## For-Next Loops:

In situations where you merely want to run the loop for a predefined number of times, it can be made by create and manage a counter for each loop, which might be called a For-Next Loop. This kind of loop allows you to specify a counter, to tell it to count from one number to another each time through the loop, and to exit once the counter has reached its upper limit. The syntax is as follow:

```
For I = (Integer) To (Integer)

  (Code to execute)

Next I
```

We used the variable name "I" above, as it is the most common name used for For-Loops; however, you can use any variable name you want, so long as the variable is of the type Integer. For more understanding let us try to write codes for the examples below:

Ex: Write a program to print on the form the numbers form (1 to 30) vertically, use the font (Arial), (Bold) with the size (14). Use For …Next statement.

1- Design part: No design part for such example.
2- Coding part: The coding part might be written as below:

```
Private Sub Form_Activate( )

Font.Name = "Arial"

Font.Bold = True

Font.Size = 10
```

```
For x = 1 To 30

Print "X = "; x

Next x

End Sub
```

In the example above, set its value to 1. Each time through the loop, the value of (x) was incremented by 1 until it reached (30), at which point the loop was executed.

## Exit For:

As with Do Loops, there is a statement that can be used to exit a For-Next loop, and it is called Exit For. Simply this statement anywhere within a For-Next loop will help to exit the loop.

## Step:

By default, the variable used in the declaration of the For-Next loop is incremented by 1 each time through the loop; however, if you want to increment this value by a different amount each time through the loop, you can simply append Step (Integer) to the end of the For-Next loop declaration. If, for instance, we wanted to print out every even number counting backward from -20 to 20, we could do this using the following code:

Ex: Write a program to print on the form the numbers form (-20 to 20) vertically, use the font (Arial), (Bold) with the size (12). Use For …Next statement.

1- Design part: No design part for such example.

2- Coding part: The coding part might be written as below:

```
Private Sub Form_Activate( )

Font.Name = "arial"

Font.Bold = True

Font.Size = 12

For x = -20 To 20 Step 2

Print "x = "; x

Next x

End Sub
```

<u>Ex:</u> Write a program that is able to solve the equation:

$$Y = 2000 - \frac{3^7}{9^5} + \frac{6^{11}}{18^6} + \frac{9^{15}}{27^7} + \frac{12^{19}}{36^8} + \frac{15^{23}}{45^9} + \frac{18^{27}}{54^{10}} + \frac{21^{31}}{63^{11}} + \cdots + \frac{36^{51}}{108^{16}}$$

Print the results on the form vertically, use the font (Arial), (Bold) with the size (14). Use For …Next statement.

3- Design part: No design part for such example.
4- Coding part: The coding part might be written as below:

```
Private Sub Form_Activate( )
```

```
Font.Name = "simplified arabic"

Font.Bold = True

Font.Size = 14

c = 5

d = 7

For i = 3 To 36 Step 3

x = (i ^ (d) / ((i * 3) ^ (c)))

Sum = Sum + x

Print "sum ="; Sum

c = c + 1

d = d + 4

Next i

y = 2000 - Sum

Print "y = "; y

End Sub
```