

### 1.) Finding Area and Perimeter of a circle:

```
/* Program to calculate area and perimeter of a circle*/
#include <stdio.h>
void main()
{
    float r, area, perimeter;
    printf("Enter radius of circle \n");
    scanf("%f", &r);
    perimeter = 2 * 3.14 * r;
    area = 3.14 * r * r;
    printf("Area of circle is %f", area);
    printf("\n perimeter / circumference is %f", perimeter);
}
```

#### Output:

## 2.) Menu Driven Program Performing Arithmetic Operations:

```
/* Menu driven program to perform arithmetic operations using switch ( ) statement*/
#include <stdio.h>
void main ()
{
    int a, b, sub, mul, add, div, mod, opt;
    printf("Menu:\n 1.Addition\n 2. Subtraction \n 3.Multiplication \n 4.Division \n 5.Modulo \n");
    printf("Enter option from menu: \n");
    scanf("%d", &opt);
    printf("\n Enter two nos:");
    scanf("%d %d", &a, &b);
    switch (opt)
    {
        case 1: add= a + b;
        printf("\n sum is=%d", add);
        break;
        case 2: sub= a - b;
        printf ("\n difference is=%d", sub);
        break;
        case 3: mul= a * b;
        printf ("\n Product is=%d", mul);
        break;
        case 4: div= a / b;
        printf ("\n Quotient is=%d", div);
        break;
        case 5: mod= a % b;
        printf ("\n Reminder is=%d", mod);
        break;
        default: printf ("\n Wrong option!!!");
        break;
    }
}
```

### 3.) Printing of Natural numbers in reverse order:

```
/* Program to print natural numbers in reverse order using a 'for' loop*/
#include <stdio.h>
void main ()
{
    int n, i;
    printf ("Enter the value of n: \n");
    scanf ("%d", &n);
    printf ("\n Natural Numbers \n");
    for (i=n; i>=1; i--)
    {
        printf ("%d \t", i);
    }
}
```

#### Output:

#### 4.) Finding Fibonacci Series for a number:

```
/* Program to print the Fibonacci series*/
#include <stdio.h>
void main ()
{
    int i, n, a=0, b=1, c;
    printf ("Enter no of terms to be printed: \n");
    scanf ("%d", &n);
    printf ("\n Fibonacci Series \n");
    for (i=0; i<n; i++)
    {
        printf ("%d \t", a);
        c = a + b;
        a = b;
        b = c;
    }
}
```

#### Output:

### 5.) Counting of number of Digits in a Number:

```
/* Program to count the no. of digits in a number*/
#include <stdio.h>
void main ()
{
    long int num, count = 0;
    printf ("Enter a number: \n");
    scanf ("%ld", &num);
    while (num > 0)
    {
        num = num/10;
        count++;
    }
    printf ("Number of digits in the no are =%ld", count);
}
```

#### Output:

### 6.) Sum of Individual Digits in a Number:

```
/* Program to find the sum of individual digits in a number*/
#include <stdio.h>
void main ( )
{
    int rem, num, sum = 0;
    printf ("Enter a number: \n");
    scanf ("%d", &num);
    while (num > 0)
    {
        rem = num % 10;
        num = num / 10;
        sum = sum + rem;
    }
    printf ("Sum of Digits in a number =%d", sum);
}
```

#### Output:

**7.) Checking whether the number is palindrome or not:**

```
/* Program to Check whether the number entered is a palindrome or not */
#include <stdio.h>
void main ()
{
    int num, rev = 0, rem, temp;
    printf ("Enter a number: \n");
    scanf ("%d", &num);
    temp = num;
    while (num > 0)
    {
        rem = num % 10;
        num = num / 10;
        rev = rev * 10 + rem;
    }
    if (rev == temp)
        printf ("it is a palindrome");
    else
        printf ("it is not a palindrome");
}
```

**Output:**

### 8.) Finding of GCD and LCM of two numbers:

```
/* Program to find the Greater Common Divisor and Least Common Multiplier of two numbers*/
#include <stdio.h>
void main ()
{
    long int num1, num2, Nr, Dr;
    long int GCD, LCM, rem;
    printf("Enter two number: \n");
    scanf("%ld %ld", &num1, &num2);
    if(num1 > num2)
    {Nr = num1;
     Dr = num2;}
    else
    {
        Nr = num2;
        Dr = num1;}
    rem = Nr % Dr;
    while (rem != 0)
    {
        Nr = Dr;
        Dr = rem;
        rem = Nr % Dr;}
    GCD = Dr;
    LCM = num1 * num2 / GCD;
    printf("GCD = %ld \n LCM = %ld\n", GCD, LCM);
}
```

### Output:

### **9.) Checking whether a number is Prime or not:**

```
/* Program to check whether the number entered is a prime number or not */
# include <stdio.h>
void main ( )
{
int n,i,count=0;
printf ("Enter a number: \n");
scanf ("%d", &num);
for (i = 1; i<=num; i++)
{
if (num%i == 0)
{
count++;
}
}
if (count== 2)
printf ("it is a prime number");
else
printf("It is not a Prime number");
}
```

#### **Output:**

### 10.) Finding the maximum, minimum and sum of given set of numbers \*

```
/* Program to find the maximum, minimum and sum of given set of numbers*/
```

```
#include <stdio.h>

void main ()
{
    int a[20], max, min, sum = 0, i, n;

    printf("Enter the size of array: \n");
    scanf("%d",&n);

    printf("Enter the elements of array: \n");

    for (i=0; i<n; i++)
        scanf("%d",&a[i]);

    max = a[0];
    min = a[0];
    for (i=0; i<n; i++)
    {
        if (a[i] > max)
            max = a[i];
        if (a[i] < min)
            min = a[i];
        sum = sum + a[i];
    }
    printf("maximum=%d\n",max);
    printf("minimum=%d\n",min);
    printf("sum=%d\n",sum);
}
```

Output:

O/P

Enter the size of array : 5

Enter the element of array :

2 5 9 15 8

maximum = 15

minimum = 2

sum = 39

**11.) Finding roots of quadratic equation \***

```
/*Program to find a roots of quadratic equation. */
#include<stdio.h>
#include<math.h>

void main()
{
    int a,b,c,D;
    float x,y;
    printf("enter coefficient of x^2,x and constant");
    scanf("%d%d%d",&a,&b,&c);
    D=b*b-4*a*c;
    if(D<0)
        printf("both roots are imaginary");
    else if(D==0)
        {
            x=-b/(2.0*a);
            printf("both roots are equal");
            roots are real & distinct
            roots are -0.697224,
            -4.302776
            printf("roots are real and distinct");
        }
    y=(-b+sqrt(D))/(2*a);
    printf("n roots are %f,%f",x,y);
}
```

Output:

1 5 6  
- 2.000000,  
- 3.000000

**12.) Computing sine value using sine series expansion \***

/\* Program to compute sin x value using sine series expansion \*/
# include <stdio.h>
sin x =  $x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$

```
int degree, i=1;
double radian, term, Nr, Dr, sum = 0.0;
printf ("Enter a value in degree: \n");
scanf ("%d",&degree);
radian = degree * 0.01745329;
Nr = radian;
Dr = 1.0;
term = Nr / Dr;
while (i <= 20)
{
    sum = sum + term;
    term = Nr / Dr;
    Dr = Dr * (i + 1) * (i + 2);
    i = i + 2;
}
printf ("sine value = %lf", sum);
```

float x,y;

printf("enter coefficient of x^2,x and constant");

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

D=b\*b-4\*a\*c;

if(D<0)

printf("both roots are imaginary");

else if(D==0)

{

x=-b/(2.0\*a);

printf("both roots are equal");

roots are real & distinct

roots are -0.697224,

-4.302776

printf("roots are real and distinct");

x=(-b+sqrt(D))/(2\*a);

y=(-b-sqrt(D))/(2\*a);

printf("n roots are %f,%f",x,y);

}

}

Output:

1 5 6  
Sine value = 0.500000

$$t = 2 + 2 \cdot 2^2$$

$$\text{value } t = 20$$

$$\text{sum } (x - \frac{x^3}{3!} + \frac{x^5}{5!})$$

3.) Computing cosine value using cosine series expansion \*

/\* Program to compute cosine x value using cosine series expansion \*/

```
#include <stdio.h>
void main()
{
    int degree, i=2;
    double radian, term, Nr, Dr, sum = 1.0;
    printf ("Enter a value in degree: \n");
    scanf ("%d", &degree);
    radian = degree * 0.01745329;
    Nr = -radian * radian;
    Dr = 2.0;
    term = Nr / Dr;
    while (i <= 20)
    {
        sum = sum + term;
        Nr = (-Nr) * radian * radian;
        Dr = Dr * (i + 1) * (i + 2);
        term = Nr / Dr;
        i = i + 2;
    }
    printf ("cosine value = %lf", sum);
}
```

Output:

Enter a value in degree:

60

Cosine value = 0.500000

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

$$NY = -x^2 \cdot x = -x^2$$

$$Dr = 2 \cdot 0$$

$$term = \frac{Nr}{Dr} = \frac{-x^2}{2!}$$

$$sum = 1.0 + \frac{x^2}{2!} - \frac{x^4}{4!} + \dots$$

$$NY = x^2 + x \cdot x = x^4$$

$$Dr = 2 \cdot 0 \times (3) \times 4 = 4!$$

$$Dr = \frac{Nr}{Dr} = \frac{-x^4}{4!}$$

$$i = i + 2;$$

$$i = 4;$$

$$i = 1 + \frac{x^2}{2}$$

14.) To convert binary to decimal \*

/\* Write a program to convert given binary number to decimal. \*/

```
14.) To convert binary to decimal *
/* Write a program to convert given binary number to decimal. */
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
void main()
{
    long int decimal = 0, binary, remainder;
    long int temp, i=0;
    printf ("Enter binary number \n");
    scanf ("%d", &binary);
    temp = binary;
    do
    {
        remainder = binary % 10;
        binary = binary / 10;
        decimal = decimal + (remainder * pow (2, i));
        i++;
    } while (binary > 0);
    printf ("\n the value of binary %d in decimal is %d", temp, decimal);
}
```

Output:

Enter binary number  
200

The value of binary 200 in decimal  
is 8

15.) To convert binary to octal \*

/\* C programming source code to convert binary to octal according to data entered by user. \*/

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

void main()
{
    long int n;
    int octal=0,decimal=0,i=0;
    printf("Enter a binary number: ");
    scanf("%d",&n);
    while(n!=0)
    {
        decimal=decimal+(n%10)*pow(2,i);
        i++;
        n=n/10;
    }
    /* At this point, the decimal variable contains corresponding decimal value of binary number. */
    i=1;
    while (decimal!=0)
    {
        octal+=(decimal%8)*i;
        decimal=decimal/8;
        i=10;
    }
    printf("Its equivalent octal number is: %d",octal);
}
```

Output:

$$\begin{array}{r} 1001 \\ \times 3 \\ \hline 1001 \\ \end{array}$$

Page 18 of 61

16.) To convert binary to hexadecimal \*

/\* C programming source code to convert binary to hexadecimal according to data entered by user. \*/

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

void main()
{
    long int n;
    int decimal=0,i=0,hex,count=0,j=0;
    char a[20];
    printf("Enter a binary number: ");
    scanf("%d",&n);
    while(n!=0)
    {
        decimal=decimal+(n%10)*pow(2,i);
        i++;
        n=n/10;
    }
    while(decimal>0)
    {
        hex=decimal %16;
        decimal=decimal /16;
        if(hex==10)
            a[i]=A;
        if(hex==11)
            a[i]=B;
        if(hex==12)
            a[i]=C;
        if(hex==13)
            a[i]=D;
        if(hex==14)
            a[i]=E;
        if(hex==15)
            a[i]=F;
        i++;
        count++;
    }
    for (i=count-1;j>=0;i--)
    {
        if(a[i]>=65 &&a[i]<=70)
            printf("%c", a[i]);
        else
            printf("%d", a[i]);
    }
}
```

Output:

Page 19 of 61

17.) To convert decimal to binary \*

```
#include<stdio.h>
void main()
{
    int decimal,i=1,binary=0,rem;
    printf("Enter a decimal number to convert:");
    scanf("%d",&decimal);
    while(decimal>0)
    {
        rem=decimal%2;
        decimal=decimal/2;
        binary=binary+(rem*i);
        i=i*10;
    }
    printf("of given number is %d",binary);
}
```

Output:

```
while (4>0)
{
    rem = 4 % 2 => 0
    dec = 4/2 => 2
    binary = 0 + (0x1) => 0
    i = 10 ;
}
while (2 >0)
{
    rem = 2 % 2 => 0
    dec = 2/2 => 1
    binary = 0 + (0x1) = 0
    i = 100
}
while (1 >0)
{
    rem = 1
    dec = 0
    binary = 0 + (1x100) = 100
    i = 10x10 => 1000
}
while (0>0) F X
```

18.) To convert octal to binary \*

```
/* C programming source code to convert octal to binary according to data entered by user.*/
#include<stdio.h>
void main()
{
    int i=0;
    char octnum[100];
    printf("Enter any Octal Number : ");
    scanf("%s",octnum);
    printf("Equivalent Binary Value = ");
    while(octnum[i])
    {
        switch(octnum[i])
        {
            case '0' : printf("000");
            break;
            case '1' : printf("001");
            break;
            case '2' : printf("010");
            break;
            case '3' : printf("011");
            break;
            case '4' : printf("100");
            break;
            case '5' : printf("101");
            break;
            case '6' : printf("110");
            break;
            case '7' : printf("111");
            break;
            default : printf("\nInvalid Octal Digit %c\n",octnum[i]);
            break;
        }
        i++;
    }
}
```

Output:

19.) To convert hexadecimal to binary \*  
/\* C programming source code to convert hexadecimal to binary according to data entered by  
user. \*/

```
#include <stdio.h>
void main()
{
    char hex[100];
    int i=0;
    printf("Enter a HexaDecimal Number: ");
    scanf("%s",hex);
    printf("Equivalent Binary Number is: ");
    while(hex[i])
    {
        switch(hex[i])
        {
            case '0':
                printf("0000");
                break;
            case '1':
                printf("0001");
                break;
            case '2':
                printf("0010");
                break;
            case '3':
                printf("0011");
                break;
            case '4':
                printf("0100");
                break;
            case '5':
                printf("0101");
                break;
            case '6':
                printf("0110");
                break;
            case '7':
                printf("0111");
                break;
            case '8':
                printf("1000");
                break;
            case '9':
                printf("1001");
                break;
            case 'A':
                printf("1010");
                break;
            case 'B':
                printf("1011");
                break;
            case 'C':
                printf("1100");
                break;
            case 'D':
                printf("1101");
                break;
            case 'E':
                printf("1110");
                break;
            case 'F':
                printf("1111");
                break;
            case 'a':
                printf("1010");
                break;
            case 'b':
                printf("1011");
                break;
            case 'c':
                printf("1101");
                break;
            case 'd':
                printf("1100");
                break;
            case 'e':
                printf("1110");
                break;
            case 'f':
                printf("1111");
                break;
        }
    }
}
```

Output:

20.) Generating Pascal triangle for a number \*

/\*C Program for Generating Pascal triangle for a number\*/

#include <stdio.h>

void main()

{ int rows, coef = 1, space, i, j;

printf("Enter number of rows: ");

scanf("%d", &rows);

for(i=0; i<rows; i++)

{ for(space=1; space <= rows-i; space++)

printf(" ");

for(j=0; j <= i; j++)

{

if(j==0 || j==i)

coef = 1;

else

coef = coef \* (i-j+1)/j;

printf("%d", coef);

}

printf("\n");

Output:

Output:  
Enter No of rows in the Pyramid

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

21.) Generating pyramid for a number \*

/\*C program for printing Pyramid\*/

#include <stdio.h>

void main()

{

int r, sp, n, temp;

printf("Enter the number of rows in the pyramid: ");

scanf("%d", &n);

temp=n;

for(r=1; r<=temp; r++)

{

for(sp=1; sp<=r; sp++)

printf(" ");

for(c=1; c<=r; c++)

{

printf("%d", r);

printf(" ");

}

printf("\n");

n=n-1;

}

}

$n = \text{size}$ .  
 $r = \text{current row}$   
 $sp = \text{space}$   
 $c = \text{column point}$

22.) Finding factorial of a number using recursion \*

/\* Program for finding factorial of a number using recursion \*/

#include<stdio.h>

int fact(int);

void main()

{

long int num,f;

printf("nEnter a number: ");

scanf("%d",&num);

f=fact(num);

printf("nFactorial of %d is: %d",num,f);

}

int fact(int n)

{

if(n==1)

return 1;

else

return(n\*fact(n-1));

}

Output:

5  
Factorial of 5 is 120

23.) Finding Fibonacci series of a number using recursion \*

#include<stdio.h>

int fibonaci(int , int , int );

void main()

{

int n,a=0,b=1;

printf("nEnter the range of the Fibonacci series: ");

scanf("%d",&n);

printf("Fibonacci Series: ");

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

fibonaci(n);

}

void fibonaci(int n)

{

if(n==1)

return 1;

else

return(n\*fibonaci(n-1));

}

static int a=0,b=1,sum;

if(n>2)

{

sum = a+b;

a=b;

b=sum;

printf("%d",sum);

fibonaci(n-1);

}

Output:

5

Enter a Number =

5

Factorial of 5 is 120

Output:

3

Enter a Number =

3

Fibonacci Series: 0 1 1 2 3

Output:

3

Enter a Number =

3

Fibonacci Series: 0 1 1 2 3

#### 24.) Finding GCD of a number using recursion \*

```
/*Program for GCD of a number using recursion*/
```

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

```
void main() → int findgcd (int ,int )
```

```
{ int n1,n2,gcd;
```

```
printf("\nEnter two numbers: ");
```

```
scanf("%d %d",&n1,&n2);
```

```
gcd=findgcd(n1,n2);
```

```
printf("\nGCD of %d and %d is: %d",n1,n2,gcd);
```

```
}
```

```
int findgcd(int x,int y)
```

```
{
```

```
while(x!=y)
```

```
{
```

```
if(x>y)
```

```
return findgcd(x-y,y);
```

```
else
```

```
return findgcd(y,x-y);
```

```
}
```

```
return x;
```

Output:

```
int n1,n2,gcd;
```

```
printf("\nEnter two numbers: ");
```

```
scanf("%d %d",&n1,&n2);
```

```
gcd=findgcd(n1,n2);
```

```
printf("\nGCD of %d and %d is: %d",n1,n2,gcd);
```

```
}
```

Output:

```
Enter Array 1 elements :
```

```
5  
4  
2
```

```
Enter Array 2 elements :
```

```
7  
8  
9
```

```
12  
12
```

```
15  
4
```

#### 25.) Performing addition on two matrices using arrays \*

```
/* Program to perform addition on two matrices using array */
```

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

```
void main()
```

```
int x[10][10], y[10][10], z[10][10];
```

```
int i,j, row, col;
```

```
printf("Enter no of rows and column: \n");
```

```
scanf("%d%d", &row, &col);
```

```
printf("Enter Array1 elements: \n");
```

```
for(i=0;i<row;i++)
```

```
for(j=0;j<col;j++)
```

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

```
printf("Enter Array2 elements: \n");
```

```
for(i=0;i<row;i++)
```

```
for(j=0;j<col;j++)
```

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

```
z[i][j] = x[i][j] + y[i][j];
```

```
printf ("In Resultant Matrix: \n");
```

```
for(i=0;i<row;i++)
```

```
for(j=0;j<col;j++)
```

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

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

```
}
```

```
}
```

```
for(i=0;i<row;i++)
```

```
for(j=0;j<col;j++)
```

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

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

```
}
```

```

26.) Performing Multiplication on two matrices using arrays *
/* Program to perform multiplication on two matrices using array */

#include <stdio.h>

void main ()
{
    int x [10][10], y [10][10], z [10][10];

    int i, j, r1, c1, r2, c2, t;

    printf("Enter the rows and column for Mat-1: \n");
    scanf("%d %d", &r1, &c1);

    printf("Enter the rows and column for Mat-2: \n");
    scanf("%d %d", &r2, &c2);

    if(c1 == r2)

    {
        printf("Matrix Multiplication possible: \n");

        printf("Enter %d elements for Mat-1: \n", r1*c1);
        for(i=0; i< r1; i++)
            for(j=0; j< c1; j++)
                scanf("%d", &x[i][j]);

        printf("Enter %d elements for Mat-2: \n", r2*c2);
        for(i=0; i< r2; i++)
            for(j=0; j< c2; j++)
                scanf("%d", &y[i][j]);

        for(i=0; i< r1; i++)
            for(j=0; j< c2; j++)
            {
                for(t=0; t< c1; t++)
                    z [i][j] = 0;
                for(t=0; t< c1; t++)
                    z [i][j] = z [i][j] + (x [i][t] * y [t][j]);
            }
        printf ("Resultant Matrix: \n");
    }
}

```

```

for(i=0; i< r1; i++)
{
    for(j=0; j< c2; j++)
        printf ("%d\t", z[i][j]);
    printf ("\n");
}

else
    printf ("matrix Multiplication is not possible");
}

```

Output:

27.) Performing Linear search using recursive procedure \*

```
#include<stdio.h>
int linear(int,int); //function prototype
void main()
{
    int a[2],key,ans,n;
    printf("Enter number of elements");
    scanf("%d",&n);
    printf("Enter %d elements\n",n);
    for(i=0;i<n-1;i++)
    {
        scanf("%d",&a[i]);
    }
    printf("\nEnter the number to be searched\n");
    scanf("%d",&key);
    ans=linear(key,0); //function call
    if(ans<n-1)
    {
        printf("Number (%d) found at position (%d)",key,ans+1);
    }
    else
    {
        printf("Didnot find the number (%d)",key);
    }
}
int linear(int key,int position)
{
    if(key==a[position])
    {
        return position;
    }
    else
    {
        position++;
        return linear(key,position); //Recursive call
    }
}
```

Output:

### 28.) Performing binary search using recursive procedure \*

```
#include<stdio.h>
#define size 10
int binsearch(int[], int, int, int);
void main()
{
    int num, i, key, position;
    int low, high, list[size];
    printf("nEnter number of elements:\n");
    scanf("%d", &num);
    printf("nEnter %d elements in ascending order:\n", num);
    for (i = 0; i < num; i++)
    {
        scanf("%d", &list[i]);
    }
    low = 0;
    high = num - 1;
    printf("nEnter element to be searched: ");
    scanf("%d", &key);
    position = binsearch(list, key, low, high);
    if (position != -1)
        printf("\nNumber present at position %d", (position + 1));
    else
        printf("\n The number is not present in the list");
}
// Binary Search function
```

```
int binsearch(int a[], int x, int low, int high)
{
    int mid;
    if (low > high)
        return -1;
    mid = (low + high) / 2;
    if (x == a[mid])
        return (mid);
    else if (x < a[mid])
        binsearch(a, x, low, mid - 1);
    else
        binsearch(a, x, mid + 1, high);
}
```

Output:

### 29.) Performing Linear search using non-recursive procedure \*

```
/* Program to demonstrate Linear search on the list of array elements */
```

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

```
void main()
```

```
{
```

```
int a[100];
```

```
int key, N, position, i;
```

```
int linearsearch (int [], int, int);
```

```
printf("Enter the size of array: \n");
```

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

```
printf("Enter %d elements:", N);
```

```
for (i=0; i<N; i++)
```

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

```
printf("Enter the key element: \n");
```

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

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

Enter the key element:

Page 36 of 61

### 30.) Performing Binary search using non-recursive procedure \*

```
/* Program to perform binary search on the list of array elements */
```

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

```
void main()
```

```
{
```

```
int a[100];
```

```
int N, key, position, i;
```

```
int linearsearch (int [], int, int);
```

```
printf("Enter the size of array: \n");
```

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

```
printf("Enter %d elements in ASCENDING ORDER:", N);
```

```
for (i=0; i<N; i++)
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

```
printf("Enter the key element: \n");
```

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

Enter the key element:

Page 37 of 61

X

$$f = \frac{1}{2}x^2 - \frac{1}{3}x^3 + \frac{1}{4}x^4 - \frac{1}{5}x^5 + \dots$$

```

}
return (-1);
}

```

Output:

~~$f = \frac{1}{2}x^2 - \frac{1}{3}x^3 + \frac{1}{4}x^4 - \frac{1}{5}x^5 + \dots$~~

X

$$1. \quad x^2 \\ 2. \quad x^3 \\ 3. \quad x^4 \\ 4. \quad x^5$$

31.) Performing Bubble sort on the given list of array elements \*

# include <stdio.h>

```

void main()
{

```

```

    int a[100];
    int trip, N, i, temp;

```

```

    printf("Enter the size of array: \n");
    scanf("%d", &N);

```

```

    printf("Enter %d elements: \n", N);
    for(i=0; i<N; i++)
    {
        scanf("%d", &a[i]);
    }

```

```

    for(trip = 1; trip < N; trip++)
    {
        for(i=0; i<N-trip; i++)
        {
            if(a[i] > a[i+1])
            {
                temp = a[i];
                a[i] = a[i+1];
                a[i+1] = temp;
            }
        }
    }

```

Enter the size of array:

5

Enter 5 elements :

7

6

2

8

3

The sorted array list is :

```

    printf("In the sorted array list is: \n");
    for(i=0; i<N; i++)
    {
        printf("%d\n", a[i]);
    }
}

```

Output:

2 3 6 7 8

32.) Performing Selection sort on the given list of array elements \*

```
/* Program to perform Selection sort on the given list of numbers */
#include <stdio.h>
void main ()
{
    int A [100];
    int small, N, i, j, temp;
    printf ("Enter the size of array: \n");
    scanf ("%d", &N);
    printf ("Enter array elements: \n", N);
    for (i=0; i<N; i++)
        scanf ("%d", &A [i]);
    for (j=0; i<N; i++)
    {
        small = i;
        for (j=i+1; j<N; j++)
            if (A [j] < A [small])
                small = j;
    }
    temp = A [i];
    A [i] = A [small];
    A [small] = temp;
}
printf ("In the sorted array list is: \n");
for (i = 0; i < N; i++)
    printf ("%d \n", A [i]);
```

Output:

Enter the size of array:  
5  
Enter 5 array elements:  
1 6 7 8 9

33.) Program illustrating pointer to 1D array \*

```
/* Program to Print array elements in reverse order */
#include <stdio.h>
```

```
void main ()
{
    void reverse (int * , int);
    int a[10], n, i;
    printf ("Enter the size: \n");
    scanf ("%d", &n);
    printf ("Enter %d array elements", n);
    for (i = 0; i < n; i++)
        scanf ("%d", &a[i]);
    reverse (a, n);
}
```

```
void reverse (int * a, int n)
{
    int i;
    for (i = n-1; i >= 0; i--)
        printf ("%d \n", a [i]);
}
```

Enter the size:  
5  
Enter 5 array elements:

9  
8  
6  
7  
1  
0  
for (i = n-1; i >= 0; i--)  
 printf ("%d \n", a [i]);

Output:

6 7 5 0 9

**34.) Program illustrating pointer to 2D array \***

// Program to Display the contents of two dimensional array using pointers.

```
#include<stdio.h>
void main()
{
    int a[3][2]={100,200,300,400,500,600},i,j,*ptr;
    ptr=&a[0][0];
    for(i=0;i<=3-1;j++)
    {
        for(j=0;j<=2-1;j++)
        {
            printf("Element = %d\n",*ptr);
            ptr++;
        }
    }
}
```

**Output:**

**35.) Program illustrating pointer to function \***

//Program to find the sum of two numbers implementing the concept of pointer to function

```
#include<stdio.h>
void main()
{
    int add(int,int);
    int a,b;
    int (*fptr)();

    fptr=&add;
    printf("enter two numbers\n");
    scanf("%d%d",&a,&b);
    printf("result=%d",(*fptr)(a,b));
}
```

```
int add(int a, int b)
{
    return(a+b);
}
```

**Output:**

36.) Program for string manipulation using library functions \*

```
/* Program for string manipulation using library functions */  
#include <stdio.h>  
  
void main ()  
{  
    char s1 [100], s2 [100], a [100], b [100];  
  
    int n;  
  
    printf("Enter the first string: ");  
  
    gets (s1);  
  
    printf("Enter the second string: ");  
  
    gets (s2);  
  
    n=strlen(s1);  
  
    printf("\nLength of first string is: %d\n",n);  
  
    n=strlen(s2);  
  
    printf("\nLength of second string is: %d\n",n);  
  
    if(strcmp (s1, s2)==0)  
        printf("\nStrings are equal\n");  
  
    else  
        printf("\nStrings are not equal\n");  
  
    strcat (s1, s2);  
  
    printf("\nThe concatenated string is: %s\n",s1);  
  
    printf("\nReverse of second string is: %s\n",strrev(s2));  
  
    printf("\nEnter a string that is to be copied: ");  
  
    gets (a);  
  
    strcpy (b, a);
```

printf("The string after being copied is: %s\n",b);  
})

Output:

Enter the first string : Programming for problem solving  
Enter the second string : c lab

length of first string is : 87  
length of second string is : 5

String are not equal

The concatenated string is : Programming for problemsolving c lab

Reverse of second string is : lab c

int a string that is to be copied : c lab

The string after being copied is : c lab

### 37.) Generation of address labels using structures \*

```
/* Program to generate address labels using structure */
```

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

```
struct address
```

```
{
```

```
char name [20];
```

```
char hno [20];
```

```
char city [20];
```

```
char state [20];
```

```
char country [20];
```

```
long int pcode;
```

```
};
```

```
void main ()
```

```
{
```

```
int size, i;
```

```
struct address a[60];
```

```
printf("Enter no. of students you want to store
```

```
details of : 
```

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

```
printf("Enter data for %d students: \n", size);
```

```
for (i = 0; i < size; i++)
```

```
{
```

```
printf("Enter name, house no, city: \n");
```

```
scanf("%s %s %s", a[i].name, a[i].hno, a[i].city);
```

```
printf("Enter state, country and pincode: \n");
```

```
scanf("%s %s %d", a[i].state, a[i].country, &a[i].pcode);
```

```
printf("Entered details of student %d: \n", i+1);
```

```
for (i = 0; i < size; i++)
```

```
{
```

```
printf("Student %d \n", i+1);
```

```
printf("Name : %s \n house no : %s \n city : %s \n state : %s \n country : %s \n pincode : %d \n", a[i].name, a[i].hno, a[i].city, a[i].state, a[i].country, a[i].pcode);
```

Enter no. of students you want to store  
details of : 2

Enter data for 2 students ?

Enter name, hno, city :

hadi 454

hyderabad

Enter state, country, pincode :

Telangana

india

500065

Entered details of student 1 :  
name : hadi  
city : hyderabad  
state : telangana

Entered details of student 2 :  
name : india  
city : hyderabad  
state : telangana

Output:

**38. Program on Illustration of a Union \***

```
/* Program illustrating a union */

#include <stdio.h>

union data
{
    int x;
    double y;
    float z;
};

void main()
{
    union data d;
    d.x = 100;
    d.y = 300.00;
    printf("integer = %d\n", d.x);
    printf("double = %f\n", d.y);
    d.z = 50.00;
    printf("float = %f\n", d.z);
}
```

**Output:**

```
int words=0, characters=0, lines=0;
char ch;

printf("Enter lines of text and press F6. Press enter to see the output:\n");
while((ch=getchar())!=EOF)
{
    characters++;
    if(ch=='\n')
        lines++;
    if((ch==' ')||(ch=='\r'))
        words++;
}

if(characters>0)
{
    words++;
    lines++;
}

printf("Characters=%d\n", characters);
printf("Words=%d\n", words);
printf("Lines=%d", lines);
}
```

**40.) Program to copy contents of one file to another \***

```

/* Program to copy contents of one file into another file */
#include <stdio.h>
void main ()
{
    FILE *ptr;
    FILE *pt;
    char ch;

    ptr = fopen ("Quote.c", "r");
    fputs ("knowledge helps you to reach your destination provided you know what your destination is", ptr);

    fclose (ptr);

    ptr = fopen ("Quote.c", "r");
    pt = fopen ("new.c", "w");
    while (!feof (ptr))
    {
        ch = fgetc (ptr);
        if (ch == 'f') // File is copied
            fputc (ch, pt);
        else
            fputc (ch, pt);
    }
    printf("File is copied");
    fclose (pt);
    fclose (ptr);
}

```

**Output:**

knowledge helps you to reach your destination provided you know what your destination is

**41.) Program to print contents of file in reverse order \***

```

/* Program to Print contents of file in reverse order */
#include <stdio.h>
void main ()
{
    int i = 0, count = 0;
    char ch, a[100];
    FILE *ptr;

    ptr = fopen ("Rev.c", "r");
    printf ("Enter a line of text and press F6: \n");
    while ((ch = getchar ()) != EOF)
    {
        fputc (ch, ptr);
    }
    fclose (ptr);

    ptr = fopen ("Rev.c", "r");
    while ((ch = getc (ptr)) != EOF)
    {
        a[i] = ch;
        i++;
        count++;
    }
    printf ("contents in reverse order :\n");
    for (i = count - 1; i >= 0; i--)
    {
        printf ("%c", a[i]);
    }
    fclose (ptr);
}

```

**Output:**

**42.) Illustration of call by value.**

```
/* Program to swap two numbers using call by value */

#include <stdio.h>
void swap (int, int);
void main ()
{
    int x = 100, y = 200;
    printf ("values before swap: \n");
    printf ("x = %d \t y = %d", x,y);
    swap (x, y);
    printf ("\n values after swap: \n");
    printf ("x = %d \t y = %d", x,y);
}
```

**Output:**

**43.) Illustration of call by reference.**

```
/* Program to swap two numbers using call by value */

#include <stdio.h>
void swap (int *, int *);
void main ()
{
    int x = 100, y = 200;
    printf ("values before swap: \n");
    printf ("x = %d \t y = %d", x,y);
    swap (&x, &y);
    printf ("\n values after swap: \n");
    printf ("x = %d \t y = %d", x,y);
}
```

**Output:**

44.) Program for string manipulation using user defined functions

```
#include <stdio.h>
void main()
{
    char s1 [100], s2 [100], a [100], b [100];
    void compare (char [], char []);
    void concat (char [], char []);
    void copy (char [], char []);

    printf ("Enter the first string: \n");
    gets (s1);

    printf ("Enter the second string: \n");
    gets (s2);

    printf ("\n\n after comparing the string, the result is : \n");
    compare (s1, s2);

    concat (s1, s2);

    printf ("\nEnter a string that is to be copied : \n");
    gets (a);

    copy (a, b);

}

void compare (char s1 [100], char s2 [100])
{
    int i = 0;
    while (s1 [i] == s2 [i] && s1 [i] != '\0' && s2 [i] != '\0')
    {
        i++;
    }
}

void concat (char s1 [100], char s2 [100])
{
    int i, j;
    char c [100];
    for (i = 0; s1 [i] != '\0'; i++)
    {
        c [i] = s1 [i];
    }
    for (j = 0; s2 [j] != '\0'; j++)
    {
        c [i + j] = s2 [j];
    }
    c [i + j] = '\0';
    printf ("\n The concatenated string is: \n %s", c);
}

void copy (char a [100], char b [100])
{
    int i = 0;
    for (i = 0; a [i] != '\0'; i++)
    {
        b [i] = a [i];
    }
    b [i] = '\0';
}
```

```
printf ("\n\n the string after being copied is: \n\n %s", b);  
}  
  
Output:
```

45.) Program illustrating Structures within structures:  
/\* Program to illustrate embedded structure \*/  
  
#include <stdio.h>  
  
void main()  
{  
 struct salary  
 {  
 char name[20];  
 char dept[10];  
 struct addon  
 {  
 int bonus;  
 int transport;  
 int medical;  
 } allowance;  
 } employee;  
  
 printf ("Enter name and department: \n");  
 scanf ("%s %s", employee.name, employee.dept);  
 printf ("Enter bonus, transport, medical allowances: \n");  
 scanf ("%d %d %d", &employee.allowance.bonus, &employee.allowance.transport, &employee.  
allowance.medical);  
  
 printf ("\n\n Details are: \n\n");  
 printf ("name: %s \n", employee.name);  
 printf ("department: %s \n", employee.dept);  
 printf ("Bonus: %d \n transport: %d \n, medical: %d \n", employee.allowance.bonus, employee.  
allowance.transport, employee.allowance.medical);  
}

Output:

**46.) Program illustrating Self Referential Structure:**

```
/* program illustrating self referential structure */
#include <stdio.h>
void main ()
{
    struct detail
    {
        char name [20];
        int age;
        struct detail *ptr;
    };
    struct detail p1 = {"John", 60};
    struct detail p2 = {"Smith", 61};
    struct detail p3 = {"Mary", 68};
    p1.ptr=&p2;
    p2.ptr=&p3;
    p3.ptr=0;
    printf ("name1: %s \t Age1: %d \n", p1.name, p1.age);
    printf ("name2: %s \t Age2: %d \n", p1.ptr->name, p1.ptr->age);
    printf ("name3: %s \t Age3: %d \n", p2.ptr->name, p2.ptr->age);
}
```

**Output:**