

Q Using \n & \t in c programming. O/T

```
#include <stdio.h>
int main()
{
    printf ("Hello World, \n This is my, \n first class");
}
```

Output:  
Hello World,  
This is my,  
First class

Q. # include (stdio.h)

```
{ int main()
{
    printf ("Welcome to \n\t CJA \t\t University");
}
```

O/T:  
Welcome to.  
CJA  
University

Q Addition of two nos. in C-programming.

```
#include<stdio.h>
int main()
{
    int a,b,c;
    a=10 ;
    b=40 ;
    c=a+b ;
    printf ("Total sum is : %d",c);
}
```

O/T  
Total sum is: 50

Q Printing diff. datatypes

```
#include <stdio.h>
int main()
{
    int x;
    float y;
    char z;
    x=10;
    y=10.5;
    z='A';
    printf ("%d",x);
    printf ("%f",y);
    printf ("%c",z);
}
```

O/T  
10  
10.5  
A

Ans. ####/####

####

Ans Printing 'A' in 'A'

#	#	#	#	#
#	#	#	#	#
#	#	#	#	#
#	#	#	#	#
#	#	#	#	#

Ans #include <stdio.h>  
int main()  
{  
 printf("# #####\n# \_ # \n#####\n# \_ # \n# \_ #");  
}

Q. Take two nos. from user and display the sum of them.

Ans #include <stdio.h>  
int main()  
{

int a,b,c;

printf("Enter first no.");

scanf("%d",&a);

printf("Enter second no.");

scanf("%d",&b);

c=a+b;

printf("Total Sum is %d",c);

}

Q Write a program to accept 2 nos. from user & calculate diff.

#include <stdio.h>

int main()

{ int a,b,c;

printf("Enter the 1<sup>st</sup> no.");

scanf("%d",&a);

printf("Enter the 2<sup>nd</sup> no.");

scanf("%d",&b);

c = a - b;

printf("The difference is %d",c);

}

Q Write a program to multiply two nos. taken from user.

```
#include<stdio.h>
int main()
{
    int a,b,c;
    printf("Enter the first no.");
    scanf("%d",&a);
    printf("Enter the 2nd no.");
    scanf("%d",&b);
    c=a*b;
    printf("The answer is %d",c);
}
```

Q Write a program to divide two nos. taken from user.

```
#include<stdio.h>
int main()
{
    int a,b,c;
    printf("Enter the first no");
    scanf("%d",&a);
    printf("Enter the second no");
    scanf("%d",&b);
    c=a/b;
    printf("The answer is %d",c);
}
```

Q Write a program to divide two nos. using single scanf.

```
#include<stdio.h>
int main()
{
    float a,b,c;
    printf("Enter two nos.");
    scanf("%f %f",&a,&b);
    c=a/b;
    printf("Result is %f",c);
}
```

Q Swapping the values of two nos. inc.

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

```
int main()
```

```
{
```

```
    int a,b,c;
```

```
    a=5;
```

```
    b=3;
```

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

```
    b=a-b;
```

```
    a=a-b;
```

```
    printf("\n a=%d",a);
```

```
    printf("\n b=%d",b);
```

```
}
```

Q Write a program to accept radius of circle & find their Area.

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

```
int main()
```

```
{
```

```
    float a,b;
```

```
    printf("Enter the radius of circle");
```

```
    scanf("%f",&a);
```

```
    b = 3.14 * a * a;
```

```
    printf("The area is %f",b);
```

```
}
```

Q Write a program to accept base & height & then find area of triangle.

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

```
int main()
```

```
{
```

```
    float a,b,h;
```

```
    printf("Enter the base & height");
```

```
    scanf("%f %f", &b, &h);
```

```
    a = 0.5 * b * h;
```

```
    printf("The area of Δ is %f");
```

```
}
```

Q Swapping using third variable.

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

```
{ int a,b,c;
```

```
a=4;
```

```
b=7;
```

```
c=b;
```

```
b=a;
```

```
a=c;
```

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

```
printf ("\n b=%d",b);
```

```
}
```

O/t

```
a=7
b=4
```

Q Finding the ASCII value of any character.

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

```
int main()
```

```
{
```

```
char x='A';
```

```
char y='a';
```

```
char z='1';
```

```
printf ("%c-->%d",x,x);
```

```
printf ("\n%c-->%d",y,y);
```

```
printf ("\n%c-->%d",z,z);
```

```
}
```

O/t

```
A-->65
a-->97
1-->49
```

Q Finding character using ASCII value?

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

```
int main()
```

```
{
```

```
int x=67;
```

```
int y=101;
```

```
int z=50;
```

```
printf ("%d-->%c",x,x);
```

```
printf ("\n%d-->%c",y,y);
```

```
printf ("\n%d-->%c",z,z);
```

```
}
```

O/t

```
67-->C
101-->e
50-->Z
```

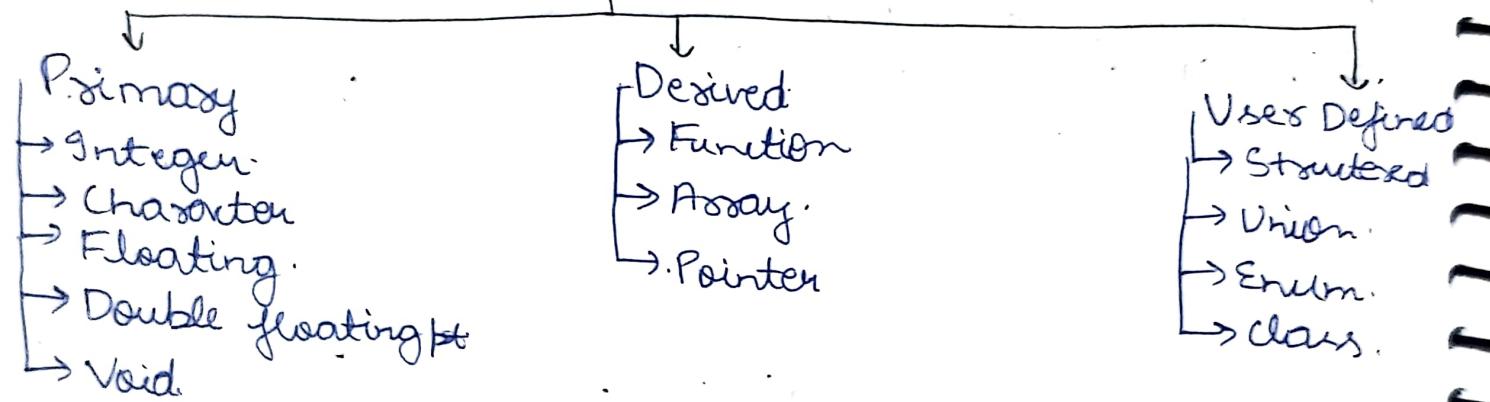
Q Using of 'size of' function.

```
#include < stdio.h>
int main() {
    printf("In size of char : %d", sizeof(char));
    printf("In size of short : %d", sizeof(short));
    printf("In size of int : %d", sizeof(int));
    printf("In size of float : %d", sizeof(float));
}
```

O/t

size of char : 1.
size of short : 2.
size of int : 4.
size of float : 4.

## Data - Types



Type	Byte	Access Specifier	Represent.
char	1	%c	character
short	2	%d	whole no.
int	2/4	%d	"
float	4	%f	Floating (decimal)
double	8	%lf	"
long int	4/8	%ld	whole no.
long double	16	%Lf	floating no.

\* Using **unsigned** function we can double the value in **unsigned int n=3;**

O/t =  $\Rightarrow n=6$ .

## # OPERATORS IN C :-

- i) Arithmetic - +, -, \*, /, %
- ii) Conditional -
- iii) Logical -
- iv) Assignment -
- v) Increment/Decrement -
- vi) Bitwise -
- vii) Ternary op -
- viii) Comparison -

### Arithmetic

- +  $\Rightarrow 2+3 = 7$
- -  $\Rightarrow 2-1 = 1$
- \*  $\Rightarrow 2*5 = 10$
- /  $\Rightarrow 10/2 = 5$
- %  $\Rightarrow \text{Remain} \Rightarrow 13 \% 5 \Rightarrow 3$

### Assignment Op :-

$$1. = ; \quad x=10$$

### Increment / Decrement Op :-

→ Pre increment / pre decrement { first 1 value will increment  
 int x, y; }  
 x=5; y= ~~5~~ ++x; // pre increment  
 printf(" x=%d", x); // x=6.  
 printf(" /n y=%d", y); // y=6

### # Bitwise Operators :-

- & - Bitwise AND Op
- | - Bitwise OR Op
- ^ - Bitwise XOR

### Q # include <stdio.h>

```
int main()
```

```
{
```

```
int x=13, y=22, a, b;
```

```
a=x&y;
```

```
b=x|y, c=x^y;
```

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

3  
}      `printf("%d", b);`

$x=13$        $\begin{array}{r} 32 \\ 16 \\ 8 \\ 4 \\ 2 \\ 1 \end{array}$   
 $y=22$        $\begin{array}{r} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{array}$   
 $a=x\&y$        $\begin{array}{r} 1 \\ 0 \\ 0 \end{array}$

Ans      4  
31  
27

$b=x|y$        $\begin{array}{r} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array}$

$c=x\wedge y$        $\begin{array}{r} 1 \\ 1 \\ 0 \\ 1 \end{array}$   
 $=27$

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

{  
    int y=22, x1, x2;  
    x1 = y  $\gg 1$ ; right shift op<sup>r</sup>       $y=22$        $\begin{array}{r} 32 \\ 16 \\ 8 \\ 4 \\ 2 \\ 1 \end{array}$   
    x2 = y  $\gg 2$ ;       $x1=y\gg 1$        $\begin{array}{r} 1 \\ 0 \\ 1 \\ 1 \end{array}$   
    printf("%d", x1); // 11       $x2=y\gg 2$        $\begin{array}{r} 1 \\ 0 \\ 1 \end{array}$   
    printf("%d", x2); // 5  
}

.....  
x1 = y  $\ll 1$       left shift op<sup>r</sup>       $y=22$        $\begin{array}{r} 64 \\ 32 \\ 16 \\ 8 \\ 4 \\ 2 \\ 1 \end{array}$   
x2 = y  $\ll 2$        $\begin{array}{r} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{array}$   
                         $O=22$   
                         $1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 = 44$   
                         $O=88$

Q # --  
int main()

{  
    int x, y, z;  
    y=100;  
    y=20;  
    z = x > y ? x : y; condition.  
    z = x > y ? x : y;      // Ternary Operator  
    True  
    False  
}

$1010 = 100$

Q Ternary operator for three variables:

#include <stdio.h>

int main()

{  
    int x=55, y=50, z=250, n;  
    n = (x>y) ? (x>z ? x : z) : (y>z ? y : z);  
    printf("Largest integer is %d", n);  
}

## # Decision Making statement :-

i) if statement  $\Rightarrow$  if (expression)  
                   {  
                       statement  
                   }

Q To print a check if given no. is even.

```
#include <stdio.h>
int main()
{
    int n=12;
    if (n%2==0); // 12%2 == 0
    {
        printf("Its an even no.");
    }
    else
    {
        printf("It is odd no.");
    }
}
```

Q To accept a no. from user & check whether its odd or even

```
#include <stdio.h>
int main()
{
    int x;
    printf("Enter the No.");  

    scanf("%d", &x);
    if (x%2==0)
    {
        printf("Its an even no.");
    }
    else
    {
        printf("Odd no.");
    }
}
```

Q To accept a no. from user & show which is greater

```
#include <stdio.h>
int main()
{
    int a, b;
    printf("Enter the two no.");
    scanf("%d %d", &a, &b);
    if (a>b);
    {
        printf("Larger no. is %d", a);
    }
}
```

```

else if (b > a)
{
    printf ("Larger no is %d", b);
}
else
{
    printf ("Both nos are equal");
}

```

Q1 Write a program to accept age & check whether eligible for voting.

- Q2 ----- & check eligible for emp. ( $>= 18 \& <= 60$ )
- Q3 ----- year & check leap year or not.
- Q4 ----- 3 no from user & check which is largest.

Ans1 #include < stdio.h >

```

int main()
{
    int x;
    printf ("Enter the age of the person:");
    scanf ("%d", &x);
    if (x >= 18)
    {
        printf ("Person is eligible to vote");
    }
    else
    {
        printf ("You are not eligible to vote");
    }
}

```

Ans2 #include < stdio.h >

```

int main()
{
    int x;
    printf ("Enter the age of the person:");
    scanf ("%d", &x);
}

```

```

if (x >= 18)
{
    if (x <= 60)
    {
        printf("Person is eligible");
    }
    else
    {
        printf("You are not eligible");
    }
}

```

Ans 3 #include <stdio.h>

```

int main()
{
    int x;
    printf("Enter the year: ");
    scanf ("%d", &x);
    if (x % 400 == 0)
    {
        printf ("%d is a leap year", x);
    }
    elseif (x % 100 == 0)
    {
        printf ("%d is not a leap year", x);
    }
    elseif (x % 4 == 0)
    {
        printf ("%d is a leap year", x);
    }
    else
    {
        printf ("%d is not a leap year", x);
    }
}

```

Ans 4 #include

```

int main()
{
    int x, y, z;
    printf("Enter the 3 nos: ");
    scanf ("%d%d%d", &x, &y, &z);
    n = (x > y) ? (x > z ? x : z) : (y > z ? y : z);
    printf ("Largest no. is %d", n);
}

```

Q Write a program & check whether given no. is +, - or 0

```
#include <stdio.h>
int main()
{
    int x;
    printf("Enter the number:");
    scanf("%d", &x);
    if (x > 0)
    {
        printf("%d is positive no.", x);
    }
    else if (x < 0)
    {
        printf("%d is negative no.", x);
    }
    else
    {
        printf("%d is zero.", x);
    }
}
```

Q Day selection Program using Switch Operator.

```
#include <stdio.h>
int main()
{
    int n;
    printf("Enter a no. from 1-7");
    scanf("%d", &n);
    switch (n)
    {
        case 1:
            printf("Today is 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;
        case 7:
            printf("Sunday");
            break;
    }
}
```

case 7:

```
    printf("Sunday");  
    break;
```

default:

```
    printf("Invalid choice.");  
}
```

}

Q Take operation from user and print result.

A #include <stdio.h>

```
int main().
```

{

```
    int a, b, n;
```

```
    printf("Enter the two nos.:");
```

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

```
    printf("Enter 1-addn, 2-subtraction, 3-multiplication,  
        4-division, 5-modulus");
```

```
switch(n).
```

{

case 1 :

```
    printf("The result is  $\frac{a+b}{.d}$ ", a+b);  
    break;
```

case 2 :

```
    printf("The result is  $\frac{a-b}{.d}$ ", a-b);  
    break.
```

case 3 :

```
    printf("The result is  $\frac{a \times b}{.d}$ ", a*b);  
    break;
```

case 4 :

```
    printf("The result is  $\frac{a}{.d}$ ", a/b);  
    break;
```

case 5 :

```
    printf("The result is  $\frac{a}{.d}$ ", a/.d);  
    break.
```

Default:

```
case 6 : printf("Invalid choice");  
    break.
```

}

Q Take me from ~~which is positive~~ which is positive show message until it ~~gives~~ gives positive.

#include <stdio.h>

```
int main()
{
```

```
    int n;
```

```
    printf("Enter the no.");
```

Positive:

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

```
    if (n < 0) {
```

```
        printf("It's -ve, Please enter the No.");
```

```
    } goto positive;
```

```
    printf("It's +ve");
```

Level Goto function

Q Infinite loop program.

#include <stdio.h>

```
int main()
{
```

```
    int n=1;
```

abc:

```
    if (n <= 10)
```

```
{
```

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

```
        goto abc; }
```

```
}
```

Q Display all even nos. from 1 to 100.

#include <stdio.h>

```
int main()
{
```

```
    int n=1;
```

even:

```
    if (n <= 100) {
```

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

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

3

    n++;  
    goto even;

3

## # LOOP statements:

Q Display all no 1-20 using while loop.

```
#  
int main()  
{  
    int i=1;  
    while (i<=20)  
    {  
        printf ("%d", i);  
        i++;  
    }  
}
```

Q Display all no 50-1 using while loop.

```
#  
int main()  
{  
    int i=50;  
    while (i>0)  
    {  
        printf ("%d", i);  
        i--;  
    }  
}
```

Q Display all no 1-20 using do-while loop.

```
#  
int main()  
{  
    int i=1;  
    do  
    {  
        printf ("%d", i);  
        i++;  
    } while (i<=20);  
}
```

Q Display all no 1-20 using for loop.

```
#  
int main()  
{  
    for (int i=1; i<=20; i++)  
}
```

# int main()

```

    {
        int i;
        for (i=1, i<=10, i++)
        {
            if (i==5)
            {
                continue;
            }
            printf("%d", i);
        }
    }

```

0/10  
1 2 3 4 6 7 8 9 10

Q

# int main()

```

    {
        int i;
        for (i=1, i<=10, i++)
        {
            if (i==5)
            {
                break;
            }
            printf("%d", i);
        }
    }

```

0/4  
1 2 3 4

4 0 0 5

b = 5  
b = 10

Q1 Accept a no. & from user & count how many digits are there.

Q2 Accept a no. & from user & calculate sum of digit

Q3 Accept a no. from user and reverse the number.

Q4 Accept a no. from user and check whether it is pallindrome

Ans1 #

```

    {
        int n, c=0;
        printf("Enter the no. ");
        scanf("%d", &n);
        while (n>0)
        {
            c++;
            n=n/10;
            printf("Result is %d", c);
        }
    }

```

Ans 2 #

int main().

```
{
    int n, a, b=0;
    printf("Enter the no: ");
    scanf("%d", &n);
    while (n>0)
    {
        a+b n;
        b = n%10;
        a = b; n=n/10;
        printf("Result is %d", a);
    }
}
```

Ans 3 #

int main()

```
{
    int n, sum=0;
    printf("Enter the no. ");
    scanf("%d", &n);
    while (n>0); //n=123
    {
        d = n%10;
        sum = sum * 10 + d; // (321)
        n = n/10;
    }
    printf("%d", sum);
}
```

```

# Math.h as a header.
#include <stdio.h>
#include <math.h>
int main()
{
    float x;
    printf ("%f", sqrt(20));
    printf ("%f", pow(5,7));
    printf ("%f", fmod(15.7, 3.1));
    printf ("%f", log10(31));
    printf ("%f", ceil(3.7)); // 4
    printf ("%f", floor(3.1)); // 3
    printf ("%f", round(3.4)); // 3
    printf ("%f", round(3.5)); // 4
    printf ("%f", round(3.9)); // 4
    printf ("%f", abs(-12)); // 12.

```

Q Write a program for finding roots of the eqn.

```

#include <stdio.h>
#include <math.h>
int main()
{
    float a=2, b=3, c=1, x1, x2;
    x1 = (-b + sqrt(b*b - 4*a*c)) / (2*a);
    x2 = (-b - sqrt(b*b - 4*a*c)) / (2*a);
    printf ("%f", x1);
    printf ("%f", x2);

```

Q Take a no from user and show whether its prime or not.

```

int main()
{
    int n, i, c=0;
    scanf ("%d", &n);
    for ((= 2; i <= n-1; i++)) {
        if (n % i == 0)
            c = 1
    }
}
```

```

    }  

    if (c == 0) {  

        printf ("prime no.");  

    }
}

```

```

else {  

    printf ("Not prime");  

}
}

```

\* We can replace  $i \leq n/2$  and can use 'break' statement after  $c=1$

Q Check whether the no is a perfect number or not.

```

#include <stdio.h>  

int main()  

{  

    int n, sum=0, i;  

    scanf ("%d", &n);  

    for (i=1; i <= n/2; i++) {  

        if (n % i == 0) {  

            sum = sum + i;  

        }
    }
    if (sum == n)  

    {  

        printf ("It's perfect no.");  

    }
    else {  

        printf ("Not perfect no");  

    }
}

```

Q Check whether the number is Armstrong or not.

$$\text{Ex: Armstrong} = 153 = 1^3 + 5^3 + 3^3$$

$$1634 = 1^4 + 6^4 + 3^4 + 4^4$$

```

#include <stdio.h>  

#include <math.h>  

int main()  

{  

    int n, d, sum=0, r;  

    printf ("Enter no.");  

    scanf ("%d", &n);  

    d = log10(n) + 1;  

    temp = n;  

    for (i=0; i < d; i++) {  

        r = temp % 10;  

        sum = sum + r;  

        temp = temp / 10;  

    }
    if (sum == n)
        printf ("It's Armstrong no.");
    else
        printf ("Not Armstrong no.");
}

```

```

while (n>0) {
    r=n%10;
    sum=sum+pow(r,d);
    n=n/10;
}
if (sum==temp) {
    printf("Armstrong no.");
}
else {
    printf("Not Armstrong");
}

```

Q1  $1+4+9+16+25 = \dots = 100$

Ans #include <stdio.h>  
int main().{

```

int i, t, sum=0;
for(i=1; i<=10, i++) {
    t=i*i;
    sum=sum+t;
    sum+=t;
}
printf("The sum is: %.d", sum);
}

```

Q2  $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{10} = ?$

Ans #include <stdio.h>  
int main() {

```

int n, t, sum=0;
for(n=0; n<=10, n++) {
    t=1/n;
    sum+=t;
}
printf("The sum is: %.10f", sum);
}

```

$$Q4. 1 + 1 + 2 + \frac{1}{2} + 3 + \frac{1}{3} + 4 + \frac{1}{4} \dots 10 + \frac{1}{10} = ?$$

Ans #include <stdio.h>

int main() {

    int i, t, sum = 0;

    for (i = 1, i <= 10, i++) {

        t = i + 1 / i;

        sum += t;

    } printf ("The sum is : %d", sum);

}

$$\text{Q5. } 1 + \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \frac{4}{5}$$

Ans #include <stdio.h>

int main() {

    int i, t, sum = 1;

    for (i = 1, i <= 10, i++) {

        sum = sum + (i / (i + 1));

}

    printf ("The sum is : %d", sum);

}

Q6 Display sum of factorial b/w 1-10

#include <stdio.h>

#int main().{

    int i, j, sum = 0, t = 1;

    for (i = 1, i <= 10, i++) {

        for (j = 1; j <= i, j++) {

            t = t \* j;

        sum = sum + t;

    } t = 1;

}

} printf ("The sum is : %d", sum);

Q7. Sum of perfect no. b/w 5-500.

Ans C/C++

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

int i, j, t = 0, sum = 0;

for (i = 5; i <= 500; i++) {

for (j = 1; j <= (i/2); j++) {

if (i % j == 0) {

sum = sum + j;

}

if (sum == i) {

~~t=t~~

}

~~t=t+i~~

3.

sum = 0

printf("Sum is: %.d", t);

}

Q8. # Frequency of each digit in given number

Ans #include <stdio.h>  
int main() {

int n, sum = 0, i, f, r, temp;

printf("Enter number: ");

scanf("%d", &n);

~~n=temp~~  
~~temp=n~~  
~~c=0~~

for (i = 0, f <= 9, i++) {

    n = temp;

    c = 0;

    while (n > 0) {

        r = n % 10;

        if (r == c) {

            c++;

        }

        n = n / 10;

    } printf("\n%d -> %d", f, c);

}

## L.C.M & H.C.F

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

```
int main() {
```

```
    int x, y, gcd, lcm, a, b;
```

```
    printf("Enter two integers: (n);
```

```
    scanf("%d %d", &x, &y);
```

```
    a = x;
```

```
    b = y;
```

```
    while (b != 0) {
```

```
        t = b;
```

```
        b = a % b;
```

```
        a = t;
```

```
    }  
    gcd = a;
```

```
    lcm = (x * y) / gcd;
```

```
    printf("GCD of %d and %d = %d\n", x, y, gcd);
```

```
    printf("LCM of %d and %d = %d\n", x, y, lcm);
```

```
}
```

## Fibonacci Series

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

```
int main() {
```

```
    int n1 = 0, n2 = 0, n3;
```

```
    int i;
```

```
    for (i = 1; i <= 8, i++) {
```

```
        n3 = n1 + n2;
```

```
        printf("%d", n3);
```

```
        n1 = n2;
```

```
        n2 = n3;
```

```
}
```

```
}
```

# Deleting an element from array?

# include < stdio.h >

int main() {

int arr[] = {33, 22, 44, 55, 6, 14, 3, 22, 3};

int i, n, c = -1;

{ printf("Enter no to delete:");

scanf("%d", &n);

for (i=0; i<7; i++) {

if (n == arr[i]) {

c = i;

break;

}

if (c != -1) {

for (i=c; i<6; i++) {

arr[i] = arr[i+1];

}

arr[7] = '0';

}

else {

printf("Element not found\n");

}

for (i=0; i<7; i++) {

printf("%d", arr[i]);

}

}

# Insert an element in an array?

# include < stdio.h >

## Arrays

- An array is a collection of similar type of data types items stored at contiguous memory locations.
- It has two types -
  - i) 1-Dimension array -
  - ii) 2-Dimension array.

Ex: #

```
int main {
```

```
    int arr[5] = {12, 43, 54, 76, 89};  
    printf ("%d", arr[0]);  
    " (%d", arr[3]);  
}
```

Output

12 76.

$\boxed{arr[5] = \{2, 4, 6, 8, 16\}}$  → Memory allocation & initialization  
[2 4 6 8 16] → Array elements.

Index → 0 1 2 3 4

Q: Display element of array using for loop.

Ans: #include <stdio.h>

```
int main () {
```

```
    int arr[] = {12, 43, 54, 76, 89};
```

```
    int i;
```

```
    for (i=0; i<=4; i++) {
```

```
        printf ("%d", arr[i]);
```

```
}
```

Output: 12, 43, 54, 76, 89

## Advantages of the array:

- i) Accessing an element is very easy by using the index number.
- ii) The search process can be applied to an array easily.
- iii) 2-D array is used to represent matrices.
- v) Array has low overhead.

## Disadvantages of the array:

- vi) Array size is fixed: The array is static, which means its size is always fixed. The memory which is allocated to it cannot be increased or decreased.
- vii) Array is homogeneous: The array is homogeneous i.e. only one type of value can be stored in the array.

Create an array & display its elements.

```
#include <stdio.h>
int main() {
    int n;
    printf("Enter size of an array:");
    scanf("%d", &n);
    int arr[n];
    int i;
    for (i=0; i<=n-1; i++) {
        printf("Enter number:");
        scanf("%d", &arr[i]);
    }
    for (i=0; i<=n-1; i++) {
        printf("\n%d", arr[i]);
    }
}
```

Ques = {32, 21, 22, 54, 66, 77, 88, 70, 54, 32, 11}

- i) Reverse Order
- ii) Sum
- iii) Even number display.
- iv) No. of even & odd terms
- v) Maximum number

```
#include <stdio.h>
int main() {
    int arr[] = {32, 21, 22, 54, 66, 77, 88, 70, 54, 32, 11};
    int n = sizeof(arr)/4;
    int i, t=0;
    printf("The reverse order is:");
    for (i=n-1; i>=0; i--) {
        printf("\n%d", arr[i]);
    }
    printf("\n\nIn the even nos are:");
    for (i=0; i<=n-1; i++) {
        if (arr[i] % 2 == 0) {
            t=t+1;
            printf("\n%d", arr[i]);
        }
    }
    int sum=0;
    for (i=0; i<=n-1; i++) {
        sum = sum + arr[i];
    }
    printf("\n\nThe sum is %d", sum);
    printf("\n\nIn the no. of even terms", t);
    printf("\n\nIn the no. of odd terms", n-t);
    int max=0;
    for (i=0; i<=n-1; i++) {
        if (arr[i] > max) {
            max=arr[i];
        }
    }
    printf("\n\nThe max. number is %d", max);
}
```

+ Inserting character in array.

#  
int main(){}

```
char arr[5] = {'a', 'e', 'i', 'o', 'u'};  
int i;  
for(i=0; i<=4; i++) {  
    printf("%c", arr[i]);  
}
```

Inserting float -

```
float arr[5] = {12.7, 19.723, 7.711, 3.14};
```

Q Find the element in given array available or  
not, also find position.

#  
int main{}

```
int arr[] = {23, 21, 40, 49, 23, 45, 0, 0, 80};  
int n = sizeof(arr)/4;  
int num;  
printf("Enter the no. you want to find");  
scanf("%d", &num);  
for(i=0; i<=n-1; i++) {  
    if(arr[i] == num) {  
        printf("Number available");  
        printf("Position is %d", i+1);  
    }  
}
```

3 3

3

# Bubble sorting Program

# include <stdio.h>

int main{

```
int i, j, t;  
int arr[] = {32, 12, 54, 45};  
int s = sizeof(arr)/sizeof(int);  
for(i=0; i<s; i++) {  
    for(j=1; j<s-i; j++) {  
        if(arr[j] < arr[j-1]) {  
            t = arr[j];  
            arr[j] = arr[j-1];  
            arr[j-1] = t;  
        }  
    }  
}
```

```
for(i=0; i<s; i++) {  
    printf("%d", arr[i]);  
}
```

3

3

3

3

3

3

3

## 2-D Array

Q Write a program to create an 2-D array.

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

```
int main() {
```

```
    int arr[3][3];
```

```
    int i, j;
```

```
    for (i=0; i<=2; i++) {
```

```
        for (j=0; j<=2; j++) {
```

```
            printf("Enter number:");
```

```
            scanf ("%d", &arr[i][j]);
```

```
}
```

```
}
```

```
for (i=0; i<=2; i++) {
```

```
    for (j=0; j<=2; j++) {
```

```
        printf ("%d", arr[i][j]);
```

```
}
```

```
}
```

\* For user desired rows and columns replace 3x3 with r & c. & Take r & c using scanf function.

5	11	9,
3	2	7,
1	2	6

① Replace all odd with zero,  
② Replace all no. with 0, except diagonals

③ sum of all rows - {Output : 5 4 9 , 18}

④  $A_1 = [ ]_{3 \times 3} \quad A_2 = [ ]_{3 \times 3} \quad A_3 = A_1 + A_2$

# Insertion &/ Declaration of elements in 2-D array

Ans 1

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

```
int main() {
```

```
    int arr[3][3] = {{2, 1, 3}, {3, 4, 5}, {7, 6, 5}};
```

```
    int i, j;
```

```
    for (i=0; i<=2; i++) {
```

```
        for (j=0; j<=2; j++) {
```

```
            printf ("%d", arr[i][j]);
```

```
            printf ("\n")
```

```
}
```

```
}
```

Ans ② #include <stdio.h>

```
int main() {
```

```
    int arr[3][3] = {{5, 4, 9}, {3, 2, 7}, {1, 2, 6}};
```

```
    int i, j;
```

```
    for (i=0; i<=2; i++) {
```

```
        for (j=0; j<=2; j++) {
```

```
            if (arr[i][j] % 2 != 0) {
```

```
                arr[i][j] = 0;
```

```
}
```

```
3
```

```
3
```

```
3
```

Print the array.

Ques ② Insert a statement after '\*'

```
if (i!=j){ }
```

Ques ③ #include <stdio.h>

```
int main(){
    int arr[3][3] = {{5,4,9},{3,2,7},{1,2,6}};
    int i=0, j=0, s=0;
    for (i=0; i<2, i++){
        s=0;
        for (j=0; j<2, j++){
            printf ("%d", arr[i][j]);
            s+=arr[i][j];
        }
        printf ("\n");
    }
}
```

Ques ④  $A_3[i][j] = A_2[i][j] + A_1[i][j]$

Ques ⑤ C program to print lower diagonal of matrix

Ques ⑥ C program to check two matrices are identical

Ques ⑦ C program to interchange rows & columns

Ques ⑧ Frequency of even numbers.

## POINTERS

\* Pointer is a variable that stores the memory address of another variable or its value

\* A pointer variable points to a data type (int, float, char, etc.) of the same type & it is created with a (\*) asterisk operator

name of the variable      10 → value  
xyz → address of the variable  
 ptos xyz  
 abc

#include <stdio.h>

int main()

```
int x;
x=10;
int *ptr;
ptr=&x;
printf ("x=%d", x);
printf ("Address of x=%u", ptr);
printf ("Value of x using ptr %d", *ptr);
```

int main()

```
char n='A', *cptos;
cptos=&n;
```

n    'A'  
 xyz

```
printf ("%c", n); - A
```

```
printf ("%u", cptos); - xyz
```

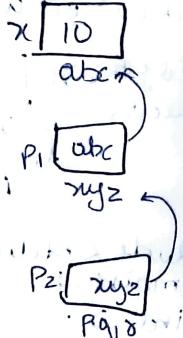
```
printf ("%c", *cptos); - A
```

```
printf ("%u", &cptos); - abc
```

cptos    xyz  
 abc

## # Pointers + Pointers

```
int main() {  
    int x=10, *P1, **P2  
    P1=&x;  
    P2=&P1;  
    printf("x.d", x); - 10  
    printf("y.u", P1); - abc  
    printf("y.d", *P1); - 10  
    printf("y.u", P2); - xyz  
    printf("%u", *P2); - abc  
    printf("%d", **P2); - 10  
    printf("%u", &P2); - p100
```



## FUNCTIONS

- A function is a block of reusable code which only runs when it is called.
- You can pass data known as parameters into a function.
- Functions are used to perform certain actions and they are important for reusing code. Define the code once, and use it many times.

There are two types of functions:

- i) Predefine function
- ii) User define function.  
'Void' is a key

## Syntax of function:

return-type function-name(list, parameter)

Condition for functions:

- i) Function declaration
- ii) Function calling.
- iii) Function definition.

Type of User-defined function:

- i) No return and no parameter.
- ii) No return and with parameter.
- iii) With return and no parameter.
- iv) With return and with parameter.

### No return and no-parameter:

```
void show(); // function declaration.
int main() {
    show(); // function calling
    printf("In thanks");
    show();
}
```

```
3
void show() {
    printf("In Hello world");
}
```

### No return and with parameter:

```
#include <stdio.h>
void addition(int, int);
void sub(int, int);
int main() {
    int a=12, b=34;
    addition(a, b);
    sub(a, b);
}
```

```
3
addition
void addition(int x, int y) {
    int sum=x+y;
    printf("\n & addition is: %d", sum);
}

3
void subtraction
void subtraction(int x, int y) {
    int s = x-y;
    printf("\n subtraction is: %d", s);
}
```

### With return with parameter:

```
#include <stdio.h>
int addition (int, int);
int main() {
    int a=addition(12, 34);
    printf("In addition is: %d", a);
}
```

```
int b=addition (10,30);
printf("In addition is: %d , b);
```

```
3
int addition (int x, int y) {
    int sum=x+y;
    return sum;
}
```

### With return and no-parameter:

```
#include <stdio.h>
int show();
int main() {
    int s = show();
    printf("Addition is: %d", s);
}

int show() {
    int x, y;
    printf("Enter two no. \n");
    scanf("%d %d", &x, &y);
    return x+y;
}
```

### Multiplication of Matrix:

```
#include <stdio.h>
int show();
int main() {
    int A[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    int B[3][3] = {
        {4, 3, 2},
        {7, 8, 9},
        {1, 2, 3}
    };
}
```

```
int M[3][3] = { i, j, k;
    for (i=0; i<=2, i++) {
```

```
for (j=0; j<2; j++) {
```

```
    M[3][j] = 0;
```

```
    for (k=0; k<2; k++) {
```

```
        M[3][j] = M[i][j] + A[i][k] * B[k];
```

```
}
```

```
}
```

```
for (i=0; i<2; i++) {
```

```
    for (i=0; i<2; i++) {
```

```
        printf(" %d", M[i][j]);
```

```
    }
```

```
}
```

```
}
```

## Passing an element to an Array:

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

```
void displayArrayElement (int arr[], int s);
```

```
int main() {
```

```
    int arr[] = {33, 22, 44, 55};
```

```
    int s = sizeof(arr) / sizeof(int);
```

```
    displayArrayElement (arr, s);
```

```
}

void displayArrayElement (int arr[], int s) {
```

```
    for (int i=0; i<s; i++) {
```

```
        printf(" %d", arr[i]);
```

```
}
```

## Variables

### # Global variable:

→ Declared outside function block.

→ Accessible everywhere.

→ Global variable is destroyed only when a program is terminated.

### # Local Variable (Automatic variable),

→ Declared inside function body.

→ Accessible only in the function.

→ Local variable is created when a function is called and is destroyed when a function returns.

### # Global ~~variable~~ variable using program:

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

```
void display();
```

```
int a = 224; //global variable.
```

```
int main() {
```

```
    int n = 12;
```

```
    if (1 == 1) {
```

```
        int x = 23; //local variable
```

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

```
        printf(" \n x=%d", x);
```

```
    } printf(" \n inside if block a=%d", a);
```

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

```
    printf(" \n inside main Function a=%d", a);
```

```
    display();
```

```
void display() {
```

```
    printf(" \n inside Function a=%d", a);
```

```
}
```

# Call by value :-

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



```
        int n=12;
```

```
        printf ("\\n before function n=%d", n);
        change(n);
```

```
        printf ("\\n after function n=%d", n);
```

```
}
```

```
void change (int n) {
```

```
    n=n+10;
```

```
    printf ("\\n inside function n=%d", n);
```

# Call by reference :-

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

```
void change (int *);
```

```
int main() { int n=12;
```

```
    printf ("\\n before function n=%d", n);
```

```
    change (&n); // actual Parameter
```

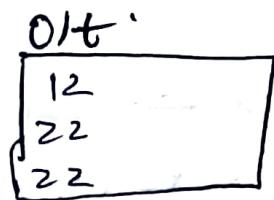
```
    printf ("\\n after function n=%d", n);
```

```
}
```

```
void change (int *p) { // formal Parameter
```

```
    *p = *p + 10;
```

```
    printf ("\\n inside function n=%d", *p);
```



Ex:- Global scope & Local scope.

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

```
int x=32; -②
```

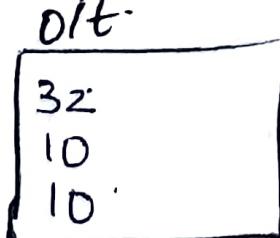
```
void show();
```

```
int main() {
```

```
    printf ("\\n", x);
```

```
    show();
```

```
    printf ("\\n", x);
```



## String

String function in string.h header file.

```

char ch1[44]
char ch2[64]
gets(ch1); // gla123
gets(ch2);
* strlenc(ch1); // 5
* strcpy(ch2, ch1); // ch2=ch1 string copy
* strcmp(ch1, ch2); // ch1==ch2 return 0 if both
        are equal otherwise return non zero value.
* strcat(ch1, ch2); // ch1=ch1+ch2.
* strrev(ch1); // ch1= 3210lg.
* toupper(ch1); // uppercase
    tolower(ch1); // lowercase.

```

Ex: #include <stdio.h>

#include <string.h>

int main{

```

char ch1[44], ch2[64];
printf("c1=");
gets(ch1);
int n = strlen(ch1);
printf("\n ch1 is: %.s", ch1);
printf("\n length is: %d", n);

```

3

Ex: COPY Function.

```

#include <stdio.h>
#include <string.h>
int main{.

```

```

char ch1[44], ch2[64];
printf("c1=");
gets(ch1);
strcpy(ch2, ch1);
printf("In ch1= %.s", ch1);
printf("In ch2= %.s", ch2);

```

Ex: COMPARE FUNCTION:

```

int main{
char ch1[44], ch2[64];
printf("ch1=");
gets(ch1);
printf("ch2=");
gets(ch2);
int n = strcmp(ch1, ch2);
if(n==0){
    printf("Both are equal");
} else {
    printf("Both are not equal");
}

```

#concat Function:

#include <stdio.h>

#include <string.h>

```

int main{char ch1[32], ch2[33];
printf("ch1=");
gets(ch1);
printf("ch2=");
gets(ch2);
strcat(ch1, ch2);
printf("In ch1= %.s", ch1);
printf("In ch2= %.s", ch2);

```

3

printf("Now reverse of ch1 is %.s", ch1);

O/t:

ch<sub>1</sub> = hi hello.  
 Now ch<sub>2</sub> = hello. of ch<sub>2</sub> ~~will be~~ Olleh ih  
**# LOWER & UPPER.**

```
int main(){
    char ch1[22], ch2[33];
    printf("ch1=%s", ch1);
    gets(ch2);
    printf("ch2=%s", ch2);
    gets(ch2);
    strlwr(ch2);
    strupr(ch2);
    printf("\n ch1 = %s", ch1);
    printf("\n ch2 = %s", ch2);
}
```

3

Q Write a program to find no. of capital, small.  
 letter, digits, special character.

```
int main{
    char email[44];
    printf("email = ");
    gets(email);
    int i, c=0, s=0, d=0, sp=0, len = strlen(email);
    for(i=0; i<len; i++){
        if(email[i] >='A' && email[i] <='Z'){
            c++;
        }
        else if(email[i] >='a' && email[i] <='z'){
            s++;
        }
        else if(email[i] >='0' && email[i] <='9'){
            d++;
        }
        else {
            sp++;
        }
    }
}
```

```
printf("\n no. of capital letter is : %d", c);
printf("\n no. of small letter is : %d", s);
printf("\n no. of digits is %d", d);
printf("\n no. of special character %d", sp);
```

3

## Dynamic Memory Allocation

```
#include < stdio.h>
#include < string.h>
#include < stdlib.h>
int main() {
    char * name;
    name = (char *) malloc(20 * sizeof(char));
    strcpy(name, "MLA University");
    printf("\n name is: %s", name);
    free(name);
    printf("\n name is: %s", name);
}

#ii - -
int main() {
    char * name;
    name = (char *) malloc(20 * sizeof(char));
    strcpy(name, "MLA University");
    printf("\n name is: %s", name);
    name = (char *) realloc(name, 100 * sizeof(char));
    strcpy(name, "MLA University MLAU University 0000000000");
    printf("\n Resized memory is: %s\n", name);
}

```

## on Array of Pointer

```
#include < stdio.h>
int main() {
    int arr[] = {12, 12, 43, 44, 54};
    int i, * ptr[5];
    for (i=0; i<5; i++) {
        ptr[i] = &arr[i];
    }
    for (i=0; i<5; i++) {
        printf("\n %d", *(ptr[i]));
    }
    printf("\n %d", ptr[0]);
}

```

## Point to Array

```
#include < stdio.h>
int main() {
    int (*a)[5];
    int b[5] = {1, 2, 3, 4, 5};
    int i=0;
    a = &b;
    for (i=0; i<5; i++) {
        printf("\n %d", (a+i));
    }
}

```

## Pointing array elements using pointer

```
#include < stdio.h>
int main() {
    int arr[] = {33, 44, 55, 66, 77};
    int i, * ptr = arr; // *ptr = &arr;
    for (i=0; i<5; i++) {
        printf("\n %d", *(ptr+i));
    }
}

```

$\begin{matrix} 33 \\ 44 \\ 55 \\ 66 \\ 77 \end{matrix} \rightarrow \begin{matrix} 33 \\ 44 \\ 55 \\ 66 \\ 77 \end{matrix}$

Larger number among three numbers.

```
#include <stdio.h>
int main(){
    int a, b, c;
    int *p1, *p2, *p3;
    printf("Enter three numbers : \n");
    scanf("%d %d %d", &a, &b, &c);
    p1 = &a;
    p2 = &b;
    p3 = &c;
    if (*p1 > *p2 && *p1 > *p3){
        printf("Larger number : %d", *p1);
    } else if (*p2 > *p1 && *p2 > *p3){
        printf("Larger number : %d", *p2);
    } else if (*p3 > *p1 && *p3 > *p2){
        printf("Larger number : %d", *p3);
    } else {
        printf("None is the larger.");
    }
}
```

Finding Factorial using pointer.

```
#include <stdio.h>
void findfact(int, int* );
int main(){
    int fact;
    int num1;
    printf("Input a number : ");
    scanf("%d", &num1);
    findfact(num1, &fact); // find fact(5)
    printf("The factorial is %d", fact);
}
```

```
void findfact (int n, int* f) {
    int i;
    *f = 1;
    for (i=1; i<=n; i++){
        *f = *f * i;
    }
}
```

Find largest in an array using pointer

```
#include <stdio.h>
void findlargest (int* arr, int N) {
```

```
    int i;
    for (i=1; i<N; i++){
        if (*arr < *(arr+i)){
            *arr = *(arr+i);
        }
    }
    printf(" %d", *arr);
}
```

```
int main() {
```

```
    int i, N=7;
    int arr [ ] = {22, 77, 33, 66, 54, 343};
    int * arr = arr;
    findlargest (arr, N);
}
```