PROGRAMMING FOR PROBLEM SOLVING LAB MANUAL

Simple numeric problems:

a. Write a program for find the max and min from the three numbers.

```
#include<stdio.h>
void main()
{
 int a,b,c;
 printf("Enter 3 numbers");
 scanf("%d%d%d",&a,&b,&c);
   if (a<=b && a <=c)
      {
          if(b<c)
             printf("Min=%d, Max= %d",a,c);
          else
             printf("Min=%d, Max= %d",a,b);
      }

   else if( b<=a && b<=c)
```

```
    {
        if(a<c)
            printf("Min=%d, Max= %d",b,c);
        else
            printf("Min=%d, Max= %d",b,a);
    }

  else
    {
        if(a<b)
            printf("Min=%d, Max= %d",c,b);
        else
            printf("Min=%d, Max= %d",c,a);
    }

}
```

RESULT:

INPUT: Enter 3 numbers 5 4 6

OUTPUT : Min=4, Max= 6

b. Write the program for the simple, compound interest.

```
#include<stdio.h>

#include<math.h>

int main()

{

        int p,t;

        float r,si,amount,ci;

        printf("Please enter principal,time and rate of
interest");

        scanf("%d%d%f",&p,&t,&r);

        si=p*t*r/100;

//Simple Interest formula is p*t*r

        printf("\nSimple interest = %.3f",si);
```

```
//Compound Interest formula is below

        amount=p*pow((1 +r/100),t);

        ci=amount-p;

        printf("\nCompound interest = %.3f",ci);

}
```

RESULT:

INPUT: Please enter principal,time and rate of interest 1000 2 12

OUTPUT : Simple interest = 240.000
Compound interest = 254.400

c. Write program that declares Class awarded for a given percentage of marks, where
mark <40%= Failed, 40% to <60% = Second class, 60% to <70%=First class,
>=70% = Distinction. Read percentage from standard input.

```
#include <stdio.h>

int main()
{
    int M1, M2, M3;
    float per;

    printf("Enter three subjects marks: ");
    scanf("%d%d%d", &M1, &M2, &M3);

    per = (M1 + M2 + M3) / 3.0;

    printf("Percentage = %.2f\n", per);

    if(per >= 70)
    {
        printf("Distinction");
    }
    else if((per>=60) && (per<= 70))
```

```
    {
        printf("First class");
    }
    else if((per>=40) && (per<= 60))
    {
        printf("Second class");
    }
    else if (per <= 40)
    {
        printf("Failed");
    }

   return 0;
}
```

RESULT- 1:

INPUT: Enter three subjects marks: 40 35 30
OUTPUT : Percentage = 35.00
Failed

RESULT- 2:

INPUT: Enter three subjects marks: 78 65 90
OUTPUT: Percentage = 77.67
Distinction

d. Write a program that prints a multiplication table for a given number and the number of rows in the table. For example, for a number 5 and rows = 3, the output should be:

5 x 1 = 5
5 x 2 = 10
5 x 3 = 15

```
#include <stdio.h>
int main()
{
    int n, i, range;

    printf("Enter an integer: ");
    scanf("%d",&n);

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

    for(i=1; i <= range; ++i)
    {
        printf("%d * %d = %d \n", n, i, n*i);
    }

    return 0;
}
```

RESULT:

INPUT:

Enter an integer: 5
Enter the range: 3

OUTPUT :

5 * 1 = 5
5 * 2 = 10
5 * 3 = 15

e. Write a program that shows the binary

equivalent of a given positive number between 0 to 255.

```
#include <stdio.h>
long decimalToBinary(long n);
int main() {
long decimal;
printf("Enter a decimal number\n");
scanf("%ld", &decimal);
printf("Binary number of %ld is %ld", decimal,
decimalToBinary(decimal));
return 0;
}

/* Function to convert a decinal number to binary
number */
long decimalToBinary(long n) {
int remainder;
long binary = 0, i = 1;

while(n != 0) {
remainder = n%2;
n = n/2;
binary= binary + (remainder*i);
i = i*10;
}
return binary;
}
```

RESULT:

INPUT:
Enter a decimal number
211

OUTPUT :
Binary number of 211 is 11010011

Expression Evaluation:

a. A building has 10 floors with a floor height of 3 meters each. A ball is dropped from the top of the building. Find the time taken by the ball to reach each floor.
(Use the formula s = ut+(1/2)at^2 where u and a are the initial velocity in m/sec (= 0) and acceleration in m/sec^2 (= 9.8 m/s^2)).

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

void main()
{
    float s=30,u=0,a=9.8,t,temp;    // s = 10*3 = 30

            /* s = ut+(1/2)at^2
               s = 0*t + (0.5)*a*t^2    */

   temp=(0.5)*a;

          /* s = 0 + temp*t ^2
             s/temp = t ^2
             sqrt(s/temp) = t   */

   t=u*t+sqrt(s/temp);
   printf("Time taken by the ball to reach each floor is
```

```
=%f ",t);

    return 0;
}
```

RESULT:

OUTPUT:

Time taken by the ball to reach each floor is = 2.474358

b. Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +,-,*, /, % and use Switch Statement)

```
#include <stdio.h>

void main()
{
   int a, b, c;
   char ch;

   printf("Enter your operator(+, -, /, *, %)\n");
   scanf("%c", &ch);

   printf("Enter the values of a and b\n");
   scanf("%d%d", &a, &b);

   switch(ch)
   {
      case '+': c = a + b;
                printf("addition of two numbers is %d", c);
                break;
      case '-': c = a - b;
```

```
            printf("substraction of two numbers is %d",
c);
            break;
      case '*': c = a * b;
            printf("multiplication of two numbers is
%d", c);
            break;
      case '/': c = a / b;
            printf("remainder of two numbers is %d",
c);
            break;
      case '%': c = a % b;
            printf("quotient of two numbers is %d", c);
            break;
      default: printf("Invalid operator");
           break;
   }

}
```

RESULT:

INPUT:

```
Enter you operator(+, -, /, *, %)
 +
Enter the values of a and b
 1 3
```

OUTPUT:

addition of two numbers is 4

c. Write a program that finds if a given number is a prime number

```
#include <stdio.h>

int main()
 {
    int n, i, c = 0;
    printf("Enter any number n:");
    scanf("%d", &n);

    for (i = 1; i <= n; i++)
       {
          if (n % i == 0)
             {
                c++;
             }
       }


  if (c == 2)
    {
      printf("The given no is a Prime number");
    }
  else
```

```
    {
      printf("The given no is not a Prime number");
    }
 return 0;
}
```

RESULT:

INPUT:

Enter any number n: 7

OUTPUT:

The given no is a Prime number

d. Write a C program to find the sum of individual digits of a positive integer and test given number is palindrome.

```
#include<stdio.h>

void main()
{
    int n,x,r,sum=0,k=0;

    printf("\nEnter The Number:");
    scanf("%d",&n);

    x=n;

    while(n!=0)
    {
        r=n%10;
        sum=sum+r;
        k=k*10+r;
        n=n/10;
    }

    printf("sum of%d is %d",x,sum);

    if(x==k)
      {
```

```
        printf("\n  %d is a Palindrome Number",x);
      }
    else
      {
        printf("\n  %d is not a Palindrome Number",x);
      }
}
```

RESULT:

INPUT:

Enter The Number: 121

OUTPUT:

sum of 121 is 4

121 is a Palindrome Number

4

121 is a Palindrome Number

e. A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.

```
#include <stdio.h>
int main()
{
  int i,n,a=0,b=1,c;
  printf("Enter n value:");
  scanf("%d",&n);
  printf("\n%d\t%d\t",a,b);
  for(i=3;i<=n;i++)
  {
    c=a+b;
    printf("%d\t",c);
    a=b;
    b=c;
   }
```

```
 return 0;
}
```

RESULT:

INPUT:

Enter n value 5

OUTPUT:

0 1 1 2 3

f. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

```
#include<stdio.h>

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

   printf("Enter any number\n");
   scanf("%d", &n);

   printf("The prime numbers between 1 to %d\n",n);

   for(i = 1; i <= n; i++)
     {
      count = 0;
       for(j = 1; j <=i; j++)
        if(i % j == 0)
          {
             count++;
```

```
        }
      if(count == 2)
        {
          printf("%d\t", i);
        }
   }
}
```

RESULT:

INPUT:

Enter any number
10

OUTPUT:

The prime numbers between 1 to 10
2 3 5 7

g. Write a C program to find the roots of a Quadratic equation.

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

int main()
{
  int a,b,c,d,reat,imaginary;
  float r1,r2;

  printf(“enter values of a,b,c”);
  scanf("%d%d%d",&a,&b,&c);

  d=(b*b)-(4*a*c);

  if(d==0)
    {
      printf("\n roots are equal ");
      r1=-b/(2*a);
      r2=-b/(2*a);
      printf("\n the value of root1 is  =%f",r1);
      printf("\n the value of root2 is =%f",r2);
    }
```

```
  else if(d>0)
    {
      printf("roots are distinct:");
      r1=(-b+sqrt(d))/(2*a);
      r2=(-b-sqrt(d))/(2*a);
      printf("\n the value of root1 is  =%f",r1);
      printf("\n the value of root2 is =%f",r2);
    }
 else
    {
        realPart = -b/(2*a);
        imaginaryPart = sqrt(-determinant)/(2*a);
        printf("root1 = %f+%f and root2 = %f-%f", real,
imaginary, real, imaginary);
      //printf("\n roots are imaginary");
    }
 return 0;
}
```

RESULT:

INPUT:

enter the values of a,b,c 1 6 9

OUTPUT:

roots are equal

the value of root1 is =-3.0000

the value of root2 is =-3.0000

h. Write a C program to calculate the following, where x is a fractional value.

1-x/2 +x^2/4-x^3/6

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

int main()
 {
      float   sum=1;
      int   x,k=1,i;

      printf("Enter x value:");
      scanf("%d",&x);

     for(i=1;i<=3;i++)
         {
              k=-k;
              sum=sum+(k*pow(x,i))/(i*2);
         }

     printf("sum of the series=%f",sum);
```

```
    return 0;
}
```

RESULT:

INPUT:
Enter x value: 4

OUTPUT:
sum of the series= -7.666

i. Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression:
1+x+x^2+x^3+.....+x^n.
(For example: if n is 3 and x is 5, then the program computes 1+5+25+125).

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

void main()
{
   int n, x, i, sum = 0;

   printf("Enter the limit\n");
   scanf("%d", &n);

   printf("Enter the value of x\n");
   scanf("%d", &x);

   if(x < 0 || n < 0)
     {
       printf("illegal  value");
     }
   else
```

```
    {
      for(i = 0; i <= n; i++)
      sum=sum + pow(x, i);
    }
  printf("sum=%d", sum);
}
```

RESULT:

INPUT:

Enter the limit
3
Enter the value of x
5

OUTPUT:

sum=156

Arrays and Pointers and Functions:

a. Write a C program to find the minimum, maximum and average in an array of integers.

```
#include <stdio.h>

int main()
{
    int a[50];
    int i, max, min, n;

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

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

    max = a[0];
    min = a[0];
```

```
   for(i=1; i<n; i++)
      {
         if(a[i] > max)
          {
             max = a[i];
          }

         if(a[i] < min)
          {
             min = a[i];
          }
      }
    printf("Maximum element = %d\n", max);
    printf("Minimum element = %d", min);

    return 0;
}
```

RESULT:

INPUT:

Enter size of the array: 5

Enter elements in the array: 1 2 3 4 5

OUTPUT:

Maximum element = 5

Minimum element = 1

b. Write a functions to compute mean, variance,

Standard Deviation, sorting of n elements in single dimension array.

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

void main()
{
float x[10], mean, variance,dev;
float sum1=0, sum2=0;
int n,i;
printf("Enter the number of integers:");
scanf("%d", &n);

printf("\nEnter the integers:");
for(i=0;i<=n-1;i++)
  {
    scanf("%f", &x[i]);
  }

//to find mean
for(i=0;i<=n-1;i++)
  {
    sum1 = sum1 + x[i];
  }

mean = sum1 /n;
printf("\nmean = %f", mean);
```

```
//to find variance
for(i=0;i<=n-1;i++)
  {
    sum2 = sum2 + (x[i] - mean) * (x[i] - mean);
  }
variance = sum2/n;
printf("\nvariance = %f", variance);

//to find deviation
dev = sqrt(variance);
printf("\ndeviation = %f", dev);

}
```

RESULT:

INPUT:

Enter the number of integers:5

Enter the integers:1 2 3 4 5

OUTPUT:

mean = 3.000000

variance = 2.000000

deviation = 1.414214

c. Write a C program that uses functions to perform the following:

i.Addition of Two Matrices

ii. Multiplication of Two Matrices

iii. Transpose of a matrix with memory dynamically allocated for the new matrix as row and column counts may not be same.

i.Addition of Two Matrices:

```
# include <stdio.h>
 # include <conio.h>
 void main ( )
 {
int i, j, a[3][3], b[3][3], sum[3][3];
      clrscr ( ) ;
printf ("Enter the elements of 1st array .\n") ;
      for (i=1;i<=3;i++)
```

```
	{
		for (j=1;j<=3;j++)
		{
			scanf ("%d", &a[i][j]) ;
		}
	}
printf ("Elements of 1st array are: \n") ;
	for (i=1;i<=3;i++)
	{
		for (j=1;j<=3;j++)
		{
			printf ("%d\t", a[i][j]) ;
		}
		printf ("\n") ;
	}
printf ("Enter the elements of 2nd array .\n") ;
	for (i=1;i<=3;i++)
	{
		for (j=1;j<=3;j++)
		{
			scanf ("%d", &b[i][j]) ;
		}
	}
printf ("Elements of 2nd array are: \n") ;
	for (i=1;i<=3;i++)
	{
		for (j=1;j<=3;j++)
		{
```

```
                printf ("%d\t", b[i][j]) ;
            }
            printf ("\n") ;
        }
printf ("The sum of both the matrices are: \n") ;
for (i=1;i<=3;i++)
        {
            for (j=1;j<=3;j++)
            {
                sum[i][j] = a[i][j] + b[i][j];
                printf ("%d\t", sum[i][j]) ;
            }
            printf ("\n") ;
        }
        getch ( ) ;
 }
```

----------OR----------

```
#include<stdio.h>
int read(int x[10][10],int m,int n);
int add(int a[10][10],int b[10][10],int c[10][10],int
m,int n);
int write(int x[10][10],int m,int n);

int main()
{
  int a[10][10],b[10][10],c[10][10],r1,r2,c1,c2;
  printf("\nEnter rows and columns of matrix1:\n");
```

```
  scanf("%d%d",&r1,&c1);
  printf("\nEnter rows and columns of matrix2:\n");
  scanf("%d%d",&r2,&c2);

  if((r1==r2)&&(c1==c2))
    {
      printf("\nEnter elements of matrix1:\n");
      read(a,r1,c1);
      printf("\nEnter elements of matrix2:\n");
      read(b,r2,c2);
      printf(“after adding a&b matrices\n”);

      add(a,b,c,r1,c1);
      write(c,r1,c1);
    }
  else
      printf("\n Addition is not possible\n");
}

int read(int x[10][10],int m,int n)
 {
   int i,j;
   for(i=0;i<m;i++)
     for(j=0;j<n;j++)
       scanf("%d",&x[i][j]);
 }

int add(int a[10][10],int b[10][10],int c[10][10],int
```

```
m,int n)
 {
    int i,j;
    for(i=0;i<m;i++)
      for(j=0;j<n;j++)
         c[i][j]=a[i][j]+b[i][j];
 }

int write(int c[10][10],int m,int n)
 {
    int i,j;
      for(i=0;i<m;i++)
       {
          printf("\n");
          for(j=0;j<n;j++)
             printf("\t%d",c[i][j]);
       }
 }
```

RESULT:

INPUT:

```
Enter rows and columns of matrix1
2   2
Enter rows and columns of matrix2
2  2
Enter elements of matrix1
```

1 2
3 4
Enter elements of matrix2
1 2
3 4

OUTPUT:

after adding a&b matrices
2 4
6 8

ii. Multiplication of two matrices

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

void main()
{
   int a[10][10],b[10][10],c[10][10],i,j,k,r1,c1,r2,c2;
   int sum=0;
```

```
    clrscr();

    printf("Enter number of rows and columns of first
matrix (MAX 10)\n");
    scanf("%d%d",&r1,&c1);
    printf("Enter number of rows and columns of
second matrix MAX 10)\n");
    scanf("%d%d",&r2,&c2);
    if(r2==c1)
    {

        printf("\n Enter First Matrix:");

        for(i=0; i<r1; i++)
        {
            for(j=0; j<c1; j++)
                scanf("%d",&a[i][j]);
        }

        printf("\n Enter Second Matrix: ");
        for(i=0; i<r2; i++)
        {
            for(j=0; j<c2; j++)
                scanf("%d",&b[i][j]);
        }

        printf("The First Matrix Is: \n");
        //print the first matrix
```

```
for(i=0; i<r1; i++)
{
    for(j=0; j<c1; j++)
        printf(" %d ",a[i][j]);
    printf("\n");
}

printf("The Second Matrix Is:\n");
// print the second matrix
for(i=0; i<r2; i++)
{
    for(j=0; j<c2; j++)
        printf(" %d ",b[i][j]);
    printf("\n");
}


printf("Multiplication of the Matrices:\n");

for(i=0; i<r1; i++)
{
    for(j=0; j<c2; j++)
    {
        c[i][j]=0;
        for(k=0; k<r1; k++)
            c[i][j]+=a[i][k]*b[k][j];
        printf("%d  ",c[i][j]);
    }
```

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

    }
    else
    {
        printf("Matrix Multiplication is Not Possible");
    }
    getch();
}
```

----------OR----------

```
#include<stdio.h>
int read(int x[10][10],int m,int n);
int mul(int a[10][10],int b[10][10], int c[10][10],int
r1,int c2,int r2);
int write(int x[10][10],int m,int n);

int main()
{
   int a[10][10],b[10][10],c[10][10],r1,r2,c1,c2;
   printf("\nEnter rows and columns of matrix1:\n");
   scanf("%d%d",&r1,&c1);
   printf("\nEnter rows and columns of matrix2:\n");
   scanf("%d%d",&r2,&c2);

   if(c1==r2)
    {
```

```
      printf("\nEnter elements of matrix1:\n");
      read(a,r1,c1);
      printf("\nEnter elements of matrix2:\n");
      read(b,r2,c2);
      printf("After multiplying a&b matrices\n");

      mul(a,b,c,r1,c2,r2);
      write(c,r1,c2);
    }
   else
      printf("\n Multiplication is not possible\n");
}


int read(int x[10][10],int m,int n)
 {
   int i,j;
   for(i=0;i<m;i++)
     for(j=0;j<n;j++)
       scanf("%d",&x[i][j]);
 }

int mul(int a[10][10],int b[10][10], int c[10][10],int
r1,int c2,int r2)
 {
   int i,j,k;
   for(i=0;i<r1;i++)
     for(j=0;j<c2;j++)
```

```
      {
        c[i][j]=0;
        for(k=0;k<r2;k++)
           c[i][j]=c[i][j]+a[i][k]*b[k][j];
      }
 }

int write(int c[10][10],int m,int n)
 {
   int i,j;
   for(i=0;i<m;i++)
     {
       printf("\n");
       for(j=0;j<n;j++)
           printf("\t%d",c[i][j]);
     }
 }
```

RESULT:

INPUT:

```
Enter rows and columns of matrix1
2   2
Enter rows and columns of matrix2
2  2
Enter elements of matrix1
```

1 2
3 4
Enter elements of matrix2
1 2
3 4

OUTPUT:

After multiplying a&b matrices
7 10
15 22

iii. Transpose of a matrix with memory dynamically allocated for the new matrix as row and column counts may not be same.

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

int main()
{
        int *a[10];
        int m,n,i,j;

        printf("\n Enter matrix order");
        scanf("%d%d",&m,&n);

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

            a[i]=(int *)malloc(n*sizeof(int));

        printf("\n Enter matrix elements:");
        for(i=0;i<m;i++)
        for(j=0;j<n;j++)
        scanf("%d",&a[i][j]);
```

```
        printf("\n The original matrix\n");
        for(i=0;i<m;i++)
        {
                for(j=0;j<n;j++)
                printf("%d\t",a[i][j]);
                printf("\n");
        }

        printf("\n The transpose of matrix\n");
        for(j=0;j<n;j++)
        {
                for(i=0;i<m;i++)
                printf("%d\t",a[i][j]);
                printf("\n");
        }
    return 0;
}
```

RESULT - 1:

INPUT:

Enter matrix order 2 3

Enter matrix elements: 1 2 3 4 5 6

OUTPUT:

The original matrix

1 2 3

4 5 6
The transpose of matrix
1 4
2 5
3 6

RESULT – 2:

INPUT:

Enter matrix order 3 2

Enter matrix elements: 1 2 3 4 5 6

OUTPUT:

The original matrix

1 2

3 4

5 6

The transpose of matrix

1 3 5

2 4 6

RESULT – 3:

INPUT:
Enter matrix order 2 2

Enter matrix elements:1 2 3 4

OUTPUT:

The original matrix

1 2

3 4

The transpose of matrix

1 3

2 4

d. Write C programs that use both recursive and non-recursive functions

i. To find the factorial of a given integer.

ii. To find the GCD (greatest common divisor) of two given integers.

iii. To find x^n

i. To find the factorial of a given integer using recursive function

and non-recursive functions:

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

```
int recfactorial(int );
int nonrecfactorial(int );

void main()
{
   int n, a, b;

   printf("Enter any number\n");
   scanf("%d", &n);

   a = recfactorial(n);
   printf("The factorial of a given number using
recursion is %d \n", a);

   b = nonrecfactorial(n);
   printf("The factorial of a given number using
nonrecursion is %d ", b);
}

int recfactorial(int x)
{
   int f;
   if(x == 0)
   {
    return(1);
   }
   else
```

```
   {
    f = x * recfactorial(x - 1);
    return(f);
   }
}

int nonrecfactorial(int x)
{
   int i, f = 1;
   for(i = 1;i <= x; i++)
   {
      f = f * i;
   }
   return(f);
}
```

RESULT:

INPUT:

Enter any number

5

OUTPUT:

The factorial of a given number using recursion is 120

The factorial of a given number using nonrecursion is 120

ii. To find the GCD (greatest common divisor) of two given integers
using recursive function and non-recursive functions:

```
#include <stdio.h>
int recgcd(int , int );
int nonrecgcd(int , int );

void main()
{
```

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

   printf("Enter two numbers a, b\n");
   scanf("%d%d", &a, &b);

   c = recgcd(a, b);
   printf("The gcd of two numbers using recursion is
%d\n", c);

   d = nonrecgcd(a, b);
   printf("The gcd of two numbers using nonrecursion
is %d", d);
}

int recgcd(int x, int y)
{
   if(y == 0)
   {
      return(x);
   }
   else
   {
     return(recgcd(y, x % y));
   }
}

int nonrecgcd(int x, int y)
{
```

```
   int z;
   while(x % y != 0)
   {
    z = x % y;
    x = y;
    y = z;
   }
   return(y);
}
```

RESULT:

INPUT:

Enter two numbers a, b 3 6

OUTPUT:

The gcd of two numbers using recursion is 3

The gcd of two numbers using nonrecursion is 3

iii. To find x^n using recursive function and non-recursive functions:

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

long recpower(int, int);
long nonrecpower(int , int );

int main()
```

```
{    int p, num;
     long result1, result2;
     printf("Enter a number : ");
    scanf("%d", & num);
    printf("Enter it's power: ");
    scanf("%d", &p);

   result1 = recpower(num, p);
   printf("Recursive function result :%d^%d is %ld",
num, p, result1);

   result2 = nonrecpower(num, p);
   printf("\nNon recursive function result: %d^%d is
%ld", num, p, result2);

   return 0;
}



long recpower(int num, int p)
{
  if (p==1)
   {
      return num;
     }
else
{
  return (num * recpower(num, p – 1));
```

```
}
   }
  return 1;
}


long nonrecpower(int num, int p)
{
    long power = 1;
    int i;

   for(i=1; i<=p; i++)
    {
        power = power * num;
    }

    return power;
}
```

RESULT:

INPUT:

Enter a number : 2

Enter it's power: 2

OUTPUT:

Recursive function result : 2^2 is 4
Non recursive function result : 2^2 is 4

e. Write a program for reading elements using pointer into array and display the values using array.

```
void main()
{
   int a[50],*p,i,n;
```

```
   p=a;

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

   printf("Enter elements of array:");

   for(i=0;i<n;++i)
      scanf("%d",p+i);

   for(i=0;i<n;++i)
      printf("%d\t",*(p+i));

}
```

RESULT:

INPUT:

Enter size of array: 5
Enter elements of array:2 5 8 9 1

OUTPUT:

2 5 8 9 1

f. Write a program for display values reverse order from array using pointer.

```
#include<stdio.h>
#define MAX 30

void main()
{
int size,i,arr[MAX];
int *ptr;
```

```
ptr=&arr[0];

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

printf("Enter %d integers into array: ",size);
   for(i=0;i<size;i++)
   {
     scanf("%d",ptr);
     ptr++;
   }

ptr=&arr[size-1];

printf("Elements of array in reverse order are:\n");

 for(i=size-1;i>=0;i--)
   {
     printf("\nElement%d is %d\n",i,*ptr);
     ptr--;
   }

}
```

RESULT:

INPUT:

Enter the size of array : 5
Enter 5 integers into array: 10 20 30 40 50

OUTPUT:

Elements of array in reverse order are:

Element4 is 50

Element3 is 40

Element2 is 30

Element1 is 20

Element0 is 10

g. Write a program through pointer variable to sum of n elements from array.

```
#include<stdio.h>
int main()
{
   int array[20];
   int n,i,sum=0;
   int *ptr;

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

```
   printf("\nEnter array elements:");
   for(i=0;i<5;i++)
      scanf("%d",&array[i]);

   /* array is equal to base address * array = &array[0]
*/

   ptr = array;

   for(i=0;i<n;i++)
   {
      //*ptr refers to the value at address
      sum = sum + *ptr;
      ptr++;
   }

   printf("\nThe sum is: %d",sum);
}
```

RESULT:

INPUT:

Enter the array size: 5

Enter array elements:1 2 3 4 5

OUTPUT:

The sum is: 15

Files:

a. Write a C program to display the contents of a file to standard output device.

```
#include<stdio.h>
#include<string.h>
void main()
{
char ch;
char str[20];
FILE *fp;

printf("Enter the source file name");
gets(str);
```

```
fp=fopen(str,"r");

if(fp==NULL)
  {
    printf("Error in opening file");
  }

while((ch=getc(fp))!=EOF)
  {
    putchar(ch);
  }
fclose(fp);
}
```

RESULT:

INPUT:

hello.txt (created file with text "this is kmit")

OUPUT:

```
Enter the source file name
hello.txt
this is kmit
```

b. Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents.

```
#include <stdio.h>
#include<stdlib.h>
void main()
{
   FILE *fp,*ft;
   char ch;

  fp=fopen("w.c","r");
  ft=fopen("ww.c","w");

  if(fp==NULL)
   {
     puts("Cannot open file");
     exit(0);
   }
```

```
  if(ft==NULL)
   {
      puts("Cannot open file");
   }

  while((ch=fgetc(fp))!=EOF)
   {
      if(ch>=97&&ch<=122)
      ch=ch-32;
      fputc(ch,ft);
   }
  fclose(fp);
  fclose(ft);
}
```

RESULT:

INPUT:

```
//w.c created file with following c program code
#include<stdio.h>
void main()
{
int a=10,b=20;
printf("addition of a,b=%d",a+b);
}
```

OUTPUT:

```
//ww.c generated by compiler with following code by
```

convert lowercase letters to uppercase letters

```
#INCLUDE<STDIO.H>
VOID MAIN()
{
INT A=10,B=20;
PRINTF("ADDITION OF A,B=%D",A+B);   }
```

c. Write a C program to count the number of times a character occurs in a text file. The file name and the character are supplied as command line arguments.

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

int main(int argc, char *argv[])
{
    FILE *finp   = NULL;
    FILE *output = NULL;

    int   letter; // character to be found
    int   ex=0; // character occurrence counter

    if( argc!=4 )
      {
         printf( "pass correct way of command line
arguments like inputfile outputfile character want
```

```
                to search\n");
    exit(0);
  }


  finp=fopen(argv[1], "r");

if( finp==NULL)
  {
     perror( "fopen for the input file failed" );
     exit(0);
  }


output=fopen(argv[2], "w");

if(output==NULL )
  {
    perror( "fopen for the output file failed" );
    fclose( finp );
    exit(0);
  }

while(  (letter = fgetc(finp) )!=EOF)
  {
    if( argv[3][0]==letter)  /*From ASCII TABLE*/
       {
           ex++;
```

```
        }
    }


    fprintf(output,"The search character is '%c' and it occurred %d times\n", argv[3][0], ex);
    fclose(finp);
    fclose(output);

    return 0;
}
```

RESULT:

OUTPUT:

$./a.out hi.txt hi2.txt s
(hi.txt file contained text: hello studentsssss.
As a result: hi2.txt got: The search character is 's' and it occurred 6 times)

d. Write a C program that does the following:

It should first create a binary file and store 10 integers, where the file name and 10 values are given in the command line. (hint: convert the strings using atoi function)

Now the program asks for an index and a value from the user and the value at that index should be changed to the new value in the file. (hint: use fseek function)

The program should then read all 10 values and print them back.

e. Write a C program to merge two files into a third file (i.e., the contents of the firs t file followed by those of the second are put in the third file).

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

void main()
{
   // Open two files to be merged
      FILE *fp1 = fopen("w.c", "r");
      FILE *fp2 = fopen("ww.c", "r");

   // Open file to store the result
      FILE *fp3 = fopen("r.c", "w");

   char c;
```

```
   if (fp1 == NULL || fp2 == NULL || fp3 == NULL)
    {
       puts("Could not open files");
       exit(0);
    }

   // Copy contents of first file to file3.txt
      while ((c = fgetc(fp1)) != EOF)
         fputc(c, fp3);

   // Copy contents of second file to file3.txt
      while ((c = fgetc(fp2)) != EOF)
         fputc(c, fp3);

   printf("Merged w.c and ww.c into r.c");
   fclose(fp1);
   fclose(fp2);
   fclose(fp3);
}
```

RESULT:

INPUT:

w.c has following code

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

```
void main()
{
int a=10,b=20;
printf("addition of a,b=%d",a+b);
}
```

ww.c has following code

```
#INCLUDE<STDIO.H>
VOID MAIN()
{
INT A=10,B=20;
PRINTF("ADDITION OF A,B=%D",A+B);
}
```

OUTPUT:

Merged w.c and ww.c into r.c.
r.c has following code

```
#include<stdio.h>
void main()
{
int a=10,b=20;
printf("addition of a,b=%d",a+b);
}
#INCLUDE<STDIO.H>
VOID MAIN()
{
```

```
INT A=10,B=20;
PRINTF("ADDITION OF A,B=%D",A+B);
}
```

Strings:

a. Write a C program to convert a Roman numeral ranging from I to L to its decimal equivalent.

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

void main()
{
   char rom[30];
   int a[30], l, i, k, dec;

   printf("Enter the roman number\n");
   scanf("%s", &rom);

   l =strlen(rom);
   for(i = 0; i < l; i++)
   {
      switch (rom[i])
```

```
   {
      case 'I':  a[i] = 1;
                 break;
     case 'V':  a[i] = 5;
                 break;
     case 'X':  a[i] = 10;
                 break;
     case 'L':  a[i] = 50;
                 break;
     case 'C':  a[i] = 100;
                break;
     case 'D':  dec = dec + 500;
                break;
     case 'M':  a[i] = 1000;
                break;
     default :  printf("Invalid choice");
                break;
   }
}

k = a[l - 1];
for(i = l - 1; i > 0; i--)
  {
    if(a[i] > a[i - 1])
      {
         k = k - a[i - 1];
      }
```

```
        if(a[i] <= a[i - 1])
          {
             k = k + a[i - 1];
          }
      }

   printf("decimal equivalent is %d", k);
}
```

RESULT:

INPUT:

Enter the roman number

XIV

OUTPUT:

decimal equivalent is 14

b. Write a C program that converts a number

ranging from 1 to 50 to Roman equivalent

```
#include<stdio.h>

void main()
{
   int a,b,e;

   printf("Input a  number (between 1-50):");
   scanf("%d",&e);

  while (e==0||e>50)
   {
      printf ("ERROR: Invalid Input!");
      printf ("\nEnter the number again:");
      scanf ("%d",&e);
   }

a = ((e/10)%10)*10;
b = ((e/1)%10)*1;



if (a == 10)
            printf("X");
else if (a == 20)
            printf("XX");
else if (a == 30)
            printf("XXX");
```

```
else if (a == 40)
            printf("XL");
else if (a ==50)
            printf("L");

if (b == 1)
            printf("I");
else if (b == 2)
            printf("II");
else if (b == 3)
            printf("III");
else if (b == 4)
            printf("IV");
else if (b ==5)
            printf("V");
else if (b == 6)
            printf("VI");
else if (b == 7)
            printf("VII");
else if (b ==8)
            printf("VIII");
else if (b == 9)
            printf("IX");
}
```

RESULT:

INPUT:

Input a number (between 1-50): 51

ERROR: Invalid Input!
Enter the number again:50

OUTPUT:

L

c. Write a C program that uses functions to perform the following operations:

i. To insert a sub-string in to a given main string from a given position.

ii. To delete n Characters from a given position in a given string.

i. To insert a sub-string in to a given main string from a given position.

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

void insstr(int);

char MS[20], SS[20], TS[50];
int i,j,pos;

void main()
{
  puts("Enter First String=\n");
  gets(MS);

  puts("Enter Second String=\n");
  gets(SS);

  printf("Enter the position where the item has to be
inserted=\n ");
  scanf("%d",&pos);

  insstr(pos);
}
```

```
void insstr(int pos)
{
  for(i=0;i<pos;i++)
    {
      TS[i]=MS[i];
    }

  for(j=0;SS[j]!='\0';j++)
    {
      TS[i]=SS[j];
      i++;
    }

 for(j=pos;MS[j]!='\0';j++)
   {
     TS[i]=MS[j];
     i++;
   }

 TS[i]='\0';
 printf("The final string is=\n");
 puts(TS);
}
```

RESULT:

INPUT:

Enter First String=
sai

Enter Second String=
ram
Enter the position where the item has to be inserted=
3

OUTPUT:

The final string is=
sairam

ii. To delete n Characters from a given position in a given string.

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

```
void delstr(int n,int pos);

char str[15];
int i,j;

void main()
{
   int pos,n;

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

   printf("Enter the position in string from where to
delete");
   scanf("%d", &pos);

   printf("Enter no of char to be deleted");
   scanf("%d",&n);

   delstr(n,pos);
}

void delstr(int n,int pos)
{
  i=pos;

  for(j=pos+n;str[j]!='\0';j++)
   {
```

```
    str[i]=str[j];
    i++;
  }
  str[i]='\0';
  printf("\n string after deletion=%s",str);
}
```

RESULT:

INPUT:

Enter the string:
kmitcollege
Enter the position in string from where to delete 4
Enter no of char to be deleted 7

OUTPUT:

string after deletion=kmit

d. Write a C program to determine if the given string is a palindrome or not (Spelled same in both directions with or without a meaning like madam, civic, noon, abcba, etc.)

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

```
int main()
{
      char str[100];
      int i, len, flag;
      flag = 0;

      printf("\n Please Enter any String :  ");
      gets(str);

      len = strlen(str);

      for(i = 0; i < len; i++)
        {
           if(str[i] != str[len - i - 1])
             {
                flag = 1;
                break;
             }
       }
      if(flag == 0)
        {
           printf("\n %s is a Palindrome String", str);
        }
      else
        {
           printf("\n %s is Not a Palindrome String",
str);
```

```
        }

    return 0;
}
```

RESULT:

INPUT:

Please Enter any String : MALAYALAM

OUTPUT:

MALAYALAM is a Palindrome String

e. Write a C program that displays the position of a character ch in the string S or – 1 if S doesn‘t contain ch.

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

```
void main()
{
   char s[30], t[20];
   char *found;

   puts("Enter the first string: ");
   gets(s);

   puts("Enter the string to be searched: ");
   gets(t);

   found = strstr(s, t);

   if(found)
     {
        printf("Second String is found in the First String
at %d position.\n", found - s);
     }
   else
     {
        printf("-1");
     }
}
```

RESULT - 1:

INPUT:

Enter the first string:
sai ram
Enter the string to be searched:
ram

OUTPUT:

Second String is found in the First String at 4 position.

RESULT - 2:

INPUT:

Enter the first string:
sai ram
Enter the string to be searched:
raj

OUTPUT: -1

f. Write a C program to count the lines, words and characters in a given text.

```
#include<stdio.h>

void main()
{
   char line[81], a;
   int i,c,
```

```
    end = 0,
    characters = 0,
    words = 0,
    lines = 0;

   printf("TYPE ANY  TEXT.\n");
   printf("GIVE ONE SPACE AFTER EACH
WORD.\n");
   printf("WHEN COMPLETED, PRESS
'ctrl+z'.\n\n");

   while( end == 0)
      {
              /* Reading a line of text */
              c = 0;

               while((a=getchar()) != '\n')
                        line[c++] = a;
                     line[c] = '\0';

               /* counting the words in a line */
               if(line[0] == '\0')
                           break ;
               else
                     {
                          words++;
                           for(i=0; line[i] != '\0';i++)
                                   if(line[i] == ' ' || line[i] == '\t')
```

```
                                words++;
                }

                /* counting lines and characters */
                lines = lines +1;
                characters = characters + strlen(line);
        }

    printf ("\n");
    printf("Number of lines = %d\n", lines);
    printf("Number of words = %d\n", words);
    printf("Number of characters = %d\n", characters);
}
```

RESULT:

INPUT:

TYPE ANY TEXT.
GIVE ONE SPACE AFTER EACH WORD.
WHEN COMPLETED, PRESS 'ctrl+z'.

hi how r u
what r u doing
n where r u

^Z

OUTPUT:
Number of lines = 3
Number of words = 12
Number of characters = 35

Miscellaneous:

a. Write a menu driven C program that allows a user to enter n numbers and then choose between finding the smallest, largest, sum, or average. The menu and all the choices are to be functions. Use a switch statement to determine what action to take. Display an error message if an invalid choice is entered.

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

```
int main()
{
   int n,i,a[100];
   int ch;

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

   printf(“Enter elements in to array\n”);
   for(i=0;i<n;i++)
        scanf("%d",&a[i]);

   printf("\n\tMENU\n");
   printf("\n\t1.largest");
   printf("\n\t2.Smallest");
   printf("\n\t3.sum");
   printf("\n\t4.avg");
   printf("\n\n\tEnter your choice:");
   scanf("%d",&ch);

  switch(ch)
     {
       case 1:   largest(a,n);
                 break;
       case 2:   smallest(a,n);
                 break;
       case 3:   sum(a,n);
                  break;
```

```
        case 4:   avg(a,n);
                  break;
       default:   printf("\n Invalid Choice");
     }
}

int  largest(int x[],int k)
  {
    int t,i;
       t=x[0];
       for(i=1;i<k;i++)
          {
            if(x[i]>t)
             t=x[i];
          }
       printf("largest=%d",t);
  }

int smallest(int x[],int k)
  {
    int t,i;
       t=x[0];
       for(i=1;i<k;i++)
          {
            if(x[i]<t)
              t=x[i];
          }
       printf("smallest=%d",t);
```

```
  }

int sum(int x[],int k)
  {
    int i,s=0;
    for(i=0;i<k;i++)
      {
        s=s+x[i];
      }
      printf("sum=%d",s);
  }

int avg(int x[],int k)
  {
    int a,i,s=0;
    for(i=0;i<k;i++)
      {
         s=s+x[i];
      }
      a=s/k;
      printf("average =%d",a);
  }
```

RESULT:

INPUT:

Enter the size of array : 5
Enter elements in to array
2 4 6 3 1

MENU

1.largest
2.Smallest
3.sum
4.avg
Enter your choice: 1

OUTPUT:

largest=6

b. Write a C program to construct a pyramid of numbers as follows:

```
1          *          1          1          *
1 2        * *        2 3        2 2        * *
1 2 3      * * *      4 5 6      3 3 3      * * *
                                 4 4 4 4    * *
                                            *
```

I. Program to print half pyramid a using numbers.

```
#include <stdio.h>
int main()
{
    int i, j, rows;
```

```
    printf("Enter number of rows: ");
    scanf("%d",&rows);

    for(i=1; i<=rows; ++i)
    {
        for(j=1; j<=i; ++j)
        {
            printf("%d ",j);
        }
        printf("\n");
    }
    return 0;
}
```

RESULT:

INPUT:
Enter number of rows: 3

OUTPUT:
1
1 2
1 2 3

II. Program to print half pyramid using *.

```
#include <stdio.h>
int main()
{
    int i, j, rows;

    printf("Enter number of rows: ");
    scanf("%d",&rows);

    for(i=1; i<=rows; ++i)
    {
        for(j=1; j<=i; ++j)
        {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

RESULT:

INPUT:
Enter number of rows: 3

OUTPUT:
*
* *
* * *

III. Print Floyd's Triangle.

```
#include <stdio.h>
int main()
{
    int rows, i, j, number= 1;

    printf("Enter number of rows: ");
    scanf("%d",&rows);

    for(i=1; i <= rows; i++)
    {
        for(j=1; j <= i; ++j)
        {
            printf("%d ", number);
            ++number;
        }

        printf("\n");
    }

    return 0;
}
```

RESULT:

INPUT:
Enter number of rows: 3

OUTPUT:
1
2 3
4 5 6

IV. Program to print half pyramid using numbers.

```
#include<stdio.h>
int main()
{
  int n,i,j;

  printf("Enter size");
  scanf("%d",&n);

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

RESULT:

INPUT:
Enter size 4

OUTPUT:

1
2 2
3 3 3
4 4 4 4

v. Print the following pattern.

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

int main()
{
    char str[]="***";
    int i,j, len;

    len = strlen(str);

    for(i=0; i<len; i++)
    {
        for(j=0; j<=i; j++)
            printf("%c\t",str[j]);
        printf("\n");
    }

    for(i=len-2; i>=0; i--)
    {
        for(j=0; j<=i; j++)
            printf("%c\t",str[j]);
        printf("\n");
    }

    return 0;
}
```

RESULT:

OUTPUT:

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

Sorting and Searching:

a. Write a C program that uses non recursive function to search for a Key value in a given list of integers using linear search method.

```
#include<stdio.h>

int lsearch(int a[],int,int);

int main()
 {
   int i,j,a[100],key,n;

   printf("Number of elements in the array\n");
   scanf("%d",&n);

   printf("\nEnter %d elements\n",n);

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

   printf("Element to be searched\n");
   scanf("%d",&key);

   lsearch(a,n,key);
 }

int lsearch(int a[],int n,int key)
 {
   int flag=0,i;
   for(i=0;i<n;i++)
```

```
    if(key==a[i])
     {
       flag=1;
       break;
     }

  if(flag== 0)
        printf("key value not found");
  else
        printf("key value found at %d index",i);
}
```

RESULT:

INPUT:
Number of elements in the array
10

Enter 10 elements
2 5 3 6 7 8 12 10 9 15
Element to be searched
8

OUTPUT:
key value found at 5 index

b. Write a C program that uses non recursive function to search for a Key value in a given sorted list of integers using binary search method.

```
#include<stdio.h>

int  bsearch(int a[],int,int);

int main()
{
   int i,a[100],key,n;

   printf("Enter Number of elements in the array:");
   scanf("%d",&n);

   printf("\nEnter %d elements in sorted order\n",n);
   for(i=0;i<n;i++)
       scanf("%d",&a[i]);

   printf("Element to be searched\n");
   scanf("%d",&key);

   bsearch(a,n,key);
}

int bsearch(int a[],int n,int key)
{
```

```
   int flag=0,low=0,mid,high=n-1;
   while(low<=high)
    {
      mid=(low+high)/2;
      if(key==a[mid])
       {
         flag=1;
         break;
       }

      else if(key<a[mid])
         high=mid-1;

      else if(key>a[mid])
         low=mid+1;
    }

   if(flag==0)
      printf("key value not found");
   else
      printf("key value found at %d index",mid);
}
```

RESULT:

INPUT:

Enter Number of elements in the array: 7

Enter 7 elements in sorted order
12 14 15 18 20 23 65
Element to be searched
23

OUTPUT:
key value found at 5 index

c. Write a C program that implements the Bubble sort method to sort a given list of integers in ascending order.

```
#include <stdio.h>
int main()
{
  int array[100], n, i, j, temp;

  printf("Enter number of elements\n");
  scanf("%d", &n);
```

```
  printf("Enter %d integers\n", n);
  for (i = 0; i < n; i++)
     scanf("%d", &array[i]);

  for(i = 0; i<(n-1); i++)
   {
     for(j = 0;j<n-i-1; j++)
      {
         if(array[j] > array[j+1])
           {
              temp  = array[j];
              array[j]   = array[j+1];
              array[j+1] =temp;
           }
      }
   }

  printf("Sorted list in ascending order:\n");
  for ( i= 0 ; i < n ; i++ )
      printf("%d\t", array[i]);
}
```

RESULT:

INPUT:

Enter number of elements

10
Enter 10 integers
12 1 4 7 23 76 45 21 54 47

OUTPUT:
Sorted list in ascending order:
1 4 7 12 21 23 45 47 54 76

d. Write a C program that sorts the given array of integers using selection sort in descending order.

```
#include <stdio.h>

int main()
{
	int a[10] = { 3,4,7,6,5,1,2,8,10,9 };
	int n = 10;
	printf("\nArray Data : ");

	for (int i = 0; i < n; i++)   //Loop for displaying the data of array
```

```
        {
          printf(" %d ", a[i]);
        }

      for (int i = 0; i < n; i++)   //Loop for descending
ordering
        {
          for (int j = 0; j < n; j++)
           {
             if (a[j] < a[i])
               {
                int tmp = a[i];
                a[i] = a[j];
                a[j] = tmp;
               }
           }
        }

      printf("\n\nDescending : ");
      for (int i = 0; i < n; i++)
        {
          printf(" %d ", a[i]);
        }

    return 0;
}
```

RESULT:

OUTPUT:

Array Data : 3 4 7 6 5 1 2 8 10 9

Descending : 10 9 8 7 6 5 4 3 2 1

e. Write a C program that sorts the given array of integers using insertion sort in ascending order

```
#include<stdio.h>

int main()
{
    int data[100],n,temp,i,j;

    printf("Enter number of terms(should be less than 100): ");
    scanf("%d",&n);

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

```
      for(i=0;i<n;i++)
        {
          scanf("%d",&data[i]);
        }

       for(i=1;i<n;i++)
        {
          temp = data[i];
          j=i-1;
          while(temp<data[j] && j>=0)
/*To sort elements in descending order, change
temp<data[j] to temp>data[j] in above line.*/
              {
                data[j+1] = data[j];
                --j;
              }
          data[j+1]=temp;
        }

       printf("In ascending order: ");
      for(i=0; i<n; i++)
          printf("%d\t",data[i]);
   return 0;
}
```

RESULT:

INPUT:
Enter number of terms(should be less than 100): 5
Enter elements:
6 8 4 10 99

OUTPUT:
In ascending order: 4   6   8   10   99

f. Write a C program that sorts a given array of names.

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

void main()
   {

       char name[10][8], tname[10][8], temp[8];
       int i, j, n;

       printf("Enter the value of n \n");
       scanf("%d", &n);
```

```
printf("Enter %d names \n",n);

for (i = 0; i < n; i++)
  {
    scanf("%s", name[i]);
    strcpy(tname[i], name[i]);
  }

for (i = 0; i < n - 1 ; i++)
 {
    for (j = i + 1; j < n; j++)
     {
        if (strcmp(name[i], name[j]) > 0)
          {
            strcpy(temp, name[i]);
            strcpy(name[i], name[j]);
            strcpy(name[j], temp);
          }
     }
 }

printf("\n-------------------------------------\n");
printf("Input Names\tSorted names\n");
printf("-------------------------------------\n");

for (i = 0; i < n; i++)
 {
    printf("%s\t\t%s\n", tname[i], name[i]);
```

```
        }

        printf("---------------------------------------\n");

    }
```

RESULT:

INPUT:

```
Enter the value of n
7
Enter 7 names
heap
queue
stack
object
class
program
project
```

OUTPUT:

```
-------------------------------------
Input Names    Sorted names
-------------------------------------
heap           class
```

queue heap
stack object
object program
class project
program queue
project stack
