

- ★  $A = 1 + 2 + 3 + 4$
- Add 1 and 2
  - Assign the sum of 1 and 2 to K.
  - Add K and 3
  - Assign the sum of K and 3 to l.
  - Add l and 4
  - Assign the sum of l and 4 to M.
  - Now assign it to A.
  - Display it.

★  $A = 10 - 2 - 1$

- Subtract 2 from 10
- Assign the subtraction of 10 from 2 to 8.
- Subtract 1 from 8.
- Assign the subtraction of 1 from 8 to 7
- Now assign it to A.
- Display it.

★  $C = d + e + f$

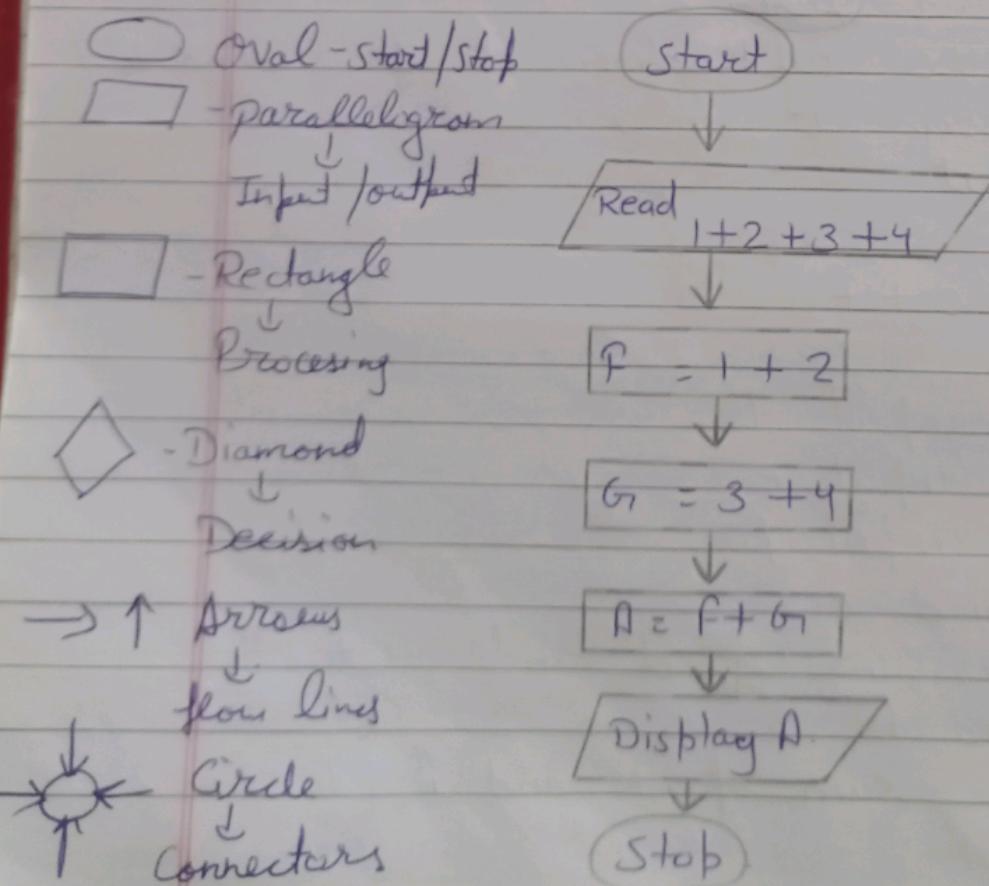
- Take the input d.
- Take the input e.
- Take the input f.
- Add d and e.
- Assign the sum of d and e to k.
- Add k and f.
- Assign the sum of k and f to M.

- (viii) Now assign it to C.  
 (ix) Display H.

\*  $a = 10, b = 20$  we have to find the value of  $a = 20, b = 10$ .

- (i) Add a and b ( $10 + 20$ )
- (ii) Assign the sum of 10 and 20 to 30(K)
- (iii) Subtract a from K ( $30 - 10$ )
- (iv) Assign the subtraction 30 from 10 to 20.
- (v) Subtract b from K ( $30 - 20$ )
- (vi) Assign the subtraction 30 from 20 to 10
- (vii) Now assign it to a and b.
- (viii) Display it.

\*  $A = 1 + 2 + 3 + 4$



\* % mode -

if  $(7 \% 2 == 0)$   
 $1 == 0$   
(odd)

if  $(100 \% 2 == 0)$   
 $0 == 0$   
(Even)

a) 50 %

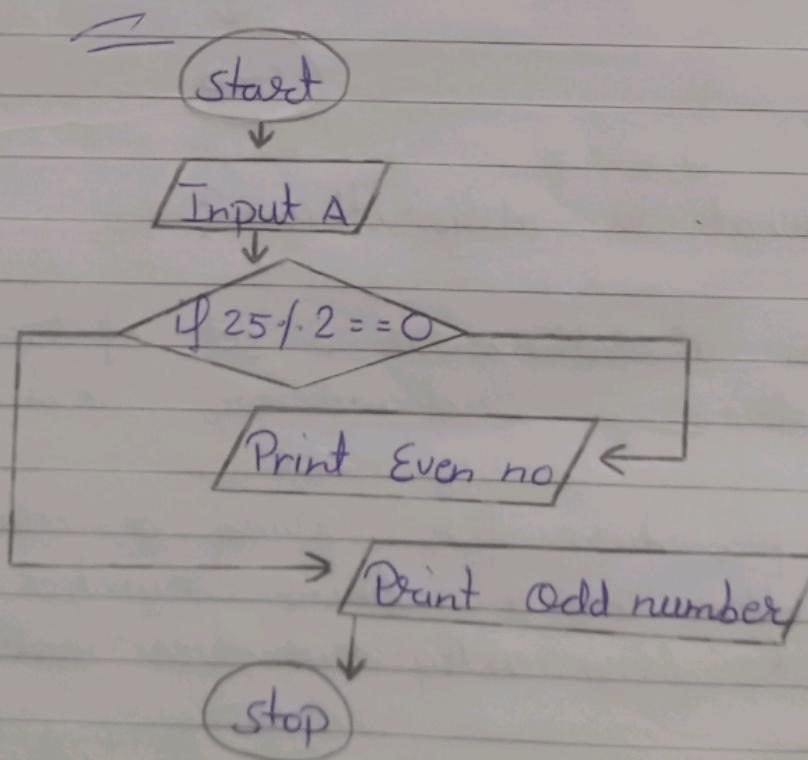
if  $(50 \% 2 == 0)$   
 $0 == 0$

Yes, this is the even number.

(ii) 25 %.

if  $(25 \% 2 == 0)$   
 $1 == 0$

No, this is not a even no. this is odd number.



SE

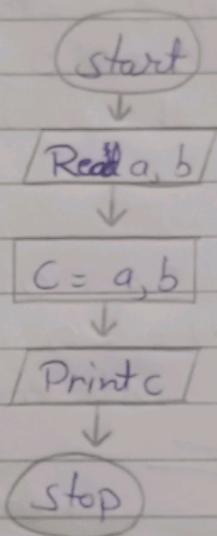
formula is always in  Box.  
e.g.  $C = a + b$ .

★ Example :-

Algorithm :-

- (i) Start
- (ii) Read a and b
- (iii)  $C = a + b$
- (iv) Print at display c.
- (v) Stop

Flow chart -



★ What is Algorithm?

An algorithm is a set of instructions for solving a problem or accomplishing a task

OR

A procedure for solving a mathematical problem in a finite number of steps that frequently involves recursive operations

=> # include <stdio.h>

B predefined = design by computer  
userdefined = .. .. user

We have to  
fill just b/w  
the curly  
brackets

void main ( ) → defined

{

( ) this is defined

f = function

ch = characteristics

( "NewLine" );

backslash \ for go to  
next line.

String (Contains no. like 4)

SE

```
printf ("HELLO");  
getch();  
}
```

semicolon means  
line ends

⇒ # include <stdio.h>

void main

{

int a = 10;

float=10.15

printf ("%d", a);

getch();

}

int = integer

string = " "

character = ''

d = integer type value or alpha

%d = from this we know

d as integer

float = 10.15

# include <stdio.h>

void main()

{

int a = 10;

int b = 20;

printf ("%d %d", a, b);

getch();

}

printf ("123+2");

printf ("\n");

printf ("%d", 123+2);

⇒ Three variables of integers type a, b, c . a+b is their sum and store them in c and print a, b, c.

# include <stdio.h>

void main()

{

int a = 10;

int b = 20;

int c; c = a + b;

printf ("%d %d %d", a, b, c);

getch();

}

$\Rightarrow a = 10, b = 2$ . we have to find c as add and even.

```
# include <stdio.h>
void main ()
{
    int a = 10;
    int b = 2;
    int c;
    c = a % b;
    printf ("%d %d %d", a, b, c);
    getch ();
}
```

$\Rightarrow a = 2, b = 2, C = a^b$

```
# include <stdio.h>
void main ()
{
    int a = 2;
    int b = 2;
    int c;
    c = a * a;
    printf ("%d", c);
    getch ();
}
```

$\left[ \begin{array}{l} \text{int } a = 2; \\ \text{int } b = 4; \\ \text{int } c; \\ c = a * a * a * a \\ \text{if } b = 5; \\ c = a * a * a * a * a \end{array} \right]$

$\Rightarrow f = 1 + 2 + 3 + 4 + 5$

```
# include <stdio.h>
void main ()
{
```

```
    int a = 1;
```

```

int b = 2;
int c = 3;
int d = 4;
int e = 5;
int f;
f = a + b + c + d + e;
printf ("%d", f);
getch ();
}

```

$$\Rightarrow \frac{PXRXT}{100}, P = 1000, R = 20\%, T = 2 \text{ Year}, SI = ?$$

```

#include <stdio.h>
void main ()
{
    int P = 1000;
    int R = 20;
    int T = 2;
    int SI;
    SI = (PXRXT) / 100;
    printf ("%d", SI);
    getch ();
}

```

$$\Rightarrow a = 10, b = 20 \quad (a = 20, b = 10)$$

```

#include <stdio.h>
void main ()
{
    int a = 10;
    int b = 20;
}

```

```

int c;
c = a;
a = b;
b = c;
printf ("%d %d", a, b);
getch ();
}

```

⇒ a = 10, b = 20 (a = 20, b = 10)  
without assigning another variable.

```

#include <stdio.h>
void main ()
{
    int a = 10;
    int b = 20;
    a = a + b; b = a - b; a = a - b;
    printf ("%d %d", a, b);
    getch ();
}

```

⇒ a = 10, b = 20, c = 30, d = 40

- I a = a + b;
- II a = a + b + c + d;
- III d = d + a;
- IV b = a + c;
- V c = c + c;
- VI c = c + d;

VII  $a = a + 10$

$$a = a + b \quad (10 + 20) = 30$$

$$a = a + b + c + d \quad (30 + 20 + 30 + 40) = 120$$

$$d = d + a \quad (40 + 120) = 160$$

$$b = a + c \quad (120 + 30) = 150$$

$$c = c + c \quad (30 + 30) = 60$$

$$c = c + d \quad (60 + 160) = 220$$

$$a = a + 10 \quad (120 + 10) = 130$$

a	10, 30, 120, 130	1
b	20, 150	2
c	30, 60, 220	3
		4
		5
		6
d	40, 160	7

# include <stdio.h>

void main()

{

int a = 10; int b = 20; int c = 30; int d = 40;

a = a + b; a = a + b + c + d; d = d + a; ~~d = a + c;~~

C = C + C; C = C + d; a = a + 10;

printf ("%d %d %d %d", a, b, c, d);

getch();

}

=> Int a, b, c

Constant int d = 20;

a = 10; b = 30; c = 50;

a = a + b;

a | 10, 40

b = a + d;

b | 30, 60

d = d + c;

c | 50

d | 20

a = a + b;  $10 + 30 = 40$

b = a + d;  $40 + 20 = 60$

d = d + c; 20 + 50, error becoz d is constant  
its value can not change.

⇒ Rules for Identifier :-

- ① It can't start from special symbol except underscore (-) & \$ (dollar)
  - ② It can't start from digit.
  - ③ It can only start from alphabet.
- ⇒ If we want to write  $a = 1.2$  in program, we have to write float  $a = 1.2$  not integer. When we print we can write  $\%f$  not  $\%d$ .
- If we want to store  $a = 'b'$  then we write char  $a = 'b'$ . char (character). When we print it we write printf "%c".
  - Only one alphabet is called character and more than one is called string. like :- "HELLO".

⇒  $a = 10.2, b = 3.6$ , write program . Add, Mul, div. and print all.

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
float a = 10.2;
```

```
float b = 3.6;
```

```
float c;
```

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

```
float d;
```

```
d = a*b;
```

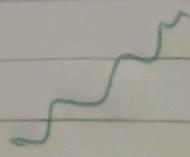
```
float e;  
c = a/b  
printf ("%f %f %f", c, d, e);  
getch();  
}
```

⇒ store character b in 'a'. b = 'a'

```
# include <stdio.h>  
void main()  
{  
    char a = 'b';  
    printf ("%c", a);  
}
```

Q Print 7's table

```
# include <stdio.h>  
void main()  
{  
    int i = 1;  
    int a = 7;  
    printf ("%d", a);  
    i = i + 1;  
    int b; b = a * i;  
    printf ("%d", b);  
    i = i + 1;  
    int c; c = a * i;  
    printf ("%d", c);  
    i = i + 1;  
    int d; d = a * i;  
    printf ("%d", d);  
}
```



⇒ Take one values and find out the factorial for given no. i=4.

```
# include <stdio.h>
void main()
{
    int i = 4;
    int a;
    a = i * (i - 1) * (i - 2) * (i - 3);
    printf ("%d", a);
}
```

⇒ Print 'a' 'b'.

```
# include <stdio.h>
void main()
{
    char x = 'a';
    char y = 'b';
    printf ("%c %c", x, y);
    getch();
}
```

⇒ Difference b/w Keywords and Identifier.

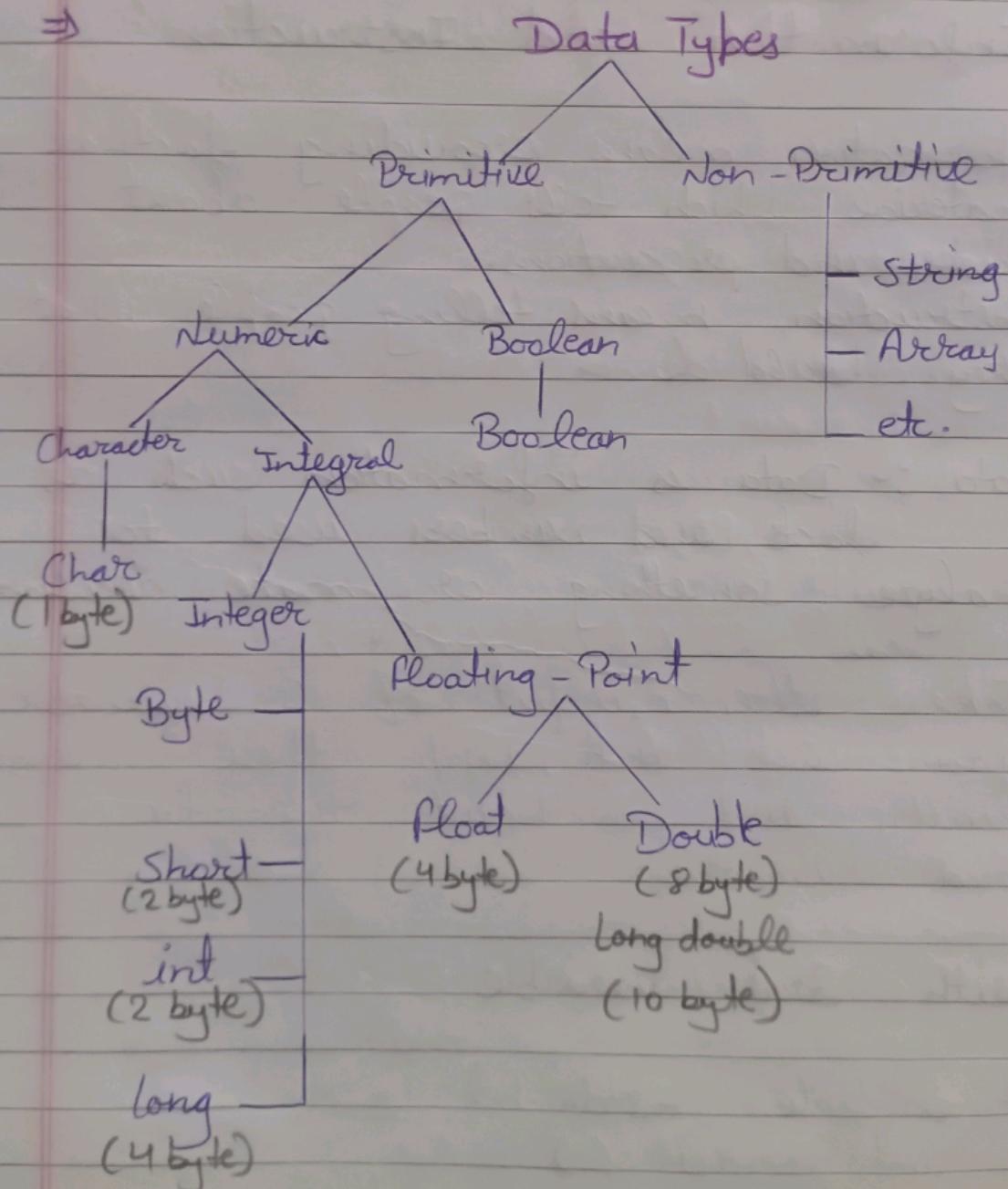
The basic diff. b/w the two is that keywords are the reserve words which are predefined and have special meaning in the language, whereas an

identifier is a unique name assigned to a variable, function, class etc.

⇒ Difference b/w variable and constant.

A constant does not change its value over time. A variable, on the other hand, changes its value dependent on the equation.

⇒



Basic data types : int, char, float, double

Derived data types : array, pointer, structure,  
union

Enumeration data types : enum

Void data types : void

⇒ Information and Instruction ?

Information means providing factual material which tells people about risks and precautions.

Instruction means telling people what they should do.

Data :- Data is information such as fact and numbers used to analyze something or make decisions.

→ Take the input of 2 variable from user and swap these variable with and without the help of 3rd variable.

// with 3rd variable

```
# include <stdio.h>
int main ()
```

```
{  
    int a, b, c;  
    printf ("enter the value of a");  
    scanf ("%d", &a);  
  
    printf ("enter the value of b");  
    scanf ("%d", &b);  
  
    c = a;  
    a = b;  
    b = c;  
    printf ("%d %d", a, b);  
    return 0;  
}
```

Without 3rd variable

```
# include <stdio.h>  
int main ()  
{  
    int a, b;  
    printf ("enter the value of a");  
    scanf ("%d", &a);  
  
    printf ("enter the value of b");  
    scanf ("%d", &b);  
    a = a + b;  
    b = a - b;  
    printf ("%d", b);  
    a = a - b;  
    printf ("%d", a);  
}
```

→ Calculate Distance Speed × Time.

```
# include <stdio.h>
int main()
{
    int speed, time, distance;
    printf ("enter the value of speed and time")
    scanf ("%d %d", &speed, &time);
    distance = speed * time;
    printf ("%d", distance);
    return 0;
}
```

→ Area of circle

```
# include <stdio.h>
int main()
{
    PI = 3.14
    float a, PI, r;
    printf ("enter value of PI and r");
    scanf ("%f %f", &PI, &r);
    a = PI * r * r;
    printf ("%f", a);
}
```

→ Factorial

```
# include <stdio.h>
int main()
{
    int a, fact;
```

```

printf (" Enter the value fact")
scanf ("%d", &fact);
fact = a * (a-1) * (a-2) * (a-3) * (a-4);
printf ("%d", fact);
}

```

Bullian value - either 1 or 0.

### Operators

= = =

< = a+b

> = a-b

a++; [a = a+1]

a--; [a = a-1]

a = 10

(a % 2 == 0)

(0 == 0)  
L == R

// - Comment  
Didn't consider line.

/\*

printf ("\n enter the no")  
new line

It tab the space.

Void main

{

if (Conditional Statement)

{

}

}

Brackets represent  
set of statement

x++ => Post increment
++x = Preincrement

⇒ Void main

```
{  
    int a = 10;  
    if (a == 10)  
    {  
        printf ("BCD-B");  
        printf ("In if block");  
    }  
    printf ("Out of if block");  
}
```

⇒ int flag = 0;

```
int n;  
if (n % 2 == 0)  
{
```

```
    printf ("even number");  
    flag++;
```

```
}  
if (flag == 0)
```

```
    printf ("odd number");  
}
```

n = 5;

n - 1 = 5 - 1 = 4

n++ = n = n + 1 = 5 + 1 = 6

Q) find out the maximum  
number among 2 variable.

```
#include <stdio.h>
int main()
{
    int a;
    int b;
    printf ("enter the value of a and b");
    scanf ("%d %d", &a, &b);
    if (a>b)
    {
        printf ("%d", a);
    }
    if (b>a)
    {
        printf ("%d", b);
    }
}
```

OR

```
int a,b; flag=0;
if (a>b)
{
    printf ("a is greater");
    flag++;
}
if (flag == 0)
{
    printf ("b is greater");
}
```

→ find out the greatest among 3 variable.

```
# include <stdio.h>
void main ()
{
    int a, b, c;
    printf ("Enter the value of a, b, c");
    scanf ("%d%d%d", &a, &b, &c);
    if ((a>b)&&(a>c))
    {
        printf ("%d", a);
    }
    if ((a<b)&&(c<b))
    {
        printf ("%d", b);
    }
    if ((c>d)&&(c>b));
    {
        printf ("%d", c);
    }
}
```

\* Print leap year

```
# include <stdio.h>
void main ()
{
    int y;
    printf ("Enter the Year");
```

```

    scanf ("%d", & Y);
    if (Y % 4 == 0)
    {
        printf ("enter leap year");
    }

```

## ⇒ Keywords (32)

- ① Keyword is a Predefined word
- ② written in lowercase letter only.
- ③ Meaning is pre-defined in the C-Compiler.
- ④ Combination of alphabetical characters.
- ⑤ Not Contains underscore character.
- ⑥ e.g. double, int, auto, char, break.

## Identifier

It is defined by user.

written in both upper and lower case.

Meaning not defined.

Combination of alphanumeric characters.

Contain underscore character

e.g. Test, count1, high-speed.

→ Data types :- Datatype specifies the size and type of values that can be stored in a variable.

Types →

- ① Primary → int (2 byte)  
char (1 byte) float (4 byte)  
double (8 byte) void (0 byte)
- ② Secondary → Array, Pointer, structure, Union

## ⇒ Constant

- ① Constant has its fixed value.
- ② Constant is type of value.
- ③ Dynamic initialization is not possible.
- ④ used in mathematical expression.
- ⑤ e.g. `int Const. x = 5;`

## Variable

- Variable doesn't have fixed value.
- Variable are symbol.
- Dynamic initialization is possible.
- used to represent letters.
- e.g. `int x;`

## ⇒ If structure

{ if (Condition)

-Statement

}

{ if (Condition 2)

— — —

}

⇒ Find out the given number is even or odd.

```
# include < stdio.h >
void main ()
{
```

```
int a;
printf ("Enter the value of a");
scanf ("%d", &a);
if (a % 2 == 0)
{
    printf ("Even Number");
}
if (a % 2 != 0)
{
    printf ("odd Number");
}
getch();
```

⇒ Enter a number and check  
Is it divisible by 7 or not.

```
#include <stdio.h>
void main()
{
    int a;
    printf ("Enter the value of a");
    scanf ("%d", &a);
    if (a % 7 == 0)
    {
        printf ("Divisible by 7");
    }
    if (a % 7 != 0)
    {
        printf ("Not Divisible by 7");
    }
}
```

&

```

int a, flag = 0;
printf ("Enter the value of a");
scanf ("%d", &a);
if (a % 7 == 0)
{
    printf ("Divisible by 7");
    flag = flag++;
}
if (flag == 0)
{
    printf ("Not Divisible by 7");
}
getch();
}

```

2) Even odd with flag.

```

int a, flag = 0;
printf ("Enter the value of a");
scanf ("%d", &a);
if (a % 2 == 0)
{
    printf ("a is even number");
    flag = flag++;
}
if (flag == 0)
{
    printf ("a is odd number");
}
getch();
}

```

⇒ Even odd with else.

```
# include <stdio.h>
void main()
{
    int n;
    printf (" enter the value of n");
    scanf ("%d", &n);
    if (n%2 == 0)
    {
        printf (" n is even number");
    }
    else
    {
        printf (" n is odd number");
    }
}
```

⇒ Which is greater

```
# include <stdio.h>
int main()
{
    int a, b, c;
    printf (" enter the value of a, b and c");
    scanf ("%d %d %d", &a, &b, &c);
    if ((a>b) && (a>c))
    {
        printf (" a is greater");
    }
}
```

```

else
{
    if ((b > a) && (b > c))
        printf ("b is greater");
    else
    {
        printf ("c is greater");
    }
    getch ();
}

```

⇒ Age

```

#include <stdio.h>
void main()
{
    int age;
    printf ("enter the value of age");
    scanf ("%d", &age);
    if (age >= 18)
    {
        printf (" eligible to vote");
    }
    else
    {
        printf (" Not eligible to vote");
    }
}

```

$\Rightarrow$  More than 80 are equal in A grade.  
" " " 60 " " is B grade.  
others C grade.

```
# include <stdio.h>
void main()
{
    int marks;
    printf (" Enter the value of marks ");
    scanf ("%d", &marks);
    if (marks >= 80)
    {
        printf (" A Grade ");
    }
    else
    {
        if (marks >= 60)
        {
            printf (" B Grade ");
        }
        else
        {
            printf (" C Grade ");
        }
    }
    getch();
}
```

⇒ Write a program to find the area and perimeter of rectangle.

```
#include <stdio.h>
void main()
{
    float length, breadth, area, perimeter;
    printf ("Enter the value of length and breadth");
    scanf ("%f %f", &length, &breadth);
    area = length * breadth;
    printf ("%f", area);
    perimeter = 2 * (length + breadth);
    printf ("%f", Perimeter);
}
```

⇒ Write a program to find simple interest and compound interest.

$$S.I. = (\text{Principle} \times \text{Rate} \times \text{Time}) / 100$$
$$C.I. = \text{Principle} (1 + \text{Rate})^{\text{Time}} - \text{Principle}$$

```
#include <stdio.h>
int main()
{
    float P, T, R, SI, CI;
    printf ("Enter the value of P, T, R");
    scanf ("%f %f %f", &P, &T, &R);
    SI = (P * R * T) / 100;
    printf ("%f", SI);
    CI = ((1 + R) ^ T - 1) * pow((1 + R / 100), T); // AP
    printf ("%f", CI);
```

⇒ write a program to convert distance from miles to km

```
# include <stdio.h>
int main()
{
    float miles, kilometre;
    printf ("enter the value of miles");
    scanf ("%f", &miles);
    Kilometre = miles * 1.61;
    printf ("%f", kilometre);
```

⇒ write a program to print the size of all the data types supported by C.

```
# include <stdio.h>
int main()
{
    printf ("size of char : %lu byte\n", size of (char));
    printf ("size of int : %lu byte\n", size of (int));
    printf ("size of long : %lu byte\n", size of (long));
    printf ("size of float : %lu byte\n", size of (float));
    printf ("size of double : %lu byte\n", size of (double));
```

# if ( condition )

}

else if ( Condition )

{

}

else if ( Condition )

{

}

else if \_\_\_\_\_

{

}

⇒ if { $(n \% 2 == 0)$

} printf (" div. by 2 ");

else if { $(n \% 3 == 0)$

} printf (" div. by 3 ");

else if { $(n \% 4 == 0)$

} printf (" div by 4 ");

else

else if { $(n \% 5 == 0)$

} printf (" div. by 5 ");

} printf ("Not divi ");

```
⇒ #include <stdio.h>
int main ()
{
    int n = 31;
    if (31 % 2 == 0)
    {
        printf ("div by 2");
    }
    else if (31 % 3 == 0)
    {
        printf ("div by 3");
    }
    else if (31 % 4 == 0)
    {
        printf ("div by 4");
    }
    else
    {
        printf ("Not div.");
    }
}
```

```
# #include <stdio.h>
int main ()
{
    char c;
    printf ("enter the value of c");
    scanf ("%c", &c);
    if (c == 'm')
    {
```

```
printf ("Yes it is Monday");  
} else if (c == 'T')  
{ printf ("Yes it is Tuesday");  
} else if (c == 'W')  
{ printf ("Yes it is Wednesday");  
} else if (c == 'T')  
{ printf ("Yes it is Thursday");  
} else if (c == 'F')  
{ printf ("Yes it is Friday");  
} else if (c == 'S')  
{ printf ("Yes it is Saturday");  
} else if (c == 'S')  
{ printf ("Yes it is Sunday");  
}
```

# Write a program to print positive, negative or zero as per the value of n which is entered by user.

```
# include <stdio.h>
int main()
{
    int n;
    printf ("enter the value of n");
    scanf ("%d", &n);
    if (n == 0)
    {
        printf ("zero");
    }
    else if (n < 0)
    {
        printf ("Negative");
    }
    else (n > 0)
    {
        printf ("Positive");
    }
}
```

# C program to print weekend based on given number.

```
# include <stdio.h>
int main()
{
    int day;
    printf ("Enter day number");
    scanf ("%d", &day);
    if (day == 1)
```

```
{  
    printf ("SUNDAY");  
}  
else if (day == 2)  
{  
    printf ("MONDAY");  
}  
else if (day == 3)  
{  
    printf ("TUESDAY");  
}  
else if (day == 4)  
{  
    printf ("WEDNESDAY");  
}  
else if (day == 5)  
{  
    printf ("THURSDAY");  
}  
else if (day == 6)  
{  
    printf ("FRIDAY");  
}  
else if (day == 7)  
{  
    printf ("SATURDAY");  
}  
else  
{  
    printf ("INVALID DAY");  
}
```

```
# # include <stdio.h>
void main()
{
    int x = 22;
    if (x == 10)
        printf ("TRUE");
    else
        printf ("FAWE");
}
```

Output : TRUE

```
# # include <stdio.h>
void main()
{
    if (-100 && 100) || (20 && -20)
        printf ("%s", " condition is true");
    else
        printf ("%s", " condition is false");
}
```

OUTPUT : Condition is true.

```
# # include <stdio.h>
int main ()
{
    char val = 1;
    if (val-- == 0)
        printf ("true");
    else
        printf ("false");
}
```

Output : False

```

# # include <stdio.h>
int main()
{
    char a = 20;
    char c, b;
    c = a + ++10;           (c = 20@1 + 10; = 30)
    b = ++a + 10;          (b = 1++@ + 10; = 32)
    printf ("%d %d", c, b); 21
}

```

## # Switch case statement

```

int a;
switch (a)
{
    case 2:
        printf ("BCA I");
    case 3:
        printf ("BCA II");
    default:
        printf ("xyz");
}

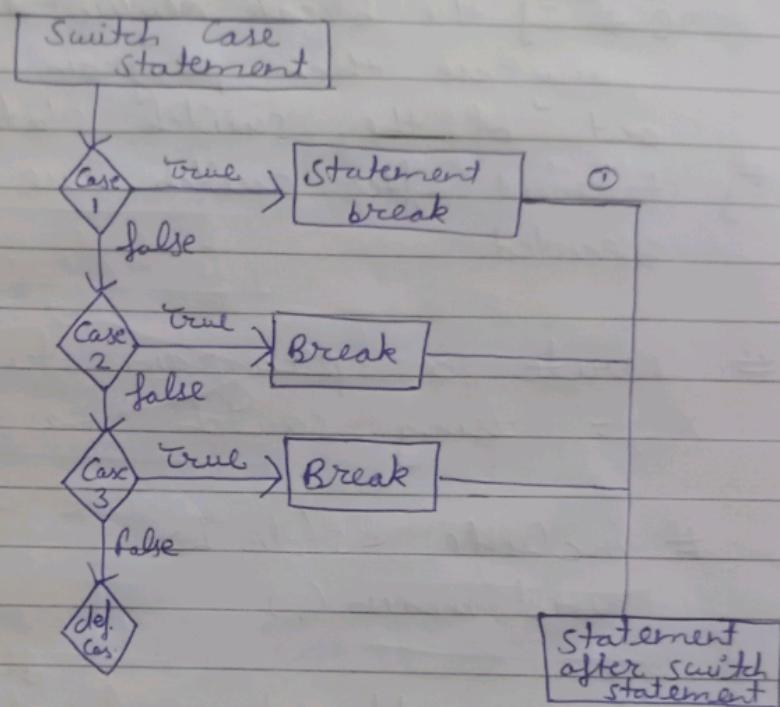
```

## # Rules :-

- ① Switch expression must be integer or character.
- ② Break is not compulsory.
- ③ Case must be int. or char.

# Switch case statement follows a selection control mechanism and allows a value to change control of execution.

# Flow chart



# Break Keyword :- This keyword is used to stop the execution inside the switch block. It helps to terminate the switch block and break out of it.

# How it works :- 1) The switch variable is evaluated.

- 2) The evaluated is matched against all the present cases.
- 3) If match occurs the associated code will execute.

3B  $\rightarrow$  if no match is occur then default case will execute.

4) 4A  $\rightarrow$  If the break keyword not present the case after matching case is executed.

4B  $\rightarrow$  If the break keyword is present in any case then program control break out of the switch statement.

5) statement after switch case statements are executed.

# write a program to perform +, -, x, ÷ using switch case statement.

```
# include <stdio.h>
void main()
{
    float a, b, add, sub, mul, div;
    printf ("Enter the value of a,b");
    scanf ("%d %d", &a, &b);
    char op;
    switch (op)
    Case '+':
        printf ("Sum %d", a+b);
    Case '-':
        printf ("Sub %d", a-b);
    Case '*':
        printf ("mul %d", a*b);
    Case '/':
        printf ("div %d", a/b);
    default:
        printf ("operator not exist");
}
```

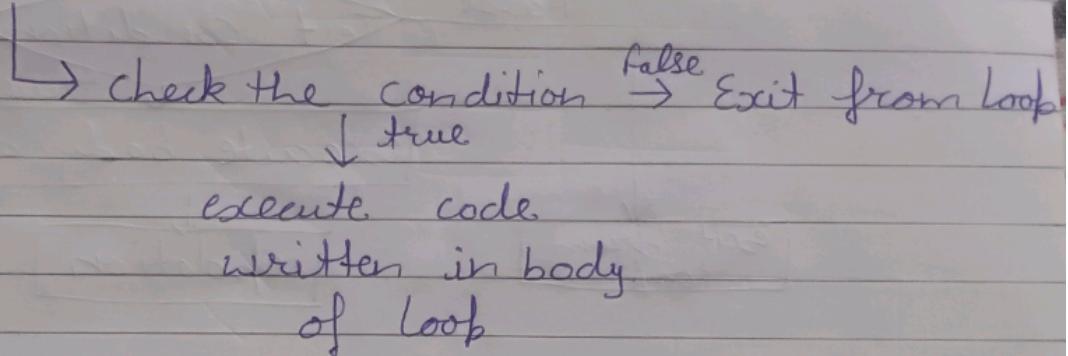
## # Loop

Advantages :-

- ① Code reuse ability
- ② Time saving
- ③ Traversing is easy

## # Flow

Loop start



## # Types of Loop

① Entry Controlled loop → In entry controlled loop the test condition is checked before entering the main body of the loop. ① for ② while

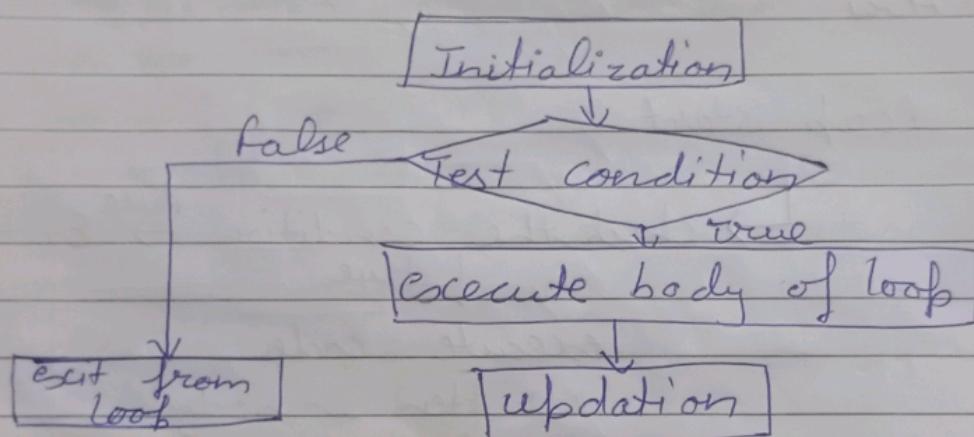
② Exit Controlled loop → In exit controlled loop the test condition is evaluated at the end of loop body. The loop will execute atleast one irrespective of weather condition is true or false. ① do-while

# for loop

```
i=0 ; i<=9.  
for (initialization; test condition; updation)  
{  
    printf ("%d", &i);  
}
```

increment i++  
decrement i--

# Flow chart



# write a program to calculate sum of first 10 Natural Number using for loop.

```
# include <stdio.h>  
void main ()  
{
```

```
int i , sum = 0;
```

```
printf ("First 10 Natural Number");
```

```
for (i=1 ; i<=10 ; i++)
```

```
{  
    sum = sum + i ;
```

```
    printf ("%d", i);
```

```
    }  
    printf ("%d", sum);  
}
```

Output : The first 10 natural number  
1 2 3 4 5 6 7 8 9 10 55

# Somebody want to print 1 to 10 with the help of loop.

```
# include <stdio.h>  
void main ()  
{  
    int i;  
    for (i=1; i<=10; i++)  
    {  
        printf ("%d\n", i);  
    }  
}
```

## # Factorial Table

```
# include <stdio.h>  
int main ()  
{  
    int n, fact = 1, i; T;  
    printf ("enter the value of n");  
    scanf ("%d", &n);  
    for (i = 1; i<=10; i++)  
    {  
        T = n * i;  
        fact = i * fact;  
        printf ("%d", fact);  
    }  
}
```

# Find out the factorial for entered number.

```
# include <stdio.h>
Void main ()
{
    int i, n, fact = 1;
    printf ("enter the value of n");
    scanf ("%d", &n);
    for (i=1; i ≤ n; i++)
    {
        fact = fact * i;
    }
    printf ("%d", fact);
}
```

Print sum write a program  
of 123 # to print sum of digit of specific no

```
# include <stdio.h>
Void main ()
{
    int n, r, sum = 0;
    printf ("enter the value of n:");
    scanf ("%d", &n);
    for ( ; n > 0; )
    {
        r = n % 10;
        n = n / 10;
        sum = sum + r;
    }
    printf ("%d", sum);
}
```

★ Write a program to check that the entered no. is prime or not.

```
# include <stdio.h>
void main ()
{
    int i, n, flag=0;
    printf (" enter n");
    scanf ("%d", &n);
    for (i=2; i<n; i++)
    {
        if (n%2 == 0)
        {
            printf (" not prime");
            flag++;
        }
        if (flag == 0)
    }
    printf (" prime number");
}
```

⇒ Continue statement is used to skip some iteration.

```
for (i=0; i<n; i++)
{
    if (i == 7)
        continue;
    printf ("%d", i); }
```

★ Write a program to print counting from 1 to 10 except 7.

```
# include <csstd.h>
void main()
{
    int i;
    for (i=1; i<=10; i++)
    {
        printf ("%d\n", i);
        if (i==7)
        {
            continue;
        }
    }
}
```

★ Write a program to enter the sum of 2 previous no.

```
# include <csstd.h>
void main()
{
    int n, F=0, sum=0, S=1;
    printf ("enter n");
    scanf ("%d", &n);
    printf ("%d %d", F, S);
    for ( ; S<n; )
    {
        sum = F+S;
        F = S;
        S = sum;
        printf ("%d", sum);
    }
}
```

⇒ while loop

```
i ≤ 10  
while (condition)  
{  
    printf ("%d", i)  
    i = i + 1;  
}
```

- \* Take the input from the user and find out factorial.

```
# include <stdio.h>  
void main ()  
{  
    int i, n, fact = 1;  
    printf ("enter the value of n");  
    scanf ("%d", &n);  
    while (i ≤ n) {  
        fact = fact * i;  
        i++;  
    }  
    printf ("%d", fact);  
}
```

- \* while 0112358 - - -

```
# include <stdio.h>  
void main ()  
{
```

```

int n, f=0, sum=0; // Set f=0, sum=0
printf ("enter the value of n"); // Print "enter the value of n"
scanf ("%d" & n); // Scan n
while (n>0) { // While n is greater than 0
    sum = f + s; // sum = f + s
    f = s; // f = s
    s = sum; // s = sum
    printf ("%d %d", f, sum); // Print f and sum
}
    
```

G)

\* print the table for 7.

```
# include <stdio.h>
```

```
void main ()
```

```
{
```

```
int n, i, T;
```

```
printf ("enter n");
```

```
i = 1;
```

```
while (i <= 10)
```

```
{
```

```
T = n * i;
```

```
i++;
```

```
}
```

```
printf ("%d\n", T);
```

```
}
```

```
}
```

\* while — palindrome number.

```
# include <stdio.h>
void main ()
{
    int n,r,c,s=0;
    printf (" Enter any Number : ");
    scanf ("%d",&n);
    c = n;
    while (n > 0)
    {
        r = n % 10;
        s = r + (s * 10);
        n = n / 10;
    }
    if (c == s)
        printf (" Palindrome Number ");
    else
        printf (" Not Palindrome Number ");
    return 0;
}
```

\* Do Syntax

```
Do
{
    Statement
}
```

```
while (condition)
```

```
while (condition)
{
    Statement
}
```

```

i=6;
Do
{
    printf ("%d", i);
}
while (i<5);
printf ("HELLO");
}

```

Output : 5 HELLO

```

i=6;
while (i<5)
{
    printf ("%d", i);
}
printf ("HELLO");
}

```

Output - HELLO

### \* Factorial of n with Do-while

```

#include <stdio.h>
int main ()
{
    int i, n, fact = 1;
    printf ("enter the value of n");
    scanf ("%d", &n);
    i = 1;
    do
    {
        fact = fact * i;
        i++;
    }
    while (i < n);
    printf ("%d", fact);
}

```

## \* Function definition :-

Return type function Name()  
{  
} body of function

## \* Function calling :-

Function-name();

## 3) function definition

```
#include <csflio.h>
void main()
{
    void sum();
}
int void sum()
{
    int a; int sum;
    int b;
    printf("Enter value of a and b:");
    scanf("%d %d", &a, &b);
    sum = a+b;
    printf("sum=%d", sum);
```

\* fact with function - ⑤

```
# include <stdio.h>
void fact()
{
    int i, n, fact = 1;
    printf ("enter the value of n");
    scanf ("%d", &n);
    for (i = 1; i <= n; i++)
    {
        fact = fact * i;
    }
    printf ("%d", fact);
}

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

\* write a program for swapping variable with swap function.

```
# include <stdio.h>
void swap()
{
    int a, b, c;
    printf ("enter the value of a, b");
    scanf ("%d %d", &a, &b);
    c = a; a = b; b = c;
    printf ("%d\n%d", a, b);
}

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

\* find the area of rectangle with function

```
# include <stdio.h>
void area ()
{
    int area , length , breadth ;
    printf ("Enter the value of length, breadth ");
    scanf ("%d %d" , &length , &breadth );
    area = length * breadth ;
    printf ("%d\n" , area );
}
```

```
void main ()
{
```

```
    area ();
    area ();
}
```

```
}
```

### Call by Value

```
# include <stdio.h> -⑦
```

```
void swap (int a , int b)
```

```
{
```

```
    int c;
```

```
c = a;
```

```
a = b;
```

```
b = c;
```

```
printf ("%d %d" , a , b );
```

```
}
```

```
void main ()
```

```
{
```

```
    int a , b;
```

```

printf ("enter the value of a, b");
scanf ("%d%d", &a, &b);
swap(a, b);
printf ("%d %d", a, b);
}

```

⇒ # include <stdio.h>

```

void swap (int a, int b)
{
    int c;
    c = a; a = b; b = c;
    printf ("%d %d", a, b);
}

```

void main()

```

int a, b, c, d, e, f, g, h, j
printf ("enter the values");
scanf ("%d%d%d%d%d%d%d%d", &a, &b, &c, &d, &e, &f, &g, &h);
swap (a, b);
swap (c, d);
swap (e, f);
swap (g, h);
}

```

c	10	c	30	e	50	c	20
b	20	b	40	b	60	b	80
a	10	a	30	a	50	a	70

ab)

10, 20

cd

30, 40

ef

50, 60

gh

70, 80

⑥ # include <stdio.h>  
void fact (int  $\hat{a}$ )  
{  
 int fact = 1, i;  
 for (i = 1; i <=  $\hat{a}$ ; i++)  
 {  
 fact = fact \* i;  
 }  
 printf ("%d", fact);  
}  
  
void main()  
{  
 int  $\hat{a}$ ;  
 printf ("enter the values");  
 scanf ("%d",  $\hat{a}$ );  
 factorial ( $\hat{a}$ );  
}

⇒ void factorial()

# include <stdio.h>  
int e = 10;  
void f()  
{  
 int a = 20;  
 printf ("%d %d", a, e);  
 e = e + 10;  
}  
  
void main()  
{  
 int a = 10;

```
printf ("%d %d", a, e);
e = 15;
f();
printf ("%d", e);
```

⇒ Fibonacci series.

```
#include <stdio.h>
void fibo (int n)
{
    int f = 0, s = 1, sum = 0;
    printf ("%d %d", f, s);
    for ( ; sum < n; )
    {
        sum = f + s;
        f = s;
        s = sum;
        printf ("%d", sum);
    }
}
void main ()
{
    int n;
    printf ("enter n");
    scanf ("%d", &n);
    fibo (n);
}
```

Output : Enter n 13  
0 1 1 2 3 5 8 13

```
# include <stdio.h>
int fact (int n)
{
    int fact = 1;
    for (int i = 1; i <= n; i++)
    {
        fact = fact * i;
    }
    return fact;
}
void main()
{
    int n, c;
    printf (" enter n");
    scanf ("%d", &n);
    c = fact (n);
    printf ("%d", c);
}
```

```
int i, j, K = 0;
for (j = 0; j < i; j++)
{
    for (i = 0; 3 < i; i++)
    {
        printf (" enter value of k% element, k");
        scanf ("%d", &a[i][j]);
        k++;
    }
}
printf (" it is %d", a[i][j])
```