Recursion Programming Exercises

There are no 'chips' in this one – you should aim to complete all of the tasks. The first four exercises are fairly straightforward, and then there are a couple of harder ones.

The Factorial Function

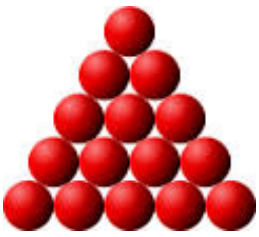
This can be calculated with the following recursive algorithm:

```
Function FactorialCalc(x)
   If x = 0 Then
        FactorialCalc ← 1
   Else
   FactorialCalc ← x * (FactorialCalc(x - 1))
End Function
```

Write a program to calculate the factorial of any number and test it. Remember that factorial values can get VERY big!

Snooker Ball Triangle

Write a recursive program to calculate the number of balls in a triangle like the ones used to set up a game of Snooker or Pool. The user should input the number of balls along one side of the triangle. Hint: A little thought should convince you that this is not that different to the previous exercise!



Text Chopping

The following procedure successively removes a letter from the end of a string.

```
Procedure Chop(AWord)
      Print AWord
      If the word length > 1 Then
         ChoppedWord ← AWord with the last character removed
         chop(ChoppedWord)
      Print AWord
End Procedure
```

You have already performed a dry run the above procedure with the word "Computing" passed to it.

Now write the program in VB and test it. In order to code it you will need to remember the string functions Left() and Len(). Does it behave as expected?

The B Procedure

Write a program to call the following procedure for an integer input. The functions DIV and MOD return the whole number and remainder part on division respectively. Mod exists as a function in VB. For whole number division use the "\" operator.

```
Procedure B (Number)
If (Number = 0) OR (Number = 1)
Then Print (Number)
Else
B (Number DIV 2)
Print (Number MOD 2)
EndIf
EndProcedure
```

Now test the program with the following integers: 10, 15, 127 & 256

Can you explain the output? Hint: If you have used the 'Writeline' method it will make more sense to read the numbers vertically!

The Fibonacci Sequence!

One possible recursive definition of the Fibonacci numbers is as follows:

$$F_n = F_{n-1} + F_{n-2},$$ Starting at n = 1, where $$F_1 = 1,\ F_2 = 1$$

Write a recursive function to calculate any Fibonacci number and then call it (iteratively, unless you want to use recursion for a simple loop) to output the first 20 numbers in the sequence.

Palindromes

Write a recursive routine to test if a string of any length is palindromic. Test it with a variety of words of different lengths (odd and even), including some palindromes.