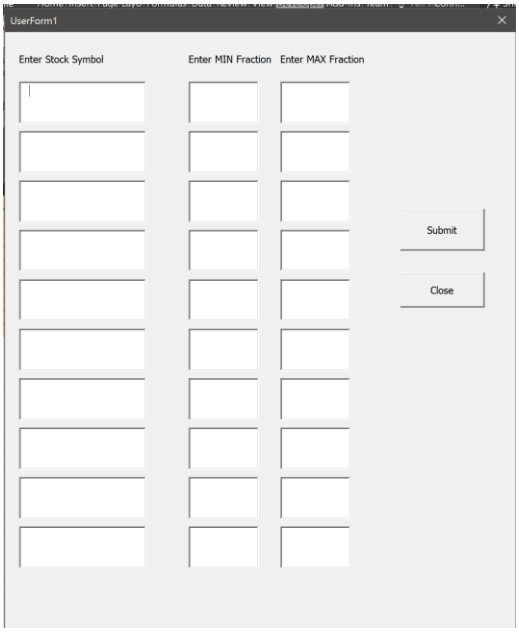
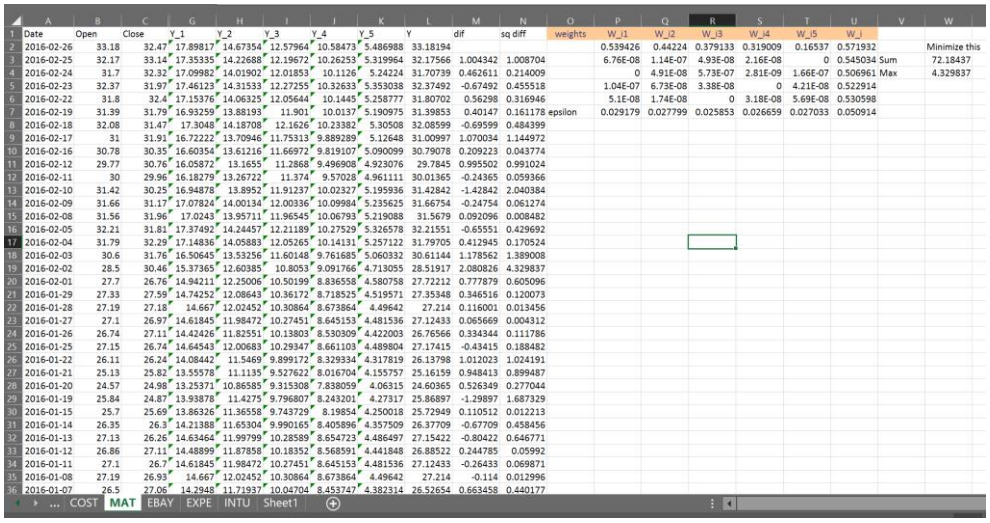
In this assignment, we put together the previous 6 assignments into one decision support system. We assume the user inputs valid NASDAQ stock symbols and does not skip any lines when inputting.



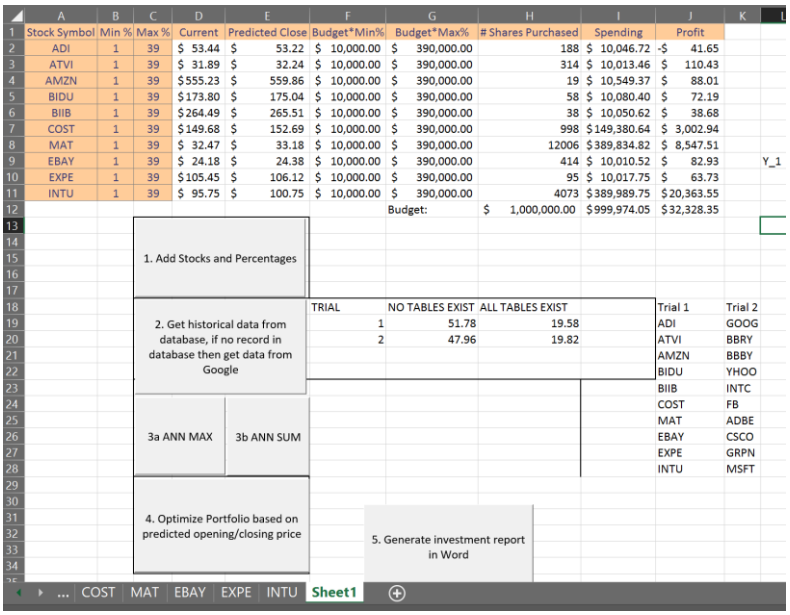
After they input their stocks and max/min percentages, the user presses the second button which pulls historical data (minimum of one year from today) from the database which is updated using data from Google finance. The data for each stock symbol is pasted onto an individual sheet (Assignments 3 and 4). The user then presses button 3a or 3b from Assignment 6 which creates an artificial neural network (ANN) to predict the opening price of each stock for the next day.



Date	Open	Close	Y_1	Y_2	Y_3	Y_4	Y_5	Y	dif	sq diff	weights	W_i1	W_i2	W_i3	W_i4	W_i5	W_i		W
2016-02-26	33.18	32.47	17.89817	14.67354	12.57964	10.58473	5.486988	33.18194				0.539426	0.44224	0.379133	0.319009	0.16537	0.571932		Minimize this
2016-02-25	32.17	33.14	17.35335	14.22688	12.19672	10.26253	5.319964	32.17566	1.004342	1.008704		6.76E-08	1.14E-07	4.93E-08	2.16E-08	0	0.545034	Sum	72.18437
2016-02-24	31.7	32.32	17.09982	14.01902	12.01853	10.1126	5.24224	31.70739	0.462611	0.214009		0	4.91E-08	5.73E-07	2.81E-09	1.66E-07	0.506961	Max	4.329837
2016-02-23	32.37	31.97	17.46123	14.31533	12.27255	10.32633	5.353038	32.37492	-0.67492	0.455518		1.04E-07	6.73E-08	3.38E-08	0	4.21E-08	0.522914		
2016-02-22	31.8	32.4	17.15376	14.06325	12.05644	10.1445	5.258777	31.80702	0.56298	0.316946		5.1E-08	1.74E-08	0	3.18E-08	5.69E-08	0.530598		
2016-02-19	31.39	31.79	16.93259	13.88193	11.901	10.0137	5.190975	31.39853	0.40147	0.161178	epsilon	0.029179	0.027799	0.025853	0.026659	0.027033	0.050914		
2016-02-18	32.08	31.47	17.3048	14.18708	12.1626	10.23382	5.30508	32.08599	-0.69599	0.484399									
2016-02-17	31	31.91	16.72222	13.70946	11.75313	9.889289	5.12648	31.00997	1.070034	1.144972									
2016-02-16	30.78	30.35	16.60354	13.61216	11.66972	9.819107	5.090099	30.79078	0.209223	0.043774									
2016-02-12	29.77	30.76	16.05872	13.1655	11.2868	9.496908	4.923076	29.7845	0.995502	0.991024									
2016-02-11	30	29.96	16.18279	13.26722	11.374	9.57028	4.961111	30.01365	-0.24365	0.059366									
2016-02-10	31.42	30.25	16.94878	13.8952	11.91237	10.02327	5.195936	31.42842	-1.42842	2.040384									
2016-02-09	31.66	31.17	17.07824	14.00134	12.00336	10.09984	5.235625	31.66754	-0.24754	0.061274									
2016-02-08	31.56	31.96	17.0243	13.95711	11.96545	10.06793	5.219088	31.5679	0.092096	0.008482									
2016-02-05	32.21	31.81	17.37492	14.24457	12.21189	10.27529	5.326578	32.21551	-0.65551	0.429692									
2016-02-04	31.79	32.29	17.14836	14.05883	12.05265	10.14131	5.257122	31.79705	0.412945	0.170524									
2016-02-03	30.6	31.76	16.50645	13.53256	11.60148	9.761685	5.060332	30.61144	1.178562	1.389008									
2016-02-02	28.5	30.46	15.37365	12.60385	10.8053	9.091766	4.713055	28.51917	2.080826	4.329837									
2016-02-01	27.7	26.76	14.94211	12.25006	10.50199	8.836558	4.580758	27.72212	0.777879	0.605096									
2016-01-29	27.33	27.59	14.74252	12.08643	10.36172	8.718525	4.519571	27.35348	0.346516	0.120073									
2016-01-28	27.19	27.18	14.667	12.02452	10.30864	8.673864	4.49642	27.214	0.116001	0.013456									
2016-01-27	27.1	26.97	14.61845	11.98472	10.27451	8.645153	4.481536	27.12433	0.065669	0.004312									
2016-01-26	26.74	27.11	14.42426	11.82551	10.13803	8.530309	4.422003	26.76566	0.334344	0.111786									
2016-01-25	27.15	26.74	14.64543	12.00683	10.29347	8.661103	4.489804	27.17415	-0.43415	0.188482									
2016-01-22	26.11	26.24	14.08442	11.5469	9.899172	8.329334	4.317819	26.13798	1.012023	1.024191									
2016-01-21	25.13	25.82	13.55578	11.1135	9.527622	8.016704	4.155757	25.16159	0.948413	0.899487									
2016-01-20	24.57	24.98	13.25371	10.86585	9.315308	7.838059	4.06315	24.60365	0.526349	0.277044									
2016-01-19	25.84	24.87	13.93878	11.4275	9.796807	8.243201	4.27317	25.86897	-1.29897	1.687329									
2016-01-15	25.7	25.69	13.86326	11.36558	9.743729	8.19854	4.250018	25.72949	0.110512	0.012213									
2016-01-14	26.35	26.3	14.21388	11.65304	9.990165	8.405896	4.357509	26.37709	-0.67709	0.458456									
2016-01-13	27.13	26.26	14.63464	11.99799	10.28589	8.654723	4.486497	27.15422	-0.80422	0.646771									
2016-01-12	26.86	27.11	14.48899	11.87858	10.18352	8.568591	4.441848	26.88522	0.244785	0.05992									
2016-01-11	27.1	26.7	14.61845	11.98472	10.27451	8.645153	4.481536	27.12433	-0.26433	0.069871									
2016-01-08	27.19	26.93	14.667	12.02452	10.30864	8.673864	4.49642	27.214	-0.114	0.012996									
2016-01-07	26.5	27.06	14.2948	11.71937	10.04704	8.453747	4.382314	26.52654	0.663458	0.440177									

Button 3a predicts by minimizing the sum of squared errors for each stock while 3b minimizes the minimum value of the squared errors for each stock. The user clicks button 4 which runs the solver macro from

Assignment 2 to take the predicted and current values (opening and closing values) to maximize the profit given the users input budget and portfolio criteria.



Stock Symbol	Min %	Max %	Current	Predicted Close	Budget*Min%	Budget*Max%	# Shares Purchased	Spending	Profit
ADI	1	39	$ 53.44	$ 53.22	$ 10,000.00	$ 390,000.00	188	$ 10,046.72	-$ 41.65
ATVI	1	39	$ 31.89	$ 32.24	$ 10,000.00	$ 390,000.00	314	$ 10,013.46	$ 110.43
AMZN	1	39	$555.23	$ 559.86	$ 10,000.00	$ 390,000.00	19	$ 10,549.37	$ 88.01
BIDU	1	39	$173.80	$ 175.04	$ 10,000.00	$ 390,000.00	58	$ 10,080.40	$ 72.19
BIIB	1	39	$264.49	$ 265.51	$ 10,000.00	$ 390,000.00	38	$ 10,050.62	$ 38.68
COST	1	39	$149.68	$ 152.69	$ 10,000.00	$ 390,000.00	998	$149,380.64	$ 3,002.94
MAT	1	39	$ 32.47	$ 33.18	$ 10,000.00	$ 390,000.00	12006	$389,834.82	$ 8,547.51
EBAY	1	39	$ 24.18	$ 24.38	$ 10,000.00	$ 390,000.00	414	$ 10,010.52	$ 82.93
EXPE	1	39	$105.45	$ 106.12	$ 10,000.00	$ 390,000.00	95	$ 10,017.75	$ 63.73
INTU	1	39	$ 95.75	$ 100.75	$ 10,000.00	$ 390,000.00	4073	$389,989.75	$20,363.55
						Budget:	$ 1,000,000.00	$999,974.05	$32,328.35

1. Add Stocks and Percentages

2. Get historical data from database, if no record in database then get data from Google

3a ANN MAX

3b ANN SUM

4. Optimize Portfolio based on predicted opening/closing price

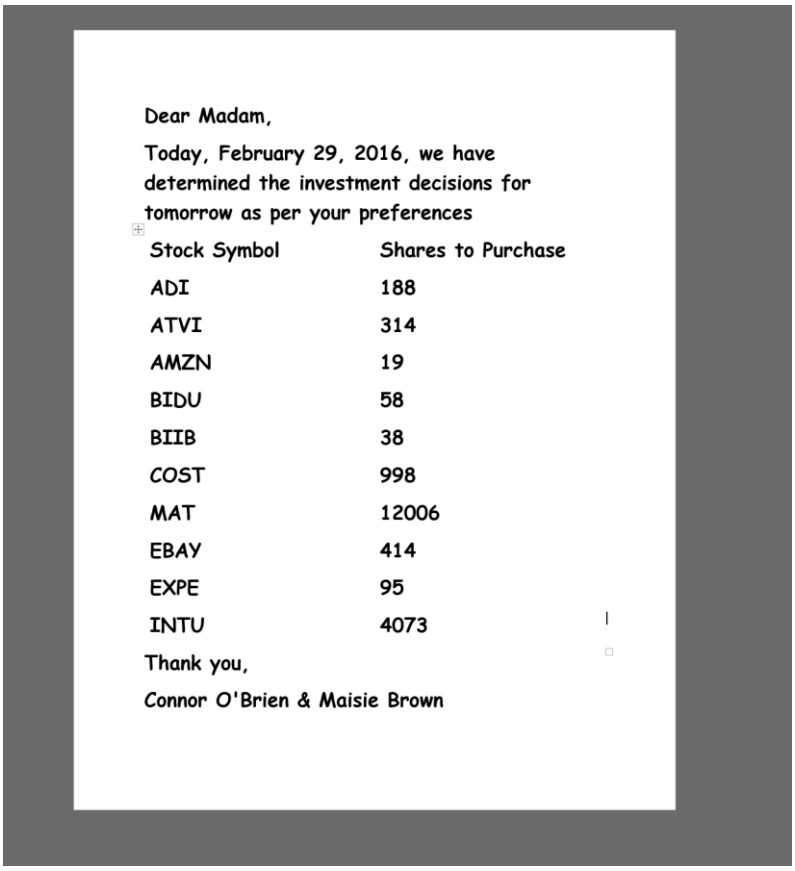
5. Generate investment report in Word

TRIAL	NO TABLES EXIST	ALL TABLES EXIST
1	51.78	19.58
2	47.96	19.82

Trial 1	Trial 2
ADI	GOOG
ATVI	BBRY
AMZN	BBBY
BIDU	YHOO
BIIB	INTC
COST	FB
MAT	ADBE
EBAY	CSCO
EXPE	GRPN
INTU	MSFT

... COST MAT EBAY EXPE INTU Sheet1

Finally, the user presses button 5 which takes the optimal investment decision and outputs it to a formal Word document that can be handed to the user as a recommended investment report for the next day.



Dear Madam,

Today, February 29, 2016, we have determined the investment decisions for tomorrow as per your preferences

Stock Symbol	Shares to Purchase
ADI	188
ATVI	314
AMZN	19
BIDU	58
BIIB	38
COST	998
MAT	12006
EBAY	414
EXPE	95
INTU	4073

Thank you,

Connor O'Brien & Maisie Brown

VBA CODE

```
Sub addstocks()
   Dim row As Integer

   Sheets("Sheet1").Select
   row = 2
   Range("A2").Select
   Do While ActiveCell.Value <> ""
      Range("A" & row).Value = ""
      Range("B" & row).Value = ""
      Range("C" & row).Value = ""
      row = row + 1
      Range("A" & row).Select
   Loop
   Range("A1").Select
   UserForm1.Show
End Sub

Sub addupdatedata()
Application.ScreenUpdating = False
Dim space As DAO.Workspace
Dim db As DAO.Database
Dim stock As String
Dim row As Integer
Dim newtable As DAO.TableDef
Dim i As Integer
Dim rs As DAO.Recordset
Dim tableexists As Boolean
Dim latest As Date
Dim start As Double
Dim elapsed As Double
Dim sheet As Object
Dim sqlstring As String
start = Timer

row = 2
Sheets("Sheet1").Select
Range("A2").Select
Set space = DBEngine.Workspaces(0)
Set db = DAO.OpenDatabase(Application.ActiveWorkbook.Path & "\miniProject4_db.accdb")
'Clear existing sheets and data
Dim ws As Worksheet
For Each ws In Worksheets
   If ws.Name <> "Sheet1" Then ws.Delete
Next

Do While ActiveCell.Value <> ""
   tableexists = False
   stock = ActiveCell.Value
   'Check if table exists
   For i = 0 To db.TableDefs.Count - 1
      If stock = db.TableDefs(i).Name Then
         tableexists = True
         Exit For
      End If
   Next i

   If tableexists = False Then
      'Create new table
      Set newtable = db.CreateTableDef(stock)
      With newtable
         .Fields.Append .CreateField("Date", dbDate)
         .Fields.Append .CreateField("Open", dbCurrency)
         .Fields.Append .CreateField("Close", dbCurrency)
```

```
    End With
    db.TableDefs.Append newtable
    'Create and Populate new sheet with Google data
    Workbooks.Open Filename:="http://www.google.com/finance/historical?q=NASDAQ:" + stock + "&output=csv"
    ActiveWindow.Visible = False
    Windows("historical").Visible = True
    Range("A1:F1").Select
    Range(Selection, Selection.End(xlDown)).Select
    Selection.Copy
    Windows("historical").Visible = False
    Sheets.Add.Name = stock
    ActiveSheet.Paste
    Application.CutCopyMode = False
    Windows("historical").Activate
    ActiveWindow.Close savechanges:=False
    'Populate table with worksheet data
    Worksheets(stock).Select
    Range("A2").Select
    Selection.End(xlDown).Select
    Set rs = db.OpenRecordset(stock, dbOpenTable)
    Do While ActiveCell.row <> 1
       rs.AddNew
       rs.Fields("Date") = ActiveCell.Value
       rs.Fields("Open") = ActiveCell.Offset(0, 1).Value
       rs.Fields("Close") = ActiveCell.Offset(0, 4).Value
       ActiveCell.Offset(-1, 0).Select
       rs.Update
    Loop
    Columns(3).EntireColumn.Delete
    Columns(3).EntireColumn.Delete
    Columns(4).EntireColumn.Delete
End If

If tableexists = True Then
    'Update database
    Workbooks.Open Filename:="http://www.google.com/finance/historical?q=NASDAQ:" + stock + "&output=csv"
    Windows("historical").Visible = True
    Range("A2").Select
    Set rs = db.OpenRecordset(stock, dbOpenTable)
    rs.MoveLast
    latest = rs.Fields("Date").Value
    Do While ActiveCell.Value <> latest
       ActiveCell.Offset(1, 0).Select
    Loop
    ActiveCell.Offset(-1, 0).Select
    Do While ActiveCell.row <> 1
       rs.AddNew
       rs.Fields("Date") = ActiveCell.Value
       rs.Fields("Open") = ActiveCell.Offset(0, 1).Value
       rs.Fields("Close") = ActiveCell.Offset(0, 4).Value
       ActiveCell.Offset(-1, 0).Select
       rs.Update
    Loop
    ActiveWindow.Close savechanges:=False
    Windows("All Assignments FINAL 3.0.xlsm").Activate
    'Populate worksheet with Database data
    Sheets.Add.Name = stock
    Set sheet = Sheets(stock)
    sheet.Activate
    For i = 0 To rs.Fields.Count - 1
       sheet.Cells(1, i + 1).Value = rs.Fields(i).Name
    Next i
    sqlstring = "SELECT * FROM " & stock & " ORDER by Date DESC"
    Set rs = db.OpenRecordset(sqlstring)
    sheet.Range("A2").CopyFromRecordset rs
```

```
   End If

   row = row + 1
   Worksheets("Sheet1").Select
   Range("A" & row).Select
Loop

db.Close
space.Close
Access.Application.Quit
elapsed = Round(Timer - start, 2)
Application.ScreenUpdating = True
MsgBox "This macro took " & elapsed & " seconds to complete"
End Sub

Sub exporttoword()

Dim wa As Word.Application
Dim i As Integer
Dim deleted As Integer

Set wa = CreateObject("Word.Application")
Set doc = wa.Documents.Add

doc.Styles.Add ("Text")
With doc.Styles("Text").Font
   .Name = "Comic Sans MS"
   .Size = 20
   .Bold = True
End With
Set sel = wa.Selection
sel.Style = doc.Styles("Text")

doc.Content.InsertAfter ("Dear Madam,")
doc.Range.InsertAfter Chr(10)
doc.Content.InsertAfter ("Today, " & Format(Now, "mmmm d, yyyy") & ", we have determined the investment decisions for tomorrow as per
your preferences.")
doc.Range.InsertAfter Chr(10)

Range("A2").Select
Set tablerange = doc.Range(118)
doc.Tables.Add tablerange, 11, 2
Set tbl = doc.Tables(1)
tbl.Borders.Enable = False
deleted = 0
i = 0
tbl.Cell(1, 1).Range.Text = "Stock Symbol"
tbl.Cell(1, 2).Range.Text = "Shares to Purchase"

For i = 1 To 10
   tbl.Cell(i + 1, 1).Range.Text = ActiveCell.Value
   If ActiveCell.Offset(0, 7) > 0 Then
      tbl.Cell(i + 1, 2).Range.Text = ActiveCell.Offset(0, 7)
   Else
      tbl.Cell(i + 1, 2).Range.Text = "No profit could be made on this stock"
   End If
   If ActiveCell.Value = "" Then
      tbl.Rows(i - deleted + 1).Delete
      deleted = deleted + 1
   End If
   ActiveCell.Offset(1, 0).Select
Next i
doc.Content.InsertAfter ("Thank you,")
doc.Range.InsertAfter Chr(10)
doc.Content.InsertAfter ("Connor O'Brien & Maisie Brown")
```

```
wa.Visible = True
wa.WindowState = wdWindowStateMaximize

End Sub


Sub ANNSUMbystock()
Application.ScreenUpdating = False
Dim row As Integer
Dim symbol As String
Dim holder As Double
Dim bottom As Integer
Dim start As Double
Dim elapsed As Double
Dim sumstring As String
sumstring = "=SUM("
start = Timer
row = 1
Sheets("Sheet1").Select
Range("$N$2:$S$7").Clear
Range("A" & row).Select

Do While ActiveCell.Offset(1, 0) <> ""
   ActiveCell.Offset(1, 0).Select
   symbol = ActiveCell.Value
   sumstring = sumstring & symbol & "!W4, "
   Range("L9:S9").Select
   Selection.Copy
   Sheets(symbol).Select
   Range("G1").Select
   ActiveSheet.Paste
   Application.CutCopyMode = False
   Sheets("Sheet1").Select
   Range("M1:U7").Select
   Selection.Copy
   Sheets(symbol).Select
   Range("O1").Select
   ActiveSheet.Paste
   Application.CutCopyMode = False
   Range("B2").Select
   Selection.End(xlDown).Select
   bottom = ActiveCell.row
   Range("W3").Formula = "=SUM(N3:N" & bottom & ")"
   Range("W4").Formula = "=MAX(N3:N" & bottom & ")"
   Range("P2").Formula = "=Sheet1!N2"
   Range("P2").Select
   Selection.AutoFill Destination:=Range("P2:U2"), Type:=xlFillDefault
   Range("P2:U2").Select
   Selection.AutoFill Destination:=Range("P2:U7"), Type:=xlFillDefault
   Range("F" & bottom).Select
   ActiveCell.Offset(-4, 0).Select
   ActiveCell.Offset(0, 1).Formula = "=SUMPRODUCT($B" & bottom & ":$B" & bottom - 4 & ",P$2:P$6)"
   ActiveCell.Offset(0, 2).Formula = "=SUMPRODUCT($B" & bottom & ":$B" & bottom - 4 & ",Q$2:Q$6)"
   ActiveCell.Offset(0, 3).Formula = "=SUMPRODUCT($B" & bottom & ":$B" & bottom - 4 & ",R$2:R$6)"
   ActiveCell.Offset(0, 4).Formula = "=SUMPRODUCT($B" & bottom & ":$B" & bottom - 4 & ",S$2:S$6)"
   ActiveCell.Offset(0, 5).Formula = "=SUMPRODUCT($B" & bottom & ":$B" & bottom - 4 & ",T$2:T$6)"
   bottom = ActiveCell.row
   ActiveCell.Offset(0, 6).Formula = "=(G" & bottom & "+$P$7)*$U$2+(H" & bottom & "+$Q$7)*$U$3+(I" & bottom & "+$R$7)*$U$4+(J" &
bottom & "+$S$7)*$U$5+(K" & bottom & "+$T$7)*$U$6+$U$7"
   ActiveCell.Offset(0, 7).Formula = "=B" & bottom - 1 & "-L" & bottom
   ActiveCell.Offset(0, 8).Formula = "=M" & bottom & "^2"
   Range("G" & bottom & ":N" & bottom & "").Select
   Selection.AutoFill Destination:=Range("G3:N" & bottom), Type:=xlFillDefault
   Range("G3:L3").Select
   Selection.AutoFill Destination:=Range("G2:L3"), Type:=xlFillDefault
   Range("G2:L3").Select
```

```
    ActiveSheet.Calculate
    holder = Range("W3").Value
    SolverReset
    SolverOk SetCell:="$W$3", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
       Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverOptions MaxTime:=0, Iterations:=0, Precision:=0.000001, Convergence:= _
       0.0001, StepThru:=False, Scaling:=False, AssumeNonNeg:=True, Derivatives:=2
    SolverOptions PopulationSize:=100, RandomSeed:=0, MutationRate:=0.075, Multistart _
       :=True, RequireBounds:=False, MaxSubproblems:=0, MaxIntegerSols:=0, _
       IntTolerance:=1, SolveWithout:=False, MaxTimeNoImp:=30
    SolverOk SetCell:="$W$3", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
       Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverAdd CellRef:="$P$2:$U$7", Relation:=3, FormulaText:="0.00000001"
    SolverOk SetCell:="$W$3", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
       Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverOk SetCell:="$W$3", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
       Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverSolve userfinish:=True

    Do While holder > Range("W3").Value
       holder = Range("W3").Value
       SolverReset
       SolverOk SetCell:="$W$3", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
          Engine:=1, EngineDesc:="GRG Nonlinear"
       SolverOptions MaxTime:=0, Iterations:=0, Precision:=0.000001, Convergence:= _
          0.0001, StepThru:=False, Scaling:=False, AssumeNonNeg:=True, Derivatives:=2
       SolverOptions PopulationSize:=100, RandomSeed:=0, MutationRate:=0.075, Multistart _
          :=True, RequireBounds:=False, MaxSubproblems:=0, MaxIntegerSols:=0, _
          IntTolerance:=1, SolveWithout:=False, MaxTimeNoImp:=30
       SolverOk SetCell:="$W$3", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
          Engine:=1, EngineDesc:="GRG Nonlinear"
       SolverOk SetCell:="$W$3", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
          Engine:=1, EngineDesc:="GRG Nonlinear"
       SolverOk SetCell:="$W$3", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
          Engine:=1, EngineDesc:="GRG Nonlinear"
       SolverSolve userfinish:=True
    Loop
    row = row + 1
    Sheets("Sheet1").Select
    Range("A" & row).Select
Loop

sumstring = sumstring & "0)"
Sheets("Sheet1").Select
ActiveSheet.Range("J16").Formula = sumstring

Sheets("sheet1").Select
Range("A2").Select
row = ActiveCell.row
Do While ActiveCell.Value <> ""
    symbol = ActiveCell.Value
    Sheets(symbol).Select
    Range("L2").Select
    Selection.Copy
    Sheets("Sheet1").Select
    Range("E" & row).PasteSpecial xlPasteValues
    Application.CutCopyMode = False
    Sheets(symbol).Select
    Range("C2").Select
    Selection.Copy
    Sheets("Sheet1").Select
    Range("D" & row).PasteSpecial xlPasteValues
    Application.CutCopyMode = False
    row = row + 1
    Range("A" & row).Select
```

```
Loop
elapsed = Round(Timer - start, 2)
MsgBox "This macro took " & elapsed & " seconds to complete"


Application.ScreenUpdating = True
End Sub
Sub ANNMAXbystock()
Application.ScreenUpdating = False
Dim row As Integer
Dim symbol As String
Dim holder As Double
Dim bottom As Integer
Dim start As Double
Dim elapsed As Double
Dim sumstring As String
sumstring = "=SUM("
start = Timer
row = 1
Sheets("Sheet1").Select
Range("$N$2:$S$7").Clear
Range("A" & row).Select

Do While ActiveCell.Offset(1, 0) <> ""
   ActiveCell.Offset(1, 0).Select
   symbol = ActiveCell.Value
   sumstring = sumstring & symbol & "!W4, "
   Range("L9:S9").Select
   Selection.Copy
   Sheets(symbol).Select
   Range("G1").Select
   ActiveSheet.Paste
   Application.CutCopyMode = False
   Sheets("Sheet1").Select
   Range("M1:U7").Select
   Selection.Copy
   Sheets(symbol).Select
   Range("O1").Select
   ActiveSheet.Paste
   Application.CutCopyMode = False
   Range("B2").Select
   Selection.End(xlDown).Select
   bottom = ActiveCell.row
   Range("W3").Formula = "=SUM(N3:N" & bottom & ")"
   Range("W4").Formula = "=MAX(N3:N" & bottom & ")"
   Range("P2").Formula = "=Sheet1!N2"
   Range("P2").Select
   Selection.AutoFill Destination:=Range("P2:U2"), Type:=xlFillDefault
   Range("P2:U2").Select
   Selection.AutoFill Destination:=Range("P2:U7"), Type:=xlFillDefault
   Range("F" & bottom).Select
   ActiveCell.Offset(-4, 0).Select
   ActiveCell.Offset(0, 1).Formula = "=SUMPRODUCT($B" & bottom & ":$B" & bottom - 4 & ",P$2:P$6)"
   ActiveCell.Offset(0, 2).Formula = "=SUMPRODUCT($B" & bottom & ":$B" & bottom - 4 & ",Q$2:Q$6)"
   ActiveCell.Offset(0, 3).Formula = "=SUMPRODUCT($B" & bottom & ":$B" & bottom - 4 & ",R$2:R$6)"
   ActiveCell.Offset(0, 4).Formula = "=SUMPRODUCT($B" & bottom & ":$B" & bottom - 4 & ",S$2:S$6)"
   ActiveCell.Offset(0, 5).Formula = "=SUMPRODUCT($B" & bottom & ":$B" & bottom - 4 & ",T$2:T$6)"
   bottom = ActiveCell.row
   ActiveCell.Offset(0, 6).Formula = "=(G" & bottom & "+$P$7)*$U$2+(H" & bottom & "+$Q$7)*$U$3+(I" & bottom & "+$R$7)*$U$4+(J" &
bottom & "+$S$7)*$U$5+(K" & bottom & "+$T$7)*$U$6+$U$7"
   ActiveCell.Offset(0, 7).Formula = "=B" & bottom - 1 & "-L" & bottom
   ActiveCell.Offset(0, 8).Formula = "=M" & bottom & "^2"
   Range("G" & bottom & ":N" & bottom & "").Select
   Selection.AutoFill Destination:=Range("G3:N" & bottom), Type:=xlFillDefault
   Range("G3:L3").Select
   Selection.AutoFill Destination:=Range("G2:L3"), Type:=xlFillDefault
```

```
    Range("G2:L3").Select

    ActiveSheet.Calculate
    holder = Range("W4").Value
    SolverReset
    SolverOk SetCell:="$W$4", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
      Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverOptions MaxTime:=0, Iterations:=0, Precision:=0.000001, Convergence:= _
      0.0001, StepThru:=False, Scaling:=False, AssumeNonNeg:=True, Derivatives:=2
    SolverOptions PopulationSize:=100, RandomSeed:=0, MutationRate:=0.075, Multistart _
      :=True, RequireBounds:=False, MaxSubproblems:=0, MaxIntegerSols:=0, _
      IntTolerance:=1, SolveWithout:=False, MaxTimeNoImp:=30
    SolverOk SetCell:="$W$4", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
      Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverAdd CellRef:="$P$2:$U$7", Relation:=3, FormulaText:="0.00000001"
    SolverOk SetCell:="$W$4", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
      Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverOk SetCell:="$W$3", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
      Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverSolve userfinish:=True

    Do While holder > Range("W4").Value
      holder = Range("W4").Value
      SolverReset
      SolverOk SetCell:="$W$4", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
        Engine:=1, EngineDesc:="GRG Nonlinear"
      SolverOptions MaxTime:=0, Iterations:=0, Precision:=0.000001, Convergence:= _
        0.0001, StepThru:=False, Scaling:=False, AssumeNonNeg:=True, Derivatives:=2
      SolverOptions PopulationSize:=100, RandomSeed:=0, MutationRate:=0.075, Multistart _
        :=True, RequireBounds:=False, MaxSubproblems:=0, MaxIntegerSols:=0, _
        IntTolerance:=1, SolveWithout:=False, MaxTimeNoImp:=30
      SolverOk SetCell:="$W$4", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
        Engine:=1, EngineDesc:="GRG Nonlinear"
      SolverOk SetCell:="$W$4", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
        Engine:=1, EngineDesc:="GRG Nonlinear"
      SolverOk SetCell:="$W$4", MaxMinVal:=2, ValueOf:=0, ByChange:="$P$2:$U$7", _
        Engine:=1, EngineDesc:="GRG Nonlinear"
      SolverSolve userfinish:=True
    Loop
    row = row + 1
    Sheets("Sheet1").Select
    Range("A" & row).Select
Loop

sumstring = sumstring & "0)"
Sheets("Sheet1").Select
ActiveSheet.Range("J16").Formula = sumstring

Sheets("sheet1").Select
Range("A2").Select
row = ActiveCell.row
Do While ActiveCell.Value <> ""
    symbol = ActiveCell.Value
    Sheets(symbol).Select
    Range("L2").Select
    Selection.Copy
    Sheets("Sheet1").Select
    Range("E" & row).PasteSpecial xlPasteValues
    Application.CutCopyMode = False
    Sheets(symbol).Select
    Range("C2").Select
    Selection.Copy
    Sheets("Sheet1").Select
    Range("D" & row).PasteSpecial xlPasteValues
    Application.CutCopyMode = False
    row = row + 1
```

```
   Range("A" & row).Select
Loop
elapsed = Round(Timer - start, 2)
MsgBox "This macro took " & elapsed & " seconds to complete"


Application.ScreenUpdating = True
End Sub

Sub SOLVAasst2()
'
' SOLVA Macro
'

'
   Range("H2:H11").Clear
   Dim check As Double
   Dim start As Double
   Dim elapsed As Double
   elapsed = 0
   start = Timer
   ActiveSheet.Calculate
   check = 1
   Do While check <> Range("J12").Value

   SolverReset
   SolverOk SetCell:="$J$12", MaxMinVal:=1, ValueOf:=0, ByChange:="$H$2:$H$11", _
      Engine:=1, EngineDesc:="GRG Nonlinear"
   SolverAdd CellRef:="$I$2:$I$11", Relation:=1, FormulaText:="$G$2:$G$11"
   SolverAdd CellRef:="$I$2:$I$11", Relation:=3, FormulaText:="$F$2:$F$11"
   SolverAdd CellRef:="$I$12", Relation:=1, FormulaText:="$H$12"
   SolverAdd CellRef:="$H$2:$H$11", Relation:=4, FormulaText:="integer"
   SolverOk SetCell:="$J$12", MaxMinVal:=1, ValueOf:=0, ByChange:="$H$2:$H$11", _
      Engine:=1, EngineDesc:="GRG Nonlinear"
   SolverOptions MaxTime:=60, Iterations:=0, Precision:=0.000001, Convergence:= _
      0.0001, StepThru:=False, Scaling:=True, AssumeNonNeg:=True, Derivatives:=1
   SolverOptions MaxTimeNoImp:=30
   SolverOk SetCell:="$J$12", MaxMinVal:=1, ValueOf:=0, ByChange:="$H$2:$H$11", _
      Engine:=1, EngineDesc:="GRG Nonlinear"
   SolverOk SetCell:="$J$12", MaxMinVal:=1, ValueOf:=0, ByChange:="$H$2:$H$11", _
      Engine:=1, EngineDesc:="GRG Nonlinear"
   SolverSolve userfinish:=True
      check = Range("J12").Value
      elapsed = elapsed + Round(Timer - start, 3)
      If elapsed > 120 Then
         MsgBox "Solver could not find a valid solution"
         Exit Do
      End If



   Loop
End Sub
```