Sheet7 - 1

```
'Retain the ComboBox shape after every selection
Private Sub UserInput_Change()
     ActiveSheet.Shapes("UserInput").Width = 100
     ActiveSheet.Shapes("UserInput").Height = 30
End Sub
```

ThisWorkbook - 1

```
Public Sub Workbook_Open()

'Populate the dropdown value when excel opens
With Sheet7.UserInput
    .AddItem "ALL"
    .AddItem "2018"
    .AddItem "2019"
    .AddItem "2020"
End With

End Sub
```

Module1 - 1

```
'# Reference used for creating this program
'#    - https://learn.microsoft.com/en-us/office/vba/api/overview/
'#    - https://www.homeandlearn.org/excel_vba_practice1.html
'  ----------------------------------------------------------------
'# Author SA
'# VBA-Challenge - This Procedure "calFnVBAChallenge" performs all the required actions for this challenge
'   --#Debugging Info#
'#  ----#This logic works only for 2018, 2019 and 2020 data sheet.
'#  ----#Any new data sheet should be added to below code logic and should updated in the combobox value.
'#  ----#The below logic will work only if column A(ticker Symbol) and Column B(Date) are sorted in the data sheet.
'#  ----#If dataset is not sorted,it should be sorted first before running the below code to get desired output.

Sub calFnVBAChallenge()

    'Cell reference variables
    Dim writeCol1 As String
    Dim writeCol2 As String
    Dim writeCol3 As String
    Dim writeCol4 As String

    'Sheet reference variables
    Dim sheet2018 As String: sheet2018 = Sheet1.Name
    Dim sheet2019 As String: sheet2019 = Sheet2.Name
    Dim sheet2020 As String: sheet2020 = Sheet3.Name
    Dim allValue As String: allValue = "All"

    Dim summarySheet As String: summarySheet = Sheet7.Name

    'UserInput variable
    Dim selectionValue As String

    'Exeute based on the selection value from the summary sheet combo box
    selectionValue = Worksheets(summarySheet).UserInput.Value

    '# START - Below code is retained only for Graders manual verification purpose only
      '## if combobox is not used and requires to be run directly from this macro uncomment the required selection and run the macro
     'selectionValue = "ALL"
     'selectionValue = "2018"
     'selectionValue = "2019"
     'selectionValue = "2020"
    '# END - Above code is retained only for Graders manual verification purpose only

'Check if the user selection is 2018 or for all sheet and perform calculation for the respective data sheet
 If (UCase(selectionValue) = UCase(sheet2018)) Or (UCase(selectionValue) = UCase(allValue)) Then

    'Write the summary data in to below respective columns defined below
     writeCol1 = "A"
     writeCol2 = "B"
     writeCol3 = "C"
     writeCol4 = "D"

    'Reset color formatting and clear content
     Worksheets(summarySheet).Range("A12 : D" & Rows.Count).Interior.Color = &HF4ECE4 '#E4ECF4
     Worksheets(summarySheet).Range("A12 : D" & Rows.Count).ClearContents
     Worksheets(summarySheet).Range("A7:D10").Interior.Color = &HF1E6DC '#DCE6F1
     Worksheets(summarySheet).Range("D7:D10").ClearContents

    'Call oneYearOutput procedure to perform one year calculation
     oneYearOutput sheet2018, writeCol1, writeCol2, writeCol3, writeCol4, summarySheet

    'Call greatestCalc procedure to find the greatest value and color formatting
     greatestCalc writeCol1, writeCol2, writeCol3, writeCol4, summarySheet
 End If

'Check if the user selection is 2019 or for all sheet and perform calculation for the respective data sheet
 If (UCase(selectionValue) = UCase(sheet2019)) Or (UCase(selectionValue) = UCase(allValue)) Then

    'Write the summary data in to below respective columns defined below
     writeCol1 = "E"
     writeCol2 = "F"
     writeCol3 = "G"
     writeCol4 = "H"

    'Reset color formatting and clear content
     Worksheets(summarySheet).Range("E12 : H" & Rows.Count).Interior.Color = &HE9EAF7 '#F7EAE9
     Worksheets(summarySheet).Range("E12 : H" & Rows.Count).ClearContents
     Worksheets(summarySheet).Range("E7:H10").Interior.Color = &H9496DA '#DA9694
```

Module1 - 2

```
        Worksheets(summarySheet).Range("H7:H10").ClearContents


       'Call oneYearOutput procedure to perform one year calculation
        oneYearOutput sheet2019, writeCol1, writeCol2, writeCol3, writeCol4, summarySheet

       'Call greatestCalc procedure to find the greatest value and color formatting
        greatestCalc writeCol1, writeCol2, writeCol3, writeCol4, summarySheet
    End If

   'Check if the user selection is 2020 or for all sheet and perform calculation for the respective data sheet
    If (UCase(selectionValue) = UCase(sheet2020)) Or (UCase(selectionValue) = UCase(allValue)) Then

       'Write the summary data in to below respective columns defined below
        writeCol1 = "I"
        writeCol2 = "J"
        writeCol3 = "K"
        writeCol4 = "L"

       'Reset color formatting and clear content
        Worksheets(summarySheet).Range("I12 : L" & Rows.Count).Interior.Color = &HECDFE4 '#E4DFEC
        Worksheets(summarySheet).Range("I12 : L" & Rows.Count).ClearContents
        Worksheets(summarySheet).Range("I7:L10").Interior.Color = &HC7A0B1 '#B1A0C7
        Worksheets(summarySheet).Range("L7:L10").ClearContents

       'Call oneYearOutput procedure to perform one year calculation
        oneYearOutput sheet2020, writeCol1, writeCol2, writeCol3, writeCol4, summarySheet

       'Call greatestCalc procedure to find the greatest value and color formatting
        greatestCalc writeCol1, writeCol2, writeCol3, writeCol4, summarySheet
    End If

End Sub
```

Module2 - 1

```
'# Reference used for creating this program
'#    - https://learn.microsoft.com/en-us/office/vba/api/overview/
'#    - https://www.homeandlearn.org/excel_vba_practice1.html
'  ---------------------------------------------------------------
'# Author SA
'# VBA-Challenge - loop through all the stocks for one year and output
'   --#The ticker symbol#
'#  --#Yearly change from the opening price at the beginning of a given year
'#      to the closing price at the end of that year.#
'#  --#The percentage change from the opening price at the beginning of a given year
'#      to the closing price at the end of that year.#
'#  --#The total stock volume of the stock.#

Sub oneYearOutput(sheetName As String, col1 As String, col2 As String, col3 As String, col4 As String, summarySheet As String)

    'Start - Declare variables used inside FOR loop
      Dim tickerValue As String   'used to store ticker symbol
      Dim yrOpenPrice As Double   'used to store yearly open price
      Dim yrClosePrice As Double  'used to store yearly closing price
      Dim yrDiff As Double        'used to store difference b/w change in open price and closing price
      Dim percentDiff As Double   'used to store %change
      Dim totStockVol As Double   'used to store total stock volume
    'Ends - Declare variables used inside FOR loop

    'Start - Declare and assign cell reference variables
      Dim startIndex As Long: startIndex = 2 'used as Row index

      Dim colOpenPrice As Long: colOpenPrice = 3     'used as column index
      Dim colClosePrice As Long: colClosePrice = 6   'used as column index
      Dim colTotalVolume As Long: colTotalVolume = 7 'used as column index

      Dim writeCol1 As String: writeCol1 = col1 'used as range reference value
      Dim writeCol2 As String: writeCol2 = col2 'used as range reference value
      Dim writeCol3 As String: writeCol3 = col3 'used as range reference value
      Dim writeCol4 As String: writeCol4 = col4 'used as range reference value
    'Ends - Declare and assign cell reference variables


  'Start - Iteration counter variabels
    Dim rowCounter As Long: rowCounter = 12
    Dim lastRow As Long: lastRow = Worksheets(sheetName).UsedRange.Rows.Count
  'Ends - Iteration counter variabels

  'Start - Calculte one year stock output for each ticker symbol

  ' Proceeds only if the year dataset contains any record/s
  If (lastRow > 1) Then

      'Assigning values to variables
        tickerValue = Worksheets(sheetName).Cells(startIndex, 1).Value
        yrOpenPrice = CDec(Worksheets(sheetName).Cells(startIndex, colOpenPrice).Value)

      'Iterate through each record and calculate one year stock output for each ticker symbol
        For x = startIndex To lastRow

          'Check if ticker value changed
           If (tickerValue <> Worksheets(sheetName).Cells(x, 1).Value) Then

             'Start - Calculate the output for the previous ticker symbol
              yrClosePrice = CDec(Worksheets(sheetName).Cells(x - 1, colClosePrice).Value)
              yrDiff = CDec(yrClosePrice - yrOpenPrice)
              percentDiff = yrDiff / yrOpenPrice
              'End - Calculate the output for the previous ticker symbol

             'Start - Write the yearly calcuated values in summary sheet
             ' #for the previous ticker symbol
              Worksheets(summarySheet).Range(writeCol1 & rowCounter).Value = tickerValue
              Worksheets(summarySheet).Range(writeCol2 & rowCounter).NumberFormat = "0.00"
              Worksheets(summarySheet).Range(writeCol2 & rowCounter).Value = yrDiff
              Worksheets(summarySheet).Range(writeCol3 & rowCounter).Value = FormatPercent(percentDiff, 2)
              Worksheets(summarySheet).Range(writeCol4 & rowCounter).Value = totStockVol
             'Ends - Write the yearly calcuated values in summary sheet

              'Start - Rest values for next iteration
               rowCounter = rowCounter + 1
               tickerValue = Worksheets(sheetName).Cells(x, 1).Value
               yrOpenPrice = CDec(Worksheets(sheetName).Cells(x, colOpenPrice).Value)
```

```
                yrClosePrice = 0
                yrDiff = 0
                percentDiff = 0
                totStockVol = 0
                totStockVol = totStockVol + Worksheets(sheetName).Cells(x, colTotalVolume).Value
               'End - Rest values for next iteration

           'Check if this is the last record
            ElseIf x = lastRow Then
               'Start - Calculate the output for the final ticker symbol
                yrClosePrice = CDec(Worksheets(sheetName).Cells(x, colClosePrice).Value)
                totStockVol = totStockVol + Worksheets(sheetName).Cells(x, colTotalVolume).Value
                yrDiff = CDec(yrClosePrice - yrOpenPrice)
                percentDiff = yrDiff / yrOpenPrice
                'End - Calculate the output for the final ticker symbol

                'Start - Write the last calcuated ticker symbol values in the summary sheet
                Worksheets(summarySheet).Range(writeCol1 & rowCounter).Value = tickerValue
                Worksheets(summarySheet).Range(writeCol2 & rowCounter).NumberFormat = "0.00"
                Worksheets(summarySheet).Range(writeCol2 & rowCounter).Value = yrDiff
                Worksheets(summarySheet).Range(writeCol3 & rowCounter).Value = FormatPercent(percentDiff, 2)
                Worksheets(summarySheet).Range(writeCol4 & rowCounter).Value = totStockVol
                'Ends - Write the last calcuated ticker symbol values in the summary sheet

            Else 'If there are more records for the current ticker symbol
                 'Add the total stock volume
                  totStockVol = totStockVol + Worksheets(sheetName).Cells(x, colTotalVolume).Value
             End If

        Next x

    Else
        'Do nothing - #just information statement here
        'There are no records in this sheet to calculate one year output
    End If

End Sub
```

```
Module3 - 1

'# Reference used for creating this program
'#    - https://learn.microsoft.com/en-us/office/vba/api/overview/
'#    - https://www.homeandlearn.org/excel_vba_practice1.html
'#    - http://dmcritchie.mvps.org/excel/colors.htm
'#    - All class material references
'  ------------------------------------------------------------
'# Author SA
'# VBA-Challenge - Add functionality to your script to return the stock with the
'#  --#Greatest % increase#
'#  --#Greatest % decrease#
'#  --#Greatest total volume#
'# Make sure to use conditional formatting that will highlight
'#  --#positive change in green
'#  --#negative change in red

Sub greatestCalc(col1 As String, col2 As String, col3 As String, col4 As String, summarySheet As String)


    'Start - Iteration counter variabels
      Dim rowStartIndex As Long: rowStartIndex = 12

      Dim tickerCol As String: tickerCol = col1         'used as range reference value
      Dim yrlyChngCol As String: yrlyChngCol = col2     'used as range reference value
      Dim percentCol As String: percentCol = col3       'used as range reference value
      Dim totVolumeCol As String: totVolumeCol = col4   'used as range reference value

      'get total count of available records for processing
      Dim gcEndRow As Long: gcEndRow = Worksheets(summarySheet).Range(tickerCol & Rows.Count).End(xlUp).Row
    'Ends - Iteration counter variabels

    'Start - Declare variables used inside FOR loop
      Dim tickerValue As String   'used to store ticker symbol
      Dim yrlyChngPrice As Double 'used to store yearly Chng price
      Dim yrlyPrcntChng As Double 'used to store %change
      Dim totStockVol As Double   'used to store total stock volume

     'Start - Declare variables for greatest value calculation
      Dim tickerGPD As String
      Dim greatestPcntDecrease As Double
      Dim tickerGPI As String
      Dim greatestPcntIncrease As Double
      Dim tickerGTSV As String
      Dim greatestTotVolume As Double
     'Ends - Declare variables for greatest value calculation
    'Ends - Declare variables used inside FOR loop

    'Proceeds only if the year dataset contains any record/s
    If (gcEndRow >= rowStartIndex) Then

       'Setting up intial values
        tickerGPD = Worksheets(summarySheet).Range(tickerCol & rowStartIndex).Value
        greatestPcntDecrease = Worksheets(summarySheet).Range(percentCol & rowStartIndex).Value
        tickerGPI = Worksheets(summarySheet).Range(tickerCol & rowStartIndex).Value
        greatestPcntIncrease = Worksheets(summarySheet).Range(percentCol & rowStartIndex).Value
        tickerGTSV = Worksheets(summarySheet).Range(tickerCol & rowStartIndex).Value
        greatestTotVolume = Worksheets(summarySheet).Range(totVolumeCol & rowStartIndex).Value

        'Iterate through each record and greatest values
          For x = rowStartIndex To gcEndRow

            'Assigning values to variables
             tickerValue = Worksheets(summarySheet).Range(tickerCol & x).Value
             yrlyChngPrice = Worksheets(summarySheet).Range(yrlyChngCol & x).Value
             yrlyPrcntChng = Worksheets(summarySheet).Range(percentCol & x).Value
             totStockVol = Worksheets(summarySheet).Range(totVolumeCol & x).Value

            'Applying conditional formatting to the yearly change column
             If yrlyChngPrice < 0 Then
                'Apply red color for negative values
                 Worksheets(summarySheet).Range(yrlyChngCol & x).Interior.ColorIndex = 3
               Else
                'Apply green color for positive values
                 Worksheets(summarySheet).Range(yrlyChngCol & x).Interior.ColorIndex = 4
             End If

            'Applying conditional formatting to the percent change column
             If yrlyPrcntChng < 0 Then
```

```
                'Apply red color for negative values
                 Worksheets(summarySheet).Range(percentCol & x).Interior.ColorIndex = 3
               Else
                'Apply green color for positive values
                 Worksheets(summarySheet).Range(percentCol & x).Interior.ColorIndex = 4
             End If

            'Checking and setting up greatest percent decrease value
             If yrlyPrcntChng < greatestPcntDecrease Then
                'Set the greatest percent decrease value
                 greatestPcntDecrease = yrlyPrcntChng
                 tickerGPD = tickerValue

            'Checking and setting up greatest percent Increase value
             ElseIf yrlyPrcntChng > greatestPcntIncrease Then
                'Set the greatest percent increase value
                 greatestPcntIncrease = yrlyPrcntChng
                 tickerGPI = tickerValue
             End If

            'Checking and setting up greatest total stock volumne value
             If totStockVol > greatestTotVolume Then
                'Set the greatest total stock volume value
                 greatestTotVolume = totStockVol
                 tickerGTSV = tickerValue
             End If

          Next x

      'write the greatest values to the summary sheet
        Worksheets(summarySheet).Range(percentCol & rowStartIndex - 5).Value = tickerGPI
        Worksheets(summarySheet).Range(totVolumeCol & rowStartIndex - 5).Value = FormatPercent(greatestPcntIncrease, 2)

      'Applying conditional formatting to greatest percent Increase cell
        If greatestPcntIncrease < 0 Then
          Worksheets(summarySheet).Range(totVolumeCol & rowStartIndex - 5).Interior.ColorIndex = 3
        Else
          Worksheets(summarySheet).Range(totVolumeCol & rowStartIndex - 5).Interior.ColorIndex = 4
        End If

      'write the greatest values to the summary sheet
        Worksheets(summarySheet).Range(percentCol & rowStartIndex - 4).Value = tickerGPD
        Worksheets(summarySheet).Range(totVolumeCol & rowStartIndex - 4).Value = FormatPercent(greatestPcntDecrease, 2)

      'Applying conditional formatting to greatest percent decrease cell
        If greatestPcntDecrease < 0 Then
          Worksheets(summarySheet).Range(totVolumeCol & rowStartIndex - 4).Interior.ColorIndex = 3
        Else
          Worksheets(summarySheet).Range(totVolumeCol & rowStartIndex - 4).Interior.ColorIndex = 4
        End If

      'write the greatest values to the summary sheet
        Worksheets(summarySheet).Range(percentCol & rowStartIndex - 3).Value = tickerGTSV
        Worksheets(summarySheet).Range(totVolumeCol & rowStartIndex - 3).Value = greatestTotVolume

      Else
          'Do nothing - #just information statement here
          'There are no records in this sheet to calculate one year output
      End If

End Sub
```