**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.

We can use the following VBA code to prompt the user to enter their name using an Input Box and then display a message box once the name is entered:

Sub EnterName()

Dim userName As String

' Prompt the user to enter their name using an Input Box

userName = InputBox("Please enter your name:")

' Check if the user entered a name or canceled the input

If userName <> "" Then

' If the user entered a name, display a message box with the entered name

MsgBox "Your name '" & userName & "' has been entered.", vbInformation, "Name Entered"

' Write the entered name to cell A1 on the active sheet

ThisWorkbook.ActiveSheet.Range("A1").Value = userName

Else

' If the user canceled the input, display a message box

MsgBox "Name entry was canceled.", vbExclamation, "Name Not Entered"

End If

End Sub

To use this code, follow these steps:

Press ALT + F11 to open the VBA editor.

In the VBA editor, click Insert from the top menu and then select Module.

Paste the above VBA code into the new module.

Close the VBA editor.

Press ALT + F8 to open the list of available macros in Excel.

Select the macro named "EnterName" from the list and click "Run."

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

UserForms are custom dialog boxes or forms that we can create in Excel VBA. They provide a graphical user interface (GUI) that allows users to interact with our VBA application. UserForms can contain various controls, such as labels, text boxes, buttons, combo boxes, list boxes, and more, to gather user input, display information, or perform specific actions.

UserForms are used for the following purposes:

Data Entry: UserForms are useful for collecting data from users in a structured and user-friendly manner. we can design a form with input fields and validation rules to ensure data integrity.

Custom Menus: we can create custom menus or toolbars with buttons that trigger specific actions when clicked.

Information Display: UserForms can be used to display information, messages, or reports in a more visually appealing format.

Settings and Preferences: UserForms can be used to allow users to customize application settings or preferences.

Interactive Applications: UserForms enable the creation of interactive applications that respond to user actions, such as performing calculations, generating reports, etc.

Now, let's see how to fill a list box using a For loop in VBA:

Assuming we have a UserForm named "ListBoxForm" with a ListBox control named "ListBox1," here's an example code to fill the ListBox with numbers from 1 to 10 using a For loop:

Open the VBA editor by pressing ALT + F11.

Double-click on the UserForm "ListBoxForm" in the Project Explorer window to open the code window for the UserForm.

In the UserForm code window, enter the following code:

Private Sub UserForm\_Initialize()

Dim i As Integer

' Clear the ListBox before filling it

ListBox1.Clear

' Fill the ListBox with numbers 1 to 10 using a For loop

For i = 1 To 10

ListBox1.AddItem i

Next i

End Sub

Close the VBA editor and return to the Excel worksheet.

To display the UserForm with the filled ListBox, we can call the UserForm's Show method. For example, we can use a button or other trigger to call the UserForm from the Excel worksheet.

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

An array is a collection of elements of the same data type that are stored together under a single variable name. It allows us to store and manipulate multiple values of the same type efficiently. In VBA, arrays can be one-dimensional or multi-dimensional, and they provide a convenient way to work with a set of related data.

To enter students and their marks from the below table using VBA, we can use a two-dimensional array. In this example, we'll use a two-dimensional array to store the student names in the first column and their corresponding marks in the second column.

Assuming we have the student data in cells A2:B7, where column A contains student names and column B contains their marks, follow these steps:

Open the VBA editor by pressing ALT + F11.

In the VBA editor, click Insert from the top menu and then select Module.

In the new module, paste the following VBA code:

Sub EnterStudentsAndMarks()

Dim studentData() As Variant

Dim ws As Worksheet

Dim i As Long, numRows As Long

' Set the worksheet where the data is located

Set ws = ThisWorkbook.Sheets("Sheet1") ' Replace "Sheet1" with the actual sheet name

' Define the range containing student data (A2:B7 in this example)

numRows = 6 ' Number of rows containing data

studentData = ws.Range("A2:B" & 1 + numRows).Value

' Loop through the array to process the student data

For i = 1 To numRows

Dim studentName As String

Dim studentMark As Double

' Extract student name and mark from the array

studentName = studentData(i, 1)

studentMark = studentData(i, 2)

' Perform any processing or actions with the student data

' For example, we can write the data to another location or perform calculations.

' In this example, we'll display the student name and mark in the Immediate window.

Debug.Print "Student: " & studentName & ", Mark: " & studentMark

Next i

End Sub

Close the VBA editor.

To execute the code and process the student data, go back to our Excel worksheet and press ALT + F8 to open the list of available macros.

Select the macro named "EnterStudentsAndMarks" from the list and click "Run."

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

Create a pie chart using VBA code, we can follow these steps:

Ensure we have the necessary data in a worksheet. Let's assume the data is in cells A1:B5, where column A contains category labels, and column B contains corresponding values.

Open the VBA editor by pressing ALT + F11.

In the VBA editor, click Insert from the top menu and then select Module.

In the new module, paste the following VBA code:

Sub CreatePieChart()

Dim ws As Worksheet

Dim chartObj As ChartObject

Dim chartDataRange As Range

Dim chartTitle As String

' Set the worksheet where the data is located

Set ws = ThisWorkbook.Sheets("Sheet1") ' Replace "Sheet1" with the actual sheet name

' Define the data range for the chart (A1:B5 in this example)

Set chartDataRange = ws.Range("A1:B5")

' Create a new chart on the worksheet

Set chartObj = ws.Shapes.AddChart(xlPie, Left:=100, Width:=375, Top:=75, Height:=225)

' Set the chart data range

chartObj.Chart.SetSourceData Source:=chartDataRange

' Set chart title and formatting

chartTitle = "Pie Chart"

chartObj.Chart.HasTitle = True

chartObj.Chart.ChartTitle.Text = chartTitle

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

chartObj.Chart.ChartTitle.Font.Size = 14

chartObj.Chart.ChartTitle.Font.Bold = True

' Customize the chart colors (optional)

' Replace the default color palette with our preferred colors

chartObj.Chart.ChartArea.Format.Fill.ForeColor.RGB = RGB(255, 192, 0) ' Yellow

chartObj.Chart.ChartGroups(1).Points(1).Format.Fill.ForeColor.RGB = RGB(0, 176, 80) ' Green

chartObj.Chart.ChartGroups(1).Points(2).Format.Fill.ForeColor.RGB = RGB(0, 32, 96) ' Blue

chartObj.Chart.ChartGroups(1).Points(3).Format.Fill.ForeColor.RGB = RGB(128, 0, 64) ' Purple

chartObj.Chart.ChartGroups(1).Points(4).Format.Fill.ForeColor.RGB = RGB(192, 0, 0) ' Red

End Sub

Close the VBA editor.

To execute the code and create the pie chart, go back to our Excel worksheet and press ALT + F8 to open the list of available macros.

Select the macro named "CreatePieChart" from the list and click "Run."

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

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

To protect our workbook using a password in Microsoft Excel, follow these step-by-step procedures:

Open the Excel workbook that we want to protect with a password.

Click on the "File" tab in the top-left corner of the Excel window.

From the File menu, select "Protect Workbook" (or "Protect Workbook with Password" in older versions of Excel). A dropdown menu will appear.

Click on "Encrypt with Password." A small dialog box labeled "Encrypt Document" will appear.

In the "Encrypt Document" dialog box, we will see an input field where we can enter a password. Enter our desired password in the input field. Make sure to choose a strong password that includes a combination of uppercase letters, lowercase letters, numbers, and special characters.

After entering the password, click on the "OK" button. A confirmation dialog box will appear, asking us to re-enter the password.

Re-enter the same password in the confirmation dialog box and click on the "OK" button.

our workbook is now protected with a password. Excel will save the workbook and prompt us to re-enter the password every time we open the file.