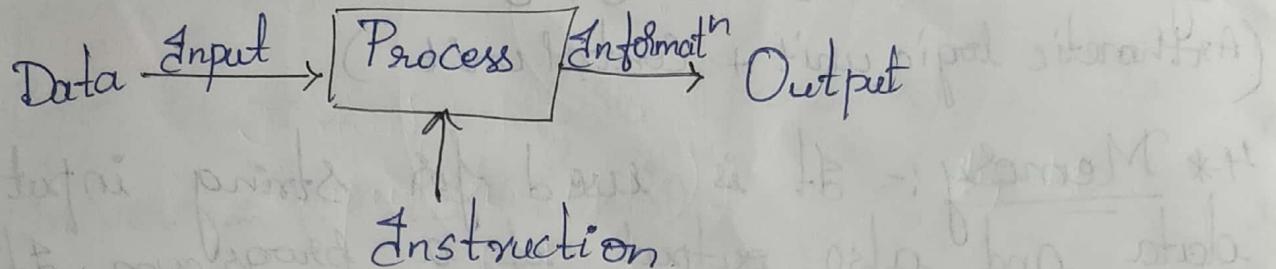
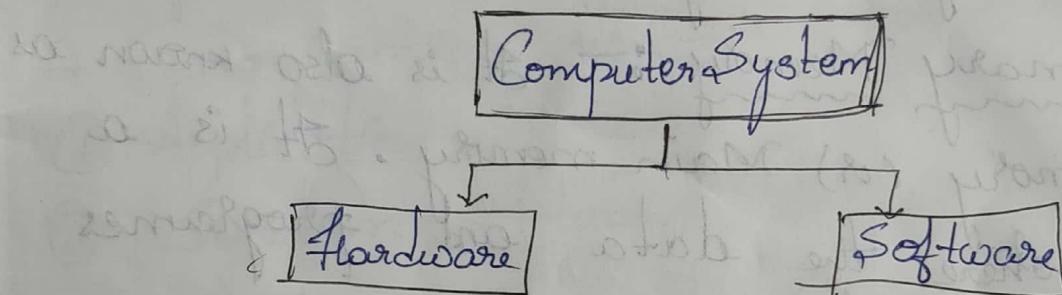


\* INSTRUCTION TO COMPUTERS \*

→ A computer is an electronic device that takes data as input from the user, processes data and provides useful information known as output.



⇒ A Computer is made up of two components i.e., ① Hardware ② Software



① Hardware :- It is a physical part of computer system. It includes the following components.  
1, Input devices      3, Processor / C.P.U  
2, Output devices      4, Memory

1\* Input :- Input devices are used for accepting data on which the operations be perform.

Ex:- Keyboard, Mouse, Scanner etc.

2\* Output :- Output devices are used for providing results i.e., operators after performing the operations.

Ex:- Printers, Speakers, Monitor etc.

Processor:- It is also called as C.P.U  
(Central processing Unit). C.P.U is responsible for executing programs (instruction).

e.g. ALU + CU = C.P.U  
(Arithmetic logic unit + Control unit)

4\* Memory:- It is used for storing input data and also output of a program. It divides into 2 types. ① Primary Memory

② Secondary Memory.

→ Primary Memory :- It is also known as Real memory (or) Main memory. It is a place where the data and programs are stored temporarily during processing. It includes RAM & ROM.

③ RAM - (RANDOM ACCESS MEMORY). It is Read / write memory. It is a volatile memory (temporary) that temporarily stores data and programs as long as they are in use.

④ ROM - (Read Only Memory). It is a non volatile memory (permanent) that stores data and programs permanently even when the computer is turned off or switched off.

→ Secondary Memory :- It is called auxiliary storage device. It is used for storing both input & output. It is a permanent memory.

Eg:- Hard disc, USB (pendrive), CD, Universal serial Buses

\* Software :- It is a group of programs for performing a task. A programs divided into two types.

① Application software.

② System software.

• Application Software :- It is used for perform a task according to requirement

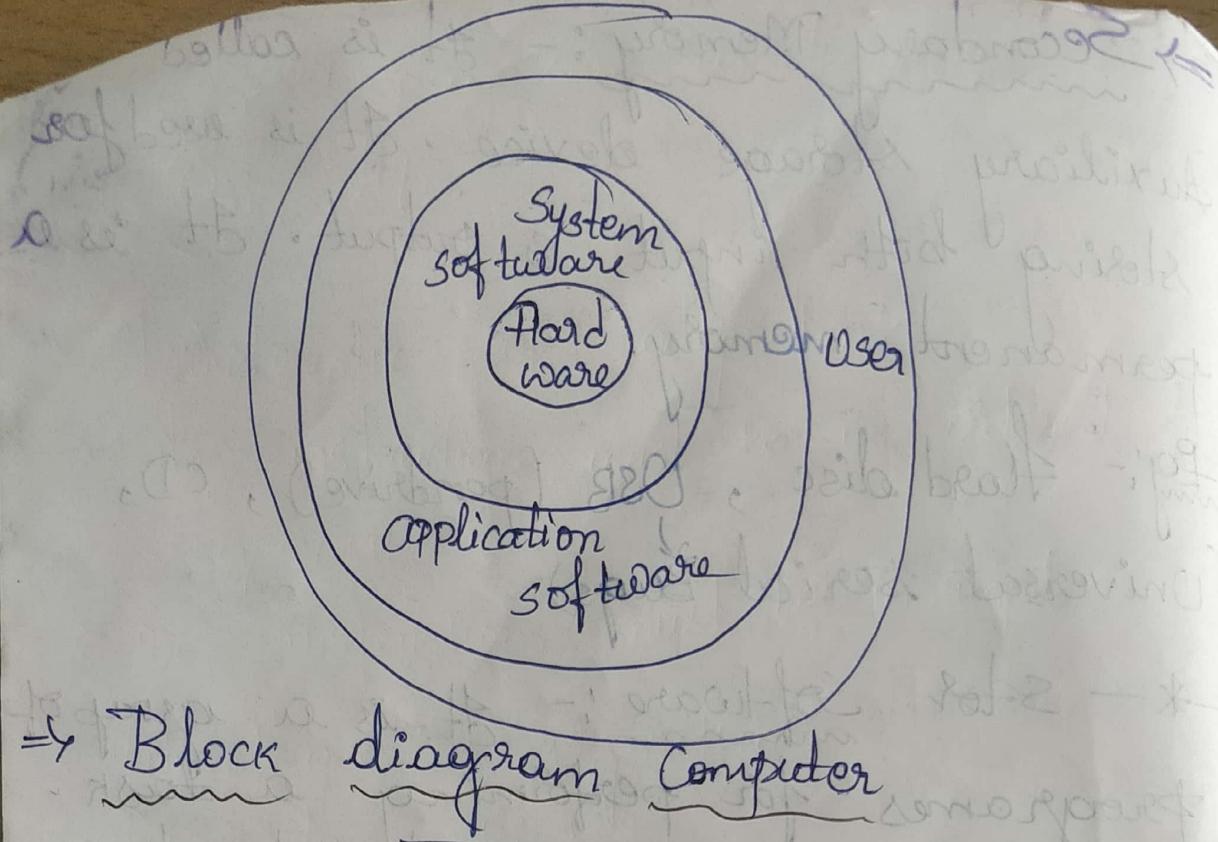
Eg:- Browsers, Ms-office, chrome, windows.

• System Software :- It is responsible for running application software.

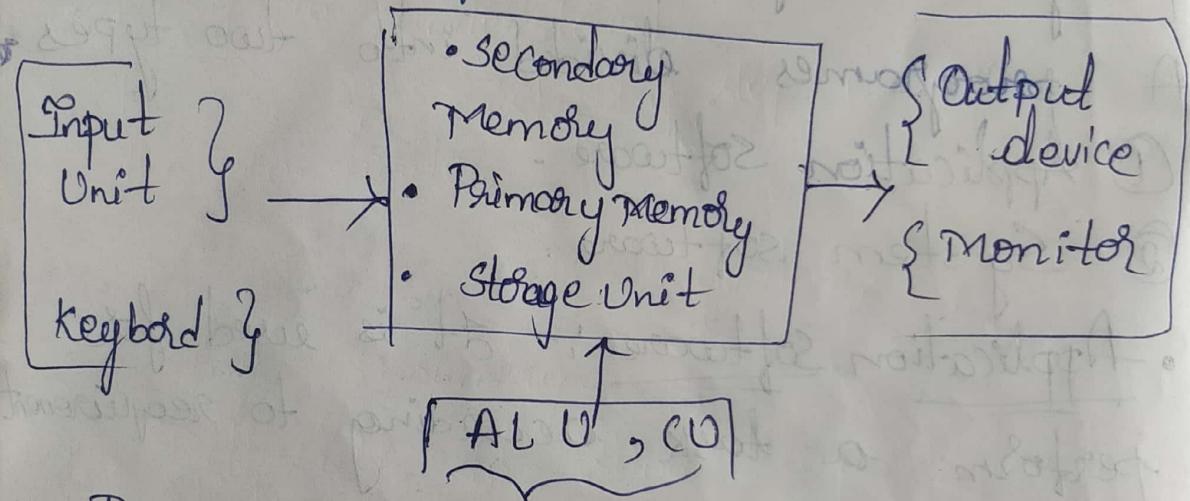
Eg:- OS, Networking system.

\* OS :- It is an interface b/w the user and computer hardware.

Eg:- MS-DOS (Disc - operating system)  
Windows, linear.



$\Rightarrow$  Block diagram Computer



$\Rightarrow$  Program in language (or) Computer language.

- ① Machine language (or) Binary language  
Low-level language (1940)
- ② Assembly (Symbolic)  $\rightarrow$  OP code  
 $\begin{array}{l} \text{Mnemonic} \\ \text{Assembly} \\ \text{Compiler} \\ \text{Interpreter} \end{array}$
- ③ High - level language. (1960)

12/8/19} Problem Solving:-

→ Algorithm , → flowchart , → Pseudo code

→ Algorithm:-

\* Problem solving is the process of solving a problem in computers by following sequence of steps. It includes the following methods  
① Algorithm ② flow chart ③ Pseudo code.

→ Algorithm:-

An algorithm is a very powerful technique i.e., used to get sol<sup>n</sup> for a given problem. It is a step by step procedure (process or method) for solving a problem  
Or An algorithm is sequence of steps i.e., within English for that specifies the task that are performed while solving a problem.

1. Write an algorithm to add two numbers.

Step-① :- Start

Step-② :- Accept the first integer as input from the user (or) read the 1<sup>st</sup> no. [Numb 1]

Step-③ :- Accept the second integer as input from the user (or) read the 2<sup>nd</sup> no. [Numb 2]

Step-④ :- Calculate the sum of the two no..

$$(\text{Sum} = \text{num } 1 + \text{num } 2)$$

Step-⑤ :- Display sum as the result.  
Step-⑥ :- Stop.

⇒ Characteristics of algorithm:-

→ The instructions must be in ordered form (sequence)

→ The instructions must be in simple.

⇒ Advantages of Algorithm:-

- Easy to understand.
- It is a step by step process.
- Easy to write.

⇒ Disadvantages of Algorithm:-

- \* Time consuming.
- \* Difficult to show loops & selection statement.

Q. Write an algorithm to find area and perimeter of a rectangle:-

Step-①:- Start

Step-②:- Accept the first integer as input  
i.e., (l) [num l]

Step-③:- Read the (b) value

Step-④:- Calculate the area  
(area = l × b)

Step-⑤:- calculate the perimeter

$$\text{Perimeter} = 2(l+b)$$

Step - ⑥ :- Display area and perimeter as result.

Step - ⑦ :- Stop.

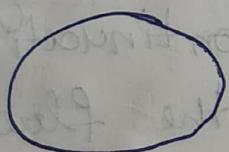
16 | 8 | 9

= \* Flow chart :- \* =

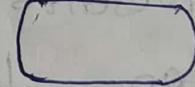
If it is a pictorial or graphical representation of an algorithm which describes the sequence and flow of steps with in the process.

\* Symbols used in a flow chart :-

① Start and stop :- It is represented by an oval or rounded rectangle and represents the starting & ending of a flow chart.

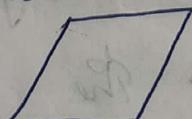


(O)



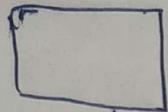
Rounded  
Rectangle

② Input and output :- It is represented by a parallelogram & represents the input (given by the user to process) and output (given to the users). It is used for input and output operators.



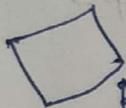
(Parallelogram)

Action (or) Process:- It is represented by a rectangle. It represent the action, logical calculation within the process.



rectangle

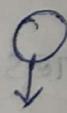
④ Decision (or) Condition:- It is represented by a rhombus. It represents the condition or decision making steps in a flow chart.



Rhombus

⑤ Arrow:- It is represented by a directed line → ↘ ↓ . It represents the flow of process and sequence of steps.

⑥ Connector:- It is represented by a circle, it represents the continuation of the flow of steps. When the flow chart continue to the next page.



\*Design Groups:- It must begin with the start & stop symbols. The standard process flow should be either from top to bottom and left to right.

\*Advantage of flow chart:-

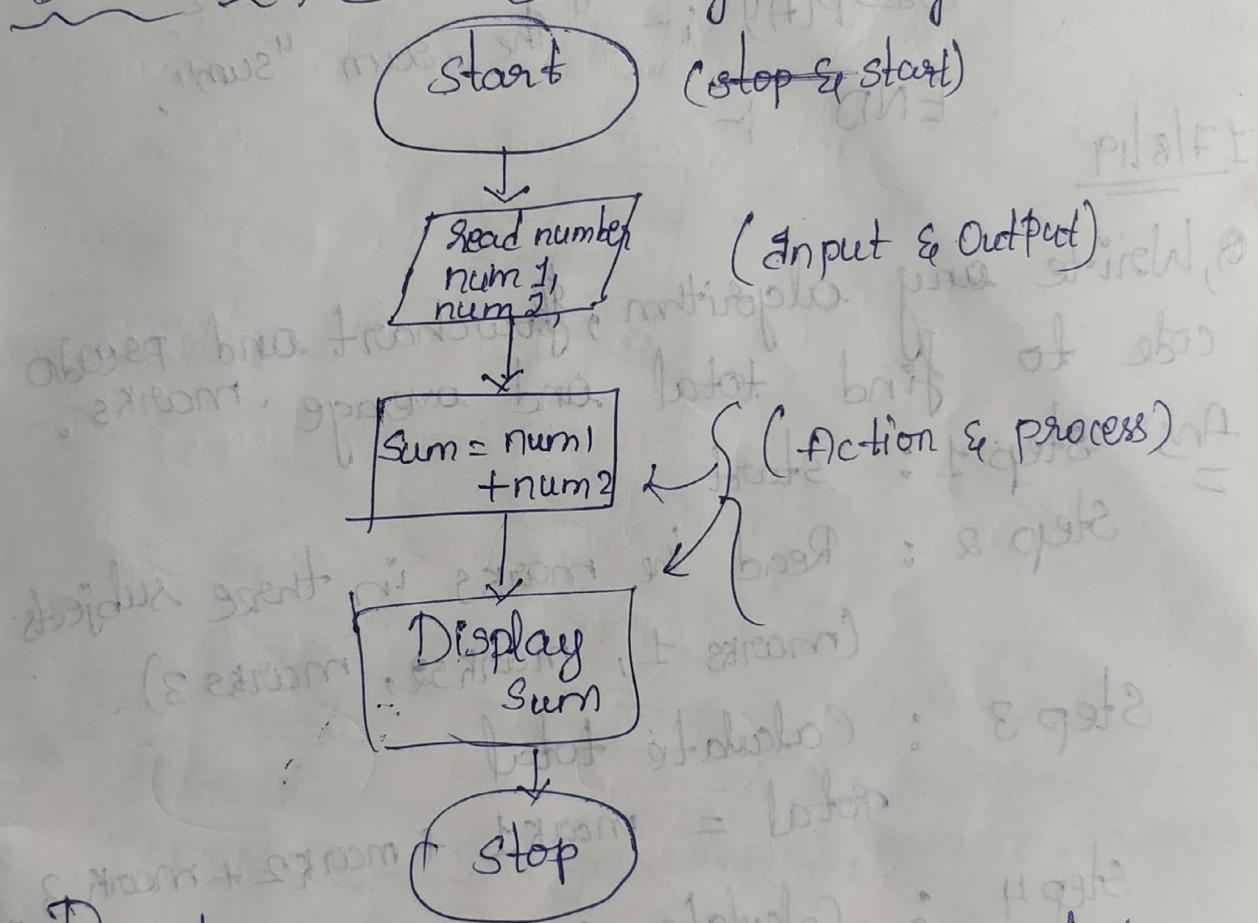
1, Flow chart works well for small programme design.

2, It helps in avoiding semantic errors.

\* Disadvantage of flow chart:

1, It difficult to draw for complex programme.

Draw a flow chart by adding two numbers



• Pseudo code: It is a technique used to develop the source code for the programme. It is written with a combination of generic syntax and normal English language. It helps the programmer understand the basic logic of the programme.

\* Write a Pseudo to adding two numbers.

Ans: BEGIN  
DEFINE :- Integer num1, num2, sum.  
DISPLAY :- "Enter the two numbers".  
READ :- num1, num2.  
SET (or) COMPUTE :- sum = num1 + num2  
DISPLAY :- "The sum "sum".  
END

17/8/19

Q. Write any algorithm, flowchart and pseudo code to find total and average marks.

Ans Step 1 : start  
Step 2 : Read the marks in three subjects  
(marks 1, marks 2, marks 3).

Step 3 : Calculate total

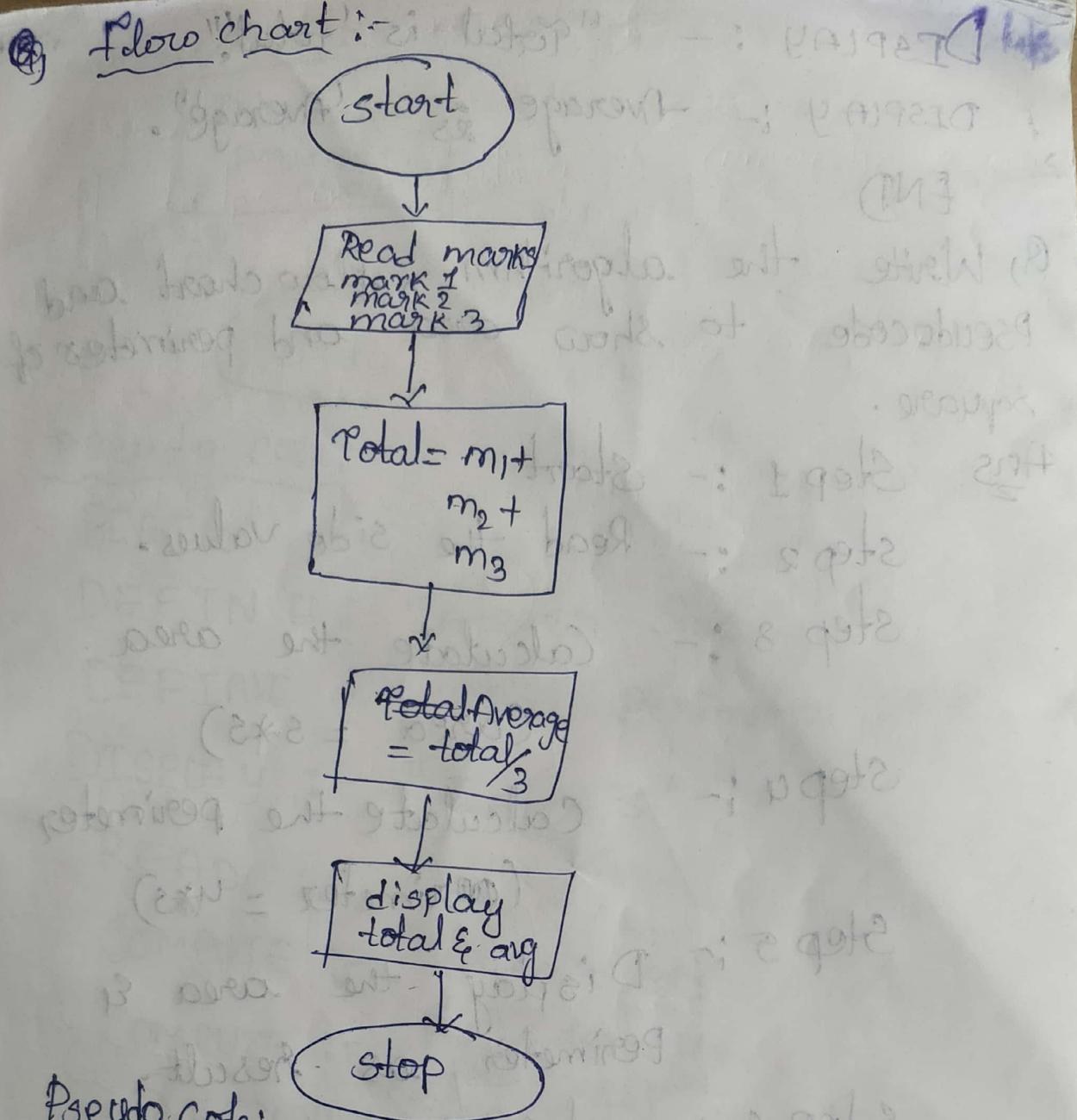
$$\text{total} = \text{mark1} + \text{mark2} + \text{mark3}$$

Step 4 : Calculate average.

$$\text{Average} = \frac{\text{total}}{3}$$

Step 5 : Display total and average  
as the result.

Step 6 : Stop.



Pseudo code:-

BEGIN

DEFINE :- Integer marks1, marks2, marks3

DISPLAY :- "Enter three numbers".

READ :- Real total & average.

DISPLAY :- "Enter the marks of three".

READ :- m<sub>1</sub>, m<sub>2</sub>, m<sub>3</sub>, total, average

SET OR COMPUTE :- Total = m<sub>1</sub> + m<sub>2</sub> + m<sub>3</sub>

COMPUTE :- Average = total / 3

get-marks  
exp =

DISPLAY :- "Total is 'total'."

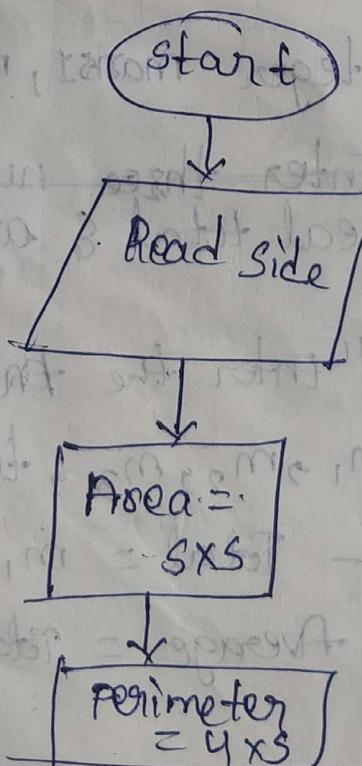
DISPLAY :- Average is 'Average'.

END

Q) Write the algorithm, flow chart and Pseudocode to show area and perimeter of square.

- Ans Step 1 :- Start
- Step 2 :- Read the side values.
- Step 3 :- Calculate the area.  
 $(\text{Area} = s \times s)$
- Step 4 :- Calculate the perimeter  
 $(\text{Perimeter} = 4 \times s)$
- Step 5 :- Display the area & Perimeter as result.
- Step 6 :- Stop.

Flowchart:-



Display  
Perimeter &  
area

Stop

Pseudo code :-

BEGIN .

DEFINE :- Integer side ,

DEFINE :- Real Perimeter , area .

DISPLAY :- " Enter s "

READ :- s

COMPUTE :- Area =  $s^2$

COMPUTE :- Perimeter =  $4s$

DISPLAY :- " The area of square is ~~Area is~~ area "

DISPLAY :- " The perimeter of square is ~~Perimeter is~~ perimeter "

END .

DISPLAY :- " The area of square is "area".

DISPLAY :- " The perimeter of square is "perimeter".

END .

Q. Write an algorithm , flowchart and  
pseudo code to find area of rectangle .

Ans Step 1 :- Start

Step 2 :- Read the l , b values .

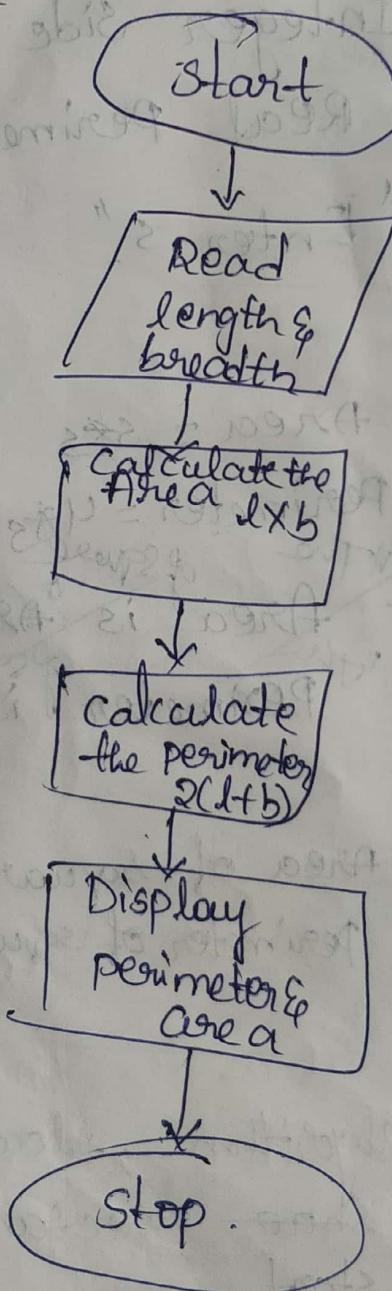
Step 3 :- Calculate the area & perimeter of rectangle is (area =  $l \times b$ )

Step 4 :- Calculate the perimeter of rectangle  
(Perimeter =  $2(l+b)$ )

Step 5 :- Display area and perimeter as result.

Step 6 :- Stop.

Flow chart:-



Pseudo code :-

BEGIN

DEFINE :- Integer l, b

DEFINE :- Real Perimeter and area.

DISPLAY :- "Enter the length & breadth value"

READ :- l & b.

COMPUTE :- Area of rectangle = l\*b

COMPUTE :- Perimeter of rectangle = 2(l+b)

DISPLAY :- "The area of rectangle is  
"area".

DISPLAY :- "The perimeter of rectangle is  
"perimeter".

END.

1Q, Write an algorithm, flowchart & Pseudo code to find area and circumference of circle.

2Q, Write an algorithm, flowchart & Pseudo code to find area & perimeter of triangle.

3Q, Write an algorithm, flowchart & pseudo code to find volume of a sphere

Sphere

4Q, Write an algorithm, flowchart & Pseudo code to find Roots of Quadratic equation.

5. Write the algorithm, flowchart & Pseudo code to find simple interest.

Ans Step 1:- START

Algorithm

Step 2:- Calculate Read the ' $\pi$ ', 'r' values.

Step 3:- Calculate the area of circle

$$\text{area} = \pi r^2$$

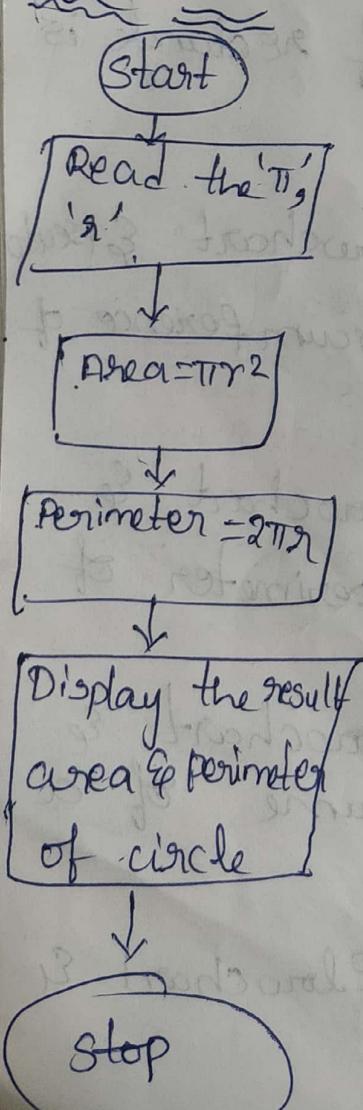
Step 4:- Calculate the perimeter of circle

$$\text{Perimeter} = 2\pi r$$

Step 5 :- Display the result of area & perimeter of circle.

Step 6:- STOP.

Flow chart



Pseudocode

BEGIN

DEFINE:- ' $\pi$ ', 'r'

DEFINE :- Real "area, perimeter",

DISPLAY :- "Enter the  $\pi$  &  $r$ ".

READ :-  $\pi$ ,  $r$

COMPUTE :- Area =  $\pi r^2$

COMPUTE :- Perimeter =  $2\pi r$

DISPLAY :- The area of the circle is "area".

DISPLAY :- The perimeter of the circle is "perimeter".

END.

## Ans Algorithm

Step 1 :- START

Step 2 :- Read the 'h', 'b', 'a', 'c' values.

Step 3 :- Calculate the area of the triangle

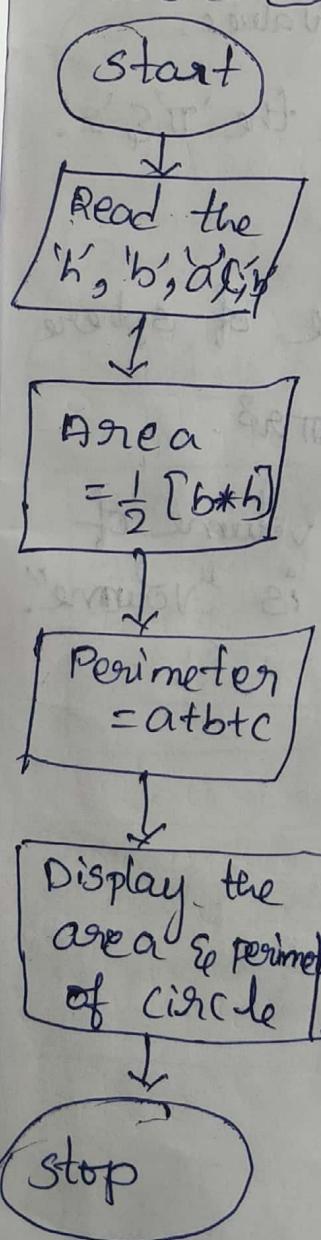
$$\text{Area} = \frac{1}{2} [b*h]$$

Step 4 :- calculate the perimeters of the triangle  
Perimeter = a+b+c

Step 5 :- Display the area & perimeter of the triangle.

Step 6 :- Stop.

## Flowchart:-



## Pseudocode

BEGIN

DEFINE :- Integers 'h', 'b', 'a', 'c'

DEFINE :- Real "area" & "perimeter"

DISPLAY :- "Enter the 'h', 'b', 'a', 'c'"

READ :- 'h', 'b', 'a', 'c'

COMPUTE :- Area =  $\frac{1}{2} (b * h)$

COMPUTE :- Perimeter = a+b+c

DISPLAY :- The area of the triangle is "area".

DISPLAY :- The perimeter of the triangle is "perimeter".

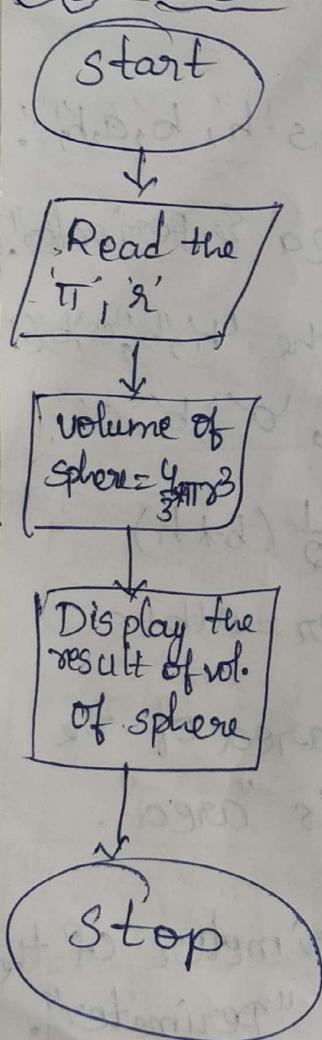
END.

## Algorithm:-

- Step 1 :- START
- Step 2 :- Read the 'π', 'r' values.
- Step 3 :- Calculate the volume of sphere.  

$$\text{Vol. of Sphere} = \frac{4}{3} \pi r^3.$$
- Step 4 :- Display the result of volume of sphere.
- Step 5 :- STOP.

## Flowchart:-



## Pseudo Code:-

```

BEGIN
  DEFINE :- 'π', 'r' values.
  DISPLAY :- Enter the 'π' & 'r'.
  READ :- π, r.
  COMPUTE :- volume of sphere
            =  $\frac{4}{3} \pi r^3$ .
  DISPLAY :- The volume of
            sphere is "volume".
END.
  
```

Ans Quadratic equation :-

$$ax^2 + bx + c = 0$$

$$\text{roots} \Rightarrow x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Algorithm:-

Step 1:- START

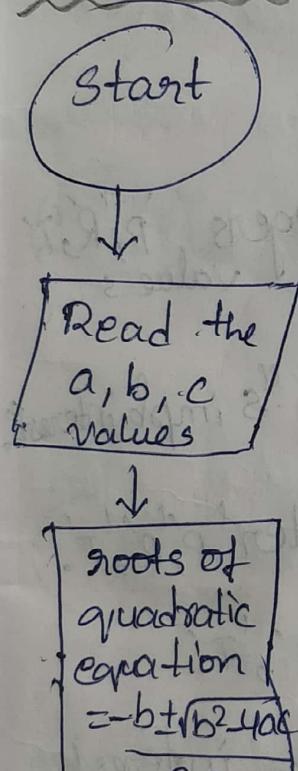
Step 2:- Read the 'a', 'b', 'c', values.

Step 3:- Calculate the quadratic equation of roots. i.e.,  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

Step 4:- Display the results of roots of quadratic equation.

Step 5:- STOP.

Flowchart



Pseudocode:-

BEGIN

DEFINE :- Integers 'a', 'b', 'c'.

DISPLAY :- Enter the 'a', 'b' & 'c'.

READ :- ~~x =~~  $b \pm \sqrt{a, b, c}$

COMPUTE :- Roots of quadratic equation =  $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

DISPLAY :- The roots of quadratic equation is "roots of quadratic equation".

END

## Simple Interest:-

### Algorithm

Step 1 :- START

Step 2 :- Read the 'P', 'R', 'T' values.

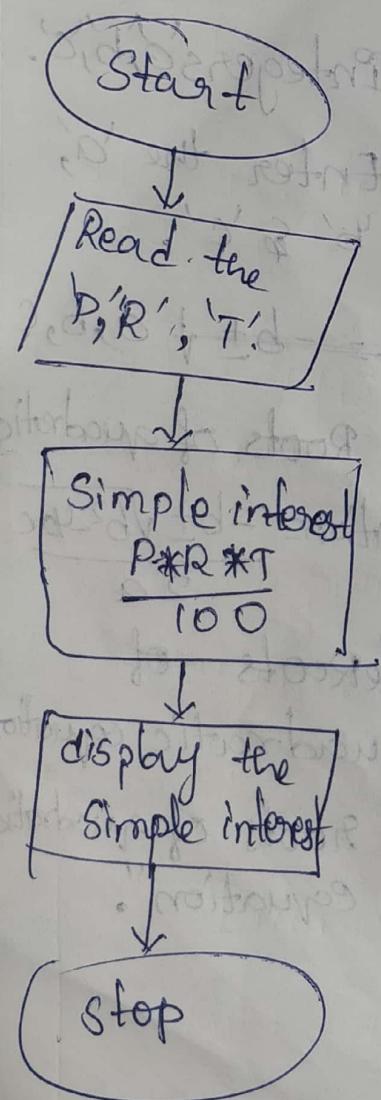
Step 3 :- Calculate the simple interest

$$\text{Simple interest} = \frac{P \times R \times T}{1000}$$

Step 4 :- Display the result of the simple interest

Step 5 :- STOP

### Flowchart



### Pseudo code

BEGIN

DEFINE :- Integers P, R, T  
values

DEFINE :- Real "Simple interest"

DISPLAY :- ENTER 'P, R, T'

READ :- 'P, R, T'

COMPUTE :- Simple interest  
 $= \frac{P \times R \times T}{100}$

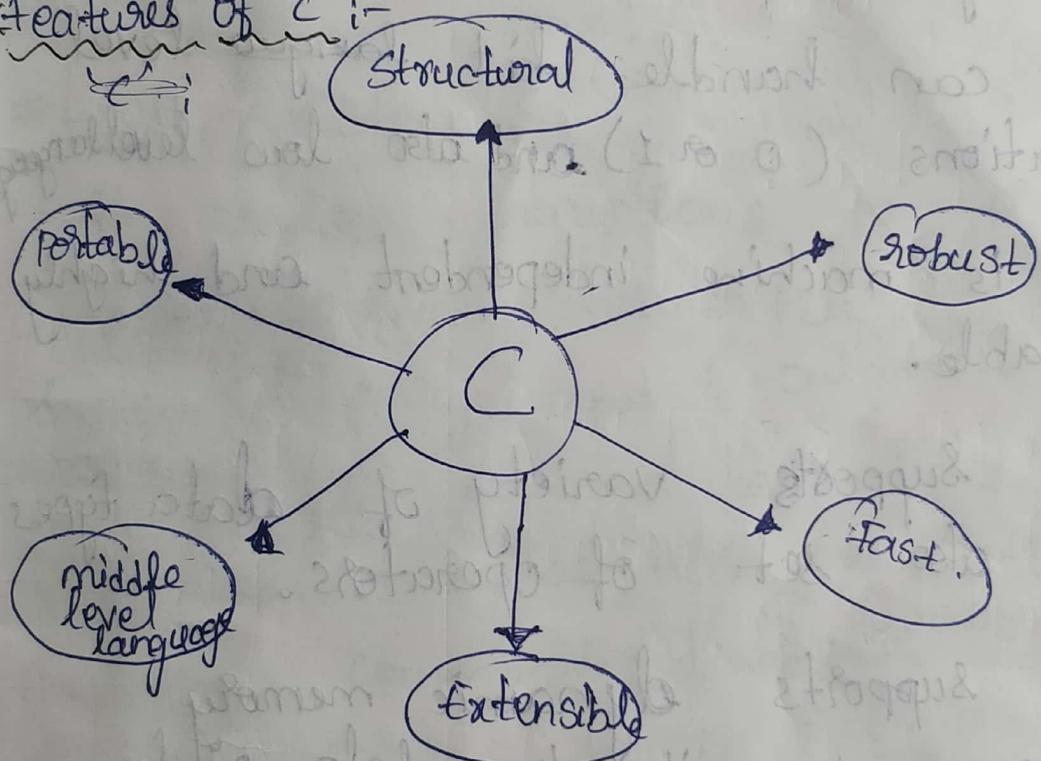
DISPLAY :- the simple interest  
is "Simple interest."

END.

## 19/8/19 - \*Introduction to 'C'\* =

'C' is a structured, powerful & portable programming language. It combines the features of high level language. 'C' is the most widely used programming language to develop system programs and also application programs(OS). [OS - operative systems, Assemblers, Compilers, Linkers and editors (word programme linker)].

### \*Features of 'C':-



### \*History of 'C':-

→ In the year of 1968 "martin Richards" who introduce language is BCPL (Basic combine programming language).

→ In the year of 1970 "ken thompson" introduce a language using many features of BCPL called "B language".

In the year of 1972 "Dennis Ritchie" introduce a language is called "c language" and to develop system & application software.

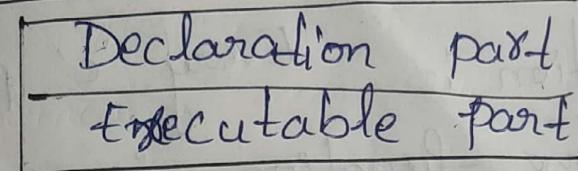
### Characteristics of 'C':

- \* It is highly structured programming language.
  - \* It uses the features of high level language.
  - \* It can handle bit language level operations (0 or 1) and also low level language.
  - \* It is machine independent and highly portable.
  - \* It supports variety of data types and also set of operators.
  - \* It supports dynamic memory management with the help of pointers.
- ### Structure of 'C' programmes:-

Documentation section
Link section (or) header file section
Definition section

## Global declaration section

Main () function name.



## Documentation section

\* It consists of set of comments giving the programme name.

## Link section:-

It provides instructions to provide compiler to link header file to the system library (input & output function).

The headerfile it must be included into the programme by as using #include

Preprocess command / directive.

\* Defination section:- It defines all symbolic constant  $\#define \ PI \ (3.14)$

\* Global declaration section:- this section is declare the variable which are used in more than 1 function is called Global variable.

- Global variable are declared before the main function (or) outside the main function.
- Global / External / Public.

- \* Main () function:- Every 'c' programme must have one function is called main () function.
- \* Every 'c' programme execution starts & begins & ends with main() only.
- \* Main() function contains two parts.
  - Declaration part
  - Executable part
- \* Declaration part :- It declares all the variables which are used in the executable part.
- \* Executable part:- It contains single statement or group of statements. These statement must be control enclosed within the opening ( { ) & closing ( } ) braces.
- \* Every statement it must be ends with semicolon ( ; ) called statement terminator.

Eg:- Write a 'C' programme to print a msg (message) on the screen.

#include <stdio.h>

Void main ()

{

    printf ("Welcome to VJIT");

}

{ A - gcc addc }

P - a.out

Welcome to VJIT

22/08/19

ctrl alt f →

\$ PPS \$ mkd 301 (create, dir, file)

\$ PPS \$ ls (list command)

\$ PPS \$ cd 301 (in - change direct)

\$ PPS \$ 301\$

\$ PPS \$ 301\$ cd .. (out, exist)

\$ PPS \$

\$ PPS \$ 301\$ gedit ex.c

#include <stdio.h>

Void / int main()

{

    printf ("Hello world");

}

printf  
= output

Ctrl S → save

Quit (ctrl q)

{  
  gcc.exe  
  \ a.out  
  hello world}

- C Token  
there are '6' types of 'C token'
- ① Key words → reserved words
  - ② Identifiers → ~~variables~~
  - ③ Constants
  - ④ Strings
  - ⑤ Operators
  - ⑥ Special symbols ([ ], { } ).

C language as 32 words

If, do, while, else, int, float, auto,  
break, continue, switch etc.

#include <stdio.h>

void main ( )

{ (function) . . . }

int a, b, c;

printf ("Enter the values (%d), %d",

scanf ("%d %d", &a, &b);

P. C = a+b;

use - printf ("%sum=%d", sum);

In a 'C' program, the smallest individual unit is called as 'C' tokens.

① Keywords: - It is also called reserved words.  
All keywords have fixed meaning. Meaning  
can not be changed. The keywords can't  
use as variable name (Identifiers)

It should be written in lower case.

It supports 32 keywords.

Ex:- if, do, while, else, int, float, auto,  
break, continue, switch etc.

22/19

② Identifiers :- It refers to the names  
of the variables, function, array &  
structure. These are used different names  
consisting of sequence of letters & digits

Rules for identifiers :-

- 1) It must be in alphabet.
- 2) Special characters are not allowed.

Except (-).

3) <sup>no</sup> Keywords

- 4) Void Space is not allowed.

- 5) It consists of letters (upper-case A-Z  
& low case a-z) (not (-))

Valid :-

total 123 total sum ✓

total 123 ✓

total ✓

Invalid :-

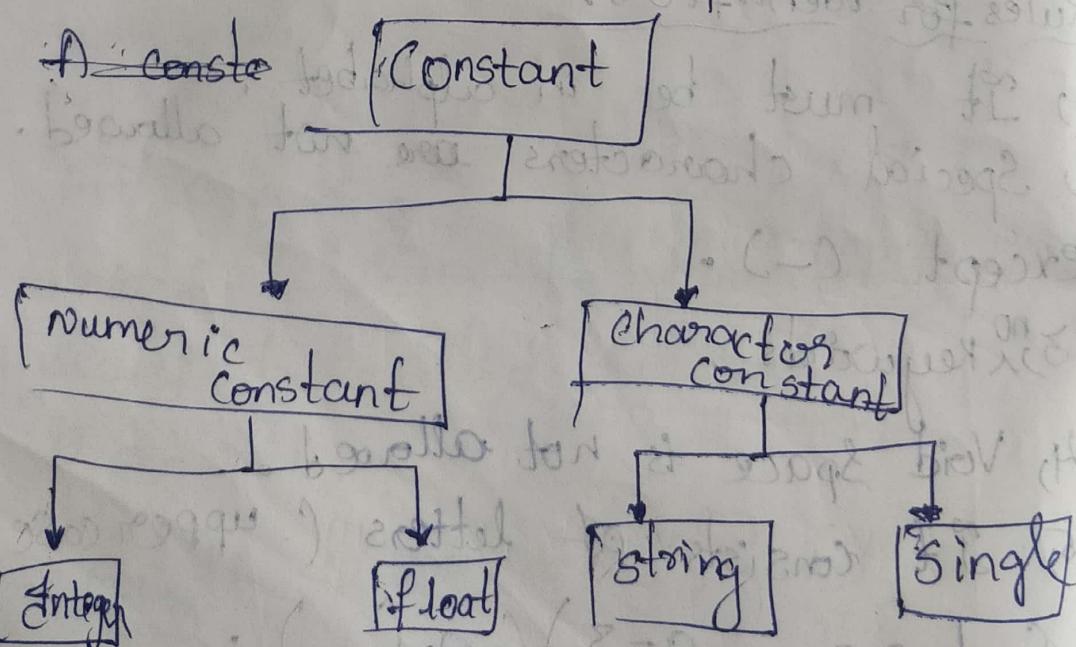
sum ✗

Ranga Reddy ✗

total 123 ✗

⑤ Constant :- It is a fixed value that does not change values. It has 5 types of constant they are. & it is a fixed value.

- |   |                                |  |
|---|--------------------------------|--|
| ① | integer constants              | that could not change execution of program |
| ② | float constants                |  |
| ③ | character constants            |  |
| ④ | string constants               |  |
| ⑤ | Back slash character constants |  |



1) Integer Constant :- It is a no. without decimal point is called integer constant. It may be either positive & negative or zero.

Eg:-  
10  
100  
-800

2) Float constant :- It is a no. with decimal point is called float constal & also called real constant.

Eg:-  
10.5  
100.30  
200.2

3) Character constant :- It is a single character, it must be enclosed with single quotes. Each character has an integer value is known as ASCII (American Standard code Information & Interchange).

Eg:- A - 65    a - 97    O - 48 ; 'x', '5', 'd'  
      Z - 91    Z - 123    @ - 52

4) String Constant :- A group of characters enclosed with double (" ") .

Eg:- "x" "21" "abc"  
Every string ends with null character.

② Back slash characters! :- (Suppose some back slash character constants that are used in printf. If is used in output function (or) printf they are

In  $\rightarrow$  (newline)

It  $\rightarrow$  (tab)

10  $\rightarrow$  (null)

= \* VARIABLE; - \* =

26/8/19

→ A variable is a name of the memory location where the value is stored. It is used to stored data.

A variable is a data name that may be used to store values. It may be (variable value) during the execution of program.

• Declaration of Variable:-

The general form of declaration of variable follows:-

→ Syntax : datatype Variable list ;

datatype  $v_1, v_2, v_3, \dots, v_n;$

{ Here  $v_1, v_2, v_3, \dots, v_n$  are the names of variable separated by comma.

→ Datatype Specifies the type of data the variable it holds.

Eg:-

- int a;
- int a, b, c;
- float f;
- char c;

→ Assigning values to the variable:-  
we can also provide values while declaring the variable.

Syntax : datatype variable name = value;

Eg:-

- int a = 10;
- int a = 10, b = 20, c = 30;
- float f = 10.5;
- char c = 'A';

→ Rules for variables / convention:-

- 1) There should not be keywords.
- 2) It must be in alphabet.
- 3) Special characters are not allowed except (-).
- 4) ~~void~~ Space is not allowed.
- 5) It consists of letters ( uppercase A-Z & lowercase a-z ) or (-).

6) It contains maximum 31 characters.

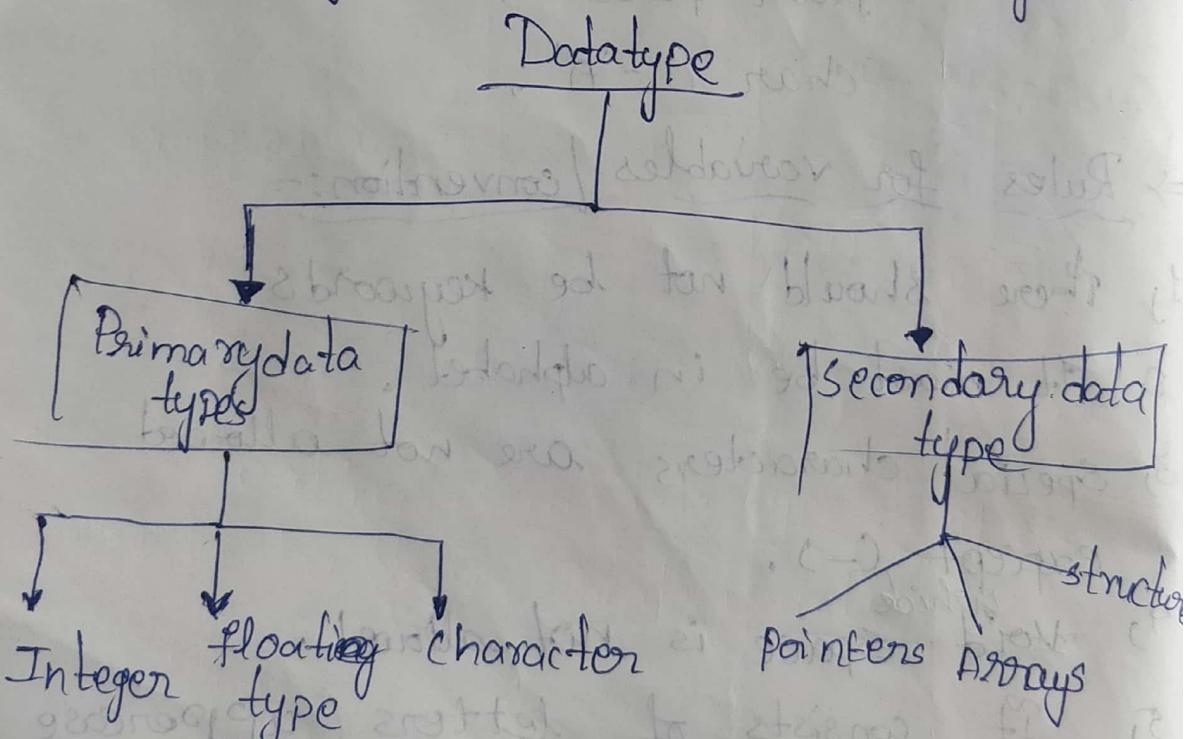
Ex:-	Invalid	Valid
	int 2 $\times$	total 123 ✓
	int abc $\times$	int a,b,c; ✓
	total 123 $\times$	total - sum ✓

### Data types:-

It specifies the type of data that a variable can store or hold.

C-language has some predefined data types to handle various kinds of data that we can use in the programme.

Data-type are divide into two types:-



\* Primary Data types:- It is also called as basic datatypes (or) fundamental data type.

The primary data type are Integer, float point type, character type.

### ① Integer Type :-

a, short int (or) short

b, Int

c, Long int (or) long

(a) short int = short int takes 1 byte of memory space (8 bits)

1 bit for signed bit & 7 bits for storing data.

(b) int = int takes 2 bytes of memory space (16 bits).

1 bit for signed bit & 15 bits for storing data.

(c) long int = it takes 4 bytes of memory space (32 bits).

1 bit for sign bit & 31 bits for storing data.

## Size and Range of integer type.

Type	size (bytes)	Range
1, short int (or) Signed short int	1	-128 to 127 (or) (-2^7 to 2^7 - 1)
2 Unsigned short int	1	0 to 255 (or) (0 to 2^8 - 1)
3, int (or) signed int	2	-2^15 to 2^15 - 1 (or) -32768 to 32767
4, <del>int (or)</del> Unsigned int	2	0 to 2^16 - 1 (or) 0 to 65,535
5) long int (or) Signed long int	4	-2,147,483,648 to <del>65,536</del> . 2,147,483,647
6) Unsigned long int	4	$-2^{31}$ to $2^{31} - 1$ $\{ 0 \text{ to } 2^{32} - 1 \}$ (or) 0 to 4,294,967,295

\* = Floating point data type :- It is used to store real number.

- a) float
- b) double
- c) long double

→ Size and Range of float point type:-

Type	Size (bytes)	Range
• float	4	$3.4E - 38$ to $3.4E + 38$
• Double	8	$1.7E - 308$ to $1.7E + 308$
• Long Double	10	$3.4E - 4932$ to $1.1E + 4932$

\* = character type:-

129/8/19

It is used to store character values.

→ Size and Range of character type:-

Type	Size (bytes)	Range
• Char	1 byte	$-2^7$ to $2^7 - 1$
• Unsigned char	1 byte	0 to $2^8 - 1$

\* = void type:-

It has no value & operations. It is used to specify the type of function.

\* = Secondary data type:-

It is also called as derived data type. It includes pointers, Arrays, structures & functions.

# Size and types Range of Types:-

Type	Size	Range	Storage Content (S)
char	1	-2 <sup>7</sup> to 2 <sup>7</sup> -1	%c
unsigned char	1	0 to 2 <sup>8</sup> -1	%c
short	1	-2 <sup>7</sup> to 2 <sup>7</sup> -1	%d
unsigned short	1	0 to 256	%u
int	2	-2 <sup>15</sup> to 2 <sup>15</sup> -1	%d
unsigned int	2	0 to 2 <sup>16</sup> -1	%u
long	4	-2 <sup>31</sup> to 2 <sup>31</sup> -1	%ld
unsigned long	4	0 to 2 <sup>32</sup> -1	%lu
float	4	3.4E-38 to 3.4E+38	.lf
double	8	1.7E-308 to 1.7E+308	.lf
long double	10	3.4E-4932 to 3.4E+4932	.lf

30/8/19 Using all datatype, write a program

```
#include <stdio.h>
Void main()
{
    int a=10;
    float b=20.5;
    char c='A';
    printf ("%d\n", a); — int
    printf ("%f\n", b); — float
    printf ("%c\n", c); — char
    printf { }
```

---

```
#include <stdio.h>
Void main()
{
    short a=1;
    long b=6;
    double c=5;
    longdouble d=10;
    printf ("%d\n", a);
    printf ("%ld\n", b);
    printf ("%lf\n", c);
    printf ("%lf\n", d);
```

write a c program to find total and average marks of 3 subjects.

Ans #include <stdio.h>  
Void main()  
{  
 Char a; int m<sub>1</sub>, m<sub>2</sub>, m<sub>3</sub>, total, average;  
 Int a=6; Printf ("enter values of  
 Double m<sub>1</sub>, m<sub>2</sub>, m<sub>3</sub>");  
 Scanf ("%d%d%d", &m<sub>1</sub>, &m<sub>2</sub>, &m<sub>3</sub>);  
 total = m<sub>1</sub>+m<sub>2</sub>+m<sub>3</sub>;  
 average =  $\frac{m_1+m_2+m_3}{3}$ ;  
 Printf ("the total is %d", total);  
 Printf ("the average is %.2f", avg);  
}

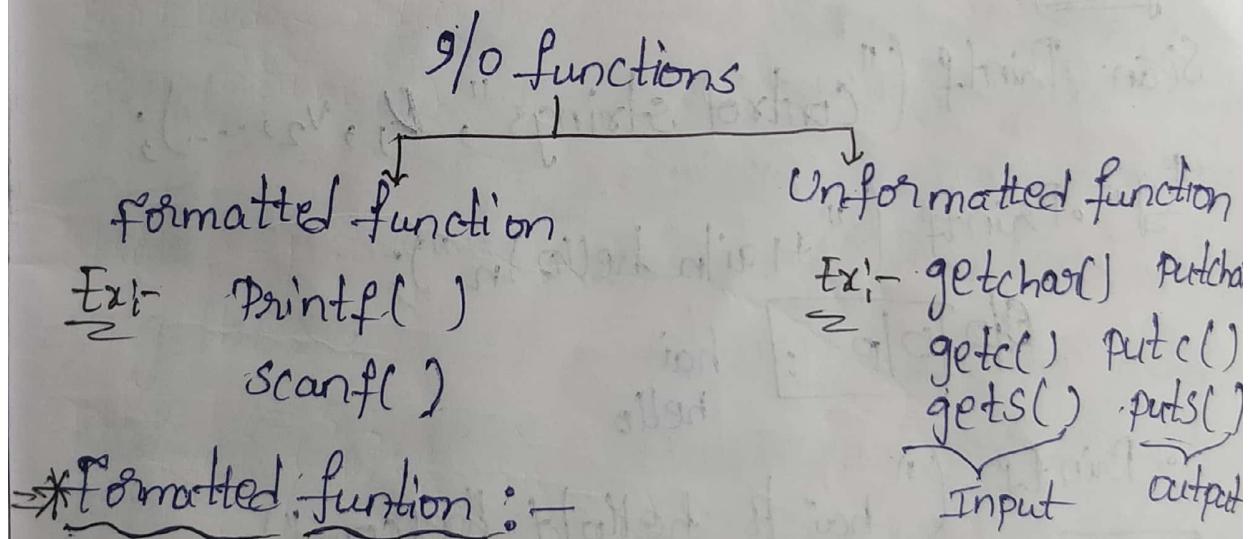
5/9/19

Input and Output functions (or) I/O statement  
The process of giving something to the programme is called input.

The process of taking something to the programme is called output.

- C language provide sum input and output functions are called predefined functions.
- Builtin (or) library.
- Input and Output function divided into two groups.

① formatted function      ② unformatted function



→ Formatted function :-

They are two types ① `scanf()` ② `printf()`

• scanf() :- It is an input function. It is used to get (read) input from the user during the execution of programme and stored in variable of specified format.

Syntax :- `Scanf("control strings", &v1, &v2, ...)`

Ex:- `scanf ("%d", &a); // for single input`

`scanf ("%d.%lf.%c", &a, &b, &c); // for multiple`

if we enter the values, through keyboard.  
10

10.5

A stored as

a  
10  
&a

b  
10.5  
&b

c  
A  
&c

- Printf () :- It is a output function. It is used to display text, constants, variable values on the screen in specified format.

Syntax :-

`Scanf Printf ("Control strings", &v1, &v2, ...);`

Eg:- `Printf ("hai\nhello\n");`

O/P :- 

O/P :	hai
	hello

• `Printf ("hai\nhello\nwelcome");`

O/P :- hai hello welcome // printf without argument.

int a=10;

• `Printf ("the value of a %.d\n", a);`

O/P :- 10

// printf with one argument.

3. int a;  
float b;  
char c;  
scanf ("%d.%f%c", &a, &b, &c);  
printf ("%d.%f%c", a, b, c);  
Output:- {  
10.5 }  
A

### Unformatted functions:-

It is used to read and write only the character values (not float & not integer).

The unformatted functions are like:-

getc()	putc()	⇒ getc() & putc()
getchar()	putchar()	functions are used
getS()	puts()	to read and write
Input	Output.	only one character at a time.

⇒ gets() & puts() - It is used to read and write the entire strings.