```
Sub 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, it was provided with the assigment challenge section 2.5.3 Run the Analysis for any year


    Worksheets("All Stocks Analysis").Activate


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


    'Create a header row, it was provided in the challenge but it was studied in sectionwe did a similar excersize in Step 6, Section 2.3.3 Reuse Code and in previous sections


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

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

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


    'Initialize array of all tickersit was provided in the challenge but it was studied in section 2.3.1, also, we did a similar excersize in Step 6, Section 2.3.3 Reuse Code


    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

'Chalenge 1a. Create a ticker Index;  we did a similar excersize in Step 4, Section 2.3.3 Reuse Code

For i = 0 To 11
   tickerIndex = tickers(i)


'Challenge 1b.Create three output arrays,

Dim tickerVolumes As Long
Dim tickerStartingPrices As Single, tickerEndingPrices As Single


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

```
 'If the next row's ticker doesn't match, increase the tickerIndex.
Worksheets(yearValue).Activate
tickerVolumes = 0

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

For j = 2 To RowCount

   ' If the next row's ticker doesn't match, increase the tickerIndex.
   If Cells(j, 1).Value = tickerIndex Then

      'Challenge 3a. Increase volume for current ticker

      tickerVolumes = tickerVolumes + Cells(j, 8).Value

   End If


 'Challenge 3b. Check if the current row is the first row with the selected tickerIndex.
 'If  Then
   If Cells(j - 1, 1).Value <> tickerIndex And Cells(j, 1).Value = tickerIndex Then

       tickerStartingPrices = Cells(j, 6).Value

  'End If
   End If

 'Challenge 3c. check if the current row is the last row with the selected ticker
 'If  Then
```

```
      If Cells(j + 1, 1).Value <> tickerIndex And Cells(j, 1).Value = tickerIndex Then

         tickerEndingPrices = Cells(j, 6).Value

      'End If
      End If

   Next j

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


      Worksheets("All Stocks Analysis").Activate

      Cells(4 + i, 1).Value = tickerIndex
      Cells(4 + i, 2).Value = tickerVolumes
      Cells(4 + i, 3).Value = tickerEndingPrices / tickerStartingPrices - 1

       'With Range("C4:C15")
                '.NumberFormat = "0.0%"
                '.Value = .Value
       'End With

Next i

'Formatting Source: Module material - 2.4.1 Static 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

    ' Formatting Source: Module Material - 2.4.2 Conditional Formating

    dataRowStart = 4
    dataRowEnd = 15
    For i = dataRowStart To dataRowEnd

        If Cells(i, 3) > 0 Then

            'Color the cell green
            Cells(i, 3).Interior.Color = vbGreen

        ElseIf Cells(i, 3) < 0 Then

            'Color the cell red
            Cells(i, 3).Interior.Color = vbRed

        Else

            'Clear the cell color
            Cells(i, 3).Interior.Color = xlNone

        End If

Next i
```

```
'Measure Code Performance Source: Module Material - 2.5.3
endTime = Timer
   MsgBox "This code ran in " & (endTime - startTime) & " seconds for the year " & (yearValue)


End Sub


Sub ClearWorksheet()
   ' Clear Year from Top Header and Calculated data
   Range("A1:A1").Value = "AllStocks(xxxx)"
   Range("A3:C15").Clear
End Sub
```