SONA COLLEGE OF ARTS AND SCIENCE
Affiliated to Periyar University, Salem -11

DEPARTMENT OF COMPUTER SCIENCE WITH CYBER SECURITY

Practical Record

Name :

Register Number :

Subject Code :

Subject Title :

Year / Sem :

ACADEMIC YEAR: 2024 – 2025

SONA COLLEGE OF ARTS AND SCIENCE
Affiliated to Periyar University, Salem

Certificate

This is to certify that the Practical Record **"Programming in C Lab"** is a bonafide work done by _____

Reg. No. _____submitted to the Department of Computer Science with Cyber Security, during the academic year 2024 – 2025.

SUBJECT IN-CHARGE **HEAD OF THE DEPARTMENT**

Submitted for University Practical Examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INDEX

Ex. No. : 1	USING INPUT / OUTPUT FUNCTIONS
Date:	

AIM :

To write the C programs using Input / Output functions

ALGORITHM :

Step 1: Start the process

Step 2: Initialize the necessary variables with valid data types

Step 3: Get the proper input by using input function

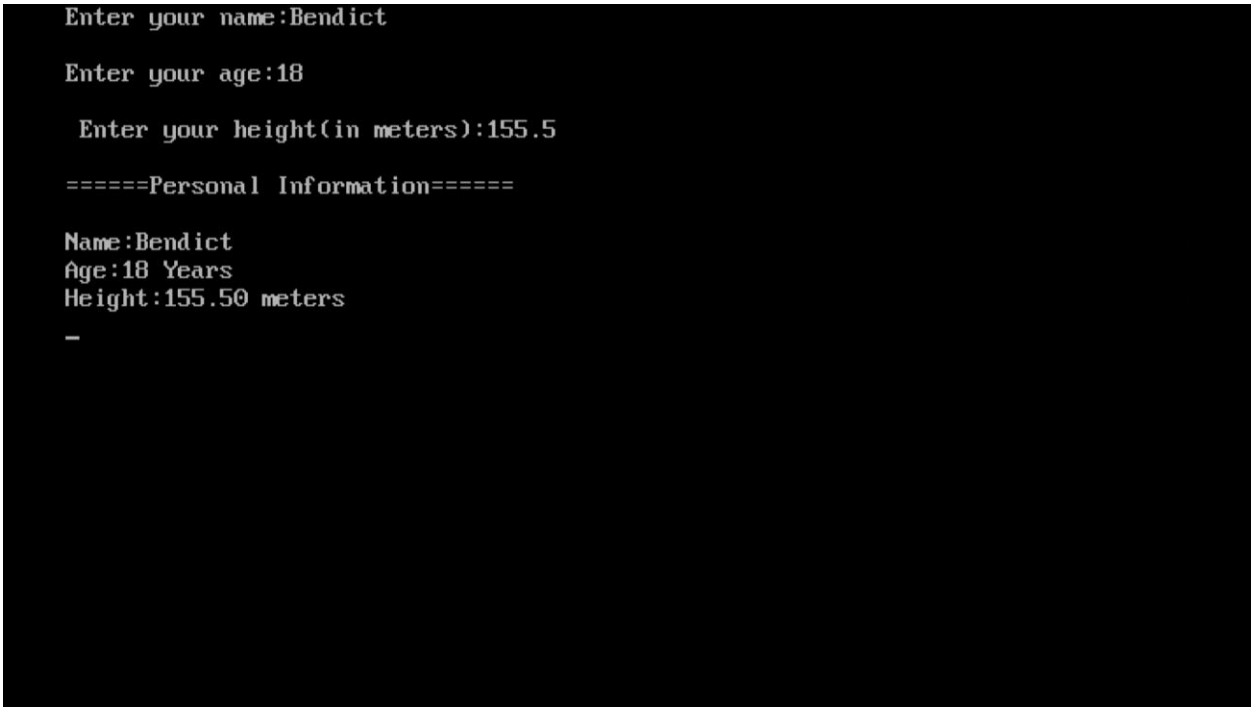
Step 4: To print the output by using output function

Step 5: Stop the process

CODING:

```
#include<stdio.h>
#include<conio.h>
int main()
{
char name[50]; int age; float height;
printf("Enter your name: ");
fgets(name, sizeof(name), stdin);
printf("Enter your age: ");
scanf("%d", &age);
printf("Enter your height (in meters): ");
scanf("%f", &height);
printf("\n=== Personal Information ===\n");
printf("Name: %s", name);
printf("Age: %d years\n", age);
printf("Height: %.2f meters\n", height);
return 0;
}
```

OUTPUT SCREENSHOT:



RESULT:

Thus the above program used various Input and Output functions and verified successfully.

Ex. No.: 2	USING CONDITIONAL STRUCTURES
Date:	

AIM:

To develop a C program that calculates a grade obtained by a student using if..else if structure.

ALGORITHM :

Setp 1: Start the program

Setp 2: Get the input values of three subject marks using scanf.

Step 3: Calculate total and average.

Step 4: Assign the grade based on the average value.

Step 5: Print the value of total, average and grade.

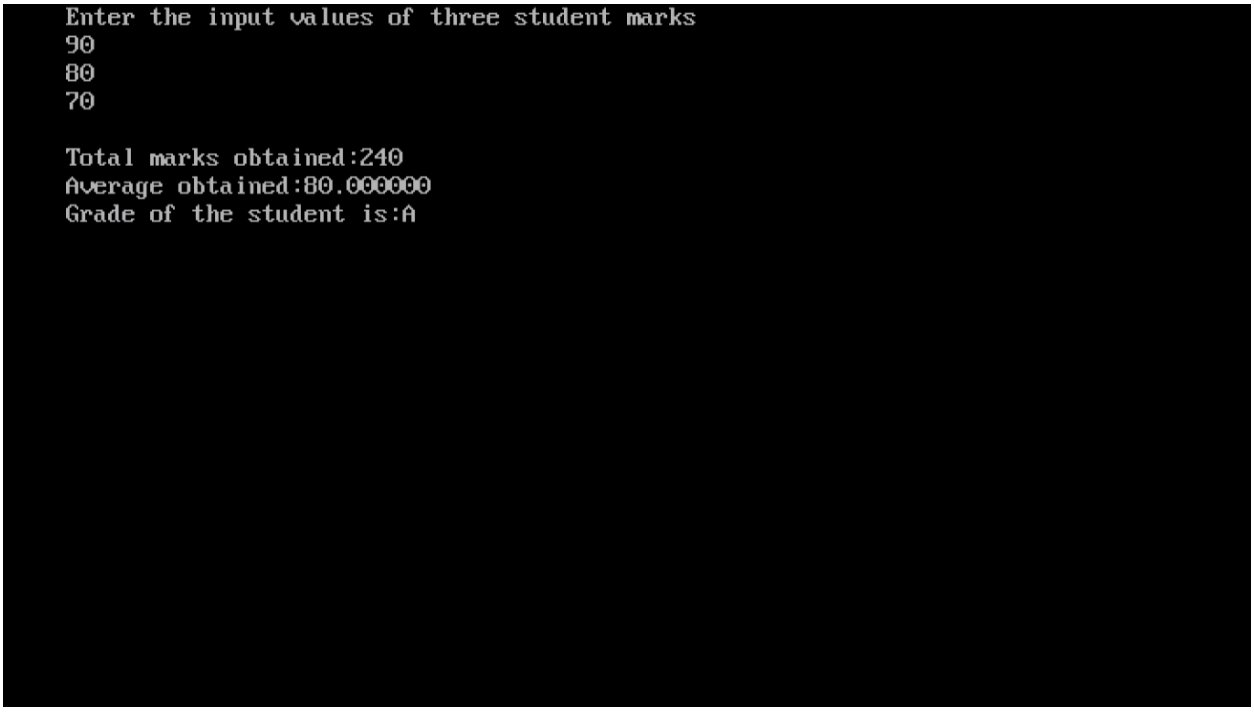
Step 6: Stop the execution.

CODING:

```
#include <stdio.h>

int main ()

{

    int m1,m2,m3,tot; float avg;

    char g;

    tot=0;

    printf("Enter the input values of 3 marks\n");

    scanf("%d%d%d",&m1,&m2,&m3);

    tot=m1+m2+m3;

    printf("Total mark obtained=%d\n",tot);

    avg=tot/3;

    printf("Average obtained=%f\n",avg);

if(avg>=90)

{

     g='O';

}
```

```
else if(avg>=80 && avg<90)
{
    g='A';
}
else if(avg>=70 && avg<80)
{
    g='B';
}
else if(avg>=60 && avg<70)
{
    g='C';
}
else if(avg>=50 && avg<60)
{
    g='D';
}
else
{
    g='E';
}
printf("Grade of a student= %c",g);
return 0;
}
```

OUTPUT SCREENSHOT:



RESULT:

Thus the above program using if- else if structure executed successfully and the results are verified.

Ex. No.: 3	COMMAND LINE ARGUMENTS
Date:	

AIM:

To develop a C program that calculates the summation of command line arguments.

ALGORITHM :

Setp 1: Start the program

Setp 2: Pass the count of total number of command line arguments as first argument to the main()

Step 3: Pass the program name and other argument values as the next to main()

Step 4: Use for loop to calculate the sum of command line arguments

Step 5: Print the values of arguments and sum

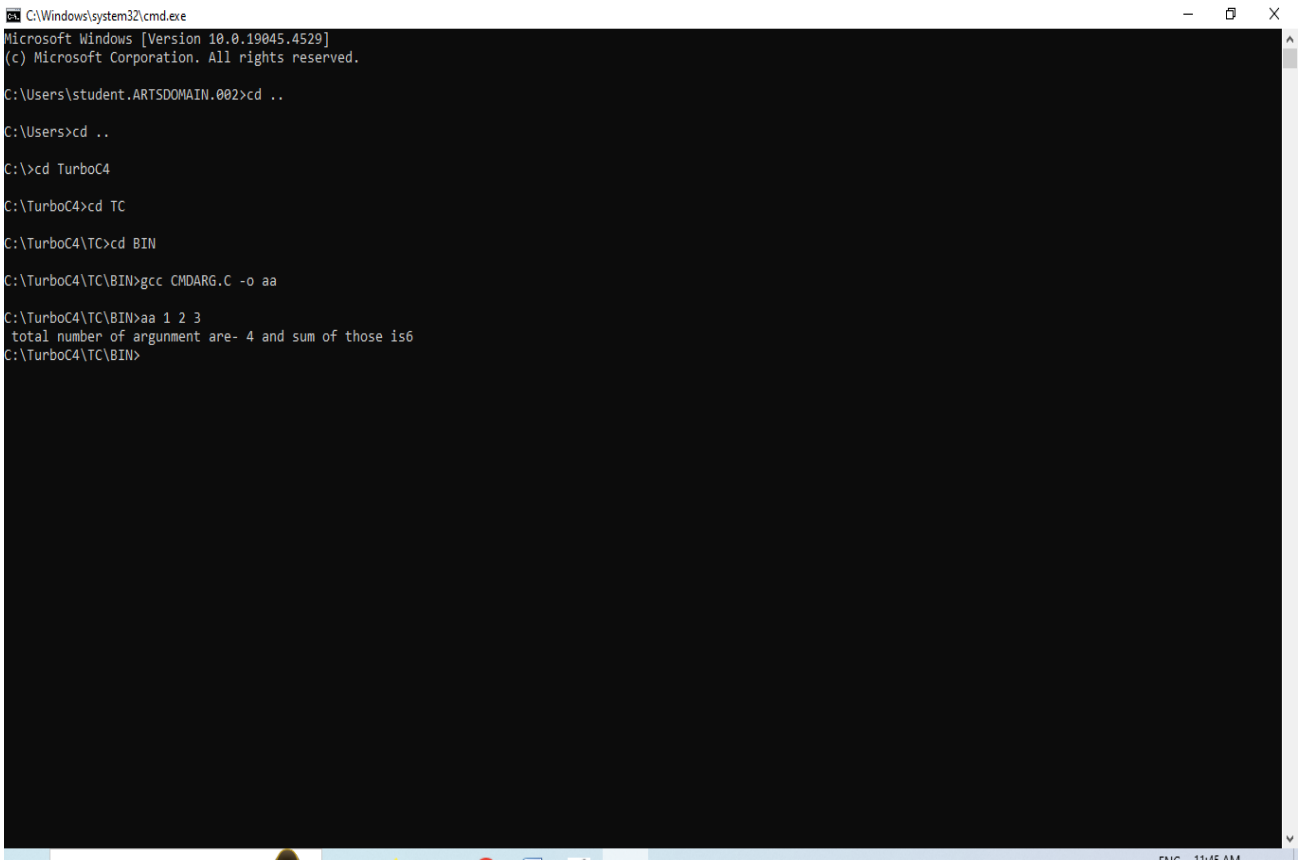
Step 6: Stop the execution.

CODING:

```
#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int main(int argc, char *argv[])

{

  int i, n, sum = 0;

  clrscr() ;

    if(argc == 1)

{

      printf("Please provide command line arguments!!!");

      return 0;

  }

  else

{

      printf("Total number of arguments are - %d and sum of those is ", argc);

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

{

       n = atoi(argv[i]);
```

```
        sum += n;
      }
      printf("%d\n", sum);
      return 0;
   }
}
```

OUTPUT SCREENSHOT:



RESULT:

Thus the above program using command line arguments executed successfully and the results are verified.

Ex. No.: 4	USING ARRAYS
Date:	

AIM:

To develop a C program that finds the largest of array elements.

ALGORITHM :

Setp 1: Start the program

Setp 2: Get the input value for size of an array

Step 3: Use a for loop to get the values of array elements

Step 4: Assign the first element array[0] to the variable large

Step 5: Check if any other element is greater than array[0]. If yes, then assign that element to the variable large

Step 6: Print the largest value of array

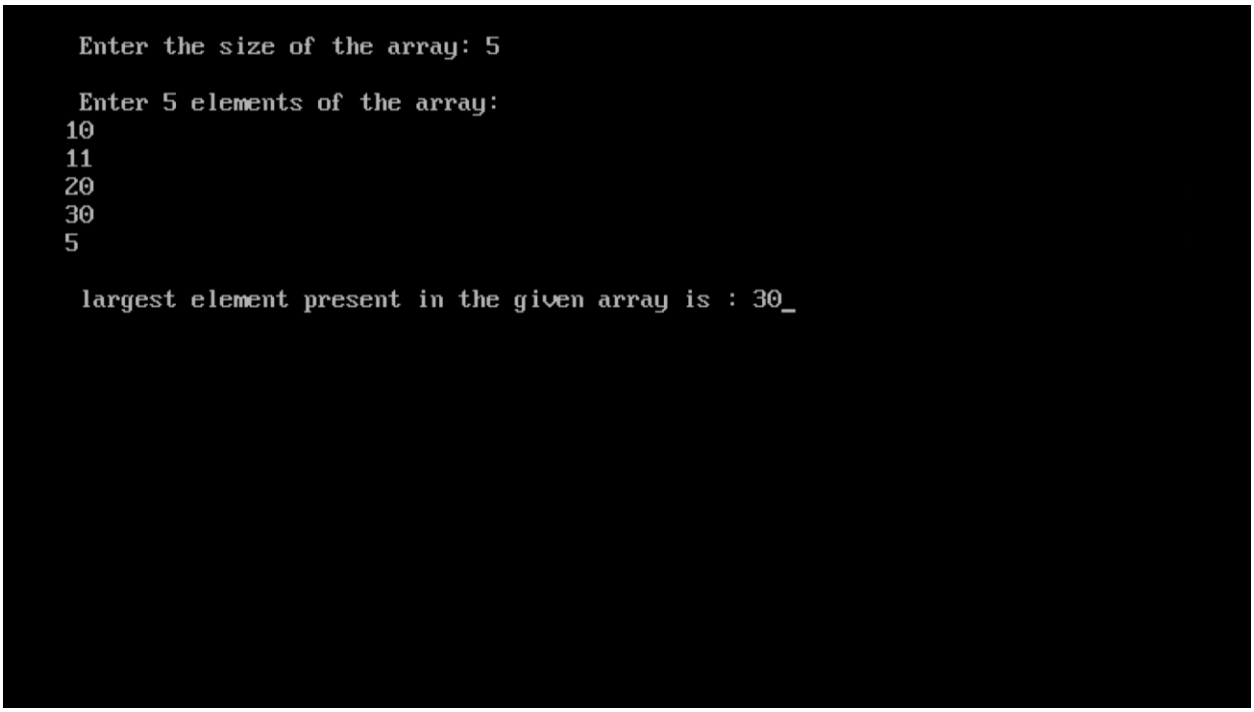
Step 7: Stop the execution.

CODING:

```
#include <stdio.h>

 #include<conio.h>

int main()

{

   int size, i, large;

  clrscr();

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

   scanf("%d", &size);

   int array[size];  //Declaring array

    //Input array elements

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

    for (i = 0; i < size; i++)
   {
      scanf(" %d", &array[i]);
   }

     //Declaring Largest element as the first element

   large = array[0];

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

   {
```

```
    if (large < array[i])
    large = array[i];
  }
  printf("\n largest element present in the given array is : %d", largest);
  getch();
  return 0;
}
```

OUTPUT SCREENSHOT:



RESULT:

Thus the above program using arrays executed successfully and the results are verified.

Ex.No.:5	STRING MANIPULATIONS
DATE:	

AIM:

To develop a C Program to perform the string handling functions

ALGORITHM:

STEP 1 : Start the Program

STEP 2 : Declare the character arrays str, str1 to give the input value for strings

STEP 3: Perform the following string handling functions

(i) strlen()

(ii) strcpy()

(iii) strcat()

(iv) strcmp()

(v) strrev()

(vi) strlwr()

(vii) strupr()

STEP 4: Print the output strings

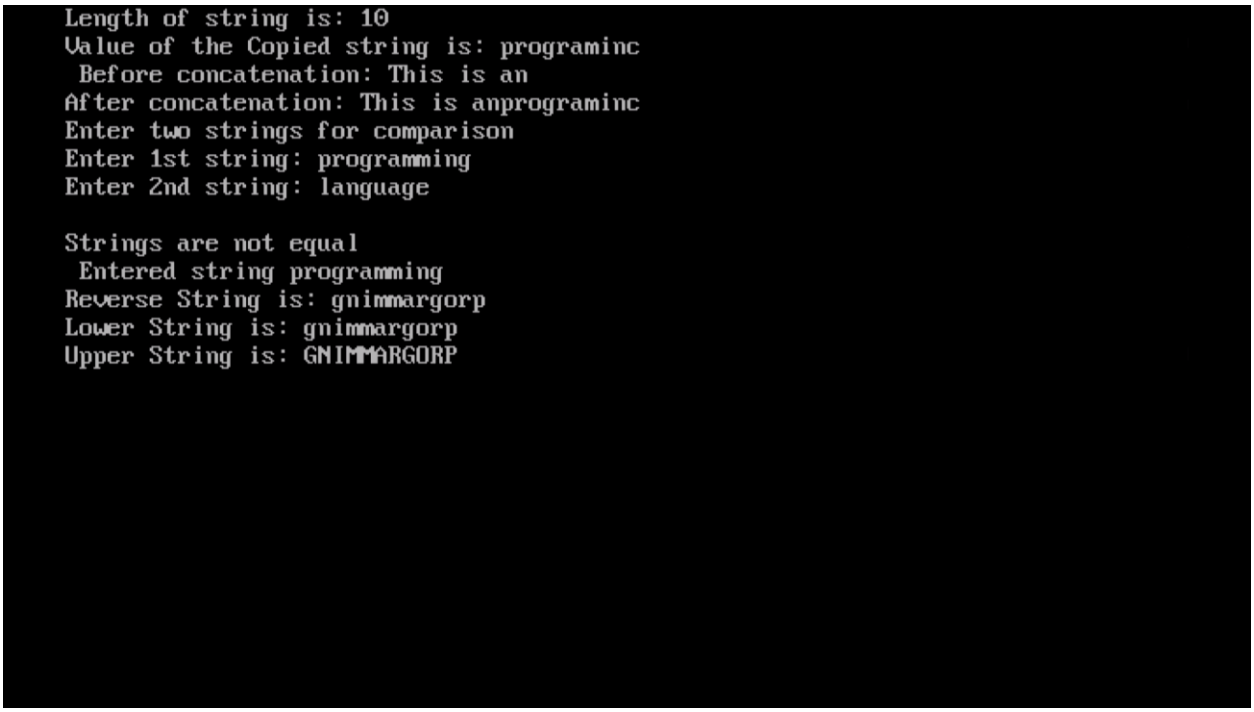
STEP 5: Stop the execution

CODING:

```
#include<stdio.h>

#include<conio.h>

#include <string.h>

int main()

{

  char str[20]={'p', 'r', 'o', 'g', 'r', 'a', 'm', 'i', 'n', 'c', '\0'};

  char str1[20], dest[50] = "This is an";

  clrscr( );

  printf("Length of the string is: %zu",strlen(str));

  strcpy(str1,str);

  printf("Value of Copied string is: %s",str1);

  printf(" Before concatenation: %s\n", dest);

  strcat(dest, str);

  printf("After concatenation: %s", dest);

  printf("Enter two strings for comparison\n");

  printf("Enter 1st string: ");

  scanf("%s",str);//reads string from console

  printf("Enter 2nd string: ");

  scanf("%s",str1);

  if(strcmp(str,str1)==0)
```

```
    printf("Strings are equal");
 else
    printf("Strings are not equal");
printf("\n Entered string %s",str);
printf("\nReverse String is: %s",strrev(str));
printf("\nLower String is: %s",strlwr(str));
printf("\nUpper String is: %s",strupr(str));
getch();
return 0;
}
```

OUTPUT SCREENSHOT:



RESULT:

Thus the program using Strings are executed successfully and the results are verified.

Ex.No.:6	USING FUNCTIONS
DATE:	

AIM:

To develop a C Program to print Fibonacci series using functions

ALGORITHM:

STEP 1 : Start the Program

STEP 2 : Declare a function to print Fibonacci series

STEP 3: Get the input value for the variable n

STEP 4: Call the fibo(n), passing n as argument

STEP 5: Within fibo(), print the Fibonacci series by assigning the a=0,b=1.

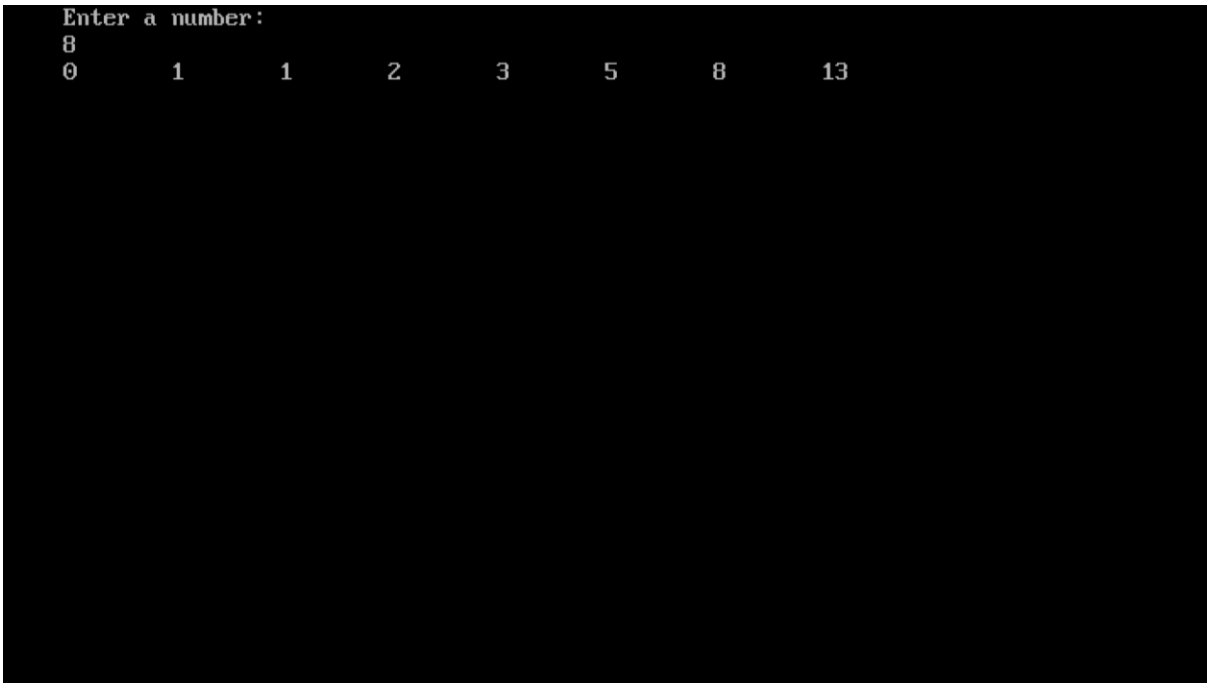
STEP 6: Print the Fibonacci series by adding the previous 2 values

STEP 7: Stop the execution

CODING:

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

OUTPUT SCREENSHOT:



RESULT:

Thus the above program using functions in C has been developed and executed successfully.

Ex.No.:7	RECURSIVE FUNCTIONS
DATE	

AIM:

To develop a C Program to print Factorial of a given number using recursion

ALGORITHM:

STEP 1: Start the Program

STEP 2: Define a function factorial to print factorial value of a given parameter using the concept of recursion

STEP 3: Define a main() function

STEP 4: Get the input value for the variable n

STEP 5: Call the function factorial with n and assign the return value to a variable fact.

STEP 6: Print the value of factorial value

STEP 7: Stop the execution

CODING:

```
#include <stdio.h>
#include<conio.h>
int factorial(int i)
{
 if(i <= 1)
   return 1;
else
  return i * factorial(i - 1);
}
void main()
{
 int n,fact;
clrscr();
printf(“Enter the number to find factorial\n”);
scanf(“%d”,&n);
fact= factorial(n);
printf("Factorial of %d =  %d",n, fact);
getch();
}
```

OUTPUT SCREENSHOT:

RESULT:

Thus the program executed successfully and the factorial value is verified.

Ex.No.:8	USING POINTERS
DATE:	

AIM:

To develop a C Program to add the sum of two integers using Pointers

ALGORITHM:

STEP 1: Start the Program

STEP 2: Declare two integer variables num, num2 and pointer variables ptr1, ptr2

STEP 3: Assign the address of num1 and num2 to ptr1 and ptr2

STEP 4: Add the content of pointer variables ptr1 and ptr2

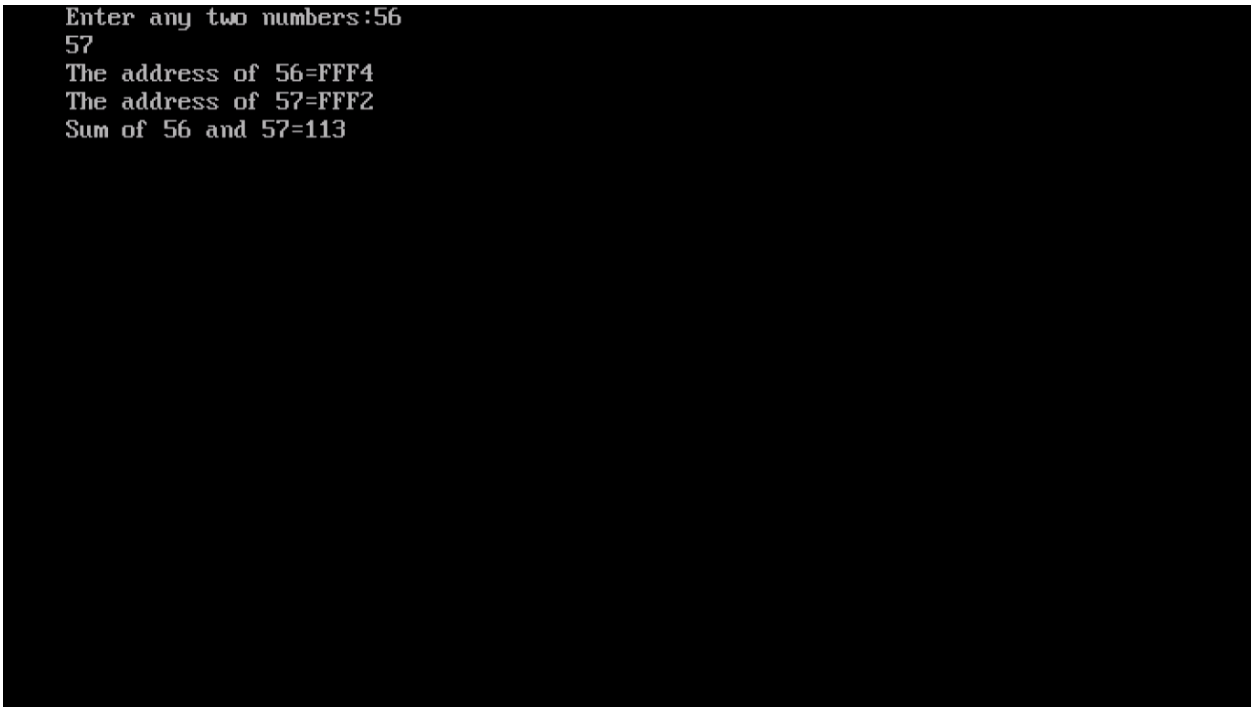
STEP 5: Print the value of variable sum

STEP 6: Stop the execution

CODING:

```
#include <stdio.h>
#include<conio.h>
int main()
{
   int num1, num2, sum;
   int *ptr1, *ptr2;
  clrscr();
   ptr1 = &num1; // ptr1 stores the address of num1
   ptr2 = &num2; // ptr2 stores the address of num2
   printf("Enter any two numbers: ");
   scanf("%d%d", ptr1, ptr2);
   sum = *ptr1 + *ptr2;
  printf("The address of %d = %p\n",num1,ptr1);
  printf("The address of %d = %p\n",num2,ptr2);
  printf("Sum of %d and %d  = %d",num1,num2, sum);
  getch();
  return 0;
}
```

OUTPUT SCREENSHOT:



RESULT:

Thus the above program using pointers are executed successfully and results are verified

Ex.No.:9	USING FILES
DATE:	

AIM:

To develop a C Program to create a new file and write student name and marks.

ALGORITHM:

STEP 1: Start the Program

STEP 2: Declare a file pointer fptr and also the required variables for student name and marks

STEP 3: Open a file student.txt in Append mode

STEP 4: Get the input values for name and marks of the students

STEP 5: write the content into the file using fprintf()

STEP 6: Close the file pointer fptr

STEP 7: Stop the execution

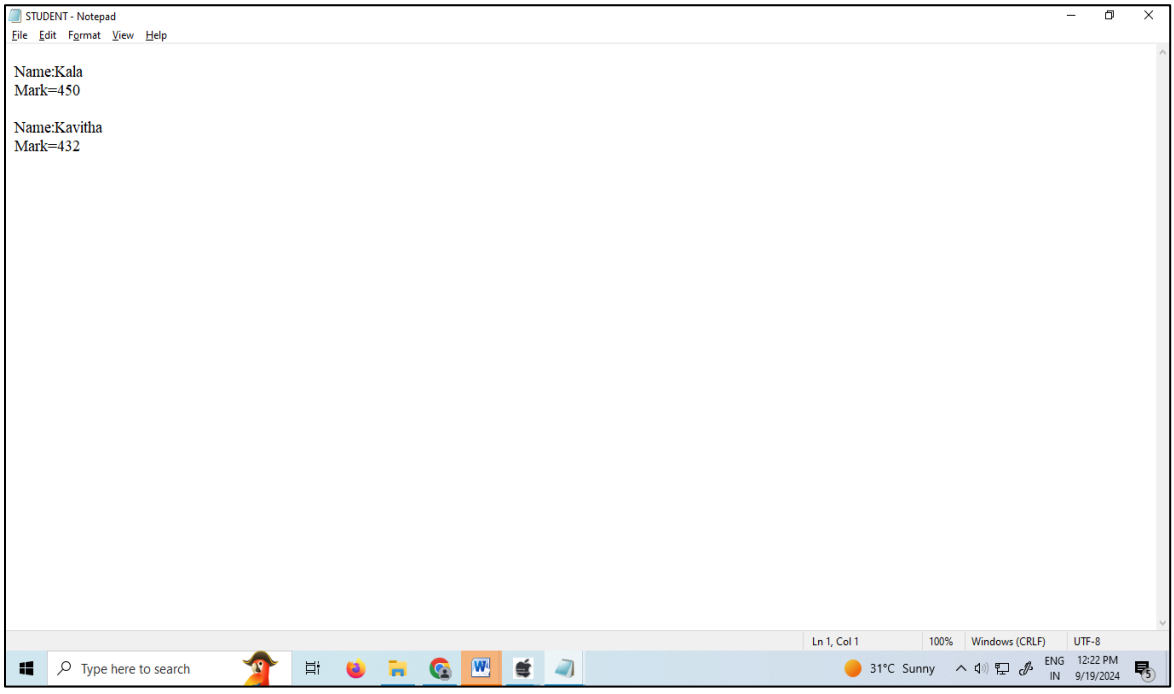
CODING:

```
#include <stdio.h>
int main()
{
  char name[50];
  int marks, i, num;
  printf("Enter number of students: ");
  scanf("%d", &num);
  FILE *fptr;
  fptr = fopen("C:\\student.txt", "a");
  if(fptr == NULL)
  {
     printf("Error!");
     exit(1);
  }
  for(i = 1; i <= num; i++)
  {
    printf("For student %d\nEnter name: ", i);
    scanf("%s", name);
    printf("Enter total marks: ");
    scanf("%d", &marks);
    fprintf(fptr,"\nName: %s \nMarks=%d \n", name, marks);
  }
  fclose(fptr);
  return 0;
}
```

OUTPUT SCREENSHOT:

```
Enter number of students:2
Enter the details for student 1
Enter name:Kala
Enter marks:450
Enter the details for student 2
Enter name:Kavitha
Enter marks:432
```

c:\turboc4\student.txt:



RESULT:

Thus the above program created a file and the contents are written into the file successfully.

Ex.No.:10	STRUCTURES AND UNIONS
DATE:	

AIM:

To develop a C Program that uses structure and union to maintain student data

ALGORITHM:

STEP 1: Start the Program

STEP 2: Declare a structure called student contains name and rollno of a student

STEP 3: Declare a union called marks within student to maintain 3 subject marks of the student

STEP 4: Get the input values for structure and union variables and calculate total

STEP 5: Display the total marks obtained by the student

STEP 6: Stop the execution

CODING:

```
#include <stdio.h>
#include<conio.h>
struct student
{
 int rollno;
char name[25];
      union marks
      {
      int m1,m2,m3;
      }m;
int tot;
}stud;
void main()
{
int x,y,z;
clrscr();
printf("Enter name and rollno\n");
scanf("%s%d",stud.name,stud.rollno);
printf("Enter mark 1\n");
scanf("%d",&stud.m.m1);
x=stud.m.m1;
printf("Enter mark 2\n");
scanf("%d",&stud.m.m2);
y=stud.m.m2;
```

```
printf(“Enter mark 3\n”);
scanf(“%d”,&stud.m.m3);
z=stud.m.m3;
stud.tot=x+y+z;
printf(“Total marks= %d”,stud.tot);
getch();
}
```

OUTPUT SCREENSHOT:

RESULT:

Thus the above program using Structures and unions executed successfully and results are verified.