**VBA CHALLENGE**

Student Name

Institutional Affiliation

Course Name

Instructor Name

Date

**VBA CHALLENGE**

**Overview of Project**

The purpose of deliverable 1 was to optimize the execution time (comparing to the old code). Refactoring the code made the code run faster with the use of an index to access data in an array, and with the use of nested loops. The code performance was measured after the use of conditional formatting, debugging and commenting on the code to optimize the execution time.

**Results**

The execution time was optimized using this code: AllStocksAnalysisRefactored()

Dim startTime As Single

Dim endTime As Single

yearValue = InputBox("What year would you like to run the analysis on?")

startTime = Timer

'Format the output sheet on All Stocks Analysis worksheet

Worksheets("All Stocks Analysis").Activate

Range("A1").Value = "All Stocks (" + yearValue + ")"

'Create a header row

Cells(3, 1).Value = "Ticker"

Cells(3, 2).Value = "Total Daily Volume"

Cells(3, 3).Value = "Return"

'Initialize array of all tickers

Dim tickers(12) As String

tickers(0) = "AY"

tickers(1) = "CSIQ"

tickers(2) = "DQ"

tickers(3) = "ENPH"

tickers(4) = "FSLR"

tickers(5) = "HASI"

tickers(6) = "JKS"

tickers(7) = "RUN"

tickers(8) = "SEDG"

tickers(9) = "SPWR"

tickers(10) = "TERP"

tickers(11) = "VSLR"

'Activate data worksheet

Worksheets(yearValue).Activate

'Get the number of rows to loop over

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

'1a) Create a ticker Index

Dim tickerIndex As Long

tickerIndex = 0

'1b) Create three output arrays

Dim tickerVolumes(11) As Long

Dim tickerStartingPrices(11) As Single

Dim tickerEndingPrices(11) As Single

''2a) Create a for loop to initialize the tickerVolumes to zero.

For i = 0 To 11

tickerVolumes(i) = 0

Next

''2b) Loop over all the rows in the spreadsheet.

For i = 2 To RowCount

'3a) Increase volume for current ticker

tickerVolumes(tickerIndex) = tickerVolumes(tickerIndex) + Cells(i, 8).Value

'3b) Check if the current row is the first row with the selected tickerIndex.

If Cells(i - 1, 1) <> tickers(tickerIndex) Then

tickerStartingPrices(tickerIndex) = Cells(i, 6)

End If

'3c) check if the current row is the last row with the selected ticker

'If the next rowâ€™s ticker doesnâ€™t match, increase the tickerIndex.

If Cells(i + 1, 1) <> tickers(tickerIndex) Then

tickerEndingPrices(tickerIndex) = Cells(i, 6)

'3d Increase the tickerIndex.

tickerIndex = tickerIndex + 1

End If

Next i

'4) Loop through your arrays to output the Ticker, Total Daily Volume, and Return.

For i = 0 To 11

Worksheets("All Stocks Analysis").Activate

Cells(i + 4, 1) = tickers(i)

Cells(i + 4, 2) = tickerVolumes(i)

Cells(i + 4, 3) = tickerEndingPrices(i) / tickerStartingPrices(i) - 1

Next i

'Formatting

Worksheets("All Stocks Analysis").Activate

Range("A3:C3").Font.FontStyle = "Bold"

Range("A3:C3").Borders(xlEdgeBottom).LineStyle = xlContinuous

Range("B4:B15").NumberFormat = "#,##0"

Range("C4:C15").NumberFormat = "0.0%"

Columns("B").AutoFit

dataRowStart = 4

dataRowEnd = 15

For i = dataRowStart To dataRowEnd

If Cells(i, 3) > 0 Then

Cells(i, 3).Interior.Color = vbGreen

Else

Cells(i, 3).Interior.Color = vbRed

End If

Next i

endTime = Timer

MsgBox "This code ran in " & (endTime - startTime) & " seconds for the year " & (yearValue)

**SCREEN SHOTS OF THE OPTIMIZED RESULTS FOR THE YEAR 2017, AND 2018 USING ABOVE CODE**

**Graphical user interface, text, application, email

Description automatically generated**

Graphical user interface, text, application

Description automatically generated

**SUMMARY**

The project was successfully completed as illustrated above that the codes ran faster for both the 2017, and 2018 stock analysis. The advantages of refactoring a code like this is to give a reader a better comprehension of the codes, facilitate maintenance and of course, optimize execution time. The most common problems with refactoring codes like this is that you might some new bugs, and errors.