Chapter 4 : Actions of Repetition (Iterative) (Loops)

Introduction :

In computer programming and algorithms, iteration refers to the process of repeatedly executing a block of code based on a certain condition. It allows you to perform repetitive tasks efficiently. Iterations (loops) allow the repetition of the same set of instructions multiple times.

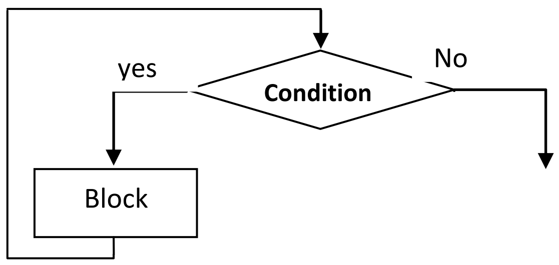
There are three primary types of iterative instructions:

1. Iterative Action "While":

It allows the repetition of one or more actions as long as the condition remains true.

Syntax:

```
While (condition) Do
   <Action Block>
End while
```



It involves repeating the execution of the instructions in the <Action Block> as long as the condition is met and stopping their execution as soon as the condition becomes false.

Example:

Write an algorithm that calculates the sum of the first n positive integers?

```
Algorithm sum-int1
        n ,i,som: integer;
Begin
    Read(n) ;
    Som← 0 ;            //the su mis initialized to 0
    i←0 ;
   while (i≤ n) do
        Som←Som + i ;
        i←i + 1;                  // incrementation of i
    end while
    write(som) ;
 end
```

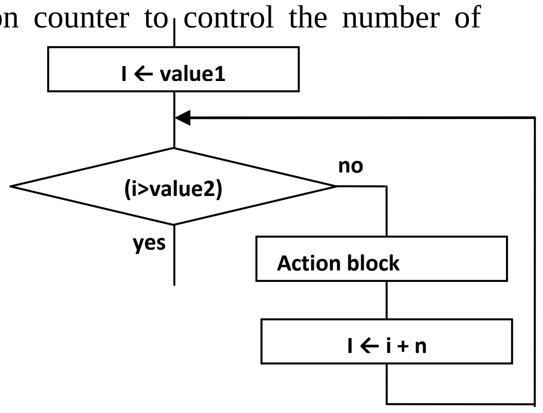
Note: The "While" iterative action is used when you do not know in advance the number of repetitions, and it is possible to not execute the actions of the block at all.

2. Iterative Action "For":

It uses a variable called a control variable or iteration counter to control the number of iterations.

Syntax:

```
For i going from Value1 to Value2 (step = n) do
   <Action Block>
End For
```



This action iterates the variable i from Value1 to Value2 with a step size of n, executing the instructions in the <Action Block> for each iteration.

Example : We will take the previous example and implement it using the "for" loop:

```
Algorithm som-int
     n , i ,som : integer;
Begin
  read(n) ;
  Som← 0 ;
  for i←1 to  n do
       Som←Som + i ;
  End for
 Write (som) ;
End.
```

```
#include <iostream>
using namespace std;

int main() {
   int n, i, sum;

   cout << "Enter a positive integer (n): ";
   cin >> n;

   sum = 0;
   for (i = 1; i <= n; i++) {
      sum += i;
   }

   cout << "The sum of the first " << n << "
positive integers is: " << sum << endl;

   return 0;
}
```

This C++ program reads a positive integer n, calculates the sum of the first n positive integers, and then displays the result.

Remarks:

- The "for" loop is used when the number of iterations is known in advance. You need to specify the initial value, final value, and optionally the step size.

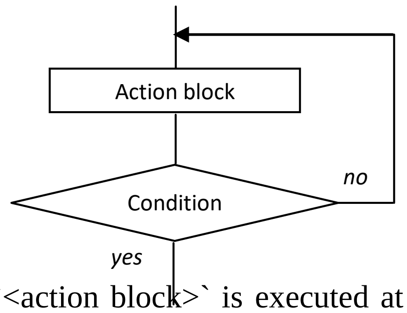
- The loop counter is automatically incremented. If the step size is not specified, it defaults to 1.

3. The iterative repeat action :

The "repeat" action is similar to the "do...while" loop in many programming languages. It repeats a block of actions until a certain condition is met. Here is the syntax in an algorithmic form:

Syntax :

```
Repeat
<Action block>
Until  (condition) ;
```



In this construct, the block of actions within `<action block>` is executed at least once, and then the condition is checked. If the condition is false, the block is executed again, and this process continues until the condition becomes true.

Here's an example in an algorithmic form:

```
repeat
read (x)
until  (x > 0)
```

In this example, the program keeps reading the value of `x` until the condition `x > 0` becomes true. This ensures that the program reads a positive value for `x`. You can think of it as "Repeat the actions until the condition is satisfied."

In C++, the equivalent loop is the "do...while" loop:

```
#include <iostream>
using namespace std;
int main() {
   int x;
   do {
      cout << "Enter a positive number: ";
      cin >> x;
   } while (x <= 0);

   cout << "You entered a positive number: " << x << endl;

   return 0;
}
```

In this C++ program, we use a "do...while" loop to repeatedly ask the user to enter a positive number until they do so. Once a positive number is entered, the loop terminates.

<u>Example :</u>

here is an example of how you can calculate the sum of the first N positive integers using the "repeat" (do...while) loop in an algorithmic form:

```
Algorithm som-int4
        n , i ,som:integer ;
begin
read (n) ;
   Som← 0 ;     // the sum is initialized to 0
    i← 0 ;         // initialization of i
    Repeat
        Som← Som + i ;
        i←i + 1;     // increment of i
   until (i> n)
   write (som) ;
end.
```

```
#include <iostream>
using namespace std;

int main() {
   int n, i = 1, som = 0;
   cout << "Enter a positive integer: ";
   cin >> n;

   do {
      som += i;
      i++;
   } while (i <= n);

   cout << "The sum of the first " << n << "
positive integers is: " << som << endl;
   return 0;
}
```