

Lesson 17: Article 5: Recursion

So far, iterative looping has been our tool for looping through several known number of times. It is also easy to implement.

In recursion, what we have is the repetitive calling of a particular module by the very same module. Still don't get what recursion is? See Recursion ;) - that's recursion. This requires vast use of the stack, and if recursion is not properly managed, it would result in stack overflow. Stack size should never be exceeded.

Recursion is not that simple to write. It should be carefully designed beforehand and tested appropriately. It also requires more system resources than the usual iterations since every module call stays resident in memory until the whole recursion backtracks. Technically speaking, backtracking means the endpoint of a recursion. The recursion then starts popping out the stored routines and completes all the necessary operations.

Recursion Example: Factorial

Factorial is a mathematical computation which for instance, the factorial of n is the multiplication of $n * n-1 * n-2 * \dots * 0$. In factorial, multiplying by 0 results in 1, ie. $\text{factorial}(0) = 1$.

Let us now see how we do it iteratively (the ordinary way):

```
Function Factorial(n : Integer) : LongInt;
Var
    Result : LongInt;
    i : Integer;

Begin
    Result := n;
    If (n <= 1) Then
        Result := 1
    Else
        For i := n-1 Downto 1 do
            Result := Result * i;
        Factorial := Result;
    End;
```

This is the iterative way. Now open your eyes and watch how we do it the recursive way:

```
Function Factorial(n : Integer) : Integer;
Var
    Result : Integer;

Begin
    If n = 1 Then
        Factorial := 1
    Else
        Factorial := n*Factorial(n-1);
    End;
```

As you can see, the difference is in the number of lines for both functions to perform the same operation. Recursion takes much less code but is harder to write and even more overall system resources (i.e. requires sufficient stack memory to operate appropriately) than iterations do. Iterations are easier to code.