SNS COLLEGE OF TECHNOLOGY

Coimbatore-36.

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE NAME : 23CST101-PROBLEM SOLVING & C PROGRAMMING

I YEAR/ I SEMESTER

UNIT – I INTRODUCTION TO PROBLEM SOLVING TECHNIQUES

Topic: Simple Strategies for Developing Algorithm

Mrs.Papithasri K

Assistant Professor

Department of Computer Science and Engineering

They are two commonly strategies used in developing algorithm
1. Iteration
2. Recursion

Iteration:

The iteration is when a loop repeatedly executes till the controlling condition becomes false.

The iteration is applied to the set of instructions which we want to get repeatedly executed.

Iteration includes "initialization, condition, and execution" of statement within loop and update (increments and decrements) the control variable.

A sequence of statements is executed until a specified condition is true is called iterations.
1. for loop
2. While loop

for loop

Syntax for For:	Example: Print n natural numbers

FOR(*start-value* to *end-value*) DO
   statement
... ENDFOR
	BEGIN
	GET n
	INITIALIZE i=1
	FOR (i<=n)DO
	    PRINT i
	    i=i+1
	ENDFOR
	END

```c
/* C Program to Print Natural Numbers from 1 to N using For Loop */

#include<stdio.h>

int main()
{
    int Number, i;

    printf("\n Please Enter any Integer Value  : ");
    scanf("%d", &Number);

    printf("\n List of Natural Numbers from 1 to %d are \n", Number);
    for(i = 1; i <= Number; i++)
    {
        printf(" %d \t", i);
    }

    return 0;
}
```

```
C:\Users\Suresh\Documents\C Programs\NNumber1.exe                  —    □    ✕

Please Enter any Integer Value  : 5                    ©tutorialgateway.org

List of Natural Numbers from 1 to 5 are
1          2          3          4          5
```

while loop

Syntax for While:	Example: Print n natural numbers

WHILE (condition) DO
   statement
 	
++
 	
ENDWHILE
	BEGIN
	GET n
	INITIALIZE i=1
	WHILE(i<=n) DO
	   PRINT i
	   i=i+1
	ENDWHILE
	END

```c
/* C Program to Print Natural Numbers from 1 to N using While Loop */

#include<stdio.h>

int main()
{
    int Number, i = 1;

    printf("\n Please Enter any Integer Value  : ");
    scanf("%d", &Number);

    printf("\n List of Natural Numbers from 1 to %d are \n", Number);
    while(i <= Number)
    {
    printf(" %d \t", i);
    i++;
    }

    return 0;
}
```

```
Please Enter any Integer Value   : 15

 List of Natural Numbers from 1 to 15 are
 1        2        3        4        5        6        7        8        9        10       11
```

Simple Strategies For Developing Algorithm /19CST101 Programming
for Problem Solving/Papithasri/ CSE/ SNSCT

Flow chart for (for loop & while loop)

Simple Strategies For Developing Algorithm /19CST101 Programming for Problem Solving/Papithasri/ CSE/ SNSCT
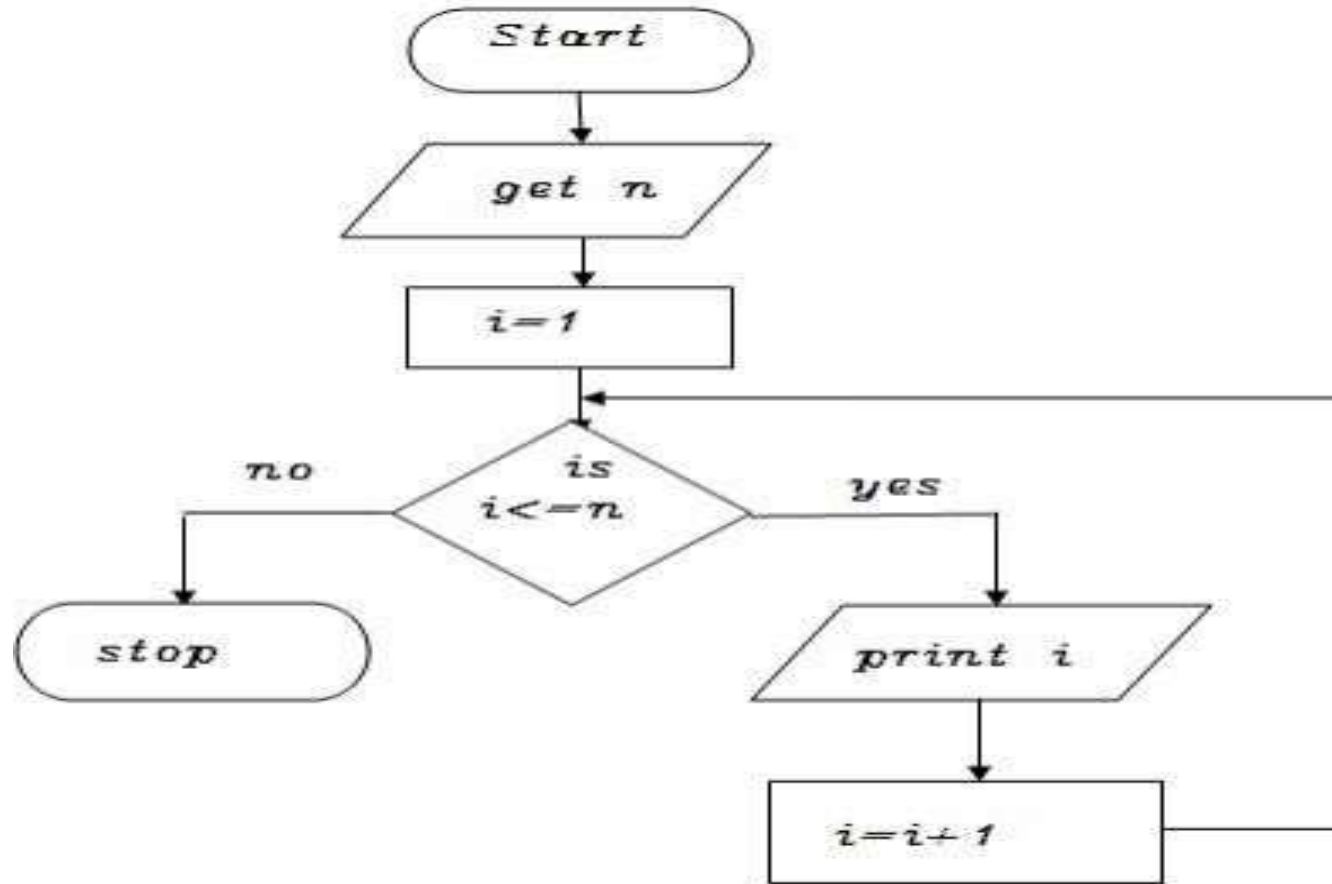
Recursions:

A function that calls itself is known as recursion.

Recursion is a process by which a function calls itself repeatedly until some specified condition has been satisfied.

Algorithm for factorial of n numbers using recursion

Main function:

        Step1: Start

        Step2: Get n

        Step3: call factorial(n)

        Step4: print fact

        Step5: Stop

Sub function factorial(n):

        Step1: if(n==1) then fact=1 return fact

        Step2: else fact=n*factorial(n-1) and return fact
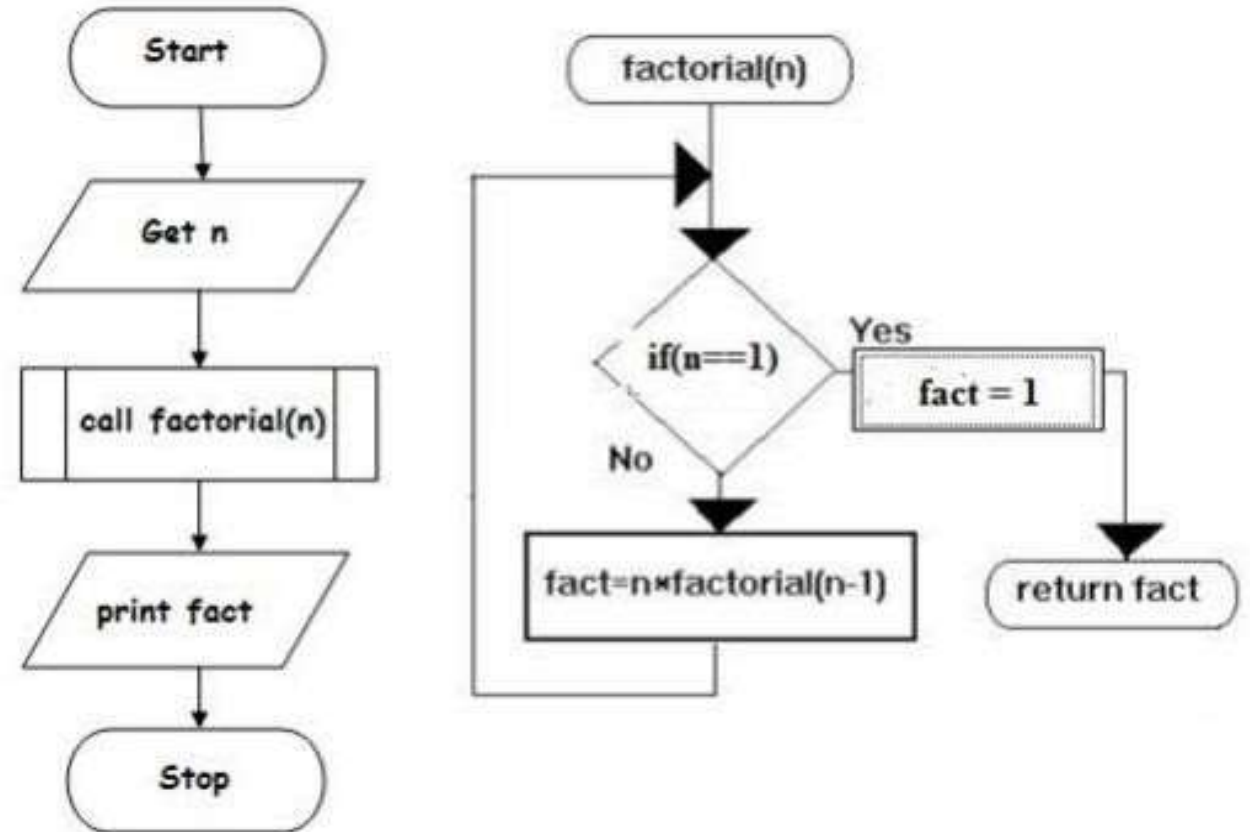
Pseudo code for factorial using recursion:

Main function:
    BEGIN
    GET n
    CALL factorial(n)
    PRINT fact
    END

Sub function factorial(n):
    IF(n==1) THEN
        fact=1
        RETURN fact
    ELSE
        RETURN fact = n * factorial (n - 1)

Recursion

```c
#include<stdio.h>
#include<conio.h>
void main()
{
int n=0,f=0;
printf("enter the number");
scanf("%d",&n);
f=fact(n);
printf("factorial of %d is %d",n,f);
}
int fact(int n)
{
 if(n==1)
 return 1;
 else
  return(n*fact(n-1));
 }
```

```
C:\Users\Ad\Documents\test.exe

enter the number 6
factorial of 6 is 720
----------------------------------
Process exited after 2.411 seconds with return value 21
Press any key to continue . . . _
```

Thank you