VBA-Challenge Code

The VBA code for Module 2 Challenge is written as below. Please see the Images at the end for screenshots.

Sub ModTwoSolution()

Dim ws As Worksheet

Dim lastRow As Long

Dim searchRange As Range

Dim searchCell As Range

Dim uniqueStrings As Collection

Dim sortedStrings() As String

Dim i As Long

Dim sumG As Double

Dim diff As Double

Dim percentage As Double

Dim outputRow As Long

Dim firstValueC As Double

Dim firstOccurrence As Boolean

Dim LastOccurrenceC As Double

Dim highestDiff As Double

Dim highestDiffString As String

Dim lowestDiff As Double

Dim lowestDiffString As String

Dim highestvol As Double

Dim highestvolString As String

Dim originalDate As String

Dim yearPart As Integer

Dim monthPart As Integer

Dim dayPart As Integer

Dim convertedDate As Date

' Initialize the highest/lowest values

highestDiff = -1E+308 ' Very low initial value

highestDiffString = ""

lowestDiff = 1E+308 ' Very high initial value

lowestDiffString = ""

highestvol = -1E+308 ' Very low initial value

highestvolString = ""

' Loop through each worksheet

For Each ws In ThisWorkbook.Worksheets

' Convert date strings in column B

lastRow = ws.Cells(ws.Rows.Count, "B").End(xlUp).Row

For i = 2 To lastRow

originalDate = ws.Cells(i, 2).Value

' Check if the date is in YYYYMMDD format

If Len(originalDate) = 8 And IsNumeric(originalDate) Then

yearPart = CInt(Mid(originalDate, 1, 4))

monthPart = CInt(Mid(originalDate, 5, 2))

dayPart = CInt(Mid(originalDate, 7, 2))

convertedDate = DateSerial(yearPart, monthPart, dayPart) ' Corrected day and month parts

ws.Cells(i, 2).Value = Format(convertedDate, "mm/dd/yyyy")

ElseIf IsDate(originalDate) Then

Dim dateParts() As String

dateParts = Split(originalDate, "/")

If UBound(dateParts) = 2 Then

Dim newDate As String

newDate = dateParts(1) & "/" & dateParts(0) & "/" & dateParts(2)

ws.Cells(i, 2).Value = newDate

End If

End If

Next i

' Add headers to columns I, J, K, and L

ws.Cells(1, "I").Value = "Ticker"

ws.Cells(1, "J").Value = "Yearly Change"

ws.Cells(1, "K").Value = "Percent Change"

ws.Cells(1, "L").Value = "Total Stock Volume"

ws.Cells(2, "O").Value = "Greatest % Increase"

ws.Cells(3, "O").Value = "Greatest % Decrease"

ws.Cells(4, "O").Value = "Greatest Total Volume"

ws.Cells(1, "P").Value = "Ticker"

ws.Cells(1, "Q").Value = "Value"

' Find the last row in column A

lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

' Define the ticker range

Set searchRange = ws.Range("A2:A" & lastRow)

' Initialize the collection for tickers

Set uniqueStrings = New Collection

' Add tickers to the collection

On Error Resume Next

For Each searchCell In searchRange

If searchCell.Value <> "" Then

uniqueStrings.Add searchCell.Value, CStr(searchCell.Value)

End If

Next searchCell

On Error GoTo 0

' Sort the tickers alphabetically

If uniqueStrings.Count > 0 Then

ReDim sortedStrings(1 To uniqueStrings.Count)

For i = 1 To uniqueStrings.Count

sortedStrings(i) = uniqueStrings(i)

Next i

QuickSort sortedStrings, LBound(sortedStrings), UBound(sortedStrings) ' Call the sorting function

' Process each unique ticker

outputRow = 2

For i = LBound(sortedStrings) To UBound(sortedStrings)

sumG = 0

firstOccurrence = False

firstValueC = 0

LastOccurrenceC = 0

diff = 0

' Sum total stock volume and find opening/closing values for the ticker

For Each searchCell In searchRange

If searchCell.Value = sortedStrings(i) Then

sumG = sumG + ws.Cells(searchCell.Row, "G").Value

' Find the first and last occurrence in column C and F

If Not firstOccurrence Then

firstValueC = ws.Cells(searchCell.Row, "C").Value

firstOccurrence = True

End If

LastOccurrenceC = ws.Cells(searchCell.Row, "F").Value

End If

Next searchCell

' Calculate the difference and percentage

diff = LastOccurrenceC - firstValueC

If firstValueC <> 0 Then

percentage = (diff / firstValueC) \* 100

Else

percentage = 0 ' Avoid division by zero

End If

' Print the results in columns I to L

ws.Cells(outputRow, "I").Value = sortedStrings(i)

ws.Cells(outputRow, "J").Value = diff

ws.Cells(outputRow, "K").Value = percentage & "%"

ws.Cells(outputRow, "L").Value = sumG

' Format cells based on yearly change

If diff > 0 Then

ws.Cells(outputRow, "J").Interior.Color = RGB(0, 255, 0) ' Green for positive

ElseIf diff < 0 Then

ws.Cells(outputRow, "J").Interior.Color = RGB(255, 0, 0) ' Red for negative

Else

ws.Cells(outputRow, "J").Interior.ColorIndex = xlNone ' No color for zero

End If

' Check for the highest/lowest value

If percentage > highestDiff Then

highestDiff = percentage

highestDiffString = sortedStrings(i)

End If

If percentage < lowestDiff Then

lowestDiff = percentage

lowestDiffString = sortedStrings(i)

End If

If sumG > highestvol Then

highestvol = sumG

highestvolString = sortedStrings(i)

End If

outputRow = outputRow + 1

Next i

End If

' Output the greatest values

ws.Cells(2, "P").Value = highestDiffString

ws.Cells(2, "Q").Value = highestDiff

ws.Cells(3, "P").Value = lowestDiffString

ws.Cells(3, "Q").Value = lowestDiff

ws.Cells(4, "P").Value = highestvolString

ws.Cells(4, "Q").Value = highestvol

Next ws

End Sub

'External function to sort the tickers alphabetically

Sub QuickSort(arr() As String, first As Long, last As Long)

Dim low As Long, high As Long

Dim midValue As String, temp As String

low = first

high = last

midValue = arr((first + last) \ 2)

Do While low <= high

Do While arr(low) < midValue

low = low + 1

Loop

Do While arr(high) > midValue

high = high - 1

Loop

If low <= high Then

temp = arr(low)

arr(low) = arr(high)

arr(high) = temp

low = low + 1

high = high - 1

End If

Loop

If first < high Then QuickSort arr, first, high

If low < last Then QuickSort arr, low, last

End Sub

Figure 1

A screenshot of a computer

Description automatically generated

Figure 2 A screenshot of a computer

Description automatically generated

Figure 3

A screenshot of a computer

Description automatically generated

Figure 4

A screenshot of a computer

Description automatically generated