

1. (15 Marks) Develop a VBA function, called `calculateCharge`, that takes two parameters, an integer number, called `pType`, as the type of the plan a user subscribes to and a double number, called `usage`, as the data usage (in GBytes) the user used in a month and calculates and returns the monthly charge for the data usage using the unit price per GByte listed in the following table. (You can simply calculate the charge as the usage multiply the appropriate unit price.)

Plan Type	<= 5 GByte	5 GByte to 10 GByte	> 10 GByte
Economy (1)	7	6.5	5.0
Premium (2)	8	6	5.0
Ultimate (3)	9	6.5	4.5

Your function should return -1 if the plan type is not valid and/or the usage is not a positive number.

```
Function calculateCharge(byVal pType as Integer, _
                        byVal usage as Double) As Double
```

```
    Dim rate As Double
    If pType = 1 Then
        If usage < 0 Then
            rate = -1
        ElseIf usage <= 5 Then
            rate = 7
        ElseIf usage <= 10 Then
            rate = 6.5
        Else
            rate = 5.0
        End If
    ElseIf pType = 2 Then
        If usage < 0 Then
            rate = -1
        ElseIf usage <= 5 Then
            rate = 8.0
        ElseIf usage <= 10 Then
            rate = 6.0
        Else
            rate = 5.0
        End If
    ElseIf pType = 3 Then
        If usage < 0 Then
            rate = -1
        ElseIf usage <= 5 Then
            rate = 9.0
        ElseIf usage <= 10 Then
            rate = 6.5
        Else
```

```

        rate = 4.5
    End If
Else
    rate = -1
End If

If rate = -1 Then
    calculateCharge = -1
Else
    calculateCharge = rate * usage
End If
End Function

```

2. (15 Marks) Develop a VBA function, called `calculateAverage`, that takes the start row number, the end row number and a column number as its parameters, and then calculate the average of all the numbers stored in the active worksheet in the indicated column from the indicated start row to the indicated end row. You can safely assume that the data in these cells are floating point numbers. Your function should return a double number in the end.

```

Function calculateAverage(byVal sRow as Integer,
                        byVal eRow as Integer,
                        byVal col as Integer) As Double

    Dim sum As Double
    Dim count As Integer
    Dim row As Integer
    sum = 0
    count = 0
    For row = sRow To eRow
        If Not IsEmpty(Cells(row, col)) Then
            sum = sum + Cells(row, col)
            count = count + 1
        End If
    Next row

    If count = 0 Then
        calculateAverage = 0
    Else
        calculateAverage = sum / count
    End If
End Function

```

3. (15 Marks) In an Excel workbook, the currently active worksheet is called "customers". In this active worksheet, Column A to D stores customer's ID (A), full name (B), address (C) and phone number (D) respectively.

The actual customer data area starts from row 4 and there is no blank lines in the data area.

Develop a VBA procedure, called `LookUpCustomerInfo` that performs the following steps:

- Ask the user for a customer's ID. (You can assume the ID is a string.)
- Search the data area to see whether there is a customer with the given ID. (The search should start from row 4, and you can use the function `IsEmpty(a cell)` to test whether the data area has ended.)
- If there is a customer with the given ID, then use a message box to display the customer's name, and phone number. Otherwise, use a message box to display an appropriate error message. (You can assume that the customer IDs are all unique.)

```
Sub LookUpCustomerInfo()  
    Dim cid As String  
    Dim row As Integer  
  
    cid = InputBox("Enter customer ID:")  
    row = 4  
    Do Until IsEmpty(Cells(row, 1)) Or Cells(row, 1) = cid  
        row = row + 1  
    Loop  
  
    If IsEmpty(Cells(row, 1)) Then  
        MsgBox "No such customer with ID of " & cid  
    Else  
        MsgBox "The Customer's name is " & Cells(row, 2) _  
            & " and his/her phone number is " & Cells(row, 4)  
    End If  
End Sub
```