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

Sub EnterNameAndDisplayMessage()

' Declare variables

Dim userName As String

' Use Input Box to enter the name

userName = InputBox("Enter your name:", "Name Entry")

' Check if the user entered a name

If userName <> "" Then

' Enter the name in cell A1

Range("A1").Value = userName

' Display a message box

MsgBox "Name has been entered successfully!", vbInformation, "Success"

Else

' Display a message if the user cancels the Input Box

MsgBox "No name entered. Operation canceled.", vbExclamation, "Canceled"

End If

End Sub

* What are Userforms? Why are they used? How to fill a list box using

for loop.

**UserForms in VBA:**

**UserForms** in VBA are custom dialog or input forms that you can create to interact with users. They provide a graphical interface for gathering information, making selections, or displaying messages. UserForms are commonly used in Excel VBA to create custom dialog boxes, data entry forms, or interactive interfaces within Excel applications.

**Why Use UserForms:**

1. **Enhanced User Interaction:** UserForms provide a more user-friendly and visually appealing way to interact with users compared to standard input boxes or message boxes.
2. **Data Entry and Validation:** You can create custom data entry forms with input fields, checkboxes, option buttons, and other controls to collect and validate user input.
3. **Customized Dialog Boxes:** UserForms allow you to design customized dialog boxes with specific functionalities, improving the user experience.
4. **Complex Input Scenarios:** When you need more complex input scenarios or when you want to guide users through a series of steps, UserForms are more suitable than standard input boxes.

**How to Fill a ListBox Using a For Loop:**

Assuming you have a UserForm with a ListBox control, here's an example of how you can fill that ListBox using a For Loop:

1. Open the Excel workbook where you want to create the UserForm.
2. Press **Alt + F11** to open the VBA editor.
3. Insert a new UserForm by right-clicking on a project item, selecting **Insert**, and choosing **UserForm**.
4. On the UserForm, add a ListBox control (you can do this by dragging it from the toolbox).
5. Double-click on the UserForm to open the code window.

* What is an array? Write a VBA code to enter students and their marks

from the below table.

Arrays in VBA:

In VBA, an array is a collection of variables of the same type stored under a common name. The variables, called elements, can be accessed by an index or subscript that specifies their position in the array. Arrays can be one-dimensional, two-dimensional, or multi-dimensional.

Example: Entering Students and Their Marks using Arrays

Let's say you have a table with students and their marks like this:

css

Copy code

| Student | Marks |

|----------|-------|

| Student1 | 85 |

| Student2 | 92 |

| Student3 | 78 |

| Student4 | 95 |

And you want to store this data in VBA arrays. Here's an example code:

vba

Copy code

Sub EnterStudentMarks()

' Declare variables

Dim students() As String

Dim marks() As Integer

Dim numRows As Integer

Dim i As Integer

' Define the number of rows in your table

numRows = 4

' Resize arrays to match the number of rows

ReDim students(1 To numRows)

ReDim marks(1 To numRows)

' Assign student names and marks to arrays

students(1) = "Student1"

students(2) = "Student2"

students(3) = "Student3"

students(4) = "Student4"

marks(1) = 85

marks(2) = 92

marks(3) = 78

marks(4) = 95

' Display the data

For i = 1 To numRows

Debug.Print "Student: " & students(i) & ", Marks: " & marks(i)

Next i

End Sub

* 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 chartSheet As ChartObject

Dim dataRange As Range

Dim chartTitle As String

' Set the data range (assuming the data is in A1:B4)

Set dataRange = ThisWorkbook.Sheets("Sheet1").Range("A1:B4")

' Set the chart title

chartTitle = "Pie Chart"

' Create a new chart sheet

Set chartSheet = ThisWorkbook.Sheets.Add

chartSheet.ChartObjects.Add(Left:=10, Width:=375, Top:=75, Height:=225).Select

' Activate the chart

ActiveChart.SetSourceData Source:=dataRange

' Set chart type to pie chart

ActiveChart.ChartType = xlPie

' Format the title

With ActiveChart

.HasTitle = True

.ChartTitle.Text = chartTitle

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

.ChartTitle.Font.Size = 14

.ChartTitle.Font.Bold = True

End With

' Apply color to data points (you can customize these colors)

ActiveChart.SeriesCollection(1).Points(1).Interior.Color = RGB(255, 0, 0) ' Red

ActiveChart.SeriesCollection(1).Points(2).Interior.Color = RGB(0, 255, 0) ' Green

ActiveChart.SeriesCollection(1).Points(3).Interior.Color = RGB(0, 0, 255) ' Blue

ActiveChart.SeriesCollection(1).Points(4).Interior.Color = RGB(255, 255, 0) ' Yellow

End Sub

* 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 ws As Worksheet

Dim pt As PivotTable

Dim ptField As PivotField

' Set the worksheet

Set ws = ThisWorkbook.Sheets(1) ' Change to the appropriate sheet index or name

' Create a Pivot Table on a new sheet

Set pt = ws.PivotTableWizard(SourceType:=xlDatabase, SourceData:=ws.UsedRange, TableDestination:=ws.Cells(1, ws.UsedRange.Columns.Count + 2), TableName:="PivotTable")

' Set the Pivot Table fields

Set ptField = pt.PivotFields("Category")

ptField.Orientation = xlRowField

ptField.Position = 1

Set ptField = pt.PivotFields("Year")

ptField.Orientation = xlColumnField

ptField.Position = 1

Set ptField = pt.PivotFields("Sales")

ptField.Orientation = xlDataField

ptField.Function = xlSum

ptField.Position = 1

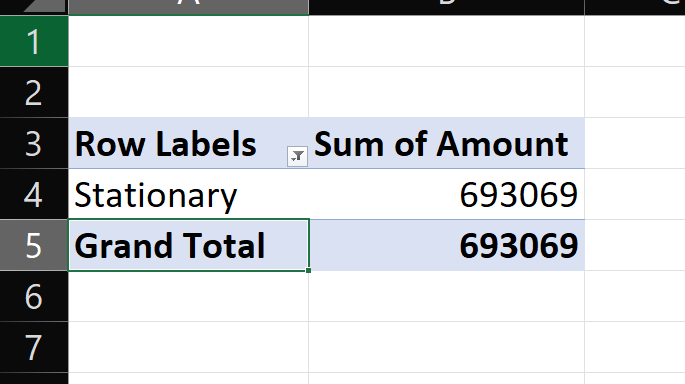
' Filter for "stationary" category

pt.PivotFields("Category").PivotFilters.Add2 Type:=xlCaptionEquals, Value1:="stationary"

End Sub

* 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 ws As Worksheet

Dim pt As PivotTable

Dim ptField As PivotField

' Set the worksheet (replace "Sheet1" with your actual sheet name)

Set ws = ThisWorkbook.Sheets("Sheet1")

' Create a Pivot Table on a new sheet

Set pt = ws.PivotTableWizard(SourceType:=xlDatabase, SourceData:=ws.UsedRange, TableDestination:=ws.Cells(1, ws.UsedRange.Columns.Count + 2), TableName:="PivotTable")

' Set the Pivot Table fields

Set ptField = pt.PivotFields("Category")

ptField.Orientation = xlRowField

ptField.Position = 1

Set ptField = pt.PivotFields("Year")

ptField.Orientation = xlColumnField

ptField.Position = 1

Set ptField = pt.PivotFields("Sales")

ptField.Orientation = xlDataField

ptField.Function = xlSum

ptField.Position = 1

' Filter for "stationary" category

pt.PivotFields("Category").PivotFilters.Add2 Type:=xlCaptionEquals, Value1:="stationary"

End Sub

* Write step by step procedure to protect your workbook using a

password.

1. **Open the Workbook:**
   * Open the Excel workbook that you want to protect.
2. **Go to the "File" Menu:**
   * Click on the "File" tab in the top left corner of the Excel window.
3. **Select "Info" on the left:**
   * In the File menu, select "Info" from the left-hand menu.
4. **Click on "Protect Workbook":**
   * In the Info pane, you'll find an option called "Protect Workbook." Click on it.
5. **Choose "Encrypt with Password":**
   * In the Protect Structure and Windows dialog box, you'll see an option to "Encrypt with Password." Click on it.
6. **Enter a Password:**
   * A new dialog box will appear asking you to enter a password. Type in the password you want to use to protect the workbook.
   * Note: Make sure to remember this password because if you forget it, you won't be able to access the workbook.
7. **Confirm the Password:**
   * After entering the password, you will be asked to confirm it. Re-enter the password to confirm.
8. **Save the Workbook:**
   * After setting the password, click on the "OK" button.
   * Save the workbook to apply the password protection. You might be prompted to save the workbook if you haven't saved it before.
9. **Close and Reopen the Workbook:**
   * Close the workbook and reopen it to see the changes.
   * When you open the workbook, you will be prompted to enter the password. Enter the correct password to access the workbook.
10. **Verify Protection:**
    * Once the workbook is open, you can verify that it's protected by going to the "File" tab, selecting "Info," and checking if "Protect Workbook" is now highlighted, indicating that protection is applied.