**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

name = InputBox("Enter your name:")

Range("A1").Value = name

MsgBox "Your name has been entered."

End Sub

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

Ans:

Userforms are graphical user interface (GUI) elements in Microsoft Excel that allow users to interact with the workbook by collecting data, displaying information, and performing calculations. They are created using the Visual Basic for Applications (VBA) programming language and can be customized to suit specific user needs and preferences.

Userforms are used to simplify and enhance the usability of a workbook by providing users with a more intuitive and user-friendly way to interact with the data. They can be used to input data, display reports or charts, provide instructions, and perform calculations.

To fill a list box using for loop in VBA, you can use the following code:

```

For i = 1 To 10

ListBox1.AddItem "Item " & i

Next i

```

This code will add ten items to the list box, each labeled "Item 1," "Item 2," etc., up to "Item 10." The for loop is used to repeat the same piece of code for each item, iterating from i=1 to i=10. You can modify this code to add different values or labels to the list box, depending on your needs.

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

Ans: An array is a data structure that can store a fixed-size sequential collection of elements of the same type. It is used in programming to efficiently store and manipulate large amounts of data.

Sub EnterMarks()

Dim MarksArray(1 To 3, 1 To 4) As Integer

'1st dimension represents students, 2nd dimension represents marks

'entering data into array

MarksArray(1, 1) = 80

MarksArray(1, 2) = 70

MarksArray(1, 3) = 90

MarksArray(2, 1) = 90

MarksArray(2, 2) = 85

MarksArray(2, 3) = 95

MarksArray(3, 1) = 70

MarksArray(3, 2) = 75

MarksArray(3, 3) = 80

'displaying data from array

For i = 1 To 3

Debug.Print "Student Name: "; GetStudentName(i)

Debug.Print "Mark 1: "; MarksArray(i, 1)

Debug.Print "Mark 2: "; MarksArray(i, 2)

Debug.Print "Mark 3: "; MarksArray(i, 3)

Next i

End Sub

Function GetStudentName(index As Integer) As String

Select Case index

Case 1: GetStudentName = "John"

Case 2: GetStudentName = "Sarah"

Case 3: GetStudentName = "Mike"

End Select

End Function

1. 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.

Ans:

```

Sub CreatePieChart()

Dim myChart As Chart

Set myChart = Sheets("Sheet1").Shapes.AddChart2(251, xlPie).Chart

'Set chart title and font properties

myChart.HasTitle = True

myChart.ChartTitle.Text = "Pie Chart"

myChart.ChartTitle.Format.TextFrame2.TextRange.Font.Name = "Times New Roman"

myChart.ChartTitle.Format.TextFrame2.TextRange.Font.Size = 14

myChart.ChartTitle.Format.TextFrame2.TextRange.Font.Bold = True

'Add data to the chart and format slices

With myChart.SeriesCollection.NewSeries

.Name = "Data"

.Values = Array(50, 30, 20)

End With

myChart.SeriesCollection(1).Points(1).Format.Fill.ForeColor.RGB= RGB(255, 192, 0)

myChart.SeriesCollection(1).Points(2).Format.Fill.ForeColor.RGB= RGB(192, 255, 0)

myChart.SeriesCollection(1).Points(3).Format.Fill.ForeColor.RGB= RGB(0, 192, 255)

'Adjust other chart properties as desired

myChart.PlotArea.Format.Fill.Visible = msoFalse

myChart.Legend.Position = xlLegendPositionRight

myChart.ChartArea.Format.Line.Visible = msoFalse

End Sub

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:

1. Set the source data range:

`Set sourceData = Sheets("Sheet1").Range("A1:C10")`

2. Set the pivot table destination:

`Set pivotDestination = Sheets("Sheet2").Range("A1")`

3. Create a pivot cache:

`Set pivotCache = ActiveWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:=sourceData)`

4. Create a new pivot table:

`Set pivotTable = pivotDestination.PivotTableWizard(PivotCache:=pivotCache)`

5. Specify the fields to be used in the pivot table:

`With pivotTable`

`.PivotFields("Field1").Orientation = xlRowField`

`.PivotFields("Field2").Orientation = xlColumnField`

`.AddDataField .PivotFields("Field3"), "Sum of Field3", xlSum`

`End With`

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

Ans:

1. Open the workbook you want to protect with a password.

2. Click on the ‘File’ tab from the top left corner of the Excel window.

3. Click on the ‘Save As’ option from the left menu.

4. Type in a file name for the workbook if you haven’t already done so.

5. Click on the ‘Tools’ drop-down menu in the bottom right corner of the ‘Save As’ dialog box.

6. Select the ‘General Options’ item from the drop-down menu.

7. In the ‘General Options’ dialog box, enter a password in the ‘Password to modify:’ field.

8. Click on the ‘OK’ button to confirm the password.

9. Re-enter the password again in the ‘Confirm Password’ field.

10. Click on the ‘OK’ button to save the password and close the dialog box.

11. Click on the ‘Save’ button to save the workbook with the password protection.

12. Close the workbook and reopen it to check if the password protection is working as intended.