## 

FACULTY OF ENGINEERING &TECHNOLOGY BACHELOR OF TECHNOLOGY

Computational Thinking for Structure Design-1

(303105104)1stSEMESTER

COMPUTERSCIENCE&ENGINEERING

DEPARTMENT

LABORATORYMANNUAL



**COMPUTATIONALTHINKINGFORSTRUCTUREDESIGN-1**

**PRACTICAL BOOK COMPUTER SCIENCEANDENGINEERINGDEPARTMENT**

# PREFACE

It gives us immense pleasure to present the first edition of ***Computational Thinking for Structure Design -1*** for the B.Tech. 1st year students for PARULUNIVERSITY.

The Fundamental of Programming theory and laboratory courses at **PARULUNIVERSITY,**

**WAGHODIA,VADODARA** are designed in such a way that students develop

The basic understanding of the subject in the theory classes and then try their hands on the computer learnt during the theoretical sessions.

This book is emphatically not focused on “the syntax of C”. Understanding the fundamental ideals, principals, and techniques is the essence of a good programmer. Only well-designed code has a chance of becoming part of a correct, reliable, and maintain able system .Also ,“thefundamentals”are what last: they will still be essential after today’s language and tools have evolved or been replaced.

We acknowledge the authors and publishers of all the books which we have consulted while developing this Practical book. Hopefully this Computational Thinking for Structure Design-1 will serve the purpose for which it has been developed.



**Instructions to students**

1. Every student should obtain a copy of laboratory Manual.
2. Dress Code:Students must come to the laboratory wearing. i.Trousers, ii. half-sleeve tops and

iii. Leather shoes. Half pants,loosely hanging garments and slippers are not allowed.

1. To avoid injury,the student must take the permission of the laboratory staff be fore handling any machine.
2. Students must ensure that their work areas are clean and dry to avoid slipping.
3. Do not eat or drink in the laboratory.
4. Do not remove anything from the computer laboratory without permission.
5. Do not touch,connect or disconnect any plug or cable without your lecturer/laboratory technician’s permission.
6. All students need to perform the practical/program.



**CERTIFICATE**

This is to certify that

Mr./Ms Singh Abhinav Santosh

With enrollment not 23UG031209/2303031260216 has success fully

Completed his/her laboratory experiments in the ***Computational Thinking for Structure Design -1 (303105104)*** from the department of Computer science & engineering

During the academic year 2023-2024.



Date of Submission: ......................... Staff In charge:...........................

Head of Department:...........................................



**INDEX**

**Class: 1st Semester Subject:-ComputationalThinkingforStructuredDesign-1**

**Y.2023-2024**  **SubjectCode:303105104**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. No**  **.** | **ExperimentTitle** | **PageNo.** | | **Date of Performance** | **Marks out0f**  **10** | **Sign** |
|  | **PracticalSet–1** | **From** | **To** |  |  |  |
| 1 | Installation CIDE, Basic Structure of C program. Format Specifiers, Escape Character.Run time input/Output Programs. | 10 | 13 |  |  |  |
| 2 | 1. Write a c program to calculate Area of Rectangle,Perimeter of a Rectangle and   Diagonal of a Rectangle.   1. Write a c program to calculate Area of square, Perimeter of a square and Diagonal of a square. 2. Write a c program to calculate total area of Cylinder and volume of a cylinder. | 14 | 19 |  |  |  |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 1. The total distance traveled by vehicle in t seconds is given by distance s = ut+1/2at? where uand aare the initial velocity (m/sec.) and acceleration(m/sec?). Write a C program to find the distance traveled at regular intervals of time given the values of and a The program should provide the flexibility to the user to select his own time intervals and repeat the calculations for different values of u and   a   1. Write a C program, which takes two integer operator and send one operator from the user ,performs the operation and then prints the result. (Consider the operators   And use Switch Statement) | 20 | 21 |  |  |  |
| 4 | 1. Write a C program to find the sum of individual digits of a positive integer. 2. A Fibonacci sequence is defined as   follows: the first and second terms in the sequences are 0 and 1.Subsequent terms are | 22 | 23 |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | found by adding the preceding two terms in the sequence.Write a C program to generate the first n terms of the sequence.  3.WriteaCprogramtogeneratealltheprime  numbersbetween1andn,where n is a value supplied by the user. |  |  |  |  |  |
| 5 | 1. Write a C program to calculate the following Sum: Sum=1-x^2/2!   +x^4/4!x^6/6!+x^8/8!x^10/10!.   1. Write a C program to find the roots of a quadratic equation. | 24 | 29 |  |  |  |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 6 | Write C programs that use both recursive and non-recursive functions.   1. To find the factorial of a given integer. 2. To find the GCD (greatest common divisor) of two given integers. | 30 | 33 |  |  |  |
| 7 | 1. Write a C program to find thelargest integer in a list of integers, 2. Write a C program that uses functions to perform the following:1.Addition of Two Matrices   2.Multiplication of Two Matrices | 34 | 37 |  |  |  |
| 8 | 1. Write a C program that uses functions to perform the following operation; 2. To insert a sub-string into a given main string from a given position. 3. To delete n Characters from a given position in a given string,   2.Write a C program to determine If the  Given string is a palindrome or not. | 38 | 43 |  |  |  |
| 9 | 1. Write a C program that displays the position or index in the string S where the string T begins, or-1if S does not contain T. 2. Write a C program to count the lines, words and characters in a given text. | 44 | 49 |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | 1. Write a C program to generate Pascal's triangle. 2. Write a C program to construct a pyramid   of numbers. | 50 | 53 |  |  |  |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11 | Write a C program to read int wo 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 computes1+5+25+125.Printx,n,the sum. | 54 | 57 |  |  |  |
|  | Perform error checking. For example, the formula does not make sense for negative exponents ◻◻if n is less than 0. Have your program print an error-message ifn<0,then go back and read in the next pair of numbers without computing the sum.  Are any values of x also illegal? If so, test for them too. | 58 | 59 |  |  |  |
| 12 | 1. 2'scomplementofanumberisobtained by   scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2◻s complement of 11100is00100.WriteaCprogramtofindthe2's complement of a binary number.   1. Write a C program to convert a Roman numeral to its decimal Equivalent. | 60 | 65 |  |  |  |
| 13 | 1. Write a c program on Given an unsorted array arr[] of size N. Rotate the array to the left (counter-clock wise direction)by D steps, where D is a positive integer. 2. Write a c Program on given two sorted arrays arr1 and arr2 of size N and M respectively and an element K .The task is to find the element that would be at the k ◻ th position of the final sorted array . Explanation:   Input :  Array1-14 23 5  Array2-78 6k =  5 Output : 5  Because The final sorted array would be-  1,2, 3,  4, 5, 6,7, 8,The 5th element of this array is6. | 66 | 72 |  |  |  |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 14 | 1. Write a c program to take multiline string input and print individual string length .2.  Write a c program to reverse the individual word of a given string  Explanation :input: Welcome To Bytexl output:  Emocle Wo TlxetyB. | 73 | 77 |  |  |  |



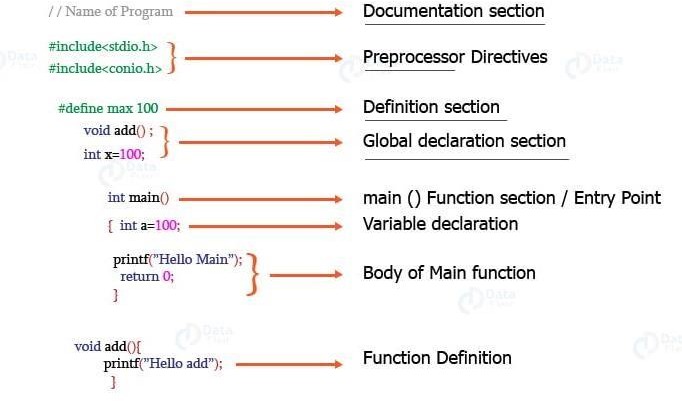
# Practical 1

**AIM: Installation CIDE ,Basic Structure of C program.FormatSpecifiers,Escape Character. Run time input/Output Programs.**

**Basic structure of C Program:**

The components of the basic structure of a C programconsistsof7 parts

1. Document section
2. Preprocessor/linkSection
3. Definition section
4. Global declaration section
5. Function declaration section
6. Main function
7. User-defined function section





Let’s understand every basic component of the C program with the help of an example.

/\*

Documentationsection

Cprogrammingstructure Author: DataFlair

\*/

#include<stdio.h>/\*Linksection\*/

intsubtract=0;/\*Globaldeclaration,definitionsection\*/ intall(int,int);/\*Functiondeclarationsection\*/intmain () /\* Main function \*/

{

printf("WelcometoDataFlairtutorials!\n\n"); printf("ThisisaCprogram \n");subtract=all (25,10);

printf("Subtractionofthetwonumbers:%d\n",subtract);**return**

0;

}

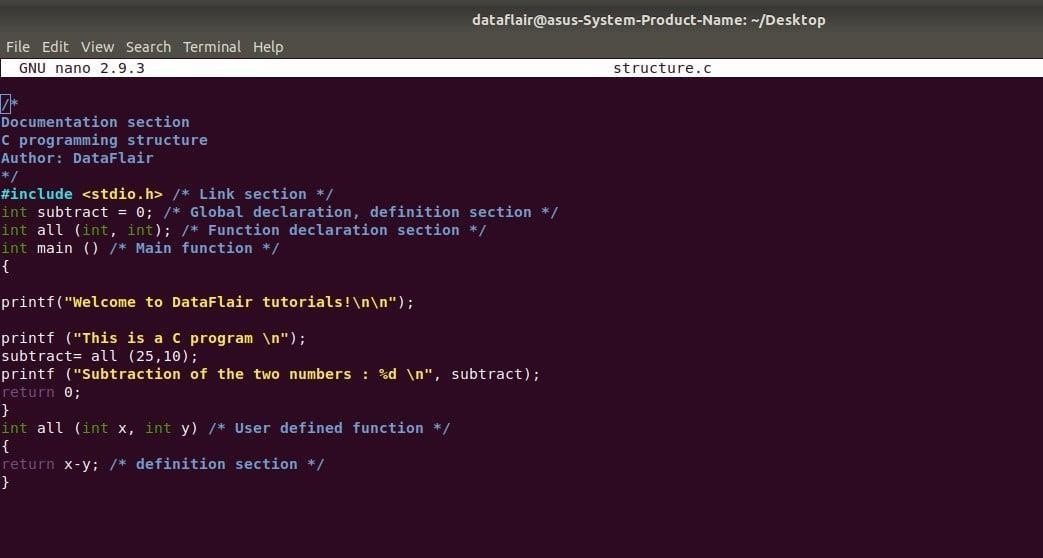
intall(intx,inty)/\*Userdefinedfunction\*/

{

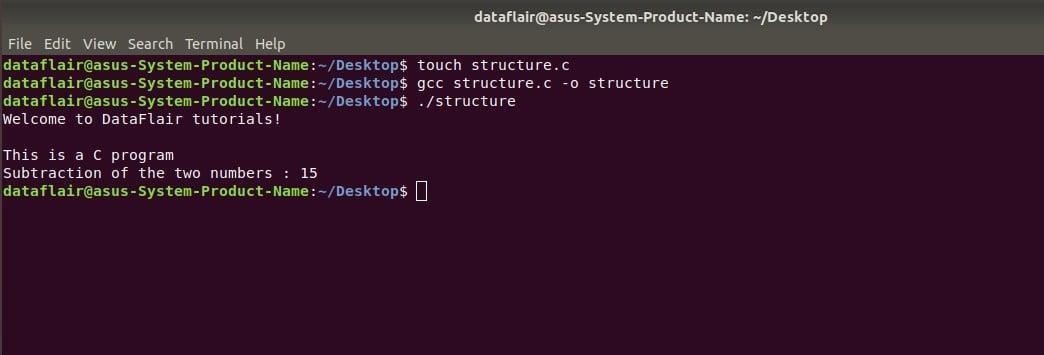
**return**x-y;/\*definitionsection\*/

}



**CodeonScreen-**

**Output:**





1. **Documentation Section:**

Itisthesectioninwhichyoucangivecommentstomaketheprogrammoreinteractive.The compiler won’t compile this and hence this portion would not be displyed on the output screen.

1. **Pre process or directives Section:**

This section involves the use of header files that are to included necessarily program.

1. **Definition section:**

This section involves the [variable definition and declaration in C.](https://data-flair.training/blogs/variables-in-c/)

1. **Global declaration Section:**

This section is used to define the global variables to be used in the programs, that means you can use these variables throughout the program.

1. **Function prototype declaration section:**

This section gives the information about a function that includes, the data type or the return type, the parameters passed or the arguments.

1. **Main function:**

It is the major section from where the execution of the program begins. The main section involves the declaration and executable section.

1. **User-defined function section:**

When you want to define your function that fulfills a particular requirement, you can define them in this section.



# Practical 2

**AIM1:Write a c program to calculate Area of Rectangle,Perimeter of a Rectangle and Diagonal of a Rectangle.**

**Program:**

#include <stdio.h>

#include <math.h>

Int main () {

int area, perimeter, width, length;

float diagonal;

printf ("enter the length\n");

scanf ("%d", &length);

printf ("enter width\n");

scanf ("%d", &width);

area = length \* width;

perimeter = 2 \* (length + width);

diagonal = sqrt ((length \* length) + (width \* width));

printf ("area of rectangle is = %d\n", area);

printf ("perimeter of rectangle is = %d\n", perimeter);

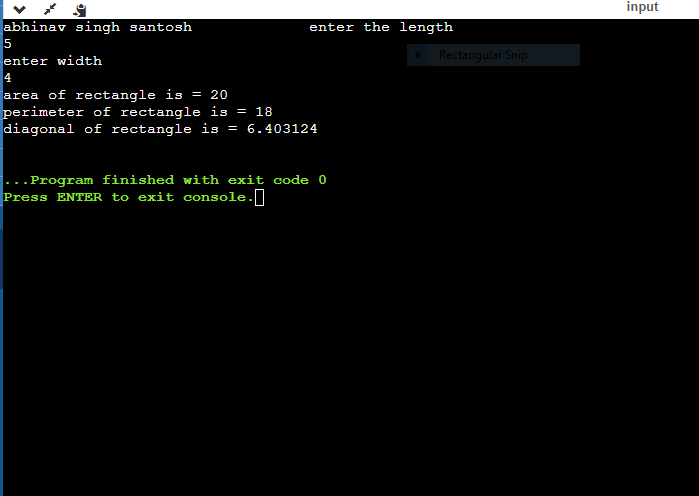
printf ("diagonal of rectangle is = %f\n", diagonal);

return 0;

}



**OUTPUT:**





**AIM2:Write a c program to calculate Area of square ,Perimeter of a square and Diagonal of a square.**

**Program:**

#include <stdio.h>

#include <math.h>

Int main ()

{

int area, perimeter, length;

float diagonal;

printf ("enter the length\n");

scanf ("%d", &length);

area = length \* length;

perimeter = 4 \* (length);

diagonal = sqrt ((length \* length) + (length \* length));

printf ("area of square is = %d\n", area);

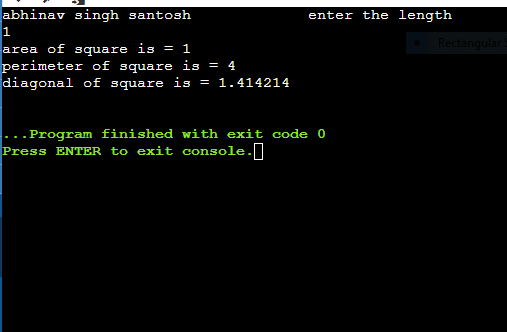
printf ("perimeter of square is = %d\n", perimeter);

printf ("diagonal of square is = %f\n", diagonal);

return 0;

}

**OUTPUT:**

****



**AIM3:Write a c program to calculate total area of Cylinder and volume of a cylinder.**

**Program:**

#Include<stdio.h>

#include <math.h>

Int main ()

{

floatsurfacearea, pi, length, radius, volume, totalsurfacearea;

printf ("enter the radius\n");

scanf ("%f", &radius);

printf ("enter the length\n");

scanf ("%f", &length);

pi = 3.14;

surfacearea = 2 \* pi \* radius \* length;

volume = pi \* (radius \* radius) \* length;

totalsurfacearea = 2 \* pi \* radius \* length + 2 \* pi \* (radius \* radius);

printf ("surface area of cylinder %f\n", surfacearea);

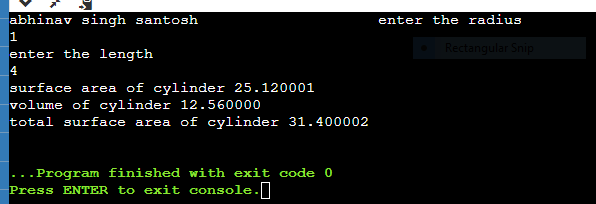
printf ("volume of cylinder %f\n", volume);

printf ("total surface area of cylinder %f\n", totalsurfacearea);

return 0;



**OUTPUT:**





# Practical 3

**AIM1: The total distance traveled by vehicle in ͚t seconds is given by distance s = ut+1/2at2? where͚ u͛ and͚ a͛ are the initial velocity (m/sec.)and acceleration(m/sec?). Write a C program to find the distance travel edit regular intervals of time given the Values of͚ u͛ and͚ a͛͛˙The program should provide the flexibility to the user to select his Own time intervals and repeat the calculations for different values of͚ u͛ and͚ a͛.˙͛**

**Program:**

#include<stdio.h>

int main(){

int t;

floatv,a,distance;

printf("enter initial velocity v\n");

scanf("%f",&v);

printf("enter acceleration a\n");

scanf("%f",&a);

printf("enter time t\n");

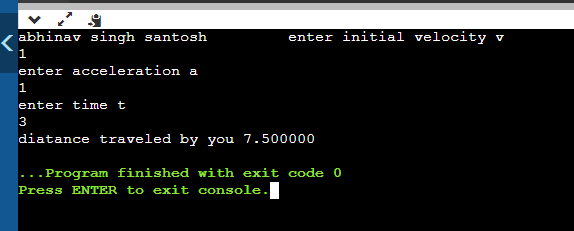
scanf("%d",&t);

distance=v\*t+a\*(t\*t)/2;

printf("diatancetraveled by you %f",distance);

return 0;

**OUTPUT:**





AIM2:**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)**

**Program:**

#include <stdio.h>

int main()

{

printf("abhinavsinghsantosh ");

inta,b,c;

printf("enter 2 numbers ");

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

printf("give your choice\n 1 for add\n 2 for subraction\n 3 for multiplication\n 4 for divide\n ");

scanf("%d",&c);

switch(c)

{

case 1:printf("addition is %d",a+b);

break;

case 2:printf("subraction is %d",a-b);

break;

case 3:printf("multiplication is %d",a\*b);

break;

case 4:printf("divide is %d",a/b);

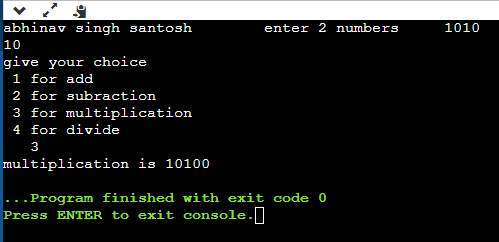
break;

default:printf("wrong choice");

} }



**OUTPUT:**





# Practical 4

**AIM1: Write a C program to find the sum of individual digits of a positive integer.**

**Program:**

#include <stdio.h>

void main(){

int n,sum=0;

printf("enter a +ve integer ");

scanf("%d",&n);

while(n>0)

{

sum +=n%10;

n=n/10;

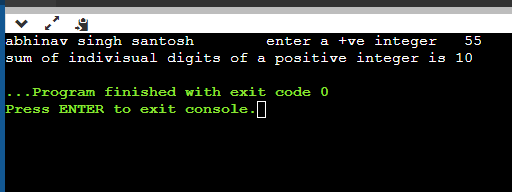
}

printf("sum of indivisual digits of a positive integer is %d",sum);

}



**OUTPUT**





**AIM2:A Fibonacci sequence is defined as follows: the first and second terms in the sequences are 0 and 1.Subsequent term are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.**

**Program:**

#include<stdio.h>

int main(){

intn,a=0,b=1,temp;

printf("Enter the number uptill where you want to generate series: ");

scanf("%d",&n);

printf("Fibonacci Series is: ");

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

printf("%d",a);

temp=a+b;

a=b;

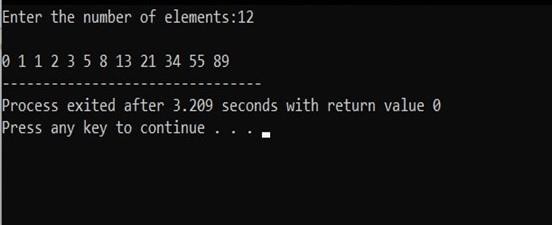
b=temp;

}

return 0;

}

**OUTPUT:**





**AIM3:Write a C program to generate all the prime numbers between1andn,wheren is a value supplied by the user.**

**Program:**

#include<stdio.h>

//#include<conio.h>int main() {

int n, i, j, count; printf("Primeno.series\n");

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++;

}

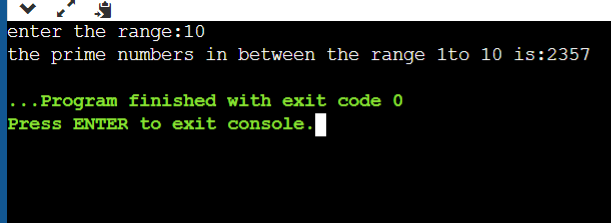
if(count==2){ printf("%d\t", i);

} }

return0;

}

**OUTPUT:**

****



# Practical 5

**AIM1:Write a C program to calculate the following Sum:Sum=1-x^2/2!**

**+x^4/4!x^6/6!+x^8/8!-x^10/10!.**

**Program:**

#include <stdio.h>

#include <math.h>

int main()

{

int counter,f\_coun;

float sum=0,x,power,fact;

printf("equation series : 1- X^2/2! + X^4/4! - X^6/6! + X^8/8! - X^10/10!");

printf("\nenter value of X : ");

scanf("%f",&x);

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

{

fact=1;

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

fact \*= f\_coun;

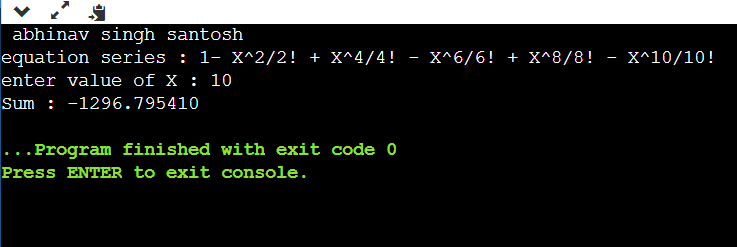
sum=sum+(pow(-1,counter)\*(pow(x,power)/fact));

}

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

}

**OUTPUT:**



**AIM2:Write a C program to find the roots of a quadratic equation. Program:**

#include <math.h>

int

main ()

{

float a, b, c, r1, r2,d;

printf ("enter the value of a b c;");

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

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

if (d > 0)

{

r1 = -b + sqrt (d) / (2 \* a);

r2 = -b - sqrt (d) / (2 \* a);

printf("the real roots =%f%f",r1,r2);

}

else if (d==0){

r1=-b/(2\*a);

r2=-b/(2\*a);

printf("roots are equal =%f %f",r1,r2);

}

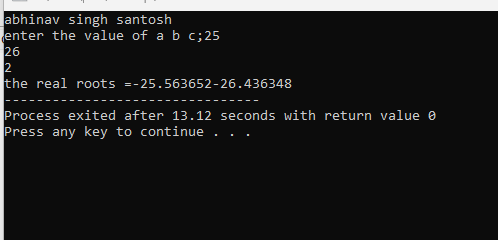
else

printf("roots are imaginary");

return 0;

}

**OUTPUT:**

****



# Practical 6

**AIM:Write C programs that use both recursive and non-recursive functions. AIM1: To find the factorial of a given integer.**

**Program:**

#include<stdio.h>

int factorial(int n);

int main(){

int fact,number;

printf("ebter a number for which you want to calculate factorial:");

scanf("%d",&number);

fact=factorial(number);

printf("factorial=%d",fact);

return 0;

}

int factorial(int n){

if(n>=1){

return n\*factorial(n-1);

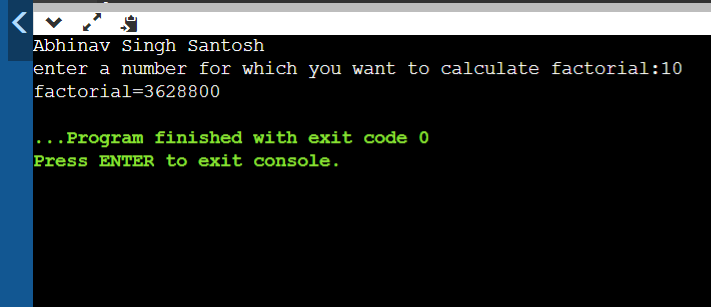
}

else

return 1;

}

**OUTPUT:**

****



**AIM2:To find the GCD (greatest common divisor) of two given integers.**

**Program:**

#include <stdio.h>

int hcf(int n1,int n2);

int main(){

int n1,n2;

printf("enter two positive integer:");

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

printf("G.C.D of %d and %d is %d.",n1,n2,hcf(n1,n2));

return 0;

}

int hcf (int n1,int n2){

if (n2=0)

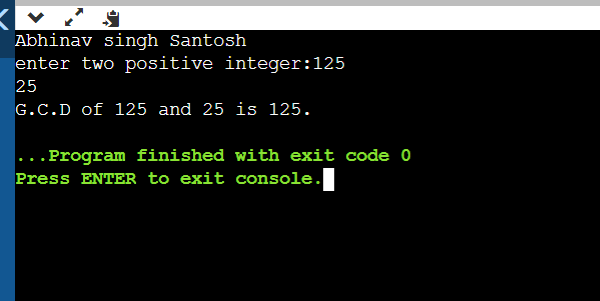
return hcf(n2,n1 % n2 );

else

return n1;

}

**OUTPUT:**

****



# Practical 7

**AIM1:Write a C program to find the largest integer in al is to find integers, Program:**

#include <stdio.h>

int main()

{

int i,arr[] = {25, 11, 7, 75, 56};

int length = sizeof(arr)/sizeof(arr[0]);

int max = arr[0];

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

if(arr[i] > max)

max = arr[i];

}

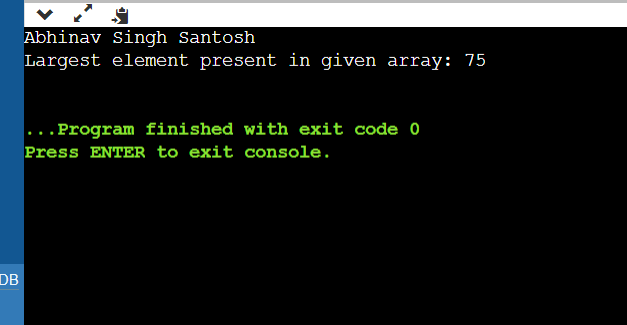
printf("Largest element present in given array: %d\n", max);

return 0;

}



**OUTPUT:**





**AIM2:Write a C program that uses function stopper form the following:**

* 1. **Addition of Two Matrices Program**

#include <stdio.h>

void addMatrix(int a[10][10], int b[10][10],

int c[10][10], int row, int column)

{ int i,j;

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

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

c[i][j] = a[i][j] + b[i][j]; } } }

void readMatrix(int matrix[10][10], int row, int column)

{ int i,j;

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

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

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

void displayMatrix(int matrix[10][10], int row, int column)

{ int i,j;

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

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

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

printf("\n"); } }

int main() {

int a[10][10];

int b[10][10];

int c[10][10];

int row, column;

printf("Enter Row and Column Sizes: ");

scanf("%d %d", &row, &column);

printf("Enter Matrix-1 Elements: \n");

readMatrix(a, row, column);

printf("Enter Matrix-2 Elements: \n");

readMatrix(b, row, column);

addMatrix(a, b, c, row, column);

printf("Resultant Matrix: \n");

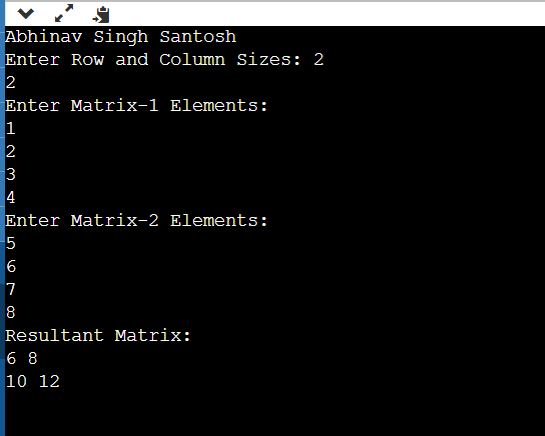
displayMatrix(c, row, column);

return 0;

}



**OUTPUT:**





# Aim:B. Write a C program that uses functions to perform the Multiplication of Two Matrices

# Program:

#include <stdio.h>

void addMatrix(int a[10][10], int b[10][10],

int c[10][10], int row, int column)

{ int i,j;

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

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

c[i][j] = a[i][j] \* b[i][j]; } } }

void readMatrix(int matrix[10][10], int row, int column)

{ int i,j;

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

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

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

void displayMatrix(int matrix[10][10], int row, int column)

{ int i,j;

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

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

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

printf("\n"); } }

int main() {

printf("Abhinav Singh Santosh\n");

int a[10][10];

int b[10][10];

int c[10][10];

int row, column;

printf("Enter Row and Column Sizes: ");

scanf("%d %d", &row, &column);

printf("Enter Matrix-1 Elements: \n");

readMatrix(a, row, column);

printf("Enter Matrix-2 Elements: \n");

readMatrix(b, row, column);

addMatrix(a, b, c, row, column);

printf("Resultant Matrix: \n");

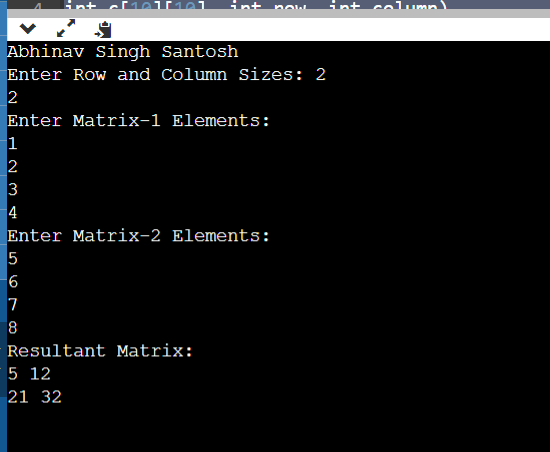
displayMatrix(c, row, column);

return 0;

}



**OUTPUT:**





# Practical 8

**AIM1:Write a C program that uses function stopper form the following operation;**

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

**Program:**

#include <stdio.h>

#include <string.h>

void insertSubstring(char mainString[], char subString[], int position);

int main() {

char mainString[100], subString[100];

int position;

printf("Enter the main string: ");

gets(mainString);

printf("Enter the sub-string: ");

gets(subString);

printf("Enter the position to insert sub-string: ");

scanf("%d", &position);

if (position < 0 || position > strlen(mainString)) {

printf("Invalid position!\n");

return 1; }

insertSubstring(mainString, subString, position);

printf("Modified String: %s\n", mainString);

return 0;

}

void insertSubstring(char mainString[], char subString[], int position) {

int i;

int mainLength = strlen(mainString);

int subLength = strlen(subString);

for (i = mainLength; i >= position; i--) {

mainString[i + subLength] = mainString[i];

}

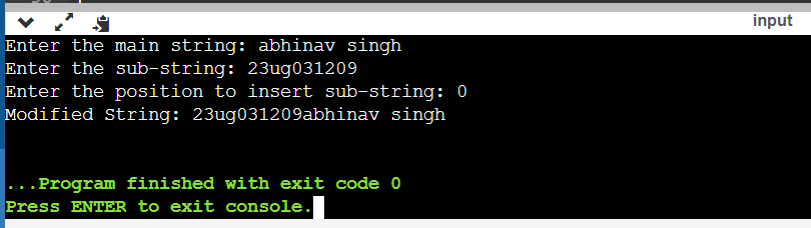
for (i = 0; i < subLength; i++) {

mainString[position + i] = subString[i];

}

}

**OUTPUT:**

****



# Aim: To delete n Characters from a given position in a given string,

**Program:**

#include <stdio.h>

#include <string.h>

void deleteCharacters(char mainString[], int position, int n);

int main() {

char mainString[100];

int position, n;

printf("Enter the main string: ");

gets("main String");

printf("Enter the position to start deletion: ");

scanf("%d", &position);

printf("Enter the number of characters to delete: ");

scanf("%d", &n);

if (position < 0 || position >= strlen(mainString)) {

printf("Invalid position!\n");

return 1; // Exit with an error code

}

deleteCharacters(mainString, position, n);

printf("Modified String: %s\n", mainString);

return 0;

}

void deleteCharacters(char mainString[], int position, int n) {

int i;

int mainLength = strlen(mainString);

for (i = position; i < mainLength - n; i++) {

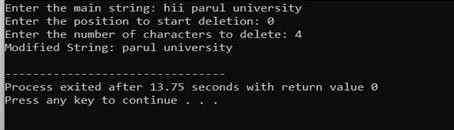
mainString[i] = mainString[i + n];

}

mainString[mainLength - n] = '\0';

}

**OUTPUT:**





**AIM2: Write a C program to determine if the given string is a palindrome or not**

**Program:**

#include <stdio.h>

#include <string.h>

int isPalindrome(char str[]);

int main() {

char inputString[100];

printf("Enter a string: ");

gets(inputString);

if (isPalindrome(inputString)) {

printf("%s is a palindrome.\n", inputString);

} else {

printf("%s is not a palindrome.\n", inputString); }

return 0; }

int isPalindrome(char str[]) {

int i, length = strlen(str);

for (i = 0; i < length / 2; i++) {

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

return 0;

}

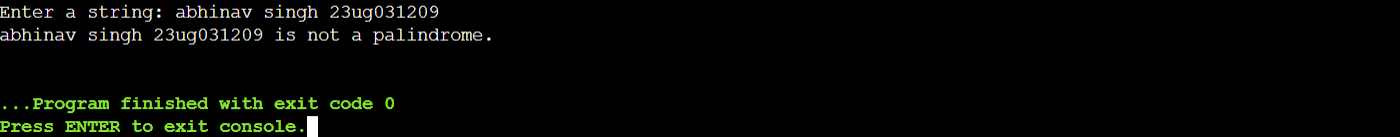
}

return 1;

}



**OUTPUT:**





# Practical 9

**AIM1:Write a C program that displays the position or index in the strings where the string T begins, or -1 if S doesn’t contain**

**Program:**

#include<stdio.h>

#include<string.h>

int 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("Secound String is found in the First string at %d position \n",found-s);

}

else

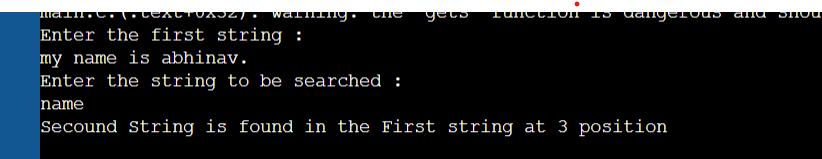
{

printf("-1");

}

return 0; }

**OUTPUT:**

****



**AIM2:Write a C program to count the lines, words and characters in a given text.**

**Program:**

#include<stdio.h>

#include<string.h>

int main()

{

char s[30];

int ch=0,w=0,l=0,i;

puts("Enter the string : ");

gets(s);

for(i=0;i<strlen(s);i++)

{

if(s[i]!=' ') ch++;

if(s[i]==' ') w++;

if(s[i]=='\n') l++;

}

if(ch>0)

{

l++;

w++;

}

printf("No. of characters : %d\n",ch);

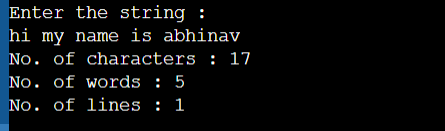
printf("No. of words : %d\n",w);

printf("No. of lines : %d",l);

return 0;

}

**OUTPUT:**

****



# Practical 10

**AIM1: Write a C program to generate Pascal's triangle.**

**Program:**

#include <stdio.h>

int factorial(int n) {

if (n <= 1)

return 1;

else

return n \* factorial(n - 1);

}

void printPascalTriangle(int numRows) {

for (int i = 0; i < numRows; i++) {

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

printf("%d ", factorial(i) / (factorial(j) \* factorial(i - j)));

}

printf("\n");

}

}

int main() {

int rows;

printf("Enter the number of rows for Pascal's Triangle: ");

scanf("%d", &rows);

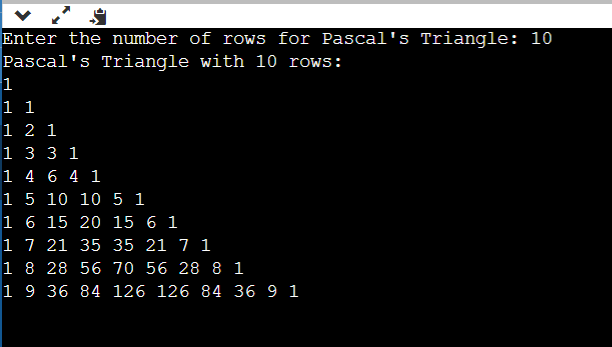
printf("Pascal's Triangle with %d rows:\n", rows);

printPascalTriangle(rows);

return 0;

}

**OUTPUT:**





**AIM2: Write a C program to construct a pyramid of numbers**

**Program:**

#include <stdio.h>

void printSpaces(int spaces) {

for (int i = 0; i < spaces; i++) {

printf(" ");

}

}

void printNumbers(int n) {

int num = 1;

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

printf("%d ", num);

num++;

}

num -= 2;

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

printf("%d ", num);

num--;

}

}

void printPyramid(int rows) {

int spaces = rows - 1;

for (int i = 1; i <= rows; i++) {

printSpaces(spaces);

printNumbers(i);

spaces--;

printf("\n");

}

}

int main() {

int rows;

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

scanf("%d", &rows);

printf("Pyramid of Numbers:\n");

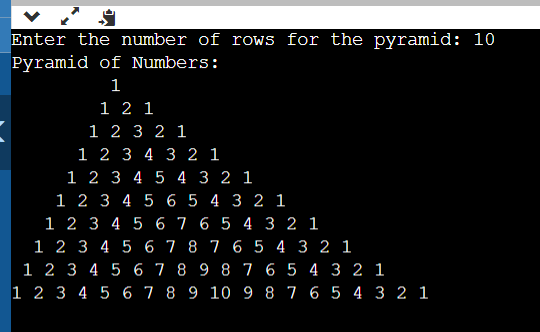
printPyramid(rows);

return 0;

}



**OUTPUT:**





# Practical11

**AIM:** **2's complement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2͛s complement of 11100 is 00100. Write a C program to find the 2's complement of a binary**

**number.**

**Program:**

#include <stdio.h>

#include <math.h>

int 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);

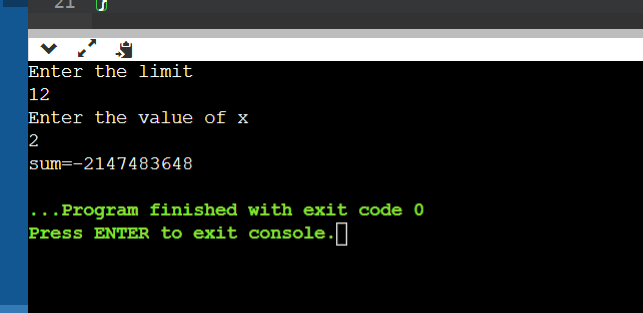
}

printf("sum=%d", sum);

return 0;

**}**

**OUTPUT:**

****



# Practical 12

**AIM1:2'scomplementofanumberisobtainedbyscanningitfromrighttoleftand**

**complementingallthebitsafterthefirstappearanceofa1.Thus2͛scomplementof11100 is 00100. Write a C program to find the 2's complement of a binary number.**

**Program:**

#include <stdio.h>

int main()

{

int n,i;

printf("Enter the number of bits do you want to enter :");

scanf("%d",&n);

char binary[n+1];

char onescomplement[n+1];

char twoscomplement[n+1];

int carry=1;

printf("\nEnter the binary number : ");

scanf("%s", binary);

printf("%s", binary);

printf("\nThe ones complement of the binary number is :");

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

{

if(binary[i]=='0')

onescomplement[i]='1';

else if(binary[i]=='1')

onescomplement[i]='0';

}

onescomplement[n]='\0';

printf("%s",onescomplement);

printf("\nThe twos complement of a binary number is : ");

for(i=n-1; i>=0; i--)

{

if(onescomplement[i] == '1' && carry == 1)

{

twoscomplement[i] = '0';

}

else if(onescomplement[i] == '0' && carry == 1)

{

twoscomplement[i] = '1';

carry = 0;

}

else

{

twoscomplement[i] = onescomplement[i];

}

}

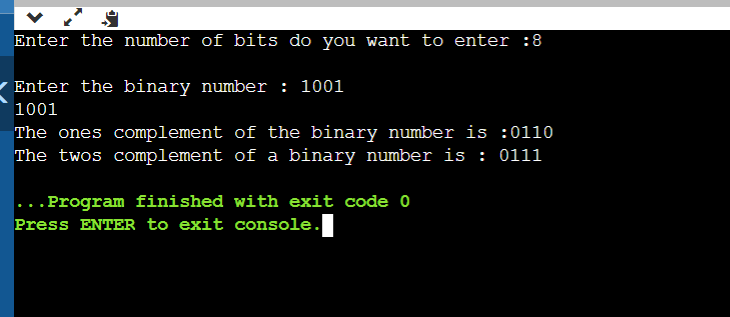
twoscomplement[n]='\0';

printf("%s",twoscomplement);

return 0;

}

**OUTPUT:**





**AIM2:Write a C program to convert a Roman numeral 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, C, D, M): \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;

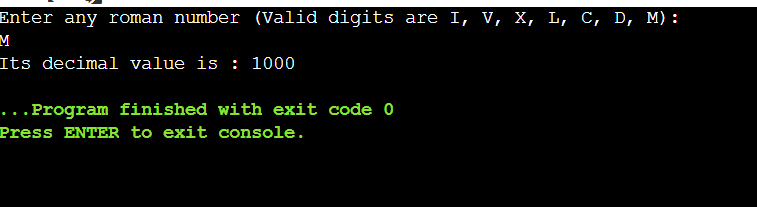
default: value = -1;

}

return value;

}

**OUTPUT:**





**Practical 13**

**AIM:Write a c program on Given an unsorted array arr[] of size N. Rotate the array to the left (counter-clockwise direction) by D steps, where D is a positive integer.**

**Program:**

#include <stdio.h>

void rotateLeft(int arr[], int n, int d) {

int i;

int temp[d];

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

temp[i] = arr[i];

}

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

arr[i - d] = arr[i];

}

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

arr[n - d + i] = temp[i];

}

}

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

int i;

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

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

}

printf("\n");

}

int main() {

int n,i, d;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

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

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

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

}

printf("Enter the number of rotations (D): ");

scanf("%d", &d);

rotateLeft(arr, n, d);

printf("Rotated array:\n");

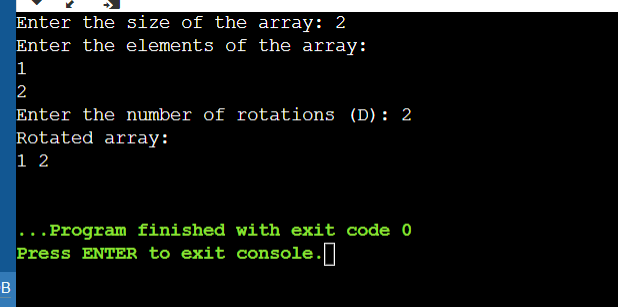
printArray(arr, n);

return 0;

}



**OUTPUT:**

****



AIM2: Write a c Program on given two sorted arrays arr1 and arr2 of size N and M respectivelyandanelementK.Thetaskistofindtheelementthatwouldbeatthek͛th position of the final sorted array.

**Explanation:**

**Input:**

**Array1-1 42 3 5**

**Array2-7 86k**

**= 5**

**Output:5**

**Because The final sorted array would be -1, 2,3,4,5, 6,7,8,The 5th element of this**

**Program:**

#include <stdio.h>

int findKthElement(int arr1[], int N, int arr2[], int M, int K) {

int i = 0, j = 0, k = 0;

int mergedArray[N + M];

while (i < N && j < M) {

if (arr1[i] < arr2[j]) {

mergedArray[k++] = arr1[i++];

} else {

mergedArray[k++] = arr2[j++];

}

}

while (i < N) {

mergedArray[k++] = arr1[i++];

}

while (j < M) {

mergedArray[k++] = arr2[j++];

}

return mergedArray[K - 1];

}

int main() {

int i,N, M, K;

printf("Enter the size of the first array (arr1): ");

scanf("%d", &N);

int arr1[N];

printf("Enter the elements of the first array (arr1) in sorted order:\n");

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

scanf("%d", &arr1[i]);

}

printf("Enter the size of the second array (arr2): ");

scanf("%d", &M);

int arr2[M];

printf("Enter the elements of the second array (arr2) in sorted order:\n");

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

scanf("%d", &arr2[i]);

}

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

scanf("%d", &K);

int result = findKthElement(arr1, N, arr2, M, K);

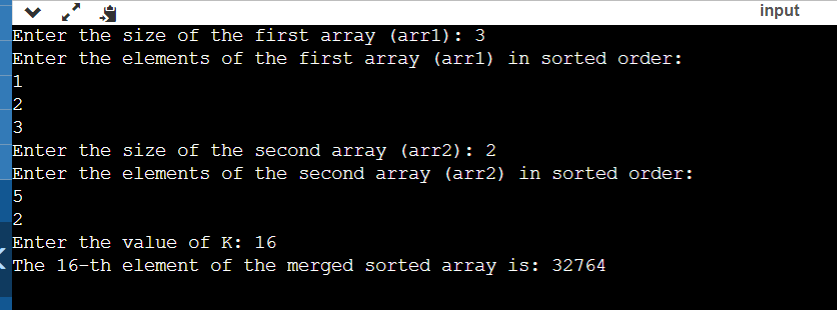
printf("The %d-th element of the merged sorted array is: %d\n", K, result);

return 0;

}



**OUTPUT:**





**Practical 14**

**AIM1:Write a c program to take multiline string input and print individual string length.**

**Program:**

#include <stdio.h>

#include <string.h>

int main() {

char input[1000];

int len;

printf("Enter a multiline string (end with a blank line):\n");

while (fgets(input, sizeof(input), stdin) != NULL && strcmp(input, "\n") != 0) {

len = strlen(input);

if (input[len - 1] == '\n') {

len--;

}

printf("Length of the string: %d\n", len);

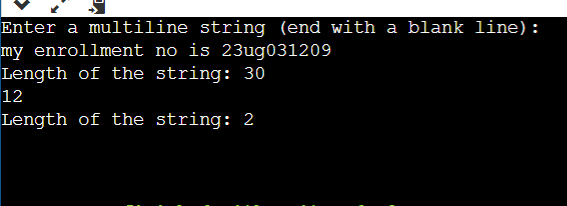
}

return 0;

}



**OUTPUT:**

****



**AIM2:** **Write a c program to reverse the individual word of a given string**

**Explanation:**

**input : Welcome To Bytexl output: emocle**

**Program:**

#include <stdio.h>

void reverse(char\* begin, char\* end)

{

char temp;

while (begin < end) {

temp = \*begin;

\*begin++ = \*end;

\*end-- = temp;

}

}

void reverseWords(char\* s)

{

char\* word\_begin = s;

char\* temp = s;

while (\*temp) {

temp++;

if (\*temp == '\0') {

reverse(word\_begin, temp - 1);

}

else if (\*temp == ' ') {

reverse(word\_begin, temp - 1);

word\_begin = temp + 1;

}

}

reverse(s, temp - 1);

}

int main()

{

char s[] = "Welcome To Bytexl ";

char\* temp = s;

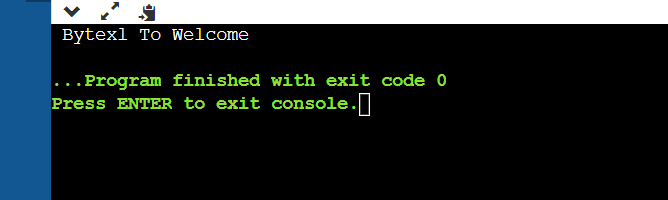
reverseWords(s);

printf("%s", s);

return 0;

}

**OUTPUT:**

****