C Programming

1> what is computer ?

⇒ computer is an electronics Device, which can accept and store information or Process them and give the desirable result or output, it's also help us to Solve the various Problem.

c → commonly	o → operated	m → machine
p → Particularly	u → used for	t → technology
e → educational	r → research.	

2> what is Programming ?

⇒ Programming is a Process to write a code to construct an application and designing and building an executable computer Program.

3> what is c ?

⇒ c is a general-purpose, high level Programming language, which is mother of all language. and c is called Procedure oriented Programming language. (POP).

4> what is compiler ?

⇒ compiler is a computer Program that translates computer code written in one Programming language into another programming language, that convert .i to .obj. $a.i \xrightarrow[\text{compiler}]{} a.obj$

5> why header files are used ?

⇒ A header file is used to define all the function, variables contained in any function library,

It is a file where Some Predefined function work.

6) why hatch(#) is used in header files?

⇒ The hash (#) symbol helps the compiler to load the header files which are needed foro the complication of whole Progroam. (It is a Preprosscer).

7) who developed c language?

⇒ Dennis Ritchie at Bell labs (USA) in 1972.

8) what is operating System?

⇒ It is a system Software, it does process management, file management, memory management, it also acts as a manager of computer system.

Ex:- windows 7, windows 8, Linux, Mac-OS etc.

9) why semicolon (;) is used?

⇒ To terminate the statement.

10) why {} is used?

⇒ To make a complete block.

11) why main funn used in c Progroam?

⇒ It is the core of every program. It contains Instruction that tell the computer to carry out whatever task your progroam is designed to do.

12) what is Algorithm?

⇒ write mathematical & logical problem step by step. It help the computer analyzing. Algorithms are universal

* Algorithm of Series:

```
Step 1 :  x = 1
Step 2 :  Print(x)
Step 3 :  x + 1
Step 4 :  if(x < 11)
              goto Step 2 ;
Step 5 :  Stop ;
```

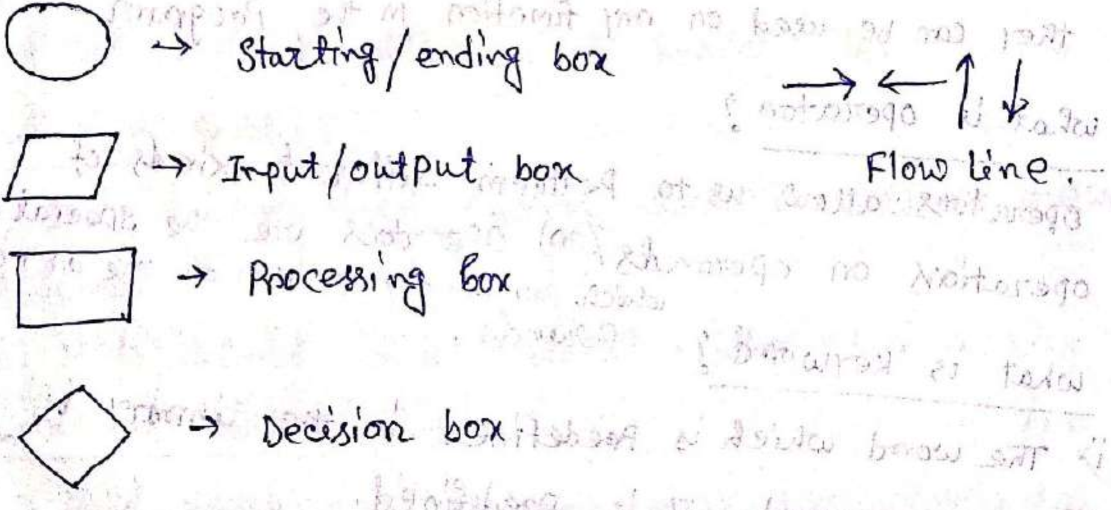
13) what is Flow chart ?

⇒ Graphical view of Algorithm is called flow chart.

i) It reduces program making time.

ii) Also reduces error in time.

Flow chart symbols



14) what is Operand ?

⇒ The data or variable on which the operation is performed is called operand.

15) what is Datatype ?

⇒ i) It is a type of data which is used in the Program.

ii) The datatype defines the amount of storage area allocated to variables.

Ex :- int (2 byte), char (1 byte), Float (4 byte), Double (8 byte)

6) Types of operator ?

> i) Arithmetic operator
ii) Relational "
(iii) Logical "
iv) Bitwise "
(v) Conditional "
(vi) Increment/Decrement operator

17) what is local variable ?

=> The variable which is defined inside a block is called as local variable.

18) what is Global variable ?

=> Global variable are declared outside any function and they can be used on any function in the Program.

19) what is operator ?

=> operators allow us to Perform different kinds of operations on operands.(or) Operators are the special Symbol which Perform any operation on one or more operands.

20) what is keyword ?

=> i) The word which is Predefined in the library is called keyword. It is Predefined.

ii) we can't use keyword as a variable.

21) what is variable ?

=> It is a name of Storage area which is used to store data or information.

22) what is Predefined function ?

=> The function which is Predefined in the library is called Predefined function. It's also called library function.

Ex :- Printf, Scanf, getch, clrscr. etc

3) What is Printf() ?

⇒ i) It is a Predefined function which is used to Print data or information, output Result on to the output Screen.

(ii) It is defined inside stdio.h header file.

4) What is Scanf() ?

⇒ (i) It is a Predefined function which is used to assign the value.

(ii) It is defined inside stdio.h header file.

25) What is clrscr() ?

⇒ i) It is a Predefind function which can clear Previous output Screen.

(ii) It is defined inside conio.h header file.

26) What is getch() ?

⇒ i) It is a Predefined function which is hold the output Screen.

(ii) It is defined inside conio.h header file.

27) What is void ?

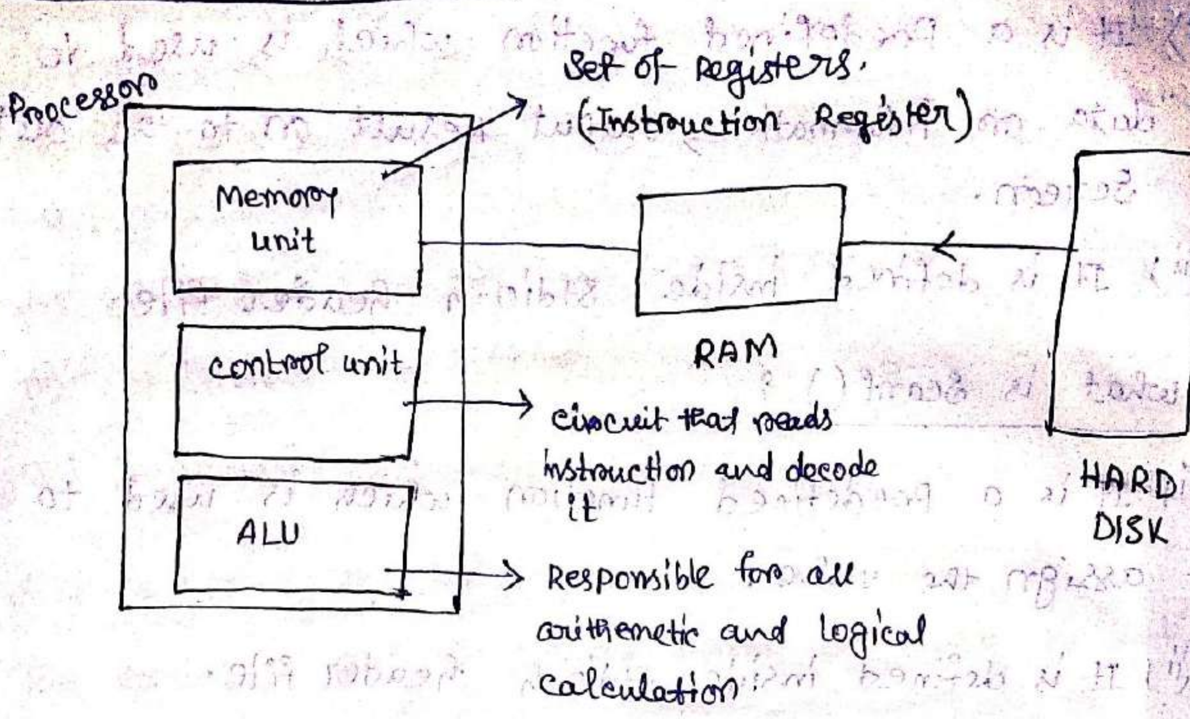
⇒ It is a Keyword. The meaning of the word is empty.

28) What is the full name of conio and Stdio ?

⇒ Stdio → Standard input output

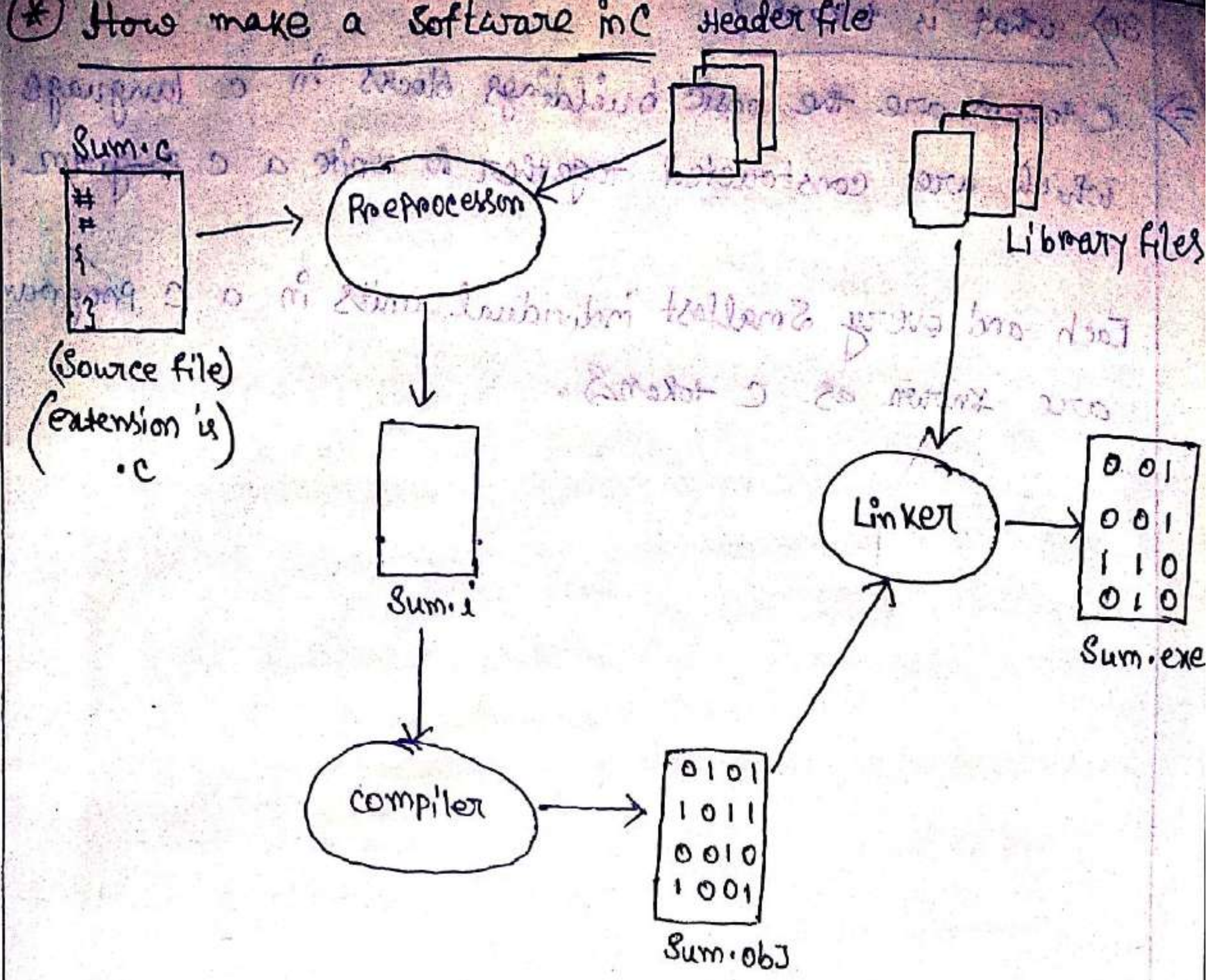
conio → console input output

(*) Execution of Program :->


Processor
Set of Registers.
(Instruction Register)
Memory unit
control unit
ALU
RAM
HARD DISK
circuit that reads instruction and decode it
Responsible for all arithemetic and logical calculation

when any Program saved in hard disk, we double click in this file and it makes a copy that load Save in Ram. Then the instruction coming at memory unit where has so many memory devise which is called as Register. In this Register there has one instruction Register where the instruction is saved. Then it comes in control unit where that reads the instruction and decode it and then instructed send at ALU, it responsible for all arithemetic and logical calculation and the instruction will exicute.

(*) How make a Software in C



Making a Software is depends upon the operating System for window OS, the software is .exe and For mac OS, the Software is .dmg file. In C language, the Source file's extension is .C. But it is not Software because operating System know .exe file as a Software. Now there was a Software that called as Preprocessor. The hash (#) Symbol helps ~~the~~ to load the header files. Preprocessor mined the contains of header files with the Source file and make a new file that's extension's is .i. Now using of compiler Software it converts in object file. In object file there are So many code that the OS is not known. So Linker connects the object file and library file and make .exe file.

30) what is tokens?

=> C tokens are the basic buildings blocks in c language which are constructed together to write a C Program.

Each and every smallest individual units in a C Program are known as C tokens.

* Structure of C

1> Header file

2> main function

3> open curly Braces

4> Logical variable declaration.

5> Input the Program

6> Process (Arithmetic & logic unit)

7> output of the Program

8> End

9> close curly Braces.

Datatype

1> Primary datatype.

2> Derived "

3> user defined "

Datatype of variable.

Primary Datatypes.

char	int	Float	Double
1 byte	2 byte	4 byte	8 byte.
Ex: a, b, c	Ex: 1, 2, 3	Ex: 1.5, 7.8	Ex: 1.04671
(%c)	(%d)	(%f)	(%lf)
(-128 to 127)	(-32,768 to 32,767)		

String → %s

Group of Char

%u → use for high range of intiger value

clrscr(); → To clear Screen.

getch(); → Hold Screen

conio → console input output

Stdio → Standard input output

Stdio.h

Printf() Scanf()

conio.h

clrscr() getch()

⊛ Sum of two numbers.

```
# include <stdio.h>  \
                      > Header file.
# include <conio.h>  /

void main()
{
   int a, b, c;
   clrscr();
   Printf("Enter two no");
   Scanf("%d%d", &a, &b);
   c = a+b;
   Printf("Sum = %d", c);
   getch();
 }
```

O/P

```
Enter two no
2  3

Sum = 5.
```

⊛ Group of char is called String.

```
# include <stdio.h>
# include <conio.h>
void main()
{
  char a[20];
  clrscr();
  Printf("Enter name");
  Scanf("%S", a);
  Printf("Name: %S", a);
  getch();
}
```

O/P

```
Enter name Anupam.
   Name: Anupam.
```

```
# include <Stdio.h>
# include <conio.h>
void main()
{  char name[30], branch[10];
   int reg;
   clrscr();
   Printf("Enter Your name");
   Scanf("%s", name);
   Printf("Your branch ");
   Scanf("%s", branch);
   Printf("Your registration no");
   Scanf("%d", &reg);
   Printf("Name: %s", name);
   Printf("\n Branch : %s", branch);
   Printf("\n Registration no : %d", reg);
   getch();
}
```

\n → used for new lines

* WAP accept two no of user and Swap without using third variable.

```
# include <Stdio.h>
# include <conio.h>
void main()
{  int a, b;
 Printf("Enter two number");
 Scanf("%d %d", &a, &b);
   a = a+b;
   b = a-b;   [b=a+b-(a=b)] →Single line.
   a = a-b;
Printf("%d %d", a, b);
```

using 3rd variable

```
c = a;
a = b;
b = c
```

⊛ Operators

operations allow us to perform different kinds of operations on operands.

types of operators :→ i) Increment/Decrement operators.

ii) Arithemetic operators

iii) Relational "

iv) Bitwise "

v) Logical "

vi) conditional "

Increment/Decrement

$P = +5$ → (unery Plus)

$x = -5$ → (unery minus)

++ (Increment)   -- (Decrement)

Types of Increment :- i) Pre increment

ii) Post increment.

(1) Incase of Pre increment :- Increment the value of right hand Side operate, then assign the value to Left hand Side operate.

Ex: $x = ++P$, $P = 9$.

$x = 10$
$P = 10$

(2) Incase of post increment :→ First assign the value of left hand Side operate and then increment the value Right hand Side operate.

Ex: $x = P++$, $P = 9$.

$P = 10$ $x = 9$

(*) Incase of Pre decrement :→ First decrement the value of Right hand side operate and then assign the value to Left hand Side operate.

$x = --P$. $P = 9$.

$x = 8$,
$P = 8$.

(*) Incase of post decrement :→ First assign the value to the left hand Side operate then decrement the value of Right hand Side operate.

$x = P--$ $P = 9$,

$x = 9$,
$P = 8$.

How to Solve the Expression

Rule 1 :→ First we calculate how many Pre increment as well as how many Pre decrement present in the given expression.

Rule 2 :→ Increment and decrement that much of time on the right hand Side operate.

Rule 3 :→ Put the updated value in all Places. calcula the final value then assign to left hand Side.

Rule 4 :→ Again count how many post increment and Pre increment present in the given expression

Rule 5 :→ Increment and decrement from the updated value of right hand Side operator.

Ex:- ++p+p--+++p+++p+--p+--p p=9

= (++p) + (p--) + (++p) + (++p) + (--p) + (--p)

= 10 + 10 + 10 + 11 + 10 + 9

= 60.

Pre increment → 1+1+1 = 3

Pre decrement → 1+1 = 2.

∴ 9+3-2 = 10

Post increment → 0

" decrement → 1

∴ 9+0-1 = 8.

Ex-2 x=9.

z = ++x + --x - x-- + ++x - x++ + x-- + --x.

= (++x) + (--x) - (x--) + (++x) - (x++) + (x--) + (--x)

= 10 + 9 - 9 + 9 - 9 + 10 + 8

= 28.

Pre increment → 2

Pre decrement → 2

9+2-2 = 9.

Post increment → 1

" decrement → 2

9+1-2 = 8.

Arithmetic operators

+ → Addition

- → Substraction

* → Multiplication

/ → Devision

% → Modulo.

a=7, b=3

Ex:

a+b = 10

a-b = 4

a*b = 21

a/b = 2 (int)

a%b = 1

(*) Find the Modulo value

```
# include <stdio.h>
# include <conio.h>
void main()
{  int a, b;
   a=10, b=5;
   clrscr();
   int c = a%b;
 Printf(" The modulo is %d", c);
 getch();
}
```

Relational operator

	Symbol	Name
1	==	equals to
2	!=	Not equals to
3	<	less than
4	>	Greater than
5	<=	Less than (or) equal to
6	>=	greater than (or) equals to

If a = 20, b = 25.

a == b → 0 (False)

a != b → 1 (True)

a < b → 1 (True)

a > b → 0 (False)

a <= b → 1 (True)

a >= b → 0 (False)

Relational operators compareing two variables after comparison it will return bolean values.

```
# include<stdio. h>
# include <conio.h>
void main()
{
  float a, b;
  clrser();
  Printf("Enter two no");
  Scanf("%f%f", &a, &b);
  if (a==b)
     Printf("True");
  else
     Printf("False");
  getch();
}.
```

Bitwise operators

The bitwise operator can be perform in between the integer value and the result will be integer value and result will be also integer value.

& → Bitwise AND

| → Bitwise OR

^ → Bitwise XOR

~ → Bitwise NOT

>> → Right shift

<< → Left shift

Decimal
↓
Binary
↓
operation
↓
Binary result
↓
Decimal

Bitwise AND

A	B	R
0	0	0
0	1	0
1	0	0
1	1	1

if $a = 3$, $b = 4$, $\therefore a \& b = ?$

$$\begin{array}{l|cccc}
 & 8 & 4 & 2 & 1 \\ \hline
a=3 & 0 & 0 & 1 & 1 \\
b=4 & 0 & 1 & 0 & 0 \\ \hline
 & 0 & 0 & 0 & 0
\end{array}$$

$\therefore a \& b = 0$

Bitwise OR

A	B	R
0	0	0
1	0	1
0	1	1
1	1	1

if $a = 3$, $b = 4$ $\therefore a | b = ?$

$$\begin{array}{l|cccc}
 & 8 & 4 & 2 & 1 \\ \hline
a=3 & 0 & 0 & 1 & 1 \\
b=4 & 0 & 1 & 0 & 0 \\ \hline
 & 0 & 1 & 1 & 1
\end{array}$$

$\therefore a | b = 4+2+1 = 7.$

Bitwise NOT

$a = 11,$

$\sim a = ?$

$$a = 11 \longrightarrow \begin{array}{cccc} 8 & 4 & 2 & 1 \\ \hline 1 & 0 & 1 & 1 \end{array}$$

⊛ Bitwise XOR

if a=6, b=12, a^b=?

a	b	R
0	0	0
1	0	1
0	1	1
1	1	0

	8	4	2	1
a=6	0	1	1	0
b=12	1	1	0	0
	1	0	1	0

= 10

⊛ Left Shift (Moving from the left side)

if a=4, a<<?

a=4 → 0100.

Here, 00000100

00010000

= 16

$\boxed{a << 2 = 16}$.

⊛ Right shift (moving from right side)

if, a=12, a>>2 = ?

a=12 → 1100

Here, 00001100

00000011 = 3.

$\boxed{\therefore\ a >> 2 = 3}$

Logical operator

Here we know need to convert to binary. we are taking as a non zero and zero.

&& → Logical AND
|| → Logical OR
! → Logical NOT

⊛ Logical AND :

a = 5 ⟶ 1
b = 0 ⟶ 0,

$\boxed{\therefore\ a \&\& b = 0}$

AND

0	1	0
0	0	0
1	0	0
1	1	1

⊛ Logical OR :

a = 10 ⟶ 1
b = 8 ⟶ 1

$\boxed{\therefore\ a || b = 1}$

NOR

0	0	0
0	1	1
1	0	1
1	1	1

⊛ Logical NOT :

a = 9 → 1 → 0.

$\boxed{!a = 0}$

Conditional operators

(*) **Ternery operators** :→ This operator can apply on three operators.

Syntax :→ (condition)? (True statement): (False statement);

```
a=7,      (a>b)? a : b
b=9
          (7>9)? 7 : 9
```

(*) **Decision control statements** :→

→ Simple if

→ if-else

→ if-else ladder

→ Nested if

→ Switch case.

(*) **Simple if**

Syntax :→
```
if (condition)
{
  statement;
}
```

(*) WAP to accept the mark of a student and check if the mark >= 45 then print Pass.

```
# include <stdio.h>
# include <conio.h>
void main()
{ int a;
 clrscr();
 printf("The mark is ");
scanf("%d", &a);
if(a>=45)
    printf("Pass");
getch();
}
```

⊛ if-else

Syntax:→

```
if(condition)
  {
     Statement;        ——→ True Statement
  }
else
    {
       Statement;      ——→ False Statement
    }
```

WAP to accept to unequal number to user and find the greater number

```
# Include <stdio.h>
# include <conio.h>
void main ()
{  int a, b;
   Printf(" Enter two no ");
   Scanf ("%d %d", &a, &b);
   if(a>b)
    {  Printf(" %d is big", a);
    }
   else
      {  Printf(" %d is ~~small~~ big", b);
      }
   getch();
}
```

2) Write a Program accept the number is even or Odd.

```
# include <stdio.h>
# include <conio.h>
void main()
{
  int a;
   Printf("Enter a number");
   Scanf("%d", &a);
   if(a%2==0)
      Printf("%d is even", a);
   else
      Printf("%d is Odd", a);
   getch();
 }
```

3) WAP to accept an alphabet from the user and check the alphabet is lower case and upper case.

```
# include <stdio.h>
# include <conio.h>
void main()
{
  char a;
  Printf("Enter one alphabet");
  Scanf("%c", &a);
  if( a>=65 && a<=90)
     Printf("It upper case");
  else
     Printf("It lower case");
  getch();
}
```

```
#include <stdio.h>
#include <conio.h>
void main()
{   int c, s;
    Printf(" Enter cost price ");
    Scanf("%d", &c);
    Printf("Enter Selling Price");
    Scanf("%d", & s);
   if ( s > c )
    {   Printf(" The Propiate is %d", s-c);
    }
   else
      Printf(" The loss is %d", c-s);
   getch ();
}
```

⊞ else if ladder

Syntax :->
```
if ( condition)
  { statement 1 ;
  }
else if(condition)
  { Statement 2 ;
  }

  - - - - - .

else
  { Statement n ;
  }
```

(*) WAP to accept three number of user and find the greatest number

```
#include <stdio.h>
#include <conio.h>
void main()
{
   int a, b, c;
   Printf(" Enter three number ");
   Scanf(" %d %d %d", &a, &b, &c);
   if (a > b && a > c)
     { Printf(" %d is big,"a);
     }
   else if( b > a && b > c)
      { Printf(" %d is big", b);
      }
   else
      { Printf(" %.d is big", c);
      }
  getch();
}
```

(*) WAP to accept the age of a Person if

age Less than 12 → child

if age 12 – 18 → teenage.

18 – 30 → Young

30 – 50 → middle

Otherwise old

```
# Include <stdio.h>
# include <conio.h>
void main ()
{
  int age ;
  Printf (" Enter Your age ");
  Scanf ("%d", & age);
  if (age < =12)
    { Printf (" You are child ");
    }
 else if (a>12 && a<=18)
    { Printf (" You are Teenage ");
    }
 else if (a>18 && a<= 30)
    { Printf (" Your are Young ");
    }
 else if (a>30 && a<=50)
    { Printf (" You are middle ");
    }
  else
    { Printf (" You are old ");
    }
 getch ();
}
```

⊛ nested if

Syntax :-

```
if (condition)
{
    if (condition)
      { Statement ;
      }
    else
      { Statement ;
      }
}
else
  { if (condition)
      { Statement ;
      }
    else
      { Statement ;
      }
  }
```

⊛ WAP to accept three number and find the greatest number.

```
⇒  # include < stdio.h >
   # include < conio.h >
   void main ()
   { int a = 10, b = 20, c = 30 ;
     ~~Printf~~ clrser();
   if (a > b)
   {   if (a > c)
       { Printf (" a is big ");
       }
     else
       { Printf ("c is big");
   }   }
else
   { if( c<b)
      {
        Printf(" b is big");
      }
     else
        { Printf(" c is big");
        }
   }
  getch();
}
```

Assignment

1) write a Program to find the Sum of n-number.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int n;
  Printf(" The number");
  Scanf(" %d", &n);
  Printf(" The Sum of 0 to %d is %d", n, n*(n+1)/2);
  getch();
}
```

2) WAP to convert doller in to roupees

```
#
#
main()
{ float n;
 Printf(" The doller Price is %");
 Scanf(" %f", &n);
 Printf(" The convert of Indian currency is %f", a* 69.35);
 getch();
}
```

3) WAP to find the volume of cylinder.

```
#
#
void main()
{ ~~int~~ float r, h;
   Printf("Enter the radious & height of cylinder");
   Scanf("%f%f", &r, &h);
   Printf("The volume is %f", 3.14*r*r*h);
   getch();
}
```

4) WAP to enter 4 digits number and find the Sum of first and last digit.

10) 1234 (102
1230
4 → 4th

1000) 1234 (1 → 1st
1000
234

```
=> # include<stdio.h>
   # include <conio.h>
   void main()
   { int a;
     Printf("Enter four digit number");
     Scanf("%d", &a);
     Printf("The Sum of 1st & 4th digits is %d", a/1000 + a%10);
     getch();
   }.
```

a/1000 ↓ 1st   a%10 ↓ 4th

5) Enter 3 digits number and find the Sum of Each digits

100) 123 (1 → 1st
100
23

10) 123 (12
120
3 → 3rd

100) 123 (1
100
10) 23 (2
20
3

```
=> # include <stdio.h>
   # include <conio.h>
   void main()
   { int a;
   Printf("Enter 3 digits no");
   Scanf("%d", &a);
```

a%100/10

```
Printf("The Sum of each digit is %d", a/100 + a%10 + a/10 %10),
getch();
}
```

6) WAP to check a year is leap year or not.

```
# include <stdio.h>
# include <conio.h>
void main()
{ int a;
  Printf(" Enter a year ");
  Scanf("%d", &a);
  if(a%4==0)
  {
        if(a%100==0)
        {     if(a%400==0)
                  Printf(" It is leap year ");
              else
                  Printf(" Not leap year");
        }
      else
          Printf(" It is leap year");
  }
else
   Printf(" Not leap year ");
 getch();
}.
```

year % 4 == 0 → leap year.

year % 100 == 0 → year%400==0. → Leap year

✱ Switch :-

Syntax :-

```
        Switch (condition)
      {
        Case label 1 : Statement ;
                       break ;
        Case label 2 : Statement ;
                       break ;

        - - - - - -

        Case label n : Statement ;
                       break ;

    default : Statement ;
    }
```

1) WAP to accept the alphabet and check they are vowel or consonant

```
# include <stdio.h>
# include <conio.h>
void main()
{  char ch;
   clrscr();
   Printf("Enter an alphabet");
   Scanf("%c", & ch);
Switch (ch)
{
   case 'a' :
   Case 'A' :
   Case 'i' :
   Case 'I' :
   Case 'e' :
   Case 'E' :
```

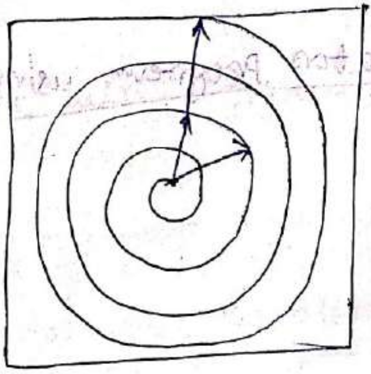
```
Case 'O':
Case 'O':
Case 'u':
Case 'U':

        Printf(" vowel");
        break;
 default : printf(" consonent");
 }
 getch();
}
```

2) WAP to accept the caculator program using Swich case

```
#
#
main()
{
 int a, b;    char op;
 clrscr();
Printf(" Enter two no ");
Scanf("%d %d", &a, &b);
Printf(" Enter operator ");
Scanf(" %c", & op);
Switch (op)
{
 Case '+' :  Printf(" The Sum is %d", a+b);
             break;
 Case '-' :  Printf(" The Substraction is %d", a-b);
             break;
 Case '*' :  Printf(" The multiplication is %d", a*b);
             break;
 Case '/':  Printf(" The devision is %d", a/b);
             break;
case '%' : Printf(" The modulo is %d", a%b);
           break;
default : Printf(" Give +, -, *, /, %");
}
getch();
}
```

⊛ LOOPS



i) initialization

ii) condition

iii) increment/decrement

* while

* do-while

* for

While statement (Entry check)

```
      initialization ;  1
      while (condition)  2
   [ {
 3 [   Statement ; 4
   [   update variable ; 5
   [ }
```

Note: If we Print n number of value the loop is exicute (n+1) time.

⊛ WAP to Print 1-10.

```
#
#
void main()
{ int i ;
 for (i = 1 ; i <= 10 ; i++)
 { Printf(" %d/n", i);
 }
 getch();
}
```

⊛ Do-while (Exit check)

Syntax

```
do
{
    Statement ;
    update variable ;
}
while (condition)
```

Example :-

```
    #
    #
    void main ()
  {  int i = 1 ;
   do
     { Printf (" %d\n", i);
        i++
     }
   while ( i <= 10);
   getch ( );
  }
```

Output

```
1
2
3
4
5
6
7
8
9
10.
```

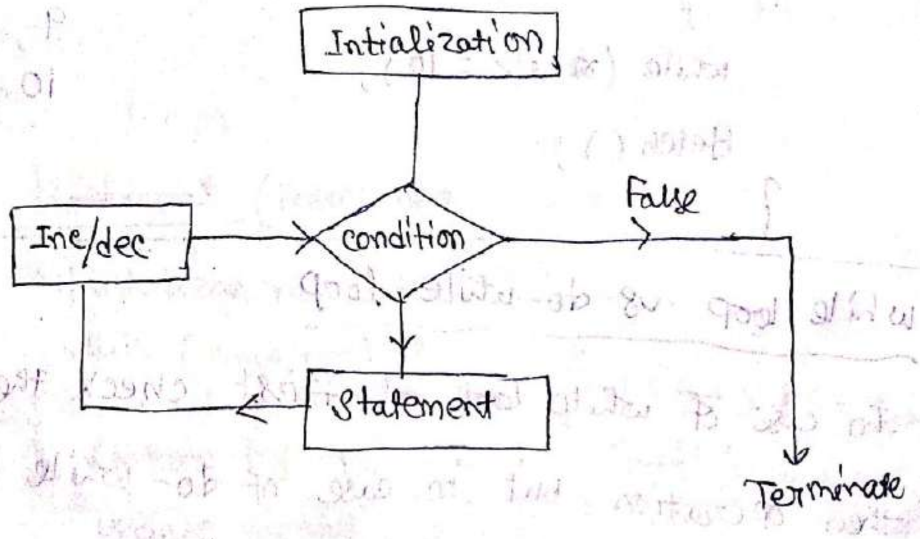
⊛ for loop

Syntax :-

```
         for (initialization ; condition; update      ) 0
                     1              2     variable
                                              5
  3 [ {
          Statement ; 4
      }
```

ex :- Print 1 - 10

```
#
#
void main ()
{ int i ;
  for (i = 1 ; i < = 10 ; i++)
  {
    Printf("%d\n", i);
  }
  Getch ();
}
```



* write a Program to Print 75-35

```
#
#
void main ()
{ int i = 75 ;
  for (i = 75 ; i >= 35 ; i--)
  {  Printf("%d\n", i);
  }
  getch();
}.
```

(*) write a Program to find the Sum and multiplication of 1-5.

```
#
#
void main()
{
   int i, sum = 0, mul = 1 ;
   for (i = 1 ; i <= 5 ; i++)
   {
       sum = sum + i ;
       mul = mul * i ;
   }
 Printf("sum = %d", sum);
 Printf("multiplication = %d", mul);
 getch();
}
```

(*) WAP to find 2+4+6+...+n

=>

```
#
#
main ()
{  int i, a, s = 0;
   Printf("Enter a no ");
   scanf("%d", &a);
   for (i = 2 ; i <= n ; i+2)
   {
     s = s + i ;
   }
   Printf("sum = %d", s);
   getch();
}
```

⊛ WAP to find $1^2+2^2+3^2+\ldots+n^2$.

```
#
#
void main()
{
   int i, n, s=0;
   Printf("Enter a no");
   Scanf("%d", &n);
   for (i=1; i<=n; i++)
   {
      s = s + i*i;
   }
   Printf("Sum is %d", s);
   getch();
}
```

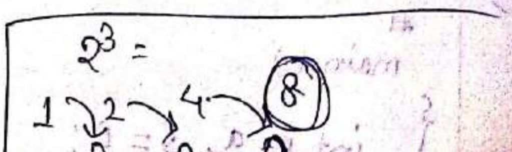
⊛ WAP to accept a number from the user and find factorial.

```
#
#
main()
{
   int i, n, f=1;
   clrscr();
   Printf("Enter a no");
   Scanf("%d", &n);
   for (i=1; i<=n; i++)
   {
      f=f*i;
   }
   Printf("Factorial = %d", f);
   getch();
}.
```

⊛ WAP to accept a number from the user find the Power of 1st number to 2nd number.

$a^b = ?$

```
#
#
main();
{  int a, b, i, s=1;
   Printf("Enter two no");
   Scanf("%d%d", &a, &b);
   for(i=1; i<=b; i++)
   s = s*a;
   Printf("%d", s);
   getch();
}
```



(*) write a Program to accept a number from the user and find the Power of it's own.

$a^a = ?$

```
#
#
main()
{ int a, i, s=1;
  Printf(" Enter a no ");
  Scanf(" %d", &a);
  for(i=1 ; i<=a ; i++)
  {   s = s*a;
  }.
  Printf (" ~~Enter~~ %d ", s);
  getch();
}
```

(*) WAP to accept two number and Print it assending order.

```
#
#
void main()
{ int a, b, i
  Printf("Enter two no");
  Scanf("%d %d", &a, &b);
  if(a>b)
  { for( i=b; i<=a; i++)
        { Printf(" %d", i);
        }
  }
else if( b>a)
   { for(i=a; i<=b; i++)
        { Printf ("%d", i);
        }
   }
else
   { Printf("%d", a);
   }
   getch();
   }
```

O/P

```
Enter two no
4
8
4 5 6 7 8
```

⊛ write a Program to find even number or odd no In between 1 to 10

```
#
#
void main()
{
  int i;
  for(i=1; i<=10; i++)
  {
      if(i%2==0)
         Printf("%d is even", i);
      else
         Printf("%d is odd", i);
  }
  getch();
}
```

⊛ WAP to find the Sum of even number and odd number

```
#
#
main()
{
  int n, i, es=0, os=0;
  clrscr();
  Printf("Enter a number");
 Scanf("%d", &n);
 for(i=1; i<=n; i++)
{
  if(i%2==0)
     es= es+i;
 else
     os=os+i;
}
Printf("The sum of even number is %d", es);
Printf("The sum of odd number is %d", os);
getch();
}
```

(*) WAP to find three number from the user and find to the Sum of Square of indivisual number

1 2 3

$1^2 + 2^2 + 3^2$

```
#
#
main()
{ int i, a[3], Sum = 0 ;
  Printf("Enter three no");
  for (i=0; i<3; i++)
  { Scanf("%d", &a[i]);
  }
 for (i=0; i<3; i++)
 {
     Sum = Sum + a[i]* a[i];
  }
 Printf("The Square Sum of this number is %d", Sum);
 getch();
 }.
```

(*) WAP to accept a number from the user check the number is Prime or not.

```
#
#
main()
{
  int i, n, c=0;
  Printf("Enter a number");
  scanf("%d",& n);
for (i=2; i<=n; i++)
{
     if(n% i == 0)
     { Printf ("Not Prime");
       C=1;
       break;
     }
 }
     if(c==0)
     {
        Printf("Number is
                  Prime");
     }
     getch();
 }
```

(*) WAP to accept the number from the user and do reverse

```
#
#
void main()
{   int n, i, rem, rev = 0;
    clrscr();
    Printf("Enter a number");
    Scanf("%d", &n);
    while (n!=0)
    {
        rem = n % 10;
        rev = 10 * rev + rem;
        n = n/10;
    }
    printf("%d is reverse", rev);
    getch();
}.
```

```
while (n!=0)
{   rem = n % 10;
    Printf("%d", rem);
    n = n/10;
}
```

(*) WAP to accept from user and check the number is Pallindrome. [The number which is equal with the reverse of the number].

```
=> #
   #
   main()
   {
     Int n, rev=0, rem;
     Printf("Enter a number");
     Scanf("%d", &n);
     while (n!=0)
     {
       rem = n%10;
       rev = 10 * rev + rem;
   }   n = n/10;
```

```
if (n== rev)
{ Printf("Palindrome");
}
else
  Printf("Not Palindrome");
getch();
}
```

(*) WAP to user and check the number is Amstrong.

[153, $1^3+5^3+3^3$ = 1+125+27 = 153].

```
#
#
main()
{
  int n, r, sum=0
  clrscr();
  Printf("Enter a no");
  Scanf("%d", &n);
  while(n!=0)
  {
    r = n%10;
    Sum= Sum + r*r*r;
    n= n/10;
  }

  if (n== Sum)
     Printf("Amstrong");
  else
     Printf("Not Amstrong");

  getch();
}.
```

(*) WAP to accept a number from the user and check the number is Perfect or not

[Note: A number is equal with the sum of factors of the number, except the number]

$$6 \downarrow 1+2+3 = 6.$$

```
#
#
main()
{
  int n, S=0, i
  Printf("Enter a no");
  Scanf("%d", &n);
  for (i=1; i<n; i++)
  {
    if(n%i == 0)
     { S = S+i;
     }
  }

  if(S==n)
     Printf("perfect");
  else
     Printf("Not Perfect");
  getch();
}.
```

Nested for :→ A for loop inside another for loop.

Syntax

```
for( initialization; condition; update variable)
{  for(initialization; condition; update variable)

      { statement;
      }
   statement;
}
```

Program for Print Pattern :→

```
* #include <stdio.h>
  # include <conio.h>
  void main()
  { int i, J;
    for(i=1; i<=4; i++)
    { for(J=1; J<=4; J++)
      {  Printf(" * ");
      }
     Printf(" \n");
    }
   getch();
  }
```

output

```
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
```

```
* #
  #
  main()
  { int i, J;
  for(i=1; i<=5; i++)
  { for(J=1; J<=5; J++)
    {  if(J<=i)
          Printf(" * ");
    else Printf("   ");
    }
```

O/P

	1	2	3	4	5
1	*				
2	*	*			
3	*	*	*		
4	*	*	*	*	
5	*	*	*	*	*

i	J
1	1
2	11, 2
3	1, 2, 3
4	1, 2, 3, 4
5	1, 2, 3, 4, 5

i	J
1	J<=1 .
2	J<=2
3	J<=3
4	J<=4
5	J<=5

$\boxed{J <= i}$

```
Printf("\n");
}
getch();
}
```

(*)

```
#
#
main()
{
  int i, J;
  for(i=1; i<=5; i++)
  {
   for(J=1; J<=5; J++)
    {
      if(i+J <= 6)
        Printf(" * ");
      else
        Printf("  ");
    }
    Printf("\n");
  }
  getch();
}
```

O/P

	1	2	3	4	5
1	*	*	*	*	*
2	*	*	*	*	
3	*	*	*		
4	*	*			
5	*				

i	J	
1	1 2 3 4 5	i+J <=6
2	1 2 3 4	
3	1 2 3	
4	1 2	
5	1	

For Extra Star Point

visit → Star Pattern Programs in C

Youtube (mysirg.com)

Compile time initialization	Runtime Initialization

void main ()
{ int a = 10, b = 20, c;
C = a+b;
Printf(" %d", c);
getch();
}
	void main()
	{ int a, b, c;
	Printf("Enter two no");
	Scanf(" %d%d", &a, &b);
	c = a+b;
	Printf(" c = %d", c);
	getch();
	}.

ARRAYS

Group of elements storing in same datatype. it can be store many value but variable can store single value.

Array is correction of similar types of data these are stored in continuous memory location.

* Base address :- That address of the first index of the array is called Base address.

Rule-1 :-> Array index always starts from 0.

Rule-2 :-> Last index of the array will be (total size - 1)

Rule-3 :-> Syntax> Datatype Arrayname[Size];

Rule-4 :-> whenever we want to take the value from the user and put into the array by using for loop.

Rule-5 :-> whenever we want to show the value from the array to the user by using for loop.

Rule-6 :→ whenever we want to do the operation of array elements at that time we using for loops we can take 1,1 elements to do the require operation.

one Dimensional Array

Syntax :→ Datatype arrayname [Size];

↓ int, double, float, char

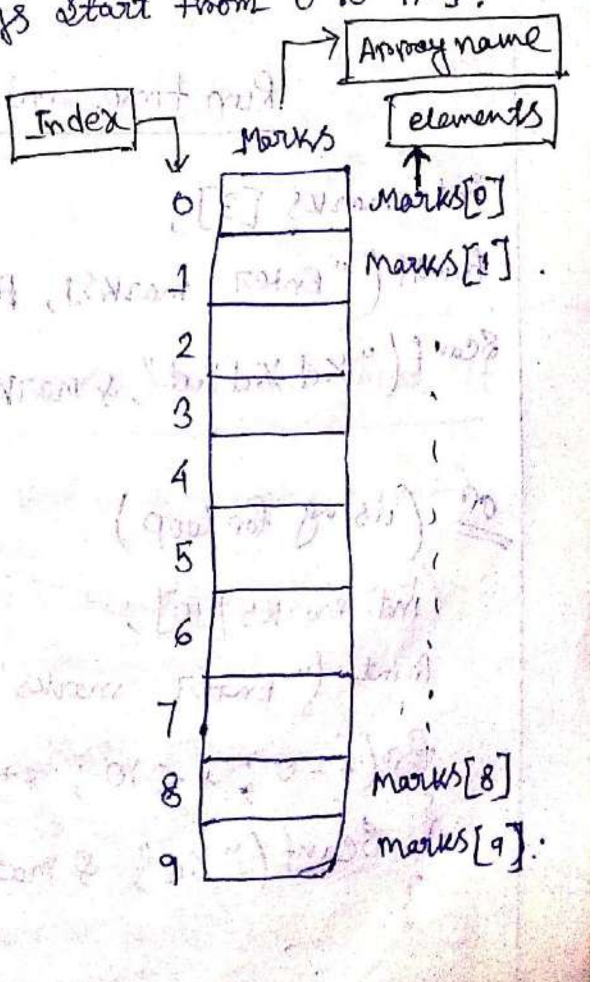
└→ maximum number of elements.

Arrays :→ 1) one Dimensional array

2) two " "

3) Multi " "

(*) int marks [10] ⟶ always start from 0 to n-1.



compile time initialization

Datatype Array name [size] = {value1, value 2, value 3};

Ex:1

int marks[4] = {70, 80, 90; 75};

marks		
0	70	marks[0]
1	80	:
2	90	:
3	75	Marks[3]

Ex:2

int marks [6] = {30, 60, 40};

marks		
0	30	marks[0]
1	60	mark[1]
2	40	:
3	0	:
4	0	:
5	0	marks[5]

Ex: 3

int marks [ ] = {30, 40, 50};

marks		
0	30	marks[0]
1	60	marks[1]
2	40	marks[2]

Run time initialization

```
int marks [3];
Printf("Enter marks1, Marks 2, marks 3");
Scanf("%d %d %d", & marks[0], & marks[1], & marks[2]);
```

or (using for loop)

```
int marks [10];
Printf("Enter marks");
for(i = 0; i < 10; i++)
{ Scanf("%d", & marks[i]);
}.
```

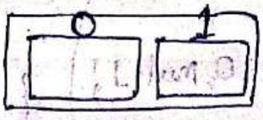
Runtime → Reading inputs from the user during Program exituation.

Two Dimensional Array

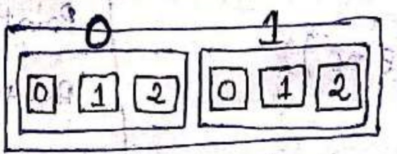
Syntax :-> Datatype arrayname [row size] [col size] ;

∴ b[2][3]

Here, b[2] =



b[2][3] =



1) WAP to accept 5 number from the user and print it.

```
# include <stdio.h>
# include <conio.h>
void main ()
{
  int n[5], i ;
  Printf (" Enter five no") ;
  for (i= 0; i<5; i++)
  {
   Scanf (" %d", & n[i]) ;
  }.
  Printf (" The number's are") ;
  for (i = 0 ; i <= 4 ; i ++)
  {
   Printf (" %d ", n[i]) ;
  }
  getch () ;
}.
```

2) WAP to accept 10 number from the user and print it reversely.

```
#.
#
void main ()
{ int n[10], i ;
  Printf (" Enter 10 no") ;
  for (i=0; i<10; i++)
  {
   Scanf (" %d", & n[i]) ;
  }
  Printf (" The reverse no are") ;
  for (i = 9 ; i >= 0 ; i--)
  {
   Printf (" %d ", n[i]) ;
  }
  getch () ;
}
```

3) WAP to accept five number from the user and print it. the Sum of all the number and multiplication of number.

```
#
#
void main()
{
  int a[5], i, Sum=0, mul=1;
  clrscr();
 Printf("Enter 5 no");
 Scanf("%d",
 for(i=0; i<5; i++)
  { Scanf(" %d", &a[i]);
  }
for(i=0; i<5; i++)
{
  Sum = Sum + a[i];
  Mul = Mul * a[i];
}
Printf("Sum = %d", Sum);
Printf("The mult = %d", Mul);
getch();
}
```

4) WAP to accept Six no from the user and find even number as well as odd number.

```
#
#
void main()
{
  int a[6], i,
  Printf("Enter Six no");
  for(i=0; i<6; i++)
  { Scanf(" %d", &a[i]);
  }
for(i=0; i<6; i++)
{
   if(a[i] % 2 == 0)
    {  Printf(" %d is even", a[i]);
    }
   else
      Printf(" %d is odd", a[i]);
}
getch();
```

5) WAP to accept find three number from the user and find to the Sum of Square of indivisual number

```
#
#
void main()
{  int n[3], i, S=0;
   clrscr();
   Printf(" Enter 3 no");
   for (i=0; i<3; i++)
   { scanf(" %d", & n[i]);
   }
for(i=0; i<3; i++)
{
   S = S + n[i] * n[i];
}
Printf(" The Answer is %d", S);
getch();
}.
```

6) WAP to any number and find the total Sum and Multiplication

```
#
#
void main()
{  int a[100], i, n, S=0, m=1;
   clrscr();
   Printf("Enter noOflimit");
   ~~for(i=0)~~ Scanf(" %d", n);
   Printf(" Enter the no");
   for(i=0; i<n; i++)
   { scanf("%d", & a[i]);
   }
for (i=0; i<n; i++)
{
   S = S + a[i];
   m = m + a[i];
}
Printf(" The Sum=%d", S);
Printf ("mult = %d", M);
getch();
}.
```

7) WAP to accept any number and find the Sum of even and odd number.

```
#
#
void main()
{
    int a[100], i, n, es=0, os=0;
    clrscr();
    Printf(" The no of Limits");
    Scanf("%d", &n);
    Printf("Enter the no ");
    for(i=0; i<n; i++)
    { Scanf("%d", & a[i]);
    }
for (i=0; i<n; i++)
{ if (a[i] % 2 ==0)
   { es = es + a[i];
   }
  else
   { os = os + a[i];
   }
}
  Printf(" The even Sum
            = %d", es);
  Printf("The odd Sum= %d", os);
  getch();
}.
```

8) WAP to accept any number from the user and find the largest number.

0	1	2	3
4	6	3	8

l=4

```
#
#
void main()
{ int a[100], n, i, l;
    Printf(" Enter the size");
  Scanf("%d", &n);
  Printf(" Enter the no");
for (i=0; i<n; i++)
{ Scanf ("%d", & a[i]);
}
l = a[0];
for (i=1; i<n; i++)
{
   if(a[i] > l)
     { l = a[i];
     }
   else
       l = a[i-1]
}
Printf(" The largest no %d", l);
  getch();
}
```

9) WAP to accept any number from the user and find the lowest number.

0	1	2	3
4	3	5	2

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int a[100], n, i, l;
  clrser();
  Printf("Enter the Size");
  Scanf("%d", &n);
  Printf(" Enter the no");
  for(i=0; i< n; i++)
     Scanf(" %d", &a[i]);
  l = a[0];
  for(i=1; i<n; i++)
  {  if( l > a[i] )
     {
        l = a[i]
     }
  }
  Printf(" The lowest no is
              %d", l);
  getch();
}.
```

10) WAP to accept 5 number to user and also accept a number check the number is given in the array or not is given then the position of the number.

```
#include <stdio.h>
#include <conio.h>
void main()
{
 Int a[5], i, n, c = 0
 clrscr();
 Printf(" Enter 5 no");
 for( i=0; i<5; i++)
    Scanf(" %d", & a[i]);
 Printf(" Enter a no to check");
 Scanf(" %d", & n);
 for (i=0; i<5; i++)
 {
    if( a[i] == n)
     { c = 1;
     } break;
 if(c==0)
 { Printf("The no is not given");
 }
 else
 {
    Printf("The no is given");
   Printf("Position = %d", i+1);
 }
 getch();
}
```

* WAP to Print 3x3 matrix.

```
# include <stdio.h>
# include <conio.h>
main()
{ int a[3][3], i, J;
  clrscr();
  Printf("Enter the elements");
  for(i=0; i<3; i++)
  {
    for(J=0; J<3; J++)
    { scanf("%d", &a[i][J]);
    }
  }
Printf("The elements are");
for(i=0; i<=2; i++)
{ for(J=0; J<=2; J++)
  {
   Printf("%d", a[i][J]);
  }
  Printf("\n");
}
 getch();
}.
```

* WAP to accept a 4x4 matrix and find it's transpose

```
# include <stdio.h>
# include <conio.h>
void main()
{ int a[4][4], i, J
  clrscr();
  Printf("Enter the elements");
  for(i=0; i<4; i++)
  { for(J=0; J<4; J++)
    { scanf("%d", &a[i][J]);
    }
  }
Printf(" The transpose is ");
for(J=0; J<4; J++)
{ for(i=0; i<4; i++)
  { Printf("%d", a[i][J]);
  }
Printf("\n");
}
 getch();
}.
```

```
Printf("The elements of 2nd matrix");
for(i=0; i<m; i++)
{ for(J=0; J<n; J++)
  { scanf("%d", & mat2[i][J]);
  }
 }

for(i=0; i<m; i++)
{ for(J=0; J<n; J++)
   { mat3[i][J] = mat1[i][J] + mat2[i][J];
   }
 } Printf("The sum of 2 matrix is\n");
for(i=0; i<m; i++)
{ for(J=0; J<n; J++)
   {
    Printf(~~"The Sum is~~"%d", mat3[i][J]);
   }
    Printf("\n");
 }
}
else
  { Printf("The addition is not Possible");
  }
getch();
}
```

(*) WAP to create mxn matrix. Find the sum of the diagonal of this matrix

```
#include<stdio.h>
#include<conio.h>
main()
{
 int i, J, mat[100][100], m,n,s=0;
 clrser();
Printf("Enter row & colum");
scanf("%d %d", &m,&n);
Printf("Enter the elements");
~~Scanf("%~~
for(i=0; i<m; i++)
{
  for(J=0; J<n; J++)
  { Scanf("%d", &mat[i][J]);
  }
}
```

```
for(i=0; i<m; i++)
{ for(J=0; J<n; J++)
  { if(i==J)
    { S=S+mat[i][J];
    }
  }
}
Printf("%d is sum", S);
getch();
}
```

(*) WAP to create two matrix mxn and Pxq and find the multiplication of this two matrix

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int mat1[100][100], mat2[100][100], mat3[100][100], m,n,p,q, i,J,K;
 clrser();
 Printf("Enter the row & colomn of 1st matrix");
 Scanf("%d%d", &m, &n);
 Printf("Enter the row & colorm of 2nd matrix");
 scanf("%d%d", &p, &q);
if(n==p)
{
  Printf("Enter the elements for 1st matrix\n");
  for(i=0; i<m; i++)
  { for(J=0; J<n; J++)
     { scanf("%d", & mat1[i][J]);
     }
  }
  Printf("Enter the elements for 2nd matrix\n");
  for(i=0; i<p; i++)
  {  for(J=0; J<q; J++)
      { scanf("%d", & mat2[i][J]);
      }
  }
  for(i=0; i<m; i++)
  { for(J=0; J<n; J++)
    {
      mat3[i][J] = 0;
     for(K=0; K<n; K++)
     {
      mat3[i][J] = mat3[i][J] + mat1[i][K] * mat2[K][J];
     }
    }
  }
  Printf("The multiplication is ");
  for(i=0; i<m; i++)
  { for(J=0; J<q; J++)
    { Printf("%d", mat3[i][J]);
    } Printf("\n");
  }
}
else
   Printf("multiplication is not Possible");
getch();
}
```

* Bubble Sort

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int n, a[100], i, J, temp;
  clrscr();
  Printf("Enter the Size");
  Scanf("%d", &n);
  Printf("Enter the no");
  for(i=0; i<n; i++)
     Scanf("%d", &a[i]);
for(i=0; i<n-1; i++)
{  for(J=0; J<n-1-i; J++)
    ~~{ if (a[i] > a[i+1])~~
     {
{ if (a[J] > a[J+1])
  {  temp = a[J];
     a[J] = a[J+1];
     a[J+1] = temp;
  }
 }
 }
Printf("The Sorted array is");
for(i=0; i<n; i++)
 {  Printf(" %d", a[i]);
 }
  getch();
 }
```

String → group of character.

String :-> collection of more than one character called String by default System store null value at tee end of the String.

(*) ASCII code of null character is 0.

\0 → null character.
(Slash zero)

character — Alphabets
numeric
Special character (%, ', ', &, ... etc).

compile time :->

char A[6] = {'H','E','L','L','O','\0'};

Input	output
Scanf()	Printf()
getchar()	Putchar()
gets()	Puts()

9 → integer
'9' → character
"9" → String.

Control identifier → %d.
& → no need.

Ex :->

```
char A[10];
~~Printf~~ Scanf("%s", A);
Printf("%s", A);
```

* Termination — Space.
* It fails to read line of text.

(*) Read Single world.

OP

```
Hello_how
Hello
```

getchar() Putchar()

* It will read line of text.

* Termination — newline (\n)

* control identifier — %c.

* Read character by character.

Ex :-
```
char ch[100];
getchar (ch);  → scanf ("%c", ch);
putchar (ch);
```

gets() Puts()

* Read line of text.

* Termination — '\n' → Enter button.

* Read String Directly.

String handling function

Function	Function name
String length	Strlen()
String copy	Strcpy()
String concatination	Strcat()
String compare	Strcmp()
String reverse	Strrev()
String duplicate	Strdup()
lower to upper	Strupr()
upper to lower	Strlwr()

⊛ # include <string.h>

└→ Predefined of string handing function.

1) WAP to accept a name from the user & Print it.

```
# include <stdio.h>
# include <conio.h>
void main()
{
 char ch[100];
 clrscr();
 Printf("Enter any name");
 scanf("%s", ch);
 Printf("The name is %s", ch);
 getch();
}
```

2) WAP to accept a string from the user and Print it vertically.

```
#include <stdio.h>
#include <conio.h>
void main()
{
   char ch[100];
   int i;
   Printf("Enter a name ");
  Scanf("%s", ch);
for(i=0; ch[i]!='\0'; i++)
  {
    Printf("%c\n", ch[i]);
  }
  getch();
}.
```

op

R	A	m	\0

3) WAP to accept a string from the user and find the strength of string.

0	1	2	3
R	A	M	\0

```
#include<stdio.h>
#include<conio.h>
void main()
{ char ch[100];
   int i;
  Printf("Enter a name");
 Scanf("%s", ch);
for(i=0; ch[i]!='\0'; i++)
 {
 }
Printf("Length = %d", i--);
 getch();
}.
```

```
 #include <stdio.h>
 #include <conio.h>
 #include <string.h>
void main()
{
 char ch[100];
 Printf("Enter a name");
 Scanf("%s", ch);
int n = strlen(ch);
Printf("The length = %d", n);
getch();
}
```

4) WAP to accept a string from user and Print it reverse.

```
#include <stdio.h>
# include <conio.h>
void main ()
{ char ch[100] ;
  int i ;
Printf ("Enter name");
 Scanf ("%s", ch) ;
for (i=0 ; ch[i] != '\0' ; i++)
{

}
 i-- ;
for (i-- ; [illegible][i] >= 0 ; i--)
{ Printf ("%c", ch[i]) ;
}
 getch() ;
}.
```

R	A	M	\0
0	1	2	3
M	A	R	

```
#include <stdio.h>
# Include <conio.h>
# include <String.h>
main ()
{ char ch[100] ; int i ;
  Printf ("Enter name") ;
  Scanf ("%s", ch) ;
   Strrev (ch) ;
  Printf ("%s", ch) ;
   getch() ;
}
```

5) WAP to accept a string from the user and copy that String to another String.

```
#
#
main ()
{ Char ch1[100], ch2[100] ;
  int i ;
 Printf ("Enter a String") ;
 Scanf ("%s", ch1) ;
 for (i=0 ; ch[i] ; i++)
      ch2[i] = ch1[i] ;
Printf (" The copy String is ") ;
Printf ("%s", ch2) ;
getch () ;
}.
```

```
#
#
# include <String.h>
Void main ()
{
 Char ch1[100], ch2[100] ;
 Printf ("Enter a String") ;
 Scanf ("%s", ch1) ;
 Strcpy (ch2, ch1) ;
 Printf ("%s", ch2) ;
 getch() ;
}.
```

6) WAP to accept a string from the user and also accept from the user and check whether the char is Present string or not. if Present then Print the Position.

```
#
#
void main ()
{
  char ch[100], n;
  int i, pos=0;
  clrser();
  Printf("Enter a String");
  Scanf("%S", ch);
  Printf("Search character");
  Scanf("%c"&n);
  for(i=0; ch[i]; i++)
  {  if(ch[i]==n)
       {    pos = i+1;
            break;
       }
  }
  if( pos = 0)
  {  Printf("The elements
              is not there");
  }
  else
  {
     Printf("The elements is
              there");
     Printf("The position is
              %d", pos);
  }
  getch();
}
```

7) WAP to accept two string from user and concadinate this two string into a single string

```
#
#
void main ()
{ char ch1[100], ch2[100], ch3[100];
  int i, a ;
  Printf("Enter a String");
  Scanf("%S", ch1);
  Printf("Enter Second String");
  Scanf("%S", ch2);
{e ch3
```

(ch1)

0	1	2	3
R	A	M	\0

(ch2)

0	1	2	3	4
S	H	A	M	\0

ch3

0	1	2	3	4	5	6	7
R	A	M	S	H	A	M	\0

```
for(i= 0; ch1[i] ; i++)
{ ch3[i] = ch1[i] ;
} a=i+1;

for(i=0 ; ch2[i] ; i++)
{ ch3[i+a] = ch2[i] ;
}
Printf("%S", ch3);
getch();
}
```

```
#
#
# include <String.h>
main()
{ char ch1[100], ch2[100] ;
Printf("Enter a String");
Scanf("%S", ch1);
Printf("Enter 2nd String");
Scanf("%S", ch2);
Strcat(ch1, ch2);
Printf("%S", ch1);
getch();
}
```

8) WAP to accept a String lower case and Print it in upper case.

```
#
#
main()
{ char ch[100], ch1[100];
  int i;
  Printf("Enter a String in
          Lower case");
  Scanf("%S", ch1) ;
for(i=0; ch[i] ; i++)
{
   if (ch[i] >= 'a' && ch[i] <= 'z')
     { ch[i] = ch[i] - 32;
 } }
 ch1[i] = '\0';
 Printf("%S", ch1);
 getch();
}
```

```
#
#
# include <String.h>
void main()
{
   Char ch[100];
   Printf("Enter a String");
   Scanf("%S", ch);
  Printf("%S", Strupr(ch));
   getch();
 }.
```

9) WAP to accept a String from the user in upper case and convert in lower case.

```
#
#
main()
{
 char ch[100],ch1[100];
  int i;
Printf("Enter a String");
Scanf("%s", ch);
for(i=0; ch[i]; i++)
{   if( ch[i] >='A' && ch[i]<='Z')
      {   ch1[i] = ch[i]+32;
      }
 }
  ch1[i] = '\0';
 Printf("%s",ch1);
 getch();
}
```

```
#
#
#include <string.h>
main()
{ char ch[100];
 Printf("Enter a string");
 Scanf("%s", ch);
 Printf("%s", strlwr(ch));
    getch();
}
```

10) WAP to accept a string from the user and find how many vowel and consnonant present in this string.

```
#
#
main()
{ char ch[100];
  int i; vowel = 0, cons = 0;
Printf("Enter a name");
Scanf("%s", ch);

for (i=0; ch[i]; i++)
{  if( ch[i] >='A' && ch[i]<='Z')
       Vowel = vowel + 1;
   else
}      cons = cons + 1;
```

```
Printf("vowel = %d", vowel);
Printf("consonant = %d", cons);
getch();
}
```

11) WAP to accept two string from the user and check both String is equal or not.

```
#
#
main()
{ char ch[20], ch1[20];
  int i, len, flag
  Printf("Enter two String");
  Scanf("%s%s", ch, ch1); → for (i=0; ch1[i]; i++);
                            len = i;
  for(i=0 ; ch[i]; i++)
  {
     if(ch[i]== ch1[i] || ch[i]== ch1[i]-32 || ch[i]== ch1[i]+32)
       {
          ~~continue~~; flag = i+1;
       }
     else
       { Printf(" Both are different");
         break;
       }
  }
  if ( Flag == Len)
       Printf (" Both are Same ");
  getch();
}
```

(*) WAP to accept two String from the users check bot the String equal or not.

```
#
#
#include <string.h>
void main()
{
 char ch[10] = "Anupam";
 char ch1[10] = "Anupam";
 int n = strcmp(ch, ch1);
 if(n==0)
      Printf("Both are Same");
 else
      Printf("Both are different");
 getch();
}
```

Strcmp()

Syntax)

Strcmp(String1, String2);

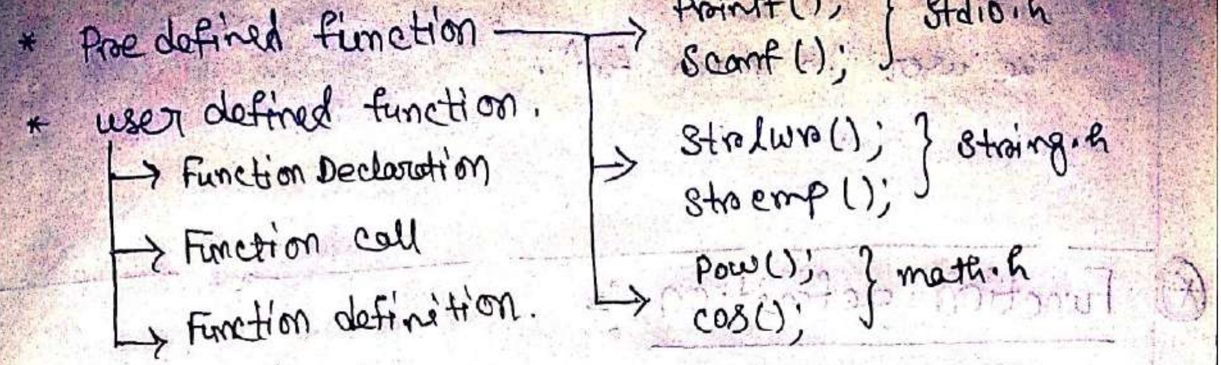
* If both are Same then return 0

(*) Assignment

```
#
#
main()
{ int i, J;
  char ch[10] = "TEMPLE";
  for(i=0; ch[i]; i++)
  {
    for(J=0; i>=J; J++)
     { Printf("%c", ch[J]);
     }
    Printf("\n");
  }
getch();
}
```

```
T
T E
T E M
T E M P
T E M P L
T E M P L E
```

Function



* Predefined function

* user defined function.

→ Function Declaration

→ Function call

→ Function definition.

Printf(); Scanf(); } Stdio.h

Strlwr(); Strcmp(); } String.h

Pow(); cos(); } math.h

(*) Predefined function > These are the fun^n which already have a definition in header files (.h).

(*) userdefined function > Here logic will be included by user.

(*) Function > A function is a group of statements that together perform a task, every C Programs has atleast one function (main() function)

i) Function Declaration :→ * above the main function

Syntan :- Returetype function_name (Datatype, Datatype);

a) In case of without returntype we write void.

b) In " " with " we write Datatype of output.

c) In case of without argumant simply put empty.

d) In case of with argument we have to decide the datatype input.

ii) Function Call :→ It should inside the main function

Syntan :- output variable = function-name (List of Parameter);

a) In case of without Rt, no need to take any output variable inside the main function.

b) In case of with Rt, we have to write output variable

c) In case of without argument no need to take any input variable inside main fun^n.

d) Incase of with argument we have to declar the input variable and take the variable from the user in main fun^n.

⊛ Function definition :→

* Here we are writing logic of the function.
* It should be out of the main function.

Syntax Returntype functionname (list of Parameter)

```
{
    Statement
    return;
}
```

* Incase of without Argument You have to declar the input variable inside the definition Path, take the input from the user and do the operation.

* In case of with argument we have to take the formal argument at the time of fun^n definition but the variable name should be different, do the operation to the require operation.

⊛ Incase of without Rt, Simply we can declear the o/p variable then do the operation asign the value to the o/p variable.

* Incase of with Rt, we may or mayn't be declear the o/p variable do the requir operation and assign the value to the variable atleast return the o/p variable.

(*) Benefits of function :→

1) Modularization :→ spliting of a bigger task into sarvel smaller task.

2) Easy to read.

3) Easy to debug (Testing the blocks easily)

4) Easy to Modify

5) Avoids rewriting of same code over and over.

6) Better memory utillization.

(*) Addition of two number using function

```
# include <stdio.h>
# include <conio.h>
                → int add (int, int) ;   → Function Declaration
void main ()
{
  int a, b, sum ;
  Printf ("Enter two no") ;
  Scanf ("%d %d", & a, & b) ;  ___ Actual Parameter
 sum = add (a, b) ;   ————→ Function call
 Printf (" The sum is %d", sum) ;

}                 ┌── Formal Parameter ,
int add (int x, int y) ;  ————→ Function Definition ,
{
  int R ;
  R = x+y ;
  return R ;
}.
```

(*) Ways to define a function

i) without Returntype without Argument

ii) without Returntype with Argument

iii) with Returntype without Argument

iv) with Returntype with Argument.

i) without Returntype without Argument

```
#
# include <conio.h>
void add ();
void main()
{

   add ();
   getch();
}.
```

```
void add ()
{
  int a, b, Sum;
  Printf ("Enter two no");
  Scanf("%d %d", &a, &b).
  Printf("Sum = %d", a+b);
}.
```

ii) without Returntype with Argument.

```
#
# include <conio.h>
void add (int, int);
void main ()
{
  int a, b;
  clrscr();
 Printf ("Enter two no");
 Scanf ("%d %d", &a, &b);
 add (a,b);
 getch();
}
```

```
void add (int x, int y)
{
   int c;
   c = x+y;
  Printf (" Sum = %d ", c);
}
```

iii) With Returntype without Argument

```
#include <Stdio.h>
#include <conio.h>
int add ();
void main ()
{
  int Sum;
Sum=add();
  Printf(" Sum= %d ",Sum);
  getch();
}
```

```
int add ( )
{
  int a,b,c;
  Printf("Enter two no");
  Scanf("%d %d",&a,&b);
  C=a+b;
  return C;
}
```

iv) With Returntype with~~out~~ Argument

```
#
# include <conio.h>
int add (int, int);
void main ()
{
  int a, b, S;
  clrscr();
  Printf("Enter two no");
  Scanf("%d %d", &a,&b);
  S= add (a,b);
  Printf("Sum = %d", S);
  getch();
}
```

```
int add (int x, int Y)
{
  int R;
  R= x+Y;
  return R;
}.
```

⊛ WAP to accept 10 diff. number and Print it using function.

```
#
#

void num ();
void main ()
{
    num ();
    getch();
}.
```

```
void num ()
{
  int a[10], i
  Printf("Enter 10 no");
  ~~Scanf("%d" &~~
  for(i=0; i<10; i++)
     Scanf("%d", &a[i]);
  Printf("Enter 10 no are ");
  for(i=0; i<=9; i++)
     Printf("%d\n", a[i]);
}.
```

⊛ Recursion

Recursion is a very Powerful technique to write a complicated Program in easy way.

'c' language Supports the technique of recursion which makes it move Powerful and unique.

The function which call itself (in function body) again and again is known as resursion function.

Syntax

```
        func();
        void main ()
        {
          func();
        }

      func()
      {
        func();
      }.
```

Ex. Program

Factorial Program using recursive function

```
#
#
int fact (int);
Void main ()
{ int n, R;
Printf ("Enter a no");
Scanf ("%d", &n);
R = fact(n);
Printf ("Factorial = %d", R);
 getch();
}.
```

```
int fact (int f)
{   if (f == 1)
        return 1;
    else
        return (f * fact(f-1));
}
```

(*) Fabonasi Series using Recursion

```
#
#
int fab (int);
void main()
{ int n, F;
 Printf ("Enter a no");
 Scanf ("%d", &n);
 F = fab (n);
 Printf (" Fabonasi is %d", F);
 getch();
}
```

```
int fab (int n)
{  if (n==0)
      return 0;
   else if (n==1)
      return 1;
   else
      return (fab (n-1) + fab(n-2));
}
```

$0 \quad \frac{n-3}{1} \quad \frac{n-2}{1} \quad \frac{n-1}{2} \quad \frac{n}{3} \quad 5 \quad 8$

⊛ **Pointer** :→ It is a variable which is stored address of another variable.

datatype *variable → Pointer variable.

* → valul

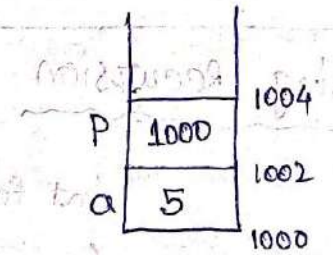
& → address.

Integer variable stores integer value.
Integer pointer stores address of Integer variable.

If any variable name consists of * operator as a Prefix it means pointer variable.

So If I want to store the address of another variable I want to use Pointer variable.

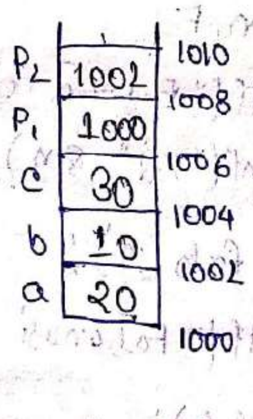
```
int a;
int *P;
a=5;
P=&a;
```



a → 7
&a → 1000
P = 1000
&P = 1002
*P = 5.

whatever datatypes memory will allocate only 2 bytes for pointer variable.

```
int a, b, c;
a = 20;
b = 10;
int *P1, * P2.
P1 = &a;
P2 = &b;
```



$c = *P_1 + *P_2$
$= 20 + 10$
$= 30$

$*P_1$ → value of address,
&a → address of a.

Address	values
$P_2 - P_1$ ✓	$*P_1 - *P_2$ ✓
$P_1{++}$ ✓	$*P_1 + *P_2$ ✓
$P_1{--}$ ✓	$*P_1 * *P_2$ ✓
$P_1 * P_2$ X	$*P_1{++}$ ✓
$P_1 + P_2$ X	$*P_2{--}$ ✓
	$*P_1 / *P_2$ X.

$P_1{++} \rightarrow P_1+1 \longrightarrow 1000+1$

$\downarrow$

1002. [for integer taks two byte].

Here $*P_1/*P_2$ not Possible because

/* ---- */ this is commmand.

⊛ wAP to accept two no from the user and find the addition, multiplication, Devision, Substraction.

```
#
#
void main()
{
   int a,b;
   int *P, *q;
  Printf("Enter two no");
  Scanf("%d %d", &a, &b);

  P = &a;
  q = &b;
Printf(" The Sum = %d\n", *P+ *q);
Printf(" The Substaraction = %d \n", *P - *q);
Printf(" The multi = %d\n", *P * *q);
```

```
Printf(" The Devision is
        = %d\n", (*P)/(*q));
getch();
}
```

(*) WAP to accept a radious of a circle and find the area and perimeter using Pointer.

```
#
#
void main()
{
 float r;
 float a, Per;
 float *p;
Printf("Enter the radious");
Scanf("%f", &r);
 P = &r;
 a = 3.14 *(*P)*(*P);
 Per = 2* 3.14 *(*P);
 Printf("area = %f", a);
 Printf("perimeter =%f", Per)
 getch();
}
```

Pointer to Pointer (or) chain Pointer

It is a Pointer variable which is stores address of another pointer variable.

```
int a = 5, b = 10;
int *P1, *P2;
P1 = &a;
P2 = &b;
int **P3, **P4;
P3 = &P1;
P4 = &P2;
```

P4	1006	1010
P3	1004	1008
P2	1002	1006
P1	1000	1004
b	10	1002
a	5	1000

$*P_1 \rightarrow 5$

$*P_2 \rightarrow 10$

$\&a \rightarrow 1000$

$\&b \rightarrow 1002$

$**P_3 \rightarrow 5$

$**P_4 \rightarrow 10$

$P_1 \rightarrow 1000$

$P_2 \rightarrow 1002.$

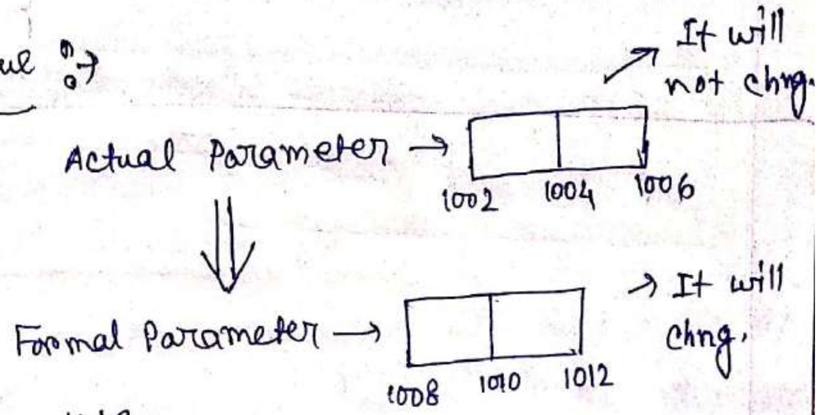
Parameter Passing Methods

⊛ Parameter :→ Inputs given by the user to the Particula function.

Actual Parameter → which are written in the function call.

Formal Parameter → which are written in the function definition.

⊛ Call by value :→



Actual Parameter will be copied to Formal Parameter.

▩ Call by value (Swap)

```
#
#
void Swap(int, int);
void main()
{
  int a = 10, b = 20;
  Swap(a,b);
  Printf("a= %d, b= %d",a, b);
} getch();
```

```
void Swap(int x, int y)
{
  int temp;
  temp = x;
  x = y;
  y = temp;
  Printf("x = %d, y = %d", x, y);
}.
```

O/P

x = 20, y = 10,

a = 20, b = 10

⊛ Call by ~~address~~ Reference

#

```
void Swap (Int *, int *);
void main ()
{
   Int a = 10, b = 20;
   Swap (&a, &b);
   getch();
}
```

```
void Swap (int *x, int *y)
{
   int temp;
   temp = *x;
   *x = *y;
   *y = temp;
  Printf("x = %d, y=%d", x,y);
}
```

Structure

If i want to store multiple elements with different data-types under the same name, we have to use structure

It is an user defined datatype. It is a collection of hetrogenious / desimilar type of data.

Syntax :→
```
        struct Structure_name
        {
          Datatype variable;
          - - - - -
          Datatype variable;
        };
```

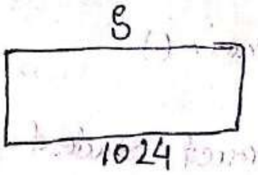
⊛ characteristic of Structure :→

i> whenever we create a structure at that time by using struct keyword, we can create the structure.

(ii) The size of the object of structure is equal to the total size of the main data declear in the structure

Ex
```
Struct Student
{ int roll;
  char ch[100];
  float fee;
};
```

Struct Student S;



→ size is 126 byte-

iii> we can change the data more than one time in the structure.

iv> when we can call the structure member through the object by using dot operator.

* Define Structure

* Declare Structure

* Initiazing Structure variable a,

* Accessing Structure.

Structure	union
i) we can use 'Struct' keyword to make the Structure.	i) we can use 'union' keywor to create the union.
ii) The size of Structure is the total size of the declar variable.	ii) The size of union is the maximum size of union member data.
iii) In case of structure we can change data any time.	iii) In case of union we can change data at once.

Ⓐ write a Program to accept name, branch, Roll no and fee and Print it.

```
#
#
Struct Student
{
 int roll;
 char name[100], br[100];
 float fee;
};
void main()
{
  Struct Student s;
  Printf("Enter the details");
 Scanf("%s%s %d %f", s.name, s.br, & s.roll, & s.fee);
 Printf(" The details are ");
 Printf("%s %s %d %f", s.name, s.br, s.roll, s.fee);
 getch();
}
```

(*) WAP to accept name, Roll, age, branch of five Student and Print it.

```
# include <stdio.h>
# include <conio.h>
Struct student
{ char name[100], br[10];
   int Roll; age;
};
void main()
{
    Struct student P[5];
    int i;
  for (i=0; i<5; i++)
  { Printf("Enter the details ");
     Scanf("%s %s %d %d", P[i].name, P[i].br, & P[i].Roll,
                                                & P[i].age);
  }
for (i=0; i<5; i++)
{ Printf("The details of %d students", i+1);
   Printf("%s %s %d %d", P[i].name, P[i].br, P[i].Roll, P[i].age);
 }
 getch();
}.
```

union

```
union union-name
{ Datatype member 1;
    - - - -
};
```

Ex. Program for union

```
union R;
{ int a;
  float b;
};
```

```
void main()
{ union R. var;
  var. a = 10;
  Printf(" b = %d", var. b.);
  var. b = 20;
  Printf(" a = %d", var. a);
  getch();
}
```

Jumping Statement

1) goto :→ The goto statements is the type of Jumping statement which transfer the control to sum other Part of Program. This Program Passes control any where in the Program that is control is transfered to any Part without its any condition.

This statement is written as goto label; where label is position where control is to be transperd.

Syntax :→

```
void main ()
{ - - - -;
   label:
   - - -;
   goto label;

}
```

Reverse of the number using Jumping Statement

```
#
#
void main ()
{
 int n, rem, rev = 0;
 Printf ("Enter a no");
 Scanf ("%d", &n);
Start ;
      rem = n%10 ;
      rev = rev * 10 + rem;
      n = n/10 ;
 if (n>0)
 goto Start ;
 Printf (" Reverse = %d", rev);
 getch ();
}
```

2) Break :→ Break is the type of Jumping Statement which allows the statement terminate to loop. The break skips from the loop. The control then automatically go the first Statement after the loop or block. The break will be assioted with all conditional Statement This will be used in all the huge statement like for while, do-while, Switch statement. It is used to handle the exception like infinte loop condition.

It is used in Switch Statement.

(*) WAP to Print 1-10 using break Statement,

```
#
#
void main()
{
   int i;
 for (i=1; i>0; i++)
 { Printf("%d", i);
     if(i>10)
     break;
 }
 getch();
}
```

3) Continue :-> The continue statement does not require any condition. The continue statement is used for continueing the next iduration. The loop does not when continue statement encounter. That is exicutable statement which are Present after continue statement can't be exicuted.

Syntax

```
while (condition)
{ - - - - ;
  if (condition)
  continue;
  - - - ;
}
```

4) Return :-> It's a type of Jumping statement which is used to return the value to the calling funn from a user defined call function. It is used to return some value or expression from a user defined type to caller functuion.

Syntax

```
returntype func name ('Parameters )
{ - - ;
  - - ;
  return (value / expression);
}
```

ex :->

```
#
#
int add (int, int);
void main()
{ int a= 20, b= 30;
  c= add (a,b);
```

```
  Pointf("sum= %d", c);
  getch();
}
int add ( int x, int y )
{
  return(x+y);
}
```