

switch - :

In the case of multiple 'if-else' statements, until the condition is satisfied all the conditions has to check. Due to this the performance of the system will be degraded, to check all the conditions.

To overcome all of these problems we can introduce switch programming.

Switch is one more decision making statement. It is executed based on the user choice, only that case will be evaluated.

- * The switch type and case type must be match. If it not match then default statement will be executed. Either the switch expression or the case values can be integer or character.

Switch executes only the required condition, not all the conditions as in multiple 'if-else' case.

Syntax -

```
switch (integer variable or  
integer expression or character  
variable) {  
    case:integer or character constant-1:  
        statement(x);  
    case:integer or character constant-2:  
        statement(x);  
    case:integer or character constant-3:  
        statement(x);  
    default:  
        statement(x);  
}
```

When we defining the case, we have to follow some of the rules that must be integers or character.

- * Cases can be define in some of the formats like -

Ex. case 0 :

~~Ex.~~ case 100 ;

Ex. case -10 :

Ex. case 56 :

~~Ex.~~ case 'A' ;

Ex. case 'A' :

Ex. case '+' :

~~Ex.~~ case '+' ;

Ex. case 1+2*3 ;

Note - If the case value is a character then it must be defined in single quote.

* cases can not be define in some of the formats like -

Ex. case 1 ;

case 1 :

'A' ; // duplicates not allowed

Ex: case 1, 2, 1 ; // commas not allowed

Ex: case Num :

// Num is a variable
and variables are not allowed.

Ex: case 46.57 :

// floating not allowed

Ex: case "abc" :

// strings not allowed.

Ex: case 5 < 6 :

// relational operators
are not allowed.

Ex: switch (a, b) // invalid.

Here a, b are variables
and variables are not allowed
in switch ; if a, b are
constants then they are
valid statement.

Ex: case 1, 2 ; // not allowed

Ex: switch (1, 2) // valid .

Note:

Every 'if' condition programming can be convert into switch.

Ex. main()

{

int choice;

clrscr();

printf("111*****109 0 0 1 * * *");

printf("111***** LOOT POLICE ***\n");

printf("111***** 108 AMBULANCE ***\n");

printf("111***** 103 ENQUIRY ***\n");

printf("Enter the vehicle choice:");

scanf("%d", &choice);

switch(choice)

{

case 100: printf("POLICE");

break;

case 108: printf("AMBULANCE");

break;

case 103: printf("ENQUIRY");

break;



Scanned with OKEN Scanner

```
default : printf ("WRONG CHOICE");
}
getch ();
}
```

O/P →

```
***** * * * * * * * * * * * * * * * * * *  
* * * 100 POLICE ***  
* * * 108 AMBULANCE ***  
* * * 103 ENQUIRY ***
```

Enter the choice : 103

ENQUIRY

Note - :

Whenever the compiler read the break statement, it immediately comes out of the switch. Break is optional statement.

Note - :

The default can be present anywhere in the switch. The default also is optional and it can present anywhere in the program.

Ex. Write a program to implement all the arithmetic operations using switch.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b;
    char choice;
    clrscr();
    printf("Enter the first value:");
    scanf("%d", &a);
    printf("Enter the second value:");
    scanf("%d", &b);
    printf("Enter the choice :");
    fflush(stdin);
    scanf("%c", &choice);

    switch(choice)
    {
        case '+': printf("Sum is : %d", a+b);
                     break;
        case '-': printf("Subtraction is : %d", a-b);
                     break;
        case '*': printf("Product is : %d", a*b);
                     break;
        case '/': printf("Quotient is : %d", a/b);
                     break;
        case '%': printf("Remainder is : %d", a%b);
                     break;
        default: printf("Invalid choice");
    }
}
```

```
break;  
case '*': printf("Multiplication :  
%d", a*b);  
break;  
case '/': printf("Division :  
%d", a/b);  
break;  
case '%': printf("Modulus :  
%d", a % b);  
break;  
default: printf("Wrong choice");  
getch();  
}
```

Enter first value: 70

Enter second value: 50

Enter the choice: *

Multiplication : 3500

Ex. 8. Avoid "main()"

```
int choice = 1;
clrscr();
switch(choice, choice+1, choice+2)
{
```

case 1: printf("In ternary");

break;

case 2: printf("In if");

break;

case 3: printf("In switch");

break;

}

getch();

}

O/P → switch.

Ex. void main()

```
int choice=2; clrscr();
switch(choice)
{
    case 1+2/3: printf("In case 1");
        break;
```

```

case 2/2 * 3 : printf("In case 3");
    break;
}
}
getch();
}
}

```

O/P → 1

Program will execute but it will not show any output.

Ex. #include <stdio.h>

#include <conio.h>

void main()

{

int a = 3, b = 2;

a = a == b == 0;

switch (1)

{

a = a + 10; getch();

}

sizeof (a++);

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

}

O/P → 1

Ex.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i = 5;
    switch (i)
    {
        case 6:
            i = 6;
    }
}
```

case 6 : printf ("ATUL");

case 5 : printf ("AKHILESH");
 }

getch();

O/P → AKHILESH

Ex. #include <stdio.h>

#include <conio.h>

void main()

{

int i = 5; if (i >= 3) then

clrscr(); clrscr(); then

a = a >= 4;

switch (2)

{

if (i >= 2) then

```
case 0 : a = 8;  
case 1 : a = 10;  
case 2 : ++a;  
case 3 : printf ("%d", a);  
}  
  
getch();  
}
```

O/P → 11 (H2) (H3) (H4) (H5) (H6)

Ex. Write a program to accept a character. Find out if the given character is vowel or not.

```
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
    char ch;  
    printf ("Enter the choice: ");  
    scanf ("%c", &ch);  
  
    switch (ch)  
    {  
        case 'a':  
        case 'e':  
        case 'i':  
        case 'o':  
        case 'u':  
            cout << "Vowel" << endl;  
        default:  
            cout << "Consonant" << endl;  
    }  
}
```

case 'e':

case 'i':

case 'o':

case 'u':

case 'A':

case 'I':

case 'O':

case 'E':

case 'U':

printf ("VOWEL");

else break;

default: printf ("CONSONENT");

}

getch();

Ex. Write a program to accept day, month and year. Find out the weak day.

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    int dd, mm, yy;
    long dp = 0;
```

int dd, mm, yy;

long dp = 0;

```
clrscr();
```

```
printf("Enter day no.:");
```

```
scanf("%d", &dd);
```

```
printf("Enter month no.:");
```

```
scanf("%d", &mm);
```

```
printf("Enter year.:");
```

```
scanf("%d", &yy);
```

```
dp = (yy - 1) * 365 + (yy - 1)/4;
```

```
switch (mm)
```

```
{
```

```
case 12: dp += 30; // Nov
```

```
case 11: dp += 31; // Oct
```

```
case 10: dp += 30; // Sep
```

```
case 9: dp += 31; // Aug
```

```
case 8: dp += 31; // Jul
```

```
case 7: dp += 30; // Jun
```

```
case 6: dp += 31; // May
```

```
case 5: dp += 30; // Apr
```

```
case 4: dp += 31; // Mar
```

```
case 3: dp += 28; // Feb
```

```
case 2: dp += 31; // Jan
```

```
case 1: dp += dd;
```

```
}
```

if ($yy \% 4 == 0 \ \&\ & \ mm > 2$)

dp++;

switch (dp % 7)

{
 case 0 : printf ("Sunday");
 break;

case 1 : printf ("Monday");
 break;

case 2 : printf ("Tuesday");
 break;

case 3 : printf ("Wednesday");
 break;

case 4 : printf ("Thursday");
 break;

case 5 : printf ("Friday");
 break;

case 6 : printf ("Saturday");
 break;

y
 printf ("In Total days passed : %d", dp);

getch();

y

Ex: main ()

{ switch (5)

{ case 5 : printf ("5");
default : printf ("LO");
case 6 : printf ("6");

} getch ();

} ; if

O/P → 5 , 20, ; 6

Ex: main ()

{ if ("pohamit") ; if

int i = 5, j=0; ; if

switch (i && ++j || --j) ; if

{ ; if

case 1: ; if

printf ("2"); ; if

default : if

printf ("default"); ; if

} ; if

O/P → 2 default

Ex: main ()
{
 float j = 1.5;
 switch (j)
 {
 case 1.0 : printf ("%f", j);
 break;
 case 1.5 : printf ("%f", j);
 break;
 case 2.5 : printf ("%f", j);
 break;
 }
 getch ();
}

O/P → Error! (case does not contain floating point).

Ex: main ()
{
 unsigned short a = -1;
 unsigned char b = a;
 printf ("%d %d", a, b);
}

O/P → -1 , 255 .

Ex. main()

```
{  
int c=0, d=5, e=10, a;  
a = c > 1 ? d > 1 || e > 1 ? 100:200:300;  
printf ("%d", a);  
}
```

O/p → a = 300

Ex. main()

```
{  
int i, j = 7;  
printf ("%d\n", i++ * j++);  
}
```

O/p → 56

Ex. What is i value? a)

main()

```
{  
int i = 5;  
if (i >= 5);  
{  
i = 100;  
printf ("%d", i);  
}
```



O/P → 100

Ex: main()

```
{  
    int var = 90;  
    if (var += var == ++var == 89)  
        printf ("%d", var);  
}
```

O/P → 91

Ex: main()

```
{  
    int j;  
    int k = 0;  
    if (k == '0')  
        printf ("One");  
    else if (k == '48')  
        printf ("Two");  
    else  
        printf ("Three");  
}
```

O/P → Three.

Ex: main()

```
{  
    int j = 2;  
    int a = 4;
```



```
if ((i+ = 3) > a) + a>0
{
```

```
printf ("True \n");
{
```

```
else if (av == b) {true
{
```

```
printf ("False \n");
{
```

```
}
```

O/P → True.

Ex. main ()

```
{
```

```
int i = -3, j = 5; //initialization
if (i-- && j--) {false
printf ("%d(%d)", i, j); //ans
{
```

O/P → -4, 4

Ex. main ()

```
{
```

```
int i = 4;
if (i = 2)
```

```
{
```

Statement 1

```
printf ("statement 1");
```

}

else

```
printf ("statement 2");
```

}

O/P → statement 1

Ex. main ()

{

```
int x = 10, y, z;
```

```
z = y = x;
```

```
y = x --;
```

```
i.e. ("before")
```

```
x = 10, y = 10, z = 10;
```

```
printf ("%d %d %d", x, y, z);
```

}

O/P → 10, 0, 2

Ex. main ()

{

```
int k = 4, j = 0;
```

```
switch (k)
```

{

```
case 3: j = 300;
```

```
case 4: j = 400;
```

case 5: $j = 500$;

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

O/P $\rightarrow 500$

Ex. main()

```
{
```

```
int x = 10, y = 20; // P = 5
```

```
if (!(!x) && x)
```

```
    printf("x: %d", x);
```

else

```
    printf("y: %d", y);
```

```
}
```

O/P $\rightarrow x = 10$

Ex.

main()

```
{
```

```
int i = -32;
```

```
char c = -34;
```

```
unsigned int u = -16;
```

```
if (c > i)
```

```
    printf("Pass 1");
```



```

if (c < u)
    printf ("Pass 2");
else
    printf ("Fail 1");

if (i < u)
    printf ("Pass 3");
else
    printf ("Fail 2");

```

Ex. main()

```

for(j=0; j<10; j++)
{
    switch (printf("%d", j))
    {
        case 1: printf("Kku1");
        break;
        case 2: printf("Kku2");
        break;
        default: getch();
    }
}

```

case 1 : $\text{printf}(\text{"Kku1"}, j)$;

$\text{break};$

() return block

$\text{getch}();$

{ }

case 2 : $\text{printf}(\text{"Kku2"}, j)$;

$\text{break};$

() return block

$\text{getch}();$

{ }

default : $\text{printf}(\text{"Kku1"}, j)$;

$\text{break};$

() return block

$\text{getch}();$

{ }

return 0; $\rightarrow \text{Kku1}$

Ex. main ()
{ if ("Aman")
 100;
 printf ("%d\n", 100);
}

O/P → 100

Nested Switch - ~~Structure~~

Nested switch
are supported by 'C' Language.
In Nested switch, it contains
switch inside switch.

Ex. Write a program to generate an
electricity Bill based on the
commercial and demonstrate by
using switch.

Rates are as per
the previous example but
something changed.

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    int nu, ph;
    char type;
    float amt, pt, sc, stamt;
```

```

printf("Init 1. Domestic (D/I)");
printf("Init 2. Commercial (C/E)");
printf("Init Enter Option : ");
scanf("%c", &type);
if (type != 'D' && type != 'd' &&
    type != 'C' && type != 'c')
{
    printf("Invalid type \n");
    getch();
    exit(0);
}
clrscr();
printf("Enter no. of Units : ");
scanf("%d", &nu);
printf("Enter Phase (1 or 3) : ");
scanf("%d", &ph);
if (ph != 1 && ph != 3)
{
    printf("Invalid Phase \n");
    getch();
}
switch(type)
{

```

case 'd':

case 'D':

if (nu <= 50)

amt = nu * 1.45;

else if (nu <= 100)

amt = (50 * 1.45) + ((nu - 50) * 2.85);

else if (nu <= 200)

amt = (50 * 1.45) + (50 * 2.85) +
((nu - 100) * 3.95);

else if (nu <= 300)

amt = (50 * 1.45) + (50 * 2.85) +
(100 * 3.95) + ((nu - 200) * 4.50);

else
amt = (50 * 1.45) + (50 * 2.85) + (100 * 3.95)
+ (100 * 4.50) + ((nu - 300) * 5.00);

sc = 20.00;

pt = nu * 0.06;

if (ph == 1)

{

if (pt < 20)

pt = 20;

}

else

{

Date _____
Page _____

if ($pt < 50$) {
 if ($amt = 0$)
 $amt = 50$;
 else if ($amt > 50$)
 $amt = amt - 50$;
 break;
}

case 'C':
case 'c':
 if ($nu <= 100$)
 $amt = nu * 3.95$;
 else
 $amt = 100 * 3.95 + (nu - 100) * 7.00$;
 $dc = 20.00$;
 $pt = nu * 0.06$;

if ($ph = 2$) {
 if ($pt < 50$)
 $amt = 50$;
 else if ($amt > 50$)
 $amt = amt - 50$;
 break;
}

else if ($amt > 50$)
 $amt = 50$;
if ($amt < 100$)
 $amt = 100$;
break;

$\text{totmt} = \text{amt} + \text{pt} + \text{sc}$;

`printf("In PURPOSE (C-Commercial,
D-Domestic): %c", type);`

`printf("In Total no. of units :
%.d", nu);`

`printf("In Phase-type : %.d", ph);`

`printf("In Bill amount : %.2f Rs",
amt);`

`printf("In Service Charge : %.2f Rs",
sc);`

`printf("In Power tax : %.2f Rs",
pt);`

`printf("In Total Bill Amount :
%.2f Rs", totmt);`

`getch();`

Output →

1. Domestic (D/d)
2. Commercial (C/c)

Enter < Option :- d for domestic

then next output page will be
open as -

Enter no. of Units : 59

Enter Phase (1 or 3) : 3

PURPOSE (C-Commercial, D-Domestic) : d

Total no of units : 59

Phase Type : 3

Billed Amount : 98.15 Rx.

Service Charge : 10.00 Rx.

Power Tax : 50.00 Rx.

Total Bill + Amount : 158.15 Rx.

Ex. Write a program to display the output in the format 1 2 3 4 5.

```
#include <stdio.h>
#include <conio.h>
{
    clrscr();
    printf("1\n2\n3\n4\n5");
    getch();
}
```

O/p → 1 2 3 4 5

Note :-

The above program is not easy to implement based on the decision making statements. To overcome this problem of writing multiple statements we have to use iterations supports of repetition. (co-repetition).

The drawback of decision making statement is they can check the condition only one time.

* Any function will return a value but receiving the value is optional.
For suppose `printf()` returns no. of characters in the double quotes of the first argument.
`scanf()` returns no. of format specifiers that is also first argument.

Ex. void main() {
 int x;
 x = printf("Kiran(8)\n");
 printf("%d", x);
 getch();
}

O/P → Kiran(8)
8

Ex. main()
{
 printf("%d", printf("Hello"));
 getch();
}

O/P → Hello 5

```
Ex. void main()
{
    printf("%d", printf("%d %d Hi",
                        10, 20));
    getch();
}
```

```
Ex. void main () {  
    int a;  
    clrscr();  
    a = printf("Hai %d Kiran", 10);  
    printf("SQL");  
    printf("%d", a);  
    getch();  
}
```

Ex: void main ()
{
int a, b, c;

a = printf("Welcome Naresh");

b = scanf("%d%d", &c);

// Entered values are 20,

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

}

0010011010 30

06 00101

O/p → Welcome Naresh

Instrumentation by [redacted] [redacted]

Ex: void main() {
 int a, b, c; // declaration

a = 100; // assignment
 b = c = 200; // assignment

clrscr(); // clear screen

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

a = printf("Welcome Naresh %d",
printf("Hello Kiran"));

a = printf("\n%d%d%d%d", scanf("%d
%d", &b, &c), a, b, c);

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

getch();

}

O/P →

Gv 100100 Hello Kiranklecome Narekh11

216100100

1010030

Unconditional Uncontrolled Statements:-

The

'goto' is not an actual looping statement. It is an unconditional uncontrolled statement. This can be written in two ways.

One is Backword Declaration and another one is Forward Declaration.

Backword Declaration:-

Syntax -

<Labelname> : ←

Statement 1;

Statement 2;

Statement 3;

Statement 4;



Scanned with OKEN Scanner

Forward Declaration :-

Syntax -

goto <labelname>;
statement 1;
statement 2;

Infinite loop
Labelname <labelname>;
Statement 1
Statement 2

Note -

Labelname is nothing but user defined statement.

Ex. main ()
{
 x :
 printf ("Atul");
 goto x;
 getch();
}

Note :-

The above program unconditionally rotating infinite times. It does not have any flexibility of checking the condition. Due to this it is called as unconditional. The process of rotating will continuously take

place that is uncontrolled. To stop the above program manually we have to press (ctrl + break).

To make this unconditional, uncontrolled statement as a conditional ~~controlled~~ statement we have to use any conditional statement like 'if'.

```
Ex. void main()
{
    clrscr();
    int i = 1;
    x:
    if (i <= 5)
    {
        printf("Atul");
        i++;
        goto x;
    }
    getch();
}
```

O/P → AtulAtulAtulAtulAtul

Explanation: In this program, the loop starts at i = 1. Inside the loop, the printf statement is executed, printing "Atul". Then, the value of i is incremented by 1. After the increment, the goto statement is executed, which jumps back to the label "x", effectively creating an infinite loop. The getch() statement at the end of the loop prevents the program from exiting immediately after the first iteration.



Q) Write a program to print max
of two numbers.

void main ()

{ int j=1,i;

char ch;

scanf("%d", &i);

if (i>=100)

{ printf("100");}

printf(" %d ", i);

i++;

getchar();

getch();

}

Q) Write a program to findout some
of series upto n terms by
accepting n value.

void main () {

int n, j=1, num=0;

char ch;

printf("Enter the value of n: ");

scanf("%d", &n);

```

series:
if (j <= n)
{
    sum += j;
    j++;
    goto series;
}
printf("Sum of series 1+2+3+--\n-- %d is %d", n, sum);
getch();

```

0/P →
Enter the value of n : 8

Sum of series 1+2+3+-- +8
is 36

Ex: Write a program to accept a no, display the no. in reverse order.

First method -

```

void main()
{
    int num;

```

```
printf("Enter the number: ");
scanf("%d", &num);
```

```
reverse:
if (num)
{
    printf("%d", num % 10);
    if (num / 10 == 0)
        goto reverse;
    getch();
}
```

o/p →
Enter the number: 5896
6985

Second method :-

EXPLANATION: First method is not the actual logic for this program. The actual logic to solve this program is given below -

```
Void main()
{
```

```
int num, rev=0, temp;
clrscr();
```

printf("Enter the number : ");
scanf("%d", &num);

temp = num;

reverse:

if (num > 0)

{
 rev = rev * 10 + (num % 10);

 num = num / 10;

 goto reverse;

}

printf("Reverse of %d is %d",
 temp, rev);

getch();

}

O/P →

Enter the number : 5973

Reverse of 5973 is 3795

Ex.

Ex: Write a program

to display

the given -

1 10

2 9

3 8

4 7

5 6

6 5

7 4

8 3

9 2

10 1

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int i = 1;
    xyz:
    if (i <= 20)
        printf("%d %d\n", i, 21 - i);
    i++;
    goto xyz;
    getch();
}

```

Ex. With what condition both the statements are displayed.

```

main()
{
    if ()
        printf("Hello");
    else
        printf("Kiran");
}

```

An - if(!printf("Hello"))

or
if (printf("Hello") - 5)

Conditional Controlled Statement:-

1. ~~for~~ while -

2. for .

1. While -:

The while is an entry controlled loop statement. First it checks the condition. If the condition is satisfied, there is an entry inside the loop. and execute the statement i. The process of loop is rotating will be taken place untill the condition is satisfied.

Syntax - with example

```
initialization ;  
while (condition)  
{  
    stmt 1;  
    stmt 2;  
    -----  
    -----  
    update;  
}
```

Ex: int
 89
num

the difference b/w looping and decision making statements, decision making statements check the condition only one time but the looping statement can check more than one time.

Ez. Write a program to print nos. from 1 to 10.

```
main()
{
    int i = 1;
    clrscr();
    while (i <= 10)
    {
        printf("%d", i);
        i++;
    }
    getch();
}
```

O/P →

1	2	3	4	5	6	7	8
9	10						

Ez. Write a program to find out squares and cubes from 1 to 10 numbers.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i = 1; clrscr();
    printf("Number \t Square \t Cube\n");
    while (i <= 20)
    {
        printf("%d \t %d \t %d\n", i, i*i, i*i*i);
        i++;
    }
    getch();
}
```

O/p →

Number	Square	Cube
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

Ex. Write a program to accept a no.
Display the number in reverse
order.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int num, rev=0, temp;
    clrscr();
    printf("Enter the number:");
    scanf("%d", &num);
    temp = num;
    while (num)
    {
        rev = rev * 10 + (num % 10);
        num = num / 10;
    }
    printf("Reverse of %d is %d", temp, rev);
    getch();
}
```

O/P → Enter the no! 5873
Reverse of 5873 is 3785

Ex: Write a program to accept a number. Find out the given no. is palindrome or not.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int num, rev=0, temp;
    clrscr();
    printf("Enter the number:");
    scanf("%d", &num);
    temp = num;
    while (num)
    {
        rev = rev * 10 + (num % 10);
        num = num / 10;
    }
    printf("Reverse of %d is %d", temp, rev);
    if (temp == rev)
        printf("In this no. is palindrome");
    else
        printf("In this no. is not palindrome");
}
```

printf("In this no. is not a
palindrome");

getch();

}

O/P →

Enter the number : 454

Reverse of 454 is 454

this no. is a palindrome

Ex. Write a program to accept a no.
Find out if the given no. is Armstrong or
not.

#include <stdio.h>

#include <conio.h>

void main()

{

int num, temp, x, sum = 0;

clrscr();

printf("Enter the number:");

scanf("%d", &num);

temp = num;

```
while (num)
```

```
{ // Armstrong number
```

```
x = num % 10;
```

```
x = x * x * x;
```

```
sum += x;
```

```
num = num / 10;
```

```
}
```

```
printf("Sum of cube of all digits  
is %d", sum);
```

```
if (temp == sum)
```

```
printf("In this no is armstrong");
```

```
else  
printf("In this no. is not armstrong");
```

```
getch();
```

```
}
```

O/P →

Enter the number : 153

Sum of cube of all digits is 153

This no. is armstrong

Ex Write a program to accept a no.
 Find the multiplication table for that no.

```
#include <stdio.h>
#include <conio.h>
main()
{
    int num, i=1;
    printf("Enter the number: ");
    scanf("%d", &num);
    printf("Multiplication table is : ");
    while(i<=10)
        printf("\n %d * %d = %d", num, i, num*i);
    i++;
}
getch();
```

O/P → Enter the number : 7

Multiplication table is :

$$7 * 1 = 7$$

$$7 * 2 = 14$$

$$7 * 3 = 21$$

$$7 * 10 = 70$$



Ex: Write a program to accept two nos. Find the product of two nos. (For ex. 2 and 3). (try to solve without using multiplication operator).

```
#include <iostream.h>
#include <conio.h>
void main()
{
    int sum=0, a, b, i=1;
    clrscr();
    printf("Enter two numbers : ");
    scanf("%d %d", &a, &b);
    printf("Product of two nos. is : ");
    while (i <= b)
    {
        sum += a;
        i++;
    }
    printf("%d * %d = %d", a, b, sum);
    getch();
}
```

O/P → Enter two numbers = 4
7

Product of two no is : $4 * 7 = 28$

Ex Write a program to accept two numbers. Display all the multiplication table between them -.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n1, n2, n, i=1;
    clrscr();
    printf("Enter two numbers : ");
    scanf("%d %d", &n1, &n2);
    printf("Multiplication table is : \n");
    while (i<=10)
    {
        n = n1;
        while (n<=n2)
        {
            printf("%d * %d = %d\n", n, i, n*i);
            n++;
        }
        i++;
    }
    printf("\n");
```



getch(); // creates a sticky key
// until the value is read
// with interrupt ident

O/P →

Enter two numbers: 5 7

Multiplication Table in C

$5 * 1 = 5$	$6 * 1 = 6$	$7 * 1 = 7$
$5 * 2 = 10$	$6 * 2 = 12$	$7 * 2 = 14$
$5 * 3 = 15$	$6 * 3 = 18$	$7 * 3 = 21$
$5 * 4 = 20$	$6 * 4 = 24$	$7 * 4 = 28$
$5 * 5 = 25$	$6 * 5 = 30$	$7 * 5 = 35$
$5 * 6 = 30$	$6 * 6 = 36$	$7 * 6 = 42$
$5 * 7 = 35$	$6 * 7 = 42$	$7 * 7 = 49$
$5 * 8 = 40$	$6 * 8 = 48$	$7 * 8 = 56$
$5 * 9 = 45$	$6 * 9 = 54$	$7 * 9 = 63$
$5 * 10 = 50$	$6 * 10 = 60$	$7 * 10 = 70$

Implementation of Looping Concept

Looping concept can be implemented in three ways -

With Body

Without Body

With Semicolon.

1. With Body :- It has a loop

Ex:- main ()

```
{ int j = 1;
while (j <= 10)
{
    printf ("%d", j);
    j++;
}
getch();
```

O/P → 1 2 3 4 5 6 7 8 9 10

2. Without Body :- It has a loop

Ex:- main ()

```
{ int j = 1;
while (j <= 10)
printf ("%d", j);
j++;
getch();
```

O/P → 1 1 1 1 ----- infinite times

Compiler will do some replacement in this program as -

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

```
{
```

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

```
}
```

```
    i++;
```

```
}
```

Note :-

In with Body compiler do not make any replacement but in without Body compiler do some replacements.

3. With Semicolon - ; in function

Ex. main ()

```
{
```

```
    int i = 1;
```

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

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

```
        i++;
```

```
}
```

O/P → Program will execute but don't display anything.

Compiler will do some replacement
in this program : ax -> j

```
while (j <= 10) {  
    if (a <= 0 || b <= 0) break;  
    printf ("%d", j);  
    j++;
```

Ex. main ()

```
{  
    int a = 5;  
    while (a)  
    {  
        printf ("%d", a);  
        a = a - 1;
```

```
    }  
}
```

O/P → 5 4 3 2 - 1 0

Ex. main ()

```
{  
    int a, b;  
    clrscr ();  
    a = b = 1;  
    while (a)
```

Ex: Place more than 10% solutions

```

a = b <= 3;      // increment side - ad
b = b + 1;
printf ("%d %d", a, b);
}

```

```

printf ("%d %d", a+10, b+10);
getch(); : (i. "bd") HAVING
}

```

O/P → 1 2 () given
 1 3 ,
 1 4 i = 0 true
 0 5 (0) false
 10 15 ,

Ex: main()

```

{
int a;
clrscr();
a = 3;
while (a--)
{
    printf ("%d", a);
    printf ("%d", a+10);
    getch();
}

```

O/P → 2 1 0 ; output



Eg main()

```
{  
    int a;  
    clrscr();
```

a = 1;

while (a++ <= 1)

while (a++ <= 2);

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

```
getch();
```

}

for loop

O/P → 5

for loop

int a = 1;

clrscr();

while (a++ <= 3)

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

```
printf("%.d", a+10);
```

```
getch();
```

}

Replacement

while (a++ <= 1)

while (a++ <= 2)

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

}

(3 times)

(3 times)

(3 times)

(3 times)

Replacement

while (a++ <= 3)

{

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

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

{

```
printf("%d", a+10);
```

O/P → 2 3 4 15

Ex. main ()

```
{
```

```
int a = 1;
```

```
clrscr();
```

```
while (a++ <= 1)
```

```
while (a++ <= 2)
```

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

```
getch();
```

```
}
```

Replacement

```
while (a++ <= 1)
```

```
{
```

```
while (a++ <= 2)
```

```
{
```

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

```
}
```

O/P → 3

Ex. main ()

```
{
```

```
int a = 1;
```

```
clrscr();
```

```
while (a-- >= 1)
```

```
while (a-- >= 0)
```

```
while (a-- >= 0);
```

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

```
getch();
```

```
}
```

Replacement

```
while (a-- >= 1)
```

```
{
```

```
while (a-- >= 0)
```

```
{
```

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

```
}
```

O/P → -3

Ex. main ()

```
{
```

```
int a = 1;
```

```
clrscr();  
while (a <= 1)
```

```
if (a % 2)
```

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

```
else
```

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

```
printf("%d", a+10);
```

```
getch();
```

```
}
```

O/P → 1 12

Ex. #include <stdio.h>

```
#include <conio.h>
```

```
void main()
```

```
{ clrscr(); }
```

```
int a = 1;
```

```
clrscr();
```

```
while (a++ >= 1);
```

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

```
getch();
```

```
}
```

O/P → -32768

Replacement
while (a <= 1)

```
{
```

```
if (a % 2)
```

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

```
else
```

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

```
}
```

```
printf("%d", a+10);
```

```
getch();
```

Replacement

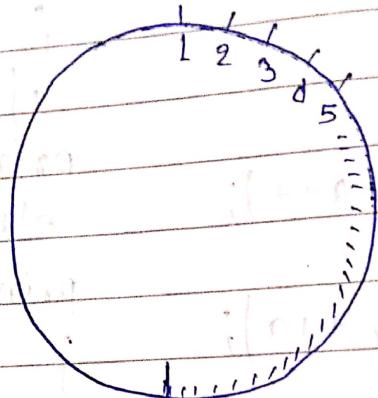
```
while (a++ >= 1)
```

```
{
```

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

```
}
```



Explanation -

32767

+ 1 = -32768

+ 1 = -32767

Ex:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a = -1;
    clrscr();
    while (a--)
        {
            printf("%d", a);
            getch();
        }
}
```

O/P → -1

Unconditional Controlled Statement -:

1. break -:

The 'break' is an unconditional controlled statement. When the compiler reads the 'break' statement, it will come out the loop or switch.

Whenever 'break' must be there inside the loop or switch. Only combination of 'if' and 'break' leads to the error.

Whenever the program result has been specified then we can use the 'break' statement and we can come out manually by using 'break'.

Syntax -

```
main()
{
    while (condition)
    {
        if (condition)
            break; // end of loop
    }
}
```



Ex. #include <stdio.h>

#include <conio.h>

void main()

{

}

int num, i = 1, sum = 0;

clrscr();

while (i <= 10)

{

printf("Enter the value of %.d : ", i);

scanf("%d", &num);

if (num == 0)

{

break;

sum += num;

i++;

{

printf("Sum : %d", sum);

}

printf("Count : %.d Sum : %.d", i, sum);

getch();

}

0/p →

Enter the value of 1: 56

Enter the value of 2: 85

Enter the value of 3: 87

Count : 3 Sum : 141

2. continue - :

The 'continue' is exactly opposite to the 'break' statement. When the compiler read the 'continue' statement, it once again re-enter inside the loop.

'Continue' must be there inside the loop only. If you want to skip specific situation in a condition, use the 'continue' statement. Then we using 'continue' statement iteration part has to prevent.

Syntax -

```
public class main{  
    public static void main()  
    {  
        int i;  
        for(i=1;i<10;i++)  
        {  
            if(i==5)  
                continue;  
            System.out.println(i);  
        }  
    }  
}
```

} End of program

Ex: Write a program to find out biggest number without accepting negative Ex:

values.

```
main() {  
    int num, i = 1, big = 0;  
    clrscr();  
    while (i <= 10) {  
        printf("Enter the number %d : ", i);  
        scanf("%d", &num);  
        if (num < 0)  
            printf("Negative nos are not  
allowed \n");  
        continue;  
    }  
    if (num > big) big = num;  
    j++;  
}  
printf("Biggest : %d", big);
```

Ex:



Ex: main()

```
{  
int j=1;  
clrscr();  
while(j<=10)  
{  
    if(j>4) break;  
    printf("%d\n",j);  
    j++;  
}  
  
printf("%d",j+10);  
 getch();  
}  
0/10 → 1 2 3 4 15
```

Ex:

```
{  
int j=0;  
clrscr();  
while(j<100){  
    if(j>40 && j<60) continue;  
    printf("%d",j);  
    j=j+2;  
}
```



Scanned with OKEN Scanner

~~getch();~~

~~if~~

~~o/p →~~

~~2 4 6 8 ----- 38, 40, 60, 62, --~~

~~-- 96, 98, 100.~~

~~if (i >= 10) break;~~

Ex: main()

```
{  
    int i = 1;  
    clrscr();  
    while (i <= 20)  
    {  
        if (i > 4 && i <= 17) // division  
        continue;  
        printf("%d", i);  
        i++;  
    }  
    getch();  
}
```

~~o/p → 1 2 3 4~~



3. `exit(0)` -:

It indicates when the compiler read the `exit()` statement, it will come out of the program. When `exit()` statement is utilizing a parameter of 0, indication to the operating system the program is successful. The `exit()` can be used anywhere in the program either in looping or decision making statement.

Mostly `exit()` will be used in the decision making statement.

Syntax -

`main() → 0; S.`

```
if (condition) {  
    //  
    //  
    //  
} → Program Successful  
→ exit(0); (Indication for O.S.)
```

out of

the

program if went to end of the program



E. Write a program to accept a no. and print its factorial.

```
main () { int num;
    long f = 1;
    clrscr();
    printf ("Enter the number : ");
    scanf ("%d", &num);
    if ( num == 0 || num == 1 )
        *
    else {
        for ( f = 1; f <= num; f++ )
            f = f * num - 1;
        printf ("Factorial is %ld ", f);
        getch ();
        exit (0);
    }
}

while ( num > 1 )
    f = f * num - 1;
printf ("Factorial is %ld ", f);
getch ();
}
```

O/P →

Enter the number: 5
Factorial is 120 .

*. Fibonacci Series :- In this series

first two values are 0 and 1 and remaining values can be find out by addition of last two values. Fibonacci Series is

given below -

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

* Perfect Number :- A number

beginning with the number 1, keep adding power of 2 (doubling the numbers) until you get sum which is a prime number.

A perfect number is obtained by multiplying the sum to last power of 2.

Ex. Sum Prime = Multiply Perfect

$$1 + 2 = 3 \text{ Yes } 3 * 2 = 6$$

$$1 + 2 + 4 = 7 \text{ Yes } 7 * 4 = 28$$

$$1 + 2 + 4 + 8 = 15 \text{ No } -----$$

$$1 + 2 + 4 + 8 + 16 = 31 \text{ Yes } 31 * 16 = 496$$

Ex'. Write a program to accept a number. Find out if the given no. is in fibonacci series or not.

```
main( ) {  
    int j, j1=0, j2=1, num, ck=0;  
    clrscr();  
    printf("Enter the number: ");  
    scanf("%d", &num);  
    if (num == 0) {  
        ck = 1; /* odd position */  
    } else {  
        ck = 0; /* even position */  
    }  
    while ((j = j1 + j2) <= num) {  
        printf("%d", j);  
        if (j == num) {  
            ck = 1; /* odd position */  
            break; /* even position */  
        }  
        j1 = j2;
```

Ex'



Page 1

Page 2

```
#include <stdio.h>
int fact(int n)
{
    if (n == 1)
        return 1;
    else
        return n * fact(n - 1);
}

int main()
{
    int no, fact, sum = 0;
    printf("Enter a number : ");
    scanf("%d", &no);
    for (int i = 1; i <= no; i++)
    {
        sum += fact(i);
    }
    printf("Sum of first %d terms of Fibonacci series is %d", no, sum);
}
```

Output
Enter the number : 27
Given no. is not present in fibonacci series

Write a program to print all the perfect no.

```
#include <stdio.h>
int fact(int n)
{
    int fact, sum = 0;
    for (int i = 1; i <= n; i++)
    {
        fact = fact(i);
        sum += fact;
    }
    if (sum == n)
        printf("%d is a perfect number", n);
    else
        printf("%d is not a perfect number", n);
}
```



Scanned with OKEN Scanner

Ex'

```
numb = 1;  
while (numb < 1000)  
{  
    fsum = 0;  
    fact = 1;  
    i = 1;  
    while (fact <= numb)  
    {  
        if (numb % fact == 0)  
        {  
            fsum = fsum + fact;  
            fact++;  
        }  
        if (fsum == numb) {  
            printf("%d is a Perfect no.\n",  
                   numb);  
            numb++;  
        }  
    }  
    getch();  
}
```

O/p →

6 is a Perfect no.

28 is a Perfect no.

496 is a Perfect no.

Ex: Write a program to compute the sum of series.

$$1 + x + x^2 + x^3 + x^4 + \dots + x^n$$

```
void main() { show sum off
    float sum = 0.0, term;
    int x, count, term, sum, i=1;
    clrscr(); cout << "Enter the value of x: ";
    printf ("In Enter the value of x: ");
    scanf ("%d", &x);
    cout << "Enter the number of terms ";
    printf ("In upto How many nos. sum
    is needed : ");
    scanf ("%d", &count); cout << "The
    term = sum = 1, acc below
    while (i < count) {
        sum = sum + term;
        term = term * x;
        i++;
    }
    printf ("The sum upto %d terms
    of series is %d", count);
    printf ("In 1 + %d + %d * %d - - -
    getch(); }
```

O/P →

Enter the value of x : 5
upto how many no. sum is needed : 3

The sum upto 3 terms of series

$$1 + 5 + 5 * 5^2 + \dots \text{ i.e } 31.$$

#include <stdio.h>

#include <conio.h>

void main ()

$$1 + 1/x + 1/x^2 + 1/x^3 + \dots + 1/x^n$$

#include <stdio.h>

#include <conio.h>

void main ()

int x, count, term, j = 1;

float sum, y;

clrscr();

printf ("Enter the value of x : ");

scanf ("%d", &x);

term = 1;

printf ("In Upto how many no.

sum is needed : ");

scanf ("%d", &count);

sum = 1;

upto

the

1 +



Scanned with OKEN Scanner

```
while (i < count) {  
    sum = sum + y;  
    i++;  
}  
  
printf("The sum upto %.d term  
of series (%.d)",  
    count);  
  
printf(" %.1f + %.1f + 1/(%.d * %d) + --  
    \n%.1f - %.1f + %.1f, x, x, sum);  
getch();  
}  
else  
{  
    printf("No value for x");  
}  
else  
{  
    printf("No value for x");  
}  
Enter the value of x : 4.  
  
Upto how many no. sum is needed:3  
Value of x: 4.000000  
The sum upto 3 term of series  
1 + 1/4 + 1/(4**2) + --- is 1.312500
```

Ex. Write a program to accept two no's or a power value.

too. Find out the power value.
base .

```
void main ( )
{
    int a, b, i = 1;
    long result = 1;
    clrscr ();
    printf ("Enter the base value : ");
    scanf ("%ld", &a);
    printf ("Enter the power value : ");
    scanf ("%ld", &b);
    while (i <= b)
    {
        result = result * a;
        i++;
    }
    printf ("\n Result is %ld", result);
}
```

O/P →

Enter the base value : 2
Enter the power value : 5
Result is 32

Write a program to accept a number. Find out the given number factors and also find out if a prime number or composite number.

```
void main()
{
    int num, count = 0, i = 1, a;
    clrscr();
    printf("Enter the number: ");
    scanf("%d", &num);
    while (i <= num) {
        if (num % i == 0) {
            count++;
            if (count > 2)
                break;
        }
        i++;
    }
    printf("%d has %d factors", num, count);
}
```



Scanned with OKEN Scanner

```

printf ("\\n Prime number");
else
printf ("\\n Composite number");
getch ();
}

O/P → Enter a number : 15
Enter the number : 15
1 3 5 15 are factors of 15
composite number

# include < stdio.h >
int main()
{
    # include < comio.h >
    # include < math.h >
    void main()
    {
        int n, p = 0;
        long unsigned bin = 0;
        clrscr();
        printf ("Enter any decimal number:");
        scanf ("%d", &n);
        while (n > 0)
        {
            p++;
            bin = bin + (n % 2) * pow(10, p);
            n = n / 2;
        }
        printf ("%d", bin);
        clrscr();
        printf ("\\n Binary");
        scanf ("%d", &p);
        if (p == bin)
            printf ("\\n Number is Palindrome");
        else
            printf ("\\n Number is not Palindrome");
    }
}

```

```
3 printf("Binary no. of given no.: %d\n",  
printf("Binary no. of given no.: %d\n",  
getch());
```

```
} // Main program
```

O/P →

Enter any decimal number : 14
Binary no. of given no. : 1110

* Write a program to accept a number
and also accept two digits. Replace
the first digit with second digit
and display the number.

```
#include <stdio.h>  
#include <conio.h>  
void main ()  
{  
    clrscr();  
    long int n, r = 0, dummy;  
    int d1, d2;  
    printf ("Enter any number : ");  
    scanf ("%ld", &n);  
    printf ("Enter first digit : ");  
    scanf ("%d", &d1);  
    printf ("Enter second digit : ");  
    scanf ("%d", &d2);  
    dummy = n;
```



```
printf("Enter second digit : ");  
scanf("%d", &d2);
```

```
while (n > 0)  
{
```

```
r = r * 10 + (n % 10);
```

```
n = n / 10;
```

```
n = 0;
```

```
while (r > 0)
```

```
dummy = r % 10;
```

```
if (dummy == d1)
```

```
dummy = d2;
```

```
n = n * 10 + dummy;
```

```
r = r / 10;
```

```
printf("%.ld", n);
```

```
getch();
```

```
O/p → 345 371
```

```
Enter any number : 345371
```

```
Enter first digit : 3
```

```
Enter second digit : 1
```

```
145171
```

for - :

'for' is an entry controlled looping statements. The 'for' work flow will take place in anti-clockwise direction.

Syntax -

for (initialization; condition; updation)

{

statement 1;

statement 2; not break

- - - - -

}; } - - - - -

shiva { good 'not' o good msg print

good 'not' o

In 'for' loop everything is optional. we can write it like this -

for (; ;) // invalid

but

while () // invalid

'for' is more flexible rather than 'while'.

In 'for', there can be any no. of initialization, any no. of condition and any no. of updation.



Ques -

Balance of a bank account
for (j = 0, j = 0 ; j <= 5, j <= 10 ; j++ , j--)
{
 j
}

No. of initialization and condition
can be separate with the
comma (,) operator. Between

The condition we can use
relational and logical operators
also.

Nested for :-

Nested for is also
supported by 'C' language. i.e.
there can be a 'for' loop inside
a 'for' loop.

Ex:-

Ex. Write a program to print no.
from 10 to 1 with the help
of for loop.

First method :-

```
main ()  
{  
    for ( i = 10 ; i >= 1 ; i-- )  
        printf ("%d", i);  
}
```

```
int i;  
for ( i = 10 ; i >= 1 ; i-- )  
    printf ("%d", i);  
}
```

```
getch();  
}
```

O/P →

10 9 8 7 6 5 4 3 2 1

Second method -:

Ex:-

```
main()  
{  
    int j;  
  
    for(j=10; j>=1; printf("%d",j--));  
  
    getch();  
}
```

O/P →

10 9 8 7 6 5 4 3 2 1

In the above program compiler will do some replacement as -

```
for(j=10; j>=1; printf("%d",j--))  
{  
}
```

Ex. void main()

```
    {  
        int a, b;  
        clrscr();  
        for(a = b = 1; a; printf("\n%d %d", a, b))  
        {  
            a = b + + ;  
            printf("\n%d %d", a + 10, b + 10);  
            getch();  
        }  
    }
```

O/p → 1 2

1 3

1 4

0 5

20 15

10 10

5 5

0 0

Replacement →

```
for(a = b = 1; a; printf("%d %d", a, b))  
{  
    a = b + + ;  
}
```

```
Ex. void main()  
{  
    int a, b;  
    clrscr();  
    for(a = b = 1; a; printf("%d %d", a, b))  
    {  
        a = b + + ;  
    }  
}
```



Scanned with OKEN Scanner

Ex. void main() {

```
int x = 0, j, k;
clrscr();
for (k = 0, j = 5; k < 5, j > 0; k = k + 2, j--)
{
    x++;
    printf("The value of k, j, x is\n");
    getch();
}
```

```
Output : 0 5 10
         2 4 8
         4 2 6
         6 0 4
         8 -2 2
         10 -4 0
         12 -2 8
         14 -4 6
         16 -2 4
         18 -4 2
         20 -2 0
         22 -4 -2
         24 -2 -4
         26 -2 -6
         28 -2 -8
         30 -2 -10
         32 -2 -12
         34 -2 -14
         36 -2 -16
         38 -2 -18
         40 -2 -20
```

The value of k, j, x is 0
The value of k, j, x is 2
The value of k, j, x is 4
The value of k, j, x is 6
The value of k, j, x is 8
The value of k, j, x is 10
The value of k, j, x is 12
The value of k, j, x is 14
The value of k, j, x is 16
The value of k, j, x is 18
The value of k, j, x is 20
The value of k, j, x is 22
The value of k, j, x is 24
The value of k, j, x is 26
The value of k, j, x is 28
The value of k, j, x is 30
The value of k, j, x is 32
The value of k, j, x is 34
The value of k, j, x is 36
The value of k, j, x is 38
The value of k, j, x is 40

Ex. void main()
{
 int x = 0, j, k;
 clrscr();
}



Scanned with OKEN Scanner

```
for (k=0, j=5; k<5 || j>0; k=k+2, j=j-1)
{
    x++;
    printf("In the value of k,j,x\n");
    %d.%d", k, j, x);
    getch();
}
```

Ex:

O/p →

the value of k,j,x is 0 5 1
the value of k,j,x is 2 4 2
the value of k,j,x is 4 3 3
the value of k,j,x is 6 2 4
the value of k,j,x is 8 1 5

Ex: void main()
{
 int x=0, j, k;
 clrscr();
 for (k=0, j=5; k<5 && j>0; k=k+2, j=j-1)
 {
 x++;
 printf("In the value of k,j,x\n");
 %d.%d", k, j, x);
 getch();
 }
}

Ex:

O/p →



Scanned with OKEN Scanner

O/P → 0 1 2 3 4 5

the value of k, j, x in it on 5 1
the value of k, j, x in it on 4 2
the value of k, j, x in it on 3 3

Ex: void main()

```
int j=0;
clrscr();
for(;;)
{
    printf("%d",j);
    j++;
    if(j>5)
        break;
}
getch();
```

O/P → 0 1 2 3 4 5

Ex: void main()

```
int a=1;
clrscr();
for(a++;a<=2;a++)
{
    printf("%d",a);
}
```



Scanned with OKEN Scanner

```
for(a++; a++ <= 7 ; a++)
```

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

```
getch();  
}
```

O/p → 1 3

Preplacement -

```
for(a++; a++ <= 2 ; a++)  
{  
    printf("%d", a);  
}
```

```
for(a++; a++ <= 7 ; a++)  
{  
    printf("%d", a);  
}
```

Ex.

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

Ex. Write a program to print

0	0	0	1	1	0	2	1	0
---	---	---	---	---	---	---	---	---

1	0	1	1	2	2	2	0	1
---	---	---	---	---	---	---	---	---



```

void main()
{
    int i, j;
    clrscr();
    for( i=0; i<=2; i++)
    {
        for( j=0; j<3; j++)
        {
            printf("%d %d\n", i, j);
            printf("or \n");
        }
    }
    getch();
}
    
```

Write a program to print

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

```

void main()
{
    int i, j;
    clrscr();
    for( i=1; i<=4; i++)
    {
        for( j=1; j<=4; j++)
        {
            printf("%d %d\n", i, j);
        }
    }
}
    
```

```
printf("%d", i);  
printf("\n");  
}  
getch();  
}
```

Ex. Write a program to print

Ex.

```
1 1 1 1  
2 2 2 2  
3 3 3 3  
4 4 4 4
```

don't use 'for' loop. Use 'while' loop.

```
void main()  
{  
    int j=1, j;  
    clrscr();  
    while (j<=4)  
    {  
        j = 1;  
        for (j=1;
```

```
        j<=4)  
        {  
            printf("%d", j);  
            j++;  
        }  
    }  
}
```



```
printf( " \\n" ) ; getch( );  
for( i = 1; i <= 4; i++ )  
    {  
        for( j = 1; j <= i; j++ )  
            printf( "%d", j );  
        printf( "\n" );  
    }  
getch( );
```

Ex. Write a program to print

```
1  
1 2 3  
1 2 3 4  
main( ) {  
    int i, j;  
    clrscr();  
    for( i = 1; i <= 4; i++ )  
        {  
            for( j = 1; j <= i; j++ )  
                printf( "%d", j );  
            printf( "\n" );  
        }  
    getch( );
```

Some program can be written
with the help of 'while' or -

```
main()
{
    int i = 1, j;
    clrscr();
    printf("Program to print\n");
    while (i <= 4)
    {
        j = 1;
        while (j <= i)
        {
            printf("%d", j);
            j++;
        }
        printf("\n");
        i++;
    }
    getch();
}
```

Ex. Write a program to print

1	1	1	1
1	2	2	2
1	2	3	3
1	2	3	4



```
#include <stdio.h> //Firstly X3
#include <conio.h>

void main()
{
    int i, j, n;
    clrscr();
    printf("Enter the N:");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    {
        j=n;
        while(j>i)
        {
            printf("%d", j);
            j--;
        }
        for(j=1; j<=i; j++)
        {
            printf("%d", j);
        }
        getch();
    }
}
```

Ex: Write a program to print -

1 2 1 () minus below

1 2 3 2 1

1 2 3 4 3 2 1 () minus below

```
void main()
{
    clrscr();
    printf("Enter the N : ");
}
```

```
scanf("%d", &n);
```

```
for (j = 1; j <= n; j++)
{
    if (j % 2 == 0)
        printf(" ");
    else
        printf("%d", j);
}
```

Ex:

```
for (j = 1; j <= n; j++)
{
    if (j % 2 == 0)
        printf(" ");
    else
        printf("%d", j);
}
```

```
j = n;
while (j > i)
{
    if (j % 2 == 0)
        printf(" ");
    else
        printf("%d", j);
    j--;
}
```

```
for (j = 1; j <= n; j++)
{
    if (j % 2 == 0)
        printf(" ");
    else
        printf("%d", j);
}
```

```
for (j = 1; j <= n; j++)
{
    if (j % 2 == 0)
        printf(" ");
    else
        printf("%d", j);
}
```



```

printf("%d", j);
j = j;
while (j > 1) {
    printf(" %d", j - 1);
    j--;
}

```

```

printf("\n");
scanf("%d");
getch();
}

```

Ex: Write a program to print

```

1 1 million billion
2 2
3 3 3 one billion three
4 4 4 4 if you do

```

```

void main() {
    clrscr();
    int i, j, n;
    printf("Enter the N : ");
    scanf("%d", &n);
    for (i = 1, j = n; i++)

```

```
{  
    for (j=1; j<=i; j++)  
        printf("%d", j);  
    printf("\n");  
    getch();  
}
```

Ex:- Write a program to print -

```
1  
2 2  
3 3 3  
4 4 4 4
```

void main ()
{
 int i,j,n;
 clrscr();

```
printf("Enter the Number : ");  
scanf("%d", &n);  
for (i=1; i<=n; i++)  
{  
    for (j=1; j<=i; j++)  
        printf("%d", j);  
    printf("\n");  
}
```

```

while(j>=i)
{
    printf(" ");
    j--;
}
for(j=1 ; j<=i ; j++)
{
    printf("%d",j);
    printf("\n");
}
getch();
}

```

Write a program to print -

```

1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4

```

```

main()
{
    int i, j, n;
    clrscr();
    printf("Enter the n: ");
    scanf("%d", &n);
}

```

```

for (j = 1; j <= n; j++)
{
    j = n;
    while (j >= i)
    {
        printf("%n");
        j--;
    }
    printf(" %n");
}
for (j = 1; j <= j; j++)
{
    printf("%d", j);
    j = j;
}
while (j > 1)
{
    j--;
    printf("%d", j);
}
printf("%d", j);
}
printf("%n");
getch();

```

Ex. Write a program to print -

```
4 3 2 1  
3 2 1  
2 1  
1
```

```
void main ()  
{  
    clrscr();  
    int i, j, n;  
    printf("Enter the (N: ");  
    scanf("%d", &n);  
  
    for (i = n; i >= 1; i--)  
    {  
        for (j = i; j >= 1; j--)  
            printf("%d ", j);  
        printf("\n");  
    }  
    getch();  
}
```

Ex:

```
void main()
{
    int j;
    clrscr();
    for(j=1; j<=4; j++)
    {
        for(j=1; j<=4; j++)
        {
            printf("%d", j);
            if(j==4)
                break;
        }
        printf("\n");
    }
    getch();
}
```

Ex:

O/P →

1 2 3 4
1 2 3 4
1 2 3 4

Date _____
Page _____

Ex. void main(void) {
 {
 int i, j;
 clrscr();
 for (i = 1; i < 6; i++) {
 {
 for (j = 1; j <= 5; j++) {
 printf("%d", j);
 }
 if ((j == i)) {
 continue;
 printf("\n%d", j);
 }
 printf("%d", j);
 }
 getch();
 }
 O/P →

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



Second logic (for spaces) :-

Ex. Write

```
void main()
{
    int i, j, r, s;
    char ch = 'A';
    clrscr();
    printf("Enter the row size:");
    scanf("%d", &r);
    for (i = 1; i <= r; i++)
    {
        for (j = 1; j <= i; j++)
        {
            if (i % 2)
                printf(" ");
            else
                printf("%c", ch);
        }
        ch++;
        printf("\n");
    }
}
```

Ex.

```
int i, j, r, s;
char ch = 'A';
clrscr();
printf("Enter the row size:");
scanf("%d", &r);
for (i = 1; i <= r; i++)
{
    for (j = 1; j <= i; j++)
    {
        if (i % 2)
            printf(" ");
        else
            printf("%c", ch);
    }
    ch++;
    printf("\n");
}
getch();
```

Ex: Write a program to print

5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5

main ()

```
{  
    int i;  
    for ( i=1, i<=5, i++) {  
        printf(" 5 ");  
        if ( i % 5 == 0 )  
            printf("\n");  
    }  
}
```

Ex: Write a program to print

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

```
main ()  
{  
    int i,j,n;  
}
```

clrscr();
printf("Enter the value : ");
scanf("%d", &n);
scanf("%c", &c);

for(j=1; j<=n; j++)
{
 if(j==1)
 printf("%d", n);
 else
 printf("0");
}

if(j==1 || j==n || j==i || j==h)
 printf("0");

printf("1");
else
 if(i%2!=0)
 k++;
 else
 k++;
 if(j>k)
 k++;
 else
 k--;
 printf("0");
}

getch();
}

Ex:

Write a program to print
1 2 3 4 5 6 7 8 9

1 2 3 4 5 6 7 8 9
program to print
int i, j, n, k = 1;
main()
{
 clrscr();
 printf("Enter the value : ");
 scanf("%d", &n);
 getch();
 for(j=1; j<=n; j++)
 {
 if(j==1)
 printf("%d", n);
 else
 printf("0");
 }
 if(j==1 || j==n || j==i || j==h)
 printf("0");
 printf("1");
 else
 if(i%2!=0)
 k++;
 else
 k++;
 if(j>k)
 k++;
 else
 k--;
 printf("0");
 }
 getch();
}



```

choice();
printf("Enter the Number : ");
scanf("%d", &n);
for (j=1; j<=n; j++)
{
    for (j=1; j<=n; j++)
    {
        printf("%d", k);
        if (j==n)
            printf("\n");
        k++;
    }
    else
        k--;
}
getch();

```



Ex. Write a program to print

```
1 0 0 0  
1 0 1 0  
0 1 0 0  
0 0 1 0  
0 1 0 1  
1 0 0 1  
1 0 1 0
```

Ex. Write a program to print prime or

```
main()  
{  
    int i,j,k,n;  
    clrscr();  
    textmode(2);  
  
    printf("Enter the number : ");  
    scanf("%d", &n);  
  
    textColor(4+128);  
    for(j=1,k=n; j<=n; j++, k--)  
    {  
        for(i=j, j<=n; j++)  
        {  
            if(i==j || k==j)  
                printf("%-4d", 1);  
            else  
                printf("%-4d", 0);  
        }  
    }  
}
```



```
g  
printf( "%d\n" );  
g  
getch();  
g
```

Ex. Write a program to accept a number. Find out this no. is prime or not.

Third logic (for spaces) :-

```

Ex. void main()
{
    int n, s;
    clrscr();
    printf("Enter the number :");
    scanf("%d", &n);
    s = 5 * n;
    printf("%c", -n, 32);
    getch();
}

```

Note :-

Here 32 is an ASCII code
for space-bar.

```

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

```

```

int i, j, k, n, m;
clrscr();
printf("%d", j);
printf("Enter the number :");
scanf("%d", &n);

```

(m = n;)

for (j = 1; j <= n; i++) b[i]

{

for (j = n; j > m; j--) b[j]

{

printf ("% - 4d", j);

}

for (k = 1; k <= m; k++) b[k]

{

printf ("% - 4d", k);

}

abos

m--;

printf ("%n");

}

getch();

}

O/p →

Enter the number(:) 328312

1 : 2 3

3 : 1 2

3 : 2 1

Ex: Write a program to print -

A B C D E F G F E D C B A
A B C D E F F > F E D C B A
A B C D E E D C B A
A B C D D C B A
A B C A > F E D C B A
A B C B A
A B A
A B A
A B C B A
A B C D C B A
A B C D E D C B A
A B C D E F E D C B A
A B C D E F G F E D C B A

#include <stdio.h>

#include <conio.h>

void main ()

{

int i, j, p, x = 7, y = 7;

clrscr ();

for (i = 1; i <= 13; i++)

{

```
p = 65; /* for ASCII value
          of A */
for (j = 1; j <= 13; j++)
{
    if (j > 1 && j >= x && j <= y)
        printf ("%c - %c", 32);
    else
        printf ("%c", p); /* for printing
                           3 letters */
    p++;
}
p--;
/* for reverse order
   F to A */
}
// inner & outer loop
if (j > 1) /* inner block */
{
    if (j < 7)
        /* for first half increasing
           space */
    x--;
    y++;
}
else
```

for record half
decreasing paper */

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, j, p, n, x;
    clrscr();
    printf("Enter the number : ");
    scanf("%d", &n);
    printf("\n");
    x = 5 * n;
    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= i; j++)
            printf("%c", x);
        printf("\n");
        x--;
    }
}
```

for (j=2; j<=n+1; j++)

{

p = j-1;
printf("%c", x, 32);

for(j=1; j<=2*(j-1); j++)

{

if (p>=0)

printf("%d-4d", p);

else

printf("%d-4d", -(p));

p--;

} //right > solution

printf("\n");

x = x - 4; //left < solution

} //left < solution

getch();

0/P→

Enter the number : 4

1 0 1

2 1 0 1

3 2 1 0 1 2

4 3 2 1 0 1 2 3

4



Scanned with OKEN Scanner

Ex Write a program to accept two no's.
Display all the prime nos. between these two no's.

```
#include <stdio.h>
#include <conio.h> //using
#include <math.h> //sqrt
void main()
{
    long int n1, n2, n, cnt = 0, t, flag;
    clrscr();
    clrscr();
    for (n = n1; n <= n2; n++)
    {
        flag = 0;
        for (t = 2; t < sqrt(n); t++)
        {
            if (n % t == 0)
                flag = 1;
        }
        if (flag == 0)
            cnt++;
    }
    printf("%d prime numbers between %d and %d are\n", cnt, n1, n2);
}
```



```
} function of square root in C programming  
getch();
```

```
o/p →  
Enter two numbers: 121 144  
1 PRIME = 11 since 121 is divisible by 11  
2 PRIME = 2 since 121 is divisible by 2  
3 PRIME = 3 since 121 is divisible by 3  
4 PRIME = 5  
5 PRIME = 7 since 144 is divisible by 7
```

Ex: Write a program to accept a number. Find out the square-root of that no. (without using sqrt function).

```
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
    float n, j;  
    int i, o = 0.01;  
    clrscr();  
    printf("Enter the number: ");  
    scanf("%f", &n);  
    for(j = 1; j * j < n; j++)
```



```

if (i * j == n)
{
    printf("Square-root of %d is %d\n", n, i);
    printf("%f\n", c % d = %.c %d, 25L, n, 24L, j);
}

```

```

else
{
    if (i < j)
        swap(i, j);
    for (j = i + 1; j * j < n; j = j + 0.00001)
        printf("Square-root of %d is %f\n", n, j);
    printf("%f\n", c % d = %.c %d, 25L, n, 24L, j);
}

```

```
getch();
```

O/p →

(1) Enter the number : 81
Square-root of 81 is 9

$$\sqrt{81} = \pm 9$$

(2) Enter the number : 56
Square-root of 56 is 7.483
 $\sqrt{56} = \pm 7.483$

Ex. Write a program to print:

Ex:

```
main()
{
    int i, j, k;
    clrscr();
    printf("1\n");
    for (j=1; j<=3; j++)
        for (k=1; k<=j; k++)
            printf("%d ", k);
    printf("\n");
}
```

Ans

```
main()
{
    int i, j, k;
    clrscr();
    printf("1\n");
    for (j=1; j<=3; j++)
        for (k=1; k<=j; k++)
            printf("%d ", k);
    printf("\n");
    for (j=1; j<=3; j++)
        for (k=1; k<=3-j; k++)
            printf("%d ", k);
    printf("\n");
    for (j=1; j<=3; j++)
        for (k=1; k<=3-j; k++)
            printf("%d ", k);
    printf("\n");
    getch();
}
```



Scanned with OKEN Scanner

classmate
Date _____
Page _____

Ex: Write a program to print -

```
4  
3 3 4 4  
2 2 2 2  
1 1 1 1 1 1  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
2 2 2 2 2 2  
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
main()  
{  
    int i, j, n, k;  
    clrscr();  
    printf("Enter the number :\n");  
    scanf("%d", &n);  
    for (i = n; i >= 1; i--)  
    {  
        for (j = n - i + 1; j <= 2 * i - 1; j++)  
            printf("%d ", j);  
        printf("\n");  
    }  
}
```



Scanned with OKEN Scanner

Ex:

```
printf("1m");
for(j=0; j<=n; j++)
{
    for(j=1; j<=j; j++)
        printf("%");
    printf("\n");
    for(k=2*(n-j)+1; k>=1; k--)
        printf("%");
    printf("\n");
}
```

E

```
getch();
f:
    if((c=='q')||(c=='Q'))
        break;
    if(c=='1')
        printf("1\n");
    else if(c=='2')
        printf("2\n");
    else if(c=='3')
        printf("3\n");
    else if(c=='4')
        printf("4\n");
    else if(c=='5')
        printf("5\n");
    else if(c=='6')
        printf("6\n");
    else if(c=='7')
        printf("7\n");
    else if(c=='8')
        printf("8\n");
    else if(c=='9')
        printf("9\n");
    else if(c=='0')
        printf("0\n");
    else
        printf("0/p\n");
    getch();
```

O/p → Enter the number : 3

3	9	8	7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---	---	---	---



Scanned with OKEN Scanner

Ex: Write a program to print:-

```
int i, j, n, k;
clrscr();
printf("Enter the row size : ");
scanf("%d", &n);
main()
{
    int i, j, n, k;
    clrscr();
    printf("Enter the row size : ");
    scanf("%d", &n);
    for (k = -n; k < = n; k++)
        printf("%c", k);
    for (i = -n; i < = n; i++)
        for (j = -n; j < = n; j++)
            printf("%c", i + j);
}
```

```

if( abx(i) );
printf("%d", abx(i));
else
printf("%n");
printf("%n");
}
for(k=-n; k<= -n; k++)
{
printf("1");
printf("\n");
}
for(k=-n; k=n+2; k++)
{
printf("1");
printf("\n");
}
getch();

```

T
D
O

卷之三

row size: 2

```

if( abx(i) == abx(j) )
    printf("%d", abx(i));
else
    printf("%n");
    printf("%d", abx(i));
    for( k=-n; k<-n; k++)
        printf("%1", "m");
    printf("%n");
    printf("%d", abx(i));
}

```

E:

```

for( k=-n; k=n+2; k++)
    printf("%1", "m");
    getch();
    printf("%d", abx(i));
    printf("%n");
    printf("%d", abx(i));
}

```

O/p →

Enter the row size: 20

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	2	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	



Scanned with OKEN Scanner

Eg. Write a program to print

3

2 1 1

0

1 1

2 1 1

3 3 3

main()

{

```
int i, j, n;
clrscr();
printf("Enter the row size : ");
scanf("%d", &n);
for (i = -n; i <= n; i++)
```

```
for (j = -n; j <= n; j++)
{
```

```
    if (abs(i) == abs(j))
        printf("%d", abs(i));
    else
        printf(" ");
}
```

```
printf("1m");
}
getch();
```

O/p → Enter the row size: 3

3

2

1

0

1

1

2

3

~~control looping statement.~~

Syntax - initialization;

```
do {  
    statement 1 ; ←  
    statement 2 ;  
    -----  
    -----  
    -----  
    -----  
    update ; } ←  
    while (condition);
```

Ex

First it will enter into the loop and execute the body atleast for one time and then it check the condition. if the condition is satisfied then it will re-enter into the loop. This process of loop rotation will be taken place until the condition is dis-satisfied.

In 'dowhile' the while will have the semicolon (;)



Difference between 'while' and 'do while' :-

'while' first check the condition and then enter into the body.

'do while' first enter into the body and then checks the condition.

Note :-

1. Inside the body 'while' will not present without semicolon (;) , it's only normal 'while' and after the body 'while' will present with semicolon (;) .
2. In 'dowhile' after the body is closed the next statement must be 'while' with condition including semicolon (;) .
3. In 'dowhile' within the body there is no while with condition including semicolon (;) .

Eg

```
int i=10;  
do {  
    printf("%d", --i);  
} while(--i);
```



Scanned with OKEN Scanner

```
getch();  
}  
}
```

O/p → 9 7 5 3 1

Ex. main () {
 int i=1; do

```
    {  
        while (i++ < 1);  
        while (i++ <= 2);  
        printf ("%d", i);  
    }  
    while (i++ <= 2)  
        getch(); printf ("%d", i);  
    }  
}
```

E

```
o/p → 5  
1  
2  
3  
4  
5
```

Ex. main () {
 int i=1; do

```
    {  
        while (i++ <= 1);  
        while (i++ <= 3);  
        while (i++ <= 5);  
        printf ("%d", i);  
    }  
    while (i++ <= 5)  
        getch(); printf ("%d", i);  
}
```

O/p → 7



Scanned with OKEN Scanner

Number System -:

Supports four types of formats. A number can be represented in any one of the below formats -

1. Binary format.
2. Octal format.
3. Hexadecimal format.
4. Decimal format (like integer)

1. Binary format :-

Base of Binary is 2. It contains 2 numbers that are 0 and 1. To represent binary data can be written with the help of 0's and 1's. Ex. 1010110, 10101. etc.

2. Octal format - It has base 8. It contains 8 numbers that are 0, 1, 2, 3, 4, 5, 6, 7. To represent octal data can be written by using the eight numbers.

If the number has to treat on octal, Define the number with 0 (zero). As per the 'C' Language, octal numbers are preceded by 0. So, some octal data can be written by using the eight numbers. If the number has to treat on octal, Define the number with 0 (zero).



Page No. _____ Date _____

Ques. 4. i) 0 4 5 6 , 0 7 6 , 0 4 5 6 etc.

Ex:

3. Hexadecimal format :- Give Base of Hexa-decimal is 16. and it contains 8 numbers. They are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Ex:
10, 11, 12, 13, 14, 15
↓ ↓ ↓ ↓ ↓
A B C D E F

Ex:

as shown above the hexadecimal format contains numbers from 0 to 9 and some alphabets like A, B, C, D, E, F. These alphabets are equivalent to 10, 11, 12, 13, 14 and 15 respectively.

If the per 'c' component is an number has to be treated as hexadecimal with '0x' (zero x). Ex. 0x 567 0x 989 0x 532 etc.



classmate
Date _____
Page _____

4. Decimal format :-
Decimal formats are integer numbers.
i.e.

Ex. 456, 8083, 53 etc.

Base of decimal is 10 and it contains 10 numbers. They are -
0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

By using these numbers we can represent any decimal number.

Note :-

In Binary format, we can write a binary data by using 0s (zero's) but it is optional.

Ex. i) 0b10100001010000000000000000000000

both are valid.

Ex. Convert into Hexa and octal
i). 100
ii). 30
iii). 20

Ex. i). 100 R
ii). 4
iii). 20

$$\begin{array}{r} 100 \\ \hline 12 \\ 8 \\ \hline 1 \\ 0 \end{array}$$

100 R

$$\begin{array}{r} 100 \\ \hline 6 \\ 4 \\ \hline 1 \\ 0 \end{array}$$

100 R

format + doer - not containing g.

it will show an error because octal

$$0 \ 1 \times 2 \ 4$$

$$Subtract 0 \ 7 \ 9 \ 4$$

$$5 - 2 = 3$$

$$18 - 10 = 8$$

$$2 - 0 = (16 + 2) - 0$$

$$18 - 12 = 6$$

$$16 + 2 = 18 - c$$

$$c = 0$$

$$\oplus \xrightarrow{\text{H}}$$

$$2 + 9 + 6 + 0$$

$$29 = 16 | 27$$

$$1 + 18 + 14 + 11$$

$$+ D + e + b$$

$$16 | 28$$

$$= 16 + 5$$

$$b + 5 + c = 11 + 5 + 12$$

$$\textcircled{1} \textcircled{2} \textcircled{3}$$

Ex. Add.

$$\begin{array}{r} \boxed{4} \\ \boxed{20} \\ \hline 0 \times 1 \quad 4 \quad D \quad C \\ 0 \times 1 \quad 9 \quad 3 \quad 6 \\ + 0 \times 1 \quad 2 \quad D \\ \hline 12 - C \end{array}$$

$$1 + 0 + 9 = 20$$

$$4 + 15 + 11 = 28$$

$$4 + D + 5$$

Add.

classmate

page

```
#include <stdio.h>
Ex: #include <conio.h>
     void main()
{
    clrscr();
    int a = 14;
}
```

```
printf("%d\n", a);
printf("%0m", a);
printf("%#0", a);
printf("%m %x", a);
printf("%m %#x", a);
printf("%m %#x", a);
printf("%m %b", a);
getch();
}
```

```
printf("%d\n", a);
printf("%0m", a);
printf("%#0", a);
printf("%m %x", a);
printf("%m %#x", a);
printf("%m %#x", a);
printf("%m %b", a);
getch();
}
```

```
O/p →
14
16
016
0xe
0xe
0xe
0xe
%
```

```
for printf("%m %b", a);  
for printf("%m %#x", a);
```



Binary codes :-

0	-	0000	00000000
1	-	0001	00000001
2	-	0010	00000010
3	-	0011	00000011
4	-	0100	00000100
5	-	0101	00000101
6	-	0110	00000110
7	-	0111	00000111
8	-	1000	00001000
9	-	1001	00001001
10	\rightarrow	1010	00001010
97	-	1100001	0001100001

Binary coding in increasing bits :-

$2^1 - 1$	=	1	\Rightarrow binary form 1
$2^2 - 1$	\Rightarrow	3	\Rightarrow binary form 11
$2^3 - 1$	\Rightarrow	7	\Rightarrow binary form 111
$2^4 - 1$	\Rightarrow	15	\Rightarrow binary form 1111
$2^5 - 1$	\Rightarrow	31	\Rightarrow binary form 11111
$2^6 - 1$	\Rightarrow	63	\Rightarrow binary form 111111
$2^7 - 1$	\Rightarrow	127	\Rightarrow binary form 1111111
$2^8 - 1$	\Rightarrow	255	\Rightarrow binary form 11111111

Ex. Add ① ① ① ① ①

$$\begin{array}{r} 1 1 1 \\ 1 1 0 1 \\ 1 0 1 0 1 \\ + 1 1 0 0 1 \\ \hline 1 0 0 1 0 0 1 \end{array}$$

Ex. Subtract
1101001 - 1010100

$$\begin{array}{r} 1 1 0 1 0 0 1 \\ - 1 0 1 0 1 0 0 \\ \hline 0 0 1 0 1 0 1 \end{array}$$

E

Ex. Calculate

$$Q = 10 + 010 + 0\text{x}10 + 0\text{x}10^2$$

First method :-

Convert all nos. into equivalent decimal nos & then add. & solve the expression.

$$\begin{aligned} Q &= 10 + (1 * 8^1 + 0 * 8^0) + (1 * 16^1 + 0 * 16^0) \\ &\quad + (1 * 16^1 + 0 * 16^0) \end{aligned}$$

$$\begin{aligned} Q &= 10 + 8 + 16 + 16 \\ Q &= 50 \end{aligned}$$





Second method :-

Convert all nos into its equivalent binary nos. and then add.

$$\alpha = 10 + 010 + 0 \times 10 + 0 \times 10$$

$$\begin{aligned}
 10 &\rightarrow 1010_2 \\
 010 &\rightarrow 0010_2 \\
 0 \times 10 &\rightarrow 00010_2 \\
 0 \times 10 &\rightarrow 000010_2
 \end{aligned}$$

Sum

$$\boxed{00110010_2}$$

Now convert it into its equivalent decimal.

$$\begin{aligned}
 \alpha &= (1 * 2^5) + (1 * 2^4) + (0 * 2^3) + (0 * 2^2) \\
 &\quad + (1 * 2^1) + (0 * 2^0)
 \end{aligned}$$

$$\alpha = 32 + 16 + 0 + 0 + 2 + 0$$

$$\alpha = 50. \text{ And } 10 \text{ ranging from } 0 \text{ to } 255.$$

Note :-

In the program in implementing if a variable is cleared then first of all it allocates some bytes of memory, for suppose char ch; (1 byte = 8 bits).

~~Character allocated~~ Since character memory -

1 byte of memory -
1 byte = 8 bits.

$$11111111 + 101 \rightarrow 2^8 = 255$$

Physical memory is created once you created a variable like above.

Note :-

Internally in the system everything is accepted and in the binary format and then its implementation will be takes place.

Ex:-
It can be represented either in 8 bit format or 16 bit format.
Better always a represent by 16 bit format.

Ex:-
0000 0000 → 0
0111 1111 0000 0000 → -1
1000 0000 1111 1111 → 32767
1111 1111 0000 0000 → -32768
→ -1

Ex:-
0000 0000 → 0
1111 1111 1111 1111 → 32767
1111 1111 0000 0000 → -32768
→ -1

Reverse Combination - bbb

1. If the given number is positive then the result will be negative by incrementing value for the positive number.

Ex:

$$9 = -10$$

$$9 \rightarrow 0000\ 0000\ 0000\ 1001$$

$$-10 \rightarrow 1111\ 1111\ 1111\ 0110$$

Ex:

$$4 = -5$$

$$4 \rightarrow 0000\ 0000\ 0000\ 0100$$

$$-5 \rightarrow 1111\ 1111\ 1111\ 1011$$

2. If the given number is negative then the result will be positive by decrementing 1 value for the negative number.

Ex:

$$-100 = 995$$

$$-100 \rightarrow 1111\ 1111\ 1001\ 1100$$

$$99 \rightarrow 0000\ 0000\ 0000\ 0110\ 0011$$

Ex: $-10 = 995$

$$-10 = 1111\ 1111\ 1111\ 10110$$

$$9 = 0000\ 0000\ 0000\ 1001$$