽ പ്രദർശിപ്പിക്കാനായി ഒരു പ്രോഗ്രാം എഴുതുക.*
2. *"TOBOCCO CAUSES CANCER" എന്ന സന്ദേശം മോണിട്ടറിൽ പ്രദർശിപ്പിക്കാനായി ഒരു പ്രോഗ്രാം എഴുതുക*

നമുക്ക് സംഗ്രഹിക്കാം

1980 - കളുടെ തുടക്കത്തിൽ Bjarne Stroustrup ആണ് C++ വികസിപ്പിച്ചത്. C++-ന് അതിന്റേതായ ക്യാരക്ടർ സെറ്റുണ്ട്. ടോക്കൺ എന്നത് പ്രോഗ്രാമിന്റെ ഏറ്റവും ചെറിയ ഘടകവും അവ നിർമ്മിക്കപ്പെട്ടിരിക്കുന്നത് C++-ലെ ഒന്നോ അതിലധികമോ അക്ഷരങ്ങൾ ചേർന്നുമാണ്. കീവേഡുകൾ, ഐഡന്റിഫയറുകൾ, ലിറ്ററലുകൾ, പങ്ചുവേറ്ററുകൾ, ഓപ്പറേറ്ററുകൾ എന്നിങ്ങനെയുള്ള അഞ്ച് തരം ടോക്കണുകളുണ്ട്. പ്രോഗ്രാമുകൾ കമ്പ്യൂട്ടറിൽ എഴുതുന്നത് ഒരു എഡിറ്ററിന്റെ സഹായത്താലാണ്. സോഴ്സ് കോഡ് കമ്പ്യൂട്ടറിൽ നൽകുന്നതിനും അത് കമ്പയിൽ ചെയ്യുന്നതിനും ഒബ്ജക്ട് കോഡ് എക്സിക്യൂട്ട് ചെയ്യുന്നതിനും വേണ്ടി GCC-യും ജിനി IDE-യും പോലുള്ള സോഫ്റ്റ്‌വെയറുകൾ സൗകര്യമൊരുക്കുന്നു.

പഠന നേട്ടങ്ങൾ

ഈ അധ്യായം പൂർത്തിയാക്കിയ പഠിതാവ്

- C++-ലെ ക്യാരക്ടർ സെറ്റ് ലിസ്റ്റ് ചെയ്യുക.
- വിവിധ ടോക്കണുകളെ തരം തിരിക്കുക.
- കീവേഡുകളെ തിരിച്ചറിയുക.
- സാധുവായ ഐഡന്റിഫയറുകൾ എഴുതുക.
- ലിറ്ററലുകൾ തരം തിരിക്കുക.
- ജിനി IDE-യുടെ പ്രധാന ഭാഗങ്ങൾ തിരിച്ചറിയുക.
- ഒരു ലളിതമായ പ്രോഗ്രാം എഴുതി കമ്പയിൽ ചെയ്ത് റൺ ചെയ്യുക.

മാതൃക ചോദ്യങ്ങൾ

വളരെ ലഘുവായ ഉത്തരങ്ങൾ എഴുതുന്നവ.

1. C++ ക്യാരക്ടർ സെറ്റിലെ വിവിധ ക്യാരക്ടറുകൾ എന്തെല്ലാം?
2. എന്താണ് എസ്കേപ്പ് സീക്വൻസ്?
3. ആരാണ് C++ വികസിപ്പിച്ചത്?
4. ടോക്കണുകൾ എന്നാലെന്ത്? C++-ലെ ടോക്കണുകളുടെ പേരെഴുതുക.
5. എന്താണ് C++-ലെ ഒരു ക്യാരക്ടർ സ്ഥിരാങ്കം?
6. എങ്ങനെയാണ് ചിത്രീകരിക്കാനാകാത്ത ക്യാരക്ടറുകൾ C++-ൽ പ്രതിനിധാനം ചെയ്യുന്നത്? ഉദാഹരണം നൽകുക.
7. \(സ്ലാഷ്), ‘ (ഏക ഉദ്ധരണി), “ (ജോഡിയായ ഉദ്ധരണി), ? (ചോദ്യച്ചിഹ്നം) തുടങ്ങിയ ക്യാരക്ടറുകൾ എസ്കേപ്പ് സീക്വൻസുകൾ ഉപയോഗിച്ച് ടൈപ്പ് ചെയ്യുന്നത് എന്ത് കൊണ്ടാണ്?
8. ഏത് എസ്കേപ്പ് സീക്വൻസുകളാണ് ന്യൂലൈൻ ക്യാരക്ടറിനേയും നൾ ക്യാരക്ടറിനെയും പ്രതിനിധാനം ചെയ്യാൻ ഉപയോഗിക്കുന്നത്?
9. ഒരു എസ്കേപ്പ് സീക്വൻസ് .... ക്യാരക്ടറുകളെ പ്രതിനിധാനം ചെയ്യുന്നു.
10. താഴെ തന്നിട്ടുള്ളവയിൽ ഏതൊക്കെയാണ് C++-ലെ സാധുവായ ക്യാരക്ടറുകൾ/സ്ട്രിങ് സ്ഥിരാംഗങ്ങൾ?

    ‘c’   ‘anu’   “anu”   mine   ‘min’s’   “ ”

    ‘char   ‘\ \’

11. എന്താണ് ഫ്ളോട്ടിങ് പോയിന്റ് സ്ഥിരാംഗം? അവ പ്രതിനിധാനം ചെയ്യുന്നതിനുള്ള വിവിധ മാർഗങ്ങൾ എന്തെല്ലാം?
12. എന്താണ് C++-ലെ സ്ട്രിങ് ലിറ്ററലുകൾ? ക്യാരക്ടർ സ്ഥിരാംഗങ്ങളും സ്ട്രിങ് ലിറ്ററലുകളും തമ്മിലുള്ള വ്യത്യാസമെന്ത്?
13. റൺ ചെയ്യുവാനുപയോഗിക്കുന്ന ഒരു C++ ഫയലിന്റെ എക്സ്റ്റൻഷൻ എന്താണ്?
14. താഴെ തന്നിട്ടുള്ളവയിൽ നിന്ന് അസാധുവായ ഐഡന്റിഫയറുകൾ കണ്ടെത്തുക. അസാധുവായതിന് കാരണമെഴുതുക.

    (a) Principal amount (b) Continue (c) Area (d) Date-of-join (e) 9B
15. ലേബൽ എന്നത് C++-ലെ ഒരു ....ആണ്

    (a) കീവേർഡ് (b) ഐഡന്റിഫയർ (c) ഓപ്പറേറ്റർ (d) ഫങ്ഷൻ
16. താഴെത്തന്നിട്ടുള്ള ടോക്കണുകൾ C++ പ്രോഗ്രാമിൽ നിന്ന് എടുത്തിട്ടുള്ളവയാണ്. താഴെ തന്നിട്ടുള്ള പട്ടികയിൽ അവ യഥാസ്ഥാനത്ത് നിരത്തുക.

    (int, cin, %, do, =, "break", 25.7, digit)

കീവേഡുകൾ	ഐഡന്റിഫയറുകൾ	ലിറ്ററലുകൾ	ഓപ്പറേറ്ററുകൾ

ലഘു ഉത്തരങ്ങളെഴുതുന്നവ

1. ഐഡന്റിഫയറുകളെ പരിപാലിക്കുന്ന നിയമങ്ങളെഴുതുക.
2. എന്താണ് C++-ലെ ടോക്കണുകൾ? എത്ര തരം ടോക്കണുകൾ C++-ൽ അനുവദിക്കുന്നു? അവ ലിസ്റ്റ് ചെയ്യുക.
3. കീവേഡുകളും, ഐഡന്റിഫയറുകളും വേർതിരിക്കുക
4. പൂർണ സംഖ്യാ സ്ഥിരാംഗങ്ങളെ C++-ൽ എങ്ങനെ പ്രതിനിധാനം ചെയ്യുന്നു? ഉദാഹരണ സഹിതം വിശദീകരിക്കുക
5. C++-ലെ ക്യാരക്ടർ സ്ഥിരാംഗങ്ങൾ എന്തെല്ലാം? എങ്ങനെ അവ പ്രയോഗത്തിൽ വരുത്തുന്നു?

വിവരണാത്മക ഉത്തരങ്ങളെഴുതുന്നവ

1. വിവിധയിനം ടോക്കണുകളെക്കുറിച്ച് ചുരുക്കി വിവരിക്കുക.
2. വിവിധയിനം ലിറ്ററലുകളെക്കുറിച്ച് ഉദാഹരണ സഹിതം വിശദീകരിക്കുക.
3. ജിനി IDE-യെക്കുറിച്ചും അതിന്റെ പ്രത്യേകതകളെക്കുറിച്ചും ചുരുക്കി വിശദീകരിക്കുക.

5

പ്രധാന ആശയങ്ങൾ

- ഡാറ്റ ഇനങ്ങൾ എന്ന ആശയം
- C++ ഡാറ്റ ഇനങ്ങൾ
- അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ
- ടൈപ്പ് മോഡിഫയറുകൾ
- വേരിയബിളുകൾ
- ഓപ്പറേറ്ററുകൾ
  - അരിത്മറ്റിക്
  - റിലേഷണൽ
  - ലോജിക്കൽ
  - ഇൻപുട്ട്/ഔട്ട്പുട്ട്
  - അസൈൻമെന്റ്
- എക്സ്പ്രഷനുകൾ
  - അരിതമെറ്റിക്
  - റിലേഷണൽ
  - ലോജിക്കൽ
- പ്രസ്താവനകൾ
  - പ്രഖ്യാപനം
  - അസൈൻമെന്റ്
  - ഇൻപുട്ട്/ഔട്ട്പുട്ട്

T5J9E4

ഡാറ്റയുടെ ഇനങ്ങളും ഓപ്പറേറ്ററുകളും

ഇതിനു മുൻപുള്ള അധ്യായത്തിൽ C++ ന്റെ അടിസ്ഥാന നിർമ്മാണ ഘടകങ്ങളും പ്രോഗ്രാം വികസിപ്പിക്കുന്നതിനുപയോഗിക്കുന്ന IDE യും നാം പരിചയപ്പെട്ടു കഴിഞ്ഞു. കമ്പ്യൂട്ടറുകളിൽ നടക്കുന്ന പ്രധാന പ്രവർത്തനം ഡാറ്റ പ്രോസസ്സിംഗ് ആണെന്ന് നമുക്കറിയാമല്ലോ. എല്ലാ പ്രോഗ്രാമ്മിംഗ് ഭാഷകളും ഡാറ്റ കൈകാര്യം ചെയ്യുന്നതിന് പ്രാധാന്യം നൽകുന്നുണ്ട്. ചില പ്രത്യേകമായ സങ്കേതങ്ങൾ ഉപയോഗിച്ചുകൊണ്ടാണ് കമ്പ്യൂട്ടറുകളിൽ ഇൻപുട്ട് ചെയ്യപ്പെടുന്ന ഡാറ്റയുടെ ക്രമീകരണവും സംഭരണവും നടക്കുന്നത്. ഡാറ്റ സംഭരിക്കുന്നതിന് C++ന് മുൻകൂട്ടി നിർവചിച്ച ഒരു രൂപരേഖയുണ്ട്. സംഭരിക്കപ്പെട്ട ഡാറ്റ പിന്നീട് ഓപ്പറേറ്ററുകൾ ഉപയോഗിച്ച് പ്രോസസ്സ് ചെയ്യപ്പെടുന്നു. ഉപയോക്തൃ നിർവചിത ഡാറ്റ ഇനങ്ങൾ എന്നു വിളിക്കുന്ന പുതിയ ഡാറ്റ ഇനങ്ങൾ നിർവചിക്കുന്നതിന് ഉപയോക്താവിനെ C++ അനുവദിക്കുന്നു.

ഇ++ഭാഷയുടെ മുഖ്യ ആശയങ്ങളായ ഡാറ്റ ഇനങ്ങൾ, ഓപ്പറേറ്ററുകൾ, പദപ്രയോഗങ്ങൾ, പ്രസ്താവനകൾ എന്നിവയെക്കുറിച്ച് നമുക്ക് ഈ അദ്ധ്യായത്തിൽ വിശദമായി പഠിക്കാം.

5.1 ഡാറ്റ ഇനങ്ങൾ എന്ന ആശയം (Concept of data types)

പരീക്ഷയ്ക്കുശേഷം ഒരു വിദ്യാർത്ഥിയുടെ പ്രോഗ്രസ് കാർഡ് തയ്യാറാക്കുന്നതിനായി നമുക്ക് അയാളുടെ രജിസ്റ്റർ നമ്പർ, റോൾ നമ്പർ, പേര്, വിലാസം, വിവിധ വിഷയങ്ങൾക്ക് ലഭിച്ച സ്കോർ, ഗ്രേഡുകൾ തുടങ്ങിയ ഡാറ്റ ആവശ്യമുണ്ട്. ഇത് കൂടാതെ, വിദ്യാർത്ഥിയുടെ സ്കോർ, ഹാജർ എന്നിവ ശതമാനത്തിൽ പ്രദർശിപ്പിക്കേണ്ടതുണ്ട്. ശാസ്ത്ര സംബന്ധിയായ ഡാറ്റ പ്രോസസിംഗ് പരിഗണിക്കുമ്പോൾ പ്രകാശത്തിന്റെ വേഗതയായ ($3x10^8$m/s), ഗുരുത്വാകർഷണത്തിന്റെ വിലയായ (9.8 m/s^2), ഇലക്ട്രോണിന്റെ ഇലക്ട്രിക് ചാർജായ ($-1.6x10^{-19}$c), തുടങ്ങിയ

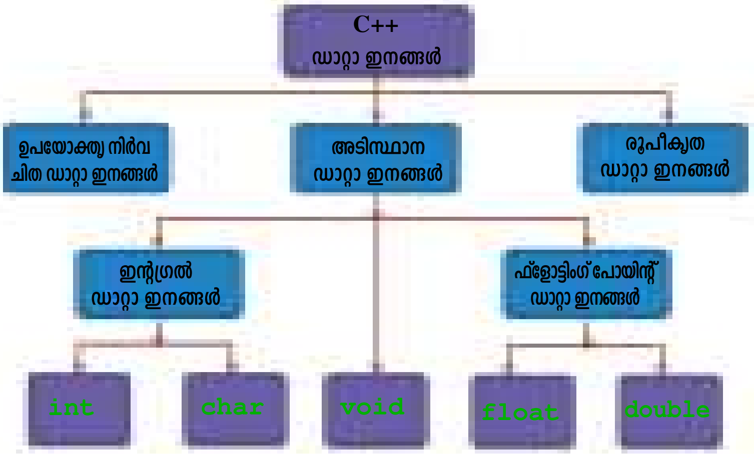
അക്കങ്ങളുടെ രൂപത്തിലുള്ള ഡാറ്റ ആവശ്യമായി വരാം.

മേൽ പറഞ്ഞതിൽ നിന്ന്, ക്യാരക്ടർ (character), ഇന്റിജർ (integer), ഭിന്ന സംഖ്യ (Real Number), വാക്കുകൾ (string) മുതലായ വിവിധതരം ഡാറ്റയുണ്ടെന്ന് നമുക്ക് അനുമാനിക്കാം. C++ലെ സാധുവായ ഒരു അക്ഷരം ഒരു ജോഡി ഏക ഉദ്ധരണികൾക്കുള്ളിൽ (single quotes) രേഖപ്പെടുത്തിയാൽ അത് ഒരു ക്യാരക്ടർ ഡാറ്റയെ പ്രതിനിധീകരിക്കുന്നു എന്ന് നാം കഴിഞ്ഞ അദ്ധ്യായത്തിൽ പഠിച്ചതാണല്ലോ. അതുപോലെ ദശാംശസ്ഥാനമില്ലാത്ത സംഖ്യകൾ ഇന്റിജർ (integer) ഡാറ്റയെ പ്രതിനിധാനം ചെയ്യുന്നു, ഭിന്നസംഖ്യകൾ ഫ്ളോട്ടിംഗ് പോയിന്റ് (floating point) ഡാറ്റ എന്നും, ഇരട്ട ഉദ്ധരണികളിൽ രേഖപ്പെടുത്തിയിരിക്കുന്നവ സ്ട്രിംഗ് (string) ഡാറ്റ എന്നും അറിയപ്പെടുന്നു. വിവിധ തരം ഡാറ്റ കൈകാര്യം ചെയ്യേണ്ടതിനാൽ എല്ലാ പ്രോഗ്രാമിംഗ് ഭാഷകളും അതിനുള്ള സംവിധാനം നൽകേണ്ടതാണ്. ഡാറ്റ ഇനങ്ങൾക്ക് പേരുകൾ നൽകികൊണ്ട് വിവിധതരം ഡാറ്റ കൈകാര്യം ചെയ്യുന്നതിനുള്ള സംവിധാനം C++ നൽകുന്നു. ഡാറ്റ ഇനങ്ങൾ (data types) എന്നാൽ ഡാറ്റയുടെ സ്വഭാവം, അതിൻമേൽ നടത്തുന്ന പ്രവർത്തനങ്ങൾ എന്നിവ തിരിച്ചറിയുന്നതിനുള്ള ഉപാധിയാണ്. ഈ സവിശേഷതകൾ വേർതിരിക്കുന്നതിനായി C++-ൽ വിവിധ ഡാറ്റ ഇനങ്ങൾ നിർവചിച്ചിരിക്കുന്നു.

അദ്ധ്യായം നാലിലെ അൽഗോരിതങ്ങളിൽ ഡാറ്റയെ സൂചിപ്പിക്കുവാൻ വേരിയബിളുകളാണ് നാം ഉപയോഗിച്ചത്. പ്രോഗ്രാമുകളിലും വേരിയബിളുകൾ തന്നെയാണ് ഡാറ്റയെ സൂചിപ്പിക്കുവാൻ ഉപയോഗിക്കുന്നത്. C++ ഭാഷയിൽ നാം പ്രോഗ്രാമുകൾ എഴുതുമ്പോൾ വേരിയബിളുകളെ അവ ഉപയോഗിക്കുന്നതിനുമുമ്പായി പ്രഖ്യാപിക്കേണ്ടതുണ്ട് (declaration of variable) ഈ വേരിയബിളുകൾ പ്രഖ്യാപിക്കുന്നതിന് ഡാറ്റ ഇനങ്ങൾ ആവശ്യമാണ്.

5.2 C++ ലെ ഡാറ്റ ഇനങ്ങൾ (C++ Data Types)

C++ വിവിധതരം ഡാറ്റാഇനങ്ങൾ കൊണ്ട് സമ്പുഷ്ടമാണ്. ഇവയെ സ്വഭാവം, വലിപ്പം, അവയുമായി ബന്ധപ്പെട്ട പ്രവർത്തനങ്ങൾ എന്നിവയുടെ അടിസ്ഥാനത്തിൽ ചിത്രം 6.1ൽ കാണുന്നതു പോലെ പലതായി തരം തിരിച്ചിട്ടുണ്ട്. ഡാറ്റ ഇനങ്ങളെ അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ അല്ലെങ്കിൽ അന്തർ നിർമിത ഡാറ്റ ഇനങ്ങൾ (built-in data types), രൂപീകൃത ഡാറ്റ ഇനങ്ങൾ (Derived data types), ഉപയോക്ത നിർവചിത ഡാറ്റ ഇനങ്ങൾ (user defined data types) എന്നിങ്ങനെ തരം തിരിച്ചിരിക്കുന്നു.



ചിത്രം 5.1 : C++ ഡാറ്റാ ഇനങ്ങളുടെ തരംതിരിക്കൽ

അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ: C++ കംപൈലറിൽ അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ നിർവ്വചിച്ചിരിക്കുന്നു. അന്തർ നിർമിത ഡാറ്റ ഇനങ്ങൾ എന്നും അവ അറിയപ്പെടുന്നു. ഇവ വീണ്ടും വിഭജിക്കുവാൻ കഴിയാത്ത ഏറ്റവും ചെറിയ ഘടകങ്ങളാണ്. char, int, float, double, void എന്നിവയാണ് C++ലെ അഞ്ച് അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ. ഇവയിൽ int, char, എന്നിവ പൂർണസംഖ്യാ ഡാറ്റ ഇനത്തിനു കീഴിൽ വരുന്നതാണ്. പൂർണ്ണ സംഖ്യകളെ മാത്രമേ ഇവയ്ക്ക് കൈകാര്യം ചെയ്യാൻ കഴിയുകയുള്ളൂ. ഭിന്നസംഖ്യകൾ സാധാരണയായി ദശാംശ സംഖ്യാ ഡാറ്റ ഇനം (floating point data type) എന്ന് അറിയപ്പെടുന്നു. ഇവയെ വ്യാപ്തിയുടെയും (range) കൃത്യതയുടെയും (presion) അടിസ്ഥാനത്തിൽ float, double എന്നിങ്ങനെ രണ്ടായി തരം തിരിച്ചിരിക്കുന്നു.

ഉപോയോക്തൃ നിർവചിത ഡാറ്റ ഇനങ്ങൾ: പ്രോഗ്രാമർക്ക് സ്വന്തമായി ഡാറ്റ ഇനം നിർവചിക്കുവാനുള്ള സൗകര്യം C++ൽ ഉണ്ട്. സ്ട്രക്ച്ചർ (struct), എന്യൂമറേഷൻ (enum), യൂണിയൻ (union), ക്ലാസ്സ് (class) തുടങ്ങിയവ ഇത്തരം ഡാറ്റ ഇനങ്ങൾക്കുള്ള ഉദാഹരണങ്ങളാണ്.

രൂപീകൃത ഡാറ്റ ഇനങ്ങൾ: അടിസ്ഥാന ഡാറ്റ ഇനങ്ങളെ ഗണങ്ങൾ ആക്കി മാറ്റിയോ വലിപ്പ മാറ്റം വരുത്തിയോ നിർമ്മിക്കപ്പെടുന്ന ഡാറ്റ ഇനങ്ങൾ രൂപീകൃത ഡാറ്റ ഇനങ്ങൾ എന്ന് അറിയപ്പെടുന്നു. അറെകൾ, പോയിന്ററുകൾ, ഫങ്ഷനുകൾ തുടങ്ങിയവ രൂപീകൃത ഡാറ്റ ഇനങ്ങൾക്കുള്ള ഉദാഹരണങ്ങളാണ്.

5.3 അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ (fundamental data types):

അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ മൗലികമായ സ്വഭാവ വിശേഷമുള്ളവയാണ്. അവയെ വീണ്ടും ചെറിയ ഭാഗങ്ങളായി വിഭജിക്കുവാൻ കഴിയില്ല. കമ്പൈലറിൽ നിർചിക്കപ്പെട്ടിരിക്കുതിനാൽ അവയുടെ വലിപ്പം (അനുവദിക്കപ്പെട്ട മെമ്മറിയുടെ അളവ്) കമ്പൈലറിനെ ആശ്രയിച്ചിരിക്കുന്നു. നാം ഉപയോഗിക്കുന്നത് GCC -ൽ ലഭ്യമായ കമ്പൈലർ ആയതിനാൽ ഡാറ്റയുടെ വലിപ്പവും അതുപോലെ ഡാറ്റയുടെ വ്യാപ്തിയും അതിന് അനുസൃതമായിരിക്കും. ടർബോ C++ IDE പോലുള്ള കംപയിലറുകൾ നിങ്ങൾ ഉപയോഗിക്കുമ്പോൾ ഇത് വ്യത്യസ്തമാകാം. അഞ്ചു അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ താഴെ വിശദീകരിച്ചിരിക്കുന്നു.

int ഡാറ്റഇനം (പൂർണ്ണ സംഖ്യകൾക്കായി): ദശാംശ ഭാഗമില്ലാത്ത പൂർണ്ണ സംഖ്യകളാണ് ഇന്റിജറുകൾ. ഇവ പോസിറ്റീവോ, പൂജ്യമോ, നെഗറ്റീവോ ആകാം. ഒരു പ്രത്യേക പരിധിക്കുള്ളിലുള്ള ഇന്റിജറുകളെ പ്രതിനിധീകരിക്കാൻ ഉപയോഗിക്കുന്ന കീ വേർഡ് ആണ് int. int ഇനത്തിലുള്ള ഇന്റിജറുകൾക്ക് GCC നാല് ബൈറ്റ് മെമ്മറി അനുവദിക്കുന്നു. ആയതിനാൽ -2147483648 മുതൽ +2147483647 വരെയുള്ള സംഖ്യകളെ int ഡാറ്റാ ഇനം ഉപയോഗിച്ച് സൂചിപ്പിക്കാം. 69, 0, -112, 17, -32768, +32767 എന്നിവ int ഡാറ്റാ ഇനത്തിനുള്ള ഉദാഹരണങ്ങളാണ്. 22000000.00, - 2147483649 എന്നിവ അനുവദനീയമായ പരിധിക്ക് പുറത്തായതിനാൽ int ഡാറ്റാ ഇനമായി പരിഗണിക്കുകയില്ല.

char ഡാറ്റ ഇനം (ക്യാരക്ടർ സ്ഥിരാംഗങ്ങൾക്കുവേണ്ടി) : C++ ഭാഷയിലെ ക്യാരക്ടർ സെറ്റിലുൾപ്പെടുന്ന ചിഹ്നങ്ങൾ ആണ് ക്യാരക്ടറുകൾ. എല്ലാ അക്ഷരങ്ങൾ അക്കങ്ങൾ പ്രത്യേക ചിഹ്നങ്ങൾ വിരാമ ചിഹ്നങ്ങൾ തുടങ്ങിയവ ഈ വിഭാഗത്തിൽ ഉൾപ്പെടുന്നു.

ഈ ക്യാരക്ടറുകൾ ഡാറ്റയായി ഉപയോഗിക്കുമ്പോൾ അവയെ C++ ലെ char ഡാറ്റഇനമായി പരിഗണിക്കപ്പെടുന്നു. Char കീ വേർഡ് C++ ലെ ക്യാരക്ടർ ലിറ്ററലുകളെ പ്രതിനിധീകരിക്കുന്നു എന്നു നമുക്ക് പറയാം.ഓരോ char ഇനത്തിലുള്ള ഡാറ്റായ്ക്കും 1 ബൈറ്റ് മെമ്മറി അനുവദിക്കുന്നു. 'a', '+', '/t', '0' മുതലായവ char ഡാറ്റാ ഇനത്തിൽപ്പെടുന്നവയാണ്.

char ഡാറ്റയെ ഇന്റിജർ ആയിട്ടാണ് പരിഗണിക്കുന്നത് എന്തുകൊണ്ടെന്നാൽ ASCII കോഡ് ഉപയോഗിച്ചാണ് കമ്പ്യൂട്ടർ ക്യാരക്ടറുകളെ തിരിച്ചറിയുന്നത്. മെമ്മറിയിൽ ക്യാരക്ടർ ഡാറ്റാ സംഭരിക്കുന്നത് അതിന്റേതായ ASCII കോഡ് ഉപയോഗിച്ചാണ്. ASCII കോഡ് ഇന്റിജറായതിനാലും അവ 8 ബിറ്റ്/1 ബൈറ്റ് സ്ഥലത്ത് സംഭരിക്കപ്പെടേണ്ടതിനാലും char ഡാറ്റാ ഇനത്തിന്റെ പരിധി -128 മുതൽ +127 വരെയാണ്.

float ഡാറ്റഇനം (ദശാംശ സംഖ്യകൾക്കായി): ദശാംശ ഭാഗത്തോടുകൂടിയ സംഖ്യകളെ ഫ്ളോട്ടിംഗ് പോയിന്റ് സംഖ്യകളെന്നു വിളിക്കുന്നു. കമ്പ്യൂട്ടറിൽ ഫ്ളോട്ടിംഗ് പോയിന്റ് സംഖ്യകൾ രേഖപ്പെടുത്തുന്നത് ശാസ്ത്രീയമായ പ്രതീകങ്ങൾ ഉപയോഗിച്ചാണ്. ഉദാഹരണത്തിന് -47281.97 എന്ന് സംഖ്യയെ ശാസ്ത്രീയ പ്രതീകങ്ങൾ ഉപയോഗിച്ചാണ് എഴുതുന്നത് 0.4728197×10^5 എന്നാണ്. ഇതിന്റെ ആദ്യഭാഗത്തെ (0.4728197) ഭിന്നാംശം (mantissa) എന്നും പത്തിന്റെ വർഗ്ഗമായ 5നെ (10^5) കൃത്യംങ്കം (exponent) എന്നും പറയുന്നു. ഫ്ളോട്ടിംഗ് പോയിന്റ് വിലകളെ പ്രതിനിധീകരിക്കാൻ കമ്പ്യൂട്ടർ സാധാരണയായി കൃത്യങ്ക മാതൃക (E-notation) ഉപയോഗിക്കുന്നു. അത് പ്രകാരം 47281.97 എന്നത് 0.4728197E5 ആയാണ് രേഖപ്പെടുത്തുന്നത്. Eക്ക് മുമ്പുള്ള സംഖ്യ ഭിന്നാംശവും Eക്ക് ശേഷമുള്ള സംഖ്യ കൃത്യങ്കവുമാണ്. C++ൽ float എന്ന കീ വേർഡാണ് ഇത്തരം സംഖ്യകളെ സൂചിപ്പിക്കാൻ ഉപയോഗിക്കുന്നത്. float ഡാറ്റ ഇനത്തിൽപ്പെട്ട സംഖ്യകൾക്ക് 4 ബൈറ്റ് മെമ്മറി GCC അനുവദിക്കുന്നു. ഈ ഡാറ്റ തരത്തിലുള്ള സംഖ്യകൾക്ക് സാധാരണയായി ദശാംശത്തിനുശേഷം 7 അക്കങ്ങൾ വരെ ആവാം.

Double ഡാറ്റ ഇനം (ഡബിൾ പ്രിസിഷൻ ദശാംശ സംഖ്യകൾക്കായി): ദശാംശത്തിനുശേഷം കൂടുതൽ അക്കങ്ങൾ വേണ്ട സംഖ്യകൾക്ക് അതായത് കൂടതുൽ കൃത്യത വേണ്ട ദശാംശ സംഖ്യകൾക്ക് ഉപയോഗിക്കുന്ന ഡാറ്റ ഇനമാണ് double. ഫ്ളോട്ട് ഡാറ്റാ ഇനം ഉപയോഗിച്ച് കൈകാര്യം ചെയ്യാവുന്ന സംഖ്യകളുടെ പരിധി ഈ ഡാറ്റാ ഇനം ഉപയോഗിച്ച് വർധിപ്പിക്കാവുന്നതാണ്. എന്തുകൊണ്ടെന്നാൽ ഈ ഡാറ്റാ ഇനം ഫ്ളോട്ട് ഡാറ്റാ ഇനത്തെക്കാൾ ഇരട്ടി മെമ്മറി ഉപയോഗിക്കുന്നു. C++ ൽ double ന്റെ കൃത്യതയും പരിധിയും കുറഞ്ഞത് float ന്റെ അത്രയെങ്കിലും ആയിരിക്കണം. gcc ഇത്തരത്തിലുള്ള ഡാറ്റ സംഭരിക്കുന്നതിന് 8 ബൈറ്റ് മെമ്മറി നീക്കിവച്ചിരിക്കുന്നു. ഡബിൾ ഡാറ്റാ ഇനത്തിൽ ദശാംശ സ്ഥാനത്തിനു ശേഷം 15 അക്കങ്ങൾ വരെ ആകാം.

void ഡാറ്റ തരം (എം.റ്റി. സെറ്റ് ഡാറ്റക്കായി): എംറ്റി സെറ്റിലെ ഡാറ്റയെ സൂചിപ്പിക്കാൻ ഉപയോഗിക്കുന്ന കീ വേഡാണ് വോയിഡ് (void). തീർച്ചയായും ഇതിന് മെമ്മറി ആവശ്യമില്ല. ഈ ഡാറ്റാ ഇനത്തിന്റെ വിശദമായ ഉപയോഗം അധ്യായം 10 ൽ ചർച്ച ചെയ്യാം.

അടിസ്ഥാന ഡാറ്റ ഇനങ്ങളെ അവയുടെ വലിപ്പത്തിന്റെ അവരോഹണ ക്രമത്തിൽ double, float, int, char.void എന്ന് ക്രമീകരിക്കാം.

Name പേര്	Description വിശദീകരണം	Size വലിപ്പം	Range പരിധി
`char`	Character	1 byte	`signed`: -128 to 127 `unsigned`: 0 to 255
`int`	Integer	4 bytes	`signed`: -2147483648 to 2147483647 `unsigned`: 0 to 4294967295
`float`	Floating point number	4 bytes	$-3.4\times10^{+/-38}$ to $+3.4 \times 10^{+/-38}$ with approximately 7 significant digits
`double`	Double precision floating point number	8 bytes	$-1.7 \times 10^{+/-308}$ to $+1.7 \times 10^{+/-308}$ with approximately 15 significant digits
`void`	Null data	0 bytes	empty set

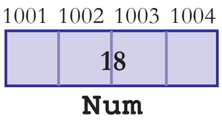
പട്ടിക 5.1: ഡാറ്റ ഇനങ്ങളും ടൈപ്പ് മോഡിഫയറുകളും

> *ഡാറ്റാഇനങ്ങൾ എങ്ങനെ വ്യത്യാസപ്പെട്ടിരിക്കുന്നു എന്ന് നിങ്ങൾക്ക് അറിയുന്നതിനുള്ള ഉദാഹരണങ്ങളാണ് ടേബിൾ 5.1 ൽ കൊടുത്തിരിക്കുന്നത്. ഇതിലുള്ള പല വിലകളും നിങ്ങളുടെ കമ്പ്യൂട്ടറിൽ വ്യത്യസ്തമായിരിക്കാം*

5.4 വേരിയബിളുകൾ (variables)

ഡാറ്റ പരാമർശിക്കുന്നതിന് മെമ്മറിയിൽ അതിന്റെ സ്ഥാനങ്ങൾ തിരിച്ചറിയേണ്ടതുണ്ട്. മെമ്മറി സ്ഥാനങ്ങൾക്ക് നൽകുന്ന പേരുകളാണ് വേരിയബിളുകൾ. മെമ്മറി സ്ഥാനങ്ങളിൽ ഡാറ്റയെ സ്റ്റോർ ചെയ്യാനും വീണ്ടെടുക്കാനും ഉപയോഗിക്കുന്ന C++ -ലെ ഐഡന്റിഫയറുകളാണ് വേരിയബിളുകൾ. ഒരു വേരിയബിളിൽ സ്റ്റോർ ചെയ്തിട്ടുള്ള ഡാറ്റയുടെ സ്വഭാവവും അതിന്റെ വലിപ്പവും ആ വേരിയബിളിന്റെ ഡാറ്റ ഇനത്തിന് അനുസരിച്ചിരിക്കും. ഒരു വേരിയബിളിന് മൂന്ന് പ്രധാനപ്പെട്ട സ്വഭാവ സവിശേഷതകളുണ്ട്.

i. **വേരിയബളിന്റെ പേര് (variable name):** മെമ്മറിയിലെ ഒരു സ്ഥലത്ത് സംഭരിച്ചിരിക്കുന്ന ഡാറ്റ പരാമർശിക്കുന്നതിന് വേണ്ടി പ്രതീകാത്മകമായ ഉപയോഗിക്കുന്ന പേരാണിത്.



ചിത്രം 5.2 : ഒരു വേരിയബിളിന്റെ മെമ്മറി പ്രതിനിധാനം

ii. **മെമ്മറി വിലാസം (memory address):** ഒരു ബൈറ്റു ഡാറ്റ വീതം സംഭരിക്കാൻ കഴിയുന്ന സെല്ലുകളുടെ (cell) ശേഖരമാണ് കമ്പ്യൂട്ടറിന്റെ RAM. RAM ലുള്ള ഓരോ സെല്ലും (ബൈറ്റ്) ഉപയോഗിക്കുന്നതിന് അവയ്ക്ക് തനതായ വിലാസങ്ങൾ നൽകപ്പെട്ടിരിക്കുന്നു. എല്ലാ വേരിയബിളുകളും RAM ലുള്ള ഒന്നോ അതിലധികമോ മെമ്മറി സ്ഥാനങ്ങളുമായി ബന്ധപ്പെട്ടിരിക്കുന്നു. അനുവദിച്ചിട്ടുള്ള മെമ്മറിയുടെ ആരംഭത്തിലെ സെല്ലിന്റെ വിലാസത്തെ പ്രാരംഭ വിലാസം (base address) എന്നു പറയുന്നു. സാധാരണഗതിയിൽ ഈ വിലാസം അനുവദിക്കുന്നത് കംപൈലർ ആണ്. ഈ വിലാസത്തെ വേരിയബിളിന്റെ എൽ മൂല്യം (L-Value) എന്നും വിളിക്കുന്നു. ചിത്രം 6.2 ൽ Num വേരിയബിളിന്റെ പ്രാരംഭ വിലാസം 1001 ആണ്.

iii. **ഉള്ളടക്കം (Content):** ഒരു മെമ്മറി സ്ഥാനത്ത് സംഭരിച്ചിരിക്കുന്ന മൂല്യത്തെ വേരിയബിളിന്റെ ഉള്ളടക്കം എന്ന് വിളിയ്ക്കുന്നു. ഇതിനെ വേരിയബിളിന്റെ ആർ. മൂല്യം (R-value) എന്നും വിളിയ്ക്കുന്നു. ഉള്ളടക്കത്തിന്റെ തരവും വലിപ്പവും വേരിയബിളിന്റെ ഡാറ്റാ ഇനത്തെ ആശ്രയിച്ചിരിക്കുന്നു.

ചിത്രം 6.2 ഒരു വേരിയബിളിന്റെ മെമ്മറിയിലെ പ്രതിനിധാനം കാണിച്ചിരിക്കുന്നു. ഇവിടെ Num എന്നതു വേരിയബിളിന്റെ പേരും 1001, 1002, 1003, 1004 എന്നീ നാലു മെമ്മറി വിലാസങ്ങൾ ഉപയോഗിക്കുന്ന 4 ബൈറ്റ് മെമ്മറിയുമാണ്. ഈ വേരിയബിളിന്റെ ഉള്ളടക്കം 18 ആണ്. അതായത് Num ന്റെ L മൂല്യം. 1001 ഉം R മൂല്യം 18 ഉം ആണ്.

5.5. ഓപ്പറേറ്ററുകൾ (Operators):

കമ്പ്യൂട്ടറുകളിൽ പ്രവർത്തനങ്ങൾ (operations) നടപ്പിലാക്കുന്നതിന് പ്രേരിപ്പിക്കുന്ന മുൻകൂട്ടി നിശ്ചയിച്ചിട്ടുള്ള ചിഹ്നങ്ങളാണ് ഓപ്പറേറ്ററുകൾ. ഒരു ഓപ്പറേഷനിൽ പങ്കെടുക്കുന്നവയെ ഓപ്പറാൻഡ്സ് (operands) എന്നു വിളിക്കുന്നു. ഒരു ഓപ്പറന്റ് വേരിയയബിളോ സ്ഥിരാംഗമോ ആകാം.

ഉദാഹരണത്തിന് a+b എന്ന അരിത്മെറ്റിക് ഓപ്പറേഷനിൽ + (സങ്കലനം) ഓപ്പറേറ്ററും, a, b എന്നിവ ഓപ്പറാൻഡുകളും ആണ്. വിവിധ മാനദണ്ഡങ്ങൾക്കനുസൃതമായി C++ലെ ഓപ്പറേറ്ററുകളെ തരം തിരിച്ചിരിക്കുന്നു. C++ൽ ഓപ്പറേഷനുപയോഗിക്കുന്ന ഓപ്പറാൻഡ്കളുടെ എണ്ണം അനുസരിച്ച് ഓപ്പറേറ്ററുകളെ യൂനറി (unary), ബൈനറി (binary), ടെറിനറി (terinary) എിങ്ങനെ മൂന്നായി തരം തിരിച്ചിരിക്കുന്നു.

യൂനറി ഓപ്പറേറ്ററുകൾ (Unary Operators): ഒരു ഓപ്പറാന്റ് മാത്രമുള്ള പ്രവർത്തനങ്ങളിലെ ഓപ്പറേറ്ററുകളാണ് യൂനറി ഓപ്പറേറ്ററുകൾ. ഒരു സംഖ്യ പോസിറ്റീവ് അല്ലെങ്കിൽ നെഗറ്റീവ് എന്നു കാണിക്കാനുപയോഗിക്കുന്ന +, (-) ചിഹ്നങ്ങളാണ് സാധാരണ ഉപയോഗിക്കുന്ന യൂനറി ഓപ്പറേറ്ററുകൾ ചിഹ്നത്തോടു കൂടിയ ഒരു നമ്പറിന് മുൻപിൽ + ഓപ്പറേറ്റർ നൽകുമ്പോൾ നിലവിലുള്ള ചിഹ്നത്തിന് മാറ്റമുണ്ടാകുന്നില്ല. എന്നാൽ - നൽകുമ്പോൾ വിലയിൽ മാറ്റമുണ്ടാകുന്നു. ചിഹ്നത്തോടു കൂടിയ ഒരു സംഖ്യയിൽ യുനറി ഓപ്പറേറ്റർ നാം പ്രയോഗിച്ചാൽ സംഖ്യയുടെ നിലവിലുള്ള ചിഹ്നം നേരേ വിപരീതമാകുന്നു. യുനറി ഓപ്പറേറ്ററിന്റെ ഉപയോഗം പട്ടിക 6.2ൽ കൊടുത്തിരിക്കുന്നു.

Variable x	Unary + +x	Unary- -x
8	8	-8
0	0	0
-9	-9	9

പട്ടിക 5.2 : യുനറി ഓപ്പറേറ്ററുകൾ

ഇൻക്രിമന്റ് (increment) ++ (decriment) -- എന്നിവയും യുനറി ഓപ്പറേറ്ററുകൾക്ക് ഉദാഹരണങ്ങളാണ്.

ബൈനറി ഓാപ്പറേറ്ററുകൾ (Binary Operator): ബൈനറി ഓപ്പറേറ്ററുകൾ രണ്ട് ഓപ്പറാൻഡുകളിൽ പ്രവർത്തിക്കുന്നു. അരിത്മെറ്റിക് ഓപ്പറേറ്ററുകൾ (arithmatic), റിലേഷണൽ ഓപ്പറേറ്ററുകൾ (relational), ലോജിക്കൽ ഓപ്പറേറ്ററുകൾ (logical) മുതലായവയാണ് സാധാരണയായി ഉപയോഗിക്കുന്ന ബൈനറി ഓപ്പറേറ്ററുകൾ.

ടെറിനറി ഓാപ്പറേറ്റർ (Terinary operator): ടെർനറി ഓപ്പറേറ്ററുകൾ മൂന്ന് ഓപ്പറാൻഡുകളിൽ പ്രവർത്തിക്കുന്നു. കണ്ടീഷണൽ (conditional) ഓപ്പറേറ്റർ (?:) ഇതിന് ഒരു ഉദാഹരണമാണ്.

മുകളിൽ പറഞ്ഞിരിക്കുന്ന ഓപ്പറേറ്ററുകളിൽ ചിലത് അടുത്ത ഭാഗങ്ങളിലും മറ്റു ചിലത് അധ്യായം ഏഴിലും ചർച്ച ചെയ്യാം.

പ്രവർത്തനരീതി അടിസ്ഥാനമാക്കി ഓപ്പറേറ്ററുകളെ അരിത്മാറ്റിക് (arithmatic), റിലേഷണൽ (relational), ലോജിക്കൽ (logical), ഇൻപുട്ട്/ ഔട്ട്പുട്ട് (input/output), അസൈൻമെന്റ് (assignment), ഷോർട്ട്-ഹാൻഡ് (short-hand), ഇൻക്രിമെന്റ്/ ഡിക്രിമെന്റ് (increment/ decrement) എന്നിങ്ങനെ തരംതിരിച്ചിരിക്കുന്നു.

5.5.1 അരിത്മെറ്റിക് ഓപ്പറേറ്ററുകൾ (Arithmetic operators)

അടിസ്ഥാന ഗണിതപ്രക്രിയകളായ സങ്കലനം, വ്യവകലനം, ഗുണനം, ഹരണം എന്നിവയ്ക്ക് ഉപയോഗിക്കുന്ന ഓപ്പറേറ്ററുകളാണ് അരിത്മെറ്റിക് ഓപ്പറേറ്ററുകൾ. യഥാക്രമം +, ,* ,/എന്നീ ചിഹ്നങ്ങൾ ഇതിനായി ഉപയോഗിക്കുന്നു. ഹരണത്തിനു ശേഷമുള്ള ശിഷ്ടം ലഭിക്കുന്നതിനായി C++ ൽ മോഡുലസ് ഓപ്പറേറ്റർ (%) എന്നൊരു പ്രത്യേക ഓപ്പറേറ്ററും ഉപയോഗിക്കുന്നു. ഇവയെല്ലാം ബൈനറി ഓപ്പറേറ്ററുകളാണ്. + ഉം, - ഉം യുനറി ഓപ്പറേറ്റുകളായും ഉപയോഗിക്കുന്നു എന്നത് ശ്രദ്ധിക്കുക. ഈ പ്രവർത്തനങ്ങൾക്ക് സംഖ്യാ സംബന്ധിയായ ഓപ്പറന്റുകളാണ് ആവശ്യമായിട്ടുള്ളത്. ഈ പ്രവർത്തനത്തിന് ശേഷം ലഭിക്കുന്ന ഫലവും ഒരു സംഖ്യയായിരിക്കും. പട്ടിക 6.3ൽ. ബൈനറി അരിത്മെറ്റിക് പ്രവർത്തനത്തിന്റെ ചില ഉദാഹരണങ്ങൾ കാണിച്ചിരിക്കുന്നു.

വേരിയബിൾ x	വേരിയബിൾ y	സങ്കലനം x + y	വ്യവകലനം x - y	ഗുണനം x * y	ഹരണം x / y
10	5	15	5	50	2
-11	3	-8	-14	-33	-3.66667
11	-3	8	14	-33	-3.66667
-50	-10	-60	-40	500	5

പട്ടിക 5.3 അരിത്മെറ്റിക് ഓപ്പറേറ്ററുകൾ

മോഡുലസ് ഓപ്പറേറ്ററുകൾ (Modulus operator (%)): മോഡ് അഥവാ മോഡുലസ് ഓപ്പറേറ്റർ ഹരണത്തിനുശേഷമുള്ള ശിഷ്ടം കണ്ടുപിടിക്കാൻ ഉപയോഗിക്കുന്നു. ഇത് ഇന്റിജർ ഓപ്പറാൻഡുകളോടൊപ്പം മാത്രമേ ഉപയോഗിക്കാൻ കഴിയു. മോഡുലസ് പ്രക്രിയയുടെ ചില ഉദാഹരണങ്ങൾ പട്ടിക 6.4 ൽ കാണിച്ചിരിക്കുന്നു. ഈ പ്രക്രിയയുടെ ഫലത്തിന്റെ ചിഹ്നം ഒന്നാമത്തെ ഓപ്പറാൻഡിന്റെ ചിഹ്നം തന്നെ ആയിരിക്കുമെന്നത് ശ്രദ്ധിക്കുക. ഇവിടെ പട്ടികയിൽ ഒന്നാമത്തെ ഓപ്പറാന്റ് x ആണ്. ഉദാഹരണം പട്ടിക 5.4ൽ.

വേരിയബിൾ x	വേരിയബിൾ y	മോഡുലസ് ഓപ്പറേഷൻ x % y	വേരിയബിൾ x	വേരിയബിൾ y	മോഡുലസ് ഓപ്പറേഷൻ x % y
10	5	0	100	100	0
5	10	5	32	11	10
-5	11	-5	11	-5	1
5	-11	5	-11	5	-1
-11	-5	-1	-5	-11	-5

പട്ടിക 5.4: മോഡുലസ് ഓപ്പറേറ്റർ ഉപയോഗിച്ചുള്ള പ്രവർത്തനങ്ങൾ

സ്വയം വിലയിരുത്താം

1. അടിസ്ഥാന ഡാറ്റ ഇനങ്ങളെ ആരോഹണ ക്രമത്തിൽ ക്രമീകരിക്കുക.
2. ഒരു സംഭരണ സ്ഥാനത്തിനു നൽകുന്ന പേര് .... എന്ന് അറിയപ്പെടുന്നു.
3. C++ ലെ ഒരു ടെർനറി ഓപ്പറേറ്ററുടെ പേരെഴുതുക.
4. x = -5, y = 3 ആയാൽ താഴെ കൊടുത്തിരിക്കുന്ന ഓപ്പറേഷനുകളുടെ ഔട്ട്പുട്ട് പ്രവചിക്കുക.

   a) -*x*   c) - *x* + -i   e) *x* % - 11   g) *x* % *y*

   b) -*y*   d) - *x* - y   f) *x* + *y*   h) *x* /*y*

   i) *x* ×-*y*   j) - *x* % -5

5.5.2 റിലേഷണൽ ഓപ്പറേറ്ററുകൾ (Relational Operators)

സംഖ്യ സംബന്ധിയായ ഡാറ്റയെ താരതമ്യം ചെയ്യുന്നതിനാണ് റിലേഷനൽ ഓപ്പറേറ്ററുകൾ ഉപയോഗിക്കുന്നത്. ഇവ ബൈനറി ഓപ്പറേറ്ററുകളാണ്. ഏതൊരു റിലേഷണൽ ഓപ്പറേഷന്റെയും ഫലം ശരി (true) അല്ലെങ്കിൽ തെറ്റ് (false) എന്നതായിരിക്കും. C++ൽ True നെ 1 കൊണ്ടും False നെ 0 കൊണ്ടും പ്രതിനിധീകരിക്കുന്നു. <(ചെറുതാണ്), < = (ചെറുതോ, തുല്യമോ ആണ്), > (വലുതാണ്), > = (വലുതോ, തുല്യമോ ആണ്), == (തുല്യമാണ്), ! = (തുല്യമല്ല). എന്നിങ്ങനെ 6 റിലേഷണൽ ഓപ്പറേറ്ററുകളാണ് C++ൽ ഉള്ളത്. തുല്യതാ പരിശോധനയ്ക്ക് രണ്ട് തുല്യ ചിഹ്നങ്ങൾ (=) ആവശ്യാമാണെന്നത് ശ്രദ്ധിക്കുക. വിവിധ റിലേഷണൽ ഓപ്പറേറ്ററുകളുടെ ഉപയോഗവും അവയുടെ ഫലങ്ങളും പട്ടിക 6.5 ൽ കാണിച്ചിരിക്കുന്നു.

m	n	m<n	m>n	m<=n	m>=n	m!=n	m==n
12	5	0	1	0	1	1	0
-7	2	1	0	1	0	1	0
4	4	0	0	1	1	0	1

പട്ടിക 5.5 റിലേഷണൽ ഓപ്പറേറ്ററുകൾ ഉപയോഗിച്ചുള്ള പ്രവർത്തനങ്ങൾ

5.5.3 ലോജിക്കൽ ഓപ്പറേറ്ററുകൾ (Logical Operators)

റിലേഷണൽ ഓപ്പറേറ്ററുകൾ ഉപയോഗിച്ച് നമുക്ക് വിലകൾ താരതമ്യം ചെയ്യാം. ഉദാഹരണത്തിന് 3<5,num !=10 മുതലായവ C++ൽ ഇത്തരം താരതമ്യ പ്രവർത്തനങ്ങളെ റിലേഷണൽ പദപ്രയോഗങ്ങൾ എന്ന് വിളിക്കുന്നു. ചില സാഹചര്യങ്ങളിൽ രണ്ടോ അതിലധികമോ താരതമ്യങ്ങൾ സംയോജിപ്പിക്കേണ്ടതായി വരും. ഗണിതശാസ്ത്രത്തിൽ a>b>c എന്ന രീതിയിലുള്ള പദപ്രയോഗങ്ങൾ നമുക്ക് ഉപയോഗിക്കാം. എന്നാൽ C++ൽ ഇത് സാധ്യമല്ല. ഇവയെ a>b എന്നും b>c എന്നും വേർതിരിച്ച് & എന്ന ലോജിക്കൽ ഓപ്പറേറ്റർ ഉപയോഗിച്ച് സംയോജിപ്പിക്കുന്നു. അതായത് (a>b) && (b>c). ഇത്തരം ലോജിക്കൽ സംയോഗങ്ങളുടെ ഫലവും (true) (1) അല്ലെങ്കിൽ (false) (0) ആയിരിക്കും. && (ലോജിക്കൽ ആൻഡ് (AND),! (ലോജിക്കൽ ഓർ (OR)), ! (ലോജിക്കൽ നോട്ട് (NOT)) എന്നിവയാണ് C++ ലെ ലോജിക്കൽ ഓപ്പറേറ്ററുകൾ.

ലോജിക്കൽ ആന്റ് (logical AND) ഓപ്പറേറ്റർ: E1, E2 എന്നീ രണ്ട് റിലേഷൻ പദപ്രയോഗങ്ങൾ logical AND ഉപയോഗിച്ച് സംയോജിപ്പിക്കുമ്പോൾ ഫലം true(1) ലഭിക്കണമെങ്കിൽ E1, E2 എന്നിവ രണ്ടും true(1) തന്നെ ആയിരിക്കണം. അല്ലാത്ത എല്ലാ സന്ദർഭങ്ങളിലും ഫലം false(0) ആയിരിക്കും. വിവിധ ഇൻപുട്ടുകൾക്ക് അനുസരിച്ചുള്ള ലോജിക്കൽ AND പ്രക്രിയയുടെ ഫലം പട്ടിക 6.6 ൽ കാണിച്ചിരിക്കുന്നു.

E1	E2	E1&&E2
0	0	0
0	1	0
1	0	0
1	1	1

പട്ടിക 5.6ആന്റ് ഓപ്പറേറ്ററിന്റെ ഉപയോഗം

ഉദാഹരണം 10>5&&15<25 ഫലം true (1). 10>5 &&100<25 ഫലം false (0).

ലോജിക്കൽ ഓർ (logical OR) ഓപ്പറേറ്റർ: E1, E2 എന്നീ രണ്ട് റിലേഷൻ പദപ്രയോഗങ്ങൾ logical OR ഉപയോഗിച്ച് സംയോജിപ്പിക്കുമ്പോൾ ഫലം false(0) ലഭിക്കുമെങ്കിൽ E1, E2 എന്നിവ രണ്ടും false (0) ആയിരിക്കണം. അല്ലാത്ത എല്ലാ സന്ദർഭങ്ങളിലും ഫലം true(1) ആയിരിക്കും. വിവിധ ഇൻപുട്ടുകൾക്ക് അനുസരിച്ചുള്ള ലോജിക്കൽ OR പ്രക്രിയയുടെ ഫലം പട്ടിക 6.7ൽ കാണിച്ചിരിയ്ക്കുന്നു.

<table>
<tr><th>E1</th><th>E2</th><th>E1||E2</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>

പട്ടിക 5.7 ഓർ ഓപ്പറേറ്ററിന്റെ ഉപയോഗം

ഉദാഹരണം: 10>15||100<25 ഫലം true(1), 10>15||100<90 ഫലം false (0).

ലോജിക്കൽ നോട്ട് (logical NOT) ഓപ്പറേറ്റർ: റിലേഷനൽ പദപ്രയോഗങ്ങളുടെ ഫലം വിപരീതമാക്കാനാണ് logical NOT ഉപയോഗിക്കുന്നത്. ഇത് ഒരു യൂനറി ഓപ്പറേഷനാണ്.

വിവിധ ഇൻപുട്ടുകൾക്ക് അനുസരിച്ചുള്ള ലോജിക്കൽ NOT പദപ്രയോഗത്തിന്റെ ഫലം പട്ടിക 6.8ൽ കാണിച്ചിരിക്കുന്നു.

E1	!E1
0	1
1	0

പട്ടിക 5.8 നോട്ട് ഓപ്പറേറ്ററിന്റെ ഉപയോഗം

ഉദാഹരണം ! (100<2) ഫലം 1

! (100>2) ഫലം 0 (False)

5.5.4 ഇൻപുട്ട് / ഔട്ട്പുട്ട് ഓപ്പറേറ്ററുകൾ (Input/Output Operators)

ഇൻപുട്ട് പ്രവർത്തനങ്ങൾക്ക് സാധാരണയായി ഉപയോക്താവിന്റെ ഇടപെടൽ ആവശ്യമാണ്. ഇൻപുട്ട് പ്രോസസ്സിൽ കീബോഡ് വഴി നൽകുന്ന ഡാറ്റ മെമ്മറി ലൊക്കേഷനുകളിൽ സൂക്ഷിക്കുന്നു. C++ൽ ഇൻപുട്ട് ഓപ്പറേഷൻ ചെയ്യുന്നതിനായി >> ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നു. ഈ ഓപ്പറേറ്റർ ഗേറ്റ് ഫ്രം (get from) അഥവാ എക്സ്ട്രാക്ഷൻ (extraction) ഓപ്പറേറ്റർ എന്ന് അറിയപ്പെടുന്നു. രണ്ട് > ചിഹ്നങ്ങൾ ഉപയോഗിച്ചാണ് ഈ ചിഹ്നം നിർമ്മിച്ചിരിക്കുന്നത്.

ഇതുപോലെ ഔട്ട്പുട്ട് പ്രവർത്തനങ്ങളിൽ ഡാറ്റ റാമിൽ നിന്നും ഔട്ട്പുട്ട് ഉപകരണത്തിലേക്ക് മാറ്റുന്നു. സാധാരണയായി ഫലം നേരിട്ട് ലഭിക്കാൻ ഉപയോഗിക്കുന്ന ഔട്ട്പുട്ട് ഉപകരണം മോണിറ്ററാണ്. പുട്ട് ടു (Put to) അഥവാ ഇൻസേർഷൻ (insertion) ഓപ്പറേറ്റർ എന്നു വിളിക്കുന്ന <<ഓപ്പറേറ്റർ ഔട്ട്പുട്ട് പ്രവർത്തനത്തിന് ഉപയോഗിക്കുന്നു. ഇത് രണ്ട് <ചിഹ്നങ്ങൾ ഉപയോഗിച്ചാണ് നിർമ്മിച്ചിരിക്കുന്നത്.

5.5.5 അസൈസ്മെന്റ് ഓപ്പറേറ്റർ (=) (Assignment operator (=))

സാധാരണയായി ഒരു വില മെമ്മറിയിൽ സംഭരിക്കുന്നതിനായി വിലനൽകൽ ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നു. ഇത് ഒരു ബൈനറി ഓപ്പറേറ്ററായതിനാൽ ഇവയ്ക്ക് രണ്ട് ഓപ്പറാന്റുകൾ ആവശ്യമാണ്. ആദ്യത്തെ ഓപ്പറാന്റ് ഒരു വേരിയബിൾ ആയിരിക്കണം. അതിലാണ് രണ്ടാമത്തെ ഓപ്പറാന്റിന്റെ മൂല്യം സൂക്ഷിക്കുന്നത്.

ഇനം	വിശദീകരണം
a=b	വേരിയബിൾ **b** യുടെ വില **a** ൽ സംഭരിക്കുന്നു
a=3	സ്ഥിരാംഗം **3** വേരിയബിൾ **a** ൽ സംഭരിക്കുന്നു

പട്ടിക 5.9 അസൈൻമെന്റ് ഓപ്പറേറ്റർ

ചില ഉദാഹരണങ്ങൾ പട്ടിക 5.9ൽ കാണിച്ചിരിക്കുന്നു.

റിലേഷണൽ ഓപ്പറേറ്ററായ = = ഉപയോഗത്തെപ്പറ്റി ഭാഗം 6.6.2ൽ നമ്മൾ ചർച്ച ചെയ്തിരുന്നു. ഈ രണ്ടു ഓപ്പറേറ്ററുകൾ തമ്മിലുള്ള വ്യത്യാസം ശ്രദ്ധിക്കുക. = ചിഹ്നം ഒരു വേരിയബിളിനു വില നൽകുന്നതിനും എന്നാൽ = = ചിഹ്നം രണ്ട് വിലകളെ തമ്മിൽ താരതമ്യം ചെയ്ത് true അല്ലെങ്കിൽ false എന്ന ഉത്തരം നൽകുന്നതിനും ഉപയോഗിക്കുന്നു.

5.6. പ്രയോഗങ്ങൾ (expressions)

ഒരു പദപ്രയോഗം ഓപ്പറേറ്ററുകളും ഓപ്പറാന്റുകളും ചേർന്നതാണ്. ഓപ്പറാന്റുകൾ സ്ഥിരാംഗങ്ങളോ വേരിയബിളുകളോ ആകാം. എല്ലാ പദപ്രയോഗങ്ങളും പൂർത്തീകരിച്ചതിനുശേഷമേ ആ പ്രയോഗത്തിന്റെ അന്തിമ ഫലം ലഭ്യമാകൂ. ഈ ഫലം പദപ്രയോഗം തിരികെ നൽകിയ വില എന്ന് അറിയപ്പെടുന്നു. ഉപയോഗിച്ചിരിക്കുന്ന ഓപ്പറേറ്ററുകളുടെ അടിസ്ഥാനത്തിൽ പദപ്രയോഗങ്ങളെ പ്രധാനമായും അരിത്മാറ്റിക് പദപ്രയോഗങ്ങൾ, റിലേഷണൽ പദപ്രയോഗങ്ങൾ, ലോജിക്കൽ പദപ്രയോഗങ്ങൾ എന്നിങ്ങനെ തരംതിരിച്ചിരിക്കുന്നു.

5.6.1 അരിത്മാറ്റിക് പ്രയോഗങ്ങൾ (arithmetic expressions)

അരിത്മാറ്റിക് ഓപ്പറേറ്ററുകൾ മാത്രം ഉപയോഗിച്ചിട്ടുള്ള പദപ്രയോഗങ്ങളെ അരിത്മാറ്റിക് പദപ്രയോഗങ്ങൾ എന്ന് വിളിക്കുന്നു. ഇവിടെ ഓപ്പറാന്റുകൾ സംഖ്യകളാണ്. അവ വേരിയബിളുകളോ സ്ഥിരാംഗങ്ങളോ ആകാം. ഈ പദപ്രയോഗത്തിൽ നിന്നും ലഭ്യമാകുന്ന വിലയും ഒരു സംഖ്യ ആയിരിക്കും. അരിത്മാറ്റിക് പദപ്രയോഗങ്ങളെ വീണ്ടും പൂർണ്ണ സംഖ്യാപദപ്രയോഗങ്ങൾ, ദശാംശസംഖ്യാ (real) പദപ്രയോഗങ്ങൾ, സ്ഥിരാംഗ പദപ്രയോഗങ്ങൾ എന്നിങ്ങനെ തരം തിരിച്ചിരിക്കുന്നു.

പൂർണ്ണസംഖ്യാ പ്രയോഗങ്ങൾ: ഒരു അരിത്മെറ്റിക് പദപ്രയോഗത്തിൽ പൂർണ്ണസംഖ്യകൾ മാത്രമേ ഉൾകൊള്ളുന്നുള്ളൂ എങ്കിൽ അതിനെ പൂർണ്ണസംഖ്യാപദപ്രയോഗം എന്ന് വിളിക്കുന്നു. ഇവയുടെ ഫലവും ഒരു പൂർണ്ണസംഖ്യ ആയിരിക്കും.

ഉദാഹരണത്തിന്: x, y എന്നിവ പൂർണ്ണസംഖ്യാ വേരിയബിളുകൾ ആണെങ്കിൽ ചില പൂർണ്ണ സംഖ്യാ പദപ്രയോഗവും അവയുടെ ഫലങ്ങളും പട്ടിക 5.10 ൽ കൊടുത്തിരിക്കുന്നു. എല്ലാ പദപ്രയോഗങ്ങളുടെയും ഫലം ഒരു പൂർണ്ണസംഖ്യ ആയിരിക്കും എന്നത് ശ്രദ്ധിക്കുക.

x	y	x + y	x / y	-x + x * y	5 + x / y	x % y
5	2	7	2	5	7	1
6	3	9	2	12	7	0

പട്ടിക 5.10 പൂർണ്ണ സംഖ്യാ പ്രയോഗങ്ങളും അവയുടെ ഫലങ്ങളും

ഫ്ളോട്ടിംഗ് പോയിന്റ് പ്രയോഗങ്ങൾ (floating point/ real expression): ഒരു അരിത്മാറ്റിക് പദപ്രയോഗത്തിൽ എല്ലാ വിലകളും ദശാംശസഖ്യകൾ ആണെങ്കിൽ അവയെ ദശാംശസഖ്യ അഥവാ ഭിന്ന സംഖ്യാപദപ്രയോഗം എന്ന് വിളിക്കുന്നു. ഇതിന്റെ ഫലം തീർച്ചയായും ഒരു ദശാംശസംഖ്യ ആയിരിക്കും. x, y എന്നിവ ദശാംശസംഖ്യാ വേരിയബിൾ ആണെന്ന് കരുതുക. ചില ദശാംശസംഖ്യാപദപ്രയോഗങ്ങളും അവയുടെ ഫലങ്ങളും പട്ടിക 5.11 ൽ കൊടുത്തിരിക്കുന്നു.

x	y	x + y	x / y	-x + x * y	5 + x / y	x * x / y
5.0	2.0	7.0	2.5	5.0	7.5	12.5
6.0	3.0	9.0	2.0	12.0	7.0	12.0

പട്ടിക 5.11: ഫ്ളോട്ടിംഗ് പോയിന്റ് സംഖ്യാ പ്രയോഗങ്ങളും അവയുടെ ഫലങ്ങളും

മുകളിൽ കൊടുത്തിരിക്കുന്ന എല്ലാ പദപ്രയോഗങ്ങളുടെയും ഉത്തരം ദശാംശസംഖ്യകളാണ് എന്ന് കാണാൻ കഴിയും.

ഒരു അരിത്മാറ്റിക് പദപ്രയോഗത്തിൽ ഉപയോഗിക്കുന്ന എല്ലാ ഓപ്പറാന്റുകളും സ്ഥിരാംഗങ്ങളാണെങ്കിൽ അതിനെ സ്ഥിരാംഗപദപ്രയോഗം (const. expression) എന്നു വിളിക്കുന്നു. ഉദാ: 20+5/2.0. സ്ഥിരാംഗങ്ങളായ 15,3.14, 'a' എന്നിവയും സ്ഥിരാംഗപദപ്രയോഗങ്ങളായി അറിയപ്പെടുന്നു.

5.6.2 റിലേഷണൽ പ്രയോഗങ്ങൾ (relational expressions)

റിലേഷണൽ ഓപ്പറേറ്ററുകൾ ഉപയോഗിക്കുന്ന പദപ്രയോഗങ്ങളെ റിലേഷണൽ പദപ്രയോഗങ്ങൾ എന്ന് വിളിക്കുന്നു. ഇവ true(1) അല്ലെങ്കിൽ false(0) എന്ന ഫലം നൽകുന്നു. ഇത്തരം പദപ്രയോഗങ്ങളിൽ ഓപ്പറാന്ററുകളായി സംഖ്യകളാണ് ഉപയോഗിക്കുക. ഇവയുടെ ചില ഉദാഹരണം പട്ടിക 5.15 ൽ കാണിച്ചിരിക്കുന്നു.

x	y	x > y	x == y	x+y !=y	x-2 == y+1	x*y == 6*y
5.0	2.0	1 (True)	0 (False)	1 (True)	1 (True)	0 (False)
6	13	0 (False)	0 (False)	1 (True)	0 (False)	1 (True)

പട്ടിക 5.12 റിലേഷണൽ പ്രയോഗങ്ങളും അവയുടെ ഫലങ്ങളും

അരിത്മാറ്റിക് ഓപ്പറേറ്ററുകൾക്ക് റിലേഷണൽ ഓപ്പറേറ്റേക്കാൾ മുൻഗണയുണ്ടെന്ന്നമുക്കറിയാം. ഒരു റിലേഷണൽ ഓപ്പറേറ്ററിന്റെ ഇരുവശങ്ങളിലായി അരിത്മാറ്റിക് പദപ്രയോഗങ്ങൾ ഉപയോഗിക്കുമ്പോൾ ആദ്യം അരിത്മാറ്റിക് ഓപ്പറേഷനുകൾ ചെയ്യുകയും അതിന് ശേഷം ആ ഫലങ്ങൾ താരതമ്യം ചെയ്യുന്നു. പട്ടികയിലെ ചില പദപ്രയോഗങ്ങളിൽ

അരിത്മാറ്റിക് ഓപ്പറേറ്ററും റിലേഷണൽ ഓപ്പറേറ്ററുകളും ഉൾപ്പെട്ടിരിക്കുന്നു. വിവിധ തരം ഓപ്പറേറ്ററുകൾ ഉൾപ്പെട്ടിട്ടുണ്ടെങ്കിലും ഇവയുടെ ഫലം true(1) അല്ലെങ്കിൽ false (0) ആയതിനാൽ അവയെ റിലേഷണൽ പദപ്രയോഗങ്ങൾ എന്നാണ് വിളിക്കുന്നത്.

5.6.3 ലോജിക്കൽ പ്രയോഗങ്ങൾ (logical expressions)

രണ്ടോ അതിലധികമോ റിലേഷണൽ പദപ്രയോഗങ്ങളെ ലോജിക്കൽ ഓപ്പറേറ്റർ ഉപയോഗിച്ച് ലോജിക്കൽ പദപ്രയോഗങ്ങൾ സംയോജിപ്പിക്കുന്നു. ഇവയുടെ ഫലം true(1) അല്ലെങ്കിൽ false (0) എന്നായിരിക്കും. ലോജിക്കൽ പദപ്രയോഗത്തിൽ വേരിയബിളുകൾ, സ്ഥിരാംഗങ്ങൾ ലോജിക്കൽ ഓപ്പറേറ്ററുകൾ, റിലേഷനൽ ഓപ്പറേറ്ററുകൾ എന്നിവ ഉൾപ്പെടാവുന്നതാണ്. ചില ഉദാഹരണങ്ങൾ പട്ടിക 5.13 ൽ കാണിച്ചിരിക്കുന്നു.

x	y	x>=y && x==20	x==5lly==0	x==y && y+2==0	!(x==y)
5.0	2.0	0 (False)	1 (True)	0 (False)	1 (True)
20	13	1 (True)	0 (False)	0 (False)	1 (True)

പട്ടിക 5.13 ലോജിക്കൽ പ്രയോഗങ്ങളും അവയുടെ ഫലങ്ങളും

പട്ടിക 5.13 ൽ കാണുന്നതു പോലെ ചില പദപ്രയോഗങ്ങളിൽ ലോജിക്കൽ ഓപ്പറേറ്ററുകളെ കൂടാതെ അരിതമാറ്റിക് ഓപ്പറേറ്ററുകളും റിലേഷൺ ഓപ്പറേറ്ററുകളും ഉൾപ്പെട്ടിട്ടുണ്ടെങ്കിലും ഈ പ്രയോഗങ്ങളെ ലോജിക്കൽ പ്രയോഗങ്ങളായി കണക്കാക്കുന്നു. അവസാനം ചെയ്യുന്ന പ്രവർത്തനം ലോജിക്കൽ പ്രവർത്തനം ആയതിനാലും അതിന്റെ ഫലം True അല്ലെങ്കിൽ False ആയത് കൊണ്ടുമാണ് ഇത്.

സ്വയം പരിശോധിക്കാം.

1. x = 5, y=3 ആയാൽ താഴെ കൊടുത്തിരിക്കുന്ന പ്രവർത്തനങ്ങളുടെ ഔട്ട്പുട്ട് പ്രവചിക്കുക

   a) x>= 10<<y>=4,   b) x>=1<<y>=3,   c) x>=11y>=4,   d)x>=1 11y>=3

2. p=5, q=3, x=2 ആയാൽ ചുവടെ ചേർക്കുന്ന പ്രയോഗങ്ങളുടെ ഔട്ട്പുട്ട് പ്രവചിക്കുക

   a) ++P-q×r/2   b) p×q--+r   c) p-q-r×2+p   d) p+=5×q+r×r/2

5.7. പ്രസ്താവനകൾ (Statements)

ഒരു ഭാഷയുടെ പഠനശ്രേണി എന്നത് അക്ഷരമാല, പദങ്ങൾ, ശൈലികൾ, വാക്യങ്ങൾ, ഖണ്ഡികകൾ തുടങ്ങിയവയാണ്. അതുപോലെ C++ന്റെ പഠനത്തിൽ അക്ഷരമാല (character set), ടോക്കണുകൾ (tokens), പദപ്രയോഗങ്ങൾ എന്നിവ നമ്മൾ മനസ്സിലാക്കി കഴിഞ്ഞു. പ്രസ്താവനകളുടെ സഹായത്തോടെ കമ്പ്യൂട്ടറുമായി യുക്തിപരമായും അർത്ഥവത്തായും സംവദിക്കാവുന്ന രീതിയിൽ നാമിപ്പോൾ എത്തിയിട്ടുണ്ട്. ഒരു പ്രോഗ്രാമിംഗ് ഭാഷയിലെ ഏറ്റവും ചെറിയ പ്രവർത്തന ഘടകമാണ് പ്രസ്താവനകൾ. ഒരു പ്രസ്താവന അവസാനിച്ചു എന്ന് സൂചിപ്പിക്കുവാൻ C++; (Semi column) ഉപയോഗിക്കുന്നു. C++ ൽ വ്യത്യസ്ത ആവശ്യങ്ങൾക്കായി പ്രഖ്യാപന പ്രസ്താവനകൾ (declaration), വില നൽകുന്ന (assignment) പ്രസ്താവനകൾ, ഇൻപുട്ട് (input) പ്രസ്താവനകൾ, നിയന്ത്രണ

പ്രസ്താവനകൾ (control), ഔട്ട്പുട്ട് (output) പ്രസ്താവനകൾ തുടങ്ങിയവ ഉപയോഗിക്കുന്നു. ഒരു C++പ്രോഗ്രാമിലെ ഓരോ പ്രസ്താവനക്കും അതിന്റേതായ ലക്ഷ്യങ്ങളുണ്ട്. ഇവയിൽ പ്രഖ്യാപന പ്രസ്താവനകൾ ഒഴികെയുള്ളവ ചില പ്രത്യേക പ്രവർത്തനങ്ങൾ കമ്പ്യൂട്ടർ ഉപയോഗിച്ച് ചെയ്യാനുള്ളവയാണ്. നിർവഹണ പ്രസ്താവനകൾ (execulable statements) കമ്പ്യൂട്ടറനുള്ള നിർദ്ദേശങ്ങളാണ്. നിയന്ത്രണപ്രസ്താവനകളുടെ പ്രവർത്തനം അധ്യായം 7ൽ ചർച്ച ചെയ്യാം.

മറ്റു ചില പ്രസ്താവനകളെ നമുക്കിവിടെ ചർച്ചചെയ്യാം.

5.7.1. പ്രഖ്യാപന പ്രസ്താവനകൾ (Declaration statement)

എല്ലാ ഉപയോക്തൃ നിർവചിത വാക്കുകളും പ്രോഗ്രാമിൽ അവ ഉപയോഗിക്കുന്നതിനു മുൻപുതന്നെ നിർവചിക്കേണ്ടതാണ്. ഒരു വേരിയബിൾ എന്നത് ഉപയോക്താവ് നിർവചിക്കുന്നതാണെും മെമ്മറിയിലെ ഒരിടത്തിനെ സൂചിപ്പിക്കുന്നതാണെന്നും നാം കണ്ടു. ഉപയോഗിക്കുന്നതിന് മുൻപ് പ്രോഗ്രാമിൽ ഇവ പ്രഖ്യാപിക്കപ്പെടേണ്ടതുണ്ട്. നാം ഒരു വേരിയബിളിനെ പ്രഖ്യാപിക്കുമ്പോൾ അതിൽ സൂക്ഷിച്ചിരിക്കുന്ന ഡാറ്റയുടെ ഇനം ഏതാണെന്ന് കംപൈലറിനെ അറിയിക്കുകയാണ് ചെയ്യുന്നത്. വേരിയബിൾ പ്രഖ്യാപിക്കുന്നതിന്റെ വാക്യഘടന:

Data Type<variable>, [<variable 2>, < variable 3>...];

Data Type എന്നത് C++ലെ ഏതെങ്കിലും അംഗീകൃതമായ ഡാറ്റ ഇനം ആകാം. ഒന്നിലധികം വേരിയബിളുകൾ പ്രയോഗിക്കുമ്പോൾ അവയെ വേർതിരിക്കാൻ കോമ (,) ഉപയോഗിക്കണം. ഒരു പ്രഖ്യാപന പ്രസ്താവന അർദ്ധവിരാമം (;) ത്തോട് കൂടി അവസാനിക്കുന്നു. സാധാരണയായി വേരിയബിളുകൾ പ്രഖ്യാപിക്കുന്നത് അവ ഉപയോഗിക്കുന്നതിന് തൊട്ട് മുൻപോ അല്ലെങ്കിൽ പ്രോഗ്രാമിന്റെ തുടക്കത്തിലോ ആയിരിക്കും. വാക്യഘടനയിൽ [ ] ൽ നൽകിയിരിക്കുന്നത് ആവശ്യമുണ്ടെങ്കിൽ മാത്രം ഉപയോഗിച്ചാൽ മതി എന്ന അർത്ഥത്തിലാണ്. താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവനകൾ വേരിയബിൾ പ്രഖ്യാപനങ്ങൾക്ക് ഉദാഹരണങ്ങളാണ്:

```
int roll number;
double wgpa, avg-score;
```

ഒന്നാമത്തെ പ്രസ്താവനയിൽ വേരിയബിൾ roll number ഒരു int ഡാറ്റ ഇനമായതിനാൽ ഇതിനായി 4 ബൈറ്റ് മെമ്മറി മാറ്റിവക്കപ്പെടുന്നു. (gcc അനുസരിച്ച്) ഇതിൽ 2147483648 മുതൽ +2147483647 വരെയുള്ള ഏതെങ്കിലും പൂർണ്ണസംഖ്യ സൂക്ഷിക്കാവുന്നതാണ്.

രണ്ടാമത്തെ പ്രസ്താവന wgpa, avg-score എന്നീ double ഡാറ്റ ഇനത്തിലുള്ള വേരിയബിളുകൾ നിർവചിക്കുന്നു. ഇവ ഓരോന്നിനും 8 ബൈറ്റ് മെമ്മറി വീതം നീക്കി വയ്ക്കുന്നു. പ്രോഗ്രാം കംപൈൽ ചെയ്യുന്ന സമയത്ത് ഇവക്കുള്ള മെമ്മറി നീക്കി വയ്ക്കുന്നു.

5.7.2 അസൈൻമെന്റ് പ്രസ്താവനകൾ (Assignment statement)

ഒരു വേരിയബിളിലേക്ക് വില നൽകുന്നതിനാണ് അസൈൻമെന്റ് ഓപ്പറേറ്റർ (=) ഉപയോഗിക്കുന്നത്. ഇങ്ങനെ ഉപയോഗിക്കുന്നതിനുള്ള പ്രസ്താവനകളെ അസൈൻമെന്റ് പ്രസ്താവന എന്ന് വിളിക്കുന്നു. താഴെപറയുന്ന ഏതെങ്കിലും രീതികളിൽ അവ എഴുതാം.

```
variable = constant;
variable1 = variable2;
variable = expression;
```

ഒന്നാമത്തേതിൽ ഒരു സ്ഥിരാംഗം വേരിയബിളിൽ സംഭരിക്കുന്നു. രണ്ടാമത്തേതിൽ വേരിയബിളിന്റെ വില മറ്റൊരു വേരിയബിളിൽ സംഭരിക്കുന്നു. മൂന്നാമത്തേതിൽ പദപ്രയോഗത്തിന്റെ ഫലം വേരിയബിളിൽ സംഭരിക്കുന്നു. അതുപോലെ നാലാമത്തേതിൽ ഫങ്ഷൻ തിരിച്ചുനൽകുന്ന വിലയാണ് വേരിയബിളിലേക്ക് സംഭരിക്കുന്നത്. ഫങ്ഷൻ എന്ന ആശയത്തെക്കുറിച്ച് അധ്യായം 10ൽ ചർച്ച ചെയ്യാം.

അസൈൻമെന്റ് പ്രസ്താവനകൾക്കുള്ള ചില ഉദാഹരണങ്ങൾ താഴെ കൊടുത്തിരിക്കുന്നു.

```
A = 15;    b = 5.8;
c = a+b;  d = (a+b) & (c+d)
r = sqrt (25);
```

അവസാനം നൽകിയിരിക്കുന്ന ഉദാഹരണത്തിൽ sqrt ( ) എന്നത് ഒരു ഫങ്ഷനാണ്. r എന്ന വേരിയബിളിൽ 25 ന്റെ വർഗമൂലമായ 5 ആണ് സംഭരിക്കപ്പെടുക. അസൈൻമെന്റ് പ്രസ്താവനകളിൽ ഇടതുവശത്ത് ഒരു വേരിയബിൾ തന്നെ ആയിരിക്കണം. പ്രോഗ്രാം പ്രവർത്തിക്കുമ്പോൾ ആദ്യം വലതുവശം പ്രവർത്തിച്ചശേഷം കിട്ടുന്ന ഫലം ഇടതുവശത്തെ വേരിയബിളിൽ (RHS) സംഭരിക്കുന്നു.

താഴെകാണിച്ചിരിക്കുന്നതുപോലെ ഒന്നിൽ കൂടുതൽ അസൈൻമെന്റുകൾ കൂട്ടിച്ചേർത്ത് ഒരേ സമയം ചെയ്യാവുന്നതാണ്. ഉദാഹരണത്തിന് x = y = z =13; ഇവിടെ 13 എന്ന വില z,y,x എന്നീ ക്രമത്തിൽ മൂന്ന് വേരിയബിളുകൾക്കും നൽകുന്നു. അസൈൻമെന്റ് പ്രസ്താവനയ്ക്കുമുമ്പ് വേരിയബിളുകൾ പ്രഖ്യാപിച്ചിരിക്കണം. ഒരു വേരിയബിളിന് നാം വില നൽകുകയാണെങ്കിൽ അതിലുള്ള പഴയ വില മാറ്റി പുതിയ വില നൽകുന്നു.

5.7.3 ഇൻപുട്ട് പ്രസ്താവന (Input statement):-

പ്രോഗ്രാമിന്റെ പ്രവർത്തനസമയത്ത് ഉപയോക്താവിന് ഡാറ്റ മെമ്മറിയിൽ സംഭരിക്കുന്നതിന് ഉപയോഗിക്കുന്ന പ്രസ്താവനകൾ ഇൻപുട്ട് പ്രസ്താവനകൾ എന്നറിയപ്പെടുന്നു. ഗെറ്റ് ഫ്രം, എക്സ്ട്രാക്ഷൻ എന്നീ പേരുകളിലറിയപ്പെടുന്ന >> ഓപ്പറേറ്ററാണ് ഇൻപുട്ട് ഓപ്പറേറ്റർ എന്നും നാം കണ്ടതാണ്. ഡാറ്റ സൂക്ഷിക്കേണ്ട RAMലെ സ്ഥാനവും ഇൻപുട്ട് നൽകുന്ന ഉപകരണവുമാണ് ഇൻപുട്ട് ഓപ്പറേറ്ററുടെ രണ്ട് ഓപെറന്റുകൾ. ഒരു അംഗീകൃത ഇൻപുട്ട് ഉപകരണമായ കീബോർഡിൽ നിന്ന് വരുന്ന തുടർച്ചയായ ഡാറ്റാ പ്രവാഹത്തെ വേരിയബിളുകൾ സൂചിപ്പിക്കുന്ന മെമ്മറി സ്ഥാനങ്ങളിൽ സംഭരിക്കപ്പെടുന്നു.. c++ ഒരു ഒബ്ജക്റ്റ് ഓറിയന്റഡ് ഭാഷയായതിനാൽ കീബോർഡ് ഒരു അംഗീകൃത ഇൻപുട്ട് സ്ട്രീം ഉപകരണമായാണ് കണക്കാക്കപ്പെടുന്നത്. സി ഇൻ (cin) എന്ന പേരിലുള്ള ഒരു ഒബ്ജക്റ്റ് ആയി തിരിച്ചറിയപ്പെടുകയും ചെയ്യുന്നു. ഒരു ഇൻപുട്ട് പ്രസ്താവനയുടെ ലളിതമായ രൂപം താഴെ കൊടുത്തിരിക്കുന്നു.

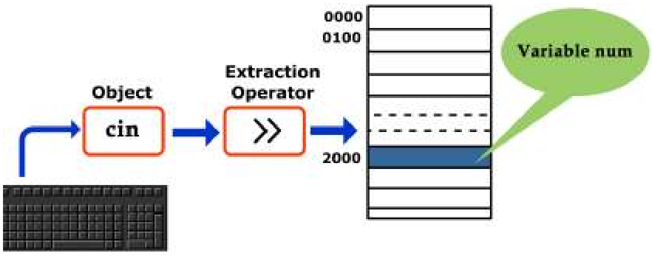
```
streamobject >> variable;
```

കീ ബോർഡ് ഒരു ഇൻപുട്ട് ഉപകരണമായി നാം ഉപയോഗിക്കുന്നതിനാൽ മുകളിൽ പറഞ്ഞ വാക്യഘടനയിൽ stream object നു പകരം cin എന്നു എഴുതുന്നു. >> എന്ന ഓപ്പ

റേറ്ററിനു നിർബന്ധമായും ഒരു വേരിയബിൾ തന്നെയാകണം ഓപ്പറന്റ് ആയി ഉപയോഗിക്കേണ്ടത്.ഉദാഹരണത്തിന് താഴെ പറയുന്ന പ്രസ്ഥാവന കീബോർഡിൽ നിന്ന് ഡാറ്റാ സ്വീകരിക്കുകയും Num എന്ന വേരിയബിളിൽ സൂക്ഷിക്കുകയും ചെയ്യുന്നു.

cin>>num;

ഡാറ്റ കീ ബോർഡിൽ നിന്നും സ്വീകരിച്ച് എങ്ങനെ വേരിയബിളിൽ സംഭരിക്കുന്നു എന്ന് ചിത്രം 5.3. ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 5.3 C++

5.7.4 ഔട്ട്പുട്ട് പ്രസ്താവന (Output statement):-

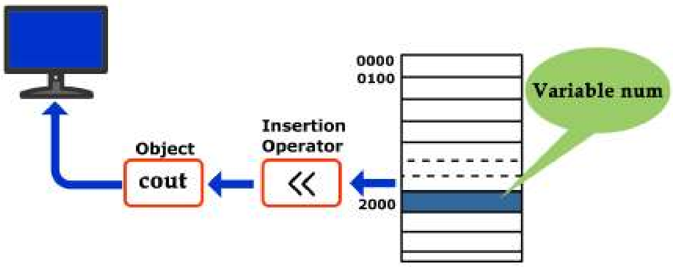
ഏതൊരു ഔട്ട്പുട്ട് ഉപകരണത്തിലൂടെയും ഉപയോക്താക്കൾക്ക് ഫലം ലഭ്യമാക്കുന്നതാണ് ഔട്ട്പുട്ട് പ്രസ്താവന. പുട്ട് ടു അല്ലെങ്കിൽ ഇൻസേർഷൻ എന്നീ പേരുകളിൽ അറിയപ്പെടുന്ന ഓപ്പറേറ്ററാണ് ഈ പ്രവർത്തനത്തിന് ഉപയോഗിക്കുന്നത്. ഇവിടെ ഔട്ട്പുട്ട് ചെയ്യേണ്ട ഡാറ്റയും ഔട്ട്പുട്ട് ഉപകരണവുമാണ് രണ്ട് ഓപെറന്റുകൾ. ഔട്ട്പുട്ട് പ്രസ്താവനയുടെ വാക്യഘടന ഇതാണ്.

streamobject << data;

stream object ഏതെങ്കിലും ഔട്ട്പുട്ട് ഉപകരണമാകാം. Data ഒരു സ്ഥിരാംഗമോ ഒരു വേരിയബിളോ അല്ലെങ്കിൽ ഒരു പദപ്രയോഗമോ ആകാം. മോണിറ്റർ ആണ് സാധാരണ ഉപയോഗിക്കുന്ന ഔട്ട്പുട്ട് ഉപകരണം. C++ ൽ cout (സി ഔട്ട് എന്ന് ഉച്ചരിക്കുന്നു) എന്നതാണ് മോണിറ്ററിനെ സൂചിപ്പിക്കുന്ന ഒബ്ജക്റ്റിന്റെ പേര്. മോണിറ്റർ ഔട്ട്പുട്ട് ഉപകരണമായി ഉപയോഗിക്കുന്ന ചില ഔട്ട്പുട്ട് പ്രസ്താവനകൾക്കുള്ള ഉദാഹരണങ്ങളാണ് താഴെ പറയുന്നവ.

```
cout << num;
cout << "hello friends";
cout << num+12;
```

ഒന്നാമത്ത പ്രസ്താവന num ന്റെ വില മോണിറ്ററിൽ പ്രദർശിപ്പിക്കുന്നു. രണ്ടാമത്തേത് hello friends എന്ന സ്ട്രിംഗ് പ്രദർശിപ്പിക്കുന്നു. അവസാനത്തേതിൽ num നോടുകൂടി 12 കൂട്ടി കിട്ടുന്ന ഫലം പ്രദർശിപ്പിക്കുന്നു. (num ൽ സംഖ്യയാണെന്ന് കരുതുക). മെമ്മറി സ്ഥാനം num ൽ നിന്ന് ഡാറ്റ എങ്ങനെയാണ് stream object (മോണിറ്റർ)ൽ ചേർത്തിരിക്കുന്നത് എന്ന് ചിത്രം 5.4. ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 5.4: C++

> ടോക്കണുകളായ cin ഉം cout ഉം കിവേഡുകളല്ല. C++ ഭാഷയുടെ ഭാഗമല്ലാത്ത മുൻ നിർവചിത വാക്കുകളാണിവ. ഉപയോക്താവിന് ഇവയെ പുനർ വ്യാഖ്യാനം ചെയ്യാനവുന്നതാണ്. C++ ഭാഷയുടെ ലൈബ്രറിയിൽ നിർവ്വചിച്ചിട്ടുള്ള അടിസ്ഥാനവാക്കണിത്. വ്യക്തമായി പറയുകയാണെങ്കിൽ മുൻകഗൂട്ടി വായിച്ച അർത്ഥമുള്ള വാക്കുകളെ പുർവ്യാഖ്യാനം ചെയ്ത് മാറ്റാവശ്യങ്ങൾക്ക് ഉപയോഗിക്കുന്നത് അപകടകരവും ചിന്താകുഴപ്പം വരുത്തുന്നതുമാണ്. ഇത് ഒഴിവാക്കേണ്ടതാണ്. ഏറ്റവും ലളിതവും സുരക്ഷിതവുമായ മാർഗ്ഗം എന്നത് എല്ലാ മുൻ നിർവചിത ഐഡന്റിഫയറുകളെയും കൈകാര്യം ചെയ്യുന്നതാണ്.

കാസ്കേഡിംങ്ങ് ഓഫ് ഇൻപുട്ട് / ഔട്ട്പുട്ട് ഓപ്പറേറ്ററുകളുടെ (Cascading of I/O operators)

ഒന്നിൽകൂടുതൽ ഇൻപുട്ട് ഔട്ട്പുട്ട് ഓപ്പറേറ്റുകൾ ഒരുമിച്ച് ഉപയോഗിക്കുന്ന വിധം.

നമുക്ക് x,y,z എന്നീ മൂന്നു പേരുകളിലായി ഡാറ്റ ഇൻപുട്ട് ചെയ്യുന്നതിനായി

```
cin>>x;
cin>>y;
cin>>z;
```

ഇങ്ങനെ 3 പ്രസ്താവനകൾ ഉപയോഗിക്കാം; താഴെ പറയുന്ന രീതിയിൽ ഇവ മൂന്നും കൂട്ടി യോജിപ്പിച്ച് ഒറ്റ പ്രസ്താവനയായി ഉപയോഗിക്കാവുന്നതാണ്.

```
cin>>x>>y>>z;
```

ഒന്നിൽകൂടുതൽ ഇൻപുട്ട് ഔട്ട്പുട്ട് ഓപ്പറേറ്ററുകൾ ഒറ്റ പ്രസ്താവനയിൽ ഉപയോഗിക്കുന്നതിന് സംയോജിപ്പിച്ച് ഉണ്ടാക്കുമ്പോൾ ഓഫ് ഇൻപുട്ട് ഔട്ട്പുട്ട് ഓപ്പറേറ്ററുകളുടെ സംയോജനം എന്നു പറയുന്നു. ഇൻപുട്ട് ഓപ്പറേറ്ററുകൾ ചെയ്യുമ്പോൾ ആദ്യം നൽകുന്ന വില ആദ്യത്തെ വേരിയബിളിന് ലഭിക്കും. രണ്ടാമത്തേ വില രണ്ടാമത്തേതിന് അങ്ങനെ ഇടത്തുനിന്ന് വലത്തേക്ക് വില ലഭിക്കും. ഉദാഹരണമായി `cin>>x>>y>>z;` ഒന്നാമത് നൽകുന്ന വില x നും രണ്ടാമത്തേത് y ക്കും മൂന്നാമത്തേത് z നും ലഭിക്കും. പ്രവർത്തനസമയത്ത് വിലനൽകുമ്പോൾ വേരിയബിളുകളുടെ വിലകൾ തമ്മിൽ വേർതിരിക്കുന്നതിന് സ്പൈസ്, ബാർ, ടാബ്, അല്ലെങ്കിൽ എന്റർ കീ ഇവ ഏതെങ്കിലും ഉപയോഗിക്കാം.

ഇതുപോലെ ഒന്നിലധികം വേരിയബിളുകളുടെ വിലകൾ മോണിറ്ററിൽ കാണിക്കുന്നതിനായി താഴെ പറയുന്ന രീതി ഉപയോഗിക്കാം.

```
cout<<x<<y<<z;
```

വേരിയബിളുകൾ സ്ഥിരാംഗങ്ങൾ പദപ്രയോഗങ്ങൾ എന്നിവ ഒരുമിച്ച് ഔട്ട്പുട്ട് ചെയ്യാനായി താഴെ പറയുന്ന രീതി ഉപയോഗിക്കാം.

```
cout<<"The number is “<<z;
```

ഔട്ട്പുട്ട് ഓപ്പറേറ്ററുകൾ കാസ്കേഡ് ചെയ്യുമ്പോൾ വലത്തുനിന്ന് ഇടത്തേക്കായിരിക്കും ഔട്ട്പുട്ട് വിലകൾ പ്രദർശിപ്പിക്കുന്നത്.

‘<<’, ‘>>’ എന്നീ ഓപ്പറേറ്ററുകളെ ഒരേ പ്രസ്താവനയിൽ ഉപയോഗിക്കാൻ കഴിയില്ല എന്നത് ശ്രദ്ധിക്കുമല്ലോ.

നമുക്ക് സംഗ്രഹിക്കാം

ഡാറ്റയുടെ തരം തിരിച്ചറിയുന്നതിനും അവയുമായി ബന്ധപ്പെട്ട ക്രിയകളെ കൈകാര്യം ചെയ്യുന്നതിനുമുള്ള ഒരു ഉപാധിയാണ് ഡാറ്റ ഇനങ്ങൾ. ഓരോ ഡാറ്റ ഇനത്തിലെ ഡാറ്റയ്ക്കും അതിന്റേതായ വലുപ്പവും പരിധിയുമുണ്ട്. വേരിയബിളുകൾ നിർവചിക്കാൻ ഡാറ്റ ഇനങ്ങൾ ഉപയോഗിക്കുന്നു. C++ ൽ വിവിധ ക്രിയകൾക്കായി വ്യത്യസ്തതരം ഓപ്പറേറ്ററുകൾ ലഭ്യമാണ്. ഓപ്പറേറ്ററുകൾ ഓപ്പറന്റുകളുമായി (ഡാറ്റ) കൂട്ടി ചേർക്കുമ്പോൾ പ്രയോഗങ്ങൾ രൂപപ്പെടുന്നു. മൂന്നു തരത്തിലുള്ള പ്രയോഗങ്ങളാണുള്ളത് - അരിത്മാറ്റിക്, റിലേഷണൽ, ലോജിക്കൽ. ഒരു പ്രോഗ്രാമിന്റെ ഏറ്റവും ചെറിയ പ്രവർത്തന ഭാഗമാണ് പ്രസ്താവന. വേരിയബിളിനെ പ്രഖ്യാപിക്കുന്ന പ്രസ്താവനകൾ പ്രോഗ്രാമിൽ ഒരു വേരിയബിളിനെ നിർവചിക്കുകയും അവക്ക് മെമ്മറി സ്ഥാനം നീക്കി വയ്ക്കുകയും ചെയ്യുന്നു. വില നൽകൽ പ്രസ്താവനകൾ, ഇൻപുട്ട് പ്രസ്താവനകൾ, ഔട്ട്പുട്ട് പ്രസ്താവനകൾ മുതലായവ കമ്പ്യൂട്ടറുകൾക്ക് നിർദ്ദേശങ്ങൾ നൽകാൻ സഹായിക്കുന്നു.

പഠന നേട്ടങ്ങൾ

ഈ അധ്യായത്തിൽ പൂർത്തികരണത്തോടെ പഠിതാവിന്

- C++ ലെ വിവിധ ഡാറ്റ ഇനങ്ങൾ തിരിച്ചറിയാൻ സാധിക്കുന്നു.
- ഉചിതമായ വേരിയബിളുകൾ തിരഞ്ഞെടുക്കുവാൻ സാധിക്കുന്നു.
- വിവിധ ഓപ്പറേറ്ററുകൾ പരീക്ഷച്ചു നോക്കുവാൻ സാധിക്കുന്നു.
- വിവിധ I/O ഓപ്പറേറ്ററുകൾ പ്രയോഗിക്കുവാൻ സാധിക്കുന്നു.
- വിവിധ പ്രയോഗങ്ങളും പ്രസ്താവനകളും എഴുതുവാൻ സാധിക്കുന്നു.

മാതൃക ചോദ്യങ്ങൾ

ഹ്രസ്വോത്തര ചോദ്യങ്ങൾ

1. ഡാറ്റ ഇനം എന്നാലെന്ത്? C++ ലെ മുൻ നിർവചിത ഡാറ്റ ഇനങ്ങളുടെ പേരെഴുതുക.
2. void ഡാറ്റ ഇനത്തിന്റെ ഉപയോഗമെന്ത്?
3. കോൺസ്റ്റന്റ് (constant) എന്നാലെന്ത്?
4. ഡയനാമിക് ഇനിഷ്യലൈസേഷൻ എന്നാലെന്ത്?
5. ഓപ്പറേറ്ററിന്റെ നിർവചനം എഴുതുക.
6. യൂനറി ഓപ്പറേഷൻ എന്നാലെന്ത്?

7. പ്രഖ്യാപന പ്രസ്താവന എന്നാലെന്ത്?

8. ">>" (ഇൻപുട്ട് ഓപ്പറേറ്റർ), "<<" (ഔട്ട്പുട്ട് ഓപ്പറേറ്റർ) എന്നിവയുടെ പേരെഴുതുക.

9. താഴെ കൊടുത്തിരിക്കുന്നവ പരിഗണിച്ച് a = 5/3 എന്ന പ്രയോഗത്തിന്റെ ഉത്തരം എഴുതുക.

   i. a ഒരു float ഡാറ്റ ഇനം ആണെങ്കിൽ

   ii. a ഒരു int ഡാറ്റ ഇനം ആണെങ്കിൽ

10. i = 4, j = 5, k = 2 ആണെങ്കിൽ താഴെ തന്നിരിക്കുന്ന പ്രയോഗങ്ങളുടെ ഉത്തരം എഴുതുക

    (i) (5*++j)%6 (ii) (5*j++)%6

11. താഴെ കൊടുത്തിരിക്കുന്ന പ്രയോഗങ്ങളിലെ ഓപ്പറേറ്ററുകളുടെ മുൻഗണനാക്രമം എഴുതുക.

    (i) i+5>=j-6 (ii) s+10<p-2+2*q

12. "ans" ന്റെ വില 6 ആണെങ്കിൽ താഴെ തന്നിരിക്കുന്ന പ്രയോഗങ്ങളുടെ ഉത്തരം എഴുതുക

    (i) cout <<ans = 8 ; (ii) cout << ans == 8

ലഘു ഉപന്യാസ ചോദ്യങ്ങൾ

1. വേരിയബിൾ എന്നാലെന്ത്? വേരിയബിളുമായി ബന്ധപ്പെട്ടിട്ടുള്ള രണ്ട് വിലകൾ ഏതെല്ലാം?

2. ലോജിക്കൽ ഓപ്പറേറ്ററുകൾ വിശദീകരിക്കുക.

3. താഴെ തന്നിട്ടുള്ള C++ പ്രസ്താവനകളിൽ തെറ്റുകൾ ഉണ്ടെങ്കിൽ കണ്ടെത്തി കാരണം വിശദമാക്കുക.

   (i) `cout<< "a=" a;`

   (ii) `m=5,n=12;015`

   (iii) `cout << "x" ; <<x;`

   (iv) `cin >> y`

   (v) `cin >> "\n" >> y ;`

   (vi) `cout >> \n "abc"`

   (vii) `a = b + c`

   (viii) `break = x`

4. റിലേഷണൽ ഓപ്പറേറ്ററുകളുടെ ധർമ്മമെന്ത്? "=", "==" എന്നിവ താരതമ്യം ചെയ്യുക.

ഉപന്യാസ ചോദ്യങ്ങൾ

1. C++ ൽ ഉപയോഗിക്കുന്ന ഓപ്പറേറ്ററുകൾ വിശദീകരിക്കുക.

2. C++ ലെ പ്രയോഗങ്ങൾ വിശദീകരിക്കുക.

6

പ്രധാന ആശയങ്ങൾ

- C++ പ്രോഗ്രാമിന്റെ ഘടന
  - പ്രിപ്രൊസ്സർ നിർദ്ദേശം
  - ഹെഡർഫയലുകൾ
  - വേരിയബിളുകൾ
  - C++ മാതൃക പ്രോഗ്രാം
- പ്രോഗ്രാം എഴുതുന്നതിനുള്ള മാർഗ്ഗ നിർദ്ദേശങ്ങൾ
- വേരിയബിളിന് പ്രാരംഭ വില നൽകൽ
- const ആക്സസ് മോഡിഫയറിന്റെ ഉപയോഗം
- ടൈപ്പ് മോഡിഫയറുകൾ
- കൂടുതൽ ഓപ്പറേറ്ററുകൾ
  - അരിത്മെറ്റിക് അസൈൻമെന്റ്
  - ഇൻക്രിമെന്റ്, ഡിക്രിമെന്റ്
  - ഓപ്പറേറ്ററുകളുടെ മുൻഗണനാക്രമം
- ഇനം മാറ്റൽ (type conversion)
  - ആന്തരിക ഇനം മാറ്റൽ (implicit conversion)
  - ബാഹ്യ ഇനം മാറ്റൽ (explicit conversion)

R2R6J1

പ്രോഗ്രാമിങ് – ഒരു ആമുഖം

C++ പ്രോഗ്രാം എഴുതുവാനുള്ള IDE നമ്മൾ പരിചയപ്പെട്ടു കഴിഞ്ഞു. എങ്ങനെയാണ് ഓപ്പറേഷനുകളെ C++ ൽ പ്രതിനിധാനം ചെയ്യുന്നത് എന്ന് കഴിഞ്ഞ അധ്യായത്തിൽ ചർച്ച ചെയ്തു. കൂടാതെ കമ്പ്യൂട്ടറിന് നിർദ്ദേശങ്ങൾ നൽകുന്നതിന് ആവശ്യമായ വിവിധതരം C++ പ്രസ്താവനകളും നാം പഠിച്ചു കഴിഞ്ഞു. ഇത്തരം പ്രസ്താവനകൾ ഉപയോഗിച്ച് ലളിതമായ പ്രശ്നപരിഹാരത്തിനുള്ള പ്രയോഗങ്ങൾ എഴുതുവാൻ നമുക്കിപ്പോൾ സാധ്യമാണ്. എന്നാൽ ഒരു കൂട്ടം പ്രസ്താവനകൾ മാത്രം കൂട്ടിച്ചേർത്താൽ ഒരു പ്രോഗ്രാമായി മാറുന്നില്ല. ഒരു C++ പ്രോഗ്രാമിന് ഒരു പ്രത്യേക ഘടനയുണ്ട്. ഈ അധ്യായത്തിൽ C++ പ്രോഗ്രാമിന്റെ ഘടനയും, അതിനനുസരിച്ച് പ്രോഗ്രാം എഴുതുന്ന രീതിയും ചർച്ച ചെയ്യുന്നു. പ്രോഗ്രാമുകൾ ഒതുക്കത്തോടെ എഴുതി വേഗത്തിൽ പ്രവർത്തിപ്പിക്കുന്നതിനാവശ്യമായ കൂടുതൽ ഓപ്പറേറ്ററുകളും ഇവിടെ പരിചയപ്പെടുത്തുന്നു.

6.1 C++ പ്രോഗ്രാമിന്റെ ഘടന:

നാം ഇതുവരെ ചർച്ച ചെയ്ത പ്രസ്താവനകൾ ഉപയോഗിച്ച് ലഘുവായ പ്രശ്നങ്ങൾ പരിഹരിക്കാവുന്ന സ്ഥിതിയിൽ നാം ഇപ്പോൾ എത്തി കഴിഞ്ഞു. എന്നാൽ ഒരു കൂട്ടം പ്രസ്താവനകൾ മാത്രം ചേർന്നാൽ ഒരു പ്രോഗ്രാമാകുകയില്ല. C++ പ്രോഗ്രാമിന് ഒരു സവിശേഷ ഘടനയുണ്ട്. അത് ഒന്നോ അതിലധികമോ ഫങ്ഷനുകളുടെ ശേഖരമാണ്. ഫങ്ഷൻ എന്നാൽ ഒരു പേരിൽ സ്വരൂപിച്ചിരിക്കുന്നതും ഒരു പ്രത്യേക കാര്യം ചെയ്യുന്നതിനായുമുള്ള പ്രസ്താവനകളുടെ കൂട്ടമാണ്. ഒരു C++ പ്രോഗ്രാമിൽ ഒന്നിലധികം ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിനാൽ അവ ഓരോന്നും ഫങ്ഷനുകളും വ്യത്യസ്ത പേരുകളിൽ ആയിരിക്കണം തിരിച്ചറിയപ്പെടേണ്ടത്. എല്ലാ പ്രോഗ്രാമിലും ഏറ്റവും അത്യാവശ്യമായി ഉണ്ടായിരിക്കേണ്ട ഫങ്ഷനാണ് main() ഫങ്ഷൻ.

ഒരു C++ പ്രോഗ്രാമിന്റെ ഘടന താഴെ കൊടുത്തിരിക്കുന്നു

```
#include <header file>
using namespace identifier;
int main()
{
     statements;
        :
        :
        :
     return 0;
}
```

ഒന്നാമത്തെ വരിയെ പ്രീ - പ്രൊസസ്സർ നിർദ്ദേശം എന്നും രണ്ടാമത്തെ വരിയെ നെയിം സ്പേസ് പ്രസ്താവന എന്നും വിളിക്കുന്നു. മൂന്നാമത്തെ വരിയിൽ ഫങ്ഷൻ ഹെഡറും തുടർന്നുള്ള വരികളിൽ ഒരു ജോഡി ബ്രാക്കറ്റുകൾക്കുള്ളിൽ ഉള്ള ഒരു കൂട്ടം പ്രസ്ഥാവ നകളുമാണ് അടങ്ങിയിരിക്കുന്നത്.

പ്രോഗ്രാമിലെ ഈ ഓരോ ഭാഗങ്ങളും നമുക്ക് ചർച്ച ചെയ്യാം.

6.1.1 പ്രീപ്രൊസസ്സർ നിർദ്ദേശങ്ങൾ (Preprocessor directive)

പ്രീ പ്രോസസ്സർ നിർദ്ദേശങ്ങളോടു കൂടിയാണ് ഒരു C++ പ്രോഗ്രാം ആരംഭിക്കുന്നത്.ഇവ കംപൈലറിനുള്ള നിർദേശ പ്രസ്താവനകളാണ്. കംപൈലേഷൻ ആരംഭിക്കുന്നതിനു മുമ്പ് കമ്പൈലർ ചെയ്യേണ്ടുന്ന കാര്യങ്ങൾ ഇത് നിർദ്ദേശിക്കുന്നു. പ്രോഗ്രാം പ്രസ്താവ നകൾ അല്ലാത്തതും എന്നാൽ പ്രോഗ്രാമിൽ ഉൾപ്പെട്ടിട്ടുള്ളതുമായ വരികളാണ് പ്രീപ്രൊ സസ്സർ നിർദ്ദേശങ്ങൾ. ഈ വരികൾ എപ്പോഴും # ചിഹ്നത്തോടു കൂടിയാണ് തുടങ്ങുന്ന ത്. പ്രോഗ്രാമിന് ആവശ്യമായ സൗകര്യങ്ങൾ ലഭ്യമാക്കാൻ C++ ലൈബ്രറിയിലെ ശീർഷക ഫയലുകളെ #include എന്നുതുടങ്ങുന്ന പ്രീപ്രോസസ്സർ നിർദ്ദേശം ഉപയോഗിച്ച് ബന്ധി പ്പിക്കുന്നു. ഈ വരികളുടെ അവസാനം അർദ്ധവിരാമം (;) ആവശ്യമില്ല. വിവിധ ശീർഷക ഫയലുകൾക്ക് വേണ്ടി വ്യത്യസ്ത #include പ്രസ്താവനകൾ ഉപയോഗിക്കണം. #define, #undef മുതലായവ മറ്റു ചില പ്രീപ്രോസസ്സർ നിർദ്ദേശങ്ങളാണ്.

6.1.2 ഹെഡർ ഫയലുകൾ (Header files)

ഫംഗ്ഷനുകൾ, ഒബ്ജക്റ്റുകൾ മുൻനിർവചിത - രൂപീകൃത ഡാറ്റഇനങ്ങൾ എന്നിവയെ ക്കുറിച്ചുള്ള വിവരങ്ങൾ കംപൈലറിനോടൊപ്പം ലഭ്യമായിട്ടുള്ള ശീർഷകംമായിട്ടുള്ള ശീർഷക ഫയലുകളിൽ അടങ്ങിയിരിക്കുന്നു. ലൈബ്രറിയിൽ സൂക്ഷിച്ചിട്ടുള്ള ഇത്തരം നിരവധി ഫയലുകൾ C++ പ്രോഗ്രാമുകളെ പിന്തുണയ്ക്കുന്നു. ഇത്തരത്തിലുള്ള ഏതെങ്കിലും വിവരങ്ങൾ ആവശ്യമുള്ള പ്രോഗ്രാമുകളിൽ അവ ഉപയോഗിക്കുന്നതിന് അതുമായി ബന്ധപ്പെട്ട ഹെഡർ ഫയൽ ഉൾപ്പെടുത്തണം. ഉദാഹരണത്തിന് മുൻ നിർവ ചിത ഒബ്ജക്ടുകളായ cin, cout എന്നിവ നമുക്ക് ഉപയോഗിക്കേണ്ടതായി വരുമ്പോൾ പ്രോഗ്രാമിന്റെ തുടക്കത്തിൽ താഴെ പറയുന്ന പ്രസ്താവന നമ്മൾ ഉപയോഗിക്കണം.

#include <iostream> എന്ന ഹെഡർ ഫയലിൽ cin, cout എന്നീ ഒബ്ജക്ടുകളുടെ വിവര ങ്ങൾ അടങ്ങിയിരിക്കും. ഹെഡർ ഫയലുകൾക്ക് h എക്സ്റ്റൻഷൻ ഉണ്ടെങ്കിലും GCC ൽ

അതു ഉപയോഗിക്കരുത്. പക്ഷേ ടർബോ C++ IDE പോലെയുള്ള മറ്റു ചില കംപൈലറുകൾക്ക് ഈ എക്സ്റ്റൻഷൻ നിർബന്ധമാണ്.

6.1.3. നെയിംസ്പേസ് എന്ന ആശയം (Concept of namespace)

ഒരു പ്രോഗ്രാമിൽ ഒരേ വ്യാപ്തിയിൽ ഒരേ പേരിലുള്ള ഒന്നിലധികം ഐഡന്റിഫയറുകൾ (വേരിയബിളുകൾ അല്ലെങ്കിൽ ഫംങ്ഷനുകൾ) ഉണ്ടായിരിക്കാൻ പാടില്ല. ഉദാഹരണത്തിന് നമ്മുടെ വീട്ടിൽ രണ്ടോ അതിലധികമോ ആളുകൾക്ക് (അല്ലെങ്കിൽ ജീവജാലങ്ങൾക്ക്) ഒരേ പേരുണ്ടാവില്ല. അങ്ങനെയുണ്ടെങ്കിൽ തീർച്ചയായും വീട്ടിനുള്ളിൽ അവരെ പേരു കൊണ്ട് തിരിച്ചറിയുക എന്നത് വിഷമകരമാകും. അതുകൊണ്ട് നമ്മുടെ വീട്ടിൽ ഓരോ പേരും അനന്യമായിരിക്കണം. എന്നാൽ നമ്മുടെ അയൽപക്കത്തെ വീട്ടിൽ സമാനമായ പേരുള്ള ഒരാൾ (അല്ലെങ്കിൽ ജീവജാലം) ഉണ്ടായിരിക്കാം. അതാത് പരിധിക്കുള്ളിൽ വ്യക്തികളെ പേരു കൊണ്ട് തിരിച്ചറിയുന്നതിന് ഇത് യാതൊരു ആശയക്കുഴപ്പവുമുണ്ടാക്കില്ല. പക്ഷേ പുറമേ നിന്നൊരു വ്യക്തിക്ക് പേരു മാത്രം ഉപയോഗിച്ച് കൊണ്ട് ഇവരെ തിരിച്ചറിയാൻ കഴിയില്ല. അതിനായി വീട്ടുപേരു കൂടി പരാമർശിക്കേണ്ടതുണ്ട്.

നെയിം സ്പേസ് എന്ന ആശയം വീട്ടുപേരിനു സമാനമാണ്. ഒരു പ്രത്യേക നെയിംസ്പേസുമായി വ്യത്യസ്ത ഐഡന്റിഫയറുകൾ ബന്ധപ്പെട്ടിരിക്കുന്നു. ഓരോ ഇനവും വ്യത്യസ്തമായിരിക്കുന്ന ഒരു ഗണത്തിന്റെ പേരാണിത്. വേരിയബിളുകൾക്കും ഫംഗ്ഷനുകൾക്കുമായി പ്രത്യേകം നെയിം സ്പേസുകൾ സൃഷ്ടിക്കുന്നതിന് ഉപയോക്താവിനു അനുവാദമുണ്ട്. ഒരു നെയിംസ്പേസിനു പേരു കൊടുക്കാൻ നമുക്ക് ഒരു ഐഡന്റിഫയർ ഉപയോഗിക്കാം. പ്രോഗ്രാമിംഗിൽ ഉപയോഗിക്കുന്ന ഘടകങ്ങളെ ഏത് നെയിം സ്പേസിൽ തിരയണമെന്ന് using എന്ന കീവേർഡ് സാങ്കേതികമായി കംപൈലറിനോട് പറയുന്നു. C++ ൽ standard എന്നതിന്റെ ചുരുക്കെഴുത്താണ് std. cin, cout തുടങ്ങി മറ്റ് പല ഒബ്ജക്ടുകളും നിർവചിച്ചിട്ടുള്ള ഒരു നെയിം സ്പേസ് ആണിത്. അതിനാൽ ഒരു പ്രോഗ്രാമിൽ ഇവ ഉപയോഗിക്കണമെങ്കിൽ std::cin, std::cout എന്ന മാതൃക നാം പിന്തുടരേണ്ടതാണ്. using name space std എന്ന പ്രസ്താവന പ്രോഗ്രാമിൽ ഉപയോഗിക്കുന്നതിലൂടെ ഇത്തരത്തിലുള്ള വിശദമായ പരാമർശങ്ങൾ ഒഴിവാക്കാവുന്നതാണ്. അത്തരമൊരു സാഹചര്യത്തിൽ കംപൈലർ cin, cout, endl മുതലായവയ്ക്കായി ഈ നെയിംസേപസിൽ തിരയുന്നു. cin, cout, endl അല്ലെങ്കിൽ അതുപോലെയുള്ളവ എപ്പോഴൊക്കെ ഒരു C++ പ്രോഗ്രാമിൽ കമ്പ്യൂട്ടർ കാണുന്നുവോ, അവയെ std::cin, std::cout, std::endl എന്നിങ്ങനെ വ്യഖ്യാനിക്കുന്നു.

using name space std എന്ന പ്രസ്താവന യഥാർത്ഥത്തിൽ പ്രോഗ്രാമിലേക്ക് ഒരു ഫംഗ്ഷനും കൂട്ടിച്ചേർക്കുന്നില്ല, #include<iostream> എന്ന നിർദ്ദേശമാണ് cin, cout, endl അതുപോലെയുള്ളവ ഉൾപ്പെടുത്തുന്നത്.

6.1.4 `main()` ഫങ്ഷൻ (The `main ()` function)

എല്ലാ C++ പ്രോഗ്രാമിലും main() എന്നു പേരുള്ള ഫങ്ഷൻ ഉൾപ്പെട്ടിരിക്കുന്നു. പ്രോഗ്രാമിന്റെ പ്രവർത്തനം ആരംഭിക്കുന്നതും അവസാനിക്കുന്നതും main() ഫങ്ഷനിലാണ്. മറ്റ് ഏതെങ്കിലും ഫങ്ഷനുകൾ നാം പ്രോഗ്രാമിൽ ഉപയോഗിക്കുന്നുണ്ടെങ്കിൽ അവയെ വിളിക്കുന്നത് main() ഫങ്ഷനിൽ നിന്നാണ്. സാധാരണയായി main() ഫങ്ഷൻ മുമ്പായി ഒരു ഡാറ്റ ഇനം ഉണ്ടായിരിക്കും. GCC -ൽ ഇത് int ആയിരിക്കണം.

main () ഫങ്ഷന്റെ ഹെഡറിനെ തുടർന്ന് ഒരു ജോഡി ബ്രാക്കറ്റുകൾക്കുള്ളിൽ ഒന്നോ അതിലധികമോ പ്രസ്താവനകൾ അടങ്ങിയ ഫങ്ഷന്റെ ചട്ടക്കൂടും ഉണ്ടായിരിക്കും. ഈ ഘടന main ഫങ്ഷന്റെ നിർവചനം എന്ന് അറിയപ്പെടുന്നു. ഓരോ പ്രസ്താവനയും ഒരു അർദ്ധവിരാമത്തിൽ ';' അവസാനിക്കുന്നു. പ്രസ്താവനകൾ നിർവഹിക്കാവുന്നവയോ നിർവഹിക്കാനാവത്തവയോ ആകാം. കമ്പ്യൂട്ടർ ചെയ്യേണ്ട കാര്യങ്ങൾക്കുള്ള നിർദേശ ങ്ങളാണ് നിർവഹണ പ്രസ്താവനകൾ പ്രതിനിധാനം ചെയ്യുന്നത്. നിർവഹിക്കാനാവാത്ത പ്രസ്താവനകൾ കമ്പയിലറിനെയോ പ്രോഗ്രാമറെയോ ഉദ്ദേശിച്ചുള്ളവയാണ്. അവ വിവ രാധിഷ്ഠിത പ്രസ്താവനകൾ ആണ്. main() ഫംഗ്ഷന് ഉള്ളിലെ അവസാനത്തെ പ്രസ്താ വന return 0 എന്നായിരിക്കും. ഈ പ്രസ്താവനകൾ നാം ഉപയോഗിച്ചില്ലെങ്കിലും പ്രോഗ്രാ മിൽ അത് ഒരു തരത്തിലുള്ള പിശകും വരുത്തുന്നില്ല. ഇതിന്റെ പ്രസക്തി അധ്യായം 10-ൽ ചർച്ച ചെയ്യാം. ഓരോ പ്രസ്താവനകളും പുതിയ വരികളിൽ തുടങ്ങണമെന്ന് നിർബ്ബ ന്ധമില്ലാത്തതിനാൽ C++ ഒരു സ്വതന്ത്ര രൂപത്തിലുള്ള ഭാഷയാണ്. അതുപോലെ ഒരു പ്രസ്താവനക്ക് ഒന്നിൽ കൂടുതൽ വരികൾ ഉപയോഗിക്കാവുന്നതാണ്.

6.1.5 ഒരു മാതൃക പ്രോഗ്രാം (A sample program)

പൂർണ്ണമായ ഒരു പ്രോഗ്രാം നമുക്ക് പരിശോധിച്ച് അതിന്റെ സവിശേഷതകൾ വിശദ മായി പരിചയപ്പെടാം. ഈ പ്രോഗ്രാം സ്ക്രീനിൽ ഒരു വാചകം പ്രദർശിപ്പിക്കും.

```
#include<iostream.h>
void main()
{
    cout<<"Hello, Welcome to C++";
}
```

താഴെ പറയുന്ന രീതിയിൽ ഈ പ്രോഗ്രാമിന് ഏഴു വരികളുണ്ട്.

വരി 1. പ്രീ പ്രോസസ്സർ നിർദേശം #include എന്നത് iostream.h എന്ന ഹെഡർ ഫയലിനെ ഈ പ്രോഗ്രാമുമായി ബന്ധിപ്പിക്കുന്നു.

വരി 2. using name space std എന്ന പ്രസ്താവന cout എന്ന ഐഡന്റിഫയറിനെ പ്രോഗ്രാമിൽ ലഭ്യമാക്കുന്നു.

വരി 3. പ്രോഗ്രാമിൽ നിർബന്ധമായും ഉണ്ടായിരിക്കേണ്ട മെയിൻ എന്ന ഫങ്ഷന്റെ ഹെഡർ

വരി 4. ആദ്യത്തെ ബ്രാക്കറ്റ് ({)പ്രസ്താവനകൾ തുടങ്ങുന്നു എന്നു കാണിക്കുന്നു.

വരി 5. നാം പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുമ്പോൾ ഈ ഔട്ട്പുട്ട് പ്രസ്താവന പ്രവർത്തിച്ച് Hello, welcome to C++ എന്ന് മോണിറ്ററിൽ പ്രദർശിപ്പിക്കുന്നു. Cout പ്രസ്താവന ഈ പ്രോഗ്രാമിൽ ഉപയോഗിക്കുന്നതിന് iostream എന്ന് ഹെഡർ ഫയൽ ഉൾപ്പെടുത്തിയിട്ടുണ്ട്.

വരി 6. return പ്രസ്താവന main() ഫങ്ഷന്റെ പ്രവർത്തനം അവസാനിപ്പിക്കുന്നു. ഈ പ്രസ്താവന main ഫങ്ഷനെ സംബന്ധിച്ചിടത്തോളം നിർബന്ധമല്ല.

വരി 7. അവസാന ബ്രാക്കറ്റ് (}) ഈ പ്രോഗ്രാം അവസാനിപ്പിച്ചതായി സൂചിപ്പിക്കുന്നു.

6.2 പ്രോഗ്രാം എഴുതുന്നതിനുള്ള മാർഗ്ഗ നിർദ്ദേശങ്ങൾ

ഒരു പ്രോഗ്രാം കോഡ് യുക്തിസഹവും സ്പഷ്ടവും തെറ്റുകൾ പെട്ടെന്ന് കണ്ടെത്തുവാൻ കഴിയുന്നതുമാണെങ്കിൽ അത് ഒരു നല്ല സോഴ്സ് കോഡ് ആയിരിക്കും. പ്രോഗ്രാമുകൾ എഴുതുമ്പോൾ ചില രീതികൾ പിന്തുടരുകയാണെങ്കിൽ ഈ സവിശേഷതകൾ നമുക്ക് അനുഭവവേദ്യമാക്കാം.

ശൈലീപരമായ പ്രോഗ്രാമുകൾ എഴുതുന്നതിനുള്ള ചില മാർഗ്ഗ നിർദ്ദേശങ്ങൾ ഈ ഭാഗത്ത് ചർച്ച ചെയ്യുന്നു.

ഐഡന്റിഫയറുകൾക്ക് യോജിച്ച പേര് നൽകുക.

ഒരു ജോലിക്കാരന്റെ കിഴിവുകൾക്കുശേഷമുള്ള ശമ്പളം നമുക്ക് കണക്കാക്കണം എന്നിരിക്കട്ടെ. താഴെ കാണുന്ന രീതിയിൽ നമുക്ക് കോഡ് ചെയ്യാവുന്നതാണ്.

ഇവിടെ A എന്നത് മിച്ച ശമ്പളവും b മൊത്ത ശമ്പളവും c ആകെ കിഴിവും ആണ്, എന്നാൽ ഈ പേരുകൾ അവയുടെ ഉപയോഗത്തെ പ്രതിഫലിപ്പിക്കുന്നില്ല. ഇതേ പ്രസ്താവന താഴെ പറയുന്ന രീതിയിലായാൽ കൂടുതൽ വ്യക്തമായിരിക്കും.

Net_Slary=Gross_Slary-Deduction;

ഇവിടെ വേരിയബിളുകളുടെ പേരുകൾ അവയുടെ മൂല്യവുമായി പൊരുത്തമുള്ളതും പെട്ടെന്ന് ഓർത്തിരിക്കാൻ പറ്റുന്നതുമാണ്. ഈ പേരുകൾ അവയുടെ ഉദ്ദേശ്യത്തെ പ്രതിഫലിപ്പിക്കുന്നു.ഇത്തരം പേരുകളെ ന്യൂമോണിക് പേരുകൾ (mnemonic names) എന്നു വിളിക്കുന്നു. പേരുകൾ സ്വീകരിക്കുമ്പോൾ താഴെപ്പറയുന്ന കാര്യങ്ങൾ ശ്രദ്ധിക്കണം.

1. വേരിയബിളുകൾ, ഫങ്ഷനുകൾ, പ്രൊസീഡറുകൾ എന്നിവക്ക് നല്ല ന്യൂമോണിക് പേരുകൾ തിരഞ്ഞെടുക്കാം.

   ഉദാഹരണം: avg_hgt, Roll_No, emp_code, Sum Of Digits, തുടങ്ങിയവ
2. ബന്ധപ്പെട്ട വേരിയബിളുകൾക്ക് നിലവാരമുള്ള പിൻ വാക്കുകളും, മുൻ വാക്കുകളും ഉപയോഗിക്കാം.

   ഉദാഹരണം: (മൂന്ന് സംഖ്യകൾക്കായി) num1, num2, num3
3. പ്രോഗ്രാമിന്റെ തുടക്കത്തിൽതന്നെ സ്ഥിരാംഗങ്ങൾക്ക് പേരുകൾ നൽകുക.

   ഉദാഹരണം: const float PI = 3.14;

വ്യക്തവും ലളിതവുമായ പ്രയോഗങ്ങൾ ഉപയോഗിക്കുക

പ്രവർത്തനസമയം കുറയ്ക്കുന്നതിനായി പ്രോഗ്രാമുകളുടെ ലാളിത്യം നഷ്ടപ്പെടുത്തുന്ന പ്രവണത ചില ആളുകൾക്കുണ്ട്. ഇത് ഒഴിവാക്കേണ്ടതാണ്. താഴെ പറയുന്ന ഉദാഹരണം പരിഗണിക്കുക. x നെ yകൊണ്ട് ഹരിച്ച്കിട്ടുന്ന ശിഷ്ടം കാണുന്നതിന് y=x-(x/n)*n; എന്ന പ്രസ്താവന ഉപയോഗിക്കാം. ഇതേ കാര്യത്തിനായി താഴെ കാണുന്ന ലളിതവും സുന്ദരവുമായ കോഡ് ഉപയോഗിക്കാവുന്നതാണ്.

y=x%n;

അതു കൊണ്ട് ഒരു പ്രോഗ്രാമിനെ ലളിതവും വ്യക്തവുമാക്കുന്നതിന് ലളിതമായ കോഡുകൾ ഉപയോഗിക്കുന്നതാണ് നല്ലത്.

ആവശ്യമുള്ളിടത്ത് കമന്റുകൾ ഉപയോഗിക്കുക.

ഒരു പ്രോഗ്രാമിന്റെ മുകളിൽ വിവരണം നൽകുന്നതിന് കമന്റുകൾ വളരെ പ്രധാനപ്പെട്ട പങ്ക് വഹിക്കുന്നു. അവയെ പ്രോഗ്രാമുകളെ വിശദീകരിക്കുവാനുള്ള വരികളായിട്ടാണ്

പ്രോഗ്രാമിനുള്ളിൽ കൂട്ടി ചേർത്തിരിക്കുന്നത്. കംപൈലറുകൾ അവയെ അവഗണിക്കുന്നു. C++ ൽ കമന്റുകൾ എഴുതുന്നതിന് രണ്ടു മാർഗ്ഗങ്ങളുണ്ട്.

ഒറ്റവരി കമന്റ്: '//' ചിഹ്നങ്ങളാണ് ഒറ്റവരി കമന്റുകൾ എഴുതാനായി ഉപയോഗിക്കുന്നത്. ഒരു വരിയിൽ // ന് ശേഷമുള്ള വാക്യങ്ങൾ കമന്റുകളായി C++ കമ്പൈലർ കണക്കാക്കുന്നു.

ഖണ്ഡിക കമന്റ് (multiline comment): /* നും */ നും ഇടയിൽ എഴുതുന്ന എന്തിനെയും കമ്പൈലർ കമന്റ് ആയി കണക്കാക്കുന്നു. ആയതിനാൽ ഒരു കമന്റിൽ എത്ര വരികൾ വേണമെങ്കിലും ഉൾപ്പെടുത്താം. പക്ഷേ പ്രോഗ്രാമിൽ ആവശ്യമായ പ്രസ്താവനകൾ കമന്റുകളിൽ ആയി പോകാതിരിക്കാൻ പ്രത്യേകം ശ്രദ്ധിക്കേണ്ടതാണ്.

കമന്റുകൾ നൽകുമ്പോൾ താഴെ പറയുന്ന കാര്യങ്ങൾ ശ്രദ്ധിക്കണം

- പ്രോഗ്രാമിന്റെ തുടക്കത്തിലുള്ള കമന്റുകൾ പ്രോഗ്രാമിന്റെ ഉദ്ദേശ്യത്തെ സംഗ്രഹിക്കുന്നതായിരിക്കണം.
- ഓരോ വേരിയബിളും, സ്ഥിരാംഗവും പ്രഖ്യാപിക്കുമ്പോൾ കമന്റുകൾ ഉപയോഗിക്കുക.
- സങ്കീർണമായ പ്രോഗ്രാം ഘട്ടങ്ങൾ വിശദീകരിക്കുന്നതിന് കമന്റുകൾ ഉപയോഗിക്കുക.
- പ്രോഗ്രാം എഴുതുന്ന സമയത്തു തന്നെ കമന്റുകൾ ഉൾപ്പെടുത്തുന്നതാണ് നല്ലത്.
- ലളിതവും വ്യക്തവുമായി കമന്റുകൾ എഴുതുക.

ഇന്റന്റേഷന്റെ ആവശ്യകത

കമ്പ്യൂട്ടർ പ്രോഗ്രാമിംഗിൽ പ്രോഗ്രാം ഘടന വ്യക്തമാക്കുന്നതിന് കോഡുകൾ മാർജിനിൽ നിന്നും നിശ്ചിത അകലത്തിൽ എഴുതുന്ന സമ്പ്രദായമുണ്ട്. ഇതിനെ ഇന്റന്റേഷൻ എന്ന് പറയുന്നു. ഇത് പ്രസ്താവനകളുടെ വായന സുഗമമാക്കുകയും പ്രോഗ്രാമിന് വ്യക്തത വരുത്തുകയും ചെയ്യുന്നു.

ഇത് പ്രോഗ്രാമിലെ പ്രസ്താവനകളുടെ വിവിധ നിലകളെ സൂചിപ്പിക്കുന്നു.

ഈ മാർഗ്ഗ നിർദ്ദേശങ്ങളുടെ ഉപയോഗം അടുത്ത ഭാഗത്തുള്ള പ്രോഗ്രാമുകളിൽ നിരീക്ഷിക്കാവുന്നതാണ്.

6.3. വേരിയബിളുകൾക്ക് പ്രാരംഭവില നൽകൽ (Veriable initialisation)

വേരിയബിളുമായി ബന്ധപ്പെട്ട L മൂല്യം (വിലാസം), R മൂല്യം (ഉള്ളടക്കം) എന്നിങ്ങനെ രണ്ട് വിലകൾ ഉണ്ടെന്ന് ഭാഗം 6.5ൽ നാം കണ്ടു. ഒരു വേരിയബിൾ പ്രഖ്യാപിക്കുമ്പോൾ അതിനായി വിലാസത്തോടുകൂടിയ ഒരു മെമ്മറി ഭാഗം നീക്കി വെക്കുന്നു. എന്തായിരിക്കും അതിന്റെ ഉള്ളടക്കം? അത് പൂജ്യം, ശൂന്യം, സ്ഥലം/ വിടവ് എന്നിവയൊന്നും ആയിരിക്കില്ല! വേരിയബിളുകൾ int ഡാറ്റയായി പ്രഖ്യാപിക്കുമ്പോൾ വേരിയബിളുകളുടെ ഉള്ളടക്കം അഥവാ R വാല്യു എന്നത് അനുവദനീയമായ പരിധിക്ക് അകത്തുള്ള ഒരു പൂർണ്ണസംഖ്യ ആയിരിക്കും. എന്നാൽ ഈ സംഖ്യ പ്രവചിക്കാൻ സാധ്യമല്ല, എല്ലായിപ്പോഴും ഒരേ വില ആയിരിക്കണമെന്നുമില്ല. അതുകൊണ്ട് ഇതിനെ ഗാർബേജ് വില (garbage value) എന്ന് വിളിക്കുന്നു. നാം വേരിയബിളിന് ഒരു വില നൽകുമ്പോൾ അതിന്റെ പഴയ വിലയെ മാറ്റി പുതിയ വില ആക്കുന്നു. വേരിയബിളിന് കംപൈലേഷൻ സമയത്തോ പ്രോഗ്രാമിന്റെ പ്രവർത്തന (execution) സമയത്തോ വില നൽകാവുന്നതാണ്. പ്രഖ്യാപന സമയത്തുതന്നെ വേരിയബിളിന് വില നൽകുന്നതിന് പ്രാരംഭ വിലനൽകൽ

(variable initialisation) എന്നു പറയുന്നു. ഈ വില കംപൈൽ സമയത്ത് മെമ്മറിയിൽ സംഭരിക്കപ്പെടുന്നു. ഇതിനായി അസ്സൈൻമെന്റ് ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നു. താഴെ കൊടുത്തിരിക്കുന്നതു പോലെ രണ്ടു രീതിയിൽ ഇത് ചെയ്യാവുന്നതാണ്.

Data type variable = value

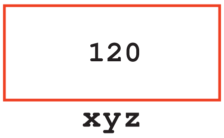
അല്ലെങ്കിൽ

Data type variable (value)

xyz എന്നൊരു വേരിയബിൾ പ്രഖ്യാപിച്ച് അതിന് 120 എന്ന വില നൽകുന്നതിനായി താഴെ പറയുന്ന രണ്ട് രീതികൾ സ്വീകാര്യമാണ്.

int xyz=120;

int xyz (120); ഈ രണ്ടു പ്രസ്താവനകളും xyz എന്ന ഇന്റിജർ വേരിയബിൾ പ്രഖ്യാപിച്ച് 120 എന്ന വില ചിത്രം 6.3 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ സൂക്ഷിക്കുകയും ചെയ്യുന്നു.



ചിത്രം 6.1: വേരിയബിളിനു പ്രാരംഭവില നൽകൽ

കൂടുതൽ ഉദാഹരണങ്ങൾ:

```
float val=0.12, b=5.234;
char k='A';
```

പ്രോഗ്രാമിന്റെ പ്രവർത്തനസമയത്തും വേരിയബിളുകൾക്ക് പ്രാരംഭവില നൽകാവുന്നതാണ്. ഇത് ഡൈനാമിക് പ്രാരംഭവില നൽകൽ എന്ന് അറിയപ്പെടുന്നു (dynamic initialisation). താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവനകളിലുള്ളതു പോലെ ഒരു പ്രയോഗത്തെ വേരിയബിളിലേക്ക് അസൈൻ ചെയ്യാൻ കഴിയും.

```
float product = x*y;
float interest = p*n*r/100.0;
```

ഒന്നാമത്തെ പ്രസ്താവനയിൽ x, y എന്നിവ പ്രവർത്തന സമയത്ത് ഗുണിച്ചു കിട്ടുന്ന ഫലമാണ് product എന്ന വേരിയബിളിന്റെ പ്രാരംഭ വില. രണ്ടാമത്തേതിൽ p*n*r/100.0; എന്നതിന്റെ ഫലമാണ്, interest എന്ന വേരിയബിളിൽ സംഭരിക്കുന്നത്.

ഡൈനാമിക് പ്രാരംഭ വിലനൽകുമ്പോൾ അസൈൻമെന്റ് ഓപ്പറേറ്ററിന് വലതുവശത്തുള്ള എല്ലാ വേരിയബിളുകളിലും സാധുവായ വില ഉണ്ടായിരിക്കണമെന്നത് ശ്രദ്ധിക്കുക. അല്ലെങ്കിൽ അപ്രതീക്ഷിത ഫലങ്ങൾ അത് സൃഷ്ടിക്കും.

6.4. Const- ആക്സസ് മോഡിഫയർ

സംഖ്യ സ്ഥിരാംഗങ്ങൾ ഉപയോഗിക്കുതിനേക്കാൾ നല്ല രീതി അവയുടെ പ്രതീകങ്ങൾ ഉപയോഗിക്കുന്നതാണ്. ഉദാഹരണമായി 3.14 അല്ലെങ്കിൽ 22.0/7.0 എന്ന് ഉപയോഗിക്കുന്നതിന് പകരം pi എന്ന പ്രതീകം നമുക്ക് ഉപയോഗിക്കാം. ഇതിനായി const എന്ന കീവേഡ് ആണ് ഉപയോഗിക്കേണ്ടത്. const എന്ന സൂചികപദം (keyword) ഉപയോഗിച്ച് ഒരു പ്രതീകാത്മക സ്ഥിരാംഗം നിർമിക്കുമ്പോൾ ആ വേരിയബിളിന്റെ വില പ്രവർത്തന സമയത്ത് മാറ്റം വരുത്താൻ കഴിയാത്തതായിതീരുന്നു. താഴെ കൊടുത്തരിക്കുന്ന പ്രസ്താവന പരിഗണിക്കുക.

```
float Pi=3.14;
```

Pi എന്ന ദശാംശസംഖ്യാ വരിയബിളിന് 3.14 എന്ന പ്രാരംഭവില നൽകിയിരിക്കുന്നു. Pi എന്നതിന്റെ മൂല്യം പ്രോഗ്രാമിന്റെ പ്രവർത്തന സമയത്ത് മാറ്റം വരുത്താവുന്നതാണ്. ഈ പ്രഖ്യാപനത്തെ താഴെപറയുന്ന രീതിയിൽ നാം പരിഷ്കരിച്ചാൽ Pi യുടെ വില പ്രോഗ്രാമിന്റെ പ്രവർത്തനസമയം മുഴുവൻ സ്ഥിരമായിരിക്കാം.

```
const float pi=3.14;
```

ഇതിന്റെ വില പിന്നീട് മാറ്റം വരുത്താൻ സാധ്യമല്ല. വേരിയബിളിൽ മൂല്യങ്ങൾ രേഖപ്പെടുത്താനും തിരിച്ചെടുക്കാനുമുള്ള അവകാശം (read/write accessibility) പരിഷ്കരിച്ച് തിരിച്ചെടുക്കുക എന്നത് മാത്രമാക്കി മാറ്റുന്നു. അതിനാൽ const എന്നത് ഒരു ആക്സസ് മോഡിഫയർ ആയി പ്രവർത്തിക്കുന്നു.

സോഫ്റ്റ്‌വെയർ വികസിപ്പിക്കുമ്പോൾ വലിയ പ്രോഗ്രാമുകൾ വികസിപ്പിക്കുന്നത് സംയുക്ത സംരംഭങ്ങളായിട്ടാണ്. ഒരേ പ്രോഗ്രാമിന്റെ പല ഭാഗങ്ങളിൽ ധാരാളം ആളുകൾ ജോലി ചെയ്യുന്നുണ്ടാകും അവർ ഒരേ വേരിയബിൾ പങ്കുവെക്കുന്നുണ്ടാകാം. ഇത്തരം സന്ദർഭങ്ങളിൽ ഒരാൾ വേരിയബിളിന്റെ ഉള്ളടക്കത്തിൽ വരുത്തുന്ന മാറ്റം മറ്റൊരാൾ തയ്യാറാക്കുന്ന കോഡിനെ ദോഷകരമായി ബാധിച്ചേക്കാം. ഇവിടെ മറ്റുള്ളവരുടെ പ്രവർത്തി വേരിയബിളിന്റെ ഉള്ളടക്കത്തെ ബാധിക്കാതെ നോക്കണം. const ഉപയോഗിച്ച് കൊണ്ട് ഇത് ചെയ്യാൻ കഴിയും.

6.5 ടൈപ്പ് മോഡിഫയറുകൾ (Type modifiers)

വ്യാപ്തി വർദ്ധിപ്പിച്ചു കൊണ്ട് അധിക സാധനങ്ങൾ ഉൾപ്പെടുത്താവുന്ന തരത്തിലുള്ള യാത്രാ ബാഗുകൾ നിങ്ങൾ കണ്ടിട്ടുണ്ടോ? സാധാരണയായി നമ്മൾ ഈ അധിക സ്ഥലം ഉപയോഗിക്കാറില്ല. ബാഗിൾ ഘടിപ്പിച്ചിട്ടുള്ള സിബ് അവയുടെ വ്യാപ്തി കൂടുന്നതിനോ കുറയ്ക്കുന്നതിനോ നമ്മെ സഹായിക്കുന്നു. അൽപ്പം വലിപ്പം കൂടിയതോ കുറഞ്ഞതോ ആയ ഡാറ്റ ഉൾക്കൊള്ളാവുന്ന ഡാറ്റ ഇനങ്ങൾ C++ലും നമുക്ക് ആവശ്യമാണ്. അതിനുള്ള C++ലെ സംവിധാനമാണ് ഡാറ്റാ ടൈപ്പ് മോഡിഫൈയറുകൾ (type modifiers). ഇവ C++ൽ ഡാറ്റ ഇനങ്ങളുടെ വലിപ്പം (size), പരിധി (range), ദശാംശത്തിനുശേഷമുള്ള സംഖ്യയുടെ വലിപ്പം (precision) എന്നിവ ക്രമീകരിക്കാൻ ഉപയോഗിക്കുന്നു. വേരിയബിൾ പ്രഖ്യാപനത്തിൽ ഡാറ്റാ ഇനത്തിന്റെ പേരിന് മുൻപായി മോഡിഫൈയറുകൾ ചേർക്കുന്നു. അതുകൊണ്ട് ഒരു ഡാറ്റാ ഇനത്തിന് അനുവദിച്ചിരിക്കുന്ന സ്ഥലവും ഡാറ്റയുടെ ചിഹ്നവും മാറ്റം വരുത്തി വിലകളുടെ പരിധി വ്യത്യാസപ്പെടുത്താൻ അനുവദിക്കുന്നു. signed, unsigned, long, short എന്നിവയാണ് പ്രധാനപ്പെട്ട മോഡിഫയറുകൾ.

ഡാറ്റ ഇനങ്ങളുടെ ശരിയായ വലിപ്പം നിങ്ങൾ ഉപയോഗിക്കുന്ന കംപ്യൂട്ടറിനെയും കംപയിലറിനെയും ആശ്രയിച്ചിരിക്കുന്നു. താഴെ പറയുന്നവ ഇത് ഉറപ്പു നൽകുന്നു.

- ഒരു double ഡാറ്റ ഇനത്തിന് ഏറ്റവും കുറഞ്ഞത് float ഡാറ്റ ഇനത്തിന്റെ വലിപ്പം ഉണ്ടാകണം.
- ഒരു long double ന് ഏറ്റവും കുറഞ്ഞത് double ഡാറ്റ ഇനത്തിന്റെയെങ്കിലും വലിപ്പമുള്ളതായിരിക്കും.

ഓരോ ഡാറ്റ ഇനങ്ങളും അവയുടെ മോഡിഫയറുകളും ടേബിൾ 6.1 ൽ കാണിച്ചിരിക്കുന്നു. (GCC കമ്പൈലറിനെ അടിസ്ഥാനമാക്കി)

Name പേര്	Description വിശദീകരണം	Size വലിപ്പം	Range പരിധി
char	Character	1 byte	signed: -128 to 127 unsigned: 0 to 255
short int (short)	Short Integer	2 bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	Integer	4 bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	Long integer	4 bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
float	Floating point number	4 bytes	$-3.4\times10^{+/-38}$ to $+3.4\times10^{+/-38}$ with approximately 7 significant digits
double	Double precision floating point number	8 bytes	$-1.7\times10^{+/-308}$ to $+1.7\times10^{+/-308}$ with approximately 15 significant digits
long double	Long double precision floating point number	12 bytes	$-3.4\times10^{+/-4932}$ to $+3.4\times10^{+/-4932}$ With approximately 19 significant digits

പട്ടിക 6.1: ഡാറ്റ ഇനങ്ങളും ടൈപ്പ് മോഡിഫയറുകളും

ഡാറ്റ ഇനങ്ങൾ എങ്ങനെ വ്യത്യാസപ്പെട്ടിരിക്കുന്നു എന്ന് നിങ്ങൾക്ക് അറിയുന്നതിനുള്ള ഉദാഹരണങ്ങളാണ് ടേബിൾ 6.1 ൽ കൊടുത്തിരിക്കുന്നത്. ഇതിലുള്ള പല വിലകളും നിങ്ങളുടെ കമ്പ്യൂട്ടറിൽ വ്യത്യസ്തമായിരിക്കാം

6.6 കൂടുതൽ ഓപ്പറേറ്ററുകൾ

സാധാരണയായി ഉപയോഗിക്കുന്ന C++ ഓപ്പറേറ്ററുകൾ നമ്മൾ ചർച്ച ചെയ്തു കഴിഞ്ഞു. പ്രോഗ്രാമുകളെ ഒതുക്കമുള്ളതാക്കുന്നതിന് ഉതകുന്ന രീതിയിലുള്ള ചില പ്രത്യേക ഓപ്പറേറ്ററുകൾ C++ൽ ഉണ്ട്. ഇവ രണ്ട് ഓപ്പറേഷനുകളെ കൂട്ടിയോജിപ്പിക്കുന്നു. അസൈൻമെന്റ് ഓപ്പറേഷനും അരിത്തമെറ്റിക് ഓപ്പറേഷനും കൂട്ടിയോജിപ്പിക്കുന്നു. ഏതാനും ചിലതു മാത്രം നമ്മൾ ഇവിടെ ചർച്ച ചെയ്യുന്നു. അരിത്തമെറ്റിക് അസൈൻമെന്റ്, ഇൻക്രിമെന്റ്, ഡിക്രിമെന്റ് എന്നീ ഓപ്പറേറ്ററുകൾ ചില ഉദാഹരണങ്ങളാണ്.

6.6.1 അരിത്മെറ്റിക് അസൈൻമെന്റ് ഓപ്പറേറ്ററുകൾ

ലളിതമായ ഒരു അരിത്മാറ്റിക് പ്രസ്താവന ചുരുക്കി സൂചിപ്പിക്കാൻ അരിത്തമാറ്റിക് വില നൽകൽ ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നു. ഉദാഹരണത്തിന് a = a +10 എന്നത് a + = 10 എന്നും എഴുതാം. ഇവിടെ + = എന്നത് അരിത്മാറ്റിക് വിലനലൽകൽ ഓപ്പറേറ്റർ ആണ്. ഈ രീതി എല്ലാ അരിത്മാറ്റിക് ഓപ്പറേറ്ററുകളിലും ഉപയോഗിക്കാവുന്നതാണ്. അവ പട്ടിക 6.10 കാണിച്ചിരിക്കുന്നു. +=, −=, *=, /=, %= എന്നിവ. C++ലെ അരിത്മാറ്റിക് വിലനൽകൽ

അരിത്മെറ്റിക് അസൈൻമെന്റ് ഓപ്പറേഷൻ	തുല്യമായ അരിത്മെറ്റിക് ഓപ്പറേഷൻ
X += 10	X = X + 10
X -= 10	X = X - 10
X *= 10	X = X *+ 10
X /= 10	X = X / 10
X %= 10	X = X % 10

പട്ടിക 6.2: C++ ഓപ്പറേറ്ററുകളുടെ മുൻഗണനാക്രമം.

ഓപ്പററ്ററുകളാണ്. C++ ചുരുക്കെഴുത്തുകൾ (short hands) എന്നുകൂടി ഇവ അറിയപ്പെടുന്നു. ഇവയെല്ലാം തന്നെ ബൈനറി ഓപ്പറേറ്ററുകളാണ്. ഇവയുടെ ഒന്നാമത്തെ ഓപ്പറാന്റ് എപ്പോഴും ഒരു വേരിയബിൾ തന്നെയായിരിക്കണം. സങ്കലനം, വില നൽകൽ എന്നീ പ്രവർത്തനങ്ങൾ സാധാരണ രീതിയേക്കാൾ വേഗത്തിൽ ചെയ്യുന്നതിനായി ഈ ഓപ്പറേറ്ററുകൾ ഉപയോഗിക്കുന്നു.

6.6.2 ഇൻക്രിമന്റ് (++) ഡിക്രിമന്റ് (--) ഓപ്പറേറ്ററുകൾ

C++ലെ രണ്ടു പ്രത്യേക ഓപ്പറേറ്ററുകളാണ് ഇൻക്രിമെന്റ്, ഡിക്രിമെന്റ് ഓപ്പറേറ്ററുകൾ. ഇവ യൂനറി ഓപ്പറേറ്ററുകളാണ്. അവയുടെ ഓപ്പറാന്റ് വേരിയബിൾ ആയിരിക്കണം.സോഴ്സ് കോഡ് സംക്ഷിപ്തമാക്കാൻ ഈ ഓപ്പറേറ്ററുകൾ സഹായിക്കും.

ഇൻക്രിമന്റ് ഓപ്പറേറ്റർ: (++) ഒരു ഇന്റിജർ വേരിയബിളിന്റെ ഉള്ളടക്കത്തെ ഒന്നു വർദ്ധിപ്പിക്കാൻ ഈ ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നു. ++x പ്രീഇൻക്രിമെന്റ് (pre increment), x++ പോസ്റ്റ്ഇൻക്രിമെന്റ് (post increment) എന്നിങ്ങനെ ഇതിനെ രണ്ടു രീതിയിൽ എഴുതാം. ഇത് x=x+1 അല്ലെങ്കിൽ x+=1 എന്നതിനു തുല്യമാണ്.

ഡിക്രിമെന്റ് ഓപ്പറേറ്റർ: ഇൻക്രിമെന്റ് ഓപ്പറേറ്ററിന്റെ നേർവിപരീതമായ ഡിക്രിമെന്റ് ഓപ്പറേറ്റർ ഇന്റിജർ വേരിയബിളിന്റെ ഉള്ളടക്കത്തെ കുറക്കുന്നു. ---x, x- എന്നിങ്ങനെ ഈ ഇൻക്രിമെന്റ്/ഡിക്രിമെന്റ് പ്രവർത്തനങ്ങളുടെ ഓപ്പറേറ്ററും രണ്ടു രീതിയിൽ ഉപയോഗിക്കാം. ഇത് x=x-1 അല്ലെങ്കിൽ x-=1 എന്നതിന് തുല്യമാണ്.

ഈ ഓപ്പറേറ്ററുകളുടെ രണ്ട് രീതിയിലുള്ള ഉപയോഗങ്ങളെ ഇൻക്രിമെന്റ്/ഡിക്രിമെന്റ് പ്രവർത്തനങ്ങളുടെ പ്രിഫിക്സ് രൂപമെന്നും, പോസ്റ്റ്ഫിക്സ് രൂപമെന്നും എന്ന് വിളിക്കുന്നു.

രണ്ടു രീതികളും ഓപ്പറാന്റിൽ ഒരേ മാറ്റമാണ് വരുത്തുന്നത് എങ്കിലും മറ്റ് ഓപ്പറേറ്ററുകളുടെ കൂടെ ഉപയോഗിക്കുമ്പോൾ ഇവയുടെ പ്രവർത്തന രീതി വ്യത്യസ്തമായിരിക്കും.

ഇൻക്രിമന്റ്/ ഡിക്രിമന്റ് ഓപ്പറേറ്ററുകളുടെ പ്രീഫിക്സ് രൂപം: പ്രീഫിക്സ് രീതിയിൽ ഓപ്പറേറ്റർ ഓപ്പറാന്റിന്റെ മുൻപായിരിക്കും എഴുതുക. ഇവിടെ ഇൻക്രിമെന്റ് / ഡിക്രിമെന്റ് ആദ്യം ചെയ്യുകയും അങ്ങനെ കിട്ടുന്ന മൂല്യം മറ്റ് ഓപ്പറേഷനുകൾക്ക് ഉപയോഗിക്കുകയും ചെയ്യും. അതുകൊണ്ട് ഈ രീതിയെ മാറ്റുക പിന്നീട് ഉപയോഗിക്കുക (change, then use) എന്ന് വിളിക്കുന്നു.

a, b, c, d എന്നീ വേരിയബിളുകൾ പരിഗണിക്കുക. അവയിൽ a, യുടെ വില 10 ഉം b യുടെ വില 5 ഉം ആണ്. C=++a എന്ന പ്രസ്താവനയിൽ a യുടെ മൂല്യം 11 ഉം c യുടെ മൂല്യം 11 ഉം ആയി ലഭിക്കും. ഇവിടെ ആദ്യം a യുടെ മൂല്യം ഒന്ന് വർദ്ധിച്ച് 11 ആകും. ഈ

കൂടിയ മൂല്യമാണ് cക്ക് നൽകുന്നത്. അതുകൊണ്ടാണ് രണ്ടിനും ഒരേ വില ലഭിക്കുന്നത്. അതുപോലെ തന്നെ d= --b എന്നതിൽ d യുടെയും b യുടെയും മൂല്യം 4 ആകും.

ഇൻക്രിമന്റ്/ഡിക്രിമെന്റ് ഓപ്പറേറ്ററുകളുടെ പോസ്റ്റ് ഫിക്സ് രൂപം: പോസ്റ്റ്ഫിക്സ് രീതിയിൽ ഇൻക്രിമന്റ് /ഡിക്രിമന്റ് ഓപ്പറേഷൻ നടത്തുമ്പോൾ ഓപ്പറേറ്റർ ഓപ്പറാന്റിനു ശേഷമാണ് എഴുതുക. വേരിയബിളിന്റെ നിലവിലുള്ള വിലയാണ് മറ്റ് ഓപ്പറേഷനുകൾക്ക് ഉപയോഗിക്കുക. അതിനുശേഷം മൂല്യം വർദ്ധിപ്പിക്കുകയോ കുറയ്ക്കുകയോ ചെയ്യും. അതിനാൽ ഈ രീതിയെ ഉപയോഗിക്കുക, പിന്നീട് മാറ്റുക (use, then change) എന്നു വിളിക്കുന്നു.

മുകളിൽ കൊടുത്ത ഉദാഹരണം അതേ തുടക്ക വിലകൾ ഉപയോഗിച്ച് നിരീക്ഷിക്കുമ്പോൾ c = a++ എന്ന പ്രയോഗത്തിൽ aയുടെ മൂല്യം 11, c യുടെ മൂല്യം 10 എന്ന് ലഭിക്കും. ഇവിടെ a യുടെ മൂല്യം c ക്ക് നൽകുകയും അതിന് ശേഷം a യുടെ മൂല്യം ഒന്ന് വർദ്ധിപ്പിക്കുകയും ചെയ്യുന്നു. അതായത് a യുടെ മൂല്യം വർദ്ധിപ്പിക്കുന്നതിന് മുൻപുതന്നെ c ക്ക് ആ വില നൽകുന്നു. അതുപോലെ d = b- - എന്ന പ്രവർത്തനത്തിനുശേഷം d യുടെ മൂല്യം 5 ഉം b യുടെ മൂല്യം 4 ഉം ആയിരിക്കും.

6.6.3 ഓപ്പറേറ്ററുകളുടെ മുൻഗണനാക്രമം

പലതരം ഓപ്പറേറ്ററുകൾ ഒരു പ്രയോഗത്തിൽ ഉപയോഗിക്കുമ്പോൾ ഏത് ക്രമത്തിലാണ് ആ ക്രിയകൾ ചെയ്യേണ്ടത് എന്ന് അറിയേണ്ടതുണ്ട്. C++ ൽ അവയുടെ മുൻഗണനാക്രമം എങ്ങനെയാണെന്ന് നോക്കാം. വിവിധ ഓപ്പറേറ്ററുകൾ ഉപയോഗിക്കുന്ന ഒരു പ്രയോഗത്തിൽ ആവരണ ചിഹ്നത്തിനാണ് ആദ്യ പരിഗണന. ( ) ആവരണ ചിഹ്നം ഇല്ലെങ്കിൽ മുൻനിശ്ചയിച്ചപ്രകാരമുള്ള ഒരു മുൻഗണനാക്രമത്തിലാണ് അവ വിലയിരുത്തപ്പെടുക. ഓപ്പറേറ്ററുകളുടെ ഈ മുൻഗണനാക്രമം പട്ടിക 6.3 ൽ നൽകിയിരിക്കുന്നു. ഒരു പ്രയോഗത്തിൽ മുൻഗണനാക്രമത്തിൽ ഒരേ സ്ഥാനം വരുന്ന ഓപ്പറേറ്ററുകൾ ഉണ്ടെങ്കിൽ അവ മിക്കവാറും ഇടത്തുനിന്ന് വലത്തേക്ക് എന്ന രീതിയിലാണ് പ്രവർത്തിക്കുക.

a=3, b=5, c=4, d=2, x എന്നീ വേരിയബിളുകളും അവയുടെ വിലകളും പരിഗണിക്കുക.

x=a+b*c-d എന്ന പ്രയോഗം ചെയ്തു കഴിയുമ്പോൾ x ന്റെ വില 21 ആയിരിക്കും. ഇവിടെ * (ഗുണനത്തി)ന് +(സങ്കലനം), -(വ്യവകലനം) എന്നിവയേക്കാൾ മുൻഗണനയുള്ളതിനാൽ b, c എന്നീ വേരിയബിളുകൾ തമ്മിൽ ഗുണിച്ചശേഷമേ അതിന്റെ ഫലം a യോടൊപ്പം കൂട്ടിചേർക്കൂ. ആ കിട്ടുന്ന ഉത്തരത്തിൽ നിന്നും d കുറച്ചാൽ അന്തിമഫലം ലഭ്യമാകും. അങ്ങനെ കിട്ടുന്ന ഉത്തരം xന് നൽകുന്നു. പ്രോഗ്രാമറുടെ ആവശ്യാനുസരണം പ്രയോഗത്തിലെ ഓപ്പറേറ്ററുകളുടെ മുൻഗണനാക്രമം മാറ്റുന്നതിനായി ( ) (ആവരണചിഹ്നം) ഉപയോഗിച്ചാൽ മതിയാകും. ഉദാഹരണത്തിന് a=5, b=4, c=3, d=2 എന്നായാൽ a+b-c*d എന്നതിന്റെ ഉത്തരം 3 ആയിരിക്കും. പ്രോഗ്രാമർക്ക് ആദ്യം വ്യവകലനം, പിന്നീട് സങ്കലനം, ഗുണനം എന്ന ക്രമത്തിൽ ചെയ്യണമെങ്കിൽ അതിനായി ശരിയായ രീതിയിൽ ആവരണ ചിഹ്നങ്ങൾ ഉപയോഗിച്ച് (a+(b-c))*d എന്ന് എഴുതണം. ഇതിന്റെ ഉത്തരം 12 ആയിരിക്കും. മുൻഗണനാക്രമം മാറ്റുതിനായി [], {} ഇത്തരം ആവരണചിഹ്നങ്ങൾ ഉപയോഗിക്കാൻ സാധിക്കില്ല.

മുൻഗണന	പ്രവർത്തനങ്ങൾ
1	( ) ആവരണം
2	++, --, ! , യൂനറി + , യൂനറി -, sizeof
3	* (ഹരണം), / (ഗുണനം), % (മോഡുലസ്)
4	+ (സങ്കലനം), - (വ്യവകലനം)
5	< (ചെറുതാണ്), <= (ചെറുതോ തുല്യമോ ആണ്), > (വലുതാണ്), >= (വലുതോ തുല്യമോ ആണ്)
6	== (തുല്യമാണ്), != (തുല്യമല്ല)
7	&& (ലോജിക്കൽ AND)
8	|| (ലോജിക്കൽ OR)
9	? : (കണ്ടീഷണൽ പ്രയോഗം)
10	= (അസൈൻമെന്റ് ഓപ്പറേറ്റർ), *=, /=, %=, +=, -= (അരിത്മെറ്റിക് അസൈൻമെന്റ് ഓപ്പറേറ്ററുകൾ)
11	, (അല്പവിരാമം)

പട്ടിക 6.3: ഓപ്പറേറ്ററുകളുടെ മുൻഗണനാക്രമം.

6.7 ഇനം മാറ്റൽ (Type conversion)

പൂർണ്ണസംഖ്യ പദപ്രയോഗം, ദശാംശസംഖ്യാ പദപ്രയോഗം എന്നിങ്ങനെ രണ്ട്തരം അരിത്മെറ്റിക് പദപ്രയോഗങ്ങൾ ഉണ്ടെന്ന് നാം മുമ്പ് ചർച്ച ചെയ്തുവല്ലോ. ഇവ രണ്ടിലും അരിത്മാറ്റിക് ഓപ്പറേഷനിൽ ഉൾപ്പെട്ടിരിക്കുന്ന ഓപ്പറാന്റുകൾ ഒരേ ഡാറ്റാ ഇനത്തിലുള്ളവയാണ്. എന്നാൽ വ്യത്യസ്ത ഇനം സംഖ്യകൾ ഉപയോഗിക്കേണ്ട സാഹചര്യങ്ങളും ഉണ്ടാകാം. ഉദാഹരണമായി C++ ൽ പൂർണ്ണസംഖ്യാ പദപ്രയോഗം 5/2, എന്നത് 2 എന്ന ഫലം തരുമ്പോൾ ദശാംശസംഖ്യാ പദപ്രയോഗമായ 5.0/2.0 എന്നത് 2.5 എന്ന ഫലം തരുന്നു. പക്ഷെ 5/2.0, അല്ലെങ്കിൽ 5.0/2 എന്നിവയുടെ ഉത്തരം എന്തായിരിക്കും? ഇനം മാറ്റൽ രീതിയാണ് ഈ സാഹചര്യത്തിൽ ഉപയോഗിക്കേണ്ടി വരുക. ഒരു ഓപ്പറാന്റിന്റെ ഡാറ്റാ ഇനം മറ്റൊന്നിലേക്ക് മാറ്റുകയാണ് ചെയ്യേണ്ടത്. ഇതിനെ ഇനം മാറ്റൽ എന്ന് പറയാം. ഇത് ആന്തരിക ഇനം മാറ്റൽ, ബാഹ്യഇനം മാറ്റൽ എന്നിങ്ങനെ രണ്ടു രീതിയിൽ ചെയ്യാം.

6.7.1 ആന്തരിക ഇനം മാറ്റൽ (implicit type conversion/ type promotion):

ആന്തരിക ഇനം മാറ്റൽ C++ കമ്പൈലർ ആന്തരികമായി ചെയ്യുന്നതാണ്. വ്യത്യസ്ഥതരം ഡാറ്റ ഉള്ള ഒരു പദപ്രയോഗത്തിൽ C++ കുറഞ്ഞ വലിപ്പത്തിലുള്ള ഓപ്പറാന്റിനെ കൂടുതൽ വലുപ്പമുള്ളതിന്റെ ഡാറ്റാ ഇനമാക്കി മാറ്റുന്നു. അതായത് എല്ലായ്പ്പോഴും ചെറിയതിനെ വലുതാക്കുക മാത്രമാണ് ചെയ്യുന്നത്. ആയതിനാൽ ഇതിനെ ടൈപ്പ് പ്രമോഷൻ എന്നും പറയുന്നു. ഡാറ്റ ഇനങ്ങൾ വലിപ്പത്തിന്റെ അവരോഹണ ക്രമത്തിൽ താഴെ പറയുന്ന രീതിയിലായിരിക്കും.

long double, double, float, unsigned long, long int, unsigned int short in

ഫലത്തിന്റെ ഡാറ്റാ ഇനം വലിയ ഓപ്പറാന്റിന്റെ ഡാറ്റാ ഇനമായിരിക്കും. ഉദാഹരണമായി 5/2*3+2.5 എന്ന പ്രയോഗത്തിന്റെ ഫലം 8.5 ആണ്. ഇത് എങ്ങിനെ ലഭിക്കുന്നു എന്ന് നോക്കാം.

ഘട്ടം 1: 5/2 → 2 പൂർണ്ണസംഖ്യയുടെ ഹരണം

ഘട്ടം 2: 2*3 → 6 പൂർണ്ണസംഖ്യയുടെ ഗുണനം

ഘട്ടം 3: 6+2.5→8.5 (ദശാംശസംഖ്യാ സങ്കലനം ഇവിടെ 6 നെ 6.0 ആക്കി മാറ്റുന്നു).

6.7.2: ബാഹ്യഇനംമാറ്റൽ (explicit type conversion type casting):

ആന്തരിക ഡാറ്റാ ഇനം മാറ്റലിൽ നിന്നും വ്യത്യസ്തമായി ചില സാഹചര്യങ്ങളിൽ ചില പ്രോഗ്രാമർ തന്നെ ഫലത്തിന്റെ ഡാറ്റ ഇനം തീരുമാനിക്കേണ്ടതായി വരും. അങ്ങനെ വരുമ്പോൾ പ്രോഗ്രാമർ തന്നെ ഡാറ്റ ഇനം ആവരണ ചിഹ്നത്തിൽ ഓപ്പറാന്റിന്റെ ഇടതു വശത്തായി നൽകണം. ഇത് ഇനം മാറ്റാൻ ഉപകരിക്കുന്നു. ഈ രീതിയിൽ പ്രോഗ്രാമർ തന്നെ ആവശ്യമായ ഇനത്തിലേക്ക് ഡാറ്റയെ ഇനം മാറ്റുന്നതിനെ ബാഹ്യഇനം മാറ്റൽ അഥവാ ടൈപ്പ് കാസ്റ്റിംഗ് എന്നു പറയുന്നു. സാധാരണയായി പദപ്രയോഗത്തിലെ വേരിയബിളുകളുടെ ഇനം മാറ്റത്തിനാണ് ഇത് ഉപയോഗിക്കുന്നത്. കൂടുതൽ ഉദാഹരണങ്ങൾ ഭാഗം 6.9.2 ൽ ചർച്ച ചെയ്യാം.

ഇനം അനുയോജ്യപ്പെടുത്തൽ

ഒരു വില നൽകൽ പ്രസ്താവന നടപ്പിലാക്കുമ്പോൾ RHS പ്രയോഗത്തിന്റെ ഡാറ്റ ഇനം LHS വേരിയബിളിൽ നിന്നും വ്യത്യസ്തമാകാം, അതിന് രണ്ട് സാധ്യതകളുണ്ട്. LHS-ലുള്ള വേരിയബിളിന്റെ ഡാറ്റ ഇനത്തിന്റെ വലിപ്പം RHS ലുള്ള വേരിയബിൾ അല്ലെങ്കിൽ പദപ്രയോഗത്തിലേതിനേക്കാളും കൂടുതലാകാം. ഈ സാഹചര്യത്തിൽ, RHS ലെ മൂല്യത്തിന്റെ ഡാറ്റ ഇനം LHSലെ വേരിയബിളിലേക്ക് ഉയർത്തപ്പെടുന്നതാണ് (ടൈപ്പ് പ്രൊമോഷൻ). താഴെയുള്ള കോഡ് ശകലം പരിഗണിക്കുക:

```
int a=5, b=2;
float p, q;
p = b;
q = a / p;
```

ഇവിടെ b യുടെ ഡാറ്റ ഇനം ടൈപ്പ് പ്രൊമോഷനിലൂടെ float ആക്കി മാറ്റുകയും 2.0 എന്നത് p യിൽ സൂക്ഷിക്കുകയും ചെയ്യുന്നു. q=a/p എന്ന പ്രസ്താവനയിൽ a യുടെ ഡാറ്റാഇനം ടൈപ്പ് പ്രൊമോഷൻ ചെയ്തുകഴിയുമ്പോൾ ഉത്തരം 2.5 എന്ന് ലഭിക്കുകയും q യിൽ സൂക്ഷിയ്ക്കപ്പെടുകയും ചെയ്യും.

LHS ന്റെ ഡാറ്റാഇനത്തിന്റെ വലിപ്പം RHS ന്റേതിനേക്കാൾ കുറവായിരിക്കാം എന്നതാണ് രണ്ടാമത്തെ സാധ്യത. RHS ന്റെ വില (truncate) ചുരുക്കി LHS നു അനുയയോജ്യമാക്കുന്നു. താഴെയുള്ള കോഡ് ഇത് വിശദീകരിക്കുന്നു.

```
float a=2.6;
int p, q;
p = a;
q = a * 4;
```

ഇവിടെ p യുടെ വില 2 ഉം q എന്നത് 10 ആകുന്നു. a*4 എന്ന പദപ്രയോഗത്തിന്റെ വില കാണുമ്പോൾ 10.4 എന്ന് ലഭിക്കുന്നു. എന്നാൽ q int ഇനത്തിൽപ്പെട്ടതിനാൽ 10 മാത്രമേ അതിൽ സൂക്ഷിക്കൂ. ഓപ്പറേറ്ററുകളുടെ ഡാറ്റ ഇനങ്ങളിൽ ചേർച്ചയില്ലാതെ വരുമ്പോൾ ആഗ്രഹിക്കുന്ന ഫലം ലഭിക്കുന്നതിനായി പ്രോഗ്രാമർക്ക് ബാഹ്യഇനം മാറ്റൽ രീതി പ്രയോഗിക്കാവുനനതാണ്.

താഴെക്കൊടുത്തിരിക്കുന്ന പ്രോഗ്രാം ശകലം ശ്രദ്ധിക്കുക.

```
int p=5, q=2;
float x, y;
x = p/q;
y = (x+p)/q;
```

മുകളിലെ കോഡ് ശകലം പ്രവർത്തിച്ചു കഴിയുമ്പോൾ x ന്റെ വില 2.0 , y യുടെ വില 3.5 ആയിരിക്കും. p/q എന്ന പദപ്രയോഗം ഒരു പൂർണ്ണസംഖ്യാ പദപ്രയോഗം ആയതിനാൽ അതിന്റെ ഫലമായി 2 ലഭിക്കുകയും, ഫ്ളോട്ടിംഗ് പോയിന്റ് വിലയായി x ൽ സംഭരിക്കപ്പെടുകയും ചെയ്യുന്നു. X+p ബ്രാക്കറ്റിനകത്ത് ആയതിനാൽ മുൻഗണന ലഭിക്കുന്നു. x ന്റെ ഇനം float ആയതിനാൽ p ടൈപ്പ് പ്രമോഷൻ ചെയ്യുകയും ഫലം 7.0 എന്ന് ലഭിക്കുകയും ചെയ്യും. പിന്നീട് 7.0 ഹരണക്രിയയുടെ ഒന്നാമത്തെ ഓപ്പെറന്റ് ആക്കി ഹരണക്രിയ നടത്തുന്നു q നെ float ആക്കി മാറ്റി ഫലം 3.5 എന്ന് ലഭിക്കുകയും ചെയ്യുന്നു. നമുക്ക് p /q ന്റെ ഫലം ദശാംശസംഖ്യയായി x ൽ സംഭരിക്കുന്നതിന് ടൈപ്പ് കാസ്റ്റിങ് ഉപയോഗിച്ച് താഴെപ്പറയുന്ന രീതിയിൽ പ്രസ്താവന മാറ്റി എഴുതാവുന്നതാണ്.

x=(float)p/q; or x=p/(float)q;

പ്രോഗ്രാമുകൾ

പ്രോഗ്രാം 6.1 ഒരു സന്ദേശം കാണിക്കുന്നതിന്

```
   /* This program displays the message
       "Smoking is injurious to health"
       on the monitor */
#include <iostreamh>   // To use the cout object
using namespace std;  // To access cout
int main() //program begins here
{     //The following output statement displays a message
      cout << "Smoking is injurious to health";
      return 0;
}     //end of the program
```

(Multiline comment — Single line comment — ഇന്റന്റേഷൻ)

ഔട്ട്പുട്ട് :

Smoking is injurious to health

ഇന്റന്റേഷൻ ഉപയോഗിച്ചുള്ള കൂടുതൽ ഉദാഹരണങ്ങൾ അധ്യായം 7 ൽ നൽകിയിരിക്കുന്നു.പ്രോഗ്രാം 6.2 രണ്ട് സംഖ്യകൾ ഇൻപുട്ടായി സ്വീകരിച്ച് അവയുടെ തുക പ്രദർശിപ്പിക്കുന്നു. ഈ പ്രോഗ്രാമിൽ അധ്യായം അഞ്ചിൽ പഠിച്ച പ്രസ്താവനകൾ ഉപയോഗിച്ചിരിക്കുന്നു.

നമുക്കു ചെയ്യാം

5.2, 2.7 തുടങ്ങിയ വ്യത്യസ്ത ഇൻപുട്ടുകൾ ഉപയോഗിച്ച് പ്രോഗ്രാം 6.2 റൺ ചെയ്യുക. ശരിയായ ഉത്തരം ലഭിക്കുന്നുണ്ടോ എന്ന് പരിശോധിക്കുക. ഇല്ലെങ്കിൽ കൂട്ടുകാരുമായി ചർച്ച ചെയ്ത് പ്രശ്ന പരിഹാരം കണ്ടെത്തുക.

പ്രോഗ്രാം 6.2: രണ്ട് പൂർണ്ണസംഖ്യകളുടെ തുക കണ്ടുപിടിക്കാൻ

```
#include <iostream>
using namespace std;
int main()
{  //Program begins
/* Two variables are declared to read user inputs and the
variable sum is declared to store the result
*/
   int num1, num2, sum;
   cout<<"Enter two numbers: "; //Prompt for input
   cin>>num1>>num2;    //Cascading to get two numbers
   sum=num1+num2;   //Assignment statement to find the sum
   cout<<"Sum of the entered numbers = "<<sum;
/* The result is displayed with proper message.
   Cascading of output operator is utilized      */
   return 0;
}
```

ഔട്ട്പുട്ട് :

Enter two numbers: 5 7

Sum of the entered numbers = 12

യൂസർ നൽകുന്ന ഡാറ്റ സ്പേയ്സ് ഉപയോഗിച്ച് വേർതിരിച്ചിരിക്കുന്നു

നമുക്ക് മറ്റൊരു പ്രശ്നം പരിഗണിക്കാം. ഒരു കുട്ടിയ്ക്ക് മൂന്ന് തുടർമൂല്യ പ്രവർത്തനങ്ങളിൽ ലഭിച്ച സ്കോറുകൾ നൽകിയിരിക്കുന്നു. കൂടിയ സ്കോർ 20 ആണ്. ശരാശരി സ്കോർ കണ്ടുപിടിക്കുക.

പ്രോഗ്രാം 6.3 : മൂന്നു CE സ്കോറുകളുടെ ശരാശരി കണ്ടെത്താൻ

```
#include <iostream>
using namespace std;
int main()
{
    int score_1, score_2, score_3;
```

```
    float avg;
    //Average of 3 numbers can be a floating point value
    cout << "Enter the three CE scores: ";
    cin >> score_1 >> score_2 >> score_3;
    avg = (score_1 + score_2 + score_3) / 3.0;
/* The result of addition will be an integer value. If 3
is written instead of 3.0, integer division will be
performed and will not get the correct result  */
    cout << "Average CE score is: " << avg;
    return 0;
}
```

ഔട്ട്പുട്ട് :

Enter the three CE scores: 17 19 20

Average CE score is: 18.666666

ശരാശരി കാണാനുള്ള എക്സ്പ്രഷൻ അസൈമെന്റ് ഓപ്പറേറ്ററി (=) ന്റെ വലതു ഭാഗത്തു നൽകിയിരിക്കുന്നു. ഈ എക്സ്പ്രഷനിൽ രണ്ട് + ഓപ്പറേറ്ററുകളും ഒരു / ഉണ്ട്. / ഓപ്പറേറ്റർ + ഓപ്പറേറ്ററിനേക്കാൾ മുൻഗണന ഉള്ളതായതിനാൽ തുക കാണുന്ന പ്രവർത്തനം ബ്രാക്കറ്റിനുള്ളിൽ നൽകിയിരിക്കുന്നു. തുക കാണുന്നതിനുള്ള ഓപ്പറന്റുകളെല്ലാം int ഡാറ്റാ തരത്തിലുള്ളതായതിനാൽ ഫലവും ഇന്റിജർ തന്നെയായിരിക്കും. ഈ ഇന്റിജറിനെ മൂന്ന് കൊണ്ട് ഹരിക്കുമ്പോൾ ഫലം വീണ്ടും ഇന്റജർ തന്നെയായിരിക്കും. അതുകൊണ്ട് പ്രോഗ്രാം 6.3 ന്റെ ഔട്ട്പുട്ട് 18 ആയിരിക്കും. ഇത് കൃത്യമല്ല. ആയതിനാൽ ഹരിക്കാൻ വേണ്ടി ഫ്ളോട്ടിംഗ് പോയിന്റ് സംഖ്യയായ 3.0 ഉപയോഗിച്ചിരിക്കുന്നു. അങ്ങനെ ഇന്റിജർ ഫ്ളോട്ടിംഗ് തരമായി ഉയർത്തപ്പെടുകയും കൃത്യമായ ഉത്തരം ലഭിക്കുകയും ചെയ്യുന്നു. π

ഒരു വൃത്തത്തിന്റെ ആരം നൽകിയെന്നിരിക്കട്ടെ. എങ്ങനെയാണ് വിസ്തീർണ്ണവും ചുറ്റളവും കണ്ടുപിടിക്കുന്നത്? വിസ്തീർണ്ണം കാണാനുള്ള സൂത്രവാക്യം r^2 ഉം ചുറ്റളവ് കാണാനുള്ള സൂത്രവാക്യം 2 r ആണല്ലോ (=3.14). പ്രോഗ്രാം 6.4 ഈ പ്രശ്നം നിർദ്ധാരണം ചെയ്യുന്നു.

പ്രോഗ്രാം 6.4 : ഒരു വൃത്തത്തിന്റെ വിസ്തീർണവും ചുറ്റളവും കണ്ടെത്താൻ

```
#include <iostream>
using namespace std;
int main()
{
    const float PI = 22.0/7; //Use of const access modifier
    float radius, area, perimeter;
    cout<<"Enter the radius of the circle: ";
```

```
    cin>>radius;
    area = PI * radius * radius;
    perimeter = 2 * PI * radius;
    cout<<"Area of the circle = "<<area<<"\n";
    cout<<"Perimeter of the circle = "<<perimeter;
    return 0;
}
```

Escape sequence '\n' prints a new line after displaying the value of Area

ഔട്ട്പുട്ട് :

```
Enter the radius of the circle: 2.5
Area of the circle = 19.642857
Perimeter of the circle = 15.714285
```

പ്രോഗ്രാം 6.4 ന്റെ അവസാനത്തെ രണ്ട് ഔട്ട്പുട്ടുകൾ വ്യത്യസ്ത ലൈനുകളിൽ പ്രദർശിപ്പിക്കുന്നു. ഇതിനു കാരണം എസ്കേപ്പ് സിക്വൻസ് ക്യാരക്ടറായ '\n' ഉപയോഗിച്ചതാണ്.

അടുത്തതായി നമുക്ക് സാധാരണ പലിശ കാണുന്നതിനുള്ള ഒരു പ്രോഗ്രാം തയ്യാറാക്കാം. പ്രിൻസിപ്പൽ എമൗണ്ട്, റെയ്റ്റ് ഓഫ് ഇൻട്രസ്റ്റ്, പീരിഡ് എന്നിവ ഇതിലെ ഇൻപുട്ടുകളാണെന്ന് നിങ്ങൾക്കറിയാമല്ലോ.

പ്രോഗ്രാം 6.5 : സാധാരണ പലിശ കണ്ടെത്താൻ

```
#include <iostream>
using namespace std;
int main()
{
    float p_Amount, n_Year, i_Rate, int_Amount;
    cout<<"Enter the principal amount in Rupees: ";
    cin>>p_Amount;
    cout<<"Enter the number of years for the deposit: ";
    cin>>n_Year;
    cout<<"Enter the rate of interest in percentage: ";
    cin>>i_Rate;
    int_Amount = p_Amount * n_Year * i_Rate /100;
    cout << "Simple interest for the principal amount "
        <<p_Amount<<" Rupees for a period of "<<n_Year
        <<" years at the rate of interest "<<i_rate
        <<" is "<<int_Amount<<" Rupees";
    return 0;
}
```

ഔട്ട്പുട്ട് :

```
Enter the principal amount in Rupees: 100
Enter the number of years for the deposit: 2
Enter the rate of interest in percentage: 10
```

പ്രോഗ്രാം 6.5 ലെ അവസാനത്തെ പ്രസ്താവന അഞ്ച് വരികളിലേക്ക് പടർന്നിരിക്കുന്നു. ഇത് ഒരു പ്രസ്താവനയാണ്. ഇതിനിടയിൽ സെമികോളൻ ഉപയോഗിച്ചിട്ടില്ലായെന്ന് ശ്രദ്ധിക്കുക. പ്രോഗ്രാം പ്രവർത്തിക്കുമ്പോൾ ഫലം മോണിട്ടറിന്റെ റെസലൂഷൻ അനുസരിച്ച് രണ്ട് വ്യത്യസ്ത വരികളിൽ പ്രദർശിപ്പിക്കപ്പെടാൻ സാധ്യതയുണ്ട്.

പ്രോഗ്രാം 6.6 ഒരു കൺവേർഷൻ പ്രശ്നമാണ്. ഇവിടെ ഡിഗ്രി സെൽഷ്യസിലുള്ള താപം ഫാരൻ ഹീറ്റിലേക്ക് മാറ്റുന്നു.

പ്രോഗ്രാം 6.6 : സെൽഷ്യസിലുള്ള താപം ഫാരൻ ഹീറ്റിലേക്ക് മാറ്റാൻ

```
#include <iostream>
using namespace std;
int main()
{
    float celsius, fahrenheit;
    cout<<"Enter the Temperature in Celsius: ";
    cin>>celsius;
    fahrenheit=1.8*celsius+32;
    cout<< celsius<<" Degree Celsius = "
        << fahrenheit<<" Degree Fahrenheit";
    return 0;
}
```

ഔട്ട്പുട്ട് :

```
Enter the Temperature in Celsius: 37
37 Degree Celsius = 98.599998 Degree Fahrenheit
```

നമുക്കറിയാം C++ ലെ ഓരോ ക്യാരക്ടറിനും തത്തുല്യമായ ASCII കോഡ് ഉണ്ട്. ഇത് ഇന്റിജർ കോഡാണ്. തന്നിരിക്കുന്ന ക്യാരക്ടറിന്റെ ASCII കോഡ് കണ്ടുപിടിക്കാൻ നമുക്ക് ഒരു പ്രോഗ്രാം എഴുതാം.

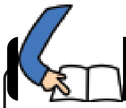
പ്രോഗ്രാം 6.7 : ക്യാരക്ടറിന്റെ ASCII കോഡ് കണ്ടെത്താൻ

```
#include <iostream>
using namespace std;
int main()
{
      char ch;
```

```
      int asc;
      cout << "Enter the character: ";
      cin >> ch;
      asc = ch;
      cout << "ASCII value of "<<ch<<"  = " << asc;
      return 0;
}
```

ഔട്ട്പുട്ട് :

```
Enter the character: A
ASCII value of A = 65
```



നമുക്ക് സംഗ്രഹിക്കാം

C++ പ്രോഗ്രാമിൽ കൃത്യമായ ഒരു ഘടനയുണ്ട്. പ്രോഗ്രാം ആകർഷകമാക്കാനും ആശയവിനിമയം ഉറപ്പു വരുത്താനും കൃത്യമായ മാർഗ്ഗ നിർദ്ദേശങ്ങൾ പാലിക്കപ്പെടേണ്ടതായിട്ടുണ്ട്. വേരിയബിളുകൾ പ്രഖ്യാപിക്കുമ്പോൾ തന്നെ അതിനു പ്രാരംഭ വില നൽകേതാണ്ടിയിട്ടുണ്ട്. 'const' ആക്സസ് മോഡിഫയർ ഉപയോഗിച്ചാൽ വേരിയബിളിന്റെ വാല്യു നമ്മൾക്ക് പ്രോഗ്രാമിലൂടെ മാറ്റാൻ സാധിക്കില്ല. കൂടുതൽ റെയ്ഞ്ചിലുള്ള ഡാറ്റ കൈകാര്യം ചെയ്യുമ്പോൾ ടൈപ്പ് മോഡിഫയറുകൾ സഹായകമാകുന്നു. ടൈപ്പ് മോഡിഫയറുകൾ പ്രഖ്യാപന സമയത്ത് ഡാറ്റ ഇനത്തിന്റെ കൂടെ ഉപയോഗിക്കുന്നു. C++ ൽ അരിത്തമെറ്റിക് ഓപ്പറേറ്റർ, അസൈൻമെന്റ് ഓപ്പറേറ്റർ എന്നിവ ഒന്നിച്ചുപയോഗിക്കാനുള്ള സൗകര്യം ഉണ്ട്. അരിത്തമെറ്റിക് ഓപ്പറേഷനുകളിൽ നിന്നും ഉദ്ദേശിച്ച ഫലം ലഭിക്കുന്നതിന് ഇനം മാറ്റൽ രീതികളും C++ ൽ ഉപയോഗിക്കപ്പെടുന്നു.

പഠന നേട്ടങ്ങൾ

ഈ അധ്യായത്തിൽ പൂർത്തികരണത്തോടെ പഠിതാവിന്

- ഡാറ്റ ടൈപ്പ് മോഡിഫയറുകൾ തിരഞ്ഞെടുക്കാൻ സാധിക്കുന്നു.
- വിവിധ ഓപ്പറേറ്ററുകൾ ഉപയോഗിക്കാൻ കഴിയുന്നു.
- I/O ഓപ്പറേഷനുകൾ നടത്താൻ കഴിയുന്നു.
- C++ പ്രോഗ്രാമിന്റെ ഘടന തിരിച്ചറിയുന്നു.
- പ്രോഗ്രാം എഴുതുമ്പോൾ ശ്രദ്ധിക്കേണ്ട മാർഗ്ഗ നിർദ്ദേശങ്ങൾ ലഭിക്കുന്നു.
- C++ IDE ഉപയോഗിച്ച് ചെറിയ പ്രോഗ്രാമുകൾ ചെയ്യാൻ സാധിക്കുന്നു.

ലാബ് പ്രവർത്തനങ്ങൾ

1. ഉപയോക്താവിൽ നിന്ന് ഭാരം ഗ്രാമിൽ സ്വീകരിച്ച് കിലോഗ്രാമിൽ പ്രദർശിപ്പിക്കുന്നതിനുള്ള C++ പ്രോഗ്രാം എഴുതുക.
2. താഴെ കൊടുത്തിരിക്കുന്ന പട്ടിക പ്രദർശിപ്പിക്കുന്നതിനുള്ള പ്രോഗ്രാം എഴുതുക.

2013	100%
2012	99.9%
2011	95.5%
2010	90.81%
2009	85%

(സൂചന : '\n', '\t' എന്നിവ ഉപയോഗിച്ച് ഒറ്റ ഔട്ട് പ്രസ്താവനയിൽ ഉത്തരമെഴുതുക.)

3. നിങ്ങളുടെ ഉയരം മീറ്ററിലും സെന്റീമീറ്ററിലുമായി സ്വീകരിച്ച് അടിയിലും ഇഞ്ചിലുമായി മാറ്റുന്നതിനാവശ്യമായ C++ പ്രോഗ്രാം എഴുതുക.

   (1 അടി = 12 ഇഞ്ച്, 1 ഇഞ്ച് = 2.54 സെ.മീ.)
4. ത്രികോണത്തിന്റെ വിസ്തീർണം കാണുവാനുള്ള പ്രോഗ്രാമെഴുതുക.
5. സാധാരണ പലിശ, കൂട്ടുപലിശ എന്നിവ കാണുവാനുള്ള പ്രോഗ്രാം എഴുതുക.
6. തന്നിരിക്കുന്ന ഡിജിന്റെ ASCII കോഡും, ബാക്ക് സ്പേസിന്റെ ASCII കോഡും പ്രദർശിപ്പിക്കുവാനുള്ള പ്രോഗ്രാം എഴുതുക.

   (സൂചന: ബാക്ക്സ്പേസിനെ ('\b) ഒരു int വേരിയബിളിൽ സംഭരിക്കുക.)
7. സമയം സെക്കന്റിൽ സ്വീകരിച്ച് hrs:mins:secs എന്ന മാതൃകയിൽ പ്രദർശിപ്പിക്കുന്നതിനുള്ള പ്രോഗ്രാം എഴുതുക.

   ഉദാഹരണത്തിന് സമയം 3700 സെക്കന്റുകൾ ആണെങ്കിൽ ഉത്തരം 1hr:1min:40secs എന്നായിരിക്കും.

മാതൃക ചോദ്യങ്ങൾ

ഹ്രസ്വോത്തര ചോദ്യങ്ങൾ

1. വേരിയബിളിന്റെ ഡയനാമിക് ഇനിഷ്യലൈസഷൻ എന്നാൽ എന്ത്?
2. ടൈപ്പ് പ്രമോഷൻ എന്നാൽ എന്ത്?
3. ടൈപ്പ് കാസ്റ്റിങ്ങ് എന്നാൽ എന്ത്?

ലഘു ഉപന്യാസ ചോദ്യങ്ങൾ

1. ഏതൊക്കെ തരത്തിലുള്ള വേരിയബിൾ പ്രഖ്യാപനങ്ങളാണ് താഴെ തന്നിരിക്കുന്ന പ്രോഗ്രാം ശകലത്തിലുള്ളത്?

```
{ int area, length = 10, wideth = 12, perimeter;
  area = length * width;
  perimeter = 2 * (length + width)
}
```

2. 'area', 'perimeter' എന്നീ വേരിയബിളുകളെ ഡയനാമിക് ഇനിഷ്യലൈസേഷൻ നടത്തുന്ന രീതിയിൽ മുകളിലുള്ള പ്രോഗ്രാം ശകലം പരിഷ്കരിക്കുക.
3. C++ ലെ ടൈപ്പ് മോഡിഫയർ വിശദീകരിക്കുക.
4. എന്തുകൊണ്ടാണ് C++ ൽ ധാരാളം ഡാറ്റ ഇനങ്ങൾ ലഭ്യമായിട്ടുള്ളത്.
5. 'const' എന്ന കീവേർഡിന്റെ ധർമ്മം എന്ത്?
6. പ്രീഇൻക്രിമെന്റ്, പോസ്റ്റ് ഇൻക്രിമെന്റ് എന്നീ ഓപ്പറേഷനുകൾ വിവരിക്കുക.
7. ഇനം മാറ്റൽ രീതികൾ വിവരിക്കുക.
8. iostream എന്ന ഫയൽ പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തിയിട്ടില്ലെങ്കിൽ എന്ത് സംഭവിക്കും?
9. main( ) ഒരു പ്രോഗ്രാമിൽ ഇല്ലെങ്കിൽ എന്ത് സംഭവിക്കും?
10. താഴെ കൊടുത്ത പ്രോഗ്രാം ശകലത്തിലെ തെറ്റുകൾ കണ്ടെത്തുക.

(i)
```
int main() { cout << "Enter two numbers"
cin >> num >> auto
float area = Length * breadth ; }
```

(ii)
```
#include <iostream>
using namespace;
void Main()
{ int a, b
cin <<a <<b
max=a % b
cout > max
}
```

11. പ്രോഗ്രാമിൽ കമന്റുകളുടെ ഉപയോഗം എന്ത്?

ഉപന്യാസ ചോദ്യങ്ങൾ

1. C++ ലെ വിവിധതരം പ്രയോഗങ്ങൾ വിവരിക്കുക.
2. അസൈൻമെന്റ് ഓപ്പറേറ്ററിന്റെ പ്രവർത്തനം എഴുതുക. അരിത്തമെറ്റിക് അസൈൻമെന്റ് ഓപ്പറേറ്ററുകൾ ഉദാഹരണസഹിതം വിശദമാക്കുക.

7

പ്രധാന ആശയങ്ങൾ

- തീരുമാനം എടുക്കുന്നതിനുള്ള പ്രസ്താവനകൾ
  - `if` പ്രസ്താവന
  - `if .. else` പ്രസ്താവന
  - നെസ്റ്റഡ് `if`
  - `else if` ഗോവണി
  - `switch` പ്രസ്താവന
  - കൺഡീഷണൽ ഓപ്പറേറ്റർ
- ആവർത്തന പ്രസ്താവനകൾ
  - `while` പ്രസ്താവന
  - `for` പ്രസ്താവന
  - `do .. while` പ്രസ്താവന

R4U8X7

നിയന്ത്രണ പ്രസ്താവനകൾ

ഇൻപുട്ട്, ഔട്ട്പുട്ട്, വില നൽകൽ എന്നിവ ചെയ്യുന്നതിനുള്ള C++-ലെ നിർവഹണ പ്രസ്താവനകൾ കഴിഞ്ഞ അധ്യായങ്ങളിൽ നാം ചർച്ച ചെയ്തു. ഇതുപയോഗിച്ച് ലളിതമായ പ്രോഗ്രാമുകൾ എങ്ങനെ എഴുതുവാൻ കഴിയുമെന്ന് നമുക്കറിയാം. ഈ പ്രോഗ്രാമുകളുടെ നിർവഹണം അനുക്രമമാണ്. അതായത്, പ്രോഗ്രാമിലെ ഓരോ നിർദ്ദേശവും ഒന്നിന് പുറകെ ഒന്നായി പ്രവർത്തിക്കുന്നു. ഈ അധ്യായത്തിൽ C++-ലെ പ്രോഗ്രാമിന്റെ തനത് പ്രവർത്തനക്രമത്തിന് മാറ്റം വരുത്തുന്ന പ്രസ്താവനകളെ കുറിച്ചാണ് നാം ചർച്ച ചെയ്യുന്നത്. അധ്യായം 3 ൽ നാം ചർച്ച ചെയ്ത തെരഞ്ഞെടുക്കൽ, ആവർത്തിക്കൽ, നീക്കംചെയ്യൽ എന്നീ പ്രസ്താവനകൾ പ്രശ്നങ്ങൾ നിർദ്ധാരണം ചെയ്യാൻ ആവശ്യമായേക്കാം. സാധാരണയായി ഇത്തരം തീരുമാനങ്ങൾ കൈക്കൊള്ളുന്നത് ചില നിബന്ധനകളെ അടിസ്ഥാനമാക്കിയാണ്. C++ ഈ ആവശ്യം നിറവേറ്റുന്നത് നിയന്ത്രണ പ്രസ്താവനകളുടെ സഹായത്തോടെയാണ്. ഈ പ്രസ്താവനകൾ പ്രോഗ്രാം നിർവഹണത്തിലെ സാധാരണ രീതിക്ക് മാറ്റം വരുത്തുന്നതിനായി ഉപയോഗിക്കുന്നു. നിയന്ത്രണ പ്രസ്താവനകളെ രണ്ടായി തരംതിരിക്കാം. (1) തീരുമാനമെടുക്കൽ/തിരഞ്ഞെടുക്കൽ (Decision making/Selection statement) (2) ആവർത്തന പ്രസ്താവനകൾ (Iteration statement). ഈ പ്രസ്താവനകളും ഇവയുടെ വാക്യഘടനയും നിർവഹണ രീതികളും നമുക്ക് ചർച്ച ചെയ്യാം.

7.1 തീരുമാനങ്ങൾ എടുക്കുന്നതിനുള്ള പ്രസ്താവനകൾ (Decision making statements)

പ്രശ്നങ്ങൾ നിർദ്ധാരണം ചെയ്യുമ്പോൾ കമ്പ്യൂട്ടറുകളിൽ എല്ലാ പ്രസ്താവനകളും എല്ലാ സന്ദർഭങ്ങളിലും ഒരുപോലെ പ്രവർത്തിക്കണമെന്നില്ല. ചില പ്രസ്താവ

നകൾ ഒരു സന്ദർഭത്തിൽ പ്രവർത്തിക്കുമെങ്കിലും മറ്റു ചില സന്ദർഭങ്ങളിൽ പ്രവർത്തിക്കണമെന്നില്ല. ഇത്തരം സന്ദർഭങ്ങളിൽ കമ്പ്യൂട്ടറിന് ആവശ്യമായ തീരുമാനങ്ങൾ എടുക്കേണ്ടതുണ്ട്. ഇതിനായി നാം ഇവിടെ അനുയോജ്യമായ നിബന്ധനകൾ നൽകുകയും അവയെ കമ്പ്യൂട്ടർ വിലയിരുത്തുകയും വേണം. ഈ ഫലത്തിന്റെ അടിസ്ഥാനത്തിൽ അത് ഒരു തീരുമാനം എടുക്കുന്നു. ഈ തീരുമാനങ്ങൾ ഒന്നുകിൽ ഒരു പ്രത്യേക പ്രസ്താവനയെ പ്രവർത്തിപ്പിക്കുന്നതിനായി തിരഞ്ഞെടുക്കുന്നതിനോ അല്ലെങ്കിൽ ചില പ്രസ്താവനകളെ പ്രവർത്തിപ്പിക്കുന്നതിൽ നിന്നും ഒഴിവാക്കുന്നതിനോ ആയിരിക്കും. ഇപ്രകാരം ചില പ്രസ്താവനകൾ മാത്രം നിർവഹണം നടത്തുന്നതിനായി C++ ൽ തീരുമാനമെടുക്കൽ പ്രസ്താവനകൾ അല്ലെങ്കിൽ തെരഞ്ഞെടുക്കൽ പ്രസ്താവനകൾ ഉപയോഗിക്കുന്നു. `if`, `switch` എന്നിവയാണ് C++ ലെ രണ്ടുതരം തിരഞ്ഞെടുക്കൽ പ്രസ്താവനകൾ.

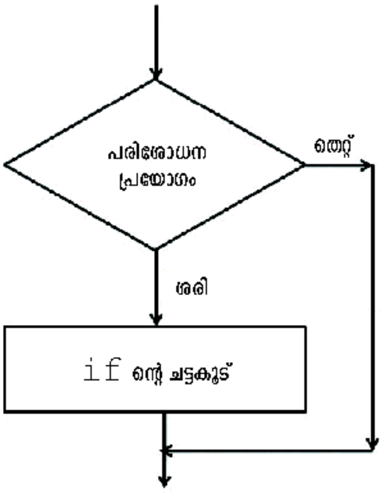
7.1.1 if പ്രസ്താവന (if statement)

ഒരു നിബന്ധനയുടെ (condition) അടിസ്ഥാനത്തിൽ ഒരു കൂട്ടം പ്രസ്താവനകളെ പ്രവർത്തിപ്പിക്കുന്നതിനായി `if` പ്രസ്താവന ഉപയോഗിക്കുന്നു. C++ ൽ നിബന്ധനകൾ (പരിശോധനാ പ്രയോഗങ്ങൾ എന്നും അറിയപ്പെടുന്നു) നൽകുന്നത് റിലേഷണൽ അല്ലെങ്കിൽ ലോജിക്കൽ പ്രയോഗങ്ങൾ ഉപയോഗിച്ചാണ്. `if` പ്രസ്താവനയുടെ വാക്യഘടന (`Syntax`) താഴെ കൊടുത്തിരിക്കുന്നു.

```
if (പരിശോധന പ്രയോഗം)
{
    പ്രസ്താവനകൾ;
}
```

if ചട്ടകൂടിലെ നിബന്ധനകൾ ശരിയാണെങ്കിൽ പ്രവർത്തിക്കേണ്ട പ്രസ്താവന

ഇവിടെ പരിശോധനാ പ്രയോഗം എന്നത് ഒന്നുകിൽ റിലേഷണൽ പ്രയോഗം അല്ലെങ്കിൽ ലോജിക്കൽ പ്രയോഗമായ ഒരു നിബന്ധനയെയാണ് സൂചിപ്പിക്കുന്നത്. പരിശോധനാ പ്രയോഗം ശരിയാണെങ്കിൽ (True-പൂജ്യം അല്ലാത്ത വില) `if` -നോടു ചേർന്നുള്ള പ്രസ്താവനയോ അല്ലെങ്കിൽ ഒരു കൂട്ടം പ്രസ്താവനകളോ പ്രവർത്തിക്കും. അല്ലെങ്കിൽ `if` -നു ശേഷമുള്ള പ്രസ്താവനയിലേക്ക് നിയന്ത്രണം കൈമാറുന്നു. ചിത്രം 7.1 `if` പ്രസ്താവനയുടെ പ്രവർത്തന രീതി കാണിക്കുന്നു. `if` ഉപയോഗിക്കുമ്പോൾ താഴെ പറയുന്ന ചില കാര്യങ്ങൾ ഓർത്തിരിക്കേണ്ടതുണ്ട്.



ചിത്രം 7.1: ശള പ്രസ്താവനയുടെ പ്രവർത്തനം

- പരിശോധനാ പ്രയോഗം എപ്പോഴും ആവരണ ചിഹ്നത്തിന് അകത്തായിരിക്കണം.
- നിബന്ധനയിലുള്ള പ്രയോഗം റിലേഷണൽ പ്രയോഗങ്ങൾ ഉപയോഗിച്ചുള്ള ലളിതമായ പ്രയോഗങ്ങളോ, ലോജിക്കൽ പ്രയോഗങ്ങൾ ഉപയോഗിച്ചുള്ള സംയുക്ത പ്രയോഗങ്ങളോ ആകാം.
- `if` പ്രസ്താവനയോടുകൂടി ഒന്നോ അതിലധികമോ പ്രസ്താവനകൾ ഉണ്ടാകാം. ഒരു പ്രസ്താ

വന മാത്രമാണെങ്കിൽ {,} എന്നീ ബ്രാക്കറ്റുകൾ നിർബന്ധമില്ല. ഒന്നിൽ കൂടുതൽ പ്രസ്താവനകൾ ഉണ്ടെങ്കിൽ ഈ ബ്രാക്കറ്റുകൾ നിർബന്ധമാണ്.

പ്രോഗ്രാം 7.1 ഒരു വിദ്യാർഥിയുടെ സ്കോർ സ്വീകരിക്കുകയും അത് 18 ഓ അതിലധികമോ ആണെങ്കിൽ "You have Passed" എന്ന് പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു. (പാസാവാൻ മിനിമം 18 സ്കോർ വേണമെന്ന് വിചാരിക്കുക)

പ്രോഗ്രാം 7.1: സ്കോർ 18 ഓ അതിലധികമോ ആണെങ്കിൽ "You have Passed" എന്ന പ്രദർശിപ്പിക്കുന്നതിന്

```
#include<iostream>
using namespace std;

{
    int score ;
    cout << "Enter your score: ";
    cin >> score;
    if (score >= 18)
         cout << "You have passed";
    return 0;
}
```

if ന്റെ ചട്ടകൂട്

പ്രോഗ്രാം 7.1 – ന്റെ മാതൃകാ ഔട്ട്പുട്ട് താഴെകൊടുത്തിരിക്കുന്നു.

```
Enter your score: 25
You have passed
```

പ്രോഗ്രാം 7.1 –ൽ ഒരു വിദ്യാർത്ഥിയുടെ സ്കോർ നൽകുകയും അത് `score` എന്ന വേരിയബിളിൽ സംഭരിക്കുകയും ചെയ്യുന്നു. പരിശോധനാപ്രയോഗം സ്കോർ എന്ന വേരിയബിളിലെ വില 18-ഓ അതിൽ അധികമോ ആണോ എന്നു നോക്കുന്നു. പരിശോധനാപ്രയോഗം ശരിയാണെങ്കിൽ `if` –ന്റെ ഭാഗം പ്രവർത്തിക്കുന്നു. അതായത് സ്കോർ 18-ഓ അതിലധികമോ ആണെങ്കിൽ `"You have Passed"` എന്ന സന്ദേശം സ്ക്രീനിൽ പ്രദർശിപ്പിക്കപ്പെടുന്നു. അല്ലാത്ത പക്ഷം ഒരു ഔട്ട്പുട്ടും ലഭിക്കുന്നില്ല.

`if`-നോടു കൂടിയുള്ള ഭാഗം ഒരു ടാബ് ദൂരത്തിന് ശേഷമാണ് എഴുതിയിട്ടുള്ളത് എന്നത് ശ്രദ്ധിക്കുക. നാം അതിനെ ഇൻഡന്റേഷൻ എന്നു വിളിക്കുന്നു. ഇത് പ്രോഗ്രാമിന്റെ വായന ക്ഷമത വർദ്ധിപ്പിക്കുകയും തെറ്റുകൾ കണ്ടുപിടിക്കാൻ സഹായിക്കുകയും ചെയ്യുന്നു. എന്നാൽ ഇവ പ്രോഗ്രാമിന്റെ പ്രവർത്തനത്തിൽ ഒരു സ്വാധീനവും ചെലുത്തുന്നില്ല.

താഴെ കൊടുത്തിരിക്കുന്ന C++ പ്രോഗ്രാം ശകലം ശ്രദ്ധിക്കുക. തന്നിരിക്കുന്ന ഇൻപുട്ട് ഒരു അക്ഷരമാണോ അല്ലെങ്കിൽ ഒരു അക്കമാണോ എന്ന് ഇത് പരിശോധിക്കുന്നു.

```
char ch;
cin >> ch;
if (ch >= 'a' && ch <= 'z')
```

ലോജിക്കൽ പ്രസ്താവന വിലയിരുത്തുന്നു.

```
        cout << "You entered an alphabet";
    if (ch >= '0' && ch <= '9')
    {
        cout << "You entered a digit\n";
        cout << "It is a decimal number ";
    }
```

ഒരു പ്രസ്താവന മാത്രമേയുള്ളൂ. അതുകൊണ്ട് {, } ആവശ്യമില്ല

ഒന്നിൽകൂടുതൽ പ്രസ്താവനകൾ ഉള്ളതു കൊണ്ട് അവ നിർബന്ധമായും {,} ന് അകത്തായിരിക്കണം

7.1.2 if... else പ്രസ്താവന (if... else statement)

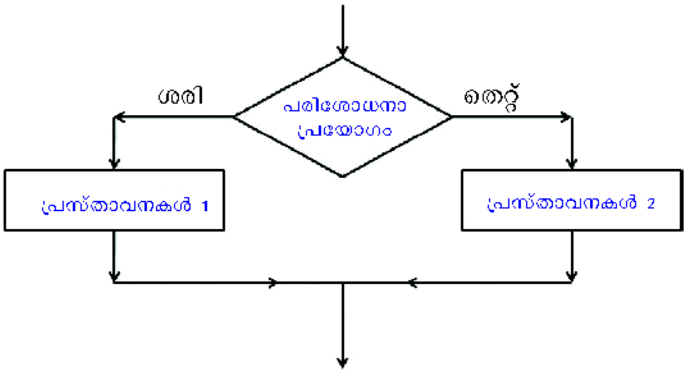
പ്രോഗ്രാം 7.1-ലെ if പ്രസ്താവന പരിഗണിക്കുക.

```
if (score >= 18)
    cout << "You have passed";
```

ഇവിടെ സ്കോർ പതിനെട്ടോ അതിൽ കൂടുതലോ ആണെങ്കിൽ മാത്രമേ ഔട്ട്പുട്ട് ലഭിക്കുന്നുള്ളൂ. നൽകിയ സ്കോർ 18-ൽ കുറവാണെങ്കിൽ എന്ത് സംഭവിക്കും? ഒരു ഔട്ട്പുട്ടും ലഭിക്കില്ല എന്ന് വ്യക്തമാണ്. പരിശോധന പ്രയോഗം വിലയിരുത്തുമ്പോൾ തെറ്റ് (false) ലഭിക്കുകയാണെങ്കിൽ മറ്റൊരു കൂട്ടം പ്രസ്താവനകൾ തിരഞ്ഞെടുക്കുന്നതിനുള്ള അവസരം നമുക്ക് ലഭിക്കാതെ വരുന്നു. നിബന്ധന തെറ്റാവുന്ന അവസരത്തിൽ ചില പ്രവർത്തനങ്ങൾ ചെയ്യണമെങ്കിൽ if പ്രസ്താവനയുടെ മറ്റൊരു രൂപമായ **if.. else** നമുക്ക് ഉപയോഗിക്കാം. ഇതിന്റെ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
    if    (പരിശോധനാ പ്രയോഗം)
    {
          പ്രസ്താവനകൾ 1 ;
    }
    else
    {
          പ്രസ്താവനകൾ 2;
    }
    if    (test expression)
    {
          statement block 1;
    }
    else
    {
          statement
block 2;
    }
```

പരിശോധനാപ്രയോഗം ശരിയാണെങ്കിൽ പ്രസ്താവനകൾ 1 ഉം തെറ്റാണെങ്കിൽ പ്രസ്താവനകൾ 2 ഉം പ്രവർത്തിക്കുന്നു. if...else പ്രസ്താവനയുടെ പ്രവർത്തനം ചിത്രം 7.2-ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 7.2 if....else പ്രസ്താവനയുടെ ഫ്ളോചാർട്ട്

താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് ശകലം if...else പ്രസ്താവനയുടെ പ്രവർത്തനം വിവരിക്കുന്നു.

```
if (score >= 18)
     cout << "Passed";
else
     cout << "Failed";
```

സ്കോർ 18 ഓ അതിൽ അധികമോ ആണെങ്കിൽ മാത്രം ഈ പ്രസ്താവന പ്രവർത്തിക്കുന്നു. (അതായത് പരിശോധനാ പ്രയോഗം ശരിയാകുമ്പോൾ)

സ്കോർ 18 ൽ കുറവാണെങ്കിൽ ഈ പ്രസ്താവന പ്രവർത്തിക്കും. (അതായത് പരിശോധനാ പ്രയോഗം തെറ്റാകുമ്പോൾ.)

രണ്ട് കുട്ടികളുടെ ഉയരം ഇൻപുട്ടായി സ്വീകരിച്ച് അവരിൽ ഉയരമുള്ള കുട്ടിയെ കണ്ടുപിടിക്കുന്നതിനുള്ള ഒരു പ്രോഗ്രാം നമുക്കെഴുതാം.

പ്രോഗ്രാം 7.2: വിദ്യാർത്ഥികളുടെ ഉയരം താരതമ്യം ചെയ്ത് അവരിൽ ഉയരം കൂടുതലുള്ള ആളെ കണ്ടുപിടിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
int main()
{
    int ht1, ht2;
    cout << "Enter heights of the two students: ";
    cin >> ht1 >> ht2;
    if (ht1 > ht2)    //decision making based on condition
        cout<<"Student with height "<<ht1<<" is taller";
    else
        cout<<"Student with height "<<ht2<<" is taller";
    return 0;
}
```

പ്രോഗ്രാം 7.2 പ്രവർത്തിക്കുമ്പോൾ ht1>ht2 എന്ന റിലേഷണൽ പ്രയോഗത്തിന്റെ ഫലത്തെ ആശ്രയിച്ച് ഏതെങ്കിലും ഒരു ഔട്ട്പുട്ട് പ്രദർശിപ്പിക്കപ്പെടും. മാതൃകാ ഔട്ട്പുട്ടുകൾ താഴെ കൊടുത്തിരിക്കുന്നു.

ഔട്ട്പുട്ട് 1:
```
Enter the height of two students: 170   165
Student with height 170 is taller
```

ഔട്ട്പുട്ട് 2:
```
Enter the height of two students: 160   171
Student with height 171 is taller
```

ഔട്ട്പുട്ട് 1-ൽ ht1 -ന് 170-ഉം ht2 -ന് 165-ഉം ഇൻപുട്ടായി നൽകിയിരിക്കുന്നു. അതുകൊണ്ട് (ht1>ht2) എന്ന പരിശോധനാപ്രയോഗം ശരിയാവുകയും തൽഫലമായി if ബ്ലോക്ക് പ്രവർത്തിക്കുകയും ചെയ്യുന്നു. ഔട്ട്പുട്ട് 2-ൽ ht1 -ന് 160-ഉം ht2-ന് 171-ഉം വിലകൾ നൽകുമ്പോൾ ht1>ht2 എന്ന പരിശോധനാ പ്രയോഗം തെറ്റാവുകയും തൽഫലമായി else ബ്ലോക്ക് പ്രവർത്തിക്കുകയും ചെയ്യുന്നു. if .. else പ്രസ്താവനയിൽ ഒന്നുകിൽ if നോട്

അനുബന്ധിച്ചുള്ള കോഡും (പ്രസ്താവനകൾ 1) അല്ലെങ്കിൽ `else` നോട് അനുബന്ധിച്ചുള്ള കോഡും (പ്രസ്താവനകൾ 2) പ്രവർത്തിക്കുന്നു.

പരിശോധനാ പ്രയോഗത്തിൽ ഒരു ഓപറൻഡ് ആയി അരിത്തമെറ്റിക് പ്രയോഗം ഉപയോഗിച്ചിരിക്കുന്ന മറ്റൊരു പ്രോഗ്രാം നമുക്ക് നോക്കാം. പ്രോഗ്രാം 7.3 ഈ ആശയം ഉപയോഗിച്ച് ഒരു സംഖ്യ ഒറ്റസംഖ്യയാണോ ഇരട്ടസംഖ്യയാണോ എന്ന് പരിശോധിക്കുന്നു.

പ്രോഗ്രാം 7.3: തന്നിരിക്കുന്ന സംഖ്യ ഒറ്റസംഖ്യയാണോ ഇരട്ടസംഖ്യയാണോ എന്ന് പരിശോധിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
int main()
{
    int num;
    cout << "Enter the number: ";
    cin >> num;
    if (num%2 == 0)
        cout << "The given number is Even";
    else
        cout << "The given number is Odd";
    return 0;
}
```

പ്രോഗ്രാം 7.3 ന്റെ ചില മാതൃക ഔട്ട്പുട്ടുകൾ താഴെ കാണിച്ചിരിക്കുന്നു

ഔട്ട്പുട്ട് 1:

```
Enter the number: 7
The given number is Odd
```

ഔട്ട്പുട്ട് 2:

```
Enter the number: 10
The given number is Even
```

ഈ പ്രോഗ്രാമിൽ `(num%2)` എന്ന പ്രയോഗം `num` ലെ വിലയെ 2 കൊണ്ട് ഹരിച്ച് കിട്ടുന്ന ശിഷ്ടത്തെ പൂജ്യമായി താരതമ്യം ചെയ്യുന്നു. അത് തുല്യമാണെങ്കിൽ `if` പ്രസ്താവനകൾ പ്രവർത്തിക്കും അല്ലെങ്കിൽ `else` പ്രസ്താവനകൾ പ്രവർത്തിക്കും.

നമുക്കു ചെയ്യാം

1. തന്നിരിക്കുന്ന സംഖ്യ പോസിറ്റീവ് ആണോ നെഗറ്റീവ് ആണോ എന്ന് പരിശോധിക്കാനുള്ള പ്രോഗ്രാം എഴുതുക. (പൂജ്യം ഒഴികെയുള്ള സംഖ്യ മാത്രമേ ഇൻപുട്ട് ചെയ്യുന്നുള്ളൂ.)
2. ഒരു അക്ഷരം സ്വീകരിച്ച് 'M' ആണെങ്കിൽ Male എന്നും 'F' ആണെങ്കിൽ Female എന്നും ഔട്ട്പുട്ട് കാണിക്കുന്നതിനുള്ള പ്രോഗ്രാം എഴുതുക.
3. നിങ്ങളുടെ പ്രായം ഇൻപുട്ടായി നൽകി അത് 18 വയസ്സിനു മുകളിലാണെങ്കിൽ വോട്ടു ചെയ്യാൻ യോഗ്യതയുണ്ടെന്നും അല്ലെങ്കിൽ യോഗ്യതയില്ലെന്നും പ്രദർശിപ്പിക്കാനുള്ള പ്രോഗ്രാം എഴുതുക.

7.1.3 നെസ്റ്റഡ് if (Nested if)

ചില സാഹചര്യങ്ങളിൽ if പ്രസ്താവനയുടെ അകത്തുനിന്നുകൊണ്ട് തീരുമാനങ്ങൾ എടുക്കേണ്ട ആവശ്യം വരും. ഒരു if ബ്ലോക്കിനകത്ത് മറ്റൊരു if ബ്ലോക്ക് എഴുതുന്നതിനെ നെസ്റ്റിങ്ങ് എന്നു പറയുന്നു. നെസ്റ്റഡ് എന്നാൽ ഒന്നിനകത്ത് മറ്റൊന്ന് എന്നാണ് അർഥം. താഴെ കൊടുത്തിരിക്കുന്ന പ്രോഗ്രാം ശകലം പരിഗണിക്കുക.

```
if (score >= 60)
{
    if(age >= 18)
        cout<<"You are selected for the course!";
}
```

പുറമെയുള്ള if — `if (score >= 60)`

അകത്തുള്ള if — `if(age >= 18)`

ഈ കോഡ് ശകലത്തിൽ Scoreന്റെ വില 60 ഓ അതിൽ കൂടുതലോ ആണെങ്കിൽ പ്രോഗ്രാമിന്റെ നിയന്ത്രണം പുറത്തെ **if** ബ്ലോക്കിനുള്ളിലേക്ക് പ്രവേശിക്കുന്നു. അതിനുശേഷം അകത്തുള്ള **if** ന്റെ പരിശോധനാ പ്രസ്താവന വിലയിരുത്തുന്നു. (അതായത് age ന്റെ വില പതിനെട്ടോ അതിൽ കൂടുതലോ എന്ന്). ഇത് ശരിയാണെങ്കിൽ "You are selected for the course!" എന്ന സന്ദേശം പ്രദർശിപ്പിക്കും. തുടർന്ന് പുറത്തെ **if** പ്രസ്താവനകൾക്ക് ശേഷമുള്ള പ്രസ്താവനകൾ പ്രവർത്തിക്കുന്നു.

ഒരു if പ്രസ്താവനക്കുള്ളിലെ മറ്റൊരു if പ്രസ്താവനയെ നെസ്റ്റഡ് if (nested **if**) എന്നു വിളിക്കുന്നു. നെസ്റ്റഡ് if-ന്റെ വിപുലീകരിച്ച വാക്യഘടന താഴെ കൊടുത്തിരിക്കുന്നു.

```
if  (test expression1)
{
    if (test expression 2)
        statement 1;
    else
        statement 2;
}
else
{
    body of else;
}
```

statement 1 — പരിശോധനാ പ്രയോഗം രണ്ടും ശരിയാകുമ്പോൾ മാത്രം പ്രവർത്തിക്കുന്നു.

statement 2 — പരിശോധനാ പ്രയോഗം 1 ശരിയാവുകയും പരിശോധനാ പ്രയോഗം 2 തെറ്റാവുകയും ചെയ്യുമ്പോൾ പ്രവർത്തിക്കുന്നു.

body of else — പരിശോധനാ പ്രയോഗം 1 തെറ്റാകുമ്പോൾ പ്രവർത്തിക്കും. പരിശോധനാ പ്രയോഗം 2 നിർദ്ധാരണം ചെയ്യുന്നില്ല.

നെസ്റ്റഡ് if മായി ബന്ധപ്പെട്ട് ശ്രദ്ധിക്കേണ്ട പ്രധാന കാര്യം ഒരു else എപ്പോഴും അതേ ബ്ലോക്കിൽ തന്നെയുള്ള ഏറ്റവും അടുത്ത if മായി ബന്ധപ്പെട്ടിരിക്കുന്നു എന്നതാണ്. ഒരു ഉദാഹരണത്തിലൂടെ നമുക്ക് ഇത് ചർച്ച ചെയ്യാം. താഴെ പറയുന്ന പ്രോഗ്രാം ശകലം പരിഗണിക്കുക.

```
cout<<"Enter your score in Computer Science exam: ";
cin>>score;
if (score >= 18)
     cout<<"You have passed";
     if(score >= 54)
          cout<<"with A+ grade !";
else
     cout<<"\nYou have failed";
```

Score ന് 45 എന്ന വില നൽകുകയാണെങ്കിൽ ഔട്ട്പുട്ട് താഴെ ഉള്ളതുപോലെയായിരിക്കും.

```
You have passed
You have failed
```

യുക്തിപരമായി ഇത് ശരിയല്ല എന്ന് നമുക്ക് അറിയാം. കോഡിന്റെ ഇൻഡന്റേഷൻ ശരിയാണെങ്കിലും പ്രവർത്തനത്തിൽ പ്രതിഫലിച്ചിട്ടില്ല. ഇതിൽ രണ്ടാമത്തെ `if` പ്രസ്താവന നെസ്റ്റഡ് `if` ആയി പരിഗണിച്ചിട്ടില്ല. പകരം അതിനെ `else` ബ്ലോക്കോടു കൂടിയ സ്വതന്ത്രമായ ഒരു `if` ആയിട്ടാണ് കണക്കാക്കിയിട്ടുള്ളത്. അതുകൊണ്ട് ആദ്യത്തെ `if` പ്രസ്താവനയിലെ പരിശോധനാ പ്രയോഗം ശരിയായതിനാൽ അതിലെ `if` ബ്ലോക്ക് പ്രവർത്തനത്തിനായി തിരഞ്ഞെടുക്കുന്നു. ഇത് ഔട്ട്പുട്ടിലെ ഒന്നാമത്തെ വരിക്ക് കാരണമാകുന്നു. അതിനുശേഷം രണ്ടാമത്തെ `if` പ്രസ്താവനയിലെ പരിശോധനാ പ്രയോഗം തെറ്റായതിനാൽ ഔട്ട്പുട്ടിലെ രണ്ടാമത്തെ വരി ലഭിക്കുന്നു. അതുകൊണ്ട് ശരിയായ ഔട്ട്പുട്ട് ലഭിക്കുന്നതിനായി കോഡ് താഴെയുള്ളതുപോലെ പരിഷ്കരിക്കണം.

```
cout<<"Enter your score in Computer Science exam: ";
cin>>score;
if (score >= 18)
{
     cout<<"You have passed";
     if(score >= 54)
          cout<<" with A+ grade !";
}
else
     cout<<"\nYou have failed";
```

ഒരു ജോഡി { } ബ്രാക്കറ്റുകളുടെ സഹായത്തോടെ നെസ്റ്റിങ്ങ് നടപ്പിലാക്കിയിരിക്കുന്നു.

else പ്രസ്താവന ഇപ്പോൾ പുറത്തെ if നോട് ചേർന്നിരിക്കുന്നു.

മുൻ ഉദാഹരണത്തിലെ ഇൻപുട്ട് ആയ 45 ഇവിടെയും നൽകിയാൽ താഴെ പറയുന്ന ഔട്ട്പുട്ട് ലഭിക്കും.

```
You have passed
```

തന്നിരിക്കുന്ന മൂന്നു സംഖ്യകളിൽ വലുത് കണ്ടെത്തുന്നതിനായി പ്രോഗ്രാം 7.4 ൽ നെസ്റ്റഡ് `if` ഉപയോഗിച്ചിരിക്കുന്നു. ഈ പ്രോഗ്രാമിൽ `if` പ്രസ്താവന `if` ബ്ലോക്കിനകത്തും `else` ബ്ലോക്കിനകത്തും ഉപയോഗിച്ചിരിക്കുന്നു.

പ്രോഗ്രാം 7.4: മൂന്ന് സംഖ്യകളിൽ നിന്നും വലുത് കണ്ടുപിടിക്കുന്നതിന്

```
#include <iostream>
using namespace std;
int main()
{
    int x, y, z;
    cout << "Enter three different numbers: ";
    cin >> x >> y >> z ;
    if (x > y)
    {
        if (x > z)
            cout << "The largest number is: " << x;
        else
            cout << "The largest number is: " << z;
    }
    else
    {
        if (y > z)
            cout << "The largest number is: " << y;
        else
            cout << "The largest number is: " << z;
    }
    return 0;
}
```

പ്രോഗ്രാം 7.4-ന്റെ ഒരു മാതൃകാ ഔട്ട്പുട്ട് താഴെ കൊടുത്തിരിക്കുന്നു.

```
Enter three different numbers: 6   2   7
The largest number is: 7
```

ഇൻപുട്ട് നൽകിയ പ്രകാരം പുറമെയുള്ള `if` ലെ പരിശോധനാപ്രയോഗം `(x>y)` ശരിയായതിനാൽ അതിലെ അകത്തെ `if` ലേക്ക് പ്രവേശിക്കുന്നു. ഇവിടെ `(x>z)` എന്ന പരിശോധനാപ്രയോഗം തെറ്റായതിനാൽ അതിന്റെ `else` ബ്ലോക്ക് പ്രവർത്തിക്കുന്നു. അതുകൊണ്ട് z-ന്റെ വില ഔട്ട്പുട്ടായി പ്രദർശിപ്പിക്കുന്നു.

സ്വയം വിലയിരുത്താം

1. ഒരു പൂർണ്ണ സംഖ്യ ഇൻപുട്ടായി സ്വീകരിച്ച്, അത് പോസറ്റീവാണോ നെഗറ്റീവാണോ പൂജ്യം ആണോ എന്ന് പരിശോധിക്കുന്നതിനുള്ള ഒരു പ്രോഗ്രാം എഴുതുക
2. മൂന്ന് സംഖ്യകളെ ഇൻപുട്ടായി സ്വീകരിച്ച് അതിലെ ചെറുത് പ്രിന്റ് ചെയ്യാനുള്ള ഒരു പ്രോഗ്രാം എഴുതുക.

7.1.4 else if ലാഡർ (The else if ladder)

ഒരു else ബ്ലോക്കിനുള്ളിൽ ഒരു if പ്രസ്താവന ഉപയോഗിക്കേണ്ട സാഹചര്യം ഉണ്ടായേക്കാം. അനേകം നിബന്ധനകൾ (condition) ആവശ്യമുള്ള പ്രോഗ്രാമുകളിൽ അത് ഉപയോഗിക്കുന്നു. പ്രവർത്തനത്തിനായി ഏത് പ്രസ്താവന തിരഞ്ഞെടുക്കണമെന്ന് അതാത് നിബന്ധന തീരുമാനിക്കുന്നു. if പ്രസ്താവനയെ അടിസ്ഥാനമാക്കിയുള്ള ഒരു സാധാരണ പ്രോഗ്രാമിംഗ് രൂപകൽപനയാണ് else if ലാഡർ. അതിന്റെ രൂപഘടനയിലുള്ള പ്രത്യേകത കാരണം അതിനെ else if സ്റ്റെയർ കെയ്സ് എന്നും പറയുന്നു. ഇത് if..else if പ്രസ്താവന എന്നും അറിയപ്പെടുന്നു. else if ലാഡറിന്റെ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
if (പരിശോധനാ പ്രയോഗം 1)
       പ്രസ്താവനകൾ 1;
          else if (പരിശോധനാ പ്രയോഗം 2)
             പ്രസ്താവനകൾ 2;
                else if (പരിശോധനാ പ്രയോഗം 3)
                   പ്രസ്താവനകൾ 3;
                      ...............
                         else
                            പ്രസ്താവനകൾ n;
```

```
if (test expression 1)
       statement block 1;
          else if (test expression 2)
             statement block 2;
                else if (test expression 3)
                   statement block 3;
                      ...............
                         else
                            statement block n;
```

ആദ്യം പരിശോധനാ പ്രയോഗം 1 വിലയിരുത്തുമ്പോൾ അത് ശരിയാണെങ്കിൽ പ്രസ്താവനകൾ 1 പ്രവർത്തിച്ചതിനുശേഷം ലാഡറിൽ നിന്ന് പുറത്തേക്ക് വരുന്നു. അതായത് ലാഡറിന്റെ ബാക്കി ഭാഗം ഒഴിവാക്കപ്പെടുന്നു. പരിശോധനാപ്രയോഗം 1 വിലയിരുത്തുമ്പോൾ അത് തെറ്റാണെങ്കിൽ പരിശോധനാ പ്രയോഗം 2 വിലയിരുത്തുന്നു. ഈ പ്രക്രിയ അങ്ങിനെ തുടരുന്നു. ഏതെങ്കിലും ഒരു പരിശോധനാപ്രയോഗം ശരിയാണെങ്കിൽ അതിനനുസൃതമായ പ്രസ്താവനകൾ പ്രവർത്തിച്ചതിനുശേഷം നിയന്ത്രണം ലാഡറിന്റെ പുറത്തേക്ക് വരുന്നു. എല്ലാ പരിശോധനാ പ്രയോഗങ്ങളും വിലയിരുത്തുമ്പോൾ തെറ്റാണെങ്കിൽ അവസാന else-നുശേഷമുള്ള പ്രസ്താവനകൾ n പ്രവർത്തിക്കുന്നു. വാക്യഘടനയിൽ കൊടുത്തിരിക്കുന്ന ഇൻഡന്റേഷൻ നിരീക്ഷിക്കുകയും else if ലാഡർ ഉപയോഗിക്കുന്നതിന് ഈ രീതി പിന്തുടരുകയും ചെയ്യുക.

ഒരു വിദ്യാർത്ഥിക്ക് ഒരു വിഷയത്തിൽ 100 ൽ ലഭ്യമായ സ്കോറിന്റെ അടിസ്ഥാനത്തിൽ ഗ്രേഡ് കണ്ടുപിടിക്കാനുള്ള പ്രോഗ്രാം else if ലാഡർ ഉപയോഗിച്ച് നമുക്ക് വിവരിക്കാം.

താഴെയുള്ള പട്ടികയിൽ കൊടുത്തിരിക്കുന്ന മാനദണ്ഡമനുസരിച്ചാണ് ഗ്രേഡ് കണ്ടുപിടിക്കേണ്ടത്.

സ്കോർ	ഗ്രേഡ്
80 ഓ അതിൽ കൂടുതലോ	A
60 മുതൽ 79 വരെ	B
40 മുതൽ 59 വരെ	C
30 മുതൽ 39 വരെ	D
30 ൽ താഴെ	E

പ്രോഗ്രാം 7.5: തന്നിരിക്കുന്ന സ്കോറിന്റെ അടിസ്ഥാനത്തിൽ വിദ്യാർത്ഥിയുടെ ഗ്രേഡ് കണ്ടുപിടിക്കുന്നതിന്

```
#include <iostream>
using namespace std;
int main()
{
    int score;
    cout << "Enter your score: ";
    cin >> score;
    if (score >= 80)
        cout << "A Grade";
      else if (score >= 60)
          cout << "B Grade ";
        else if (score >= 40)
            cout << "C grade";
          else if (score >= 30)
              cout << "D grade";
            else
                cout << "E Grade";
    return 0;
}
```

പ്രോഗ്രാം 7.5 ന്റെ മാതൃക ഔട്ട്പുട്ടുകളാണ് താഴെയുള്ളത്.

ഔട്ട്പുട്ട് 1:

```
Enter your score: 73
B Grade
```

ഔട്ട്പുട്ട് 2:

```
Enter your score: 25
E Grade
```

പ്രോഗ്രാം 7.5 ൽ ആദ്യം പരിശോധനാ പ്രയോഗം `score>=80` വിലയിരുത്തുന്നു. ഔട്ട്പുട്ട് 1-ൽ ഇൻപുട്ട് ചെയ്ത വില 73 ആയതിനാൽ പരിശോധനാ പ്രയോഗം തെറ്റ് ആകുകയും

score>=60 എന്ന അടുത്ത പരിശോധനാ പ്രയോഗം വിലയിരുത്തുകയും ചെയ്യുന്നു. ഇവിടെ ഇത് ശരിയായതിനാൽ "B Grade" എന്ന് പ്രദർശിപ്പിക്കുകയും else if ലാഡറിന്റെ ബാക്കി ഭാഗം ഒഴിവാക്കുകയും ചെയ്യുന്നു. എന്നാൽ ഔട്ട്പുട്ട് 2 - ൽ എല്ലാ പരിശോധനപ്രയോഗങ്ങളും തെറ്റാണെന്ന് വിലയിരുത്തിയതിനാൽ അവസാനത്തെ else ബ്ലോക്ക് പ്രസ്താവന പ്രവർത്തിക്കുകയും "E grade" എന്ന ഔട്ട്പുട്ട് ലഭിക്കുകയും ചെയ്യുന്നു.

തന്നിരിക്കുന്ന വർഷം അധിവർഷം (leap year) ആണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നതിനുള്ള ഒരു പ്രോഗ്രാം നമുക്ക് എഴുതാം. ഇൻപുട്ട് സംഖ്യ ശതാബ്ദമാണോ എന്ന് പരിശോധിക്കേണ്ടതുണ്ട് (നൂറുകൊണ്ട് ഹരിക്കാൻ സാധിക്കുന്ന വർഷമാണോ എന്ന്). അത് ഒരു ശതാബ്ദ വർഷമാണെങ്കിൽ അതിനെ 400 കൊണ്ട് കൂടി ഹരിക്കാമെങ്കിലേ അത് അധിവർഷമാകുന്നുള്ളൂ. ഇൻപുട്ട് സംഖ്യ ശതാബ്ദ വർഷമല്ലെങ്കിൽ അതിനെ 4 കൊണ്ട് ഹരിക്കുവാൻ സാധിക്കുമോ എന്ന് നാം പരിശോധിക്കേണ്ടതുണ്ട്. അതിനെ ഹരിക്കാൻ സാധിക്കുമെങ്കിൽ തന്നിരിക്കുന്ന വർഷം അധിവർഷം ആണ്, അല്ലെങ്കിൽ അത് ഒരു അധിവർഷമല്ല.

പ്രോഗ്രാം 7.6 തന്നിരിക്കുന്ന വർഷം അധിവർഷമാണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
void main()
{
  int year ;
  cout << "Enter the year (in 4-digits): ";
  cin >> year;
  if (year%100 == 0)// Checks for century year
  {
        if (year%400 == 0)
            cout << "Leap year\n";
        else
            cout<< "Not a leap year\n";
  }
  else if (year%4 == 0)
        cout << "Leap year\n";
  else
        cout<< "Not a leap year\n";
  return 0;
}
```

ശതാബ്ദ വർഷം അധിവർഷം ആകണമെങ്കിൽ അവയെ 400കൊണ്ട് ഹരിക്കാൻ കഴിയണം.

ശതാബ്ദ വർഷമല്ലാത്തവ അധിവർഷം ആകണമെങ്കിൽ അവയെ 4കൊണ്ട് ഹരിക്കുവാൻ കഴിയണം.

പ്രോഗ്രാം 7.6 ന്റെ ചില മാതൃക ഔട്ട്പുട്ടുകൾ നമുക്ക് നോക്കാം.

ഔട്ട്പുട്ട് 1:

```
Enter the year (in 4-digits): 2000
Leap year
```

ഔട്ട്പുട്ട് 2:

```
Enter the year (in 4-digits): 2014
Not a leap year
```

ഔട്ട്പുട്ട് 3:

```
Enter the year (in 4-digits): 2100
Not a leap year
```

ഔട്ട്പുട്ട് 4:

```
Enter the year (in 4-digits): 2004
Leap year
```

`else if` ലാഡറിന്റെ ഉപയോഗം വിവരിക്കുന്നതിനുള്ള ഒരു പ്രോഗ്രാം കൂടി നമുക്ക് എഴുതാം. പ്രോഗ്രാം 7.7-ൽ ആഴ്ചയിലെ ദിവസത്തെ സൂചിപ്പിക്കുന്നതിനായി 1 മുതൽ 7 വരെയുള്ള സംഖ്യ ഇൻപുട്ടു ചെയ്യുന്നതിന് അനുവദിക്കുകയും അതിനനുസൃതമായ ദിവസത്തിന്റെ പേര് പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു. ഇൻപുട്ട് 1 ആണെങ്കിൽ "Sunday" എന്നും 2 ആണെങ്കിൽ "Monday" എന്നും ഔട്ട്പുട്ടുകൾ പ്രദർശിപ്പിക്കപ്പെടുന്നു. ഇതുപോലെ മറ്റു ദിവസങ്ങളും. 1 മുതൽ 7 വരെയുള്ള പരിധിക്ക് പുറത്താണ് ഇൻപുട്ട് എങ്കിൽ "Wrong input" എന്നായിരിക്കും ഔട്ട്പുട്ട്.

പ്രോഗ്രാം 7.7: തന്നിരിക്കുന്ന ദിവസ സംഖ്യയ്ക്ക് അനുസൃതമായ ദിവസത്തിന്റെ പേര് പ്രദർശിപ്പിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
int main()
{
    int day;
    cout << "Enter the day number (1-7): ";
    cin >> day;
    if (day == 1)
      cout << "Sunday";
        else if (day == 2)
          cout << "Monday";
            else if (day == 3)
              cout << "Tuesday";
                else if (day == 4)
                  cout << "Wednesday";
                    else if (day == 5)
                      cout << "Thursday";
                        else if (day == 6)
                          cout << "Friday";
                            else if (day == 7)
                              cout << "Saturday";
                                else
                                  cout << "Wrong input";
    return 0;
}
```

പ്രോഗ്രാം 7.7 ന്റെ ചില മാതൃക ഔട്ട്പുട്ടുകളാണ് താഴെയുള്ളത്.

ഔട്ട്പുട്ട് 1:

```
Enter the day number (1-7): 5
Thursday
```

ഔട്ട്പുട്ട് 2:

```
Enter day number (1-7): 9
Wrong input
```

സ്വയം വിലയിരുത്താം

1. ഒരു പൂർണ്ണ സംഖ്യ ഇൻപുട്ടായി സ്വീകരിച്ച് അത് പോസറ്റീവാണോ നെഗറ്റീവാണോ പൂജ്യമാണോ എന്ന് പരിശോധിക്കുവാനുള്ള പ്രോഗ്രാം if else if പ്രസ്താവന ഉപയോഗിച്ച് എഴുതുക.
2. ഒരു അക്ഷരം (a, b, c അല്ലെങ്കിൽ d) ഇൻപുട്ട് ചെയ്യുന്നതിനും താഴെപറയുന്ന രീതിയിൽ ഔട്ട് പ്രദർശിപ്പിക്കുന്നതിനുമുളള ഒരു പ്രോഗ്രാം എഴുതുക.

   "a - abacus", "b - boolean", "c - computer", "d - debugging"
3. ഒരു അക്ഷരം ഇൻപുട്ട് ചെയ്യുന്നതിനും അത് ആൽഫബെറ്റാണോ, സംഖ്യയാണോ അതോ മറ്റേതെങ്കിലും ക്യാരക്ടർ ആണോ എന്ന് പ്രിന്റ് ചെയ്യുന്നതിനുള്ള ഒരു പ്രോഗ്രാം എഴുതുക.

7.1.5. switch പ്രസ്താവന (switch statement)

else if ലാഡറിന്റെ സഹായത്തോടെ ബഹുശാഖീകരണം (Multiple branching) എന്ന ആശയം നാം കണ്ടു കഴിഞ്ഞു. C++-ലെ മറ്റൊരു രൂപകൽപ്പനയായ switch പ്രസ്താവന ഉപയോഗിച്ച് ഇവയിൽ ചില പ്രോഗ്രാമുകൾ എഴുതുവാൻ സാധിക്കും. ഈ തിരഞ്ഞെടുക്കൽ പ്രസ്താവന ഒരു വേരിയബിളിന്റേയോ ഒരു പ്രയോഗത്തിന്റേയോ (expression) വിലയെ ഒരു കൂട്ടം പൂർണ സംഖ്യകളുമായോ അക്ഷര സ്ഥിരാങ്കങ്ങളുമായോ തുടർച്ചയായി പരിശോധിക്കുന്നു. switch പ്രസ്താവനയുടെ വാക്യഘടന ചുവടെ ചേർത്തിരിക്കുന്നു.

```
switch (പ്രയോഗം)
        case സ്ഥിരാംഗം_1      : പ്രസ്താവനകൾ 1;
                                break;
        case സ്ഥിരാംഗം_2      : പ്രസ്താവനകൾ 2;
                                break;
        case സ്ഥിരാംഗം_3      : പ്രസ്താവനകൾ 3;
                                break;
                              :
                              :
        case സ്ഥിരാംഗം_n-1    : പ്രസ്താവനകൾ n-1;
                                break;
        default               : പ്രസ്താവനകൾ n;
    }
```

```
switch (expression)
        'case' constant_1  : statement block 1;
                             break;
        'case' constant_2  : statement block 2;
                             break;
        'case' constant_3  : statement block 3;
                             break;
                           :
                           :
        'case' constant_n-1  :  statement block n-1;
                             break;
        default            : statement block n;
    }
```

വാക്യഘടനയിൽ switch, case, break, default എന്നിവ കീ വേർഡുകളാണ് (Keyword). ഒരു പൂർണ്ണസംഖ്യയോ ഒരു ക്യാരക്ടർ കോൺസ്റ്റന്റോ കിട്ടാവുന്ന രീതിയിൽ പ്രയോഗത്തെ വിലയിരുത്തുകയും അത് case പ്രസ്താവനകളിൽ കൊടുത്തിരിക്കുന്ന സ്ഥിരാംഗങ്ങൾക്ക് തുല്യമാണോ എന്ന് നോക്കുകയും ചെയ്യുന്നു. ഒരു തുല്യത കണ്ടെത്തിയാൽ ആ case-നോട് അനുബന്ധിച്ചുള്ള പ്രസ്താവനകൾ പ്രവർത്തിക്കും (break പ്രസ്താവന വരെയോ അല്ലെങ്കിൽ switch പ്രസ്താവനയുടെ അവസാനം വരെയോ). തുല്യത കണ്ടെത്തിയില്ലെങ്കിൽ default ബ്ലോക്കിലെ പ്രസ്താവന കൂട്ടം പ്രവർത്തിക്കും. default പ്രസ്താവന നിർബന്ധമല്ല. അത് ഉപയോഗിച്ചിട്ടില്ലെങ്കിൽ തുല്യത കണ്ടെത്താനാവാത്ത സന്ദർഭങ്ങളിൽ മറ്റൊന്നും പ്രവർത്തിക്കുകയില്ല.

switch-നകത്ത് ഉപയോഗിച്ചിരിക്കുന്ന break പ്രസ്താവന C++-ലെ ഒരു ജമ്പ് പ്രസ്താവനയാണ്. break പ്രസ്താവനയിൽ എത്തുമ്പോൾ പ്രോഗ്രാം നിയന്ത്രണം switch പ്രസ്താവനയ്ക്ക് ശേഷമുള്ള പ്രസ്താവനകളിലേക്ക് പോകുന്നു.

ഭാഗം 7.3.2 ൽ break സ്റ്റേറ്റ്മെന്റിനെക്കുറിച്ച് വിശദമായി നമുക്ക് ചർച്ച ചെയ്യാം. പ്രോഗ്രാം 7.7നെ switch പ്രസ്താവന ഉപയോഗിച്ച് എഴുതാവുന്നതാണ്. ഇത് കോഡിന്റെ വായനാസുഖവും ഫലപ്രാപ്തിയും വർദ്ധിപ്പിക്കുന്നു. പ്രോഗ്രാം 7.8 ൽ വരുത്തിയ ഭേദഗതികൾ ശ്രദ്ധിക്കുക.

പ്രോഗ്രാം 7.8: switch പ്രസ്താവന ഉപയോഗിച്ച് ആഴ്ചയിലെ ദിവസം പ്രദർശിപ്പിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
int main()
{     int day ;
      cout << "Enter a number between 1 and 7: ";
      cin >> day ;
      switch (day)
      {
          case 1: cout << "Sunday";
                  break;
```

```
        case 2: cout << "Monday";
                break;
        case 3: cout << "Tuesday";
                break;
        case 4: cout << "Wednesday";
                break;
        case 5: cout << "Thursday";
                break;
        case 6: cout << "Friday";
                break;
        case 7: cout << "Saturday";
                break;
        default: cout << "Wrong input";
     }
     return 0;
}
```

പ്രോഗ്രാം 7.7-ന്റെ ഔട്ട്പുട്ട് തന്നെയായിരിക്കും പ്രോഗ്രാം 7.8-നും. ചില മാതൃകകൾ താഴെ കൊടുത്തിരിക്കുന്നു.

ഔട്ട്പുട്ട് 1:

```
Enter a number between 1 and 7: 5
Thursday
```

ഔട്ട്പുട്ട് 2:

```
Enter a number between 1 and 7: 8
Wrong input
```

പ്രോഗ്രാം 7.8-ൽ day വേരിയബിന്റെ വില case പ്രസ്താവനയിലെ സ്ഥിരാങ്കങ്ങളുമായി താരതമ്യം ചെയ്തിരിക്കുന്നു. ഒരു തുല്യത കണ്ടെത്തുമ്പോൾ ആ case-നോട് അനുബന്ധിച്ചുള്ള ഔട്ട്പുട്ട് പ്രസ്താവന പ്രവർത്തിക്കുന്നു. വേരിയബിൾ day യ്ക്ക് 5 എന്ന വില നാം ഇൻപുട്ടായി കൊടുത്താൽ അഞ്ചാമത്തെ case പ്രസ്താവന തുല്യമാവുകയും cout << "Thursday"; എന്ന പ്രസ്താവന പ്രവർത്തിക്കുകയും ചെയ്യുന്നു. ഇൻപുട്ട് 8 ആണെങ്കിൽ തുല്യത കണ്ടെത്തുവാൻ ആകില്ല. ആയതിനാൽ default ബ്ലോക്ക് പ്രവർത്തിക്കും.

എല്ലാ break പ്രസ്താവനകളെയും ഒഴിവാക്കുകയാണെങ്കിൽ പ്രോഗ്രാം 7.8-ന്റെ ഔട്ട്പുട്ട് നിങ്ങൾക്ക് പ്രവചിക്കാമോ? case സ്ഥിരാങ്കങ്ങളുമായി day യുടെ വില താരതമ്യം ചെയ്യുന്നു. ആദ്യത്തെ തുല്യത കണ്ടെത്തുമ്പോൾ അനുബന്ധ പ്രസ്താവനകളും തുടർന്നുള്ള പ്രസ്താവനകളും പ്രവർത്തിക്കുന്നു (അവശേഷിക്കുന്ന സ്ഥിരാങ്കങ്ങൾ പരിശോധിക്കാതെ). ചില സാഹചര്യങ്ങളിൽ break പ്രസ്താവന മനഃപൂർവ്വം ഒഴിവാക്കാറുണ്ട്. ഒരു switch ലെ തന്നിട്ടുള്ള എല്ലാ case നോടും അനുബന്ധിച്ചുള്ള പ്രസ്താവനകൾ ഒരുപോലെയാണെങ്കിൽ അത്തരം പ്രസ്താവനകൾ അവസാനത്തെ case ൽ നാം എഴുതിയാൽ മതിയാകും. പ്രോഗ്രാം 7.9 ഈ ആശയം വിവരിക്കുന്നു.

പ്രോഗ്രാം 7.9: തന്നിരിക്കുന്ന അക്ഷരം സ്വരാക്ഷരം ആണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
int main()
{
    char ch;
    cout<<"Enter the character to check: ";
    cin>>ch;
    switch(ch)
    {
        case 'A' :
        case 'a' :
        case 'E' :
        case 'e' :
        case 'I' :
        case 'i' :
        case 'O' :
        case 'o' :
        case 'U' :
        case 'u' : cout<<"The given character is a vowel";
                   break;
        default  : cout<<"The given character is not a
vowel";
    }
    return 0;
}
```

പ്രോഗ്രാം. 7.9 നൽകുന്ന ചില ഔട്ട്പുട്ടുകൾ താഴെ കാണിച്ചിരിക്കുന്നു.

ഔട്ട്പുട്ട് 1:

```
Enter a character to check: E
The given character is a vowel
```

ഔട്ട്പുട്ട് 2:

```
Enter a character to check: k
The given character is not a vowel
```

Switch ഉപയോഗിക്കുന്നതിന്റെ അനുയോജ്യതയും ആവശ്യകതയും

Switch പ്രസ്താവന else if ലാഡറിന്റെ അനേകം ശാഖകളുള്ള പ്രസ്താവനകൾക്ക് പകരമായാണ് ഉപയോഗിക്കുന്നതെങ്കിലും ഇവ രണ്ടും ഒരേ രീതിയിലല്ല പ്രവർത്തിക്കുന്നത്. C++-ൽ എല്ലാ Switch പ്രസ്താവനകളെയും else if ലാഡർ ഉപയോഗിച്ച് മാറ്റിയെഴുതാം. എന്നാൽ എല്ലാ else if ലാഡറുകളും switch ഉപയോഗിച്ച് മാറ്റിയെഴുതാൻ സാധിക്കില്ല. switch

പ്രസ്താവന ഉപയോഗിച്ച് അനേകം ശാഖകൾ നടപ്പിൽ വരുത്തുന്നതിന് താഴെ പറയുന്നവ ആവശ്യമാണ്.

- നിബന്ധനകളിൽ തുല്യത പരിശോധന മാത്രമെ ഉള്ളൂ. മറ്റ് അവസരങ്ങളിൽ അതിനെ തുല്യതപ്രയോഗങ്ങളാക്കി മാറ്റിയെഴുതണം.
- എല്ലാ തുല്യതാ പ്രയോഗങ്ങളിലേയും ആദ്യത്തെ ഓപ്പറൻഡ് (operand) ഒരേ വേരിയബിളോ പ്രയോഗമോ ആയിരിക്കണം.
- ഈ പ്രയോഗങ്ങളിലെ രണ്ടാമത്തെ ഓപ്പറൻഡ് (operand) പൂർണസംഖ്യ (integer) അല്ലെങ്കിൽ ക്യാരക്ടർ കോൺസ്റ്റന്റ് ആയിരിക്കണം.

ഈ അധ്യായത്തിൽ ഇതുവരെ ചർച്ച ചെയ്ത പ്രോഗ്രാം 7.3, പ്രോഗ്രാം 7.7 എന്നിവയിലെ ശാഖകൾ മാത്രമേ `switch` ഉപയോഗിച്ച് മാറ്റിയെഴുതുവാൻ സാധിക്കുകയുള്ളൂ. പ്രോഗ്രാം 7.5-ൽ പരിശോധനാ പ്രയോഗങ്ങൾ `score/10==10`, `score/10==9`, `score/10==8` എന്നിങ്ങനെ മാറ്റം വരുത്തിയാൽ `switch` ഉപയോഗിക്കാം. ഇതുപോലെ മറ്റു സ്കോറുകൾ മാറ്റി എഴുതുക. താഴെകൊടുത്തിരിക്കുന്ന പ്രോഗ്രാംശകലം `else if` ഗോവണിക്കു പകരമായി ഉപയോഗിക്കാം.

```
switch(score/10)
{
    case 10:
    case 9: case 8: cout<< "A Grade"; break;
    case 7: case 6: cout<< "B Grade"; break;
    case 5: case 4: cout<< "C Grade"; break;
    case 3:         cout<< "D Grade"; break;
    default: cout<< "E Grade";
}
```

`switch` പ്രസ്താവന	`else if` ഗോവണി
1. അനേകം ശാഖകൾ അനുവദിക്കുന്നു.	1. അനേകം ശാഖകൾ അനുവദിക്കുന്നു.
2. തുല്യത (equality) ഓപ്പറേറ്റർ ഉള്ള നിബന്ധനകൾ മാത്രം വിലയിരുത്തുന്നു.	2. ഏതൊരു റിലേഷണൽ/ലോജിക്കൽ പ്രയോഗങ്ങളും വിലയിരുത്തുന്നു.
3. `case` സ്ഥിരാംഗം എപ്പോഴും പൂർണസംഖ്യയോ അക്ഷരമോ ആയിരിക്കണം.	3. ഫ്ളോട്ടിങ് പോയിന്റ് സ്ഥിരാംഗങ്ങളോ ഒരു പരിധിയിലുള്ള വിലകളോ നിബന്ധനകളിലുൾപ്പെടുത്താം.
4. ഒരു തുല്യതയും ലഭിക്കാത്തപ്പോൾ default പ്രസ്താവന പ്രവർത്തിക്കുന്നു.	4. ഒരു പ്രയോഗവും ശരിയായില്ലെങ്കിൽ else ബ്ലോക്ക് പ്രവർത്തിക്കുന്നു.
5. `switch` പ്രസ്താവനയിൽ നിന്നും പുറത്തു കടക്കുന്നതിന് `break` പ്രസ്താവന ആവശ്യമാണ്.	5. ഒരു ബ്ലോക്ക് പൂർത്തീകരിച്ചതിനുശേഷം പ്രോഗ്രാമിന്റെ നിയന്ത്രണം സ്വയം ബ്ലോക്കിന് പുറത്തു പോകുന്നു.
6. ഒരേ വേരിയബിളോ പ്രയോഗവോ ഒരു കൂട്ടം വിലകളുമായി തുല്യത പരിശോധിക്കുന്നതിന് കൂടുതൽ ഫലപ്രദമാണ്.	6. `switch`-നെക്കാൾ വഴക്കമുള്ളതും എളുപ്പത്തിൽ ഉപയോഗിക്കുവാൻ സാധിക്കുന്നതുമാണ്.

പട്ടിക 7.1: switch Dw else if ഗോവണിയും തമ്മിലുള്ള താരതമ്യം

switch ഉം else if ലാഡറും തമ്മിലുള്ള ഒരു താരതമ്യം പട്ടിക 7.1-ൽ കൊടുത്തിരിക്കുന്നു.

7.1.6 കണ്ടിഷണൽ ഓപ്പറേറ്റർ (Conditional operator (?:))

അധ്യായം 5- ൽ സൂചിപ്പിച്ചതുപോലെ C++ ൽ ഒരു ടെർണറി ഓപ്പറേറ്റർ ഉണ്ട്. ?ഉം :ഉം എന്നീ ചിഹ്നങ്ങൾ (ചോദ്യചിഹ്നവും കോളനും) ഉൾപ്പെടുന്ന കണ്ടിഷണൽ ഓപ്പറേറ്റർ (Conditional operator) ആണ് അത്. ഇത് ഉപയോഗിക്കുന്നതിന് മൂന്ന് ഓപ്പറന്റുകൾ ആവശ്യമാണ്. if...else പ്രസ്താവനക്ക് പകരമായി ഇതിനെ ഉപയോഗിക്കാം. ഇതിന്റെ വാക്യഘടന താഴെ കൊടുത്തിരിക്കുന്നു.

പരിശോധനാ പ്രയോഗം ? പ്രയോഗം ശരിയാകുമ്പോൾ പ്രവർത്തിക്കുന്ന കോഡ് : പ്രയോഗം തെറ്റാകുമ്പോൾ പ്രവർത്തിക്കുന്ന കോഡ് ;

```
Test expression ? True_case code : False_case code;
```

പരിശോധനാ പ്രയോഗം ഏതെങ്കിലും റിലേഷണലോ ലോജിക്കലോ ആയ പ്രയോഗം ആകാം. പ്രയോഗം ശരിയാകുമ്പോൾ പ്രവർത്തിക്കുന്ന കോഡ്, പ്രയോഗം തെറ്റാകുമ്പോൾ പ്രവർത്തിക്കുന്ന കോഡ് എന്നിവ സ്ഥിരവിലയോ, വേരിയബിളോ, പ്രയോഗമോ അല്ലെങ്കിൽ പ്രസ്താവനയോ ആകാം. ഇതിന്റെ പ്രവർത്തനം if else പ്രസ്താവനയുടെ സഹായത്തോടു കൂടി താഴെ കാണിച്ചിരിക്കുന്നു.

```
if Test experssion (പരിശോധനാ പ്രയോഗം)
    {
    True_case code: (പ്രയോഗം ശരിയാകുമ്പോൾ പ്രവർത്തിക്കുന്ന കോഡ്:)
    }
    else
    {
    False_case code: (പ്രയോഗം തെറ്റാകുമ്പോൾ പ്രവർത്തിക്കുന്ന കോഡ്;)
    }
if  (Test expression)
    {
        True_case code;
    }
    else
    {
        False_case code;
    }
```

if...else പ്രവർത്തിക്കുന്നതുപോലെയാണ് കണ്ടിഷണൽ ഓപ്പറേറ്ററും പ്രവർത്തിക്കുന്നത്. പരിശോധനാ പ്രയോഗം വിലയിരുത്തി അത് ശരിയാണെങ്കിൽ 'പ്രയോഗം ശരിയാകുമ്പോൾ പ്രവർത്തിക്കുന്ന കോഡും' (true_case code) തെറ്റാണെങ്കിൽ 'പ്രയോഗം തെറ്റാകുമ്പോൾ പ്രവർത്തിക്കുന്ന കോഡും' (False_case code) തിരഞ്ഞെടുക്കുന്നു. കണ്ടീഷണൽ ഓപ്പറേറ്ററിന്റെ പ്രവർത്തനം പ്രോഗ്രാം 7.10 ൽ വിവരിച്ചിരിക്കുന്നു.

പ്രോഗ്രാം 7.10: കണ്ടീഷണൽ ഓപ്പറേറ്റർ ഉപയോഗിച്ച് ഏറ്റവും വലിയ സംഖ്യ കണ്ടുപിടിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
int main()
{
int num1, num2;
cout << "Enter two numbers: ";
cin>> num1 >> num2 ;
(num1>num2)? cout<<num1<<" is larger" : cout<<num2<<" is larger";
return 0;
}
```

ഈ പ്രോഗ്രാമിലെ `return 0` പ്രസ്താവനയ്ക്ക് മുമ്പുള്ള പ്രസ്താവനയിൽ കണ്ടീഷണൽ ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നതുകൊണ്ട് അതിനെ കണ്ടീഷണൽ പ്രസ്താവന എന്നു വിളിക്കുന്നു. ഈ പ്രസ്താവനയെ താഴെയുള്ള കോഡ് ശകലം ഉപയോഗിച്ച് മാറ്റി എഴുതാവുന്നതാണ്.

```
int big = (n1>n2) ? n1 : n2;
cout<< big << "is larger";
```

പരിശോധനാ പ്രയോഗം ശരിയാണെങ്കിൽ `num1`-ന്റെ വിലയും തെറ്റാണെങ്കിൽ `num2` ന്റെ വിലയും ആയിരിക്കും `big` ലേക്ക് ശേഖരിക്കുക. ഇവിടെ കണ്ടീഷണൽ ഓപ്പറേറ്റർ ഉപയോഗിച്ചാണ് ഒരു കണ്ടീഷണൽ പ്രയോഗം ഉണ്ടാക്കിയിരിക്കുന്നത്. ഈ പ്രയോഗത്തിൽ നിന്നും ലഭിക്കുന്ന വില `big` ലേക്ക് ശേഖരിക്കുന്നു.

സ്വയം വിലയിരുത്താം

1. 1 മുതൽ 12 വരെയുള്ള സംഖ്യകൾ ഇൻപുട്ട് ചെയ്ത് അതിനനുസൃതമായ മാസത്തിന്റെ പേര് പ്രദർശിപ്പിക്കുന്നതിനുള്ള പ്രോഗ്രാം എഴുതുക. (1 ആണെങ്കിൽ January, 2 ആണെങ്കിൽ February എന്നിങ്ങനെ)
2. `switch` പ്രസ്താവനയ്ക്കകത്തുള്ള `break` പ്രസ്താവനയുടെ പ്രാധാന്യം എന്താണ്?
3. താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന `if..else` ഉപയോഗിച്ച് മാറ്റിയെഴുതുക.

```
result = (mark>30)? 'P' : 'F';
```

7.2. ആവർത്തന പ്രസ്താവനകൾ (Interation statements)

അധ്യായം 4-ൽ നാം ചർച്ച ചെയ്ത ചില പ്രശ്നങ്ങളുടെ ഉത്തരങ്ങളിൽ ആവർത്തന സ്വഭാവമുള്ള പ്രവർത്തികൾ അടങ്ങിയിട്ടുണ്ട്. പ്രോഗ്രാമുകൾ എഴുതുമ്പോൾ ഒന്നോ അതിലധികമോ പ്രസ്താവനകളെ പല തവണ പ്രവർത്തിപ്പിക്കുന്നതിനായി ഭാഷയുടെ പ്രത്യേക രൂപകൽപ്പനകൾ നാം ഉപയോഗിക്കുന്നു. ഇത്തരം രൂപകൽപ്പനകളെ **ആവർത്തന പ്രസ്താവനകൾ** (Iteration statements) അല്ലെങ്കിൽ **ലൂപ്പിങ് പ്രസ്താവനകൾ** എന്നു വിളിക്കുന്നു. C++-ൽ മൂന്ന് തരം ആവർത്തന പ്രസ്താവനകൾ ഉണ്ട്. ഒരു നിബന്ധന ശരിയാകുമ്പോൾ ഒരു കൂട്ടം പ്രസ്താവനകൾ ആവർത്തിച്ച് പ്രവർത്തിപ്പിക്കുവാൻ ഇവ അനുവദിക്കുന്നു.

ലൂപ്പ് എന്ന ആശയം നിത്യജീവിതത്തിൽ നാം പ്രയോഗിക്കാറുണ്ട്. നമുക്ക് ഒരു സാഹചര്യം പരിഗണിക്കാം. പരീക്ഷയിൽ A+ ഗ്രേഡ് ലഭിക്കുന്ന എല്ലാ വിദ്യാർത്ഥികൾക്കും നിങ്ങളുടെ ക്ലാസ് ടീച്ചർ ഒരു സമ്മാനം തരുമെന്ന് പ്രഖ്യാപിച്ചു എന്ന് വിചാരിക്കുക. സമ്മാനം പൊതിയാനുള്ള ചുമതല നിങ്ങളെ ഏൽപ്പിക്കുന്നു. സമ്മാനം പൊതിയേണ്ടതെങ്ങനെയെന്ന് താഴെ കൊടുത്ത രീതിയിൽ ടീച്ചർ വിശദീകരിക്കുന്നു.

ഘട്ടം 1 : സമ്മാനം എടുക്കുക.

ഘട്ടം 2 : പൊതിയാനുള്ള പേപ്പർ മുറിക്കുക.

ഘട്ടം 3 : സമ്മാനം പൊതിയുക.

ഘട്ടം 4 : റിബൺ ഉപയോഗിച്ച് കവർ കെട്ടുക.

ഘട്ടം 5 : കാർഡിൽ പേരെഴുതി സമ്മാനത്തിന് മുകളിൽ ഒട്ടിക്കുക.

പരീക്ഷയിൽ 30 വിദ്യാർഥികൾക്ക് A+ ഗ്രേഡ് ഉണ്ടെങ്കിൽ ഇതേ പ്രവർത്തി 30 തവണ നിങ്ങൾ ആവർത്തിക്കേണ്ടതുണ്ട്. സമ്മാനം പൊതിയുന്ന ഈ പ്രവർത്തി 30 തവണ ആവർത്തിക്കുന്നതിന് താഴെ കൊടുത്ത രീതിയിൽ നിർദ്ദേശങ്ങൾ പുനഃക്രമീകരിക്കാം.

താഴെ കൊടുത്ത ഘട്ടങ്ങൾ 30 തവണ ആവർത്തിക്കുക

{

അടുത്ത സമ്മാനം എടുക്കുക.

പൊതിയാനുള്ള പേപ്പർ മുറിക്കുക.

സമ്മാനം പൊതിയുക.

റിബൺ ഉപയോഗിച്ച് കവർ കെട്ടുക.

കാർഡിൽ പേരെഴുതി സമ്മാനത്തിന് മുകളിൽ ഒട്ടിക്കുക.

}

ഇനി വേറൊരു ഉദാഹരണമെടുക്കാം. കമ്പ്യൂട്ടർ ആപ്ലിക്കേഷൻ വിഷയത്തിൽ ലഭിച്ച സ്കോറുകളുടെ ക്ലാസ് ശരാശരി നമുക്ക് കണ്ടുപിടിക്കണമെന്ന് കരുതുക. അതിനായി താഴെ പറയുന്ന ഘട്ടങ്ങളിലൂടെ കടന്നു പോകണം.

ആകെ-സ്കോറിന് പ്രാരംഭ വിലയായി പൂജ്യം കൊടുക്കുക.

താഴെ പറയുന്ന ഘട്ടങ്ങൾ ആദ്യത്തെ വിദ്യാർഥിക്ക് മുതൽ അവസാനത്തെ ആൾ വരെ ആവർത്തിക്കുക.

{

വിദ്യാർത്ഥിയുടെ സ്കോർ ആകെ-സ്കോറിനോട് കൂട്ടുക.

അടുത്ത വിദ്യാർത്ഥിയുടെ സ്കോർ എടുക്കുക.

}

ശരാശരി = ആകെ-സ്കോർ/ക്ലാസ്സിലെ ആകെ വിദ്യാർത്ഥികളുടെ എണ്ണം

ഈ രണ്ടു ഉദാഹരണങ്ങളിലും ചില ഘട്ടങ്ങൾ നാം പല തവണ ചെയ്യുന്നു. പ്രക്രിയ എത്ര തവണ ആവർത്തിച്ചു എന്നറിയുന്നതിന് നാം ഒരു കൗണ്ടർ (counter) ഉപയോഗിക്കുന്നു.

പ്രവർത്തനം തുടരണമോ വേണ്ടയോ എന്ന് കൗണ്ടറിന്റെ വില തീരുമാനിക്കുന്നു. നിബന്ധനയ്ക്ക് വിധേയമായി ലൂപ്പുകൾ പ്രവർത്തിക്കുന്നതിനാൽ കൗണ്ടർ പോലുള്ള വേരിയബിൾ ലൂപ്പ് നിർമ്മിക്കുന്നതിന് ഉപയോഗിക്കുന്നു. ഈ വേരിയബിൾ പൊതുവെ ലൂപ്പ് നിയന്ത്രണവേരിയബിൾ (Loop control variable) എന്നറിയപ്പെടുന്നു. എന്തുകൊണ്ടെന്നാൽ യഥാർത്ഥത്തിൽ ഇതാണ് ലൂപ്പിന്റെ പ്രവർത്തനത്തെ നിയന്ത്രിക്കുന്നത്. അധ്യായം 3-ൽ ഒരു ലൂപ്പിന്റെ 4 ഘടകങ്ങളെക്കുറിച്ച് നാം ചർച്ച ചെയ്തു. നമുക്ക് അതൊന്ന് ഓർത്തെടുക്കാം.

1. **പ്രാരംഭ വില നൽകൽ (Initialisation)** : ലൂപ്പിലേക്ക് പ്രവേശിക്കുന്നതിനു മുമ്പ് അതിന്റെ നിയന്ത്രണ വേരിയബിളിന് പ്രാരംഭ വില നൽകണം. അങ്ങനെ ലൂപ്പ് നിയന്ത്ര വേരിയബിളിന് അതിന്റെ ആദ്യത്തെ വില ലഭിക്കും. പ്രാരംഭ വില നൽകുന്ന പ്രസ്താവന ലൂപ്പിന്റെ തുടക്കത്തിൽ മാത്രമേ പ്രവർത്തിക്കുന്നുള്ളൂ.

2. **പരിശോധനാ പ്രയോഗം (Test Expression)** : ഇത് ഒരു റിലേഷണൽ അല്ലെങ്കിൽ ലോജിക്കൽ പ്രയോഗമാണ്. ഇതിന്റെ വില ശരി അല്ലെങ്കിൽ തെറ്റ് ആയിരിക്കും. ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിക്കണോ വേണ്ടയോ എന്ന് ഇത് തീരുമാനിക്കുന്നു. പരിശോധനാ പ്രയോഗം ശരിയാണെങ്കിൽ ലൂപ്പ് പ്രവർത്തിക്കുന്നു. അല്ലെങ്കിൽ അത് പ്രവർത്തിക്കില്ല.

3. **പരിഷ്കരിക്കൽ പ്രസ്താവന (Updation Statement)** : പരിഷ്കരിക്കൽ പ്രസ്താവന ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിന്റെ വിലയിൽ മാറ്റം വരുത്തുന്നു. ഈ പ്രസ്താവന അടുത്ത ആവർത്തനത്തിന് മുന്നേ പ്രവർത്തിക്കുന്നു.

4. **ലൂപ്പിന്റെ ചട്ടക്കൂട് (Body of loop)** : ആവർത്തിക്കപ്പെടേണ്ട പ്രസ്താവനകൾ ഉപയോഗിച്ച് ലൂപ്പിന്റെ ചട്ടക്കൂട് രൂപപ്പെടുത്തുന്നു. ഇതിൽ ഒന്നോ അതിലധികമോ പ്രസ്താവനകൾ ഉണ്ടായിരിക്കും. ലൂപ്പുകളെ പൊതുവെ ആഗമന നിയന്ത്രണ ലൂപ്പുകൾ (Entry controlled loop) എന്നും ബഹിർഗമന നിയന്ത്രണ ലൂപ്പുകൾ (Exit controlled loop) എന്നും തരംതിരിച്ചിരിക്കുന്നു എന്ന് അധ്യായം 3-ൽ നാം പഠിച്ചു. C++ ൽ മൂന്നുതരം ലൂപ്പ് പ്രസ്താവനകൾ ഉണ്ട്: `while loop`, `for loop`, `do-while loop`. ഓരോന്നിന്റേയും പ്രവർത്തനം വിശദമായി നമുക്ക് ചർച്ച ചെയ്യാം.

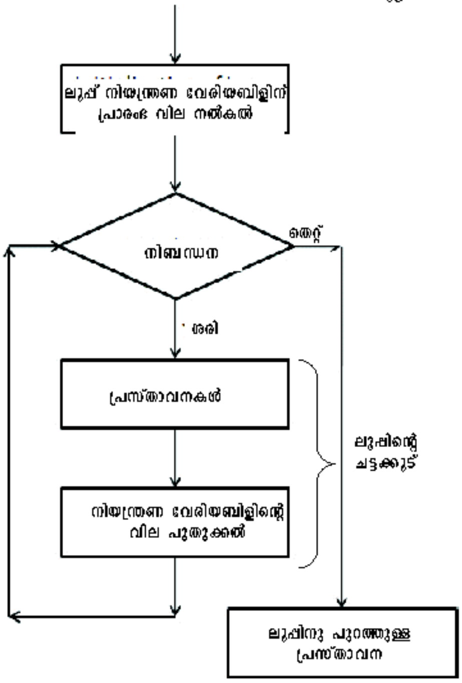
7.2.1 while പ്രസ്താവന (while statement)

`while` ലൂപ്പ് ഒരു ആഗമന നിയന്ത്രണ ലൂപ്പ് ആണ്. നിബന്ധന (Condition) ആദ്യം പരിശോധിക്കുകയും അത് ശരിയാണെങ്കിൽ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിക്കുകയും ചെയ്യുന്നു. അതായത് നിബന്ധന ശരിയാകുന്നിടത്തോളം ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിക്കും. `while` ലൂപ്പിന്റെ വാക്യഘടന ഇതാണ്.

```
നിയന്ത്രണ വേരിയബിളിന്റെ പ്രാരംഭ വില നൽകൽ;
    while(പരിശോധനാ പ്രയോഗം)
    {
        ലൂപ്പിന്റെ ചട്ടക്കൂട്;
        ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിനെ പുതുക്കൽ;
    }
intialisation of loop control variable;
while (test expression)
```

```
{
    body of the loop;
    updation of loop control variable;
}
```

ഇവിടെ പരിശോധനാ പ്രയോഗം നിബന്ധന നിർവചിക്കുകയും അത് ലൂപ്പിനെ നിയന്ത്രിക്കുകയും ചെയ്യുന്നു. ലൂപ്പിന്റെ ചട്ടക്കൂട് ഒരു പ്രസ്താവനയോ ഒന്നിലധികം പ്രസ്താവനകളോ അല്ലെങ്കിൽ പ്രസ്താവനകളില്ലാതെയോ ആകാം. ആവർത്തിച്ചു പ്രവർത്തിക്കുന്നതിനുള്ള ഒരു കൂട്ടം പ്രസ്താവനകളാണ് ലൂപ്പിന്റെ ചട്ടക്കൂട്. ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിന്റെ വില വ്യത്യാസപ്പെടുത്തുന്ന പ്രസ്താവനയാണ് പരിഷ്കരിക്കൽ പ്രസ്താവന. ഒരു while ലൂപ്പിൽ ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിന് ലൂപ്പ് തുടങ്ങുന്നതിനുമുമ്പ് പ്രാരംഭവില നൽകുകയും ലൂപ്പ് ചട്ടക്കൂടിനുള്ളിൽ വച്ച് അതു പുതുക്കുകയും ചെയ്യുന്നു. ഒരു while ലൂപ്പിന്റെ പ്രവർത്തനം ചിത്രം 7.3-ലെ ഫ്ളോചാർട്ടിൽ വിവരിച്ചിരിക്കുന്നു.



ചിത്രം 7.3. while ലൂപ്പിന്റെ പ്രവർത്തനം

നിയന്ത്രണ വേരിയബിളിന് പ്രാരംഭ വില നൽകുകയാണ് ആദ്യം ചെയ്യുന്നത്. പിന്നീട് പരിശോധനാ പ്രയോഗം വിലയിരുത്തുന്നു. അത് ശരിയാണ് എങ്കിൽ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിക്കുന്നു. അതുകൊണ്ടാണ് while ലൂപ്പിനെ ആഗമന നിയന്ത്രണ ലൂപ്പ് എന്ന് വിളിക്കുന്നത്. ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിക്കുന്നതിനൊപ്പം ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിന്റെ വിലയും പുതുക്കുന്നു. ലൂപ്പ് ചട്ടക്കൂട്ടിന്റെ പ്രവർത്തനം കഴിഞ്ഞതിനുശേഷം പരിശോധനാപ്രയോഗം വീണ്ടും വിലയിരുത്തുന്നു. നിബന്ധന ശരിയായിരിക്കുന്നിടത്തോളം ഈ പ്രക്രിയ തുടരുന്നു. while ലൂപ്പിന്റെ പ്രവർത്തനം വിവരിക്കുന്നതിനുള്ള ഒരു കോഡ് ശകലം നമുക്ക് ഇപ്പോൾ പരിഗണിക്കാം.

```
int k=1;
while(k<=3)
{
    cout << k << '\t';
    ++k;
}
cout << "\n Program Ends";
```

ലൂപ്പിന് മുമ്പേ പ്രാരംഭ വില നൽകുന്നു.

പരിശോധനാ പ്രയോഗം

ലൂപ്പിന്റെ ചട്ടക്കൂട്

ലൂപ്പ് ചട്ടക്കൂടിനകത്തെ പുതുക്കൽ പ്രസ്താവന

ഈ കോഡ് ശകലത്തിൽ k എന്ന ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിന് 1 എന്ന വില ആദ്യം നൽകിയിരിക്കുന്നു. പിന്നീട് k<=3 എന്ന പരിശോധനാ പ്രയോഗമായ വിലയിരുത്തുന്നു. ഇത് ശരിയായതു കൊണ്ട് ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിക്കുന്നു. അതായത് k-യുടെ വിലയായ 1 സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുന്നു. അതിനുശേഷം പരിഷ്കരിക്കൽ പ്രസ്താവനയായ (update statement) ++k പ്രവർത്തിച്ച് k യുടെ വില 2 ആയി മാറുകയും ചെയ്യുന്നു. നിബന്ധന (k<=3) ഒന്നുകൂടി പരിശോധിച്ച് ശരിയാണെന്ന് കണ്ടെത്തുകയും ചെയ്തു. പ്രോഗ്രാമിന്റെ നിയന്ത്രണം ലൂപ്പിനകത്ത് പ്രവേശിച്ച് k യുടെ വില 2 എന്ന് സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുന്നു. വീണ്ടും പരിഷ്കരിക്കൽ പ്രസ്താവന ആവർത്തിക്കുകയും k യുടെ വില 3 ആകുകയും ചെയ്യുന്നു. നിബന്ധന ഇപ്പോഴും ശരിയായതിനാൽ ലൂപ്പ് പ്രവർത്തിച്ച് 3 എന്ന് സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുന്നു. k യുടെ വില വീണ്ടും പരിഷ്കരിച്ച 4 ആവുകയും ഇപ്പോൾ പരിശോധന പ്രയോഗത്തിനന്റെ ഫലം തെറ്റാവുകയും ചെയ്യുന്നു. നിയന്ത്രണം ലൂപ്പിന് പുറത്തേക്ക് വരുകയും while ലൂപ്പിന് പുറത്തുള്ള അടുത്ത പ്രസ്താവന പ്രവർത്തിക്കുകയും ചെയ്യുന്നു. ചുരുക്കത്തിൽ കോഡിന്റെ ഔട്ട്പുട്ട് താഴെ കൊടുത്തിരിക്കുന്നത് പോലെയായിരിക്കും.

```
1     2     3
Program ends
```

k-യുടെ പ്രാരംഭ വില 5 ആണെങ്കിൽ എന്ത് സംഭവിക്കുമെന്ന് സങ്കൽപ്പിക്കുക? ആദ്യം വിലയിരുത്തുമ്പോൾ തന്നെ പരിശോധന പ്രയോഗം തെറ്റായതിനാൽ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിക്കുകയില്ല. ലൂപ്പിന്റെ ചട്ടക്കൂടിലെക്കുള്ള പ്രവേശനം while loop നിയന്ത്രിക്കുന്നുവെന്ന് ഇത് വ്യക്തമായി കാണിക്കുന്നു.

ആദ്യത്തെ 10 എണ്ണൽ സംഖ്യകളെ while loop ഉപയോഗിച്ച് പ്രിന്റ് ചെയ്യുന്നതിനുള്ള ഒരു പ്രോഗ്രാം നമുക്ക് നോക്കാം.

പ്രോഗ്രാം 7.11: ആദ്യത്തെ 10 എണ്ണൽ സംഖ്യകൾ പ്രിന്റ് ചെയ്യുന്നതിന്

```
#include<iostream>
using namespace std;
int main()
{
    int n = 1;
    while(n <= 10)
    {
        cout<< n << "  ";
        ++n;
    }
    return 0;
}
```

ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിന് പ്രാരംഭ വില നൽകുന്നു (int n = 1;)

പരിശോധനാ പ്രയോഗം (while(n <= 10))

ലൂപ്പിന്റെ ചട്ടക്കൂട്

ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിന്റെ വില പരിഷ്കരിക്കുന്നു (++n;)

പ്രോഗ്രാം 7.11 ന്റെ ഔട്ട്പുട്ട് താഴെ കൊടുത്തിരിക്കുന്ന പോലെ ആയിരിക്കും.

```
1   2   3   4   5   6   7   8   9   10
```

20 വരെയുള്ള ഇരട്ട സംഖ്യകളുടെ തുക കണ്ട്പിടിക്കുന്നതിന് പ്രോഗ്രാം 7.12 while ലൂപ്പ് ഉപയോഗിക്കുന്നു. ലൂപ്പ് വേരിയബിളിന്റെ വില ഏത് ഓപ്പറേഷനുപയോഗിച്ചും പരിഷ്കരിക്കാമെന്ന് ഈ പ്രോഗ്രാം കാണിക്കുന്നു.

പ്രോഗ്രാം 7.12: 20 വരെയുള്ള ഇരട്ടസംഖ്യകളുടെ തുക കണ്ടുപിടിക്കുന്നതിന്

```
#include<iostream>
using namespace std;
int main()
{
      int i, sum = 0;
      i = 2;
      while( i<= 20)
      {
            sum = sum + i;
            i = i + 2;
      }
      cout<<"\nThe sum of even numbers up to 20 is: "<<sum;
      return 0;
}
```

നിലവിലുള്ള വിലയോട് രണ്ട് കൂട്ടി ച്ചേർത്തു കൊണ്ട് ലൂപ് നിയന്ത്രണ വേരിയബിന്റെ വില പരിഷ്കരി ക്കുന്നു.

പ്രോഗ്രാം 7.12 ന്റെ ഔട്ട്പുട്ട് താഴെ കൊടുത്തിരിക്കുന്നു.

```
The sum of even numbers up to 20 is: 110
```

നമുക്കു ചെയ്യാം

1. *100 - നും 200 - നും ഇടയിലുള്ള എല്ലാ ഒറ്റ സംഖ്യകളും പ്രദർശിപ്പിക്കുവാനായി പ്രോഗ്രാം 7.11 പരിഷ്കരിക്കുക.*
2. *പ്രോഗ്രാം 7.12 പരിഷ്കരിച്ച് ആദ്യത്തെ N എണ്ണൽ സംഖ്യകളുടെ ശരാശരി കണ്ടുപിടിക്കുക.*

while പ്രസ്താവനയിലെ പരിശോധനാ പ്രയോഗത്തിന് ശേഷം നാം ഒരു അർധവിരാമം (;) ഇട്ടാൽ വാക്യഘടനയിൽ തെറ്റൊന്നുമില്ല. എന്നാൽ അതിനുശേഷമുള്ള ബ്രാക്കറ്റുകളിലെ പ്രസ്താവനകളെ ലൂപ് ചട്ടക്കൂടായി പരിഗണിക്കുന്നില്ല. പരിശോധനാ പ്രയോഗം ശരിയാണെങ്കിൽ while ലൂപ്പിനുശേഷമുള്ള കോഡ് പ്രവർത്തിക്കുകയുമില്ല പ്രോഗ്രാം അവസാനിക്കുകയുമില്ല എന്നതാണ് ഏറ്റവും പരിതാപകരമായ അവസ്ഥ. ഇത് ഒരു അനന്തമായ ലൂപ്പിന് കാരണമാകുന്നു

7.2.2 for പ്രസ്താവന (for statement)

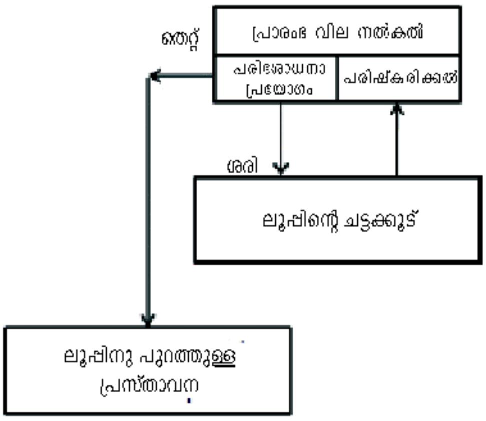
for ലൂപ്പും C++-ലെ ഒരു ആഗമന നിയന്ത്രണ ലൂപ്പ് ആണ്. ലൂപ്പിലെ ഘടകങ്ങളായ പ്രാരംഭ വില നൽകൽ, പരിശോധന പ്രയോഗം, പരിഷ്കരിക്കൽ പ്രസ്താവന എന്നിവ ഒരുമിച്ചാണ് for പ്രസ്താവനയിൽ നൽകിയിരിക്കുന്നത്. അതുകൊണ്ട് പ്രോഗ്രാം ഒതുക്കമുള്ളതായി തീരുന്നു. വാക്യഘടന ഇതാണ്:

```
for (പ്രാരംഭവില നൽകൽ; പരിശോധന പ്രയോഗം; പരിഷ്കരിക്കൽ പ്രസ്താവന)
{
     ലൂപ്പിന്റെ ചട്ടക്കൂട്;
}
for (initialisation; test expression; update statement)
{
     body-of-the-loop;
{
```

for ലൂപ്പിന്റെ പ്രവർത്തനം while ലൂപ്പിന്റേതുപോലെയാണ്. while ലൂപ്പിന്റെ ഫ്ളോചാർട്ട് for ലൂപ്പിന്റെ പ്രവർത്തനം വിശദമാക്കുന്നതിന് ഉപയോഗിക്കാവുന്നതാണ്.

for ലൂപ്പിൽ മൂന്നു ഘടകങ്ങളും ഒരുമിച്ചു വന്നതിനാൽ എണ്ണുന്ന (Counting) സാഹചര്യങ്ങളിൽ ഈ പ്രസ്താവന ഉപയോഗിക്കുന്നത് അഭികാമ്യമാണ്.

ചിത്രം 7.4 ൽ കൊടുത്തിരിക്കുന്ന ഫ്ളോചാർട്ട് സാധാരണയായി for പ്രസ്താവനയുടെ പ്രവർത്തനം കാണിക്കുന്നതിന് ഉപയോഗിക്കുന്നു.



ചിത്രം 7.4: ഫോർലൂപ്പിന്റെ പ്രവർത്തനം.

തുടക്കത്തിൽ, പ്രാരംഭ വില നൽകൽ നടക്കുന്നു, തുടർന്ന് പരിശോധനാ പ്രയോഗം വിലയിരുത്തുന്നു. ഇതിന്റെ ഫലം ശരിയാണെങ്കിൽ ലൂപ്പ് ചട്ടക്കൂട് പ്രവർത്തിക്കുന്നു. അല്ലെങ്കിൽ പ്രോഗ്രാം നിയന്ത്രണം ലൂപ്പിനു പുറത്തേക്കു പോകുന്നു. ലൂപ്പ് ചട്ടക്കൂടിന്റെ പ്രവർത്തനത്തിനുശേഷം പരിഷ്കരിക്കൽ പ്രയോഗം പ്രവർത്തിക്കുകയും പരിശോധനാ പ്രയോഗം വീണ്ടും വിലയിരുത്തുകയും ചെയ്യുന്നു. പരിശോധനാ പ്രയോഗം തെറ്റാവുന്നതുവരെ ഈ മൂന്നു ഘട്ടങ്ങളും (പരിശോധന, ചട്ടക്കൂട്, പരിഷ്കരിക്കൽ) തുടർന്നുകൊണ്ടേയിരിക്കും.

പ്രോഗ്രാം 7.11 ൽ ഉപയോഗിച്ചിരിക്കുന്ന ലൂപ്പ് ശകലത്തെ for ലൂപ്പ് ഉപയോഗിച്ച് താഴെ കാണും വിധം മാറ്റി എഴുതാം.

```
for (n=1; n<=10; ++n)
     cout << n << "  ";
```

while ലൂപ്പിലേതുപോലെ തന്നെ ഈ കോഡ് പ്രവർത്തിക്കുന്നു.

നമുക്കു ചെയ്യാം

തൊട്ട് മുമ്പ് സൂചിപ്പിച്ച for ലൂപ്പിന്റെ പ്രവർത്തനക്രമത്തിലെ ഒന്നും രണ്ടും ഘട്ടങ്ങൾ താഴെ കൊടുത്തിരിക്കുന്നു. ബാക്കി ഘട്ടങ്ങൾ എഴുതുക.

ഘട്ടം 1 : n= 1, നിബന്ധന ശരിയാണ്, ഒന്ന് പ്രദർശിപ്പിക്കുന്നു, n ന്റെ വില 2 ആകുന്നു.

ഘട്ടം 2 : നിബന്ധനശരിയാണ്, 2 പ്രദർശിപ്പിക്കുന്നു, n ന്റെ വില 3 ആകുന്നു.

ഘട്ടം 3 : ....

for ലൂപ്പ് ഉപയോഗിച്ച് ഒരു സംഖ്യയുടെ ഫാക്റ്റോറിയൽ കണ്ടുപിടിക്കാനുള്ള പ്രോഗ്രാം നമുക്കു എഴുതാം. N എന്ന സംഖ്യയുടെ ഫാക്റ്റോറിയൽ എന്നത് N! എന്ന് സൂചിപ്പിക്കുന്നു. ഇത് ആദ്യത്തെ N എണ്ണൽ സംഖ്യകളുടെ ഗുണനഫലമാണ്. ഉദാഹരണത്തിന് 5 ന്റെ ഫാക്റ്റോറിയൽ (5!)കണക്കാക്കുന്നത് $1 \times 2 \times 3 \times 4 \times 5 = 120$ എന്നാണ്.

പ്രോഗ്രാം 7.13: for ലൂപ് ഉപയോഗിച്ച് ഒരു സംഖ്യയുടെ ഫാക്ടോറിയൽ കണ്ടുപിടിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
int main()
{   int n, i;
    long fact=1;
    cout<<"Enter the number: ";
    cin>>n;
    for (i=1; i<=n; ++i)
        fact = fact * i;
    cout << "Factorial of " << n << " is " << fact;
    return 0;
}
```

പ്രാരംഭ വില നൽകൽ, പരിശോധനാ പ്രയോഗം, പരിഷ്കരിക്കൽ പ്രസ്താവന

ലൂപ് ചട്ടക്കൂട്

പ്രോഗ്രാം 7.13 ന്റെ ഒരു മാതൃക ഔട്ട്പുട്ട് താഴെ കൊടുത്തിരിക്കുന്നു.

```
Enter the number:  6
Factorial of 6 is 720
```

കമ്പ്യൂട്ടർ ആപ്ലിക്കേഷൻസ് എന്ന വിഷയത്തിലെ സ്കോറുകളുടെ ശരാശരി കാണുന്നതിനുള്ള മറ്റൊരു പ്രോഗ്രാമാണ് താഴെ കൊടുത്തിരിക്കുന്നത് പ്രോഗ്രാം 7.14-ൽ n നു(കുട്ടികളുടെ എണ്ണം) വില സ്വീകരിക്കുകയും പിന്നീട് ഓരോ വിദ്യാർഥികളേയും സ്കോർ ഇൻപുട്ടായി സ്വീകരിച്ച് ശരാശരി സ്കോർ പ്രിന്റ് ചെയ്യുന്നു.

പ്രോഗ്രാം 7.14 n വിദ്യാർഥികളുടെ ശരാശരി സ്കോർ കണ്ടുപിടിക്കുന്നതിന്

```
#include<iostream>
using namespace std;
int main()
{
    int i, sum, score, n;
    float avg;
    cout << "How many students? ";
    cin >> n ;
    for( i=1, sum=0;  i<=n; ++i)
    {
        cout << "Enter the score of student " << i << ": ";
        cin >> score;
```

പ്രാരംഭ വില നൽകുന്ന ഭാഗത്ത് രണ്ട് പ്രയോഗങ്ങൾ അടങ്ങിയിരിക്കുന്നു.

```
        sum = sum + score;
    }
    avg = (float)sum / n;
    cout << "Class Average: " << avg;
    return 0;
}
```

സ്പഷ്ടമായ ജനം മാറ്റം

പ്രോഗ്രാം 7.14 ന്റെ ഒരു മാതൃക ഔട്ട്പുട്ട് താഴെ കൊടുത്തിരിക്കുന്നു.

```
How many students? 5
Enter the score of student 1: 45
Enter the score of student 2: 50
Enter the score of student 3: 52
Enter the score of student 4: 34
Enter the score of student 5: 55
Class Average: 47.2
```

for ലൂപ്പ് ഉപയോഗിക്കുമ്പോൾ ചില കാര്യങ്ങൾ ശ്രദ്ധിക്കേണ്ടതുണ്ട്. തന്നിരിക്കുന്ന നാലു കോഡ് ശകലങ്ങൾ ഈ പ്രത്യേക സാഹചര്യത്തിൽ വിശദീകരിക്കുന്നു. കോഡിൽ ഉപയോഗിച്ചിട്ടുള്ള എല്ലാ വേരിയബിളുകളും int ഡാറ്റാ തരത്തിലുള്ളതാണ് എന്ന് കരുതുക.

കോഡ് ശകലം 1:

```
for (n=1; n<5; n++);
    cout<<n;
```

for പ്രസ്താവനയുടെ ബ്രായ്ക്കറ്റ് കഴിഞ്ഞ് ഒരു അർധവിരാമം കാണപ്പെടുന്നു. ഇത് വാക്യഘടനയിലെ തെറ്റ് (syntax error) അല്ല. ഇതിന്റെ ഔട്ട്പുട്ട് നിങ്ങൾക്ക് പ്രവചിക്കാൻ കഴിയുമോ? 5 ആണെങ്കിൽ നിങ്ങൾ പറഞ്ഞത് ശരിയാണ്. ഈ ലൂപ്പിന് ചട്ടക്കൂട് ഇല്ല. പക്ഷേ ഇതിന്റെ പ്രവർത്തനം സാധാരണപോലെ പൂർത്തികരിക്കുന്നു. പ്രാരംഭവില നൽകുന്ന പ്രസ്താവന **n** ന് 1 എന്ന വില നൽകുകയും നിബന്ധന വിലയിരുത്തുമ്പോൾ ശരിയാവുകയും ചെയ്യുന്നു. അവിടെ ലൂപ്പ് ചട്ടക്കൂട് ഇല്ലാത്തതിനാൽ പരിഷ്ക്കരിക്കൽ പ്രസ്താവന പ്രവർത്തിക്കുകയും **n** ന്റെ വില 5 ആകുന്നതുവരെ ഈ പ്രവർത്തനം തുടരുകയും ചെയ്യുന്നു. ഈ സന്ദർഭത്തിൽ നിബന്ധന വിലയിരുത്തി തെറ്റാവുകയും പ്രോഗ്രാമിന്റെ നിയന്ത്രണം ലൂപ്പിൽ നിന്നും പുറത്തു വരികയും ചെയ്യുന്നു. ഔട്ട്പുട്ട് പ്രസ്താവന പ്രവർത്തിക്കുമ്പോൾ സ്ക്രീനിൽ 5 എന്ന് പ്രദർശിപ്പിക്കുന്നു.

കോഡ് ശകലം 2:

```
for (n=1; n<5; )
    cout<<n;
```

ഈ കോഡിൽ പരിഷ്ക്കരിക്കൽ പ്രസ്താവന (update expression) ഇല്ല. ഇത് കോഡിന്റെ വാക്യഘടനയിൽ തെറ്റ് ഉണ്ടാക്കുന്നില്ല. പക്ഷെ ലൂപ്പ് പ്രവർത്തിക്കുമ്പോൾ ഒരിക്കലും അവസാനിക്കുന്നില്ല. 1 എന്ന സംഖ്യ അനന്തമായി പ്രദർശിപ്പിക്കുന്നു. ഇതിനെ നമുക്ക് അനന്തമായ ലൂപ്പ് (infinite loop) എന്നു വിളിക്കാം.

കോഡ് ശകലം 3:

```
for ( ; n<5; n++)
    cout<<n;
```

ഈ കോഡിന്റെ ഔട്ട്പുട്ട് പ്രവചിക്കുവാൻ സാധ്യമല്ല. കാരണം നിയന്ത്രണവേരിയബിളിന് (control variable) പ്രാരംഭ വില (initialisation) നൽകിയിട്ടില്ല. അതിനാൽ നിയന്ത്രണ വേരിയബിൾ **n** - ന് ചില പൂർണസംഖ്യകൾ കിട്ടുന്നു. ചിലപ്പോൾ അത് 5 - നെക്കാൾ കുറവാണെങ്കിൽ നിബന്ധന (condition) തെറ്റാവുന്നതുവരെ ചട്ടക്കൂട്

പ്രവർത്തിക്കും. **n** ന്റെ തനത് വില 5-ഓ അതിൽ കൂടുതലോ ആണെങ്കിൽ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിക്കാതെ തന്നെ ലൂപ്പ് അവസാനിക്കുന്നു.

കോഡ് ശകലം 4:

```
for (n=1; ; n++)
    cout<<n;
```

മുകളിൽ കൊടുത്തിരിക്കുന്ന കോഡിൽ പരിശോധന പ്രയോഗം (test expression) നൽകിയിട്ടില്ല. ഇത്തരം ഘട്ടത്തിൽ പരിശോധനാ പ്രയോഗത്തിന്റെ ഫലം ശരിയായി എടുക്കുകയും ലൂപ്പ് അനന്തമായി മാറുകയും ചെയ്യുന്നു.

മുകളിൽ കൊടുത്തിരിക്കുന്ന നാലു കോഡ് ശകലങ്ങളും സൂചിപ്പിക്കുന്നത് **for** ലൂപ്പിലെ എല്ലാ ഘടകങ്ങളും നിർബന്ധമില്ല എന്നാണ്. എന്നാൽ **while, do...while** പ്രസ്താവനകളുടെ കാര്യം ഇങ്ങനെയല്ല. ഈ രണ്ടു ലൂപ്പുകൾക്കും പരിശോധന പ്രയോഗങ്ങൾ നിർബന്ധമാണ്. എന്നാൽ മറ്റു ഘടകങ്ങൾ നിർബന്ധമില്ല. എന്നാൽ ഔട്ട്പുട്ട് സംബന്ധിച്ച് ജാഗ്രത പുലർത്തണം.

പ്രോഗ്രാം 7.14-ൽ പ്രാരംഭ വില നൽകുന്ന പ്രസ്താവനയിൽ ഒരു കോമ ഉപയോഗിച്ച് വേർതിരിച്ച രണ്ട് പ്രയോഗങ്ങൾ (`i=1, sum=0`) അടങ്ങിയിരിക്കുന്നു. `i, sum` എന്നീ വേരിയബിളുകൾക്ക് അവയുടെ ആദ്യ വിലയായ `0, 1` യഥാക്രമം കിട്ടുന്നു. `i<=n` എന്ന പരിശോധന പ്രയോഗം വിലയിരുത്തുകയും അത് ശരിയായതിനാൽ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിക്കുകയും ചെയ്യുന്നു. ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിച്ചതിനുശേഷം പരിഷ്കരിക്കൽ പ്രസ്താവനയായ `++i` പ്രവർത്തിക്കുന്നു. വീണ്ടും `i<=n` എന്ന പരിശോധന പ്രയോഗം വിലയിരുത്തുകയും നിബന്ധന ശരിയായതിനാൽ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിക്കുകയും ചെയ്യുന്നു. പരിശോധന പ്രയോഗം തെറ്റായ വില തിരിച്ചു തരുന്നതുവരെ ഈ പ്രക്രിയ തുടരുന്നു. മാതൃക ഔട്ട്പുട്ടിൽ ഇത് സംഭവിക്കുന്നത് `i`-യുടെ വില 6 ആകുമ്പോഴാണ്.

നമുക്കു ചെയ്യാം

തന്നിരിക്കുന്ന സംഖ്യയുടെ ഗുണനപട്ടിക പ്രദർശിപ്പിക്കുവാനുള്ള ഒരു പ്രോഗ്രാം എഴുതുക. സംഖ്യ ഇൻപുട്ട് ചെയ്യുന്നത് `n` എന്ന വേരിയബിളിലാണെന്ന് കരുതുക. ലൂപ്പിന്റെ ചട്ടക്കൂട് താഴെ കൊടുത്തിരിക്കുന്നു.

```
cout<<i<<" x " <<n<<" = "<< i * n << "\n";
```

ഔട്ട്പുട്ട് കൂടി കാണിക്കുക.

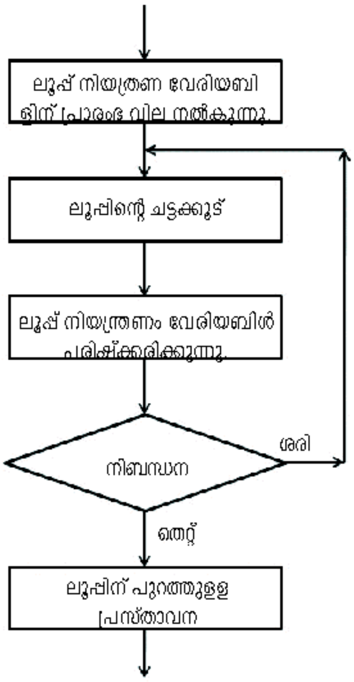
7.2.3 do...while പ്രസ്താവന (do...while statement)

`for` ലൂപ്പിന്റെയും , `while` ലൂപ്പിന്റെയും കാര്യത്തിൽ ലൂപ്പ് ചട്ടക്കൂട് പ്രവർത്തിക്കുന്നതിന് മുമ്പ് പരിശോധന പ്രയോഗം വിലയിരുത്തുന്നു. ആദ്യ തവണ തന്നെ പരിശോധന പ്രയോഗം തെറ്റാണെങ്കിൽ ലൂപ്പ് പ്രവർത്തിക്കില്ല. എന്നാൽ ചില സാഹചര്യങ്ങളിൽ പരിശോധന പ്രയോഗത്തിന്റെ ഫലം പരിഗണിക്കാതെ തന്നെ ലൂപ്പിന്റെ ചട്ടക്കൂട് ഒരു പ്രാവശ്യമെങ്കിലും പ്രവർത്തിപ്പിക്കേണ്ടത് ആവശ്യമായി വരും. അത്തരം സാഹചര്യത്തിൽ **do...while** ലൂപ്പ് ഉപയോഗിക്കുന്നതാണ് നല്ലത്. `do...while` ലൂപ്പിന്റെ വാക്യഘടന (syntax) ഇതാണ്.

നിയന്ത്രണവേരിയബിളിന്റെ പ്രാരംഭ വില നൽകൽ;

```
do
{
ലൂപ്പിന്റെ ചട്ടക്കൂട്;
ലൂപ്പ് നിയന്ത്രണവേരിയബിളിന്റെ വില പുതുക്കൽ;
} while (പരിശോധന പ്രയോഗം) ;
```

```
initialisation of loop control variable;
do
{
      body of the loop;
      updation of loop control variable;
} while(test expression);
```



ചിത്രം 7.5: do...while ലൂപ്പിന്റെ ഫ്ളോചാർട്ട്

ചിത്രം 7.5-ൽ ഈ ലൂപ്പിന്റെ പ്രവർത്തന ക്രമം കാണിച്ചിരിക്കുന്നു. ഇവിടെ ലൂപ്പ് ചട്ടക്കൂട് പ്രവർത്തിച്ചതിനുശേഷം മാത്രമാണ് പരിശോധന പ്രസ്താവന വിലയിരുത്തുന്നത്. അതിനാൽ do...while ലൂപ്പ് ഒരു ബഹിർഗമന നിയന്ത്രണ ലൂപ്പ് (Exit controlled loop) ആകുന്നു. പരിശോധന പ്രയോഗം തെറ്റാണെങ്കിൽ ലൂപ്പിന്റെ പ്രവർത്തനം അവസാനിക്കുന്നു. ഇത് അർത്ഥമാക്കുന്നത് പരിശോധന പ്രയോഗത്തിന്റെ ഫലം പരിഗണിക്കാതെ തന്നെ ലൂപ്പിന്റെ ചട്ടക്കൂട് ഒരു പ്രാവശ്യം പ്രവർത്തിക്കുന്നു എന്നാണ്.

do...while ലൂപ്പിന്റെ പ്രവർത്തനം വിശദീകരിക്കുന്നതിനായി താഴെ കൊടുത്തിരിക്കുന്ന പ്രോഗ്രാം ശകലം നമുക്ക് പരിശോധിക്കാം.

```
int k=1;
do
{
    cout << k << '\t';
    ++k;
} while(k<=3);
 cout << "\n Program Ends";
```

ലൂപ് തുടങ്ങുന്നതിനുമുമ്പ് പ്രാരംഭ വില നൽകുന്നു.

ലൂപ്പിന്റെ ചട്ടക്കൂട്

ലൂപ്പിന്റെ ചട്ടക്കൂടിനുള്ളിൽ ലൂപ് നിയന്ത്രണ വേരിയബിളിന്റെ വില പരിഷ്കരിക്കുന്നു.

പരിശോധനാ പ്രയോഗം

ആദ്യം വേരിയബിൾ **k**-യുടെ വിലയായി 1 നൽകുന്നു. അതിനുശേഷം ലൂപ്പ് ചട്ടക്കൂട് പ്രവർത്തിക്കുകയും **k** യുടെ വിലയായ 1 എന്ന് പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു. തുടർന്ന് **k**-യുടെ വില 1

വർദ്ധിപ്പിക്കുന്നു (ഇപ്പോൾ **k**=2). അതിനുശേഷം **k<=3** എന്ന വ്യവസ്ഥ പരിശോധിക്കുന്നു. ആ വ്യവസ്ഥ ശരിയായതിനാൽ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിച്ച് **k**-യുടെ വില 2 എന്ന് സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുന്നു. പുതുക്കൽ പ്രക്രിയ വീണ്ടും നടത്തി **k**-യുടെ വില 3 ആക്കുകയും **k<=3** എന്ന നിബന്ധന വീണ്ടും പരിശോധിക്കുകയും ചെയ്യുന്നു. നിബന്ധന ശരിയായതിനാൽ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിപ്പിച്ച് **k**-യുടെ വിലയായ 3 പ്രദർശിപ്പിക്കുന്നു. **k**-യുടെ വില വീണ്ടും പരിഷ്കരിച്ച് 4 ആകുന്നു. ഇത് പ്രോഗ്രാമിന്റെ നിയന്ത്രണം ലൂപ്പിന് പുറത്ത് വരുന്നതിനും തുടർന്നുള്ള പ്രസ്താവന പ്രവർത്തിക്കുന്നതിനും കാരണമാകുന്നു. ആയതിനാൽ കോഡിന്റെ ഔട്ട്പുട്ട് ഇങ്ങനെയായിരിക്കും.

```
    1    2    3
```

ഈ ലൂപ്പ് മറ്റു രണ്ടു ലൂപ്പിൽ നിന്നും എങ്ങനെ വ്യത്യാസപ്പെട്ടിരിക്കുന്നു എന്ന് ഇപ്പോൾ നമുക്കു നോക്കാം. **k**-യുടെ പ്രാരംഭവില 5 ആണെന്ന് സങ്കൽപിക്കുക. എന്ത് സംഭവിക്കും? ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിച്ച് **k**-യുടെ വിലയായ 5 സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുന്നു. അതിനുശേഷം **k**-യുടെ വില ഒന്ന് വർദ്ധിപ്പിച്ച് 6 ആയി തീരുന്നു. **k<=3** എന്ന നിബന്ധന പരിശോധിച്ചപ്പോൾ പരിശോധന പ്രയോഗം തെറ്റാവുകയും നിയന്ത്രണം ലൂപ്പിന് പുറത്തേയ്ക്കു വരുകയും ചെയ്യുന്നു. `do...while` ലൂപ്പിന്റെ ചട്ടക്കൂടിലേക്ക് ആദ്യത്തെ പ്രാവശ്യം പ്രവേശിക്കുന്നതിന് യാതൊരു നിയന്ത്രണവും ഇല്ലെന്നാണ് ഇത് കാണിക്കുന്നത്. അതുകൊണ്ടു നിബന്ധനയുടെ ശരി (`True`) വില മാത്രം അനുസരിച്ചാണ് ലൂപ്പ് ചട്ടക്കൂട് പ്രവർത്തിക്കേണ്ടതെങ്കിൽ `while` ലൂപ്പോ, `for` ലൂപ്പോ ഉപയോഗിക്കുക.

ഉപയോക്താവിന്റെ ആവശ്യത്തിനനുസരിച്ച് പ്രവർത്തിക്കുന്ന ഒരു പ്രോഗ്രാം നമുക്കു നോക്കാം. ഇത്തരം പ്രോഗ്രാമുകൾ ഉപയോക്താവിന്റെ പ്രതികരണം സ്വീകരിച്ചുകൊണ്ട് കോഡ് ശകലം ആവർത്തിച്ചു പ്രവർത്തിപ്പിക്കുന്നു.

ഉപഭോക്താവിൽ നിന്നും ഓരോ ചതുരത്തിന്റേയും നീളവും വീതിയും ഇൻപുട്ടായി സ്വീകരിച്ച് ചതുരങ്ങളുടെ വിസ്തീർണം കണ്ടുപിടിക്കുന്നതിനുള്ള ഒരു പ്രോഗ്രാം `do...while` ലൂപ്പ് ഉപയോഗിച്ച് എഴുതിയിരിക്കുന്നു. (പ്രോഗ്രാം 7.15)

പ്രോഗ്രാം 7.15 ചതുരത്തിന്റെ വിസ്തീർണം കാണുന്നതിന്

```
#include <iostream>
using namespace std;
int main()
{
    float length, breadth, area;
    char ch;
    do
    {
        cout << "Enter length and breadth: ";
        cin >> length >> breadth;
        area = length * breadth;
        cout << "Area = " << area;
```

```
        cout << "Any more rectangle (Y/N)? ";
        cin >> ch;
    } while (ch == 'Y' || ch == 'y');
    return 0;
}
```

പ്രോഗ്രാം 7.15 ന്റെ ഒരു മാതൃക ഔട്ട്പുട്ട് താഴെ കൊടുക്കുന്നു.

```
Enter length and breadth: 3.5      7
     Area = 24.5
     Any more rectangle (Y/N)? Y
     Enter length and breadth: 6      4.5
     Area = 27
     Any more rectangle (Y/N)? N
```

ഉപയോക്താവ് ഇൻപുട്ട് നൽകുന്നു

ഉപയോക്താവ് ഇൻപുട്ട് നൽകുന്നു

ഉപയോക്താവ് ഇൻപുട്ട് നൽകുന്നു

C++ ലെ മൂന്ന് ലൂപ്പിങ്ങ് പ്രസ്താവനകളെക്കുറിച്ചും നാം ചർച്ച ചെയ്തു. പട്ടിക 7.2 ൽ ഈ പ്രസ്താവനകൾ താരതമ്യം ചെയ്തിരിക്കുന്നു.

for ലൂപ്പ്	while ലൂപ്പ്	do...while ലൂപ്പ്
ആഗമന നിയന്ത്രണ ലൂപ്പ് (Entry controlled loop)	ആഗമന നിയന്ത്രണ ലൂപ്പ് (Entry controlled loop)	ബഹിർഗമന നിയന്ത്രണ ലൂപ്പ് (Extit controlled loop)
ലൂപ്പിന്റെ നിർവചനത്തോടൊപ്പം തന്നെ പ്രാരംഭ വിലയും നൽകുന്നു.	ലൂപ്പ് നിർവചനത്തിനു മുമ്പ് പ്രാരംഭവില നൽകുന്നു.	ലൂപ്പ് നിർവചനത്തിനു മുമ്പ് പ്രാരംഭവില നൽകുന്നു.
ലൂപ്പിന്റെ ചട്ടക്കൂട് ഒരു പ്രാവശ്യമെങ്കിലും പ്രവർത്തിക്കുമെന്ന് ഉറപ്പില്ല	ലൂപ്പിന്റെ ചട്ടക്കൂട് രു പ്രാവശ്യമെങ്കിലും പ്രവർത്തിക്കുമെന്ന് ഉറപ്പില്ല.	നിബന്ധന തെറ്റാണെങ്കിലും ലൂപ്പിന്റെ ചട്ടക്കൂട് ഒരു പ്രാവശ്യം പ്രവർത്തിക്കും.

പട്ടിക 7.2: C++ ലെ ലൂപ്പ് പ്രസ്താവനകളുടെ താരതമ്യം

നമുക്ക് സംഗ്രഹിക്കാം

ഒരു പ്രോഗ്രാമിൽ തീരുമാനങ്ങൾ എടുക്കുന്നതിനോ ആവർത്തന പ്രവർത്തനങ്ങൾ നടപ്പാക്കുന്നതിനോ ഉള്ള സൗകര്യമൊരുക്കുന്ന പ്രസ്താവനകൾ നിയന്ത്രണ പ്രസ്താവനകൾ എന്ന് അറിയപ്പെടുന്നു. നിയന്ത്രണ പ്രസ്താവനകൾ ഒരു കമ്പ്യൂട്ടർ പ്രോഗ്രാമിന്റെ നട്ടെല്ലാണ്. ഈ അധ്യായത്തിൽ വിവിധ തരം നിയന്ത്രണ പ്രസ്താവനകളായ തിരഞ്ഞെടുക്കൽ പ്രസ്താവനകൾ (`if`, `if...else`, `if...else if`, `switch`), ആവർത്തന പ്രസ്താവനകൾ (`for`, `while`, `do...while`) എന്നിവ നാം പഠിച്ചു. സങ്കീർണ്ണമായ പ്രോഗ്രാമുകൾ എഴുതുന്നതിന് ഈ നിയന്ത്രണ പ്രസ്താവനകൾ നമ്മെ സഹായിക്കുന്നു. വിവിധ C++ പ്രോഗ്രാമിങ് പ്രവർത്തനങ്ങൾ നടത്തുന്നതിന് ഈ പ്രസ്താവനകൾ അത്യാവശ്യമാണ്.

പഠന നേട്ടങ്ങൾ

ഈ അധ്യായം പൂർത്തിയാകുന്നതോടെ പഠിതാവിന്

- പ്രശ്നങ്ങൾ നിർദ്ധാരണം ചെയ്യുന്നതിന് C++ ലെ നിയന്ത്രണ പ്രസ്താവനകൾ ഉപയോഗിക്കുന്നു.
- നിയന്ത്രണ പ്രസ്താവനകൾ ഒരു പ്രോഗ്രാമിൽ ഏതു സാഹചര്യത്തിലാണ് ഉപയോഗിക്കേണ്ടത് എന്ന് തിരിച്ചറിയുന്നു.
- സാഹചര്യത്തിന് അനുയോജ്യമായ ശരിയായ നിയന്ത്രണ പ്രസ്താവനകൾ ഉപയോഗിക്കുന്നു.
- വിവിധ തരം നിയന്ത്രണ പ്രസ്താവനകളെ തരം തിരിക്കുന്നു.
- നിയന്ത്രണ പ്രസ്താവനകൾ ഉപയോഗിച്ച് C++ പ്രോഗ്രാം എഴുതുന്നു.

ലാബ് പ്രവർത്തനങ്ങൾ

1. ഒരു ഡിജിറ്റ് ഇൻപുട്ട് ചെയ്ത് അത് വാക്കിൽ പ്രദർശിപ്പിക്കുന്നതിനുള്ള പ്രോഗ്രാം എഴുതുക.
2. ആദ്യത്തെ n ഒറ്റ സംഖ്യകൾ പ്രദർശിപ്പിക്കുന്നതിനുള്ള ഒരു പ്രോഗ്രാം എഴുതുക.
3. ആദ്യത്തെ n ഒറ്റ സംഖ്യകളുടെ വർഗ്ഗങ്ങളുടെ തുക കണ്ടുപിടിച്ച് പ്രദർശിപ്പിക്കുന്നതിനുള്ള ഒരു പ്രോഗ്രാം എഴുതുക.
4. ഒന്നിനും n-നുമിടയിൽ m കൊണ്ട് പൂർണ്ണമായും ഹരിക്കാൻ കഴിയുന്ന അക്കങ്ങളെ പ്രദർശിപ്പിക്കുന്നതിനുള്ള ഒരു പ്രോഗ്രാം എഴുതുക.

മാതൃകാ ചോദ്യങ്ങൾ

1. switch പ്രസ്താവനയിൽ break- പ്രസ്താവനയുടെ പ്രാധാന്യം എഴുതുക. switch പ്രസ്താവനയിൽ break-ന്റെ അഭാവം എന്ത് ഫലം ഉളവാക്കും?
2. താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് ശകലത്തിന്റെ ഔട്ട്പുട്ട് എന്തായിരിക്കും?
```
for(i=1;i<=10;++i) ;
cout<<i+5;
```
3. 1000-നും 2000-നും ഇടയിൽ 132 കൊണ്ട് ഹരിക്കാവുന്ന സംഖ്യകൾ പ്രിന്റ് ചെയ്യുന്നതിനുള്ള ഒരു പ്രോഗ്രാം For ലൂപ്പ് ഉപയോഗിച്ച് എഴുതുക.
4. താഴെ പറയുന്ന പ്രസ്താവനയെ **while, do while** ലൂപ്പുകൾ ഉപയോഗിച്ച് മാറ്റി എഴുതുക.
```
for (i=1; i<=10; i++) cout<<i;
```
5. താഴെ കൊടുത്തിരിക്കുന്ന ലൂപ്പ് എത്ര തവണ പ്രവർത്തിക്കും.
```
int  s=0, i=0;
while(i++<5)
    s+=i;
```

ലഘു ഉപന്യാസ ചോദ്യങ്ങൾ

1. താഴെ കൊടുത്തിരിക്കുന്ന രണ്ട് കോഡ് ശകലങ്ങൾ പരിഗണിക്കുക.

```
// version 1
cin>>mark;
if (mark > = 90)
cout<<" A+";
if (mark > = 80 && mark <90)
cout<<" A";
if (mark > = 70 && mark <80)
cout<<" B+";
if (mark > = 60 && mark <70)
cout<<" B";
```

```
//version 2
cin>>mark;
if (mark>=90)
cout<<" A+";
else if (mark>=80 && mark <90)
cout<<" A";
else if (mark>=70 && mark <80)
cout<<" B+";
else if (mark>=60 && mark <70)
cout<<" B";
```

വേർഷൻ 2 ന് വേർഷൻ 1 നെ അപേക്ഷിച്ചുള്ള മേന്മകൾ ചർച്ച ചെയ്യുക.

2. ഒരു for ലൂപ്പിന്റെ പ്രവർത്തനം അതിന്റെ വാക്യഘടന (Syntax) യോടുകൂടി ചുരുക്കി വിവരിക്കുക. നിങ്ങളുടെ ഉത്തരം സാധൂകരിക്കുന്നതിന് for ലൂപ്പിന്റെ ഒരു ഉദാഹരണം നൽകുക.
3. വിവിധ സാഹചര്യങ്ങളിൽ മൂന്നു ലൂപ്പുകളുടെ അനുയോജ്യത താരതമ്യം ചെയ്ത് ചർച്ച ചെയ്യുക.
4. z=3 ആണെങ്കിൽ താഴെ കൊടുത്തിരിക്കുന്ന while പ്രസ്താവനയിലെ തെറ്റ് എന്താണ്?

```
while(z>=0)
      sum+=z;
```

5. താഴെ കൊടുത്തിരിക്കുന്ന if.. else if പ്രസ്താവന പരിഗണിക്കുക. switch പ്രസ്താവന കൊണ്ട് അത് മാറ്റി എഴുതുക.

```
if (a==1)
      cout << "One";
else if (a==0)
      cout << "Zero";
else
      cout << "Not a binary digit";
```

6. ഒരു ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിന്റെ പ്രാധാന്യം എഴുതുക. ഒരു ലൂപ്പിന്റെ വിവിധ ഭാഗങ്ങളെക്കുറിച്ച് ചുരുക്കി വിവരിക്കുക.

ഉപന്യാസ ചോദ്യങ്ങൾ

1. C++- ലെ ലഭ്യമായ വിവിധ തരം തീരുമാനമെടുക്കൽ പ്രസ്താവനകൾ വിശദീകരിക്കുക.
2. C++- ലെ വിവിധ ആവർത്തന പ്രസ്താവനകൾ വാക്യഘടനയോടും ഉദാഹരണത്തോടും കൂടി വിവരിക്കുക.

8

M8T5C2

പ്രധാന ആശയങ്ങൾ

കമ്പ്യൂട്ടർ ശൃംഖലകൾ (Computer Networks)
- ശൃംഖലയുടെ ആവശ്യകത
- ചില പ്രധാന പദങ്ങൾ

ഡാറ്റ വിനിമയ സംവിധാനം (Data Communication System)

വിനിമയ മാധ്യമം (Communication Medium)
- ഗൈഡഡ് മീഡിയം (Guided Medium)
- അൺ ഗൈഡഡ് മീഡിയം (Unguided Medium)
- റേഡിയോ തരംഗങ്ങൾ ഉപയോഗിച്ചുള്ള വയർ ലെസ്സ് വിനിമയ സാങ്കേതികവിദ്യകൾ

ഡാറ്റ വിനിമയ ഉപകരണങ്ങൾ (Data Communication Devices)
- എൻ.ഐ.സി. (NIC), ഹബ് (HUB), സ്വിച്ച് (SWITCH), റിപീറ്റർ (Repeater), ബ്രിഡ്ജ് (Birdge), റൂട്ടർ (Router), ഗേറ്റ്‌വേ (Gagteway)

ഡാറ്റ ടെർമിനൽ ഉപകരണങ്ങൾ (Data Terminal Equipments)
- മോഡം (Modem), മൾട്ടിപ്ലെക്സർ (Multiplexer) /ഡിമൾട്ടിപ്ലെക്സർ (Demultiplexer)
- ശൃംഖല ക്രമീകരണ രീതികൾ (Network Topologies)
- ബസ് (Bus), സ്റ്റാർ (Star), റിങ് (Ring), മെഷ് (Mesh)

വിവിധ തരം ശൃംഖലകൾ
- പാൻ (PAN), ലാൻ (LAN), മാൻ (MAN), വാൻ (WAN)

ശൃംഖലയുടെ യുക്ത്യാധിഷ്ഠിത തരംതിരിവുകൾ/വിഭജനം
- പീർ - ടു - പീർ (Peer - to - peer)
- ക്ലയന്റ് സർവർ (Client - Server)

ശൃംഖലാ പെരുമാറ്റ ചട്ടങ്ങൾ/നിയമങ്ങൾ (Network Protocol)
- TCP/IP (HTTP, FTP, DNS)

ഉപയോക്താവിനെയും കമ്പ്യൂട്ടറുകളെയും ശൃംഖലയിൽ തിരിച്ചറിയൽ
- MAC വിലാസം (MAC Address)
- ഐപി വിലാസം (IP Address)
- യൂണിഫോം റിസോഴ്സ് ലൊക്കേറ്റർ (Uniform Resoure Locator)

കമ്പ്യൂട്ടർ ശൃംഖലകൾ

പത്താം ക്ലാസ് പരീക്ഷയുടെ ഫലം അറിയുവാനോ പതിനൊന്നാം ക്ലാസ്സിൽ പ്രവേശനം കിട്ടിയോ എന്ന് പരിശോധിക്കുന്നതിനോ നിങ്ങൾ ഇന്റർനെറ്റ് ഉപയോഗിച്ചിട്ടുണ്ടോ? പണം പിൻവലിക്കുന്നതിനായി നിങ്ങൾ എ ടി എം സന്ദർശിച്ചിട്ടുണ്ടോ? കമ്പ്യൂട്ടറിൽ നിന്ന് പാട്ടുകൾ, ചിത്രങ്ങൾ, സിനിമാ ശകലങ്ങൾ എന്നിവ സെൽഫോണിലേക്ക് മാറ്റുവാനോ, ഇന്റർനെറ്റ് ഉപയോഗിച്ച് ട്രെയിൻ ടിക്കറ്റ് ബുക്ക് ചെയ്യുവാനോ നിങ്ങൾ ശ്രമിച്ചിട്ടുണ്ടോ? ഈ ചോദ്യങ്ങളിൽ ഏതെങ്കിലും ഒന്നിന് നിങ്ങളുടെ ഉത്തരം 'അതെ' എന്നാണെങ്കിൽ, നിങ്ങൾ കമ്പ്യൂട്ടർ ശൃംഖലയുടെ സേവനം ഉപയോഗപ്പെടുത്തിയിട്ടുണ്ട് എന്ന് അനുമാനിക്കാം. കമ്പ്യൂട്ടർ ശൃംഖലയുടെ പ്രവർത്തനങ്ങളെക്കുറിച്ചും അവയുടെ ഗുണങ്ങളെക്കുറിച്ചുമാണ് ഈ അദ്ധ്യായത്തിൽ പഠിക്കുന്നത്. ഇതോടൊപ്പം ഈ മേഖലയിൽ ഉപയോഗിക്കുന്ന വിവിധ ഉപകരണങ്ങളെക്കുറിച്ചും മാധ്യമങ്ങളെക്കുറിച്ചും നമുക്ക് ചർച്ച ചെയ്യാം. കൂടാതെ വിവിധതരം കമ്പ്യൂട്ടർ ശൃംഖലകളെക്കുറിച്ചും ശൃംഖലകളിലൂടെ വിനിമയം നടത്തുവാനാവശ്യമായ നിയമങ്ങളെക്കുറിച്ചും ചർച്ച ചെയ്യാം.

8.1 കമ്പ്യൂട്ടർ ശൃംഖല (Computer network)

ഒരു വിനിമയ ഇലക്ട്രോണിക് മാധ്യമത്തിലൂടെ പരസ്പരം ബന്ധിപ്പിച്ചിട്ടുള്ള കമ്പ്യൂട്ടറുകളുടെയും

മറ്റു കമ്പ്യൂട്ടിങ് ഹാർഡ്‌വയർ ഉപകരണങ്ങളുടെയും (പ്രിന്ററുകൾ, സ്കാനറുകൾ, മോഡം, CD ഡ്രൈവുകൾ തുടങ്ങിയവ) ഒരു കൂട്ടമാണ് കമ്പ്യൂട്ടർ ശൃംഖല. ഈ ഉപകരണങ്ങൾക്ക് പരസ്പരം വിവരങ്ങൾ വിനിമയം നടത്തുവാനും, നിർദ്ദേശങ്ങൾ കൈമാറുവാനും, ഡാറ്റയും ഉപകരണങ്ങളും പരസ്പരം പങ്കിടുവാനും സാധിക്കുന്നു. ഒരു ശൃംഖലയിൽ ഉള്ള കമ്പ്യൂട്ടറുകളെ കേബിളുകൾ, ടെലിഫോൺ ലൈനുകൾ, റേഡിയോ തരംഗങ്ങൾ, ഇൻഫ്രാറെഡ് തരംഗങ്ങൾ, ഉപഗ്രഹങ്ങൾ ഇവയിലേതെങ്കിലും ഉപയോഗിച്ച് പരസ്പരം ബന്ധിപ്പിക്കാം.

8.1.1 ശൃംഖലയുടെ ആവശ്യകത (Need for network)

ഒരു കമ്പ്യൂട്ടർ ശൃംഖലയ്ക്ക് ഉത്തമ ഉദാഹരണമാണ് ഇന്റർനെറ്റ്. ഇമെയിൽ, ഓൺലൈൻ പത്രങ്ങൾ, ബ്ലോഗുകൾ, ചാറ്റിങ് ഇന്റർനെറ്റ് അധിഷ്ഠിത സേവനങ്ങൾ തുടങ്ങിയവ ഇല്ലാത്ത ഒരു ലോകത്തെ കുറിച്ച് നമുക്ക് ചിന്തിക്കുവാൻ കഴിയില്ല. പരസ്പരം ബന്ധിപ്പിച്ചിട്ടില്ലാത്ത കമ്പ്യൂട്ടറുകൾ ഉപയോഗിക്കുന്നതിനേക്കാൾ പലമേന്മകളും പരസ്പരം ബന്ധിപ്പിച്ച കമ്പ്യൂട്ടറുകൾക്ക് ഉണ്ട്. അവയിൽ ചിലത് ചുവടെ ചേർക്കുന്നു.

- വിഭവം പങ്കുവെയ്ക്കൽ (Resource sharing)
- വില പ്രകടന അനുപാതം (Price preformance ratio)
- വിവര വിനിമയം (Communication)
- വിശ്വാസ്യത (Reliability)
- വിപുലീകരിക്കുവാനുള്ള സാധ്യത (Scalability)

വിഭവം പങ്കുവെയ്ക്കൽ: കമ്പ്യൂട്ടർ ശൃംഖലയിൽ ലഭ്യമായ ഹാർഡ്‌വെയറും സോഫ്റ്റ്‌വെയറും പങ്കിടുന്നതിനെയാണ് വിഭവങ്ങളുടെ പങ്കുവെയ്ക്കൽ എന്നതുകൊണ്ട് ഉദ്ദേശിക്കുന്നത്. ഉദാഹരണത്തിന് ഒരു കമ്പ്യൂട്ടറിന്റെ ഡിവിഡി ഡ്രൈവിൽ നിന്നാണ് ഒരു ഡിവിഡി യുടെ ഉള്ളടക്കം മറ്റൊരു കമ്പ്യൂട്ടറിൽ ഉപയോഗിക്കുന്നത്. അതുപോലെ, മറ്റ് ഹാർഡ്‌വെയർ ഉപകരണങ്ങളായ ഹാർഡ് ഡിസ്ക്, പ്രിന്റർ, സ്കാനർ, തുടങ്ങിയവയും സോഫ്റ്റ്‌വെയറുകളായ അപ്ലിക്കേഷൻ സോഫ്റ്റ്‌വെയർ, ആന്റി വൈറസുകൾ തുടങ്ങിയവയും കമ്പ്യൂട്ടർ ശൃംഖല വഴി പരസ്പരം പങ്കിടാം.

വില പ്രകടന അനുപാതം: ഒരു കമ്പ്യൂട്ടറിൽ ലഭ്യമായ വിഭവങ്ങൾ ശൃംഖലയിലുള്ള മറ്റ് കമ്പ്യൂട്ടറുകളുമായി എളുപ്പത്തിൽ പങ്കിടുവാൻ കഴിയുന്നു. ലൈസൻസുള്ള സോഫ്റ്റ്‌വെയർ ഓരോ കമ്പ്യൂട്ടറിനും വാങ്ങുന്നതിനുള്ള ചെലവ് അത്തരം സോഫ്റ്റ്‌വെയറിന്റെ ശൃംഖല പതിപ്പുകൾ വാങ്ങിക്കൊണ്ടു കുറയ്ക്കുവാൻ കഴിയും. വിഭവങ്ങളുടെ ഇത്തരത്തിലുള്ള ഉപയോഗം കമ്പ്യൂട്ടറിന്റെ പ്രകടനത്തെ ബാധിക്കാത്ത വിധത്തിലും, കൂടാതെ കുറഞ്ഞ ചിലവിൽ, ഗണ്യമായ ലാഭത്തിലേക്കു നയിക്കുന്ന തരത്തിലും ആയിരിക്കും.

വിവര വിനിമയം: ഇമെയിൽ, ചാറ്റിങ്, വീഡിയോ കോൺഫെറൻസിങ് തുടങ്ങിയ സേവനങ്ങളിലൂടെ ശൃംഖലയിലുള്ള മറ്റേതെങ്കിലും ഉപഭോക്താവുമായി വിവര വിനിമയം നടത്തുവാൻ കമ്പ്യൂട്ടർ ശൃംഖല സഹായിക്കുന്നു. ഉദാഹരണമായി ലക്ഷ്യസ്ഥാന

ത്തിലേക്കുള്ള ദൂരം കണക്കിലെടുക്കാതെ വളരെ വേഗത്തിൽ സന്ദേശങ്ങൾ അയക്കുവാനും സ്വീകരിക്കുവാനും കഴിയുന്നു.

വിശ്വാസ്യത : കമ്പ്യൂട്ടർ ശൃംഖല ഉപയോഗിച്ച് ഒന്നിലധികം കമ്പ്യൂട്ടറുകളിൽ ആവശ്യമായ വിവരങ്ങളുടെ നിരവധി പകർപ്പുകൾ സൂക്ഷിക്കുവാൻ കഴിയുന്നു. ഉദാഹരണത്തിന്, ഒരു കമ്പ്യൂട്ടറിൽ സംരക്ഷിച്ചിട്ടുള്ള C++ ഫയലുകൾ, ചിത്രങ്ങൾ അല്ലെങ്കിൽ പാട്ടുകൾ എന്നിവ ഇതേ ശൃംഖലയിലെ മറ്റു കമ്പ്യൂട്ടറുകളിൽ സൂക്ഷിക്കാവുന്നതാണ്. ഇങ്ങനെ സൂക്ഷിക്കുന്നത് കൊണ്ട്, ഏതെങ്കിലും കമ്പ്യൂട്ടറിന് തകരാറുണ്ടായാൽ (ശരിയായി പ്രവർത്തിക്കാതിരിക്കുക, യാദൃശ്ചികമായി ഫയലുകൾ നഷ്ടപ്പെട്ട് പോകുക) ഈ ഫയലുകളെ കമ്പ്യൂട്ടർ ശൃംഖലയിൽ നിന്നും വീണ്ടെടുക്കുവാൻ സാധിക്കുന്നു.

വിപുലീകരിക്കുവാനുള്ള സാധ്യത: കമ്പ്യൂട്ടർ ശൃംഖലയിലേയ്ക്ക് കമ്പ്യൂട്ടറുകളുടെ എണ്ണം കൂട്ടിയും കുറച്ചും ശൃംഖലയുടെ പ്രവർത്തന ക്ഷമത ഉയർത്തുകയും താഴ്ത്തുകയും ചെയ്യാം. ഇതിനുപുറമെ ശൃംഖലയിലേയ്ക്ക് കൂടുതൽ സംഭരണ ഉപകരണങ്ങൾ ഉൾപ്പെടുത്തി ശൃംഖലയുടെ സംഭരണ ശേഷി വർദ്ധിപ്പിക്കാം

8.1.2 ചില പ്രധാന പദങ്ങൾ (Some key terms)

കമ്പ്യൂട്ടർ ശൃംഖലയുമായി ബന്ധപ്പെട്ട ചില പ്രധാന പദങ്ങൾ ചുവടെ വിശദമാക്കുന്നു.

ബാൻഡ്‌വിഡ്ത്ത് (Bandwidth) : ബാൻഡ് വിഡ്ത് എന്നാൽ നിശ്ചിത സമയത്ത് നിശ്ചിത മാധ്യമത്തിലൂടെ അയയ്ക്കാവുന്ന ഡാറ്റയുടെ അളവാണ്. നിങ്ങൾ ഒരു ഹൈവേയിലൂടെയോ അല്ലെങ്കിൽ ഒരു പൊതുറോഡിലൂടെയോ സഞ്ചരിക്കുകയാണ് എന്ന് വിചാരിക്കുക. റോഡിന്റെ വീതി കൂടുന്തോറും അതിലൂടെ കടന്നു പോകുവാൻ കഴിയുന്ന വാഹനങ്ങളുടെ എണ്ണം കൂടുന്നതായി കാണാം. മാത്രമല്ല ഇവിടെ ഇടുങ്ങിയ റോഡിനേക്കാൾ വേഗത്തിൽ വാഹനങ്ങൾക്ക് സഞ്ചരിക്കാം. അതുകൊണ്ടു ഒരു വീതിയുള്ള റോഡിന്, ഇടുങ്ങിയ റോഡിനേക്കാൾ ബാൻഡ് വിഡ്ത് കൂടുതലാണ് എന്ന് നമുക്ക് മനസിലാക്കാം.

ഒരുശൃംഖലയിൽ കമ്പ്യൂട്ടറുകൾക്കിടയിൽ പരമാവധി കൈമാറ്റം ചെയ്യുവാൻ കഴിയുന്ന ഡാറ്റയുടെ അളവിനെ ബാൻഡ്‌വിഡ്ത്ത് എന്ന് പറയാം. ബിറ്റ്സ് പെർ സെക്കന്റ് (പ്രതി നിമിഷമാത്രകൾ) (ബിപിഎസ്) എന്ന രീതിയിൽ ഡിജിറ്റൽ സമ്പ്രദായത്തിൽ ഇതിനെ അളക്കുന്നു. ബാൻഡ്‌വിഡ്ത് കൂടുതലാവുമ്പോൾ ഡാറ്റയ്ക്കു വേഗത്തിൽ സഞ്ചരിക്കുവാൻ കഴിയുന്നു, ആയതിനാൽ ഒരു പ്രത്യേക സമയപരിധിക്കുള്ളിൽ ശൃംഖലയിലൂടെ വലിയ അളവിൽ ഡാറ്റ കൈമാറ്റം ചെയ്യാവുന്നതാണ്. ഉദാഹരണത്തിന് കേബിൾ മോഡം വഴിയുള്ള ഇന്റർനെറ്റ് കണക്ഷൻ 25 Mbps ബാൻഡ് വിഡ്ത് നൽകുന്നു.

നോയ്സ് (Noise): ഡാറ്റ സിഗ്നലിന്റെ ഗുണനിലവാരം കുറയ്ക്കുന്നതോ, സിഗ്നലുകളുടെയോ ഡാറ്റയുടെ നീക്കത്തെ തടസപ്പെടുത്തുന്നതോ ആയ മറ്റൊരു അനഭിമതമായ തരംഗമാണ് 'നോയ്സ്' (Noise). സമീപത്തുള്ള സംപ്രേക്ഷണ ഉപകരണങ്ങളിൽ നിന്നും,

മറ്റു യന്ത്രങ്ങളിൽ നിന്നും കേബിളുകളിൽ നിന്നും, പുറത്തു വരുന്ന സിഗ്നലുകളാണ് ഇതിനു കാരണം. ഒരു ശൃംഖലയിൽ (Network) കൈമാറ്റം ചെയ്യപ്പെടുന്ന ടെക്സ്റ്റുകൾ, പ്രോഗ്രാമുകൾ, ചിത്രങ്ങൾ, ഓഡിയോ തുടങ്ങിയ എല്ലാ ഡാറ്റയേയും നോയ്സ് പ്രതികൂലമായി ബാധിക്കുന്നു.

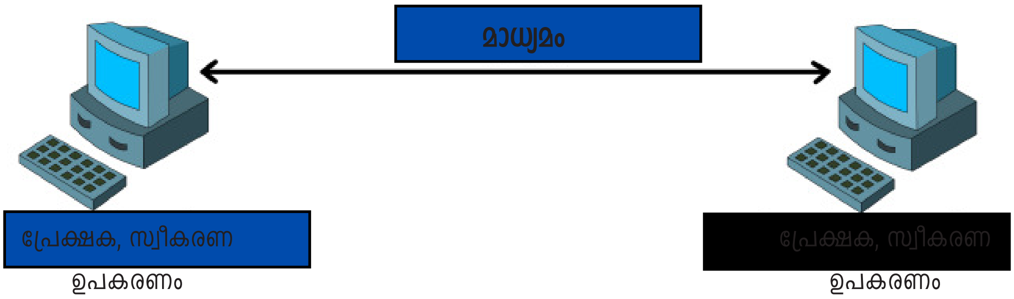
നോഡ് (Node): കമ്പ്യൂട്ടർ ശൃംഖലയിലേക്കു നേരിട്ട് ബന്ധിപ്പിച്ചിട്ടുള്ള ഏത് ഉപകരണത്തെയും (കമ്പ്യൂട്ടർ, സ്കാനർ, പ്രിൻറർ മുതലായവ) നോഡ് എന്ന് പറയുന്നു. ഉദാഹരണമായി, സ്കൂളിൽ കമ്പ്യൂട്ടർ ശൃംഖലയിലേക്ക് ബന്ധിപ്പിച്ചിരിക്കുന്ന കമ്പ്യൂട്ടറുകളെ നോഡ് എന്നാണ് അറിയപ്പെടുന്നത്. നമ്മുടെ കമ്പ്യൂട്ടറിനെ ഇന്റർനെറ്റുമായി ബന്ധിപ്പിക്കുമ്പോൾ, ആ കമ്പ്യൂട്ടർ ഇന്റർനെറ്റിലെ ഒരു നോഡ് ആയി മാറുന്നു.

നമുക്കു ചെയ്യാം

നിങ്ങളുടെ സ്കൂളിലെ കമ്പ്യൂട്ടർ ശൃംഖലയിൽ ഉപയോഗിച്ചിട്ടുള്ള ഹാർഡ് വെയറിന്റെയും സോഫ്റ്റ്‌വെയറിന്റെയും പട്ടിക തയ്യാറാക്കുക.

8.2 ഡാറ്റയുടെ വിനിമയ സമ്പ്രദായം (Data communication system)

വിവരവിനിമയത്തിനും പങ്കുവെയ്ക്കലിനും വേണ്ടി ഒരു കമ്പ്യൂട്ടർ ശൃംഖലയിലെ ഉപകരണങ്ങളെ വിവിധ രീതിയിൽ ബന്ധിപ്പിച്ചിരിക്കുന്നു. ഒരു സംപ്രേക്ഷണ മാധ്യമത്തിലൂടെ രണ്ടു ഉപകരണങ്ങൾ തമ്മിൽ നടത്തുന്ന ഡിജിറ്റൽ വിവരങ്ങളുടെ കൈമാറ്റത്തെ ഡാറ്റയുടെ വിനിമയം അഥവാ ഡാറ്റ കമ്മ്യൂണിക്കേഷൻ (Data Communication) എന്ന് പറയുന്നു. ചിത്രം 8.1 ൽ ഡാറ്റയുടെ വിനിമയ സംവിധാനത്തിന്റെ പൊതു പ്രാതിനിധ്യം കാണിക്കുന്നു.



ചിത്രം 8.1: ഡാറ്റ വിനിമയ സംവിധാനം

ഒരു ഡാറ്റയുടെ വിനിമയ സംവിധാനം നിർമ്മിക്കുന്നതിന് താഴെപ്പറയുന്ന അഞ്ച് അടിസ്ഥാന ഘടകങ്ങൾ ആവശ്യമാണ്.

സന്ദേശം (Message) : വിനിമയം ചെയ്യേണ്ട പ്രധാന വിവരങ്ങൾ ആണ് ഇത്. ഇതിൽ ടെക്സ്റ്റുകൾ, ചിത്രങ്ങൾ, ഓഡിയോ, വീഡിയോ തുടങ്ങിയവ ഉൾപ്പെടുന്നു.

പ്രേക്ഷകൻ (Sender): സന്ദേശം അയയ്ക്കുവാൻ ഉപയോഗിക്കുന്ന കമ്പ്യൂട്ടറിനെയും, ഉപകരണങ്ങളെയും, പ്രേക്ഷകനെന്നോ, ഉറവിടം എന്നോ, സംപ്രേക്ഷണ സാമഗ്രി എന്നോ വിളിക്കാം.

സ്വീകർത്താവ് (Receiver): സ്വീകർത്താവ് എന്നത് സന്ദേശങ്ങൾ സ്വീകരിക്കുന്ന കമ്പ്യൂട്ടറോ അനുബന്ധ ഉപകരണങ്ങളോ ആകാം.

മാധ്യമം (Medium):- പ്രേക്ഷകനിൽ നിന്ന് സ്വീകർത്താവിലേയ്ക്ക് സന്ദേശം സഞ്ചരിക്കുന്ന ഭൗതിക പാതയാണ് ഇത്. നോഡുകൾ തമ്മിൽ പരസ്പരം ബന്ധിപ്പിച്ചിരിക്കുന്ന രീതിയെ ഇത് സൂചിപ്പിക്കുന്നു.

പ്രൊട്ടോക്കോൾ (Protocol):- പ്രേക്ഷകനും സ്വീകർത്താവും സന്ദേശങ്ങൾ കൈമാറ്റം ചെയ്യുമ്പോൾ പാലിക്കേണ്ട നിയമങ്ങളെ പ്രോട്ടോകോൾ (protocol) എന്ന് വിളിക്കാം.

8.3 വിവര വിനിമയ മാധ്യമം (Communication medium)

ഒരു ഉപകരണത്തിൽ നിന്ന് മറ്റൊന്നിലേക്കു സന്ദേശം വഹിക്കുവാൻ കഴിയുന്ന ഒരു മാധ്യമം ഉണ്ടെങ്കിൽ മാത്രമേ ഡാറ്റയുടെ വിനിമയ പ്രക്രിയ പൂർണ്ണമാകുകയുള്ളൂ. ഒരു കമ്പ്യൂട്ടർ ശൃംഖലയിൽ ഡാറ്റ കൈമാറ്റം ചെയ്യുവാൻ ഉപയോഗിക്കുന്ന മാധ്യമത്തെ വിവരവിനിമയ പാത അല്ലെങ്കിൽ വിനിമയ മാധ്യമം എന്ന് വിളിക്കാം. ഒരു കമ്പ്യൂട്ടർ ശൃംഖലയിൽ വിവരവിനിമയത്തിനായി രണ്ടു തരത്തിലുള്ള മാധ്യമങ്ങളെ ഉപയോഗിക്കാം. ഗൈഡഡ് മാധ്യമവും അൺഗൈഡഡ് മാധ്യമവും ഗൈഡഡ് മാധ്യമത്തിൽ കേബിളുകൾ ഉപയോഗിക്കുന്നു. അതെ സമയം അൺഗൈഡഡ് മാധ്യമത്തിൽ റേഡിയോ തരംഗങ്ങൾ, മൈക്രോവേവ് തരംഗങ്ങൾ അല്ലങ്കിൽ ഇൻഫ്രാറെഡ് തരംഗങ്ങൾ എന്നിവയാണ് ഡാറ്റ അയയ്ക്കുവാനായി ഉപയോഗിക്കുന്നത്.

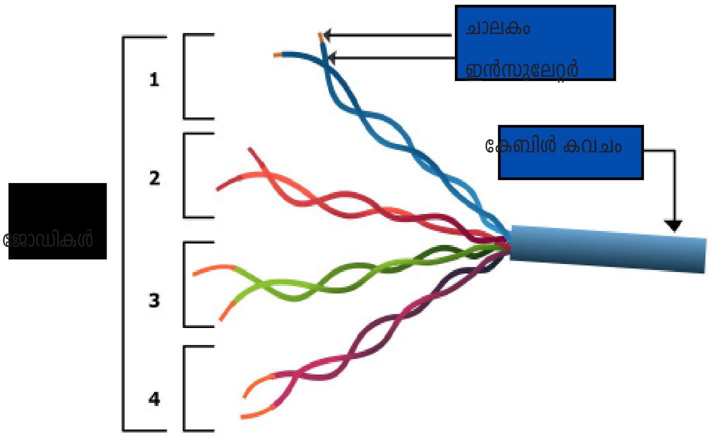
83.1 ഗൈഡഡ് മാധ്യമം (Guided Medium (wired))

കോയാക്സിൽ കേബിൾ (Coaxial cable), ട്വിസ്റ്റഡ് പെയർ കേബിൾ (Twisted pair cable), ഒപ്റ്റിക്കൽ ഫൈബർ കേബിൾ (Optical fibre cable) എന്നിവ കമ്പ്യൂട്ടർ ശൃംഖലയിൽ ഡാറ്റ കൈമാറുവാനായി ഉപയോഗിക്കുന്ന ഗൈഡഡ് മാധ്യമങ്ങളാണ്.

a. ട്വിസ്റ്റഡ് പെയർ കേബിൾ (ഈതർനെറ്റ് കേബിൾ) (Twisted pair cable (Ethernet calbe)

ചെറിയ കമ്പ്യൂട്ടർ ശൃംഖലയ്ക്ക് അനുയോജ്യവും, ഏറ്റവും വ്യാപകമായി ഉപയോഗിയ്ക്കുന്നതുമാണ് ഈ മാധ്യമം. വ്യത്യസ്ത നിറങ്ങൾ കൊണ്ട് തിരിച്ചറിയുവാൻ കഴിയുന്ന നാല് ജോഡി വയറുകളെ ഒരു കവചം കൊണ്ട് സംരക്ഷിച്ചു കൊണ്ടുള്ള രൂപകല്പനയാണിത്. ട്വിസ്റ്റഡ്പെയർ രണ്ടു തരത്തിലാണ് ഉള്ളത് 1) അൺഷീൽഡ് ട്വിസ്റ്റഡ് പെയർ ((Unshielded Twisted Pair (UTP)), 2) ഷീൽഡ് ട്വിസ്റ്റഡ് പെയർ (Shielded Twisted Pair) (STP) എന്നും

അൺഷീൽഡഡ് ട്വിസ്റ്റഡ് പെയർ (Unshielded Twisted Pair (UTP)): പേരു പോലെ തന്നെ കവചം ഇല്ലാത്ത തരം കേബിൾ ആണിത്.



ചിത്രം 8.2: UTP കേബിളിന്റെ പ്രധാന ഭാഗങ്ങൾ കാണിച്ചിരിക്കുന്നു.

ഇതിന്റെ പ്രധാന സവിശേഷതകൾ

- വളരെ കുറഞ്ഞ ചെലവിൽ ചെറിയ ശൃംഖലകൾ നിർമ്മിക്കാം.
- കനം കുറഞ്ഞതും വഴക്കമുള്ളതും ആയ കേബിളാണ്.
- വളരെ എളുപ്പത്തിൽ ശൃംഖലാ ഉപകരണങ്ങളെ ബന്ധിപ്പിക്കാം.
- 100 m ദൂരത്തിൽ വരെ ഡാറ്റയെ വഹിച്ചു കൊണ്ട് പോകുവാനുള്ള കഴിവ് ഉണ്ട്

ഷീൽഡഡ് ട്വിസ്റ്റഡ് പെയർ (Shielded Twisted Pair (STP)): UTP കേബിളിനെ പ്പോലെ തന്നെയാണ് എങ്കിലും STP യിൽ ജോഡികളായ വയറുകളെ പൊതിഞ്ഞു സൂക്ഷിക്കുന്നു. UTP കേബിളിനെ പോലെ പിന്നീട് എല്ലാറ്റിനെയും പൊതിഞ്ഞു കൊണ്ട് ഒരു കവചവും ഉണ്ടാകും.

ഇതിന്റെ പ്രധാന സവിശേഷതകൾ

- നോയിസ് (Noise) ന് എതിരെ ശക്തമായ പ്രതിരോധ സംവിധാനമാണ് ഈ കേബിളിന് ഉള്ളത്.
- ഇതിന് UTP കേബിളിനേക്കാൾ വില കൂടുതൽ ആണ്.
- UTP കേബിളുമായി താരതമ്യം ചെയ്യുമ്പോൾ STP കേബിൾ സ്ഥാപിക്കുവാൻ പ്രയാസമാണ്.

ചിത്രം 8.3: എസ്ടിപി കേബിളും RJ-45 കണക്ടറും

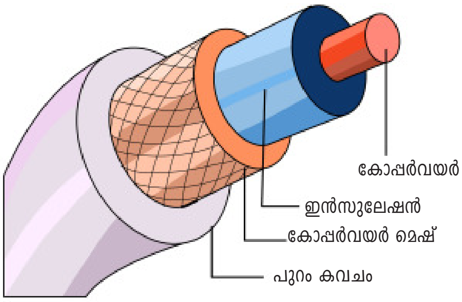
RJ45 എന്ന കണക്ടർ ഉപയോഗിച്ചാണ് UTP/STP കേബിളുകൾ കമ്പ്യൂട്ടറുമായി ബന്ധിപ്പിച്ചിരിക്കുന്നത്.

b. കൊയാക്സിയൽ കേബിൾ (Coaxial cable)

ഒരു കൊയാക്സിയൽ കേബിളിന്റെ ഉൾഭാഗത്ത് ഒരു ചാലകത്തെ പൊതിഞ്ഞു കൊണ്ട് ഒരു ഇൻസുലേറ്റർ ട്യൂബും വീണ്ടും അതിനെ പൊതിഞ്ഞു കൊണ്ട് ഒരു ചാലകവും (ഷീൽഡ്) ഉണ്ടായിരിക്കും. ഇതിനു പുറമെ ഒരു പ്രതിരോധ കവചവും കൂടി കാണും. ചിത്രം 8.4 കൊയാക്സിയൽ കേബിളിന്റെ ഘടന ചിത്രീകരിച്ചിരിക്കുന്നു.

കൊയാക്സിയൽ കേബിളിന്റെ സവിശേഷതകൾ.

- ദീർഘ ദൂരത്തേക്ക് (ഏകദേശം 185 m മുതൽ 500 m വരെ) ഒറ്റയടിക്ക് ഡാറ്റയെ വഹിച്ചു കൊണ്ട് പോകുവാൻ കഴിയും.
- വളരെ ഉയർന്ന ബാൻഡ്‌വിഡ്ത് ആണ് ഉള്ളത്.
- പുറംചട്ട (കവചം) ഉള്ളതുകൊണ്ട് വളരെ കുറഞ്ഞ തോതിലുള്ള വൈദ്യുതകാന്തിക തരംഗങ്ങളുടെ തടസ്സപ്പെടുത്തൽ മാത്രമേ ഉണ്ടാകുന്നുള്ളൂ.



ചിത്രം 8.4: കൊയാക്സിയൽ കേബിൾ

- ട്വിസ്റ്റഡ് പെയർ കേബിളിനേക്കാൾ കനം കൂടിയ രൂപകല്പനയാണ്.
- ട്വിസ്റ്റഡ് പെയർ കേബിളിനേക്കാൾ വഴക്കം വളരെ കുറവാണ്.
- ട്വിസ്റ്റഡ്പെയറുമായി താരതമ്യം ചെയ്യുമ്പോൾ സ്ഥാപിക്കുവാൻ പ്രയാസമാണ്.

c. ഒപ്റ്റിക്കൽ ഫൈബർ കേബിൾ (Optical fibre cable)

ഡാറ്റയെ പ്രകാശ കണികാ രൂപത്തിൽ ഒരു നീളം കൂടിയ കനം കുറഞ്ഞ ഗ്ലാസ് ട്യൂബിലൂടെ കടത്തിവിടുന്ന രൂപകല്പനയാണ് ഒപ്റ്റിക്കൽ ഫൈബറുകൾക്കുള്ളത്. പ്രകാശത്തിന്റെ വേഗതയിൽ ഡാറ്റയെ വളരെ ദൂരത്തേക്ക് സംപ്രേക്ഷണം ചെയ്യുവാൻ കഴിയുന്നു. ചിത്രം 8.5 ഒപ്റ്റിക്കൽ ഫൈബറിന്റെ പ്രധാന ഭാഗങ്ങൾ കാണിച്ചിരിക്കുന്നു.

ചിത്രം 8.5: ഒപ്റ്റിക്കൽ ഫൈബർ

ഒപ്റ്റിക്കൽ ഫൈബറിന് താഴെപ്പറയുന്ന ഭാഗങ്ങൾ ഉണ്ട്.

- കോർ: മധ്യഭാഗത്തു കൂടി പ്രകാശം കടന്നു പോകുന്ന കനം കുറഞ്ഞ ഗ്ലാസിന്റെ കുഴലാണ് ഇത്.
- ക്ലാഡിങ് : കോർ ഭാഗത്തെ പൊതിഞ്ഞു കൊണ്ട് പ്രകാശത്തെ കോറിനുള്ളിലേക്കു തന്നെ പ്രതിഫലിപ്പിക്കുന്ന പുറം ഭാഗമാണ് ഇത്.
- കോട്ടിങ് : ഈർപ്പത്തിൽ നിന്നും, തകരാറിൽ നിന്നും സംരക്ഷിക്കുന്നതിനായിട്ടുള്ള കേബിളിന്റെ പ്ലാസ്റ്റിക് കവചമാണ് ഇത്.

നൂറുകണക്കിനോ ആയിരക്കണക്കിനോ ആയ ഒപ്റ്റിക്കൽ ഫൈബർ കേബിളുകളെ പൊതിഞ്ഞിരിക്കുന്ന കവചത്തെ ജാക്കറ്റ് എന്ന് വിളിക്കുന്നു.

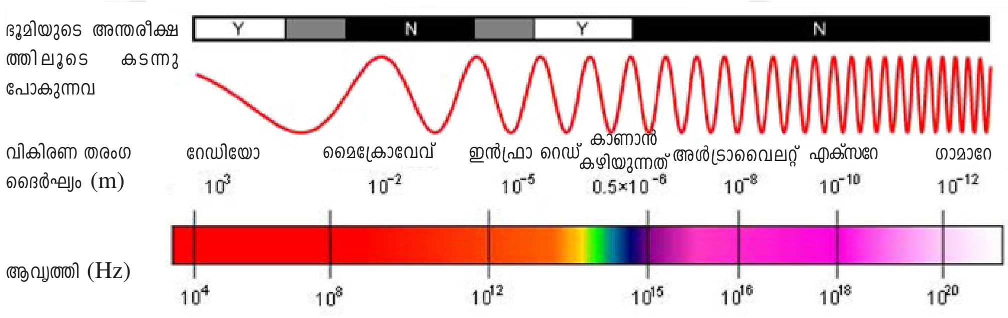
അർദ്ധചാലക ഉപകരണങ്ങളായ ലൈറ്റ് എമിറ്റിങ് ഡയോഡുകൾ (LED) ലേസർ ഡയോഡുകൾ എന്നിവ ഉപയോഗിച്ച് ഉത്ഭവ സ്ഥാനത്തുവെച്ചു ഒപ്റ്റിക്കൽ ട്രാൻസ്മിറ്റർ, വൈദ്യുത തരംഗങ്ങളെ പ്രകാശ തരംഗങ്ങൾ ആക്കി മാറ്റുന്നു (മോഡുലേഷൻ). മറുഭാഗത്ത്, ഫോട്ടോ ഡിറ്റക്ടർ അടങ്ങിയ ഒപ്റ്റിക്കൽ റിസീവർ, പ്രകാശ തരംഗങ്ങളെ ഫോട്ടോ ഇലക്ട്രിക്ക് പ്രഭാവം ഉപയോഗിച്ച് തിരികെ വൈദ്യുത തരംഗങ്ങൾ ആക്കി മാറ്റുന്നു (ഡീമോഡുലേഷൻ). ലേസർ ഡയോഡുകൾക്കു ദൂരപരിധിയും, കൈമാറ്റ വേഗതയും LED ഡയോഡുകളേക്കാൾ കൂടുതൽ ആണ് .

ഓപ്റ്റിക്കൽ ഫൈബർ കേബിളിന്റെ സവിശേഷതകൾ

- ഉയർന്ന ബാൻഡ് വിഡ്ത്തിൽ ശബ്ദവും, വീഡിയോയും ഡാറ്റയും കൈമാറുന്നു
- ഒറ്റയടിക്ക് ഡാറ്റയെ ദീർഘ ദൂരത്തേയ്ക്ക് എത്തിക്കുന്നു.
- ഡാറ്റ കൈമാറ്റം ചെയ്യുവാൻ പ്രകാശ കണികകൾ ഉപയോഗിക്കുന്നതിനാൽ വൈദ്യുത കാന്തിക തരംഗങ്ങളുമായി ഒരു കൂടിച്ചേരലും നടക്കുന്നില്ല.
- കമ്പ്യൂട്ടർ ശൃംഖലയ്ക്കു ലഭ്യമായതിൽ വെച്ച് ഏറ്റവും ചെലവേറിയതും കാര്യക്ഷമത കൂടിയതുമായ മാദ്ധ്യമമാണിത്.
- പരിപാലനവും സ്ഥാപിക്കലും (Maintenance and installation) പ്രയാസകരവും സങ്കീർണവുമാണ്.

8.3.2 അൺ ഗൈഡഡ് മീഡിയം (വയർരഹിതം) (Unguided medium (Wireless))

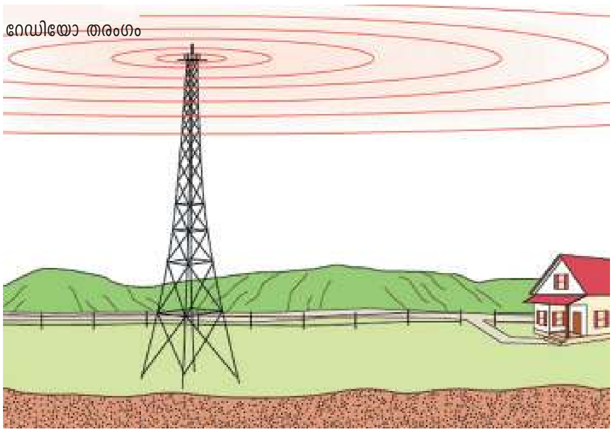
വൈദ്യുതകാന്തിക തരംഗങ്ങൾ ആണ് വയർലെസ്സ് വിവരവിനിമയത്തിനായി കമ്പ്യൂട്ടർ ശൃംഖലയിൽ ഉപയോഗിക്കുന്നത്. തരംഗദൈർഘ്യത്തെ ഹേർട്സ് (Hetrz (Hz))ൽ ആണ് കണക്കാക്കുന്നത്. ചിത്രം 8.6 ൽ ആവൃത്തിയെ അടിസ്ഥാനമാക്കി വിവിധ തരം വൈദ്യുതകാന്തികതരംഗങ്ങൾ കാണിച്ചിരിക്കുന്നു. ഈ വിഭാഗത്തിൽ വയർലസ് വിവര വിനിമയത്തിനായി, റേഡിയോ തരംഗങ്ങളും മൈക്രോ തരംഗങ്ങളും ഇൻഫ്രാറെഡ് തരംഗങ്ങളുമാണ് ഉപയോഗിക്കുന്നത് എന്ന് നമുക്ക് മനസ്സിലാക്കാം.



ചിത്രം 8.6 : വൈദ്യുത കാന്തിക വർണ്ണരാജി (സ്പെക്ട്രം)

a. റേഡിയോ തരംഗങ്ങൾ (Radio waves)

റേഡിയോ തരംഗങ്ങളുടെ ആവൃത്തി 3 KHz മുതൽ 3 GHz വരെയാണ്. റേഡിയോ തരംഗങ്ങൾ ഹ്രസ്വ / ദീർഘ ദൂര സംപ്രേക്ഷണത്തിനു ഉപയോഗിക്കുന്നു. ഇത്തരം തരംഗങ്ങളെ വളരെ എളുപ്പത്തിൽ ഉൽപ്പാദിപ്പിക്കാം എന്നതിന് പുറമെ അവയ്ക്കു തടസ്സങ്ങൾ മറികടക്കുവാനുള്ള കഴിവും ഉണ്ട്. ഇക്കാരണത്താൽ വിവരവിനിമയത്തിനായി എല്ലാ മേഖലയിലും (കെട്ടിടങ്ങൾക്ക് ഉള്ളിലും പുറത്തും) റേഡിയോ തരംഗങ്ങൾ ഉപയോഗിക്കുന്നു. കോഡ്‌ലസ് ഫോൺ, AM, FM റേഡിയോ സംപ്രേക്ഷണം, മൊബൈൽ ഫോൺ തുടങ്ങിയവയിൽ റേഡിയോ തരംഗങ്ങൾ ആണ് ഉപയോഗിക്കുന്നത്.

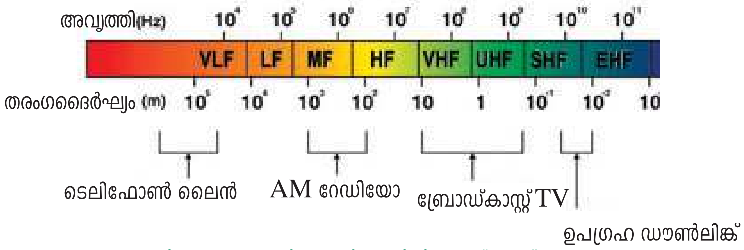


ചിത്രം 8.7 : റേഡിയോ തരംഗ പ്രസരണം

റേഡിയോ തരംഗങ്ങളുടെ വിവരവിനിമയ സവിശേഷതകൾ

- എല്ലാ ദിശയിലേക്കും റേഡിയോ തരംഗങ്ങൾക്ക് സഞ്ചരിക്കാൻ കഴിവുള്ളതിനാൽ, സ്വീകരിക്കുവാനും പ്രസാരണം ചെയ്യുവാനും ഉപയോഗിക്കുന്ന ഉപകരണങ്ങൾ നേർക്കുനേർ വരണമെന്നില്ല.
- വയർ അധിഷ്ഠിത മാധ്യമവുമായി താരതമ്യം ചെയ്യുമ്പോൾ ഇതിന് ചെലവ് കുറവാണ്.

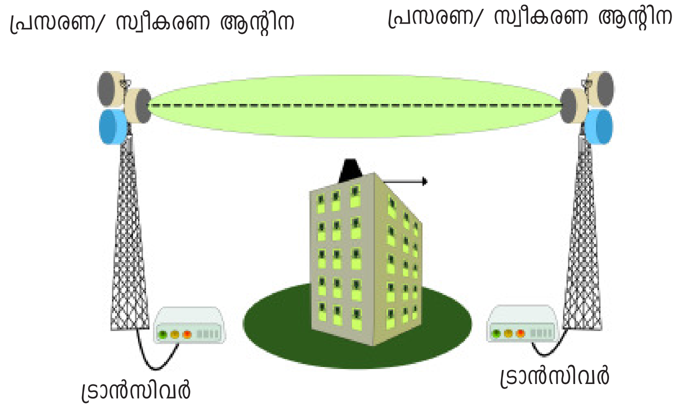
- മിക്ക വസ്തുക്കൾക്കുള്ളിലൂടെയും കടന്നു പോകുവാനുള്ള കഴിവുണ്ട്.
- പ്രസാരണത്തെ മോട്ടോറുകളും ഇലക്ട്രിക്ക് ഉപകരണങ്ങളും സ്വാധീനിക്കാൻ സാധ്യതയുണ്ട്.
- സുരക്ഷിതത്വം കുറഞ്ഞ വിനിമയ രീതിയാണ്.
- റേഡിയോ തരംഗങ്ങളുടെ ഉപയോഗത്തിന് ബന്ധപ്പെട്ട അധികാരികളുടെ അനുവാദം ആവശ്യമാണ്.



ചിത്രം 8.8: റേഡിയോ വിവരവിനിമയ സ്പെക്ട്രം

b. മൈക്രോ തരംഗങ്ങൾ (സൂക്ഷ്മതരംഗം) (Micro waves)

മൈക്രോ തരംഗങ്ങളുടെ ആവൃത്തി 300 MHz (0.3GHz) മുതൽ 300 GHz വരെയാണ്. മൈക്രോ തരംഗങ്ങൾ നേർരേഖയിൽ സഞ്ചരിക്കുന്നതും ഖരപദാർഥങ്ങൾക്കുള്ളിലൂടെ കടന്നു പോകാത്തതും ആണ്. ആയതിനാൽ വളരെ ഉയരം കൂടിയ ടവറുകൾ ഉണ്ടാക്കി അതിനു മുകളിൽ മൈക്രോവേവ് ആന്റിനകൾ ഉറപ്പിച്ചാണു



ചിത്രം 8.9: മൈക്രോവേവ് പ്രസരണം

ദീർഘ ദൂര പ്രസരണം സാധ്യമാക്കുന്നത്. തരംഗങ്ങൾ നേർരേഖയിൽ സഞ്ചരിക്കുന്നതിനാൽ പ്രസരണം ചെയ്യുന്നതിനും സ്വീകരിക്കുന്നതിനും ഉള്ള ആന്റിനകൾ പരസ്പരം അഭിമുഖമായാണ് സ്ഥാപിച്ചിരിക്കുന്നത്. രണ്ടു മൈക്രോവേവ് ടവറുകൾ തമ്മിലുള്ള അകലം നിശ്ചയിക്കുന്നത് തരംഗങ്ങളുടെ ആവൃത്തിയും ടവറുകളുടെ ഉയരവും അനുസരിച്ചാണ്. ചിത്രം 8.9 യിൽ ഒരു മൈക്രോവേവ് പ്രസരണ സംവിധാനത്തിന്റെ ഭാഗങ്ങൾ ചിത്രീകരിച്ചിരിക്കുന്നു.

മൈക്രോവേവ് സംപ്രേക്ഷണത്തിന്റെ സവിശേഷതകൾ

- വയേർഡ് മാധ്യമവുമായി താരതമ്യം ചെയ്യുമ്പോൾ ഇതിന് ചെലവ് കുറവാണ്
- ദുഷ്കരമായ ഭൂപ്രദേശങ്ങളിൽ സുഗമമായ വിവര വിനിമയം സാധ്യമാകുന്നു
- പ്രസാരണം നേർരേഖയിൽ ആയതിനാൽ പ്രസരണ ഉപകരണവും സ്വീകരണ ഉപകരണവും അഭിമുഖമായിത്തന്നെ സ്ഥാപിക്കണം.

c ഇൻഫ്രാറെഡ് തരംഗങ്ങൾ (Infrared waves)

ഇൻഫ്രാറെഡ് തരംഗങ്ങൾ 300 GHz മുതൽ 400 THz വരെ ആവൃത്തിയുള്ളവയാണ് ഹ്രസ്വദൂര സംപ്രേക്ഷണത്തിനാണ് ഇത് ഉപയോഗിക്കുന്നത് (ഏകദേശം 5m). ആപ്ലിക്കേഷനുകളെ നിയന്ത്രിക്കുവാനും വിലയിരുത്തുവാനും കൂടാതെ വിവിധതരത്തിലുള്ള വയർലെസ്സ് വിവരവിനിമയത്തിനും ഇത് ഉപയോഗിക്കുന്നു.

ചിത്രം 8.10 : ഇൻഫ്രാറെഡ് പ്രസരണം

വിവിധ ഗാർഹിക വിനോദ ഉപകരണങ്ങളിലെ റിമോട്ടുകൾ, കോർഡ്ലെസ് മൗസ്, അനധികൃതമായി കടന്നുകയറുന്നത് ശ്രദ്ധയിൽപ്പെടുത്തുന്ന ഉപകരണങ്ങൾ തുടങ്ങിയവയിൽ ഇൻഫ്രാറെഡ് തരംഗങ്ങൾ ഉപയോഗിക്കുന്നു. (ചിത്രം 8.10 പരിശോധിക്കുക)

ഇൻഫ്രാറെഡ് തരംഗങ്ങളുടെ സവിശേഷതകൾ

- നേർരേഖയിലുള്ള വിവര വിനിമയം നടക്കുന്നതിനാൽ, വിവരങ്ങൾ ചോർത്തപ്പെടുന്നില്ല.
- രണ്ടു ഉപകരണങ്ങൾക്ക് മാത്രമേ ഒരു സമയത്തു വിവര വിനിമയം നടത്തുവാൻ സാധിക്കൂ.
- ഖര പദാർത്ഥങ്ങളെ മറികടക്കുവാനുള്ള കഴിവില്ല (റിമോട്ട് കൺട്രോളിനും ടീവിയ്ക്കും ഇടയിൽനിന്നു കൊണ്ട് റിമോട്ട് കൺട്രോൾ പ്രവർത്തിക്കുന്നുണ്ടോ എന്ന് പരിശോധിക്കാവുന്നതാണ്).
- എത്തിപ്പെടാവുന്ന ദൂരം കൂടുന്തോറും തരംഗശേഷി കുറയുന്നു.

8.3.3 റേഡിയോ തരംഗങ്ങൾ ഉപയോഗിച്ചുള്ള വയർരഹിത വിനിമയ സംവിധാനം (Wireless communication technologies using radio waves)

a. ബ്ലൂടൂത്ത് (Bluetooth)

റേഡിയോ തരംഗങ്ങൾ ആണ് ബ്ലൂ ടൂത്ത് സംവിധാനത്തിൽ ഉപയോഗിക്കുന്നത്. ഇതിന്റെ ആവൃത്തി 2.402 GHz മുതൽ 2.480 GHz വരെയാണ്. ഹ്രസ്വ ദൂര വിവര വിനിമയത്തിന് ഉപയോഗിക്കുന്ന വയർലസ് ഉപകരണങ്ങളിൽ (ഏകദേശം 10m) ഇത് ഉപയോഗിക്കുന്നു. സെൽ ഫോൺ, ലാപ്ടോപ്, മൗസ്, കീബോർഡ്, ടാബ്ലെറ്റുകൾ, ഹെഡ്സെറ്റ്, ക്യാമറ, എന്നിവ ബ്ലൂടൂത്ത് ഉപയോഗിക്കുന്ന ചില ഉപകരണങ്ങൾ ആണ്. (ചിത്രം 8.11 പരിശോധിക്കുക.)

ചിത്രം 8.11 : ബ്ലൂ - ടൂത്ത് പ്രസരണം

ബ്ലൂടൂത്തിന്റെ വിനിമയ സവിശേഷതകൾ

- വിവരവിനിമയം നടത്തുവാൻ നേർരേഖയിൽ പ്രസരണ ഉപകരണങ്ങൾ സ്ഥാപിക്കേണ്ട ആവശ്യമില്ല.
- ബ്ലൂടൂത്ത് ഉപയോഗിച്ച് ഒരേ സമയം എട്ടു ഉപകരണങ്ങളേവരെ ബന്ധിപ്പിക്കാം.
- വേഗതകുറഞ്ഞ വിനിമയ മാർഗമാണ് ഇത് (1 Mbps വരെ).

b. വൈ-ഫൈ (Wi-Fi)

റേഡിയോ തരംഗങ്ങൾ ഉപയോഗിച്ച് ആണ് വൈ-ഫൈ ശൃംഖല പ്രവർത്തിക്കുന്നത്. സെൽഫോൺ, ടെലിവിഷൻ, റേഡിയോ തുടങ്ങിയ ഉപകരണങ്ങളിൽ വിവരങ്ങൾ കൈമാറ്റം ചെയ്യുവാൻ വൈ-ഫൈ ഉപയോഗിക്കുന്നു. വൈ-ഫൈ ശൃംഖലയിൽ ഉപയോഗിക്കുന്ന റേഡിയോ തരംഗങ്ങളുടെ ആവൃത്തി 2.4GHz മുതൽ 5 GHz വരെയാണ്. വയർലെസ്സ് ശൃംഖലയിൽ ഇരുദിശകളിലേക്കും ഉള്ള വിവരവിനിമയമാണ് നടക്കുന്നത്.

കമ്പ്യൂട്ടറിൽ ഉള്ള വയർലെസ്സ് അഡാപ്റ്റർ ഡാറ്റയെ റേഡിയോ തരംഗങ്ങൾ ആക്കി മാറ്റുകയും അവയെ ഒരു ആന്റിന ഉപയോഗിച്ച് സംപ്രേക്ഷണം ചെയ്യുകയും ചെയ്യുന്നു. വയർലെസ് റൂട്ടർ ഇവയെ സ്വീകരിച്ച് പരിവർത്തനം ചെയ്യുന്നു. പരിവർത്തനം ചെയ്യപ്പെട്ട ഡാറ്റയെ ഇന്റർനെറ്റിലേക്കോ, ശൃംഖലയിലേക്കോ ഒരു വയേർഡ് ഈതർനെറ്റ് (ethernet) /വയർലെസ്സ് കണക്ഷൻ വഴി അയയ്ക്കപ്പെടുന്നു. ഇതുപോലെ ഇന്റർനെറ്റ് വഴി ലഭിക്കുന്ന ഡാറ്റ റൂട്ടർ വഴികടന്നു പോകുകയും, അവയെ റേഡിയോ തരംഗങ്ങൾ ആക്കി ഒരു കമ്പൂട്ടറിൽ ഉള്ള വയർലസ്സ് ആഡാപ്റ്റർ സ്വീകരിക്കുന്നത് ചിത്രം 11.12 ൽ സൂചിപ്പിച്ചിരിക്കുന്നു. ഇപ്പോൾ ഈ സാങ്കേതികവിദ്യ ലാപ്ടോപ്പിലും ഡെസ്ക്ടോപ്പിലും ഇന്റർനെറ്റ് കണക്ഷൻ പങ്കിടുവാൻ വ്യാപകമായി ഉപയോഗിക്കുന്നു.

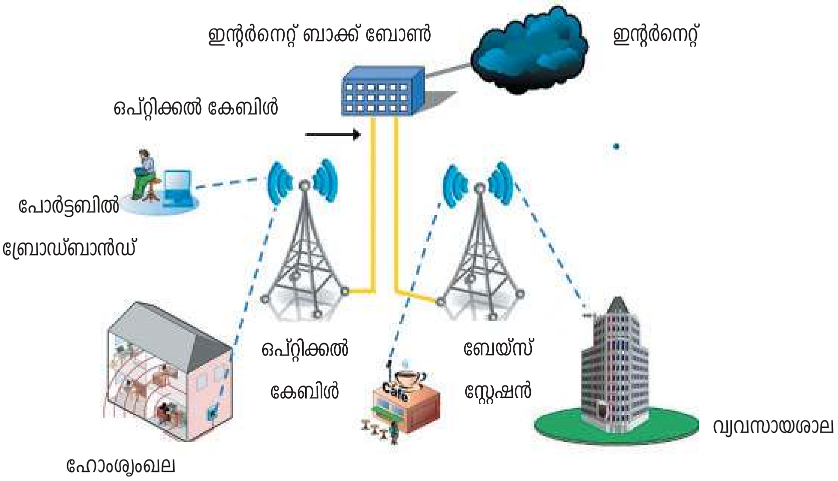
ചിത്രം 8.12: വൈ - ഫൈ പ്രസരണം

വൈ-ഫൈ പ്രസരണത്തിന്റെ സവിശേഷതകൾ

- ഉപകരണങ്ങൾ തമ്മിലുള്ള നേർകാഴ്ച ഇവിടെ ആവശ്യമില്ല.
- സംപ്രേക്ഷണത്തിന്റെ വേഗത 54Mbps വരെയാണ്.
- ഒരേ സമയം കൂടുതൽ ഉപകരണങ്ങളെ വൈ ഫൈ ഉപയോഗിച്ച് ബന്ധിപ്പിക്കാം.
- 114m (375 അടി) വരെയുള്ള വിനിമയത്തിന് ഉപയോഗിക്കുന്നു.

c. വൈ-മാക്സ് (Wi-MAX)

വേൾഡ് വൈഡ് ഇന്ററോപ്പറബിളിറ്റി ഫോർ മൈക്രോവേവ് അക്സസ് (വൈ-മാക്സ്)ന്റെ അടിസ്ഥാനം 802.16e ആണ്. ബ്രോഡ്ബാൻഡിന്റെയും വയർലെസ്സിന്റെയും സവിശേഷതകൾ സംയോജിപ്പിച്ചാണ് വൈ-മാക്സിനു രൂപം കൊടുത്തിരിക്കുന്നത്. വൈ-മാക്സിന്റെ ആവൃത്തി 2GHz മുതൽ 11 GHz വരെയാണ്. വൈ-മാക്സ് അതിവേഗത്തിലും ദീർഘ ദൂരത്തിലും ഇന്റർനെറ്റ് ഉപയോഗം സാധ്യമാക്കുന്നു (നഗരത്തിലുടനീളം). അടിസ്ഥാനതലത്തിൽ വൈ മാക്സിനു രണ്ട് തരത്തിലുള്ള സജ്ജീകരണങ്ങൾ ആണ് ഉള്ളത്. സേവനദാതാവ് സാങ്കേതികവിദ്യ വിന്യസിക്കുവാൻ ആയി ഒരു പ്രത്യേക മേഖലയിൽ ഉപയോഗിച്ചിരിക്കുന്ന ഉപകരണങ്ങളും, ഉപഭോക്താവ് സ്ഥാപിച്ചിരിക്കുന്ന സ്വീകരണ ഉപകരണങ്ങളും ചേർന്നതാണ് ബെയ്സ് സ്റ്റേഷൻ. വൈ-മാക്സ് പ്രസരണത്തിന് ഉപയോഗിക്കുന്ന അടിസ്ഥാന ഉപകരണങ്ങൾ ചുവടെ ചിത്രം 8.13 ചിത്രീകരിച്ചിരിക്കുന്നു.



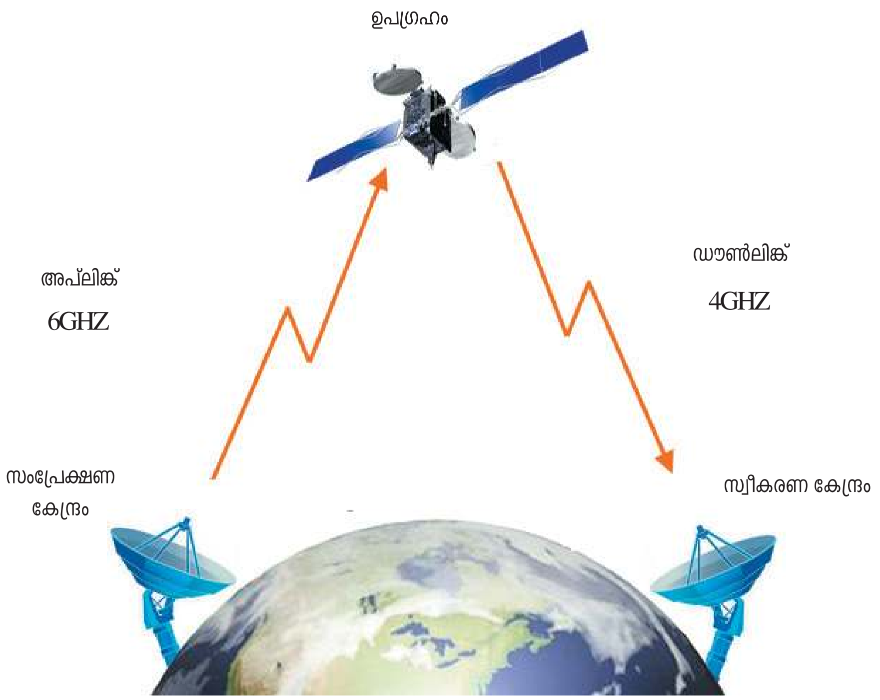
ചിത്രം 8.13 : വൈ - മാക്സ് സംപ്രേഷണം

വൈ-മാക്സ് സംപ്രേക്ഷണത്തിന്റെ സവിശേഷതകൾ

- 100 കണക്കിന് ഉപഭോക്താക്കൾക്ക് ഒരു സംപ്രേക്ഷണ നിലയവുമായി ബന്ധപ്പെടുവാൻ കഴിയുന്നു.
- 45 KM പരിധിയിൽ 70 Mbps വരെ വേഗത്തിൽ വിവരവിനിമയം നടക്കുന്നു.
- ഉപകരണങ്ങൾ തമ്മിൽ നേർരേഖയിൽ ഉള്ള വിനിമയം ഇവിടെ ആവശ്യമില്ല.
- സംപ്രേക്ഷണത്തെ മഴ, കാറ്റ് തുടങ്ങിയ പ്രതികൂല കാലാവസ്ഥ തടസ്സപ്പെടുത്തുന്നു.
- അമിതമായി ഊർജം ഉപയോഗിക്കുന്നു.
- സ്ഥാപിക്കുവാനും പ്രവർത്തിപ്പിക്കുവാനും ഉള്ള ഉയർന്ന ചെലവ്.

d. ഉപഗ്രഹ സംപ്രേക്ഷണം (Satellite link)

ദീർഘദൂര വിനിമയത്തിന് ഉപഗ്രഹശൃംഖല ഉപയോഗിച്ച് ഡാറ്റാ കൈമാറ്റം ചെയ്യപ്പെടുന്നു. സാധാരണയായി ഡാറ്റ നേർരേഖയിൽ ആണ് സഞ്ചരിക്കുന്നത്, ആയതിനാൽ ഭൂമിയെ വലം വെച്ച് വിദൂരതയിൽ ഉള്ള ഉദ്ദേശ്യ ലക്ഷ്യത്തിൽ എത്തുവാനുള്ള കഴിവ് ഡാറ്റയ്ക്ക് ഉണ്ടാവില്ല. ഇങ്ങനെയുള്ള സന്ദർഭങ്ങളിൽ ഡാറ്റയെ ഭൂസ്ഥിര ഉപഗ്രഹങ്ങളിലേക്ക് അയയ്ക്കുകയും, ഉപഗ്രഹം അടുത്ത ഉപഗ്രഹങ്ങളിലേക്കോ, വിദൂരതയിലുള്ള ലക്ഷ്യത്തിലേക്കോ എത്തിക്കുകയും ചെയ്യുന്നു. ഭൂമിയുടെ ഭ്രമണപഥത്തിൽ അതേ ദിശയിലും ഭ്രമണ വേഗതയിലും സഞ്ചരിക്കുന്ന ഉപഗ്രഹങ്ങളെ ഭൂസ്ഥിര ഉപഗ്രഹങ്ങൾ എന്ന് പറയുന്നു. ഇത്തരത്തിലുള്ള ഉപഗ്രഹങ്ങൾ ഭൂമിക്ക് മുകളിൽ നിശ്ചിത സ്ഥാനത്തുതന്നെ സ്ഥിരമായി കാണപ്പെടുന്നു. ഈ ഉപഗ്രഹങ്ങളിലെ ഇലക്ട്രോണിക് ഉപകരണങ്ങൾ ആയ ട്രാൻസ്പോണ്ടറുകൾ ഉപയോഗിച്ച് ഡാറ്റ സ്വീകരിക്കുകയും, തരംഗങ്ങളുടെ ശക്തി വർദ്ധിപ്പിച്ച് (ആംപ്ലിഫൈയിങ്), ഭൂമിയിലേക്ക് പുനഃസംപ്രേക്ഷണം നടത്തുകയും ചെയ്യുന്നു.



ചിത്രം 8.14 : ഉപഗ്രഹ സംപ്രേഷണം

ഭൂമിയിൽ നിന്നും ഉപഗ്രഹത്തിലേക്കു തരംഗങ്ങളെ അയയ്ക്കുന്നതിനെ അപ് ലിങ്ക് എന്നും. ഉപഗ്രഹത്തിൽ നിന്ന് ഭൂമിയിലേക്ക് സംപ്രേക്ഷണം ചെയ്യുന്നതിനെ ഡൗൺ ലിങ്ക് എന്നും പറയുന്നു. ഒന്നിൽ കൂടുതൽ മൈക്രോ വേവ് ആവർത്തി തരംഗങ്ങൾ ഉപഗ്രഹ സംപ്രേക്ഷണത്തിനായി ഉപയോഗിക്കുന്നു. അപ് ലിങ്കിനു വേണ്ടി ഉപയോഗിക്കുന്ന ആവൃത്തി 1.6GHz മുതൽ 30.0 GHz വരെയും ഡൗൺ ലിങ്കിനു വേണ്ടിയുള്ളത് 1.5 GHz മുതൽ 20 GHz വരെയുമാണ്. ഡൗൺ ലിങ്കിന്റെ ആവൃത്തി അപ്‌ലിങ്കിനേക്കാൾ കുറവായിരിക്കും.

ഉപഗ്രഹ സംവിധാനം ചെലവേറിയതാണ്, പക്ഷെ വളരെ കൂടിയ വ്യാപ്തിയിൽ സേവനം ലഭ്യമാക്കുവാൻ കഴിയും. പല രാജ്യങ്ങളിലും സാധാരണ, സർക്കാരുകളുടെയോ, സർക്കാർ അംഗീകരിച്ചസ്ഥാപനങ്ങളുടെയോ നിയന്ത്രണത്തിലായിരിക്കും വാർത്താവിനിമയ ഉപഗ്രഹങ്ങൾ.

ഉപഗ്രഹ സംപ്രേക്ഷണത്തിന്റെ സവിശേഷതകൾ

- വളരെ വലിയ വ്യാപ്തിയിൽ ഉപഗ്രഹങ്ങൾ ഉപയോഗിച്ച് വിവര വിനിമയം നടത്തുവാൻ സാധിക്കുന്നു.
- ഈ സംവിധാനം ചെലവേറിയതാണ്.
- നിയമപരമായ അംഗീകാരവും അനുമതിയും ആവശ്യമാണ്.

നമുക്കു ചെയ്യാം

ഇൻസ്റ്റിറ്റ്യൂട്ട് ഓഫ് ഇലക്ട്രിക്കൽ ആൻഡ് ഇലക്ട്രോണിക്സ് എഞ്ചിനീയേഴ്സ് എന്ന സംഘടന നിർവചിച്ച വയർലെസ് ബ്രോഡ്ബാൻഡ് സാങ്കേതികതയുടെ അടിസ്ഥാന നിർവചനമാണ് IEEE 802.16e എന്നത്. വയർലെസ് മെട്രോപൊളിറ്റൻ ഏരിയ ശൃംഖലയുടെ അടിസ്ഥാന നിർവചനം നൽകുവാനാണ് 1999 ൽ ഈ സംഘടന രൂപീകൃതമായത്.

സ്വയം വിലയിരുത്താം

1. ഡാറ്റയുടെ വിനിമയ വ്യവസ്ഥയ്ക്ക് ആവശ്യമായ ഘടകങ്ങൾ ഏവ?
2. വിഭവങ്ങളുടെ പങ്കിടൽ (resource sharing) നിർവചിക്കുക.
3. കമ്പ്യൂട്ടർ ശൃംഖലയിൽ ഉപയോഗിക്കുന്ന രണ്ടു വ്യത്യസ്ത വിനിമയ മാധ്യമങ്ങൾ ഏതൊക്കെ?
4. UTP/STP കേബിളിനെ കമ്പ്യൂട്ടറുമായി ബന്ധിപ്പിക്കുവാൻ ഉപയോഗിക്കുന്ന കണക്ടർ ഏത്?
5. വളരെ ദൂരത്തേക്ക് പ്രകാശ തരംഗങ്ങൾ ഉപയോഗിച്ച് ഡാറ്റ തരംഗങ്ങൾ അയക്കുവാനുള്ള ഗൈഡ് മാധ്യമമാണ് __________.
6. AM/FM റേഡിയോ സംപ്രേക്ഷണത്തിനും മൊബൈലിലും വിനിമയത്തിനായി ഉപയോഗിക്കുന്ന മാധ്യമമാണ് __________
7 ടീവിയിലെ റിമോട്ട് കൺട്രോൾ, മൗസ് തുടങ്ങിയവയിൽ ഉപയോഗിക്കുന്ന മാധ്യമമാണ് __________.
8. സംപ്രേക്ഷണ ഉപകരണങ്ങൾ തമ്മിൽ നേർരേഖ കാഴ്ച ആവശ്യമില്ലാത്ത ഹ്രസ്വദൂര വിനിമയ സാങ്കേതികവിദ്യയാണ് __________.
9 ചെലവേറിയതും എന്നാൽ മറ്റു വയർലെസ്സ് സാങ്കേതികവിദ്യയേക്കാൾ കൂടുതൽ വ്യാപ്തിയിൽ സേവനം നടത്തുവാൻ കഴിയുന്നതുമായ വിവരവിനിമയ സാങ്കേതികവിദ്യയാണ് __________.

8.4 ഡാറ്റ വിനിമയ ഉപകരണങ്ങൾ (Data communication devices)

കമ്പ്യൂട്ടറും വിനിമയ മാധ്യമവും തമ്മിലുള്ള സമ്പർക്കമുഖ (interface) മായി ഒരു ഡാറ്റ വിനിമയ ഉപകരണം പ്രവർത്തിക്കുന്നു. ഈ ഉപകരണങ്ങൾ ഉപയോഗിച്ച് ഡാറ്റ തരംഗങ്ങളെ സംപ്രേക്ഷണം ചെയ്യുവാനും, സ്വീകരിക്കുവാനും, ശക്തി കൂട്ടുവാനും വിവിധ വിനിമയ മാധ്യമ ശൃംഖലകൾ ഉപയോഗിച്ച് വഴിതിരിച്ചു വിടുവാനും കഴിയുന്നു

8.4.1 നെറ്റ്വർക്ക് ഇന്റർഫേസ് കാർഡ് (Network Interface Card (NIC))

കമ്പ്യൂട്ടർ ശൃംഖലയിലേക്ക് ഒരു കമ്പ്യൂട്ടറിനെ ബന്ധിപ്പിക്കുവാനും വിവര വിനിമയം നടത്തുവാനും പ്രാപ്തമാക്കുന്ന ഉപകരണമാണ് NIC. കമ്പ്യൂട്ടറിനും ശൃംഖലയ്ക്കും

ഇടയിലുള്ള ഹാർഡ്‌വെയർ ഇന്റർഫേസ് ഉപകരണമായി ഇത് പ്രവർത്തിക്കുന്നു. ഇത് കമ്പ്യൂട്ടറിലെ പ്രത്യേക ഭാഗമായോ മദർബോർഡിന്റെ ഭാഗമായോ സ്ഥാപിച്ചിരിക്കുന്നു. കമ്പ്യൂട്ടർ ശൃംഖലയിലേക്കു ഡാറ്റയെ സജ്ജമാക്കുവാനും അയയ്ക്കുവാനും, സ്വീകരിക്കുവാനും നിയന്ത്രിക്കുവാനും NIC യ്ക്കു കഴിയും. ഡാറ്റയെ നിയന്ത്രിത രൂപത്തിലുള്ള ഘടകങ്ങളാക്കി മാറ്റുകയും, പ്രൊട്ടോക്കോളിനു വിധേയമായി പരിവർത്തനപ്പെടുത്തി, അയയ്ക്കേണ്ട മാധ്യമത്തിലേക്ക്, മേൽവിലാസം തിരിച്ചറിയുവാനുള്ള കഴിവുണ്ടാക്കി നൽകുകയും ചെയ്യുന്നു.

ചിത്രം. 8.15 (a) : NIC കാർഡ്

ചിത്രം. 8.15 (b) : വയർലസ്സ് NIC കാർഡ്

ചിത്രം 8.15(a), 8.5(b) എന്നിവയിൽ യഥാക്രമം ഒരു NIC കാർഡിന്റെയും ഒരു വയർലെസ്സ് NIC കാർഡിന്റെയും ചിത്രങ്ങൾ കൊടുത്തിരിക്കുന്നു. ചില NIC കാർഡുകൾ കേബിൾ ഉപയോഗിച്ചും (Ethernet), ചിലതു കേബിൾ ഇല്ലാതെയും (Wi-Fi) പ്രവർത്തിക്കുന്നു. കേബിൾ ശൃംഖലയിലേക്കുള്ള ജാക്കുകൾ ആണ് ഈതർനെറ്റ് NIC യിൽ ഉള്ളത്. എന്നാൽ വയർരഹിതമായ വിനിമയത്തിനുള്ള ബിൽറ്റ്-ഇൻ-ട്രാൻസ്മിറ്ററുകളും റീസിവറുകളും ആന്റിനയുമാണ് വൈ-ഫൈ NIC യിൽ ഉള്ളത്. NIC യ്ക്ക് 1Gbps വേഗതയിൽ ഡാറ്റ കൈമാറ്റം ചെയ്യുവാൻ കഴിയുന്നു.

8.4.2 ഹബ് (Hub)

ഒരു വയേർഡ് ശൃഖലയിൽ ഉൾപ്പെട്ടിരിക്കുന്ന കമ്പ്യൂട്ടറുകളെയും ഉപകരണങ്ങളെയും പരസ്പരം ബന്ധിപ്പിക്കുവാൻ ഉപയോഗിക്കുന്ന ഉപകരണമാണ് ഹബ്. ചെറുതും ലളിതവും നിഷ്ക്രിയവും വിലകുറഞ്ഞതുമായ ഉപകരണമാണ് ഇത്. ചിത്രം 8.16 പരിശോധിക്കുക. കമ്പ്യൂട്ടറുകളെ ഹബിലെ പോർട്ട് വഴി ഈതർനെറ്റ് കേബിൾ ഉപയോഗിച്ച് ബന്ധിപ്പിക്കുന്നു. ഹബിലേക്കു വരുന്ന വിവരങ്ങളുടെ പകർപ്പുകൾ പ്രസ്തുത ശൃംഖലയിൽ ഉൾപ്പെട്ടിരിക്കുന്ന എല്ലാ കമ്പ്യൂട്ടറുകളിലേക്കും കൈമാറുകയാണ് ഹബ് ചെയ്യുന്നത്. ഓരോ കമ്പ്യൂട്ടറിനും അവരവരുടെ ഡാറ്റ പാക്കറ്റുകൾ തിരിച്ചറിയുവാനുള്ള ബാധ്യതയുണ്ട്. ഒരു കമ്പ്യൂട്ടറിനെ ഉദ്ദേശിച്ച് അയച്ച പാക്കറ്റുകൾ അവ തന്നെ സ്വീകരിക്കേണ്ടതും മറ്റു കമ്പ്യൂട്ടറുകൾ അത് തിരസ്കരിക്കേണ്ടതും ആണ്. കമ്പ്യൂട്ടർ ശൃംഖലയിലെ എല്ലാ ഉപകരണങ്ങളിലേക്കും എല്ലാ ഡാറ്റയും അയയ്ക്കുന്നതിനാൽ ശൃംഖല തിരക്കേറിയതായിത്തീരുകയും ഡാറ്റ കൈമാറുവാനുള്ള ബാൻഡ് വിഡ്ത് കുറയുകയും ചെയ്യുന്നു എന്നതാണ് ഹബിന്റെ പ്രധാന പോരായ്മ.

ചിത്രം 8.16 : ഹബ്

8.4.3 സ്വിച്ച് (Switch)

നിരവധി കമ്പ്യൂട്ടറുകളെ പരസ്പരം ബന്ധിപ്പിച്ചു ഒരു ശൃംഖല രൂപീകരിക്കുവാൻ ശേഷിയുള്ള നിർമ്മിത ബുദ്ധിയോടുകൂടിയ ഉപകരണമാണ് സ്വിച്ച്. ഹബിനെക്കാൾ ഉയർന്ന പ്രവർത്തശേഷിയുള്ള ഉപകരണമാണ് സ്വിച്ച്. കാഴ്ചയിൽ ഹബിനോട് അടുത്ത സാമ്യമുണ്ട്. എന്നാൽ സ്വിച്ച് ഡാറ്റയ്ക്ക് എത്തിച്ചേരേണ്ട ലക്ഷ്യ സ്ഥാനം കൃത്യമായി ഉറപ്പു വരുത്തുകയും, ഡാറ്റ പാക്കറ്റുകൾ ഉദ്ദേശിച്ച സ്ഥാനത്തേയ്ക്ക് മാത്രം അയയ്ക്കുകയും ചെയ്യുന്നു. ശൃംഖലയിൽ ബന്ധിപ്പിച്ചിട്ടുള്ള എല്ലാ ഉപകരണങ്ങളുടെയും വിലാസം പട്ടികയായി സംഭരിച്ചു വെയ്ക്കുന്നതിനാലാണ് സ്വിച്ചിനു ഇങ്ങനെ പ്രവർത്തിക്കുവാൻ കഴിയുന്നത്. ശൃംഖലയിലെ ഒരു ഉപകരണത്തിലേക്കു ഡാറ്റ അയയ്ക്കുവാനായി, സ്വിച്ച് ഈ പാക്കറ്റിലെ വിലാസം മുൻകൂട്ടി ശേഖരിച്ച വിലാസങ്ങളുമായി താരതമ്യം ചെയ്യുന്നു, വിലാസം കണ്ടെത്തിയാൽ ലക്ഷ്യ സ്ഥാനത്തുള്ള ഉപകരണത്തിലേക്കു മാത്രം ഡാറ്റ അയയ്ക്കുന്നു. വളരെ തിരക്ക് കൂടിയ കമ്പ്യൂട്ടർ ശൃംഖലയിൽ ഹബ്ബിനെക്കാൾ നന്നായി സ്വിച്ച് പ്രവർത്തിക്കുന്നു. കാരണം വളരെ കുറഞ്ഞ അളവിൽ സന്ദേശങ്ങൾ അയയ്ക്കുന്നതിനാൽ ശൃംഖലയിൽ തിരക്ക് ഉണ്ടാകുന്നില്ല.

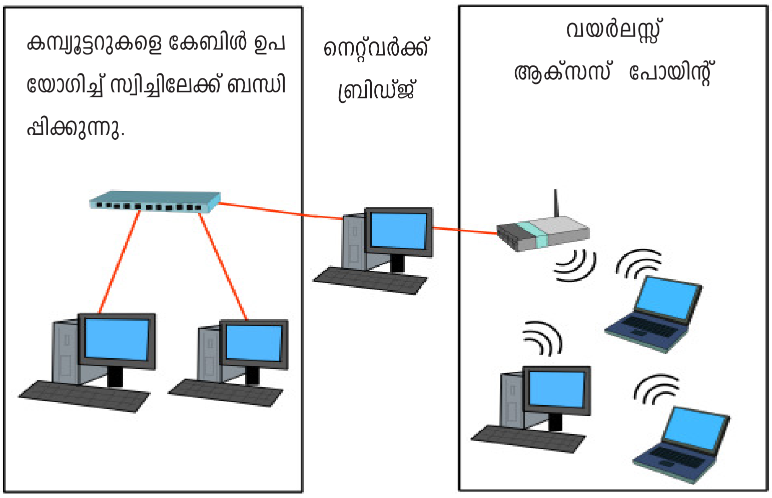
8.4.4 റിപ്പീറ്റർ (Repeater)

വിനിമയമാധ്യമത്തിലൂടെ വരുന്ന വൈദ്യുത കാന്തിക പ്രകാശ തരംഗങ്ങളെ ശക്തിപ്പെടുത്തുന്ന ഉപകരണമാണ് റിപ്പീറ്റർ. (ചിത്രം 8.17) വയേർഡ് മാധ്യമത്തിലൂടെയോ വയർലെസ്സിലൂടെയോ ഡാറ്റയ്ക്കു പരിമിതമായ ദൂരത്തേക്ക് മാത്രമേ ശക്തി ക്ഷയിക്കാതെ സഞ്ചരിക്കുവാൻ കഴിയൂ. ഇതിനു കാരണം നോയ്സ് ആണ്. റിപ്പീറ്റർ ഇങ്ങനെ വരുന്ന തരംഗങ്ങളെ സ്വീകരിച്ചു ശക്തി കൂട്ടി ലക്ഷ്യ സ്ഥാനത്തേയ്ക്ക് പുനഃസംപ്രേക്ഷണം നടത്തുന്നു.

ചിത്രം 8.17: വയർലസ്സ് റിപ്പീറ്റർ

8.4.5 ബ്രിഡ്ജ് (Bridge)

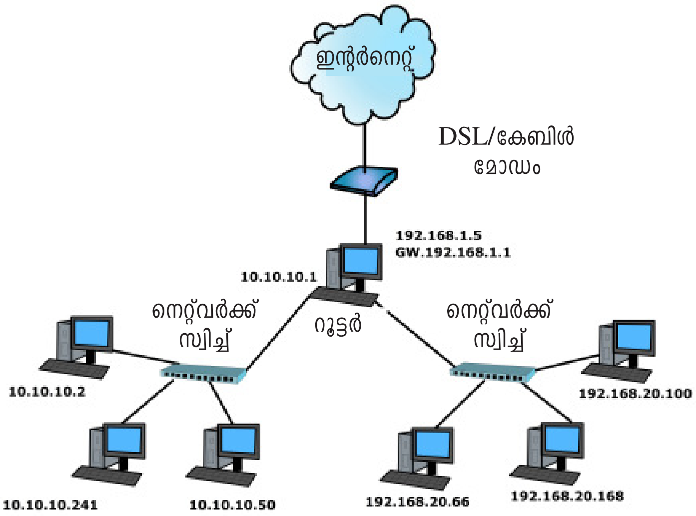
ഒരു കമ്പ്യൂട്ടർ ശൃംഖലയെ പല വിഭാഗങ്ങളാക്കി വേർതിരിക്കുവാൻ ഉപയോഗിക്കുന്ന ഉപകരണമാണ് ബ്രിഡ്ജ്. നിലവിലുള്ള ശൃംഖലയെ പല വിഭാഗങ്ങളായി തരംതിരിക്കുകയും ഇവയെ ബ്രിഡ്ജ് ഉപയോഗിച്ച് ബന്ധിപ്പിക്കുകയും ചെയ്യുന്നു. ശൃംഖലയിലുള്ള ട്രാഫിക് കുറയ്ക്കുവാൻ ഇത് സഹായിക്കുന്നു. ഒരു ബ്രിഡ്ജിൽ ഡാറ്റ പാക്കറ്റുകൾ എത്തുമ്പോൾ, അതിലെ മേൽവിലാസം പരിശോധിച്ചു ബ്രിഡ്ജിന്റെ ഏതു ഭാഗത്തെ ഇവ പ്രതിനിധാനം ചെയ്യുന്നു എന്ന് കണ്ടുപിടിക്കുന്നു (ഇതേ ഭാഗത്തേക്കുള്ള നോഡുകളിലേക്കോ അതോ മറുഭാഗത്തേയ്ക്കോ എന്ന്). ഒരു മേഖലയെ പ്രതിനിധാനം ചെയ്യുന്ന ഡാറ്റ പാക്കറ്റുകളെ മാത്രം ആ ഭാഗത്തേയ്ക്ക് ബ്രിഡ്ജ് കടത്തി വിടുന്നു. ബാക്കി ഉള്ളവ ഒഴിവാക്കുന്നു. ബ്രിഡ്ജ് വഴി കടന്നു പോകുന്ന പാക്കറ്റുകൾ മറു ഭാഗത്തുള്ള എല്ലാ നോഡുകളിലേക്കും പ്രക്ഷേപണം ചെയ്യുകയും, ലക്ഷ്യത്തിലുള്ള നോഡുകൾ മാത്രം അവ സ്വീകരിക്കുകയും ചെയ്യുന്നു. ചിത്രം 8.18 ബ്രിഡ്ജിന്റെ ധർമ്മങ്ങൾ വിശദമാക്കുന്നു.



ചിത്രം 8.18: ബ്രിഡ്ജ്

8.4.6 റൂട്ടർ (Router)

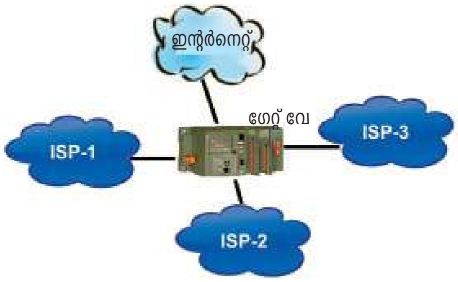
ഒരേ വിഭാഗത്തിൽപ്പെട്ടതും ഒരേ പോലുള്ള പെരുമാറ്റ ചട്ടങ്ങൾ ഉള്ളതുമായ രണ്ടു ശൃംഖലകളെ ബന്ധിപ്പിക്കുന്ന ഉപകരണമാണ് റൂട്ടർ. ഡാറ്റയ്ക്ക് സഞ്ചരിക്കുവാനാവ ശ്യമായ ഉചിതമായ പാത കണ്ടെത്തുന്നതിനും അങ്ങനെ ശൃംഖലയിലെ ട്രാഫിക്കിന്റെ അളവ് കുറയ്ക്കുന്നതിനും ഇവയ്ക്കു കഴിയുന്നു. ബ്രിഡ്ജിന്റെ പ്രവർത്തന രീതികളോട് ഇവയ്ക്കു സാമ്യം ഉണ്ടെങ്കിലും അവയേക്കാൾ കഴിവ് ഇതിനുണ്ട്. റൂട്ടറിനു ഉപകരണ ത്തിന്റെ വിലാസവും, ശൃംഖലയുടെ വിലാസവും പരിശോധിക്കുവാനുള്ള കഴിവുള്ളതോ ടൊപ്പം അൽഗോരിതം ഉപയോഗിച്ച് ഏറ്റവും ഉചിതമായ പാതയിലൂടെ പാക്കറ്റുകളെ ലക്ഷ്യ സ്ഥാനത്ത് എത്തിക്കുവാനും സാധിക്കുന്നു. ചിത്രം 8.19 റൂട്ടറിന്റെ ധർമം വിശദമാ ക്കുന്നു.



ചിത്രം 8.19: റൂട്ടർ

8.4.7 ഗേറ്റ്‌വേ (Gateway)

വിവിധ തരത്തിലും പ്രൊട്ടോക്കോളിലും പ്രവർത്തിക്കുന്ന ശൃംഖലകളെ ബന്ധിപ്പിക്കുവാൻ ഗേറ്റ്‌വേ ഉപയോഗിക്കുന്നു. ചിത്രം 8.20 പരിശോധിക്കുക. ഒരു തരത്തിലുള്ള പ്രൊട്ടോക്കോളിനെ മറ്റൊരു തരത്തിലേക്ക് വിവർത്തനം ചെയ്യുവാനും ഇവയ്ക്കു കഴിയുന്നു. ഒരു ശൃംഖലയിൽ നിന്ന് മറ്റൊരു ശൃംഖലയിലേക്കുള്ള പ്രവേശന കവാടമായി ഇത് പ്രവർത്തിക്കുന്നു. റൂട്ടറിനു സമാനമായ പ്രവർത്തനരീതിയാണ് ഇവയ്ക്കും ഉള്ളത്. ഉപകരണത്തിന്റെയും ശൃംഖലയുടെയും വിലാസം പരിശോധിക്കുകയും അൽഗോരിതത്തിന്റെ സഹായത്താൽ ഉചിതമായ പാത സ്വീകരിച്ചു പാക്കറ്റുകളെ ലക്ഷ്യ സ്ഥാനത്തു എത്തിക്കുകയും ചെയ്യുന്നു. വ്യത്യസ്തമായ പ്രൊട്ടോക്കോളുള്ള ശൃംഖലകൾ തമ്മിൽ ഒരു പരസ്പര ധാരണ ഉണ്ടായിരിക്കണം. ഒരു ഗേറ്റ്‌വേയ്ക്ക് ശൃംഖലയുടെ വിലാസ ഘടനയെ കുറിച്ച് ശരിയായ ധാരണ ഉള്ളതിനാൽ തടസ്സം ഇല്ലാതെ തുടർച്ചയായി പാക്കറ്റുകളെ ശൃംഖലയിലെ നോഡുകൾക്കിടയിൽ കൈമാറ്റം ചെയ്യുവാനുള്ള കഴിവുണ്ട്.



ചിത്രം 8.20 : ഗേറ്റ് വേ

11.5 ഡാറ്റ ടെർമിനൽ ഉപകരണങ്ങൾ (Data Terminal Equipments (DTE))

കമ്പ്യൂട്ടറിലേക്കും പുറത്തേയ്ക്കും ഉള്ള ഡാറ്റയുടെ ഒഴുക്കിനെ നിയന്ത്രിക്കുന്ന ഉപകരണമാണ് ഡാറ്റ ടെർമിനൽ ഉപകരണങ്ങൾ (Data Terminal Equipments (DTE)). ഈ ഉപകരണങ്ങൾ ടെലികമ്മ്യൂണിക്കേഷൻ ലിങ്കുമായി സംപ്രേക്ഷണ മാധ്യമത്തിന്റെ അഗ്രഭാഗത്തു ബന്ധിപ്പിച്ചിരിക്കുന്നു. പൊതുവായി ഉപയോഗിക്കുന്ന DTE ഉപകരണങ്ങളായ മോഡം, മൾട്ടിപ്ലെക്സർ എന്നിവയെ കുറിച്ച് ഇവിടെ ചർച്ച ചെയ്യുന്നു.

8.51. മോഡം (Modem)

ടെലിഫോൺ ലൈൻ ഉപയോഗിച്ച് കമ്പ്യൂട്ടറുകൾ തമ്മിൽ വിനിമയം നടത്തുവാൻ സഹായിക്കുന്ന ഇലക്ട്രോണിക് ഉപകരണമാണ് മോഡം. (ചിത്രം 8.21). മോഡുലേറ്റർ

ചിത്രം 8.21 : മോഡം ഉപയോഗിച്ചുള്ള ആശയവിനിമയം

(Modulator)ഡി മോഡുലേറ്റർ (Demodulator) എന്നതിന്റെ ചുരുക്കമാണ് മോഡം (Modem). കമ്പ്യൂട്ടറിൽ നിന്ന് സ്വീകരിക്കുന്ന ഡിജിറ്റൽ സിഗ്നലിനെ ടെലിഫോൺ ലൈനിലൂടെ കടന്നുപോകുവാനായി അനലോഗ് സിഗ്നലാക്കി മാറ്റുന്നു (Modulation). അതോടൊപ്പം ടെലിഫോൺ ലൈൻ വഴിവരുന്ന അനലോഗ് സിഗ്നലിനെ ഡിജിറ്റിലായി പരിവർത്തനം ചെയ്തത് കംപ്യൂട്ടറിലേക്കു നൽകുന്നു (Demodulation). ടെലിഫോൺ ലൈൻ വഴി വിവരങ്ങൾ അയയ്ക്കുകയും സ്വീകരിക്കുകയും ചെയ്യുന്നതിന്റെ വേഗതയെ അടിസ്ഥാനമാക്കിയാണ് മോഡത്തിന്റെ വേഗത നിർണ്ണയിക്കുന്നത്. മോഡത്തിന്റെ വേഗത അളക്കുന്നത് ബിറ്റ്സ്/സെക്കന്റ് (bits / second) ആണ്.

8.5.2 മൾട്ടിപ്ലെക്സർ/ഡി മൾട്ടിപ്ലെക്സർ (Multiplexer/Demultiplexer)

ഒറ്റ കേബിൾ ഉപയോഗിച്ച് 200 ഓ അതിലധികമോ ചാനലുകളെ കൈകാര്യം ചെയ്യുന്നത് നിങ്ങളെ എപ്പോഴെങ്കിലും അതിശയിപ്പിച്ചിട്ടുണ്ടോ? ഇതിനെയാണ് മൾട്ടിപ്ലെക്സിങ് എന്ന് പറയുന്നത്. ഇതേ രീതിയിലാണ് ശൃംഖലയിലുള്ള ഡാറ്റ കൈമാറ്റവും. ഭൗതിക മാധ്യമത്തിലൂടെ ഒന്നിലേറെ തരംഗങ്ങളെ സംയോജിപ്പിച്ച് സങ്കീർണതയേറിയ ഒറ്റ തരംഗമാക്കി മാറ്റി ഒരേ സമയം വിടുന്നതിനെ മൾട്ടിപ്ളെക്സിങ് എന്നും, മറുഭാഗത്ത് ഈ തരംഗത്തെ വിഘടിപ്പിച്ചു പ്രത്യേക തരംഗങ്ങളാക്കി മാറ്റുന്നതിനെ ഡി-മൾട്ടിപ്ലക്സിങ് എന്നും പറയുന്നു. ഭൗതിക മാധ്യമത്തെ മൾട്ടിപ്ളെക്സിങ് വിവിധ ഭാഗങ്ങളാക്കി മാറ്റുന്നു. ഇതിനെ ഫ്രീക്വൻസി ചാനൽ എന്ന് പറയുന്നു.

മൾട്ടിപ്ലെക്സർ വിവിധ ഉറവിടത്തിൽ നിന്നുള്ള തരംഗങ്ങളെ സംയോജിപ്പിച്ച്, മാധ്യമത്തിന്റെ വിവിധ ചാനലുകൾ വഴി അയയ്ക്കുന്നു. സംയോജിപ്പിച്ച തരംഗങ്ങൾ മാധ്യമത്തിലൂടെ ഒരേ സമയത്തു സഞ്ചരിക്കുന്നു. ലക്ഷ്യ സ്ഥാനത്തു ഇവയെ വിഭജിച്ച് വെവ്വേറെ തരംഗങ്ങളാക്കി, ഓരോ തരംഗത്തിനും എത്തേണ്ട സ്ഥലത്തേയ്ക്ക് അയയ്ക്കുന്നു. ചിത്രം 8.22 മൾട്ടിപ്ലെക്സറിന്റെയും ഡി-മൾട്ടിപ്ലെക്സറിന്റെയും പ്രവർത്തനം വിവരിക്കുന്നു.

ചിത്രം 8.22 : മൾട്ടിപ്ലക്സർ/ഡി- മൾട്ടിപ്ലക്സർ

നമുക്കു ചെയ്യാം

പത്തു നോഡുകൾ ഉള്ള ഒരു ചെറിയ കമ്പ്യൂട്ടർ ശൃംഖല നിർമിക്കുവാൻ ആവശ്യമായ ഉപകരണങ്ങളുടെയും മാധ്യമങ്ങളുടെയും പട്ടിക തയ്യാറാക്കുക .

സ്വയം വിലയിരുത്താം

1. ഹബും സ്വിച്ചും തമ്മിൽ താരതമ്യം ചെയ്യുക.
2. റിപീറ്ററിന്റെ ആവശ്യകത എന്ത്?
3. ഒരേപോലുള്ള രണ്ടു ശൃംഖലകളെ തമ്മിൽ ബന്ധിപ്പിക്കുന്ന ഉപകരണമാണ് __________.
4. റൂട്ടറും ബ്രിഡ്ജും തമ്മിലുള്ള വ്യത്യാസം എന്താണ്?
5. വ്യത്യസ്ത പ്രൊട്ടോക്കോൾ ഉള്ള രണ്ടു വ്യത്യസ്ത ശൃംഖലകളെ പരസ്പരം ബന്ധിപ്പിക്കുന്ന ഉപകരണമാണ് __________.
6. ടെലിഫോൺ ലൈനിലൂടെ രണ്ടു കമ്പ്യൂട്ടറുകൾ തമ്മിൽ വിവരവിനിമയം നടത്തുവാൻ ഉപയോഗിക്കുന്ന ഇലക്ട്രോണിക് ഉപകരണമാണ് __________.

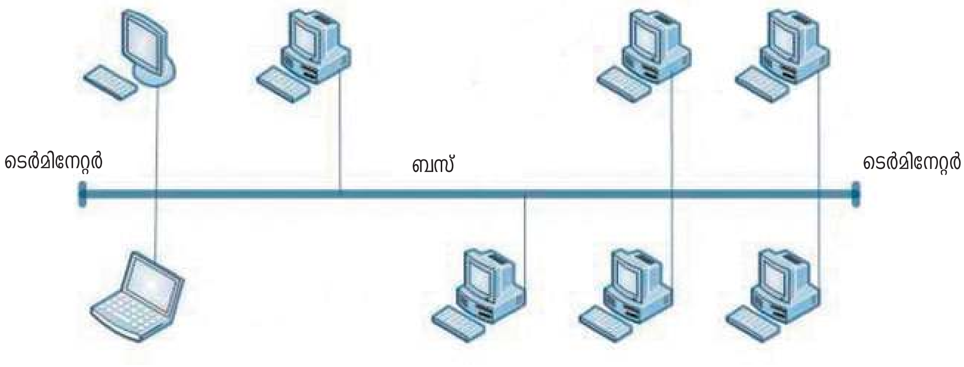
11.6 നെറ്റ്‌വർക്ക് ടോപ്പോളജികൾ (Network topologies)

പത്തു കമ്പ്യൂട്ടറുകൾ അടങ്ങിയ ഒരു ശൃംഖല രൂപകൽപ്പന ചെയ്യണമെന്ന് കരുതുക. ഏതൊക്കെ വിധത്തിൽ നമുക്ക് ഇവയെ പരസ്പരം ബന്ധിപ്പിക്കാം? ലഭ്യമായ മാദ്ധ്യമങ്ങളും ചില നിബന്ധനകളും വഴി നമുക്ക് ഇവയെ പല വിധത്തിൽ ബന്ധിപ്പിക്കാം ഭൗതികമായി കമ്പ്യൂട്ടറുകളെ പരസ്പരം ബന്ധിപ്പിച്ചു ശൃംഖല രൂപ കൽപ്പന ചെയ്യുന്ന രീതിയെ ടോപ്പോളജി എന്ന് പറയുന്നു. ബസ്, റിങ്, സ്റ്റാർ, മെഷ് എന്നിവയാണ് പ്രധാന ടോപ്പോളജികൾ.

8. 6.1 ബസ് ടോപ്പോളജി (Bus topology)

ബസ് ടോപ്പോളജിയിൽ (ചിത്രം 11.23) പ്രധാന കേബിൾ ആയ ബസിലേയ്ക്ക് നോഡുകളെ ബന്ധിപ്പിച്ചിരിക്കുന്നു. ഒരു നോഡിനു ഡാറ്റ അയയ്ക്കണമെങ്കിൽ, അത് ബസിലേയ്ക്ക് അയയ്ക്കുന്നു. ബസിന്റെ എല്ലാ ഭാഗത്തും ഈ ഡാറ്റ എത്തിച്ചേരുന്നു. എല്ലാ നോഡുകളും ബസിൽ വരുന്ന ഡാറ്റയെ പരിശോധിക്കുന്നു. ഏതു നോഡിലേ

ക്കാണോ ഡാറ്റ അയച്ചിരിക്കുന്നത് അത് ഡാറ്റയെ സ്വീകരിക്കുന്നു. ബസിന്റെ അഗ്രഭാഗങ്ങളിൽ ഒരു ചെറിയ ഉപകരണമായ ടെർമിനേറ്റർ ഘടിപ്പിച്ചിരിക്കുന്നു. തരംഗങ്ങൾ ബസിന്റെ അഗ്രഭാഗത്തു എത്തിയാൽ അവയെ ടെർമിനേറ്റർ ആഗിരണം ചെയ്തു നീക്കം ചെയ്യുന്നു. ഈ അവസരത്തിൽ ബസ് അടുത്ത തരംഗങ്ങളെ വഹിക്കുവാൻ പൂർണ സജ്ജമായിത്തീരുന്നു. കേബിളിലേക്കുള്ള തരംഗങ്ങളുടെ പ്രതിഫലനം ഒഴിവാക്കുവാനും, തരംഗങ്ങൾ തമ്മിൽ കൂടിച്ചേരുന്ന സാഹചര്യം ഒഴിവാക്കുവാനും ഇതിനാൽ സാധിക്കുന്നു. ഒരു നോഡിൽ നിന്ന് മറ്റെല്ലാ നോഡുകളിലേക്കും തരംഗങ്ങളെ അയയ്ക്കുന്നതിനെ ബ്രോഡ്കാസ്റ്റിംഗ് എന്ന് പറയുന്നു.



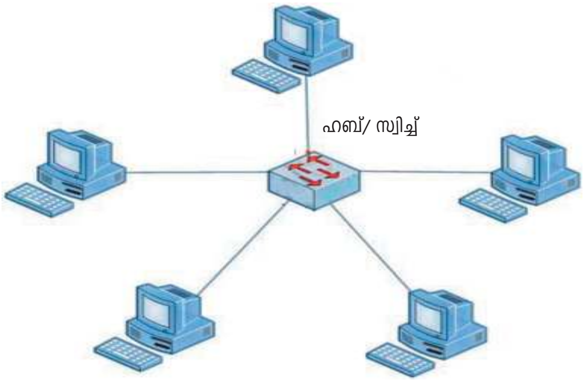
ചിത്രം 8.23: ബസ് ടോപ്പോളജി

ബസ് ടോപ്പോളജിയുടെ സവിശേഷതകൾ

- അനായാസമായി സ്ഥാപിക്കാം.
- ഇവ നിർമ്മിക്കുവാൻ വളരെ കുറച്ച് കേബിളുകൾ ഉപയോഗിക്കുന്നതിനാൽ ചെലവ് കുറവാണ്.
- ഒരു നോഡിന്റെ തകരാറ് ശൃംഖലയെ ബാധിക്കുന്നില്ല.
- ബസിന്റെയോ ടെർമിനേറ്ററിന്റെയോ തകരാറ് ശൃംഖലയെ മൊത്തമായി ബാധിക്കുന്നു.
- തകരാറ് കണ്ടെത്തുക എന്നത് ശ്രമകരമാണ്.
- ഒരു നോഡിനു മാത്രമേ ഒരു സമയത്തു ഡാറ്റ അയയ്ക്കുവാൻ കഴിയൂ.

8.6.2 സ്റ്റാർ ടോപ്പോളജി (Star topology)

ചിത്രം 8.24 ചിത്രീകരിച്ചിരിക്കുന്നതു പോലെ സ്റ്റാർ ടോപ്പോളജിയിൽ ഓരോ നോഡും ഹബ്ബിലേക്കോ അല്ലെങ്കിൽ സ്വിച്ചിലേക്കോ നേരിട്ട് ബന്ധിപ്പിച്ചിരിക്കുന്നു. ഇതിൽ ഏതെങ്കിലും ഒരു നോഡിനു ഡാറ്റ അയയ്ക്കണമെങ്കിൽ അത് സ്വിച്ചിലേക്കോ ഹബിലേക്കോ അയയ്ക്കുന്നു. ഹബിന്റെ കാര്യത്തിൽ ഈ തരംഗങ്ങളെ എല്ലാ നോഡുകളിലേക്കും സംപ്രേക്ഷണം ചെയ്യുകയും, ഉദ്ദേശിച്ച നോഡുകൾ മാത്രം അവയെ സ്വീകരിക്കുകയും ചെയ്യുന്നു. സ്വിച്ചിന്റെ കാര്യത്തിലാണെങ്കിൽ ഈ തരംഗങ്ങളെ ഉദ്ദേശിച്ച നോഡിലേക്ക് മാത്രം അയയ്ക്കുന്നു.



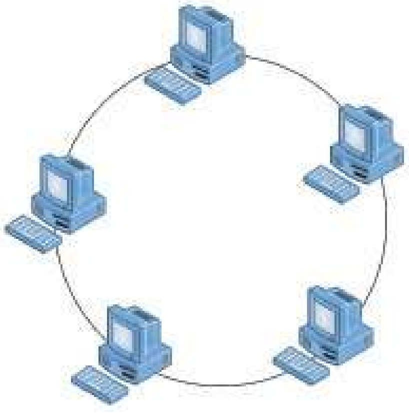
ചിത്രം 8.24 : സ്റ്റാർ ടോപ്പോളജി

സ്റ്റാർ ടോപ്പോളജിയുടെ സവിശേഷതകൾ

- ബസ് ടോപ്പോളജിയെ അപേക്ഷിച്ചു പ്രായോഗിക ക്ഷമത കൂടുതലാണ്.
- അനായാസമായി സ്ഥാപിക്കാം.
- തകരാറ് കണ്ടെത്തുക എളുപ്പമാണ്.
- കേന്ദ്രസ്ഥാനത്തുള്ള ഹബ്/സ്വിച്ച് ന്റെയും ബന്ധിപ്പിക്കുവാനുള്ള കഴിവ് അനുസരിച്ചു ശൃംഖലയിൽ നോഡുകളെ കൂട്ടിച്ചേർത്തു ശൃംഖല വിപുലീകരിക്കാം.
- ഹബ്ബിനോ/സ്വിച്ചിനോ തകരാറുണ്ടായാൽ ശൃംഖലയെ മൊത്തത്തിൽ ബാധിക്കുന്നു
- ബസ് ടോപ്പോളജിയെ അപേക്ഷിച്ച് ശൃംഖല നിർമ്മിക്കുവാൻ കൂടുതൽ കേബിൾ ആവശ്യമാണ്.

8.6.3 റിങ് ടോപ്പോളജി (Ring topology)

റിങ് ടോപ്പോളജിയിൽ നോഡുകളെ കേബിൾ ഉപയോഗിച്ച് വൃത്താകൃതിയിൽ ബന്ധിപ്പിച്ചിരിക്കുന്നു. തുടക്കമോ അവസാനമോ ഇല്ലാത്ത ഒരു വൃത്താകൃതിയാണ് റിങ് ടോപ്പോളജിക്കുള്ളത് (ചിത്രം 8.25). ടെർമിനേറ്ററിന്റെ ആവശ്യം റിങ് ടോപ്പോളജിക്ക് ഇല്ല. ഒരു ദിശയിലേക്കു മാത്രമാണ് ഡാറ്റ സഞ്ചരിക്കുന്നത്. ഒരു നോഡിൽ നിന്ന് മറ്റൊരു നോഡിൽ എത്തുന്ന തരംഗങ്ങളെ പുനരുജ്ജീവിപ്പിച്ച് അടുത്തതിലേക്ക് അയയ്ക്കുന്നു. ഉദ്ദേശിച്ച നോഡിൽ എത്തുന്നതുവരെ ഈ പ്രക്രിയ തുടരുന്നു. എല്ലാ നോഡുകളിലൂടെയും


ചിത്രം 8.25 : റിങ് ടോപ്പോളജി

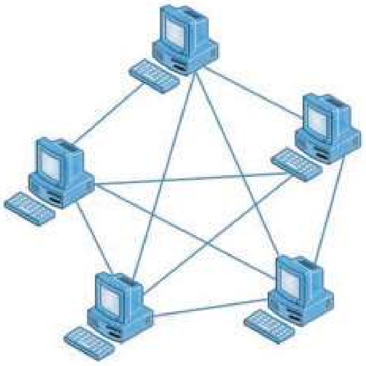
സഞ്ചരിക്കുന്ന തരംഗങ്ങൾ അവസാനം സംപ്രേക്ഷണം ചെയ്ത നോഡിൽ തിരിച്ചെത്തുകയും, അവിടെ നിന്നു ഇവയെ നീക്കം ചെയ്യുകയും ചെയ്യുന്നു.

റിങ് ടോപ്പോളജിയുടെ സവിശേഷതകൾ

- ഓരോ നോഡും തരംഗങ്ങളുടെ ശക്തി വർധിപ്പിക്കുന്നതിനാൽ, തരംഗങ്ങളുടെ ശക്തി വർധിപ്പിക്കേണ്ടി വരുന്നില്ല.
- വളരെ കുറച്ച് മാത്രം കേബിൾ ഉപയോഗിക്കുന്നതിനാൽ ചെലവ് കുറവാണ്.
- ഒരു നോഡ് തകരാറിലായാൽ അത് ശൃംഖലയെ മുഴുവനായി ബാധിക്കുന്നു.
- ശൃംഖലയിലേക്ക് പുതിയ നോഡുകളെ കൂട്ടിച്ചേർക്കുക പ്രയാസകരമാണ്

8.6.4 മെഷ് ടോപ്പോളജി (Mesh topology)

മെഷ് ടോപ്പോളജിയിൽ എല്ലാ നോഡുകളെയും പരസ്പരം ബന്ധിപ്പിച്ചിരിക്കുന്നു. ചിത്രം 8.26 കാണിച്ചിരിക്കുന്നത് പോലെ രണ്ടു നോഡുകൾക്കിടയിൽ ഒന്നിലേറെ പാതകൾ ഉണ്ടായിരിക്കും. ഒരു പാതയിൽ തടസ്സമുണ്ടായാലും മറ്റൊരു പാതയിലൂടെ ഡാറ്റ ലക്ഷ്യ സ്ഥാനത്തു എത്തിച്ചേരുന്നു.



ചിത്രം 8.26 : മെഷ് ടോപ്പോളജി

മെഷ് ടോപ്പോളജിയുടെ സവിശേഷതകൾ

- രണ്ടു നോഡുകൾക്കിടയിൽ ഉള്ള പാത തകരാറായാലും ശൃംഖലയ്ക്കു തകരാറ് ഉണ്ടാകുന്നില്ല.
- കൂടുതൽ കേബിൾ വേണ്ടതിനാൽ ചെലവ് കൂടുതലാണ്.
- വളരെ സങ്കീർണവും കൈകാര്യം ചെയ്യുവാൻ പ്രയാസവുമാണ്.

നമുക്കു ചെയ്യാം

നിങ്ങളുടെ സ്കൂൾ ലാബിലെ ശൃംഖലയുടെ ക്രമീകരണരീതി എന്താണ് എന്ന് മനസിലാക്കുക.

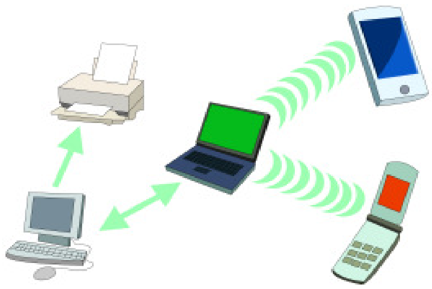
8.7 വിവിധതരം ശൃംഖലകൾ (Type of networks)

ഒരു കമ്പ്യൂട്ടർ ശൃംഖല ഭൗമ വിസ്തൃതിയിൽ പല അളവുകളിലായി വ്യാപിച്ചു കിടക്കുന്നു. ഇത് വേണമെങ്കിൽ ഒരു മേശയുടെ മുകളിലോ ഒരു റൂമിലോ ഒരു കെട്ടിടത്തിലോ ഒരു നഗരത്തിലോ, ഒരു രാജ്യത്തിനുള്ളിലോ ഭൂഖണ്ഡങ്ങളിലോ ലോകം മുഴുവനുമോ വ്യാപിച്ചു കിടക്കാം. കമ്പ്യൂട്ടർ ശൃംഖലയെ അവയുടെ വ്യാപനത്തെ അടിസ്ഥാനമാക്കി ചുവടെ ചേർത്ത രീതിയിൽ വേർതിരിക്കാം.

- PAN - പേർസണൽ ഏരിയ നെറ്റ്വർക്ക്
- LAN - ലോക്കൽ ഏരിയ നെറ്റ്വർക്ക്
- MAN - മെട്രോപൊളിറ്റൻ ഏരിയ നെറ്റ്വർക്ക്
- WAN - വൈഡ് ഏരിയ നെറ്റ്വർക്ക്

8.7.1 പേർസണൽ ഏരിയ നെറ്റ്വർക്ക് (Personal Area Network)

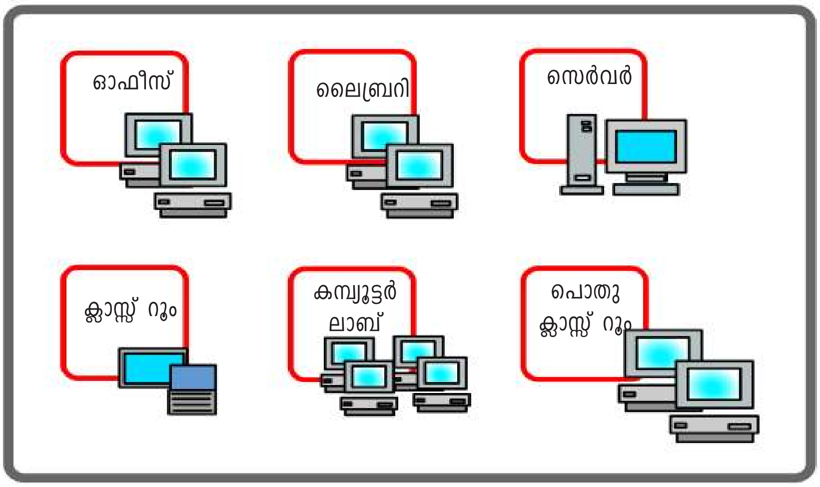
ഒരു വ്യക്തിയുടെ പരിധിയിലുള്ള വിനിമയ ഉപകരണങ്ങളുടെ (കമ്പ്യൂട്ടർ, മൊബൈൽ, ടാബ്ലെറ്റ്, പ്രിൻറർ എന്നിവ) ശൃംഖലയാണ് PAN. ഏതാനും മീറ്റർ വൃത്ത പരിധിക്കുള്ളിൽ ഇവ വ്യാപിച്ചു കിടക്കുന്നു. ചിത്രം 8.27 ഒരു പാട്ട് ഒരു മൊബൈലിൽ നിന്ന് മറ്റൊന്നിലേക്കോ, ഒരു കമ്പ്യൂട്ടറിൽ നിന്ന് MP3 പ്ലെയറിലേക്കോ അയയ്ക്കുവാൻ നാം PAN ശൃംഖല ഉണ്ടാക്കാറുണ്ട്. PAN ശൃംഖല ഉണ്ടാക്കുവാൻ ഗൈഡഡ് മാധ്യമവും (USB), അൺ ഗൈഡഡ് മാധ്യമവും (ബ്ലൂടൂത്ത്, ഇൻഫ്രാറെഡ്) ഉപയോഗിക്കാം.



ചിത്രം 8.27: പാൻ

8.7.2 ലോക്കൽ ഏരിയ നെറ്റ്വർക്ക് (Local Area Network)

ഒരു LAN ശൃംഖലയിലെ വിവര വിനിമയത്തിനും കമ്പ്യൂട്ടിങ്ങിനുമുള്ള ഉപകരണങ്ങൾ ഒരു മുറിയ്ക്കുള്ളിലോ, ഒരു കെട്ടിടത്തിനുള്ളിലോ ഒരു സ്ഥാപന പരിധിയ്ക്ക് ഉള്ളിലോ ആയിരിക്കും പരസ്പരം ബന്ധിപ്പിച്ചിരിക്കുന്നത്. ഏതാനും മീറ്ററോ ഏതാനും കിലോ മീറ്ററോ വൃത്ത പരിധിയ്ക്കുള്ളിൽ ആയിരിക്കും ഇവയുടെ പ്രവർത്തനം. സാധാരണയായി ഓഫീസിലും സ്കൂളിലും റൂമിലും ഒരു LAN ശൃംഖലമാത്രമാണ് ഉണ്ടാകാറുള്ളത്, എന്നാൽ



ചിത്രം 8.28 : ലോക്കൽ ഏരിയ നെറ്റ്വർക്ക്

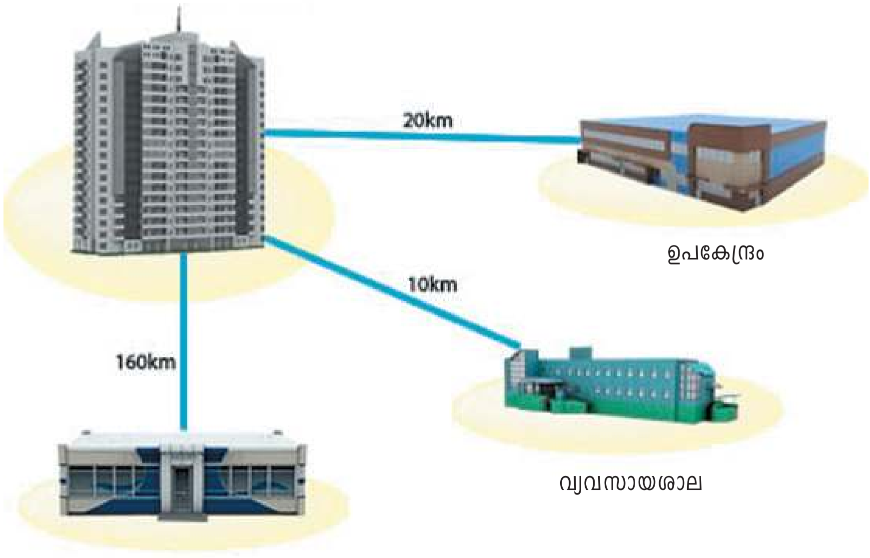
ഒരു കെട്ടിടത്തിൽ തന്നെ ഒന്നിൽ കൂടുതൽ LAN ചിലപ്പോൾ ഉണ്ടായെന്നു വരാം. (ചില സ്കൂളുകളിൽ ഓരോ ലാബിലും ഓരോ LAN ശൃംഖല ഉള്ളതുപോലെ). ചിലപ്പോൾ LAN അടുത്തടുത്ത കെട്ടിടത്തിലേക്കും വ്യാപിച്ചിരിക്കും

LAN ശൃംഖലയുടെ നിയന്ത്രണവും പരിപാലനവും, ഒരു വ്യക്തിയുടെയോ, ഒരു സ്ഥാപനത്തിന്റെയോ ഉടമസ്ഥതയിലായിരിക്കും.

ഗൈഡഡ് മാധ്യമം (വയേർഡ് മീഡിയ) (UTP കേബിളുകൾ കോയാക്സിൽ കേബിളുകൾ തുടങ്ങിയവ) ഉപയോഗിച്ചും വയർലെസ്സ് മാധ്യമം (ഇൻഫ്രാറെഡ്, റേഡിയോ തരംഗങ്ങൾ തുടങ്ങിയവ) ഉപയോഗിച്ചും ലാൻ സ്ഥാപിക്കാവുന്നതാണ്. അൺ ഗൈഡഡ് മാധ്യമം (Unguided Media) ഉപയോഗിച്ചാണ് LAN സ്ഥാപിക്കുന്നതെങ്കിൽ അതിനെ വയർലെസ്സ് LAN (WLAN (Wireless LAN)) എന്ന് വിളിക്കാം.

8.7.3 മെട്രോപൊളിറ്റൻ ഏരിയ നെറ്റ്‌വർക്ക് (Metropolitan Area Network (MAN))

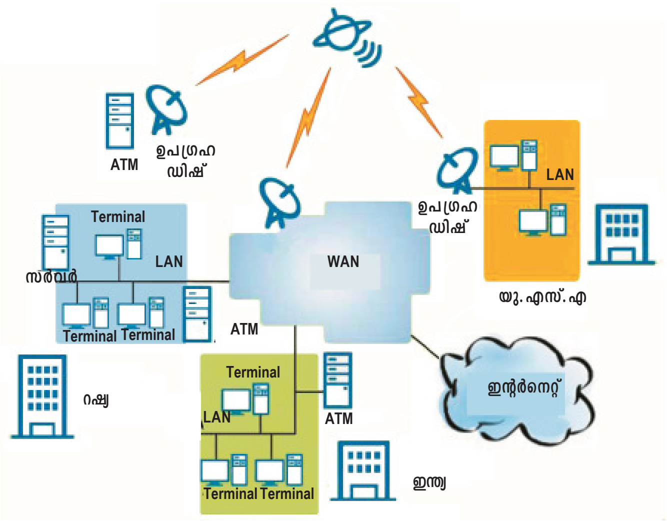
MAN ശൃംഖലയുടെ കമ്പ്യൂട്ടിങ്ങും പ്രവർത്തനവും വിനിമയ ഉപകരണങ്ങളുടെ വ്യാപനവും ഒരു നഗര പരിധിക്കുള്ളിൽ നിൽക്കുന്നു. ഇതിന്റെ വൃത്ത പരിധി നൂറു കിലോമീറ്റർ വരെ വ്യാപിച്ചു കിടക്കും. ലാൻ (LAN) ശൃംഖലകളെയും, സ്വകാര്യ കമ്പ്യൂട്ടറുകളെയും പരസ്പരം ബന്ധിപ്പിച്ചാണ് MAN സ്ഥാപിക്കുന്നത്. എല്ലാവിധ മാധ്യമങ്ങളും (ഗൈഡഡും അൺ-ഗൈഡഡും) ഇതിനായി ഉപയോഗിക്കുന്നു. MAN ന്റെ ഉടമസ്ഥതയും നിയന്ത്രണവും ഗവൺമെന്റിനോ, ഒരു വലിയ സ്ഥാപനത്തിനോ ആയിരിക്കും (ചിത്രം 8.29)



ചിത്രം 8.29 : മെട്രോപൊളിറ്റൻ ഏരിയ നെറ്റ് വർക്ക്

8.7.4 വൈഡ് ഏരിയ നെറ്റ്‌വർക്ക് (Wide Area Network (WAN))

പല നഗരങ്ങളിലും രാജ്യങ്ങളിലും ഭൂഖണ്ഡങ്ങളിലുമായി വ്യാപിച്ചു കിടക്കുന്ന വിവര വിനിമയ കമ്പ്യൂട്ടിങ് ഉപകരണങ്ങൾ WAN ശൃംഖലയിൽ ഉൾപ്പെടുന്നു. നൂറു കിലോമീറ്റർ വൃത്തപരിധിയ്ക്കും അപ്പുറത്തേയ്ക്ക് ഇവയുടെ പ്രവർത്തനം വ്യാപിച്ചിരിക്കുന്നു. സ്വകാര്യ കമ്പ്യൂട്ടറുകൾ, LAN, MAN കൂടാതെ മറ്റു WANകളും ഇതിൽ അംഗങ്ങൾ ആയിരിക്കും. എല്ലാ തരത്തിലും ഉള്ള വിനിമയ മാധ്യമങ്ങൾ (ഗൈഡഡും അൺ ഗൈഡഡും) ഇവിടെ ഉപയോഗിക്കുന്നു ചിത്രം 8.30. WANന് ഉത്തമ ഉദാഹരണമാണ് ഇന്റർനെറ്റ്. ലോകത്തിലെ ഏറ്റവും വലിയ WAN ആയിട്ടാണ് ഇന്റർനെറ്റിനെ കണക്കാക്കുന്നത്. രാജ്യത്തിനുള്ളിലും, വിവിധ ഭൂഖണ്ഡങ്ങളിലുമായി വ്യാപിച്ചു കിടക്കുന്ന ATM ശൃംഖല, ബാങ്ക് ശൃംഖല, ഗവൺമെന്റിന്റെയും, അന്താരാഷ്ട്ര സ്ഥാപനങ്ങളുടെയും ശൃംഖലകൾ എന്നിവ WANനു ഉദാഹരണങ്ങളാണ്.



ചിത്രം 11.30: വൈഡ് ഏരിയ നെറ്റ്‌വർക്ക്

അളവുകോൽ	PAN	LAN	MAN	WAN
വ്യാപ്തി	ചെറിയ വിസ്തീർണത്തിൽ (10m വൃത്ത പരിധി)	ഏതാനും മീറ്റർ മുതൽ കിലോമീറ്റർ വരെ (10 km വൃത്ത പരിധി)	നഗര പരിധിയിൽ (100 km വൃത്ത പരിധി)	രാജ്യങ്ങളിലും ഭൂഖണ്ഡങ്ങളിലും ലോകമാകമാനവും
വിനിമയ വേഗത	അതിവേഗം	അതിവേഗം	സാമാന്യ വേഗത	വേഗത കുറവ്
സ്ഥാപിക്കുവാനുള്ള ചിലവ്	തീരെ കുറവ്	ചിലവ് കുറവ്	സാമാന്യം ചിലവ്	ചിലവേറിയ

പട്ടിക 8.1 PAN, LAN, MAN, WAN സവിശേഷതകളുടെ സംഗ്രഹം

11.8 ശൃംഖലയുടെ യുക്ത്യാധിഷ്ഠിത തരംതിരിവ് (Logical classification of networks)

ശൃംഖലയിലെ കമ്പ്യൂട്ടറുകളുടെ ചുമതലകളെ അടിസ്ഥാനമാക്കി രണ്ടായി തരംതിരിക്കാം. പീർ ടു പീർ (Peer - to - peer), ക്ലയന്റ് സെർവർ (Client - Server).

8.8.1 പീർ ടു പീർ (Peer to peer)

പിർ ടു പിർ ശൃംഖലയിൽ ഒരു കമ്പ്യൂട്ടറിനും ശൃംഖലയുടെ മുഴുവൻ ചുമതല ഉണ്ടായിരിക്കില്ല. ഇവിടെ വിവരങ്ങൾ കൈമാറുന്നതിനും ഉപകരണങ്ങൾ പങ്കിടുന്നതിനും കമ്പ്യൂട്ടറുകളെ തമ്മിൽ പരസ്പരം ബന്ധിപ്പിക്കുകയാണ് ചെയ്യുന്നത്. എല്ലാ കമ്പ്യൂട്ടറുകൾക്കും തുല്യ പരിഗണനയാണ് ഉള്ളത്. ഏതു കമ്പ്യൂട്ടറിനും ഏതു സമയത്തും ക്ലയന്റ് ആയിട്ടും സെർവർ ആയിട്ടും പ്രവർത്തിക്കാം.

ചെറിയ കമ്പ്യൂട്ടർ ശൃംഖലകൾ ആവശ്യമുള്ളതും, എന്നാൽ പൂർണ്ണ ചുമതല ഉള്ള സെർവറുകളുടെ ആവശ്യമില്ലാത്തതുമായ സ്ഥലങ്ങളിൽ (വീടുകൾ, ചെറിയ വ്യാപാര സ്ഥാപനങ്ങൾ) ഇവ അനുയോജ്യമാണ്.

8.8.2 ക്ലയന്റ് സെർവർ (Client-Server)

ഭൂരിഭാഗം ശൃംഖലകളും ക്ലയന്റ്-സെർവർ രീതിയിൽ അധിഷ്ഠിതമാണ്. ഒരു ഭക്ഷണശാലയിൽ ചെന്ന്, ആഹാര സാധനങ്ങളുടെ പട്ടിക നോക്കി, അതിൽ നിന്ന് ഇഷ്ടമുള്ളത് കടയിലെ ജോലിക്കാരനോട് (സെർവർ) ആവശ്യപ്പെടുന്നതിന് തുല്യമാണ് ഇവയുടെ പ്രവർത്തനം. ഭക്ഷണശാലയിൽ അത് ലഭ്യമാണെങ്കിൽ ആവശ്യക്കാരന് (ക്ലൈന്റ്) അത് വിതരണം ചെയ്യുകയും, ലഭ്യമല്ലെങ്കിൽ ആവശ്യം നിരാകരിക്കപ്പെടുകയും ചെയ്യുന്നു.

ചിത്രം. 8.31 : ക്ലയന്റ് - സർവർ

ക്ലയന്റ്-സെർവറിന്റെ ഘടനയിൽ ശൃംഖലയിലെ ശക്തി കൂടിയ കമ്പ്യൂട്ടർ (സെർവർ), ശക്തി കുറഞ്ഞ കമ്പ്യൂട്ടറിനു (ക്ലയന്റ്) സേവനങ്ങൾ ലഭ്യമാക്കുന്നു. ക്ലയന്റിന്റെ അഭ്യർത്ഥന അനുസരിച്ചു ഒരു സെർവർ നിർദിഷ്ട സേവനങ്ങൾ (Response) ലഭ്യമാക്കുന്നു. ഈ സേവനങ്ങളിൽ ഡാറ്റയുടെയും സോഫ്റ്റ്‌വെയറിന്റെയും ഹാർഡ്‌വെയറിന്റെയും പങ്കിടൽ ഉൾപ്പെടുന്നു. ചിത്രം 8.31 ക്ലയന്റ് സെർവറിന്റെ ഘടന ചിത്രീകരിച്ചിരിക്കുന്നു.

ക്ലയന്റ് സെർവറിന്റെ ഘടന കേന്ദ്രീകൃത സോഫ്റ്റ്‌വെയർ മാനേജ്മെന്റിന് ഉദാഹരണമാണ്. സെർവറിൽ സോഫ്റ്റ്‌വയർ ലോഡ് ചെയ്യുമ്പോൾ അവ ക്ലയന്റുകൾക്കിടയിൽ പങ്കുവെയ്ക്കപ്പെടുകയും, സെർവർ സോഫ്റ്റ്‌വയറിൽ ഉണ്ടാകുന്ന ഏതു മാറ്റവും ക്ലയന്റിൽ പ്രതിഫലിക്കുകയും ചെയ്യുന്നു. ഓരോ കമ്പ്യൂട്ടറിലും പുതിയ ഫയലും അതിന്റെ പരിവർത്തന ഫയലും ഇടുവാനുള്ള അധിക ഊർജവും സമയവും ഇതിനാൽ ലാഭിക്കാം.

സെർവറുകളുടെ തരംതിരിക്കൽ

a) **ഫയൽ സെർവർ:** ഒന്നിലധികം ഉപഭോക്താക്കളുടെ ഫയലുകൾ സൂക്ഷിക്കാനും കൈകാര്യം ചെയ്യുവാനും ഉള്ള കമ്പ്യൂട്ടർ ആണിത്.

b) **വെബ് സെർവർ :** വെബ് പേജിനുള്ള അഭ്യർഥന കൈകാര്യം ചെയ്യുന്ന കമ്പ്യൂട്ടറാണിത്.

c) **പ്രിന്റ് സെർവർ :** ക്ലയന്റുകളിൽ നിന്നും പ്രിന്ററുകളിലേക്കുള്ള പ്രിന്റിങ് ജോലികളെ മുൻഗണനയ്ക്ക് അനുസരിച്ചു പൂർത്തീകരിക്കുന്ന കമ്പ്യൂട്ടർ ആണിത്.

d) **ഡാറ്റാബേസ് സെർവർ:** പൊതുവായി സൂക്ഷിച്ചിരിക്കുന്ന ഡാറ്റായെ കാണാനും മാറ്റങ്ങൾ വരുത്താനും നീക്കം ചെയ്യുവാനും അംഗീകൃത ഉപഭോക്താവിനെ (ക്ലയന്റ്) സഹായിക്കുന്ന കമ്പ്യൂട്ടർ ആണിത്.

സ്വയം വിലയിരുത്താം

1. ബസ് ടോപ്പോളജിയിൽ ബസിന്റെ അഗ്രഭാഗത്തു എത്തുന്ന തരംഗങ്ങളെ ________ ആഗിരണം ചെയ്യുകയും ബസിൽ നിന്ന് നീക്കം ചെയ്യുകയും ചെയ്യുന്നു.
2. ________ ടോപ്പോളജിയിൽ ഓരോ നോഡും ഹബ് / സ്വിച്ച് ലേക്ക് നേരിട്ട് ബന്ധിപ്പിച്ചിരിക്കുന്നു
3. ________ടോപ്പോളജിയിൽ ഓരോ നോഡും മറ്റു നോഡുകളുമായി നേരിട്ടു ബന്ധിപ്പിച്ചിരിക്കുന്നു.
4. താഴെപ്പറയുന്ന വിവിധ ശൃംഖലകളെ തരം തിരിക്കുക.

   ATM ന്റെ ശൃംഖല, കേബിൾ ടെലിവിഷൻ ശൃംഖല, ഒരു സ്കൂളിനുള്ളിലെ ശൃംഖല,

   ബ്ലൂടൂത്ത് ഉപയോഗിച്ചുള്ള വീടിനുള്ളിലെ ശൃംഖല, ടെലിഫോൺ ശൃംഖല, റെയിൽവേ ശൃംഖല

5 എന്താണ് PAN?

6 എന്താണ് പീർ ടു പീർ ശൃംഖല ?

8.9 ശൃംഖലയിൽ കമ്പ്യൂട്ടറുകളുടെ തിരിച്ചറിയൽ (Identification of computers over a network)

അമേരിക്കയിൽ ഉള്ള ഒരു കൂട്ടുകാരന് ഇന്ത്യയിൽ ഉള്ള നിങ്ങൾ ഒരു കത്ത് എഴുതുന്നു എന്ന് സങ്കൽപ്പിക്കുക. നിങ്ങൾ ഒരു കത്ത് എഴുതി, കവറിൽ ഇട്ടു, കവറിനു പുറത്ത് കൂട്ടുകാരന്റെ മേൽവിലാസവും എഴുതി, പുറകിൽ നിങ്ങളുടെയും മേൽവിലാസവും എഴുതുന്നു. ഈ കത്ത് ഇന്ത്യയിലെ പോസ്റ്റോഫീസിൽ ഇടുമ്പോൾ അതിനു മുകളിൽ ഇന്ത്യൻ തപാൽ വകുപ്പിന്റെ സീലും തീയതിയും അതിൽ പതിപ്പിക്കുന്നു. വിവിധ മാർഗങ്ങളിലൂടെ സഞ്ചരിച്ച കത്ത് അമേരിക്കയിലെ തപാൽ വകുപ്പിൽ എത്തുന്നു. അവിടെ വെച്ച് അമേരിക്കൻ തപാൽ വകുപ്പിന്റെ സീലും തീയതിയും പതിക്കുന്നു. അവസാനം പോസ്റ്റുമാൻ കത്ത് മേൽ വിലാസക്കാരന് കൈമാറുന്നു. കമ്പ്യൂട്ടർ ശൃംഖലയിലും ഡാറ്റയെ പാക്കറ്റുകളാക്കി ഇതേ രീതിയിൽ ആണ് കൈമാറ്റം ചെയ്യുന്നത്.

ഒരു ശൃംഖല സജ്ജീകരിച്ചു കഴിഞ്ഞാൽ, നോഡുകൾ തമ്മിൽ പരസ്പരം വിവര വിനിമയം നടത്താം. ശരിയായ വിവരവിനിമയത്തിന് നോഡുകളെ അന്യോന്യം തിരിച്ചറിയേണ്ടത്

ആവശ്യമാണ്. X എന്ന നോഡ് Y എന്ന നോഡിലേക്കു വിവരങ്ങൾ കൈമാറണമെങ്കിൽ, X ഉം Y ഉം ശൃംഖലയിൽ അന്യോന്യം തനതായി തിരിച്ചറിയത്തക്ക ആയിരിക്കണം. ഇത് എങ്ങനെ സാധിക്കുന്നു എന്ന് പരിശോധിക്കാം.

8.9.1 മീഡിയ അക്സസ് കൺട്രോൾ വിലാസം (Media Access Control (MAC) address)

ഓരോ NIC (Network Interface Card) യിലും അത് നിർമിച്ച കമ്പനിക്കാർ നൽകുന്ന വ്യത്യസ്തവും സ്ഥിരമായതും ആഗോളപരമായി അംഗീകരിച്ചിട്ടുള്ളതുമായ (പന്ത്രണ്ടക്ക ഹെക്സാ ഡെസിമൽ നമ്പറുകൾ) മേൽവിലാസമാണ് MAC അഡ്രസ്. ഒരു NIC ഉള്ള മെഷീനെ അതിന്റെ MAC വിലാസം ഉപയോഗിച്ച് തിരിച്ചറിയുന്നു. NIC യിലെ MAC വിലാസം സ്ഥിരമായിരിക്കും.

MAC വിലാസം എന്നത് 12 അക്ക ഹെക്സ ഡെസിമൽ അല്ലെങ്കിൽ 48 ബിറ്റ് ബൈന റിയാണ്. താഴെ കാണിച്ചിരിക്കുന്ന ഏതെങ്കിലും ഒരു രീതിയിൽ ആണ് MAC വിലാസം എഴുതാറുള്ളത്

MM:MM:MM:SS:SS:SS അല്ലെങ്കിൽ MM-MM-MM-SS-SS-SS

MAC വിലാസത്തിന്റെ ആദ്യഭാഗം (MM:MM:MM) അത് നിർമിച്ച കമ്പനിയുടെ തിരിച്ച റിയൽ അക്കവും രണ്ടാമത്തെ പകുതി (SS:SS:SS) NIC യ്ക്ക് ആയി കമ്പനി നൽകി യിരിക്കുന്ന ക്രമ നമ്പറുമാണ്. MAC വിലാസത്തിനു ഉദാഹരണമാണ്. താഴെ കൊടുത്തി രിക്കുന്നത്.

00:AO:C9	:	14:C8:35

ചിത്രം. 8.32 : MAC Id

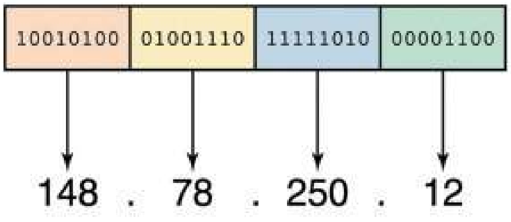
ആദ്യ പകുതി 00:AO:C9 എന്നത് ഇത് നിർമിച്ചത് ഇന്റൽ കോർപറേഷൻ ആണ് എന്ന് സൂചിപ്പിക്കുന്നു. അവസാന മൂന്നക്ക നമ്പർ ഇന്റൽ കോർപറേഷൻ NIC യ്ക്ക് നൽകിയ ക്രമനമ്പറാണ്.

8.9.2 ഇന്റർനെറ്റ് പ്രോട്ടോകോൾ (Internet Protocol (IP))

ശൃംഖലയിലെ ഓരോ നോഡിനും നൽകിയിട്ടുള്ള 4 ഭാഗങ്ങൾ ഉള്ള തനതായ നമ്പ റാണ് IP മേൽവിലാസം അഥവാ IP അഡ്രസ്. ശൃംഖല മേധാവി (നെറ്റ്‌വർക്ക് അഡ്മിനി സ്ട്രേറ്റർ) അല്ലെങ്കിൽ ഇന്റർനെറ്റ് സർവീസ് പ്രൊവൈഡർ ആണ് ഓരോ നോഡിനു മുള്ള IP അഡ്രസ് രേഖപ്പെടുത്തുന്നത്. 4 ഭാഗങ്ങളാണ് ഇതിനുള്ളത്. ഓരോ ഭാഗത്തെയും ഡോട്ട് ഉപയോഗിച്ച് വേർതിരിക്കും. ഓരോ ഭാഗത്തും 0 മുതൽ 255 വരെ ഉള്ള ഒരു നമ്പ റാണ് ഉണ്ടാകുക. ഒരു IP അഡ്രസ് 4 ബൈറ്റ് (32 ബിറ്റുകൾ) നമ്പർ ഉപയോഗിച്ചാണ് തയാറാക്കുന്നത്.

ഓർത്തിരിക്കുവാൻ എളുപ്പത്തിനായി IP അഡ്രസിനെ ഡെസിമൽ രൂപത്തിൽ ഡോട്ട് ഉപ യോഗിച്ച് വേർതിരിച്ച നമ്പറായി, രൂപകൽപ്പന ചെയ്തിരിക്കുന്ന (ചിത്രം 8.32) ൽ നൽകി യിരിക്കുന്നു.

ഒരു ശൃംഖലയിൽ ഒരു ഉപകരണത്തിന്റെ IP മേൽവിലാസം, അതിനെ തിരിച്ചറിയുവാനായി ഉപയോഗിക്കുന്നു. ഉപകരണത്തിന്റെ IP മേൽവിലാസം ഉപയോഗിച്ച് IP പ്രോട്ടോക്കോൾ പാക്കറ്റുകളെ വഴിതിരിച്ചു വിടുന്നു.



ചിത്രം. 8.33 : IP മേൽവിലാസം

IP മേൽവിലാസത്തിനു രണ്ടു പതിപ്പുകൾ ആണ് ഉള്ളത്. പതിപ്പ് 4 (version 4) IPv4 പതിപ്പ് 6 (Version 6) IPv6. IPv4 പ്രകാരം 32 ബിറ്റ് വലുപ്പമുള്ള മേൽവിലാസം ആണ് കംപ്യൂട്ടറിനു നൽകുന്നത്, IPv6 പ്രകാരം 128 ബിറ്റ് വലുപ്പമുള്ള മേൽവിലാസം ആണ് കംപ്യൂട്ടറിനു നൽകുന്നത്. IPv4 ഉപയോഗിച്ച് 2^{32}(ഏകദേശം 4 ലക്ഷം കോടി) വ്യത്യസ്ത ഉപകരണങ്ങളെ പ്രതിനിധീകരിക്കുവാൻ കഴിയും.

ശൃംഖലയിലേക്കു ബന്ധിപ്പിക്കേണ്ട ഉപകരണങ്ങളുടെ (മൊബൈൽ ഫോൺ, വീട്ടുപകരണങ്ങൾ, വ്യക്തിഗത വിനിമയോപാധികൾ) എണ്ണം നാൾക്കുനാൾ അതിവേഗം വർധിച്ചു വരുന്നതിനാൽ IPv4 വിഭാഗത്തിലുള്ള വിലാസങ്ങൾ ഉപയോഗിച്ച് തീരുന്നു. ഈ പ്രതിസന്ധി മറികടക്കുന്നതിനായാണ് IPv6 വികസിപ്പിച്ച് എടുത്തത്. അത് ഇപ്പോൾ ഉപയോഗിച്ച് തുടങ്ങിയിരിക്കുന്നു.

IPv6 ഉപയോഗിച്ച് 2^{128}(ഏകദേശം 4 ലക്ഷം കോടി × 4 ലക്ഷം കോടി × 4 ലക്ഷം കോടി ×4 ലക്ഷം കോടി) വിവിധതരം ഉപകരണങ്ങളെ പ്രതിനിധീകരിക്കാം.

നമുക്കു ചെയ്യാം

നിങ്ങളുടെ സ്കൂളിലെ കമ്പ്യൂട്ടർ ശൃംഖലയിലെ ഓരോ ഉപകരണങ്ങളുടെയും MAC ID യും IP അഡ്രസും കണ്ടുപിടിച്ചു ഒരു പട്ടിക താഴെ കാണിച്ചിരിക്കുന്നത് പോലെ തയാറാക്കുക (IPCONFIG/ALL എന്ന നിർദേശം, കമാൻഡ് പ്രോംറ്റിൽ ഉപയോഗിക്കുക)

ക്രമ നം.	കമ്പ്യൂട്ടറിന്റെ പേര്	IP	MAC
1.			
2.			
3.			

11.10 ശൃംഖലകളിലെ പ്രോട്ടോക്കോളുകൾ (Network Protocols)

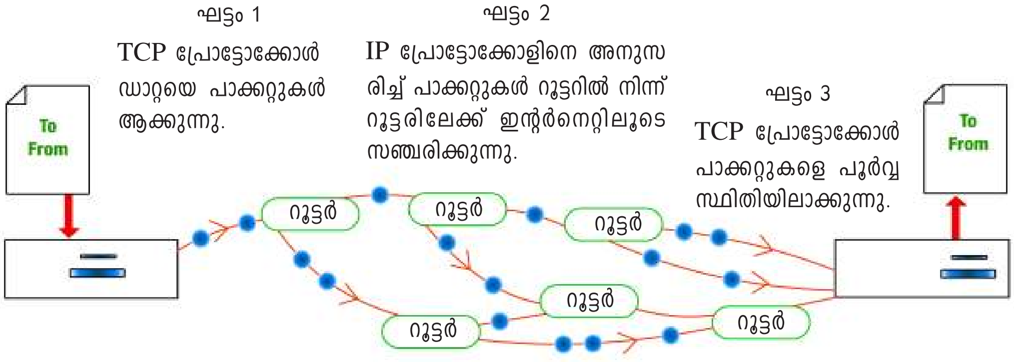
ശൃംഖലയിലെ ഉപകരണങ്ങൾ തമ്മിൽ വിവരങ്ങൾ പരസ്പരം കൈമാറുമ്പോൾ സ്വീകരിക്കേണ്ട പ്രത്യേക നിയമങ്ങളാണ് പ്രോട്ടോക്കോളുകൾ. ഡാറ്റ ഫോർമാറ്റിങ്, ഡാറ്റ കംപ്രസ്സിങ്, പിശകുകളുടെ പരിശോധന, തിരിച്ചറിയൽ, പരസ്പരം ബന്ധിപ്പിക്കൽ, ഡാറ്റാ ലക്ഷ്യസ്ഥാനത്തു എത്തിച്ചേർന്നു എന്ന് ഉറപ്പുവരുത്തൽ എന്നിവയ്ക്കായി ഓരോ പ്രോട്ടോക്കോളിനും അതിന്റേതായ നിയമങ്ങളുണ്ട്.

പ്രത്യേക ഉദ്ദേശ്യങ്ങൾക്കു വേണ്ടിയും, സാഹചര്യങ്ങൾക്കു വേണ്ടിയും നിരവധി കമ്പ്യൂട്ടർ ശൃംഖല പ്രോട്ടോക്കോളുകൾ നിർമ്മിച്ചിട്ടുണ്ട്. TCP/IP, SPx/IPx തുടങ്ങിയവയാണ് പൊതുവായി ഉപയോഗിക്കുന്ന ചില പ്രോട്ടോക്കോളുകൾ (Protocols).

TCP/IP

ഇന്റർനെറ്റിലും സാധാരണ ശൃംഖലകളിലും പരസ്പരം ബന്ധിപ്പിച്ചിട്ടുള്ള ഉപകരണങ്ങളിൽ ഉപയോഗിക്കുന്ന നിയമങ്ങളാണ് TCP/IP (ട്രാൻസ്മിഷൻ കൺട്രോൾ പ്രോട്ടോകോൾ/ ഇന്റർനെറ്റ് പ്രോട്ടോക്കോൾ) (TCP/IP Transmission control protocol/Internet protocol) എന്നത്. ഇന്റർനെറ്റിൽ ഇലക്ട്രോണിക് ഉപകരണങ്ങൾ (കമ്പ്യൂട്ടർ പോലുള്ള) എങ്ങനെ ബന്ധിപ്പിക്കണമെന്നും അവ തമ്മിൽ എങ്ങനെ വിവര വിനിമയം നടത്തണമെന്നും TCP/IP നിർവചിച്ചിരിക്കുന്നു

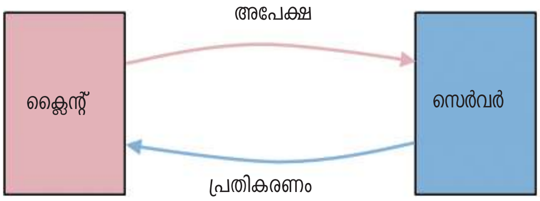
ഒരു കമ്പ്യൂട്ടറിൽ നിന്ന് മറ്റൊന്നിലേക്ക് ഡാറ്റ അയയ്ക്കുമ്പോൾ, TCP/IP ആദ്യം അവയെ വിഭജിച്ചു ചെറിയ പാക്കറ്റുകൾ ആക്കുകയും പിന്നീട് അയയ്ക്കുകയും ചെയ്യുന്നു. സ്വീകരിക്കേണ്ട കമ്പ്യൂട്ടറിൽ ഈ പാക്കറ്റുകൾ കിട്ടിക്കഴിഞ്ഞാൽ, ഈ പാക്കറ്റുകളിൽ തെറ്റുകളോ കേടുപാടുകളോ ഉണ്ടോ എന്ന് പരിശോധിക്കുന്നു. തകരാറുകൾ കണ്ടെത്തിയാൽ, ഈ പാക്കറ്റുകൾ വീണ്ടും അയയ്ക്കുന്നതിനുള്ള നിർദേശം TCP സമർപ്പിക്കുന്നു. തകരാറൊന്നും ഇല്ലെങ്കിൽ പാക്കറ്റുകളെ TCP യിൽ നിർദ്ദേശിച്ചിട്ടുള്ള നിയമങ്ങൾക്കനുസരിച്ചു സംയോജിപ്പിച്ച് യഥാർഥ സന്ദേശം ആക്കി മാറ്റുകയും ചെയ്യുന്നു. TCP/IP നിയമങ്ങളിൽ നടക്കുന്ന വിവിധ പ്രവർത്തനങ്ങൾ ചിത്രം 8.32 കൊടുത്തിരിക്കുന്നു. ഈ പാക്കറ്റുകൾ ലക്ഷ്യ സ്ഥാനത്തു എത്തിച്ചേരുന്നു എന്ന് ഉറപ്പാക്കുന്നത് ഇന്റർനെറ്റ് പ്രോട്ടോക്കോൾ ആണ്. ഒരേ സന്ദേശത്തിന്റെ വിവിധ പാക്കറ്റുകൾ പല പാതകളിലൂടെയാണ് സഞ്ചരിക്കുന്നതെങ്കിലും അവ ഒരേ ലക്ഷ്യസ്ഥാനത്തു എത്തിച്ചേരുകയും അവയെ അവിടെ വെച്ച് സംയോജിപ്പിക്കുകയും ചെയ്യുന്നു. HTTP, FTP, DNS തുടങ്ങിയ പ്രോട്ടോക്കോളുകളും TCP/ IP പ്രോട്ടോക്കോളിനുണ്ട്.



ചിത്രം 8.34 : TCP/IP യുടെ പ്രവർത്തനം

a. HTTP

ഹൈപ്പർ ടെക്സ്റ്റ് ട്രാൻസ്ഫർ പ്രോട്ടോകോൾ (Hyper Text Transfer Protocol) എന്നാണ് HTTP യുടെ പൂർണ രൂപം. ക്ലയന്റിൽ നിന്നുള്ള അഭ്യർഥന കൈമാറ്റം ചെയ്യുവാനും, സെർവറിൽ നിന്ന് പ്രതികരണങ്ങൾ സ്വീകരിക്കുവാനുമുള്ള അംഗീകൃത പെരുമാറ്റ ചട്ടങ്ങളാണിത്. ക്ലയന്റിൽ നിന്ന് ബ്രൗസർ വഴി അപേക്ഷ സ്വീകരിക്കുന്ന സർവർ, HTTPവഴി സേവനം നൽകുകയും ചെയ്യുന്നു. ഇത്തരം അഭ്യർഥനയുടെയും പ്രതികരണത്തിന്റെയും ജോഡികൾ HTTP സെഷൻ എന്ന് അറിയപ്പെടുന്നു. (ചിത്രം 8.35)



*ചിത്രം . 8.35 : **HTTP സെഷൻ***

ക്ലയന്റിൽ നിന്നുള്ള നിർദ്ദേശത്തെ തുടർന്ന് സെർവർ പ്രതികരിക്കുന്നത് രണ്ടു രീതിയിലാണ്. സെർവറിൽ മുൻകൂട്ടി സൂക്ഷിച്ചിട്ടുള്ള ഫയൽ അയച്ചു കൊടുത്തോ (Static രീതി) സെർവറിൽ സൂക്ഷിച്ചിട്ടുള്ള പ്രോഗ്രാം കോഡിന്റെ പ്രവർത്തന ഫലമായിട്ടുള്ള ഫയൽ അയച്ചു കൊടുത്തോ (Dynamic രീതി) ആകാം അത്.

HTTP യുടെ രണ്ടു പ്രധാന സവിശേഷതകൾ

- HTTP യിൽ വിവര വിനിമയ മാധ്യമത്തിന്റെ സ്വാധീനമില്ല.
- HTTP അസ്ഥിരമാണ് (അഭ്യർഥനയുടെയും പ്രതികരണത്തിന്റെയും സമയത്തു മാത്രം) ക്ലയന്റ് സർവർ ബന്ധം പരസ്പരം നിലനിർത്തുകയും അതിനുശേഷം ബന്ധം നിശേഷം വിച്ഛേദിക്കുകയും ചെയ്യുന്നു.

b. FTP

എഫ് ടി പി യുടെ പൂർണരൂപം ഫയൽ ട്രാൻസ്ഫർ പ്രോട്ടോകോൾ (File Transfer Protocol) എന്നാണ്. ഡാറ്റയും പ്രോഗ്രാം ഫയലുകളും ശൃംഖല വഴി പരസ്പരം കൈമാറ്റം ചെയ്യുവാൻ ഉപയോഗിക്കുന്ന അടിസ്ഥാന പ്രോട്ടോകോൾ ആണിത്. ഇന്റർനെറ്റിലൂടെ ലളിതമായ രീതിയിൽ കമ്പ്യൂട്ടറുകൾ തമ്മിൽ ഫയലുകൾ കൈമാറാനുള്ള മാർഗമാണ് ഇത്. TCP യും IP യും ഉപയോഗിച്ച് അയയ്ക്കുകയും സ്വീകരിക്കുകയും ചെയ്യുന്നു.

സെർവറിലെ സുരക്ഷാ മാർഗങ്ങൾ ആയ യൂസർ നാമവും പാസ്സ് വേർഡും ഉപയോഗിച്ച് ഫയലുകൾ സുരക്ഷിതമായി കൈമാറ്റം ചെയ്യുന്നത് ക്ലയന്റ് സെർവർ ഘടനയായ FTP ഉപയോഗിച്ചാണ്. FTP ക്ലയന്റ് പ്രോഗ്രാമുകളായ FileZ-illa, CUTEFTP എന്നിവ ഉപയോഗിച്ച് ഫയലുകൾ വളരെ എളുപ്പത്തിൽ അയയ്ക്കുവാനും സ്വീകരിക്കുവാനും കഴിയുന്നു.

c. DNS

ഡൊമൈൻ നെയിം സിസ്റ്റം (Domain Name System) എന്നാണ് DNS ന്റെ പൂർണ രൂപം. വെബ് ബ്രൗസറിന്റെ അഡ്രെസ്സ് ബാറിൽ നമ്മൾ ടൈപ്പ് ചെയ്യുന്ന വെബ് മേൽവിലാ

സത്തിന്റെ (ഡൊമൈൻ നാമം) IP മേൽവിലാസം DNS നമുക്ക് നൽകുന്നു. (മൊബൈൽ ഫോണിൽ ഒരു പേര് തിരഞ്ഞെടുക്കുമ്പോൾ അതിൽ ഫോൺ നമ്പർ ഉള്ളത് പോലെ)

DNS നു അതിന്റെതായ ശൃംഖലകൾ ഉണ്ട്. ഇന്റർനെറ്റിൽ ഉള്ള എല്ലാ വെബ്സൈറ്റുകളുടെയും IP മേൽവിലാസങ്ങളും ഡൊമെയിൻ നാമങ്ങളും ഒരു ഡാറ്റാബേസിൽ ശേഖരിച്ചിട്ടുണ്ട്. ഇന്റർനെറ്റിലെ ഓരോ നോഡിന്റെയും IP മേൽവിലാസം സ്ഥിരമാണ് എന്നതാണ് DNS ന്റെ അടിസ്ഥാനം. ഒരു DNS നു ഒരു ഡൊമൈൻ നാമത്തിനെ വിവർത്തനം ചെയ്തു IP മേൽവിലാസമാക്കുവാൻ കഴിഞ്ഞില്ലെങ്കിൽ അത് അടുത്ത DNS നോടും, അതിനും കഴിഞ്ഞില്ലെങ്കിൽ അതിനടുത്തതിനോടും വിവരവിനിമയം നടത്തും. ഈ പ്രക്രിയ ശരിയായ IP മേൽവിലാസം കിട്ടുന്നത് വരെ തുടരുന്നു.

TCP/IP, HTTP, FTP, DNS എന്നിവയല്ലാതെ ഏതെങ്കിലും അഞ്ചു പ്രോട്ടോക്കോളുകളെക്കുറിച്ച് കുറിപ്പ് തയാറാക്കുക.

നമുക്കു ചെയ്യാം

8.11 യൂണിഫോം റിസോഴ്സ് ലൊക്കേറ്റർ (Uniform Resource Location (URL))

യൂണിഫോം റിസോഴ്സ് ലൊക്കേറ്റർ എന്നതാണ് URL- ന്റെ പൂർണ രൂപം. URL എന്നത് ക്രമീകരിച്ച വാക്കുകൾ ഉപയോഗിച്ച് വെബ് ബ്രൗസറുകൾ, ഇമെയിൽ പ്രോഗ്രാമുകൾ, മറ്റു സോഫ്റ്റ്‌വെയറുകൾ തുടങ്ങിയവയെ ഇന്റർനെറ്റിൽ തിരിച്ചറിയുവാൻ സഹായിക്കുന്ന ഒന്നാണ്. ഇന്റെർനെറ്റിലുള്ള എല്ലാ വിഭവങ്ങൾക്കും (resources) തനതായ URL ഉണ്ടായിരിക്കും. ഫയലുകൾ, അതുൾപ്പെടുന്ന വെബ്പേജുകൾ മറ്റു ഡോക്യൂമെന്റുകൾ, ഗ്രാഫിക്സ്, പ്രോഗ്രാമുകൾ തുടങ്ങിയവയാണ് ശൃംഖല വിഭവങ്ങൾ (Network resources). ഒരു URL-ൽ അക്ഷരങ്ങൾ, അക്കങ്ങൾ, ചിഹ്നങ്ങളും ഉണ്ട്.

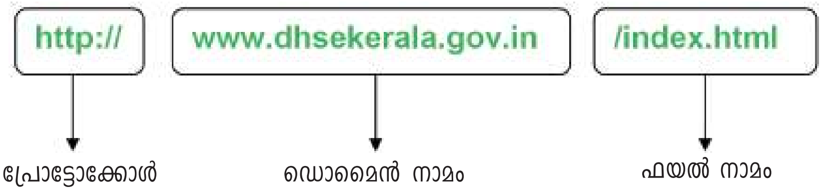
ഒരു URL മേൽവിലാസത്തെ മൂന്നായി തരം തിരിച്ചിരിക്കുന്നു

a) നെറ്റ്‌വർക്ക് പ്രോട്ടോക്കോൾ

b) ഡൊമൈൻ നാമം (ഹോസ്റ്റിന്റെ പേര് അല്ലെങ്കിൽ വിലാസം)

c) ഫയൽ നാമം

ഉദാഹരണത്തിന് http://www.dhsekerala.gov.in/index.html എന്ന URL ന് മൂന്നു ഭാഗങ്ങൾ ഉണ്ട്. ചിത്രം 8.36 ഈ URL ന്റെ വിവിധ ഘടകങ്ങൾ കാണിച്ചിരിക്കുന്നു.



*ചിത്രം 8.36 : **URL** ന്റെ ഘടകങ്ങൾ*

മൂന്നു വിഭാഗങ്ങളുടെയും വിശദ വിവരങ്ങൾ ചുവടെ ചേർത്തിരിക്കുന്നു.

a) പ്രോട്ടോക്കോൾ (Protocol)

ഡൊമൈനിൽ നിന്ന് വിവരങ്ങൾ ഏതു പ്രോട്ടോകോൾ ഉപയോഗിച്ചാണ് സ്വീകരിക്കേണ്ടത് എന്നത് ബ്രൗസറിനെ അറിയിക്കുന്നു.

b) ഡൊമൈൻ നാമം (Domain name)

ഡൊമൈൻ നാമം എന്നത് ഡൊമൈൻ നെയിം സിസ്റ്റം വഴി സെർവറിനു നൽകിയ പേരാണ്. ഒരു URL ലെ ഡൊമൈൻ നാമം ഒരു വെബ് സെർവറിനെ കണ്ടെത്തുവാൻ സഹായിക്കുന്നു. വളരെ എളുപ്പത്തിൽ ഓർമ്മ നിൽക്കുന്ന വിധത്തിൽ ഇന്റർനെറ്റ് ഉപഭോക്താക്കൾക്ക് ഹ്രസ്വനാമത്തിൽ കിട്ടുന്നു. കമ്പ്യൂട്ടർ ഉപയോഗിച്ച് ഇന്റർനെറ്റിലൂടെ വിവരവിനിമയം നടത്താൻ, IP അഡ്രസ്സ് ഉപയോഗിക്കാവുന്നതാണ്. എന്നാൽ എല്ലാ കമ്പ്യൂട്ടറുകളുടെയും IP അഡ്രസ് ഓർത്തിരിക്കുക എന്നത് പ്രായോഗികമല്ല. അതുകൊണ്ട് വെബ് സർവറിനു പേര് നൽകുകയും, ഈ പേരിനു തുല്യമായ IP വിലാസങ്ങളുടെ ഒരു പട്ടിക ഉണ്ടാക്കി സൂക്ഷിക്കുക എന്ന സമ്പ്രദായം കൊണ്ട് വന്നു. ഇതിനെയാണ് ഡൊമൈൻ നാമം എന്ന് പറയുന്നത്. ഉദാഹരണം dhsekerala.gov.in, keralaresults. nic.in ,google.com, gmail.com.

ഒരു ഡൊമൈൻ നാമത്തിനു സാധാരണ ഒന്നിൽ കൂടുതൽ ഭാഗങ്ങൾ ഉണ്ട്. ടോപ് ലെവൽ ഡൊമൈൻ അഥവാ പ്രാഥമിക ഡൊമൈൻ, ഉപ ഡൊമൈൻ എന്നിവ.

മുകളിൽ സൂചിപ്പിച്ച ഉദാഹരണത്തിൽ in എന്നത് പ്രാഥമിക ഡൊമൈനും, gov എന്നത് in ന്റെ ഉപ ഡൊമൈനും, dhsekerala എന്നത് gov യുടെ ഉപ ഡൊമൈനുമാണ്.

വളരെ കുറച്ചു പ്രാഥമിക ഡൊമൈനുകൾ ആണ് ഉള്ളത്. അവയെ രണ്ടു വിഭാഗങ്ങൾ ആയി തരം തിരിച്ചിരിക്കുന്നു. പൊതുവായ ഡൊമൈൻ നാമങ്ങൾ (Generic domain names) എന്നും രാജ്യത്തിന്റെ പ്രത്യേക ഡൊമൈൻ നാമങ്ങൾ (Country specific domain names) എന്നും. പൊതുവായ/രാജ്യ ഡൊമൈൻ നാമങ്ങളുടെ ഉദാഹരണങ്ങൾ പട്ടിക 8.2 ൽ കൊടുത്തിരിക്കുന്നു.

Generic Domain Names	
·com	Commercial business
·edu	Educational institutions
·gov	Government agencies
·mil	Military
·net	Network organizations
·org	Organizations (nonprofit)

Country Specific Domain Names	
·in	India
·au	Australia
·ca	Canada
·ch	China
.jp	Japan
.us	United States of America

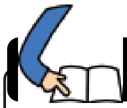
പട്ടിക 8.2 : പൊതുവായതും രാജ്യത്തിന്റെ പ്രത്യേക ഡൊമൈൻ നാമങ്ങളും

c.ഫയൽ നാമം (File Name)

ഏതു ഫയൽ ആണോ തുറക്കേണ്ടത് അതിനെ സൂചിപ്പിക്കുന്നതാണ് ഈ ഭാഗം. ചിത്രം 8.35ലെ ഉദാഹരണത്തിൽ കൊടുത്തിരിക്കുന്ന ഡൊമൈൻ നാമം നൽകുമ്പോൾ വെബ് സർവർ index.html എന്ന ഫയലാണ് അയച്ചു തരിക.

നമുക്കു ചെയ്യാം

പൊതുവായ ഡൊമൈനും രാജ്യത്തിന്റെ ഡൊമൈൻ നാമവും ഉൾകൊള്ളുന്ന URL ന്റെ സാധുവായ ഉദാഹരണ പട്ടിക തയ്യാറാക്കുക. തുറന്ന് വന്ന ഫയലിന്റെ പേര് എന്താണ് എന്ന് ശ്രദ്ധിക്കുക. (തുറന്നതിനുശേഷം അഡ്രസ് ബാറിൽ കാണുന്ന പേര് ആകും ഫയലിന്റെ പേര്)



നമുക്ക് സംഗ്രഹിക്കാം

ഈ നൂറ്റാണ്ടിന്റെ അവശ്യഘടകമായ കമ്പ്യൂട്ടർ ശ്യംഖലയെ കുറിച്ച് നമ്മൾ ഈ അധ്യായത്തിൽ പഠിച്ചു. ശ്യംഖലയുടെ പ്രാധാന്യത്തെക്കുറിച്ചും അവ നൽകുന്ന നേട്ടങ്ങളെ കുറിച്ചും ചർച്ച ചെയ്തു. വിവിധ വിവര വിനിമയ ഭൗതിക മാദ്ധ്യമങ്ങളുടെ നിർമ്മിതിയെക്കുറിച്ചും അവയുടെ നേട്ടങ്ങളും കോട്ടങ്ങളും അവയുടെ പ്രവർത്തനങ്ങളെക്കുറിച്ചും നാം ചർച്ച ചെയ്തു. ശ്യംഖല രൂപകൽപ്പന ചെയ്യുമ്പോൾ, ഉപയോഗിക്കുന്ന വിവിധതരം ഉപകരണങ്ങളെക്കുറിച്ചും മനസ്സിലാക്കി.

വിവിധ ശ്യംഖലയെക്കുറിച്ചും ചർച്ച ചെയ്യുന്നതിന് മുൻപ്,ടോപ്പോളജി എന്ന പദത്തിലൂടെ വിവിധ വിധത്തിലുള്ള ശ്യംഖലയുടെ ക്രമീകരണളെക്കുറിച്ചു പഠിച്ചു. TCP/IP പോലുള്ള ശ്യംഖല പെരുമാറ്റ ചട്ടങ്ങൾ ഉപയോഗിച്ച് വിവരങ്ങൾ കൈമാറ്റം ചെയ്യുന്നത് എങ്ങനെ എന്ന് ചർച്ച ചെയ്തു.ഒരു ശ്യംഖലയിലെ നോഡിനെ കണ്ടെത്തുന്നത് എങ്ങനെ എന്ന് പഠിച്ചു. URL നെ കുറിച്ചുള്ള ചർച്ചയോടു കൂടി ഈ പാഠഭാഗം ഉപസംഹരിച്ചു.

പഠന നേട്ടങ്ങൾ

ഈ അധ്യായത്തിന്റെ പൂർത്തീകരണത്തോടുകൂടി പഠിതാവിന്

- വിവരവിനിമയ മാധ്യമത്തെ തിരഞ്ഞെടുക്കുവാനും മനസ്സിലാക്കുവാനും കഴിയുന്നു.
- വ്യത്യസ്ത ശ്യംഖലകളെ താരതമ്യം ചെയ്യാൻ
- [redacted]ക്താധിഷ്ഠിത തരം തരിവുകൾ
- [redacted] അയയ്ക്കുന്നത് മനസ്സിലാക്കുക
- ലളിതമായ ഒരു ശ്യംഖല നിർമ്മിക്കുക
- ശ്യംഖലയിലെ ഒരു നോഡിനെ എങ്ങനെ തിരിച്ചറിയാം
- ഒരു URL ന്റെ വിവിധ ഭാഗങ്ങൾ തിരിച്ചറിയുക

മാതൃക ചോദ്യങ്ങൾ

ഹ്രസ്വോത്തര ചോദ്യങ്ങൾ

1. പ്രകാശ തരംഗങ്ങളുടെ രൂപത്തിൽ വിവരങ്ങൾ വഹിച്ചു കൊണ്ട് പോകുന്ന സംപ്രേക്ഷണ മാദ്ധ്യമമാണ്

   എ) കൊയാക്സിൽ കേബിൾ ബി) ട്വിസ്റ്റഡ് പെയർ
   സി) വൈ ഫൈ ഡി) ഒപ്റ്റിക്കൽ ഫൈബർ

2. വ്യത്യസ്തത പെരുമാറ്റ ചട്ടങ്ങളുള്ള വ്യത്യസ്ത ശൃംഖലകളെ പരസ്പരം ബന്ധിപ്പിക്കുന്ന ഉപകരണമാണ് ....

   എ) റൂട്ടർ ബി) ബ്രിഡ്ജ് സി) സ്വിച്ച് ഡി) ഗേറ്റ് വെയ്

3. .....ക്രമീകരണത്തിൽ ഒരു കമ്പ്യൂട്ടറിന്റെ തകരാർ മൊത്തം ശൃംഖലയുടെ പ്രവർത്തനത്തെയും ബാധിക്കുന്നു

   എ) ബസ് ബി) റിങ് സി) സ്റ്റാർ ഡി) ഇവയൊന്നും ഇല്ല

4. വിവിധ ഉപകരണങ്ങളിൽ നിന്നുള്ള തരംഗങ്ങളെ ഒരൊറ്റ വിനിമയ മാദ്ധ്യമത്തിലൂടെ ഒരേ സമയത്തു കടത്തിവിടുവാൻ . ... ഉപകരണം ഉപയോഗിക്കുന്നു

   എ) മോഡം ബി) സ്വിച്ച് സി) റൂട്ടർ ഡി) മൾട്ടിപ്ളെക്സർ

5. ബ്ലൂ ടൂത്ത് ഉപയോഗിക്കുന്നത്

   എ) ദീർഘ ദൂര വിനിമയത്തിന് ബി) ഹ്രസ്വദൂര വിനിമയത്തിന്
   സി) മൊബൈൽ ഫോണിൽ മാത്രം ഡി) ഇവയൊന്നും അല്ല

6. സാറ്റലൈറ്റ് ലിങ്കുകൾ പൊതുവെ ഉപയോഗിക്കുന്നത്

   എ) PANS ബി) LANS സി) MANS ഡി) ഇവയിലെല്ലാം

7. ഒരു ഡൊമൈൻ നാമം ബന്ധിപ്പിച്ചിരിക്കുന്നത്

   എ) URL ബി) ഐപി അഡ്രസ് സി) വെബ്സൈറ്റ് ഡി) ഇവയെല്ലാം

ലഘു ഉപന്യാസ ചോദ്യങ്ങൾ

1. ബാൻഡ് വിഡ്ത് നിർവ്വചിക്കുക
2. ശൃംഖലകളുമായി ബന്ധപ്പെട്ട രണ്ടു ഉപകരണങ്ങളാണ് സ്വിച്ചും ഹബ്ബും. ഇവയെ വേർതിരിക്കുക.
3. ഐപി അഡ്രസ് എന്നാൽ എന്താണ്? ഒരു ഉദാഹരണം എഴുതുക.
4. എന്താണ് TCP/IP? ഇതിന്റെ പ്രാധാന്യം എന്ത്?
5. കമ്പ്യൂട്ടർ ശൃംഖലയെ നിർവ്വചിക്കുക
6. എന്താണ് ബ്ലൂടൂത്ത്?
7. എന്താണ് മോഡം?
8. റൂട്ടറും ഗേറ്റ് വേയും തമ്മിലുള്ള വ്യത്യാസം എന്താണ്?
9. കമ്പ്യൂട്ടർ ശൃംഖല നിർമ്മിക്കുന്നതിന്റെ ആവശ്യകത വിശദീകരിക്കുക?
10. കമ്പ്യൂട്ടർ ശൃംഖലയുടെ ഉപയോഗങ്ങൾ എന്തൊക്കെയാണ്?

11. മൈക്രോവേവ് സംപ്രേക്ഷണത്തിന്റെ പോരായ്മകൾ എന്തൊക്കെയാണ് ? എങ്ങനെ അതിനെ മറികടക്കാം?

12. വൈ ഫൈ യുടെ സവിശേഷതകൾ എന്തൊക്കെയാണ് ?

13. ഒരു അന്തർദേശീയ സ്കൂൾ 45 m ചുറ്റളവിൽ സ്ഥാപിച്ചിരിക്കുന്ന കമ്പ്യൂട്ടറുകളെ തമ്മിൽ ബന്ധിപ്പിക്കുവാൻ ആലോചിക്കുന്നു.ഇതിന് ഉതകുന്ന സാമ്പത്തിക ലാഭമുള്ളതും അതിവേഗതയുള്ളതും ആയ മാദ്ധ്യമം തെരഞ്ഞെടുക്കുക

14. എന്താണ് NIC? ശൃംഖലയിൽ അവയുടെ പ്രാധാന്യം എന്താണ് ?

15. ഒരു സ്ഥാപനത്തിലെ കമ്പ്യൂട്ടർ ശൃംഖലയുടെ മേലധികാരിയാണ് നിങ്ങൾ എന്ന് സങ്കൽപ്പിക്കുക . ശൃംഖലയിലെ 10 MBPS ന്റെ SWITCH മാറ്റി 10 MBPS ന്റെ HUB വെയ്ക്കുവാൻ നിങ്ങളോടു മേലധികാരി നിർദ്ദേശിക്കുന്നു ? ഇതിനോട് നിങ്ങൾ യോജിക്കുന്നുണ്ടോ ? നിങ്ങളുടെ അഭിപ്രായം സാധൂകരിക്കുക ?

16. നിങ്ങളുടെ ബയോഡാറ്റ 10KM അകലെയുള്ള കൂട്ടുകാരന്റെ കംമ്പ്യൂട്ടറിലേക്കു ടെലിഫോൺ ശൃംഖല വഴി കൈമാറ്റം ചെയ്യണമെങ്കിൽ

    എ) രണ്ട് ഭാഗത്തും ഉണ്ടായിരിക്കേണ്ട ഉപകരണത്തിന്റെ പേര് എഴുതുക?

    ബി) രണ്ടു കമ്പ്യൂട്ടറുകൾ തമ്മിൽ ബന്ധം സ്ഥാപിച്ചു കഴിഞ്ഞാൽ, ഈ ഉപകരണത്തിലൂടെ ഫയലുകൾ അയയ്ക്കുകയും സ്വീകരിക്കുകയും ചെയ്യുന്നത് എങ്ങനെയാണ്?

17. ഒരു കമ്പ്യൂട്ടർ ശൃംഖലയിൽ റിപീറ്റർ ഉപയോഗിക്കേണ്ടി വരുന്നത് എപ്പോൾ?

18. ഇൻഫ്രാറെഡും ബ്ലൂടൂത്ത് സംപ്രേക്ഷണവും തമ്മിൽ താരതമ്യം ചെയ്യുക ?

19. ടെലിഫോൺ ശൃംഖലയുമായി കമ്പ്യൂട്ടറുകളെ ബന്ധിപ്പിക്കുവാൻ ഉപയോഗിക്കുന്ന ഉപകരണമേത്? ഇതിന്റെ പ്രവർത്തനം വിശദീകരിക്കുക?

20. ലാൻ ക്രമീകരണം (LAN TOPOLOGY) വിശദീകരിക്കുക ?

21. TCP/IP പെരുമാറ്റ ചട്ടങ്ങൾ ചുരുക്കി എഴുതുക?

22. എന്താണ് MAC അഡ്രസ് ? MAC അഡ്രസ് IP അഡ്രസ് ഉം തമ്മിലുള്ള വ്യത്യാസം എന്താണ് ?

ഉപന്യാസ ചോദ്യങ്ങൾ

1. കമ്പ്യൂട്ടർ ശൃംഖലകളെ അവയുടെ വലുപ്പമനുസരിച് എങ്ങനെ തരം തിരിച്ചിരിക്കുന്നു ?

2. വ്യത്യസ്ത ലാൻ ക്രമീകരണങ്ങളെ (LAN TOPOLOGY) കളെ താരതമ്യം ചെയ്യുക ?

3. വിവിധ തരത്തിലുള്ള ഗൈഡഡ് വിനിമയ ചാനലുകളെ കുറിച്ച് വിശദീകരിക്കുക ?

4. വ്യത്യസ്ത അൺ ഗൈഡഡ് മാദ്ധ്യമങ്ങൾ തമ്മിൽ താരതമ്യം ചെയ്യുക ?

5. പെരുമാറ്റച്ചട്ടം (പ്രോട്ടോകോൾ) എന്ന പദം നിർവ്വചിക്കുക? ഏതെങ്കിലും രണ്ടു വിനിമയ പെരുമാറ്റ ചട്ടങ്ങൾ ചുരുക്കി വിശദീകരിക്കുക ?

6. ശൃംഖലയിൽ ഉപയോഗിക്കുന്ന വിവിധ തരത്തിലുള്ള വിവര വിനിമയ ഉപകരണങ്ങളെ കുറിച്ച് ചുരുക്കി വിശദീകരിക്കുക ?

7. താഴെ പറയുന്ന സന്ദർഭങ്ങളിൽ ഏതു തരത്തിലുള്ള വിനിമയ മാദ്ധ്യമമാണ് അനുയോജ്യമാകുക?

a. LAN സ്ഥാപിക്കുക

b. ലാപ്ടോപ്പിൽ നിന്നും മൊബയിലേക്കു ഡാറ്റ കൈമാറുക

c. ഒരു മൊബയിൽ ഫോണിൽ നിന്ന് മറ്റൊരു മൊബൈൽ ഫോണിലേക്കു ഡാറ്റ കൈമാറുക

d. ഒന്നിൽ കൂടുതൽ ഉപകരണങ്ങൾ നിയന്ത്രിക്കുന്ന ഒരു റിമോട്ട് കൺട്രോൾ ഉണ്ടാക്കുക.

e. രണ്ടു രാജ്യത്തുള്ള രണ്ടു സ്ഥാപനങ്ങൾ തമ്മിലുള്ള അതിവേഗ വിവര വിനിമയം

f. കുന്നിൻപ്രദേശത്തുള്ള (മലയോര മേഖലകളിൽ) വിവരവിനിമയം

g. നഗരത്തിനുള്ളിലോ നഗര പരിധിക്കുള്ളിലോ കേബിൾ ഉപയോഗിച്ചുള്ള ചിലവേറിയ വിവര വിനിമയം?

9

പ്രധാന ആശയങ്ങൾ

- ഇന്റർനെറ്റിന്റെ ചരിത്രം
- കമ്പ്യൂട്ടറിലേക്ക് ഇന്റർനെറ്റ് ബന്ധിപ്പിക്കുന്നു.
- ബന്ധിപ്പിക്കൽ രീതികൾ
  - ഡയൽ അപ്
  - വയേർഡ് ബ്രോഡ്ബാന്റ്
  - വയർലെസ് ബ്രോഡ്ബാന്റ്
  - ഇന്റർനെറ്റ് പങ്കുവയ്ക്കുന്ന രീതികൾ
- ഇന്റർനെറ്റിലെ സേവനങ്ങൾ
  - WWW
  - സേർച്ച് എൻജിനുകൾ
  - ഈ-മെയിൽ
  - സാമൂഹിക മാധ്യമങ്ങൾ
- സൈബർ സുരക്ഷ
  - കമ്പ്യൂട്ടർ വൈറസ്, വേം, ട്രോജൻ ഹോർസ്, സ്പാം, ഹാക്കിങ്ങ്, ഫിഷിങ്ങ്, ഡിനൈൽ ഓഫ് സർവ്വീസ് ആക്രമണം, മാൻ ഇൻ ദ മിഡിൽ ആക്രമണങ്ങൾ.
- ശൃംഖലാക്രമണം തടയുന്നു.
  - ഫയർവാൾ ആന്റിവൈറസ് സ്കാനറുകൾ, കുക്കീസ്
- കമ്പ്യൂട്ടറുകളിൽ ഇന്റർനെറ്റ് ഉപയോഗിക്കുമ്പോൾ അനുവർത്തിക്കേണ്ട മാർഗ നിർദേശങ്ങൾ

D9S7I9

ഇന്റർനെറ്റ്

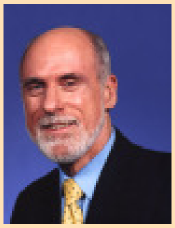
ലോകത്തിലെ ഏറ്റവും വലിയ കമ്പ്യൂട്ടർ ശൃംഖല ഇന്റർനെറ്റ് ആണെന്ന് കഴിഞ്ഞ അധ്യായത്തിൽ നമ്മൾ പഠിച്ചു. എസ്.എസ്.എൽ.സി (S.S.L.C) പരീക്ഷാഫലം അറിയുക, ഹയർ സെക്കന്ററി പ്രവേശനത്തിന് അപേക്ഷ സമർപ്പിക്കുക, സമയാസമയങ്ങളിൽ അപേക്ഷയുടെ പുതുക്കിയ വിവരങ്ങൾ അറിയുക, വിവിധയിനം സ്കോളർഷിപ്പുകളെപ്പറ്റി മനസ്സിലാക്കി അപേക്ഷ സമർപ്പിക്കുക തുടങ്ങിയവയ്ക്ക് ഇന്റർനെറ്റ് സൗകര്യം പ്രയോജനപ്പെടുത്തുന്നു. ഇക്കാലത്ത് ഇന്റർനെറ്റ് ഇല്ലാതെയുള്ള ജീവിതം നിങ്ങൾക്ക് സങ്കൽപ്പിക്കാനാകുമോ? ഈ പറഞ്ഞ ജോലികളെല്ലാം ഇന്റർനെറ്റ് ഇല്ലാതെ കൈകാര്യം ചെയ്യുക എന്നത് തുലോം ദുഷ്കരമായിരിക്കും. ഇന്റർനെറ്റ് നമ്മുടെ ജീവിതം നിശ്ചയമായും ആയാസരഹിതമാക്കിത്തീർക്കുന്നു. ഇന്റർനെറ്റ് നമ്മുടെ നിത്യജീവിതത്തിൽ ഒരു വലിയ അളവോളം സ്വാധീനം ചെലുത്തുന്നു. ഇന്റർനെറ്റിന്റെ വ്യാപക ജനപ്രിയതയും ഉപയോഗത്തിന്റെ വർദ്ധനവും കാരണം ഇന്റർനെറ്റ് ബന്ധം ഉള്ള ടെലിവിഷൻ സെറ്റുകൾ പോലും കമ്പോളത്തിൽ എത്തിക്കഴിഞ്ഞു.

വിവരങ്ങൾ തിരയുവാനും, ഇ-മെയിൽ നോക്കുന്നതിനും, ബില്ലുകൾ അടയ്ക്കുന്നതിനും, ഓൺലൈൻ ഷോപ്പിങ്ങിനും, ഓൺലൈൻ ബാങ്കിങ്ങിനും, സോഷ്യൽ നെറ്റ് വർക്കിംഗ് സൈറ്റുകളിലൂടെ മറ്റ് ആളുകളുമായി ബന്ധപ്പെടുന്നതിനും ജനങ്ങൾ പൊതുവെ ഇന്റർനെറ്റ് ഉപയോഗിക്കുന്നു. ഇന്റർനെറ്റിന്റെ സാധ്യതകൾ വളരെ വിപുലവും അത് നമ്മുടെ ചെലവും സമയവും കുറയ്ക്കുവാനും സഹായിക്കുന്നു. ഇതൊക്കെയാണെങ്കിലും ഓൺലൈനിലൂടെയുള്ള മറ്റുള്ളവരുടെ സ്വകാര്യതയിലേക്കുള്ള

നുഴഞ്ഞുകയറ്റം, തട്ടിപ്പ്, സൈബർ ആക്രമം തുടങ്ങിയവ ഇപ്പോൾ നമ്മുടെ സമൂഹത്തിൽ വ്യാപിച്ചുകൊണ്ടിരിക്കുന്നു. ഇന്റർനെറ്റും അതിന്റെ പ്രവേശന രീതികൾക്കുമുപരി ഇന്റർനെറ്റിന്റെ സേവനങ്ങളായ സെർച്ച് എൻജിൻ, ഇ-മെയിൽ, സാമൂഹിക മാധ്യമങ്ങൾ, ഇന്റർനെറ്റ് എന്നിവ ഉപയോഗിക്കുമ്പോൾ ഉണ്ടാകുന്ന ഭീഷണികളും അവയ്ക്കുള്ള പ്രതിരോധ മാർഗങ്ങളും ഈ അധ്യായത്തിൽ നമുക്ക് ചർച്ച ചെയ്യാം.

9.1 ഇന്റർനെറ്റിന്റെ ചരിത്രം (History of the Internet)

അമേരിക്കൻ പ്രതിരോധ വകുപ്പിന്റെ പദ്ധതിയായ ARPANET (Advanced Research Projects Agency Network) എന്ന പേരിൽ ഒരു ചെറിയ ശൃംഖല (നെറ്റ്‌വർക്ക്) യിലൂടെ ഇന്റർനെറ്റിന് തുടക്കമിട്ടു. ആയിരത്തിതൊള്ളായിരത്തി എഴുപതുകളിൽ അമേരിക്കൻ പ്രതിരോധ സേനക്ക് വേണ്ടി പ്രവർത്തിച്ചിരുന്ന സർവ്വകലാശാലകളിലേയും തൊഴിൽ സ്ഥാപനങ്ങളിലെയും കമ്പ്യൂട്ടറുകൾ ഈ ശൃംഖലയുമായി ബന്ധിപ്പിച്ചു. 1984 ൽ ARPANET ൽ നിന്നും വേർതിരിഞ്ഞ് അമേരിക്കൻ സൈന്യത്തിന്റെ ഉപയോഗത്തിന് മാത്രമായി സൈനിക ശൃംഖലയായ MILNET രൂപം കൊണ്ടു. ആശയ വിനിമയത്തിനായി TCP/IP പ്രോട്ടോകോൾ ഉപയോഗിച്ചിരുന്ന ARPANET അതിനു ശേഷം ശാസ്ത്രീയ ഗവേഷണങ്ങൾക്കും ആശയ വിനിമയത്തിനും ഉപയോഗിച്ചു. പിന്നീട് മറ്റനവധി ശൃംഖലകൾ കൂട്ടിചേർത്ത് ARPANET ഒരു വലിയ ശൃംഖലയായി. ലോകത്തിലെ ആദ്യ വൈഡ് ഏരിയ നെറ്റ്‌വർക്ക് ആയി ARPANET കണക്കാക്കപ്പെടുന്നു. ഇന്റർനെറ്റിന്റെയും TCP/IP പ്രോട്ടോകോളിന്റെയും വികാസത്തിന് പങ്ക് വഹിച്ച വിന്റൻ ഗ്രേ സെർഫിനെ ഇന്റർനെറ്റിന്റെ പിതാവായി അറിയപ്പെടുന്നു.



വിന്റൻ ഗ്രേ സെർഫ് (1943 -) ഒരു അമേരിക്കൻ കമ്പ്യൂട്ടർ ശാസ്ത്രജ്ഞനാണ് വിന്റ് സെർഫ് എന്ന് വിളിക്കപ്പെടുന്ന ഇദ്ദേഹം ഇന്റർനെറ്റിന്റെ പിതാവ് എന്ന് അറിയപ്പെടുന്നു. അദ്ദേഹത്തിന്റെ സഹവർത്തിയും അമേരിക്കൻ കമ്പ്യൂട്ടർ ശാസ്ത്രജ്ഞനുമായ ബോബ് കാഹ്നിനൊടൊപ്പം ഇന്റർനെറ്റിന്റെ പ്രാരംഭ വികാസത്തിന് കാരണഭൂതനായി. അദ്ദേഹം അമേരിക്കയുടെ Defence Advanced Research Project Agency വകുപ്പിൽ ജോലി ചെയ്യുകയും TCP/IP പ്രോട്ടോകാളിന്റെ വികാസത്തിന് പ്രധാന പങ്ക് വഹിക്കുകയും ചെയ്തു. ഇദ്ദേഹം ICANN ന്റെ രൂപീകരണത്തിലും ഉൾപ്പെട്ടിട്ടുണ്ട്.

1989 ൽ ഗവേഷകനായ ടിം ബെർണേഴ്സ് ലീ വേൾഡ് വൈഡ് വെബ് (WWW) എന്ന ആശയം മുന്നോട്ട് വച്ചു. ടിം ബർണേഴ്സ് ലീയും അദ്ദേഹത്തിന്റെ കൂട്ടാളികളും കൂടി ഹൈപ്പർ ടെക്സ്റ്റ് ട്രാൻസ്ഫർ പ്രോട്ടോകോൾ (HTTP), HTML, വെബ് ബ്രൗസർ, വെബ് സെർവറിന്റെ സാങ്കേതിക വിദ്യ എന്നീ കണ്ടുപിടിത്തങ്ങളിലൂടെ ഖ്യാതി നേടി. ഇതിലൂടെ വെബ് ഡവലപ്പർക്ക് ഹൈപ്പർ ടെക്സ്റ്റ് ഉൾകൊള്ളിച്ച് വെബ് പേജുകൾ തമ്മിൽ ബന്ധിപ്പിക്കുവാൻ ഹൈപ്പർ ലിങ്കുകൾ ഉപയോഗിക്കാൻ സാധിച്ചു. ശബ്ദം, വാക്യം ചിത്രം എന്നിവ ഉൾപ്പെടുത്തി ആകർഷകമായ വെബ് പേജുകൾ രൂപകൽപ്പന ചെയ്യുവാൻ കഴിഞ്ഞു. 1990 കളിൽ ഈ മാറ്റം ഇന്റർ നെറ്റിന്റെ ബൃഹത്തായ വികാസത്തിന് കാരണമായി.

ചിത്രം 9.1 ടിം ബർണേഴ്സ് ലീ.

ഇന്റർനെറ്റിനെ ഒരു ആഗോള പ്രതിഭാസമാക്കി മാറ്റാൻ ഭൂമി ശാസ്ത്രപരമായി വിദൂര സ്ഥലങ്ങളിലുള്ള പലവിധ സ്ഥാപനങ്ങളിലെ വിഭിന്നമായ ഓപ്പറേറ്റിങ്ങ് സിസ്റ്റങ്ങളാൽ ഉപയോഗിക്കപ്പെടുന്ന വിവിധതരം കമ്പ്യൂട്ടറുകൾ കൂട്ടിച്ചേർത്തു. ഇന്റർനെറ്റിന്റെ കമ്മ്യൂണിക്കേഷൻ പ്രോട്ടോകോൾ ആയി ഉപയോഗിക്കുന്നത് TCP/IP ആണ്. ഇന്റർനെറ്റിലേക്ക് കൂട്ടിചേർക്കുന്ന ഏത് കമ്പ്യൂട്ടറും TCP/IP പ്രോട്ടോകോൾ പിന്തുടരണം. 1998 ൽ ഇന്റർനെറ്റ് കോർപ്പറേഷൻ ഫോർ സൈൻഡ് നെയിംസ് ആൻഡ് നമ്പേഴ്സ് സ്ഥാപിക്കപ്പെട്ടു. ICANN ഇന്റർനെറ്റിന്റെ ഉള്ളടക്കത്തെ നിയന്ത്രിക്കുകയല്ല, മറിച്ച് അത് ഇന്റർനെറ്റിന്റെ യൂണിഫോം റിസോഴ്സ് ലൊക്കേറ്ററിന്റെ നയങ്ങൾ വികസിപ്പിക്കുകയാണ് ചെയ്യുന്നത്.

ഇന്ന് ഇ-മെയിൽ, വിവരങ്ങൾ തിരയൽ, ഫയൽ കൈമാറ്റം, സോഷ്യൽ നെറ്റ്‌വർക്ക് തുടങ്ങിയ ഒട്ടനവധി സേവനങ്ങൾ നൽകുന്ന ലോകത്തിലെ കോടിക്കണക്കിന് വരുന്ന കമ്പ്യൂട്ടറുകളെ പരസ്പരം ബന്ധിപ്പിക്കുന്ന ഏറ്റവും വലിയ പൊതുവായ ശൃംഖലയാണ് ഇന്റർനെറ്റ്. ഇന്റർനെറ്റ് ലോകത്തെവിടെയുമുള്ള ഉപയോക്താക്കളെ സേവിക്കുന്ന പരസ്പരം ബന്ധിപ്പിച്ച കമ്പ്യൂട്ടർ ശൃംഖലയാണ്.

ഇന്റർനെറ്റ് പോലെ തന്നെ ഒരു സ്ഥാപനത്തിനുള്ളിൽ TCP/IP പ്രോട്ടോകോൾ ഉപയോഗിച്ച് പങ്കുവയ്ക്കുന്ന വിവരം, സോഫ്റ്റ്‌വെയർ അല്ലെങ്കിൽ സേവനങ്ങൾ എന്നിവക്കുള്ള സ്വകാര്യ കമ്പ്യൂട്ടർ ശൃംഖലയാണ് ഇൻട്രാനെറ്റ്. ഇൻട്രാനെറ്റിലൂടെ വെബ് ഹോസ്റ്റ്, ഇ-മെയിൽ സേവനം, ഫയൽ കൈമാറ്റം, ഇന്റർനെറ്റിലൂടെയുള്ള മറ്റ് സേവനങ്ങൾ എന്നിവ സാധ്യമാകും.

ഒരു കമ്പനിയുടെ സ്വകാര്യ ശൃംഖലയുടെ ഭാഗമല്ലാത്ത ചില കമ്പ്യൂട്ടറുകൾക്ക് ഇൻട്രാനെറ്റ് ലഭ്യമാകുമ്പോൾ അതിനെ എക്സ്ട്രാനെറ്റ് എന്ന് വിളിക്കുന്നു. കമ്പനിയുടെ കൂട്ടുകച്ചവടക്കാർക്കും, വിൽപ്പനക്കാർക്കും വിഭവങ്ങൾ ഉപയോഗിക്കാൻ അനുവദിക്കുന്നത് എക്സ്ട്രാനെറ്റിന്റെ ഉദാഹരണമായി പരിഗണിക്കാം.

9.2 ഇന്റർനെറ്റിലേക്ക് കമ്പ്യൂട്ടർ ബന്ധിപ്പിക്കുന്നു (Connection the computer to the Internet)

ഇന്ന് വളരെയധികം ജനകീയമായ ഇന്റർനെറ്റിലേക്ക് ലോകമുടനീളമുള്ള സ്ഥാപനങ്ങളും ജനങ്ങളും ചേരുന്നു എന്ന് നമുക്ക് അറിയാം. നേരത്തെ വിവരങ്ങൾ തിരയുവാനും ഈ മെയിൽ പരിശോധിക്കുവാനും മാത്രമാണ് ജനങ്ങൾ ഇന്റർനെറ്റ് ഉപയോഗിച്ചിരുന്നതെങ്കിൽ, ഇന്ന് അത് ട്രെയിൻ ടിക്കറ്റ് ബുക്കിങ്ങ്, മൊബൈൽ ഫോൺ റീചാർജ്ജ്, ഇന്റർനെറ്റ് ബാങ്കിങ്ങ് എന്നിങ്ങനെ മറ്റനവധി ആവശ്യങ്ങൾക്കായും ഇന്റർനെറ്റ് ഉപയോഗിക്കുന്നുണ്ട്. അതിനാൽ നമ്മുടെ കമ്പ്യൂട്ടറിലും മൊബൈൽ ഫോണിലും ഇന്റർനെറ്റ് ബന്ധം അനിവാര്യമാണ്.

ഒരു കമ്പ്യൂട്ടറിനെ ഇന്റർനെറ്റിലേക്ക് ബന്ധിപ്പിക്കുവാൻ ആവശ്യമായ ഹാർഡ്‌വെയറുകളും സോഫ്റ്റ്‌വെയറുകളും താഴെ കൊടുത്തിരിക്കുന്നു.

- NIC കാർഡ് സൗകര്യം ഉള്ള ഒരു കമ്പ്യൂട്ടറും TCP/IP പ്രോട്ടോകാൾ പിന്തുണയ്ക്കുന്ന ഓപ്പറേറ്റിങ് സിസ്റ്റവും
- മോഡം
- ടെലിഫോൺ കണക്ഷൻ
- ഒരു ഇന്റർനെറ്റ് സർവ്വീസ് പ്രൊവൈഡർ (ISP) നൽകിയ ഇന്റർനെറ്റ് ആക്കൗണ്ട്.
- ബ്രൗസർ, ഇ-മെയിൽ ന് വേണ്ട ക്ലയന്റ് ആപ്ലിക്കേഷൻ, ചാറ്റ് തുടങ്ങിയ സോഫ്റ്റ്‌വെയറുകൾ

ഈ കാലത്ത് ഡെസ്ക്ടോപ്പും ലാപ്ടോപ്പും മാത്രമല്ല ഇന്റർനെറ്റ് ബന്ധിപ്പിക്കാനായി നമ്മൾ ഉപയോഗിക്കുന്നത്. ഇന്റർനെറ്റ് ബ്രൗസിങ്ങിന് വേണ്ടി ജനങ്ങൾ ടാബ്, സ്മാർട്ട് ഫോൺ എന്നിവ ഉപയോഗിച്ച് തുടങ്ങിയിട്ടുണ്ട്. ഇതിൽ ചില ഉപകരണങ്ങൾ ബിൽറ്റ് ഇൻ മോഡം ഉള്ളതും എന്നാൽ മറ്റ് ചിലത് ഡോങ്കിളിൽ നിന്നോ മോഡത്തിൽ നിന്നോ വയർലെസ് രീതിയിലും ഇന്റർനെറ്റ് ഉപയോഗിക്കുന്നുണ്ട്.

9.3 വിവിധ തരം ഇന്റർനെറ്റ് ബന്ധിപ്പിക്കൽ രീതികൾ (Types of connectivity)

ഇന്ന് വെബ് പേജുകൾ കൂടുതൽ ആകർഷകമാക്കുവാൻ വേണ്ടി ചിത്രങ്ങളും മൾട്ടി മീഡിയയും ഉപയോഗപ്പെടുത്തുന്നു. ധാരാളം വെബ് സൈറ്റുകളിൽ വീഡിയോ കാണുവാനും ഡൗൺലോഡ് ചെയ്യുവാനുമുള്ള സൗകര്യം ഉണ്ട്. സി.ഡി കളോ, മറ്റ് സ്റ്റോറേജ് മീഡിയ വഴിയോ സോഫ്റ്റ്‌വെയർ വിതരണം ചെയ്യുന്നതിന് പകരം ഇന്ന് പല വിൽപനക്കാരും ഓൺലൈൻ വഴിയാണ് അവ നൽകുന്നത്. സോഫ്റ്റ്‌വെയറുകളായ വേർഡ് പ്രോസസ്സറുകൾ, സ്പ്രെഡ് ഷീറ്റുകൾ, ആന്റിവൈറസ് എന്നിവ ഓരോ കമ്പ്യൂട്ടറുകളിൽ ഇൻസ്റ്റാൾ ചെയ്യുന്നതിന് പകരം, ഓൺലൈനിൽ വാടകയ്ക്ക് ഉപയോഗിക്കാവുന്നതാണ് ഇപ്പോഴത്തെ പ്രവണത. ഈ കാര്യങ്ങളിലെല്ലാം ധാരാളം ഡാറ്റ ഓൺലൈനിലൂടെ കൈമാറ്റം ചെയ്യേണ്ടതായി വരും. അതിനാൽ ഇന്റർനെറ്റിന്റെ വേഗത അഥവാ ഡാറ്റ കൈമാറ്റ നിരക്ക് (data transfer rate) ഒരു പ്രധാനപ്പെട്ട കാര്യമാണ്. ഡാറ്റ കൈമാറ്റനിരക്ക് എന്നത് യൂണിറ്റ് സമയത്തിൽ ഉപകരണങ്ങളിലൂടെ കൈമാറ്റം ചെയ്യപ്പെടുന്ന ശരാശരി ബിറ്റുകളുടെ എണ്ണമാണ്.

1 kbps = 1000 ബിറ്റ്സ്/സെക്കന്റ്
1 Mbps = 1000 kbps
1 Gbps = 1000 Mbps

b, B യും എന്നീ യൂണിറ്റ് ചിഹ്നങ്ങൾ തമ്മിലുള്ള വ്യത്യാസങ്ങൾ.

b - ബിറ്റ് എന്നതിനെ സൂചിപ്പിക്കുന്നു.

B - ബൈറ്റ് എന്നതിനെ സൂചിപ്പിക്കുന്നു.

k, K എന്നീ യൂണിറ്റ് ചിഹ്നങ്ങൾ തമ്മിലുള്ള വ്യത്യാസം

$$k = 1000 = 10^3$$
$$K = 1024 = 2^{10}$$

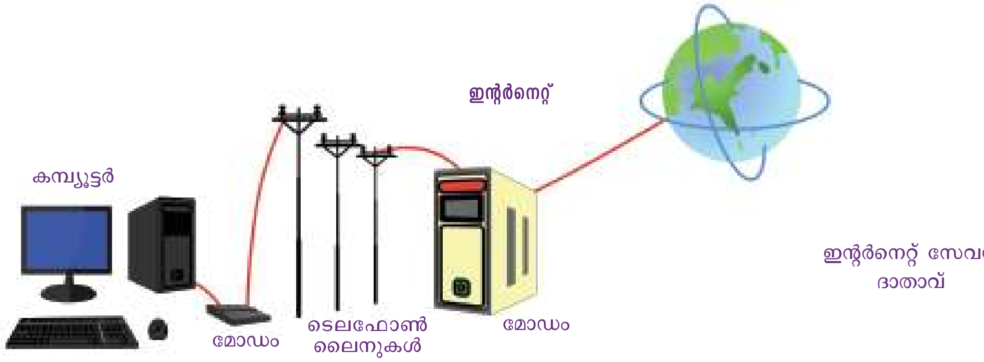
ഇവിടെ k എന്നത് ഡെസിമൽ യൂണിറ്റും K എന്നത് ബൈനറി യൂണിറ്റിലുമുള്ള അളവാണ്. എന്നാൽ മെഗാ, ജിഗാ, ടെറാ തുടങ്ങിയ യൂണിറ്റുകൾ, ഡെസിമലിനും ബൈനറിയ്ക്കും ‘M’, ‘G’, ‘T’ എന്നീ ചിഹ്നങ്ങൾ തന്നെയാണ് യഥാക്രമം ഉപയോഗിക്കുന്നത്. ഉപയോഗിക്കുന്ന സന്ദർഭത്തിന് അനുസരിച്ച് അവ വ്യത്യാസപ്പെട്ടിരിക്കും.

ഡാറ്റ കൈമാറ്റ നിരക്ക് അളക്കുന്നത് ദശക യൂണിറ്റിലും എന്നാൽ മെമ്മറി അളക്കുന്നത് ബൈനറി യൂണിറ്റിലും ആണ് എന്നത് ശ്രദ്ധിക്കുക.

ഇന്റർനെറ്റ് ആക്സസ് വേഗതയെ സ്വാധീനിക്കുന്ന പ്രധാന ഘടകം ഇന്റർനെറ്റ് ബന്ധത്തിനായി തിരഞ്ഞെടുക്കുന്ന രീതി ആണ്. ഇന്റർനെറ്റ് ബന്ധത്തിന്റെ വേഗതയും അതിൽ ഉപയോഗിച്ചിരിക്കുന്ന സാങ്കേതികവിദ്യയുടേയും അടിസ്ഥാനത്തിലാണ് ഇന്റർനെറ്റ് ബന്ധിപ്പിക്കൽ രീതി തരം തിരിച്ചിരിക്കുന്നത്. അവയെ ഡയൽ അപ് കണക്ടിവിറ്റി, വയേർഡ് ബ്രോഡ്ബാന്റ്, വയർലെസ് ബ്രോഡ്ബാന്റ് എന്നിങ്ങനെ തരം തിരിച്ചിരിക്കുന്നു. സാങ്കേതികവിദ്യയുടെ പുരോഗതി അനുസരിച്ച് ഓരോ തരം ബന്ധിപ്പിക്കൽ രീതിയുടേയും ഡാറ്റ ട്രാൻസ്ഫർ റേറ്റ് വ്യത്യാസപ്പെട്ടിരിക്കുന്നു.

9.3.1 ഡയൽ അപ് ബന്ധം (Dial-up connectivity)

ഇതിൽ പരമ്പരാഗത ടെലഫോൺ ലൈനിലൂടെ ഒരു ഡയൽ അപ് മോഡം ഉപയോഗിച്ച് ഡയൽ ചെയ്താണ് ഇന്റർനെറ്റ് സർവ്വീസ് പ്രൊവൈഡറുടെ സെർവ്വറുമായി ബന്ധിപ്പിച്ചിരിക്കുന്നത്. ഡയൽ -അപ് കണക്ടിവിറ്റി സർവീസിന്റെ ചിത്രമാണ് 12.2 കാണിച്ചിരിക്കുന്നത്. സെർവ്വറിലേക്ക് ഡയലിങ്ങിലൂടെ ബന്ധം സ്ഥാപിക്കുന്നതിനാൽ കമ്പ്യൂട്ടറിൽ നിന്നും ISP യുടെ സെർവറിലേക്ക് ബന്ധിപ്പിക്കുവാൻ കൂടുതൽ സമയം എടുക്കും. ഈ ബന്ധത്തിനായി സാധാരണ പരാമധി 56kbps വേഗതയിൽ ഡാറ്റ സംപ്രേഷണം ചെയ്യാവുന്ന 56kbps മോഡമാണ് ഉപയോഗിക്കുന്നത്. മറ്റ്



ചിത്രം 9.2 ഡയൽ അപ് ബന്ധിപ്പിക്കൽ രീതി

തരത്തിലുള്ള കണക്ഷനുകളുമായി താരതമ്യം ചെയ്യുമ്പോൾ വേഗത കുറഞ്ഞ ഈ ബന്ധത്തിന് ചെലവ് കുറവാണ്. ടെലിഫോൺ ലൈൻ ഡയൽ അപ് കണക്ഷന് വേണ്ടി മാത്രമായി മാറ്റി വയ്ക്കണം എന്നതാണ് ഇതിന്റെ മറ്റൊരു പോരായ്മ. അതായത് ഇന്റർനെറ്റ് ഉപയോഗിക്കുമ്പോൾ നമ്മൾക്ക് ടെലിഫോൺ കോളുകൾ നടത്തുവാൻ കഴിയില്ല. ഇപ്പോഴാകട്ടെ ഡയൽ-അപ് കണക്ഷനു പകരം കൂടുതൽ വേഗതയുള്ള ബ്രോഡ് ബാന്റ് ബന്ധം സ്ഥാനം പിടിച്ചിട്ടുണ്ട്.

9.3.2 വയേർഡ് ബ്രോഡ് ബാന്റ് ബന്ധം (Wired broadband connectivity)

നൂതന സാങ്കേതിക വിദ്യകൾ ഉപയോഗിച്ച് നമ്മൾക്ക് ഉയർന്ന നിരക്കിൽ ഡാറ്റാ കൈമാറുന്ന ഇന്റർനെറ്റ് ബന്ധത്തിനെയാണ് ബ്രോഡ് ബാന്റ് എന്ന പദം സൂചിപ്പിക്കുന്നത്. വയേർഡ് ബ്രോഡ്ബാന്റ് ബന്ധങ്ങൾ എപ്പോഴും ലഭ്യമാകുന്നതും ഡയൽ ചെയ്ത് ബന്ധിപ്പിക്കേണ്ട ആവശ്യം ഇല്ലാത്തതുമാണ്. ബ്രോഡ്ബാൻറ് ബന്ധങ്ങൾക്ക്, ബ്രോഡ്ബാന്റ് മോഡം ഉപയോഗിക്കണം. കൂടാതെ നമുക്ക് വയേർഡ് ഇന്റർനെറ്റ് ഉപയോഗത്തോടൊപ്പം ടെലിഫോൺ ഉപയോഗിക്കാവുന്നതാണ്. ഡയൽ-അപ് കണക്ഷനും ബ്രോഡ്ബാന്റ് കണക്ഷനും തമ്മിലുള്ള താരതമ്യം പട്ടിക 12.1 കാണിച്ചിരിക്കുന്നു.

ചിത്രം 9.3 ബ്രോഡ്മാൻ: മോഡം

ഡയൽ അപ് കണക്ഷൻ	വയേർഡ് ബ്രോഡ്ബാന്റ് കണക്ഷൻ
• കുറഞ്ഞ, വേഗത 56 Kbps വരെ	• ഉയർന്ന വേഗതയുള്ള ബന്ധം. വേഗത സാധാരണയായി 256 Kbps ൽ കൂടുതലാണ്.
• ISP യുമായി ബന്ധപ്പെടാൻ ഡയലിങ്ങ് ആവശ്യമാണ്	• എല്ലായ്പ്പോഴും ലഭ്യമാണ്
• ടെലിഫോൺ ലൈൻ ഇന്റർനെറ്റിനായി മാത്രം ഉപയോഗിക്കുന്നു.	• ഒരേ സമയം ശബ്ദവും ഇന്റർനെറ്റും ഉപയോഗിക്കാം
• ഡയൽ അപ് മോഡം ഉപയോഗിക്കുന്നു	• ബ്രോഡ്ബാന്റ് മോഡം ഉപയോഗിക്കുന്നു

പട്ടിക 9.1: ഡയൽ അപ്, വയേർഡ് ബ്രോഡ് ബാന്റ് എന്നീ ബന്ധങ്ങൾ തമ്മിലുള്ള താരതമ്യം.

ജന സമ്മിതിയുള്ള ബ്രോഡ്ബാന്റ് സാങ്കേതികവിദ്യകൾ ഇന്റഗ്രേറ്റഡ് സർവ്വീസ് ഡിജിറ്റൽ നെറ്റ്വർക്ക് (ISDN), കേബിൾ ഇന്റർനെറ്റ്, ഡിജിറ്റൽ സബ്സ്ക്രൈബർ ലൈൻ (DSL), ലീസ്ഡ് ലൈനുകൾ, ഫൈബർ ടു ഹോം എന്നിവയാണ്.

a. ഇന്റഗ്രേറ്റ്ഡ് സർവ്വീസസ് ഡിജിറ്റൽ നെറ്റ്വർക്ക് (Intergrated Services Digital Network (ISDN))

ISDN എന്നത് ശബ്ദവും ഡിജിറ്റൽ ഡാറ്റയും വഹിക്കാൻ കഴിവുള്ള ഒരു ബ്രോഡ്ബാന്റ് സേവനമാണ്. ISDN ബന്ധം ഉപയോഗിക്കുന്ന ടെലഫോൺ കമ്പനികൾ രണ്ട് ലൈനുകൾ നൽകുന്നു. ഉപഭോക്താവിന് ഒരു ലൈൻ ശബ്ദത്തിനും, അടുത്തത് ഡാറ്റക്കും അല്ലെങ്കിൽ രണ്ടും കൂടി ഡാറ്റക്കായി ഉപയോഗിക്കാം. ഈ ലൈനിലൂടെ 2 Mbps വരെ വേഗതയിൽ ഡാറ്റ കൈമാറ്റം ചെയ്യാനാകും.

b. കേബിൾ ഇന്റർനെറ്റ് (Cable Internet)

ടെലിവിഷൻ സിഗ്നലുകൾ കൈമാറുന്നതിനായി വീടുകളിൽ സ്ഥാപിച്ചിട്ടുള്ള കൊയാക്സിയൽ കേബിൾ വഴി ഇന്റർനെറ്റ് സ്വീകരിക്കുന്ന രീതിയാണ് കേബിൾ ഇന്റർനെറ്റ് ആക്സസ്. സേവന ദാതാവ്, നമ്മുടെ വീടുകളിൽ സ്ഥാപിച്ചിട്ടുള്ള ഒരു കേബിൾ മോഡം ഉപയോഗപ്പെടുത്തിയാണ് കമ്പ്യൂട്ടറും കേബിൾ ശൃംഖലയുമായി ബന്ധിപ്പിക്കുന്നത്. ഇത് ഉയർന്ന ബാന്റ് വിഡ്ത്ത് സേവനം നൽകുന്നതിനാൽ കേബിൾ ഇന്റർനെറ്റിന് 1Mbps മുതൽ 10Mbps വരെ വേഗത കാണും.

c. ഡിജിറ്റൽ സബ്സ്ക്രൈബർ ലൈൻ (Digital Subscriber Line (DSL))

സാധാരണ ടെലിഫോൺ ലൈനിലൂടെ ഇന്റർനെറ്റ് ബന്ധം ലഭ്യമാക്കുന്ന മറ്റൊരു ബ്രോഡ്ബാന്റ് സേവനമാണ് DSL. ടെലിഫോൺ ലൈനിലൂടെ ഒരേസമയം ഫോൺ വിളിക്കാനും, ഇന്റർനെറ്റ് ആശയവിനിമയത്തിനും DSL, ഉപയോക്താവിന് അനുവാദം നൽകുന്നു. ഇതിൽ നിരവധി ഉപ വിഭാഗങ്ങൾ ഉൾപ്പെടുത്തിയിട്ടുണ്ട് ഇതിൽ ഏറ്റവും സാധാരണയായത് അസിമട്രിക് ഡിജിറ്റൽ സബ്സ്ക്രൈബർ ലൈൻ Asymetric Digital Subscriber Line (ADSL)) ആണ്. ഒരു ടെലിഫോൺ ലൈനിലൂടെ ഏറ്റവും കൂടുതൽ വേഗതയിൽ വിവരങ്ങൾ സഞ്ചരിക്കുന്ന ആശയവിനിമയ സങ്കേതിക വിദ്യയാണ് ADSL. ADSL സേവനത്തിലെ ഡൗൺലോഡ് വേഗത സാധാരണയായി. 256 kbps മുതൽ 24 Mbps വരെയാണ്. ഈ ബന്ധത്തിന് ഒരു ADSL മോഡം നമ്മുടെ വീട്ടിലോ ഓഫീസിലോ ആവശ്യമാണ്. ഇന്ത്യയിൽ ലഭ്യമായ ബ്രോഡ്ബാന്റ് സേവനങ്ങളിൽ ഏറ്റവും ജനപ്രീതി ആർജ്ജിച്ചതാണ് ADSL.

d. ലീസ്ഡ് ലൈൻ (Leased Line)

ഇന്റർനെറ്റ് സേവനദാതാക്കൾ, വ്യാപാരികൾ, മറ്റ് വലിയ സ്ഥാപനങ്ങൾ തുടങ്ങിയവക്ക് ഇന്റർനെറ്റ് സേവനം നൽകുന്നതിന് മാത്രമായി ഉപയോഗിക്കുന്ന ലൈനുകളാണ് ലീസ്ഡ് ലൈൻ. ഒരു ഇന്റർനെറ്റ് ലീസ്ഡ് ലൈൻ 2Mbps മുതൽ 10Mbps വരെ വേഗത പ്രദാനം ചെയ്യുന്നതും എന്നാൽ താരതമ്യേന ചെലവ് കൂടിയതുമായ അധികമൂല്യമുള്ള ഇന്റർനെറ്റ് ബന്ധം ആകുന്നു. അതു കൊണ്ട് ലീസ്ഡ് ലൈനുകൾ വിദ്യാഭ്യാസ സ്ഥാപനങ്ങൾ പോലെയുള്ള വലിയ സ്ഥാപനങ്ങളുടെ വളപ്പിൽ മാത്രം ഉപയോഗിക്കുന്നത്.

e. ഫൈബർ ടു ദ ഹോം (Fibre To The Home (FTTH))

FTTH ഡാറ്റ പ്രസാരണത്തിനായി ഒപ്റ്റിക്കൽ ഫൈബറുകളാണ് ഉപയോഗിക്കുന്നത്. ഇന്റർ നെറ്റ് സേവനദാതാവിന്റെ (TSP) സ്ഥലം മുതൽ നമ്മുടെ വീടുവരെ ഒപ്റ്റിക്കൽ ഫൈബറുകൾ സ്ഥാപിക്കുന്നു. വീടുകളിലേക്ക് അതിവേഗതയിലുള്ള ഇന്റർനെറ്റ് സ്ഥാപിക്കുന്നതിന് ആഗോ ളതലത്തിൽ സ്വീകാര്യമായ സാങ്കേതികവിദ്യാണ് FTTH. ഒപ്റ്റിക്കൽ ഫൈബറുകൾക്ക് ഉയർന്ന ബാന്റ് വിഡ്ത്തും കുറഞ്ഞ ഡാറ്റ നഷ്ടവും ഉള്ളതിനാൽ ഒപ്റ്റിക്കൽ ഫൈബറുകൾ ഉയർന്ന വേഗതയിലുള്ള ബന്ധം നൽകുന്നു. നമ്മുടെ വീട്ടിൽ സ്ഥാപിക്കപ്പെട്ട ഒരു Network Termination Unit (NTU) FTTH മോഡം വഴി നമ്മുടെ കമ്പ്യൂട്ടറിലേക്ക് ബന്ധിപ്പിച്ചിരിക്കുന്നു.

9.3.3 വയർലെസ്സ് ബ്രോഡ്ബാന്റ് ബന്ധിപ്പിക്കൽ രീതി (Wireless broadband connectivity)

ഏറെക്കുറെ വയേർഡ് ബ്രോഡ്ബാന്റ് ബന്ധത്തിന്റെ അതേ വേഗത നൽകുന്നവയാണ് വയർലെസ്സ് ബ്രോഡ്ബാന്റ് ബന്ധിപ്പിക്കൽ രീതി. മൊബൈൽ ബ്രോഡ്ബാന്റ്, വൈ-മാക്സ്, ഉപഗ്രഹ ബ്രോഡ്ബാന്റ്, വൈ- ഫൈ എന്നിവ ജനപ്രിയമായ വയർലെസ്സ് ബ്രോഡ് ബാന്റ് രീതികളാണ്. ഇന്റർനെറ്റിലേക്ക് ബന്ധിപ്പിക്കാൻ ഉപയോഗിക്കുന്ന ചില വയർലെസ്സ് മോഡങ്ങൾ ചിത്രം 12.4ൽ കാണിക്കുന്നു.

ചിത്രം 12.4 വയർലെസ്സ് ബ്രോഡ്ബാന്റ് മോഡങ്ങൾ

a. മൊബൈൽ ബ്രോഡ്ബാന്റ് (Mobile broadband)

മൊബൈൽ ഫോൺ, USB വയർലെസ്സ് മോഡം, ടാബ് അല്ലെങ്കിൽ മറ്റ് മൊബൈൽ ഉപകര ണങ്ങൾ എന്നിവ ഉപയോഗിച്ചുള്ള വയർലെസ്സ് ഇന്റർനെറ്റ് പ്രവേശനരീതിയാണ് മൊബൈൽ ബ്രോഡ്ബാന്റ്. മൊബൈൽ ഫോണുകൾ, ടാബുകൾ, USB ഡോങ്കിളുകൾ തുടങ്ങിയവയിലെല്ലാം മോഡം സ്ഥാപിച്ചിട്ടുണ്ട്. നമ്മൾ സഞ്ചരിക്കുമ്പോൾ എവിടെ നിന്നുവേണമെങ്കിലും ഇന്റർനെറ്റ് സ്വീകരിക്കുവാൻ ഉപയോക്താവിന് മൊബൈൽ ബ്രോഡ്ബാന്റ് അനുവാദം നൽകുന്നു. ഈ സാങ്കേതിക വിദ്യ ഡാറ്റ പ്രസാരണത്തിനായി മൊബൈൽ ഫോണുകളുടെ സെല്ലുലാർ ശൃംഖല ഉപയോഗിക്കുന്നു. ഡാറ്റ പ്രസാരണത്തിന് വേണ്ട മൊബൈൽ സാങ്കേതിക വിദ്യ രണ്ടാം തലമു റയിൽ (2G) തുടങ്ങി മൂന്നാം തലമുറയിലൂടെ (3G) ഇപ്പോഴത്തെ നാലാം തലമുറയിലേക്ക് (4G) മാറിക്കഴിഞ്ഞു. മൊബൈൽ സാങ്കേതിക വിദ്യയിലെ തലമുറകളുടെ പുരോഗതിക്കനുസരിച്ച് ഡാറ്റ പ്രസാരണ വേഗത വർദ്ധിക്കുന്നു.

b. വൈ-മാക്സ് (Wi-Max)

വേൾഡ്വൈഡ് ഇന്റർ ഓപ്പെറബിലിറ്റി ഫോർ മൈക്രോവേവ് ആക്സസ് (Wi-MAX) വയേർഡ് ബ്രോഡ്ബാന്റിന് പകരം ഉപയോഗിക്കുന്നു എന്നത് കഴിഞ്ഞ അധ്യായത്തിലൂടെ നമ്മൾ പഠിച്ചതാണ്. Wi-MAX, 50 കിലോമീറ്റർ ദൂര പരിധിയിൽ വയർലെസ്സ് ഇന്റർനെറ്റ് ലഭ്യമാക്കാൻ കഴിയുന്ന ഒരു മെട്രോപൊളിറ്റൻ ഏരിയ നെറ്റ്വർക്ക് വാഗ്ദാനം ചെയ്യുന്നു. Wi-MAX ഹാൻഡ്സെറ്റുകൾ, USB ഡോങ്കിളുകൾ, Wi-MAX ഉപകരണങ്ങൾ അടങ്ങിയ ലാപ്ടോപ്പ് തുടങ്ങിയവയിൽകൂടി ബന്ധിപ്പിക്കാൻ സാധിക്കുന്നു. ഈ സാങ്കേതികവിദ്യ പരമാവധി 70 Mbps വരെ വേഗത നൽകുന്നു.

c. സാറ്റലൈറ്റ് ബ്രോഡ്ബാന്റ് (Satellite broadband)

ഉപഗ്രഹത്തിന്റെ സഹായത്തോടെ ഇന്റർനെറ്റ് ബന്ധം സാധ്യമാക്കുന്ന സാങ്കേതികവിദ്യയാണ് സാറ്റലൈറ്റ് ബ്രോഡ്ബാന്റ്. ഇതിന് ഉപയോക്താവിന്റെ സ്ഥലത്ത് ഒരു VSAT ഡിഷ് ആന്റിനയും ഒരു ട്രാൻസീവർ (ട്രാൻസ്മിറ്റർ ഉം റിസീവറും) ആവശ്യമുണ്ട്. ഉപയോക്താവിന്റെ പക്കലുള്ള മോഡം ഉപയോഗിച്ച് കമ്പ്യൂട്ടറും ട്രാൻസീവറും തമ്മിൽ ബന്ധിപ്പിക്കുന്നു. ഈ സാങ്കേതിക വിദ്യയുടെ ഡൗൺലോഡ് വേഗത 1Gbps ആണ്. ബ്രോഡ്ബാന്റ് ഇന്റർനെറ്റ് ഉപയോഗത്തിലെ ഏറ്റവും ചിലവേറിയ രീതിയാണിത്. ബാങ്കുകൾ, സ്റ്റോക്ക് എക്സ്ചേഞ്ച്, സർക്കാരുകൾ, എന്നിവ കൂടാതെ ഉൾപ്രദേശങ്ങളിലേയും ഇന്റർനെറ്റ് സ്വീകരണ മാർഗ്ഗമായി സാറ്റലൈറ്റ് ബ്രോഡ്ബാന്റ് ഉപയോഗിക്കുന്നു.

9.3.4 ഇന്റർനെറ്റ് പങ്കുവയ്ക്കൽ രീതികൾ (Internet access sharing method)

ഒരു ലോക്കൽ ഏരിയ നെറ്റ്വർക്ക്, വൈ -ഫൈ, അല്ലെങ്കിൽ ലൈ-ഫൈ നെറ്റ്വർക്ക് എന്നിവ ഉപയോഗിച്ച് ഒരു ഇന്റർനെറ്റ് ബന്ധം ധാരാളം കമ്പ്യൂട്ടറുകളിൽ പങ്കുവയ്ക്കാനാകും.

a LAN ഉപയോഗിച്ച് (Using LAN)

ഒരു LANലെ ഒരു കമ്പ്യൂട്ടറിൽ ലഭ്യമായ ഇന്റർനെറ്റ് ബന്ധം ആ നെറ്റ്വർക്കിലെ മറ്റു കമ്പ്യൂട്ടറുകൾക്ക് പങ്കുവയ്ക്കാനാകും. ഇത് ഓപ്പറേറ്റിങ്ങ് സിസ്റ്റത്തിന്റെ പ്രത്യേകത ഉപയോഗിച്ചോ അല്ലെങ്കിൽ കമ്പോളത്തിൽ ലഭ്യമായ പ്രോക്സി സെർവ്വർ സോഫ്റ്റ്വെയറിലൂടെയോ നടത്തുവാൻ കഴിയും. കേബിൾ ഉപയോഗിച്ച് കമ്പ്യൂട്ടറുകളെ നേരിട്ട് റൗട്ടറുമായി ബന്ധിപ്പിച്ചും ഇന്റർനെറ്റ് ലഭ്യമാക്കാം.

b. വൈ-ഫൈ ശൃംഖല ഉപയോഗിച്ച് (Using Wi-Fi network)

ഉന്നത വിദ്യാഭ്യാസ സ്ഥാപനങ്ങൾ, കോഫീ ഷോപ്പുകൾ, ഷോപ്പിങ്ങ് മാളുകൾ, ഹോട്ടലുകൾ എന്നിവിടങ്ങളിലെ Wi-Fi ഉപയോഗത്തെ കുറിച്ച് നാം കേട്ടിട്ടുണ്ട്. നാം അറിയുന്നത് പോലെ നമ്മുടെ വീട്ടിലും സ്കൂളിലും ഉള്ള ബ്രോഡ്ബാന്റ് മോഡമുകളിൽ വൈ-ഫൈ ഇന്റർനെറ്റ് ലഭ്യമാണ്. വൈ-ഫൈ എന്നത് ചെറിയ ദൂരത്തിലേക്കുള്ള നെറ്റ്വർക്ക് (ഇന്റർനെറ്റ്) ലഭ്യമാക്കാനുള്ള ഡാറ്റ പ്രസരണ സാങ്കേതിക വിദ്യയാണ്. നമ്മൾ ഇതിന് മുമ്പേ പ്രതിപാ

ചിത്രം 12.5: വൈഫൈ നെറ്റ്വർക്ക്.

ദിച്ച ഏതെങ്കിലും വയേർഡ് അല്ലെങ്കിൽ വയർലെസ്സ് ബ്രോഡ്ബാന്റ് പ്രവേശന രീതി ഉപയോഗിച്ച് വൈ-ഫൈ സ്ഥലത്തേക്ക് ഇന്റർനെറ്റ് ലഭ്യമാക്കാൻ കഴിയും. ഇതിനായി ഒരു വൈ-ഫൈ റൗട്ടർ അല്ലെങ്കിൽ വയർലെസ് ശൃംഖല അക്സസ് പോയിന്റ് ഉപയോഗിക്കുന്നു. ആ ആക്സസ് പോയിന്റിനെ സാധാരണയായി ഹോട്ട്സ്പോട്ട് എന്ന് വിളിക്കുകയും ഇവയുടെ പരിധി കെട്ടിടത്തിനകത്ത് 100 മീറ്ററും തുറന്ന സ്ഥലത്ത് അതിൽ കൂടുതലും ആണ്. ഈ ഹോട്ട്സ്പോട്ടിലൂടെ വൈ-ഫൈ സൗകര്യം ഉള്ള ഉപകരണങ്ങളായ ലാപ്ടോപ്പ്, ടാബ്‌ലെറ്റ്, മൊബൈൽ ഫോൺ എന്നിവയ്ക്ക് ഇന്റർനെറ്റ് സൗകര്യം ലഭ്യമാക്കാൻ കഴിയും. വയേർഡ് കണക്ഷനുകളെ അപേക്ഷിച്ച് സുരക്ഷ കുറവാണ് എന്നതാണ് ഇതിന്റെ ദോഷ വശം .

c. ലൈ-ഫൈശൃംഖല ഉപയോഗിച്ച് (Using Li-Fi network)

വൈഫൈയുടെ വേഗതയേറിയ ഓപ്റ്റിക്കൽ പതിപ്പായ ലൈഫൈ (Li-Fi). ഡാറ്റ പ്രസരണത്തിനായി സാധാരണ പ്രകാശമാണ് ഉപയോഗിക്കുന്നത്. ഈ വാർത്താ വിനിമയ മാർഗത്തിന്റെ പ്രധാന ഭാഗമായ തെളിച്ചമുള്ള LED ക്ക് (ലൈറ്റ് എമിറ്റിങ്ങ് ഡയോഡ്) ഡാറ്റ സംപ്രേഷണം ചെയ്യാനും, ഫോട്ടോ ഡയോഡിന് ഡാറ്റ സ്വീകരിക്കാനും കഴിയും. LED പ്രകാശം ഓഫും ഓണും ചെയ്ത് 0, 1 എന്നിവ ഉൾപ്പെട്ട ബൈനറി സ്ട്രിങ് നിർമിക്കും. LED യുടെ ഈ മങ്ങിക്കത്തൽ വളരെ വേഗത്തിലായതിനാൽ മനുഷ്യ നേത്രങ്ങൾക്ക് തിരിച്ചറിയുവാൻ സാധിക്കില്ല. പ്രകാശത്തിന് ഉയർന്ന ബാൻഡ്‌വിഡ്ത്ത് ആയതിനാൽ ഈ സാങ്കേതികത്വം ഉപയോഗിച്ച് 100 Mbps ൽ കൂടുതൽ ഡാറ്റാ നിരക്ക് സാധ്യമാണ്. ലൈ-ഫൈയിൽ പ്രകാശം ഉപയോഗിക്കുന്നതിനാൽ റേഡിയോ തരംഗങ്ങൾക്ക് തടസ്സം ഉണ്ടായേക്കാവുന്ന വിമാനങ്ങളിലും ആശുപത്രികളിലും ലൈ-ഫൈ ഉപയോഗിക്കാം എന്നുള്ളത് ഇതിന്റെ മറ്റൊരു നേട്ടം. വൈ-ഫൈ ഉപയോഗിക്കാൻ സാധ്യമല്ലാത്ത വെള്ളത്തിനടിയിലും ലൈ-ഫൈ ഉപയോഗിക്കാം. പ്രകാശത്തിന് ഭിത്തികൾ കടന്നുപോകാൻ സാധിക്കാത്തതിനാൽ വൈ-ഫൈയെ അപേക്ഷിച്ച് ലൈ-ഫൈ കൂടുതൽ സുരക്ഷ നൽകുന്നു. വീക്ഷണ രേഖയിലാണെങ്കിലേ ലൈ-ഫൈയുടെ പ്രവർത്തനം സാധ്യമാകൂ എന്നതാണ് ഇതിന്റെ പോരായ്മ. ഭാവിയിൽ ഈ സാങ്കേതികവിദ്യ വികസിച്ച് സാധാരണ ബൾബുകൾ വരെ ഇന്റർനെറ്റിന്റെ ഉറവിടമായി മാറിയേക്കാം.

സ്വയം വിലയിരുത്താം

1. ARPANET ന്റെ പൂർണ്ണ രൂപമാണ്....................
2. www എന്ന ആശയം നിർദേശിച്ചത് ആരാണ്?
3. ഇന്റർനെറ്റ് വിനിമയത്തിനുള്ള പ്രോട്ടോകോൾ .... ആകുന്നു.
4. "always on" എന്ന ബന്ധത്തിന് നിങ്ങൾ എന്ത് അർത്ഥം നൽകും.
5. വളരെ ചെറിയ ദൂരത്തേക്കും വയർലെസ് ഇന്റർനെറ്റ് പ്രവേശന മാർഗത്തെ .... എന്ന് വിളിക്കുന്നു.

നമുക്കു ചെയ്യാം

വിവിധ രീതിയിലുള്ള ഇന്റർനെറ്റ് ബന്ധത്തിന്റെ താരതമ്യം വ്യക്തമാക്കുന്ന ചാർട്ട് തയ്യാറാക്കുക.

9.4 ഇന്റർനെറ്റിലെ സേവനങ്ങൾ (Services on Internet)

ഇന്റർനെറ്റ് വിവിധതരം സേവനങ്ങൾ വാഗ്ദാനം ചെയ്യുന്നു. ഇന്റർനെറ്റിന്റെ സേവനങ്ങളായ www, ഇ-മെയിൽ, സേർച്ച് എൻജിനുകൾ, സാമൂഹ്യ മാധ്യമങ്ങൾ, എന്നിവ ലോകത്തിലെവിടെയും ഉപയോഗിക്കുന്നു. ഈ ഭാഗത്തിൽ നമുക്ക് ഇന്റർനെറ്റിലെ ചില സേവനങ്ങളെകുറിച്ച് ചർച്ച ചെയ്യാം.

9.4.1 വേൾഡ് വൈഡ് വെബ് (World Wide Web (WWW))

ഇന്റർനെറ്റിലൂടെ ലഭ്യമാക്കുന്ന പരസ്പരം കൂട്ടിയിണക്കിയ ഹൈപ്പർ ടെക്സ്റ്റ് ഡോക്യുമെന്റ് സംവിധാനമാണ് വേൾഡ് വൈഡ് വെബ് (WWW). ഇന്റർനെറ്റിന്റെ അടിസ്ഥാന ഘടകങ്ങൾ ഉപയോഗിച്ച് നൽകുന്ന ഒരു സേവനമാണ് WWW. പരസ്പര ബന്ധമുള്ള ലക്ഷോപലക്ഷം ക്ലയന്റുകളും സെർവറുകളും അടങ്ങുന്ന ക്ലയന്റ് -സെർവർ സംവിധാനമാണിത്. ഓരോ സെർവറിലും ഡോക്യുമെന്റുകൾ സൂക്ഷിക്കുകയും, അവ യൂണിഫോം റിസോഴ്സ് ലൊക്കേറ്ററിലെ സൂചന ഉപയോഗിച്ച് ലഭ്യമാക്കാനും കഴിയുന്നു. ഈ ഡോക്യുമെന്റുകളിൽ വാക്യങ്ങൾ, ചിത്രങ്ങൾ, വീഡിയോകൾ, വിവിധ സെർവറുകളിലെ ഡോക്യുമെന്റുകളിലേക്കുള്ള ഹൈപ്പർ ലിങ്ക്, മൾട്ടിമീഡിയ എന്നിവ ഉൾപ്പെട്ടിരിക്കുന്നു. ഒരു ഹൈപ്പർ ലിങ്ക് തിരഞ്ഞെടുക്കുന്നതിലൂടെ ആ ഡോക്യുമെന്റ് അല്ലെങ്കിൽ വെബ് പേജ്, സെർവറിൽ നിന്നും തേടി കൊണ്ടു വരുന്നു. ശൃംഖലകളിലെ ഡോക്യുമെന്റുകൾക്കുള്ളിൽ ഹൈപ്പർ ലിങ്കുകൾ സ്ഥാപിച്ചാണ് വേൾഡ് വൈഡ് വെബ് പ്രവർത്തിക്കുന്നത്. സെർവ്വറുകളിൽ സൂക്ഷിച്ചിരിക്കുന്ന ഡോക്യുമെന്റുകൾ ബ്രൗസർ എന്ന സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് ലഭ്യമാക്കാൻ ക്ലയന്റിന് കഴിയും. ഡോക്യുമെന്റുകൾ ഉചിതമായി പ്രദർശിപ്പിക്കേണ്ടത് ബ്രൗസറിന്റെ ഉത്തരവാദിത്വമാണ്.

a. ബ്രൗസർ (Browser)

വെബ് ബ്രൗസർ എന്ന സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് WWW യിൽ അടങ്ങിയ ഡോക്യുമെന്റുകളിലൂടെ കടന്നു പോകുകയും, വിവരങ്ങൾ പ്രദർശിപ്പിക്കുകയോ തിരിച്ചെടുക്കുവാനോ കഴിയും. പ്രദർശിപ്പിക്കേണ്ട ഡോക്യുമെന്റ് URL ഉപയോഗിച്ച് തിരിച്ചറിയേണ്ടതുണ്ട്. ഒരു URL ൽ അതിന്റെ DNS ചേരുന്നതോടൊപ്പം ഫയലിന്റെ പേരും അടങ്ങിയിരിക്കുന്നു. നെറ്റ്‌വർക്കിലൂടെ ഡോക്യുമെന്റുകൾ കൈമാറ്റം ചെയ്യുന്നതിനുള്ള നിയമവും ഇതിൽ പ്രതിപാദിക്കുന്നു. ഒരു വെബ് പേജ്/ ഡോക്യുമെന്ററിനുള്ളിലുള്ള വാക്യങ്ങൾ, ചിത്രങ്ങൾ, ഹൈപ്പർ ലിങ്കുകൾ, വീഡിയോകൾ, ശബ്ദങ്ങൾ, വെബ്പേജിലുള്ള പ്രോഗ്രാം ശകലങ്ങൾ എന്നിവ പ്രദർശിപ്പിക്കുവാൻ ബ്രൗസറിന് സാധിക്കുന്നു. ഒട്ടു മിക്ക www ഡോക്യുമെന്റുകളും വെബ് പേജുകൾ എന്നറിയപ്പെടുന്ന ഹൈപ്പർ ടെക്സ്റ്റ് മാർക്ക് അപ് ലാങ്ങ്വേജ് (HTML) ടാഗുകൾ ഉപയോഗിച്ച് ഉണ്ടാക്കിയതാണ്. വെബ് ബ്രൗസർ ഈ ടാഗുകളെ തർജമ ചെയ്ത് രൂപ ഘടനയിൽ മാറ്റം വരുത്തി പ്രദർശിപ്പിക്കുന്നു. ഇത് വെബ് പേജുകളിലുള്ള ഹൈപ്പർ ലിങ്ക് ഉപയോഗിച്ച് നമുക്ക് വിവിധ പേജുകളിലുള്ള സഞ്ചാരം അനുവദിക്കുന്നു. സാധാരണയായി ഉപയോഗിക്കുന്ന വെബ് ബ്രൗസറുകൾ ഗൂഗിൾ ക്രോം,

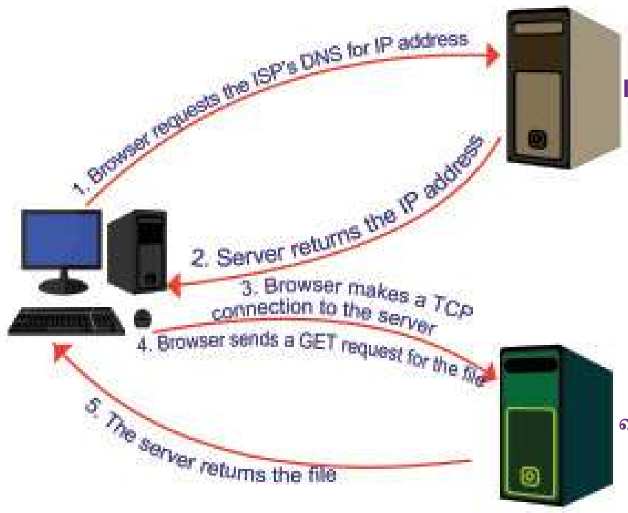
ഇന്റർനെറ്റ് എക്സ്പ്ലോറർ, മോസില്ല ഫയർ ഫോക്സ് എന്നിവയാണ്. ജനപ്രിയമായ ചില ബ്രൗസറുകളുടെ ലഘു ചിത്രങ്ങൾ ചിത്രം 12.6 ൽ കാണിച്ചിരിക്കുന്നു. ഇതിൽ ചില ബ്രൗസറുകളുടെ മൊബൈൽ പതിപ്പ് ലഭ്യമാണ്. അവ മൊബൈൽ ഓപ്പറേറ്റിങ്ങ് സിസ്റ്റത്തിൽ ഉപയോഗിക്കാം.

ഗൂഗിൾ ക്രോം | ഇന്റർനെറ്റ് എക്സ്പ്ലോറർ | മോസില്ല ഫയർഫോക്സ് | ഒപേറ | സഫാരി

ചിത്രം 9.6 : ജനപ്രിയ ബ്രൗസറുകളുടെ ലോഗോകൾ

b. വെബ് ബ്രൗസിങ്ങ് (Web browsing)

നമ്മൾ എല്ലായിപ്പോഴും വെബ് ബ്രൗസറിൽ വെബ് അഡ്രസ്സ് നൽകിയതിനുശേഷം, പിന്നീട് ഇതിലുള്ള ഹൈപ്പർ ലിങ്കുകൾ ഉപയോഗിച്ചു വെബ് പേജുകൾ സന്ദർശിക്കുന്നു. വേൾഡ് വൈഡ് വെബിലെ വെബ് പേജുകളിലൂടെയുള്ള സഞ്ചാരത്തെ വെബ് ബ്രൗസിങ്ങ് എന്ന് അറിയപ്പെടുന്നു. വെബ് ബ്രൗസിങ്ങിൽ നടക്കുന്ന പ്രധാന പ്രവർത്തനങ്ങൾ ചിത്രം 9.7 ൽ കാണിച്ചിട്ടുണ്ട്.



ചിത്രം 9.7 വെബ് ബ്രൗസിങ്ങ്

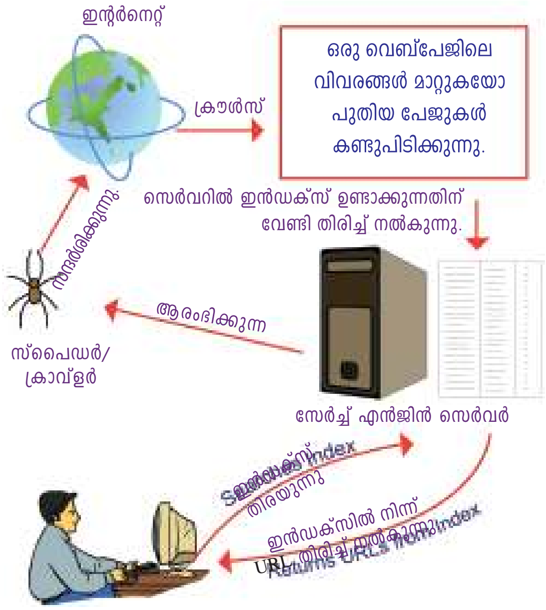
നിങ്ങൾക്ക് www. kerala.gov. in എന്ന വെബ് സൈറ്റ് സന്ദർശിക്കണം എന്ന് കരുതുക. നിങ്ങൾ എന്ത് ചെയ്യും അതിനായി അഡ്രസ് ബാറിൽ URL ടൈപ്പ് ചെയ്യുകയും Enter കീ അമർത്തുകയും ചെയ്യും. ബ്രൗസർ ഒരു വെബ് പേജ് പ്രദർശിപ്പിക്കുന്നതിനാവശ്യമായ പ്രവർത്തനങ്ങൾ താഴെ ചുരുക്കി വിവരിക്കുന്നു.

1. നൽകിയ URL ശരിയാണോ എന്ന് വെബ് ബ്രൗസർ ഉറപ്പു വരുത്തുന്നു.
2. URL ന്റെ IP അഡ്രസ് ലഭിക്കുന്നതിനായി ഉപയോക്താവിന്റെ ഇന്റർനെറ്റ് സേവന ദാതാവിന്റെ DNS സെർവറിലേക്ക് ബ്രൗസർ അഭ്യർത്ഥന അയക്കുന്നു.

3. ISP യിലെ DNS സെർവർ മറുപടിയായി IP അഡ്രസ്സ് നൽകുന്നു.
4. വെബ് ബ്രൗസർ വെബ് സെർവറിലെ IP അഡ്രസ്സിലേക്ക് TCP ബന്ധം സ്ഥാപിക്കുന്നു.
5. വെബ് ബ്രൗസർ ആവശ്യമുള്ള ഫയലിനായി വെബ് സെർവറിലേക്ക് GET അഭ്യർത്ഥന അയക്കുന്നു.
6. വെബ് സെർവർ വെബ് പേജ് തിരികെ നൽകുന്നു.
7. TCP ബന്ധം വിച്ഛേദിക്കുന്നു.
8. വെബ് പേജിന്റെ ഉള്ളടക്കങ്ങൾ മാറ്റങ്ങളോടെ പ്രദർശിപ്പിക്കുന്നു.

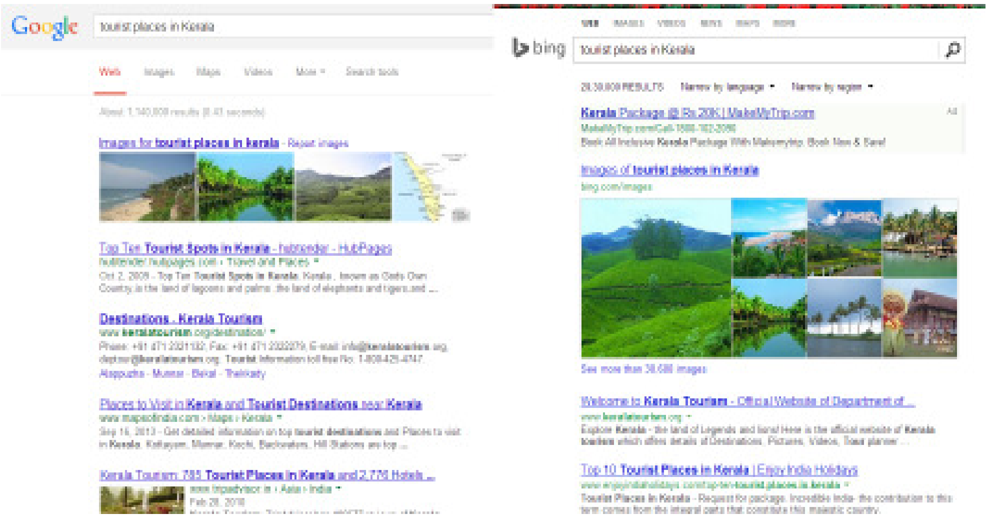
9.4.2 സേർച്ച് എൻജിനുകൾ (Search engines)

ഇന്റർനെറ്റിൽ വിവിധങ്ങളായ വിഷയങ്ങൾ ഉൾക്കൊള്ളുന്ന ലക്ഷക്കണക്കിന് വെബ് പേജുകൾ ലഭ്യമാണ്. എന്നാൽ വെബ് പേജുകളുടെ വലിയ ശേഖരത്തിൽ നിന്നും ഒരു വിഷയം തിരഞ്ഞ് കണ്ടുപിടിക്കുക എന്നത് വളരെ ശ്രമകരമാണ്. ജനങ്ങൾക്ക് ഉപകാരപ്രദമായി വേൾഡ് വൈഡ് വെബിൽ ലഭ്യമായ വിവരങ്ങൾ കണ്ടുപിടിക്കുന്നതിനായി സഹായിക്കുന്ന രീതിയിൽ തയ്യാറാക്കിയിട്ടുള്ള പ്രത്യേകതരം പ്രോഗ്രാമുകളാണ് ഇന്റർനെറ്റ് സെർച്ച് എൻജിൻ പ്രോഗ്രാമുകൾ. പ്രത്യേക വാക്കുകൾക്കായി വേൾഡ് വൈഡ് വെബിൽ ലഭ്യമായ ഡോക്യുമെന്റുകളിൽ തിരഞ്ഞ് സാദൃശ്യമുള്ള പ്രത്യേക വാക്കുകൾ ഉൾക്കൊള്ളുന്ന വെബ് പേജിന്റെ പട്ടിക സെർച്ച് എൻജിൻ പ്രോഗ്രാമുകൾ തിരിച്ചു നൽകുന്നു. നമുക്ക് ഈ വെബ് സൈറ്റിന്റെ പുറകിലുള്ള സാങ്കേതികവിദ്യയെക്കുറിച്ച് ചർച്ച ചെയ്യാം.



ചിത്രം 9.8 ഒരു സേർച്ച് എൻജിന്റെ പ്രവർത്തനം

ക്രോളേഴ്സ്, സ്പൈഡേഴ്സ്, റോബോട്ട് എന്നീ പേരുകളിലുള്ള പ്രോഗ്രാമുകൾ ഉപയോഗിക്കുന്നു. വെബ് ക്രാളറുകൾ വിവിധ വെബ് സെർവറുകളിൽ സംഭരിച്ചിരിക്കുന്ന വെബ് പേജുകൾ തിരഞ്ഞ് സാധ്യമായ കീവേഡുകൾ കണ്ടെത്തുന്നു. ഒരു ഇൻഡക്സ് രൂപപ്പെടുത്താനായി ഈ കീ വേഡുകളും അവയുടെ URL കളും അടങ്ങുന്ന പട്ടിക തയ്യാറാക്കി സെർച്ച് എൻജിന്റെ വെബ്സെർവറിൽ സംഭരിക്കുന്നു. നമ്മൾ ഒരു പ്രത്യേക വിഷയം തിരയുവാനായി സേർച്ച് എൻജിൻ വെബ് സൈറ്റ് ഉപയോഗിക്കുമ്പോൾ അത് വേൾഡ് വൈഡ് വെബിൽ തിരയില്ല. അത് വെബ് ക്രോളർ പ്രോഗ്രാമുകൾ വിഷയത്തിന് വേണ്ടി സെർച്ച് എൻജിന്റെ വെബ് സെർവറിൽ



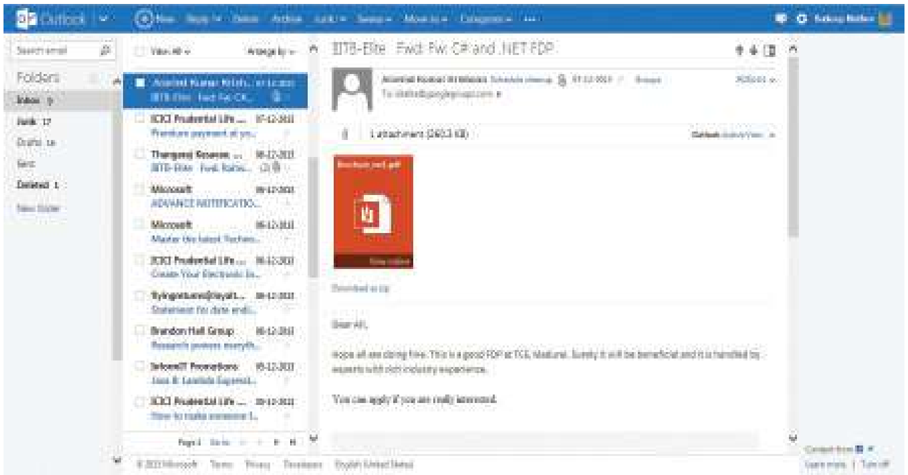
ചിത്രം 9.9 വിവിധ സേർച്ച് എൻജിനുകളിലെ തിരയൽ ഫലം

തയ്യാറാക്കിയ സൂചികയിൽ മാത്രം തിരയുന്നു. സെർച്ച് എൻജിനുകൾ സൂചികയിൽ നിന്നും പ്രത്യേക വിഷയത്തിൽ കണ്ടെത്തിയ URL കളുടെ പട്ടിക തയ്യാറാക്കി പ്രദർശിപ്പിക്കുന്നു. സെർച്ച് എൻജിന്റെ പ്രവർത്തനം ചിത്രം 9.8 ൽ കാണിച്ചിരിക്കുന്നു.

9.4.3 ഇ-മെയിൽ (E-mail)

ലോകത്തിൽ ഉള്ള ഏതൊരു വ്യക്തിയുമായും സെക്കന്റുകൾക്കുള്ളിൽ സന്ദേശങ്ങൾ കൈമാറുവാൻ ഇ-മെയിൽ നമ്മെ സഹായിക്കുന്നു. ഓരോ ദിവസവും ഇന്റർനെറ്റിലൂടെ നൂറ് കോടിയിൽ പരം സന്ദേശങ്ങളാണ് അയക്കുന്നത്. ഇന്റർനെറ്റിലെ കമ്പ്യൂട്ടറുകൾക്കിടയിൽ ഡിജിറ്റൽ സന്ദേശങ്ങൾ കൈമാറുന്ന രീതിയാണ് ഇ-മെയിൽ അല്ലെങ്കിൽ ഇലക്ട്രോണിക് മെയിൽ.

അങ്ങേയറ്റം ജനപ്രിയമായ ആശയ വിനിമയ ഉപകരണമായി ഇ-മെയിൽ മാറിയിട്ടുണ്ട്. ഇ-മെയിൽ അയച്ചു കഴിഞ്ഞാലുടൻ തന്നെ സ്വീകർത്താവിന്റെ മെയിൽ ബോക്സിൽ (Inbox) അത്



ചിത്രം 9.10 - ഒരു ഈ-മെയിൽ സന്ദേശത്തിന്റെ മാതൃക

എത്തിച്ചേരും. ടെക്സ്റ്റ് വിവരങ്ങൾക്ക് പുറമെ നമുക്ക് ഫയലുകൾ, ഡോക്യുമെന്റുകൾ, ചിത്രങ്ങൾ എന്നിവ ഇ-മെയിലിലും കൂട്ടിച്ചേർക്കാൻ കഴിയും. ഒരു ഇ-മെയിൽ സന്ദേശം ഒരേ സമയത്ത് നിരവധിപേർക്ക് അയക്കുവാൻ കഴിയും. ചിത്രം 9.10 ൽ ഒരു ഇ-മെയിൽ സന്ദേശത്തിന്റെ മാതൃക കൊടുത്തിരിക്കുന്നു. നിങ്ങളിൽ കൂടുതൽ പേർക്കും ഇ-മെയിൽ വിലാസം ഉണ്ടാകുമല്ലോ? ഒരു ഇ-മെയിൽ വിലാസത്തിന്റെ ഘടന username@domainname എന്നാകുന്നു. ഒരു ഇ-മെയിൽ വിലാസത്തിന്റെ ഉദാഹരണമാണ് scertkerala@gmail.com.

ഒരു ഇ-മെയിൽ വിലാസത്തിൽ @ ചിഹ്നത്താൽ വേർതിരിക്കപ്പെടുന്ന രണ്ട് ഭാഗങ്ങൾ ഉണ്ട്. ഇ-മെയിൽ വിലാസക്കാരനെ തിരിച്ചറിയുന്നതിനുള്ള യൂസർ നെയിം ആദ്യ ഭാഗത്തും ഇ-മെയിൽ സെർവറിന്റെ ഡൊമൈൻ പേര് gmail.com അതായത് ഇ-മെയിൽ സേവന ദാതാവിന്റെ പേര് രണ്ടാം ഭാഗത്തും ആകുന്നു. വെബ്സൈറ്റുകളായ gmail.com, hotmail.com തുടങ്ങിയവ ഇ-മെയിൽ സേവനം നൽകുന്ന വെബ്സൈറ്റുകളാണ്. ഇ-മെയിൽ അയക്കുക, സ്വീകരിക്കുക, മറുപടി നൽകുക, ക്രമീകരിക്കുക എന്നിവ. വെബ് ആപ്ലിക്കേഷനുകളുടെ ധർമ്മങ്ങളാണ്. ഇത്തരം ജനപ്രിയവും സാധാരണവുമായ സൗകര്യങ്ങളെ വെബ് മെയിൽ എന്ന് അറിയപ്പെടുന്നു. നമ്മുടെ കമ്പ്യൂട്ടറുകളിൽ സ്ഥാപിക്കപ്പെട്ട ഇ-മെയിൽ ക്ലയന്റ് സോഫ്റ്റ് വെയർ ഉപയോഗിച്ചും ഇ-മെയിൽ നോക്കാവുന്നതാണ്. ഇങ്ങനെയുള്ള സോഫ്റ്റ്വെയറിലൂടെ ഇ-മെയിൽ സേവന ദാതാവിന്റെ സെർവറിൽ നിന്നും ഇ-മെയിൽ വീണ്ടെടുക്കുകയും നമ്മുടെ ഇ-മെയിൽ വിലാസവും പാസ്വേഡും ഉപയോഗിച്ച് അത് നമ്മുടെ കമ്പ്യൂട്ടറിൽ സംഭരിക്കുകയും ചെയ്യുന്നു. ഇ-മെയിൽ അയക്കുവാനും, സ്വീകരിക്കുവാനും, ക്രമീകരിക്കുവാനും ഇ-മെയിൽ ക്ലയന്റ് സോഫ്റ്റ്വെയർ സൗകര്യം നൽകുന്നു. കമ്പ്യൂട്ടറിനും ഇന്റർനെറ്റ് ബന്ധം ഇല്ലാതിരുന്നപ്പോൾ (offline) അയച്ച സന്ദേശങ്ങൾ പ്രോഗ്രാമിലൂടെ സംഭരിക്കുകയും പിന്നീട് ഓൺലൈനിലാകുമ്പോൾ അയക്കുകയും ചെയ്യുന്നു. സാധാരണയായി സന്ദേശങ്ങൾ സ്വീകരിക്കാൻ ഇ-മെയിൽ ക്ലയന്റ് ആപ്ലിക്കേഷനുകൾ ഇന്റർനെറ്റ് മെസ്സേജ് ആക്സസ് പ്രോട്ടോകോളോ (IMAP), പോസ്റ്റ് ഓഫീസ് പ്രോട്ടോകോളോ (POP) ഉപയോഗിക്കുന്നു. മൈക്രോസോഫ്റ്റ് ഔട്ട്ലുക്ക്, മോസില്ല തണ്ടർ ബേർഡ് എന്നിവ ജനപ്രിയ ഇ-മെയിൽ ക്ലയന്റ് ആപ്ലിക്കേഷനുകൾ ആകുന്നു.

a. ഒരു ഇ-മെയിലിന്റെ വിഭാഗങ്ങൾ (Sections of an e-mail)

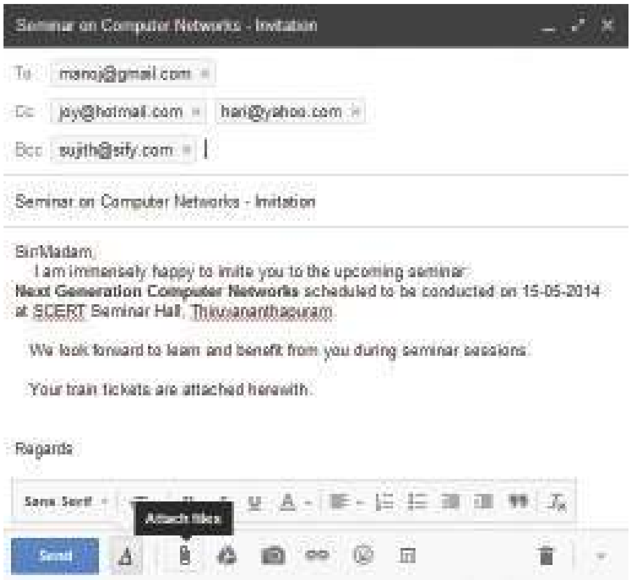
ഒരു ക്ലയന്റ് സോഫ്റ്റ്വെയർ താഴെ പറഞ്ഞിരിക്കുന്ന വിഭാഗങ്ങൾ ചേർക്കുവാനുള്ള സംവിധാനം നൽകുന്നു. ഒരു ഇ-മെയിലിന്റെ പ്രധാന വിഭാഗങ്ങൾ ചിത്രം 9.11 ൽ കാണിക്കുന്നു.

To (സ്വീകർത്താവിന്റെ വിലാസം). ഇ-മെയിൽ അയക്കേണ്ട പ്രാഥമിക സ്വീകർത്താവിന്റെ (പ്രൈമറി റെസിപ്പിയന്റ്) ഇ-മെയിൽ വിലാസം നൽകുന്ന സ്ഥലം.

Cc സന്ദേശം അയക്കേണ്ട ദ്വിതീയ സ്വീകർത്താക്ക (സെക്കന്ററി റെസിപ്പിയന്റ്)ളുടെ ഇ-മെയിൽ വിലാസം എഴുതുന്ന സ്ഥലം.

Bcc (Blind carbon copy) - സന്ദേശം സ്വീകരിക്കുന്ന ത്രിതീയ സ്വീകർത്താക്ക (ടെറിഷ്യറി റെസിപ്പിയന്റ്) ളുടെ ഇ-മെയിൽ വിലാസം എഴുതുന്ന സ്ഥലം. പ്രാഥമിക, ദ്വിതീയ സ്വീകർത്താക്കൾക്ക് സന്ദേശം ലഭിക്കുമ്പോൾ ത്രിതീയ സ്വീകർത്താക്കളുടെ വിലാസം കാണുവാൻ സാധിക്കില്ല. ഉപയോഗിക്കുന്ന ഇ-മെയിൽ സേവനങ്ങളുടെ അടിസ്ഥാനത്തിൽ ത്രിതീയ സ്വീകർത്താക്കൾക്ക് bccയിലെ അവരുടെ ഇ-മെയിൽ വിലാസം മാത്രം കാണുവാനോ അല്ലെങ്കിൽ എല്ലാ സ്വീകർത്താക്കളുടേയും ഇ-മെയിൽ വിലാസം മാത്രം കാണുവാനോ കഴിഞ്ഞേക്കാം.

ഉള്ളടക്കം (Content): നിങ്ങളുടെ സന്ദേശം ഇവിടെ നൽകുക. ഇന്ന് ഒട്ടുമിക്ക ഇ-മെയിൽ സേവന ദാതാക്കളും ആകർഷകമായ രീതിയിൽ സന്ദേശങ്ങൾ തയ്യാറാക്കുന്നതിനായി നിറം,



ചിത്രം 9.11 ഒരു ഇ - മെയിൽ സജ്ജമാക്കുന്നു

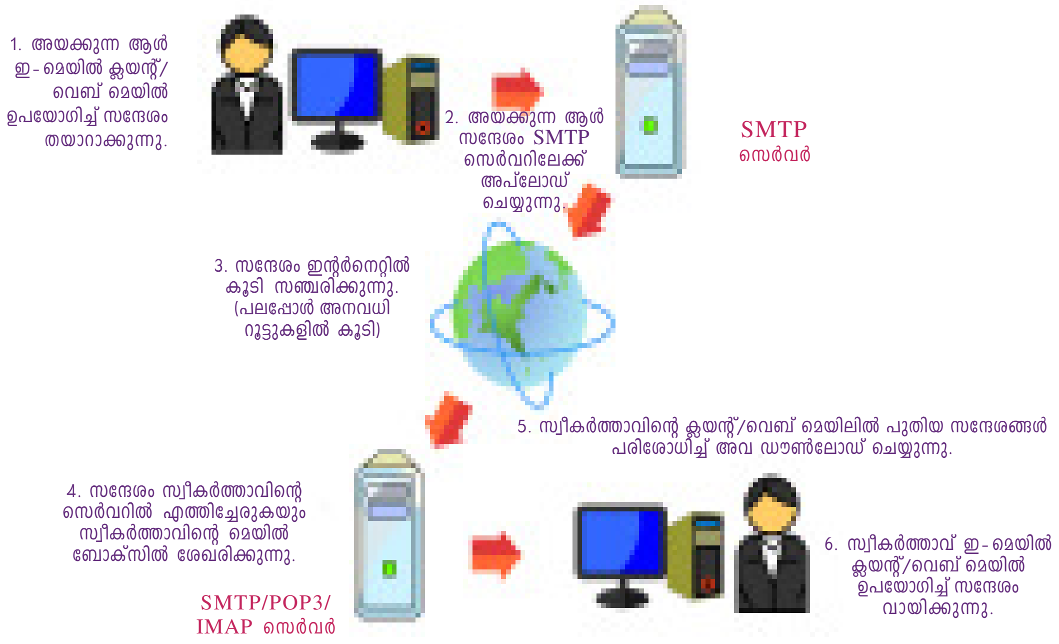
അക്ഷര ശൈലി മാറ്റുക, വലിപ്പം തുടങ്ങിയവ മാറ്റി നൽകുന്നതിന് വേണ്ട വിശേഷഗുണങ്ങൾ വാഗ്ദാനം ചെയ്യുന്നു. ഒരു ഇ-മെയിലിനോടൊപ്പം ഡോക്യുമെന്റുകൾ, ചിത്രങ്ങൾ തുടങ്ങിയവ അയക്കുവാൻ അറ്റാച്ച്മെന്റ് സൗകര്യം നമ്മെ അനുവദിക്കുന്നു. നാം സന്ദേശങ്ങൾ അയക്കുവാൻ send ബട്ടൺ ഉപയോഗിക്കുന്നു. നമുക്ക് സന്ദേശങ്ങൾ അയച്ചവർക്ക് മറുപടി നൽകുന്നതിന് 'Reply' ബട്ടൺ നമ്മെ അനുവദിക്കുന്നു. നമുക്ക് ലഭിച്ച സന്ദേശം മറ്റൊരാൾക്ക് അയക്കുന്നതിന് 'Forward' ബട്ടൺ നമ്മെ സഹായിക്കുന്നു.

വിഷയം (Subject): നിങ്ങളുടെ സംവേദനത്തിന് വേണ്ട അർത്ഥവത്തായ വിഷയം നൽകുക. പിന്നീട് നിങ്ങളുടെ ഇ-മെയിൽ തിരയുമ്പോൾ ഒരു പ്രത്യേക വ്യക്തിയുമായി നിങ്ങൾക്കുള്ള സംഭാഷണം തിരിച്ചറിയാൻ ഇതിലൂടെ സാധിക്കും.

b. ഇ-മെയിലിന്റെ പ്രവർത്തനം (Working of e-mail)

നിങ്ങളുടെ കമ്പ്യൂട്ടറിൽ നിന്നും ലോകത്തിന്റെ മറ്റേതെങ്കിലും ഭാഗത്തുള്ള സുഹൃത്തിന് ഒരു ഇ-മെയിൽ എങ്ങനെ അയച്ചുവെന്നത് നിങ്ങൾ ചിന്തിച്ചിട്ടുണ്ടോ? നിങ്ങളുടെ കമ്പ്യൂട്ടറിൽ നിന്നും വെബ് മെയിൽ അല്ലെങ്കിൽ ഇ-മെയിൽ ക്ലയന്റ് സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് ഒരു ഇ-മെയിൽ അയക്കുമ്പോൾ അത് നിങ്ങളുടെ ഇ-മെയിൽ സേവന ദാതാവിന്റെ ഇ-മെയിൽ സെർവറിൽ എത്തുന്നു. അവിടെ നിന്നും, സന്ദേശം പല വഴികളിലൂടെ ഗതിമാറ്റം ചെയ്യപ്പെട്ട് സ്വീകർത്താവിന്റെ ഇ-മെയിൽ സെർവറിൽ എത്തിച്ചേരുന്നു. പിന്നീട് സ്വീകർത്താവിന്റെ ഇ-മെയിൽ സെർവർ മെയിൽ സ്വീകർത്താവിന്റെ ബോക്സിൽ അവ സംഭരിച്ചതിനുശേഷം ഉപയോക്താവ് അത് വായിക്കുന്നതിനായി സൂക്ഷിക്കുകയും ചെയ്യുന്നു.

ഇന്റർനെറ്റിൽ കൂടിയുള്ള ഇ-മെയിൽ സംപ്രേഷണത്തിന് സിംപിൾ മെയിൽ ട്രാൻസ്ഫർ പ്രോട്ടോകോൾ (Simple Mail Transfer Protocol (SMTP) ഉപയോഗിക്കുന്നു. ഇ-മെയിലിന്റെ പ്രവർത്തനം ചിത്രം 9.12 ൽ കാണിക്കുന്നു.



ചിത്രം 9.12 ഇ-മെയിലിന്റെ പ്രവർത്തനം.

c. ഇ-മെയിൽ ഉപയോഗിക്കുന്നതിന്റെ നേട്ടങ്ങൾ

ഇ-മെയിൽ സൗകര്യം ഉപയോഗിക്കുന്നതിന്റെ നേട്ടങ്ങൾ ചുവടെ നൽകിയിരിക്കുന്നു.

- **വേഗത:** ഒരു ഇ-മെയിൽ ഭൂലോകത്തിലെ ഏത് സ്ഥലത്തേക്കും അപ്പോൾ തന്നെ അയക്കാം. ഒരു ഇ-മെയിൽ ഒരേ സമയം ധാരാളം ഉപഭോക്താക്കൾക്ക് അയക്കാം.
- **ഉപയോഗിക്കുവാൻ എളുപ്പമാണ്:** നമുക്ക് ഇ-മെയിലുകൾ അയക്കുകയും സ്വീകരിക്കുകയും ദൈനംദിന സംഭാഷണങ്ങൾ ക്രമപ്പെടുത്തി അവ നമ്മുടെ കമ്പ്യൂട്ടറിൽ എളുപ്പത്തിൽ സംഭരിക്കാനുള്ള സൗകര്യം നൽകുന്നു.
- **കൂട്ടിച്ചേർക്കാനുള്ള സൗകര്യം:** ഇ-മെയിലിനോടൊപ്പം തന്നെ ചിത്രങ്ങൾ, ഫയലുകൾ, രേഖകൾ തുടങ്ങിയവ കൂട്ടിച്ചേർക്കാം.
- **പരിസ്ഥിതി സൗഹാർദ്ദം:** പേപ്പർ ഉപയോഗിക്കാത്തതിനാൽ അവ ഉണ്ടാക്കുവാൻ ആവശ്യമായ മരങ്ങൾ മുറിക്കാതെ സംരക്ഷിക്കുവാൻ കഴിയുന്നു.
- **ഒരു ഇ-മെയിലിന്റെ മറുപടി:** ഒരു ഇ-മെയിലിന് മറുപടി നൽകുമ്പോൾ മുന്നോട്ടുള്ള ഇ-മെയിലുകൾ സൂചനയായി കൂട്ടിച്ചേർക്കുവാൻ നമുക്ക് കഴിയും. ഇത് സ്വീകർത്താവിന് വിഷയത്തെക്കുറിച്ചുള്ള ഓർമ്മ പുതുക്കുവാൻ സഹായിക്കുന്നു.
- **ചിലവ് കുറവ്:** സാധാരണ തപാലിനോടോ ഫാക്സിനോടോ താരതമ്യം ചെയ്യുമ്പോൾ ഇ-മെയിലിന് ചിലവ് കുറവാണ്.
- **ഏത് സമയത്തും എവിടെയും ലഭ്യമാണ്:** ഉപഭോക്താവിന്റെ സൗകര്യാർത്ഥം സന്ദേശങ്ങൾ വായിക്കാം. ഏതു സമയത്തും മെയിൽ ബോക്സ് ലഭ്യമാണ്.

നമ്മുടെ ദൈനം ദിന ജീവിതത്തിൽ ഇ-മെയിൽ സേവനം പ്രയോജനകരമാണെങ്കിലും അതിന്റെ വിവിധ രീതികളിലുള്ള കോട്ടങ്ങൾ ചുവടെ ചേർക്കുന്നു.

- **ഇ-മെയിലുകളിലൂടെ വൈറസുകൾ വ്യാപിച്ചേക്കാം.:** ഇ-മെയിലിന്റെ കൂടെ അയക്കുന്ന വൈറസുകൾക്ക് നമ്മുടെ കമ്പ്യൂട്ടറിന് കേടുപാടുകൾ വരുത്താം. നമ്മുടെ അറിവില്ലാതെ ഇ-മെയിൽ അഡ്രസ്സ് ബുക്കിലെ എല്ലാ ഇ-മെയിൽ വിലാസങ്ങളിലേക്കും വൈറസ് അടങ്ങിയ സന്ദേശം വ്യാപിപ്പിച്ചേക്കാം.
- **ജങ്ക് മെയിലുകൾ:** ആവശ്യമില്ലാത്ത മെയിലുകൾ പരിശോധിക്കുകയും നശിപ്പിക്കുകയും ചെയ്യുമ്പോൾ ധാരാളം സമയം നഷ്ടമാകുന്നു.

I-ter-et of Thi-gs (IoT)

നിങ്ങളുടെ വീട്ടിലെ ഫ്രിഡ്ജ്, അതിനുള്ളിലെ മുട്ടകളുടെ എണ്ണം പരിശോധിച്ചതിന് ശേഷം മുട്ട വാങ്ങാൻ നിങ്ങളുടെ മൊബൈലിലേക്ക് സന്ദേശം അയക്കുകയോ അടുത്ത കടയിൽ നിന്നും മുട്ടകൾ വാങ്ങാനുള്ള നിർദേശം നൽകുകയോ മൊബൈൽ ഫോണിലൂടെ എയർകണ്ടീഷണൽ ഓൺ അല്ലെങ്കിൽ ഓഫ് ചെയ്യുവാനോ നിങ്ങളുടെ കാറിലെ ഇന്ധനം നിറക്കുന്ന കാര്യം മൊബൈൽ ഫോണിലൂടെ ഓർമ്മപ്പെടുത്തുകയോ ചെയ്യുന്ന കാര്യം നിങ്ങൾക്ക് സങ്കൽപിക്കാനാകുമോ?

മേൽപറഞ്ഞ കാര്യങ്ങളെല്ലാം ഇന്റർനെറ്റ് ഓഫ് തിങ്ങ്സ് (IOT) ഉപയോഗിച്ചാണ് സാധ്യമാകുന്നത്. മൊബൈൽ ഫോണുകൾ, ഫ്രിഡ്ജുകൾ, കാറുകൾ, എയർകണ്ടീഷണറുകൾ, വിളക്കുകൾ തുടങ്ങിയ ഉപകരണങ്ങൾ ഇന്റർനെറ്റുമായി ബന്ധിപ്പിക്കുന്ന സങ്കൽപമാണ് IOT. ഇതിൽ ഓരോ ഉപകരണത്തിനെയും തിരിച്ചറിയുവാൻ സമാനമില്ലാത്ത IP അഡ്രസ്സ് നൽകുകയും മനുഷ്യന്റെ ഇടപെടൽ ഇല്ലാതെ ഡാറ്റ കൈമാറ്റം നടത്തുകയും ചെയ്യുന്നു. IPV6 ഉപയോഗിക്കുന്നതിനാൽ IP അഡ്രസ്സിന്റെ എണ്ണത്തിലുള്ള വർദ്ധനവാണ് ഈ സാങ്കേതികവിദ്യയുടെ അടിസ്ഥാനം. IOT ഉപയോഗിച്ച് രോഗികളുടെ രോഗാവസ്ഥ നിരീക്ഷിക്കുകയും അടിയന്തിര ഘട്ടത്തിൽ ഡോക്ടറെ അറിയിക്കുവാനും സാധിക്കുന്നു. IOT യിലൂടെ ഊർജ്ജം, ജലം എന്നിവ പാഴാക്കുന്നത് കുറയ്ക്കുവാനും അതുവഴി നമ്മുടെ തൊഴിൽ, ജീവിതശൈലികൾ മെച്ചപ്പെടുത്തുവാനും കഴിഞ്ഞേക്കാം.

9.4.4 സാമൂഹിക മാധ്യമങ്ങൾ (Social Media)

ഇന്റർനെറ്റിലെ സൗജന്യ സർവ്വവിജ്ഞാനകോശമായ വിക്കീപീഡിയ നമുക്കേവർക്കും പരിചിതമാണ്. ഫെയ്സ് ബുക്ക്, ട്വിറ്റർ തുടങ്ങിയവയിലൂടെ സാമൂഹിക പ്രശ്നങ്ങളിൽ ജനങ്ങൾ പ്രതികരിക്കുന്നത് നിങ്ങൾ കേട്ടിട്ടുണ്ടാകുമല്ലോ? ജനങ്ങൾ യൂട്യൂബ് ഉപയോഗിച്ച് വീഡിയോകൾ പങ്കിടുകയും ഉൽപന്നങ്ങൾ, വ്യാപാരം എന്നിവയെ പ്രോത്സാഹിപ്പിക്കുകയും ചെയ്യുന്ന കാര്യം നമുക്ക് അറിയാവുന്നതാണ്. സാമൂഹിക മാധ്യമങ്ങളുടെ ഭാഗമായ ഇവയെല്ലാം നമ്മുടെ ആശയ വിനിമയം, ജീവിതം, വിനോദം എന്നിവയ്ക്ക് പുതിയ മാനങ്ങൾ നൽകുന്നു. വ്യക്തികൾക്കും സമൂഹത്തിനും ഉള്ളടക്കം സൃഷ്ടിക്കുവാനും, പങ്കുവയ്ക്കുവാനും, ചർച്ച ചെയ്യുവാനും പരിഷ്കരിക്കുവാനും കഴിയുന്ന മൊബൈൽ വെബ് അധിഷ്ഠിത സാങ്കേതിക വിദ്യയാണ് സാമൂഹിക മാധ്യമങ്ങൾ.

സാമൂഹിക മാധ്യമത്തിൽ ജനങ്ങളുടെ പരസ്പര ബന്ധം സാങ്കൽപ്പിക സമൂഹങ്ങളിലും ഇന്റർനെറ്റ് ശൃംഖലകളിലുമായിരിക്കും. ഈ ഡിജിറ്റൽ സാങ്കേതികവിദ്യകൾ പൗര സമൂഹങ്ങൾ കെട്ടിപ്പടുക്കാനും പ്രവർത്തിക്കുവാനും വലിയൊരളവുവരെ സഹായിക്കുന്നു.

a. സാമൂഹിക മാധ്യമങ്ങളുടെ വർഗ്ഗീകരണം (Classification of Social Media)

ചിത്രം 9.13 ജനപ്രിയ സാമൂഹിക മാധ്യമങ്ങളുടെ ലോഗോകൾ.

ഇന്റർനെറ്റ് വേദികൾ, സമൂഹിക ബ്ലോഗുകൾ, മൈക്രോ ബ്ലോഗുകൾ, വിക്കീപീഡിയ, സാമൂഹിക ശൃംഖലകൾ, ഉള്ളടക്കം, സമൂഹങ്ങൾ എന്നിങ്ങനെ ഒട്ടനവധി സാമൂഹിക മാധ്യമങ്ങൾ ഇന്റർനെറ്റിൽ നിലനിൽക്കുന്നു. ജനപ്രിയമായ സാമൂഹിക മാധ്യമങ്ങളുടെ ലോഗോ ചിത്രം 12.13ൽ കാണിച്ചിരിക്കുന്നു. ജനപ്രിയ സാമൂഹിക മാധ്യമങ്ങളുടെ വർഗീകരണത്തെകുറിച്ച് നമുക്ക് ചർച്ച ചെയ്യാം.

1. ഇന്റർനെറ്റ് വേദികൾ (Internet forums)

സന്ദേശങ്ങൾ അയക്കുന്നതിലൂടെ ആളുകൾ തമ്മിലുള്ള സംഭാഷണത്തിൽ ഏർപ്പെടാൻ സഹായകമായ ഓൺലൈൻ വെബ്സൈറ്റാണ് ഇന്റർനെറ്റ് വേദി. വിവിധ വിഷയങ്ങൾക്കായി ഓരോ വേദികൾക്കും ഉപ വേദികൾ ഉണ്ടായേക്കാം. ഒരു വിഷയത്തിലുള്ള ചർച്ചയെ ത്രെഡ് എന്ന് വിളിക്കുന്നു. ജനങ്ങൾക്ക് ലോഗിൻ ചെയ്ത് പുതിയ ത്രെഡ് തുടങ്ങുകയോ ചർച്ച ചെയ്യുന്ന ഒരു ത്രെഡിനോട് പ്രതികരിക്കുകയോ ചെയ്യാം. ചില വേദികളിൽ അജ്ഞാത ലോഗിൻ അനുവദിക്കുന്നു. ചർച്ചകൾ പ്രോഗ്രാമിങ്ങ്, സാമൂഹിക രാഷ്ട്രീയ പ്രശ്നങ്ങൾ ഫാഷൻ തുടങ്ങിയവയെ കുറിച്ചായേക്കാം. ഈ വിധമായ ചർച്ചകൾ നമ്മുടെ പ്രശ്നങ്ങളെ കുറിച്ച് അറിയുവാനും അവയുടെ പരിഹാര മാർഗ്ഗത്തിനും സഹായിക്കുന്നു. ഉബുണ്ടു ഉപഭോക്താക്കൾക്ക് സഹായം നൽകുന്ന ജനപ്രിയ ഇന്റർനെറ്റ് വേദിയാണ് ഉബണ്ടു ഫോറം.

2. സാമൂഹിക ബ്ലോഗുകൾ (Social blogs)

വിപരീത കാലക്രമം (reverse chronological order) അനുസരിച്ച് (അതായത് ഏറ്റവും പുതിയവ ആദ്യം വരുന്ന രീതിയിൽ) എഴുത്തുകളോ കുറിപ്പുകളോ ഉൾപ്പെടുന്ന വിജ്ഞാന പ്രദമോ സംവാദ പ്രദമോ ആയ വെബ് സൈറ്റാണ് സാമൂഹിക ബ്ലോഗുകൾ (വെബ് ലോഗ്). ചില ബ്ലോഗുകളിലൂടെ ഒരു പ്രത്യേക വിഷയത്തിൽ അഭിപ്രായം രേഖപ്പെടുത്താം. എന്നാൽ ചിലത് ഓരോരുത്തരുടെ ഓൺലൈൻ ദിനക്കുറിപ്പുകളായും മറ്റു ചിലത് പ്രത്യേക വ്യക്തിയുടേയോ കമ്പനിയുടേതോ ഉത്പന്നങ്ങളുടെ ഓൺലൈൻ പരസ്യത്തിനായും ഉപയോഗിക്കുന്നു. തുടക്കത്തിൽ ബ്ലോഗുകൾ ഒരു ഉപയോക്താവിന് വേണ്ടി മാത്രം നിർമിച്ചതാണെങ്കിലും ഇപ്പോൾ ഒന്നിലധികം എഴുത്തുകാർ ഒരുമിച്ച് എഴുതുന്ന ബ്ലോഗുകളുമുണ്ട്. ജനപ്രീതിയാർജ്ജിച്ച ബ്ലോഗിങ്ങ് സൗകര്യം തരുന്ന വെബ്സൈറ്റുകളാണ് blogger.com, wordpress.com എന്നിവ.

3. മൈക്രോ ബ്ലോഗുകൾ (Micro blogs)

ഉപയോക്താക്കൾക്ക് ഹ്രസ്വ വാക്യങ്ങൾ, വ്യക്തിഗത ചിത്രങ്ങൾ അല്ലെങ്കിൽ വീഡിയോ ലിങ്കുകൾ എന്നിവ കൈമാറാൻ മൈക്രോ ബ്ലോഗുകൾ ഉപയോഗിക്കുന്നു. ആളുകൾക്ക് അവരുടെ ചുറ്റുപാടിൽ നിരീക്ഷിക്കുന്ന കാര്യങ്ങളും ഏതെങ്കിലും സംഭവങ്ങളെക്കുറിച്ചുള്ള വിവരങ്ങളും വൈവിധ്യമാർന്ന മേഖലകളിൽ നിന്നുള്ള അഭിപ്രായങ്ങളും പങ്കിടുവാൻ ജനം മൈക്രോ ബ്ലോഗ് ഉപയോഗിക്കുന്നു. നൈസർഗ്ഗീകമായ വാർത്താവിനിമയ സമ്പ്രദായം വാഗ്ദാനം ചെയ്യുന്ന മൈക്രോബ്ലോഗിങ്ങ് പൊതു ജനാഭിപ്രായത്തിനെ സ്വാധീനിക്കാൻ കഴിയും. twitter.com ഒരു ജനപ്രിയ മൈക്രോ ബ്ലോഗിങ്ങ് സൈറ്റ് ആകുന്നു.

4. വിക്കികൾ (Wikis)

ഏതൊരാൾക്കും ഒരു വെബ് പേജിൽ വിവരങ്ങൾ കൂട്ടിച്ചേർക്കാനും നിലവിലുള്ള വിവരങ്ങളിൽ മാറ്റം വരുത്തുവാനുമുള്ള സ്വാതന്ത്ര്യം നൽകുകയും സാമൂഹിക വസ്തുതകൾ, കൈകാര്യം ചെയ്യുകയും ചെയ്യുന്ന ഒരു സംവിധാനമാണ് വിക്കികൾ. വിക്കികളിൽ ഉപയോക്താക്കൾ വരുത്തുന്ന തിരുത്തലുകൾ മറ്റുള്ളവർ സൂക്ഷ്മ നിരീക്ഷണം നടത്തുന്നതിനാൽ തെറ്റായ വിവരങ്ങൾ പരസ്യങ്ങൾ തുടങ്ങിയവ അപ്പോൾ തന്നെ വിക്കികളിൽ നിന്നും നീക്കം ചെയ്യുന്നു. സ്വതന്ത്ര ഓൺലൈൻ വിജ്ഞാനകോശമായ വിക്കിപീഡിയയുടെ (wikipedia.org) മലയാള ഭാഷ പതിപ്പാണ് (ml.wikipedia.org). വെബിലെ ഏറ്റവും കൂടുതൽ ജനപ്രിയമായ സ്വതന്ത്ര ഓൺലൈൻ സർവ്വ വിജ്ഞാനകോശമാണ് wikipedia.org.

ഏതൊരാൾക്കും വിവരങ്ങൾ കൂട്ടിച്ചേർക്കാനും തിരുത്തുവാനും കഴിയുന്ന ഓൺലൈൻ വിജ്ഞാനകോശമാണ് വിക്കിപീഡിയ. 2001 ജനുവരി 15 ന് ജിമ്മി വെയിൽസ്, ലാറി ഡാങ്കർ എന്നിവർ വിക്കിയുടെ സാങ്കേതികത്വവും സങ്കൽപവും ഉപയോഗിച്ചാണ് വിക്കിപീഡിയ തുടങ്ങിയത്. വിക്കിപീഡിയയിൽ 300 ഭാഷകളിലായി വിക്കിപീഡിയയിൽ ഏകദേശം 300 കോടി ലേഖനങ്ങൾ അടങ്ങിയിരിക്കുന്നു. ഇന്റർനെറ്റിൽ ഏറ്റവും കൂടുതൽ ആളുകൾ സന്ദർശിച്ച വിക്കിപീഡിയയുടെ ആംഗലേയ ഭാഷ പതിപ്പിൽ തന്നെ 44 ലക്ഷത്തോളം ലേഖനങ്ങൾ ഉണ്ട്. വിവിധ വിഷയങ്ങളിലുള്ള ലേഖനങ്ങൾ വളരെ വിപുലവും വ്യക്തവുമായി ഇതിൽ നൽകുന്നു. ഇതിലെ ഓരോ ലേഖനങ്ങളിൽ നിന്നും വിക്കിപീഡിയയിലേക്കോ മറ്റ് പുറമെയുള്ള വിവരങ്ങളിലേക്കോ ലിങ്കുകൾ ലഭ്യമാണ്. എല്ലാ ഉപയോക്താക്കൾക്കും ഇതിൽ വിവരങ്ങൾ കൂട്ടിച്ചേർക്കുകയോ മാറ്റം വരുത്തുകയോ ചെയ്യാവുന്നതിനാൽ വിവരങ്ങളുടെ ഗുണനിലവാരം ലേഖന കർത്താവിനെ ആശ്രയിച്ചിരിക്കും. വിക്കിപീഡിയയുടെ മലയാള പതിപ്പ് www.ml.wikipedia.org യിൽ ലഭ്യമാണ്.

5. സാമൂഹ്യശൃംഖല (Social networks)

വ്യക്തിഗതമായ വെബുപേജുകൾ നിർമിക്കാൻ ആളുകളെ അനുവദിക്കുകയും അവ ഉപയോഗിച്ച് സുഹൃത്തുക്കളുമായി സംവേദനം നടത്തുവാനോ ആശയങ്ങൾ പങ്കുവയ്ക്കാൻ സഹായിക്കുന്നവയാണ് സാമൂഹ്യ ശൃംഖല സൈറ്റുകൾ. ഇവ ഉപയോഗിച്ച് ലേഖനങ്ങൾ, ചിത്രങ്ങൾ, വീഡിയോകൾ തുടങ്ങിയവ പങ്കിടുകയും അവയെ പറ്റി അഭിപ്രായ പ്രകടനം നടത്തുവാനും കഴിയും. ഈ വെബ് സൈറ്റുകളിൽ പൊതുവായ വിഷയങ്ങൾക്കോ അല്ലെങ്കിൽ പ്രൊഫഷണൽ ശൃംഖലയിലേത് പോലെ ഒരു പ്രത്യേക വിഷയമോ ചർച്ച ചെയ്യാൻ ഉപയോഗിക്കുന്നു. facebook.com, linkedin.com തുടങ്ങിയവ ജനപ്രീതിയാർജ്ജിച്ച സാമൂഹ്യ ശൃംഖലകളാകുന്നു.

6. ഉള്ളടക്ക സമൂഹങ്ങൾ (Content communities)

ചിത്രങ്ങൾ, വീഡിയോകൾ തുടങ്ങിയ ഉള്ളടക്കങ്ങൾ ക്രമപ്പെടുത്തുകയും പങ്കിടുകയും ചെയ്യുന്ന വെബ്സൈറ്റുകളാണ് ഉള്ളടക്ക സമൂഹങ്ങൾ ജന പ്രീതിയാർജ്ജിച്ച youtube.com വീഡിയോകളും flicker.com ചിത്രങ്ങളും പങ്കിടുന്നു.

ഇന്നത്തെ മിക്ക സാമൂഹിക മാധ്യമ വെബ്സൈറ്റുകളും ഒന്നിലധികം സേവനങ്ങൾ ലഭ്യമാകുന്നു. അതായത് സാമൂഹിക ശൃംഖല, മൈക്രോബ്ലോഗിങ്ങ് ഇന്റർനെറ്റ് വേദി ബ്ലോഗിങ്ങ് എന്നീ സാമൂഹിക മാധ്യമങ്ങൾ സാമൂഹ്യ സ്വാധീനം ചെലുത്തുന്നവയാണെന്ന് പഠനങ്ങളിലൂടെ വെളിപ്പെടുത്തിയിട്ടുണ്ട്.

b. സാമൂഹിക മാധ്യമങ്ങളുടെ നേട്ടങ്ങൾ (Advantages of social media)

- **ആളുകളെ ഒരുമിച്ചു ചേർക്കുന്നു:** നീണ്ടകാത്തിരിപ്പിന് ശേഷം ബാല്യകാല സുഹൃത്തുക്കളെ കണ്ടെത്തുവാനും പുതിയ സൗഹൃദവലയം സൃഷ്ടിക്കുവാനും സാമൂഹിക മാധ്യമങ്ങളിലൂടെ കഴിയുന്നു.
- **പരിപാടികളുടെ ആസൂത്രണവും ക്രമീകരണം:** ചില പരിപാടികളിൽ പങ്കെടുക്കുവാനും, സംഘടിപ്പിക്കുവാനും ഉപയോക്താക്കളെ സഹായിക്കുന്നു.
- **വാണിജ്യ പ്രചരണം:** വാണിജ്യ ഇടപാടുകാരുമായി ബന്ധപ്പെടുക, വിപണന പ്രചാരം, മതിപ്പ് നിലനിർത്തുക തുടങ്ങിയ കാര്യങ്ങളിലൂടെ വാണിജ്യ അവസരങ്ങൾ സാമൂഹിക മാധ്യമങ്ങൾ ഒരുക്കുന്നു.
- **സാമൂഹിക നൈപുണ്യം:** ആളുകൾക്ക് ചില പ്രത്യേക പ്രശ്നങ്ങളിലൂടെ അഭിപ്രായം രേഖപ്പെടുത്തൽ കാഴ്ചപാടുകൾ പ്രതിഫലിപ്പിക്കുവാനും അതു വഴി സാമൂഹിക മാറ്റത്തിന്റെ പ്രതിനിധിയായിത്തീരാൻ ഇവ സഹായകരമാണ്.
- **പൊതുവായ അഭിപ്രായ രൂപീകരണം:** അവഗണിക്കപ്പെട്ട പ്രധാന വിഷയങ്ങൾക്ക് കൂടുതൽ പ്രാമുഖ്യം നൽകുകയും അതു വഴി കൂടുതൽ ആളുകളിലേക്കും മറ്റ് മുഖ്യധാര മാധ്യമങ്ങളിലേക്കും ഇവയെ എത്തിച്ച് പൊതു ജനാഭിപ്രായം ഏകീകരിക്കാൻ കഴിയുന്നു. (ഉദാഹരണം ഹാഷ് ടാഗുകൾ)

c. സാമൂഹിക മാധ്യമങ്ങൾ ഉപയോഗിക്കുന്നതിലെ പരിമിതികൾ (Limitations in use of social media)

- **സ്വകാര്യതയിലേക്കുള്ള കടന്ന് കയറ്റം :** ആൾക്കാരുടെ വ്യക്തിഗത വിവരങ്ങൾ നിയമവിരുദ്ധ പ്രവർത്തനങ്ങൾക്ക് ഉപയോഗിച്ചേക്കാം. വ്യക്തിപര വിവരങ്ങളായ ഇ-മെയിൽ വിലാസം, പേര്, സ്ഥലം, പ്രായം തുടങ്ങിയവ ഓൺലൈൻ കുറ്റകൃത്യങ്ങൾക്ക് ഉപയോഗിക്കുന്നു.
- **ആസക്തി (അഡിക്ഷൻ):** ഈ വെബ്സൈറ്റുകളിലുള്ള ആസക്തി നമ്മുടെ വിലയേറിയ സമയം നഷ്ടപ്പെടുത്തുന്നു. നമ്മുടെ മാനസ്സികാവസ്ഥയെ പ്രതികൂലമായി ബാധിച്ച് വിഷാദത്തിനും പിരിമുറുക്കം സൃഷ്ടിക്കുന്നതിനും കാരണമാകുന്നു. ഇത് തൊഴിൽ സ്ഥാപനങ്ങളിലെ ജീവനക്കാരുടെ ഉത്പാദനക്ഷമത കുറക്കുകയും വിദ്യാർത്ഥികളുടെ ഏകാഗ്രത നഷ്ടപ്പെടുത്തുകയും പഠനത്തിലുള്ള അവരുടെ താൽപര്യത്തെ കുറക്കുകയും ചെയ്യും.
- **അപവാദം പരത്തുന്നു:** സാമൂഹിക മാധ്യമങ്ങളിലൂടെ വളരെ വേഗം വിവരങ്ങൾ പ്രചരിക്കുന്നു. ഇവയിലൂടെ തെറ്റായ വാർത്തകളും അപൂർണ്ണ വിവരങ്ങളും വളരെ വേഗതയിൽ പ്രചരിച്ച് പ്രശ്നങ്ങൾ കൂടുതൽ വഷളാക്കുന്നു.

നമുക്കു ചെയ്യാം

- *വിവിധങ്ങളായ സാമൂഹിക മാധ്യമങ്ങളുടെ വെബ്സൈറ്റും അവയുടെ ഉപയോഗങ്ങളും വ്യക്തമാക്കുന്ന ചാർട്ട് തയ്യാറാക്കുക.*
- *നിങ്ങളുടെ ക്ലാസിനെക്കുറിച്ച് ഒരു ബ്ലോഗ് നിർമിക്കുകയും അതിൽ കല, കായികം, പഠന പ്രവർത്തനങ്ങളായ പരീക്ഷ, എന്നിവയിലെ നേട്ടങ്ങൾ കൂട്ടിച്ചേർക്കുക.*
- *ഏറ്റവും കൂടുതൽ പേർ ഉപയോഗിക്കുന്ന വെബ് ബ്രൗസർ ഏതെന്ന് കണ്ടുപിടിക്കാനുള്ള സർവ്വേ നടത്തുക. ഇതിലൂടെ ശേഖരിച്ച വിവരങ്ങൾ ചിത്രീകരിച്ച് കാണിക്കുക.*

d. സാമൂഹിക മാധ്യമങ്ങളിലെ ഇടപെടലുകൾ – നല്ല ശീലങ്ങൾ (Social media interaction Best practices)

- ഇ-മെയിൽ വിലാസം, ഫോൺ നമ്പർ, വിലാസം ചിത്രങ്ങൾ വീഡിയോകൾ തുടങ്ങിയവ അനാവശ്യമായി സാമൂഹിക മാധ്യമങ്ങളിൽ അപ്‌ലോഡ് ചെയ്യരുത്.
- ഉപയോഗസമയം ചുരുക്കി ഇത്തരം വെബ് സൈറ്റുകൾ ഉപയോഗിച്ചാൽ സമയ നഷ്ടം കുറയ്ക്കാം.
- സാമൂഹിക മാധ്യമങ്ങളായ വിക്കികൾ, ബ്ലോഗുകൾ തുടങ്ങിയവയിലൂടെ ചിത്രങ്ങളും വീഡിയോകളും ഏതൊരാൾക്കും ഉപയോഗിക്കാവുന്നതിനാൽ, അയക്കുന്നതിന് മുമ്പ് അവയുടെ പ്രസക്തിയെപറ്റി ബോധവാനാകുക. അയച്ചതിന് ശേഷം അവയെ ഓർത്ത് പരിതപിക്കേണ്ടി വരരുത്.
- സാമൂഹിക മാധ്യമങ്ങളിലെ മൂന്ന് അടിസ്ഥാന സ്വകാര്യ തലങ്ങൾ. (Privacy level) സ്വകാര്യം (private) പൊതുജനം (Public) സുഹൃത്തുക്കൾ (Friends) എന്നിവയാണ്. ഇവ ഉപയോഗിച്ച് നിങ്ങളുടെ ലേഖനങ്ങളും ചിത്രങ്ങളും ആർക്കൊക്കെ കാണുവാനും പങ്കിടാനും സാധിക്കും എന്ന് കൃത്യമായി സജ്ജീകരിക്കുക.

സ്വയം വിലയിരുത്താം

1. ഒരു ഇ-മെയിൽ വിലാസത്തിന് ഉദാഹരണം നൽകുക.
2. താഴെ നൽകുന്നവയിൽ നിന്നും സെർച്ച് എൻജിൻ അല്ലാത്തത് തിരഞ്ഞെടുക്കും.
   (a) ഗൂഗിൾ (b) ബിങ്ങ് (c) ഫേസ് ബുക്ക് (d) ആസ്ക് (ask)
3. എന്താണ് ബ്ലോഗ്?
4. ഇന്റർനെറ്റിലൂടെ ഇ-മെയിൽ പ്രസാരണത്തിന് ഉപയോഗിക്കുന്ന പ്രോട്ടോകോളിന്റെ പേര് എഴുതുക.
5. ഇന്റർനെറ്റിലൂടെ ലഭ്യമാകുന്ന ഏതെങ്കിലും രണ്ട് സേവനങ്ങളുടെ പേര് എഴുതുക.
6. വെബിലെ ഓരോ ഡോക്യുമെന്റും ____________ ഉപയോഗിച്ച് പരാമർശിക്കുന്നു.

9.5 സൈബർ സുരക്ഷ (Cyber security)

ഇന്ന് ആളുകൾ ഇന്റർനെറ്റ് ഉപയോഗിക്കുന്നത് വ്യക്തിപരവും, വളരെ രഹസ്യാത്മക വിവരങ്ങൾ കൈമാറാനും, പണം അടക്കുവാനും, ബാങ്കുകളിലെ ധനകാര്യ ഇടപാടുകൾ നടത്തുന്നതും കമ്പ്യൂട്ടർ ശൃംഖലകൾ ഉപയോഗിച്ചാണ്. റെയിൽവേ ടിക്കറ്റുകൾ നൽകുവാനും, ട്രെയിനുകളുടെ സമയക്രമം ലഭ്യമാക്കാനും മുതലായവക്കായി റെയിൽവെ അവരുടെ സ്വന്തം ശൃംഖല ഉപയോഗിക്കുന്നു. ഒരു ചെറിയ ഇടവേളയിലേക്ക് പോലും ഈ ശൃംഖലകൾ ലഭ്യമാകാതിരുന്നാൽ ഉണ്ടാകാവുന്ന ധന നഷ്ടത്തെയും മറ്റ് പ്രശ്നങ്ങളെയും കുറിച്ച് നിങ്ങൾക്ക് സങ്കൽപ്പിക്കുവാൻ കഴിയുമോ?

പ്രധാപ്പെട്ട വിവരങ്ങൾ നഷ്ടപ്പെട്ടേക്കാനും സ്വകാര്യതക്ക് ഭംഗം വരാവുന്നതുകൊണ്ടും കമ്പ്യൂട്ടർ ശൃംഖലയുടെ സുരക്ഷ നിർണ്ണായകമാകുന്നു. നെറ്റ്വർക്കിനെ ആക്രമിക്കുന്നതു വഴി ജോലിയോ വ്യാപാരമോ മണിക്കൂറുകളോ ദിവസങ്ങളോളമോ തടസ്സപ്പെട്ടേക്കാം. ഇന്റർനെറ്റിന്റെ വരവോട് കൂടി ആളുകൾ അത് പരസ്പരം ആശയ വിനിമയങ്ങൾക്കും വ്യാപാരത്തിനുള്ള ഉപകരണമാക്കിയതിനാൽ സുരക്ഷ അത്യാവശ്യമായി.

ഏതൊരു സ്ഥാപനവും നുഴഞ്ഞു കയറ്റവും മറ്റ് ആക്രമണവും തടയുന്നതിനായി കമ്പ്യൂട്ടർ ശൃംഖലാ നിരീക്ഷണം നടത്തണം. ഇവിടെ നമ്മൾ ഒരു കമ്പ്യൂട്ടർ ശൃംഖലയെ ബാധിക്കാവുന്ന പൊതു ഭീഷണികളെ പറ്റി ചർച്ച ചെയ്യുന്നു.

9.5.1 കമ്പ്യൂട്ടർ വൈറസ് (Computer virus)

ഒരു പ്രോഗ്രാം അതിനെ തന്നെ മറ്റൊരു പ്രോഗ്രാമുമായി കൂട്ടിച്ചേർക്കുകയോ, നമ്മൾ അറിയാതെ തന്നെ അതിനെ മറ്റ് കമ്പ്യൂട്ടറുകളിലേക്ക് വ്യാപിപ്പിക്കുകയോ, കമ്പ്യൂട്ടറുകളുടെ സാധാരണ പ്രവർത്തനങ്ങളിൽ ഇടപെടുകയോ ചെയ്യുന്ന പ്രോഗ്രാമുകളാണ് കമ്പ്യൂട്ടർ വൈറസ്. ഒരു വൈറസ്സിന് നമ്മുടെ കമ്പ്യൂട്ടറിലെ വിവരങ്ങൾ തിരുത്തുവാനോ നശിപ്പിക്കുവാനോ അവ തന്നെ തനിപകർപ്പുകൾ ഉണ്ടാക്കി മറ്റ് കമ്പ്യൂട്ടറുകളിലേക്ക് വ്യാപനം ചെയ്യുവാനോ ഹാർഡ് ഡിസ്കിലെ വിവരങ്ങൾ പൂർണ്ണമായി നശിപ്പിക്കുവാനോ സാധിക്കും. മിക്കവാറും വൈറസുകളും എക്സിക്യൂട്ടബിൾ ഫയലിലേക്ക് കൂട്ടിച്ചേർക്കപ്പെടുന്നു. ഒരു കമ്പ്യൂട്ടറിൽ വൈറസ് ഉണ്ടായിരുന്നാലും അത് തുറക്കുകയോ പ്രവർത്തിപ്പിക്കുമ്പോഴോ മാത്രമേ അത് കമ്പ്യൂട്ടറിനെ ബാധിക്കുന്നുള്ളു. ഒരു കമ്പ്യൂട്ടറിൽ നിന്നും മറ്റൊന്നിലേക്ക് പോർട്ടബിൾ സ്റ്റോറേജ് മീഡിയ (പെൻ ഡ്രൈവ്, എക്സ്റ്റേണൽ ഹാർഡ് ഡിസ്ക് തുടങ്ങിയവ) ഉപയോഗിച്ച് ഫയൽ കൈമാറ്റം ചെയ്യുമ്പോഴോ, ഇ-മെയിൽ അറ്റാച്ച്മെന്റ്, ഫയൽ പങ്കുവയ്ക്കൽ എന്നിവയിലൂടെയോ ആണ് വൈറസ് വ്യാപിക്കുന്നത്. വൈറസുകൾ ഇന്റർനെറ്റിൽ ധാരാളം പ്രശ്നങ്ങൾ സൃഷ്ടിക്കുകയും കോടികളുടെ നഷ്ടം ഉണ്ടാക്കുകയും ചെയ്യുന്നു.

9.5.2 വേം (Worm)

മറ്റ് കമ്പ്യൂട്ടറിലേക്ക് വ്യാപിക്കാവുന്ന തരത്തിൽ സ്വയം തനിപകർപ്പുകൾ സൃഷ്ടിക്കുന്ന അപകടകാരിയായ സോഫ്റ്റ്വെയറാണ് കമ്പ്യൂട്ടർ വേം എന്ന് അറിയപ്പെടുന്നത്. ഇത്തരത്തിൽപ്പെട്ടവ, വൈറസുകളെ അപേക്ഷിച്ച് മറ്റ് പ്രോഗ്രാമുകളുമായി കൂടിച്ചേരാതെ ഒരു കമ്പ്യൂട്ടറിൽ നിന്നും മറ്റുള്ളവയിലേക്ക് സ്വയം വ്യാപനം ചെയ്യുന്നു. ഇവ കമ്പ്യൂട്ടർ ശൃംഖലയുടെ ബാൻഡ്വിഡ്ത്ത് അപഹരിച്ച് വിവര കൈമാറ്റത്തിന്റെ വേഗത കുറയ്ക്കുന്നു. ഒരു ഇ-മെയിൽ അഡ്രസ് ബുക്കിലെ എല്ലാ ഇ-മെയിൽ വിലാസത്തിലേക്കും വേംമിന്റെ ഒരു പകർപ്പ് അയക്കുവാൻ കഴിയും. അങ്ങനെ ഓരോ സ്വീകർത്താക്കളുടേയും ഇ-മെയിൽ അഡ്രസ്സ് ബുക്കിലെ ഇ-മെയിലേക്ക് പകർപ്പ് അയയ്ക്കുന്ന പ്രക്രിയ തുടർന്നു കൊണ്ടേയിരിക്കുന്നു.

ഐ ലവ് യു വേം

2000 - ൽ കമ്പ്യൂട്ടറുകളെ ബാധിച്ച ഈ വേം അതിലെ ഒട്ടുമിക്ക ഫയലുകളും മാറ്റി എഴുതി. ഉപയോക്താക്കൾക്ക് I LOVE YOU എന്ന വിഷയത്തോടെ ലഭ്യമായ ഇ - മെയിലിൽ LOVE-LETTER-FOR-YOU.TXT.vbs എന്ന ഫയൽ അറ്റാച്ച്മെന്റായി നൽകി. ഇതിൽ ക്ലിക്കു ചെയ്യുമ്പോൾ വേം കമ്പ്യൂട്ടറിനെ ബാധിക്കുകയും ഫയലുകൾ നഷ്ടപ്പെടുകയും ചെയ്യുന്നു.

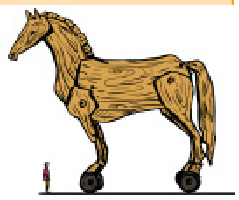
9.5.3 ട്രോജൻ ഹോഴ്സ് (Trojan horse)

നമുക്ക് ഉപകാരപ്രദമായ സോഫ്റ്റ്വെയറാണ് എന്ന തോന്നൽ ഉണ്ടാക്കുകയും എന്നാൽ യാഥാർത്ഥത്തിൽ നമ്മുടെ കമ്പ്യൂട്ടറുകളിൽ ഇൻസ്റ്റാൾ ചെയ്യുകയും എക്സിക്യൂട്ട് ചെയ്യുമ്പോൾ നഷ്ടം വരുത്തുന്നവയുമാണ് ട്രോജൻ ഹോഴ്സ്. ട്രോജൻ ഹോഴ്സ് ഉപയോക്താക്കളെ തന്ത്രപരമായാണ് അവരുടെ കമ്പ്യൂട്ടറുകളിൽ ഇത്തരം സോഫ്റ്റ്വെയർ സംഭരിക്കുകയോ പ്രവർത്തിപ്പിക്കുകയയോ ചെയ്യിപ്പിക്കുന്നത്. ഒരു കമ്പ്യൂട്ടറിൽ ട്രോജൻ ഹോർസ് ബാധിച്ചു കഴിഞ്ഞാൽ അവ നമ്മുടെ കമ്പ്യൂട്ടറിൽ സംഭരിച്ചിരിക്കുന്ന ഫയലുകൾ നശിപ്പിച്ച് വിവരങ്ങൾ നഷ്ടപ്പെടുത്തുന്നു. ചില ട്രോജനുകൾ കമ്പ്യൂട്ടറിന്റെ പിൻവാതിലിലൂടെ അപകടകാരികളായ ഉപയോക്താക്കൾക്ക് പ്രവേശന മാർഗം തുറന്നു കൊടുക്കുകയും അതിലൂടെ രഹസ്യ വിവരങ്ങളോ സ്വകാര്യ വിവരങ്ങളോ ശൃംഖലയിലൂടെ ചോർത്തുവാൻ സാധിക്കുന്നു. എന്നാൽ വൈറസ്, വേം എന്നിവയുമായി താരതമ്യം ചെയ്യുമ്പോൾ ട്രോജനുകൾ സ്വയം പെരുകുകയോ ബാധിച്ച ഫയലുകളുടെ പകർപ്പ് ഉണ്ടാക്കുകയോ ചെയ്യുന്നില്ല.

Ie0199.exe Troja-

ഇന്റർനെറ്റ് എക്സ്പ്ലോറർ സൗജന്യമായി നവീകരിക്കാം എന്ന വാഗ്ദാനത്തോടെ Ie_0199.exe എന്ന ഫയൽ ഉൾപ്പെടുത്തിയ ഒരു സന്ദേശം ഇ-മെയിൽ ഉപയോക്താക്കൾക്ക് ലഭിച്ചു. ഉപയോക്താക്കളോട് ഈ പ്രോഗ്രാം ഡൗൺലോഡ് ചെയ്ത് ഇൻസ്റ്റാൾ ചെയ്യുവാൻ ഈ ഇ-മെയിൽ സന്ദേശത്തിലൂടെ നിർദ്ദേശിക്കപ്പെട്ടു. ഈ നിർദ്ദേശം പാലിച്ച എല്ലാ ഉപയോക്താക്കളുടേയും ഫയലുകളിൽ ഇത് ബാധിക്കുകയും ചെയ്തു.

ട്രോജൻ യുദ്ധം (Trojan war)



ഗ്രീക്ക് പുരാണത്തിൽ, ഗ്രീക്ക് രാജകുമാരിയായ ഹെലനെ ട്രോയി പട്ടണത്തിലെ രാജകുമാരൻ അപഹരിച്ച് കൊണ്ടുപോയതിന് ശേഷം ഗ്രീക്കുകാരും ട്രോയി പട്ടണവും തമ്മിൽ നടന്ന യുദ്ധമാണ് ട്രോജൻ യുദ്ധമായി അറിയപ്പെടുന്നത്. നീണ്ട പത്ത് വർഷക്കാലം ട്രോയി പട്ടണവുമായി ഗ്രീക്കുകാർ യുദ്ധത്തിൽ ഏർപ്പെട്ടു. യുദ്ധത്തിൽ മനം മടുത്ത ഗ്രീക്ക് പട്ടാളക്കാർ അവരവരുടെ വീടുകളിലേക്ക് മടങ്ങുവാനുള്ള ആഗ്രഹം പ്രകടിപ്പിച്ചു. പിന്നീട് ഗ്രീക്ക് ദേവതയായ 'അഥീന' യുദ്ധം അവസാനിപ്പിക്കാനുള്ള ആശയം ഗ്രീക്കുകാർക്ക് നൽകി. ഇതിനുവേണ്ടി അവർ പൊള്ളയായ ഒരു മരക്കുതിരയെ നിർമിക്കുകയും അതിൽ പട്ടാളക്കാരെ നിറയ്ക്കുകയും ചെയ്തു. അതിനുശേഷം ട്രോജൻകാർക്കുള്ള സമ്മാനമെന്ന രീതിയിൽ ഉപേക്ഷിക്കുകയും ചെയ്തു. ബാക്കിയുള്ള ഗ്രീക്ക് പട്ടാളക്കാർ അവരുടെ താവളം ഉപേക്ഷിക്കുന്നു എന്ന തോന്നൽ ഉളവാക്കി പിന്മാറുകയും ചെയ്തു. ഇതോടെ ട്രോജൻകാർ യുദ്ധത്തിൽ വിജയിച്ചു എന്ന് കരുതുകയും ഗ്രീക്കുകാരുടെ മരകുതിരയെ ട്രോയി പട്ടണത്തിലേക്ക് വലിച്ച് കൊണ്ടു പോയി വിജയാഘോഷം ആരംഭിച്ചു. രാത്രിയിൽ എല്ലാവരും ഗാഢനിദ്രയിലാർന്ന സമയത്ത് ഗ്രീക്ക് പട്ടാളക്കാർ മരകുതിരയുടെ ഉള്ളിൽ നിന്ന് പുറത്തിറങ്ങി ഉറങ്ങിക്കിടന്ന ട്രോജൻ പട്ടാളക്കാരെ വധിക്കുകയും രാജകുമാരി ഹെലനെ രക്ഷിക്കുകയും ചെയ്തു.

9.5.4 സ്പാമുകൾ (Spams)

ചിത്രം 9.14 ഈ - മെയിൽ മെനുവിലെ സ്പാമുകളുടെ ശേഖരണം.

ഇ-മെയിൽ ഉപയോക്താക്കളുടെ സമ്മതമില്ലാതെ ഒരു ഉൽപ്പന്നത്തിന്റെയോ സേവനത്തിന്റെയോ പ്രചരണത്തിന് വേണ്ടി എല്ലാവർക്കും അയക്കപ്പെടുന്ന സന്ദേശങ്ങളാണ് സ്പാമുകൾ അല്ലെങ്കിൽ ജങ്ക് മെയിലുകൾ. ഇത്തരം ഇ-മെയിലുകൾ അയക്കുന്നവർ ചാറ്റ് റൂമുകൾ, വെബ്സൈറ്റുകൾ. ഉപയോക്താക്കളുടെ പട്ടികകൾ, വാർത്താക്കൂട്ടങ്ങൾ തുടങ്ങിയവയിൽ നിന്ന് ആണ് ഇ-മെയിൽ വിലാസങ്ങൾ ശേഖരിക്കുന്നത്. സ്പാമുകളിലെ ലിങ്കുകളിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ ചില വൈറസുകൾ ഉൾക്കൊള്ളുന്ന വെബ്സൈറ്റുകൾ ഉപയോക്താക്കൾക്ക് അയച്ച് തന്നേക്കാം. ഇന്ന് ഒട്ടുമിക്ക ഇ-മെയിൽ സേവനദാതാക്കളും യഥാർത്ഥ ഇ-മെയിലുകളേയും സ്പാമുകളേയും തമ്മിൽ വേർതിരിക്കുന്ന ഇ-മെയിൽ ഫിൽറ്ററുകൾ ചിത്രം 9.14 ൽ സൂചിപ്പിച്ചിരിക്കുന്നതുപോലെ ലഭ്യമാക്കുന്നു.

9.5.5 ഹാക്കിങ്ങ് (Hacking)

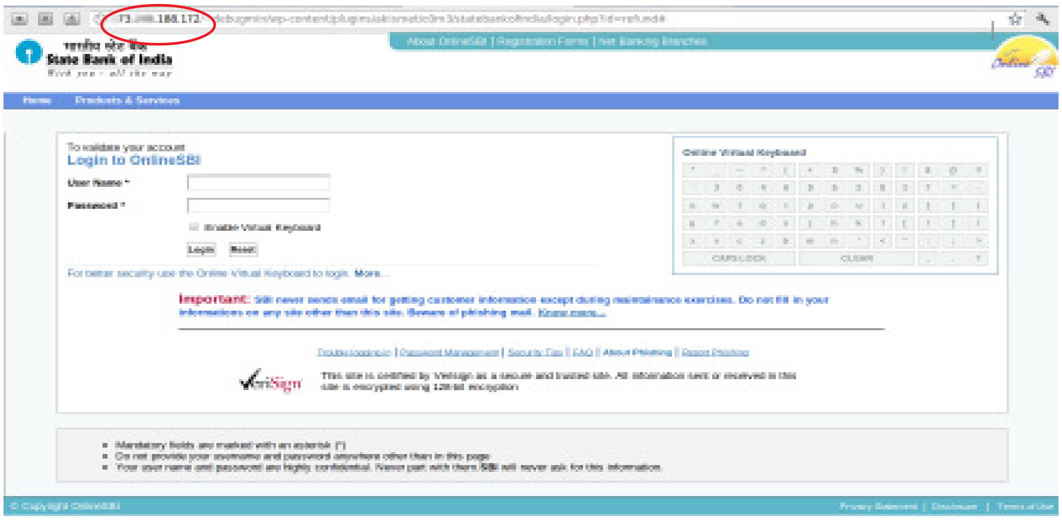
കമ്പ്യൂട്ടർ ശൃംഖലകളുടേയും അവയുമായി ബന്ധിപ്പിച്ച കമ്പ്യൂട്ടറുകളുടേയും സ്വാഭാവിക പ്രവർത്തനങ്ങൾക്ക് വിഘാതം സൃഷ്ടിക്കുന്ന സാങ്കേതിക പ്രവർത്തനമാണ് ഹാക്കിങ്ങ്. കമ്പ്യൂട്ടർ സുരക്ഷ വിദഗ്ധരും, കമ്പ്യൂട്ടർ കുറ്റവാളികളും ഹാക്കിങ്ങ് നടത്താറുണ്ട്. കമ്പ്യൂട്ടർ ശൃംഖലയുടെ സുരക്ഷ പരിശോധിക്കാനും പോരായ്മകൾ കണ്ടെത്തുവാനുമാണ് കമ്പ്യൂട്ടർ വിദഗ്ധർ ഹാക്കിങ്ങ് നടത്തുന്നത്. ഇങ്ങനെയുള്ള കമ്പ്യൂട്ടർ വിദഗ്ധരെ വൈറ്റ് ഹാറ്റ്സ് എന്നും ഇത്തരം ഹാക്കിങ്ങിനെ എത്തിക്കൽ ഹാക്കിങ്ങ് എന്നും വിളിക്കുന്നു.

കമ്പ്യൂട്ടർ കുറ്റവാളികൾ സുരക്ഷിത ശൃംഖലകൾ തകർക്കുന്നത് വിവരങ്ങൾ നശിപ്പിക്കുവാനോ ഉത്തരവാദിത്ത്വപ്പെട്ടവർക്ക് ശൃംഖല ഉപയോഗിക്കുന്നത് തടയുന്നതിന് വേണ്ടിയോ ആണ്. വളരെ രഹസ്യമായ വിവരങ്ങൾ മോഷ്ടിക്കുവാനോ ഫയലുകൾ നശിപ്പിക്കുവാനോ ആണ് ഇത്തരക്കാരുടെ ഉദ്ദേശം. ഇത്തരം കുറ്റവാളികളെ ബ്ലാക്ക് ഹാറ്റ്സ് എന്ന് വിളിക്കുന്നു.

വൈറ്റ്, ബ്ലാക്ക് ഹാക്കേഴ്സിന് ഇടയിൽ വരുന്നവരാണ്. ഗ്രേ ഹാറ്റ് ഹാക്കേഴ്സ് ഇവർ സദുദ്ദേശത്തോടെയും കൂടുതൽ സുരക്ഷയ്ക്കും വേണ്ടി നിയമവിരുദ്ധമായി പ്രവർത്തിച്ച് ശൃംഖലകളുടെ ദോഷ പൂർണ്ണത തിരിച്ചറിയുന്നു.

9.5.6 ഫിഷിങ്ങ് (Phishing)

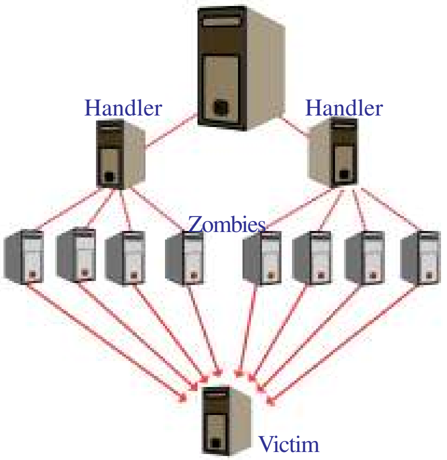
ഓൺലൈനിലൂടെ നടക്കുന്ന ഒരു തരം വ്യക്തി വിവരമോഷണമാണ് ഫിഷിങ്ങ്. പ്രമുഖ ബാങ്കുകളുടേയും ധനകാര്യ സ്ഥാപനങ്ങളുടേയും വെബ്സൈറ്റുകളോട് സാമ്യം തോന്നത്തക്ക രീതിയിൽ URL ഉം ഹോം പേജും നിർമ്മിച്ച് അതിലൂടെ ഇടപാടുകാരുടെ യൂസർ നെയിം പാസ് വേർഡ്, ക്രെഡിറ്റ് കാർഡുകളുടെ വിശദാംശങ്ങൾ തുടങ്ങിയവ കൈക്കലാക്കാൻ ശ്രമിക്കുന്നതാണ് ഫിഷിങ്ങ്. ഇത്തരം തെറ്റിദ്ധരിപ്പിക്കുന്ന വെബ് പേജ് നിർമാണത്തെ സ്പൂഫിങ്ങ് (Spoofing) എന്ന് അറിയപ്പെടുന്നു. ആളുകളുടെ ഇത്തരം വ്യാജമായ വെബ്സൈറ്റുകളിലേക്ക് ഇ-മെയിൽ വഴി വശീകരിച്ച് അവരുടെ യൂസർ നെയിം, പാസ് വേർഡ്, ക്രെഡിറ്റ് കാർഡ് വിവരങ്ങൾ എന്നിവ മോഷ്ടിക്കുകയും അതുപയോഗിച്ച് പണം തട്ടിയെടുക്കുകയും ചെയ്യുന്നു. ഇക്കാലത്ത് ഇന്റർനെറ്റിലെ ഏറ്റവും വലിയ സാമ്പത്തിക ഭീഷണിയാണ് ഫിഷിങ്ങ്. 9.15 ചിത്രത്തിലെ URL പരിശോധിച്ചാൽ അത് ഒരു ഫിഷിങ്ങ് വെബ്സൈറ്റാണെന്ന് മനസ്സിലാകും.



ചിത്രം 9.15: ഒരു ഫിഷിങ്ങ് വെബ്സൈറ്റ്

9.5.7 ഡിനൈൽ ഓഫ് സർവ്വീസ് ആക്രമണം (Denial of Service attack (DoS))

DoS അറ്റാക്ക് സാധാരണയായി വെബ് സെർവറുകളെ ആണ് ലക്ഷ്യമിടുന്നത്. ഈ ആക്രമണത്തിൽ സെർവറുകൾ പ്രവർത്തന രഹിതമാകുന്നു. ഇതിന ഡിനൈൽ ഓഫ് സർവ്വീസ് അറ്റാക്ക് എന്നു പറയുന്നു. ഇതിലൂടെ വെബ് സെർവറിന്റെ പതിവ് ഉപഭോക്താക്കൾക്കുള്ള സേവനം തടസ്സപ്പെടുത്തുന്നു. ഇതിന് ഒരു കമ്പ്യൂട്ടർ മാത്രം ഉപയോഗിച്ചാണെങ്കിൽ ഡിനൈൽ ഓഫ് സർവീസ് അറ്റാക്ക് എന്നും, മറിച്ച് ഒന്നിൽ കൂടുതൽ കമ്പ്യൂട്ടറുകളാണ് ഉപയോഗിക്കുന്നെങ്കിൽ ഡിസ്ട്രിബ്യൂട്ടഡ് ഡിനൈൽ ഓഫ് സർവീസ് അറ്റാക്ക് എന്നും വിളിക്കുന്നു. നമ്മൾ ബ്രൗസർ സോഫ്റ്റ്‌വെയറിലൂടെ വെബ്സൈറ്റ് അഡ്രസ് നൽകി വെബ് സെർവറിൽ നിന്നും ആ പേജ് ലഭിക്കുവാൻ അപേക്ഷിക്കുന്നു. DoS ആക്രമണത്തിലൂടെ ഇത്തരം ധാരാളം അപേക്ഷ വെബ് സെർവറിൽ എത്തുകയും സെർവറിന് ഇതിന്റെ ഭാരം താങ്ങുവാൻ സാധിക്കാതെ പ്രവർത്തനം നിർത്തുന്നു. ഒരു കമ്പ്യൂട്ടർ ഉപയോഗിച്ചുള്ള DoS ആക്രമണത്തിലൂടെ ശൃംഖലകളുടെ പ്രവർത്തന വേഗത കുറക്കുകയും ചെയ്യുന്നു. DoS ആക്രമണം ബാധിക്കപ്പെട്ട സോംബീസ് (Zombies) എന്ന പേരിൽ അറിയപ്പെടുന്ന കമ്പ്യൂട്ടറുകൾ ഒരുമിച്ച് പ്രവർത്തിച്ച് ധാരാളം തെറ്റായ സന്ദേശങ്ങളും അപേക്ഷകളും സെർവറിന് നൽകുന്നു. ഇങ്ങനെ സെർവറിലേക്കുള്ള ഡാറ്റയുടെ പോക്കുവരവ് വർദ്ധിപ്പിച്ച് ഉപഭോക്താക്കൾക്ക് സേവനം ലഭ്യമാക്കാതിരിക്കുന്നു. അങ്ങനെ ആ കമ്പ്യൂട്ടർ പ്രവർത്തനരഹിതമാക്കി വീണ്ടും പ്രവർത്തിപ്പിക്കാൻ (restart ചെയ്യാൻ) നിർബന്ധമാവുകയും കുറച്ച് സമയത്തേക്ക് ഉപഭോക്താക്കൾക്ക് സെർവറിന്റെ സേവനം ലഭ്യമാകാതിരിക്കുന്നു.



ചിത്രം 9.16: Distributed Denial of Service (DDoS) ആക്രമണങ്ങൾ

ഇത്തരം ആക്രമണത്തിലൂടെ കുറച്ചുനേരത്തേക്ക് സെർവറിന്റെ പ്രവർത്തനം ലഭ്യമല്ലെങ്കിലും വൈറസ് ആക്രമണവുമായി താരതമ്യം ചെയ്യുമ്പോൾ ഫയലുകൾക്ക് നഷ്ടം സംഭവിക്കുന്നില്ല.

9.5.8 മാൻ ഇൻ ദി മിഡിൽ അറ്റാക്ക് (Man-in-the-Middle attacks)

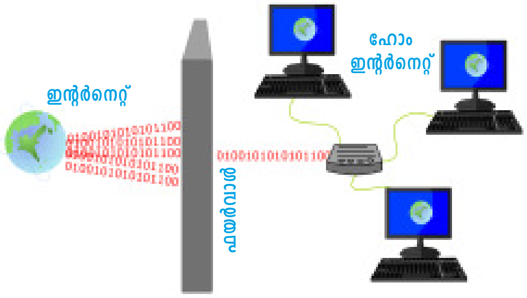
മാൻ ഇൻ ദി മിഡിൽ അറ്റാക്ക് (Man-in-the-middle attacks) ആക്രമണങ്ങളിൽ അക്രമകാരി, പ്രേക്ഷകന്റെയും സ്വീകർത്താവിന്റെയും ഇടയിലുള്ള ഇലക്ട്രോണിക് സന്ദേശങ്ങളുടെ വഴി തടഞ്ഞ് പിടിച്ചെടുക്കുകയും അവയിൽ മാറ്റങ്ങൾ വരുത്തുകയോ കൂട്ടിച്ചേർക്കുകയോ ചെയ്ത് സന്ദേശം പ്രസാരണം ചെയ്യുന്നു. പ്രേഷകൻ സന്ദേശങ്ങൾക്ക് വേണ്ട സുരക്ഷ ഒരുക്കാതെ പ്രസരണം ചെയ്യുന്നതിനാൽ ആക്രമകാരിക്ക് ശൃംഖലയുടെ ദോഷ പൂർണ്ണത ചൂഷണം ചെയ്യുവാനാകും. ശൃംഖലാ പ്രസരണം തടസ്സമില്ലാതെ നടക്കുന്നതിനാൽ പ്രേക്ഷകനും സ്വീകർത്താവിനും സന്ദേശങ്ങളിലുള്ള രൂപ മാറ്റം മനസ്സിലാക്കാൻ സാധിക്കാതെ വരുന്നു. ഇത്തരം കമ്പ്യൂട്ടറിലൂടെ ഓൺലൈൻ വ്യവഹാരങ്ങൾ നടത്തുമ്പോൾ ഇടയിലുള്ള ആൾ നമ്മുടെ ബാങ്ക് അക്കൗണ്ട് നമ്പർ, പാസ്സ്‌വേർഡ് എന്നിവ പിടിച്ചെടുത്ത് പണം അപഹരിച്ച് ധനനഷ്ടത്തിലേക്ക് നയിച്ചേക്കാം. രഹസ്യ കോഡുകൾ ഉൾക്കൊള്ളുന്ന HTTPS (HTTP സെക്യൂർ), SFTP (സെക്യൂർ FTP) തുടങ്ങിയവ ഉപയോഗിക്കുമ്പോൾ അതിക്രമിച്ചു കടക്കുന്ന ആൾക്ക് സന്ദേശങ്ങളിൽ മാറ്റം വരുത്താൻ കഴിയില്ല.

9.6 ശൃംഖലാക്രമണം തടയൽ (Preventing network attacks)

ഇന്റർനെറ്റിലൂടെ വിവരങ്ങൾ ലഭ്യമാവുകയും കൈമാറ്റപ്പെടുകയും ചെയ്യുന്നിടത്തോളം കാലം കമ്പ്യൂട്ടറുകൾക്കും നെറ്റ്‌വർക്കുകൾക്കും ഉള്ള ഭീഷണി മുഖ്യ പ്രശ്നമായി നില നിൽക്കും. ഇത്തരം ആക്രമണങ്ങൾ നേരിടുന്നതിന് പലവിധം പ്രതിരോധ മാർഗ്ഗങ്ങളും, കണ്ടെത്തൽ സംവിധാനങ്ങളും രൂപപ്പെടുത്തിയിട്ടുണ്ട്.

9.6.1 ഫയർവാൾ (Firewall)

ഒരു സ്ഥാപനത്തിലെ കമ്പ്യൂട്ടർ ശൃംഖലക്ക് സുരക്ഷ നൽകുന്ന കമ്പ്യൂട്ടർ ഹാർഡ്‌വെയറും സോഫ്റ്റ്‌വെയറും ഉൾപ്പെടുന്ന ഒരു സംവിധാനമാണ് ഫയർവാൾ. ഫയർവാൾ കമ്പ്യൂട്ടർ ശൃംഖലയിലേക്ക് വിനാശകരമായ വിവരങ്ങൾ പ്രവേശിക്കുന്നത് തടയുന്ന ചിത്രം 9.17 ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 9.17 ഫയർവാൾ

സാന്റ് ബോക്സിങ്

വൈറസ് ബാധിച്ചു എന്ന് സംശയിക്കുന്ന പ്രോഗ്രാമുകളെ പ്രവർത്തിപ്പിക്കുവാൻ ഉപയോഗിക്കുന്ന സാങ്കേതികവിദ്യയാണ് സാന്റ് ബോക്സിങ്. സാന്റ് ബോക്സിങ്ങിലൂടെ ഇത്തരം പ്രോഗ്രാമുകൾ ഒരു പ്രത്യേക മെമ്മറി ഭാഗത്ത് പ്രവർത്തിപ്പിക്കുന്നതിനാൽ നമ്മുടെ ഓപ്പറേറ്റിങ് സിസ്റ്റത്തിന് നാശം വരുത്തുവാൻ സാധിക്കുന്നില്ല.

9.6.2 ആന്റി വൈറസ് സ്കാനറുകൾ (Antivirus Scanners)

വൈറസുകൾ, വേം, ട്രാജൻ ഹോഴ്സ് എന്നിവ വിനാശകരമായ സോഫ്റ്റ്‌വെയറുകൾക്ക് ഉദാഹരണങ്ങളാണ്. ആന്റി വൈറസ് ഉപകരണങ്ങൾ ഉപയോഗിച്ച് ഇവയെ കണ്ടുപിടിക്കുകയും

സിസ്റ്റത്തിൽ നിന്നും നീക്കുകയും ചെയ്യുന്നു.. ആന്റിവൈറസ് സോഫ്റ്റ്‌വെയറുകൾ കമ്പ്യൂട്ടർ സിസ്റ്റത്തിൽ നിന്നും അറിയപ്പെടുന്ന വൈറസുകളെ കണ്ടുപിടിച്ച് നീക്കം ചെയ്യുന്നു. അറിയപ്പെടുന്ന വൈറസുകളുടേയും മാൽവെയറുകളുടേയും വിശദാംശങ്ങൾ അടങ്ങിയ വൈറസ് നിർവചന ഫയലുകൾ ആന്റി വൈറസ് സ്കാനറുകൾ ഉപയോഗിക്കുന്നു. ഒരു ആന്റി വൈറസ് പ്രോഗ്രാം ഫയലുകൾ പരിശോധിക്കുമ്പോൾ, മാൽവെയറോ വൈറസോ അതിൽ കണ്ടെത്തിയാലുടൻ തന്നെ അതിന്റെ പ്രവർത്തനം നിറുത്തി ആ ഫയലിനെ ക്വാറൻറ്റൈനിലേക്ക് മാറ്റുന്നു. വൈറസ് ബാധിച്ച ഫയലുകൾ സൂക്ഷിക്കുന്ന പ്രത്യേക സ്ഥലമാണ് ക്വാറൻറ്റൈൻ. പിന്നീട് ഈ ഫയലിലെ വൈറസിനെ നീക്കം ചെയ്യുകയോ ഫയലിനെ നശിപ്പിക്കുകയോ ചെയ്യാവുന്നതാണ്. ആന്റി വൈറസ് സോഫ്റ്റ്‌വെയറിന്റെ ഫലപ്രദമായ ഉപയോഗത്തിന് വൈറസ് നിർവചന ഫയലുകൾ പതിവായി ഓൺ ലൈനിലൂടെ നവീകരിക്കേണ്ടതുണ്ട്.

9.6.3 കൂക്കീസ് (Cookies)

നമ്മൾ ബ്രൗസർ ഉപയോഗിച്ച് ഒരു വെബ്സൈറ്റ് സന്ദർശിക്കുമ്പോൾ സൃഷ്ടിക്കപ്പെടുന്ന ചെറിയ ടെക്സ്റ്റ് ഫയലുകളാണ് കുക്കീസ്. വെബ്സെറ്റിലെ നമ്മളുടെ യൂസർ നെയിം, പാസ്സ്‌വേർഡ്, ഇ-മെയിൽ മുതലായവ വിവരങ്ങൾ ഇവ സൂക്ഷിക്കുന്നു. ബ്രൗസറുകൾ ഈ വിവരങ്ങൾ കമ്പ്യൂട്ടറിലെ കുക്കീസ് ഫോൾഡറിൽ സൂക്ഷിക്കുന്നു. സമയ ലാഭത്തിനും ഫലപ്രദവുമായ ബ്രൗസിങ്ങിനും വേണ്ടിയാണ് വെബ് സൈറ്റുകളിൽ കൂക്കീസ് ഉപയോഗിക്കുന്നത്.

കുക്കികളിലെ ഡാറ്റ വൈറസ്സുകൾ അല്ലെങ്കിലും ഹാക്കർമാർ ഇതിനെ ദുരുദ്ദേശപരമായി ഉപയോഗിക്കുന്നു. ഒരു സ്പൈവെയർ ആയും കുക്കികളെ ഉപയോഗിച്ചേക്കാം. നമ്മുടെ സ്വകാര്യത നശിപ്പിക്കുന്ന വിനാശകരമായ കുക്കികൾ ഉപയോഗിക്കുന്ന ധാരാളം വെബ്സൈറ്റുകൾ നിലവിലുണ്ട്. ഇത്തരം വെബ്സൈറ്റുകൾ നമ്മുടെ കമ്പ്യൂട്ടറിൽ പ്രത്യേകമായ ഒരു കുക്കി സംഭരിച്ച് അതിൽ സന്ദർശിച്ച വെബ്സൈറ്റുകൾ, വാങ്ങിയ ഉൽപന്നങ്ങൾ, പൂരിപ്പിച്ച അപേക്ഷകൾ എന്നിവ സൂക്ഷിക്കുന്നു. മിക്കവാറും ബ്രൗസറുകളിൽ ഉപദ്രവകാരികളായ കുക്കികളെ നിയന്ത്രിക്കുകയോ നശിപ്പിക്കുകയോ ചെയ്യാനുള്ള സൗകര്യമുണ്ട്. നിയമ വിധേയമല്ലാത്ത ഉപയോഗത്തിനും, സ്വകാര്യ വിവരങ്ങൾ ഉപയോഗിക്കുന്നത് തടയുവാനും കുക്കികളുടെ അടിക്കടിയുള്ള നശീകരണത്തിലൂടെ സാധിക്കും.

9.7 കമ്പ്യൂട്ടറിൽ ഇന്റർനെറ്റ് ഉപയോഗിക്കുന്നതിനുള്ള മാർഗ്ഗ നിർദ്ദേശങ്ങൾ (Guidelines for using computers over Internet)

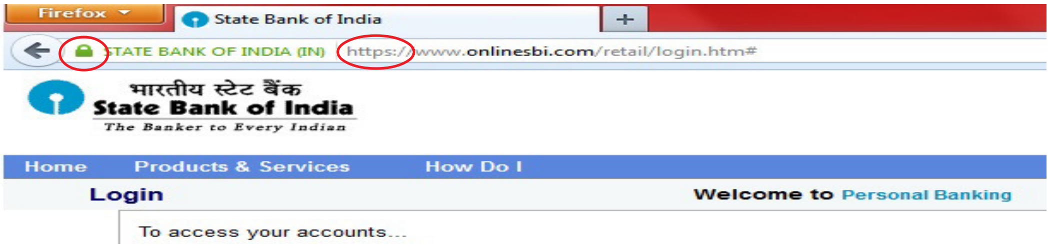
കമ്പ്യൂട്ടറിലൂടെ ഇന്റർനെറ്റ് ഉപയോഗിക്കുന്നതിനുള്ള പ്രധാന മാർഗ്ഗ നിർദ്ദേശങ്ങൾ താഴെ കൊടുത്തിരിക്കുന്നു.

- മിക്കവാറും വൈറസുകൾ ഇ-മെയിൽ അറ്റാച്ച്മെന്റ് വഴിയാണ് വ്യാപിക്കുന്നത്. പ്രേഷകരാണെന്ന് നിങ്ങൾക്ക് അറിയില്ലെങ്കിൽ അത്തരം ഇ-മെയിലിലെ അറ്റാച്ച്മെന്റുകൾ തുറക്കരുത്.
- ഉത്ഭവം സ്ഥാനം അറിയാത്ത സോഫ്റ്റ്‌വെയറുകൾ ഉപയോഗിക്കുകയോ പകർപ്പെടുക്കുകയോ ചെയ്യരുത്.
- പോപ് - അപ് പരസ്യങ്ങൾ ക്ലിക്ക് ചെയ്യുന്നതിന് പകരം അവ ക്ലോസ് ചെയ്യണം.
- USB ഡ്രൈവുകൾ ശ്രദ്ധയോടെ ഉപയോഗിക്കുക. മറ്റുള്ളവരുടെ USB സ്റ്റോറേജ് നിങ്ങളുടെ കമ്പ്യൂട്ടറിൽ ഉപയോഗിക്കുമ്പോഴും, നിങ്ങളുടെ USB സ്റ്റോറേജ് സുരക്ഷയില്ലാത്ത ഇന്റർനെറ്റ് കഫേയിലെ കമ്പ്യൂട്ടറിൽ ഉപയോഗിക്കുമ്പോഴും USB സ്റ്റോറേജിലൂടെ വൈറസ് വ്യാപനം ചെയ്യാം.
- ഫയർവാളുകൾ പ്രവർത്തന സജ്ജമാണോ എന്ന് പരിശോധിക്കുക.
- ശക്തമായ പാസ്‌വേഡ് ഉപയോഗിക്കുക. കൂടാതെ നിശ്ചിത ഇടവേളകളിൽ പാസ്‌വേർഡുകൾ മാറ്റുക.
- ഓൺലൈനിലൂടെ ആന്റിവൈറസിന്റെ വൈറസ് നിർവചനങ്ങൾ പതിവായി നവീകരിക്കുക.
- DVD യിലോ, മറ്റൊരു ഹാർഡ് ഡിസ്കിലോ സുപ്രധാന ഫയലുകൾ പതിവായി ബാക്ക് അപ്പ് ചെയ്യുക.

ശക്തമായ പാസ്‌വേഡ് ഒരുക്കുന്നതിനുള്ള മാർഗനിർദേശങ്ങൾ

- ഒരു പാസ്‌വേഡിന് ചുരുങ്ങിയത് 8 ക്യാരക്ടറുകൾ ഉണ്ടായിരിക്കണം.
- ഒരു പാസ്‌വേഡിൽ താഴെപറയുന്നവ നിർബന്ധമായും ഉൾക്കൊള്ളിക്കണം.
  - വലിയ അക്ഷരങ്ങൾ (Upper Case)
  - ചെറിയ അക്ഷരങ്ങൾ (Lower Case)
  - സംഖ്യകൾ
  - @, #, $ തുടങ്ങിയ ചിഹ്നങ്ങൾ
- സ്വകാര്യ വിവരങ്ങളായ പേര്, ജനനതീയതി തുടങ്ങിയവയോ അല്ലെങ്കിൽ പൊതുവായ വാക്യങ്ങളോ ആയിരിക്കരുത്.
- നിങ്ങളുടെ പാസ്‌വേഡ് മറ്റുള്ളവരോട് വെളിപ്പെടുത്തരുത്.
- പാസ്‌വേഡ് എഴുതി പേപ്പറിലോ അല്ലെങ്കിൽ അത് നിങ്ങളുടെ കമ്പ്യൂട്ടറിലെ ഫയലിലോ സൂക്ഷിക്കരുത്.
- എല്ലാ ലോഗിന് വേണ്ടിയും ഒരേ പാസ്‌വേഡ് ഉപയോഗിക്കരുത്.
- പതിവായി പാസ്‌വേഡ് മാറ്റുക.

- സ്വകാര്യ വിവരങ്ങൾ ഓൺലൈനിലൂടെ നൽകുമ്പോൾ ജാഗ്രത പുലർത്തുക. സ്വകാര്യ വിവരങ്ങളായ ഫോൺ നമ്പർ, വിലാസം, ഇ-മെയിൽ വിലാസം, ക്രെഡിറ്റ് കാർഡ് എന്നിവ ആവശ്യപ്പെട്ടുകൊണ്ടുള്ള ഇ-മെയിൽ സന്ദേശങ്ങൾ അവഗണിക്കുക.
- ധനകാര്യ ഇടപാടുകൾക്ക് ബാങ്കിന്റെ URL അഡ്രസ് ബാറിൽ ടൈപ്പ് ചെയ്യുക. ഇ-മെയിലിലുള്ള ലിങ്ക് ക്ലിക്ക് ചെയ്ത് ബാങ്ക് വെബ് സൈറ്റ് സന്ദർശിക്കാതിരിക്കുക. ബാങ്കുകളോ അവരുടെ ചുമതലക്കാരോ നമ്മുടെ സ്വകാര്യ വിവരങ്ങൾ യൂസർ നെയിം എന്നിവ ഫോൺ, SMS, ഇ-മെയിൽ എന്നിവ ആവശ്യപ്പെടുകയില്ല. നിങ്ങളുടെ പാസ്‌വേർഡോ, ATM കാർഡ് വിവരങ്ങൾ മറ്റുള്ളവർക്ക് വെളിപ്പെടുത്തരുത്.
- ധനകാര്യ ഇടപാടുകൾ വെബ്സൈറ്റിലൂടെ നടത്തുന്നതിന് മുൻപ് അവ സുരക്ഷിതമാണോ എന്ന് പരിശോധിക്കുക. വെബ് അഡ്രസ്സ് https:// എന്നതിൽ തുടങ്ങുന്നുവോ എന്നും താഴിന്റെ ചിഹ്നം അഡ്രസ് ബാറിൽ ഉണ്ടോ എന്നും പരിശോധിക്കുക.
- ഓൺലൈൻ അക്കൗണ്ടുകൾ പതിവായി പരിശോധന നടത്തുകയും ഏതെങ്കിലും സംശയാസ്പദമായ ഇടപാട് ശ്രദ്ധയിൽ പെട്ടാലുടൻ ബാങ്ക് അധികൃതരുമായോ ക്രെഡിറ്റ് കാർഡ് നൽകുന്നവരുമായോ ബന്ധപ്പെടുക.



ചിത്രം 9.18 സുരക്ഷിതമാക്കിയ ബാങ്കിങ്ങ് - https ഉം താഴ്ചിഹ്നവും

നമുക്കു ചെയ്യാം

- *"നിങ്ങളുടെ കമ്പ്യൂട്ടർ സിസ്റ്റത്തിലെ സുരക്ഷാഭീഷണികൾ/സൈബർ ആക്രമണം" എന്ന വിഷയത്തിൽ ഗ്രൂപ്പ് ചർച്ച നടത്തുകയും ഇതിലൂടെ ലഭ്യമായ വിവരങ്ങൾക്കനുസൃതമായി ഒരു ബാർ ഡയഗ്രം ഉപയോഗിച്ച് ചിത്രീകരിച്ച് കാണിക്കുകയും ചെയ്യുക.*
- *വിവിധ വൈറസുകളുടെ പേരും അവയുടെ പ്രത്യേകതകൾ ഉൾക്കൊള്ളുന്ന ചാർട്ട് തയാറാക്കുക.*

സ്വയം വിലയിരുത്താം

1. എന്താണ് വൈറസ്?
2. ഫിഷിങ് എന്നത് കൊണ്ട് എന്താണ് അർത്ഥമാക്കുന്നത്?
3. നമ്മുടെ ഇ-മെയിൽ വിലാസങ്ങൾ, യൂസർ നെയിം തുടങ്ങിയവ ശേഖരിച്ച് വയ്ക്കുന്ന ബ്രൗസറുകൾ ഉപയോഗിക്കുന്ന ചെറിയ ടെക്സ്റ്റ് ഫയലുകൾ ______________ എന്ന പേരിൽ അറിയപ്പെടുന്നു.
4. സുരക്ഷിതമായ ശൃംഖലകളെ തകർത്ത് അതിലെ, ഡാറ്റ നശിപ്പിക്കുന്ന പ്രവർത്തനത്തെ ______________ എന്ന പേരിൽ അറിയപ്പെടുന്നു.
5. എന്താണ് ക്വാറൻറ്റെൻ?

നമുക്ക് സംഗ്രഹിക്കാം

അമേരിക്കൻ പ്രതിരോധ വകുപ്പിനെ പ്രവർത്തനങ്ങൾക്കായി തുടക്കമിട്ട ഇന്ററർനെറ്റ് ഇന്ന് നമ്മുടെ ദൈനംദിന ജീവിതത്തിന്റെ ഭാഗമായി മാറി. ഇന്ന് ഇന്റർനെറ്റ് ഡെസ്ക് ടോപ് കമ്പ്യൂട്ടറിനെ അപേക്ഷിച്ച് മൊബൈൽ ഉപകരണങ്ങളിലാണ് കൂടുതൽ ഉപയോഗിക്കുന്നത്. ഇത് മൂലം ഇന്റർനെറ്റിന്റെ വേഗത ഒരു പ്രധാനഘടകമായി മാറി. ഇന്റർനെറ്റുമായി ബന്ധിപ്പിക്കുന്ന പുതിയ സാങ്കേതിക വിദ്യകൾ വിവര വിനിമയ നിരക്കിനെ (data transmission rate) ലക്ഷ്യമാക്കുന്നു. ഇന്റർനെറ്റ് സേവനങ്ങളായ ഇ - മെയിൽ, സാമൂഹിക മാധ്യമങ്ങൾ, തിരച്ചിൽ തുടങ്ങിയവ നമ്മുടെ വാർത്താവിനിമയ രീതികൾക്ക് പുതിയ മാനങ്ങൾ നൽകി. ഈ സേവനങ്ങൾ ലഭ്യമാക്കുന്നതിന് കമ്പ്യൂട്ടർ ശൃംഖലകൾ മുഖ്യ പങ്ക് വഹിച്ചു. ഇവക്കെല്ലാം അതിന്റേതായ ഗുണങ്ങളും ദോഷങ്ങളും ഉണ്ട്. കമ്പ്യൂട്ടർ ശൃംഖലകളിൽ വൈറസ്, വേം, ട്രോജൻ ഹോഴ്സ്, ഫിഷിങ്ങ് തുടങ്ങിയ രീതികളിലൂടെ ഭീഷണികൾ വർധിച്ചു. വിവിധ രീതിയിലുള്ള ആക്രണങ്ങളിൽ നിന്നും കമ്പ്യൂട്ടർ ശൃംഖലകളെ സംരക്ഷിക്കുവാൻ ആന്റി വൈറസ്സുകൾ സോഫ്റ്റ്വെയർ, ഫയർവാൾ തുടങ്ങിയവ ഉപയോഗിച്ചും കമ്പ്യൂട്ടറുകളിൽ ഇന്റർനെറ്റ് ഉപയോഗിക്കുമ്പോൾ പാലിക്കേണ്ട മാർഗ നിർദേശങ്ങൾ നടപ്പിലാക്കുക വഴി ശൃംഖലാക്രണത്തിന്റെ അപകട സാധ്യത കുറക്കുവാനായി.

പഠന നേട്ടങ്ങൾ

ഈ അധ്യായത്തിന്റെ പൂർത്തീകരണത്തോടെ പഠിതാവ്

- ഇന്റർനെറ്റിന്റെ ആവിർഭാവത്തിന് കാരണഭൂതരായ മഹത് വ്യക്തികളെക്കുറിച്ച് മനസ്സിലാക്കുന്നു.
- ഇന്റർനെറ്റ് ബന്ധത്തിനുവേണ്ട സോഫ്റ്റ്വെയർ, ഹാർഡ്വെയർ ഉപകരണങ്ങളുടെ ആവശ്യകത തിരിച്ചറിയുന്നു.
- ഇന്റർനെറ്റിൽ ലഭ്യമായ വിവിധതരം സേവനങ്ങൾ ഉപയോഗിക്കുന്നു.
- സാമൂഹിക മാധ്യമങ്ങളെ വിവിധ തരത്തിൽ വർഗീകരിക്കുന്നു.
- സാമൂഹിക മാധ്യമങ്ങളിലൂടെ ഇടപെടുമ്പോൾ ഉണ്ടാകുന്ന അപകട സാധ്യത മനസ്സിലാക്കുന്നു.
- ശൃംഖലയുടെ സുരക്ഷാ ഭീഷണികൾ മനസ്സിലാക്കുന്നു.

മാതൃക ചോദ്യങ്ങൾ

ഹ്രസ്വോത്തര ചോദ്യങ്ങൾ

1. ഇന്റർനെറ്റിന്റെ വികാസത്തിന് HTTP, HTML എന്നിവയുടെ കണ്ടുപിടിത്തം വളരെ പ്രധാനപ്പെട്ട വഴിത്തിരിവായി കരുതപ്പെടുന്നത് എന്തുകൊണ്ട്?
2. ഇൻട്രാനെറ്റും എക്സ്ട്രാനെറ്റും തമ്മിൽ താരതമ്യം ചെയ്യുക.
3. ലഘുവിവരണം നൽകുക.

   a. മൊബൈൽ ബ്രോഡ്ബാന്റ് b. വൈ-മാക്സ് (wi-max)

4. വെബ് ബ്രൈസർ (web browser) വെബ് ബ്രൗസിങ് (web browsing) എന്നീ പദങ്ങൾ വിവരിക്കുക.
5. ബ്ലോഗ്, മൈക്രോ ബ്ലോഗ് ഇവ താരതമ്യം ചെയ്യുക.
6. എന്താണ് വിക്കിസ് (wikis)?
7. എന്താണ് ഫയർവാൾ?

ലഘു ഉപന്യാസ ചോദ്യങ്ങൾ

1. നിങ്ങളുടെ അയൽക്കാരനായ രവി അയാളുടെ വ്യക്തിഗത ഉപയോഗങ്ങൾക്കായി ഒരു പുതിയ പേഴ്സണൽ കമ്പ്യൂട്ടർ വാങ്ങി. രവിക്ക് അയാളുടെ കമ്പ്യൂട്ടറിനെ ഇന്റർനെറ്റുമായി ബന്ധിപ്പിക്കാൻ ആവശ്യമായ ഭാഗങ്ങൾ സൂചിപ്പിക്കുക.
2. ഡയൽ-അപ് ബന്ധത്തിനുപരി ബ്രോഡ്ബാന്റ് ഇന്റർനെറ്റ് ബന്ധത്തിനുള്ള ഗുണങ്ങൾ എന്തൊക്കെയാണ്.
3. XYZ എൻജിനീയറിങ്ങ് കോളേജിന്റെ പരസ്യത്തിൽ അവരുടെ കോളേജ് അങ്കണത്തിലുടനീളം വൈ-ഫൈ ലഭ്യമാണ് എന്ന് കാണിച്ചിരിക്കുന്നു. എന്താണ് വൈ-ഫൈ എന്നും അത് കോളേജ് അങ്കണത്തിൽ എങ്ങനെ സ്ഥാപിക്കാം എന്നും വിശദീകരിക്കുക.
4. മധുവിന് ഒരു സ്ലൈഡ് തയാറാക്കേണ്ടതുണ്ട്. അതിനായി വിവരങ്ങൾ ശേഖരിക്കുന്നതായി www. google.co.in എന്ന വെബ്സൈറ്റ് ഉപയോഗിച്ചു. അയാൾ സെർച്ച് ബോക്സിൽ phishing എന്ന് ടൈപ്പ് ചെയ്ത് സെർച്ച് ബട്ടണിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ ഗൂഗിളിൽ നിന്നും വിവരങ്ങൾ ലഭ്യമാകുന്നത് എങ്ങനെ?
5. മനോജിന്റെ ഇ-മെയിൽ വിലാസമായ manoj@gmail.com നിന്നും ജോസഫിന്റെ ഇ-മെയിൽ വിലാസമായ joseph@yahoo.com ലേക്ക് ഒരു ഇ-മെയിൽ അയക്കുവാൻ മധു ആഗ്രഹിക്കുന്നു. മധുവിന്റെ കമ്പ്യൂട്ടറിൽ നിന്നും ജോസഫിന്റെ കമ്പ്യൂട്ടറിലേക്ക് ഇ-മെയിൽ അയക്കുന്നത് എങ്ങനെയാണ്?
6. ഒരു കമ്പ്യൂട്ടറിൽ ട്രോജൻ ഹോൾസ് ബാധിക്കുന്നത് എങ്ങനെയാണ്?

ഉപന്യാസ ചോദ്യങ്ങൾ

1. നിങ്ങൾക്ക് കേരള സ്കൂൾ കലോത്സവത്തിന്റെ വെബ് പേജ് ആയ www.schoolkalolsavam.in സന്ദർശിക്കുകയും നിങ്ങൾ അഡ്രസ് ബാറിൽ GRL നൽകുകയും ചെയ്യുന്നു എന്ന് വിചാരിക്കുക. ഹോം പേജ് പ്രദർശിപ്പിക്കുന്നതുവരെ പിന്തുടരേണ്ട ഘട്ടങ്ങൾ എഴുതുക.
2. സോഷ്യൽ മീഡിയയുടെ കോട്ടങ്ങൾ എഴുതുക. സോഷ്യൽ മീഡിയയുടെ കോട്ടങ്ങൾ ഒഴിവാക്കുന്നതിനാവശ്യമായ വിവിധ മാർഗങ്ങൾ എന്തൊക്കെയാണ്?
3. വിവിധ ബ്രോഡ്ബാന്റ് സാങ്കേതികവിദ്യകൾ ലഭ്യമാക്കുന്നതിന് ആവശ്യമായ ഇന്റർനെറ്റ് സ്വീകരണമാർഗങ്ങൾ വിശദീകരിക്കുക?

References

- Pradeep K. Sinha, Priti Sinha. *Computer Fundamentals* : BPB Publication
- V. Rajaraman (2010). *Fundamentals of Computers* : PHI Publication
- Thomas L. Floyd (2011). *Digital Fundamentals* : Pearson Education
- Craig Zacker, John Rourke (2008). *PC Hardware: The Complete Reference* : TMH Publication
- Abraham Silberschatz, Greg Gagne, Peter B. Galvin (2005) *Operating System Concepts* : John Wiley & Sons
- Herbert Schildt (2003). *C++ A beginners Guide* : McGraw-Hill Publication
- Bjarne Stroustrup (2013). *The C++ Programming Language* : Addison-Wesley Professional
- Robert Lafore (2009). *Object-Oriented Programming in C++* : Sams Publishing
- E. Balagurusamy (2008). *Object Oriented Programming with C++* : Tata McGraw-Hill Education
- Yashavant P Kanetkar(2000). *Let Us C++*: BPB Publication
- Andrew S. Tanenbaum, David J. Wetherall (2010). *Computer Networks* : Prentice Hall

10

വിവര സാങ്കേതികവിദ്യയുടെ ഉപയോഗം

പ്രധാന ആശയങ്ങൾ

- ഇ–ഭരണം
  - വിവിധ തരം വ്യവഹാരങ്ങൾ
  - അടിസ്ഥാന സൗകര്യം
  - പ്രയോജനങ്ങൾ
  - വെല്ലുവിളികൾ
- ഇ–വ്യാപാരം
  - ഇ-വാണിജ്യവും ഇ-വ്യാപാരവും
  - ഇലക്ട്രോണിക് രൂപത്തിൽ പണം നൽകുന്ന സമ്പ്രദായം
  - ഇ-ബാങ്കിങ്
  - മേൻമകൾ
  - വെല്ലുവിളികൾ
- ഇ–പഠനം
  - ഇ-പഠനത്തിന് ഉപയോഗിക്കുന്ന ഉപകരണങ്ങൾ
  - മേൻമകൾ
  - വെല്ലുവിളികൾ
- ആരോഗ്യ പരിപാലനതത്തിൽ ഐസിടിയുടെ ഉപയോഗങ്ങൾ
  - വൈദ്യ സംബന്ധമായ ഉപകരണങ്ങൾ
  - രോഗിയെ സംബന്ധിക്കുന്ന രേഖകൾ
  - വെബ് അധിഷ്ഠിത രോഗ നിർണ്ണയം
  - ടെലി മെഡിസിൻ
- ഐസിടി അധിഷ്ഠിത സേവനങ്ങൾ
  - ബി പി ഒ
  - കെ പി ഒ
  - കോൾ സെന്ററുകൾ

N5P1R8

നമ്മുടെ നിത്യജീവിതത്തിന്റെ ഭാഗമായുള്ള സർവ്വ മേഖലകളെയും സ്പർശിക്കുന്ന തരത്തിൽ വിവരസാങ്കേതികവിദ്യ അതിവേഗം വളർന്നു കൊണ്ടിരിക്കുന്നു. നിത്യജീവിതം സുഗമമാക്കുന്ന വ്യത്യസ്ത ഐ ടി അപ്ലിക്കേഷനുകൾ ഇന്ന് ലഭ്യമാണ്. ഉദാഹരണമായി, വിവിധ തരം ബില്ലുകൾ അടയ്ക്കുവാനും, തീവണ്ടി, വിമാന ടിക്കറ്റുകൾ ബുക്ക് ചെയ്യുവാനും, ഓൺലൈൻ വഴി സാധനങ്ങൾ വാങ്ങുവാനും, സമയത്തിന്റെയോ സ്ഥലത്തിന്റെയോ അതിർവരമ്പുകൾ ഇല്ലാതെ ഇന്ന് സാധ്യമാകുന്നു. വിദ്യാഭ്യാസ മേഖലയിലും ഐ ടി വ്യാപകമായി ഉപയോഗിക്കുന്നു. സാങ്കൽപിക ക്ലാസ്മുറികൾ ഉപയോഗിച്ചുള്ള ഇ-പഠനത്തിനും (e-Learning) ഇത് ഉപയോഗിക്കുന്നു. ഇ-ഭരണം (e-Governance), ഇ-വ്യാപാരം (e-Business), ആരോഗ്യ പരിപാലനം തുടങ്ങിയ മേഖലയിലെല്ലാം ഐ ടിയുടെ സാധ്യതകൾ വലിയ തോതിൽ ഉപയോഗിച്ചു വരുന്നു.

ഇന്നത്തെ കാലത്ത് ഐ ടി വ്യാപകമായി ഉപയോഗിക്കുന്ന ചില മേഖലകളെ കുറിച്ചാണ് ഈ അധ്യായത്തിൽ നാം ചർച്ച ചെയ്യുന്നത്. ഉദഹരണമായി, ഓൺലൈൻ ഷോപ്പിങ്ങിൽ ആവശ്യമുള്ള സാധനങ്ങൾ ഓർഡർ ചെയ്യുന്നതിനും അവ നമ്മുടെ വീടുകളിൽ എത്തിക്കു ന്നതിനും കഴിയും. പണമിടപാടുകളും ഓൺലൈൻ വഴി ചെയ്യാവുന്നതാണ്. ഐ ടിയുടെ വികാസത്തിനനുസരിച്ച് വിദ്യാഭ്യാസ മേഖലയും പുരോഗമിച്ചിട്ടുണ്ട്. സാമ്പ്രദായിക വിദ്യാഭ്യാസ അധ്യയന രീതിയിലെപോലെ അധ്യാപകരും വിദ്യാർഥികളും ഒരേ ക്ലാസ്മുറികളിൽ ഉണ്ടായിരിക്കേണ്ടതില്ല. വ്യത്യസ്ഥ സ്ഥലങ്ങളിൽ ഇരുന്നു കൊണ്ട്

സാങ്കൽപിക ക്ലാസ്മുറികളുടെ സഹായത്താൽ ഇ-പഠന സാമഗ്രികൾ ഉപയോഗിക്കാം. ഐ ടി ഉപയോഗിക്കുന്ന മറ്റൊരു പ്രധാന മേഖലയാണ് ആരോഗ്യപരിപാലനം. കമ്പ്യൂട്ടറിന്റെ സഹായത്താൽ പ്രവർത്തിക്കുന്ന മെഡിക്കൽ ഇമേജിങ്ങ് ഉപകരണങ്ങൾ ഉപയോഗിച്ചു കൊണ്ട് തുടക്കത്തിൽ തന്നെ രോഗ നിർണയം നടത്തുവാൻ സാധിക്കും. പുതിയ മരുന്നുകൾ ഉൽപാദിപ്പിക്കുന്നതിനുള്ള ഗവേഷണത്തിലും ഐ ടി പ്രമുഖ സ്ഥാനം വഹിക്കുന്നു. വിവരസാങ്കേതിക വിദ്യയുടെ പര്യായ പദമായി മിക്കപ്പോഴും ഉപയോഗിക്കുന്ന മറ്റൊരു പദമാണ് ഐ സി ടി. വിവര ആശയ വിനിമയ സാങ്കേതിക വിദ്യ/Information Communi-cation Technology -ICT അധിഷ്ഠിതമായ വ്യത്യസ്ഥ സേവനങ്ങളെ പറ്റിയും ഈ അധ്യായത്തിൽ പരാമർശിക്കുന്നു. വാർത്താ വിനിമയവും കമ്പ്യൂട്ടർ കൂട്ടിയോജിപ്പിച്ച് ഡാറ്റ ശേഖരണം (Capturing), സംഭരണം (Storing), സംസ്കരണം (Processing), വ്യാഖ്യാനം (Interpreting), ശേഖരിച്ച വിവരങ്ങളുടെ അച്ചടി (Printing information) എന്നിവക്കുള്ള സാങ്കേതികവിദ്യ ICT എടുത്തു പറയുന്നു.

10.1 ഇ–ഭരണം (e-Governance)

സർക്കാരിന്റെ പ്രവർത്തനങ്ങൾ വളരെ അനായാസമാക്കുന്നതിന് ഐ ടി വലിയ പങ്ക് വഹിക്കുന്നുണ്ട്. വിവരങ്ങൾ ശേഖരിക്കുക, സംസ്കരിക്കുക, വിശദീകരിക്കുക, സംഭരിക്കുക, കൈമാറ്റം ചെയ്യുക എന്നിവയ്ക്ക് വേണ്ടിയുള്ള പല സാങ്കേതിക വിദ്യകൾ ഇതിൽ ഉൾപ്പെടുന്നു. വളരെ പെട്ടെന്ന് ന്യായമായ തീരുമാനങ്ങൾ എടുക്കുവാൻ ഇത് സർക്കാരുകളെ സഹായിക്കുന്നു. എല്ലാ സർക്കാർ സേവനങ്ങളിലും സുതാര്യതയും ഉത്തരവാദിത്തവും വർധിപ്പിക്കുന്നു. ഐ സി ടി ഉപയോഗിച്ച് കൊണ്ടുള്ള ഭരണ നിർവ്വഹണം ആണ് ഇഭരണം എന്ന പേരിൽ പൊതുവെ അറിയപ്പെടുന്നു.

സർക്കാർ സേവനങ്ങൾ സൗകര്യപ്രദവും, കാര്യക്ഷമവും, സുതാര്യവുമായ രീതിയിൽ പൗരന്മാരിലേക്ക് എത്തിക്കുവാൻ ഐ സി ടി ഉപയോഗിച്ച് കൊണ്ടുള്ള ഇ-ഭരണത്തിന് സാധിക്കുന്നു. സർക്കാർ സേവനങ്ങൾ ഫലപ്രദമായ രീതിയിൽ തൽക്ഷണം ജനങ്ങളിലേക്ക് എത്തുന്നു എന്ന് ഉറപ്പു വരുത്തുകയാണ് ഇതിന്റെ പ്രധാന ലക്ഷ്യം. കേരളത്തിൽ മോട്ടോർ വാഹന വകുപ്പ്, വിദ്യാഭ്യാസ വകുപ്പ്, റവന്യൂ വകുപ്പ് തുടങ്ങിയവയിൽ ഇ-ഭരണം വളരെ വിജയകരമായ രീതിയിൽ നടപ്പിലാക്കിയിട്ടുണ്ട്. ഇനി നമുക്ക് ഇ ഭരണത്തിലെ വിവിധ വ്യവഹാരങ്ങളെ കുറിച്ച് ഇപ്പോൾ ചർച്ച ചെയ്യാം.

10.1.1 ഇ-ഭരണത്തിലെ വിവിധതരം വ്യവഹാരങ്ങൾ (Types of interactions in e-Governance)

ഭരണ നിർവ്വഹണത്തിൽ വിവിധതരം പങ്കാളികളുമായുള്ള പരസ്പര വ്യവഹാരങ്ങൾ ഇ-ഭരണം വഴി സുഗമമാകുന്നു. ഇത്തരത്തിലുള്ള വിവിധതരം വ്യവഹാരങ്ങളെ കുറിച്ച് ചുവടെ പ്രതിപാദിക്കുന്നു.

സർക്കാരിൽ നിന്നും സർക്കാരിലേക്ക്/Government to Government (G2G)

സർക്കാർ സംവിധാനത്തിലുള്ള ഏജൻസികൾ, വകുപ്പുകൾ, സംഘടനകൾ എന്നിവയുടെ ഇടയിൽ നടക്കുന്ന ഇലക്ട്രോണിക് രൂപത്തിലുള്ള ഡാറ്റയുടെയും വിവരങ്ങളുടെയും പങ്കുവയ്ക്കലാണിത്. ആശയ വിനിമയം, ഡാറ്റയുടെ ഉപയോഗം, ഡാറ്റ പങ്കുവയ്ക്കൽ എന്നിവ മെച്ചപ്പെടുത്തി ഇ-ഭരണ സംരംഭങ്ങളെ പിൻതുണക്കുക എന്നതാണ് G2G യുടെ ലക്ഷ്യം.

സർക്കാരിൽ നിന്നും പൗരന്മാരിലേക്ക്/Government to Citizens (G2C)

ഇത് സർക്കാരിനും പൗരന്മാർക്കുമിടയിൽ സമ്പർക്കം സാധ്യമാക്കുന്നു. ഇവിടെ പൊതു ജനങ്ങൾ ധാരാളം പൊതുസേവനങ്ങൾ ഉപയോഗപ്പെടുത്തുന്നു. പൊതുസേവനങ്ങളുടെ ലഭ്യതയും ഉപയോഗക്ഷമതയും ഇത് വർധിപ്പിക്കുന്നു. സേവനങ്ങളുടെ ഗുണനിലവാരവും ഇത് ഉയർത്തുന്നു. സർക്കാരിനും പൗരന്മാർക്കും ഇടയിൽ ഒരു സൗഹൃദ വലയം ഒരുക്കുക എന്നതാണ് ഇതിന്റെ പ്രധാന ഉദ്ദേശം.

സർക്കാരിൽ നിന്നും വ്യാപാരത്തിലേക്ക്/Government to Business (G2B)

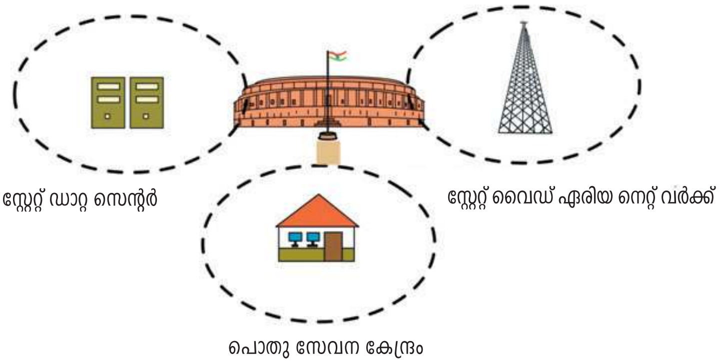
ഇവിടെ ഇഭരണ സംവിധാനം വ്യാപാര സമൂഹത്തിന് സർക്കാരുമായി സംവദിക്കുവാനുള്ള അവസരം ഒരുക്കുന്നു. ചുവപ്പുനാട പ്രവണത ഒഴിവാക്കുക, സമയനഷ്ടം പരിഹരിക്കുക, പ്രവർത്തന ചെലവ് കുറയ്ക്കുക, സർക്കാരുമായുള്ള വ്യാപാര ഇടപാടുകളിൽ കൂടുതൽ സുതാര്യത വരുത്തുക എന്നിവയാണ് ഇതിന്റെ ലക്ഷ്യം.

സർക്കാരിൽ നിന്നും ജീവനക്കാരിലേക്ക് / Government to Employees (G2E)

സർക്കാരിന്റെ നയങ്ങൾ ജീവനക്കാരിലൂടെയാണ് നടപ്പിലാക്കുന്നത്. ഇതിനായി സർക്കാരും ജീവനക്കാരും പരസ്പരം പതിവായി സമ്പർക്കം പുലർത്തേണ്ടതുണ്ട്. ഗവൺമെന്റും ജീവനക്കാരും തമ്മിൽ തമ്മിൽ സമ്പർക്കം പുലർത്തുന്നു. സർക്കാരിന്റെ വിവിധ പരിപാടികൾ നടപ്പിലാക്കുന്നതിനുള്ള നയങ്ങളും മാർഗ്ഗ നിർദ്ദേശങ്ങളും സർക്കാർ ഉത്തരവുകളുടെയും വിജ്ഞാപനങ്ങളുടെയും രൂപത്തിൽ ഇ-ഭരണ പോർട്ടലുകൾ വഴി ജീവനക്കാർക്ക് ലഭ്യമാക്കിയിട്ടുണ്ട്. ഇ-ഭരണ സേവനങ്ങൾ ഉപയോഗപ്പെടുത്തിക്കൊണ്ടാണ് സർക്കാർ ജീവനക്കാരുടെ ശമ്പളം, വ്യക്തിഗത വിവരങ്ങൾ എന്നിവ കൈകാര്യം ചെയ്യുന്നത്. (ഉദാ: സ്പാർക്ക് വെബ് സൈറ്റ് - www.spark.gov.in) ഐ സിടിയുടെ സൗകര്യങ്ങൾ ഉപയോഗപ്പെടുത്തി ഇത്തരം പ്രവർത്തനങ്ങൾ വളരെ വേഗത്തിലും കാര്യക്ഷമതയോടെയും കൈകാര്യം ചെയ്യുവാൻ സാധിക്കുന്നു.

10.1.2 ഇ-ഭരണത്തിന്റെ അടിസ്ഥാന സൗകര്യം (E-Governance Infrastructure)

ഇ-ഭരണത്തിന്റെ അടിസ്ഥാന സൗകര്യവും സംഭരണവും നൽകുന്നതിനായുള്ള സ്റ്റേറ്റ് ഡാറ്റ സെന്ററുകൾ (SDC), അവയെ ബന്ധിപ്പിക്കുന്നതിനായുള്ള സ്റ്റേറ്റ് വൈഡ് ഏരിയ നെറ്റ്‌വർക്ക് (SWAN), പൊതുജനങ്ങൾക്ക് സേവനങ്ങൾ എത്തിക്കുന്നതിനുള്ള പൊതു സേവന കേന്ദ്രങ്ങൾ (കോമൺ സർവീസ് സെന്ററുകൾ) എന്നീ ഘടകങ്ങളാണ് ഇന്ത്യയിലെ ഇഭരണത്തിന്റെ ഘടനയിൽ ഉൾപ്പെടുന്നത്. ഇ-ഭരണത്തിന്റെ ഈ മൂന്നു ഘടകങ്ങളുടേയും സംയോജനം ചിത്രം 10.1 ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 10.1 : ഇ - ഭരണത്തിന്റെ അടിസ്ഥാന സൗകര്യം

a. സ്റ്റേറ്റ് ഡാറ്റ സെന്റർ / State Data Centre (SDC)

ഇ-ഭരണം സാധ്യമാക്കുന്നതിനു ആവശ്യമായ ഒരു പ്രധാന ഘടകമാണ് ദേശീയ ഇ ഭരണ പദ്ധതി/National e-Governance Plan (NeGP) മുൻകൈ എടുത്തു രൂപപ്പെടുത്തിയ സംസ്ഥാന ഡാറ്റ കേന്ദ്രം (SDC). സേവനങ്ങൾ, ഉപയോഗങ്ങൾ, അടിസ്ഥാന സൗകര്യം എന്നിവയെ സംയോജിപ്പിക്കുകയും, G2G, G2C, G2B എന്നീ സേവനങ്ങൾക്ക് ഇലക്ട്രോണിക് ഡാറ്റാ വിതരണം കാര്യക്ഷമമായ രീതിയിൽ നടത്തുന്നതിനും വേണ്ടിയാണ് NeGP യുടെ കീഴിൽ സംസ്ഥാന ഡാറ്റ കേന്ദ്രം നിർമ്മിക്കുവാൻ തീരുമാനിച്ചു. ഒരു പൊതുവിതരണ സംവിധാനത്തിലൂടെ ഓകോ സംസ്ഥാന സർക്കാരിനും ഈ സേവനങ്ങൾ നിർവഹിക്കാവുന്നതാണ്. സ്റ്റേറ്റ് വൈഡ് ഏരിയ നെറ്റ്‌വർക്ക് (SWAN), ഗ്രാമങ്ങളിൽ വരെ വ്യാപിച്ചു കിടക്കുന്ന പൊതു സേവന കേന്ദ്രങ്ങൾ (CSC) മുതലായവ ഈ പൊതുവിതരണ സംവിധാനങ്ങൾക്ക് പിന്തുണ നൽകുന്നു. സംസ്ഥാന ഡാറ്റ കേന്ദ്രം നിരവധി സേവനങ്ങൾ നൽകുന്നു. സംസ്ഥാനങ്ങളുടെ പരമപ്രധാനമായ ഡാറ്റ സംരക്ഷിക്കുക, വിവിധ സേവനങ്ങൾക്കു വേണ്ടിയുള്ള ഓൺലൈൻ സംവിധാനം ഒരുക്കുക, സംസ്ഥാന ഇൻട്രാനെറ്റ് സംവിധാനത്തിന്റെ കവാടമാകുക, പൗരൻമാർക്കു സേവനങ്ങളുടെ കവാടമാകുക, ദുരന്ത നിവാരണ സംവിധാനമൊരുക്കുക മുതലായ പ്രവർത്തനങ്ങൾ സംസ്ഥാന ഡാറ്റ കേന്ദ്രം എന്നിവ അവയിൽ ഉൾപ്പെടുന്നു. മാനേജ്മെന്റ് നിയന്ത്രണങ്ങൾ, ഡാറ്റ മാനേജ്മെന്റിന്റെ മൊത്തം ചെലവ് കുറയ്ക്കൽ, വിഭവ നിർവ്വഹണം, വിന്യാസം, തുടങ്ങിയവയെല്ലാം സംസ്ഥാന ഡാറ്റ കേന്ദ്രങ്ങൾ നൽകുന്ന സൗകര്യങ്ങളാണ്.

b. കേരള സ്റ്റേറ്റ് വൈഡ് ഏരിയ നെറ്റ്‌വർക്ക് /Kerala State Wide Area Network (KSWAN)

KSWAN, സ്റ്റേറ്റ് ഇൻഫർമേഷൻ ഇൻഫ്രാസ്ട്രക്ച്ചർ/State Information Infrastructure (SII)ന്റെ നട്ടെല്ലായി പ്രവർത്തിക്കുന്നു. ഇത് തിരുവനന്തപുരം, കൊച്ചി, കോഴിക്കോട് എന്നീ പ്രധാന നഗരങ്ങളെ താൽപര്യകേന്ദ്രങ്ങളാക്കി (Hubs) 14 ജില്ലകളിലെ 152 ബ്ലോക്ക് പഞ്ചായത്തുകളെ തമ്മിൽ ബന്ധിപ്പിക്കുന്നു. ഈ ശൃംഖല വഴി അനേകം സർക്കാർ കാര്യാലയങ്ങളേയും ബന്ധിപ്പിക്കുന്നു. സ്റ്റേറ്റ് ഡാറ്റ സെന്ററിൽ നിന്നുള്ള അപേക്ഷകൾക്ക് അനുസരിച്ചു G2G, G2C സേവനങ്ങളെ ഏകോപിപ്പിച്ചു പ്രവർത്തിക്കുക എന്നത് കേരള സ്റ്റേറ്റ് വൈഡ് ഏരിയ നെറ്റ്‌വർക്ക് (KSWAN) ന്റെ പ്രധാന ചുമതലയാണ്.

c. പൊതുസേവന കേന്ദ്രങ്ങൾ / Common Service Cetnre (CSC)

സർക്കാർ, സ്വകാര്യ, സാമൂഹ്യ വകുപ്പുകളുടെ സേവനങ്ങൾ രാജ്യത്തെ ഗ്രാമീണർക്ക് എത്തിക്കുന്നതിനുള്ള സേവന കേന്ദ്രങ്ങളാണ് പൊതു സേവന കേന്ദ്രങ്ങൾ. വെബ് അധിഷ്ഠിത ഇ-ഭരണ സേവനങ്ങൾ ഗ്രാമ പ്രദേശങ്ങളിൽ എത്തിക്കുന്നു എന്നതാണ് **CSC** യുടെ പ്രധാന സവിശേഷത. ഈ സംവിധാനം ഉപയോഗപ്പെടുത്തിക്കൊണ്ട് നമുക്ക് വൈദ്യുതി ബിൽ, ടെലിഫോൺ ബിൽ, ജലഉപയോഗ ബിൽ എന്നിവ അടയ്ക്കുവാൻ സാധിക്കുന്നു. കൂടാതെ അപേക്ഷകൾ ഓൺലൈൻ ആയി സമർപ്പിക്കുവാനും ആവശ്യക്കാർക്ക് യോഗ്യതാപത്രം (certificate) തയ്യാറാക്കി വിതരണം ചെയ്യുവാനും ഇത് വഴി സാധിക്കുന്നു. പൊതു സേവനകേന്ദ്രങ്ങൾ വഴി ലഭ്യമാകുന്ന മറ്റു സേവനങ്ങൾ ചുവടെ ചേർക്കുന്നു.

- കാർഷിക സേവനങ്ങൾ
- വിദ്യാഭ്യാസവും പരിശീലനവുമായി ബന്ധപ്പെട്ട സേവനങ്ങൾ
- ആരോഗ്യ സേവനങ്ങൾ
- ഗ്രാമീണ ബാങ്കിന്റെയും ഇൻഷുറൻസിന്റെയും സേവനങ്ങൾ
- വിനോദ സംബന്ധമായ സേവനങ്ങൾ
- വാണിജ്യപരമായ സേവനങ്ങൾ

കേരളത്തിൽ അക്ഷയ സെന്ററുകൾ പൊതു സേവന കേന്ദ്രങ്ങളായി പ്രവർത്തിക്കുന്നു.

അക്ഷയ സെന്ററുകൾ

കേരളത്തിലെ മലപ്പുറം ജില്ലയിൽ 2002 ൽ അക്ഷയ സെന്ററുകൾക്ക് തുടക്കം കുറിച്ചു. സാധാരണക്കാരായ പൗരന്മാരിൽ ഇ-സാക്ഷരത (e-literacy) അല്ലെങ്കിൽ കമ്പ്യൂട്ടർ സാക്ഷരത ഉണ്ടാക്കിയെടുക്കുക എന്നതാണ് ഇതിന്റെ ഉദ്ദേശം. കേരള സംസ്ഥാന വിവര സാങ്കേതികവിദ്യ മിഷൻ /Kerala State Information Technology Mission (KSITM)ന്റെ ഒരു അതിരടയാളം (landmark) ആയാണ് അക്ഷയ സെന്ററുകൾ രൂപകൽപന ചെയ്തിരിക്കുന്നത്. ഐ.സി.ടി യുടെ പ്രയോജനങ്ങൾ ഡിജിറ്റൽ വേർതിരിവില്ലാതെ സംസ്ഥാനത്തെ എല്ലാ ജനവിഭാഗങ്ങളിലേക്കും എത്തിക്കുക എന്നതാണ് ഇതുകൊണ്ടു ഉദ്ദേശിക്കുന്നത്. ഇ-ഗ്രാന്റ്സ് (e-gratnz), ഇ-ഫയലിംഗ് (e-filing), ഇ-ഡിസ്ട്രിക്ട് (e-district), ഇ ടിക്കറ്റിങ് (e-ticketing) എന്നിവ ഇതിന്റെ സേവനങ്ങളിൽ ചിലതാണ്. അതുപോലെ റേഷൻ കാർഡ്, തെരഞ്ഞെടുപ്പ് ഐ ഡി എന്നിവയുടെ അപേക്ഷ സമർപ്പിക്കുവാനും, ആധാർ കാർഡിൽ പേര് ചേർക്കുവാനും, മറ്റ് ആധാർ സേവനങ്ങൾക്കും, ഇൻഷുറൻസ് ബാങ്കിങ് സേവനങ്ങൾക്കുമെല്ലാം നമുക്ക് അക്ഷയ സെന്ററുകളെ ഉപയോഗപ്പെടുത്താം.

2002 നവംബർ 18ന് മുൻ രാഷ്ട്രപതി ഡോ. എ പി ജെ അബ്ദുൾകലാം അക്ഷയ പദ്ധതി ഉദ്ഘാടനം ചെയ്തു.

സ്വകാര്യ പൊതുപങ്കാളിത്ത / Private Public Part-ership (PPP Model) മാതൃകയിൽ ആണ് അക്ഷയ കേന്ദ്രങ്ങൾ പ്രവർത്തിക്കുന്നത്. അക്ഷയ സെന്ററിന്റെ ഉടമസ്ഥാവകാശം ഒരു സ്വകാര്യ വ്യക്തിക്കാണെങ്കിൽ പോലും തദ്ദേശ സ്വയംഭരണ വകുപ്പാണ് ഈ വ്യക്തിയെ തെരഞ്ഞെടുക്കുന്നത്. അക്ഷയ സെന്റർ പ്രവർത്തിക്കുന്ന പഞ്ചായത്തിലെ വ്യക്തികൾക്കാണ് മുൻഗണന. ഉടമസ്ഥന്റെ വരുമാനം എന്നത് ഓരോ ഇടപാടിനും പൗരന്മാരിൽ നിന്നും ഈടാക്കുന്നതോ അല്ലെങ്കിൽ സർക്കാരിൽ നിന്നും ലഭിക്കുന്നതോ ആയ നാമമാത്രമായ സേവന വേതനം മാത്രമാണ്.

അക്ഷയ
അവസരങ്ങളുടെ ജാലകം
www.akshaya.kerala.gov.in

ഭാരത സർക്കാരിന്റെ കീഴിലുള്ള യുണീക് ഐഡന്റിഫിക്കേഷൻ അതോറിറ്റി ഓഫ് ഇന്ത്യ / U-ique Ide-tificatio- Authortiy of I-dia (UIDAI). ഇ-ഭരണ സംവിധാനത്തിന്റെ ഭാഗമായി പൗരന്മാർക്ക് നൽകുന്ന 12 അക്ക വ്യക്തിഗത തിരിച്ചറിയൽ സംഖ്യയാണ് ആധാർ. ഇന്ത്യയിൽ എവിടെയും ഒരു തിരിച്ചറിയൽ രേഖയായും, മേൽ വിലാസത്തിന്റെ തെളിവ് ആയും ഇത് ഉപയോഗിക്കാം. ഇന്ത്യയിലെ ഏതൊരു പൗരനും പ്രായ ലിംഗഭേദം ഇല്ലാതെ ആധാറിൽ പേര് രേഖപ്പെടുത്താവുന്നതാണ്. ഓരോ വ്യക്തിക്കും സൗജന്യമായി ഒരു പ്രാവശ്യം പട്ടികയിൽ ചേർക്കാവുന്നതാണ്. പേര് ചേർക്കുന്ന സമയത്ത് പൗരന്റെ അടിസ്ഥാന വിവരങ്ങൾ (പേര്, വയസ്സ്, ലിംഗം, വിലാസം), ബയോമെട്രിക് വിശദാംശങ്ങൾ (ഫോട്ടോ, 10 വിരലിന്റെയും 2 കൃഷ്ണമണികളുടെയും അടയാളം) എന്നിവ കൃത്യമായി രേഖപ്പെടുത്തുന്നതാണ്. ഓരോ ആധാർ നമ്പറും അദ്വിതീയവും വ്യക്തിയുടെ ജീവിതാവസാനം വരെ നിയമ സാധുതയുള്ളതുമായിരിക്കും. ബാങ്കിങ് സേവനങ്ങൾ പ്രയോജനപ്പെടുത്തുവാനും , മൊബൈൽഫോൺ കണക്ഷൻ എടുക്കുന്നതിനും മറ്റു സർക്കാർ, സർക്കാർ ഇതര സേവനങ്ങൾക്കും ആധാർ നമ്പർ ഉപയോഗപ്പെടുത്താം.

10.1.3 ഇ-ഭരണത്തിന്റെ പ്രയോജനങ്ങൾ (Benefits of e-Governance)

ഇ-ഭരണം പ്രയോഗത്തിൽ വരുത്തുന്നത് കൊണ്ട് ധാരാളം മേന്മകൾ ഉണ്ട്. സർക്കാരിന്റെ പ്രവർത്തനങ്ങളെ പറ്റിയും, സർക്കാർ നടപ്പിൽ വരുത്താൻ ഉദ്ദേശിക്കുന്ന നയപരിപാടികളെ കുറിച്ചുമുള്ള അവബോധം ജനങ്ങളിൽ സൃഷ്ടിച്ചെടുക്കുവാൻ ഇ-ഭരണത്തിന് കഴിയുന്നു. ഇഭരണത്തിന്റെ പ്രധാന പ്രയോജനങ്ങൾ ചുവടെ ചേർക്കുന്നു.

- ഇ-ഭരണം സർക്കാർ സർവീസുകളിൽ കമ്പ്യൂട്ടർ വത്കരണം സാധ്യമാക്കുന്നു. ജനക്ഷേമ പ്രവർത്തനങ്ങളുമായി ബന്ധപ്പെട്ട വിവരങ്ങൾ എളുപ്പത്തിൽ ജനങ്ങളിൽ എത്തിച്ചേരുന്നു എന്ന് ഉറപ്പു വരുത്തുന്നു.
- ഇ-ഭരണം സർക്കാരിന്റെ എല്ലാ തലങ്ങളിലും പൗരന്മാരുടെ ഇടപെടലുകൾ ഉറപ്പു വരുത്തിക്കൊണ്ട് ജനാധിപത്യം ശക്തിപ്പെടുത്തുന്നു.
- ഇത് സർക്കാർ പ്രവർത്തനങ്ങൾ കൂടുതൽ സുതാര്യമാക്കുകയും അഴിമതി തടയാൻ സഹായിക്കുകയും ചെയ്യുന്നു.
- സൂക്ഷ്മമായി നിരീക്ഷിക്കപ്പെടുന്നു എന്ന ബോധ്യം ഉള്ളതിനാൽ എല്ലാ സർക്കാർ വകുപ്പുകളുടെയും പ്രവർത്തനങ്ങൾ കാര്യക്ഷമവും ഉത്തരവാദിത്തപൂർണവും ആകുന്നു.
- ഇ-ഭരണം കാര്യക്ഷമമായി പ്രയോഗത്തിൽ വരുത്തുന്നതിലൂടെ പൊതുജനത്തിന് നിരന്തരമായി കാര്യാലയങ്ങൾ സന്ദർശിക്കേണ്ട അവസ്ഥ ഒഴിവാകുന്നു. ഇത് സമയവും സാമ്പത്തും ലാഭിക്കുന്നു.

10.1.4 ഇ-ഭരണത്തിന്റെ വെല്ലുവിളികൾ (Challenges to e-Governance)

ഇ-ഭരണത്തിന് ഒരുപാട് മേന്മകൾ ഉണ്ടെങ്കിലും, ചില വെല്ലുവിളികളും ഇതിന് അഭിമുഖീകരിക്കേണ്ടി വരുന്നു. ഇ-ഭരണം പ്രയോഗത്തിൽ വരുത്തുമ്പോൾ അഭിമുഖീകരിക്കേണ്ടി വരുന്ന ചില വെല്ലുവിളികൾ എന്തൊക്കെ എന്ന് നോക്കാം.

- ഡിജിറ്റൽ വിഭജനം മൂലം ഇ-ഭരണ സേവനം കൃത്യമായി ഉപയോഗപ്പെടുത്തുന്നതിന് ചില തടസ്സങ്ങൾ നേരിടുന്നു. വിദൂര സ്ഥലങ്ങളിൽ ജീവിക്കുന്ന കമ്പ്യൂട്ടർ സാക്ഷരത കുറഞ്ഞ ആളുകൾക്ക് ഇഭരണ സേവനങ്ങൾ പ്രയോജനപ്പെടുത്തുന്നതിന് ബുദ്ധിമുട്ടുകൾ നേരിടുന്നു.
- ഇ-ഭരണ സംവിധാനം കമ്പ്യൂട്ടർ/വെബ് അധിഷ്ഠിതമായതിനാൽ മികച്ച രീതിയിലുള്ള സുരക്ഷാ സംവിധാനങ്ങൾ ആവശ്യമാണ്. അല്ലാത്ത പക്ഷം സൈബർ ആക്രമണങ്ങൾക്കുള്ള സാധ്യത വളരെ കൂടുതലാണ്.
- ഇ-ഭരണ സേവനങ്ങൾ പ്രയോഗത്തിൽ വരുത്തുന്നതിനും അതിന്റെ പരിപാലനത്തിനും വലിയ തോതിലുള്ള മുതൽമുടക്കും ആസൂത്രണവും ആവശ്യമാണ്.
- വ്യക്തിഗത വിവരങ്ങളും, ബയോമെട്രിക് ഡാറ്റയും നിർദ്ദിഷ്ട ഏജൻസികളുമായി പങ്കുവയ്ക്കുന്നതിൽ ഭൂരിഭാഗം ആളുകളും ഉത്കണ്ഠാകുലരാണ്.
- ഇ-ഭരണം കാര്യശേഷി ഉള്ളതും ഫലപ്രാപ്തവുമാകണമെങ്കിൽ വിവിധ വകുപ്പുകളുടെ ഏകോപനം വളരെ അത്യാവശ്യമാണ്.

10.1.5 ഇ-ഭരണവുമായി ബന്ധപ്പെട്ട വെബ്സൈറ്റുകൾ (Useful e-Governance websites)

കേരള സർക്കാരിന്റെ വിവിധ സേവനങ്ങൾ ജനങ്ങളിലേക്ക് എത്തിക്കുന്നതിന് വേണ്ടിയുള്ള ചില വെബ്സൈറ്റ് വിലാസങ്ങൾ പട്ടിക 10.1 ൽ കാണിച്ചിരിക്കുന്നു. ചിത്രം 10.2 ൽ DHSE. ചിത്രം 10.3ൽ ഇ-ഡിസ്ട്രിക്ട് പ്രധാന പേജും കാണിച്ചിരിക്കുന്നു.

വെബ്സൈറ്റുകളും മേൽവിലാസവും	ഉപയോഗം / സേവനം
www.dhsekerala.gov.in	കേരള സർക്കാരിന്റെ കീഴിലുള്ള ഹയർ സെക്കണ്ടറി വിദ്യാഭ്യാസ വകുപ്പിന്റെ ഔദ്യോഗിക വെബ്സൈറ്റ്. വിദ്യാർത്ഥികൾക്കും അധ്യാപകർക്കും സ്കൂൾ ഭരണാധികാരികൾക്കും ആവശ്യമായ വിവിധ സൗകര്യങ്ങളും സേവനങ്ങളും ഇത് ലഭ്യമാക്കുന്നു.
www.edistrict.kerala.gov.in	ഇത് സർക്കാർ സേവനങ്ങൾ പൊതുജനങ്ങളിലേക്ക് എത്തിക്കുന്നു.
www.incometaxindia.gov.in	ഇത് ആദായനികുതി വകുപ്പിന്റെ സേവനങ്ങൾ ലഭ്യമാക്കുന്നു.
www.keralamvd.gov.in	ഇതാണ് കേരള മോട്ടോർ വാഹന വകുപ്പിന്റെ ഔദ്യോഗിക വെബ്സൈറ്റ്.
www.rti.gov.in	സർക്കാരിൽ നിന്നോ അധികാരികളിൽ നിന്നോ വിവരങ്ങൾക്കായി അപേക്ഷ നൽകിയ പൗരന് വിവരാവകാശ നിയമം 2005 ഉത്തരവ് അനുസരിച്ച് സമയബന്ധിതമായി മറുപടി നൽകുന്നു.
www.itmission.kerala.gov.in	കേരള സർക്കാരിന്റെ ഈ വെബ്സൈറ്റ് വിവിധ ഐ ടി സംരംഭങ്ങൾക്ക് ഭരണപരമായ പിന്തുണ നൽകുന്നു.
www.spark.gov.in	ജീവനക്കാരന്റെ സേവന വേതന കാര്യങ്ങൾ കൈകാര്യം ചെയ്യുന്ന G2E വിഭാഗത്തിലുള്ള വെബ്സൈറ്റ്.

പട്ടിക 10.1 ഇ-ഭരണവുമായി ബന്ധപ്പെട്ട വെബ്സൈറ്റുകൾ

ചിത്രം 10.2: DHSE വെബ്സൈറ്റിന്റെ പ്രധാന പേജ്

www.dhsekerala.gov.i-

ഹയർ സെക്കന്ററി പരീക്ഷകൾക്ക് രജിസ്റ്റർ ചെയ്യൽ, ഹാൾ ടിക്കറ്റ് വിതരണം, പരീക്ഷാഫല പ്രസിദ്ധീകരണം, വിവിധ ഉത്തരവുകളുടെയും വിജ്ഞാപനങ്ങളുടെയും വിതരണം മുതലായവ ഇതിന്റെ സേവന പരിധിയിൽ ഉൾപ്പെടുന്നു. വിവിധ ഫോറങ്ങളും, മാതൃക ചോദ്യ കടലാസുകൾ ഉൾപ്പെടെയുള്ള പഠന സാമഗ്രികളും ഡൗൺലോഡ് ചെയ്യാനുള്ള സൗകര്യം ഈ വെബ്സൈറ്റിൽ ലഭ്യമാണ്.

ചിത്രം 10.3: ഇ - ഡിസ്ട്രിക്ട് (e-District) പോർട്ടലിന്റെ പ്രധാന പേജ്

ദേശീയ ഇഭരണ പദ്ധതിയ്ക്ക് കീഴിൽ സംസ്ഥാനം ഏറ്റെടുത്തിരിക്കുന്ന ദൗത്യ നിർവ്വഹണ പദ്ധതി (State Missio- Mode Project) ആണ് ഇഡിസ്ട്രിക്ട് (e-District). കമ്പ്യൂട്ടർവത്കൃത സേവനങ്ങൾ പൊതുജനങ്ങളിലെത്തിക്കുക എന്നതാണ് ഈ പദ്ധതിയുടെ ലക്ഷ്യം.

കേരളത്തിലെ റവന്യു വകുപ്പുമായി ബന്ധപ്പെട്ട വിവിധ യോഗ്യത പത്രങ്ങളുടെ (Certificates) വിതരണം ഇഡിസ്ട്രിക്ട് പോർട്ടലുമായി സമന്വയിപ്പിച്ചിരിക്കുന്നു ഇത് സർക്കാർ കാര്യാലയങ്ങളിലെ പല സങ്കീർണമായ വ്യവസ്ഥകളെയും ലഘൂകരിക്കുകയും പൗരന്റെ ജീവിത രീതിയെ എളുപ്പമാക്കുകയും ചെയ്യുന്നു. പൗരനെ സംബന്ധിച്ചിടത്തോളം പൊതുരേഖകളുടെ പകർപ്പുകൾ നേടുന്നതോ, അപേക്ഷകൾ സമർപ്പിക്കുന്നതോ, അധികാരികളെ കാണുന്നതോ, മറ്റു ദൈനം ദിനാവശ്യങ്ങൾക്കുള്ള വിവരങ്ങൾ ആരായുന്നതോ ബുദ്ധിമുട്ടേറിയ കാര്യമാണ്. ഇത് അവന്റെ/അവളുടെ യാത്രാ ചെലവ് വർധിപ്പിക്കുകയും, ദിവസ വരുമാനവും സമയവും നഷ്ടപ്പെടുത്തുകയും ചെയ്യുന്നു. സർക്കാർ സ്ഥാപനങ്ങളിൽ ചിലപ്പോൾ ബന്ധപ്പെട്ട രേഖകളോ, വിവരങ്ങളോ അല്ലെങ്കിൽ അധികാരികളോ യഥാസമയം ലഭ്യമാകണമെന്നില്ല. ആയതിനാൽ സ്ഥാപനത്തിലേക്കുള്ള സന്ദർശനം ആവർത്തിക്കേണ്ടി വരികയും ചെലവ് വർധിക്കാനിടയാകുകയും ചെയ്യുന്നു. ആവശ്യമായ രേഖകൾ സഹിതം ഒരാൾക്ക് ഇഡിസ്ട്രിക്ട് വെബ് പോർട്ടലിൽ നേരിട്ടോ, അക്ഷയ കേന്ദ്രങ്ങൾ മുഖാന്തരമോ അപേക്ഷകൾ സമർപ്പിച്ച് സേവനങ്ങൾ തേടാവുന്നതാണ്. അക്ഷയ കേന്ദ്രങ്ങൾ പൊതു പൗരനുവേണ്ടി ഓൺലൈൻ അപേക്ഷകൾ സമർപ്പിക്കുന്നു. ഈ അപേക്ഷകൾ വിവിധ ഘട്ടങ്ങളിലൂടെ വില്ലേജ് ഓഫീസറുടെ അരികിൽ എത്തിച്ചേരുന്നു. ഈ അപേക്ഷകളിലുള്ള വില്ലേജ് നടപടിക്ക് ശേഷം ഓഫീസറുടെ തൊട്ടടുത്ത ഉന്നത അധികാരിക്ക് സമർപ്പിക്കുകയോ, അല്ലെങ്കിൽ അംഗീകരിച്ചതിനു ശേഷം ഡിജിറ്റൽ ഒപ്പ് സഹിതം ആവശ്യക്കാരന് വിതരണം ചെയ്യുകയോ ചെയ്യുന്നു. യോഗ്യത പത്രം ലഭ്യമായാൽ അപേക്ഷകനെ എസ് എം എസ് വഴി അറിയിക്കുകയും ചെയ്യുന്നു.

സ്വയം വിലയിരുത്താം

1. ഐ സി ടി ഉപയോഗിച്ച് സർക്കാർ സേവനങ്ങൾ സൗകര്യപ്രദവും, കാര്യക്ഷമവും, സുതാര്യവുമായ രീതിയിൽ പൗരന് എത്തിക്കുവാനുള്ള പ്രയോഗത്തിന്റെ പേരെന്ത്?
2. ഇ-ഭരണം എന്ന പദം നിർവ്വചിക്കുക.
3. ഭരണ നിർവ്വഹണത്തിൽ വിവിധതരം പങ്കാളികളുമായുള്ള പരസ്പര വ്യവഹാരങ്ങൾ ഇ-ഭരണം സുഗമമാക്കുന്നു. ഈ പ്രസ്താവന ശരിയോ തെറ്റോ എന്ന് പറയുക.
4. ഒരു ഇ-ഭരണ വെബ്സൈറ്റിന് ഉദാഹരണം നൽകുക.
5. എന്താണ് KSWAN?

10.2 ഇ-വ്യാപാരം (e-Business)

നമ്മൾ പലപ്പോഴും സാധനങ്ങൾ വാങ്ങാൻ കടകളിൽ പോവുകയും അതിനായി ഒരുപാട് സമയവും അധ്വാനവും നീക്കി വെക്കുകയും ചെയ്യാറുണ്ട്. വീട് വിട്ട് പുറത്തിറങ്ങാതെ സാധനങ്ങൾ വാങ്ങുക എന്ന ആശയം എങ്ങനെയുണ്ട്? എന്നാൽ ഇന്ന് ഇതൊരു യാഥാർത്ഥ്യമാണ്. ഒരു വിധം എല്ലാ സേവനങ്ങളും വ്യാപാരവും ഇന്ന് ഓൺലൈനിൽ ലഭ്യമാണ്. അത് ഏതാനും മൗസ് ക്ലിക്കുകൾ മാത്രം അകലെയാണ്.

ഇ-വ്യാപാരം (e-Business) എന്നത് കൊണ്ട് ഉദ്ദേശിക്കുന്നത്, വ്യാപാര വിവരങ്ങൾ പങ്കുവയ്ക്കുക, വ്യാപാര ബന്ധങ്ങൾ നിലനിർത്തുക, ഐ സി ടി ഉപയോഗിച്ച് വ്യാപാര ഇടപാടുകൾ നടത്തുക എന്നിവയാണ്. കമ്പനികകൾ (വ്യാപാരം), പൊതു സ്ഥാപനങ്ങൾ (ഭരണസമിതി), വ്യക്തികൾ (ഉപഭോക്താവ്) മുതലായവർ ഇ-വ്യാപാരത്തിന്റെ ഭാഗമായി വരുന്നു. ഇ-വാണിജ്യ സേവനങ്ങൾ കാര്യക്ഷമമായി ഉപയോഗിച്ചുകൊണ്ട് മികച്ച രീതിയിലുള്ള വിപണനത്തിനും, വില്പന വർധനവിനും ഇ-വ്യാപാരം സഹായിക്കുന്നു. ഓൺലൈൻ കൂടിക്കാഴ്ചകൾ സംഘടിപ്പിക്കുക, പ്രവർത്തനതലം പങ്കുവെക്കുക മുതലായ കാര്യങ്ങൾ വഴി നിരന്തര സമ്പർക്കവും, യാത്രാച്ചെലവും കുറയ്ക്കാൻ സാധ്യമാകുന്നു.

10.2.1 ഇ-വാണിജ്യവും ഇ-വ്യാപാരവും (e-Commerce and e-Business)

ഇ-വാണിജ്യവും ഇ-വ്യാപാരവും പലപ്പോഴും ഇടകലർത്തി ഉപയോഗിക്കുന്ന പദങ്ങളാണെങ്കിലും ഇവ വ്യത്യാസപ്പെട്ടിരിക്കുന്നു. പണം കൈമാറ്റം ചെയ്യുന്ന വ്യാപാര ഇടപാടുകളാണ് ഇവാണിജ്യത്തിൽ ഉൾപ്പെടുന്നതെങ്കിൽ വ്യാപാരം നടത്തിക്കൊണ്ടുപോകുവാൻ ആവശ്യമായ എല്ലാ ഘടകങ്ങളും ഇ-വ്യാപരത്തിൽ ഉൾപ്പെടുന്നു. വിപണനം, അസംസ്കൃത വസ്തുക്കളുടെയും ചരക്കുകളുടെയും വാങ്ങൽ, ഉപഭോക്താക്കൾക്കുള്ള പരിശീലനം, വിതരണക്കാരെ കണ്ടെത്തൽ എന്നിവയെല്ലാം ഇ-വ്യാപാരത്തിൽ ഉൾപ്പെടുന്നു. അതിനാൽ ഇ-വാണിജ്യത്തിന്റെ വിപുലീകരണമാണ് ഇ-വ്യാപാരം.

10.2.2 ഇലക്ട്രോണിക് രൂപത്തിൽ പണം നൽകുന്ന സമ്പ്രദായം (Eletcronic Payment System (EPS))

കീശയിൽ പണമില്ലാതെയുള്ള ജീവിതം നിങ്ങൾക്കു ചിന്തിക്കുവാൻ കഴിയുമോ? പണം കൈകൊണ്ടു തൊടാതെ ഒരാഴ്ചയോളം തള്ളി നീക്കി എന്ന് ആരെങ്കിലും പറഞ്ഞാൽ നിങ്ങൾ വിശ്വസിക്കുമോ? ഇന്ന് നാം ജീവിക്കുന്ന ലോകത്തു ഒരു വിധം എല്ലാ കച്ചവട സാധനങ്ങളിലും അതിന്റെ വില അനുബന്ധമായി ചേർത്തിരിക്കും (Price tag). അപ്പോൾ പിന്നെ കാശ് നൽകാതെ ഒരു സാധനം വാങ്ങുക എന്നത് എങ്ങനെ സാധ്യമാകും?

ഇലക്ട്രോണിക് രൂപത്തിൽ പണം നൽകുന്ന സമ്പ്രദായം Eletcronic Payment System (EPS) ഇതിനു പരിഹാരമാകുന്നു. നമ്മൾ ഇലക്ട്രോണിക് വ്യാപാരത്തിലേക്ക് കടന്നു ചെല്ലുമ്പോൾ പണമിടപാടുകളും ഇലക്ട്രോണിക് രൂപത്തിലാകേണ്ടതുണ്ട്. സമുചിതമായ രീതിയിൽ പ്രയോഗത്തിൽ വരുത്തിയാൽ ഇത് സൗകര്യപ്രദവും സുരക്ഷിതവുമായിരിക്കും. ഈ വ്യാപാരത്തിൽ ഇ പി എസ് ഒരു പ്രധാന പങ്ക് വഹിക്കുന്നു. വളരെ വലിയ പ്രാധാന്യമുണ്ട്. ഓൺലൈൻ സാഹചര്യത്തിൽ ഉപഭോക്താവും വിൽപ്പനക്കാരനും

തമ്മിൽ നടത്തുന്ന സാമ്പത്തിക ഇടപാടുകളെ ആണ് ഇലക്ട്രോണിക് രൂപത്തിലുള്ള പണമിടപാട് സമ്പ്രദായം (Eletcronic Payment System) എന്ന് പറയുന്നത്. ഒരു ബാങ്കിന്റെയോ ഇടനിലക്കാരന്റെയോ പിന്തുണയോടെയുള്ള ഡിജിറ്റൽ സാമ്പത്തിക ഉപകരണം ഉപയോഗപ്പെടുത്തിയാണ് ഇവിടെ സാമ്പത്തിക ഇടപാടുകൾ സുഗമമാക്കുന്നത് (അതായത് ക്രെഡിറ്റ്/ഡെബിറ്റ് കാർഡ്, ഇലക്ട്രോണിക് ചെക്ക് അല്ലെങ്കിൽ ഡിജിറ്റൽ പണം)

10.2.3 ഇ-ബാങ്കിങ് (e-Banking)

വിവരസാങ്കേതിക വിദ്യയുടെ പുരോഗതിക്കനുസരിച്ചു ബാങ്കിങ് മേഖലയിലും വലിയ തോതിലുള്ള മാറ്റങ്ങൾ സംഭവിച്ചിട്ടുണ്ട്. ബാങ്കിങിലെ മേഖലയിലെ നമ്മുടെ പരമ്പരാഗത സങ്കൽപ്പങ്ങൾ ഒരുപാട് മാറിക്കഴിഞ്ഞു. ഉദാഹരണമായി, ഇന്ന് സാമ്പത്തിക ഇടപാടുകൾക്ക് ബാങ്ക് സന്ദർശിക്കേണ്ട ആവശ്യം തന്നെ ഇല്ല. യാത്രകളിൽ പോലും ഓൺലൈൻ ബാങ്കിങ് സൗകര്യങ്ങൾ ഉപയോഗപ്പെടുത്തി നമുക്ക് ഇടപാടുകൾ നടത്താം. എ ടി എം, ഡെബിറ്റ് കാർഡ്, ക്രെഡിറ്റ് കാർഡ്, ഇന്റർനെറ്റ് ബാങ്കിങ്, കോർ ബാങ്കിങ് എന്നിങ്ങനെയുള്ള സൗകര്യങ്ങൾ പരമ്പരാഗത ബാങ്കിങ് സമ്പ്രദായത്തിൽ നിന്നും ഇ-ബാങ്കിങ് സമ്പ്രദായത്തിലേക്കു മാറുവാൻ സഹായിച്ചിട്ടുണ്ട്. ബാങ്കിങ് സേവനങ്ങൾ ഉപഭോക്താക്കൾക്ക് നേരിട്ട് ഇലക്ട്രോണിക് മാർഗ്ഗത്തിലൂടെ നൽകുന്നതിനെയാണ് ഇ -ബാങ്കിങ് അല്ലെങ്കിൽ ഇലക്ട്രോണിക് ബാങ്കിങ് എന്ന് പറയുന്നത്. ഇതിൽ ഭൂമിശാസ്ത്രപരമായ പരിമിതികൾ ഇല്ലാതെ ഡാറ്റ ഉപയോഗിക്കുവാൻ കഴിയും.

10.2.4 ഇ-വ്യാപാരത്തിന്റെ മേൻമകൾ (Advantages of e-Business)

ഇ-വ്യാപാരം ഉപഭോക്താക്കൾക്കും സ്ഥാപനത്തിനും ധാരാളം മേന്മകൾ പ്രദാനം ചെയ്യുന്നു. ഇ-വ്യാപാരത്തിന്റെ ഉപയോഗം കൊണ്ട് ലഭ്യമാകുന്ന ചില പ്രധാന ഗുണങ്ങളെ പറ്റി നമുക്ക് ചർച്ച ചെയ്യാം.

- ***ഭൂമിശാസ്ത്രപരമായ പരിമിതികൾ മറികടക്കുന്നു.*** സാധാരണ രീതിയിലുള്ള കച്ചവടത്തിൽ നിങ്ങൾക്ക് സേവനങ്ങൾ ലഭിക്കുന്നത് പരിമിതമായ സ്ഥലങ്ങളിൽ നിന്ന് മാത്രമായിരിക്കും. പക്ഷെ ഈ പരിമിതി ഇ-വാണിജ്യം വഴി പരിഹരിക്കപ്പെടുന്നു.
- ***ഇ-വ്യാപാരം പ്രവർത്തന ചെലവ് കുറയ്ക്കുന്നു.*** വ്യാപാരിക്ക് ഇ-വാണിജ്യത്തിൽ ഒരു പ്രധാന സ്ഥലത്തെ ഭൗതിക സാഹചര്യങ്ങൾ ആവശ്യമില്ല; അതിനാൽ പ്രവർത്തന ചെലവ് കുറയുന്നു. ഇതുവഴി ലഭ്യമാകുന്ന ലാഭത്തിന്റെ ഒരു വിഹിതം ഇളവ് (Discount) രൂപത്തിൽ ഉപഭോക്താവിന് തന്നെ കൊടുക്കുവാൻ സാധിക്കുന്നു.
- ***യാത്രാ സമയവും ചെലവും കുറയ്ക്കുന്നു.*** ചിലപ്പോൾ ഉപഭോക്താക്കൾക്ക് അവരുടെ പ്രിയപ്പെട്ട വിപണനശാലകളിൽ എത്തിച്ചേരുവാൻ ദീർഘദൂരം യാത്ര ചെയ്യേണ്ടതായി വന്നേക്കാം. ഇ-വ്യാപാരം വഴി ഇതേ വിപണനശാലസാങ്കൽപ്പികമായി സന്ദർശിക്കുവാൻ അനുവദിക്കുന്നു.
- ***എല്ലായ്പ്പോഴും തുറന്നു പ്രവർത്തിക്കുന്നു.*** ഇ-വ്യാപാര സേവനം എല്ലായ്പ്പോഴും ലഭ്യമാണ് (24x7). വ്യാപാരിയെ സംബന്ധിച്ചിടത്തോളം, അവർക്ക് വിൽപന വർദ്ധിക്കുന്നു. ഉപഭോക്താവിനാണെങ്കിൽ "എല്ലായ്പ്പോഴും" തുറന്നു പ്രവർത്തിക്കുന്ന വിപണനശാലകൾ കൂടുതൽ സൗകര്യപ്രദമാണ്.

- ***ഒരുപാട് ഉൽപന്നങ്ങൾക്കിടയിൽ നിന്ന് നമുക്കാവശ്യമായ ഉൽപന്നം വളരെ പെട്ടെന്ന് കണ്ടെത്തുവാൻ കഴിയും.*** ഇവ്യാപാര വെബ്സൈറ്റ് വഴി ഉപഭോക്താക്കൾക്ക് വ്യത്യസ്ത വ്യാപാരികളുടെ അനേകം ഉൽപന്നങ്ങൾക്കിടയിൽ നിന്ന് ഇഷ്ടമുള്ളത് തെരഞ്ഞെടുക്കുവാൻ കഴിയുന്നു. ചില വെബ്സൈറ്റുകൾ ഉപഭോക്താവിന്റെ മുൻഗണനയും വാങ്ങിച്ച വസ്തുക്കളുടെ പട്ടികയും ഓർമിപ്പിക്കുന്നതിനാൽ ഉൽപന്നങ്ങൾ ആവർത്തിച്ചു വാങ്ങുന്നവർക്ക് ഇത് സൗകര്യപ്രദമാണ്. ഉൽപന്നങ്ങളുടെ സ്വഭാവം, വിലയുടെ താരതമ്യം മുതലായ സവിശേഷതകൾ ഇവ്യാപാരത്തിന്റെ മറ്റൊരു ആകർഷണമാണ്.

മൊബൈൽ ഉപകരണങ്ങളായ മൊബൈൽ ഫോൺ, പി ഡി എ (PDA) മുതലായവ ഉപയോഗിച്ച്കൊണ്ടുള്ള വ്യാപാര പ്രവർത്തനങ്ങളെയും, പ്രക്രിയകളെയും, ഉപയോഗങ്ങളെയും ആണ് എം വ്യാപാരം (m-Busi-ess) അല്ലെങ്കിൽ മൊബൈൽ വ്യാപാരം (mobile busi-ess) എന്ന പദം കൊണ്ട് അർത്ഥമാക്കുന്നത്. എം വ്യാപാരം ഇവ്യാപാരത്തിന്റെ ഒരു ശാഖയായി കണക്കാക്കാം.

ഇന്റർനെറ്റ് ഉപയോഗിച്ച് ലഘു വ്യാപാരം നടത്തുന്നതിനെയാണ് ഇടൈലിങ് e-taili-g (അല്ലെങ്കിൽ ഇലക്ട്രോണിക് റീടൈലിങ് (eletcro-ic retaili-g) എന്ന് പറയുന്നത്. ഇത് B2C (ബിസിനസ് ടു കൺസ്യൂമർ) യുടെ സർവ്വ സാധാരണമായ ഒരു രൂപമാണ്.

10.2.5 ഇ-വ്യാപാരത്തിന്റെ വെല്ലുവിളികൾ (Chalanges to e-Business)

വ്യാപാരലോകത്ത് ഇവ്യാപാരത്തിന് അതിശക്തമായ സാധ്യതകൾ ഉണ്ടെങ്കിലും ഇന്ത്യ പോലുള്ള വികസ്വര രാഷ്ട്രങ്ങളിൽ ഇതിന് പലവിധ വെല്ലുവിളികളെയും അഭിമുഖീകരിക്കേണ്ടി വരുന്നു. പ്രധാനപ്പെട്ട വെല്ലുവിളികൾ താഴെ ചർച്ച ചെയ്യുന്നു.

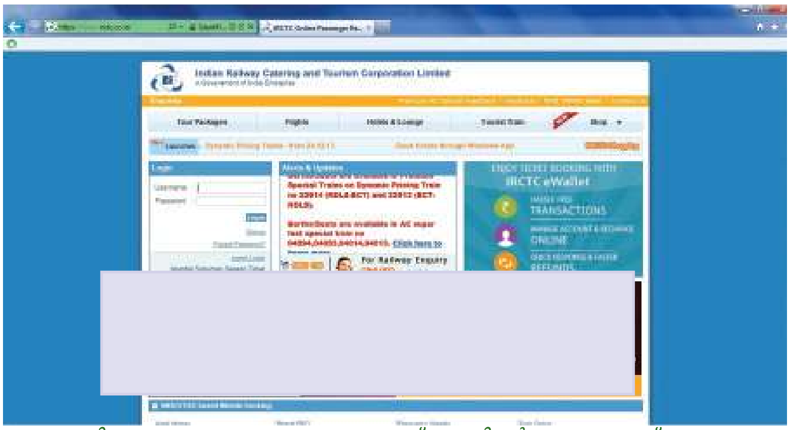
- ജനസംഖ്യയുടെ നല്ലൊരു ശതമാനവും ഐ ടിയുടെ പ്രയോഗത്തെക്കുറിച്ചും ഉപയോഗത്തെ കുറിച്ചും ബോധവാന്മാരല്ല. സ്ഥിരമായി ഇന്റർനെറ്റ് ഉപയോഗിക്കുന്ന ആളുകൾക്ക് പോലും ഓൺലൈൻ വ്യാപാരത്തെ കുറിച്ചും അതിന്റെ സാധ്യതകളെ കുറിച്ചും അറിവില്ല എന്നത് അത്ഭുതകരമാണ്.
- ഇ-വ്യാപാരത്തിന് വളരെ അത്യാവശ്യമായ പ്ലാസ്റ്റിക് പണം എന്നറിയപ്പെടുന്ന ക്രെഡിറ്റ് കാർഡ്, ഡെബിറ്റ് കാർഡ് എന്നിവയും ഇന്റർനെറ്റ് ബാങ്കിങ് സംവിധാനവും ഭൂരിഭാഗം ആളുകളും പ്രത്യേകിച്ച് ഗ്രാമീണ മേഖലയിലുള്ളവർക്ക് പ്രാപ്യമല്ല.
- വേണ്ട മുൻകരുതലുകൾ ഇല്ലാതെ ഉപയോഗിക്കുന്നതുകൊണ്ട് ഉപഭോക്താക്കൾക്ക് അവരുടെ വിലപ്പെട്ട വിവരങ്ങളായ ക്രെഡിറ്റ്/ഡെബിറ്റ് കാർഡ് നമ്പർ, രഹസ്യകോഡ് (Password) എന്നിവ നഷ്ടപ്പെടാൻ സാധ്യതയുണ്ട്.
- വസ്ത്രങ്ങൾ, കരകൗശല വസ്തുക്കൾ, ആഭരണങ്ങൾ തുടങ്ങിയ ഉൽപന്നങ്ങൾ ഭൗതിക സാഹചര്യത്തിൽ പരിശോധിച്ച് വാങ്ങാവുന്നതാണ്. എന്നാൽ ഓൺലൈൻ വ്യാപാരത്തിൽ ഉപഭോക്താവിന് 'സ്പർശിച്ചും അനുഭവിച്ചറിഞ്ഞും' ഉൽപന്നങ്ങൾ വാങ്ങുവാൻ സാധ്യമല്ല.
- ഇ-വ്യാപാരത്തിന്റെ വിജയത്തിനായി വ്യാപാര സ്ഥാപനങ്ങൾ കുറ്റമറ്റതും, കാര്യക്ഷമവുമായ രീതിയിൽ ഉൽപന്നങ്ങൾ വിതരണം ചെയ്യേണ്ടത് അത്യാവശ്യമാണ്.

10.2.6 ഇ-വ്യാപാരത്തിന് പ്രയോജനകരമായ വെബ്സൈറ്റുകൾ (Useful e-Business websites)

ജനപ്രീതിയാർജ്ജിച്ച ചില ഇ-വ്യാപാര വെബ്സൈറ്റുകൾ പട്ടിക 10.2 ൽ കൊടുത്തിരിക്കുന്നു. കൂടാതെ ചില ഇ-വ്യാപാര വെബ്സൈറ്റുകളുടെ പ്രധാന താളുകൾ ചിത്രം 10.4 ലും 10.5 ലും കാണിച്ചിരിക്കുന്നു.

വെബ്സൈറ്റുകളും മേൽവിലാസവും	ഉപയോഗം / സേവനം
www.irctc.co.in	തീവണ്ടി, വിമാന ടിക്കറ്റുകൾ ബുക്ക് ചെയ്യുവാനും, റദ്ദ് ചെയ്യുവാനും, ഹോട്ടലുകളിൽ മുറികൾ ബുക്ക് ചെയ്യുവാനും വേണ്ടിയുള്ള ഇന്ത്യൻ റെയിൽവേ കാറ്ററിംഗ് ആൻഡ് ടൂറിസം ലിമിറ്റഡ് (IRCTC) ന്റെ ഈ വെബ്സൈറ്റ് ഉപയോഗിക്കാം.
www.amazon.com	വാഷിംഗ്ടൺ ആസ്ഥാനമായി പ്രവർത്തിക്കുന്ന അമേരിക്കൻ ഓൺലൈൻ വ്യാപാരി.
www.ebay.in	ഇന്ത്യയിലെ ഏറ്റവും വലിയ ഓൺലൈൻ വ്യാപാര വെബ്സൈറ്റുകളിൽ ഒന്ന്.
www.licindia.in	ഇൻഷുറൻസ് (Insurance) സ്ഥാപനത്തിന്റെ വെബ്സൈറ്റ്.
www.airindia.com	എയർ ഇന്ത്യ വിമാന ടിക്കറ്റുകൾ ബുക്ക് ചെയ്യുന്നതിനുള്ള ഓൺലൈൻ വെബ്സൈറ്റ്.
www.keralartc.com	കെഎസ്ആർടിസി ബസ് ടിക്കറ്റുകൾ ബുക്ക് ചെയ്യുന്നതിനുള്ള വെബ്സൈറ്റ്.
www.bookmyshow.com	സിനിമാശാലകളിലേക്കുള്ള ടിക്കറ്റ് ബുക്ക് ചെയ്യുന്നതിനുള്ള വെബ്സൈറ്റ്.

പട്ടിക 10.2: ചില ഇ-വ്യാപാര വെബ്സൈറ്റുകൾ



ചിത്രം 10.4: www.irctc.co.in വെബ്സൈറ്റിന്റെ പ്രധാന പേജ്

www.irctc.co.i-

IRCTC ഒരു ഭാരത സർക്കാർ സംരംഭമാണ്. ഭക്ഷണം തയ്യാറാക്കി വിതരണം ചെയ്യുക (Cateri-g), വിനോദസഞ്ചാരം (Tourism), തീവണ്ടി, വിമാന ടിക്കറ്റ് ബന്ധപ്പെട്ട പ്രവർത്തനങ്ങൾ ഇത് കൈകാര്യം ചെയ്യുന്നു. ഓൺലൈൻ വഴി ടിക്കറ്റ് ബുക്ക് ചെയ്യാനുള്ള സൗകര്യം ഒരുക്കുന്നതോടൊപ്പം റിസർവേഷൻ നില, തീവണ്ടി സമയം, ഹോട്ടൽ ബുക്കിങ്ങ് എന്നിവയ്ക്കും ഈ വെബ്സൈറ്റ് ഉപയോഗിക്കാം.

ചിത്രം 10.5: www.Amazon.in വെബ്സൈറ്റിന്റെ പ്രധാന പേജ്

www.amazo-.com

ആമസോൺ ലോകത്തിലെ ഏറ്റവും വലിയ ചില്ലറ വ്യാപാരിയാണ്. ആമസോൺ തുടക്കത്തിൽ ഒരു ഓൺലൈൻ പുസ്തകശേഖരമായി ആരംഭിച്ചു. ഇത് പിന്നീട് സോഫ്റ്റ്വെയർ, വീഡിയോ ഗെയിംസ്, ഇലക്ട്രോണിക് ഉപകരണങ്ങൾ, ഗൃഹോപകരണങ്ങൾ, ഭക്ഷണ സാധനങ്ങൾ, കളിപ്പാട്ടങ്ങൾ, ആഭരണങ്ങൾ എന്നിവയുടെ വില്പനക്കാരായി മാറി. ഈ കമ്പനി ഉപഭോക്തൃ ഇലക്ട്രോണിക്സും (co-sumer eletcro-ics) നിർമ്മിക്കുന്നു. വിവിധ രാജ്യങ്ങളിൽ വ്യാപാരം നടത്തുന്നതിനായി ഇവർക്ക് വ്യത്യസ്ത വ്യാപാര വെബ്സൈറ്റുകൾ നിലവിലുണ്ട്. 2013 ൽ **www.amazo-.i-** എന്ന വെബ്സൈറ്റ് ഇന്ത്യയിൽ തുടക്കം കുറിച്ചു.

സ്വയം വിലയിരുത്താം

1. ഓൺലൈൻ ചുറ്റുപാടിൽ വില്പനക്കാരനും വാങ്ങുന്നവനും തമ്മിൽ നടത്തുന്ന സാമ്പത്തിക ഇടപാട് സംവിധാനത്തെ .... എന്ന് അറിയപ്പെടുന്നു.
2. ഇ - വ്യാപാരം നിർവ്വചിക്കുക.
3. ഇ - ബാങ്കിങ് നിർവ്വചിക്കുക.
4. താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന ശരിയോ തെറ്റോ എന്ന് പരിശോധിക്കുക.

   'ഇ - വ്യാപാരം എന്നത് ഇ - വാണിജ്യത്തിന്റെ വിപുലീകരണമാണ്.

10.3 ഇ–പഠനം (e-Learning)

അധ്യാപകർ, വിദ്യാർത്ഥികൾ, പഠന ഉപകരണങ്ങൾ മുതലായവ അടങ്ങുന്ന ഒരു ക്ലാസ് മുറി നമുക്കെല്ലാവർക്കും സുപരിചിതമാണ്. എന്നാൽ ഒരു സാങ്കൽപിക ക്ലാസ് മുറിയെ പറ്റി ചിന്തിച്ചുനോക്കൂ, അവിടെ അധ്യാപകരും വിദ്യാർത്ഥികളും വ്യത്യസ്തമായ സ്ഥലങ്ങളിലോ ഒരു പക്ഷേ പല രാജ്യങ്ങളിലോ ആയിരിക്കും ഉണ്ടാകുക. ഇത് സാധ്യമാകുമോ? അതെ അത് സാധ്യമാണ്. ഇ-പഠനം വഴി സാമ്പ്രദായിക പഠന രീതിയിലുള്ള പല

പരിമിതികളെയും മറികടക്കുവാൻ കഴിയുന്നു. ഇലക്ട്രോണിക് മാധ്യമങ്ങളെയും, ഐ സി ടിയെയും ഉൾക്കൊള്ളിച്ചുകൊണ്ടുള്ള പഠനരീതിയെയാണ് ഇ-പഠനം എന്ന് വിശേഷിപ്പിക്കുന്നത്.

10.3.1 ഇ-പഠനത്തിന് ഉപയോഗിക്കുന്ന ഉപകരണങ്ങൾ (e-Learning tools)

ഇ-പഠനം മെച്ചപ്പെടുത്തുന്നതിനായി നിരവധി ഉപകരണങ്ങൾ ഉപയോഗിക്കുന്നുണ്ട്. ഇ-പഠന ഉപകരണങ്ങൾ താഴെ ചർച്ച ചെയ്യുന്നു.

a. ഇലക്ട്രോണിക് ബുക്ക് റീഡർ (e-Book Reader)

ഒരു വലിയ ബാഗും അതിൽ നിറയെ പുസ്തകങ്ങളുമായി സ്കൂളിൽ പോകുന്നതിന് പകരം നമുക്ക് ആവശ്യമായ എല്ലാ വിവരങ്ങളും ഒരു ചെറിയ ഉപകരണത്തിൽ സൂക്ഷിച്ച് സ്കൂളിൽ പോകുന്നത് ചിന്തിച്ചു നോക്കൂ? അത്! രസകരമാണ് അല്ലേ? ഒരുപാട് പുസ്തകങ്ങളിലായി അടങ്ങിയിട്ടുള്ള ആശയങ്ങളെ ഡിജിറ്റൽ രൂപത്തിലാക്കി സൂക്ഷിച്ചുവെക്കുവാനും കയ്യിൽ കൊണ്ട് നടക്കുവാനും കഴിയുന്ന ഉപകരണമാണ് ഇ-ബുക്ക് റീഡർ. ഇ-ബുക്ക് റീഡറിൽ ഏത് പുസ്തകവും അതിന്റെ ഏതു പേജും അനായാസേന നമുക്ക് തുറക്കാം. ഈ ഉപകരണം ഉപയോഗിച്ച് ഇന്റർനെറ്റിൽ നിന്ന് വെബ് പ്രമാണങ്ങൾ ഡൗൺലോഡ് ചെയ്ത് എപ്പോൾ വേണമെങ്കിലും വായിക്കുവാനും ഓൺലൈനായി പണമടയ്ക്കുവാനും സാധിക്കുന്നു. ഇ-ബുക്ക് റീഡറിലെ ഓഡിയോ ഔട്ട്പുട്ട് സംവിധാനം ഉപയോഗപ്പെടുത്തി പാഠഭാഗങ്ങൾ വായിച്ച് കേൾക്കുവാനും കഴിയുന്നു.

b. ഇ-ടെക്സ്റ്റ്/e-Text (ഇ-പുസ്തകം)

പുസ്തകത്തിലുള്ള വിവരങ്ങൾ ഇലക്ട്രോണിക് രൂപത്തിൽ ലഭ്യമാക്കുന്നതിനെയാണ് ഇ-ടെക്സ്റ്റ് (ഇ-പുസ്തകം) എന്നു പറയുന്നത്. ഇ-ബുക്ക് റീഡറോ, കമ്പ്യൂട്ടറോ, മറ്റേതെങ്കിലും ഇലക്ട്രോണിക് ഉപകരണമോ ഉപയോഗിച്ച് ഇ-ടെക്സ്റ്റ് വായിക്കുവാനാകും. നമുക്ക് താൽപര്യമുള്ള രീതിയിൽ ടെക്സ്റ്റിന്റെ രൂപഘടനയിൽ സോഫ്റ്റ്വെയറുകൾ ഉപയോഗിച്ച് മാറ്റം വരുത്താവുന്നതാണ്. ഒരു കമ്പ്യൂട്ടറിന്റെയോ ഇ-ടെക്സ്റ്റ് റീഡറിന്റെയോ സഹായത്തോടെ ഇ-ടെക്സ്റ്റ് വായിച്ച് കേൾക്കുവാനാകും. ഇത് കാഴ്ച ശക്തി കുറഞ്ഞവർക്ക് വളരെയധികം സഹായകരമാണ്.

c. ഓൺലൈൻ ചാറ്റ് (Online chat)

ഇന്റർനെറ്റ് ഉപയോഗിച്ച് രണ്ടോ അതിലധികമോ വ്യക്തികൾ തമ്മിൽ ടെക്സ്റ്റ്, രൂപത്തിൽ സന്ദേശങ്ങൾ കൈമാറുന്ന രീതിയാണ് ഓൺലൈൻ ചാറ്റ്. സാങ്കൽപിക ക്ലാസ് മുറികളിൽ അധ്യാപകരും വിദ്യാർത്ഥികളും തമ്മിൽ വിവിധ വിഷയങ്ങളെകുറിച്ച് ചർച്ച നടത്തുന്നത് ഓൺലൈൻ ചാറ്റിങ്ങിലൂടെയാണ്. ഇന്റർനെറ്റിന്റെ വേഗത കുറവാണെങ്കിലും ടെക്സ്റ്റ് രൂപത്തിലുള്ള ഓൺലൈൻ ചാറ്റിങ്ങ് നടത്താവുന്നതാണ്. പരസ്പരം കണ്ട് സംസാരിക്കുന്നതിനും ഓൺലൈൻ ചാറ്റിങ്ങിൽ സാധ്യമാണ്. ഇതിന് വേഗതയേറിയ ഇന്റർനെറ്റ് ബന്ധം, വെബ് ക്യാമറ, സ്പീക്കർ തുടങ്ങിയ ഉപകരണങ്ങളും ആവശ്യമാണ്. ഓൺലൈൻ ചാറ്റ് സാമൂഹിക ഇടപെടലുകൾക്കുള്ള സാധ്യതയും ഒരുക്കുന്നു. രണ്ട് വ്യക്തികൾ തമ്മിൽ വിവിധ സ്ഥലങ്ങളിൽ ഇരുന്ന് ആശയവിനിമയം നടത്തുന്നതിന് ഇത് സഹായിക്കുന്നു. ഈ സംവിധാനം ഉപയോഗിച്ചാണ് അധ്യാപകരുടെ ഓൺലൈനിലൂടെയുള്ള തത്സമയക്ലാസുകൾ ലഭ്യമാകുന്നത്. വിദ്യാർത്ഥികൾക്ക് വിദൂര സ്ഥലങ്ങളിൽ ഇരുന്ന് ലോഗിൻ ചെയ്ത് ക്ലാസുകളിൽ പങ്കെടുക്കാം.

d. ഇ-കണ്ടന്റ് (e-Content)

വീഡിയോകൾ, അവതരണങ്ങൾ, ഗ്രാഫിക്സ്, ആനിമേഷൻ എന്നിവ ഉൾപ്പെടുത്തി ഇ-കണ്ടന്റ് നിർമ്മിക്കുകയും അതുവഴി ഇ-പഠനം സാധ്യമാക്കുകയും ചെയ്യാം. ഇവ ടെലിവിഷനിലൂടെയോ, വെബ്സൈറ്റുകൾ മുഖേനയോ പ്രക്ഷേപണം ചെയ്യുവാനും സാധിക്കുന്നു. അപ്‌ലോഡ് ചെയ്ത ഉള്ളടക്കങ്ങൾ പിന്നീട് വെബ്സൈറ്റുകളിൽ നിന്ന് ഡൗൺലോഡ് ചെയ്യുവാനും എപ്പോൾ വേണമെങ്കിലും ഉപയോഗിക്കുവാനും സാധിക്കുന്നു. ഉള്ളടക്കങ്ങൾ എളുപ്പത്തിൽ ജനങ്ങളിലേക്ക് എത്തിക്കുവാൻ മൾട്ടി മീഡിയ ഉള്ളടക്കങ്ങൾ വഴി സാദ്ധ്യമാക്കുന്നു. സാമ്പ്രദായിക ക്ലാസ് മുറികളിൽ അസാധ്യമായത് കൃത്യമായി നിർമിച്ച് കാണിക്കുവാനും ഇത് വഴി സാധ്യമാകുന്നു. കേരള സർക്കാരിന്റെ സമഗ്ര പോർട്ടൽ ഇ-കണ്ടന്റിന് ഒരു ഉദാഹരണമാണ്. (www.samagra.kite.kerala. gov.in)

e. വിദ്യാഭ്യാസ ടി വി ചാനലുകൾ (Educational TV Channels)

ഇ-പഠനം സാധ്യമാക്കുന്നതിന് നിരവധി ടി വി ചാനലുകളും വെബ്ചാനലുകളും ലഭ്യമാണ്. ഈ ചാനലുകൾ ഉപയോഗിച്ച് വിവിധ വിഷയങ്ങളിലെ ക്ലാസുകൾ, വിദഗ്ധരുമായി നടത്തിയ അഭിമുഖങ്ങൾ, ലാബുകളിൽ നടത്തിയ പരീക്ഷണങ്ങൾ എന്നിവ സംപ്രേഷണം ചെയ്യുവാൻ സാധിക്കുന്നു. ഇവയിൽ ചില ചാനലുകൾ ഇന്റർനെറ്റിലും ലഭ്യമാണ്. ദൂരദർശന്റെ വ്യാസ്, കേരളാ ഗവൺമെന്റിന്റെ വിക്ടേഴ്സ് തുടങ്ങിയവ വിദ്യാഭ്യാസ ടെലിവിഷൻ ചാനലുകൾക്ക് ഉദാഹരണങ്ങളാണ്.

10.3.2 ഇ-പഠനത്തിന്റെ മേൻമകൾ (Advantages of e-Learning)

ഇ-പഠനത്തിന് ധാരാളം മേൻമകളുണ്ട്. അവ താഴെ കൊടുത്തിരിക്കുന്നു.

- വിദൂര സ്ഥലങ്ങളിലുള്ള വിദ്യാർത്ഥികൾക്ക് വൈവിധ്യമാർന്നതും ഗുണമേൻമയുള്ളതുമായ വിവിധ വിഷയങ്ങളിലുള്ള കോഴ്സുകൾ ഇ-പഠനം വഴി പഠിക്കുവാൻ കഴിയുന്നു.
- ഇ-പഠനത്തിന് ചെലവ് കുറവാണ്. ഇത് വഴി യാത്രാസമയവും, അധ്യാപക വേതനവും ലാഭിക്കാം.
- പരിമിതമായ സാമ്പത്തിക സ്ഥിതിയുള്ളവർക്ക് പോലും കുറഞ്ഞ ചിലവുള്ള ഇ-പഠനം സാധ്യമാണ്.
- ഈ സംവിധാനം ഉപയോഗിച്ച് തദ്ദേശീയവും, അന്തർദേശീയവുമായ സ്ഥാപനങ്ങളിൽ നിന്നും ഓൺലൈൻ കോഴ്സുകൾ പഠിക്കുന്നതിന് അവസരം ലഭിക്കുന്നു.
- സ്ഥലവും, സമയവും ഇ-പഠനത്തിന് തടസമാകുന്നില്ല.

10.3.3 ഇ-പഠനത്തിന്റെ വെല്ലുവിളികൾ (Challanges to e-Learning)

ഇ-പഠനം നേരിടുന്ന ചില വെല്ലുവിളികൾ താഴെ പറയുന്നവയാണ്.

- അധ്യാപകനും വിദ്യാർത്ഥിയും തമ്മിൽ മുഖാമുഖമുള്ള പഠനം സാധ്യമല്ല.

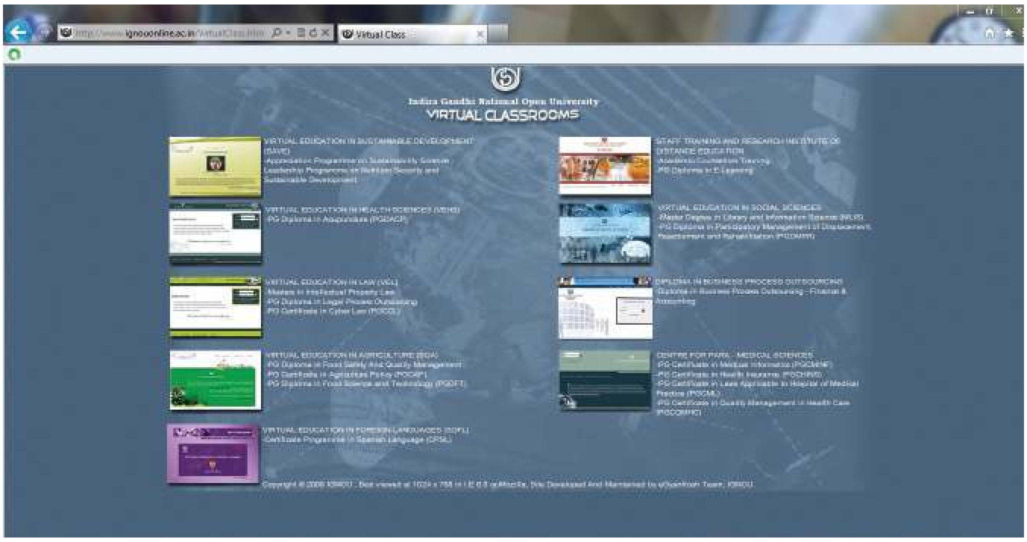
- അടിസ്ഥാന സൗകര്യങ്ങളുടെ പരിമിതി കാരണം അധ്യാപകരും വിദ്യാർത്ഥികളും തമ്മിലുള്ള ഇടപെടലുകൾ പരിമിതപ്പെടുത്തുന്നു.
- ഉപകരണങ്ങളുടെയും സാങ്കേതികവിദ്യയുടെയും (കമ്പ്യൂട്ടറും വേഗതയേറിയ ഇന്റർനെറ്റും) ആവശ്യകത ഇ-പഠനത്തിന് തടസമാകുന്നു.
- നിരന്തരമായ പ്രോത്സാഹനം ലഭിക്കേണ്ട വിദ്യാർത്ഥികൾക്ക് അത് വേണ്ട രീതിയിൽ ലഭിക്കാതെ പോകുന്നു.
- ലബോറട്ടറികളുടെ അഭാവം കാരണം പ്രായോഗിക പഠനത്തെ പരിമിതപ്പെടുത്തുന്നു.

10.3.4 ഇ-പഠനത്തിന് പ്രയോജനകരമായ വെബ് സൈറ്റുകൾ (Useful e-Learning Website)

ഇ-പഠനത്തിന് ഉപകരിക്കുന്ന വെബ്സൈറ്റുകളുടെ ഒരു പട്ടിക 10.3 ൽ സംഗ്രഹിച്ചിരിക്കുന്ന ചിത്രം 10.6 ൽ ഇഗ്നോയുടെ (ഇന്ദിരാഗാന്ധി നാഷണൽ ഓപ്പൺ യൂണിവേഴ്സിറ്റി) പ്രധാന പേജാണ് കാണിച്ചിരിക്കുന്നത്.

വെബ്സൈറ്റുകളും മേൽവിലാസവും	ഉപയോഗം / സേവനം
www.ignouonline.ac.in	വിവിധ കോഴ്സുകളിൽ വിദൂര വിദ്യാഭ്യാസം നൽകുന്ന ഓപ്പൺ യൂണിവേഴ്സിറ്റികളിൽ ഒന്നായ ഇന്ദിരാഗാന്ധി ഓപ്പൺ യൂണിവേഴ്സിറ്റിയുടെ ഔദ്യോഗിക വെബ്സൈറ്റാണിത്.
www.nptel.iitm.ac.in	എഞ്ചിനിയറിംഗ്, സയൻസ്, ഹ്യുമാനിറ്റീസ് വിഷയങ്ങളുടെ ഇ- കണ്ടെന്റുകളും വീഡിയോകളും തയ്യാറാക്കി ഓൺലൈനായി കോഴ്സുകൾ നൽകുന്ന നാഷണൽ പ്രോഗ്രാം ഓൺ ടെക്നോളജി എൻഹാൻസ്ഡ് ലേണിങ്ങിന്റെ വെബ്സൈറ്റാണിത്.
www.ncert.nic.in	നാഷണൽ കൗൺസിൽ ഓഫ് എഡ്യുക്കേഷണൽ റിസേർച്ച് ആൻഡ് ട്രെയിനിങ്ങിന്റെ വെബ്സൈറ്റാണിത്. വിവിധ പഠനസഹായികളും, സംശയനിവാരണ സാമഗ്രികളും ഹിന്ദി, ഇംഗ്ലീഷ് എന്നീ ഭാഷകളിൽ ഈ വെബ്സൈറ്റിൽ ഉൾപ്പെടുത്തിയിരിക്കുന്നു.
www.spoken_tutorial.org	മുംബൈ ഐ.ഐ.ടിയുടെ ഒരു സംരഭമാണിത്. ഇവ സ്പോക്കൺ ട്യൂട്ടോറിയൽസ് വഴി സോഫ്റ്റ്‌വെയർ പരിശീലനം സാധ്യമാക്കുന്നു. എല്ലാ കോഴ്സുകളും കേന്ദ്ര സർക്കാരിന്റെ പൂർണ്ണസഹായത്തോടുകൂടി സൗജന്യമായി നടത്തപ്പെടുന്നു.
www.w3schools.com	വെബ്സൈറ്റുകൾ നിർമ്മിക്കുന്നതിനാവശ്യമായ HTML, CSS, JavaScript, PHP, SQL, JQuery തുടങ്ങിയവ പഠിക്കുന്നതിന് ആവശ്യമായ പഠനസഹായികളും സംശയനിവാരണ സാമഗ്രികളും w3schools ൽ ലഭ്യമാണ്.
www.gcflearnfree.org	ഗുഡ്‌വിൽ കമ്മ്യൂണിറ്റി ഫൗണ്ടേഷൻ സഹായത്തോടുകൂടി പ്രവർത്തിക്കുന്ന ഒരു വെബ്സൈറ്റാണ് gcflearnfree.org. ഗണിതം, ശാസ്ത്രം, സാങ്കേതികവിദ്യ, ഇംഗ്ലീഷ് മുതലായ വിഷയങ്ങളിലെ പാഠഭാഗങ്ങൾ വളരെ ലളിതമായി ഇതിൽ കൊടുത്തിരിക്കുന്നു. ആർക്കും എവിടെ വൈദഗ്ധ്യവും അറിവുനേടുന്നതിന് ഇത് സഹായിക്കുന്നു.

പട്ടിക 10:3 ചില ഇ- പഠന വെബ്സൈറ്റുകൾ



ചിത്രം. 10:6 IGNOU വെബ്സൈറ്റിന്റെ പ്രധാനപേജ്.

www.ig-ouo-li-e.ac.i-

ഭാരത സർക്കാരിന്റെ കീഴിൽ പ്രവർത്തിക്കുന്ന ഒരു ഓപ്പൺ സർവ്വകലാശാലയാണ് ഇഗ്നോ (ഇന്ദിരാഗാന്ധി നാഷണൽ ഓപ്പൺ യൂണിവേഴ്സിറ്റി). ഈ വെബ്സൈറ്റിലൂടെ **eGya-kosh** പോലുള്ള മികച്ച സേവനങ്ങൾ നൽകിവരുന്നു. പഠനകുറിപ്പുകളുടെയും വീഡിയോകളുടെയും ഒരു ഡിജിറ്റൽ കലവറയാണ് **eGya-kosh**. ഗ്യാൻദർശൻ **(Gya-darsha-)**, ഗ്യാൻവാണി **(Gya-va-i)**, എഡ്യുസാറ്റ് **(EDUSAT)** മുതലായ വിദ്യാഭ്യാസ ചാനലുകളും ഈ വെബ്സൈറ്റ് ഉപയോഗിച്ച് സംപ്രേഷണം ചെയ്യുന്നു. ഇത് കൂടാതെ സർവ്വകലാശാല നടത്തിവരുന്ന എല്ലാ ഓൺലൈൻ പ്രോഗ്രാമുകളും സാങ്കൽപ്പിക ക്ലാസ് മുറികളിലൂടെ ലഭ്യമാണ്.

ചിത്രം. 10.7 NPTEL വെബ്സൈറ്റിന്റെ പ്രധാന പേജ്

www.-ptel.iitm.ac.i-

നാഷണൽ പ്രോഗ്രാം ഓൺ ടെക്നോളജി എൻഹാൻസ്ഡ് ലേണിങ് എന്നതിന്റെ ചുരുക്ക പേരാണ് NPTEL. ഇത് ഏഴ് ഐ.ഐ.ടികളുടെയും ഐ.ഐ.എസ്.സിയുടെയും സംയുക്ത സംരംഭമാണ്. എഞ്ചിനിയറിങ്, സയൻസ് വിഷയങ്ങൾക്ക് ആവശ്യമായ പഠന സാമഗ്രികൾ നിർമ്മിക്കുക എന്നതാണ് ഇതിന്റെ ലക്ഷ്യം. ഇത്തരം വെബ്സൈറ്റുകൾ വെബ് അധിഷ്ഠിത പഠന സാമഗ്രികൾ നിർമ്മിക്കുകയും അവ സൗജന്യമായി ലഭ്യമാക്കുകയും ചെയ്യുന്നു. വീഡിയോ ഉൾപ്പെടുത്തികൊണ്ടുള്ള അധ്യാപനവും ഇത് വഴി സാധ്യമാണ്.

സ്വയം വിലയിരുത്താം

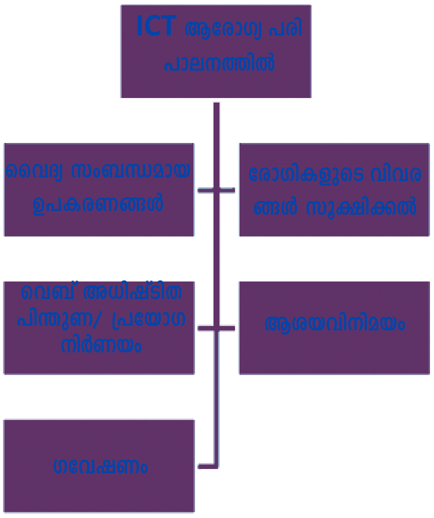
1. ഇന്റർനെറ്റ് ഉപയോഗിച്ച് രണ്ടോ അതിലധികമോ വ്യക്തികൾ തമ്മിൽ ടെക്സ്റ്റ് രൂപത്തിലുള്ള സന്ദേശങ്ങൾ കൈമാറുന്ന രീതിയാണ്.................
2. വേറിട്ടു നിൽക്കുന്നത് ഏത്?
   a. ഇ - ബുക്ക് റീഡർ   b. ഇ - ടെക്സ്റ്റ്
   c. ടെലിവിഷൻ ചാനൽ   d. ഇ - ബിസിനസ്
3. ഇ - ടെക്സ്റ്റ് നിർവ്വചിക്കുക.
4. ഇ - ലേണിങ് ഉപകരണത്തിന് ഒരു ഉദാഹരണം നൽകുക?

10.4 ആരോഗ്യ പരിപാലനത്തിൽ ഐ സി ടി യുടെ ഉപയോഗങ്ങൾ (ICT Applications in Health care)

ഐ സി ടിയുടെ വളർച്ച വൈദ്യപഠനത്തേയും വൈദ്യപരിചരണത്തേയും വളരെയധികം സ്വാധീനിച്ചിട്ടുണ്ട്. ഐ.ടി ആരോഗ്യപരിപാലനത്തിന് ഉപയോഗിക്കുന്നതിലൂടെ വളരെ അധികം ചിലവ് കുറക്കുന്നതിനും സേവനം മെച്ചപ്പെടുന്നതിനും ഒട്ടേറെ അവസരങ്ങൾ ഉണ്ട്. നമ്മൾ അടുത്തതായി ചർച്ച ചെയ്യുന്നത് ആരോഗ്യപരിപാലനത്തിലും വൈദ്യശാസ്ത്ര മേഖലയിലും കൊണ്ടുവന്നിട്ടുള്ള മാറ്റങ്ങൾ നമുക്ക് ചർച്ച ചെയ്യാം.

10.4.1 വൈദ്യസംബന്ധമായ ഉപകരണങ്ങൾ (MEDICAL EQUIPMENTS)

മിക്കവാറും എല്ലാ വൈദ്യോപകരണങ്ങളും പ്രവർത്തിക്കുന്നത് കമ്പ്യൂട്ടറിന്റെ സഹായത്തോടുകൂടിയാണ്. സ്കാനറുകളായ എം ആർ ഐയും, സി. ടിയും പ്രവർത്തിക്കുന്നത് കമ്പ്യൂട്ടറിനെ ആശ്രയിച്ചാണ്. കമ്പ്യൂട്ടറിന്റെ സഹായത്താലാണ് ഇവ ഡാറ്റ പ്രോസസ്സ് ചെയ്ത് വിവരങ്ങൾ ആക്കുന്നു. ഇതിന് സഹായകരമായ ഉപകരണങ്ങളെ മൈക്രോപ്രൊസസർ ഉപയോഗിച്ചാണ് നിയന്ത്രിക്കുന്നത്. കമ്പ്യൂട്ടർ നിയന്ത്രിത ലേസറുകളാണ് ബ്രെയിൻ ട്യൂമർ നീക്കം ചെയ്യുന്നതുൾപ്പെടെയുള്ള ശസ്ത്രക്രിയകൾക്ക് ഉപയോഗിക്കുന്നത്. രോഗികൾക്ക് സാധാരണജീവിതം നയിക്കുവാൻ ഉതകുന്ന തര



ചിത്രം. 10.8: ഐ.സി.റ്റി. ആരോഗ്യപരിപാലനത്തിൽ

ത്തിലുള്ള കൈയ്യിൽ കൊണ്ട് നടക്കാവുന്ന ഉപകരണങ്ങളും ലഭ്യമാണ്. പ്രമേഹ രോഗികൾക്ക് രക്തത്തിലെ പഞ്ചസാരയുടെ അളവ് അറിയുന്നതിനുപയോഗിക്കുന്ന ഉപകരണം ഇതിന് ഉദാഹരണമാണ്.

10.4.2 ഇലക്ട്രോണിക് മെഡിക്കൽ റെക്കോർഡ് (Electronic Medical Record -EMR)

എല്ലാ രോഗികളുടെയും വൈദ്യസംബന്ധമായ രേഖകളും ഡിജിറ്റൽ രൂപത്തിലേക്ക് മാറ്റാൻ ഒരു നീക്കമുണ്ട്. ഇത് ഇലക്ട്രോണിക് മെഡിക്കൽ റെക്കോർഡ് എന്ന് അറിയപ്പെടുന്നു. ഡോക്റ്റർമാരാണ് ഇത് ഏറ്റവും കൂടുതലായി ഉപയോഗിക്കുന്നത്. രോഗനിർണയത്തിനും ചികിത്സയ്ക്കും വേണ്ടിയാണ്. ഇങ്ങനെ രോഗിയുടെ എല്ലാ വിവരങ്ങളും ഇലക്ട്രോണിക് രൂപത്തിലേക്ക് മാറ്റുന്നത് കൊണ്ട് ധാരാളം ഗുണങ്ങളുണ്ട്. അവയിൽ ചിലത് താഴെ വിവരിക്കുന്നു.

- രോഗിയുടെ എല്ലാ വിവരങ്ങളും സൂക്ഷിച്ചുവെക്കുന്നതിനുള്ള ഏറ്റവും ചിലവ് കുറഞ്ഞ രീതിയാണിത്.
- രോഗിയുടെ ചികിത്സാ ചരിത്രം വളരെ വേഗത്തിലും എളുപ്പത്തിലും ലഭ്യമാകുന്നു.
- രോഗിയുടെ എല്ലാ രേഖകളും വളരെ വേഗത്തിൽ മറ്റൊരു ഡോക്ടറിലേക്കും ഹോസ്പിറ്റലിലേക്കും മാറ്റുവാൻ കഴിയുന്നു.
- രോഗിയുടെ രോഗാവസ്ഥയെ നിരന്തരം വിശകലനം ചെയ്യുവാൻ സാധിക്കുന്നു.
- മരുന്ന് കുറിപ്പ് കൃത്യമായി വായിക്കുവാനും അത് ഭാവിയിലെ പരിശോധനകൾക്ക് വേണ്ടി സൂക്ഷിച്ചുവെക്കുവാനും സാധിക്കുന്നു.

10.4.3. രോഗ നിർണയത്തിനും വൈദ്യസഹായത്തിനുമുള്ള വെബ് അധിഷ്ഠിത സേവനങ്ങൾ (Web-bas\ed support/diagnosis)

വിവിധതരം രോഗങ്ങൾ, അവയ്ക്കുള്ള ചികിത്സകൾ, മറ്റ് ചികിത്സാരീതികൾ, വൈദ്യസംബന്ധമായ ലേഖനങ്ങൾ,വിവിധ അവസ്ഥയിലുള്ള രോഗികളുടെ വിവരങ്ങൾ, എന്നിവയെല്ലാം ഇന്റർനെറ്റിലൂടെ ലഭ്യമാകുന്നു.

ചില വെബ്സൈറ്റുകൾ രോഗ ലക്ഷണങ്ങൾ കണ്ടെത്തുവാൻ സഹായിക്കുന്നതിനാൽ, ഡോക്ടറുടെ സഹായമില്ലാതെ തന്നെ ആളുകൾക്ക് അവരുടെ ആരോഗ്യ പ്രശ്നങ്ങളെ കുറിച്ചുള്ള പ്രാഥമിക അറിവ് ലഭിക്കുന്നു. എന്നാൽ ഇതിനു ചില അപകടസാധ്യതതൾ ഉണ്ട്. ഇത് തെറ്റായ രോഗനിർണയത്തിലേക്ക് നയിക്കുകയും ഉത്കണ്ഠയ്ക്കും മാനസിക സംഘർഷത്തിലേക്കും നമ്മെ നയിക്കുന്നു.

10.4.4 ടെലി മെഡിസിൻ (Telemedicine)

ടെലിമെഡിസിൻ ഉപയോഗിച്ച് ആരോഗ്യമേഖലയിലെ ഏറ്റവും പരിചയസമ്പന്നരായ ഡോക്ടറുടെ കണ്ടെത്തലുകൾ പങ്കുവെയ്ക്കുവാനും, നിർദ്ദേശങ്ങൾ സ്വീകരിക്കുവാനും സാധിക്കുന്നു. ശസ്ത്രക്രിയ നടക്കുന്ന സമയത്തുപോലും ഒരു ഡോക്ടർക്ക് വളരെ അകലെയുള്ള മറ്റൊരു ഡോക്ടറിൽ നിന്ന് വിദഗ്ധോപദേശം തേടാനാകും. ഇത് വഴി സമയവും പണവും ലാഭിക്കാം. അത്യാസന്ന നിലയിലുള്ള ഒരു രോഗിയുടെ ശസ്ത്രക്രിയയ്ക്ക്

ഡോക്ടറുടെ അസാന്നിദ്ധ്യത്തിൽ പോലും, അദ്ദേഹത്തിന്റെ വിദഗ്ധോപദേശം ശസ്ത്ര ക്രിയാ മുറിയിൽ ലഭ്യമാക്കുവാൻ കഴിയും. ജീവനക്കാർക്കും, ഗതാഗത സൗകര്യമൊരുക്കുന്നവർക്കും അത്യാഹിത ഘട്ടങ്ങളിൽ ആശുപത്രിയുമായി ബന്ധപ്പെടുവാൻ വയർലെസ്സ് സംവിധാനം സഹായകരമാകുന്നു.

10.4.5 ഗവേഷണവും പുരോഗതിയും (Research and development)

ഈ അടുത്ത കാലം വരെ പുതിയ മരുന്നുകളുടെ കണ്ടുപിടുത്തം എന്നത് ഏതൊരു മരുന്ന് കമ്പനിയെ സംബന്ധിച്ചിടത്തോളവും ഉത്കണ്ഠ ഉളവാക്കുന്നതായിരുന്നു. അനിയന്ത്രിതമായ വിലക്കയറ്റവും വർദ്ധിച്ചുവരുന്ന സങ്കീർണതയും പരമ്പരാഗത ഗവേഷണരീതികളിൽ മാറ്റം വരുത്തേണ്ടതായി വന്നു. ഇന്ന് ജൈവ സാങ്കേതികവിദ്യയെയും കേന്ദ്രീകൃത വിവരസമ്പ്രദായവും കൂട്ടിചേർത്ത് മരുന്ന് നിർമ്മാണ മേഖലയിൽ വിപ്ലവകരമായ മാറ്റങ്ങൾ വരുത്തുകയും ഏറ്റവും മികച്ച കമ്പ്യൂട്ടർ സാങ്കേതിക വിദ്യ ഉപയോഗിച്ച് വ്യത്യസ്ഥ ആവശ്യങ്ങൾക്കുള്ള മരുന്നുകൾ നിർമ്മിക്കുകയും ചെയ്യുന്നു. ഐ ടിയുടെ ഉപയോഗം വരും വർഷങ്ങളിൽ മരുന്ന് ഗവേഷണ രംഗത്ത് ലാഭകരമായ ഒരു മാറ്റം ഉണ്ടാക്കിയേക്കാം. ഇങ്ങനെയെല്ലാം ഐ ടി രോഗനിർണയത്തിനും, ചികിത്സയ്ക്കും, ഹോസ്പിറ്റലിന്റെ സുഖമമായ നടത്തിപ്പിനുമെല്ലാം സഹായിക്കുന്നു.

10.5 ഐ സി ടി അധിഷ്ഠിത സേവനങ്ങൾ (ICT enabled services)

ഐ സി ടി അധിഷ്ഠിത സേവനങ്ങളെ വെബ് അധിഷ്ഠിത സേവനങ്ങൾ എന്നും പറയുന്നു. ഇതിൽ ഐ സി ടിയുടെ സഹായത്തോടുകൂടി വ്യത്യസ്ഥ തലത്തിലുള്ള പ്രവർത്തനങ്ങൾ ഉൾപ്പെടുത്തി ഒരു സ്ഥാപനം നൽകുന്ന സേവനങ്ങൾ മെച്ചപ്പെടുത്തുകയും ചെയ്യുന്നു. ഈ സേവനങ്ങൾ കോൾ സെന്റർ , ബി പി ഒ, മെഡിക്കൽ ട്രാൻസ്ക്രിപ്ഷൻ എന്നിവ ഉൾപ്പെടെയുള്ള മേഖലകളിൽ ജോലി സാധ്യത ലഭ്യമാക്കുന്നു. ഏറ്റവും പ്രധാനപ്പെട്ട ചില ഐ.സി.ടി സേവനങ്ങളാണ് ഇനിയുള്ള ഭാഗങ്ങളിൽ ചർച്ച ചെയ്യുന്നത്.

10.5.1 ബിസിനസ് പ്രോസസ്സ് ഔട്ട്സോഴ്സിങ് (Business Process Outsourcing (BPO))

വ്യപാരമേഖലയിൽ ഉപയോഗിക്കുന്ന ഒരു പദമാണ് പുറംജോലി കരാർ (ഔട്ട്സോഴ്സിംങ്). ഒരു കമ്പനിയുടെ ചെലവ് കുറയ്ക്കുന്നതിന് വേണ്ടി ചില ജോലികൾ മറ്റൊരു കമ്പനിയെ ഏൽപ്പിക്കുന്ന രീതിയെയാണ് പുറംജോലി കരാർ എന്ന് പറയുന്നത്. ബി പി ഒ എന്നു പറയുന്നത് പുറംജോലി കരാറിന്റെ ഒരു ഭാഗമാണ്. ഒരു കമ്പനിയുടെ പ്രവർത്തനങ്ങളെയും ഉത്തരവാദിത്തങ്ങളെയും മറ്റൊരു കമ്പനിക്ക് കരാർ കൊടുത്തുകൊണ്ടുള്ള രീതിയാണിത്. ഒരു സ്ഥാപനത്തിൽ നിന്ന് മറ്റൊരു സ്ഥാപനത്തിലേക്ക് ജോലിക്കാരെയോ ആസ്തികളെയോ മാറ്റുന്ന പ്രവർത്തനങ്ങൾ പുറംജോലി കരാറിൽ ഉൾപ്പെടുന്നു.

ഒരു സ്ഥാപനം, ബി പി ഒ സംവിധാനം നടപ്പിൽ വരുത്തുന്നത്, അവരുടെ കാര്യക്ഷമത വർദ്ധിപ്പിക്കുവാനും, ഉൽപാദന ചെലവ് കുറയ്ക്കുവാനുമാണ്. ബിപിഒ ഉപയോഗപ്പെടുത്തി നടപ്പിൽ വരുത്തുന്ന ചില പ്രവർത്തനങ്ങളാണ് ഹ്യൂമൺ റിസോഴ്സ് മാനേജ്മെന്റ് (Human Resource Management), ഉപഭോക്തൃ സേവന കേന്ദ്രങ്ങൾ, ശമ്പള പട്ടിക (Payroll) തുടങ്ങിയവ.

10.5.2 നോളജ് പ്രോസസ് ഔട്ട് സോഴ്സിംങ് (Knowledge Process Outsourcing (KPO))

പുറംജോലി കരാറിന്റെ മറ്റൊരു രൂപമാണ് നോളജ് പ്രോസസ് ഔട്ട് സോഴ്സിങ്. ഇവിടെ വിജ്ഞാനവും, അറിവും വ്യത്യസ്ത സ്ഥാപനങ്ങളിലേക്കോ, അല്ലെങ്കിൽ അതേ സ്ഥാപനത്തിന്റെ വേറെ ഭാഗങ്ങളിലേക്കോ പുറംജോലി കരാർ ചെയ്യാവുന്നതാണ്. ഇത് ചെലവും വിഭവങ്ങളും കുറയ്ക്കുവാൻ സഹായിക്കുന്നു. കെ പി ഒയുടെ ചില പ്രധാനപ്പെട്ട പ്രക്രിയകളാണ് ഡാറ്റയെ കണ്ടെത്തൽ, ഏകീകരണം, കമ്പോള വിലയിരുത്തൽ, വഞ്ചനാഅപഗ്രഥനം തുടങ്ങിയവ. കെ പി ഒ മേഖലയിൽ അഭൂതപൂർവ്വമായ വളർച്ചയാണ് സമീപകാലത്ത് ഉണ്ടായിട്ടുള്ളത്. വ്യവസായങ്ങൾക്ക് കെ പി ഒ വളരെ കൂടുതലായി ഉപയോഗിക്കുന്നുണ്ട്. നിയമം, സാമ്പത്തിക സേവനങ്ങൾ, ഭരണം, വിദഗ്ധോപദേശം, തുടങ്ങിയ മേഖലകളിലെല്ലാം കെ പി ഒ ഉപയോഗിക്കുന്നു.

10.5.3 കോൾ സെന്റർ (Call centre)

വ്യാപാര മേഖലയിൽ കോൾ സെന്ററുകൾക്ക് വളരെയധികം പ്രാധാന്യമുണ്ട്. പൊതുജനങ്ങൾക്ക് ബില്ലുകൾ അടക്കുന്നതിനും, സാധനങ്ങൾ വാങ്ങുന്നതിനും കോൾസെന്ററുകൾ സൗകര്യമേകുന്നു. ഒരു സ്ഥാപനത്തിലേക്ക് വരുന്നതും പോവുന്നതുമായ കോളുകളെ നിയന്ത്രിക്കുവാനുള്ള ടെലിഫോൺ സേവനമാണ് കോൾസെന്റർ. കോൾസെന്ററുകൾ സർവീസ് സെന്ററുകൾ, സെയിൽസ് സെന്ററുകൾ, കോൺടാക്ട് സെന്ററുകൾ എന്നീ പേരുകളിലും അറിയപ്പെടുന്നു. കോൾ സെന്ററിന്റെ പ്രധാന ഉത്തരവാദിത്തം ഒരു സ്ഥാപനം നൽകി വരുന്ന സേവനങ്ങൾക്കായുള്ള ടെലിഫോൺകോളുകളെ നിയന്ത്രിക്കുക എന്നതാണ്. ടെലി മാർക്കറ്റിങ്ങ് സ്ഥാപനങ്ങളും, കമ്പ്യൂട്ടർ പ്രോഡക്ട് ഡെസ്ക്കുകളും കൂടാതെ ടെലിഫോൺ ഉപയോഗിച്ച് വിൽപ്പന നടത്തുന്ന സ്ഥാപനങ്ങളുമാണ് കോൾസെന്ററുകളെ പരമാവധി ഉപയോഗപ്പെടുത്തുന്നത്. കോളുകളെ അകത്തേക്ക് വരുന്നകോളുകൾ (in-bound calls) എന്നും പുറത്തേക്ക് പോകുന്ന കോളുകൾ (out-bound calls) എന്നും രണ്ടായി തരം തിരിക്കാം. ഇൻബൗണ്ട് കോളുകൾ ഏതെങ്കിലും രീതിയിലുള്ള സഹായയ്ക്കോ ഓർഡറുകൾ ചെയ്യുന്നതിനോ വേണ്ടിയുള്ളതാവാം. എന്നാൽ ഔട്ട്ബൗണ്ട് കോളുകൾ വിൽപ്പന മെച്ചപ്പെടുന്നതിനോ മറ്റ് വിവരങ്ങൾക്കോപ വേണ്ടിയുള്ളതാവാം.

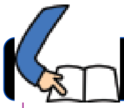
ചില സ്ഥാപനങ്ങൾ കോളുകൾ മാത്രം (ഇൻബൗണ്ട് അല്ലെങ്കിൽ ഔട്ട്ബൗണ്ട്) കൈകാര്യം ചെയ്യുന്നു. എല്ലാവിധ ടെലിഫോൺ ഓർഡറുകളും കൈകാര്യം ചെയ്യുന്നത് കോൾസെന്ററുകളാണ്, അല്ലാതെ നിർമ്മാതാക്കളോ, വിതരണക്കാരോ, സേവനദാതാക്കളോ അല്ല.

10.5.4 ടെലികോൺഫറൻസിങ്ങ് (Teleconferencing)

ടെലി കോൺഫറൻസിങ്ങ് എന്നുപറയുന്നത് വിദൂരസ്ഥലങ്ങളിലുള്ള ഒന്നിലധികം ആളുകൾ തമ്മിൽ ഐ സി ടിയുടെ സഹായത്തോടുകൂടി നടത്തുന്ന ഒത്തുചേരലോ, ചർച്ചയോ ആകാം. വ്യത്യസ്ത സ്ഥലങ്ങളിലുള്ള ആളുകളെ തമ്മിൽ ബന്ധിപ്പിക്കുന്നതിന് ടെലി കോൺഫറൻസിങ്ങ് ഉപയോഗിക്കുന്നു. വിദ്യാഭ്യാസം, ബിസിനസ്, ഭരണ നിർവ്വഹണം എന്നീ മേഖലകളിലാണ് ഇത് കൂടുതലായി ഉപയോഗിക്കുന്നത്. സമയലാഭം, യാത്രാ ചെലവ് ചുരുക്കൽ, തീരുമാനങ്ങൾ എടുക്കുന്ന പ്രക്രിയയിലേക്ക് കൂടുതൽ ആളുകളുടെ പങ്കാളിത്തം ഉറപ്പാക്കൽ മുതലായവ ഇതിന്റെ മേൻമകളാണ്.

10.5.5 വീഡിയോ കോൺഫറൻസിങ്ങ് (Video conferencing)

വീഡിയോ കൂടി ഉൾപ്പെടുത്തിക്കൊണ്ടുള്ള ടെലി കോൺഫറൻസിങ്ങ് സംവിധാനമാണ് വീഡിയോ കോൺഫറൻസിങ്ങ്. വിദൂരസ്ഥലങ്ങളിലുള്ള ആളുകൾക്ക് സ്ക്രീനിൽ പരസ്പരം കണ്ട് കൊണ്ട് ചർച്ചയിൽ ഏർപ്പെടാനും സന്ദേശങ്ങൾ കൈമാറുവാനും ഇതുവഴി സാധിക്കുന്നു. ഇവിടെ ഇന്റർനെറ്റിൽ ഘടിപ്പിച്ചിട്ടുള്ള കമ്പ്യൂട്ടറും, വെബ്ക്യാമറയും, സ്പീക്കർ ഫോണും ഉപയോഗപ്പെടുത്തുന്നു. ഓഡിയോ, വീഡിയോ ഉപകരണങ്ങളുടെ സഹായത്തോടുകൂടി ദൂരസ്ഥലങ്ങളിലുള്ള ആളുകളുമായി ചർച്ചകൾ സംഘടിപ്പിക്കുന്നതിന് ഇതുവഴി സാധിക്കുന്നു.



നമുക്ക് സംഗ്രഹിക്കാം

ഐ ടി വളരെ വേഗത്തിൽ വളർന്നു കൊണ്ടിരിക്കുന്ന ഒരു മേഖലയാണ്. ഇത് നമ്മുടെ ദൈനംദിന ജീവിതത്തിൽ വളരെയധികം ഉപയോഗിക്കുന്നു. വ്യാപാര വിനിമയം, സർക്കാരിന്റെ പ്രവർത്തനങ്ങൾ, പഠന പ്രക്രിയകൾ, ആരോഗ്യമേഖലയിലെ ഗവേഷണം എന്നിവയ്ക്ക് ഐ ടി പരമാവധി പ്രയോജനപ്പെടുത്താം. ആശയ വിനിമയ രംഗത്തുണ്ടായ പെട്ടെന്നുള്ള വളർച്ചയ്ക്ക് ഐ ടി കാരണമായിട്ടുണ്ട്. ഐ ടിയുടെ ഉപയോഗം വഴി ആളുകൾക്ക് വളരെ കുറഞ്ഞ ചെലവിലും ബുദ്ധിമുട്ടില്ലാതെയും ആശയവിനിമയം നടത്താൻ സാധിക്കുന്നു. ഐ സി ടി യുടെ പര്യായമാണ് ഐ ടി. ഐ സി ടിയുടെ സേവനങ്ങളാണ് ബി.പി.ഒ, കെ.പി.ഒ, കോൾസെന്റർ, ടെലി കോൺഫറൻസിങ്ങ്, വീഡിയോ കോൺഫറൻസിങ്ങ് തുടങ്ങിയവയെല്ലാം. അതുകൊണ്ടു തന്നെ പുതുതലമുറയുടെ അവിഭാജ്യഘടകങ്ങളാണ് ഐ ടി യും ഐ സി ടി യും.

പഠന നേട്ടങ്ങൾ

പാഠഭാഗത്തിന്റെ പൂർത്തീകരണത്തിന് ശേഷം പഠിതാവിന്

- ഇ - ഗവണൻസ്, ഇ - വ്യാപാരം, ഇ - പഠനം തുടങ്ങിയവയുടെ ഉപയോഗം ക്രമപ്പെടുത്തുവാൻ കഴിയുന്നു.
- ഇ - പഠനത്തിന് ഉപയോഗിക്കുന്ന ഉപകരണങ്ങളെപ്പറ്റി വിശദീകരിക്കുവാൻ കഴിയുന്നു.
- ആരോഗ്യ മേഖലയിലെ ഐ സി ടിയുടെ ഉപയോഗങ്ങൾ ക്രമീകരിക്കുവാൻ കഴിയുന്നു.
- ഇ - ഗവർണൻസ്, ഇ - വ്യാപാരം, ഇ - പഠനം തുടങ്ങിയവ പ്രാവർത്തികമാകുമ്പോൾ ഉണ്ടാവുന്ന ഗുണങ്ങളും, നേരിടുന്ന വെല്ലുവിളികളും ക്രമീകരിക്കുവാൻ കഴിയുന്നു.
- ഓരോ വിഭാഗത്തിലേയും വെബ്സൈറ്റുകളെ ഉപയോഗപ്പെടുത്തുക.
- ഐ സി ടി അധിഷ്ഠിത സേവനങ്ങൾ പരമാവധി ഉപയോഗപ്പെടുത്തുവാൻ കഴിയുന്നു.

മാതൃക ചോദ്യങ്ങൾ

ഹ്രസ്വോത്തര ചോദ്യങ്ങൾ

1. വളരെ എളുപ്പത്തിൽ ഇ-ടെക്സ്റ്റ് വായിക്കുവാൻ സഹായിക്കുന്ന ഇലക്ട്രോണിക് ഉപകരണത്തിന്റെ പേര് എഴുതുക.
2. ബി പി ഒ എന്നതിന്റെയും കെ പി ഒ എന്നതിന്റെയും പൂർണ്ണരൂപം എഴുതുക.
3. ഏതെങ്കിലും രണ്ട് ഇ-പഠന ഉപകരണങ്ങളുടെ പേരെഴുതുക.

ലഘു ഉപന്യാസ ചോദ്യങ്ങൾ

1. വിവിധ തരത്തിലുള്ള ഇ-ഭരണത്തിന്റെ പരസ്പര വ്യവഹാരങ്ങളുടെ ഒരു പട്ടിക തയ്യാറാക്കുക
2. ബി പി ഒ എന്താണെന്നും കെ പി ഒ എന്താണെന്നും വേർതിരിച്ച് എഴുതുക.
3. ഇ-ഭരണത്തിന്റെ ഗുണങ്ങൾ എന്തെല്ലാമാണ്?
4. അക്ഷയയുടെ കർത്തവ്യം എന്താണ്?
5. കോൾസെന്ററിന്റെ പ്രവർത്തികൾ എന്തെല്ലാമാണെന്ന് എഴുതുക.

ഉപന്യാസ ചോദ്യങ്ങൾ

1. ഇ-ഭരണം പ്രാവർത്തികമാക്കിയാൽ നമ്മൾ അഭിമുഖീകരിക്കേണ്ട പ്രധാനപ്പെട്ട വെല്ലുവിളികൾ എന്തെല്ലാം ?
2. ഇ-വ്യാപാരത്തിന്റെ ഗുണങ്ങളും ദോഷങ്ങളും എന്തെല്ലാമാണെന്ന് താരതമ്യം ചെയ്യുക.
3. ഏതെങ്കിലും മൂന്ന് ഐ ടി അധിഷ്ഠിത സേവനങ്ങളെക്കുറിച്ച് വിശദീകരിക്കുക.
4. ഐ ടി യുടെ ആരോഗ്യ മേഖലയിലെ സേവനങ്ങളെക്കുറിച്ച് വിശദമായി എഴുതുക.
5. എന്താണ് കോമൺ സർവ്വീസ് സെന്റർ? എന്തെല്ലാം സേവനങ്ങളാണ് കോമൺ സർവ്വീസ് സെന്റർ വാഗ്ദാനം ചെയ്യുന്നതെന്ന് ക്രമീകരിച്ച് എഴുതുക.

References

- Pradeep K. Sinha, Priti Sinha. *Computer Fundamentals* : BPB Publication
- V. Rajaraman (2010). *Fundamentals of Computers* : PHI Publication
- Thomas L. Floyd (2011). *Digital Fundamentals* : Pearson Education
- Craig Zacker, John Rourke (2008). *PC Hardware: The Complete Reference*: TMH Publication
- Abraham Silberschatz, Greg Gagne, Peter B. Galvin (2005). *Operating System Concepts* : John Wiley & Sons
- GIMP Documentation Team. *GIMP User Manual* : gimp.org
- Open Office.org. *OpenOffice.org3: Getting Started Guide* : openoffice.org
- Andrew S. Tanenbaum, David J. Wetherall (2010). *Computer Networks* : Prentice Hall

പദശേഖരം

അസംബ്ലി ഭാഷ	:	assembly language
അൺ ഗൈഡഡ് മീഡിയം	:	unguided medium
ആവർത്തന പ്രസ്താവനകൾ	:	interation statements
ആഗമന നിയന്ത്രണ ലൂപ്പുകൾ	:	entry controlled loop
ഇടം നൽകൽ	:	allocation
ഇ- രണം	:	e-governance
ഇ-പഠനം	:	e-learning
ഉപയോക്തൃനാമം	:	user name
കമ്പ്യൂട്ടർ ശൃംഖലകൾ (ഗണനീ ശൃംഖല)	:	computer networks
ഗൈഡഡ് മീഡിയം	:	guided medium
ഡാറ്റ വിവരവിനിമയ സംവിധാനം	:	data communication system
ഡാറ്റ വിവരവിനിമയ ഉപകരണങ്ങൾ	:	data communication devices
ഡാറ്റ ടെർമിനൽ ഉപകരണങ്ങൾ	:	data terminal equipments
ഡിവൈസ് കൈകാര്യം ചെയ്യുക.	:	device management
തനത് വില	:	default value
തരംഗങ്ങൾ	:	waves
ദ്വിതീയ സംഭരണം	:	secondary storage
നൽകിയ ഇടം തിരികെ എടുക്കൽ	:	de-allocation
നിർദ്ദേശം വ്യാഖ്യാനിക്കുക.	:	command interpretation
നിയന്ത്രണ വേരിയബിൾ	:	control variable
പരിഷ്ക്കരിക്കൽ പ്രസ്താവന	:	updation statement
പരിശോധനാ പ്രയോഗം	:	test expression
ഫയൽ കൈകാര്യം ചെയ്യുക	:	file management
പ്രാഥമിക സംഭരണം	:	primary storage
ബഹിർഗമന നിയന്ത്രണ ലൂപ്പുകൾ	:	exit controlled loop

ഭാഷ പ്രൊസസ്സർ	:	language processor
ഭൗതിക ഘടകങ്ങൾ	:	physcial component
മാദ്ധ്യമം	:	medium
യന്ത്ര ഭാഷ	:	machine language
ലോജിക്കൽ പ്രയോഗം	:	logical expression
വാക്യഘടന	:	syntax
വിഭവ പങ്കിടൽ/ വിഭവം പങ്കുവയക്കൽ	:	resource Sharing
വില പ്രകടന അനുപാതം	:	price performance ratio
വിവര വിനിമയ മാദ്ധ്യമം	:	communication medium
വിവര വിനിമയം	:	communication
വെബ്നാമ സംവിധാനം	:	DNS
ശക്തി വർധകം	:	repeater
ശൃംഖല ക്രമീകരണരീതികൾ	:	network topologies
ശൃംഖലാ പെരുമാറ്റചട്ടങ്ങൾ	:	network protocol
സമാസമ രീതികൾ	:	peer - peer
സ്പഷ്ടമായ ഇനം മാറ്റം	:	explicit type conversion
സംയുക്ത പ്രയോഗം	:	compound expression
സ്വതന്ത്ര ഓപ്പൺ സോഴ്സ്	:	free and open source
സൈബർ സുരക്ഷ	:	cyber security
റിലേഷണൽ പ്രയോഗം	:	relational expression