**1) AIM: Write a Program that prints the results of all the Operators available in C**

**PROGRAM:**

#include <stdio.h>

void main() {

int a, b, result, result2;

printf("Enter a and b values : ");

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

printf("Addition of a and b : %d\n",a+b ); // Addition of a and b

printf("Substraction of a and b : %d\n",a-b ); // Subtract b from a

printf("Multiplication of a and b : %d\n",a\*b ); // Multiply a and b

printf("Remainder of a and b : %d\n",a%b ); // Numerator is a and denominator is b

printf("Division of a and b : %d\n",a/b ); // Numerator is a and denominator is b

result =(a>0&b<=10) ;

//The condition is a is greater than 0 and b is less than or equal to 10 printf("Logical AND result : %d\n", result);

result2 = (a==b|b!=0); // The condition is a is equal to b or b is not equal to 0

printf("Logical OR and NOT result : %d\n",result2);

printf("Bitwise AND : %d\n",a&b ); // Bitwise and of a and b

printf("Bitwise OR : %d\n",a|b ); // Bitwise or of a and b

printf("Bitwise NOT : %d\n",!a ); // Bitwise not of a

printf("Bitwise complement : %d\n",~a ); // Complement of a

printf("Bitwise XOR : %d\n",a^b ); // Bitwise xor of a and b

printf("Bitwise shift right : %d\n",a>>2 ); // Right shift a by 2 times

printf("Bitwise shift left : %d\n",b<<3 ); // Left shift b by 3 times

printf("Integer size : %d, Floating point size : %d\n",sizeof(int),sizeof(float) );

// Sizes of int and float

printf("Conditional expression : %d\n",(a-b)>50?100:200 );

// if (a - b) is greater than 50 then it is 100, otherwise 200

printf("Preincrement : %d\n",++a ); // Pre increment a

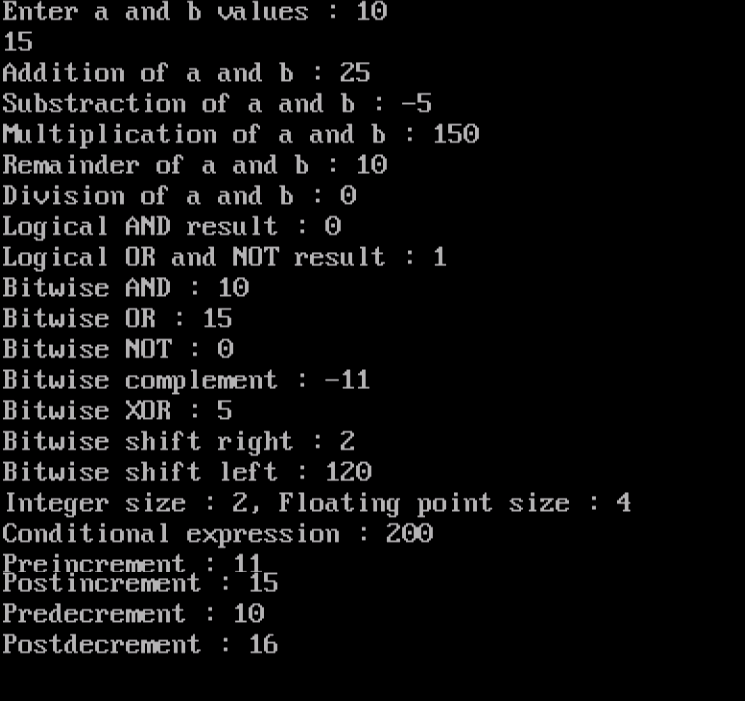
printf("Postincrement : %d\n",b++ ); // Post increment b

printf("Predecrement : %d\n",--a ); // Pre decrement a

printf("Postdecrement : %d\n",b-- ); // Post decrement b

}

**OUTPUT:**

****

**2) AIM: Write a program to demonstrate the processes of implicit type**

**conversion and explicit type conversion.**

**PROGRAM :**

#include <stdio.h>

void main() {

int a;

float b,x,y,z;

printf("Enter an integer value : ");

scanf("%d",&a);

printf("Enter a float value : ");

scanf("%f",&b);

x=a+b;

y=b\*2;

z=a/5.0;

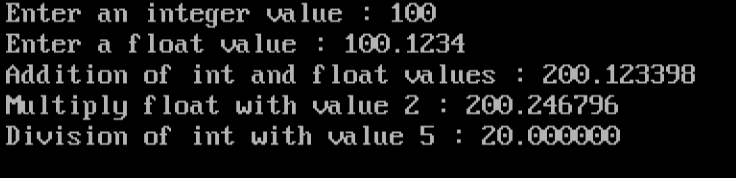
printf("Addition of int and float values : %f\n",x);

printf("Multiply float with value 2 : %f\n",y);

printf("Division of int with value 5 : %f\n",z);

}

**OUTPUT:**

****

**3) AIM : Write a program to find the largest and smallest of the three given integers.**

**PROGRAM :**

#include<stdio.h>

void main() {

int a,b,c;

printf("Enter 3 integer numbers : ");

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

if(a>b&&a>c)

printf("%d is the largest number\n",a);

else if(b>a&&b>c)

printf("%d is the largest number\n",b);

else

printf("%d is the largest number\n",c);

if(a<b&&a<c)

printf("%d is the smallest number\n",a);

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

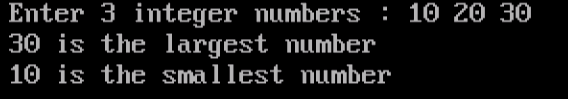
printf("%d is the smallest number\n",b);

else

printf("%d is the smallest number\n",c);

}

**OUTPUT:**



**4) AIM : Write a program to calculate the simple interest and compound interest by**

**reading principle amount, rate of interest and time.**

**PROGRAM :**

#include<stdio.h>

#include<math.h>

void main()

{

int p;

float simintr,compintr,r,q,t;

printf("Enter principle amount, rate of interest, time : ");

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

simintr=(p\*t\*r)/100;

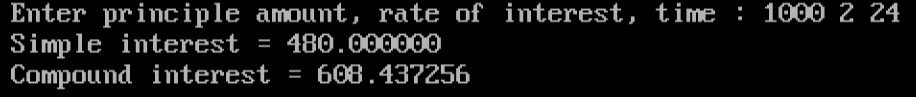
compintr=p\*pow(1+(r/100),t)-p;

printf("Simple interest = %f\n",simintr);

printf("Compound interest = %f\n",compintr);

}

**OUTPUT:**



**5)AIM :Write program that declares grade awarded for a given percentage of marks,**

**where**

**marks >= 90% is grade A**

**marks >= 80% and < 90% is grade B.**

**marks >= 70% and < 80% is grade C.**

**marks >= 60% and < 70% is grade D.**

**marks >= 40% and < 60% is grade E.**

**marks < 40% is grade Fail.**

**Read marks of five different subjects english, chemistry, computers, physics**

**and maths from standard input, then find total and percentage.**

**PROGRAM:**

#include<stdio.h>

main()

{

int a,b,c,d,e,t;

float p;

printf("Enter the five subjects marks : ");

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

t=a+b+c+d+e;

p=(t/5.0);

printf("Total marks : %d",t);

printf("\nPercentage : %f\n",p);

if(p>=90)

printf("Grade : A\n");

else if(p>=80&&p<90)

printf("Grade : B\n");

else if(p>=70&&p<80)

printf("Grade : C\n");

else if(p>=60&&p<70)

printf("Grade : D\n");

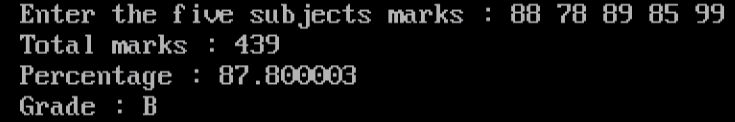
else if(p>=40&&p<60)

printf("Grade : E\n");

else

printf("Grade : Fail\n");

}

**OUTPUT:**

**6)AIM: Write a program to print the multiplication table for a given number with the**

**number of rows in the table.**

**PROGRAM :**

#include<stdio.h>

main()

{

int n,r,i;

printf("Enter an integer number : ");

scanf("%d",&n);

printf("Enter number of rows : ");

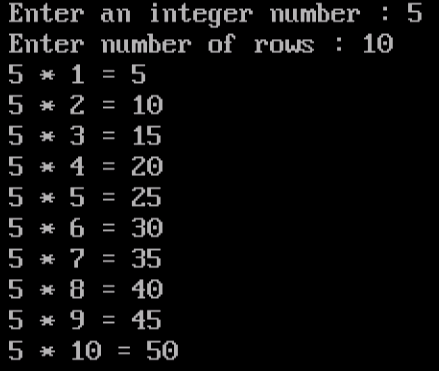
scanf("%d",&r);

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

printf("%d \* %d = %d\n",n,i,n\*i);

}

**OUTPUT:**



**7)AIM: Write a program that shows the binary equivalent of a given**

**positive decimal number between 0 to 255.**

**PROGRAM :**

#include<stdio.h>

void main()

{

int number,bi=0,rem,temp=1;

printf("Enter a decimal number : ");

scanf("%d",&number);

while(number!=0)

{

rem=number%2;

bi=bi+rem\*temp;

temp=temp\*10;

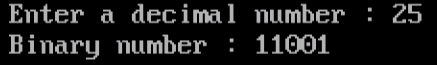
number=number/2;

}

printf("Binary number : %d\n",bi);

}

**OUTPUT:**

****

**8)AIM: A building has F floors with a floor height of H 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+(12)at2s = ut+(12)at2 where u and a are the initial velocity in m/sec (=0) and acceleration = 9.8 msec2msec2.**

**PROGRAM:**

**#include <stdio.h>**

**#include <math.h>**

**#define a 9.8**

**void main()**

**{**

**int n,h,i,s;**

**float t;**

**printf("Enter number of the floors : ");**

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

**printf("Enter height of each floor : ");**

**scanf("%d",&h);**

**for(i=n,s=h;s<=(n+1)\*h;s=s+h,i--)**

**{**

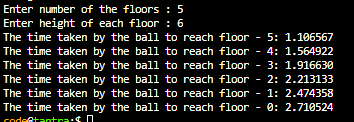
**t=sqrt((2\*s)/a);**

**printf("The time taken by the ball to reach floor - %d: %f\n",i,t);**

**}**

**}**

**OUTPUT:**

****

**9) AIM : Write a program to read two integer values and an arithmetic operator,**

**depending on the operator perform different arithmetic operations.**

**PROGRAM :**

#include<stdio.h>

int main()

{

int a,b;

char ch;

printf("Enter two integer values : ");

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

printf("Enter an arithmetic operator : ");

scanf(" %c",&ch);

switch(ch)

{

case '+' :

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

break;

case '-' :

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

break;

case '\*' :

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

break;

case '/' :

if(b==0)

{

printf("Division is not possible! Divide by zero error\n");

}

else

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

break;

case '%' :

if(b==0)

{

printf("Modulo division is not possible! Divide by zero error\n");}

else

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

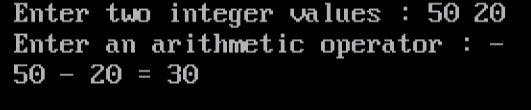
break;

default:printf("Error! Operator is not correct\n");

}

}

**OUTPUT:**



**10) AIM: Write a sample program to check whether the given number is a prime**

**number or not.**

**PROGRAM:**

#include <stdio.h>

int main()

{

int n, i, flag = 0;

printf("Enter any number : ");

scanf("%d", &n);

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

{

// condition for nonprime number

if(n%i == 0)

{

flag = 1;

break;

}

}

//if (n == 1)

//{

//printf("1 is neither a prime nor a composite number.");

//}

//else

//{

if (flag == 0)

printf("The given number %d is a prime number\n", n);

else

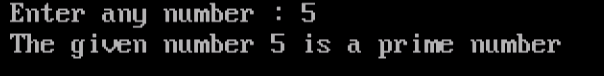
printf("The given number %d is not a prime number\n", n);

//}

return 0;

}

**OUTPUT:**

****

**11)AIM : Write a program to find the sum of individual digits of a given number.**

**PROGRAM:**

#include<stdio.h>

void main() {

int n,sum=0,rev=0,temp;

printf("Enter an integer : ");

scanf("%d",&n);

temp=n;

while(n>0)

{

rev=n%10;

sum=sum+rev;

n=n/10;

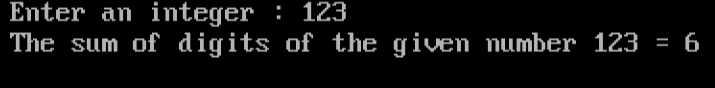
// printf("The sum of digits of the given number %d = %d\n",n,sum);

}

printf("The sum of digits of the given number %d = %d\n",temp,sum);

}

**OUTPUT:**

****

**12) AIM: Write a program to find the reverse of an integer number and check whether it**

**is Palindrome or not.**

**PROGRAM:**

#include <stdio.h>

int main()

{

int n, reversedInteger = 0, remainder, originalInteger;

printf("Enter an integer : ");

scanf("%d", &n);

originalInteger = n;

while( n!=0 )

{

remainder = n%10;

reversedInteger = reversedInteger\*10 + remainder;

n /= 10;

}

if (originalInteger == reversedInteger){

printf("The reverse of a given number : %d\n",reversedInteger);

printf("%d is a palindrome\n", originalInteger);}

else{

printf("The reverse of a given number : %d\n",reversedInteger);

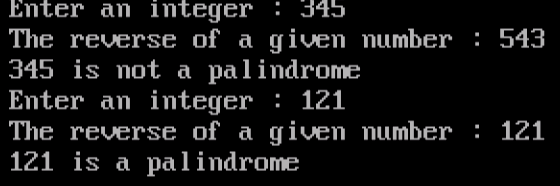
printf("%d is not a palindrome\n", originalInteger);

}

return 0;

}

**OUTPUT:**

****

**13) AIM: Write a program to print the Fibonacci seriesi.e., 0 1 1 2 3 5 8 13 21.....,**

**PROGRAM:**

int main()

{

int i, n, t1 = 0, t2 = 1, nextTerm;

printf("Enter the maximum limit to generate the Fibonacci series : ");

scanf("%d",&n);

printf("The Fibonacci series is :");

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

{

if(n>=t1){

printf(" %d",t1);

nextTerm = t1 + t2;

t1 = t2;

t2 = nextTerm;}

}

return 0;

}

**OUTPUT:**



**14) AIM: Write a sample code to find all the prime numbers between the limits.**

**PROGRAM:**

#include<stdio.h>

int main()

{

int i, prime,upper,lower, n;

printf("Enter lower and upper limits : ");

scanf("%d", &lower);

scanf("%d", &upper);

printf("Prime numbers between %d and %d are : ",lower,upper);

for(n=lower; n<=upper; n++)

{

prime = 1;

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

if(n%i == 0)

{

prime = 0;

break;

}

if(prime==1)

printf("%d ", n);

}

}

**OUTPUT:**



**15) AIM: Write a C program to find all Roots of a Quadratic equation**

**PROGRAM:**

#include <stdio.h>

#include <math.h>

int main()

{

double x,a, b, c, discriminant, root1, root2, realPart, imaginaryPart;

printf("Enter coefficients a, b and c : ");

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

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

// condition for real and different roots

if(a==0&&b==0&&c==0)

printf("Invalid coefficients\nEnter valid inputs\n");

else if(a==0){

x=(float)-c/b;

printf("Linear equation\nRoot = %lf\n",x);

}

else if (discriminant > 0)

{

// sqrt() function returns square root

root1 = (-b+sqrt(discriminant))/(2\*a);

root2 = (-b-sqrt(discriminant))/(2\*a);

printf("The roots are real and distinct\nroot1 = %f and root2 = %f\n",root1 , root2);

}

//condition for real and equal roots

else if (discriminant == 0)

{

root1 = root2 = -b/(2\*a);

printf("The roots are real and equal\nroot1 = root2 = %lf\n", root1);

}

// if roots are not real

else

{

realPart = -b/(2\*a);

imaginaryPart = sqrt(-discriminant)/(2\*a);

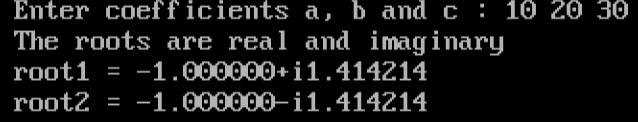
printf("The roots are real and imaginary\nroot1 = %lf+i%lf\nroot2 = %f-i%f\n", realPart, imaginaryPart, realPart, imaginaryPart);

}

return 0;

}

**OUTPUT:**

****

**16)AIM : Write a C program to calculate the following series value, where x is a fractional**

**value. 1- X2/2! + X4/4! - X6/6! + X8/8! - X10/10!**

**PROGRAM:**

#include <stdio.h>

#include <math.h>

main()

{

int counter,f\_coun;

float sum=0,x,power,fact;

// printf("\tEQUATION SERIES : 1- X^2/2! + X^4/4! - X^6/6! + X^8/8! - X^10/10!");

printf("Enter value of x : ");

scanf("%f",&x);

for(counter=0,power=0; power<=10; counter++,power=power+2)

{

fact=1;

//Factorial of POWER value.

for(f\_coun=power; f\_coun>=1; f\_coun--)

fact \*= f\_coun;

//The main equation for sum of series is...

sum=(sum+(pow(-1,counter)\*(pow(x,power)/fact)));

}

printf("Sum of series : %f\n",sum);

}

**OUTPUT:**



**17) AIM: Write a C program to read two numbers, x and n, and then compute the sum of**

**the series (1 + x + x2 + x3 + …………. +xn).**

**PROGRAM:**

#include<stdio.h>

#include<math.h>

void main()

{

int i,n;

int x,sum=0;

printf("Enter x and n : ");

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

if(x<0||n<0)

printf("Illegal values\n");

else

{

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

sum+=pow(x,i);

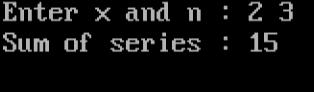
sum++;

printf("Sum of series : %d\n",sum);

}

}

**OUTPUT:**



**18) AIM: Write a program to read an array of integers (with max size 10) and print**

**the largest and the smallest of the given numbers.**

**PROGRAM:**

#include<stdio.h>

void main()

{

int a[10],i,n,largest,smallest;

printf("Enter how many values you want to read : ");

scanf("%d",&n);

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

{

printf("Enter the value of a[%d] : ",i);

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

}

largest=smallest=a[0];

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

{

if(a[i]>largest)

{

largest=a[i];

}

if(a[i]<smallest)

{

smallest=a[i];

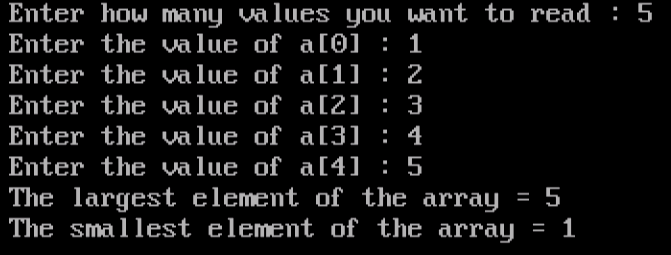
}

}

printf("The largest element of the array = %d\n",largest);

printf("The smallest element of the array = %d\n",smallest);

}

**OUTPUT:**

**19) AIM: Write a C program using pointers to compute sum, mean and standard**

**deviation of all elements sorted in an array of n real numbers.**

**PROGRAM:**

#include <stdio.h>

#include <math.h>

void main()

{

float x[20];

int i, n;

float average, variance, std\_deviation, sum = 0, sum1 = 0;

printf("Enter n value : ");

scanf("%d", &n);

printf("Enter %d elements : ", n);

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

{

scanf("%f", &x[i]);

}

/\* Compute the sum of all elements \*/

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

{

sum = sum + x[i];

}

average = (float)sum / n;

/\* Compute variance and standard deviation \*/

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

{

sum1 = sum1 + pow((x[i] - average), 2);

}

variance = sum1 / (float)n;

std\_deviation = sqrt(variance);

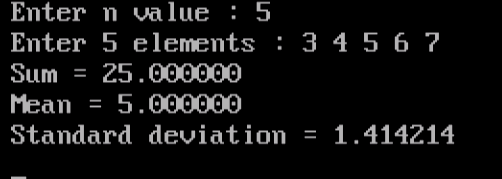
printf("Sum = %f\n", sum);

printf("Mean = %f\n", average);

printf("Standard deviation = %f\n", std\_deviation);

}

**OUTPUT:**



**20) AIM: Write a program to find the addition of two matrices using functions.**

**PROGRAM:**

void read(int a[10][10],int row,int col)

{

int i,j;

printf("Enter %d elements : ", row\*col);

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

{

for(j=1;j<=col;j++)

{

//printf("Enter Element %d %d : ",i,j);

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

}

}

}

void additionOfTwoMatrices(int m1[10][10],int m2[10][10],int row,int col)

{

int i,j;

int m3[10][10];

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

{

for(j=1;j<=col;j++)

{

m3[i][j] = (m1[i][j] + m2[i][j]);

}

}

printf("The Addition Matrix is\n");

display(m3,row,col);

}

void display(int m[10][10],int row,int col)

{

int i,j;

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

{

for(j=1;j<=col;j++)

{

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

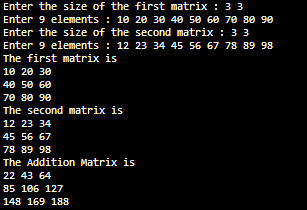
}

printf("\n");

}

}

**OUTPUT:**

****

**21) AIM: Write a program to find the multiplication of two matrices using functions.**

**PROGRAM:**

#include <stdio.h>

#include "Program717a.c"

void main() {

int a[10][10], b[10][10], m, n, p, q;

printf("Enter the size of the first matrix : ");

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

read(a, m, n);

printf("Enter the size of the second matrix : ");

scanf("%d%d", &p, &q);

read(b, p, q);

printf("The first matrix is\n");

display(a, m, n);

printf("The second matrix is\n");

display(b, p, q);

if (n == p) {

mulitiplicationOfTwoMatrices(a, b, m, n, q);

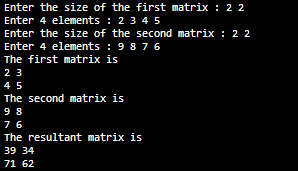
} else {

printf("Multiplication is not possible\n");

}

}

**OUTPUT:**

****

**22) AIM: Write a program to find the transpose of a given matrix using pointers.**

**PROGRAM:**

void read(int \*a[5],int m,int n) { // Write the formal arguments

int i, j;

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

for (j = 0; j < n; j++) {

scanf("%d",&a[i][j] ); // Complete the statement

}

}

}

void display(int \*a[5],int m,int n ) { // Write the formal arguments

int i, j;

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

for (j = 0; j < n; j++) {

printf("%d ",a[i][j] ); // Complete the statement

}

printf("\n");

}

}

void transpose(int \*a[5],int \*b[5],int m,int n)

{

int i,j;

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

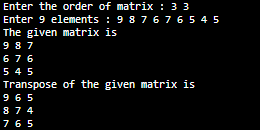
for(j=0; j<n; ++j)

{

b[j][i] = a[i][j];

}}

**OUTPUT:**

****

**23) AIM: Write a program to find the factorial of a given number.**

**PROGRAM:**

#include<stdio.h>

long factorial(int);

int main()

{

int n;

long f;

printf("Enter an integer : ");

scanf("%d", &n);

if (n < 0)

printf("Factorial of negative integers isn't defined.\n");

else

{

f = factorial(n);

printf("Factorial of given number %d = %ld\n", n, f);

}

return 0;

}

long factorial(int n)

{

if (n == 0)

return 1;

else

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

}

**OUTPUT:**

****

**24) AIM: Write a program to find the factorial of a given number using recursion process.**

**PROGRAM:**

long int factorial(long int n)

{

if(n==0)

return 1;

else

return n\*factorial(n-1);

}

**OUTPUT:**

****

**25) Write a C program to find GCD (**[**Greatest Common Divisor**](https://en.wikipedia.org/wiki/Greatest_common_divisor)**) of two integer numbers.**

**PROGRAM:**

#include<stdio.h>

void main() {

int n1,n2,i,gcd;

printf("Enter two integer numbers : ");

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

for(i=1;i<=n1&&i<=n2;i++)

{

if((n1%i==0)&&(n2%i==0))

gcd=i;

}

printf("GCD of %d and %d : %d\n",n1,n2,gcd);

}

**OUTPUT:**

****

**26) AIM: Write a program to find the gcd (Greatest Common Divisor) of a given two**

**numbers using recursion process.**

**PROGRAM:**

int gcd(int a,int b)

{

if(a==0)

return b;

else

return gcd(b%a,a);

}

**OUTPUT:**

****

**27) AIM: Write a sample code that calculates the value of an, given two positive non-zero**

**integers a and n.**

**PROGRAM:**

#include<stdio.h>

#include<math.h>

main()

{

int a,n,p;

printf("Enter two integer values for a and n : ");

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

p=pow(a,n);

printf("a\_power\_n = %d\n",p);

}

**OUTPUT:**

****

**28) AIM: Write a program to find the power of a given number using recursion process.**

**PROGRAM:**

int power(int m,int n)

{

if(n==0)

return 1;

else

return m\*power(m,n-1);

}

**OUTPUT:**

****

**29) Write a program to read and display the elements of an array using pointers.**

**PROGRAM:**

void read(int \*p,int n)

{

int i;

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

{

scanf("%d",(p+i));

}

}

void display(int \*p,int n)

{

int i;

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

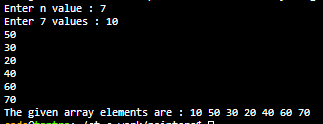
{

printf("%d ",\*(p+i));

}

}

**OUTPUT:**

****

**30) AIM: Write a program to find the sum of n elements by allocating memory by**

**using malloc() function.**

**PROGRAM:**

int \*allocateMemory(int n)

{

(int\*)calloc(n,sizeof(int));

}

void read(int \*p,int n)

{

int i;

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

{

scanf("%d",p+i);

}

}

int sum(int \*p,int n)

{

int i,s=0;

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

{

s=s+\*(p+i);

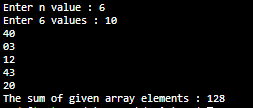
}

return s;

//printf("The sum of given array elements : %d ",s);

}

**OUTPUT:**

****

**31) AIM: Write a program to read and display the elements of an array in reverse order**

**using pointers.**

**PROGRAM:**

void read(int \*p,int n)

{

int i;

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

{

scanf("%d",p+i);

}

}

int displayInReverse(int \*p,int n)

{

int i,s=0;

for(i=n;i>=1;i--)

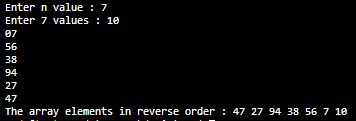
{

printf("%d ",\*(p+i));

}

}

**OUTPUT:**

****

**32) AIM: Write a C program to Open a File and to Print its contents on the screen**

**PROGRAM:**

#include <stdio.h>

void main() {

FILE \*fp;

char ch;

fp = fopen("SampleText1.txt", "w"); // Open a new file in write mode

printf("Enter the text with @ at end : ");

while ((ch=getchar())!='@') { // Repeat loop till read @ at the end

putc(ch,fp); // Put read character onto the file

}

putc('@',fp); // Put delimiter @ at the end on the file

fclose(fp); // Close the file

fp = fopen("SampleText1.txt","r"); // Open the existed file in read mode

printf("Given message is : ");

while ((ch=getc(fp))!='@') { // Repeat loop till get @ at the end of existed file

putchar(ch); // Put the character on the screen

}

printf("\n");

fclose(fp); // Close the file

}

**OUTPUT:**

****

**33) AIM: Write a program to copy contents of one file into another file. Follow the**

**instructions given below to write a program to copy the contents of one file to**

**another file**

**PROGRAM:**

#include <stdio.h>

void main() {

FILE \*fp, \*fp1, \*fp2;

char ch;

fp = fopen("SampleTextFile1.txt", "w"); // Open a new file in write mode

printf("Enter the text with @ at end : ");

while ((ch=getchar())!='@') { // Read loop till get @ at the end

putc(ch, fp);

}

putc(ch, fp);

fclose(fp);

fp1 = fopen("SampleTextFile1.txt", "r "); // Open an existed file in read mode

fp2 = fopen("SampleTextFile2.txt", "w"); // Open a new file in write mode

while ((ch=getc(fp1))!='@') { // Repeat loop till get @ at the end of existed file

putc(ch, fp2);

}

putc(ch, fp2);

fclose(fp1); // Close the existed file

fclose(fp2); // Close the copied file

fp2 = fopen("SampleTextFile2.txt", "r"); // Open the copied file in read mode

printf("Copied text is : ");

while ((ch=getc(fp2))!='@') { // Repeat loop till get @ at the end of copied file

putchar(ch);

}

printf("\n");

fclose(fp2); // Close the copied file

}

**OUTPUT:**

****

**34) AIM: Write a program to merge two files and stores their contents in another file.**

**PROGRAM:**

#include <stdio.h>

void main() {

FILE \*fp1, \*fp2, \*fp3;

char ch;

fp1 = fopen("SampleDataFile1.txt", "w"); // Open file in corresponding mode

printf("Enter the text with @ at end for file-1 :\n");

while ((ch=getchar())!='@') { // Write the condition

putc(ch, fp1);

}

putc(ch, fp1);

fclose(fp1);

fp2 = fopen("SampleDataFile2.txt", "w "); // Open file in corresponding mode

printf("Enter the text with @ at end for file-2 :\n");

while ((ch=getchar())!='@') { // Write the condition

putc(ch, fp2);

}

putc(ch, fp2);

fclose(fp2);

fp1 = fopen("SampleDataFile1.txt", "r"); // Open a first existed file in read mode

fp3 = fopen("SampleDataFile3.txt", "w"); // Open a new file in write mode

while ((ch=getc(fp1))!='@') { // Repeat loop till get @ at the end of existed file

putc(ch, fp3);

}

fclose(fp1); // Close the first existed file

fp2 = fopen("SampleDataFile2.txt", "r"); // Open a secong existed file in read mode

while ((ch=getc(fp1))!='@') { // Repeat loop till get @ at the end of existed file

putc(ch, fp3);

}

putc(ch, fp3);

fclose(fp2);

fclose(fp3);

fp3 = fopen("SampleDataFile3.txt", "r"); // Open the merged file in read mode

printf("Merged text is : ");

while ((ch=getc(fp3))!='@') { // Repeat loop till get @ at the end of merged file

putchar(ch);

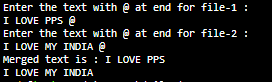
}

printf("\n");

fclose(fp3); // Close the merged file

}

**OUTPUT:**

****

**35) AIM: Write a program to insert a substring into a given main string from a given**

**position.**

**PROGRAM:**

#include <stdio.h>

#include <conio.h>

#include <string.h>

void main()

{

char a[25];

char b[25];

char c[25];

int p=0,r=0,i=0;

int t=0;

int x,g,s,n,o;

// clrscr();

printf("Enter the main string : ");

gets(a);

printf("Enter the sub string : ");

gets(b);

printf("Enter the position where you want insert substring : ");

scanf("%d",&p);

r = strlen(a);

n = strlen(b);

i=0;

// Copying the input string into another array

while(i <= r)

{

c[i]=a[i];

i++;

}

s = n+r;

o = p+n;

// Adding the sub-string

for(i=p;i<s;i++)

{

x = c[i];

if(t<n)

{

a[i] = b[t];

t=t+1;

}

a[o]=x;

o=o+1;

}

printf("After insertion the main string : %s\n", a);

//getch();

}

**OUTPUT:**

****

**36) AIM: Write a program to delete n characters from a given position in a string.**

**PROGRAM:**

#include <stdio.h>

#include <conio.h>

#include <string.h>

void delchar(char \*x,int a, int b);

void main()

{

char string[50];

int n,pos,p;

//clrscr();

printf("Enter a string : ");

gets(string);

printf("Enter the position : ");

scanf("%d",&pos);

printf("Enter the no of characters you want to delete : ");

scanf("%d",&n);

delchar(string, n,pos);

//getch();

}

void delchar(char \*x,int a, int b)

{

if ((a+b-1) <= strlen(x))

{

strcpy(&x[b],&x[a+b]);

printf("After deletion the string is : %s\n",x);

}

}

**OUTPUT:**

****

**37) AIM: Write a program to check whether the given string is a palindrome or not.**

**PROGRAM:**

#include <stdio.h>

#include <string.h>

void main(){

char string1[20];

int i, length;

int flag = 0;

printf("Enter a string : ");

scanf("%s", string1);

length = strlen(string1);

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

if(string1[i] != string1[length-i-1]){

flag = 1;

break;

}

}

if (flag) {

printf("The given string %s is not a palindrome\n",string1);

}

else {

printf("The given string %s is a palindrome\n", string1);

}

}

**OUTPUT:**

****

****

**38) AIM: Write a program to display the position or index in the string S where the string**

**T begins, or -1 if S doesn't contain T.**

**PROGRAM:**

#include<stdio.h>

#include<string.h>

#include<conio.h>

void main()

{

char s1[40], s2[20];

char \*found;

printf("Enter the sentence : ");

gets(s1);

printf("Enter the string to be searched : ");

gets(s2);

found=strstr(s1,s2);

if(found)

printf("String found in the sentence at the position : %ld\n", found-s1);

else

printf("-1\n");

}

**OUTPUT:**

****

\

**39) AIM: Write a C program to count the lines, words and characters in a given text.**

**PROGRAM:**

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include<stdlib.h>

void main()

{

char str[500];

int i,len,l = 0, w = 0,c=0;

printf("Enter lines of text (enter END to complete)\n");

gets(str);

len=strlen(str);

while(strcmp(str,"END")!=0)

{

w++;

c=c+len;

for(i=0;str[i]!='\0'; i++)

{

if(str[i]==' '||str[i]=='\t')

{ w++;

}}

l++;

gets(str);

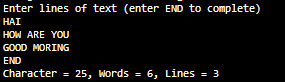
len=strlen(str);

}

printf("Character = %d, Words = %d, Lines = %d\n", c,w,l);

}

**OUTPUT:**

****

**40) AIM: 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.**

**PROGRAM:**

#include <stdio.h>

void smallest(int[], int);

void largest(int[], int);

void sum(int[], int);

void average(int[], int);

void main() {

int n,arr[10],i,x;

printf("Enter number : ");

scanf("%d",&n);

printf("Enter %d numbers : ",n);

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

{

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

}

printf("The menu driven is:\n1.smallest\n2.largest\n3.sum\n4.average\n");

printf("Enter an option : ");

scanf("%d",&x);

switch(x)

{

case 1:smallest(arr,n);

break;

case 2:largest(arr,n);

break;

case 3:sum(arr,n);

break;

case 4:average(arr,n);

break;

default:printf("Invalid choice\n");

}

}

void smallest(int arr[], int n) {

int small,i;

small=arr[0];

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

{

if(arr[i]<small)

small=arr[i];

}

printf("The minimum element is %d",small);

}

void largest(int arr[], int n) {

int large,i;

large=arr[0];

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

{

if(arr[i]>large)

large=arr[i];

}

printf("The largest element is %d",large);

}

void sum(int arr[], int n) {

int sum=0,i;

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

{

sum=sum+arr[i];

}

printf("The sum of array elements is %d",sum);

}

void average(int arr[], int n) {

float avg;

int sum=0,i;

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

{

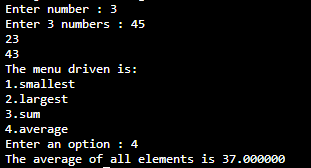
sum=sum+arr[i];

}

avg=(float)sum/n;

printf("The average of all elements is %f\n",avg);

}

****

**41) AIM: Write a program to print a right-angled triangle with \* (stars) separated by**

**spaces for a given number of rows.**

**PROGRAM:**

#include <stdio.h>

void main() {

int rows, rowIndex, starsCount;

printf("Enter number of rows : ");

scanf("%d", &rows);

for (rowIndex=1;rowIndex<=rows;rowIndex++ ) {// Fill in the missing code

for (starsCount=1;starsCount<=rowIndex;starsCount++ ) { // Fill in the missing code

printf("\* ");

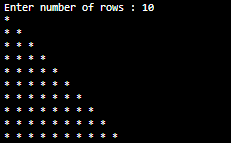
}

printf("\n");

}

}

**OUTPUT:**

****

**42) AIM: Write a program to generates a right-angled triangle with numbers starting from**

**1 till that row's number as shown below.**

**PROGRAM:**

#include <stdio.h>

void main() {

int rows, rowIndex, num;

printf("Enter number of rows : ");

scanf("%d", &rows);

for (rowIndex=1;rowIndex<=rows;rowIndex++ ) { //Fill in the missing code

for (num=1;num<=rowIndex;num++) { //Fill in the missing code

printf("%d ", num);

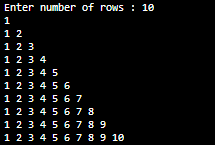
}

printf("\n");

}

}

**OUTPUT:**

****

**43) AIM: Write a program to print a right-angled triangle with numbers separated by a**

**space starting with 1 and ending with that particular row's number.**

**PROGRAM:**

#include <stdio.h>

void main() {

int rows, rowIndex, i;

printf("Enter number of rows : ");

scanf("%d" ,&rows);

for (rowIndex=1;rowIndex<=rows;rowIndex++ ) { //Fill in the missing code

for (i=1;i<=rowIndex;i++ ) { //Fill in the missing code

printf("%d ", rowIndex);

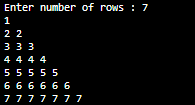
}

printf("\n");

}

}

**OUTPUT:**

****

**44) AIM: Write a program to print a pattern of \* (stars) separated by spaces for the given**

**number of rows.**

**PROGRAM:**

#include<stdio.h>

main()

{

int r,n,i,j,k;

printf("Enter number of rows : ");

scanf("%d",&r);

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

{

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

{

printf(" ");}

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

printf("\*");}

printf("\n");

}

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

{

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

{

printf(" ");}

for(k=0;k<=r-j;k++)

{

printf("\*");

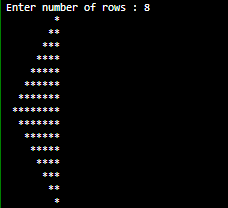
}

printf("\n");

}

}

**OUTPUT:**



**45) AIM: Write a C program to print floyd’s triangle**

**PROGRAM:**

#include <stdio.h>

int main()

{

int n, i, c, a = 1;

printf("Enter n value : ");

scanf("%d", &n);

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

{

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

{

printf("%d ",a);

a++;

}

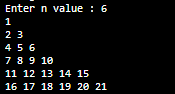
printf("\n");

}

return 0;

}

**OUTPUT:**

****

**46) AIM: Write a program to sort (Ascending order) the given elements using bubble sort**

**Technique**

**PROGRAM:**

#include<stdio.h>

void main() {

int a[20], i, n, j, temp;

printf("Enter value of n : ");

scanf("%d", &n);

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

printf("Enter element for a[%d] : ", i);

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

}

printf("Before sorting the elements in the array are\n");

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

printf("Value of a[%d] = %d\n", i, a[i]);

}

for (i = 0; i < n - 1; i++) {

for (j = 0; j < n - i - 1; j++) {

if (a[j] > a[j+1]) {

temp = a[j];

a[j] = a[j+1];

a[j+1] = temp;

}

}

}

printf("After sorting the elements in the array are\n");

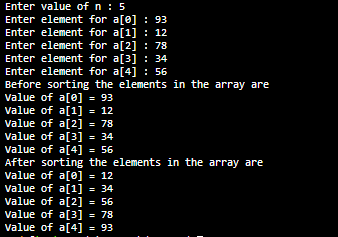
for (i = 0; i < n; i++) {

printf("Value of a[%d] = %d\n", i, a[i]);

}

}

**OUTPUT:**

****

**47) AIM: Write a program to sort (Ascending order) the given elements using insertion**

**sort technique**.

**PROGRAM:**

#include<stdio.h>

void main() {

int a[20], i, n, j, temp;

printf("Enter value of n : ");

scanf("%d", &n);

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

printf("Enter element for a[%d] : ", i);

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

}

printf("Before sorting the elements in the array are\n");

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

printf("Value of a[%d] = %d\n", i, a[i]);

}

for (i = 0; i < n - 1; i++) {

for (j = 0; j < n - i - 1; j++) {

if (a[j] > a[j+1]) {

temp = a[j];

a[j] = a[j+1];

a[j+1] = temp;

}

}

}

printf("After sorting the elements in the array are\n");

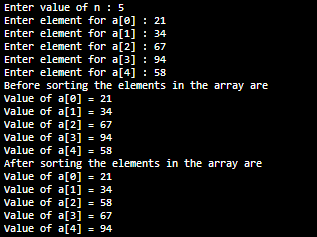
for (i = 0; i < n; i++) {

printf("Value of a[%d] = %d\n", i, a[i]);

}

}

**OUTPUT:**

****

**48) AIM: Write a program to sort (Ascending order) the given array elements**

**using selection sort largest element method.**

**PROGRAM:**

#include<stdio.h>

void main() {

int a[20], i, n, j, temp;

printf("Enter value of n : ");

scanf("%d", &n);

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

printf("Enter element for a[%d] : ", i);

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

}

printf("Before sorting the elements in the array are\n");

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

printf("Value of a[%d] = %d\n", i, a[i]);

}

for (i = 0; i < n - 1; i++) {

for (j = 0; j < n - i - 1; j++) {

if (a[j] > a[j+1]) {

temp = a[j];

a[j] = a[j+1];

a[j+1] = temp;

}

}

}

printf("After sorting the elements in the array are\n");

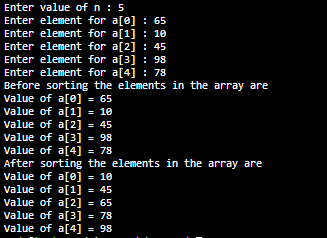
for (i = 0; i < n; i++) {

printf("Value of a[%d] = %d\n", i, a[i]);

}

}

**OUTPUT:**

****

**49) AIM: Write a program to sort (Ascending order) the given array elements**

**using selection sort smallest element method.**

**PROGRAM:**

#include<stdio.h>

void main() {

int a[20], i, n, j, temp;

printf("Enter value of n : ");

scanf("%d", &n);

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

printf("Enter element for a[%d] : ", i);

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

}

printf("Before sorting the elements in the array are\n");

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

printf("Value of a[%d] = %d\n", i, a[i]);

}

for (i = 0; i < n - 1; i++) {

for (j = 0; j < n - i - 1; j++) {

if (a[j] > a[j+1]) {

temp = a[j];

a[j] = a[j+1];

a[j+1] = temp;

}

}

}

printf("After sorting the elements in the array are\n");

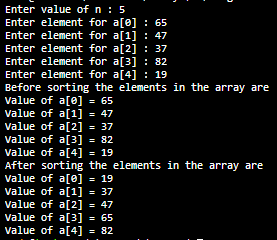
for (i = 0; i < n; i++) {

printf("Value of a[%d] = %d\n", i, a[i]);

}

}

**OUTPUT:**



**50) AIM: Write a program to search the given element from a list of elements with linear**

**search technique using functions.**

**PROGRAM:**

/\*#include<stdio.h>

void main() {

int a[20], i, n, key, flag = 0, pos;

printf("Enter value of n : ");

scanf("%d", &n);

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

printf("Enter element for a[%d] : ", i);

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

}

printf("Enter key element : ");

scanf("%d", &key);

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

if (key == a[i]) {

flag = 1;

pos = i;

break;

}

}

if (flag == 1) {

printf("The key element %d is found at the position %d\n", key, pos);

} else {

printf("The Key element %d is not found in the array\n", key);

}

}

\*/

void read(int\* a, int n) {

int i;

printf("Enter %d elements : ", n);

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

// printf("Enter element for a[%d] : ", i);

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

}

}

int linearSearch(int\* a, int n, int key) {

int i,flag,pos;

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

if (key == a[i]) {

flag = 1;

pos = i;

break;

} else {

pos =-1;

}

}

// if(pos<0)

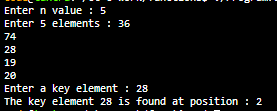
// return -1;

// else

return pos;

}

**OUTPUT:**



**51) AIM: Write a program to search the given element from a list of elements with binary**

**search technique using functions.**

**PROGRAM:**

/\*#include<stdio.h>

void main() {

int a[20], i, n, j, temp;

printf("Enter value of n : ");

scanf("%d", &n);

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

printf("Enter element for a[%d] : ", i);

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

}

printf("Before sorting the elements in the array are\n");

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

printf("Value of a[%d] = %d\n", i, a[i]);

}

for (i = 0; i < n - 1; i++) {

for (j = 0; j < n - i - 1; j++) {

if (a[j] > a[j+1]) {

temp = a[j];

a[j] = a[j+1];

a[j+1] = temp;

}

}

}

printf("After sorting the elements in the array are\n");

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

printf("Value of a[%d] = %d\n", i, a[i]);

}

}

\*/

void read(int\* a, int n) {

int i;

// printf("Enter %d elements : ", n);

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

printf("Enter element for a[%d] : ", i);

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

}

}

void display(int\* a, int n) {

int i;

// printf("After sorting the elements in the array are\n");

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

printf("Value of a[%d] = %d\n", i, a[i]);

}

}

void bubbleSort(int\* a, int n) {

int i, temp,j;

for (i = 0; i < n - 1; i++) {

for (j = 0; j < n - i - 1; j++) {

if (a[j] > a[j+1]) {

temp = a[j];

a[j] = a[j+1];

a[j+1] = temp;

.}

}

}

}

int binarySearch(int\* a, int n, int key) {

int i, pos,flag;

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

if (key == a[i]) {

flag = 1;

pos = i;

break;

} else {

pos =-1;

}

}

// if(pos<0)

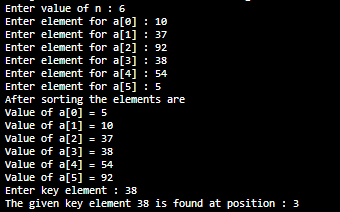
// return -1;

// else

return pos;

}

**OUTPUT:**



**52) AIM: Write a program to sort the names in alphabetical order.**

**PROGRAM:**

#include <stdio.h>

#include <string.h>

void main()

{

char name[10][15], tname[10][15], temp[15];

int i, j, n;

printf("Enter n value : ");

scanf("%d", &n);

printf("Enter %d strings : ", 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("Before sorting the names are : ");

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

{

printf("%s ",tname[i]);

}

printf("\nAfter sorting the names are : ");

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

{

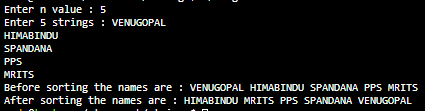
printf("%s ",name[i]);

}

printf("\n");

}

**OUTPUT:**

****

**53) AIM: Write a C program to convert a Roman numeral ranging from I to L to its**

**decimal equivalent**.

**PROGRAM:**

#include<stdio.h>

#include<string.h>

int digit(char);

int main(){

char roman\_Number[1000];

int i=0;

long int number =0;

printf("Enter any roman number (Valid digits are I, V, X, L): \n");

scanf("%s",roman\_Number);

while(roman\_Number[i]){

if(digit(roman\_Number[i]) < 0){

printf("Invalid roman digit : %c",roman\_Number[i]);

return 0;

}

if((strlen(roman\_Number) -i) > 2){

if(digit(roman\_Number[i]) < digit(roman\_Number[i+2])){

printf("Invalid roman number");

return 0;

}

}

if(digit(roman\_Number[i]) >= digit(roman\_Number[i+1]))

number = number + digit(roman\_Number[i]);

else{

number = number + (digit(roman\_Number[i+1]) - digit(roman\_Number[i]));

i++;

}

i++;

}

printf("Its decimal value is : %ld",number);

return 0;

}

int digit(char c){

int value=0;

switch(c){

case 'I': value = 1; break;

case 'V': value = 5; break;

case 'X': value = 10; break;

case 'L': value = 50; break;

// case 'C': value = 100; break;

//case 'D': value = 500; break;

//case 'M': value = 1000; break;

// case '\0': value = 0; break;

default: value = -1;

}

return value;

}

**OUTPUT:**

****

**54) AIM: Write a C programthat converts a number ranging from 1 to 50 to Roman**

**equivalent**.

**PROGRAM:**

#include <stdio.h>

int main(void)

{

int num, rem;

printf("Enter the number: ");

scanf("%d", &num);

printf("Roman number is : ");

while(num != 0)

{

if (num >= 50) // 50 - l

{

printf("l");

num -= 50;

}

else if (num >= 40) // 40 - xl

{

printf("xl");

num -= 40;

}

else if (num >= 10) // 10 - x

{

printf("x");

num -= 10;

}

else if (num >= 9) // 9 - ix

{

printf("ix");

num -= 9;

}

else if (num >= 5) // 5 - v

{

printf("v");

num -= 5;

}

else if (num >= 4) // 4 - iv

{

printf("iv");

num -= 4;

}

else if (num >= 1) // 1 - i

{

printf("i");

num -= 1;

}

}

return 0;

}

**OUTPUT:**

****