EX 1	**PROGRAM USING I/O STATEMENTS AND EXPRESSIONS**
DATE:	

AIM:

i] Program to display your personal details

ALGORITHM:

Step 1: Start

Step 2: Declare and initialize the variables for name, address, date of birth, mobile number and age

Step 3: Display name, address, date of birth, mobile number and age

Step 4: End

PROGRAM:

```
#include<stdio.h>
void main()
{
char name[20]= "SAI RAM";
char address[80]= "west tambharam,chennai";
int date=20;
int month=10;
int year=1990;
int mobile=987456321;
int age=25;
printf("\n====================");
printf("\n NAME: %s",name);
printf("\n ADDRESS:%s", address);
printf("\n DOB:%d:%d:%d", date , month, year);
printf("\n MOBILE NUMBER:%d", mobile);
printf("\n AGE:%d", age);
printf("\n====================");
}
```

OUTPUT:

NAME: SAI RAM
ADDRESS:west tambaram,chennai
DOB:20:10:1990
MOBILE NUMBER:987456321
AGE:25

RESULT

Thus the C Program to display the personal details has been executed and the output was verified.

AIM:

ii] Program to get the user details and display it.

ALGORITHM:

Step 1: Start

Step 2: Declare the variables for name, address, date, month, year, mobile number, age.

Step 3: Read values of name, address , date, month, year, mobile number, age from the user.

Step 4: Display name, address, date, month, year, mobile number, age.

Step 5: End

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
char name[20];
char address[80];
int date;
int month;
int year;
long int mobile;
char gender[20];
int age;
printf("\n ENTER YOUR NAME:=");
gets(name);
printf("\nENTER YOUR ADDRESS=");
gets(address);
printf("\nENTER YOUR date/month/year=");
scanf("%d/%d/%d",&date,&month,&year);
printf("\n ENTER YOUR AGE=");
scanf("%d",&age);
printf("\n ENTER YOUR GENDER(MALE/FEMALE)=");
scanf("%s",gender);
printf("\nENTER YOUR MOBILE NUMBER=");
scanf("%ld" ,&mobile);
```

```
printf("\n====================");
printf("\n NAME: %s",name);
printf("\n ADDRESS:%s", address);
printf("\n DOB:%d:%d:%d", date , month, year);
printf("\n AGE:%d", age);
printf("\n GENDER:%s", gender);
printf("\n MOBILE NUMBER:%d", mobile);
printf("\n====================");
return 0;
}
```

OUTPUT:

ENTER YOUR NAME:=karthikeyan

ENTER YOUR ADDRESS=west tambharam,chennai.

ENTER YOUR date/month/year=03/12/1992

ENTER YOUR AGE=28

ENTER YOUR GENDER(MALE/FEMALE)=MALE

ENTER YOUR MOBILE NUMBER=987654321

```
====================
NAME:  karthikeyan
ADDRESS:west tambharam,chennai.
DOB:3:12:1992
AGE:28
GENDER:MALE
MOBILE NUMBER:987654321
========================
```

RESULT:

Thus the C Program to read and display the user details has been executed and the output was verified.

EX 2A	PROGRAM TO DISPLAY BIGGEST OF TWO NUMBERS
DATE:	

AIM:

ii] Program to check biggest of three numbers

ALGORITHM:

Step 1:Start

Step 2:Read three numbers A,B & C

Step 3:If A>B,then go to step 6

Step 4:If B>C,then print B & go to step 8

Step 5:print C is greatest & go to step 8

Step 6:If A>C,then print A is greatest & go to step 8

Step 7:Print C is greatest

Step 8:end

PROGRAM:

```
#include <stdio.h>
void main()
{
   int A,B,C;
   printf("Enter 3 integer number \n");
   scanf("%d",&A);
   scanf("%d",&B);
   scanf("%d",&C);
   if(A>B){
      if(A>C){
         printf(" %d is the Greatest Number \n",A);
      }
      else{
         printf("%d is the greatest Number \n",C);
      }
   }
   else{
         if(B>C){
```

```
            printf("%d is the greatest Number \n",B );
        }
        else{
            printf("%d is the greatest Number \n", C);
        }
    }
}
```

OUTPUT:

Enter three numbers: -4.5
3.9
5.6
5.60 is the largest number.

RESULT:

Thus the C Program to display the personal details has been executed and the output was verified.

EX 2B	PROGRAM TO CHECK WHETHER THE ENTERED CHARACTER IS VOWEL OR NOT(USE SWITCH CASE)
DATE:	

AIM:

ii] Program to check whether the entered character is vowel or not(Use switch case)

ALGORITHM:

Step 1: Start

Step 2: Declare and initialize the variables

Step 3: Get the input from the user and compare with each cases

Step 4: if match found, print vowel otherwise print consonant

Step 5: End

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
int main()
{
   char ch;
   printf("Enter a character: ");
   scanf("%c",&ch);
   //condition to check character is alphabet or not
   if((ch>='A' && ch<='Z') || (ch>='a' && ch<='z'))
   {

      switch(ch)
      {
         case 'A':
         case 'E':
         case 'I':
         case 'O':
         case 'U':
         case 'a':
         case 'e':
         case 'i':
         case 'o':
         case 'u':
            printf("%c is a VOWEL.\n",ch);
```

```
            break;
        default:
            printf("%c is a CONSONANT.\n",ch);
      }
   }
   else
   {
      printf("%c is not an alphabet.\n",ch);
   }

   return 0;
}
```

OUTPUT:

Enter a character

E

E is a vowel

Enter a character

X

X is a consonant

Enter a character

+

+ is not an alphabet

RESULT:

Thus the C Program check whether the entered character is vowel or not (Use switch case) has been executed and the output was verified.

EX 3	PROGRAM TO FIND WHETHER THE GIVEN YEAR IS LEAP YEAR ORNOT
DATE:	

AIM:

To write a C Program to find whether the given year is leap year or not.

ALGORITHM:

Step 1: Start

Step 2: Take integer variable year

Step 3: Check if year is divisible by 400 then DISPLAY "is a leap year"

Step 4: Check if year is not divisible by 100 AND divisible by 4 then DISPLAY "is a leap year"

Step 5: Otherwise, DISPLAY "is not a leap year"

Step 6: Stop

PROGRAM:

```
#include  <stdio.h>
#include <conio.h>
void main()
{
   int year;
   printf("Enter a year :\n");
   scanf("%d", &year);
   if ((year % 400) == 0)
      printf("%d is a leap year \n",year);
   else
      if ((year % 100) != 0 && (year % 4) == 0)
      printf("%d is a leap year \n",year);
   else
      printf("%d is not a leap year \n",year);
}
```

OUTPUT:

Enter a year:
2000
2000 is a leap year

Enter a year:
1900
1900 is not a leap year

RESULT:

Thus the C Program to find whether the given year is leap year or not has been executed successfully and the output was verified.

EX 4	PROGRAM TO PERFORM THE CALCULATOR OPERATIONS, NAMELY, ADDITION, SUBTRACTION, MULTIPLICATION, DIVISION AND SQUARE OF ANUMBER
DATE:	

AIM:

To perform the Calculator operations, namely, addition, subtraction, multiplication, division and square of a number

ALGORITHM:

Step1 : Start the program

Step 2 : Declare the functions

Step 3: Get the input values

Step 4: Perform the operations using functions

Step 5: Print the results

Step 6: Stop the program

PROGRAM:

```
#include<stdio.h>

// functions declaration

int add(int n1, int n2);

int subtract(int n1, int n2);

int multiply(int n1, int n2);

int divide(int n1, int n2);

int square(int n1);

// main function

int main()

{

  int num1, num2;

  printf("Enter two numbers: ");
```

```
  scanf("%d %d", &num1, &num2);
  printf("%d + %d = %d\n", num1, num2, add(num1, num2));
  printf("%d - %d = %d\n", num1, num2, subtract(num1, num2));
  printf("%d * %d = %d\n", num1, num2, multiply(num1, num2));
  printf("%d / %d = %d\n", num1, num2, divide(num1, num2));
  printf("%d^%d=%d\n",num1,num1,square( num1));
  return 0;
}
// function to add two integer numbers
int add(int n1, int n2)
{
  int result;
  result = n1 + n2;
  return result;
}
// function to subtract two integer numbers
int subtract(int n1, int n2)
{
  int result;
  result = n1 - n2;
  return result;
}
// function to multiply two integer numbers
int multiply(int n1, int n2)
{
```

```
  int result;
  result = n1 * n2;
  return result;
}
// function to divide two integer numbers
int divide(int n1, int n2)
{
  int result;
  result = n1 / n2;
  return result;
}
// function to find square of a number
int square(int n1)
{
  int result;
  result = n1*n1;
  return result;
}
```

OUTPUT:

Enter two numbers: 20 5

20 + 5 = 25

20 – 5 = 15

20 * 5 = 100

20 / 5 = 4

20^20 = 400

RESULT

Thus the C Program to perform the Calculator operations, namely, addition, subtraction, multiplication, division and square of a number has been executed and results are verified.

EX 5	PROGRAM TO CHECK WHETHER A GIVEN NUMBER IS ARMSTRONGNUMBER OR NOT?
DATE:	

EX 5:

AIM:

Program to check whether the given number is Armstrong number or not

ALGORITHM:

Step 1: Start

Step 2: Declare Variable sum, temp, num

Step 3: Read num from User

Step 4: Initialize Variable sum=0 and temp=num

Step 5: Repeat Until num>=0 5.1 sum=sum + cube of last digit i.e [(num%10)*(num%10)*(num%10)] 5.2 num=num/10

Step 6: IF sum==temp Print "Armstrong Number" ELSE Print "Not Armstrong Number"

Step 7: Stop

PROGRAM:

```
#include<stdio.h>
int main()
{
  int num,copy_of_num,sum=0,rem;
  printf("\nEnter a number:");
  scanf("%d",&num);
  while (num != 0)
  {
    rem = num % 10;
    sum = sum + (rem*rem*rem);
    num = num / 10;
  }
  if(copy_of_num == sum)
    printf("\n%d is an Armstrong Number",copy_of_num);
  else
    printf("\n%d is not an Armstrong Number",copy_of_num);
  return(0);
}
```

OUTPUT:

Enter a number: 370
370 is an Armstrong Number

RESULT:

Thus the C Program to check whether a given number is Armstrong or not has been executed and the output was verified.

EX 6	PROGRAM TO CHECK WHETHER A GIVEN NUMBER IS ODD OR EVEN
DATE:	

AIM:

Program to check whether a given number is odd or even

ALGORITHM:

Step 1: Start the program

Step 2: Get the number

Step 3: Check the number if it is odd or even using if statement.

Step 4: If the number is even check the condition as n%2 ==0 else it is even.

Step 5: Display the result

Step 6: Stop the program

PROGRAM:

```
#include <stdio.h>
 int main()
{
int number;
printf("Enter an integer: ");
scanf("%d", &number);
// True if the number is perfectly divisible by 2 if(number % 2 == 0)
printf("%d is even.", number);
else
printf("%d is odd.", number);
return 0;
}
```

OUTPUT:

Enter an integer: -7

-7 is odd.

Enter an integer : 8

8 is even

RESULT

Thus the C Program to find whether the given number is odd or even number has been successfully executed and verified

EX 7	PROGRAM TO FIND FACTORIAL OF A GIVEN NUMBER
DATE:	

AIM:

To find factorial of a given number

ALGORITHM:

Step 1. Start the program

Step 2. Get the number

Step 3. If the number < 0 print "Error for finding a factorial"

Step 4. Else Initialize variables factorial←1 i←1

Step 5. Read value of n

Step 6. Repeat the steps until i=n 6.1: factorial←factorial, i←i+1

Step 7. Display factorial

Step 8. Stop the program

PROGRAM:

```
int main()
 {
int n, i; longfactorial = 1;
printf("Enter an integer: ");
scanf("%d",&n);
// show error if the user enters a negative integer
if (n < 0)
printf("Error! Factorial of a negative number doesn't exist.");
 else
      {
for(i=1; i<=n; ++i)
{
```

```
        factorial *= i;
        // factorial = factorial*i;
}
        printf("Factorial of %d = %lu", n, factorial);
              }
        return 0;
           }
```

OUTPUT:

```
Enter an integer: 10
Factoriaof 10 = 3628800
```

RESULT

Thus the C Program to find the factorial of a given number has been successfully executed and verified.

EX 8	**PROGRAM TO FIND OUT THE AVERAGE OF 4 INTEGERS**
DATE:	

AIM:

To find average of 4 integers

ALGORITHM:

Step 1. Start

Step 2. Declare variables

Step 3. Read the 4 numbers

Step 4. Calculate avg=sum/n

Step 5. Display the output

Step 6. Stop

PROGRAM:

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

{

inti,n,sum=0,nu[100];
float avg;

clrscr();

printf("\nEnter the numbers\n");


for(i=0;i<3;i++)

 {

scanf("%d",&nu[i]);
sum = sum + nu[i];

 }

avg = (float)sum/n;
```

```
printf("\nAverage is : %.2f\n",n,avg);
getch();
}
```

OUTPUT:

Enter the numbers

32

45

54

22

Average is 38.25

RESULT

Thus the C Program to find the average of 4 numbers has been executed and verified.

EX 9	PROGRAM TO DISPLAY ARRAY ELEMENTS USING 2D ARRAYS
DATE:	

AIM:

To display array elements using 2D arrays

ALGORITHM:

Step1 : Start the program

Step2 : Get the elements of the array

Step 3 : Display the array elements

Step 4 : Stop the program

PROGRAM:

```
#include<stdio.h>
int main(){
  /* 2D array declaration*/
  int disp[2][3];
  /*Counter variables for the loop*/
  int i, j;
  for(i=0; i<2; i++) {
    for(j=0;j<3;j++) {
      printf("Enter value for disp[%d][%d]:", i, j);
      scanf("%d", &disp[i][j]);
    }
  }
  //Displaying array elements
  printf("Two Dimensional array elements:\n");
  for(i=0; i<2; i++) {
```

```
    for(j=0;j<3;j++) {
      printf("%d ", disp[i][j]);
      if(j==2){
        printf("\n");
      }
    }
  }
  return 0;
}
```

OUTPUT:

Enter value for disp[0][0]:1

Enter value for disp[0][1]:2

Enter value for disp[0][2]:3

Enter value for disp[1][0]:4

Enter value for disp[1][1]:5

Enter value for disp[1][2]:6

Two Dimensional array elements:

1 2 3

4 5 6

RESULT:

Thus the C Program to display the array elements of the 2D array has been executed and the result was verified

EX 10	PROGRAM TO PERFORM SWAPPING USING FUNCTIONS
DATE:	

AIM:

To perform swapping using functions

ALGORITHM:

Step 1. Start the program

Step 2. Declare and get the two integer variables a and b.

Step 3. call the swap () function

In swap definition use the temporary variable and assign temp =a

b=temp

Step 4. Print the a and b values

Step 5. Stop the program

PROGRAM:

```
#include<stdio.h>

void main()

{

void swap(int,int);
inta,b,r;
clrscr();
printf("enter value for a&b: ");
scanf("%d%d",&a,&b);
swap(a,b);
getch();

}

void swap(inta,int b)

int temp;
temp=a;
a=b;
```

```
b=temp;
printf("after swapping the value for a & b is : %d %d",a,b);

}
```

OUTPUT:

Enter the value of a & b : 34 78

after swapping the value for a & b is 78,34

RESULT:

Thus the C Program to swap two numbers using functions has been executed and verified

EX 11	PROGRAM TO DISPLAY ALL PRIME NUMBERS BETWEEN TWO INTERVALS USING FUNCTIONS
DATE:	

AIM:

To display all prime numbers between two intervals using functions

ALGORITHM:

Step1: Start the Program

Step 2: Get the intervals

Step 3: Find and Display the prime numbers ie., the numbers that are divisible by 1 and itself between the intervals

Step 4: Stop the Program

PROGRAM:

```
#include <stdio.h>

/* Function declarations */

int isPrime(int num);

void printPrimes(int lowerLimit, int upperLimit);

int main()

{

int lowerLimit, upperLimit;

printf("Enter the lower and upper limit to list primes: ");

scanf("%d%d", &lowerLimit, &upperLimit);

/* Call function to print all primes between the given range*/

printPrimes(lowerLimit, upperLimit);

return 0;
```

```
}
/* Print all prime numbers between lower limit and upper limit*/
void printPrimes(int lowerLimit, int upperLimit)
{
printf("All prime number between %d to %d are: ", lowerLimit, upperLimit);
while(lowerLimit <= upperLimit)
{
/* Print if current number is prime*/
if(isPrime(lowerLimit))
{
printf("%d, ", lowerLimit);
}
lowerLimit++;
}
}
/*Check whether a number is prime or not*/
/*Returns 1 if the number is prime otherwise 0*/
int isPrime(int num)
{
int i;
for(i=2; i<=num/2; i++)
```

```
{
/* If the number is divisible by any number*/
/*other than 1 and self then it is not prime*/
if(num % i == 0)
{
return 0;
}
}
return 1;
}
```

OUTPUT:

Enter the lower and upper limit to list primes:

1 100

All prime number between 1 100 are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

RESULT:

Thus the C Program to find the prime numbers between two intervals has been executed and verified.

EX 12	PROGRAM TO REVERSE A SENTENCE USING RECURSION
DATE:	

AIM:

To reverse a sentence using recursion

ALGORITHM:

Step 1: Start

Step 2: Declare the function reverse

Step 3: Call the reverse function

Step 4: Get the sentence from the user and reverse it recursively

Step 5: stop the execution.

PROGRAM:

```
#include  <stdio.h>
void reverseSentence();
int main() {
   printf("Enter a sentence: ");
   reverseSentence();
   return 0;
}

void reverseSentence() {
   char c;
   scanf("%c", &c);
   if (c != '\n') {
      reverseSentence();
      printf("%c", c);
   }
}
```

OUTPUT:

```
Enter a sentence: margorp emosewa
awesome PROGRAM:
```

RESULT

Thus the C Program to reverse a sentence has been executed and verified

EX 13	PROGRAM TO GET THE LARGEST ELEMENT OF AN ARRAY USINGFUNCTION
DATE:	

AIM:

To get the largest element of an array using function

ALGORITHM:

Step 1: Start the program

Step 3: Initialize the array elements

Step 4: Find the largest number of the array

Step 5: Display the largest number

Step 6: Stop the program

PROGRAM:

```
#include <stdio.h>
#include <conio.h>
max(int  [],int);
void main()
{
        int a[]={10,5,45,12,19};
        int n=5,m;
        clrscr();
        m=max(a,n);
        printf("\nmaximum number is %d",m);
        getch();
}
max(int x[],int k)
{
```

```
    int t,i;
    t=x[0];
    for(i=1;i<k;i++)
  {
        if(x[i]>t)
            t=x[i];
    }
    return(t);
}
```

OUTPUT:

Maximum number is 45

RESULT:

Thus the C Program to display the largest number in an array using function has been executed and verified.

EX 14	PROGRAM TO CONCATENATE TWO STRINGS
DATE:	

AIM:

To concatenate two strings

ALGORITHM:

Step1: Start
Step 2: Get the two Strings to be concatenated.
Step 3: Declare a new String to store the concatenated String.
Step 4: Insert the first string in the new string.
Step 5: Insert the second string in the new string.
Step 6: Print the concatenated string.
Step 7: Stop

PROGRAM:

```
#include <stdio.h>
#include <string.h>
int main()
{
        char destination[] = "Hello ";
         char source[] = "World!";
         printf("Concatenated String: %s\n", strcat(destination,source));
         return 0;
}
```

OUTPUT:

Concatenated String: Hello World!

RESULT:

Thus the C Program to concatenate two strings has been executed and the result was verified.

EX 15	PROGRAM TO FIND THE LENGTH OF STRING
DATE:	

AIM:

To find the length of the given string

ALGORITHM:

Step 1 : Start the program

Step 2: Get the string

Step 3: Find the length of the string

Step 4 : Display the length of the string

Step 5 : Stop the program

PROGRAM:

i) Using Library Function

```
#include <stdio.h>
#include <string.h>
int main()
{
char a[100];
int length;
printf("\n Enter a string to calculate its length=");
gets(a);
length = strlen(a);
printf("\nLength of the string = %d\n", length);
return 0;
}
```

ii) Without Using Library Function

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

```
{
char i=0;a[100];
int length;
printf("\nEnter a string to calculate its length=");
scanf(“%s”,str);
while(string1[i] !='\0') {
i++;
}
length=i;
printf (“\n Length of the string = %d\n",length);
return 0;
}
```

OUTPUT:

Enter a string to calculate its length=Introduction

Length of the string: 12

RESULT:

Thus the C Program to find the length of the string has been executed and verified.

EX 16	PROGRAM TO FIND THE FREQUENCY OF A CHARACTER IN A STRING
DATE:	

AIM:

To find the frequency of a character in a string.

ALGORITHM:

Step 1 : Start the program

Step 2 : Get the string

Step 3 : Get the character for which frequency needs to be found

Step 4 : Display the frequency

Step 5 : Stop the program

PROGRAM:

```
#include <stdio.h>
int main() {
   char str[1000], ch;
   int count = 0;

   printf("Enter a string: ");
   fgets(str, sizeof(str), stdin);

   printf("Enter a character to find its frequency: ");
   scanf("%c", &ch);

   for (int i = 0; str[i] != '\0'; ++i) {
      if (ch == str[i])
         ++count;
   }

   printf("Frequency of %c = %d", ch, count);
   return 0;
```

```
}
```

OUTPUT:

```
Enter a string: This website is awesome.

Enter a character to find its frequency: e

Frequency of e = 4
```

RESULT:

Thus the C Program to find the frequency of a character in a string has been executed and verified.

EX 17	PROGRAM TO STORE STUDENT INFORMATION IN STRUCTURE AND DISPLAY IT
DATE:	

AIM:

To store student information in structure and display it

ALGORITHM:

Step 1: START

Step 2: Read student details like name, mark1,2,3

Step 3: Calculate total, and average

Step 4: Display the grade

Step 5: STOP

PROGRAM:

```
#include<stdio.h>
struct student
{
int roll_no, mark1, mark2, mark3, total;
float average;
char name[10],grade;
};
void struct_funct_student(struct student stu);
int main()
{
struct student stud;
printf("\nRoll No.=");
scanf("%d",&stud.roll_no);
printf("Name=");
scanf("%s",stud.name);
printf("Mark1=");
scanf("%d",&stud.mark1);
printf("Mark2=");
scanf("%d",&stud.mark2);
printf("Mark3=");
```

```
scanf("%d",&stud.mark3);
struct_funct_student(stud);
return 0;
}
void struct_funct_student( struct student stu)
{
stu.total = stu.mark1 + stu.mark2 + stu.mark3;
stu.average = stu.total / 3;
if(stu.average >= 90)
stu.grade='S';
else if(stu.average >= 80)
stu.grade='A';
else if(stu.average >= 70)
stu.grade='B';
else if(stu.average >= 60)
stu.grade='C';
else if(stu.average >= 50)
stu.grade='D';
else
stu.grade='F';
printf("\n ROLL NO. \t NAME \t TOTAL \t AVG \t
GRADE \n");
printf("%d \t %s \t %d \t %f \t %c",
stu.roll_no,stu.name,stu.total,stu.average,stu.grade);
}
```

OUTPUT:

Roll No.= 1

Name= a

Mark1= 95

Mark2= 94

Mark3= 96

ROLL NO.	NAME	TOTAL	AVG	GRADE
1	a	285	95.000000	S

RESULT:

Thus the C Program to store and display student details using structures has been executed and the result was verified.

EX 18	PROGRAM TO READ THE STUDENT DATA AND CALCULATE THETOTAL MARKS
DATE:	

AIM:

To read the student data and calculate the total marks

ALGORITHM:

Step 1: Start the program

Step 2: Get the details of the 10 students in five subjects

Step 3: Calculate the total marks of each student

Step 4 : Calculate the student who got the highest total marks

Step 5: Display the results

Step 6: Stop the Program

PROGRAM:

```
#include<stdio.h>
struct student
{
int sub1;
int sub2;
int sub3;
int sub4;
int sub5;
};
void main()
{
 struct student s[10];
 int i,total=0;
```

```
    clrscr();
    for(i=0;i<=9;i++)
      {
          printf("\nEnter Marks in Five Subjects = ");
          scanf("%d%d%d",& s[i].sub1,&s[i].sub2,&s[i].sub3,&s[i].sub4,&s[i].sub5);
          total=s[i].sub1+s[i].sub2+s[i].sub3+s[i].sub4+s[i].sub5;
          printf("\nTotal marks of s[%d] Student= %d",i,total);
      }
          getch();
}
```

OUTPUT:

Enter Marks in Five Subjects

80 70 90 80 98

Total Marks of 1 student = 83.6

RESULT:

Thus the C Program to print the student details has been executed and the result was verified.

EX :19	TELEPHONE DIRECTORY USING RANDOM ACCESS FILE
DATE:	

AIM :

To insert, update, delete and append telephone details of an individual or a company into a telephone directory using random access file.

ALGORITHM :

Step 1: Create a random access file
Step 2: call the respective procedure to insert, update, delete or append based on user choice
Step 3: Access the random access file to make the necessary changes and save

PROGRAM

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

struct dir
{
char name[20];
char number[10];
};

void insert(FILE *);
void update(FILE *);
void del(FILE *);
void display(FILE *);
void search(FILE *);

int record = 0;
int main(void) {
int choice = 0;
FILE *fp = fopen( "telephone.dat", "rb+" );
if (fp == NULL ) perror ("Error opening file");
while (choice != 6)
{
printf("\n1 insert\t 2 update\n");
printf("3 delete\t 4 display\n");
printf("5 search\t 6 Exit\n Enter choice:");
scanf("%d", &choice);
switch(choice)
{
case 1: insert(fp); break;
case 2: update(fp); break;
case 3: del(fp); break;
case 4: display(fp); break;
case 5: search(fp); break;
default: ;
}
}
```

```
fclose(fp);
return 0;
}

void insert(FILE *fp)
{

struct dir contact, blank;
fseek( fp, -sizeof(struct dir), SEEK_END );
fread(&blank, sizeof(struct dir), 1, fp);
printf("Enter individual/company name: ");
scanf("%s", contact.name);
printf("Enter telephone number: ");
scanf("%s", contact.number);
fwrite(&contact, sizeof(struct dir), 1, fp);
}

void update(FILE *fp)
{
char name[20], number[10];
int result;
struct dir contact, blank;
printf("Enter name:");
scanf("%s", name);
rewind(fp);
while(!feof(fp))
{
result = fread(&contact, sizeof(struct dir), 1, fp);
if(result != 0 && strcmp(name, contact.name) == 0)
{
printf("Enter number:");
scanf("%s", number);
strcpy(contact.number, number);
fseek(fp, -sizeof(struct dir), SEEK_CUR);
fwrite(&contact, sizeof(struct dir), 1, fp);
printf("Updated successfully\n");
return;
}
}
printf("Record not found\n");
}

void del(FILE *fp)
{
char name[20], number[10];
int result, record=0;
struct dir contact, blank = {"", ""};
printf("Enter name:");
scanf("%s", name);
rewind(fp);
while(!feof(fp))
{
result = fread(&contact, sizeof(struct dir), 1, fp);
if(result != 0 && strcmp(name, contact.name) == 0)
{
```

```
fseek(fp, record*sizeof(struct dir), SEEK_SET);
fwrite(&blank, sizeof(struct dir), 1, fp);


printf("%d Deleted successfully\n", record-1);
return;
}
record++;
}
printf("not found in %d records\n", record);

}

void display(FILE *fp)
{
struct dir contact;
int result;
rewind(fp);
printf("\n\n    Telephone directory\n");
printf("%20s %10s\n", "Name", "Number");
printf("********************************\n");
while(!feof(fp))
{
result = fread(&contact, sizeof(struct dir), 1, fp);
if(result != 0 && strlen(contact.name) > 0)
printf("%20s %10s\n",contact.name, contact.number);
}
printf("********************************\n");

}

void search(FILE *fp)
{
struct dir contact;
int result; char name[20];
rewind(fp);
printf("\nEnter name:");
scanf("%s", name);
while(!feof(fp))
{
result = fread(&contact, sizeof(struct dir), 1, fp);
if(result != 0 && strcmp(contact.name, name) == 0)
{
printf("\n%20s %10s\n",contact.name, contact.number);
return;
}
}
printf("Record not found\n");

}
```

OUTPUT:

1 insert 2 update

```
3 delete        4 display
5 search        6 Exit
 Enter choice: 4
   Telephone directory
            Name     Number
*******************************
              bb      11111
*******************************

1 insert        2 update
3 delete        4 display
5 search        6 Exit
 Enter choice: 5

Enter name: bb

              bb      11111

1 insert        2 update
3 delete        4 display
5 search        6 Exit
 Enter choice: 1
Enter individual/company name:  aa
Enter telephone number:  222222

1 insert        2 update
3 delete        4 display
5 search        6 Exit
 Enter choice: 2
Enter name: aa
Enter number: 333333
Updated successfully

1 insert        2 update
3 delete        4 display
5 search        6 Exit
 Enter choice:

   Telephone directory
            Name     Number
*******************************
              bb      11111
              aa     333333
*******************************
1 insert        2 update
3 delete        4 display
5 search        6 Exit
```

```
 Enter choice: 3

Enter name: aa
1 Deleted successfully

1 insert        2 update
3 delete        4 display
5 search        6 Exit
 Enter choice: 4


   Telephone directory
            Name     Number
*******************************
              bb     11111
*******************************

1 insert        2 update
3 delete        4 display
5 search        6 Exit
Enter choice: 6
```

RESULT:

Thus the C program To insert, update, delete and append telephone details of an individual or a company into a telephone directory using random access file was successfully written and executed.

EX :20	PROGRAM TO COUNT THE NUMBER OF ACCOUNT HOLDERS WHOSEBALANCE IS LESS THAN THE MINIMUM BALANCE USING SEQUENTIAL ACCESS FILE
DATE:	

AIM:

To count the number of account holders whose balance is less than the minimum balance using sequential access file.

ALGORITHM :

Step 1 : Start the program

Step 2 : Read choice to insert records & count minimum balance account

1. If choice is 1, then
- Open a dat file in write mode
- Read the No. of records
- Write the records into the file using fprintf() function
- Close the file

2. If Choice is 2, then
- Open the file in Read mode
- Read the records one by one using fscanf(0 function until reach the end of file.
- Check the account balance with min bal.
- If account balance is less than min balance, then display the account details
- Close the file

Step 3 : Stop the program

PROGRAM:

```
#include <stdio.h>

void insert();
void count();

int main(void)
{
  int choice = 0;
  while (choice != 3)
  {
    printf("\n1 insert records\n");
    printf("2 Count min balance holders\n");
    printf("3 Exit\n");
    printf("Enter choice:");
    scanf("%d", &choice);
```

```
    switch(choice)
    {
      case 1: insert(); break;
      case 2: count(); break;
    }
  }
}

void insert()
{
  unsigned int account,i;
  char name[30];
  double balance;
  FILE* cfPtr;

  if ((cfPtr = fopen("clients.dat", "w")) == NULL) {
    puts("File could not be opened");
  }
  else {
    int records,i=0;
    printf("Enter the No. of records ");
    scanf("%d", &records);
    while (i<records)
     {
      printf("Enter the account, name, and balance.");
      scanf("%d%29s%lf", &account, name, &balance);
      fprintf(cfPtr, "%d %s %.2f\n", account, name, balance);
      i++;
     }
    fclose(cfPtr);
  }
}

void count()
{
  unsigned int account;
  char name[30];
  double balance;
  float minBal = 5000.00;
  int count = 0;
  FILE *cfPtr;
  if ((cfPtr = fopen("clients.dat", "r")) == NULL)
    printf("File could not be opened");
   else
```

```
  {
   printf("%-10s%-13s%s\n", "Account", "Name", "Balance");
   fscanf(cfPtr, "%d%29s%lf", &account, name, &balance);

  while (!feof(cfPtr))
  {
    if (balance < minBal)
    {
     printf("%-10d%-13s%7.2f\n", account, name, balance);
     count++;
    }
    fscanf(cfPtr, "%d%29s%lf", &account, name, &balance);
   }

   fclose(cfPtr);
   printf("The number of account holders whose balance is less than the minimum balance:
%d", count);
  }
}
```

OUTPUT:

```
1 insert records
2 Count min balance holders
3 Exit
Enter choice:1
Enter the No. of records 2
Enter the account, name, and balance.1001 A 10000
Enter the account, name, and balance.1002 B 300

1 insert records
2 Count min balance holders
3 Exit
Enter choice:2
Account   Name        Balance
1002      B           300.00
The number of account holders whose balance is less than the minimum balance: 1
1 insert records
2 Count min balance holders
3 Exit
Enter choice:
```

*

RESULT :

Thus the C Program to count the number of account holders whose balance is less than the minimum balance using sequential access file