EXAMPLE 6 – A Loop within a Loop

Problem: We will write a program to find all the prime numbers through 100. (Recall that an integer greater than one is prime if and only if it cannot be written as the product of two smaller integers that are themselves greater than 1.) For example 5 is prime; but 6 is not prime, because it can be written as 6 = 2*3.) But what we are really going to focus on here is a *method* for going about building this kind of structure – a loop within a loop. The key is to recognize that there are two separate things to be done, and do them one at a time. ***The important aspect of this example is not the subject matter but the divide-and-conquer strategy. Break a problem down into separate parts, solve each part individually, then put those solutions together.***

We have to find a way to determine if a number is prime. This will require a loop. Then, we have to have a loop to go through all the numbers from 2 to 100. Finally, we will put the two parts together.

First we will write some code to find out if a number X is prime. We assume that X is an integer greater than one. The strategy is to hunt for divisors. If we find one, the number is not prime. So we will start out assuming that the number is prime. Remember the trick for finding out if an integer is even? We will use the same trick here, but with a number other than 2

```
Declare TestDivisor as Integer
Declare Boolean IsPrime
Set IsPrime = TRUE

For ( TestDivisor = 2 Step 1 Until (X - 1) ) [See notes 1 and 2]
        If (TestDivisor*(X/TestDivisor) == X) IsPrime ==FALSE
End For
```

Now let's set that aside, and turn to the other problem: testing the numbers from 2 to 100. We can use a For loop for that too. Here is the code for that task:

```
Declare Counter as Integer
For (Counter = 2 Step 1 until 100)
        Find out whether Counter is prime; call the result IsPrime
        If (IsPrime)
                Print "The number " + Counter + " is prime"
        Else
                Print "The number " + Counter + " is not prime"
        EndIf
End For
```

Now all we have to do is put these two parts together, then check all the variables to make sure that the parts match up correctly. What we do is to replace *Find out whether Counter is prime; call the result IsPrime* with the work we did first. I'll highlight the replacement so that you can easily distinguish the outside loop from the inside one.

```
Declare Counter as Integer
For (Counter = 2 Step 1 until 100)
        //  Find out whether Counter is prime; call the result IsPrime
                Declare TestDivisor as Integer
                Declare Boolean IsPrime
                Set IsPrime = TRUE

                For ( TestDivisor = 2 Step 1 Until (X - 1) )
                        If (TestDivisor*(X/TestDivisor) == X) IsPrime ==FALSE
                End For
        If (IsPrime)
                Print "The number " + Counter + " is prime"
        Else
                Print "The number " + Counter + " is not prime"
        EndIf
End For
```

But we have a problem here: X has never been declared, and anyway we are not interested in testing some X but in testing Counter. So we just change X in the above to Counter. The result is

```
Declare Counter as Integer
For (Counter = 2 Step 1 until 100)
        //  Find out whether Counter is prime; call the result IsPrime
                Declare TestDivisor as Integer
                Declare Boolean IsPrime
                Set IsPrime = TRUE

                For ( TestDivisor = 2 Step 1 Until (Counter - 1) )
                        If (TestDivisor*(Counter/TestDivisor) == Counter) IsPrime ==FALSE
                End For
        If (IsPrime)
                Print "The number " + Counter + " is prime"
        Else
                Print "The number " + Counter + " is not prime"
        EndIf
End For
```

Note 1

The pseudocode style in For (Counter = 2 Step 1 until 100)
is very common and is derived from the BASIC programming language. It says: do whatever is to be done for each of each of the values 2, 3, 4, …, 100.

Note 2

Why are we starting at 2? Why are we stopping at X – 1?