1. A simple way to gather user information is to use an InputBox function to get a string of input. The general format is:

```vba
strUserInput = InputBox(Prompt [, Title] [, Default] [, XPos] [YPos])
```

where stuff in the [ ] brackets are optional. The Default is the initial value in the InputBox with the default value of an empty string, and the center of the screen is used if no XPos or YPos are used. For example, the InputBox is shown for the call: `InputBox(“Enter the number of scores”, “Average Test Scores”)`

![InputBox Dialogue Box](image)

a) Why do you suppose usage of the InputBox is frowned upon?

b) Where do you suppose XPos = 0 and YPos = 0 is at on the screen?

2. Suppose I had a stack of test papers and I wanted to find out the class average. What steps would I do by hand to calculate the class average?
3. A loop in statement that causes a block of code to be repeated. Visual Basic has three types of loops: the Do While loop, the Do Until loop, and the For ... Next loop to use in different situations. The pretest format and flowchart of the Do While loop are:

```
Do While boolean condition
  statement1
  statement2
  ...
Loop
```

The condition is checked before the first loop body. If the condition is True, the loop body is executed and then the condition is rechecked. The loop continues to iterate until the condition is False when we drop out of the loop.

Complete the following example of a counter controlled loop to calculate the average of a set of test scores.

Public Class Form1

  Private Sub btnEnterData_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEnterData.Click
    Dim strUserInput As String
    Dim intNumberOfScores As Integer
    Dim intCounter As Integer
    Dim decScore As Decimal
    Dim decTotal As Decimal

    strUserInput = InputBox("Enter the number of scores", "Average Test Scores")
    If Integer.TryParse(strUserInput, intNumberOfScores) Then
      intCounter = 0
      Do While intCounter < intNumberOfScores
        strUserInput = InputBox("Enter a score")
        decScore = CDec(strUserInput)
        decTotal = decTotal + decScore
        intCounter = intCounter + 1
      Loop
      MessageBox.Show("The average score is ", (decTotal / intNumberOfScores).ToString("n1"))
    Else
      MessageBox.Show("Incorrect Number of Scores")
    End If
  End Sub
End Class
4. The posttest format of the Do While loop is:

\[
\text{Do}
\]
\[
\text{statement}_1
\]
\[
\text{statement}_2
\]
\[
\ldots
\]
\[
\text{Loop While boolean condition}
\]

The body of the loop is always done at least once before the condition is checked to see if the loop should iterate again. Draw the flowchart for the posttest version of the Do While loop.

5. Another type of loop is the Do Until loop. Its pretest and posttest formats are given below. Draw the flowchart for each version.

\[
\text{Do Until boolean condition}
\]
\[
\text{statement}_1
\]
\[
\text{statement}_2
\]
\[
\ldots
\]
\[
\text{Loop}
\]

\[
\text{Do}
\]
\[
\text{statement}_1
\]
\[
\text{statement}_2
\]
\[
\ldots
\]
\[
\text{Loop Until boolean condition}
\]

6. Using a Do Until loop how could we make the code in question 3 more robust to validate correctly formatted scores.
7. The **For** ... **Next** loop is useful in situations requiring a counter since it automatically initializes, tests, and increments the counter. The general format of the **For** ... **Next** loop is:

```plaintext
For CounterVariable = startValue To endValue [Step increment]
    statement_1
    statement_2
    . . .
Next  [CounterVariable]
```

Rewrite the loop from question 3 utilizing the **For** ... **Next** loop.