

RR_Assignment_21

Q1. 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.

```
Sub EnterName()  
    Dim name As String  
    name = InputBox("Please enter your name:")  
    Range("A1").Value = name  
    MsgBox "Your name, " & name & ", has been entered in cell  
A1.", vbInformation, "Name Entered"  
End Sub
```

Q2. What are Userforms? Why are they used? How to fill a list box using for loop.

Userforms are custom graphical interfaces that can be created in Excel VBA. They are used to allow users to interact with Excel workbooks in a more user-friendly way than using just the worksheet. Userforms can contain various controls, such as text boxes, labels, command buttons, checkboxes, option buttons, and list boxes, that can be used to input or display data.

Userforms are used for a variety of purposes, such as data entry, data filtering, data analysis, and data visualization. They can be used to automate tasks, improve data accuracy, and save time.

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

Add a list box control to the userform by clicking on the list box control in the toolbox and then clicking on the userform.

Set the properties of the list box control, such as Name, Width, Height, and RowSource.

Add the following code to the userform module:

```
Private Sub UserForm_Initialize()  
    Dim i As Integer  
    For i = 1 To 10  
        ListBox1.AddItem "Item " & i  
    Next i  
End Sub
```

Change "ListBox1" to the name of your list box control.

Run the userform by pressing F5 or clicking the Run button.

The list box control should be filled with the items "Item 1" to "Item 10".

The above code initializes the userform and fills the list box control with items using a for loop. The loop adds an item to the list box for each value of the counter variable "i". In this example, the loop runs from 1 to 10 and adds an item with the text "Item i" to the list box for each value of i. You can modify the code to fill the list box with any desired items.

Q3. What is an array? Write a VBA code to enter students and their marks from the below table.

```
Sub EnterStudentMarks()  
    Dim studentMarks(1 To 4, 1 To 2) As Variant  
    studentMarks(1, 1) = "John"  
    studentMarks(1, 2) = 85  
    studentMarks(2, 1) = "Mary"  
    studentMarks(2, 2) = 92  
    studentMarks(3, 1) = "David"  
    studentMarks(3, 2) = 78  
    studentMarks(4, 1) = "Sarah"  
    studentMarks(4, 2) = 90  
  
    Range("A1").Resize(4, 2).Value = studentMarks  
End Sub
```

Q4. Use the following data to create a pie chart using VBA code. Use Font - 'Times new Roman', Size -14, Bold, Title - Piechart' and you are per to use colours as per your taste.

```
Sub CreatePieChart()
```

'Declare variables

Dim cht As Chart

Dim rng As Range

Dim data As Variant

Dim i As Integer

'Define data for chart

data = Array(20, 30, 40, 10)

'Set range for chart data

Set rng = Range("A1:A4")

'Create chart

Set cht = ActiveSheet.Shapes.AddChart2(251, xlPie).Chart

'Set chart title and font

With cht

 .HasTitle = True

 .ChartTitle.Text = "Pie Chart"

 .ChartTitle.Font.Name = "Times New Roman"

 .ChartTitle.Font.Size = 14

 .ChartTitle.Font.Bold = True

End With

'Add data to chart

With cht.SeriesCollection.NewSeries

 .Values = data

 .XValues = rng

 .Name = "Data"

End With

'Set chart colors

For i = 1 To UBound(data)

 cht.SeriesCollection(1).Points(i).Interior.Color = RGB(255
- i * 20, i * 30, i * 40)

Next i

End Sub

Q5. Check the dataset in the link given below and create a pivot table using VBA showing the sales for the year from stationary category.

Sub CreatePivotTable()

 'Declare variables

 Dim pvt As PivotTable

 Dim rngData As Range

 Dim rngPivot As Range

 'Set range for data and pivot table

 Set rngData = Range("A1:C100") 'Change to match your data
range

 Set rngPivot = Range("E3") 'Change to match the starting cell
of your pivot table

 'Create pivot table

 Set pvt =

ActiveSheet.PivotTableWizard(SourceType:=xlDatabase,

```
SourceData:=rngData, TableDestination:=rngPivot,  
TableName:="StationarySalesPivotTable")
```

```
'Add fields to pivot table
```

```
With pvt
```

```
    .PivotFields("Category").Orientation = xlRowField
```

```
    .PivotFields("Category").Position = 1
```

```
    .PivotFields("Year").Orientation = xlColumnField
```

```
    .PivotFields("Year").Position = 1
```

```
    .AddDataField .PivotFields("Sales"), "Sales", xlSum
```

```
    .PivotFields("Category").CurrentPage = "Stationary"
```

```
End With
```

```
'Format pivot table
```

```
With pvt.TableRange1
```

```
    .Font.Name = "Calibri"
```

```
    .Font.Size = 12
```

```
    .HorizontalAlignment = xlCenter
```

```
    .VerticalAlignment = xlCenter
```

```
End With
```

```
End Sub
```

Q6. Write step by step procedure to protect your workbook using a password.

Open your workbook in Excel.

Click on the "File" tab in the ribbon at the top of the screen.

Click on "Save As" in the left-hand menu.

Choose a location to save the workbook and enter a name for the file if necessary.

Click on the "Tools" dropdown menu at the bottom of the "Save As" dialog box.

Click on "General Options" in the dropdown menu.

Enter a password in the "Password to open" field. This password will be required to open the workbook.

Re-enter the password in the "Confirm password" field.

(Optional) Enter a password in the "Password to modify" field. This password will be required to make changes to the workbook.

Re-enter the password in the "Confirm password" field.

Click on the "OK" button to close the "General Options" dialog box.

Click on the "Save" button to save the protected workbook.

Now, when you or someone else tries to open the workbook, they will be prompted to enter the password you specified in step 7. If you also specified a password to modify the workbook in step 9, users will need to enter that password to make changes to the workbook.