Excel Assignment - 21

1. Write a VBA code to enter your name in A1 Cell using Input Box and once you enter the name display a message box that says the name has been entered.

Ans.

```
Sub EnterName()
   Dim name As String

   ' Prompt the user to enter their name
   name = InputBox("Enter your name:")

   ' Check if a name is entered
   If Len(name) > 0 Then
      ' Display a message box with the entered name
      MsgBox "Your name has been entered: " & name
      ' Enter the name in cell A1
      Range("A1").Value = name
   End If
End Sub
```

- Open the VBA editor by pressing Alt+F11.
- Insert a new module.
- Copy and paste the above code into the module.
- Close the VBA editor.
- Run the macro "EnterName" by pressing Alt+F8, selecting "EnterName", and clicking "Run".

2. What are Userforms? Why are they used? How to fill a list box using for loop.

Ans.

UserForms in VBA are custom dialog boxes used to create interactive and user-friendly interfaces in Excel. They are used to collect user input, display information, and perform actions. To fill a ListBox using a For loop, you can write code in the UserForm module's Initialize event that adds items to the ListBox using the AddItem method within a For loop.

To fill a list box using a For loop, you can follow these steps:

- Insert a UserForm by clicking on "Insert" > "UserForm" in the VBA editor.
- Add a ListBox control to the UserForm by clicking on "Toolbox" > "ListBox" and drawing a ListBox on the UserForm.
- Write the following code in the UserForm module:

```
Private Sub UserForm_Initialize()
   Dim i As Integer

   For i = 1 To 10
      ListBox1.AddItem "Item " & i
   Next i
End Sub
```

3. What is an array? Write a VBA code to enter students and their marks from the below table.
Ans. An array in VBA is a data structure that can store multiple values of the same data type in a single variable. It provides a way to group related data elements under one variable name.

```
Sub EnterStudentsMarks()
   Dim studentsArray() As Variant
   Dim rng As Range
   Dim cell As Range
   Dim numRows As Long
   Dim i As Long

   ' Set the range where the student names and marks are located
   Set rng = Range("A1:B5") ' Assuming the data is in columns A and B
```

```
    ' Determine the number of rows in the range
    numRows = rng.Rows.Count

    ' Resize the array based on the number of rows
    ReDim studentsArray(1 To numRows, 1 To 2)

    ' Loop through each cell in the range and populate the array
    For i = 1 To numRows
        Set cell = rng.Cells(i, 1)
        studentsArray(i, 1) = cell.Value ' Student name
        studentsArray(i, 2) = cell.Offset(0, 1).Value ' Student marks
    Next i

    ' Access the array values for further processing or display
    ' For example, you can loop through the array and print the student names and marks
    For i = 1 To numRows
        Debug.Print "Student: " & studentsArray(i, 1) & ", Marks: " & studentsArray(i, 2)
    Next i
End Sub
```

rng represents the range where the student names and marks are located

4. Use the following data to create a VBA code pie chart. Use Font - ‘Times new Roman’, Size -14, Bold, Title - Piechart’ and you are per to use colours as per your taste.

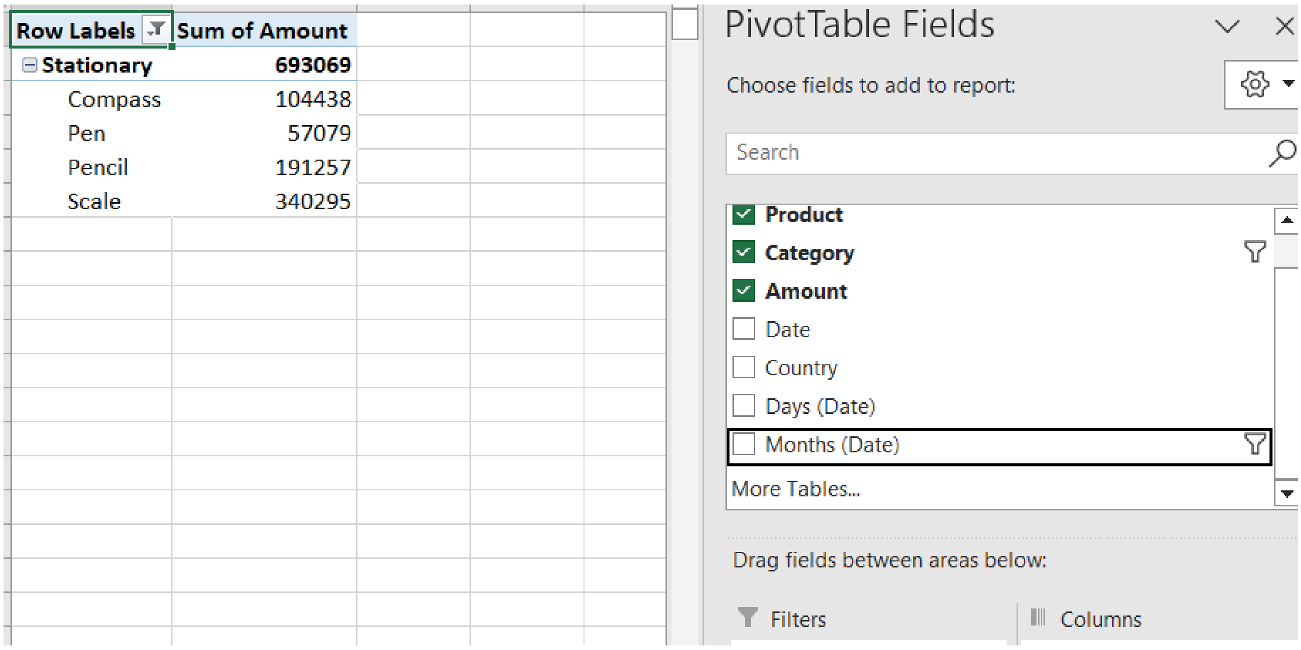
Ans.

```
Sub CreatePieChart()
    Dim ws As Worksheet
    Dim chartRange As Range
    Dim chartObject As ChartObject
    Dim dataRange As Range
    Dim chartTitle As String

    ' Set the worksheet where the chart will be created
    Set ws = ThisWorkbook.Worksheets("Sheet1") ' Change "Sheet1" to your worksheet name
    ' Set the range of the data for the chart
    Set dataRange = ws.Range("A1:B5") ' Adjust the range as per your data

    ' Set the title for the chart
    chartTitle = "Pie Chart"

    ' Create a new chart object and set its properties
    Set chartObject = ws.ChartObjects.Add(Left:=100, Width:=300, Top:=100, Height:=300)
    With chartObject.Chart
        .ChartType = xlPie
        .SetSourceData Source:=dataRange
        .HasTitle = True
        .ChartTitle.Text = chartTitle
        .ChartTitle.Font.Name = "Times New Roman"
        .ChartTitle.Font.Size = 14
        .ChartTitle.Font.Bold = True
        .SeriesCollection(1).ApplyDataLabels
        .SeriesCollection(1).DataLabels.Font.Name = "Times New Roman"
        .SeriesCollection(1).DataLabels.Font.Size = 14
    End With

    ' Adjust the chart position and size
    chartObject.Left = ws.Range("D1").Left
    chartObject.Top = ws.Range("D1").Top
    chartObject.Width = 300
    chartObject.Height = 300
End Sub
```

Sheet1, worksheet name where chart have to create.

5. Check the dataset in the link given below and create a pivot table using VBA showing the sales for the year from stationary category.
https://docs.google.com/spreadsheets/d/1IRSEnmgz8Ro276-GslknRNk0zlrB5CZH1YrnT71kqFM/edit?usp=sharing

Ans.



Row Labels	Sum of Amount
Stationary	693069
Compass	104438
Pen	57079
Pencil	191257
Scale	340295

PivotTable Fields

Choose fields to add to report:

Search

☑ Product
☑ Category
☑ Amount
□ Date
□ Country
□ Days (Date)
□ Months (Date)

More Tables...

Drag fields between areas below:

```
Sub CreatePivotTable()
    Dim wsData As Worksheet
    Dim wsPivot As Worksheet
    Dim dataRange As Range
    Dim pivotTable As PivotTable
    Dim pivotCache As PivotCache

    ' Set the worksheet with the data
    Set wsData = ThisWorkbook.Worksheets("Sheet1") ' Replace "Sheet1" with the name of your data worksheet

    ' Set the range of your data, including headers
    Set dataRange = wsData.Range("A1:F187") ' Adjust the range as per your dataset

    ' Create a new worksheet for the pivot table
    Set wsPivot = ThisWorkbook.Worksheets.Add
    wsPivot.Name = "PivotTable"

    ' Create a pivot table cache
    Set pivotCache = ThisWorkbook.PivotCaches.Create(SourceType:=xlDatabase,
SourceData:=dataRange)

    ' Create a pivot table on the new worksheet
    Set pivotTable = wsPivot.PivotTables.Add(PivotCache:=pivotCache,
TableDestination:=wsPivot.Range("A1"))

    ' Configure the pivot table fields
    With pivotTable
        ' Add fields to the pivot table
        .PivotFields("Order ID").Orientation = xlRowField
        .PivotFields("Product").Orientation = xlRowField
        .PivotFields("Category").Orientation = xlRowField
        .PivotFields("Amount").Orientation = xlDataField

        ' Apply filters
        .PivotFields("Category").PivotFilters.Add Type:=xlCaptionEquals, Value1:="Stationary"

        ' Customize layout and appearance
        .ColumnGrand = False
        .RowGrand = False
    End With
End Sub
```

6. Write a step-by-step procedure to protect your workbook using a password.

Ans.

- Open the workbook that you want to protect.
- Click on the "File" tab in the Excel ribbon to access the backstage view.
- In the backstage view, click on "Protect Workbook" under the "Info" section.

- From the drop-down menu, select "Encrypt with Password".
- A dialog box titled "Encrypt Document" will appear. Enter a password of your choice in the "Password" field. Note that the password is case-sensitive.
- Click on the "OK" button to confirm the password.
- A confirmation dialog box will appear, asking you to re-enter the password. Re-enter the same password in the "Reenter password" field.
- Click on the "OK" button to confirm the password once again.
- Save your workbook by clicking on the "Save" button or using the keyboard shortcut Ctrl + S.