

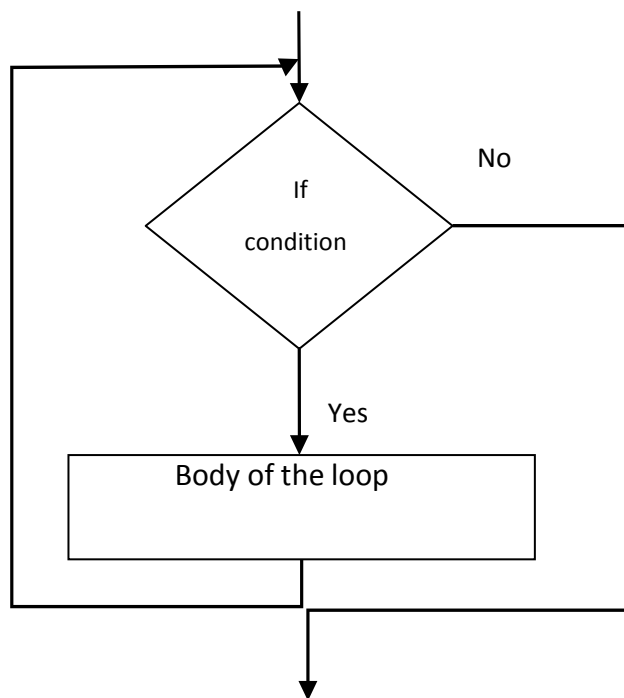
SIMPLE STRATEGIES FOR DEVELOPING ALGORITHMS

Iteration and **Recursion** are the simple strategies for developing algorithms.

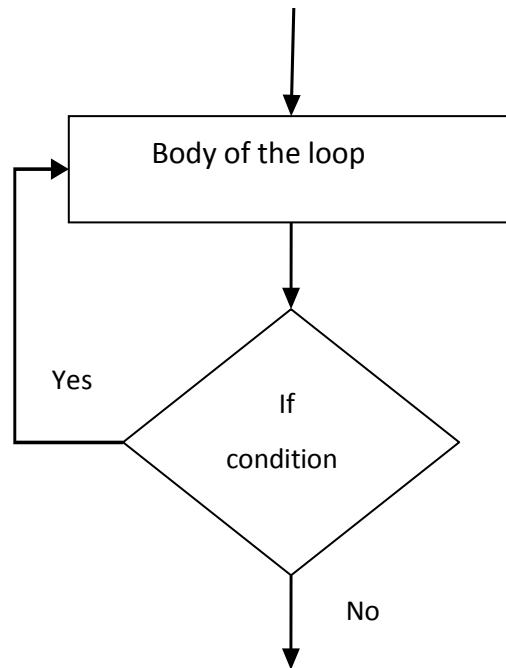
Iteration

Iteration is the process of repeating the same set of statements again and again until the condition becomes false. Iteration is done in two ways.

- i) In iteration loop, the condition is checked at the beginning of the loop. If the condition is true, then the loop will be executed. If the condition is false, the loop will not be executed.



- ii) Condition is checked at the bottom of the loop. If the condition is true, loop will be executed. Otherwise the loop will not be executed.

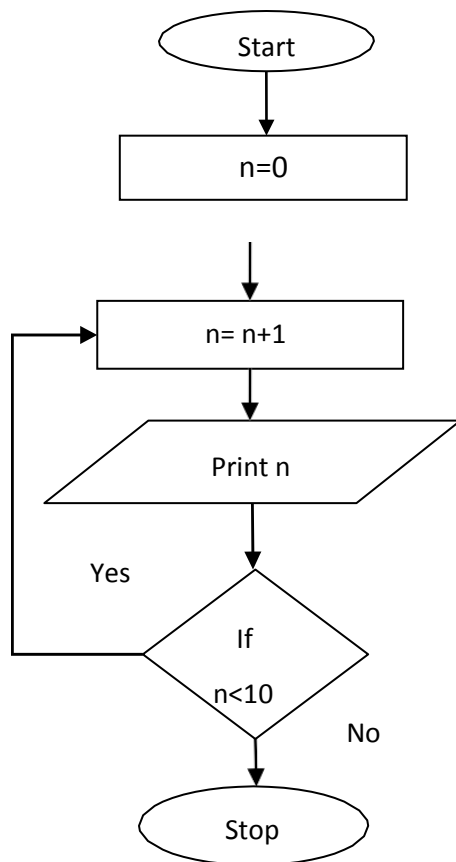


Example: Write an algorithm, pseudo code and draw the flowchart to print the numbers from 1 to 10.

Algorithm:

1. Start
2. Initialize $n=0$
3. Increment n by 1
4. Print n

5. Check condition $n < 10$
 - i. If yes goto step 3
 - ii. Otherwise goto step 6
6. Stop

Flowchart:**Pseudo code:**

```
BEGIN
OBTAIN n
INITIALIZE n=0

INCREMENT
n=n+1
DISPLAY n

IF n<10 REPEAT the loop
```

ENDIF

END

Recursion

Recursion is a programming technique that has a recursive function that calls itself again and again until the condition is reached.

Syntax:

function():

function():

In the above syntax, function() is a function which call itself until the condition is reached.

Difference between recursion and iteration

Sl.No	Recursion	Iteration
1	Function calls itself until the base condition is reached.	Repetition of loop until condition false.
2	It keeps code short and simple.	It keeps the code longer.
3	It is slower due to overhead of maintaining the stack	It is faster.
4	It takes more memory.	It takes less memory.
5	Tracing is difficult if any problem occurs.	Tracing is easier if any problem occurs.

6.	It is a process always applied to a function	It is applied to the set of instructions which we want to get repeatedly executed .
----	---	--

Example: Calculating factorial of a number using recursion.

Algorithm:

1. Start
2. Read n
3. Call the function fact(n)
4. Print value of the factorial
5. Stop

fact(n) function:

1. Start function
2. If n=0, return 1
3. If n>0 return function fact(n)
4. Return

Flowchart :

The recursive function used to calculate factorial of a number is

$$\text{fact}(n) = \begin{cases} \text{fact}(n) = 1 & \text{if } n=0 \\ n * \text{fact}(n-1) & \text{if } n>0. \end{cases}$$

