Lab-5: Complex Branching Control Structures

The purpose of this lab is to build upon the previous lab by creating more complex branching control structures.

Part I: Pollutants Table

1. This exercise requires the user to write **two macro**s.

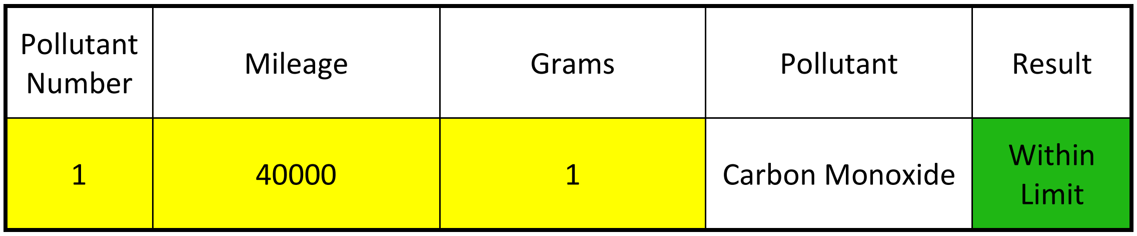
   a. Enter the following table into a new worksheet using B8 as the Upper-Left corner:

B8	Any pollutant amount is allowed for mileage above 100,000 miles	Mileage does not exceed:	Greater than 50,000 miles but does not exceed:	
Pollutant Number:	Pollutant:	50,000	100,000	
1	Carbon Monoxide	3.4	4.2	The Maximum Gram amounts allowed.
2	Hydrocarbons	0.31	0.39	
3	Nitrogen	0.4	0.5	
4	Non-methane Hydrocarbons	0.25	0.31	

   b. Code is provided to begin writing a macro for this exercise. Complete the given macro to report if a given amount of a pollutant for a specified mileage is within the limits shown in the table.

   c. Note that any pollutant amount is acceptable for an odometer reading above 100,000 miles

d. The “user supplied inputs” and “macro output results” should be displayed in a second table as shown here:



Pollutant Number	Mileage	Grams	Pollutant	Result
1	40000	1	Carbon Monoxide	Within Limit

- The inputs are the pollutant number, mileage, and grams of pollutant emitted per mile.

- The outputs are:
  - the pollutant name; and,
  - a statement declaring if the amount of pollutant is within the limit; or, is not within the limit.

When the macro is run, the table will display if the user specified parameters (pollutant number, mileage, and grams) are within the pollutant limit or not within the pollutant limit according to the first table. Suitable messages include: “Within Limit” or “Exceeds Limit”. Use B2 as the Upper-Left corner of this table.

2. Color the cell ***green*** if within the pollutant limit, color the cell ***red*** when the pollutant limit is exceeded: Range(“cell addres”).Interior.ColorIndex = x.

3. Write a second macro to clear the inputs and outputs from the user-input/output table (bottom row of the second table). Select a range of cells to clear and use ClearContents: Range(“select a range of cells”).ClearContents.

4. Test and debug your macros until they work correctly.

5. VBA code is provided to help you get started (next page).

   a. Select the Developer tab
   b. Select the Visual Basic icon to start the VBE (Alt-F11 is the shortcut key).
   c. Select Insert from the VBE ribbon and from the drop-down menu select Module.
   d. Select Module1 from the Project window on the right and then rename Module1 in the Properties window to something descriptive.
   e. Give your module and worksheet meaningful names.
   f. Copy and Paste the code provided within this document (next page) and modify it until it works. Some clues are provided in the code annotations.

```
Sub pollutantsTable()

   Dim polType As Integer, polAmt As Double, miles As Double

   polType = InputBox( "Enter the pollutant number: ")               '  remove this line
   miles =  InputBox( "Enter the odometer reading in miles: ")     ' remove this line
   polType = InputBox( "Enter the pollutant amount in grams: ") ' remove this line

   Range(“B3”).Value = polType       ' remove this line
   Range(“C3”).Value = miles           ' remove this line
   Range(“D3”).Value = polAmt       ' remove this line

  '  Convert the program to read values out of the input/output table as shown here:
  ' polType = Range("B3").Value      ' Get the pollutant number
                                                  ' Get the odometer reading by completing the code
                                                  ' Get the pollutant amount by completing the code

   Dim FML As Double, SML As Double          ' Get the Two Mileage limits:
   FML = Range("D9").Value
   SML =                                                 ' Complete this line of code

   Dim fgl As Double, sgl As Double    ' these are the two gram limits

   Select Case polType        ' Set the gram limits for each Case
      Case 1                              ' Carbon Monoxide
         fgl = Range("D10").Value
         sgl = Range("E10").Value
         Range("E3").Value = Range("C10").Value
      Case 2                              ' Hydrocarbons
         fgl =
         sgl =
         Range("E3").Value =
      Case 3                              ' Nitrogen

      Case 4                              ' Non-methane Hydrocarbons

   End Select

' Include a nested If-control structure here. The outer If should focus on the three distinct mileage
intervals:
'  milage less than or equal to 50,000, between 50,000 and 100,000 (inclusive); and finally,
'  above 100,000 miles.
'
' The inner nested-Ifs should focus on each of the two distinct gram limits.
' Enter the correct message into your output cell: “Within Limit” or “Exceeds Limit”
' Color the message-cell green or red depending on the output message.
 ' Example:
 ' Range("F3").Value =  the correct output message
 ' Range("F3").Interior.ColorIndex =  the correct color index value (look it up on the internet!)

End Sub
```

Part II: Locating a point in the Cartesian plane.

1. Write a VBA function that determines where a coordinate pair is located in the Cartesian plane. The function accepts two input arguments: the x-coordinate and the y-coordinate. The function returns one of these seven strings:
    - X-axis
    - Y-axis
    - Origin
    - Q-1
    - Q-2
    - Q-3
    - Q-4

2. The return value is based on the (x, y) coordinate pair location in the Cartesian plane.

3. You will need this function for a future lab exercise.