Laboratory Program 7

COMPUTE SIN(X)/COS(X) USING TAYLOR SERIES APPROXIMATION. COMPARE YOUR RESULT WITH THE BUILT-IN LIBRARY FUNCTION. PRINT BOTH THE RESULTS WITH APPROPRIATE INFERENCES.

AIM: To develop a program to compute sin(x)/cos(x) using taylor series approximation and compare the result with built-in function.

ALGORITHM:

Step-1: Start

Step-2: [Read the value of x in degree]- Read x

Step-3: [Initialization and Radians Conversion]

Temp = x

x=x*(3.142/180.0)

Term = x

sinx = term

n=1

Step-4: [Computer sin value]

Repeat while (term>FLT_EPSILON)

Fact = 2*n*(2*n+1)

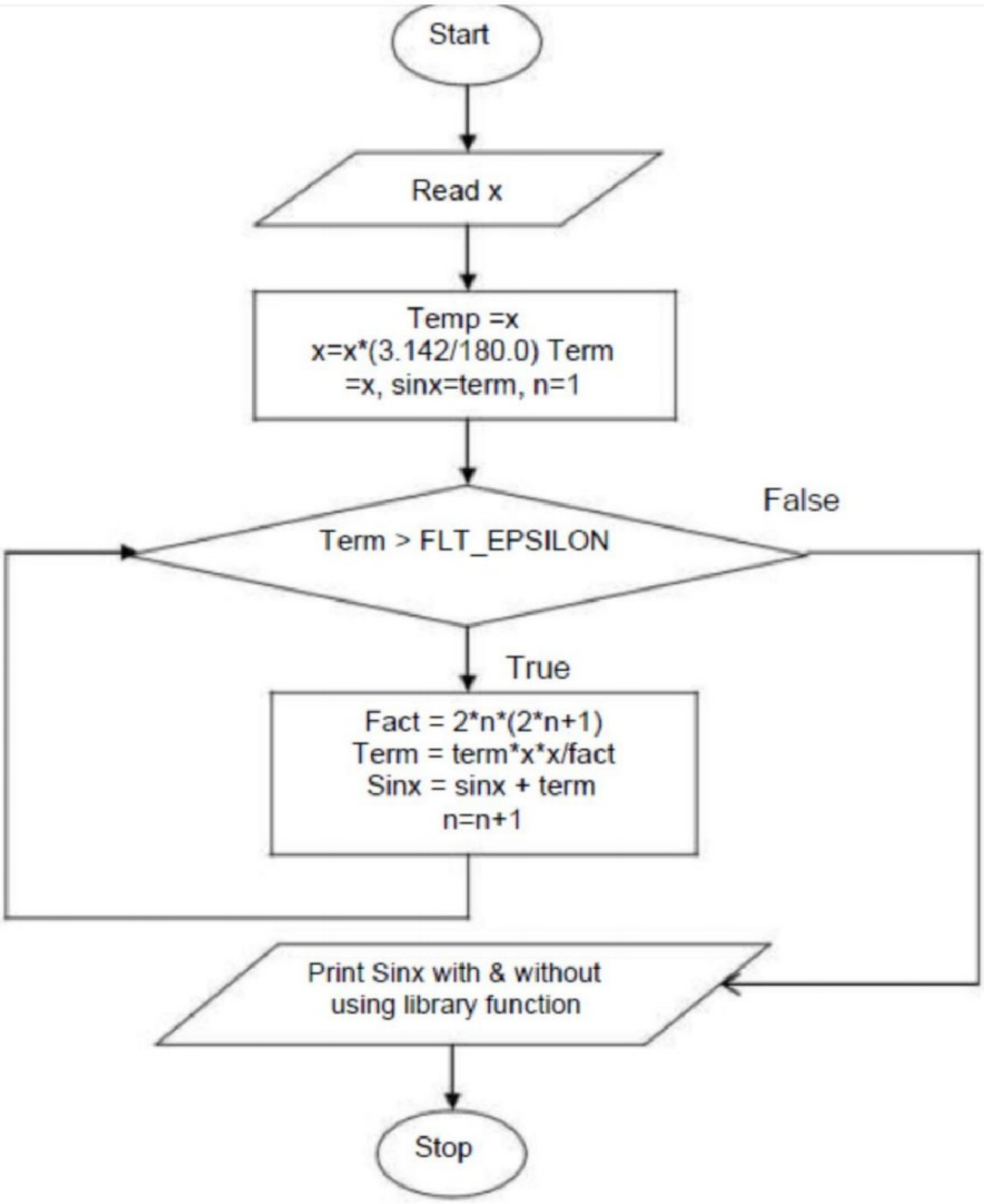
Term = term * x * x /fact Sinx

sinx +term = n + 1n

Step-5: [Output]- Print sin(x) without using library function of Print sin(x) and with using library function

Step-6: Stop

FLOWCHART



Laboratory Program 8

SORT THE GIVEN SET OF N NUMBERS USING BUBBLE SORT.

AIM: To develop a program to sort the given set of N numbers using Bubble sort.

ALGORITHM

Step **1.** [Initialize] Start

2. [Input number of elements] read *n*

3. [Input unsorted elements in array] read elements in array *a*[ ]

4. print elements of array *a*[ ]

5. [Iterate array *a*[ ] in two loops.

Outer loop gives number of passes.

Inner loop does swap task. In each pass, compare each pair of adjacent items.

If former element is greater than latter one, swap them.

[Iterate array a[ ] with

for each value *i* in array *a[i] to n* **do**

for each value *j* in array *a[j] to n-1* **do**

[Compare each pair of adjacent elements]

if (*a[j]* > a[j+1])**then**

[Swap these elements using *temp* variable]

temp ← *a[j]*

a[j] ← *a[j+1]*

a[j+1] ← *temp*

endif

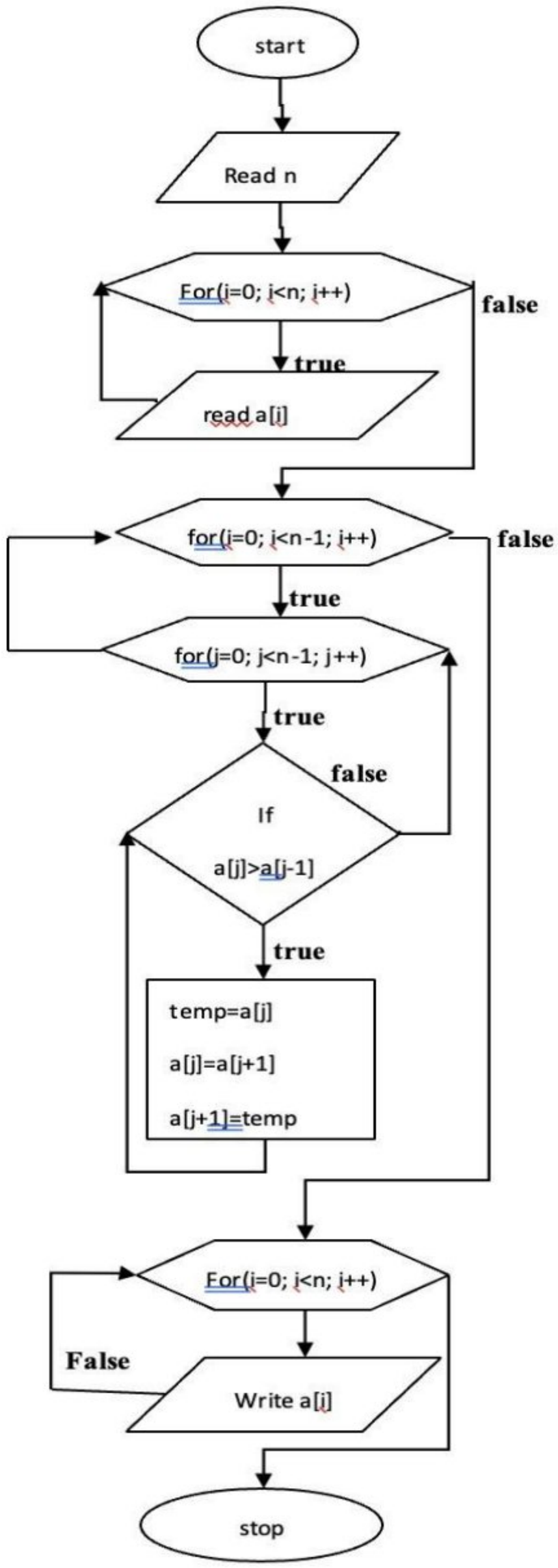
end for

end for

6. Print array with sorted elements

7. [Finished] End.

FLOWCHART



PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int n,i,j,a[10],temp;
        clrscr();
        printf("Enter the no. of elements : \n");
        scanf("%d",&n);
        printf("Enter the array elements \n");
        for(i = 0 ; i < n ; i++)
        scanf("%d",&a[i]);
        printf("The original elements are \n");
        for(i = 0 ; i < n ; i++)
        printf("%d ",a[i]);
      for(i = 0 ; i < n-1 ; i++)
      {
        for(j = 0 ; j< (n-i)-1; j++)
        {
          if(a[j] > a[j+1])
          {
            temp = a[j];
            a[j] = a[j+1];
            a[j+1] = temp;
          }
        }
      }
        printf("\n The Sorted elements are \n");
        for(i = 0 ; i < n ; i++)
        printf("%d ",a[i]);
        getch();
}
```

OUTPUT

First Run:

Enter the no. of elements : 5

Enter the array elements

30 10 50 20 40

The original elements are

30 10 50 20 40

The Sorted elements are

10 20 30 40 50

Second Run:

Enter the no. of elements : 6

Enter the array elements

6 5 4 3 2 1

The original elements are

6 5 4 3 2 1

The Sorted elements are

1 2 3 4 5 6

RESULT: -Thus, the program to sort the elements using bubble sort has been executed successfully and the output was verified.

VIVA QUESTIONS: -

a. Why the name bubble sort?

b. What are the different types of sorting techniques?

c. Explain the logic of bubble sort with an example.

d. What is nested for loop?

Laboratory Program 9

WRITE FUNCTIONS TO IMPLEMENT STRING OPERATIONS SUCH AS COMPARE, CONCATENATE, STRING LENGTH. USE THE PARAMETER PASSING TECHNIQUES.

AIM: To write functions to implement string operations such as compare, concatenate, string length.

ALGORITHM

Step 1: Start

Step 2: [Input two source strings]

read source1,source2

Step 3: [Calculate length1 of source1 by calling the user defined function,strlength();

Repeat the same for length2 of source2]

length1=strlength(source1)

length2=strlength(source2)

Step 4: [Ouput length1,length2]

Print length1,length2

Step 5: [Compare the two strings by calling the user defined function, strcompare()]

k=strcompare(source1,source2)

Step 6: [check k, to find the whether the strings are same or not]

if(k==0)

print Both strings are same

else

print strings are different

end if

Step 7: [Concatenate two strings by calling the user defined function, strconcat() and the concatenated string is stored in source1]

strconcat(source1,source2)

print source1

Step 8: Stop

User Defined Function - strlength()

Step 1: Start

Step 2: [set i=0]

i=0

Step 3: [receive the source string as str, read character by character, count one by one until we reach NULL character]

while(str[i]!='\0')

i++ end while

Step 4: [return i to the calling function]

return i

User Defined Function - strcompare( )

Step 1: Start

Step 2: [set i=0]

i=0

Step 3: [Receive both the source strings as str1 and str2, read character by character until they match, if the matched character is a NULL then go out of while loop, when any unmatched character then go out of loop]

while(str1[i] = = str2[i])

if(str1[i] = = '\0')

break

end if

i++

end while

Step 4: [calculate k]

k=str1[i]-str2[j]

Step 5: [return i to the calling function]

return k

User Defined Function - strconcat( )

Step 1: Start

Step 2: [set i=0]

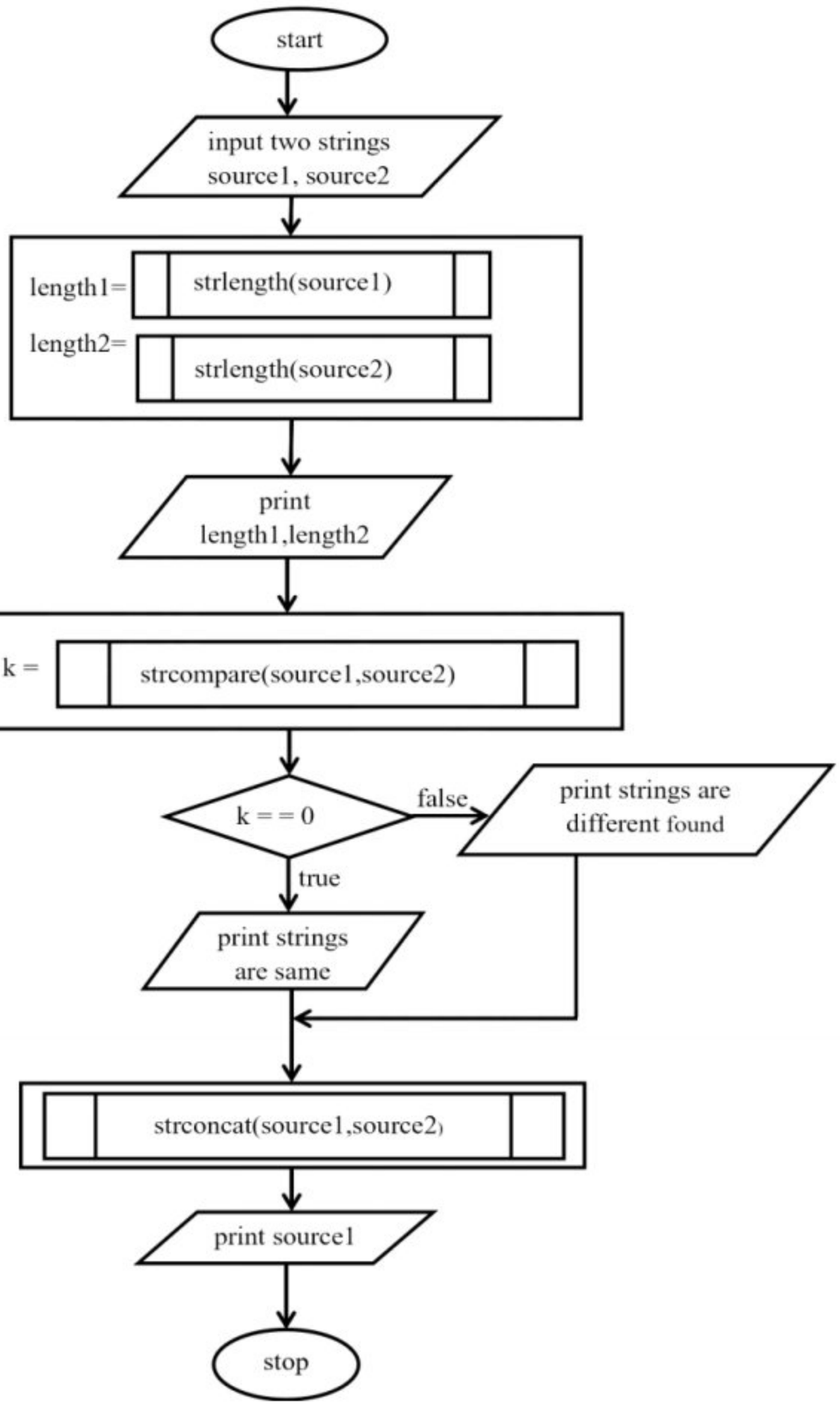
i=0

Step 3: [Receive both the source strings as str1 and str2, calculate length of str1 using strlength() as l]

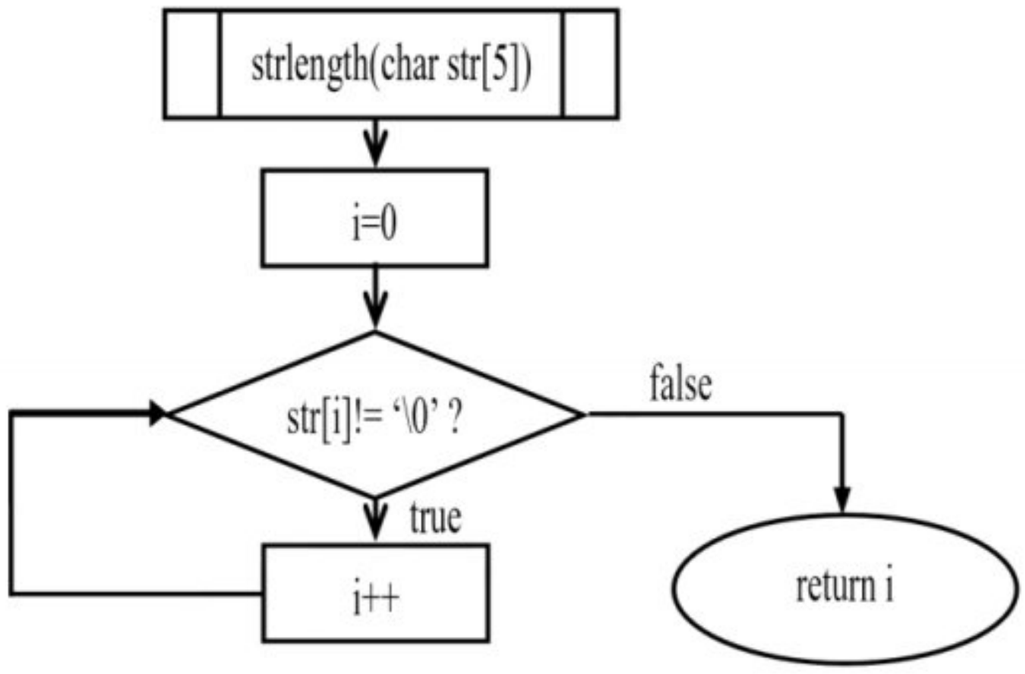
```
l=strlength(str1)
Step 4: [read str2 character by character and store it from the end of str1]
        while(str2[i]!='\0')
        str1[l+i]=str2[i]
        i++
        end while
Step 5: [return to the calling function]
return;
```

return

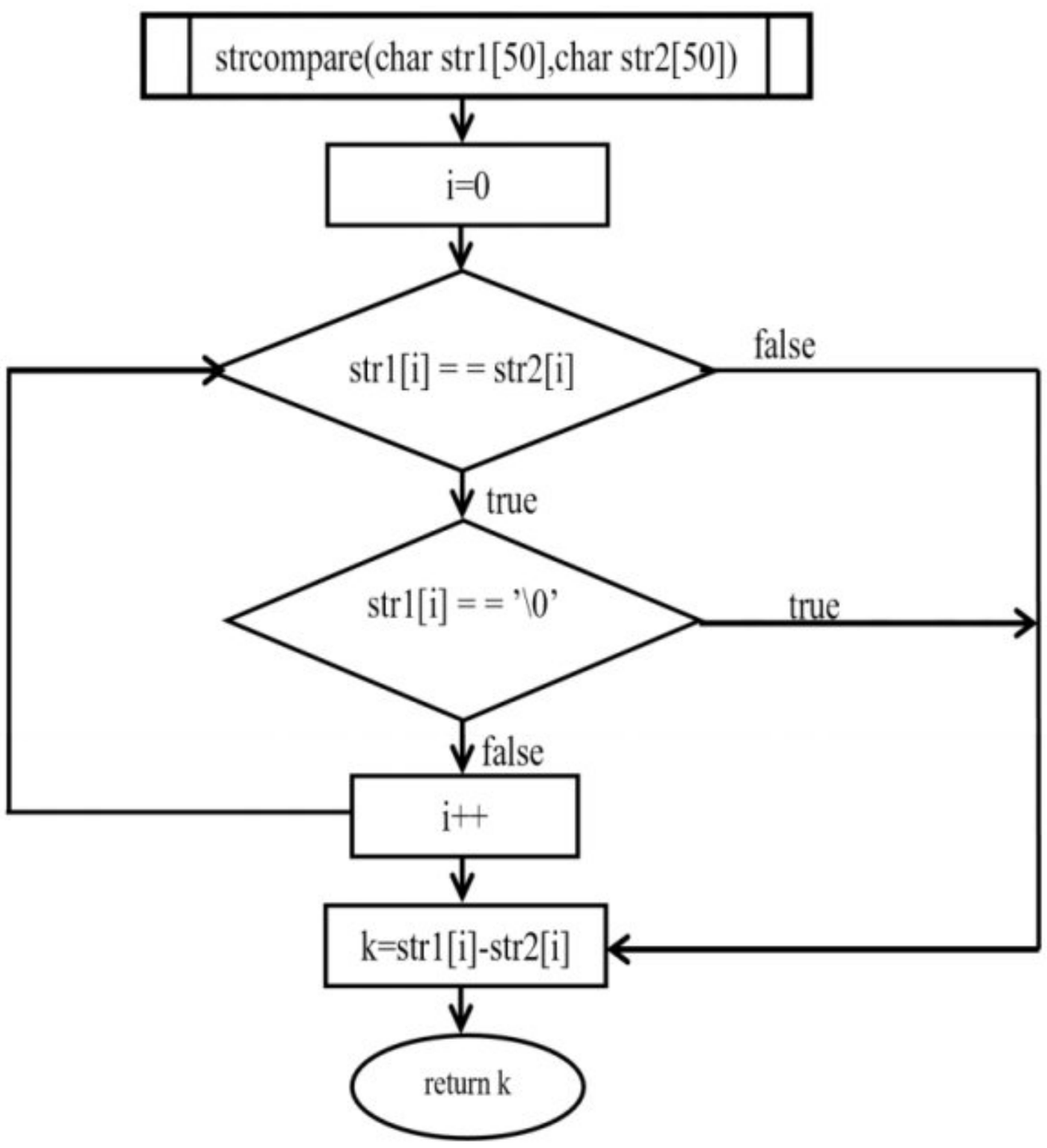
Flowchart:



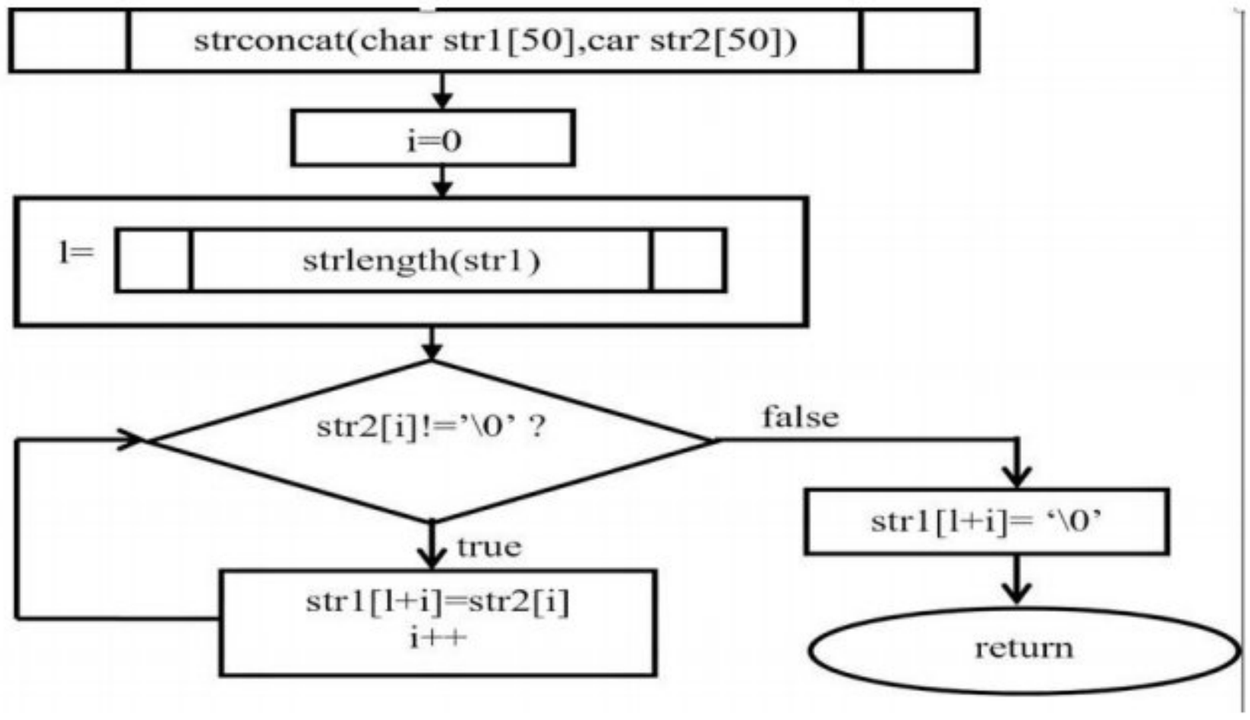
strlength( )



strcompare( )



strconcat( )



PROGRAM

```
#include<stdio.h>
#include<conio.h>
int strlength(char str1[50]);

void strconcat(char str1[50],char str2[50]);
int strcompare(char str1[50],char str2[50]);
int strlength(char str[50])
{
int i=0;
while(str[i]!='\0')
i++;
return i;
}
void strconcat(char str1[50],char str2[50])
{
int i=0,l;
l=strlength(str1);
while(str2[i]!='\0')
{
str1[l+i]=str2[i];
```

```
i++;
}
str1[l+i]='\0';
}
int strcompare(char str1[50],char str2[50])
{
int i=0, k;
while(str1[i]==str2[i])
{
if(str1[i]=='\0')
break;
i++;
}
k=str1[i]-str2[i];
return k;
}
void main()
{
char source1[50],source2[50],dest[50];
int length1,length2,k;
clrscr();
printf("\n Enter the source string 1:");
gets(source1);
printf("\n Enter the source string 2:");
gets(source2);
length1=strlength(source1);
length2=strlength(source2);
printf("\n string length of string 1 is %d",length1);
printf("\n string length of string 2 is %d",length2);
k=strcompare(source1,source2);
if(k==0)
printf("\n Both string are same");
```

```
else
printf("\n Both string are different");
strconcat(source1,source2);
printf("\n concatenated string is ");
puts(source1);
getch();
}
```

OUTPUT

```
First Run:
Enter the source string1: good
Enter the source string2: night
String length of string1 is: 4
String length of string2 is: 5
strings  are different
concatenated string is: goodnight
Second Run:
Enter the source string1: good
Enter the source string2: good
String length of string1 is: 4
String length of string2 is: 4
Both strings are same
concatenated string is: goodgood
```

RESULT: - Thus, the program to implement string operations such as compare, concatenate, string length has been executed successfully and the output was verified.

VIVA QUESTIONS: -

a. What is string?

b. How to declare string?

c. What are the string manipulation function?

d. What is gets() and puts() function in string?

Laboratory Program 10

IMPLEMENT STRUCTURES TO READ, WRITE AND COMPUTE AVERAGE- MARKS OF THE STUDENTS, LIST THE STUDENTS SCORING ABOVE AND BELOW THE AVERAGE MARKS FOR A CLASS OF N STUDENTS.

AIM: To write a C Program to implement structures to read, write and compute average-marks of the students, list the students scoring above and below the average marks for a class of N students.

ALGORITHM

Step-1: Start

Step-2: Read number of students

Step-3: For every student, read the student id, name , marks for all the subjects

Step-4: Calculate the average marks and store it in the avg field

Step-5: Print the results

Step-6: Initialise loop

Step-7: Read the average of every student

Step-8: Check for if avg>35.00

Step-9: If yes than print the result else goto next iteration

Step-10: Initialise the loop

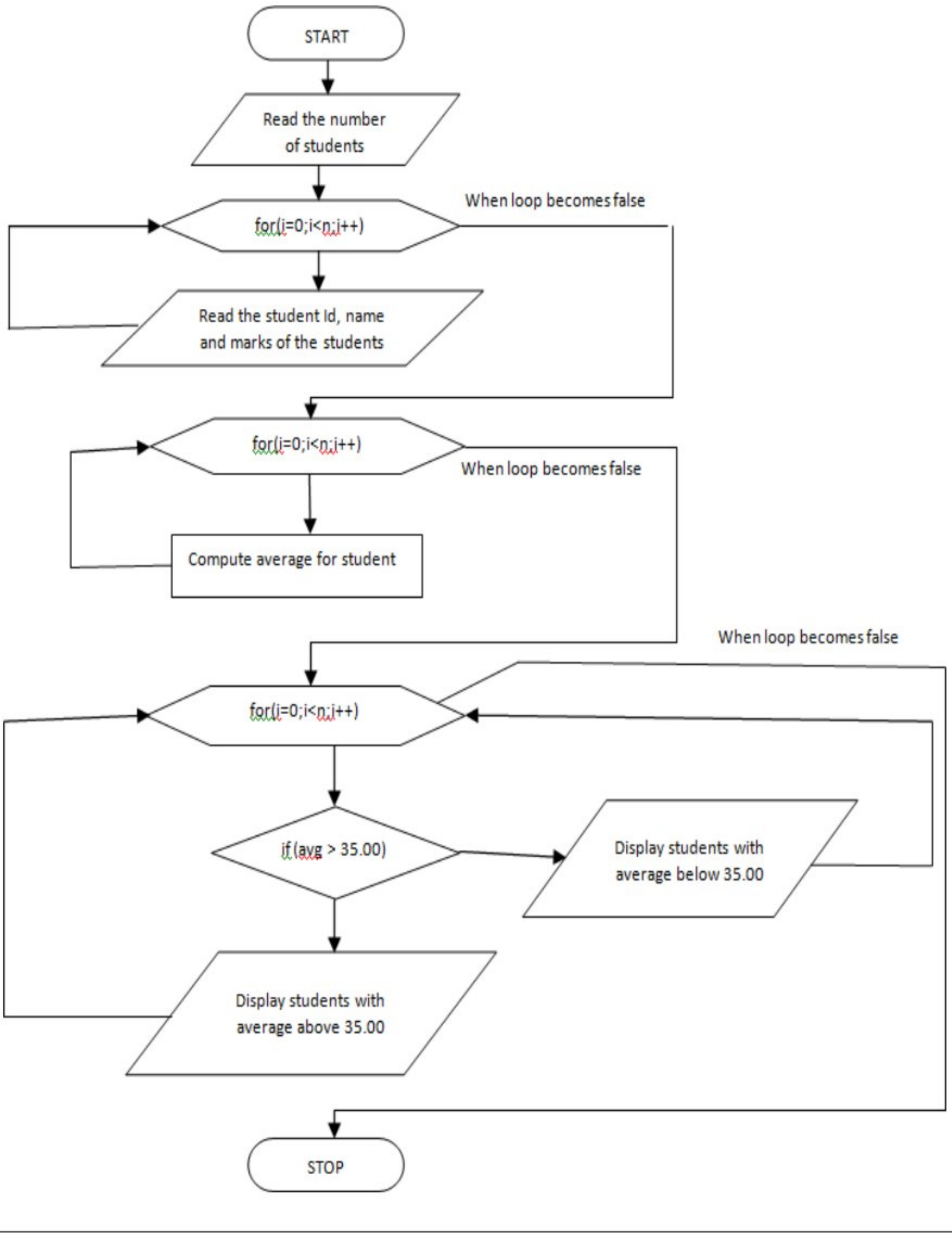
Step-11: Read average of every student

Step-12: Check if avg<35.00

Step-13: If yes than print result else goto next iteration

Step-14: Stop

FLOWCHART



PROGRAM

```
#include<stdio.h>
#include<conio.h>
struct student
{
char usn[10];
char name[10];
int m1,m2,m3;
float avg, total;
};
void main()
{
struct student s[20];
int n,i;
float tavg,sum=0.0;
clrscr();
printf("Enter the number of students");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Enter the detail of %d students\n",i+1);
printf("\n Enter USN=");
scanf("%s",s[i].usn);
printf("\n Enter Name=");
scanf("%s",s[i].name);
printf("\nEnter the three subjects marks\n");
scanf("%d%d%d",&s[i].m1,&s[i].m2,&s[i].m3);
s[i].total=s[i].m1+s[i].m2+s[i].m3;
s[i].avg=s[i].total/3;
}
for(i=0;i<n;i++)
{
```

```
 if(s[i].avg>=35)
 printf("\n %s has scored above the average marks",s[i].name);
 else
 printf("\n %s has scored below the average marks",s[i].name);
}
getch();
}
```

OUTPUT

```
Enter the number of students2
Enter the detail of student 1
Enter USN=1
Enter Name=Arun
Enter the three-subject score
23 45 67
Enter the detail of student 2
Enter USN=2
Enter Name=Tharun
Enter the three-subject score
5 3 2
 Arun has scored above the average marks
 Tharun has scored below the average marks
```

RESULT: -Thus, the program to implement structures to read, write and compute average- marks of the students, list the students scoring above and below the average marks for a class of N students has been executed successfully and the output was verified.

VIVA QUESTIONS: -

a. What is structure?

b. How to declare a structure?

c. What is structure member?

d. What is difference between array and structure?

e. What is nested structure?

f. What is typedef?

Laboratory Program 11

DEVELOP A PROGRAM USING POINTERS TO COMPUTE THE SUM, MEAN AND STANDARD DEVIATION OF ALL ELEMENTS STORED IN AN ARRAY OF N REAL NUMBERS.

AIM: To Develop a C program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of n real numbers.

ALGORITHM

Step-1: Start

Step-2: Read n

Step-3: For every value of n read the x

Step-4: Initialize sum=0 and i=0

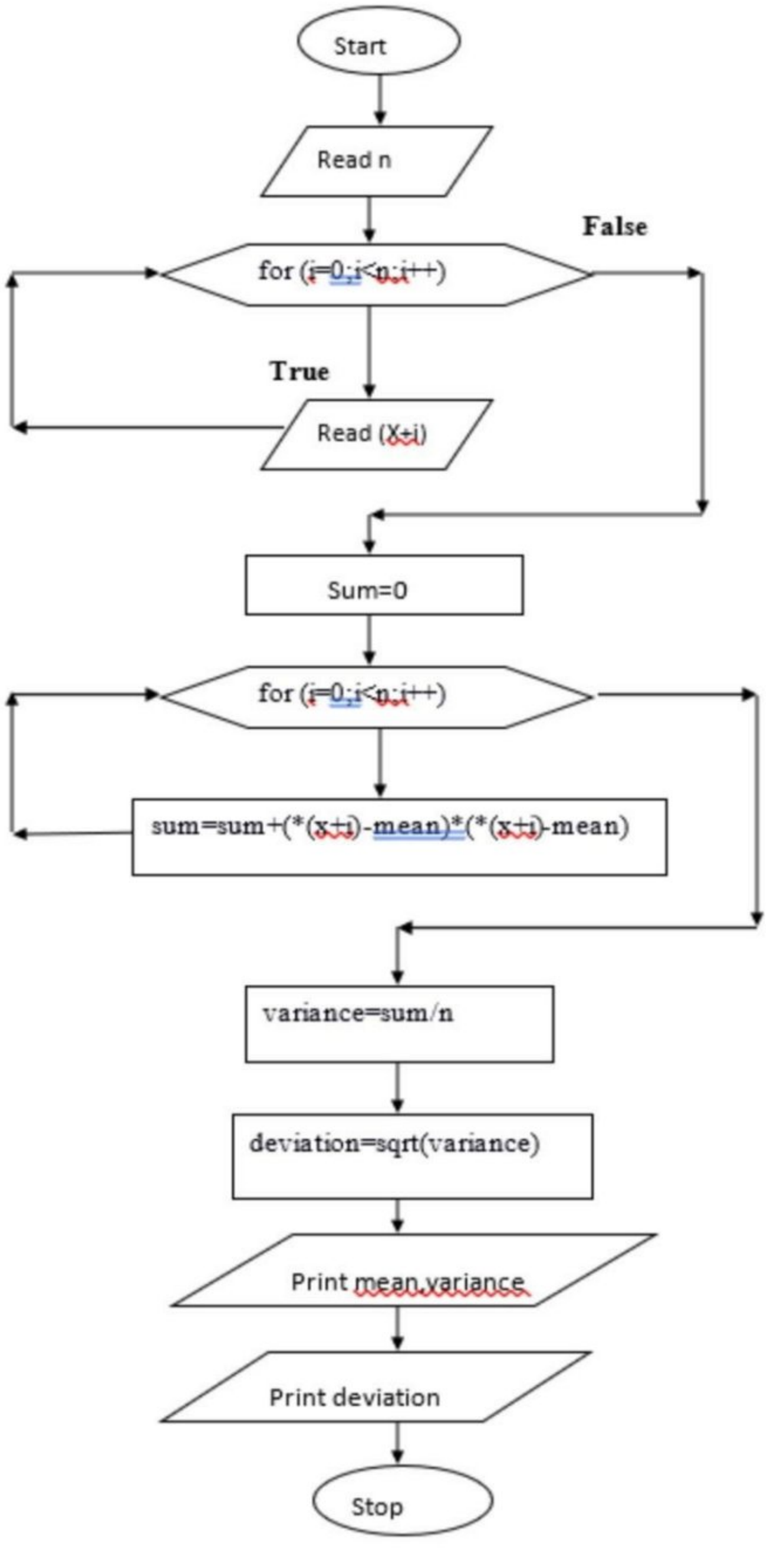
Step-5: For every value of n and i, comuter sum using sum = sum + (*(x+i) – mean) * (* (x+i) – mean)

Step-6: Using the sum value computer variance = sum / n and deviation = sqrt (variance)

Step-7: Display mean, variance, deviation

Step-8: Stop

FLOWCHART



PROGRAM

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
int n , i;
float x[20],sum,mean;
float variance , deviation;
clrscr();
printf("Enter the value of n \n");
scanf("%d",&n);
printf("enter %d real values \n",n);
for (i=0;i<n;i++)
 {
 scanf("%f",(x+i));
 }
sum=0;
for(i=0;i<n;i++)
{
sum= sum+*(x+i);
}
printf("sum=%f\n",sum);
mean=sum/n;
sum=0;
for(i=0;i<n;i++)
{
sum=sum+(*(x+i)-mean)*(*(x+i)-mean);
}
variance=sum/n;
deviation=sqrt(variance);
printf("mean(Average)=%f\n",mean);
```

```
printf("variance=%f\n",variance);
printf("standard deviation=%f\n",deviation);
getch();
}
```

Output:

```
Enter the value of n
5
Enter the 5 real values
3
7
23
1
4
Sum = 38.0000
Mean ( Average ) = 7.6000
Variance = 63.039997
Standard deviation = 7.9397
```

RESULT: -Thus, the program to implement pointers to compute the sum, mean and standard deviation of all elements stored in an array of n real numbers has been executed successfully and the output was verified.

VIVA QUESTIONS:-

a. Define pointer
b. Define array of pointer
c. Difference between (x+i) and *(x+i)
d. Define array

Laboratory Program 12

WRITE A C PROGRAM TO COPY A TEXT FILE TO ANOTHER, READ BOTH THE INPUT FILE NAME AND TARGET FILE NAME.

AIM: To Write a C program to copy a text file to another, read both the input file name and target file name.

ALGORITHM

Step 1 : Start

Step 2: Read the source file name fname1

Step 3: Open the file fname1 in read mode

Step 4: if fptr1 is equal to NULL

print " File does not found or error in opening.!!"

goto Step 12

Step 5: Read the new file name fname2

Step 6: Open the file fname2 in write mode

Step 7: if fptr2 is equal to NULL

print " File does not found or error in opening.!!"

goto Step 12

Step 8: Repeat while(1)

ch=fgetc(fptr1);

if ch is equal to EOF

break;

else

fputc(ch, fptr2);

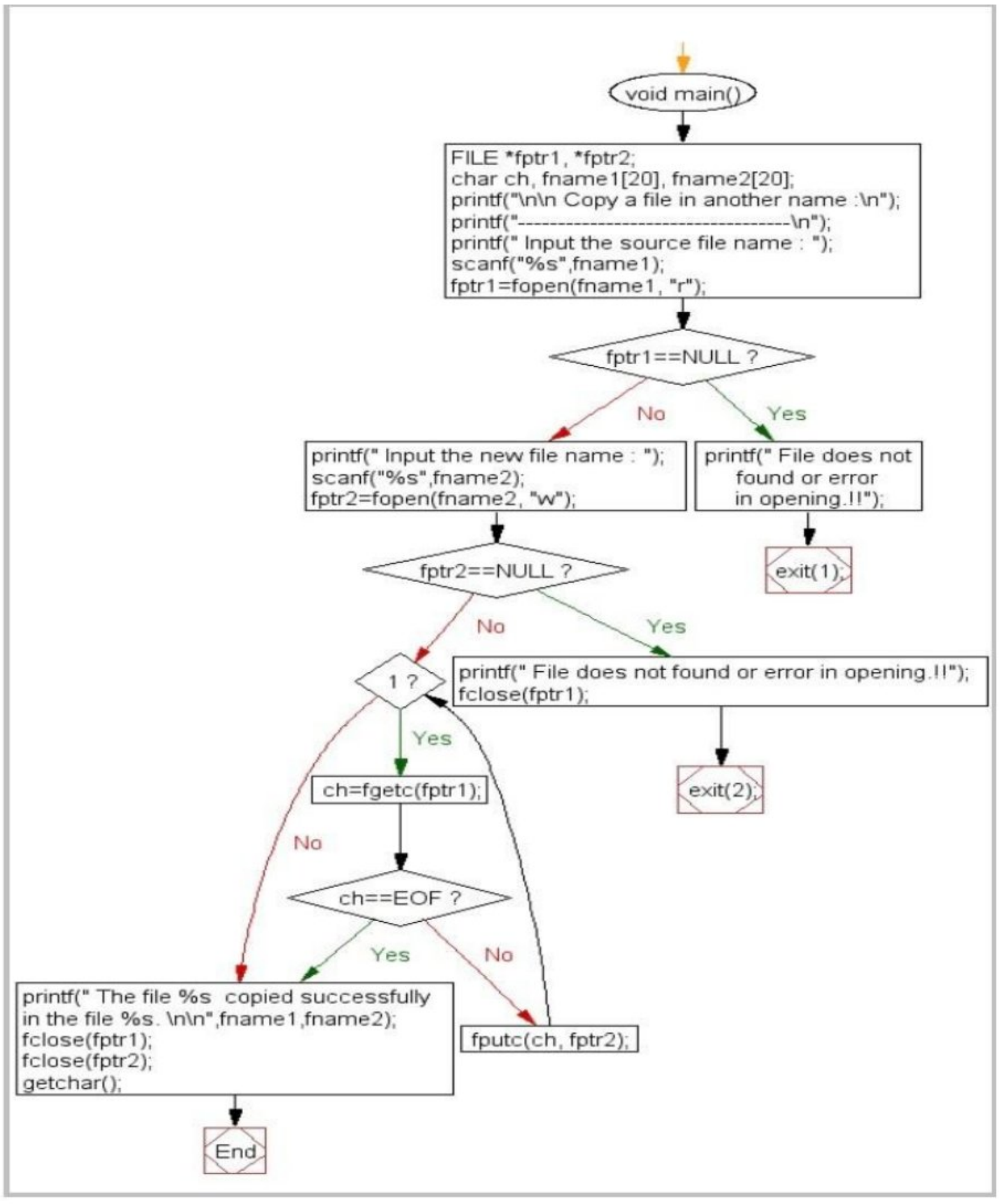
Step 9: print "The file fname1 copied successfully in the file fname2"

Step 10: close file pointer fptr1

Step 11: close file pointer fptr2

Step 12: Stop

FLOWCHART



PROGRAM

```
#include  <stdio.h>
#include <stdlib.h>
#include <conio.h>
void main()
{
        FILE *fptr1, *fptr2;
        char ch, fname1[20], fname2[20];
        clrscr();

        printf("\n\n Copy a file in another name :\n");
        printf("-----------------------------------------.\n");

        printf(" Input the source file name : ");
        scanf("%s",fname1);

        fptr1=fopen(fname1, "r");
        if(fptr1==NULL)
        {
                printf(" File does not found or error in opening.!!");
                exit(1);
        }
        printf(" Input the new file name : ");
        scanf("%s",fname2);
        fptr2=fopen(fname2, "w");
        if(fptr2==NULL)
        {
                printf(" File does not found or error in opening.!!");
                fclose(fptr1);
                exit(2);
        }
        while(1)
        {
                ch=fgetc(fptr1);
                if(ch==EOF)
                {
                        break;
                }
                else
```

```
            {
                fputc(ch, fptr2);
            }
        }
        printf(" The file %s copied successfully in the file %s. \n\n",fname1,fname2);
        fclose(fptr1);
        fclose(fptr2);
        getchar();
}
```

OUTPUT:

```
Copy a file in another name :
----------------------
 Input the source file name : test.txt
 Input the new file name : test1.txt
 The file test.txt  copied successfully in the file test1.txt.
```

RESULT: - Thus, the program to copy a text file to another, read both the input file name and target file name has been executed successfully and the output was verified.

VIVA QUESTIONS:-

a. Define file.
b. What is fopen() and fclose() in file?
c. What is fgetc() and fputc() in file?