

VIJAYABHERI



**MALAPPURAM DISTRICT PANCHAYATH
EDUCATIONAL PROJECT**

കമ്പ്യൂട്ടർ സയൻസ്

Focus Plus One - 2021

STUDY MATERIAL FOR PLUS ONE EXAMINATION 2021

ACADEMIC SUPPORT: ACT MALAPPURAM

ആമുഖം

മലപ്പുറം ജില്ലാ പഞ്ചായത്ത് വിജയഭേരി വിദ്യാഭ്യാസ പദ്ധതിയുടെ ഭാഗമായി കഴിഞ്ഞ വർഷം പ്ലസ് ടു / വി.എച്ച്.എസ്.ഇ. രണ്ടാം വർഷ വിദ്യാർത്ഥികൾക്കായി ഫോക്കസ് 21 എന്ന പേരിൽ തയ്യാറാക്കിയ സ്റ്റഡി മെറ്റീരിയലുകൾക്ക് അധ്യാപകരിൽ നിന്നും വിദ്യാർത്ഥികളിൽ നിന്നും വളരെ നല്ല പ്രതികരണമാണ് ലഭിച്ചത്. കോവിഡ് മഹാമാരിമൂലം സാധാരണ ക്ലാസുകൾ ലഭിക്കാത്ത വിദ്യാർത്ഥികൾക്ക് പ്രസ്തുത മെറ്റീരിയൽ ഏറെ സഹായകരമായെന്ന് അവർ സാക്ഷ്യപ്പെടുത്തുന്നു.

ഒന്നാം വർഷ വിദ്യാർത്ഥികൾക്കുള്ള പരീക്ഷ സെപ്റ്റംബർ ആദ്യവാരം നടക്കുകയാണ്. ഫോക്കസ് പാഠ ഭാഗങ്ങൾക്കായി വിജയഭേരി ഫോക്കസ് +1 എന്ന പേരിൽ കഴിഞ്ഞ വർഷത്തേതു പോലെ ഈ വർഷവും വിവിധ വിഷയങ്ങൾക്ക് പ്രത്യേക സ്റ്റഡി മെറ്റീരിയൽ മലപ്പുറം ജില്ലാ പഞ്ചായത്ത് വിജയഭേരി വിദ്യാഭ്യാസ പദ്ധതിയുടെ ഭാഗമായി പുറത്തിറക്കുകയാണ്. മലപ്പുറം ഡയറ്റാണ് പ്രസ്തുത മെറ്റീരിയലിനുള്ള അക്കാദമിക പിന്തുണ നൽകിയിട്ടുള്ളത്. വിവിധ വിഷയങ്ങളുടെ ജില്ലാ തല അധ്യാപകരുടെ അസോസിയേഷനാണ് ഈ പ്രവർത്തനങ്ങൾക്ക് ഞങ്ങളുടെ കൂടെ നിന്നത്. എല്ലാവരെയും ഈ അവസരത്തിൽ നന്ദിയോടെ സ്മരിക്കുന്നു.

ഈ ഉദ്യമം അധ്യാപകർക്കും വിദ്യാർത്ഥികൾക്കും ഏറെ ഉപകാരപ്പെടുമെന്ന് പ്രതീക്ഷിക്കുന്നു. എല്ലാ വിദ്യാർത്ഥികൾക്കും മികച്ച വിജയം കൈവരിക്കാൻ കഴിയട്ടെ എന്നാശംസിക്കുന്നു.

എം.കെ. റഫീഖ്
പ്രസിഡണ്ട്
ജില്ലാ പഞ്ചായത്ത്, മലപ്പുറം

നസീബ അസീസ്
ചെയർ പേഴ്സൺ
ആരോഗ്യ വിദ്യാഭ്യാസ സ്ഥിര സമിതി

ശ്രീമതി. സ്നേഹലത
RDD മലപ്പുറം

ശ്രീ .ഉബൈദുള്ള
AD VHSE

ശ്രീ.പി.മുഹമ്മദ് മുസ്തഫ
പ്രിൻസിപ്പൽ ഇൻചാർജ്ജ്
ഡയറ്റ് മലപ്പുറം

ടി. സലീം
കോഡിനേറ്റർ
വിജയഭേരി വിദ്യാഭ്യാസ പദ്ധതി

തയാറാക്കിയത്

- ◆ ഷബീർ എം മുണ്ടേക്കൊട്ട്
 - ◆ ഷിജു പി.ബി
 - ◆ ശ്രീകാന്ത് കെ
 - ◆ Dr. സജിൻ എസ് ചന്ദ്രൻ
 - ◆ രാധിക സി.വി
 - ◆ ശിവപ്രസാദ് കെ.എ
- ജി.എച്ച്.എസ്.എസ് ഏഴൂർ
 - ജി.എച്ച്.എസ്.എസ് കുറ്റിപ്പുറം
 - ജി.ജെ.എച്ച്.എസ്.എസ് നടുവട്ടം
 - കെ.എം.ജി.വി.എച്ച്.എസ്.എസ് തവനൂർ
 - ജി.എച്ച്.എസ്.എസ് ആതവനാട്
 - ഐ.കെ.ടി.എച്ച്.എസ്.എസ് ചെറുകുളമ്പ്

ഉള്ളടക്കം

1	കമ്പ്യൂട്ടിങ് വിജ്ഞാനശാഖ	5
2	ഡാറ്റയുടെ പ്രതിനിധാനവും ബുളിയൻ ബീജ ഗണിതവും	6
3	കമ്പ്യൂട്ടർ സിസ്റ്റത്തിന്റെ ഘടകങ്ങൾ	12
4	പ്രോഗ്രാമിങ്ങിന്റെ തത്വങ്ങളും പ്രശ്നപരിഹാരവും	16
5	സി++ പ്രോഗ്രാമിങ്ങ് - ഒരു ആമുഖം	18
6	ഡാറ്റ ഇനങ്ങളും ഓപ്പറേറ്ററുകളും	19
7	നിയന്ത്രണ പ്രസ്താവനകൾ	23
8	അറൈകൾ	26
9	സ്ട്രിങ് കൈകാര്യം ചെയ്യലും ഇൻപുട്ട് / ഔട്ട്പുട്ട് ഫങ്ഷനുകളും	29
10	ഫങ്ഷനുകൾ	31
11	കമ്പ്യൂട്ടർ ശൃംഖലകൾ	34
12	ഇൻ്റർനെറ്റും മൊബൈൽ കമ്പ്യൂട്ടിംഗും	40

Evolution of the Computing machine

1. Abacus

Abacus എന്നാൽ Calculating board. ഗണിത പ്രവർത്തനങ്ങൾ നടത്താൻ വേണ്ടി ഉപയോഗിക്കുന്നു. Place value system അടിസ്ഥാനത്തിലാണ് ഇത് പ്രവർത്തിക്കുന്നത്.

2. Difference Engine

Difference Engine വികസിപ്പിച്ചെടുത്തത് Charles Babbage ആണ്.

Arithmetic calculations നടത്താനും results അച്ചടിക്കാനും കഴിയുന്ന ഒരു യന്ത്രമാണിത്.

3. Analytical Engine

Analytical Engine വികസിപ്പിച്ചെടുത്തത് Charles Babbage ആണ്.

General purpose calculations നടത്താൻ ഇതിന് കഴിഞ്ഞു.

Engine ന് ഒരു 'store' (മെമ്മറി), ഒരു 'mill' (പ്രോസസർ) എന്നിവ ഉണ്ടായിരുന്നു Machine നായുള്ള നിർദ്ദേശങ്ങൾ എഴുതിയത് Augusta Ada King ആണ്.

Generations of Computers

Criteria	Generation				
	First	Second	Third	Fourth	Fifth
Technology	Vacuum Tube	Transistor	Integrated Circuit	Microprocessor	Artificial Intelligence
Operating System	None	None	Yes	Yes	Yes
Language	Machine	Assembly	High Level	High Level	High Level
Period	1940-1956	1956-1963	1964-1971	1971-Present	Present and Yet to come

Questions

1. Difference engine was developed by -----.
2. Who is the first programmer in the world?
3. Expand ENIAC.
4. Prepare short note about the generations of computers.
5. Compare difference engine and analytical engine.

ഡാറ്റയുടെ പ്രതിനിധാനവും ബൂളിയൻ ബീജ ഗണിതവും

Number System

ഒരു Number System ൽ ഉപയോഗിക്കുന്ന ചിഹ്നങ്ങളുടെ എണ്ണത്തെ Base അല്ലെങ്കിൽ Radix എന്ന് വിളിക്കുന്നു.

ഡെസിമൽ നമ്പർ സിസ്റ്റം
 അടിസ്ഥാനം - 10
 0-9 അക്കങ്ങൾ അടങ്ങിയിരിക്കുന്നു.

ബൈനറി നമ്പർ സിസ്റ്റം
 അടിസ്ഥാനം - 2
 0, 1 അക്കങ്ങൾ അടങ്ങിയിരിക്കുന്നു.
 ഒരു ബൈനറി നമ്പറിന്റെ ഓരോ അക്കത്തെയും ബിറ്റ് (Binary Digit) എന്ന് വിളിക്കുന്നു.

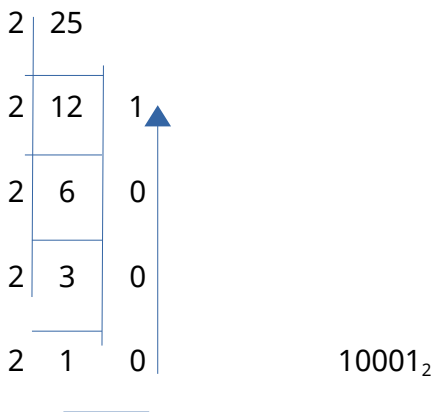
ഒക്ടൽ നമ്പർ സിസ്റ്റം
 അടിസ്ഥാനം - 8
 0-7 അക്കങ്ങൾ അടങ്ങിയിരിക്കുന്നു.

ഹെക്സാഡെസിമൽ നമ്പർ സിസ്റ്റം
 അടിസ്ഥാനം - 16
 0-15 അക്കങ്ങൾ അടങ്ങിയിരിക്കുന്നു
 A,B,C,D,E,F എന്നീ ചിഹ്നങ്ങൾ യഥാക്രമം 10,11,12,13,14,15 എന്നീ സംഖ്യകളെ പ്രതിനിധീകരിക്കാൻ ഉപയോഗിക്കുന്നു.

Number conversion

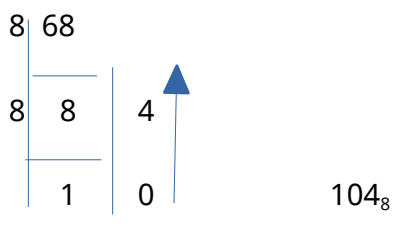
1. Decimal to binary conversion

Decimal number നെ 2 കൊണ്ട് ആവർത്തിച്ച് ഹരിക്കുക. ശിഷ്ടങ്ങൾ വിപരീതക്രമത്തിൽ എഴുതുക.



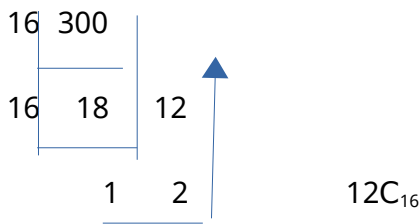
2. Decimal to octal conversion

Decimal number നെ 8 കൊണ്ട് ആവർത്തിച്ച് ഹരിക്കുക. ശിഷ്ടങ്ങൾ വിപരീതക്രമത്തിൽ എഴുതുക.



3. Decimal to hexadecimal conversion

Decimal number നെ 16 കൊണ്ട് ആവർത്തിച്ച് ഹരിക്കുക. ശിഷ്ടങ്ങൾ വിപരീതക്രമത്തിൽ എഴുതുക.



4. Binary to Decimal conversion

സ്ഥാനവില ($2^0, 2^1, 2^2, \dots$) അനുസരിച്ച് ഓരോ അക്കവും ഗുണിച്ച്, ഗുണനഫലങ്ങളുടെ തുക കണ്ടെത്തുക.

$$1011_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11_{10}$$

2^3	2^2	2^1	2^0
1	0	1	1

5. Octal to Decimal conversion

സ്ഥാനവില ($8^0, 8^1, 8^2, \dots$) അനുസരിച്ച് ഓരോ അക്കവും ഗുണിച്ച്, ഗുണനഫലങ്ങളുടെ തുക കണ്ടെത്തുക.

$$142_8 = 1 \times 8^2 + 4 \times 8^1 + 2 \times 8^0 = 64 + 32 + 2 = 98_{10}$$

6. Hexadecimal to Decimal conversion

സ്ഥാനവില ($16^0, 16^1, 16^2, \dots$) അനുസരിച്ച് ഓരോ അക്കവും ഗുണിച്ച്, ഗുണനഫലങ്ങളുടെ തുക കണ്ടെത്തുക.

$$201_{16} = 2 \times 16^2 + 0 \times 16^1 + 1 \times 16^0 = 512 + 1 = 513_{10}$$

Short cut conversions

1. Octal to binary
2. Binary to octal

Octal to Binary conversion ഉം, Binary to octal conversion ഉം താഴെക്കാണിച്ച പട്ടിക ഉപയോഗിക്കാം.

Octal	0	1	2	3	4	5	6	7
Binary	000	001	010	011	100	101	110	111

$$100001_2 = 41_8$$

3. Hexadecimal to binary
4. Binary to Hexadecimal

Hexadecimal to Binary conversion ഉം, Binary to Hexadecimal conversion ഉം താഴെക്കാണിച്ച പട്ടിക ഉപയോഗിക്കാം.

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

$$89_{16} = 10001001_2$$

Representation of Integers

കമ്പ്യൂട്ടർ മെമ്മറിയിൽ ഒരു സംഖ്യയെ പ്രതിനിധീകരിക്കുന്നതിനുള്ള മൂന്ന് രീതികൾ ഉണ്ട്.

Sign and magnitude representation

Positive numbers - Leftmost bit is 0

Negative numbers - Make the leftmost bit is 1.

E.g. To represent -32 in sign magnitude form,

--- 32 ന് തുല്യമായ ബൈനറി കണ്ടെത്തുക --- > 100000

--- 8 ബിറ്റ് ഫോർമാറ്റിലേക്ക് മാറ്റുക ---> 00100000

--- ഇടത്തേ അക്കം 1 ആക്കുക ---> 10100000

1's complement representation

Positive numbers - represented as it is.

Negative numbers - represented as 1's complement of the corresponding number.

(1 ഉം 0 ഉം പരസ്പരം കൈമാറ്റം ചെയ്യുന്നതിലൂടെ 1's complement ലഭിക്കും).

E.g. To represent -27 in 1's complement form

--- 27 ന് തുല്യമായ ബൈനറി കണ്ടെത്തുക ---> 11011

--- 8 ബിറ്റ് ഫോർമാറ്റിലേക്ക് മാറ്റുക ---> 00011011

--- 1's complement കണ്ടെത്തുക ---> 11100100

2's complement representation

Positive numbers - represented as it is.

Negative numbers - represented as 2's complement of the corresponding number.

(1's complement നോടൊപ്പം 1 കൂട്ടിയാൽ 2's complement ലഭിക്കും)

E.g. To represent -20 in 2's complement form

--- 20 ന് തുല്യമായ ബൈനറി കണ്ടെത്തുക ---> 10100

--- 8 ബിറ്റ് ഫോർമാറ്റിലേക്ക് മാറ്റുക ---> 00010100

--- 2's complement കണ്ടെത്തുക ---> 11101011 +

$$\begin{array}{r} 1 \\ \hline 11101011 \\ \hline 11101100 \end{array}$$

Representation of characters

ASCII

- American Standard Code for Information Interchange

- ASCII - 7 - 7 ബിറ്റ് ഉപയോഗിക്കുന്നു

- ASCII-8 - 8 ബിറ്റ് ഉപയോഗിക്കുന്നു.

Unicode

- Unicode consortium വികസിപ്പിച്ചത്

- ലോകത്തിലെ എല്ലാ ലിഖിത ഭാഷകളെയും പ്രതിനിധീകരിക്കാൻ കഴിയും.

Boolean Algebra

Boolean algebra deals with the values 'true' and 'false', denoted as 1 and 0. The name Boolean is given to honour George Boole, British Mathematician.

Boolean operators and logic gates

Boolean operators or logical operators -

To perform operations on boolean values. The operands will be 0 or 1.

Basic logical operators are :

- OR - Logical Addition
- AND - Logical Multiplication
- NOT - Logical Negation

Logic gates

കമ്പ്യൂട്ടറിൽ logical പ്രവർത്തനങ്ങൾ നടത്താൻ ഉപയോഗിക്കുന്ന electronic circuit കളാണ് logic circuits.

Logic gates ഉപയോഗിച്ച് ഒരു logic circuit നിർമ്മിച്ചിരിക്കുന്നു.

ഒന്നോ അതിലധികമോ logical inputs ഉപയോഗിച്ച് logical operations നടത്താനും ഒരു output സൃഷ്ടിക്കാനും കഴിയുന്ന ഒരു ഫീസിക്കൽ ഉപകരണമാണ് logic gate.

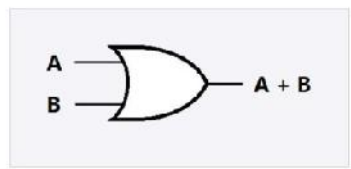
3 ലോജിക് ഗേറ്റുകൾ ഇവയാണ്: AND, OR, NOT

Logical operation ൽ സാധ്യമായ എല്ലാ input കളും അവയുടെ output കളും കാണിക്കുന്ന ഒരു പട്ടികയാണ് truth table.

OR gate

- logical addition നുവേണ്ടി ഉപയോഗിക്കുന്നു.
- + symbol ഉപയോഗിക്കുന്നു. $A+B$ എന്നത് A OR B എന്ന് വായിക്കുന്നു.

ലോജിക്കൽ OR operation നടപ്പിലാക്കാൻ logical OR gate ഉപയോഗിക്കുന്നു.

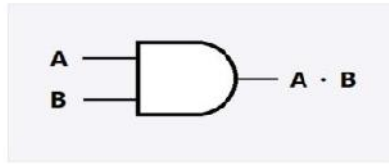


Truth table for OR operation

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

AND gate

- Logical Multiplication നുവേണ്ടി ഉപയോഗിക്കുന്നു.
- . symbol ഉപയോഗിക്കുന്നു. $A \cdot B$ എന്നത് A AND B എന്ന് വായിക്കുന്നു

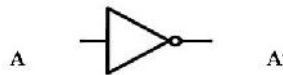


Truth table for AND operation

A	B	A · B
0	0	0
0	1	0
1	0	0
1	1	1

NOT gate

- Logical negation നു വേണ്ടി ഉപയോഗിക്കുന്നു.
- 1 operand മാത്രം ഉപയോഗിക്കുന്നു.
- \bar{A} , A' എന്നീ symbols ഉപയോഗിക്കുന്നു. \bar{A} , A' എന്നത് NOT A or A complement എന്ന് വായിക്കുന്നു.
- Inverter എന്നും അറിയപ്പെടുന്നു.



Truth table for NOT operation

A	A'
0	1
1	0

Circuit Design for simple Boolean Expressions

Consider the expression $\bar{A} + B$



Questions

1. Explain any three number systems.
2. Represent -53 in all the three forms.
3. Draw the logic gate and truth table of AND operation.
4. Convert 101010_2 to Octal and Hexadecimal number systems.
5. Write short notes on :
 - (a) ASCII
 - (b) Unicode

കമ്പ്യൂട്ടർ സിസ്റ്റത്തിന്റെ ഘടകങ്ങൾ

I. Processor (പ്രോസസ്സർ)

- ഒരു പ്രോസസ്സറിനെ CPU or Central Processing Unit എന്നും വിളിക്കാവുന്നതാണ്. ഒരു പ്രോസസ്സർ രണ്ടു കാര്യങ്ങളാണ് പ്രധാനമായി ചെയ്യുന്നത്.
 - എല്ലാ കണക്കുകൂട്ടലുകളും യുക്തിപരമായ തീരുമാനങ്ങളും എടുക്കുന്നു.
 - മറ്റ് പ്രവർത്തനങ്ങളെ ഏകോക്തീകരിക്കുകയും ചെയ്യുന്നു.
- സാധാരണയായി ഉപയോഗിക്കുന്ന പ്രോസസ്സറുകളാണ് ഇൻറൽ കോർ i3, ഇൻറൽ കോർ i5, ഇൻറൽ കോർ i7, AMD Quadcore തുടങ്ങിയവ.

Registers (രജിസ്റ്റേഴ്സ്)

- നിർദ്ദേശങ്ങളും ഡാറ്റയും താൽക്കാലികമായി സംഭരിക്കാനുള്ള CPU വിനുള്ളിലെ സ്ഥലങ്ങളാണ് രജിസ്റ്ററുകൾ.
- CPU വിനുള്ളിലെ പ്രധാനപ്പെട്ട രജിസ്റ്ററുകൾ താഴെ കൊടുക്കുന്നു.

- 1. Accumulator**
- 2. Memory Address Register(MAR)**
- 3. Memory Buffer Register(MBR)**
- 4. Instruction Register(IR)**
- 5. Program Counter(PC)**

II. Memory (മെമ്മറി)

ഡാറ്റയോ നിർദ്ദേശങ്ങളോ ഫലങ്ങളോ താൽക്കാലികമായോ സ്ഥിരമായോ സൂക്ഷിച്ചു വെക്കാനുള്ള സ്ഥലമാണ് memory. താഴെപ്പറയുന്നവയാണ് മെമ്മറിയെ അളക്കുന്ന യൂണിറ്റുകളുടെ ഏകകങ്ങൾ:

Binary Digit =1 bit	1MB (Mega Byte) =1024 KB
1 Nibble = 4 Bits	1GB (Giga Byte) =1024 MB
1 Byte = 8 bits	1TB (Tera Byte) = 1024 GB
1KB (Kilo Byte) =1024 Bytes	1PB (Peta Byte) = 1024 TB

റാൻഡം ആക്സസ് മെമ്മറി (RAM)

- RAM നുള്ളിൽ എവിടെ നിന്ന് വേണമെങ്കിലും ശേഖരിക്കാനും തിരിച്ചെടുക്കാനും സാധിക്കും.
- കമ്പ്യൂട്ടർ പ്രവർത്തിക്കുമ്പോൾ ഡാറ്റയും, അപ്ലിക്കേഷൻ പ്രോഗ്രാമും ഓപ്പറേറ്റിംഗ് സിസ്റ്റവും ഇത് സൂക്ഷിക്കുന്നു.
- ഡാറ്റയും വിവരങ്ങളും വീണ്ടെടുക്കാൻ അനുവദനീയമാണ്.
- കമ്പ്യൂട്ടർ ഓഫാക്കിയാൽ ഇതിലെ ഉള്ളടക്കം നഷ്ടപ്പെടുന്നതിനാൽ ഇതൊരു അസ്ഥിരമെമ്മറിയാണ്. (volatile).

III. e-Waste (ഇ -വേസ്റ്റ്)

- ഉപേക്ഷിക്കപ്പെട്ട കമ്പ്യൂട്ടറുകൾ, ഓഫീസ് ഇലക്ട്രോണിക് ഉപകരണങ്ങൾ, വിനോദത്തിനുപയോഗിക്കുന്ന ഉപകരണങ്ങൾ, മൊബൈൽ ഫോൺ, ടെലിവിഷൻ, റഫ്രിജറേറ്റർ ഇവയെല്ലാം ഇ-വേസ്റ്റ് എന്ന വിഭാഗത്തിൽപ്പെടുന്നു.

e-Waste disposal methods (ഇ-വേസ്റ്റ് നിർമ്മാജ്ജന മാർഗ്ഗങ്ങൾ)

- ഇ-വേസ്റ്റ് നിർമ്മാജ്ജനത്തിനുവേണ്ടി താഴെ പറയുന്ന മാർഗ്ഗങ്ങൾ ഉപയോഗിക്കാം.

a. Reuse (പുനരുപയോഗം)

- സെക്കൻഡ് ഹാൻഡ് ഉപയോഗം അഥവാ കേടുപാടുകൾ പരിഹരിച്ചു മെച്ചപ്പെടുത്തി ഉപയോഗിക്കുക.
- ഇത് ഇ-വേസ്റ്റിന്റെ തോത് കുറയാൻ സഹായിക്കും.

b. Incineration (കത്തിച്ചു കളയൽ)

- പ്രത്യേകം രൂപകൽപന ചെയ്ത ഇൻസിനറേറ്ററിൽ 900 മുതൽ 1000 ഡിഗ്രി സെൽഷ്യസ് വരെ ഉള്ള ഉയർന്ന ഊഷ്മാവിൽ നിയന്ത്രണവിധേയമായി കത്തിച്ചു കളയുന്നു.

c. Recycling of e-Waste (ഇ-വേസ്റ്റിന്റെ പുനരുൽപാദനം)

- ഉൽപ്പന്നങ്ങളിൽ നിന്നും ഉപയോഗിക്കാൻ പറ്റുന്ന ഘടകങ്ങളുപയോഗിച്ചു പുതിയ ഉപകരണങ്ങൾ നിർമ്മിക്കുന്നതിനെയാണ് പുനരുൽപാദനം എന്ന് പറയുന്നത്.

d. Land filling (മണ്ണിട്ടുമുടൽ)

- ഏറ്റവും കൂടുതൽ ഉപയോഗിക്കുന്നതും എന്നാൽ ശുപാർശ ചെയ്യപ്പെടാത്തതുമായ മാർഗ്ഗമാണ് മണ്ണിട്ടു മുടൽ.

IV. System Software (സിസ്റ്റം സോഫ്റ്റ്‌വെയർ)

- കമ്പ്യൂട്ടർ സിസ്റ്റത്തിന്റെ ഉപയോഗത്തിൽ മനുഷ്യനെ സഹായിക്കാൻ വേണ്ടി രൂപകൽപന ചെയ്ത പൊതു പ്രോഗ്രാമുകളെയാണ് സിസ്റ്റം സോഫ്റ്റ്‌വെയർ എന്ന് പറയുന്നത്.
- സിസ്റ്റം സോഫ്റ്റ്‌വെയറിന്റെ ഘടകങ്ങൾ താഴെ കൊടുക്കുന്നു:
- ഓപ്പറേറ്റിംഗ് സിസ്റ്റം
- ഭാഷാ പ്രോസസ്സറുകൾ
- യൂട്ടിലിറ്റി സോഫ്റ്റ്‌വെയർ

Operating system (ഓപ്പറേറ്റിംഗ് സിസ്റ്റം)

1. ഉപയോക്താവിനെയും കമ്പ്യൂട്ടർ ഹാർഡ്‌വെയറിനെയും ബന്ധിപ്പിക്കാനായുള്ള ഒരു കൂട്ടം പ്രോഗ്രാമുകളെയാണ് ഓപ്പറേറ്റിംഗ് സിസ്റ്റം എന്നു പറയുന്നത്

പ്രാഥമികലക്ഷ്യം: കമ്പ്യൂട്ടർ സിസ്റ്റത്തെ ഉപയോഗയോഗ്യമാക്കുക.

ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന്റെ സവിശേഷതകൾ

1. ഉപയോക്താവിന് പ്രോഗ്രാമുകൾ നടപ്പിലാക്കാൻ അനുയോജ്യമായ പരിസ്ഥിതി ഒരുക്കിക്കൊടുക്കുക.
2. കാര്യക്ഷമമായ രീതിയിൽ കമ്പ്യൂട്ടർ ഹാർഡ്‌വെയർ ഉപയോഗിക്കാൻ സഹായിക്കുന്നു.
3. കമ്പ്യൂട്ടറിലെ എല്ലാവിധ പ്രവർത്തനങ്ങളെയും നിയന്ത്രിക്കുകയും ഏകോക്തിപ്പെടുത്തുകയും ചെയ്യുന്നു.
4. കമ്പ്യൂട്ടറിലെ ഹാർഡ് ഡിസ്കിൽ നിന്നും ആദ്യം എടുക്കുന്ന പ്രോഗ്രാമും ഓഫ് ആക്കുന്നതുവരെ മെമ്മറിയിൽ നിലനിൽക്കുന്ന പ്രോഗ്രാമുമാണിത്.
5. കമ്പ്യൂട്ടർ സിസ്റ്റത്തിലെ വിഭവങ്ങളുടെ മാനേജരായി പ്രവർത്തിക്കുന്നു.
6. കമ്പ്യൂട്ടറിന്റെ അനുചിതമായ ഉപയോഗവും തെറ്റുകൾ സംഭവിക്കുന്നത് തടയാനും ഇത് ശ്രമിക്കുന്നു.

Functions of an OS (പ്രധാന ധർമ്മങ്ങൾ)

- പ്രോസസ്സ് കൈകാര്യം ചെയ്യുക
- മെമ്മറി കൈകാര്യം ചെയ്യുക
- ഫയൽ കൈകാര്യം ചെയ്യുക
- ഡിവൈസ് കൈകാര്യം ചെയ്യുക

Language Processors (ഭാഷാ പ്രോസസ്സറുകൾ)

- കമ്പ്യൂട്ടറുമായി ആശയവിനിമയം നടത്താൻ കമ്പ്യൂട്ടറിനു അറിയാവുന്ന ഒരു ഭാഷ ഉപയോഗിക്കാൻ ആവശ്യമായി വരുന്നു.
- ഉയർന്നതലത്തിലുള്ള ഭാഷകളിലോ അസംബ്ലി ഭാഷകളിലോ എഴുതിയിരിക്കുന്ന പ്രോഗ്രാമുകളെ അതിനു സമാനമായ യന്ത്ര ഭാഷകളിലേക്ക് മാറ്റം ചെയ്യുന്ന സിസ്റ്റം പ്രോഗ്രാമുകളെയാണ് ഭാഷാ പ്രോസസ്സറുകൾ (Language Processors) എന്ന് വിളിക്കുന്നത്.
- കമ്പ്യൂട്ടർ ഭാഷകളെ രണ്ടായി തരംതിരിച്ചിരിക്കുന്നു.

- a. **High level languages(HLL)**
- b. **Low level languages(LLL)**
 - i. **Machine language**
 - ii. **Assembly language**

a High level languages(HLL)

- ഇംഗ്ലീഷ് ഭാഷയെപ്പോലെ ഉള്ളതാണ്.
- ലളിതമായി മനസ്സിലാക്കാൻ സാധിക്കുന്നതുമാണ്.
- കമ്പ്യൂട്ടറിനു മനസ്സിലാക്കാൻ സാധിക്കുകയില്ല.
- ഉദാഹരണം: **C++**.

b. Low level languages(LLL)

- * യന്ത്രാധിഷ്ഠിതഭാഷ എന്നറിയപ്പെടുന്നു.
- * രണ്ടു തരത്തിലുണ്ട്.

1. Machine language

- ബൈനറി അക്കങ്ങൾ ഉപയോഗിക്കുന്ന ഭാഷകളെ യന്ത്ര ഭാഷ എന്ന് വിളിക്കുന്നു.
- യന്ത്രഭാഷയിൽ പ്രോഗ്രാം എഴുതുന്നത് വളരെ ബുദ്ധിമുട്ടാണ്.

2. Assembly language

- അസംബ്ലി ഭാഷ മധ്യവർത്തിയായ പ്രോഗ്രാമിങ് ഭാഷയാണ്.
- യന്ത്രഭാഷയെ അപേക്ഷിച്ച് അസംബ്ലി ഭാഷയിൽ കമ്പ്യൂട്ടർ പ്രോഗ്രാം എഴുതാൻ എളുപ്പമാണ്.

Need for a language processor (ഭാഷാ പ്രോസസ്സറിന്റെ ആവശ്യകത)

- കമ്പ്യൂട്ടറിനു മനസ്സിലാക്കാൻ സാധിക്കാത്ത ഉയർന്നതലത്തിലുള്ള ഭാഷയിൽ എഴുതി വെച്ചിരിക്കുന്ന പ്രോഗ്രാമുകളെ താഴ്ന്നതലത്തിലുള്ള കമ്പ്യൂട്ടർ ഭാഷയിലേക്ക് മാറ്റുന്നതിനാണ് ഭാഷാ പ്രോസസ്സറുകൾ ഉപയോഗിക്കുന്നത്.

വിവിധതരത്തിലുള്ള ഭാഷാ പ്രോസസ്സറുകൾ

മൂന്ന് തരത്തിലുണ്ട്

a. Assembler(അസംബ്ലർ)

- അസംബ്ലി ഭാഷയിൽ എഴുതിയ പ്രോഗ്രാമുകളെ യന്ത്ര ഭാഷയിലേക്കു തർജ്ജമ ചെയ്യുന്നതിനാണ് അസംബ്ലർ ഉപയോഗിക്കുന്നത്.

b. Interpreter(ഇന്റർപ്രെറ്റർ)

- ഉയർന്നതലത്തിലെഴുതിയ പ്രോഗ്രാമുകളെ വരിവരിയായി യാന്ത്രിക ഭാഷയിലേക്കു മൊഴിമാറ്റം നടത്താൻ വേണ്ടിയാണ് ഇന്റർപ്രെറ്റർ ഉപയോഗിക്കുന്നത്.

c. Compiler(കംപൈലർ)

- ഉയർന്നതലത്തിലെഴുതിയ പ്രോഗ്രാമുകളെ ഒറ്റത്തവണ കൊണ്ടുതന്നെ യാന്ത്രിക ഭാഷയിലേക്കു മൊഴിമാറ്റം നടത്താൻ വേണ്ടിയാണ് കംപൈലർ ഉപയോഗിക്കുന്നത്.

V. സ്വതന്ത്ര ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയർ

- ഉപയോഗിക്കുന്നതിനും പകർപ്പ് എടുക്കുന്നതിനും വിതരണം ചെയ്യുന്നതിനും പരിശോധിക്കുന്നതിനും മാറ്റങ്ങൾ വരുത്തുന്നതിനും മെച്ചപ്പെടുത്തുന്നതിനും ഉപയോക്താവിന് സ്വാതന്ത്ര്യം നൽകുന്ന സോഫ്റ്റ് വെയറാണ് സ്വതന്ത്ര ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയർ.

FSF (Free Software Foundation) നിർവചിക്കുന്ന സ്വാതന്ത്ര്യങ്ങൾ

- 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 നവീകരിക്കുവാനും നവീകരിച്ചവയെ പുറത്തിറക്കുവാനുള്ള സ്വാതന്ത്ര്യം.

സ്വതന്ത്ര ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയറുകളുടെ ഉദാഹരണങ്ങൾ

- **GNU/Linux (ഗ് നു /ലിനക്സ്):** കമ്പ്യൂട്ടർ ഓപ്പറേറ്റിംഗ് സിസ്റ്റമാണ്.
- **GIMP(ജിമ്പ്):** ചിത്രങ്ങളെ ചിത്രപ്പെടുന്ന സോഫ്റ്റ്‌വെയർ ആണ്.
- **Mozilla Firefox (മോസില്ല ഫയർഫോക്സ്):** വെബ് ബ്രൗസർ ആണ്.
- **OpenOffice.org (ഓപ്പൺ ഓഫീസ്. ഒആർജി):** സമ്പൂർണ്ണ ഓഫീസ് പാക്കേജ് ആണ്.

ചോദ്യങ്ങൾ

- 1) 1 ബെറ്റ് =ബിറ്റ്സ്.
- 2) ഇനിപ്പറയുന്നവയിൽ ഏതാണ് ഇ-വേസ്റ്റ് നിർമ്മാർജ്ജനത്തിന് ശുപാർശ ചെയ്യപ്പെടുന്ന രീതി അല്ലാത്തത്?

a) പുനരുപയോഗം	b) കത്തിച്ചു കളയൽ
c) മണ്ണിട്ടുമൂടൽ	d) പുനരുൽപാദനം
- 3) താഴെ കൊടുത്തിരിക്കുന്നവയിൽ ഫ്രീ ആൻഡ് ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയർ അല്ലാത്തത് ഏതാണ്?

a. ഗ് നു /ലിനക്സ്	b. എം സി ഓഫീസ്
c. ജിമ്പ്	d. മോസില്ല ഫയർഫോക്സ്
- 4) ഏതെങ്കിലും നാല് CPU രജിസ്റ്ററുകളുടെ പേരെഴുതുക?
- 5) ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന്റെ നാല് പ്രധാന പ്രവർത്തനങ്ങൾ പട്ടികപ്പെടുത്തുക.
- 6) ഇ-വേസ്റ്റ് എന്ന പദം നിർവചിക്കുക? ഏതെങ്കിലും രണ്ട് ഇ-വേസ്റ്റ് നിർമ്മാർജ്ജന രീതികളെക്കുറിച്ച് ഹ്രസ്വമായി വിശദീകരിക്കുക.
- 7) സിസ്റ്റം സോഫ്റ്റ് വെയറിന്റെ വിവിധ ഘടകങ്ങൾ ലിസ്റ്റ് ചെയ്യുകയും വിശദീകരിക്കുകയും ചെയ്യുക.

പ്രോഗ്രാമിങ്ങിന്റെ തത്വങ്ങളും പ്രശ്നപരിഹാരവും

പ്രോഗ്രാമിങ്ങിലെ വിവിധ ഘട്ടങ്ങൾ (Phases/Steps in Programming)

ഒരു പ്രശ്ന പരിഹാരത്തിനായി കമ്പ്യൂട്ടറിന് കൃത്യമായ ക്രമത്തിലുള്ള നിർദ്ദേശങ്ങൾ നൽകുന്ന ജോലിയാണ് പ്രോഗ്രാമിങ്. ഇതിന്റെ വിവിധ ഘട്ടങ്ങൾ പറയുന്നു.

- പ്രശ്നം തിരിച്ചറിയൽ
- അൽഗോരിതങ്ങളും ഫ്ലോചാർട്ടുകളും
- പ്രോഗ്രാം കോഡ് തയ്യാറാക്കൽ
- പരിഭാഷ
- ഡിബഗ്ഗിംഗ്
- പ്രവർത്തനവും പരീക്ഷണവും
- വിവരണം തയ്യാറാക്കൽ

- Problem Identification**
- Algorithms & flowcharts**
- Coding**
- Translation**
- Debugging**
- Execution and Testing**
- Documentation**

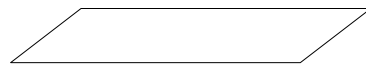
അൽഗോരിതം:- ഒരു പ്രശ്ന പരിഹാരത്തിനായി നൽകുന്ന ഒരു കൂട്ടം നിർദ്ദേശങ്ങൾ ആണിത്.

ഫ്ലോചാർട്ട്:- അൽഗോരിതത്തിന്റെ ചിത്രരൂപമാണ് ഫ്ലോചാർട്ട്. താഴെ നൽകുന്ന ചിഹ്നങ്ങൾ ഫ്ലോചാർട്ടിൽ ഉപയോഗിക്കുന്നു.

1. ടെർമിനൽ (ആരംഭവും അവസാനവും)
Terminal (To show start or Stop)



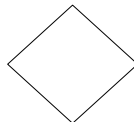
2. സമാന്തര ഭൂജം (ഇൻപുട്ട്/ഔട്ട്പുട്ട്)
Parallelogram (Input/Output)



3. ദീർഘചതുരം (പ്രവർത്തനം)
Rectangle (Process)



4. ചതുർഭുജം (തീരുമാനം)
Rhombus (Decision)



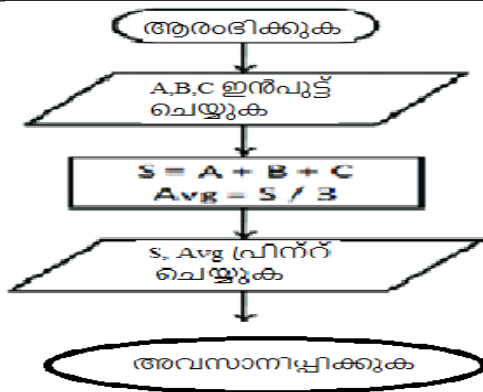
5. ആരോ / ഫ്ലോലൈനുകൾ
Arrow or Flow lines

6. വൃത്തം (കണക്ടർ)
Circle (Connector)



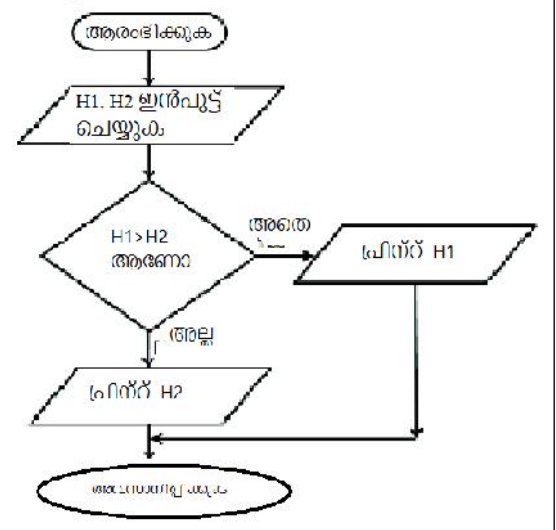
തുകയും ശരാശരിയും കണ്ടെത്തുന്നതിനുള്ള അൽഗോരിതവും ഫ്ലോചാർട്ടും

- Step 1: ആരംഭിക്കുക
- Step 2: A,B,C ഇൻപുട്ട് ചെയ്യുക
- Step 3: $S = A+B+C$
- Step 4: $Avg = S/3$
- Step 5: S, Avg പ്രിന്റ് ചെയ്യുക
- Step 6: അവസാനിപ്പിക്കുക



ഏറ്റവും കൂടിയ ഉയരം കണ്ടെത്തുന്നതിനുള്ള അൽഗോരിതവും ഫ്ലോചാർട്ടും

- Step 1: ആരംഭിക്കുക
- Step 2: H1, H2 ഇൻപുട്ട് ചെയ്യുക
- Step 3: അഥവാ $H1 > H2$ ആണെങ്കിൽ
- Step 4: H1 പ്രിന്റ് ചെയ്യുക
- Step 5: അല്ലെങ്കിൽ
- Step 6: H2 പ്രിന്റ് ചെയ്യുക
- Step 7: പരിശോധന അവസാനിപ്പിക്കുക
- Step 8: അവസാനിപ്പിക്കുക



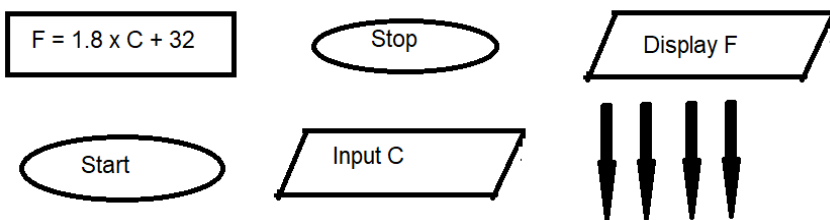
ഡിബഗ്ഗിംഗ്

പ്രോഗ്രാമിലെ തെറ്റുകൾ കണ്ടെത്തി തിരുത്തുന്നതാണ് ഡിബഗ്ഗിങ്. പ്രോഗ്രാമിലെ തെറ്റുകളെ ബഗ് (**Bug**) എന്നും അത് തിരുത്തുന്നതിനെ ഡിബഗ്ഗിങ് എന്നും വിളിക്കുന്നു. മൂന്ന് തരം തെറ്റുകൾ (**Errors**) ഉണ്ട്.

- സിൻറാക്സ് എറർ:- പ്രോഗ്രാമിങ് ഭാഷയുടെ നിയമങ്ങൾ പാലിക്കാതിരിക്കുക. ഉദാ: ഒരു വാക്യത്തിന്റെ അവസാനം ; (സെമിക്കോളൻ) ഉപയോഗിക്കാതിരിക്കൽ
- ലോജിക്കൽ എറർ:- നിങ്ങൾ നൽകുന്ന ഇൻപുട്ടിന് തെറ്റായ റിസൾട്ട് ലഭിക്കൽ. ഉദാ: + ന് പകരം * ഉപയോഗിക്കുക
- റൺ ടൈം എറർ:- പ്രോഗ്രാം പ്രവർത്തനം തടസ്സപ്പെടുത്തുന്ന എറർ. ഉദാ: 0 കൊണ്ടുള്ള ഹരണം

ചോദ്യങ്ങൾ

1. ഫ്ലോചാർട്ടിൽ പ്രോസസ്സിനെ സൂചിപ്പിക്കാൻ ----- ചിഹ്നം ഉപയോഗിക്കുന്നു.
2. പ്രോഗ്രാമിലെ തെറ്റുകൾ കണ്ടെത്തി തിരുത്തുന്നതാണ് -----
3. ഡിബഗ്ഗിങ് എന്നാൽ എന്ത്? വിവിധ തരം എററുകൾ ഏതെല്ലാം
4. താഴെ തന്നിട്ടുള്ള പ്രോഗ്രാമിങ്ങിലെ ഘട്ടങ്ങളെ ശരിയായ ക്രമത്തിൽ എഴുതുക. ഡിബഗ്ഗിങ്, പ്രശ്നം തിരിച്ചറിയൽ, പ്രവർത്തനവും പരീക്ഷണവും, കോഡിംഗ്
5. പ്രോഗ്രാമിങ്ങിലെ വിവിധ ഘട്ടങ്ങൾ എഴുതുക.
6. മൂന്ന് സംഖ്യകളുടെ ശരാശരി (**Average**) കാണുവാൻ അൽഗോരിതം എഴുതുക.
7. രണ്ട് സംഖ്യകളിൽ നിന്നും ചെറുതിനെ കണ്ടെത്തുവാൻ ഫ്ലോചാർട്ട് വരയ്ക്കുക.
8. താഴെ തന്നിട്ടുള്ളത് ഒരു ഫ്ലോചാർട്ടിലെ വിവിധ ചിഹ്നങ്ങൾ ആണ്. അതിനെ ക്രമത്തിൽ ആക്കുക.



സി++ പ്രോഗ്രാമിങ്ങ് - ഒരു ആമുഖം

ടോക്കൺകൾ:- സി++ ലെ വാക്കുകളാണ് ടോക്കൺ. ഒരു പ്രോഗ്രാമിന്റെ അടിസ്ഥാന ഘടങ്ങൾ ആണ് ഇവ. അവ ലെക്സിക്സ് യൂണിറ്റ് എന്നറിയപ്പെടുന്നു. സി++ ലെ വിവിധ തരം ടോക്കൺകൾ താഴെ പറയുന്നവയാണ്

1. കീവേർഡുകൾ
2. ഐഡന്റിഫയറുകൾ
3. ലിറ്ററലുകൾ
4. പഞ്ചുവേറ്ററുകൾ
5. ഓപറേറ്ററുകൾ

കീവേർഡുകൾ:- സി++ ൽ പ്രത്യേക അർത്ഥമുള്ള വാക്കുകളാണിവ. ഇവയുടെ അർത്ഥം മാറ്റുക സാധ്യമല്ല. ഇവ ചെറിയ അക്ഷരങ്ങളിൽ സൂചിപ്പിക്കണം.

ഉദാ: *int, float, double, for, if, switch, break, continue, char, void*

ഐഡന്റിഫയറുകൾ:- വേരിയബിൾ, ലേബൽ, ഫങ്ഷൻ എന്നിവക്ക് പേര് നൽകാൻ ഉപയോഗിക്കുന്നവയാണിവ. താഴെ പറയുന്ന നിയമങ്ങൾ പാലിക്കണം.

- a) **A-Z, a-z** അക്ഷരങ്ങളിൽ / അണ്ടർ സ്കോർ (**_**) ഉപയോഗിച്ച് തുടങ്ങിയിരിക്കണം
- b) അക്ഷരങ്ങൾ, അക്കങ്ങൾ, അണ്ടർ സ്കോർ എന്നിവ ഉപയോഗിക്കാം
- c) വൈറ്റ് സ്പെയ്സോ (ടാബ്, സ്പെയ്സ്), പ്രത്യേക ചിഹ്നങ്ങളോ (**@,%,\$ etc.**) പാടില്ല.
- d) ഇത് ഒരു കീവേർഡ് ആകാൻ പാടില്ല. (**int, float** മുതലായവ സാധ്യമല്ല)

ഉദാ: total mark2 a5 name_of_student axb ഇവ വാലിഡ് ആണ്
 +A, 2K, grand total, a*b ഇവ വാലിഡ് അല്ല (ഇൻവാലിഡ്)

ലിറ്ററലുകൾ:- പ്രോഗ്രാം പ്രവർത്തിച്ച് കൊണ്ടിരിക്കെ വിലകൾ മാറാത്ത ഇനങ്ങളാണിവ. ഉദാ: 2, 3.14, 'A', "Hello". വിവിധ തരം ലിറ്ററലുകൾ താഴെ വിവരിക്കുന്നു.

- **ഇന്റീജർ ലിറ്ററൽ:-** ദശാംശം ഇല്ലാത്ത സംഖ്യകൾ ആണിവ. ഉദാ: **125, -65, 7895**
- **ഫ്ലോട്ടിങ് പോയിന്റ് ലിറ്ററൽ:-** ദശാംശം ഉൾപ്പെടുന്ന സംഖ്യകൾ ആണിവ. ഉദാ: **2.75, -100.658**
- **ക്വാറക്ടർ ലിറ്ററൽ:-** ഒരു ജോഡി സിംഗിൾ **quotes** ഇൽ ഉൾക്കൊള്ളിച്ചിട്ടുള്ള കാരക്ടർ ആണിവ. ഉദാ: **'A', '+', '9'**
- **സ്ട്രിങ് ലിറ്ററൽ:-** ഒരു ജോഡി ഡബിൾ **quotes** ഇൽ ഉൾക്കൊള്ളിച്ചിട്ടുള്ള ഒന്നോ അതിലധികമോ കാരക്ടർ ആണിവ. ഉദാ: **"Nandana", "Welcome to C++", "9", "A", "int"**

പഞ്ചുവേറ്ററുകൾ:- സി++ ൽ പ്രസ്താവനകൾക്കിടയിൽ ഉപയോഗിക്കുന്ന സ്പെയ്സ്, കോമ (**,**), സെമിക്കോളൻ (**;**) പോലുള്ള ചിഹ്നങ്ങൾ ആണിവ.

ഓപറേറ്ററുകൾ:- ഒരു പ്രവർത്തനത്തെ സൂചിപ്പിക്കുന്ന ചിഹ്നങ്ങൾ ആണിവ. ഉദാ: സങ്കലനത്തിന് **+**, ഗുണനത്തിന് ***** എന്നിവ.

ചോദ്യങ്ങൾ

- i. ടോക്കൺ എന്നാൽ എന്ത്? നാല് ടോക്കൺകളുടെ പേരെഴുതുക.
- ii. താഴെ തന്നിട്ടുള്ളവയിൽ വാലിഡ് ആയ ഐഡന്റിഫയർ ഏത്?
 - a. A+B b. A-B c. AxB d. None of these
- iii. താഴെ തന്നിട്ടുള്ള സി++ ലെ ടോക്കൺകൾ തമ്മിലുള്ള വ്യത്യാസം എഴുതുക.
A 'A' "A"
- iv. ഒരു ഐഡന്റിഫയറിന് പേര് നൽകുമ്പോൾ പാലിക്കേണ്ട നിയമങ്ങൾ എന്തെല്ലാം?
- v. വിവിധ തരം ലിറ്ററലുകൾ ഉദാഹരണ സഹിതം വിവരിക്കുക.

ഡാറ്റ ഇനങ്ങളും ഓപ്പറേറ്ററുകളും

Datatypes

ഡാറ്റയുടെ സ്വഭാവവും ഡാറ്റയിൽ നടപ്പിലാക്കാൻ കഴിയുന്ന പ്രവർത്തനങ്ങളും തിരിച്ചറിയാനുള്ള ഉപാധികളാണ് .

Fundamental datatypes: (അടിസ്ഥാന ഡാറ്റാടൈപ്പുകൾ)

Datatype	Type of value	Memory (bytes)	Eg:
void	ശൂന്യ data	0	
char	character values	1	'A', '\n'
int	പൂർണ്ണസംഖ്യകൾ	4	84, -4
float	ദശാംശ സംഖ്യകൾ	4	5.6, -89.5
double	ദശാംശ സംഖ്യകൾ (float നേക്കാൾ കൃത്യത)	8	

Integral datatypes

Floating point datatypes

Numerical datatypes

Variable(വേരിയബിൾ) : മെമ്മറി ലൊക്കേഷനുകൾക്ക് നൽകുന്ന പേരുകൾ.

Eg: int Num=18;

1001

18

Num

- i. വേരിയബിൾ നാമം: വേരിയബിളിന്റെ പേര്. (Num)
- ii. മെമ്മറി address (L value): മെമ്മറി വിലാസം. (1001)
- iii. ഉള്ളടക്കം (R value): വേരിയബിളിൽ സംഭരിച്ചിരിക്കുന്ന മൂല്യം (18)

Operators: ഒരു പ്രവർത്തനത്തെ സൂചിപ്പിക്കുന്ന ചിഹ്നങ്ങൾ.. Eg: +, <, *, &&

$a + b$ എന്ന എക്സ്പ്രഷനിൽ , + ഓപ്പറേറ്ററും a, b എന്നിവ ഓപ്പറാൻഡുകളുമാണ്.

ഓപ്പറേറ്റർസ് ന്റെ തരംതിരിക്കൽ

(1) ഓപ്പറാൻഡുകളുടെ എണ്ണത്തെ അടിസ്ഥാനമാക്കി:

Category	ഓപ്പറാൻഡുകളുടെ എണ്ണം	Example
Unary	1	Unary +, unary -, ++,--
Binary	2	+, &&, <
Ternary	3	?:

2. പ്രവർത്തനത്തിന്റെ സ്വഭാവത്തെ അടിസ്ഥാനമാക്കി

(a) അരിത്മറ്റിക് ഓപ്പറേറ്റർസ് : കൂട്ടുക (+), കുറയ്ക്കുക (-), ഗുണനം (*), ഹരിക്കുക (/), മോഡ്യൂലസ് (%) പോലുള്ള ഗണിത പ്രവർത്തനങ്ങൾക്ക് ഉപയോഗിക്കുന്നു.

x	y	x+y	x-y	x*y	x/y	x%y
7	3	10	4	21	2	1(ശിഷ്ടമൂല്യം)

(b) റിലേഷണൽ ഓപ്പറേറ്റർസ്:

സംഖ്യാ ഡാറ്റ താരതമ്യം ചെയ്യാൻ ഉപയോഗിക്കുന്നു, ശരി (1) അല്ലെങ്കിൽ തെറ്റ് (0) എന്ന result ലഭിക്കുന്നു. <(ചെറുത്), >(വലുത്), <=(ചെറുതോ തുല്യമോ), >= (വലുതോ തുല്യമോ), ==(തുല്യം), != (തുല്യമല്ല) എന്നിവയാണ് റിലേഷണൽ ഓപ്പറേറ്റർസ്.

x	y	x<y	x>y	x<=y	x>=y	x==y	x!=y
7	3	0	1	0	1	0	1

(c) ലോജിക്കൽ ഓപ്പറേറ്റർസ്:

ലോജിക്കൽ മൂല്യങ്ങൾ സംയോജിപ്പിക്കാൻ ഉപയോഗിക്കുന്നു. ശരി (1) അല്ലെങ്കിൽ തെറ്റ് (0) എന്ന result ലഭിക്കുന്നു. && (AND), || (OR), !(NOT) എന്നിവയാണ് ലോജിക്കൽ ഓപ്പറേറ്റർസ്.

E1	E2	E1&&E2	E1 E2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

E	!(E)
0	1
1	0

Eg: (2>5) && (4<6) --> 0
 (2>5) || (4<6) --> 1
 !(2>5) --> 1

Input / Output operators:

Input ന് get from (എക്സ്ട്രാക്ഷൻ) ഓപ്പറേറ്റർ (>>) ഉപയോഗിക്കുന്നു. ഉദാ: cin >> a;
 Output ന് put to (ഇൻസെർഷൻ) ഓപ്പറേറ്റർ (<<) ഉപയോഗിക്കുന്നു. ഉദാ: cout << a;

Assignment (അസൈൻമെന്റ്) operator (=)

ഒരു വേരിയബിളിൽ ഒരു മൂല്യം സംഭരിക്കാൻ ഉപയോഗിക്കുന്നു. ഉദാ: a = 5;

=, == ഓപ്പറേറ്റർസ് തമ്മിലുള്ള വ്യത്യാസം

=	==
Assignment operator	Relational operator
ഒരു വേരിയബിളിന് മൂല്യം നൽകുന്നു	മൂല്യങ്ങൾ താരതമ്യം ചെയ്യുന്നു

എക്സ്പ്രഷനുകൾ: ഓപ്പറേറ്റർസ് ന്റെയും ഓപ്പറാൻഡുകളുടെയും സംയോജനം. ഉദാ: m + n * y

(a) Arithmetic എക്സ്പ്രഷനുകൾ:

- Arithmetic operators അടങ്ങിയിരിക്കുന്നു.
- (i) Integer എക്സ്പ്രഷനുകൾ: ഇൻറിജർ ഓപ്പറാൻഡുകൾ അടങ്ങിയിരിക്കുന്നു.
- (ii) ഫ്ലോട്ടിംഗ് പോയിൻറ് (റിയൽ) എക്സ്പ്രഷനുകൾ: ദശാംശ സംഖ്യാമൂല്യങ്ങൾ അടങ്ങിയിരിക്കുന്നു.
- (iii) Constant expression: സ്ഥിരമൂല്യങ്ങൾ അടങ്ങിയിരിക്കുന്നു. (eg: 5+m/3)

(b) റിലേഷണൽ എക്സ്പ്രഷനുകൾ:

റിലേഷണൽ operators അടങ്ങിയിരിക്കുന്നു. ഉദാ: x > y

(c) ലോജിക്കൽ എക്സ്പ്രഷനുകൾ:

ലോജിക്കൽ operators അടങ്ങിയിരിക്കുന്നു. ഉദാ: (a<b)&& (c>d)

Statements(സ്റ്റേറ്റ്‌മെന്റുകൾ) :

ഒരു പ്രോഗ്രാമിന്റെ ഏറ്റവും ചെറിയ എക്സിക്യൂട്ടബിൾ യൂണിറ്റ്. c ++ സ്റ്റേറ്റ്‌മെന്റ് അവസാനിപ്പിക്കുന്നത് ; എന്ന ചിഹ്നം ഉപയോഗിച്ചാണ് .

- 1. **ഡിക്ലറേഷൻ സ്റ്റേറ്റ്‌മെന്റ്:** ഒരു വേരിയബിളിൽ സൂക്ഷിക്കുന്ന ഡാറ്റയുടെ തരം വ്യക്തമാക്കുന്നു. syntax: datatype var_name; ഉദാ: float avg;
- 2. **അസൈൻമെന്റ് സ്റ്റേറ്റ്‌മെന്റുകൾ:** ഇത് ഒരു വേരിയബിളിന് ഒരു മൂല്യം നൽകുന്നു. ഉദാ: a=15;
- 3. **Input സ്റ്റേറ്റ്‌മെന്റുകൾ:** ഇൻപുട്ട് ചെയ്യാൻ ഉപയോഗിക്കുന്നു. ഉദാ: cin>>score;
- 4. **Output സ്റ്റേറ്റ്‌മെന്റ്:** output ചെയ്യാൻ ഉപയോഗിക്കുന്നു. ഉദാ: cout<<score;

ഒരൊറ്റ statement ലെ input അല്ലെങ്കിൽ output operators ന്റെ ഒന്നിലധികം ഉപയോഗത്തെ **I/O operators ന്റെ കാസ്കേഡിംഗ്** എന്ന് വിളിക്കുന്നു. Eg: cin>>x>>y>>z;

C ++ പ്രോഗ്രാമിന്റെ അടിസ്ഥാന ഘടന

```
#include <headerfile>      ----> line 1
using namespace identifier; ----> line 2
int main()                  ----> line 3
{ statements;
  :
  return 0; }
```

Line 1: Preprocessor directive (പ്രീപ്രോസസ്സർ നിർദ്ദേശം):

യഥാർത്ഥ compilation ന് മുമ്പ് ഒരു പ്രവർത്തനം നടത്താൻ കംപൈലറോട് നിർദ്ദേശിക്കുന്നു. # ചിഹ്നത്തിൽ ആരംഭിക്കുന്നു. ഉദാ: # include <iostream> - iostream എന്ന header ഫയൽ ലിങ്കുചെയ്യുന്നതിനുള്ള നിർദ്ദേശം.

Line 2: Namespace statement:

ഈ പ്രോഗ്രാമിൽ std എന്ന നെയിംസ്പേസ് ഉപയോഗിക്കാൻ കംപൈലറോട് പറയുന്നു. cin, cout എന്നിവയുൾപ്പെടെ മറ്റു പല ഒബ്ജക്റ്റുകളും നിർവ്വചിക്കുന്ന സ്റ്റാൻഡേർഡ് നെയിംസ്പേസ് ആണ് std.

Line 3: Function header:

ഓരോ c ++ പ്രോഗ്രാമിനും അത്യാവശ്യമായ function ആണ് main (). ഒരു c++ പ്രോഗ്രാം execution main() ഫംഗ്ഷനുള്ളിൽ ആരംഭിക്കുകയും അവസാനിക്കുകയും ചെയ്യുന്നു.

A sample program

```
#include<iostream>
using namespace std;
int main()
{
  cout<<"Hello, Welcome to C++";
  return 0;
}
```

ചോദ്യങ്ങൾ

1. ശൂന്യമായ data യ്ക്ക് ____ ഡാറ്റാ ടൈപ്പ് ഉപയോഗിക്കുന്നു.
2. ഇനിപ്പറയുന്നവയിൽ insertion ഓപ്പറേറ്റർ ഏതാണ്?
(<<, >>, <,>)
3. 8 ബൈറ്റ് മെമ്മറി ഉപയോഗിക്കുന്ന അടിസ്ഥാന ഡാറ്റാ ടൈപ്പ് ____ ആണ്.
4. $M = 7$ ഉം $n = 2$ ഉം ആണെങ്കിൽ ഇനിപ്പറയുന്ന expressios ൽ output പ്രവചിക്കുക
(i) m / y (ii) $m \% y$ (iii) $(m == n)$ (iv) $(m > 10) \&\& (n < 5)$
5. $==$, $=$ operators തമ്മിൽ വ്യത്യാസം എന്ത് ?
6. ഡാറ്റാ ടൈപ്പിന്റെ പ്രാധാന്യം എന്താണ്? c++ ൽ 2 ഇൻഗ്രൽ ഡാറ്റാ ടൈപ്പുകളുടെ പേര് നൽകുക.
7. ഇനിപ്പറയുന്ന c ++ സ്റ്റേറ്റ്‌മെന്റിൽ നിന്ന് total എന്ന വേരിയബിളിന്റെ Rvalue, മെമ്മറി സൈസ് എന്നിവ കണ്ടെത്തുക
`float total= 34.6;`
8. ഒരു ബൈനറി ഓപ്പറേറ്റർ ഒരു unary ഓപ്പറേറ്ററിൽ നിന്ന് എങ്ങനെ വ്യത്യാസപ്പെട്ടിരിക്കുന്നു?
ഓരോന്നിനും ഒരു ഉദാഹരണം നൽകുക.
9. ഒരു വിദ്യാർത്ഥിയുടെ mark സംഭരിക്കുന്നതിന് ഒരു വേരിയബിൾ declare ചെയ്യുന്ന c ++ സ്റ്റേറ്റ്‌മെന്റ് എഴുതുക.
10. c++ ലെ പ്രസ്താവനകൾ (statement) എന്നാലേന്ത്? ഏതെങ്കിലും മൂന്ന് തരം പ്രസ്താവനകൾ ഓരോ ഉദാഹരണം ഉപയോഗിച്ച് വിശദീകരിക്കുക.

നിയന്ത്രണ പ്രസ്താവനകൾ

നിയന്ത്രണ പ്രസ്താവനകൾ(Control Statements):

പ്രോഗ്രാം എക്സിക്യൂഷന്റെ സാധാരണയായുള്ള തുടർച്ചയായ രീതിയെ മാറ്റാൻ ഉപയോഗിക്കുന്ന പ്രസ്താവനകൾ.

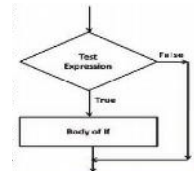
രണ്ട് തരം: (1) Decision (2) ആവർത്തനം(Iteration)

1. Decision (തിരഞ്ഞെടുക്കൽ) Statement കൾ: ഒരു condition അടിസ്ഥാനമാക്കി execution നായി statement കൾ തിരഞ്ഞെടുക്കുന്നു.

(a) **if**

Syntax: if (condition)
 { statement block ; }

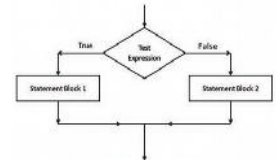
eg: if(mark>35)
 { cout<<"Passed"; }



(b) **if-else**

Syntax: if (condition)
 {statement block 1;}
else
 {statement block 2;}
.....
else
 statement block n;

eg: if(mark>35)
 {cout<<"Passed";}
else
 {cout<<"Failed";}



(c) **if-else if ladder**

*Syntax:*if (condition 1)
 statement block 1 ;
 else if (condition 2)
 statement block 2 ;

 else
 statement block n ;

eg: if (score >= 80)
 cout<<"A Grade";
 else if (score >= 60)
 cout<<"B Grade ";
 else if (score >= 40)
 cout<<"C grade";
 else
 cout<<"D grade";

(d) **switch**

Syntax:
switch (expression)
{
 case constant_1: statement block 1;break;
 case constant_2: statement block 2;break;
 :
 default : statement block n ;
}

Example
switch (game)
{
 case 1: cout<<"Football"; break;
 case 2: cout<<"Cricket"; break;
 default: cout<<"Invalid ";
}

എല്ലാ switch സ്റ്റേറ്റ്‌മെന്റുകളും if else if ലേക്ക് മാറ്റാൻ സാധിക്കും. എന്നാൽ എല്ലാ if else if കളും switch ലേക്ക് മാറ്റാൻ കഴിയില്ല

switch	if else if ladder
അനേകം ശാഖകൾ അനുവദിക്കുന്നു	അനേകം ശാഖകൾ അനുവദിക്കുന്നു
തുല്യത മാത്രമേ പരിശോധിക്കാൻ സാധിക്കൂ	ഏതു റിലേഷനൽ /ലോജിക്കൽ കണ്ടിഷനും ആകാം
case constant ഒരു പൂർണ്ണസംഖ്യ /കാരക്ടർ ആകണം	ഒന്നിലധികം values , ദശാംശസംഖ്യകൾ എന്നിവ ആകാം
case match ചെയ്തില്ലെങ്കിൽ default ഉപയോഗിക്കുന്നു, ബ്ലോക്കിന് പുറത്തു കടക്കാൻ break ഉപയോഗിക്കാം.	ഒരു കണ്ടിഷനും match ചെയ്തില്ലെങ്കിൽ else ബ്ലോക്ക് എക്സിക്യൂട്ട് ചെയ്യുന്നു

2. ആവർത്തന (ലൂപ്പിംഗ്) Statement കൾ

ഒന്നോ അതിലധികമോ statement കളുടെ കൂട്ടം ആവർത്തിച്ച് നടപ്പിലാക്കാൻ അനുവദിക്കുന്നു. **c++ ൽ 3 loop കൾ ഉണ്ട്: for, while, do-while**

ഒരു ലൂപ്പിന് ഇനിപ്പറയുന്ന ഘടകങ്ങൾ ഉണ്ട്:

- (1) ഇനിഷ്യലൈസേഷൻ - ലൂപ്പ് വേരിയബിളിന് ആരംഭ മൂല്യം നൽകുന്ന സ്റ്റേറ്റ്‌മെന്റ് (i = 1)
- (2) കണ്ടിഷൻ - നിബന്ധന (test expression) (i <= 10)
- (3) അപ്ഡേഷൻ - ലൂപ്പ് വേരിയബിളിലെ മൂല്യം മാറ്റുന്ന സ്റ്റേറ്റ്‌മെന്റ് (i ++)
- (4) ലൂപ്പ് ബോഡി : ആവർത്തിക്കേണ്ട സ്റ്റേറ്റ്‌മെന്റുകൾ (cout << i;)

for loop

```
for (i=1;i<=10;i++)
{ cout << i; }
```

while loop

```
i=1;
while(i<=10)
{
    cout<<i;
    i++;
}
```

do while loop

```
i=1;
do
{
    cout<<i;
    i++;
} while(i<=10);
```



ഇവിടെ, i ആണ് ലൂപ്പ് വേരിയബിൾ (അതിന്റെ മൂല്യം ലൂപ്പിനെ നിയന്ത്രിക്കുന്നു). [മുകളിൽ നൽകിയിരിക്കുന്ന മൂന്ന് ലൂപ്പുകളും ഒരേ output നൽകുന്നു - 1 മുതൽ 10 വരെയുള്ള സംഖ്യകൾ പ്രദർശിപ്പിക്കുന്നു]

Entry controlled loop

(എൻട്രി കൺട്രോൾഡ് ലൂപ്പ്)

- 1. ലൂപ്പ് ബോഡിക്കു മുൻപ് കണ്ടിഷൻ
- 2. condition ശരിയാണെങ്കിൽ മാത്രമേ ലൂപ്പ്

eg: for, while

Exit controlled

(എക്സിറ്റ് കൺട്രോൾഡ് ലൂപ്പ്)

- 1. ലൂപ്പ് ബോഡിക്കു ശേഷം കണ്ടിഷൻ
- 2. condition ശരിയോ തെറ്റോ ആണെങ്കിൽ പോലും ലൂപ്പ് ഒരു തവണയെങ്കിലും പ്രവർത്തിക്കും

eg: do while

Jump Statements

പ്രോഗ്രാം നിയന്ത്രണം ഒരിടത്തു നിന്ന് മറ്റൊരിടത്തേക്ക് മാറ്റാൻ സഹായിക്കുന്ന statements. സി ++ ൽ നാല് തരം jump statements ആണ് ഉള്ളത് - break, continue, goto, return. അവയിൽ രണ്ടെണ്ണം ചുവടെ വിശദീകരിച്ചിരിക്കുന്നു:

break statement

program control ന് loop നു പുറത്തേക്ക് കടക്കാൻ ഉപയോഗിക്കുന്നു.

continue statement

loop body യിലെ കോഡിന്റെ ഒരു ഭാഗം ഒഴിവാക്കുന്നതിനും അടുത്ത iteration തുടരുന്നതിനും ഉപയോഗിക്കുന്നു.

break	continue
ലൂപ്പ് അവസാനിപ്പിക്കാൻ നിർബന്ധിക്കുന്നു	ലൂപ്പിന്റെ അടുത്ത ആവർത്തനത്തെ പ്രേരിപ്പിക്കുന്നു.
സ്വിച്ച്, ലൂപ്പുകൾ എന്നിവയ്ക്കൊപ്പം ഉപയോഗിക്കുന്നു.	ലൂപ്പുകളിൽ മാത്രം ഉപയോഗിക്കുന്നു.

ചോദ്യങ്ങൾ

- 1) if else പ്രസ്താവനയുടെ സിന്റാക്സ് എഴുതുക
- 2) എൻട്രി നിയന്ത്രിത ലൂപ്പ് എക്സിറ്റ് നിയന്ത്രിത ലൂപ്പുകളിൽ നിന്ന് എങ്ങനെ വ്യത്യാസപ്പെട്ടിരിക്കുന്നു?
- 3) കണ്ടീഷൻ true ആണെങ്കിലും ഒരു ലൂപ്പിൽ നിന്നും പുറത്തേക്ക് പ്രവേശിക്കാൻ ----- പ്രസ്താവന ഉപയോഗിക്കാം
- 4) 'break', 'continue' എന്നിവ തമ്മിലുള്ള വ്യത്യാസം എഴുതുക.
- 5) സി++ ലെ 'for' പ്രസ്താവനയുടെ സിന്റാക്സ് എഴുതുക
- 6) നിയന്ത്രണ statement കൾ രണ്ടായി തരംതിരിക്കാം, ഓരോന്നും ഉദാഹരണസഹിതം വിശദീകരിക്കുക.
- 7) അനുയോജ്യമായ ഉദാഹരണത്തിലൂടെ ലൂപ്പ് ഘടകങ്ങൾ വിശദീകരിക്കുക
- 8) "switch സ്റ്റേറ്റ്‌മെന്റ് ഉപയോഗിച്ച് if ..else if കോവണി മാറ്റിയെഴുതാൻ കഴിയാതെ വരുന്ന സാഹചര്യങ്ങൾ എന്തൊക്കെയാണ് ?
- 9) നിർദ്ദേശിച്ചതുപോലെ ചെയ്യുക:
 - (a) switch ഉപയോഗിച്ച് മാറ്റിയെഴുതുക
if (ch==1) cout<<"Red";
else if (ch==2) cout<<"Blue";
else if (ch==3) cout<<"Green";
else cout<<"Wrong code";
 - (b) do while ഉപയോഗിച്ച് മാറ്റിയെഴുതുകയും ലൂപ്പ് ഘടകങ്ങൾ തിരിച്ചറിയുകയും ചെയ്യുക.
for(a=5; a<=20; a++)
{ cout<<a*a ; }

CHAPTER 8 അറേകൾ

ഒരേ പേരിൽ പരാമർശിക്കപ്പെടുന്ന ഒരേ തരത്തിലുള്ള ഡാറ്റാ വേരിയബിളുകളുടെ ഒരു ശേഖരമാണ് array

OR

ഒരേ പേരിൽ പരാമർശിക്കപ്പെടുന്ന ഹോമോജീനിയസ് വാല്യൂസിന്റെ ഒരു ശേഖരമാണ്

Need of arrays

ഒരു ക്ലാസിലെ 60 വിദ്യാർത്ഥികളുടെ മാർക്ക് സ്വീകരിക്കാൻ കഴിയുന്ന ഒരു പ്രോഗ്രാം എഴുതാൻ ആഗ്രഹിക്കുന്നുവെങ്കിൽ, അതുപോലെ മാർക്ക് ആവറേജ് കണക്കാക്കണം. വേരിയബിളുകൾ ഉപയോഗിച്ച് ഈ പ്രശ്നം പരിഹരിച്ചാൽ, വ്യത്യസ്ത മാർക്ക് സംഭരിക്കുന്നതിന് 60 വേരിയബിളുകൾ പ്രഖ്യാപിക്കേണ്ടതുണ്ട്. ഈ വേരിയബിളുകളെല്ലാം മാനേജ് ചെയ്യുന്നത് എളുപ്പമുള്ള കാര്യമല്ല, മാത്രമല്ല ഇത് പ്രശ്നം സങ്കീർണ്ണമാക്കുകയും പ്രോഗ്രാം മനസ്സിലാക്കാൻ കഴിയാതിരിക്കുകയും ചെയ്യും.

ഈ മാർക്കുകൾ സംഭരിക്കുന്നതിന് ഒരു array പ്രഖ്യാപിക്കുകയാണെങ്കിൽ, ഒരു വേരിയബിൾ മാത്രമേ കൈകാര്യം ചെയ്യേണ്ടതുള്ളൂ , array name

1. array വേരിയബിളുകളെ പ്രതിനിധീകരിക്കാനും കൈകാര്യം ചെയ്യാനും എളുപ്പമാണ്
2. array ഒരു തുടർച്ചയായ മെമ്മറി ലൊക്കേഷനുകൾ ഉപയോഗിക്കുന്നു.
3. ഒരു പ്രോഗ്രാമിന്റെ വായനാക്ഷമത വർദ്ധിച്ചു.

Declaration of Arrays

data type array name [size];

Eg :- int mark[5];

integer data type ന്റെ 5 values സൂക്ഷിക്കാൻ കഴിയുന്ന array ആണ് mark. എല്ലാ array values ഉം തുടർച്ചയായ മെമ്മറി സ്ഥാനങ്ങളിൽ ഉൾക്കൊള്ളുന്നു..

mark[0]	mark[1]	mark[2]	mark[3]	mark[4]
55	48	36	58	45
4 bytes	4 bytes	4 bytes	4 bytes	4 bytes

subscript or index value ഉപയോഗിച്ച് ഒരു അറേയിലെ elements നെ ആക്സസ് ചെയ്യാൻ കഴിയും

അറേകളുടെ elements നെ ഇങ്ങനെ പരാമർശിക്കാം
arrayname [index];

The index value (subscript) എല്ലായ്പ്പോഴും 0 (zero) യിൽ ആണ് ആരംഭിക്കുക.

മുകളിലത്തെ ഉദാഹരണത്തിൽ പറഞ്ഞിട്ടുള്ള അറേ , mark നെ താഴെ കൂടുതൽ വിശദീകരിക്കുന്നു.

- Array mark ലെ ആദ്യ വാല്യൂ (55 എന്ന് വിചാരിക്കുക) പരാമർശിക്കപ്പെടുന്നത് mark[0]
- Array mark ലെ രണ്ടാമത്തെ വാല്യൂ (48) പരാമർശിക്കപ്പെടുന്നത് mark[1]
- Array mark ലെ മൂന്നാമത്തെ വാല്യൂ (36 പരാമർശിക്കപ്പെടുന്നത് mark[2]
- Array mark ലെ നാലാമത്തെ വാല്യൂ (55) പരാമർശിക്കപ്പെടുന്നത് mark[3]
- Array mark ലെ അഞ്ചാമത്തെ വാല്യൂ (45) പരാമർശിക്കപ്പെടുന്നത് mark[4]

array ഡിക്ലറേഷൻ, int mark[20]; ഈ അറേയ്ക്ക് എത്ര മെമ്മറി അനുവദിക്കാൻ കഴിയും?

മെമ്മറിയിൽ സംഭരിക്കുന്നതിനായി ഓരോ സംഖ്യയും 4 ബൈറ്റുകൾ ഉൾക്കൊള്ളുന്നുവെന്ന് നമുക്കറിയാം. mark[20] അറേയിൽ 20 പൂർണ്ണസംഖ്യകൾ store ചെയ്യാൻ കഴിയും. അതിനാൽ $20 \times 4 = 80$ ബൈറ്റുകൾ മെമ്മറിയിൽ ഉപയോഗിക്കും

- array ഒരു derived ഡാറ്റാ ടൈപ്പ് ആണ്
- array വാല്യൂസിന്റെ ഡാറ്റാ തരം അറേയുടെ ബേസ് ടൈപ്പ് എന്നറിയപ്പെടുന്നു
- ഒരൊറ്റ വേരിയബിൾ പരാമർശിക്കുന്ന തുടർച്ചയായ മെമ്മറി സ്ഥാനങ്ങളുടെ ഒരു ശേഖരമാണ് array.

Types of Arrays

Arrays are of two types

- i) One dimensional array (Single dimensional Array)
- ii) Multi Dimensional Array

Initialization of Arrays

An array can be initialized in two ways

1) Along with the declaration

Eg :- int mark [5] = { 55, 48, 36, 58, 45} ;

2) direct Assignment

Eg :- mark[0] = 55; mark[1] = 48; mark[2] =36; mark [3] = 58 ; mark [4] = 45;

Accessing elements of array

ഏത് ലൂപ്പിന്റെയും സഹായത്തോടെ നമുക്ക് അറേയിലെ വാല്യൂസ് ആക്സസ് ചെയ്യാൻ കഴിയും. അറേയുടെ index വാല്യൂ, ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിനെ ആശ്രയിച്ചിരിക്കുന്നു.. ഓരോ എലമെന്റ്സും സബ്സ്ക്രിപ്റ്റിനൊപ്പം അറേയുടെ പേര് ഉപയോഗിച്ച് വ്യക്തമാക്കുന്നു.

Array operations

- **Traversal-** array എലമെന്റ്സിലൂടെ നീങ്ങുന്നു. array elements പ്രദർശിപ്പിക്കുന്നത് ട്രാവേർസലിന്റെ ഒരു ഉദാഹരണമാണ്
- **Searching-**അറേയിൽ സ്റ്റോർ ചെയ്തിട്ടുള്ള എലമെന്റ്സിന്റെ സ്ഥാനം കണ്ടെത്തുന്നു. രണ്ട് രീതികൾ (Linear Search, Binary Search)
- **insertion.**
- **deletion**
- **sorting-** ചില ലോജിക്കൽ ക്രമത്തിൽ അറേ ഘടകങ്ങൾ ക്രമീകരിക്കുന്നു (such as ascending or descending). Two methods selection sort, Bubble sort.
- **Merging**

Note: ഇനിപ്പറയുന്ന പ്രോഗ്രാം അറേ ട്രാവേഴ്സൽ പ്രവർത്തനം എങ്ങനെ നടത്തുന്നുവെന്ന് വിശദീകരിക്കുന്നു. ട്രാവേഴ്സൽ നടത്താൻ for ലൂപ്പ് ഉപയോഗിച്ചു. അറേ എലമെന്റ്സ് എങ്ങനെ ഉൾപ്പെടുത്താം , എങ്ങനെ ഡിസ്പ്ലേ ചെയ്യാം എന്ന് ഇനിപ്പറയുന്ന ഉദാഹരണം പഠിപ്പിക്കുന്നു

Eg :- Write a program to enter 10 elements into an array and print it.

```
# include <iostream>
using namespace std;
int main( )
{
    int num[10], i;
    cout << " Enter elements in to the Array \n ";
    for ( i = 0 ; i <= 9; ++ i)
        cin >> num[i] ;
    cout << " \n The elements in the array are \n ";
    for ( i = 0 ; i <= 9; ++ i)
        cout << num [ i] << "\n";
return 0;
}
```

മുകളിലുള്ള പ്രോഗ്രാമിൽ ഫോർ ലൂപ്പ് 10 തവണ വർക്ക് ചെയ്യുന്നു. ലൂപ്പ് കൺട്രോൾ വേരിയബിൾ i, 0 മുതൽ 9 വരെ. ഈ വാല്യൂസ് array index വാല്യൂസ് ആണ് . ഇൻപുട്ടിന് ശേഷം മറ്റൊരു for ലൂപ്പ് ഉപയോഗിച്ച് ഔട്ട്പുട്ട് നിർവഹിക്കുന്നു.

ചോദ്യങ്ങൾ

1. സി ++ ലെ സിംഗിൾ ഡെമൻഷണൽ അറേയുടെ മെമ്മറി അലോക്കേഷൻ വിവരിക്കുക
2. 5 വിദ്യാർത്ഥികളുടെ സ്കോറുകൾ ഇൻപുട്ട് ചെയ്യുന്നതിന് ഒരു സി ++ പ്രോഗ്രാം എഴുതുക, ഒരു അറേ ഉപയോഗിച്ച് വിപരീത ക്രമത്തിൽ പ്രദർശിപ്പിക്കുക
3. എന്താണ് ഒരു അറേ? നിങ്ങളുടെ ക്ലാസിലെ എല്ലാ വിദ്യാർത്ഥികളുടെയും കമ്പ്യൂട്ടർ സയൻസ് മാർക്കുകൾ സംഭരിക്കുന്നതിന് ആവശ്യമായ സി ++ പ്രോഗ്രാം എഴുതുക
4. 8,7,2,4,6 എന്നീ സംഖ്യകൾ ഉൾക്കൊള്ളുന്ന സൈസ് 5 ആയ ഒരു അറേ നിർമ്മിക്കുക
5. അറേ ട്രാവേഴ്സൽ ചിത്രീകരിക്കുന്നതിന് ഒരു സി ++ പ്രോഗ്രാം എഴുതുക.

സ്‌ട്രിങ്ങ് കൈകാര്യം ചെയ്യലും ഇൻപുട്ട് / ഔട്ട്പുട്ട് ഫങ്ഷനുകളും

Array declaration for string

കാർക്ടർ അറേ ഉപയോഗിച്ചാണ് സ്‌ട്രിങ്ങുകൾ ഡിക്ലെയർ ചെയ്യുന്നത് .

```
ഉദാ : char my_name[10];
```

Initialisation

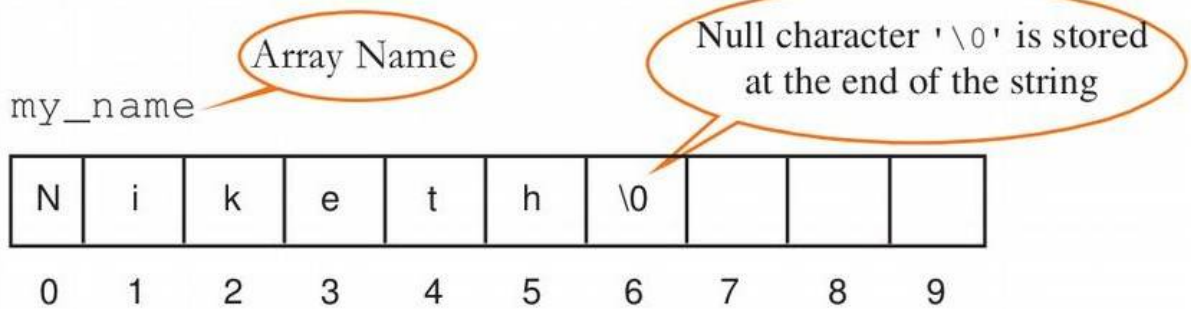
താഴെ പറയുന്ന രീതിയിലാണ് കാർക്ടർ അറേകൾ ഇനിഷ്യലൈസ് ചെയ്യുന്നത്

```
char my_name[10]={'N','i','k','e','t','h'};
```

Memory alloction

ഒരു സ്‌ട്രിങ്ങിനെ കാർക്ടർ അറേയിലേക്ക് ഇൻപുട്ട് ചെയ്യുമ്പോൾ താഴെ കാണുന്ന രീതിയിലാണ് മെമ്മറി അലോക്കേറ്റു ചെയ്യുന്നത്

'\0' എന്ന നൾ കാർക്ടർ സ്‌ട്രിങ്ങിന്റെ അവസാനം ഓട്ടോമാറ്റിക് ആയി കൂട്ടിച്ചേർക്കപ്പെടുന്നു.



അതായത് ഒരു സ്‌ട്രിങ്ങിനെ സ്റ്റോർ ചെയ്യാൻ ആ സ്‌ട്രിങ്ങിലുള്ള കാർക്ടറുകളുടെ എണ്ണത്തേക്കാൾ ഒരു ബെറ്റ് കൂടി അധികമെമ്മറി ആവശ്യമായി വരുന്നു. ('\0' ന് വേണ്ടി).

ഒരു കാർക്ടർ അറേയിലേക്ക് ഒരു സ്‌ട്രിങ്ങ് വില ഇനിഷ്യലൈസ് ചെയ്യുന്നതിന് തന്നിട്ടുള്ള രൂപം ഉപയോഗിക്കാം.

```
char my_name[10]="Niketh";
```

ഇൻപുട്ട് / ഔട്ട്പുട്ട് ഫങ്ഷൻസ്

സി ++ ൽ **ഇൻപുട്ട് / ഔട്ട്പുട്ട്** പ്രവർത്തനങ്ങൾ നിയന്ത്രിക്കുന്നത് ഒരു കൂട്ടം ഫംഗ്ഷനുകളാണ്, `iostream`, `cstdio` എന്നീ ഹെഡർ ഫയലുകളിലൂടെയാണ് ഇവ നിർവചിക്കപ്പെട്ടിരിക്കുന്നത്. `'cstdio'` ൽ നിർവചിച്ചിരിക്കുന്ന ഫംഗ്ഷനുകളെ കൺസോൾ I / O ഫംഗ്ഷനുകൾ എന്നും `'iostream'` ൽ നിർവചിച്ചിരിക്കുന്ന ഫംഗ്ഷനുകളെ സ്‌ട്രീം I / O ഫംഗ്ഷനുകൾ എന്നും വിളിക്കുന്നു.

കൺസോൾ I / O ഫംഗ്ഷനുകൾ

`stdio` എന്ന ഹെഡ്ഡർ ഫയലിൽ ആണ് ഇതിനെ നിർവചിച്ചിട്ടുള്ളത്

1. `gets()` ഫങ്ഷൻ (ഇൻപുട്ട് ഫങ്ഷൻ)

കീബോർഡിലൂടെ വൈറ്റ്സ്‌പേസ് ഉൾപ്പെടെയുള്ള ഒരുസ്‌ട്രിങ്ങിനെ കാർക്ടർ അറിയിലേക്കു സ്വീകരിക്കാൻ ഉപയോഗിക്കുന്ന ഒരു കൺസോൾ ഇൻപുട്ട് ഫങ്ഷനാണിത്. `cstdio` എന്ന ഹെഡ്ഡർ ഫയലിൽ ആണ് ഇവയെ നിവചിച്ചിട്ടുള്ളത് .

സിന്റാക്സ്

```
gets(character_array_name);
```

2. puts() ഫങ്ഷൻ (ഔട്ട്പുട്ട് ഫങ്ഷൻ)

മോണിറ്ററിൽ സ്ക്രീൻ ഡാറ്റ പ്രദർശിപ്പിക്കാൻ ഉപയോഗിക്കുന്നു

സിന്റാക്സ്

puts(string_data);

കാരക്ടർ I / O കൺസോൾഫംഗ്ഷനുകൾ

ഫങ്ഷൻ	സിന്റാക്സ്	പ്രവർത്തനം
getchar()	ch=getchar();	ch എന്ന വേരിയബിളിലേക്ക് കീബോർഡിലൂടെ ഒരു കാരക്ടറിനെ ഇൻപുട്ട് ചെയ്യുന്നു
putchar()	putchar(ch); putchar('A');	ch എന്ന വേരിയബിളിലെ കണ്ടെന്റ് ഡിസ്പ്ലൈ ചെയ്യുന്നു 'A' എന്ന കാരക്ടർ ഡിസ്പ്ലൈ ചെയ്യുന്നു

സ്ക്രീം I / O ഫംഗ്ഷനുകൾ

iostream എന്ന ഹെഡ്ഡർ ഫയലിൽ ആണ് ഇതിനെ നിർവചിച്ചിട്ടുള്ളത്

ഇൻപുട്ട് ഫങ്ഷൻസ്

ഫങ്ഷൻ	സിന്റാക്സ്	പ്രവർത്തനം
get()	cin.get(ch) cin.get(str,10);	ഒരു കാരക്ടറിനെ ch എന്ന വേരിയബിളിൽ സൂക്ഷിക്കുന്നു പത്തു കാരക്ടർ വരെയുള്ള ഒരു സ്ക്രീനിനെ കാരക്ടർ അറയിലേക്കു സ്വീകരിക്കുന്നു
getline()	cin.getline(str,10); cin.getline(str,10,ch);	പത്തു കാരക്ടർ വരെയുള്ള ഒരു സ്ക്രീനിനെ കാരക്ടർ അറയിലേക്കു സ്വീകരിക്കുന്നു പത്തു കാരക്ടർ വരെയുള്ള ഒരു സ്ക്രീനിനെ കാരക്ടർ അറയിലേക്കു സ്വീകരിക്കുന്നു അല്ലെങ്കിൽ ch എന്ന character വേരിയബിളിൽ പറഞ്ഞിരിക്കുന്ന വാല്യു വരെയുള്ള അക്ഷരങ്ങളെ സ്വീകരിക്കുന്നു. ഏതാണ് ആദ്യം സംഭവിക്കുന്നത് എന്നത് അനുസരിച്ച്

ഔട്ട്പുട്ട് ഫങ്ഷൻസ്

ഫങ്ഷൻ	സിന്റാക്സ്	പ്രവർത്തനം
put()	cout.put(ch);	ch ലെ കണ്ടെന്റ് ഡിസ്പ്ലൈ ചെയ്യുന്നു
	cout.put('B');	B ഡിസ്പ്ലൈ ചെയ്യുന്നു
	cout.put(65);	A ഡിസ്പ്ലൈ ചെയ്യുന്നു (65 is ASCII value of A)
write()	cout.write(str,10);	str എന്ന അറയിലെ ആദ്യത്തെ 10 കാരക്ടർ ഡിസ്പ്ലൈ ചെയ്യുന്നു

ചോദ്യങ്ങൾ

1. put() ഉം write() ഉം തമ്മിലുള്ള വ്യത്യാസം ഉദാഹരണ സഹിതം വിവരിക്കുക
2. "HELLO WORLD" എന്ന സ്ക്രീൻ സംഭരിക്കാൻ എത്ര ബൈറ്റുകൾ ആവശ്യമാണ്?
3. സി ++ ൽ ഉപയോഗിക്കുന്ന രണ്ട് ഇൻപുട്ട് സ്ക്രീം ഫംഗ്ഷനുകളുടെ പേര് എഴുതുക .
4. മെമ്മറിയിൽ സ്ക്രീൻ ഡിലിമിറ്റ് ചെയ്യാൻ ഉപയോഗിക്കുന്ന കാരക്ടർ ഏതാണ് ?
5. സ്ക്രീൻ ഡാറ്റ ഇൻപുട്ട് ചെയ്യുന്നതിന് സി ++ പ്രോഗ്രാമിൽ gets () ഫംഗ്ഷൻ ഉപയോഗിക്കുന്നതിന്റെ പ്രയോജനം എന്താണ്? ഒരു ഉദാഹരണം ഉപയോഗിച്ച് വിശദീകരിക്കുക.

ഫങ്ഷനുകൾ

മോഡുലാർ പ്രോഗ്രാമിംഗിന്റെ ആശയം: പ്രോഗ്രാമിംഗിൽ, വലിയ പ്രോഗ്രാമുകളെ ചെറിയ ഉപ പ്രോഗ്രാമുകളായി വിഭജിക്കപ്പെടും, അവ പ്രത്യേക പ്രോഗ്രാമുകൾ എഴുതുന്നതിലൂടെ പരിഹരിക്കാനാകും. ഇത്തരത്തിലുള്ള സമീപനത്തെ മോഡുലാർ പ്രോഗ്രാമിംഗ് എന്ന് വിളിക്കുന്നു. വലിയ പ്രോഗ്രാമുകളെ ചെറിയ ഉപ പ്രോഗ്രാമുകളായി വിഭജിക്കുന്ന പ്രക്രിയയെ മോഡുലറെസേഷൻ എന്ന് വിളിക്കുന്നു.

മോഡുലാർ പ്രോഗ്രാമിംഗിന്റെ ഗുണങ്ങൾ:

- ➔ പ്രോഗ്രാമിന്റെ വലുപ്പം കുറയ്ക്കുന്നു
- ➔ പിശകിനുള്ള സാധ്യത കുറവാണ്
- ➔ പ്രോഗ്രാമിംഗ് സങ്കീർണ്ണത കുറയ്ക്കുന്നു
- ➔ വീണ്ടും ഉപയോഗക്ഷമത മെച്ചപ്പെടുത്തുന്നു

മോഡുലാർ പ്രോഗ്രാമിംഗിന്റെ കുറവുകൾ:

- ➔ ഉപ പ്രോഗ്രാമുകളായി വിഭജിക്കുക എന്നത് ഒരു വെല്ലുവിളിയാണ്.
- ➔ മൊഡ്യൂളുകളുടെ എക്സിക്യൂഷന്റെ ശ്രേണി ക്രമീകരിക്കുമ്പോൾ ശ്രദ്ധിക്കണം.

ഫംഗ്ഷൻ: ഒരു നിർദ്ദിഷ്ട ചുമതല നിർവഹിക്കുന്നതിന് ഒരു സി ++ പ്രോഗ്രാമിലെ നെയ്മിഡ് യൂണിറ്റാണ് ഫംഗ്ഷൻ.

ബിൽറ്റ്-ഇൻ ഫംഗ്ഷനുകൾ: ഉപയോഗത്തിന് തയ്യാറായ ഫംഗ്ഷനുകൾ ബിൽറ്റ്-ഇൻ ഫംഗ്ഷനുകളാണ്.

1. സ്ട്രിംഗ് ഫംഗ്ഷനുകൾ: ഇവ സ്ട്രിംഗുകളെ കൈകാര്യം ചെയ്യുന്നതിന് ഉപയോഗിക്കുന്നു. ഈ ഫംഗ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് സി ++ പ്രോഗ്രാമിൽ `cstring` (ടർബോ സി ++ ലെ `string.h`) എന്ന ഹെഡ്ഡർ ഫയൽ ഉൾപ്പെടുത്തണം.

ഫംഗ്ഷൻ	സിന്റാക്സ്	പ്രവർത്തനം
strlen()	<code>int strlen(string);</code>	ഒരു സ്ട്രിംഗിന്റെ ദൈർഘ്യം കണ്ടെത്താൻ (ഒരു സ്ട്രിംഗിലെ അക്ഷരങ്ങളുടെ എണ്ണം)
strcpy()	<code>strcpy(string1,string2);</code>	ഒരു സ്ട്രിംഗ് മറ്റൊന്നിലേക്ക് പകർത്താൻ. ഇവിടെ സ്ട്രിംഗ് 2 സ്ട്രിംഗ് 1 ലേക്ക് പകർത്തുന്നു
strcmp()	<code>strcmp(string1,string2);</code>	രണ്ട് സ്ട്രിംഗുകൾ താരതമ്യം ചെയ്യാൻ, കേസ് സെൻസിറ്റീവ്. രണ്ട് സ്ട്രിംഗുകളും തുല്യമാണെങ്കിൽ 0 റിട്ടേൺ ചെയ്യുന്നു. സ്ട്രിംഗ് 1 സ്ട്രിംഗ് 2 നെക്കാൾ അക്ഷരമാലാക്രമത്തിൽ ഉയർന്നതാണെങ്കിൽ, ഒരു +ve മൂല്യം നൽകുന്നു, അല്ലെങ്കിൽ ഒരു -ve മൂല്യം നൽകുന്നു.
strcmpi()	<code>strcmpi(string1,string2);</code>	രണ്ട് സ്ട്രിംഗുകൾ താരതമ്യം ചെയ്യാൻ, കേസ് സെൻസിറ്റീവ് അല്ല, രണ്ട് സ്ട്രിംഗുകളും തുല്യമാണെങ്കിൽ 0 റിട്ടേൺ ചെയ്യുന്നു. സ്ട്രിംഗ് 1 സ്ട്രിംഗ് 2 നെക്കാൾ അക്ഷരമാലാക്രമത്തിൽ ഉയർന്നതാണെങ്കിൽ, ഒരു +ve മൂല്യം നൽകുന്നു, അല്ലെങ്കിൽ ഒരു -ve മൂല്യം നൽകുന്നു.
strcat()	<code>strcat(string1, string2);</code>	ഒരു സ്ട്രിംഗ് മറ്റൊരു സ്ട്രിംഗിലേക്ക് കൂട്ടിച്ചേർക്കാൻ. ഇവിടെ സ്ട്രിംഗ് 2 സ്ട്രിംഗ് 1 ലേക്ക് കൂട്ടിച്ചേർക്കുന്നു

2. മാത്തമാറ്റിക്കൽ ഫംഗ്ഷനുകൾ: മാത്തമാറ്റിക്കൽ കാൽക്കുലേഷൻ ഉപയോഗിക്കുന്നു. പ്രോഗ്രാമിൽ ഈ ഫംഗ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് ഹെഡ്ഡർ ഫയൽ cmath (ടർബോ സി ++ ലെ math.h) ഉൾപ്പെടുത്തണം.

ഫംഗ്ഷൻ	സിന്റാക്സ്	പ്രവർത്തനം
abs()	int abs(int);	ഒരു സംഖ്യയുടെ സമ്പൂർണ്ണ മൂല്യം കണ്ടെത്താൻ. ഉദാ. abs(-25) എന്നത് 25 ആണ്.
fabs()	double fabs(double);	ഒരു ഫ്ലോട്ടിംഗ് പോയിന്റ് നമ്പറിന്റെ കേവല മൂല്യം കണ്ടെത്തുന്നതിന് ഉദാ. fabs(-25.89) എന്നത് 25.89 ആണ്
sqrt()	double sqrt(double);	ഒരു സംഖ്യയുടെ റൂട്ട് കണ്ടെത്തുന്നതിന്. ഉദാ: sqrt (100) എന്നത് 10 ആണ്.
pow()	double pow(double,int);	ഒരു സംഖ്യയുടെ പവർ കണ്ടെത്താൻ. ഉദാ. pow (3,2) എന്നത് 3 ² അതായത് 9 ആണ്
sin()	double sin(double);	ഒരു കോണിന്റെ സൈൻ മൂല്യം കണ്ടെത്താൻ.
cos()	double cos(double);	ഒരു കോണിന്റെ കോസൈൻ മൂല്യം കണ്ടെത്താൻ.

3. കാരക്ടർ ഫംഗ്ഷനുകൾ: കാരക്ടറുകളുടെ വിവിധ പ്രവർത്തനങ്ങൾ നടത്താൻ ഈ ഫംഗ്ഷനുകൾ ഉപയോഗിക്കുന്നു. ഒരു പ്രോഗ്രാമിൽ ഈ ഫംഗ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് ഹെഡ്ഡർ ഫയൽ ctype (ടർബോ സി ++ ൽ ctype.h) ഉൾപ്പെടുത്തണം.

ഫംഗ്ഷൻ	സിന്റാക്സ്	പ്രവർത്തനം
isupper()	int isupper(char c);	ഒരു അക്ഷരം വലിയക്ഷരത്തിലാണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നു . നൽകിയ അക്ഷരം വലിയ ക്ഷരത്തിലാണെങ്കിൽ 1 റിട്ടേൺ ചെയ്യുന്നു അല്ലെങ്കിൽ 0 റിട്ടേൺ ചെയ്യുന്നു.
islower()	int islower(char c);	ഒരു അക്ഷരം ചെറിയ ക്ഷരത്തിലാണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നു . നൽകിയ അക്ഷരം ചെറിയ ക്ഷരത്തിലാണെങ്കിൽ 1 റിട്ടേൺ ചെയ്യുന്നു അല്ലെങ്കിൽ 0 റിട്ടേൺ ചെയ്യുന്നു.
isalpha()	int isalpha(char c);	തന്നിരിക്കുന്ന കാരക്ടർ ആൽഫബെറ്റ് ആണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നതിന്. നൽകിയ കാരക്ടർ ആൽഫബെറ്റ് ആണെങ്കിൽ 1 റിട്ടേൺ ചെയ്യുന്നു അല്ലെങ്കിൽ 0 റിട്ടേൺ ചെയ്യുന്നു.
isdigit()	int isdigit(char c);	തന്നിരിക്കുന്ന കാരക്ടർ ഒരു അക്കമാണോ എന്ന് പരിശോധിക്കുന്നതിന്. അക്കമാണെങ്കിൽ 1 റിട്ടേൺ ചെയ്യുന്നു അല്ലെങ്കിൽ 0 റിട്ടേൺ ചെയ്യുന്നു.
isalnum()	int isalnum (char c);	ഒരു കാരക്ടർ ഒരു ആൽഫാന്യൂമെറിക് ആണോ എന്ന് പരിശോധിക്കുന്നതിന്. ആൽഫാന്യൂമെറിക് ആണെങ്കിൽ 1 റിട്ടേൺ ചെയ്യുന്നു അല്ലെങ്കിൽ 0 റിട്ടേൺ ചെയ്യുന്നു.
toupper()	char toupper(char c);	തന്നിരിക്കുന്ന അക്ഷരത്തെ അതിന്റെ വലിയക്ഷരത്തിലേക്ക് പരിവർത്തനം ചെയ്യാൻ. തന്നിരിക്കുന്ന അക്ഷരത്തിന്റെ വലിയക്ഷരം റിട്ടേൺ ചെയ്യുന്നു തന്നിരിക്കുന്ന അക്ഷരം വലിയക്ഷരത്തിലാണെങ്കിൽ, ഔട്ട്പുട്ട് സമാനമായിരിക്കും.
tolower()	char tolower(char c);	തന്നിരിക്കുന്ന അക്ഷരത്തെ അതിന്റെ ചെറിയക്ഷരത്തിലേക്ക് പരിവർത്തനം ചെയ്യാൻ. തന്നിരിക്കുന്ന അക്ഷരത്തിന്റെ ചെറിയക്ഷരം റിട്ടേൺ ചെയ്യുന്നു തന്നിരിക്കുന്ന അക്ഷരം ചെറിയ അക്ഷരത്തിലാണെങ്കിൽ, ഔട്ട്പുട്ട് സമാനമായിരിക്കും.

യൂസർ ഡിഫൈൻഡ് ഫംഗ്ഷൻ:

യൂസർ നിർമ്മിക്കുന്ന ഫംഗ്ഷനുകളെ യൂസർ ഡിഫൈൻഡ് ഫംഗ്ഷൻ എന്ന് പറയുന്നു

സിന്റാക്സ്

```
data_type function_name(argument_list)
{
    statements in the body;
}
```

ഡാറ്റ ടൈപ്പ് എന്നാൽ സി ++ ലെ ഏതെങ്കിലും സാധുവായ ഡാറ്റ ടൈപ്പ് ആണ് . യൂസർ നൽകുന്ന പദമാണ് ഫംഗ്ഷൻ നെയിം (ഐഡന്റിഫയർ). ആർഗ്യുമെന്റ്_ലിസ്റ്റ്, ഓപ്ഷണലാണ്, ഫംഗ്ഷൻ ബോഡിയിൽ ഫംഗ്ഷൻ നൽകിയിട്ടുള്ള ടാസ്ക് നിർവഹി ക്കുന്നതിന് ആവശ്യമായ സി ++ സ്റ്റേറ്റ്‌മെന്റുകൾ അടങ്ങിയിരിക്കുന്നു.

ഉദാഹരണം :

```
int sum(int a,int b)
{
    c=a+b;
}
```

ഫംഗ്ഷനുകളുടെ ആർഗ്യുമെന്റുകൾ: കോളിംഗ് ഫംഗ്ഷനിൽ നിന്ന് വിളിക്കുന്ന ഫംഗ്ഷനിലേക്ക് മൂല്യങ്ങൾ കൈമാറുന്നതിനുള്ള മാർഗമാണ് ആർഗ്യുമെന്റുകൾ അല്ലെങ്കിൽ പാരാമീറ്ററുകൾ. ഫംഗ്ഷൻ നിർവചനത്തിൽ ആർഗ്യുമെന്റുകളായി ഉപയോഗിക്കുന്ന വേരിയബിളുകളെ ഫോർമൽ ആർഗ്യുമെന്റുകൾ എന്ന് വിളിക്കുന്നു. ഫംഗ്ഷൻ കോളിൽ ഉപയോഗിക്കുന്ന കോൺസ്റ്റന്റ്സ് , വേരിയബിളുകൾ അല്ലെങ്കിൽ എക്സ്‌പ്രഷനുകൾ ആക്ഷുൽ ആർഗ്യുമെന്റുകൾ എന്നറിയപ്പെടുന്നു.

റിട്ടേൺ മൂല്യം: റിട്ടേൺ സ്റ്റേറ്റ്‌മെന്റ് കോളിംഗ് ഫംഗ്ഷന് ഒരു മൂല്യം നൽകുകയും പ്രോഗ്രാം നിയന്ത്രണം കോളിംഗ് ഫംഗ്ഷനിലേക്ക് മാറ്റുകയും ചെയ്യുന്നു.

ചോദ്യങ്ങൾ

1. താഴെ പറയുന്ന പ്രവർത്തികൾ ചെയ്യുവാൻ ഏറ്റവും അനുയോജ്യമായ സി++ ലെ ബിൽട് ഇൻ ഫങ്ഷനുകൾ നിർദ്ദേശിക്കുക
 - a. 5³ ന്റെ ഉത്തരം ലഭിക്കാൻ.
 - b. "KERALA" എന്ന സ്ട്രിങ്ങിലെ അക്ഷരങ്ങളുടെ എണ്ണം ലഭിക്കാൻ.
 - c. 'M' എന്ന അക്ഷരത്തെ 'm' ആക്കി മാറ്റുന്നതിന്.
2. ഒരു ഫങ്ഷനിൽ ആർഗ്യുമെന്റുകൾ ഉപയോഗിക്കുന്നതിന്റെ ഉദ്ദേശം എന്താണ് ?
3. മാത്തമറ്റിക്കൽ ഫങ്ഷൻ ആവശ്യമുള്ള ഹെഡർ ഫയൽ ഏത് ?
4. മോഡ്യൂലർ പ്രോഗ്രാമിങ് എന്ന് പറഞ്ഞാൽ എന്താണ് ? മോഡ്യൂലർ പ്രോഗ്രാമിന്റെ ഗുണങ്ങൾ എന്തൊക്കെയാണ് ?
5. ഒരു ക്യാരക്ടർ ആൽഫ ന്യൂമെറിക് ആണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്ന ഫങ്ഷൻ ഏതാണ് ?
6. strlen() ഫങ്ഷൻ ഉപയോഗിച്ച് ഒരു സ്ട്രിങ്ങിന്റെ ലെങ്ത് കാനാനുള്ള പ്രോഗ്രാം എഴുതുക
7. strcmp() ഉം strcmpi() ഉം തമ്മിൽ താരതമ്യം ചെയ്യുക.

കമ്പ്യൂട്ടർ ശൃംഖലകൾ

കമ്പ്യൂട്ടർ ശൃംഖലകൾ

ഒരു ആശയവിനിമയ മാധ്യമം ഉപയോഗിച്ച് കമ്പ്യൂട്ടറുകളുടെയും പ്രിന്ററുകൾ, സ്കാനറുകൾ തുടങ്ങിയ ഹാർഡ്‌വെയർ ഉപകരണങ്ങളുടെയും ഇന്റർ കണക്ഷനാണ് ഇത്.

Advantages

1) റിസോഴ്സ് പങ്കിടൽ - ഒരു സിസ്റ്റത്തിലെ ഏത് ഹാർഡ്‌വെയർ / സോഫ്റ്റ് വെയർ റിസോഴ്സും നെറ്റ് വർക്കിലെ മറ്റ് സിസ്റ്റങ്ങളുമായി പങ്കിടാൻ കഴിയും.

2) വില-പ്രകടന അനുപാതം - ഓരോ കമ്പ്യൂട്ടറിനുംലൈസൻസുള്ള സോഫ്റ്റ് വെയർ വാങ്ങുന്നതിനുള്ള ചെലവ്, ഒരു നെറ്റ്‌വർക്കിൽഅത്തരംസോഫ്റ്റ് വെയറിന്റെ നെറ്റ്‌വർക്ക് പതിപ്പുകൾ വാങ്ങുന്നതിലൂടെ കുറയ്ക്കാൻ കഴിയും.

3) ആശയവിനിമയം - ഇ-മെയിൽ, ചാറ്റിംഗ്, വീഡിയോകോൺഫറൻസിംഗ് തുടങ്ങിയ സേവനങ്ങളിലൂടെനെറ്റ്‌വർക്കിലെമറ്റേതെങ്കിലും കമ്പ്യൂട്ടറുമായി ആശയവിനിമയം നടത്താൻ കമ്പ്യൂട്ടർ നെറ്റ്‌വർക്ക് ഉപയോക്താക്കളെ സഹായിക്കുന്നു.

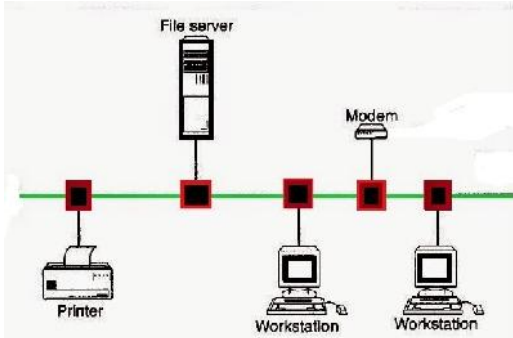


Fig 11.1 Computer Networks

4) വിശ്വാസ്യത- ഒരു നെറ്റ്‌വർക്കിൽ, ഒന്നിലധികം കമ്പ്യൂട്ടറുകളിൽ ഡാറ്റ ബാക്കപ്പ് ചെയ്യുന്നത് സാധ്യമാണ്. ഡാറ്റ ആക്സസ് ചെയ്യുന്നതിൽ പരാജയപ്പെട്ടാൽ ഡാറ്റ വീണ്ടെടുക്കാൻ ഇത് ഉപയോക്താക്കളെ സഹായിക്കുന്നു.

5) സ്കേലബിലിറ്റി - നെറ്റ്‌വർക്കിലേക്ക് കമ്പ്യൂട്ടറുകൾ ചേർക്കുന്നതിലൂടെയോ നീക്കംചെയ്യുന്നതിലൂടെയോ കമ്പ്യൂട്ടിംഗ് ശേഷി എളുപ്പത്തിൽ വർദ്ധിപ്പിക്കുകയോ കുറയ്ക്കുകയോ ചെയ്യാം.

Some key terms

Bandwidth

ഒരു നിശ്ചിത സമയത്തിനുള്ളിൽ ഒരു നിർദ്ദിഷ്ട കണക്ഷനിലൂടെ അയയ്ക്കാൻ കഴിയുന്ന ഡാറ്റയുടെ അളവിനെ ബാൻഡ് വിഡ്ത് എന്ന് പറയുന്നു

Noise:

അനാവശ്യ വൈദ്യുത അല്ലെങ്കിൽ വൈദ്യുതകാന്തിക എന്നർജിയെയാണ് നോയിസ് എന്ന് പറയുന്നത് . ഇത് ഡാറ്റ സിഗ്നലുകളുടെ അളവ് കുറയ്ക്കുന്നു .

Node:

ഒരു നെറ്റ്‌വർക്കിലേക്ക് നേരിട്ട് കണക്റ്റുചെയ്തിരിക്കുന്ന ഏത് ഉപകരണത്തെയും നോഡ് എന്ന് വിളിക്കുന്നു.

Data communication devices:

ഒരു കമ്പ്യൂട്ടറും ആശയവിനിമയ ചാനലും തമ്മിലുള്ള ഒരു ഇന്റർഫേസ് നൽകുന്നു.

1) Switch:

- ഒരു നെറ്റ്‌വർക്ക് രൂപീകരിക്കുന്നതിന് നിരവധി കമ്പ്യൂട്ടറുകളെ ബന്ധിപ്പിക്കുന്ന ഒരു ഉപകരണമാണ് സ്വിച്ച്.
- ഇത് ഒരു ബുദ്ധിമാനായ ഉപകരണമാണ്, കാരണം സ്വീകരിച്ച ഡാറ്റാ പാക്കറ്റുകൾക്ക് മാത്രം കൈമാറാൻ ഇതിന് കഴിയും.
- ഇത് കണക്റ്റുചെയ്തിരിക്കുന്ന എല്ലാ ഉപകരണങ്ങളുടെയും വിലാസങ്ങൾ സംഭരിക്കും.

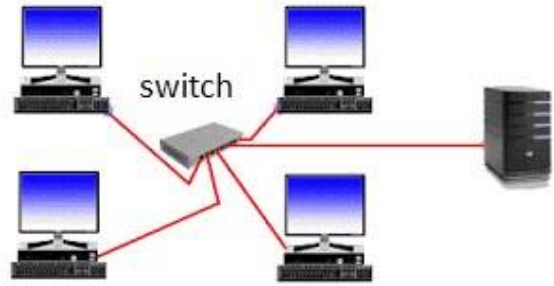


Fig 11.2 Switch

2) Bridge

- ഒരു ശൃംഖലയെ തരംതിരിക്കുന്നതിന് ഉപയോഗിക്കുന്ന ഉപകരണമാണ് ബ്രിഡ്ജ്.
- ഒരു നെറ്റ്‌വർക്കിനെ വ്യത്യസ്ത സെഗ്മെന്റുകളായി വിഭജിക്കാനും ബ്രിഡ്ജ് ഉപയോഗിച്ച് പരസ്പരം ബന്ധിപ്പിക്കാനും കഴിയും.
- ഇത് ഒരു നെറ്റ്‌വർക്കിലെ ട്രാഫിക്സിന്റെ അളവ് കുറയ്ക്കുന്നു.

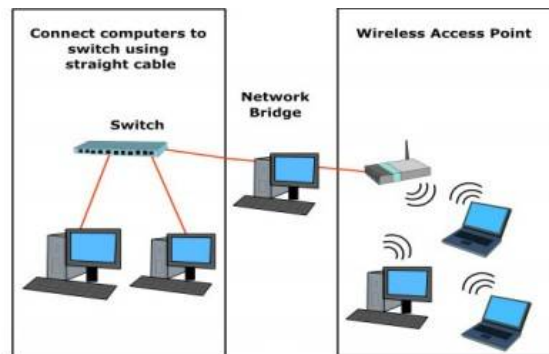


Fig 11.4 Bridge

3) Router

- ഒരേ പ്രോട്ടോക്കോൾ ഉപയോഗിച്ച് ഒരേ തരത്തിലുള്ള രണ്ട് നെറ്റ്‌വർക്കുകൾ പരസ്പരം ബന്ധിപ്പിക്കാൻ കഴിയുന്ന ഒരു ഉപകരണമാണ് റൂട്ടർ.
- ഡാറ്റാ പാക്കറ്റുകൾക്ക് യാത്ര ചെയ്യുന്നതിനും ഒരു നെറ്റ്‌വർക്കിലെ ട്രാഫിക്സിന്റെ അളവ് കുറയ്ക്കുന്നതിനും ഇത് അനുയോജ്യമായ പാത കണ്ടെത്താനാകും.

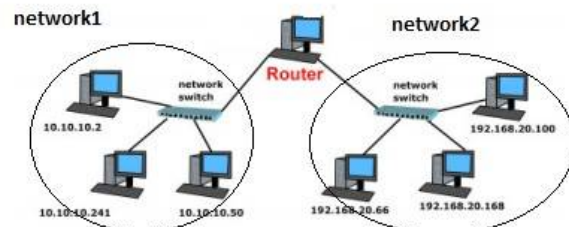


Fig 11.5 Router

4) Gateway :

- വ്യത്യസ്ത പ്രോട്ടോക്കോൾ ഉള്ള രണ്ട് വ്യത്യസ്ത നെറ്റ്‌വർക്കുകൾ പരസ്പരം ബന്ധിപ്പിക്കാൻ കഴിയുന്ന ഒരു ഉപകരണമാണ് ഗേറ്റ്‌വേ.
- ഇതിന് ഒരു പ്രോട്ടോക്കോൾ മറ്റൊന്നിലേക്ക് വിവർത്തനം ചെയ്യാൻ കഴിയും.
- ലക്ഷ്യസ്ഥാനത്തെത്താൻ പാക്കറ്റുകൾക്ക് അനുയോജ്യമായ പാത ഇതിന് കണ്ടെത്താനാകും.

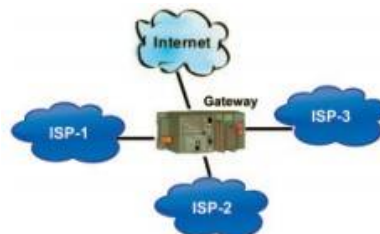


Fig 11.6 Gateway

Data Terminal Equipments:

ഒരു കമ്പ്യൂട്ടറിലേക്കോ അതിൽ നിന്നോ ഉള്ള ഡാറ്റ നിയന്ത്രിക്കുന്ന ഒരു ഉപകരണമാണ് ഡാറ്റ ടെർമിനൽ ഉപകരണങ്ങൾ.

1) Modem :

- ടെലിഫോൺ ലൈനുകൾ വഴി കമ്പ്യൂട്ടറുകൾ തമ്മിലുള്ള ആശയവിനിമയത്തിന് ഉപയോഗിക്കുന്ന ഉപകരണമാണ് മോഡം.
- മോഡുലേറ്റർ, ഡെമോഡുലേറ്റർ എന്നിവയിൽ നിന്നാണ് പേര് രൂപപ്പെടുന്നത്.
- ഇത് ഒരു കമ്പ്യൂട്ടറിൽ നിന്ന് ലഭിച്ച ഡിജിറ്റൽ സിഗ്നലുകളെ ടെലിഫോൺ ലൈനുകൾക്കായുള്ള അനലോഗ് സിഗ്നലുകളിലേക്കും തിരിച്ചും പരിവർത്തനം ചെയ്യുന്നു.

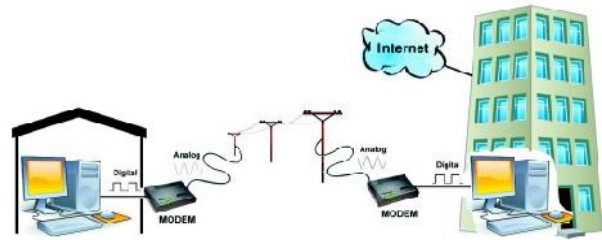


Fig 11.7 Modem

Network topologies Topology :

ഒരു നെറ്റ്‌വർക്ക് രൂപീകരിക്കുന്നതിന് നോഡുകൾ പരസ്പരം ബന്ധിപ്പിച്ചിരിക്കുന്ന രീതി. ബസ്, സ്റ്റാർ, റിംഗ്, മെഷ് എന്നിവയാണ് പ്രധാന ടോപ്പോളജികൾ.

1)Bus topology:

- ബസ് ടോപ്പോളജിയിൽ എല്ലാ നോഡുകളും ബസ് എന്ന പ്രധാന കേബിളുമായി ബന്ധിപ്പിച്ചിരിക്കുന്നു.
- ടെർമിനേറ്റർ എന്ന് വിളിക്കുന്ന ഒരു ചെറിയ ഉപകരണം ബസിന്റെ ഓരോ അറ്റത്തും ഘടിപ്പിച്ചിരിക്കുന്നു.

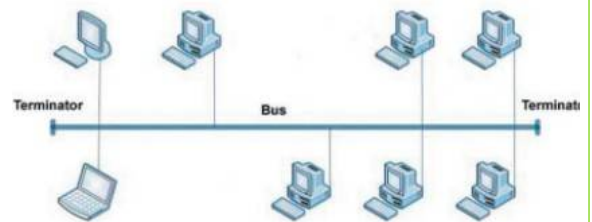


Fig 11.8 Bus topology

ഒരു നോഡിന് മറ്റൊരു നോഡിലേക്ക് ഡാറ്റ അയയ്ക്കേണ്ടിവന്നാൽ, അത് ബസിലേക്ക് ഡാറ്റ അയയ്ക്കുന്നു. സിഗ്നൽ ബസ്സിലൂടെ സഞ്ചരിക്കുന്നു, ഓരോ നോഡും ബസ് പരിശോധിക്കുന്നു, ഉദ്ദേശിച്ച നോഡ് മാത്രമേ ഡാറ്റ സ്വീകരിക്കുകയുള്ളൂ. സിഗ്നൽ ബസിന്റെ അവസാനത്തിൽ എന്തുവോൾ, ടെർമിനേറ്റർ ബസ്സിൽ നിന്നുള്ള സിഗ്നൽ ആഗിരണം ചെയ്യുന്നു.

Characteristics of bus topology

- 1) ഇൻസ്റ്റാൾ ചെയ്യാൻ എളുപ്പമാണ്.
- 2) കേബിൾ ദൈർഘ്യം കുറവാണ്, അതിനാൽ ഇത് ചെലവ് കുറഞ്ഞതാണ്.
- 3) ഒരു നോഡിന്റെ പരാജയം നെറ്റ്‌വർക്കിനെ ബാധിക്കില്ല.
- 4) കേബിൾ (ബസ്) അല്ലെങ്കിൽ ടെർമിനേറ്റർ എന്നിവയുടെ പരാജയം മുഴുവൻ നെറ്റ്‌വർക്കിന്റെയും തകർച്ചയിലേക്ക് നയിക്കുന്നു.
- 5) തെറ്റുകൾ നിർണ്ണയിക്കാൻ ബുദ്ധിമുട്ടാണ്.
- 6) ഒരു സമയം ഒരു നോഡിന് മാത്രമേ ഡാറ്റ കൈമാറാൻ കഴിയൂ.

2) Star topology

- സ്റ്റാർ ടോപ്പോളജിയിൽ ഓരോ നോഡും നേരിട്ട് ഒരു ഹബ്ബ് / സ്വിച്ചിലേക്ക് ബന്ധിപ്പിച്ചിരിക്കുന്നു.
- ഏതെങ്കിലും നോഡിന് മറ്റേതെങ്കിലും നോഡിലേക്ക് കുറച്ച് വിവരങ്ങൾ അയയ്ക്കേണ്ടിവന്നാൽ, അത് ഹബ്ബ് / സ്വിച്ചിലേക്ക് സിഗ്നൽ അയയ്ക്കുന്നു.
- സിഗ്നൽ എല്ലാ നോഡുകളിലേക്കും (ഒരു ഹബ്ബിന്റെ കാര്യത്തിൽ) പ്രക്ഷേപണം ചെയ്യുന്നു, പ്രക്ഷേപിച്ച നോഡ് മാത്രമേ സ്വീകരിക്കുകയുള്ളൂ.
- ഒരു സ്വിച്ചിന്റെ കാര്യത്തിൽ, ഉദ്ദേശിച്ച നോഡിലേക്ക് മാത്രമേ സിഗ്നൽ അയയ്ക്കൂ.

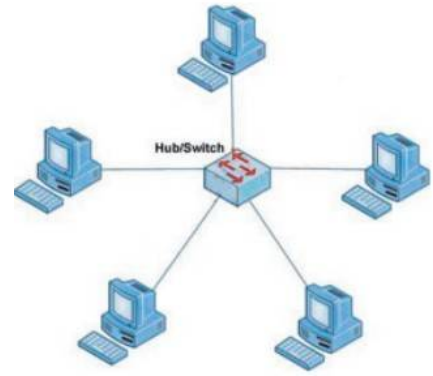


Fig 11.9 Star topology

Characteristics of star topology

- 1) ബസ് ടോപ്പോളജിയുമായി താരതമ്യപ്പെടുത്തുമ്പോൾ കൂടുതൽ കാര്യക്ഷമമാണ്.
- 2) ഇൻസ്റ്റാൾ ചെയ്യാൻ എളുപ്പമാണ്.
- 3) തെറ്റുകൾ നിർണ്ണയിക്കാൻ എളുപ്പമാണ്.
- 4) സെൻട്രൽ ഹബ്ബ് / സ്വിച്ചിന്റെ സവിശേഷതകളെ ആശ്രയിച്ച് വികസിപ്പിക്കാൻ എളുപ്പമാണ്.
- 5) ഹബ്ബ് / സ്വിച്ച് പരാജയം മുഴുവൻ നെറ്റ് വർക്കിന്റെയും പരാജയത്തിലേക്ക് നയിക്കുന്നു.
- 6) ബസ് ടോപ്പോളജിയുമായി താരതമ്യപ്പെടുത്തുമ്പോൾ കൂടുതൽ കേബിൾ ദൈർഘ്യം ആവശ്യമാണ്.

3) Ring topology

- റിംഗ് ടോപ്പോളജിയിൽ എല്ലാ നോഡുകളും റിംഗ് അല്ലെങ്കിൽ സർക്കിൾ ലൂപ്പ് ചെയ്യുന്ന കേബിൾ ഉപയോഗിച്ച് ബന്ധിപ്പിച്ചിരിക്കുന്നു.
- ഒരു റിംഗ് ടോപ്പോളജി ഒരു സർക്കിളിന്റെ രൂപത്തിലാണ്.
- ഒരു റിംഗിൽ ഒരു ദിശയിൽ മാത്രമേ ഡാറ്റ സഞ്ചരിക്കൂ.
- ഓരോ നോഡും സിഗ്നലിനെ പുനരുജ്ജീവിപ്പിക്കുകയും ഉദ്ദേശിച്ച നോഡ് എത്തുന്നതുവരെ അടുത്ത നോഡിലേക്ക് കടക്കുകയും ചെയ്യുന്നു.

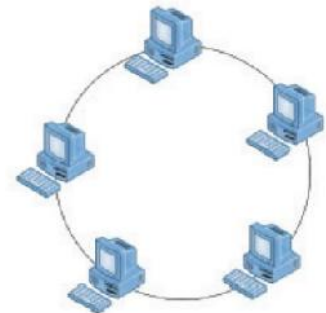


Fig 11.10 Ring topology

Characteristics of ring topology

- 1) ഓരോ നോഡും സിഗ്നൽ വർദ്ധിപ്പിക്കുന്നതിനാൽ സിഗ്നൽ ആംപ്ലിഫിക്കേഷൻ ആവശ്യമില്ല.
- 2) കേബിൾ ദൈർഘ്യം കുറവാണ്, അതിനാൽ ചെലവ് കുറവാണ്.
- 3) ഒരു നോഡ് പരാജയപ്പെടുകയാണെങ്കിൽ, മുഴുവൻ നെറ്റ് വർക്കും പരാജയപ്പെടും.
- 4) നെറ്റ് വർക്കിലേക്ക് നോഡുകൾ ചേർക്കുന്നത് ബുദ്ധിമുട്ടാണ്.

4) Mesh topology :

- മെഷ് ടോപ്പോളജിയിൽ, ഓരോ നോഡും മറ്റ് നോഡുകളുമായി ബന്ധിപ്പിച്ചിരിക്കുന്നു.
- രണ്ട് നോഡുകൾക്കിടയിൽ ഒന്നിൽ കൂടുതൽ പാത ഉണ്ടാകും.
- ഒരു പാത്ത് പരാജയപ്പെട്ടാൽ, ഡാറ്റ മറ്റൊരു പാതയിലൂടെ ലക്ഷ്യസ്ഥാനത്ത് എത്തും

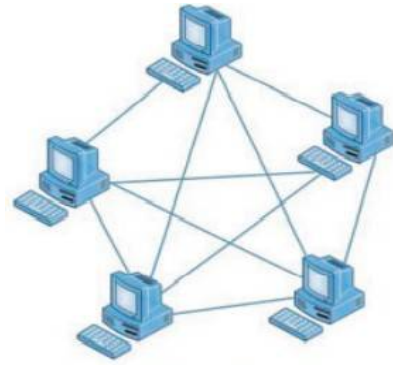


Fig 11.11 Mesh topology

Characteristics of mesh topology

- 1) നോഡുകൾക്കിടയിലുള്ള ഒരു പാത പരാജയപ്പെട്ടാലും നെറ്റ്‌വർക്ക് പരാജയപ്പെടുകയില്ല.
- 2) അധിക കേബിളുകൾ ആവശ്യമുള്ളതിനാൽ ചെലവേറിയത്.
- 3) വളരെ സങ്കീർണ്ണവും കൈകാര്യം ചെയ്യാൻ പ്രയാസവുമാണ്.

Identification of Computers on a network

1) Media Access Control (MAC) address:

- i. ഓരോ എൻഐസിക്വം (നെറ്റ്‌വർക്ക് ഇന്റർഫേസ് കാർഡ്) അതിന്റെ നിർമ്മാതാവ് നൽകിയിട്ടുള്ള സാർവ്വത്രികമായി സവിശേഷമായ അഡ്രസ്സ് (12 അക്ക ഹെക്സാഡെസിമൽ നമ്പർ).
- ii. MAC വിലാസങ്ങൾ സാധാരണയായി ഇനിപ്പറയുന്ന രണ്ട് ഫോർമാറ്റുകളിൽ ഒന്നിൽ എഴുതുന്നു:
- iii. MM: MM: MM: SS: SS: SS അല്ലെങ്കിൽ MM - MM - MM - SS - SS - SS
- iv. ഒരു MAC അഡ്രസിന്റെ ആദ്യ പകുതിയിൽ (MM: MM: MM) അഡാപ്റ്റർ നിർമ്മാതാവിന്റെ ഐഡി നമ്പർ അടങ്ങിയിരിക്കുന്നു.
- v. ഒരു MAC അഡ്രസിന്റെ രണ്ടാം പകുതി (SS: SS: SS) അതിന്റെ നിർമ്മാതാവ് അഡാപ്റ്ററിലേക്ക് (എൻഐസി) നിയോഗിച്ചിട്ടുള്ള സീരിയൽ നമ്പറിനെ പ്രതിനിധീകരിക്കുന്നു.

eg. 00:A0:C9 : 14:C8:35

2) IP address:

- i. ഒരു നെറ്റ്‌വർക്കിലെ ഓരോ നോഡിനും അവരുടെ അദ്വിതീയ തിരിച്ചറിയലിനായി നിയുക്തമാക്കിയിരിക്കുന്ന 4 ഭാഗങ്ങളുള്ള ഒരു അദ്വിതീയ വിലാസമാണ് ഐപി അഡ്രസ്സ് .
- ii. നാല് ബൈറ്റുകളുടെ (അല്ലെങ്കിൽ 32 ബിറ്റുകൾ) ഒരു ഗ്രൂപ്പാണ് ഐപി അഡ്രസ്സ് , അവയിൽ ഓരോന്നും 0 മുതൽ 255 വരെയുള്ള സംഖ്യകളാകാം.

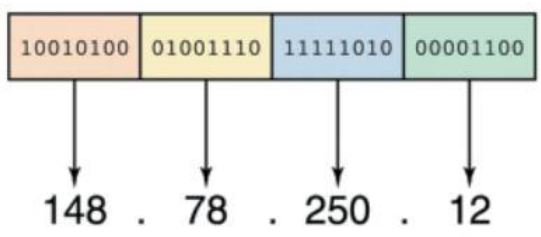


Fig 11.15 IP address

ചോദ്യങ്ങൾ

- a) ടെലിഫോൺ ലൈനുകൾ വഴി മറ്റ് കമ്പ്യൂട്ടറുകളുമായി ബന്ധപ്പെടാനും ആശയവിനിമയം നടത്താനും നിങ്ങളെ അനുവദിക്കുന്ന ഒരു കമ്പ്യൂട്ടർ പെരിഫറൽ ആണ് -----.
- b) ----- ടോപ്പോളജിയിൽ, എല്ലാ ഉപകരണങ്ങളും ഒരു കേന്ദ്ര ഹബിലേക്ക് / സ്വിച്ചിലേക്ക് ബന്ധിപ്പിച്ചിരിക്കുന്നു.
- c) കമ്പ്യൂട്ടർ നെറ്റ്‌വർക്കുകളുടെ ഏതെങ്കിലും രണ്ട് ഗുണങ്ങൾ വിശദീകരിക്കുക.
- d) സ്റ്റാൻഡ്-എലോൺ കമ്പ്യൂട്ടറുകൾക്ക് പകരം നെറ്റ്‌വർക്കുചെയ്ത കമ്പ്യൂട്ടറുകൾ ഉപയോഗിക്കുന്നതിൽ ധാരാളം ഗുണങ്ങളുണ്ട്. അവയിൽ ഏതെങ്കിലും നാല് ഗുണങ്ങൾ എഴുതുക.
- e) സ്വിച്ച് എന്ന നെറ്റ്‌വർക്ക് ഉപകരണത്തിന്റെ ധർമ്മം വിവരിക്കുക.
- f) ടോപ്പോളജി എന്ന പദം നിർവചിക്കുക. നിങ്ങളുടെ കമ്പ്യൂട്ടർ lab നെ നെറ്റ്‌വർക്ക് ചെയ്യാൻ നിങ്ങളുടെ പ്രിൻസിപ്പൽ തീരുമാനിച്ചു. ഏത് ടോപ്പോളജി നിങ്ങൾ നിർദ്ദേശിക്കും? നിങ്ങളുടെ ഉത്തരം ന്യായീകരിക്കുക.
- g) റിംഗ് ടോപ്പോളജി, മെഷ് ടോപ്പോളജി എന്നിവ താരതമ്യം ചെയ്യുക.
- h) കമ്പ്യൂട്ടർ നെറ്റ്‌വർക്കുമായി ബന്ധപ്പെട്ട ഇനിപ്പറയുന്ന പദങ്ങൾ നിർവചിക്കുക
- a) Bandwidth b) Noise c) Node

ഇന്റർനെറ്റും മൊബൈൽ കമ്പ്യൂട്ടിംഗും

ഇന്റർനെറ്റിലെ സേവനങ്ങൾ

- വേൾഡ് വൈഡ് വെബ് (WWW)
- സേർച്ച് എൻജിനുകൾ
- ഇ-മെയിൽ
- സാമൂഹിക മാധ്യമങ്ങൾ (Social Media)

വേൾഡ് വൈഡ് വെബ് (WWW)

ടെക്സ്റ്റ്, ചിത്രങ്ങൾ, വീഡിയോകൾ എല്ലാം ഉൾക്കൊള്ളുന്ന നിരവധി ഡോക്യുമെന്റുകൾ ശേഖരിച്ച് വെച്ചിട്ടുള്ള സെർവറുകളുടെ ഒരു കൂട്ടമാണ് വേൾഡ് വൈഡ് വെബ്. ഇതിൽ ഓരോ ഡോക്യുമെന്റിനെയും തിരിച്ചറിയുന്നതിനായി ഒരു **URL (Uniform Resource Locator)** ഉണ്ടായിരിക്കും. ആളുകൾക്ക് ഈ ഡോക്യുമെന്റുകളെ ബ്രൗസർ സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് ദൃശ്യമാക്കാം. **Google Chrome, Mozilla Firefox, Opera, Internet Explorer, Safari** എന്നിവ ഇത്തരത്തിൽ ഉപയോഗിക്കുന്ന ബ്രൗസറുകളാണ്. ഇങ്ങിനെ ബ്രൗസറുകൾ ഉപയോഗിച്ച് ഡോക്യുമെന്റുകൾ ദൃശ്യമാക്കുന്നതാണ് വെബ് ബ്രൗസിംഗ്.

സേർച്ച് എൻജിനുകൾ

WWW ഇൽ ലഭ്യമായ വിവരങ്ങൾ കണ്ടു പിടിക്കുന്നതിനുള്ള ഒരു പ്രത്യേക പ്രോഗ്രാം ആണ് സേർച്ച് എൻജിൻ. ഒരു പ്രത്യേക വാക്ക് ഉൾക്കൊള്ളുന്ന നിരവധി പേജുകൾ പ്രദർശിപ്പിക്കാൻ ഇതിന് കഴിയും. പ്രചാരത്തിലുള്ള സേർച്ച് എൻജിനുകൾ **Google, Bing, Yahoo Search, Ask** എന്നിവയാണ്.

ഇ-മെയിൽ

ഇന്റർനെറ്റും കമ്പ്യൂട്ടറും ഉരുപയോഗിച്ച് ഡിജിറ്റൽ മെസ്സേജുകൾ കൈമാറ്റം ചെയ്യുന്ന രീതിയാണ് ഇ-മെയിൽ. മെസ്സേജുകൾക്ക് പുറമെ ഡോക്യുമെന്റ്, ചിത്രങ്ങൾ, വീഡിയോകൾ എന്നിവയും ചേർത്ത് (**attach**) അയക്കാം. **gmail.com, hotmail.com** എന്നിവ വളരെ പ്രചാരത്തിലുള്ള ഇ-മെയിൽ സേവന ദാതാക്കളാണ്. ഇ-മെയിൽ അയക്കുന്ന ആൾക്കും ലഭിക്കേണ്ട ആൾക്കും ഇ-മെയിൽ വിലാസം (**id**) ഉണ്ടായിരിക്കണം. ഇ-മെയിൽ വിലാസത്തിന്റെ രൂപം താഴെ.

ഉപയോക്താവിന്റെ പേര് @ ഡൊമൈൻ പേര്. ഉദാ: **scertkerala@gmail.com**

ഒരു ഇ-മെയിലിൽ **To** (ലഭിക്കേണ്ട വിലാസം), **cc** (കാർബൺ കോപ്പി), ബുക്ക് (ബ്ലൈൻഡ് കാർബൺ കോപ്പി), വിഷയം, ഉള്ളടക്കം എന്നിവക്ക് പുറമെ അറ്റാച്ച് ചെയ്ത ഫയലുകളും ഉണ്ടാകാം.

നേട്ടങ്ങൾ

- വേഗത
- ഉപയോഗയ്ക്കുന്നതിനു എളുപ്പമാണ്
- അറ്റാച്ച് ചെയ്യുവാനുള്ള സൗകര്യം
- ഏത് സമയവും എവിടെയും ലഭ്യമാണ്.

പരിമിതികൾ

1. ഇ-മെയിൽ വൈറസ് വാഹകരാകാം
2. ജങ്ക് മെയിൽസ്:- ഇത് അർത്ഥമാക്കുന്നത് അനാവശ്യ മെയിലുകൾ എന്നാണ്

സാമൂഹിക മാധ്യമങ്ങൾ (Social Media)

വെബ് അധിഷ്ഠിത സങ്കേതവും മൊബൈലും ഉപയോഗിച്ച് ഉള്ളടക്കം (**content**) നിർമ്മിക്കുകയും, കൈമാറ്റം ചെയ്യുകയും, ചർച്ച ചെയ്യുകയും എല്ലാം സാധ്യമാകുന്ന ഒന്നാണിത്. ഇത് ഒരു അയഥാർത്ഥ ലോകമാണ്.പല തരത്തിലുണ്ട്.

1. **ഇന്റർനെറ്റ് വേദികൾ (Internet Forums):-** ഇവിടെ വ്യക്തികൾക്ക് ഒരു ത്രേഡ് ആരംഭിക്കുകയോ പ്രതികരിക്കുകയോ ചെയ്യാം. ത്രേഡ് എന്നാൽ ചർച്ചാ വിഷയമാണ്. **Ubuntu Forum** ഒരു ഇന്റർനെറ്റ് വേദിയാണ്.
2. **സാമൂഹിക ബ്ലോഗുകൾ (Social blogs) :-** വിപരീത കാലക്രമത്തിൽ (ഏറ്റവും പുതിയത് ആദ്യം) എഴുത്തുകളോ കുറിപ്പുകളോ ഉൾപ്പെടുന്ന വെബ് സൈറ്റ് ആണ് ബ്ലോഗ് (വെബ് ലോഗ്). ദിനകുറിപ്പുകളായും പരസ്യത്തിനും മറ്റും ഇത് ഉപയോഗിക്കുന്നു. ഉദാ: **Blogger.com, Wordpress.com.**
3. **മൈക്രോ ബ്ലോഗുകൾ:-** ആളുകൾക്ക് അവരുടെ ചുറ്റുപാടിൽ നിന്നും ലഭ്യമായ ഹ്രസ്വ വിവരങ്ങൾ, ചിത്രങ്ങൾ, വീഡിയോ ലിങ്കുകൾ എന്നിവ ഷെയർ ചെയ്യാം. ഉദാ: **Twitter.com**
4. **വികിസംഗ്രഹം:-** ഇതിൽ ഏതൊരാൾക്കും വിവരങ്ങൾ കൂട്ടിച്ചേർക്കാനും മാറ്റം വരുത്തുവാനും സാധിക്കും. ഇത് നിരീക്ഷിക്കുവാൻ ഒരു വിദഗ്ധ സംഘം ഉണ്ടായിരിക്കും. അതിനാൽ, പരസ്യങ്ങളും തെറ്റായ വിവരങ്ങളും ഉടൻ നീക്കം ചെയ്യപ്പെടും. ഉദാ: **wikipedia.org**
5. **സാമൂഹ്യശൃംഖല (Social networks):-** വ്യക്തിഗത വെബ് പേജ് നിർമ്മിച്ച് ടെക്സ്റ്റ്, ചിത്രം, വീഡിയോ എന്നിവ പങ്കുവെയ്ക്കാൻ കഴിയും. ഉദാ: **facebook.com, linkedin.com.**
6. **ഉള്ളടക്ക സമൂഹങ്ങൾ (Content communities):-** ഇതിൽ ചിത്രങ്ങൾ,വീഡിയോകൾ തുടങ്ങിയ ഉള്ളടക്കങ്ങൾ ക്രമപ്പെടുത്തുകയും പങ്കിടുകയും ചെയ്യുന്നു. ഉദാ: വീഡിയോ പങ്കു വെക്കുന്ന **Youtube.com** ചിത്രങ്ങൾ പങ്ക് വെക്കുന്ന **flickr.com**

സാമൂഹിക മാധ്യമങ്ങളുടെ നേട്ടങ്ങൾ

- ആളുകളെ ഒരുമിച്ച് ചേർക്കുന്നു
- പരിപാടികളുടെ ആസൂത്രണവും ക്രമീകരണവും
- വാണിജ്യ പ്രചാരണം

സാമൂഹിക മാധ്യമങ്ങൾ ഉപയോഗിക്കുന്നതിലെ പരിമിതികൾ

- സ്വകാര്യതയിലേക്കുള്ള കടന്ന് കയറ്റം
- ആസക്തി (അധിക്ഷേപം)
- അപവാദം പരത്തുന്നു (rumours)

സൈബർ സുരക്ഷ:- കമ്പ്യൂട്ടർ ശൃംഖലയെ ബാധിക്കാവുന്ന ഭീഷണികൾ

- **കമ്പ്യൂട്ടർ വൈറസ്:-** മറ്റൊരു പ്രോഗ്രാമുമായി കൂടിച്ചേർന്ന് വ്യാപിക്കുന്ന ഒന്നാണ് വൈറസ്. കമ്പ്യൂട്ടറുകളുടെ പ്രവർത്തനത്തെ ഇവ സാരമായി ബാധിക്കുന്നു. ഹാർഡ് ഡിസ്കിലെ വിവരങ്ങൾ നശിപ്പിക്കുന്നു. ഇതു കൂടുതലും വ്യാപിക്കുന്നത് പെൻ ഡ്രൈവ്, പോർട്ടബിൾ ഹാർഡ് ഡ്രൈവ്, ഇ-മെയിൽ അറ്റാച്ച്മെന്റ് തുടങ്ങിയവയിലൂടെയാണ്.
- **ട്രോജൻ ഹോഴ്സ്:-** ഇവ നമുക്ക് ഉപകാരപ്രദമെന്ന് തോന്നിപ്പിക്കുന്നു. എന്നാൽ, പ്രവർത്തിപ്പിച്ചാൽ ഫയലുകൾ നഷ്ടപ്പെടുത്തി കമ്പ്യൂട്ടറിനെ തകരാറിലാക്കുന്നു. ചില ട്രോജനുകൾ കമ്പ്യൂട്ടറിലെ രഹസ്യ വിവരങ്ങൾ ചോർത്തിയെടുക്കുവാൻ ഹാക്കർമാരെ സഹായിക്കുകയും ചെയ്യും.
- **ഹാക്കിങ്ങ്:-** ഒരു കമ്പ്യൂട്ടറിനെ സാധാരണ പ്രവർത്തനത്തിന് വിഘാതം സൃഷ്ടിക്കുന്ന പ്രവർത്തനമാണിത്. കമ്പ്യൂട്ടർ സുരക്ഷാ വിദഗ്ധർ പോരായ്മകൾ കണ്ടെത്തുവാൻ ഹാക്ക് ചെയ്യുന്നതിനെ എത്തിക്കൽ ഹാക്കിങ്ങ് എന്ന് പറയും. ഇവരെ വൈറ്റ് ഹാറ്റ് ഹാക്കേഴ്സ് എന്ന് വിളിക്കും. കമ്പ്യൂട്ടർ കുറ്റവാളികൾ വിവരങ്ങൾ മോഷ്ടിക്കാൻ ഹാക്ക് ചെയ്യാറുണ്ട്. ഇവരെ ബ്ലാക്ക് ഹാറ്റ് ഹാക്കേഴ്സ് എന്ന് വിളിക്കും.
- **ഫിഷിങ്ങ്:-** ബാങ്കുകളുടെയും മറ്റ് സ്ഥാപനങ്ങളുടെയും വ്യാജ വെബ് സൈറ്റ് നിർമ്മിച്ച് വിവരങ്ങൾ തട്ടിയെടുക്കുന്ന രീതിയാണ് ഫിഷിങ്ങ്. ഇത്തരം വെബ് പേജുകൾ നിർമ്മിക്കുന്നതിനെ സ്പൂഫിംഗ് എന്ന് അറിയപ്പെടും. ഇ-മെയിലുകളിലൂടെയാണ് ആളുകളെ ഇത്തരം വെബ് പേജുകളിലേക്ക് ആകർഷിക്കുന്നത്.

ചോദ്യങ്ങൾ

1. ഒറ്റയായത് തിരഞ്ഞെടുക്കുക.

a. Yahoo	b. Gmail	c. Google	d. Bing
----------	----------	-----------	---------
2. നിങ്ങൾക്ക് പരിചയമുള്ള ഏതെങ്കിലും രണ്ട് ബ്രൗസറുകളുടെ പേരെഴുതുക.
3. സാമൂഹ്യ മാധ്യമങ്ങളുടെ ഉപയോഗത്തിലെ പരിമിതികൾ എന്തെല്ലാം?
4. വൈറസ്, ട്രോജൻ ഹോഴ്സ് എന്നിവ വിശദമാക്കുക.
5. താഴെ തന്നിട്ടുള്ളവയെ വിവിധ സാമൂഹിക മാധ്യമങ്ങൾ ആയി തരം തിരിക്കുക.

(a) Youtube	(b) Wikipedia	(c) Twitter
(d) Facebook	(e) Ubuntu forum	(f) blogger
6. ഇന്റർനെറ്റിലെ വിവിധ സേവനങ്ങളുടെ പേരെഴുതുക. ഏതെങ്കിലും രണ്ടെണ്ണം വിവരിക്കുക.
7. ഇ-മെയിൽ എന്നാൽ എന്ത്? ഇ-മെയിൽ വിലാസത്തിന്റെ ഘടകങ്ങൾ വിവരിക്കുക.
8. വെബ് ബ്രൗസിംഗ് എന്നാൽ എന്ത്? വെബ് ബ്രൗസിങ്ങിന് ഉപയോഗിക്കുന്ന സോഫ്റ്റ് വെയറുകളുടെ പേരെഴുതുക.
9. ഇ-മെയിലുകൾ ഉപയോഗിക്കുന്നതിന്റെ നേട്ടങ്ങൾ എന്തെല്ലാം?
10. ഹാക്കിങ്ങ് എന്നാൽ എന്ത്? വൈറ്റ് ഹാറ്റ്, ബ്ലാക്ക് ഹാറ്റ് ഹാക്കേഴ്സ് എന്തെന്ന് വിവരിക്കുക.
11. ഒരു പ്രത്യേക വാക്ക് ഉപയോഗിച്ച് വെബ് പേജുകൾ കണ്ടെത്താൻ ഉപയോഗിക്കാവുന്ന ഇന്റർനെറ്റ് സേവനം ഏത്? അത്തരത്തിലുള്ള മൂന്നെണ്ണത്തിന്റെ പേരെഴുതുക.