Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester:2nd (Summer, Year:2022), B.Sc. in CSE (Day)

LAB REPORT NO: 01

Course Title: Structured Programming Lab

Course Code: CSE 104 / Section: DK

Lab Experiment Name: Expression in C

Student Details

Name	ID
Khondokar Saim	221902353

Lab Date : 21 / 06 / 2022

Submission Date : 07 / 08 / 2022

Course Teacher's Name : Monoshi Kumar Roy

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: …… **Signature:**.....

Comments:..... **Date:**.....

- **Tittle of the Lab experiment :** Expression in C

- **Objectives :**

We learnt many things of c programming. Such as,

- Display any input from users.
- Summation, subtraction/differentiation,multiplication,division related problem solving with two integers taken from user.
- Display math with symbol and show the result.

1. Write a C Program to Calculate Area of a Square, take length of one side as user input.

Algorithm :

Step-1 : Start

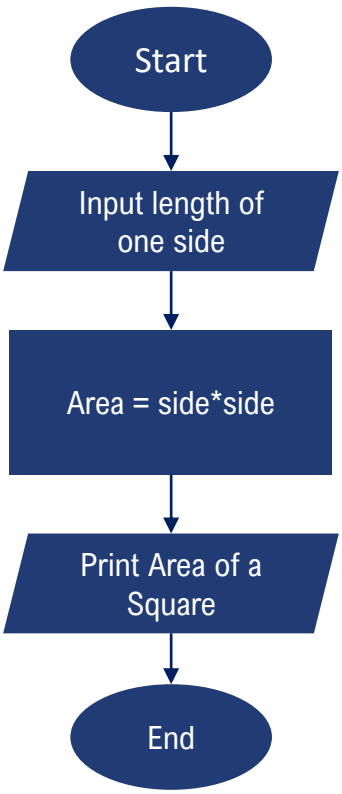
Step-2 : Input length of one side

Step-3 : Area = side*side

Step-4 : Print Area of a Square

Step-5 : End

Flowchart :



Code :

```
/* Write a C Program to Calculate Area of a Square, take length of one side as user input
*/
#include<stdio.h>
int main()
{
   int side;
   int area;

   printf("Enter the Length of one Side : ");
   scanf("%d",&side);

   area = side * side;
   printf("Area of Square : %d",area);

   return 0;
}
```

Output :

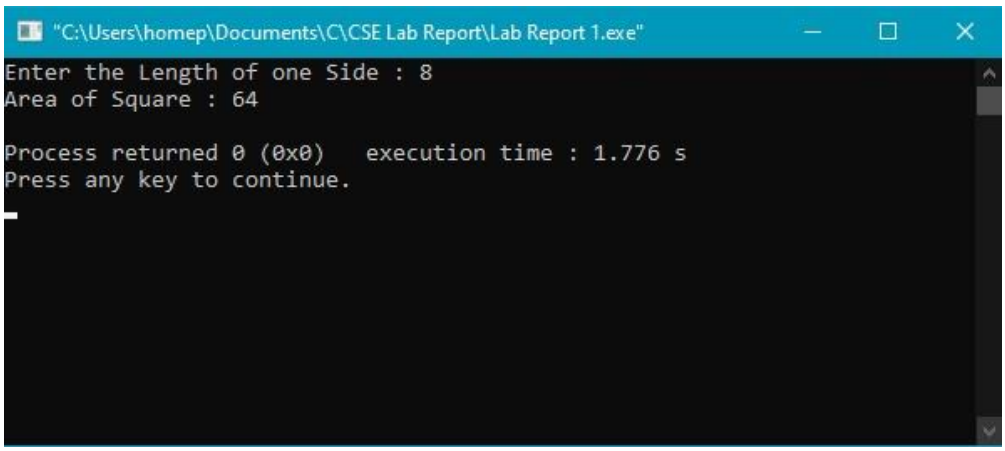


Fig 1.1: C Program to Calculate Area of a Square, take length of one side as user input.

2. Write a C program to enter temperature in °Celsius and convert it into °Fahrenheit

Algorithm :

Step-1 : Start

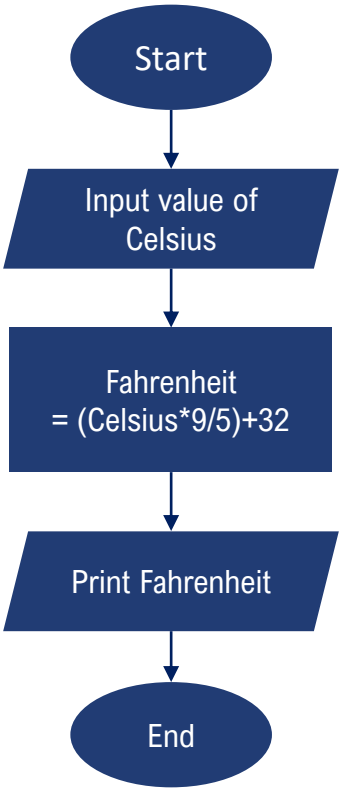
Step-2 : Input value of Celsius

Step-3 : Fahrenheit = (Celsius*9/5)+32

Step-4 : Print Fahrenheit

Step-5 : End

Flowchart :



Code :

```
#include<stdio.h>
int main()
{
    float Celsius;
    float Fahrenheit;

    printf("Enter Celsius temperature = ");
    scanf("%f",&Celsius);

    Fahrenheit = (float)(Celsius * 9 / 5) + 32; //used type casting
    printf("Fahrenheit temperature = %.2f\n",Fahrenheit);

    return 0;
}
```

Output :

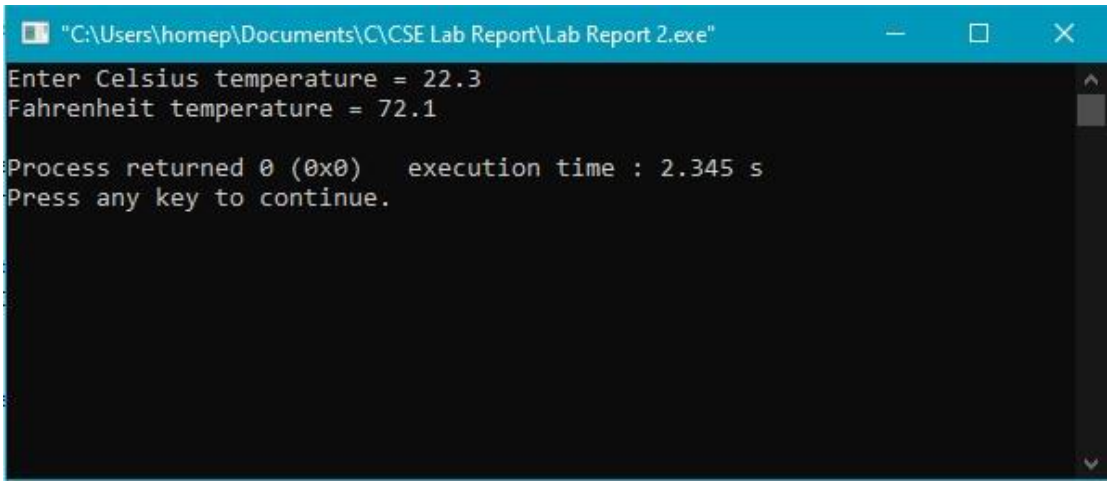


Fig 1.2: C program to enter temperature in °Celsius and convert it into °Fahrenheit

3. Write a C program to enter temperature in Fahrenheit (°F) and convert it into Celsius(°C)

Algorithm :

Step-1 : Start

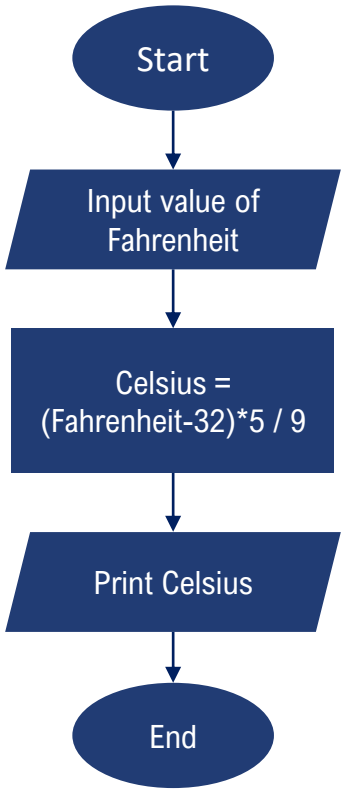
Step-2 : Input value of Fahrenheit

Step-3 : Celsius = (Fahrenheit-32)*5 / 9

Step-4 : Print Celsius

Step-5 : End

Flowchart :



Code :

```
/* Write a C program to enter temperature
in Fahrenheit(°F) and convert it into Celsius(°C)
*/
#include<stdio.h>
int main()

{
    float Fahrenheit;
    float Celsius;

    printf("Enter Fahrenheit temperature = ");
    scanf("%f",&Fahrenheit);

    Celsius = (float)(Fahrenheit - 32) * 5 / 9;
    printf("Celsius temperature = %.2f\n",Celsius);

    return 0;

}
```

Output :

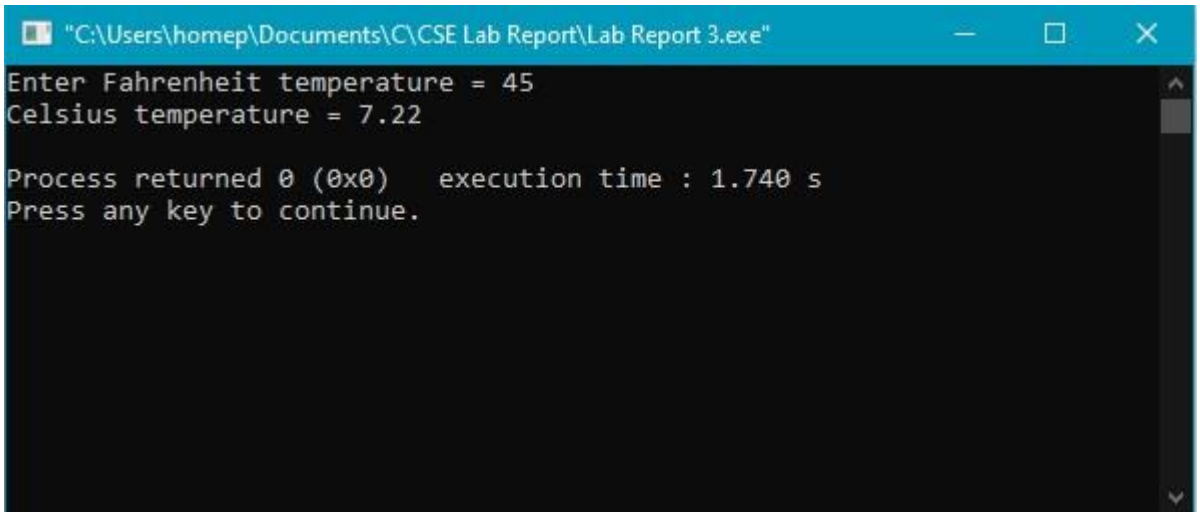


Fig 1.3 : C program to enter temperature in Fahrenheit (°F) and convert it into Celsius(°C)

4. Write a C program to enter marks of five subjects and calculate total and average marks

Algorithm :

Step-1 : Start

Step-2 : Input marks of five Subjects English,Physics, Biology,Math,ICT

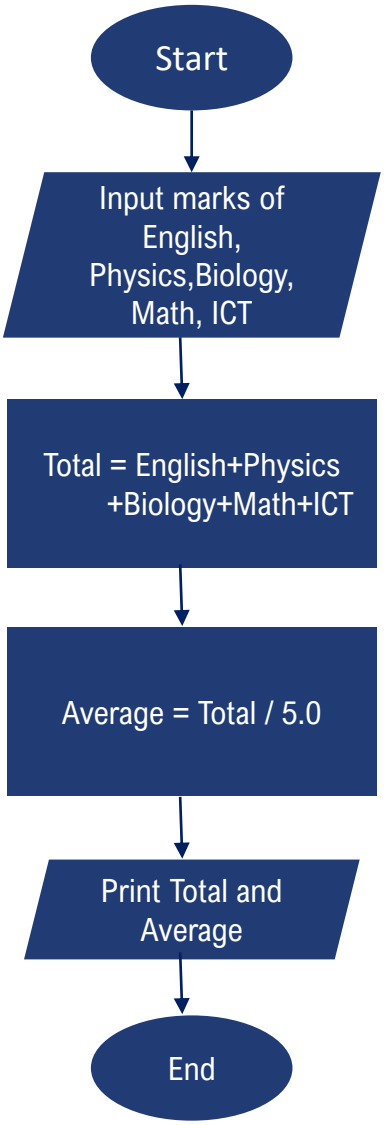
Step-3 : Total = English + Physics + Biology + Math + ICT

Step-4 : Average = Total / 5.0

Step-5 : Print Total and Average

Step-6 : End

Flowchart :



Code :

```
#include<stdio.h>
int main()
{
    float English,Physics,Biology,Math,ICT;
    float Total,Average;

    /* Input marks of all five subjects */
    printf("English = ");
    scanf("%f",&English);

    printf("Physics = ");
    scanf("%f",&Physics);

    printf("Biology = ");
    scanf("%f",&Biology);

    printf("Math = ");
    scanf("%f",&Math);

    printf("ICT = ");
    scanf("%f",&ICT);

    /* Calculate total,average*/
    Total = English + Physics + Biology + Math + ICT;
    Average = Total / 5.0;

    /* Print all results */
    printf("\nTotal marks = %.2f\n", Total);
    printf("\nAverage marks = %.2f\n", Average);

    return 0;

}
```

Output :

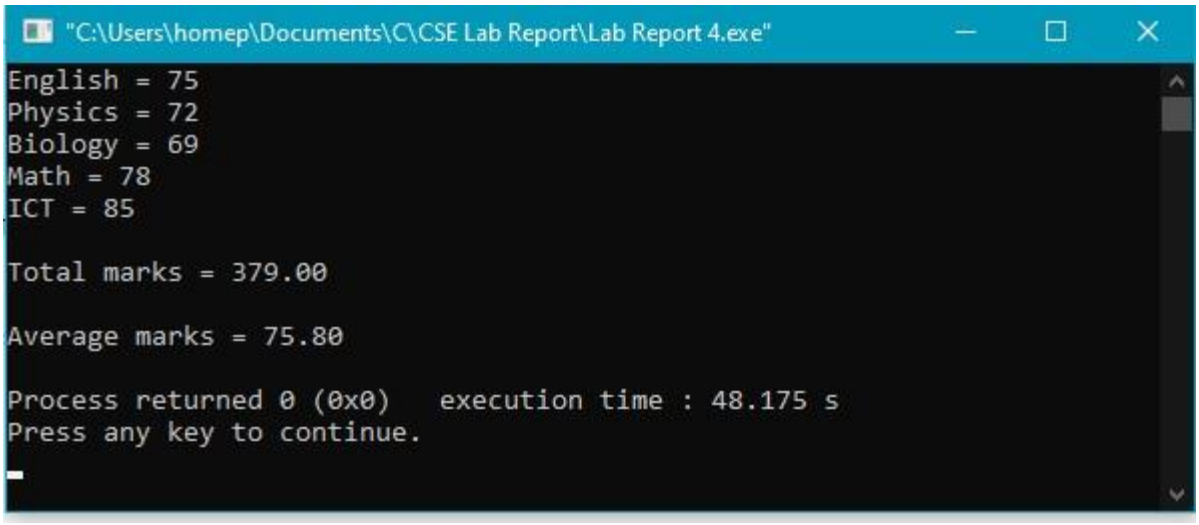


Fig 1.4 : C program to enter marks of five subjects and calculate total and average marks

• ANALYSIS AND DISCUSSION :

1. We got the exact result on output. Sometimes the result was wrong but we found the right implementation.

2. The problem of display anything in output is the easiest implementation. We solve that very easy.

3. In this assignment, we were faced some problem on last two question but by the teachers help we solve it.

4. Nothing is very/most difficult parts in my program to implement.

5. All program is easy to understand and these helped me a lot to remove my confusion about c programming.

6. I learnt- display something in program, summation, total, multiplication, division, use float numbers on program and many basic things about c programming.

Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester:2nd (Summer, Year:2022), B.Sc. in CSE (Day)

LAB REPORT NO: 02

Course Title: Structured Programming Lab

Course Code: CSE 104 / Section: DK

Lab Experiment Name: Decision making in C

Student Details

Name	ID
Khondokar Saim	221902353

Lab Date : 28 / 06 / 2022

Submission Date : 07 / 08 / 2022

Course Teacher's Name : Monoshi Kumar Roy

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: …… **Signature:.....**

Comments:..... **Date:.....**

- **Tittle of the Lab experiment :** Decision making in C

- **Objectives :**

We learnt many things from decision making in C. Such as,

- If the condition is true, then a set of statements are executed.
- If the condition is false then another set of statements is executed.
- Using a step-by-step decision-making process can help make more deliberate, thoughtful decisions by organizing relevant information and defining alternatives.

1. Write a C program to check whether a number is divisible by 5 and 11 or not.

Algorithm :

Step-1 : Start

Step-2 : Input Number

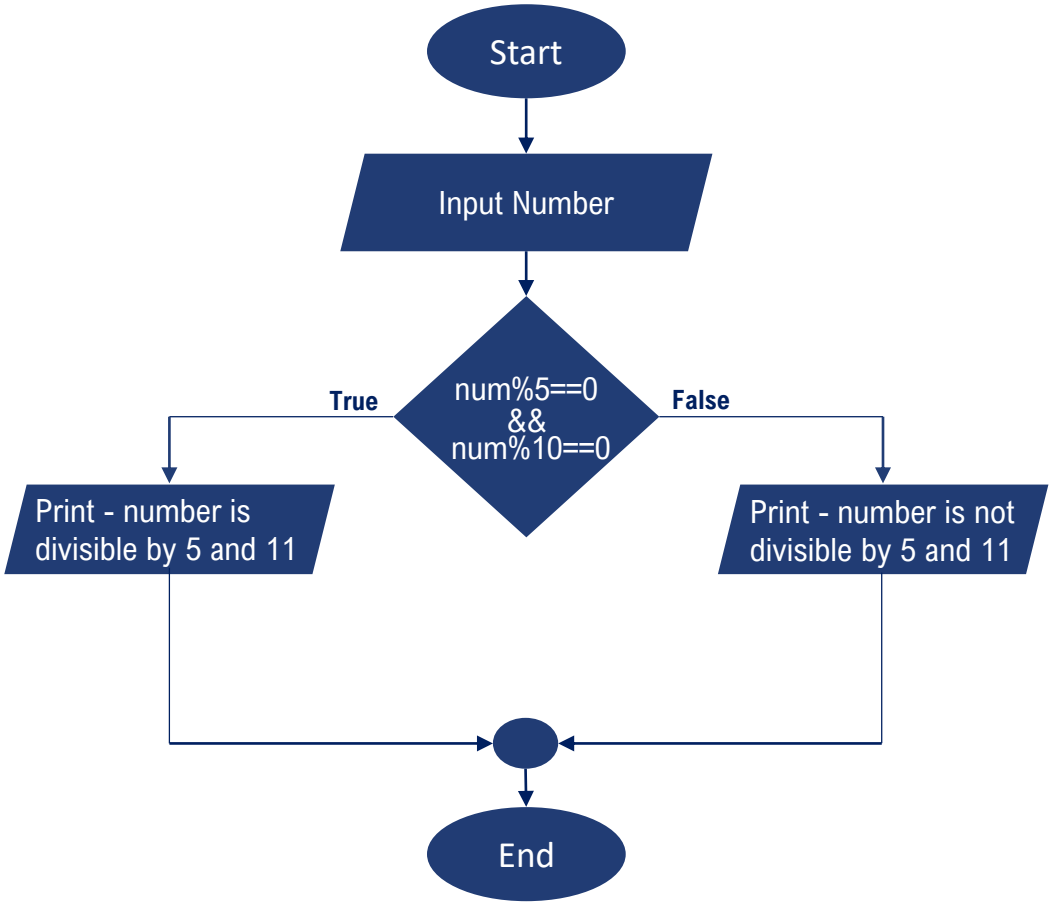
Step-3 : Is (num%5==0) and (num%11==0)

i) True, print input number is divisible by both 5 and 11

ii) False, print input number is not divisible by both 5 and 11

Step-6 : End

Flowchart :



Code :

```
// C Program To Check Whether a Number is Divisible by 5 and 11.
#include <stdio.h>
int main(){
    int num;
    printf("Enter any number: ");
    scanf("%d", &num);

    if ((num % 5 == 0) && (num % 11 == 0)) {
        printf("\n%d is divisible by both 5 and 11.\n", num);
    }
    else
        printf("\n%d is not divisible by both 5 and 11.\n", num);
    }
    return 0;
}
```

Output :

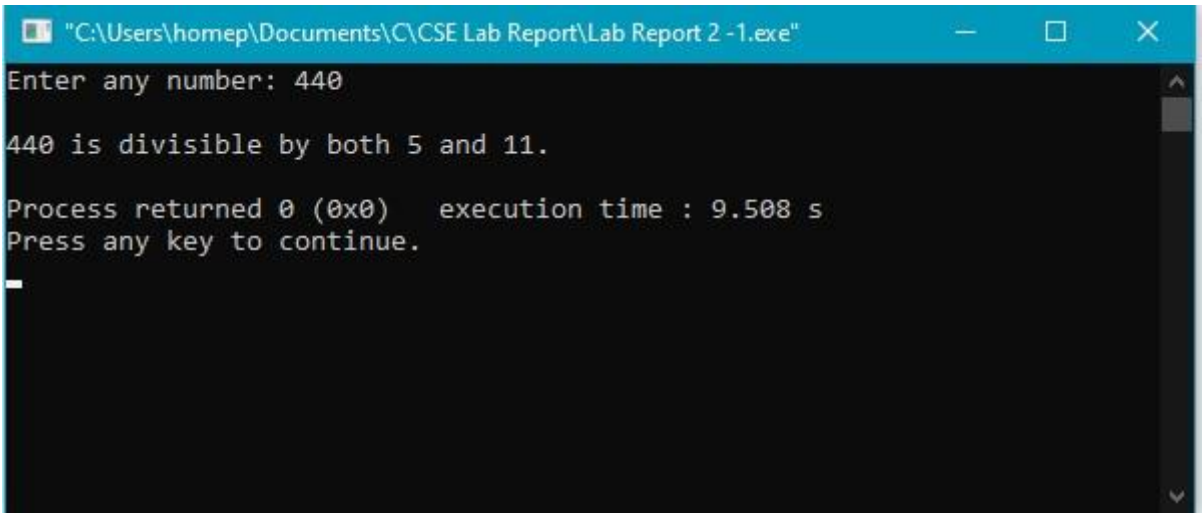


Fig 2.1 : C program to check whether a number is divisible by 5 and 11 or not.

2. Write a C program to find maximum between three numbers.

Algorithm :

Step-1 : Start

Step-2 : Input num1 , num2 , num2

Step-3 : Is num1>num2 and num1>num3

i) True, print num1.

ii) False, go to next step.

Step-4 : Is num2>num1 and num2>num3

i) True, print num2.

ii) False, go to next step.

Step-5 : Is num3>num1 and num3>num2

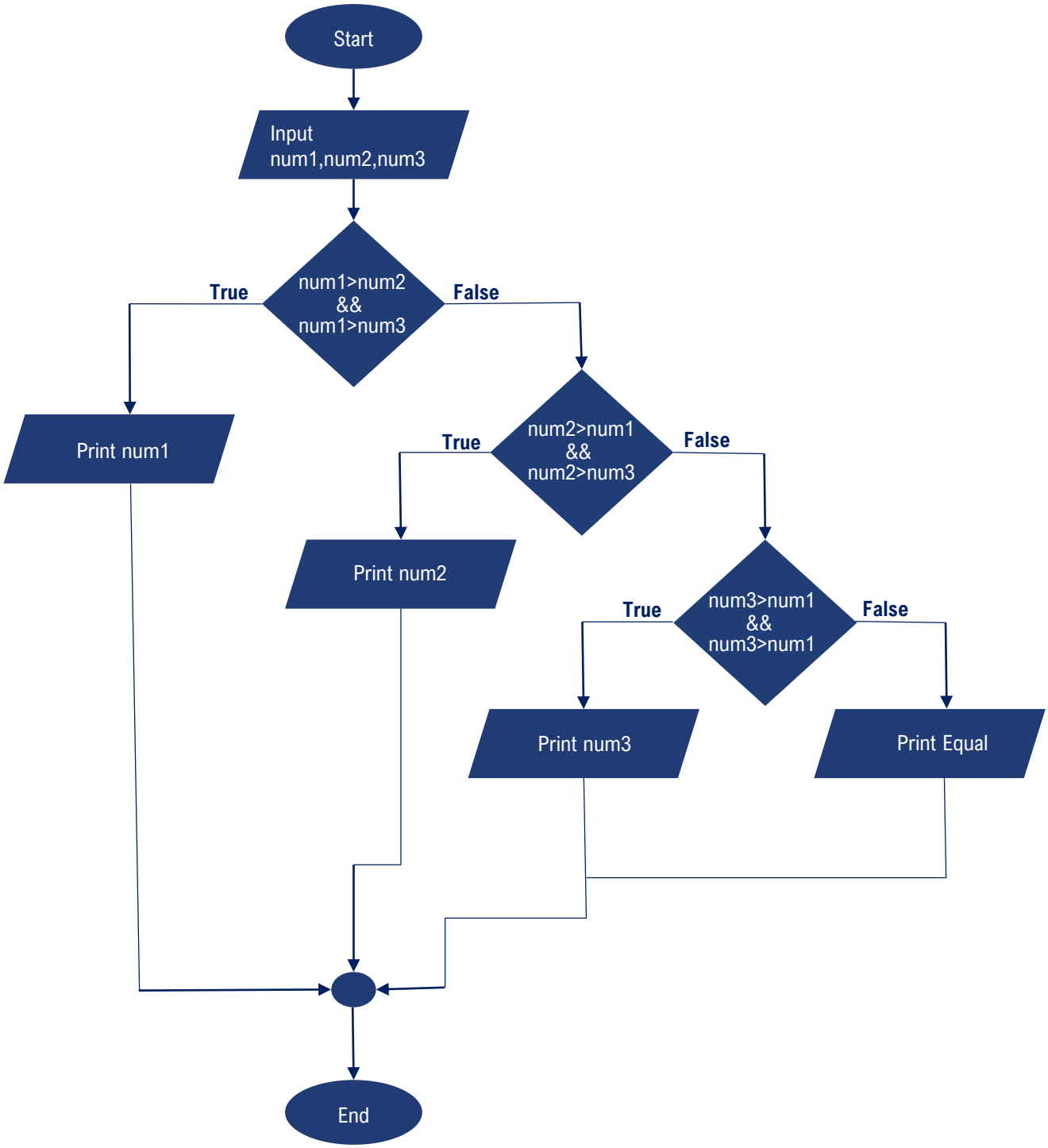
i) True, print num3.

ii) False, Equal.

Step-6 : End.

Flow chart on next page

Flowchart :


Start
Input num1,num2,num3
num1>num2 && num1>num3
True
False
Print num1
num2>num1 && num2>num3
True
False
Print num2
num3>num1 && num3>num1
True
False
Print num3
Print Equal
End

Code :

```
/*Write a C program to find maximum between three numbers.
*/
#include<stdio.h>
int main()
{
    int num1;
    int num2;
    int num3;

    printf("Enter 1st number = ");
    scanf("%d",&num1);

    printf("Enter 2nd number = ");
    scanf("%d",&num2);

    printf("Enter 3rd number = ");
    scanf("%d",&num3);

    if(num1>num2 && num1>num3)
        printf("\nLarge = %d\n",num1);

    else if(num2>num1 && num2>num3)
            printf("\nLarge = %d\n",num2);

    else if(num3>num1 && num3>num2)
            printf("\nLarge = %d\n",num3);

    else
        printf("\nThis number is Equal");

    return 0;
}
```

Output :

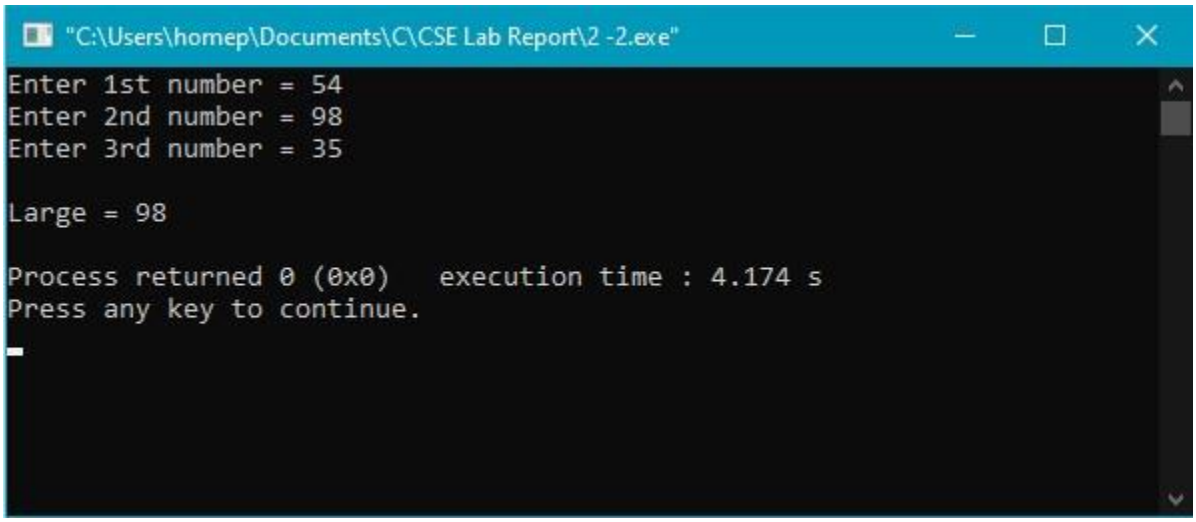


Fig 2.2 : C program to find maximum between three numbers.

3. Write a Program to take the value from the user as input any alphabet and check whether it is vowel or consonant (Using the switch statement).

Algorithm :

Step-1 : Start

Step-2 : Input Ch

Step-3 : Switch Ch

Step-4 : Is switch – a , e , i , o , u

i) True, print Vowel.

ii) break.

iii) False, go to next step.

Step-4 : Is switch – A , E , I , O , U

i) True, print Vowel.

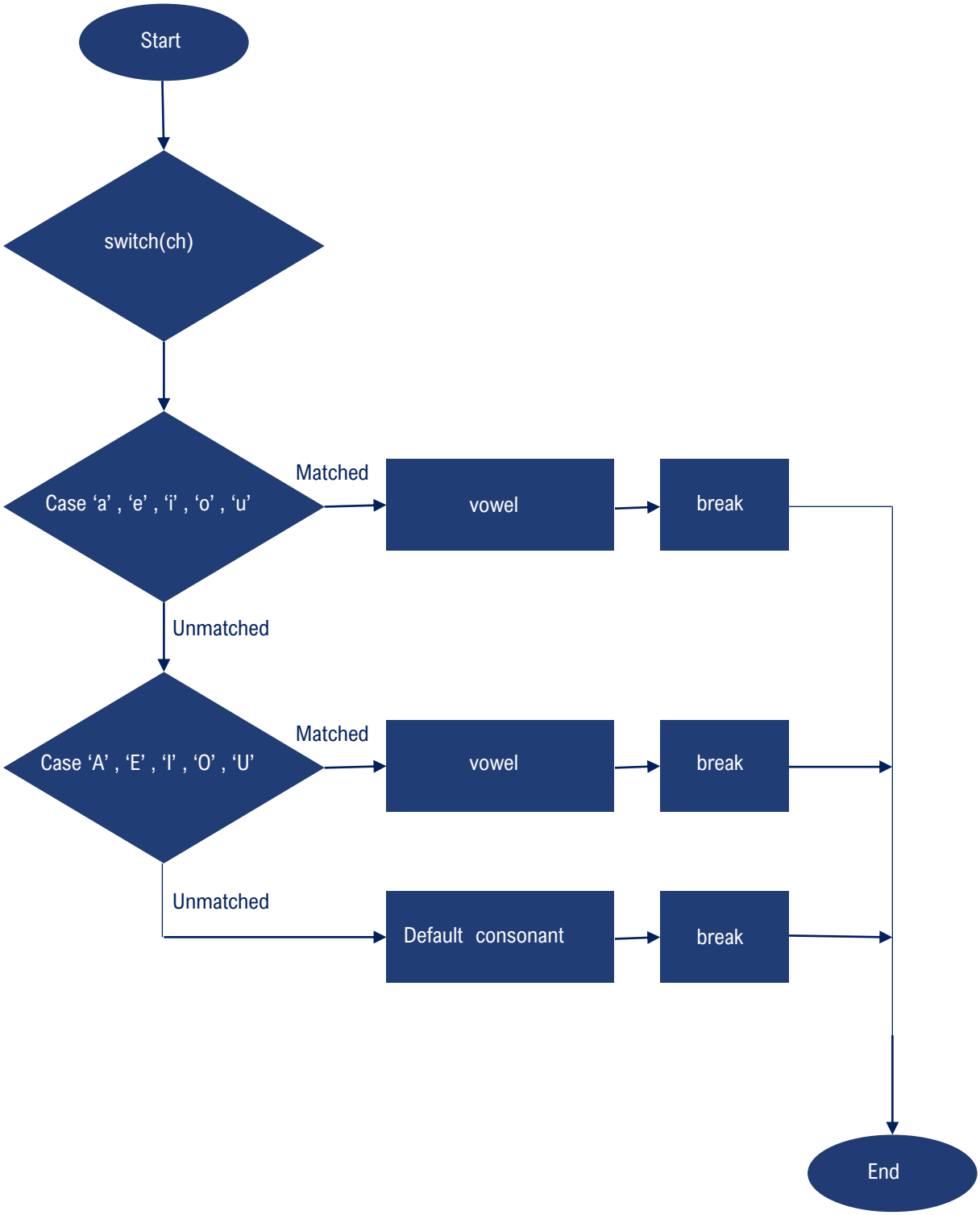
ii) break.

iii) False, default: print consonant.

Step-6 : End.

Flow chart on next page

Flowchart :

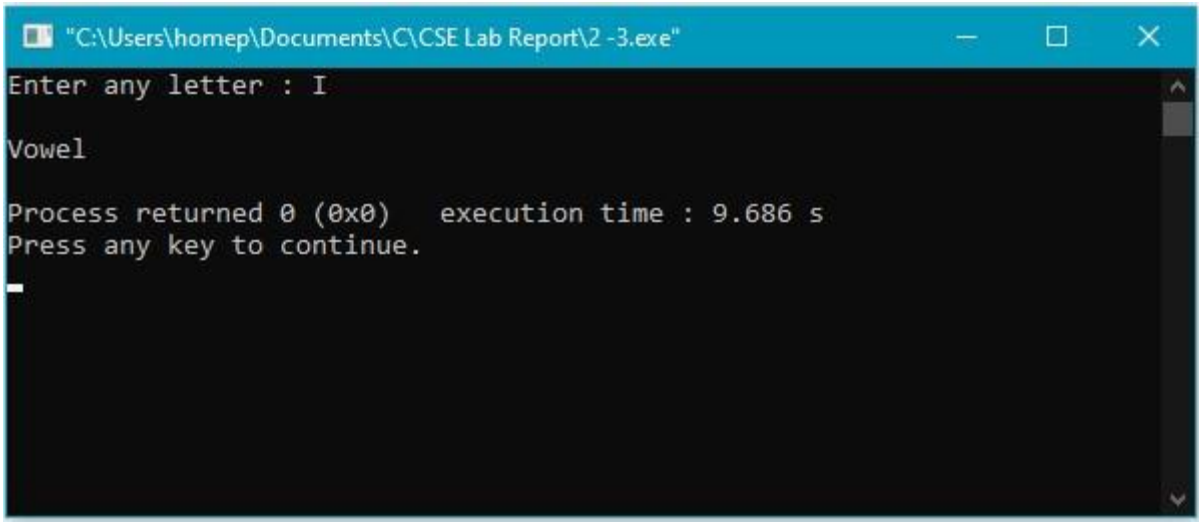

Start
switch(ch)
Case ‘a’ , ‘e’ , ‘i’ , ‘o’ , ‘u’
Matched
vowel
break
Unmatched
Case ‘A’ , ‘E’ , ‘I’ , ‘O’ , ‘U’
Matched
vowel
break
Unmatched
Default consonant
break
End

Code :

```
/*Write a Program to take the value from the user as input any alphabet and check whether it is vowel/consonant (Using the switch statement).
*/
#include<stdio.h>
int main()
{
    char ch;

    printf("Enter any letter : ");
    scanf("%c",&ch);

    switch(ch)
    {
    case 'a' :
    case 'e' :
    case 'i' :
    case 'o' :
    case 'u' :
        printf("\nVowel\n");
        break;
    case 'A' :
    case 'E' :
    case 'I' :
    case 'O' :
    case 'U' :
        printf("\nVowel\n");
        break;

    default:
         printf("\nConsonant\n");
    }
return 0;
}
```

Output :

Case - 1



Case - 2

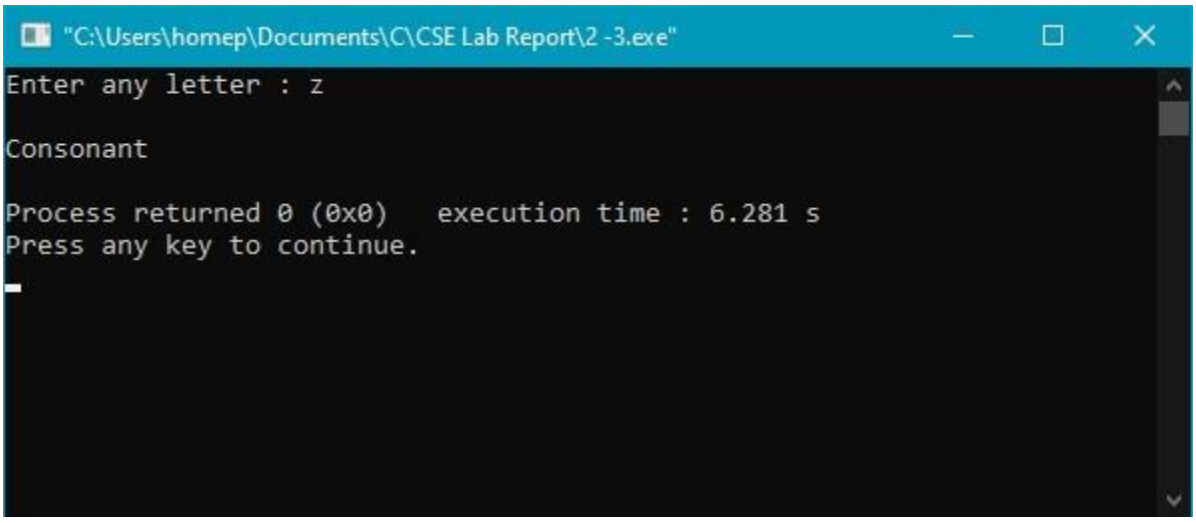


Fig 2.3 : C Program to take the value from the user as input any alphabet and check whether it is vowel or consonant (Using the switch statement).

4. Write a C program to check whether a year is leap year or not.

Algorithm :

Step-1 : Start

Step-2 : Input year

Step-3 : Is year%400==0

i) True, print Leap year.

ii) False, go to next step.

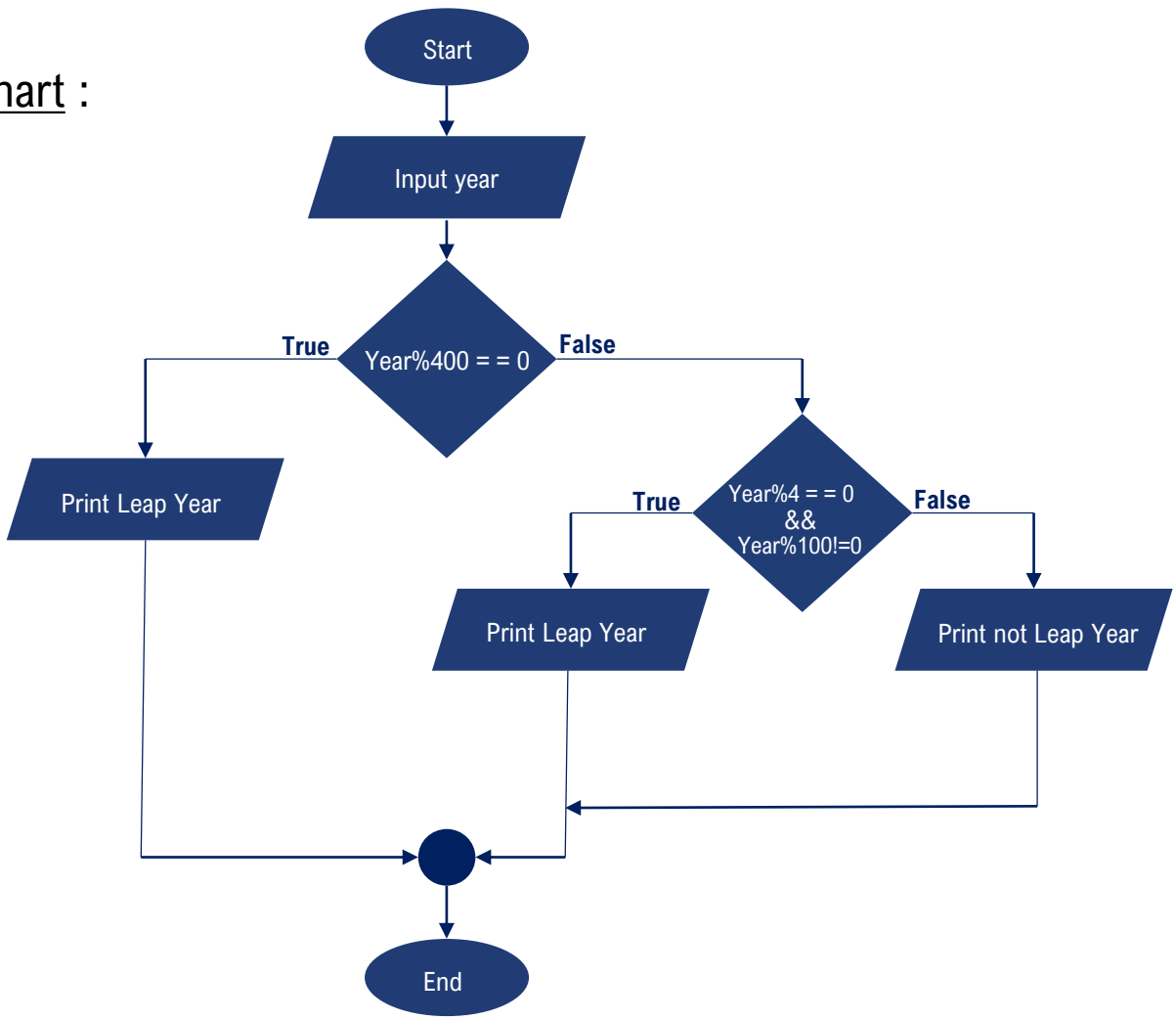
Step-4 : Is year%4==0 and year%100!=0

i) True, print Leap year.

ii) False, Not Leap year.

Step-5 : End.

Flowchart :



Code :

```
// Write a C program to check whether a year is leap year or not.
#include<stdio.h>
int main()
{
    int year;

    printf("Enter any year: ");
    scanf("%d",&year);

    if(year%400==0)
    {
        printf("\n%d is a Leap Year\n",year);
    }
    else if(year%4==0 && year%100!=0)
    {
        printf("\n%d is a Leap Year\n",year);
    }
    else
        printf("\n%d is not a Leap Year\n",year);

return 0;
}
```

Output :

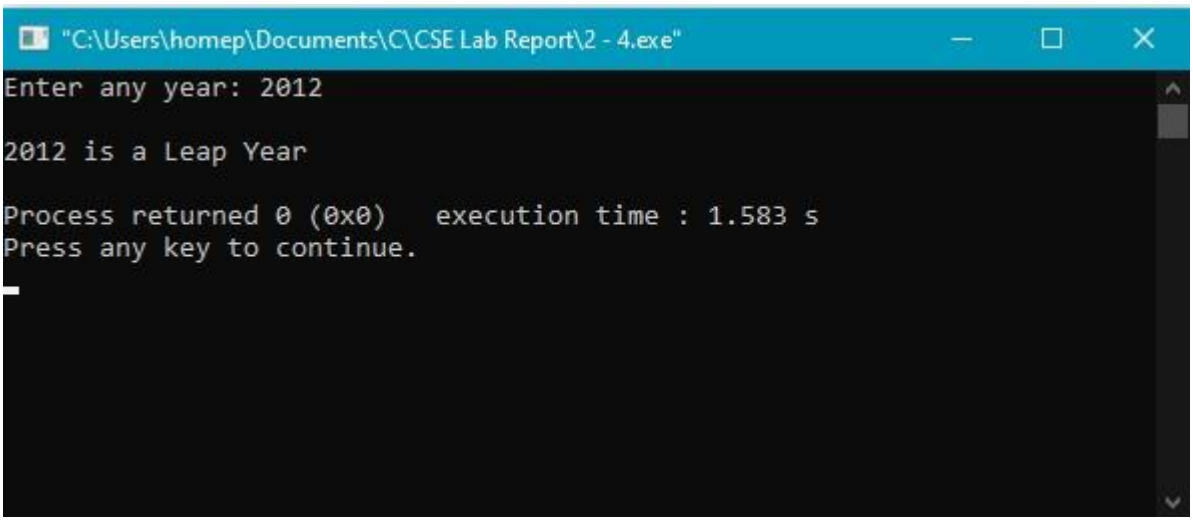


Fig 2.4 : Write a C program to check whether a year is leap year or not.

• ANALYSIS AND DISCUSSION :

1. We got the exact result on output. Sometimes the result was wrong but we found the right implementation.

2. The problem of display anything in output is the easiest implementation. We solve that very easy.

3. In this assignment, we were faced some problem on last two question but by the teachers help we solve it.

4. Nothing is very/most difficult parts in my program to implement.

5. All program is easy to understand and these helped me a lot to remove my confusion about conditional statement in c programming.

6. I learnt- Relational operator , conditional statement , control statement , Descision statement ,if-else if-else on program and many basic things about c programming.

Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester:2nd (Summer, Year:2022), B.Sc. in CSE (Day)

LAB REPORT NO: 03

Course Title: Structured Programming Lab

Course Code: CSE 104 / Section: DK

Lab Experiment Name: While and Do -While Loop Control Structure in C.

Student Details

Name	ID
Khondokar Saim	221902353

Lab Date : 05 / 07 / 2022

Submission Date : 07 / 08 / 2022

Course Teacher's Name : Monoshi Kumar Roy

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: …… **Signature:.....**

Comments:..... **Date:.....**

- **Tittle of the Lab experiment :** While and Do -While Loop Control Structure in C.

- **Objectives :**

We learnt many things from control statement and loop in C. Such as,

- Control structures alter the normal sequential flow of a statement execution. Loops allow the a block of statements to be executed repeatedly without actually writing them down numerous times.

- "C" supports mainly three types of control statements.

  - Decision making statements.
  - Loop control statements.
  - Unconditional control statements

- Loops allow the a block of statements to be executed repeatedly without actually writing them down numerous times.

There are 3 types of loop we're gonna discuss about -

- For loop.
- While loop.
- Do while loop.

1. Write a C program to find sum of first and last digit of any number.

Algorithm :

Step-1 : Start

Step-2 : Input num

Step-3 : Initialization,
Sum=0
first digit = num
last digit = num%10
Sum= first digit + last digit

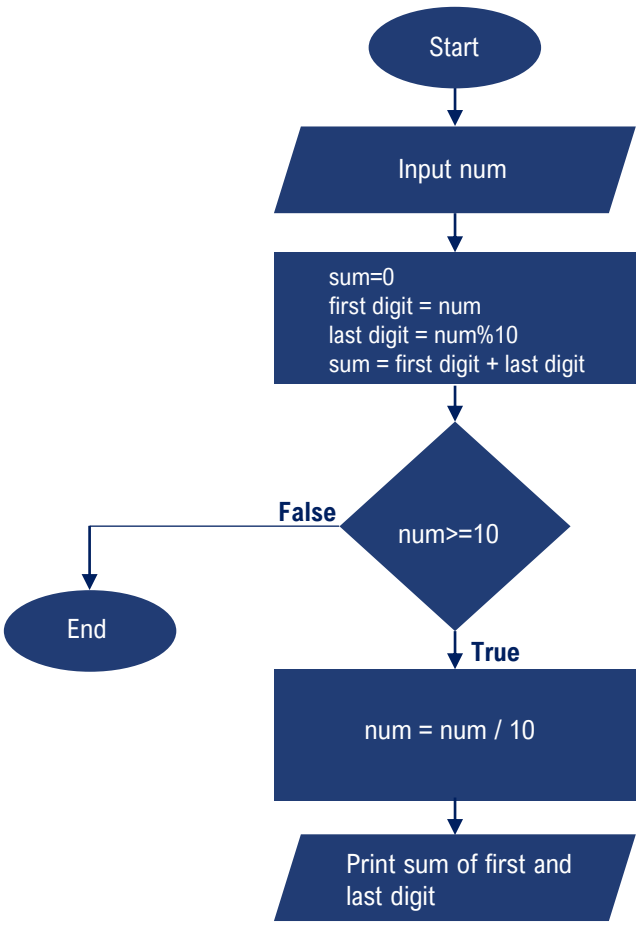
Step-4 : Is num>=10

i) True, num = num / 10.

ii) False, exit.

Step-5 : Print sum of first and last digit.

Flowchart :



Code :

```
//Write a C program to find sum of first and last digit of any number.
#include <stdio.h>
int main()
{
    int num, sum=0, firstdigit, lastdigit;

    /* Input a number from user */
    printf("Enter any number to find sum of first and last digit: ");
    scanf("%d", &num);

    //Find last digit to sum
    lastdigit = num % 10;


    /* Copy num to first digit */
    firstdigit = num;

    /* Find the first digit by dividing num by 10 until first digit is left */
    while(num >= 10)
    {
        num = num / 10;
    }
    firstdigit = num;


    /* Find sum of first and last digit*/
    sum = firstdigit + lastdigit;

    printf("\nSum of first and last digit = %d\n", sum);

    return 0;
}
```

Output :

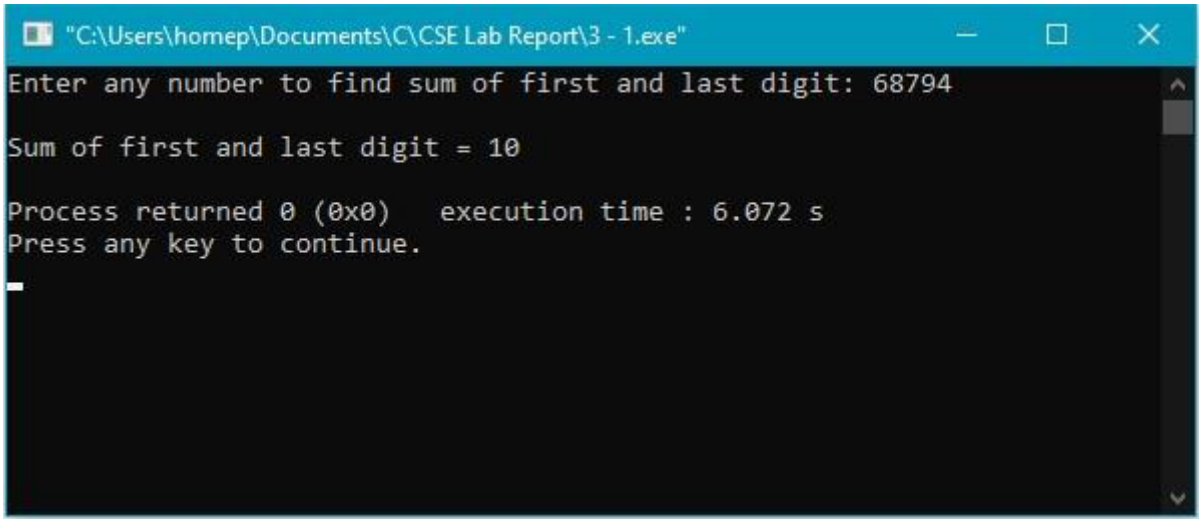


Fig 3.1 : C program to find sum of first and last digit of any number.

2. Write a C program to swap first and last digits of any number.

Algorithm :

Step-1 : Start

Step-2 : Input num

Step-3 : Initialization,
temp=num
last = temp % 10
count=(int)log10(temp)

Step-4 : Is temp>=10

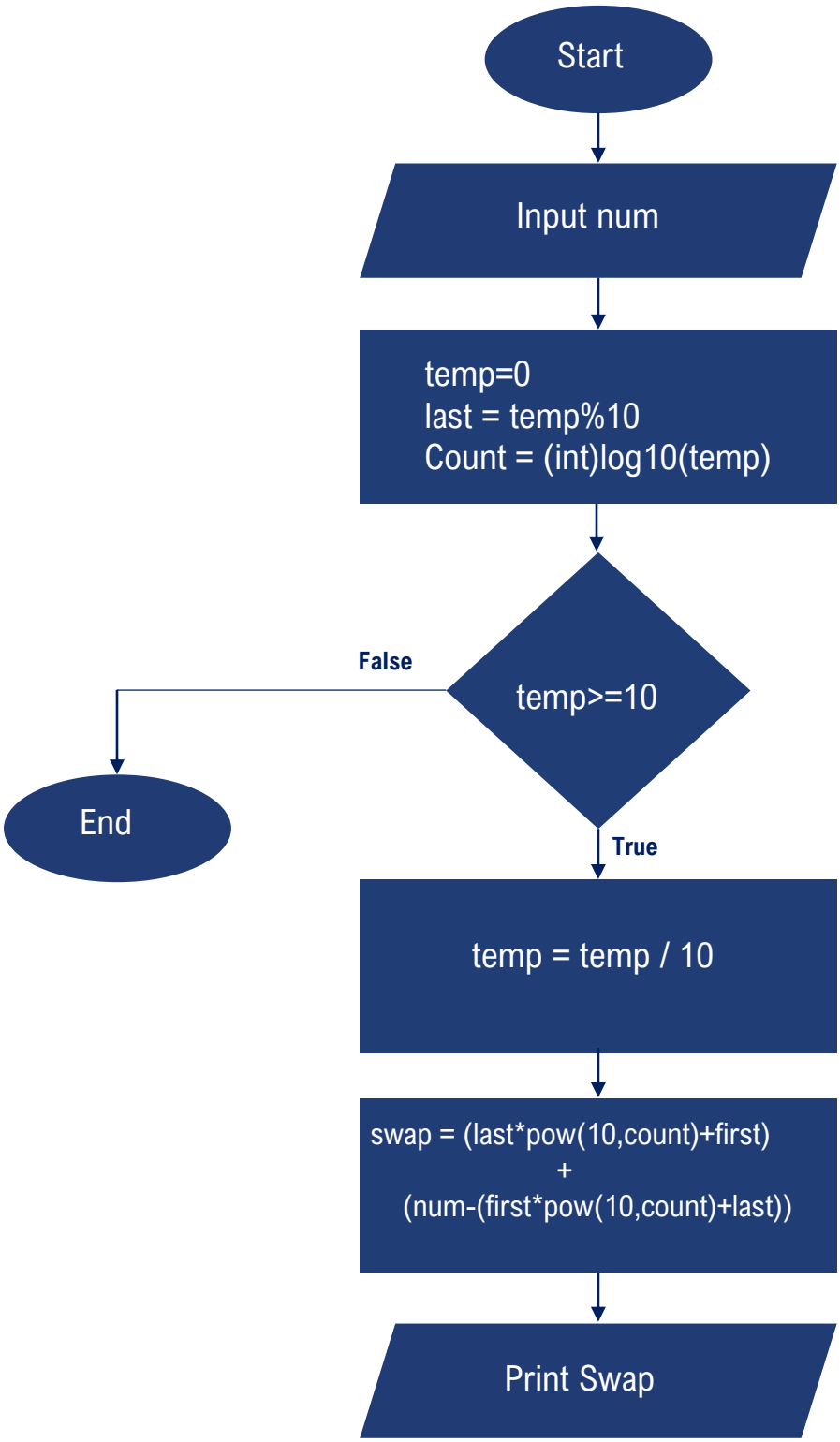
i) True, temp = temp / 10.
first = temp.
swap = (last*pow(10,count)+first)
+
(num-(first*pow(10,count)+last))

ii) False, End.

Step-5 : Print swap.

Flow chart on next page

Flowchart :


Start
Input num
temp=0
last = temp%10
Count = (int)log10(temp)
temp>=10
False
End
True
temp = temp / 10
swap = (last*pow(10,count)+first)
+
(num-(first*pow(10,count)+last))
Print Swap

Code :

```
/* Write a C program to swap first and last digits of any number.
*/
#include <stdio.h>
 int main()
 {
    int num, last, first, temp, swap, count = 0;

    printf("Enter any number: ");
    scanf("%d", &num);

    temp = num;
    last = temp % 10;
    count = (int)log10(temp);

    while(temp>=10)
    {
        temp /= 10;
    }
    first = temp;
    swap = (last * pow(10, count) + first) + (num - (first * pow(10, count) + last));

    printf("\n%d is swapped to %d\n", num, swap);
    return 0;
 }
```

Output :

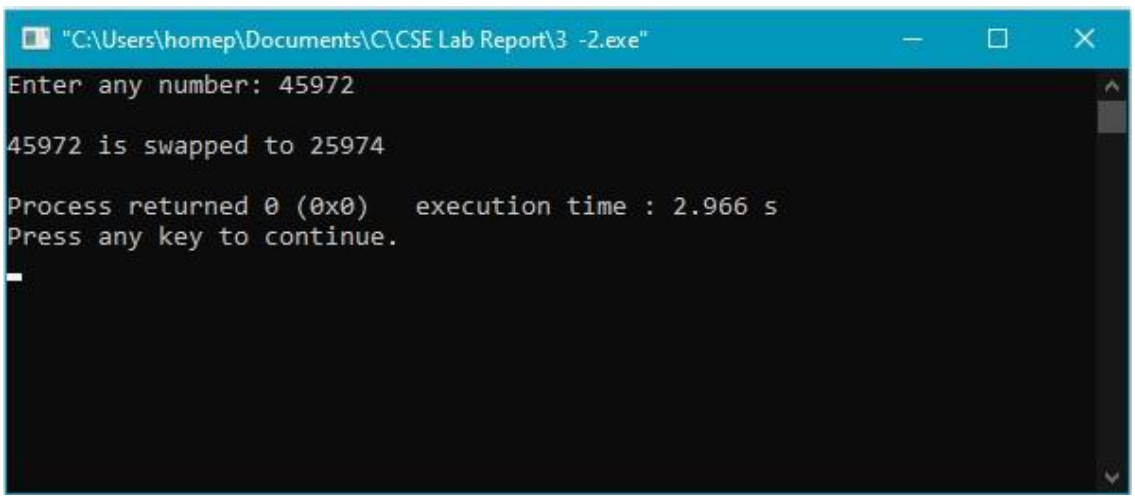


Fig 3.2 : C program to swap first and last digits of any number.

3. Write a C program to calculate product of digits of any number.

Algorithm :

Step-1 : Start

Step-2 : Input Number

Step-3 : Initialization,
Product = 1

Step-4 : Is Number>0

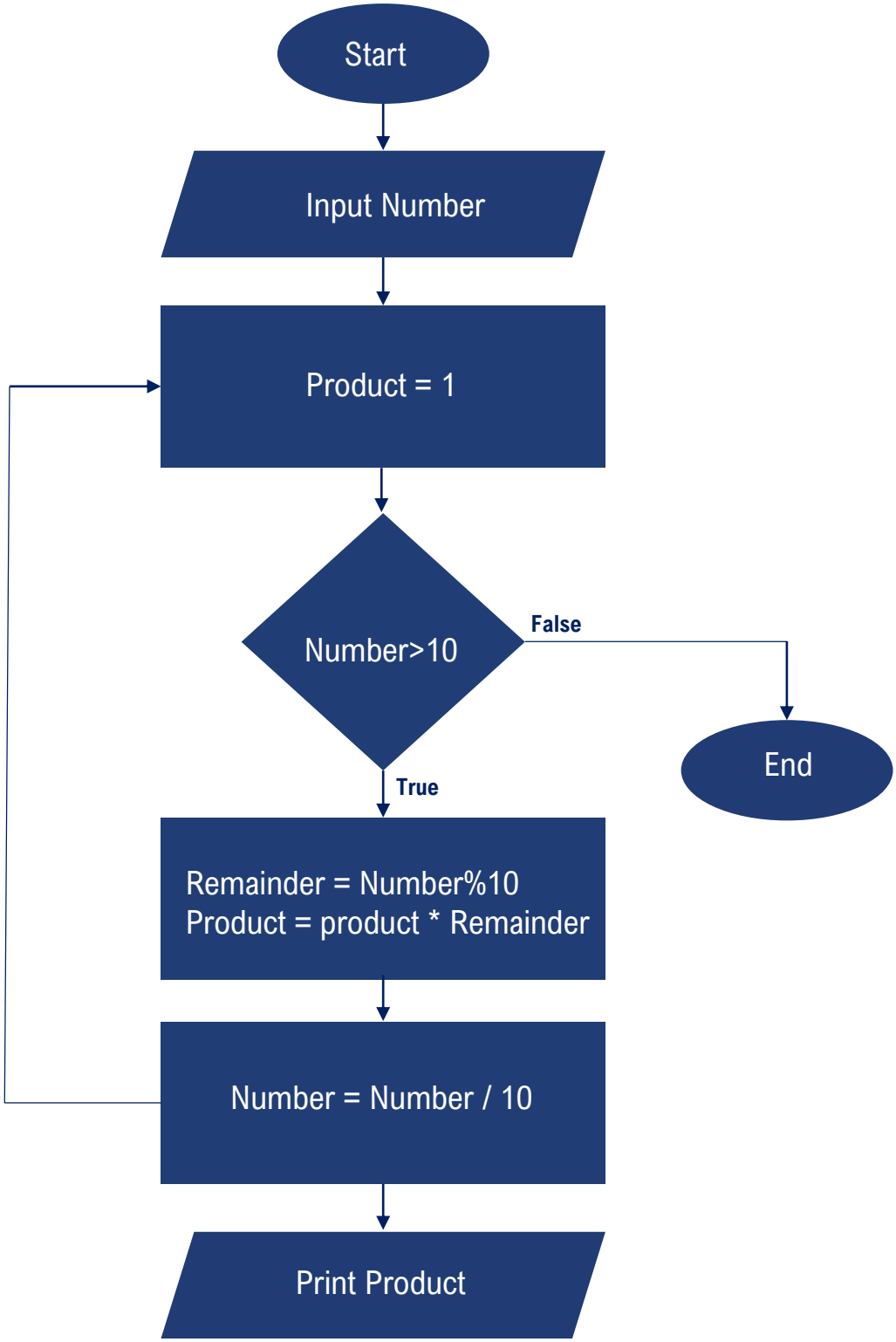
i) True, Remainder = Number % 10
Product = product * Remainder.
Go to next step.
Number = Number / 10.

ii) False, End.

Step-5 : Print Product.

Flow chart on next page

Flowchart :


Start
Input Number
Product = 1
Number>10
False
End
True
Remainder = Number%10
Product = product * Remainder
Number = Number / 10
Print Product

Code :

```
/* Write a C program to calculate product of digits of any number.
*/
#include <stdio.h>
int main()
{
	int Number, Reminder, Product;

	printf("Please Enter any Number : ");
	scanf("%d", & Number);

	for(Product = 1; Number > 0; Number = Number / 10)
	{
		Reminder = Number % 10;
		Product = Product * Reminder;
	}

	printf("\nThe Product of Digits of a Given Number = %d\n", Product);

	return 0;
}
```

Output :

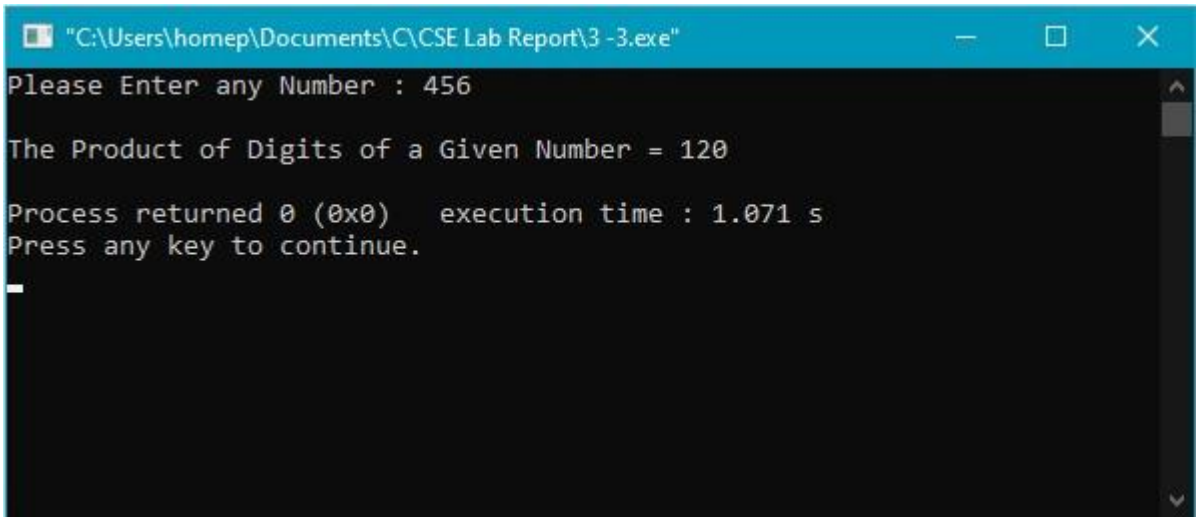


Fig 3.3 : C program to calculate product of digits of any number.

4. Write a program in C to find the sum of the series 1 +11 + 111 + 1111 + .. n

Algorithm :

Step-1 : Start

Step-2 : Input num

Step-3 : Initialization,
i = 1

Step-4 : Is i<=num

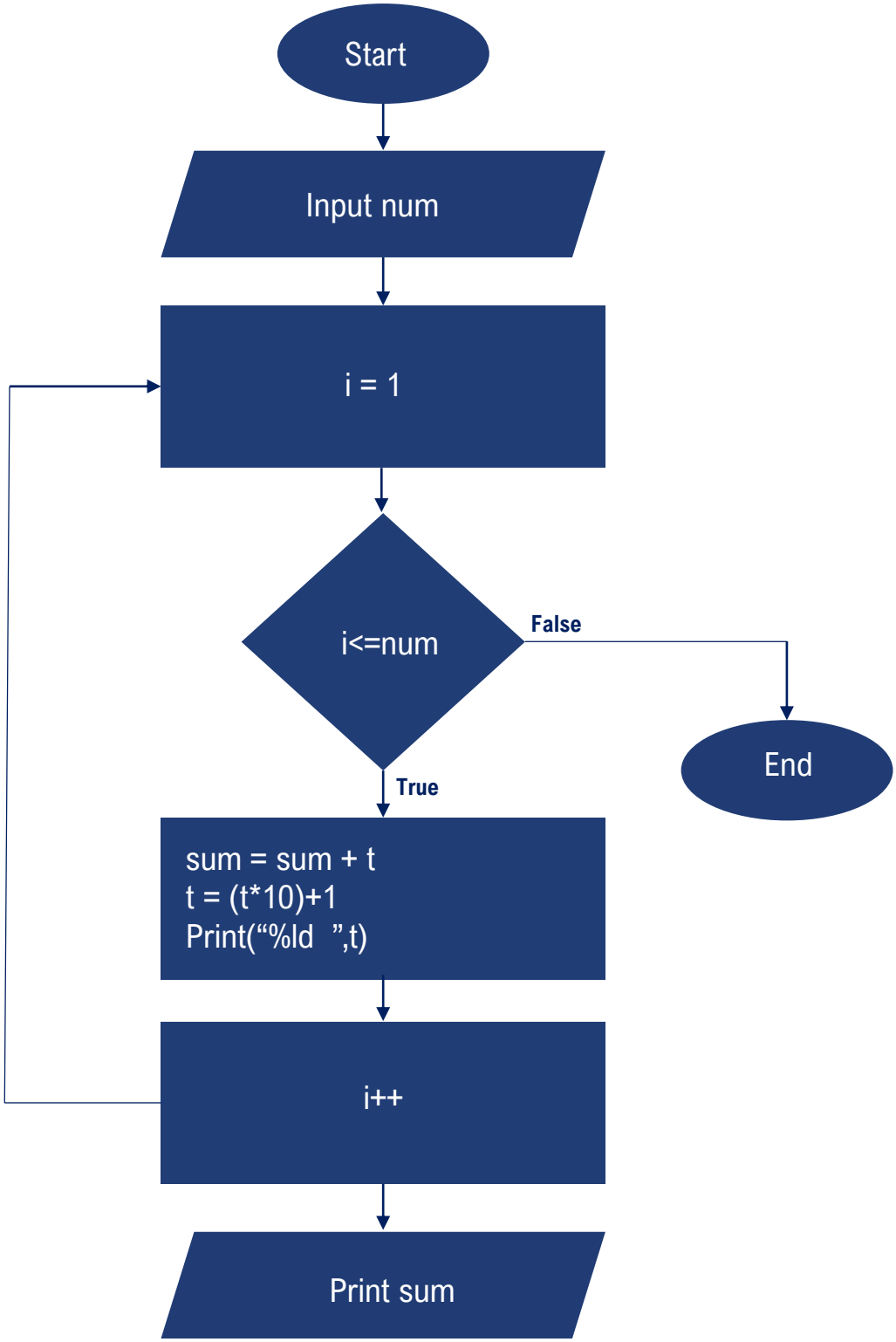
i) True, sum = sum + t
t = (t*10)+1
Print ("%ld ",t).
Go to next step.
i++

ii) False, End.

Step-5 : Print sum.

Flow chart on next page

Flowchart :


Start
Input num
i = 1
i<=num
False
End
True
sum = sum + t
t = (t*10)+1
Print("%ld ",t)
i++
Print sum

Code :

```
/* Write a program in C to find the sum of
the series 1 +11 + 111 + 1111 + .. n terms.
*/
#include <stdio.h>
void main()
{
  int num,i;
  long sum=0;
  long int t=1;

  printf("Input the number of terms : ");
  scanf("%d",&num);

  for(i=1;i<=num;i++)
  {
     printf("%ld  ",t);
      if (i<num)
      {
          printf("+ ");
      }
     sum=sum+t;
     t=(t*10)+1;
  }
  printf("\nThe Sum is : %ld\n",sum);
}
```

Output :

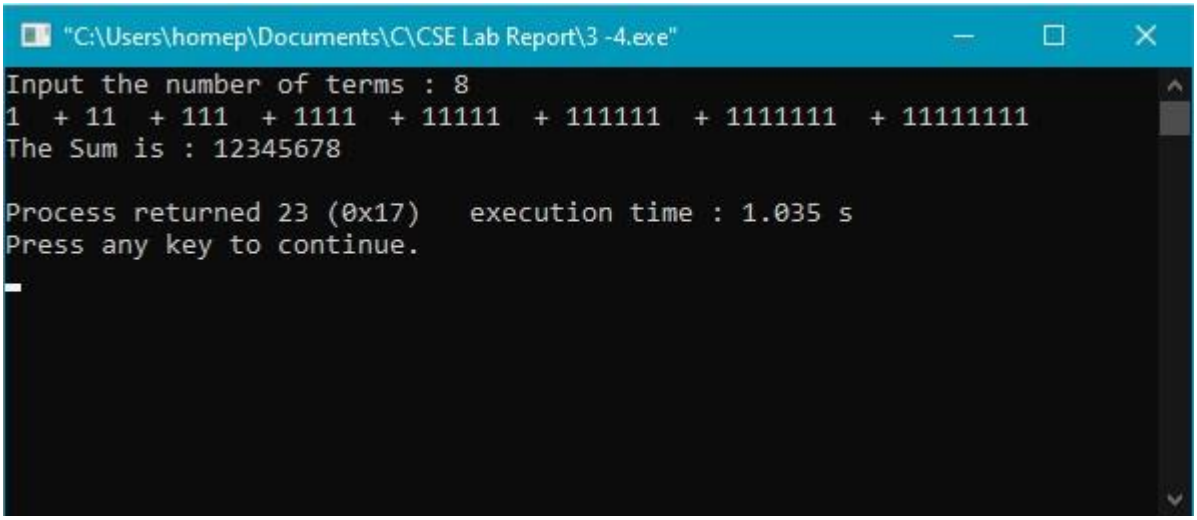


Fig 3.4 : C program in C to find the sum of the series 1 +11 + 111 + 1111 + .. n.

• ANALYSIS AND DISCUSSION :

1. We got the exact result on output. Sometimes the result was wrong but we found the right implementation.

2. The problem of display anything in output is the easiest implementation. We solve that very easy.

3. In this assignment, we were faced some problem on last two question but by the teachers help we solve it.

4. Nothing is very/most difficult parts in my program to implement.

5. All program is easy to understand and these helped me a lot to remove my confusion about control statement and loop in c programming.

6. I learnt- control statement and also loop – for, while and do while loop , series type of program and many basic things about c programming.

Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester:2nd (Summer, Year:2022), B.Sc. in CSE (Day)

LAB REPORT NO: 04

Course Title: Structured Programming Lab

Course Code: CSE 104 / Section: DK

Lab Experiment Name: Problem solving using for Loop .

Student Details

Name	ID
Khondokar Saim	221902353

Lab Date : 12 / 07 / 2022

Submission Date : 07 / 08 / 2022

Course Teacher's Name : Monoshi Kumar Roy

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: …… **Signature:.....**

Comments:..... **Date:.....**

- **Tittle of the Lab experiment :** Problem solving using for Loop.

- **Objectives :**

We learnt many things from loop in C. Such as,

- It is generally common knowledge that computers are great at performing repetitive tasks an infinite number of times, and doing so very quickly. It is also common knowledge that computers really don't do anything unless someone programs them to tell them what to do. Loop statements are the primary mechanism for telling a computer that a sequence of tasks needs to be repeated a specific number of times.

- Loops allow the a block of statements to be executed repeatedly without actually writing them down numerous times.

There are 3 types of loop we're gonna discuss about -

- For loop.
- While loop.
- Do while loop.

1. Write a C program to find whether a given number is a prime number or not.

Algorithm :

Step-1: Begin

Step-2: Display "Enter a number: "

Step-3: Read n

Step-4: Initialize c to 0

Step-5: For i = 1 to n, do

Step 5.1: If "n%i==0"

Step 5.1.1: Increment c by 1

Step 5.2: EndIf;

Step 5.3: Increment i by 1

Step-6: EndFor;

Step-7: If "c<=2"

Step 7.1: Display n " is prime number"

Step-8: Else

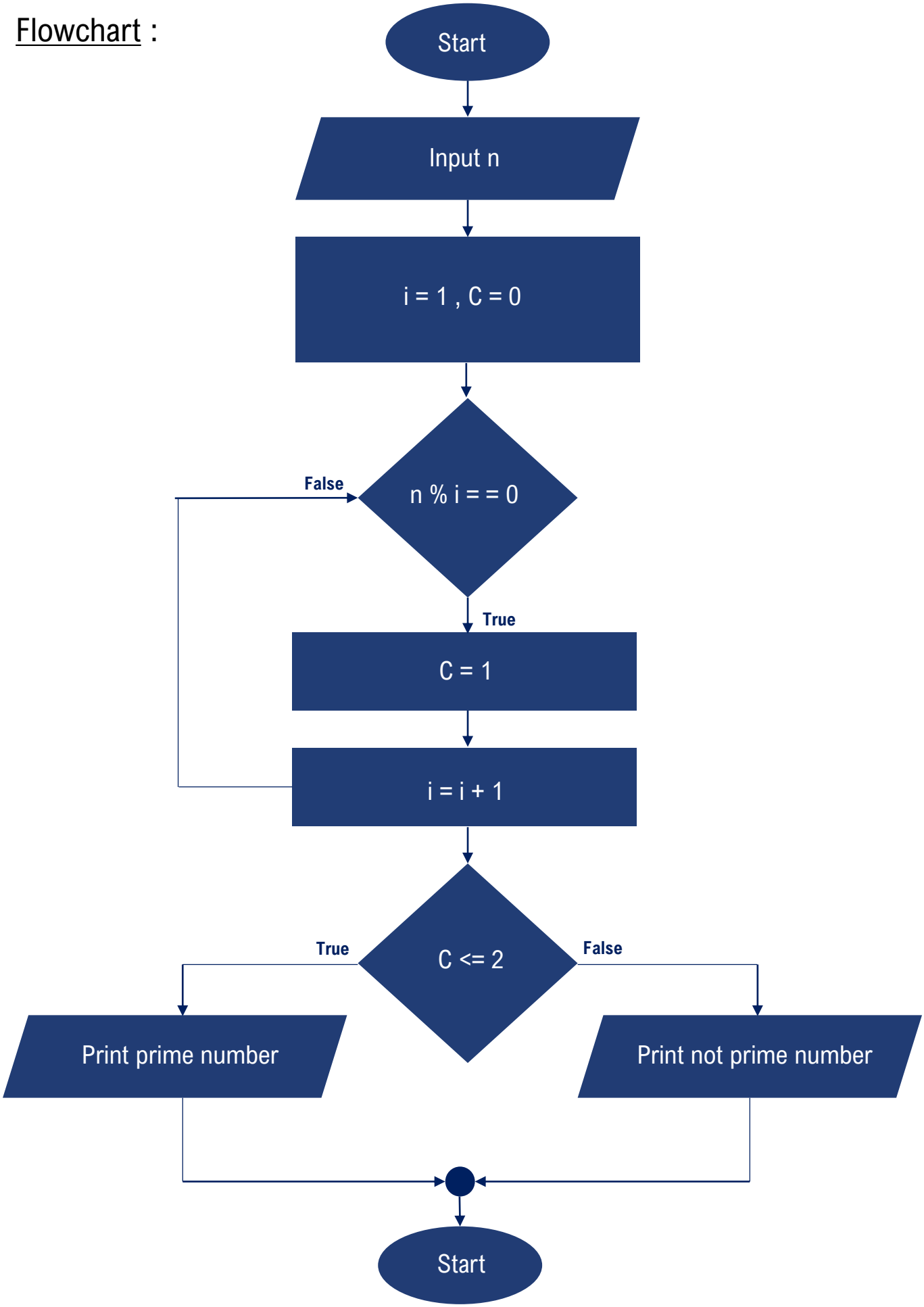
Step 8.1: Display n " is not prime number"

Step-9: EndIf;

Step 10: End

Flow chart on next page

Flowchart :

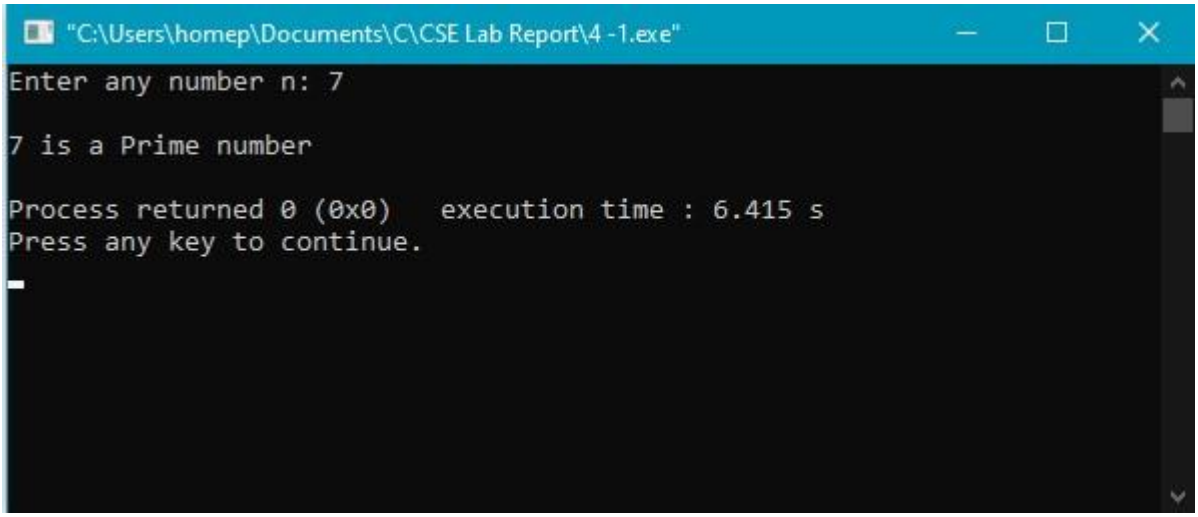

Start
Input n
i = 1 , C = 0
n % i = = 0
False
True
C = 1
i = i + 1
C <= 2
True
False
Print prime number
Print not prime number
Start

Code :

```
/* Write a C program to find whether a given number is a prime
number or not
*/
#include <stdio.h>
main()
{
  int n, i, c = 0;

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

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

  if (c <= 2)
    {
        printf("\n%d is a Prime number\n",n);
    }
  else
    {
         printf("\n%d is not a Prime number.\n",n);
    }
  return 0;
}
```

Output :

Case - 1



Case - 2

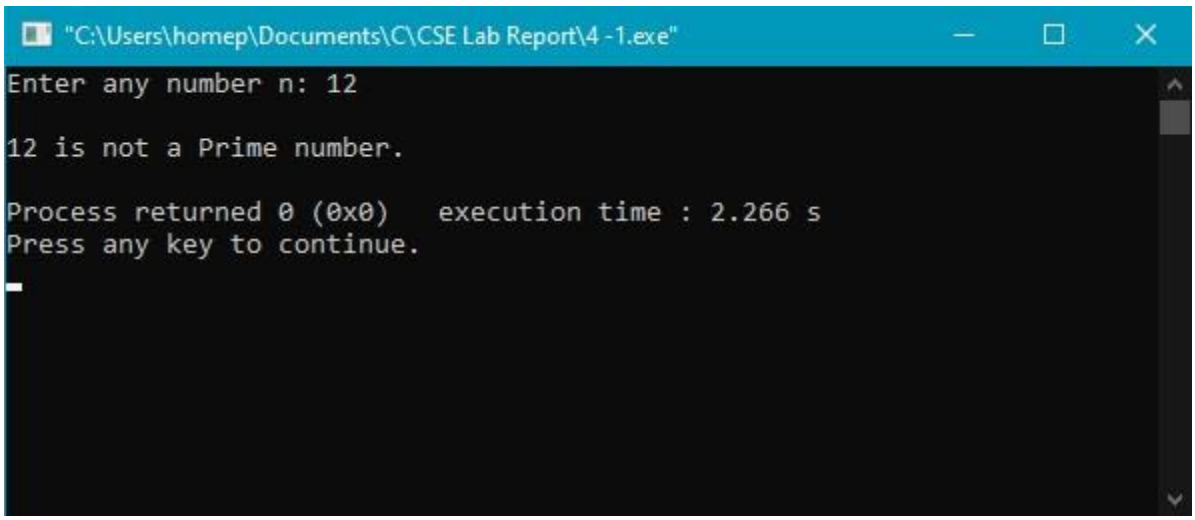


Fig 4.1 : C program to find whether a given number is a prime number or not.

2. Print Fibonacci series until a given number. For instance, if a user wants to print Fibonacci series until 1000, print all the Fibonacci number below 1000.

Algorithm :

Step-1 : Start

Step-2 : Input num

Step-3 : Initialization,
first_num = 0
second_num=1
i=0

Step-4 : Is i< num

i) True, is i<=1

1.True, next_num = i

2.False, next_num = first_num + second_num;
first_num = second_num;
second_num = next_num;

ii) False, is next_num > num

1.True, break

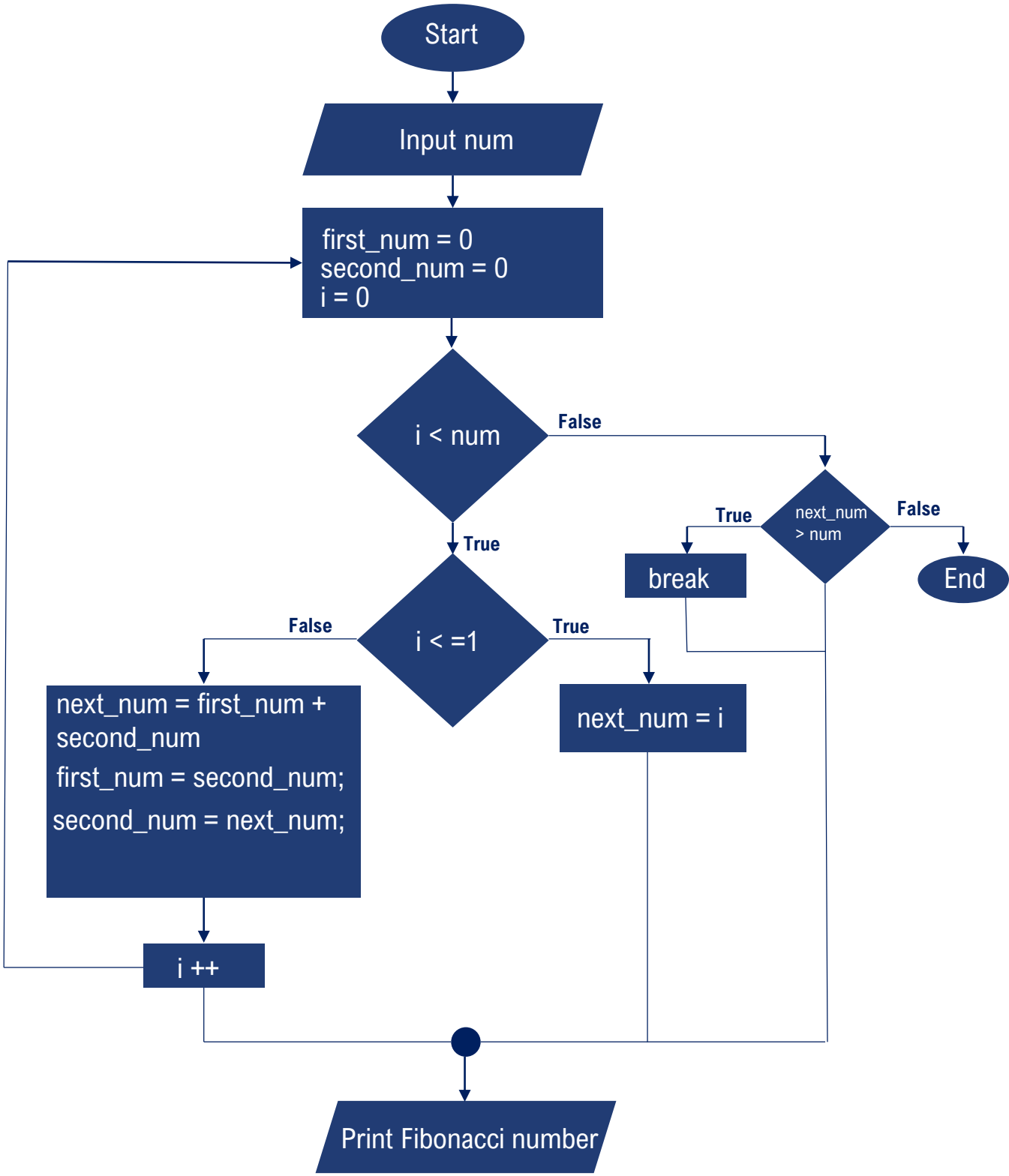
2.False, End

Step-5 : I++

Step-6 : Print Fibonacci number

Flow chart on next page

Flowchart :


Start
Input num
first_num = 0
second_num = 0
i = 0
i < num
False
True
next_num > num
True
False
break
End
i < =1
False
True
next_num = first_num + second_num
first_num = second_num;
second_num = next_num;
next_num = i
i ++
Print Fibonacci number

Code :

```
/* Print Fibonacci series until a given number.
 print all the Fibonacci number below 1000.
*/
#include<stdio.h>
int main() {
 int num;
 int first_num = 0;
 int second_num = 1;
 int next_num;
 int i;

   printf("Input number : ");
   scanf("%d",&num);

   for ( i = 0 ; i < num ; i++ ) {
      if ( i <= 1 )
         next_num = i;
      else
      {
         next_num = first_num + second_num;
         first_num = second_num;
         second_num = next_num;

         if(next_num > num)
        {
             break;
        }
      }
      printf("%d  ",next_num);
   }
getch();

}
```

Output :

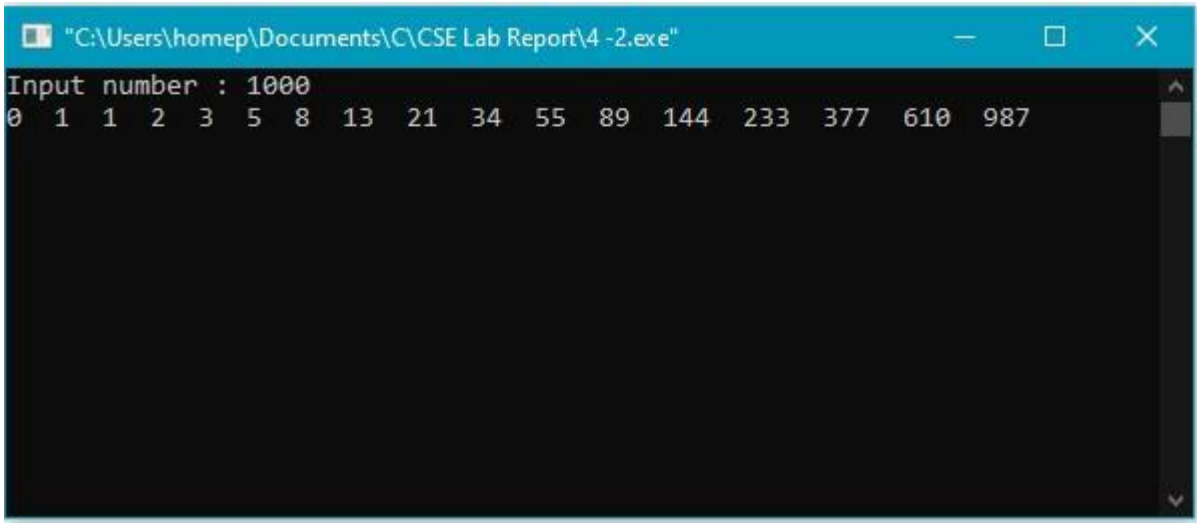


Fig 4.2 :Print Fibonacci series until a given number.

3. Display Pascal's Triangle until a given row. For instance, if a user selects row = 6, the pascal triangle for the choice would be something like below:

Algorithm :

Step-1 :Start

Step-2 : Declare variables *x, y, n, a, z, s*

Step-3 :Enter the limit

Step-4 : Initialize the value of variables, s=n , x=0, y=0 , z=s

Step-5 :Do the following operations in loop,

1. x = 0 to n
2. a= 1, x++
3. z=s to 0
4. print space
5. Z --
6. y = o to x
7. print a
8. a = a*(x-y)/(y+1)
9. y= y+1
10. go to next line

Step-6 :Repeat the process to n

Step-7 :Print the final required triangle

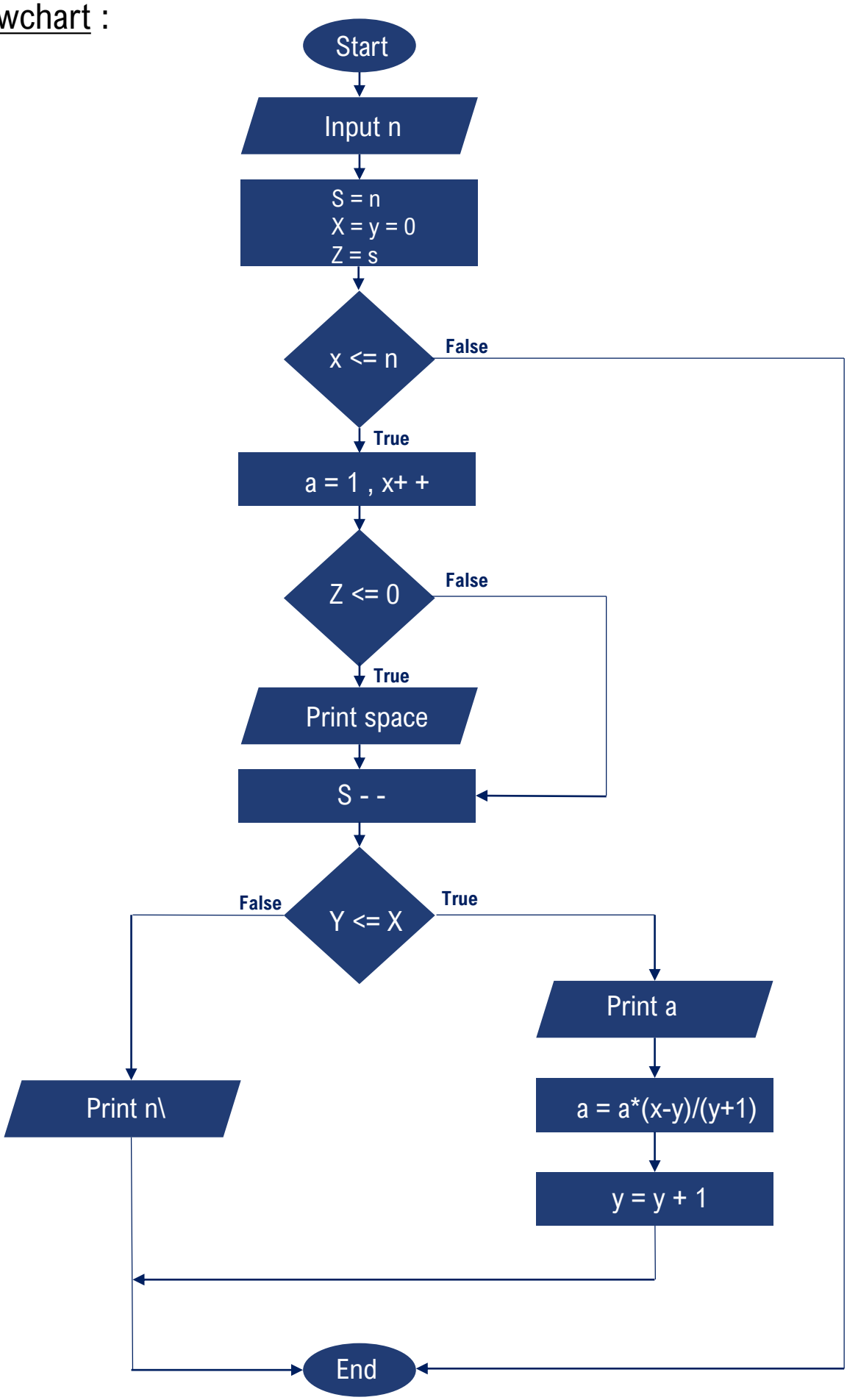
Step-8 :Stop

Code :

```
/*Display Pascal's Triangle until a given row. For instance, if a user
selects row = 6,
the pascal triangle for the choice would be something like below:
*/
#include<stdio.h>
int main()
{
    int x,y,n,a,z,s;

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

    s=n;
    for(x=0; x<=n; x++)
    {
        a=1;
        for(z=s; z>=0; z--)
            printf(" ");

        s--;
        for(y=0; y<=x; y++)
        {
            printf("%d ",a);
            a=(a*(x-y)/(y+1));
        }
        printf("\n");
    }
    getch();
}
```

Flow chart on next page

Flowchart :


Start
Input n
S = n
X = y = 0
Z = s
x <= n
False
True
a = 1 , x+ +
Z <= 0
False
True
Print space
S - -
Y <= X
False
True
Print a
Print n\
a = a*(x-y)/(y+1)
y = y + 1
End

Output :

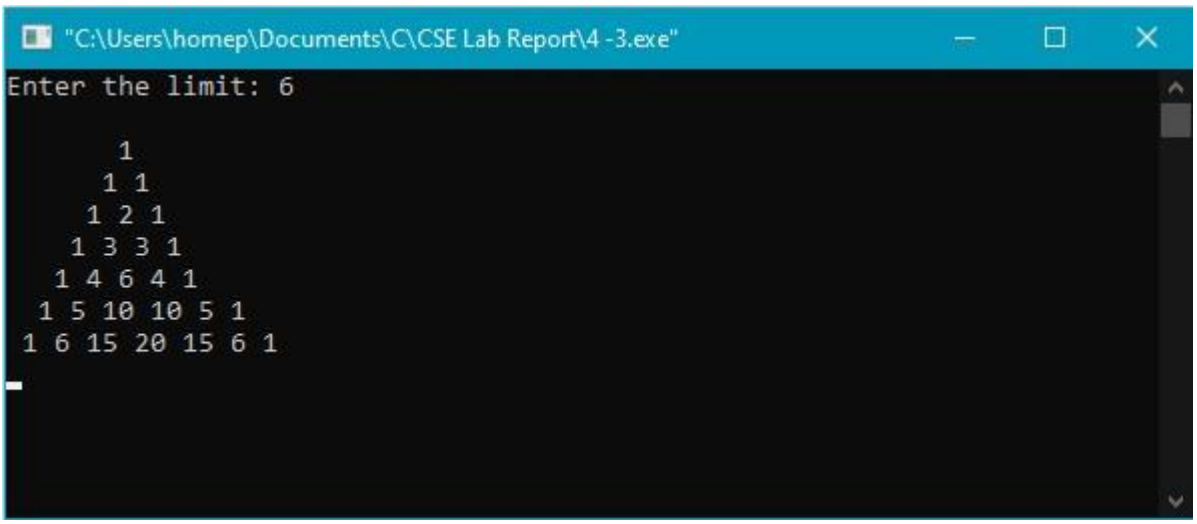


Fig 4.3 : Display Pascal's Triangle until a given row.

• ANALYSIS AND DISCUSSION :

1. We got the exact result on output. Sometimes the result was wrong but we found the right implementation.

2. The problem of display anything in output is the easiest implementation. We solve that very easy.

3. In this assignment, we were faced some problem on last two question but by the teachers help we solve it.

4. Nothing is very/most difficult parts in my program to implement.

5. All program is easy to understand and these helped me a lot to remove my confusion about loop in c programming.

6. I learnt- loop , for, while and do while loop , Pattern and series type of program and many basic things about c programming.

Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester:2nd (Summer, Year:2022), B.Sc. in CSE (Day)

LAB REPORT NO: 05

Course Title: Structured Programming Lab

Course Code: CSE 104 / Section: DK

Lab Experiment Name: Arrays .

Student Details

Name	ID
Khondokar Saim	221902353

Lab Date : **19 / 07 / 2022**

Submission Date : **07 / 08 / 2022**

Course Teacher's Name : **Monoshi Kumar Roy**

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status	
Marks: ……	Signature:.....
Comments:.....	Date:.....

- **Tittle of the Lab experiment :** Arrays.

- **Objectives :**

We learnt many things from Arrays in C. Such as,

- Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

- A specific element in an array is accessed by an index. All arrays consist of contiguous memory locations.

- The lowest address corresponds to the first element and the highest address to the last element.

- C programming language provides the following types of Arrays:

  - One-dimensional arrays .
  - Multidimensional arrays .

1. Write a C Program to Calculate mean, median and Standard Deviation.

Code :

```
/* Write a C Program to Calculate mean, median and Standard Deviation
*/
  #include <stdio.h>
  #include <stdlib.h>
  #include <math.h>
  int main() {
        float *x, mean = 0, median, sd, var;
        int n, i, j, temp;
        printf("Enter the no of entries:");
        scanf("%d", &n);
        x   = (float *)malloc(sizeof (float) * n);

        /* get n inputs from user */
        printf("Enter your inputs:\n");
        for (i = 0; i < n; i++)
                scanf("%f", &x[i]);

        /* calculate the mean */
        for (i = 0; i < n; i++)
                mean = mean + x[i];
        mean = mean / n;

        /* calculate the variance*/
        for (i = 0; i < n; i++)
                var = var + pow((x[i] - mean) , 2);

        var = var / n;

         /*square root of variance is SD */
        sd  = sqrt(var);
```

```
/* sort the given inputs to find median */
        for (i = 0; i < n - 1; i++)
                for (j = i; j < n; j++) {
                        if (x[i] > x[j]) {
                                temp = x[i];
                                x[i] = x[j];
                                x[j] = temp;
                        }
                }

        /* calculate the median */
        if ((n + 1) % 2 == 0) {
                median = x[((n + 1) / 2) - 1];
        } else {
                median = (x[((n + 1) / 2) - 1] + x[((n + 2) / 2) - 1]) / 2;
        }

        /* print the outputs */
        printf("Standard deviation: %f\n", sd);
        printf("Variance: %f\n", var);
        printf("Mean : %f\n", mean);
        printf("Median: %f\n", median);
        return 0;
  }
```

Output :

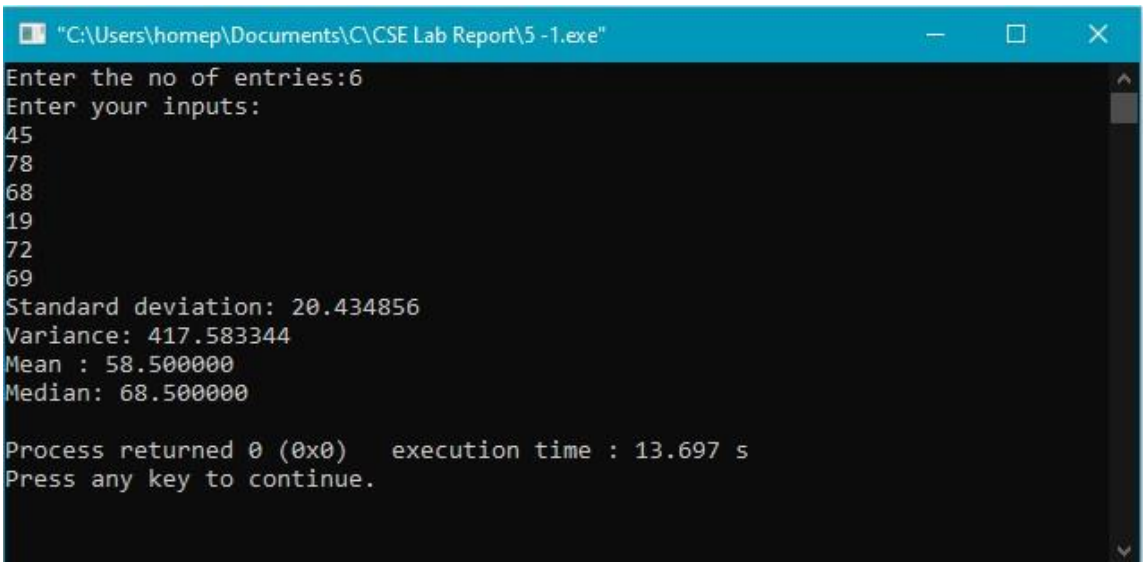


Fig 5.1 : C Program to Calculate mean, median and Standard Deviation.

2. Write a C program to convert Decimal to Binary number system.

Algorithm :

Step-1 : Start

Step-2 : Input decNo

Step-3 : Print biNo

Step-4 : Long convert binary (int dec no)

Step-5 : Static long , biNo , r , fctor=1

Step-6 : Is decNo ! = 0

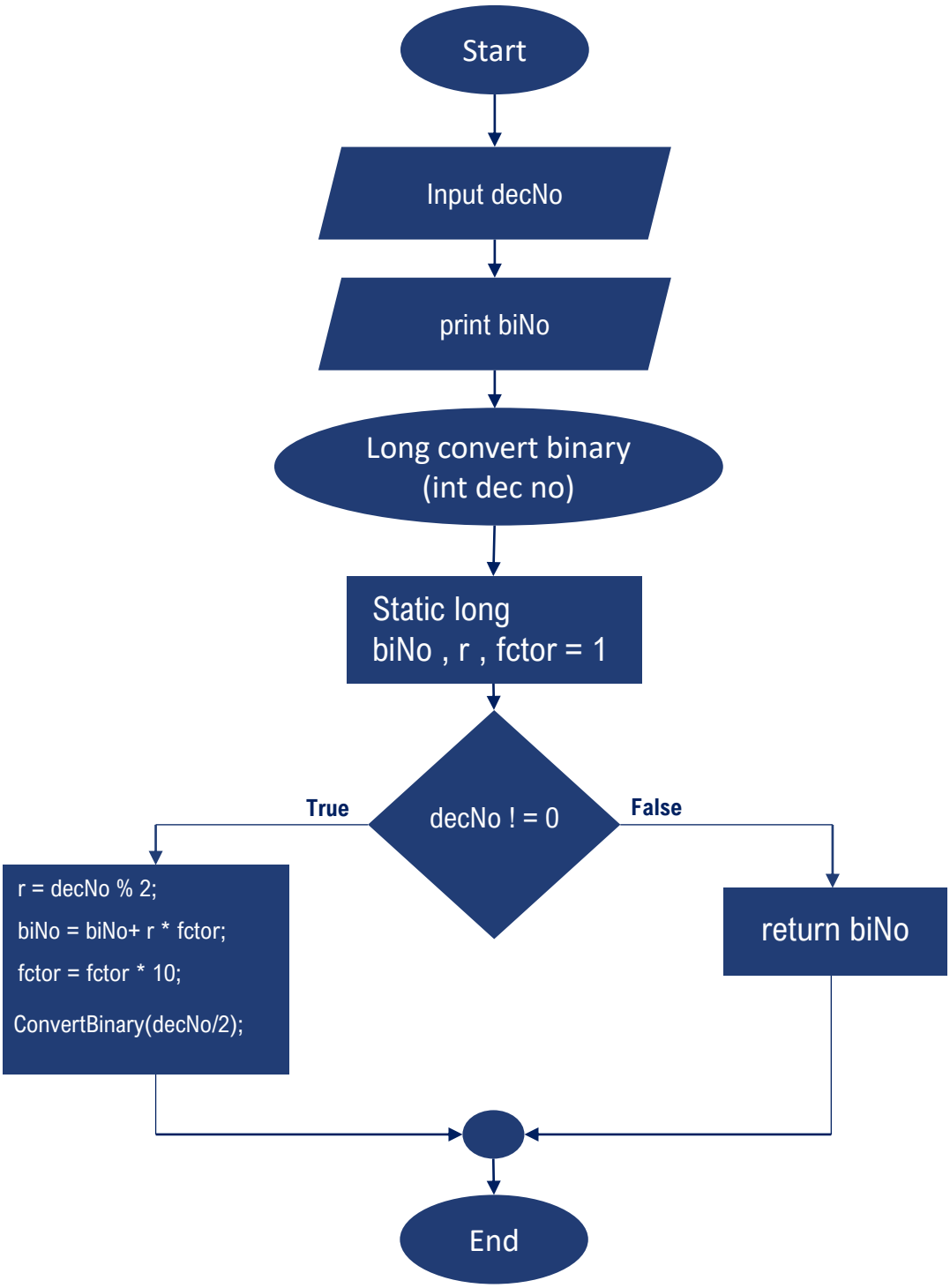
i) True,
r = decNo % 2;
biNo = biNo+ r * fctor;
fctor = fctor * 10;
ConvertBinary(decNo/2);

ii) False, return biNo

Step-7 : End

Flow chart on next page

Flowchart :


Start
Input decNo
print biNo
Long convert binary
(int dec no)
Static long
biNo , r , fctor = 1
decNo ! = 0
True
False
r = decNo % 2;
biNo = biNo+ r * fctor;
fctor = fctor * 10;
ConvertBinary(decNo/2);
return biNo
End

Code :

```
/*
Write a C program to convert Decimal to Binary number system.
*/
#include<stdio.h>
long convertBinary(int);

int main()
{
    long biNo;
    int decNo;

       printf("Convert decimal number to binary :\n");
       printf("-----------------------------------------------\n");

    printf("Input any decimal number : ");
    scanf("%d",&decNo);

    biNo = convertBinary(decNo);
    printf("The Binary value of decimal no. %d is : %ld\n\n",decNo,biNo);
    return 0;
}
long convertBinary(int decNo)
{
    static long biNo,r,fctor = 1;

    if(decNo != 0)
    {
        r = decNo % 2;
        biNo = biNo + r * fctor;
        fctor = fctor * 10;
        convertBinary(decNo / 2);
    }
    return biNo;
}
```

Output :

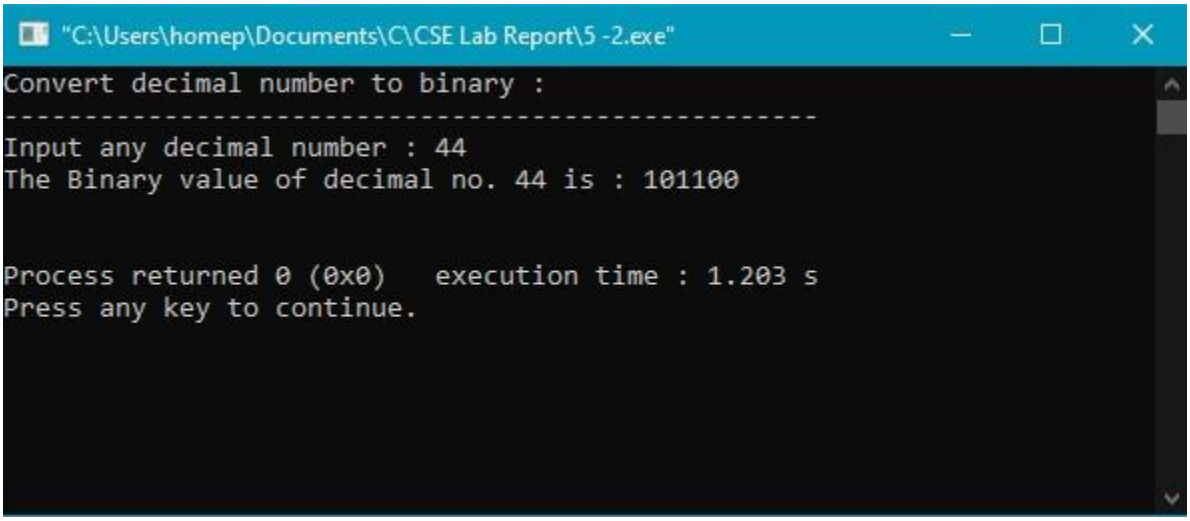


Fig 5.2 : C program to convert Decimal to Binary number system.

3. Write a C program to count frequency of each element in an array.

Algorithm :

Step-1 : Start

Step-2 : Input n

Step-3 : i = 0

Step-4 : Condition- 1 is i<n

1.1) True,
printf("element %d: ",i)
scanf("%d",&arr1[i])
fr1[i] = -1
Increment i++ and return to condition 1

1.2) False,
i = 0

Step-5 : Condition – 2 is i<n

2.1) True,
ctr = 1
j = i+1
Condition 3 - Is i<n

3.1) True,
Condition - 4 is arr1[i] = = arr1[j]

4.1) True,
ctr ++
fr [j]=0

4.2) False,
Increment j++ and return to condition 4

3.2) False, condition -5 is fr1[i]!=0

5.1) True,

fr1[i]=ctr

5.2) False,

Increment i++ and return to condition 2

2.2) False,

printf("frequency of all element");

i = 0

Condition -6 is i<n

6.1) True,

Condition -7 is fr1[i]!=0

7.1) True,

printf("%d occurs%d times")

arr1[i] , fr1[i]

7.1) False,

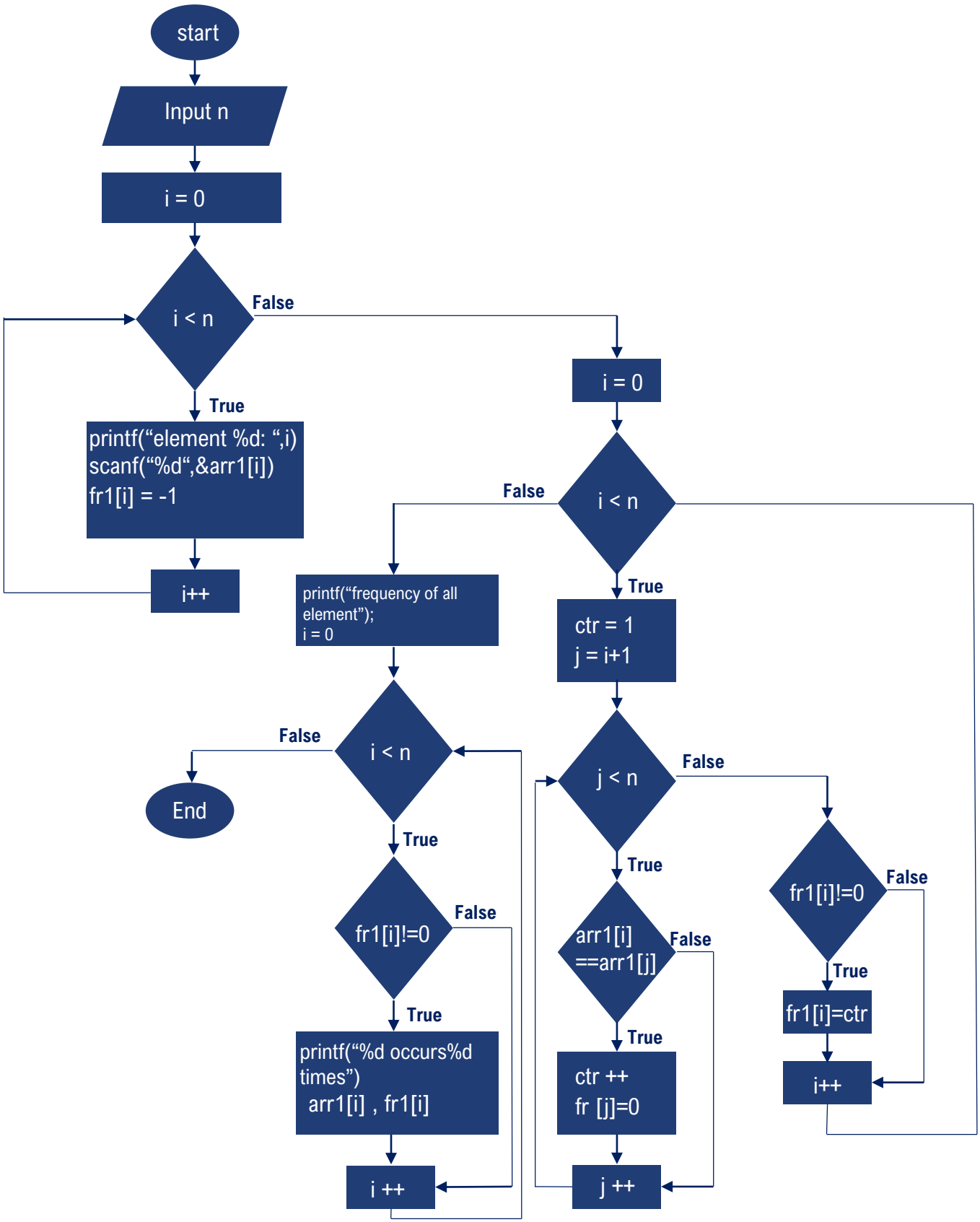
Increment i++ and return to condition 6

6.2) False,

End

Flow chart on next page

Flowchart :


start
Input n
i = 0
i < n
False
True
printf("element %d: ",i)
scanf("%d",&arr1[i])
fr1[i] = -1
i++
i = 0
i < n
False
True
printf("frequency of all element");
i = 0
ctr = 1
j = i+1
i < n
False
End
True
fr1[i]!=0
False
True
printf("%d occurs%d times")
arr1[i] , fr1[i]
i ++
j < n
False
True
arr1[i] ==arr1[j]
False
True
ctr ++
fr [j]=0
j ++
fr1[i]!=0
False
True
fr1[i]=ctr
i++

Code :

```
/*
Write a C program to count frequency of each element in an array.
*/
#include <stdio.h>
void main()
{
    int arr1[100], fr1[100];
    int n, i, j, ctr;


       printf("Count frequency of each element of an array:\n");
       printf("----------------------------------------------\n");

       printf("Input the number of elements to be stored in the array :");
       scanf("%d",&n);

       printf("Input %d elements in the array :\n",n);
       for(i=0;i<n;i++)
            {
             printf("element - %d : ",i);
             scanf("%d",&arr1[i]);
                fr1[i] = -1;
            }
    for(i=0; i<n; i++)
    {
        ctr = 1;
        for(j=i+1; j<n; j++)
        {
            if(arr1[i]==arr1[j])
            {
                ctr++;
                fr1[j] = 0;
            }
        }
```

```
    if(fr1[i]!=0)
        {
            fr1[i] = ctr;
        }
    }
    printf("\nThe frequency of all elements of array : \n");
    for(i=0; i<n; i++)
    {
        if(fr1[i]!=0)
        {
            printf("%d occurs %d times\n", arr1[i], fr1[i]);
        }
    }
}
```

Output :

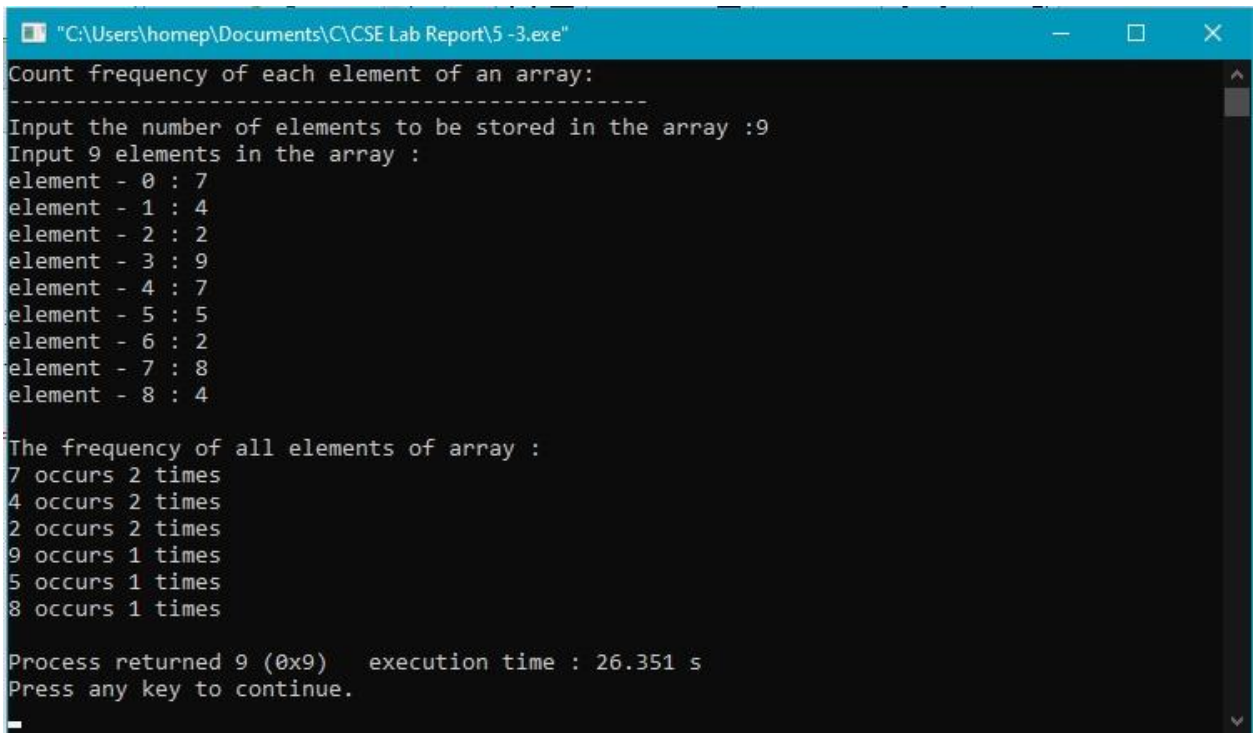


Fig 5.3 : C program to count frequency of each element in an array.

4. Write a C Program to Find Transpose of a Matrix.

Algorithm :

Step-1 : Start

Step-2 : Input r , c

Step-3 : i = 0

Step-4 : Condition- 1 is i<r

1.1) True, j = 0

Condition- 2 is i<c

2.1) True,
Print i,j
scan arr1[i] && i , j
c; j++ and return to condition 2

2.2) False,
r; i++ and return to condition 1

1.2) False, i = 0

Condition- 3 is i<r

3.1) True,
Printf("\n")
j = 0
Condition- 4 is j<c

4.1) True,
Printf("%d\tarr1[i][j]")
c; j++ and return to condition 4

4.2) False,
r; i++ and return to condition 3

3.2) False, i = 0

Condition- 5 is i<r

5.1) True, j = 0

j = 0

Condition- 6 is j<c

6.1) True,
brr1[j][i] = = arr1[j][i]
c; j++ and return to condition 6

6.2) False,
r; i++ and return to condition 5

5.2) False, i = 0

Condition- 7 is i<c

7.1) True,
Printf("\n")
j = 0
Condition- 8 is j<r

8.1) True,
Printf("%d\tbrr1[i][j]")
r; i++ and return to condition 8

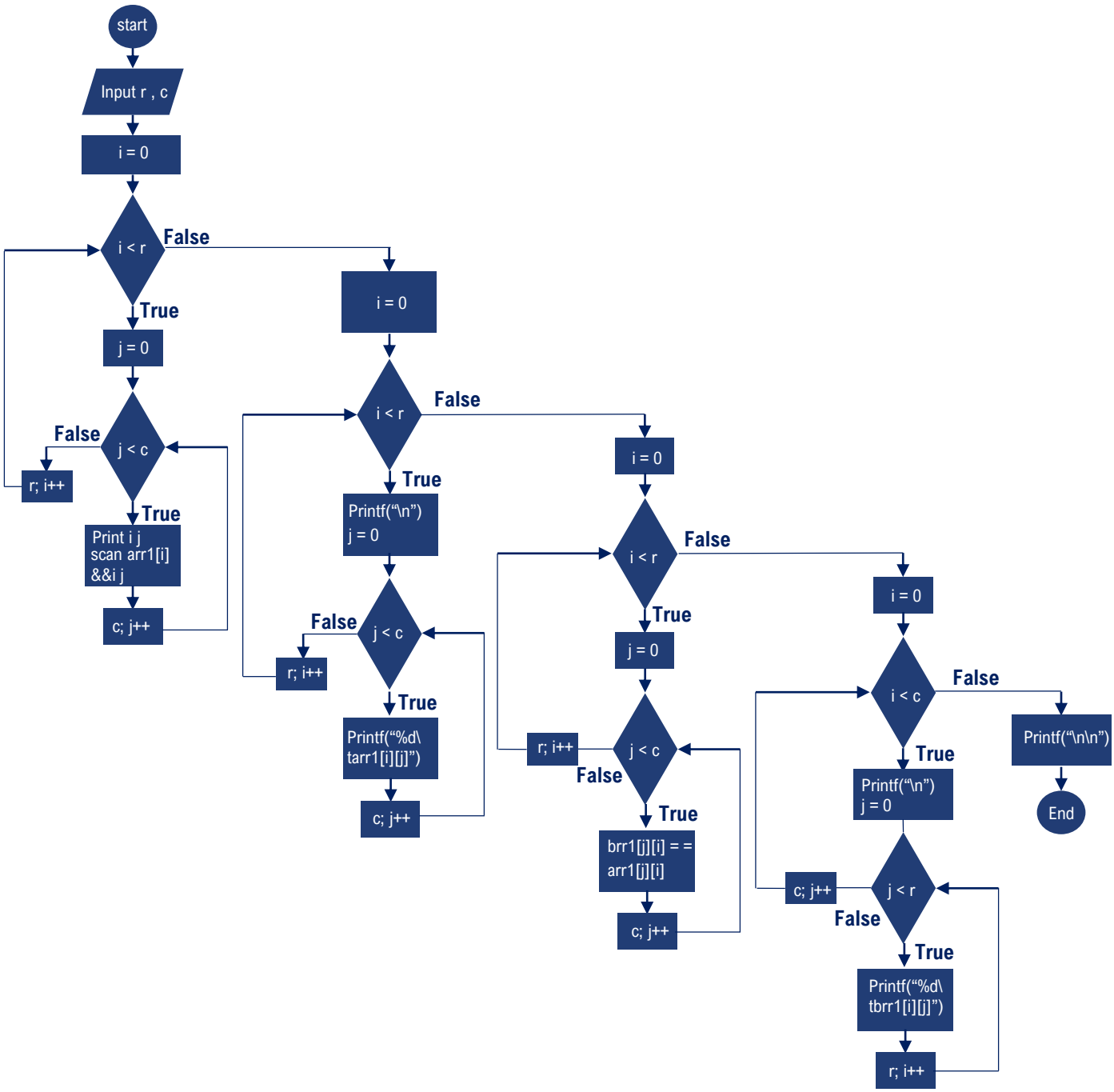
8.2) False,
c; j++ and return to condition 7

7.2) False,
Printf("\n\n")

Step-4 : End

Flow chart on next page

Flowchart :


start
Input r , c
i = 0
i < r
False
True
j = 0
j < c
False
True
r; i++
Print i j
scan arr1[i]
&&i j
c; j++
i = 0
i < r
False
True
Printf("\n")
j = 0
j < c
False
True
r; i++
Printf("%d\
tarr1[i][j]")
c; j++
i = 0
i < r
False
True
j = 0
j < c
False
True
r; i++
brr1[j][i] = =
arr1[j][i]
c; j++
i = 0
i < c
False
True
Printf("\n")
j = 0
j < r
False
True
c; j++
Printf("%d\
tbrr1[i][j]")
r; i++
Printf("\n\n")
End

Code :

```
/*
Write a C Program to Find Transpose of a Matrix.
*/
#include <stdio.h>
void main()

 {
 int arr1[50][50],brr1[50][50],i,j,r,c;

      printf("Transpose of a Matrix :\n");
      printf("-------------------------\n");


      printf("Input the rows and columns of the matrix : ");
      scanf("%d %d",&r,&c);

      printf("Input elements in the first matrix :\n");
      for(i=0;i<r;i++)
       {
          for(j=0;j<c;j++)
          {
                printf("element - [%d],[%d] : ",i,j);
                scanf("%d",&arr1[i][j]);
          }
       }

       printf("\nThe matrix is :\n");
             for(i=0;i<r;i++)
             {
             printf("\n");
             for(j=0;j<c;j++)
             printf("%d\t",arr1[i][j]);
             }
```

```
 for(i=0;i<r;i++)
    {
     for(j=0;j<c;j++)
          {
               brr1[j][i]=arr1[i][j];
          }
     }

     printf("\n\nThe transpose of a matrix is : ");
     for(i=0;i<c;i++){
     printf("\n");
     for(j=0;j<r;j++){
         printf("%d\t",brr1[i][j]);
     }
  }
     printf("\n\n");
}
```

Output :

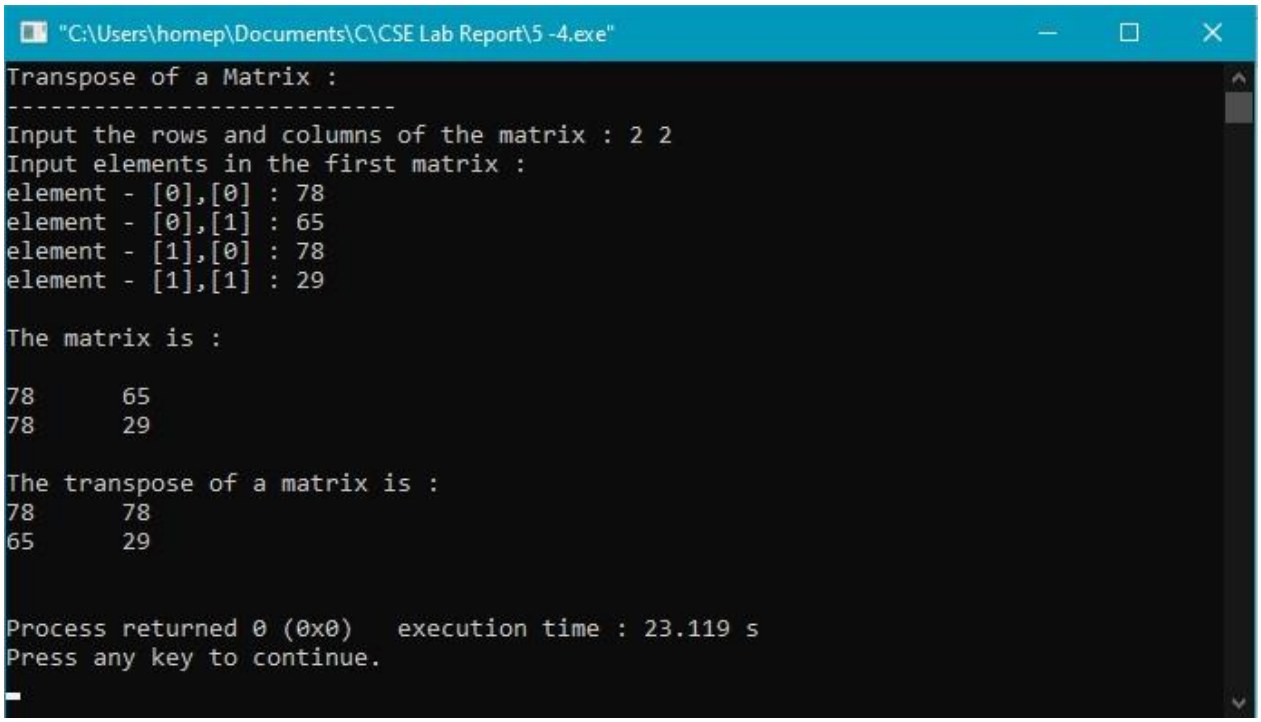


Fig 5.4 : C Program to Find Transpose of a Matrix.

• ANALYSIS AND DISCUSSION :

1. We got the exact result on output. Sometimes the result was wrong but we found the right implementation.

2. The problem of display anything in output is the easiest implementation. We solve that very easy.

3. In this assignment, I was faced some problem on first and last question but by the teachers help I solve it.

4. Nothing is very/most difficult parts in my program to implement.

5. All program is easy to understand and these helped me a lot to remove my confusion about Array in c programming.

6. I learnt- how to Calculate mean, median and Standard Deviation, Decimal to Binary number system , count frequency of each element in an array , Find Transpose of a Matrix. program and many basic things about c programming.