Function CalcEmotionRating(ByVal InputMsg As String) As Double

' You need to calculate the emotion rating here using this steps:

' 1) Count the number of positive words in the input message (code is already

' given below)

' 2) Count the number of negative words in the input message (to be done by you)

' 3) Calculate the emotion rating using the word counts (to be done by you)

' 4) Return the value (to be done by you)

' Change the InputMsg to lower case with the LCase() function

InputMsg = LCase(InputMsg)

' Initialize the positive word count to 0

PositiveWordCount = 0

' The while-loop for positive word counting retreives a positive word from the table in each

' execution of the loop. The positive word table will be scanned row by row and the CurrentRow

' variable indicates the current row which the loop is looking at. CurrentRow starts with 1 and

' will increase by 1 at the end of every execution of the loop

CurrentRow = 1

' While the cell at the location (CurrentRow, 1) in the "Positive Word List" worksheet

' is not empty, continue with the counting procedure (i.e., increase PositiveWordCount

' by 1 if the current word, which is in the mentioned cell, appears in the InputMsg)

While Worksheets("Positive Word List").Cells(CurrentRow, 1).Value <> ""

' Assign the current word, which is in the current cell at location (CurrentRow, 1),

' to the variable CurrentWord

CurrentWord = Worksheets("Positive Word List").Cells(CurrentRow, 1)

' See if the CurrentWord is contained in the InputMsg or not

If InStr(1, InputMsg, CurrentWord) > 0 Then

' If yes, increase the PositiveWordCount by 1

PositiveWordCount = PositiveWordCount + 1

End If

' Move to the next row - to check the next word in the next cell

CurrentRow = CurrentRow + 1

Wend

End Function