```
include<stdio.h>
nt main()

nt sum=0,i,n,p;
intf("enter the number:");
canf("%d",&n);
=n;
hile(i != 0)
p=i%10;
=i/10;
um=sum+p*p*p;

f (sum==n){
rintf("the number is Armstrong number");}
lse
printf("not an armstrong ");}
eturn 0;
```

three

16

```
guest-7iyxzo@student-desktop: ~/Desktop/Amal
guest-7iyxzo@student-desktop:~/Desktop/Amal$ ./a.out
Enter number
153
Armstrongguest-7iyxzo@student-desktop:~/Desktop/Amal$ ./a.out
Enter number
147
not Armstrongguest-7iyxzo@student-desktop:~/Desktop/Amal$
```

D.Y. PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Grade:
Sign:
Date:

EXPERIMENT NO. 4

Name: Sunidhishree Prabakaran
Roll no: 22CE1116
Batch: B3
Branch: CE Core

Experiment - IV

Write a program to compute (i) Square root of number (ii) Square of number (iii) cube of number (iv) check for prime (v) factorial of number (vi) prime factors of given number using switch

Theory :

C switch statement.

The switch statement in C is an alternate to if-else-if ladder statement which allows us to execute multiple operations for the different possibles values of a single variable called switch variable. Here, we can define various statements in the multiple cases for the different values of a single variable

The syntax of switch statement in C language is given below:

```
switch (expression){
case value1:
// code to be executed;
break; //optional
case value 2:
// code to be executed;
```

Algorithm

Step 1: Start
Step 2: Declare the variables
Step 3: Input the number (ch)
Step 4: Start with the switch condition
Step 5: In case 1; A = ch * ch, print "A" i.e. square of number
Step 6: In case 2 (square root of number), B = sqrt(ch), print "B"
Step 7: In case 3 (cube of number), C = ch * ch * ch, print "C"
Step 8: In case 4 (prime number), using for loop condition, if condition satisfies then it will print "the number is prime number" otherwise it will print "the number is not prime no."
Step 9: In case 5 (factorial of number), using for loop, print "factorial"
Step 10: In case 6 (prime factors of number), using for loop condition, prime factors will be printed
Step 11: Stop

break; //optional

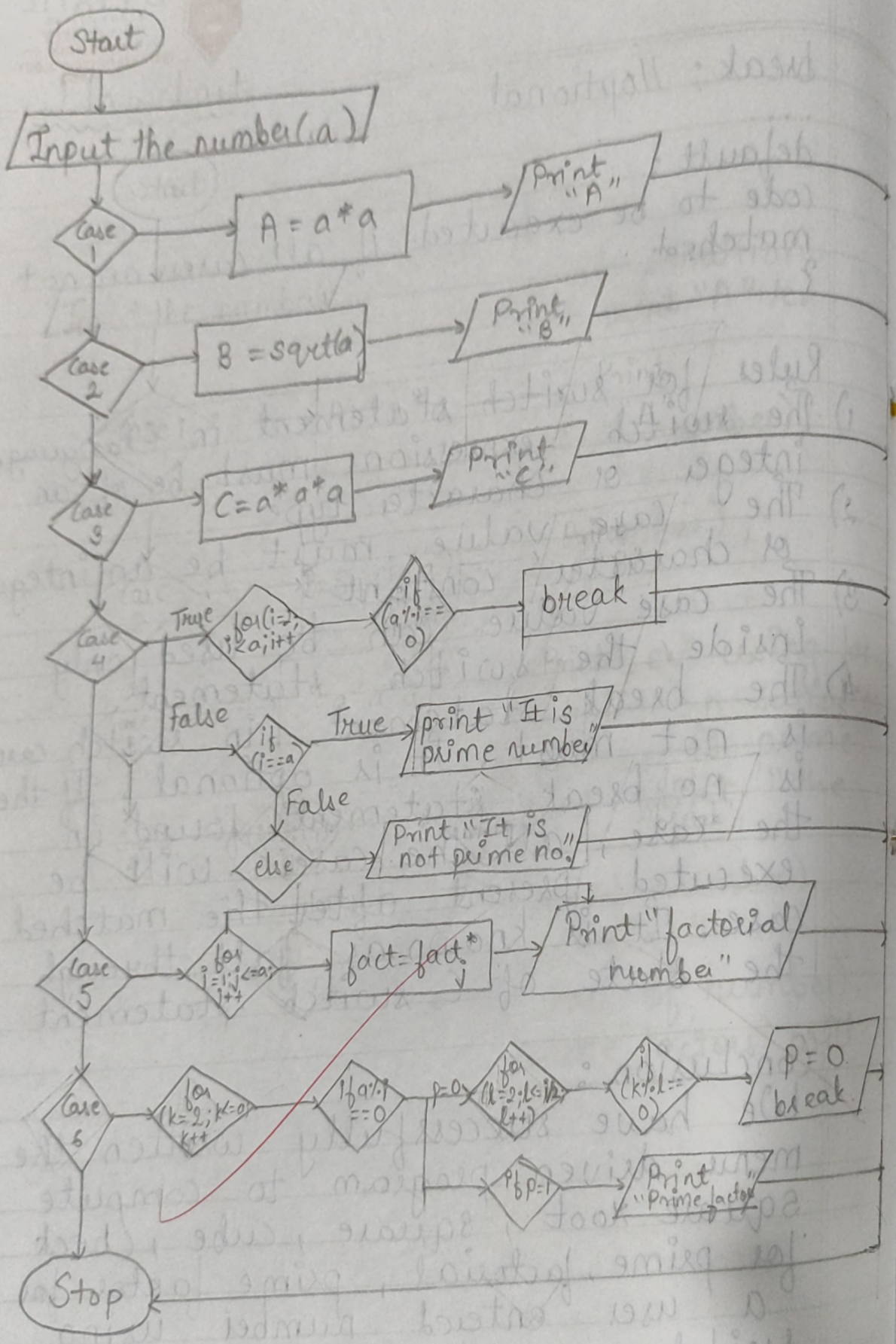
...

default :
code to be executed if all cases are not matched.
}

Rules for switch statement in C language
1) The switch expression must be of an integer or character type.
2) The case value must be an integer or character constant
3) The case value can be used only inside the switch statement.
4) The break statement in switch case is not must. It is optional. If there is no break statement found in the case, all the cases will be executed present after the matched case. It is known as fall through the state of C switch statement

Conclusion :
We have successfully written the menu driven program to compute square root, square, cube, check for prime, factorial, prime factor for a user entered number using switch.

Flow chart


Start
Input the number(a)
Case 1
A = a*a
Print "A"
Case 2
B = sqrt(a)
Print "B"
Case 3
C=a*a*a
Print "C"
Case 4
True
for(i=2; i<a; i++)
if (a%i==0)
break
False
if (i==a)
True
print "It is prime number"
False
else
Print "It is not prime no."
Case 5
for i=1; i<=a; i++
fact=fact*i
Print "factorial number"
Case 6
for (k=2; k<=a; k++)
if a%k==0
p=0
for (l=2; l<=k/2; l++)
if (k%l==0)
p=0
break
if p=1
Print "Prime factor"
Stop

File Edit Selection View Go Run Terminal Help 7mar.c - New folder (Workspace) - Visual Studio Code

7mar.c

New folder > C > C++ > 7mar.c > ...

```c
#include<stdio.h>
#include<math.h>
int main(){
    int a,i,p,fact=1,j,k,l,ch,A,B,C;
    printf("enter the number:");
    scanf("%d",&a);
    printf("enter the choice: 1.square root of number 2.square of number 3.cube of number  4.check for prime number 5.factorial of number 6
    scanf("%d",&ch);
    switch(ch){
    case 1:
    {A=a*a;
    printf("square of number:%d",A);
    break;}
    case 2:
    {B=sqrt(a);
    printf("square root of number:%d",B);
    break;}
    case 3:
    {C=a*a*a;
    printf("cube of number:%d",C);
    break;}
    case 4:
    {i=2;
    for(i=2;i<a;i++){
        if(a%i==0)
        break;
    }
    if(i==a){
        printf("it is prime number");
    }
```

0 △ 0 Watch Sass Ln 2, Col 17 Spaces: 4 UTF-8 CRLF C Go Live

Search

File Edit Selection View Go Run Terminal Help 7mar.c - New folder (Workspace) - Visual Studio Code

7mar.c

New folder > C > C++ > 7mar.c > ...

```c
        {C=a*a*a;
        printf("cube of number:%d",C);
        break;}
        case 4:
        {i=2;
        for(i=2;i<a;i++){
            if(a%i==0)
            break;
        }
        if(i==a){
            printf("it is prime number");
        }
        else{printf("it is not prime number");}
        break;}
        case 5:
        {for(j=1;j<=a;j++){fact=fact*j;}
        printf("the factorial of number :%d",fact);
        break;}
        case 6:
        {for(k=2;k<=a;k++){
            if (a%i==0){
                p=0;
                for(l=2;l<=i/2;l++){
                    if(k%l==0){
                        p=0;
                        break;
                    }
                }
                if(p==1){
```

File Edit Selection View Go Run Terminal Help 7mar.c - New folder (Workspace) - Visual Studio Code

7mar.c

New folder > C > C++ > 7mar.c > main()

```c
        }
        else{printf("it is not prime number");}
        break;}
        case 5:
        {for(j=1;j<=a;j++){fact=fact*j;}
        printf("the factorial of number :%d",fact);
        break;}
        case 6:
        {for(k=2;k<=a;k++){
            if (a%i==0){
                p=0;
                for(l=2;l<=i/2;l++){
                    if(k%l==0){
                        p=0;
                        break;
                    }
                }
                if(p==1){
                    printf("prime factors:",k);
                }
            }
        }
        break;}
        default:printf("wrong choice:");}
        return 0;
}
```

Watch Sass Ln 39, Col 21 Spaces: 4 UTF-8 CRLF C Go Live

Search

7mar.c - New folder (Workspace) - Visual Studio Code

File Edit Selection View Go Run Terminal Help

7mar.c

New folder > C > C++ > 7mar.c > main()

```c
        if(i==a){
            printf("it is prime number");
        }
        else{printf("it is not prime number");}
        break;}
        case 5:
        {for(j=1;j<=a;j++){fact=fact*j;}
        printf("the factorial of number :%d",fact);
        break;}
        case 6:
        {for(k=2;k<=a;k++){
            if (a%i==0){
                p=0;
                for(l=2;l<=i/2;l++){
                    if(k%l==0){
                        p=0;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\SUNIDHISHREE\New folder> cd "c:\Users\SUNIDHISHREE\New folder\C\C++\" ; if ($?) { gcc 7mar.c -o 7mar } ; if ($?) { .\7mar }
enter the number:2
enter the choice: 1.square root of number 2.square of number 3.cube of number  4.check for prime number 5.factorial of number 6.prime factors of number1
square of number:4
PS C:\Users\SUNIDHISHREE\New folder\C\C++> cd "c:\Users\SUNIDHISHREE\New folder\C\C++\" ; if ($?) { gcc 7mar.c -o 7mar } ; if ($?) { .\7mar }
enter the number:2
enter the choice: 1.square root of number 2.square of number 3.cube of number  4.check for prime number 5.factorial of number 6.prime factors of number2
square root of number:1
PS C:\Users\SUNIDHISHREE\New folder\C\C++> cd "c:\Users\SUNIDHISHREE\New folder\C\C++\" ; if ($?) { gcc 7mar.c -o 7mar } ; if ($?) { .\7mar }
enter the number:2
enter the choice: 1.square root of number 2.square of number 3.cube of number  4.check for prime number 5.factorial of number 6.prime factors of number3
cube of number:8
PS C:\Users\SUNIDHISHREE\New folder\C\C++>
```

D.Y. PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Grade :
Sign :
Date :

EXPERIMENT NO.

Name : Sunidhishree Prabakaran
Roll no : 22CE1116
Batch : B3
Branch : CE Core

Experiment.

Write a program to display patterns using for loop.

```
*                    1 2 3 4 5
* *                  1 2 3 4
* * *                1 2 3
* * * *              1 2
* * * * *            1
```

Theory.

A for loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

Syntax of for loop:

```
for (init; condition; increment){
        statement(s);
}
```

A pattern is defined as a design that repeats a specific number of times. The pattern can be decorative in real life, like, for example, a rangoli.

A pattern program in C helps improve the concept of looping and algorithms.

Algorithms

Pattern I:-

Step 1: Start
Step 2: Declare the variables
Step 3: Input the number of rows (n)
Step 4: Check for condition
(i) for (i=1 ; i<=n ; i++){
(ii) for (j=1 ; j<=i ; j++)
Step 5: If the condition (ii) is true then print ("*") and also condition (i) is true then print ("\n"), otherwise go to step 6
Step 6: Stop

Pattern II:-

Step 1: Start
Step 2: Declare the variable
Step 3: Input the number of rows (n)
Step 4: Check for condition
(i) for (i=n ; i>=1 ; i--)
(ii) for (j=1 ; j<=i ; j++)
Step 5: If the condition (ii) is true then print ("j") and also if condition (i) is true print ("\n") otherwise goto step 6
Step 6: Stop

D.Y. PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

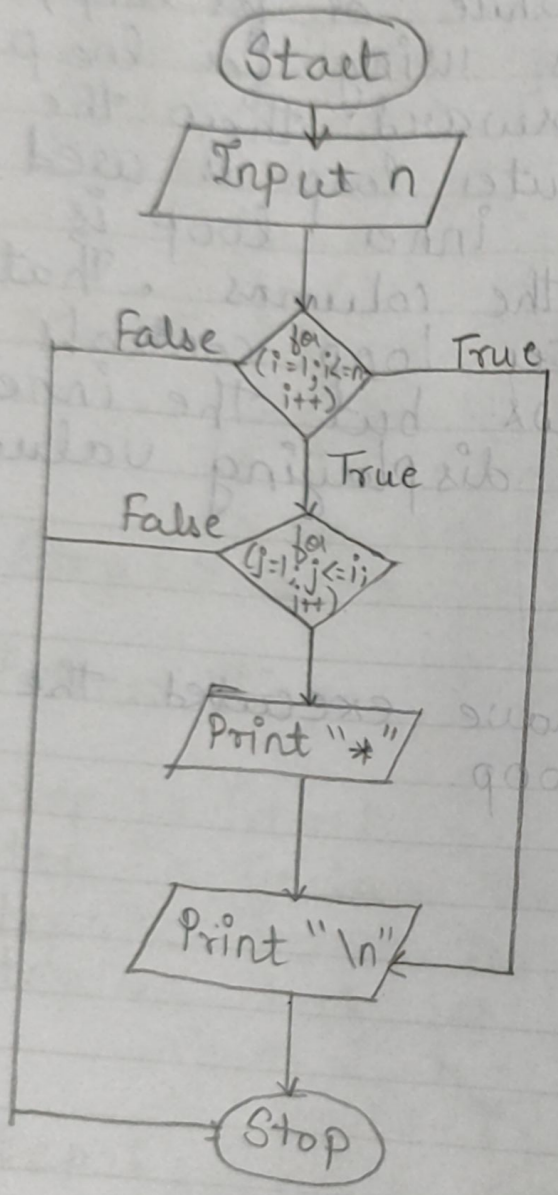
They are easily displayed with the help of nested loops. Loops can be made using the while or for loop, but writing programs using for loop is more straightforward than the while lopp. The outer loop is used for rows and the inner loop is used to display the columns. That is why the outer loop controls the number of rows but the inner loop is used for displaying values in those columns.

Conclusion

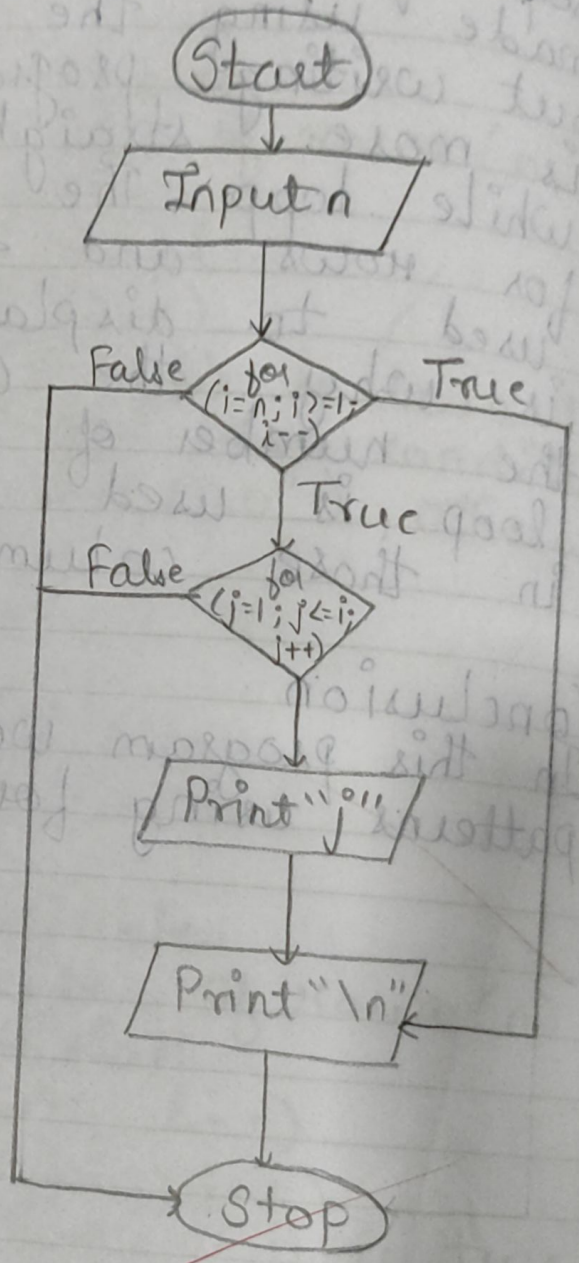
In this program we have executed the patterns using for loop.

Flowchart

Pattern I :-



Pattern II :-



File Edit Selection View Go Run Terminal Help

7mar.c - New folder (Workspace) - Visual Studio Code

C 7mar.c

New folder > C > C++ > C 7mar.c > main()

```c
#include <stdio.h>
int main()
{
int i, j, n;
printf("Enter the number of rows: ");
scanf("%d", &n);
for(i = 1; i <= n; i++)
{
    for(j = 1; j <= i; j++)
    {
        printf("* ");
    }
    printf("\n");
}
return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\SUNIDHISHREE\New folder> cd "c:\Users\SUNIDHISHREE\New folder\C\C++\" ; if ($?) { gcc 7mar.c -o 7mar } ; if ($?) { .\7mar }
Enter the number of rows: 5
*
* *
* * *
* * * *
* * * * *
PS C:\Users\SUNIDHISHREE\New folder\C\C++>
```

Watch Sass Ln 12, Col 6 Spaces: 4 UTF-8 CRLF C Go Live

7mar.c

New folder > C > C++ > 7mar.c > ...

```c
#include <stdio.h>
int main()
{
int i, j, n;
printf("Enter the number of rows: ");
scanf("%d", &n);
for (i = n; i >= 1; i--)
{
    for (j = 1; j <= i; j++)
    {
        printf("%d",j);
    }
    printf("\n");
}
return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\SUNIDHISHREE\New folder> cd "c:\Users\SUNIDHISHREE\New folder\C\C++\" ; if ($?) { gcc 7mar.c -o 7mar } ; if ($?) { .\7mar }
Enter the number of rows: 5
12345
1234
123
12
1
PS C:\Users\SUNIDHISHREE\New folder\C\C++>
```

D.Y. PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Grade :
Sign :
Date :

EXPERIMENT NO.

A+

Name : Suridhishree Prabakaran.
Roll no : 22CE1116
Batch : B3
Branch : CE Core

D.Y. PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Experiment

Write a program to accept the number of terms and find the sum of sine series

Theory :-
Sinx is a series of sin function of trigonometry ; it can expand up to infinite number of terms. Through this series, we can find out value of sin x at any radian value of sinx graph.

Sin x series

$$\sin x = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1}$$

$$= x - \frac{x^3}{3!} + \frac{x^5}{5!}$$

Conclusion:-
In this program we have calculated the sine series using for loop

Algorithm

Step 1: Start

Step 2: Input the no. of terms & angle in degree

Step 3 :- Check for condition

for (i = 2 ; i <= n ; i+=2)

If condition is true, then fact = fact * i * (i-1)

Step 4 :- Check for condition

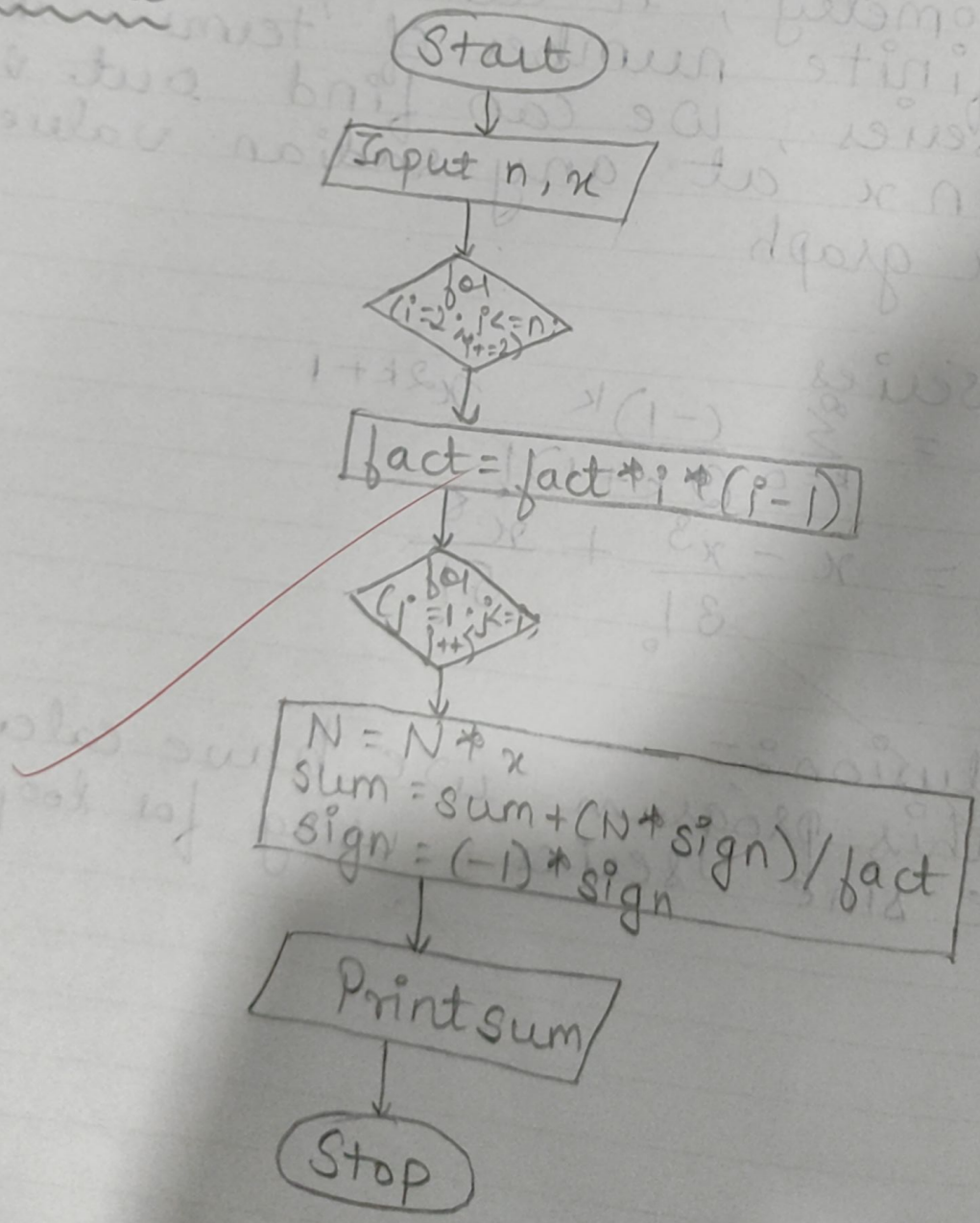
for (j = 1 ; j <= i ; j++)

If condition is true, N = N * x, sum = sum + (N * sign) / fact, sign (-1) * sign

Step 5 :- Print sum

Step 6 :- Stop

Flowchart



```
#include<stdio.h>
#include<math.h>
int main()
{
int i,j,N=1,n,fact=1,sum=0,x,sign=1;
printf("enter the number of terms:");
scanf("%f",&n);
printf("enter the angle in degrees:");
scanf("%f",&x);
for (i=2;i<=n;i+=2){
fact=fact*i*(i-1);
for(j=1;j<=i;j++)
N=N*x;}
sum=sum+(N*sign)/fact;
sign=(-1)*sign;


printf("the addition of numbers:%f",sum);
return 0;
```

```
guest-dwerm2@student-HP-Pro-3330-MT: ~/Desktop/22ce1116
guest-dwerm2@student-HP-Pro-3330-MT:~$ cd Desktop
guest-dwerm2@student-HP-Pro-3330-MT:~/Desktop$ cd 22ce1116
guest-dwerm2@student-HP-Pro-3330-MT:~/Desktop/22ce1116$ gcc abc.c
guest-dwerm2@student-HP-Pro-3330-MT:~/Desktop/22ce1116$ ./a.out
enter the number of terms:4
enter the angle in degrees:30
the addition of numbers:-nanguest-dwerm2@student-HP-Pro-3330-MT:~/Desktop/22ce1
6$ gcc abc.c
guest-dwerm2@student-HP-Pro-3330-MT:~/Desktop/22ce1116$ ./a.out
enter the number of terms:4
the angle in degrees:30
the addition of numbers:-30374550.000000guest-dwerm2@student-HP-Pro-3330-MT:~/
ktop/22ce1116$
```

D.Y. PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Grade:
Sign:
Date:

EXPERIMENT NO.

Name: Sunidhishree Prabakaran
Roll no: 22CF1116
Batch: B9
Branch: CE Core

D.Y. PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Experiment

Write a menu driven program to driven the following operations:

a. Perform addition of 3x3 matrices
b. Obtain transpose of given 3x3 matrices
c. Multiplication of 3x3 matrices

Theory

An array is defined as the collection of similar type of data items stored at contiguous memory locations. Arrays are the derived data type in C programming language which can store the primitive type of data such as int, char, double, float, etc.

Advantage of C array

1. Code optimization : Less code to access the da
2. Ease of transversing: By using the for loop, we can retrieve the elements of an array easily
3. Ease of sorting : To sort the elements of the array, we need few lines of code on
4. Random access : We can access any element randomly using the array

Disadvantage of C array

Fixed size : Whatever size, we define at th

Experiment

Write a menu driven program to driven the following operations:

a. Perform addition of 3x3 matrices
b. Obtain transpose of given 3x3 matrices
c. Multiplication of 3x3 matrices

Theory

An array is defined as the collection of similar type of data items stored at contiguous memory locations. Arrays are the derived data type in C programming language which can store the primitive type of data such as int, char, double, float, etc.

Advantage of C array

1. Code optimization : less code to access the data
2. Ease of transversing: By using the for loop, we can retrieve the elements of an array easily
3. Ease of sorting : To sort the elements of the array, we need few lines of code only
4. Random access : We can access any element randomly using the array

Disadvantage of C array

Fixed size : Whatever size, we define at the

Algorithm :-

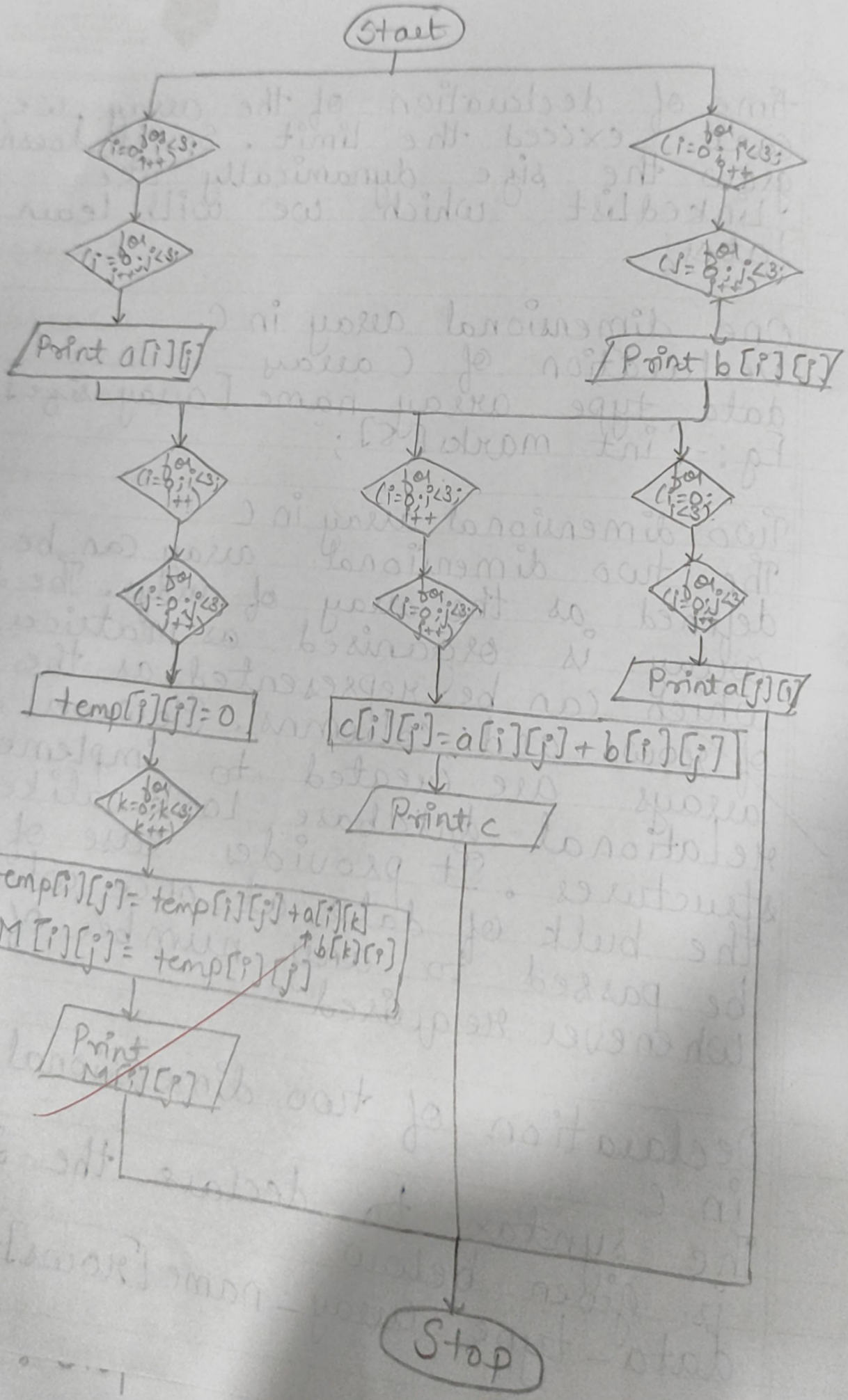
Step 1: Start
Step 2: Declare the variables
Step 3 : Using for loop, print matrix a and b
Step 5: For addition of matrices, check for condition
(i) for (i = 0 ; i < 3 ; i++)
(ii) for (j = 0 ; j < 3 ; j++)
Step 6 :- If condition of for loop (i) & (ii) is true then
c[i][j] = a[i][j] + b[i][j]
print (c[i][j])
Step 7 : For multiplication of matrices, check for condition
(i) for (i = 0 ; i < 3 ; i++)
(ii) for (j = 0 ; j < 3 ; j++)
(iii) for (k = 0 ; k < 3 ; k++)
Step 8 :- If condition of for loop (i), (ii), (iii) is true then
temp[i][j] = temp[i][j] + a[i][k] * b[k][j]
M[i][j] = temp[i][j]
print (M[i][j]) using for loop.
Step 9 :- For transpose of matrix, check for condition
(i) for (i = 0 ; i < 3 ; i++)
(ii) for (j = 0 ; j < 3 ; j++)
Step 10: If condion (i) & (ii) is true then print
print a[j][i]
Step 11 : Stop

time of declaration of the array, we can't exceed the limit. So, it doesn't grow the size dynamically like LinkedList which we will learn later.

One dimensional array in C.
Declaration of C array
data_type array_name [array_size];
Eg:- int marks[5];

Two dimensional array in C
The two dimensional array can be defined as the array of arrays. The 2D array is organised as matrices which can be represented as the collection of rows and columns. However, 2D arrays are created to implement a relational database look alike data structures. It provides ease of holding the bulk of data at once which can be passed to any number of function whenever required.

Declaration of two dimensional array in C.
The syntax to declare the 2D array is given below
data_type array_name [rows] [columns];


Start
for (i=0; i<3; i++)
for (j=0; j<3; j++)
Print a[i][j]
for (i=0; i<3; i++)
for (j=0; j<3; j++)
Print b[i][j]
for (i=0; i<3; i++)
for (j=0; j<3; j++)
temp[i][j]=0
for (k=0; k<3; k++)
temp[i][j]=temp[i][j]+a[i][k]*b[k][j]
M[i][j]=temp[i][j]
Print M[i][j]
for (i=0; i<3; i++)
for (j=0; j<3; j++)
c[i][j]=a[i][j]+b[i][j]
Print c
for (i=0; i<3; i++)
for (j=0; j<3; j++)
Print a[j][i]
Stop

D.Y. PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Eg:- int two dimen [4].[3];

Conclusion:
In this program, we have performed addition, multiplication & transpose of matrix.

Terminal

```c
#include<stdio.h>
int main(){
int a[3][3],b[3][3],c[3][3],d[3][3],temp[3][3],M[3][3],i,j,k;
printf("enter the matrix a :");
for (i=0;i<3;i++){
for(j=0;j<3;j++){
scanf("%d",&a[i][j]);}}
printf("The matrix a :\n");
for (i=0;i<3;i++){
for(j=0;j<3;j++){
printf("%d\t",a[i][j]);}
printf("\n");}


printf("enter the matrix b :");
for (i=0;i<3;i++){
for(j=0;j<3;j++){
scanf("%d",&b[i][j]);}}
printf("The matrix b :\n");
for (i=0;i<3;i++){
for(j=0;j<3;j++){
printf("%d\t",b[i][j]);}
printf("\n");}

//addition of matrices
printf("The matrix addition  :\n");
for (i=0;i<3;i++){
for(j=0;j<3;j++){
c[i][j]=a[i][j]+b[i][j];
printf("%d\t",c[i][j]);}
printf("\n");}


//multiplication of matrices
printf("The matrix multiplication:\n");
for (i=0;i<3;i++){
for(j=0;j<3;j++){
temp[i][j]=0;
for(k=0;k<3;k++){
temp[i][j]=temp[i][j]+a[i][k]*b[k][i];}
M[i][j]=temp[i][j];}}
for (i=0;i<3;i++){
for(j=0;j<3;j++){
printf("%d\t",M[j][i]);}
printf("\n");}



//tanspose of matrix
printf("the transpose of matrix a:\n");
for (i=0;i<3;i++){
for(j=0;j<3;j++){
printf("%d\t",a[j][i]);}
printf("\n");}
return 0;}
```

```
guest-qwsrno@student-HP-Pro-3330-MT: ~/Desktop/sunidhi
guest-qwsrno@student-HP-Pro-3330-MT:~$ cd Desktop
guest-qwsrno@student-HP-Pro-3330-MT:~/Desktop$ cd sunidhi
guest-qwsrno@student-HP-Pro-3330-MT:~/Desktop/sunidhi$ gcc ce.c
guest-qwsrno@student-HP-Pro-3330-MT:~/Desktop/sunidhi$ ./a.out
enter the matrix a :1
2
3
4
5
6
7
8
9
The matrix a :
1       2       3
4       5       6
7       8       9
enter the matrix b :9
8
7
6
5
4
3
2
1
The matrix b :
9       8       7
6       5       4
3       2       1
The matrix addition  :
10      10      10
10      10      10
10      10      10
The matrix multiplication:
30      69      90
30      69      90
30      69      90
the transpose of matrix a:
1       4       7
2       5       8
3       6       9
guest-qwsrno@student-HP-Pro-3330-MT:~/Desktop/sunidhi$
```

D.Y. PATIL DEEMED TO BE UNIVERSITY
RAMRAO ADIK INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Grade:
Sign:
Date:

EXPERIMENT NO.

A

Name: Sunidhishree Prabakaran
Batch: B3
Roll no: 22CE1116
Branch: CE Core.

D.Y. PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Experiment -

Write a program to accept from user the number of Fibonacci numbers to be generated and print the Fibonacci series using recursion.

Theory

A function is a group of statements that together perform a task. Every C program has atleast one function which is main() and all the most trivials can define additional functions.

For writing program using function we required three steps as follows

1. Defining a function

Syntax

```
return-type function_name (parameter list)
{
    body of the function
}
```

2. Function Declarations

A function declaration tells the compiler about a function name and how to call the function. The actual body

Algorithms :-

Step 1 : Start
Step 2 : Declare the variables
Step 3 : Input the Limit
Step 4 : Assign a = 0, b = 1 and print a, b
Step 5 : Check for condition
for (i=1 ; i<=m ; i++)
If the condition, is true then
c = a + b ; a = b ; b = c ; and print (c)
Step 6 : Stop

D.Y. PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

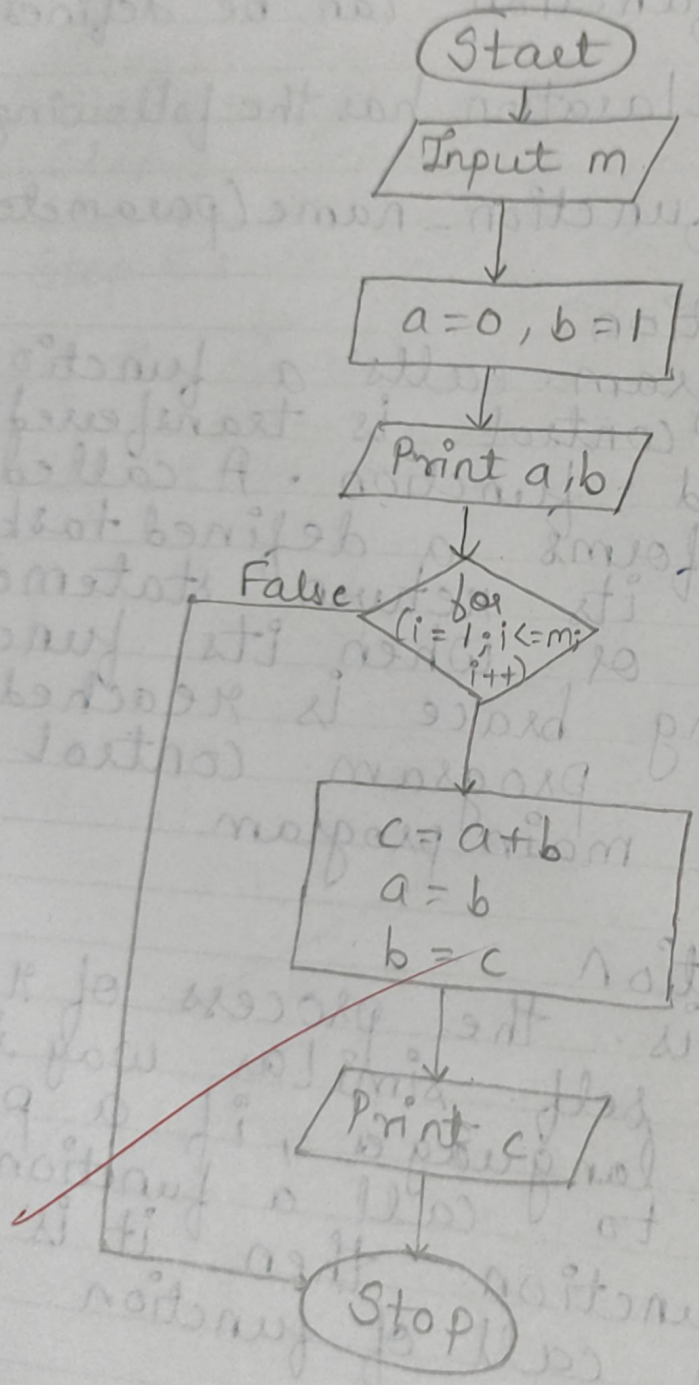
body of the function can be defined separately.
A function declaration has the following part:-
return_type function_name(parameter list)

3. Calling a function
When a program calls a function, the program control is transfered to the called function. A called function performs a defined task and when its return statement is executed or when its function-ending closing brace is reached, it returns the program control back to the main program.

Recursive function
Recursion is the process of repeating items in a self similar way in programming language, if a program allows you to call a function inside the same function then it is called a recursive call of function.

```
void recursion()
{
  recursion();  /*function calls itself*/
}
```

Flowchart :-


Start
Input m
a=0 , b=1
Print a,b
False
for (i=1;i<=m; i++)
c=a+b
a=b
b=c
Print c
Stop

D.Y. PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

```
int main ()
{
  recursion ();
}
```

Conclusion.
In this program. we have executed Fibonacci series using recursion.

File Edit Selection View Go Run Terminal Help 7mar.c - New folder (Workspace) - Visual Studio Code

7mar.c

New folder > C > C++ > 7mar.c > main()

```c
#include <stdio.h>
int main()
{
int n, a, b, c, i;
printf("Enter limit:");
scanf("%d",&n);
a=0;
b=1;
printf("\n%d\t%d\t",a,b);
for(i=1;i<=n;i++)
{
    c=a+b;
    a=b;
    b=c;
    printf("%d\t",c);
}
return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\SUNIDHISHREE\New folder> cd "c:\Users\SUNIDHISHREE\New folder\C\C++\" ; if ($?) { gcc 7mar.c -o 7mar } ; if ($?) { .\7mar }
Enter limit:5

0       1       1       2       3       5       8
PS C:\Users\SUNIDHISHREE\New folder\C\C++>
```

Watch Sass Ln 16, Col 1 Spaces: 4 UTF-8 CRLF C Go Live Win32