Option Explicit

Option Base 1

Private PUB\_ARR(1 To 14) As Variant

Private Const PUB\_TOL\_VAL As Double = 2 ^ 52

'----------------------------------------------------------------------------------------------

'When creating financial models for company valuation, circular references always occur.

'Although the Excel built-in iteration system might generate the correct answer, a

'user-defined function is preferred as we can actively modify the function to improve

'its speed, accuracy and functionality. Further, a user-defined function can be used as

'part of a program that automates building financial models.

'----------------------------------------------------------------------------------------------

'Motivation for Creating Circular Reference Solving Functions

'----------------------------------------------------------------------------------------------

'With financial statement data accessible online, there are programs that aggregate data and

'perform data sanitization to generate historical financial statements. Given this fact, it

'is viable to build a program that automates financial modeling. While most calculations in

'financial models are simple arithmetic, there are several key circular references that require

'identification of the difference equation in order to find the equilibrium solution. Normally,

'the Excel built-in iteration system is used to solve for circular reference. It cannot be

'used once the calculations are conducted off worksheet. Therefore, a user-defined function

'for solving circular reference is required for financial modeling automation.

'----------------------------------------------------------------------------------------------

'Coding a set of Circular Reference Solving Function

'----------------------------------------------------------------------------------------------

'i. Driving philosophy of coding a function to solve circular reference

'----------------------------------------------------------------------------------------------

'The goal of a circular reference solving function is to find the stable equilibrium solution

'of a difference function in the form of x{n+1}=f(x{n},c,v), where x represents the element

'that the function will solve, c is a vector of constants, and v is a vector of variables

'dependent on x{n} and c.

'----------------------------------------------------------------------------------------------

'ii. Structure of a General Circular Reference Solving Function

'----------------------------------------------------------------------------------------------

'There are three parts of each set of circular reference solving functions:

'1) The main body (e.g., Function FINAL\_CASH\_SOLVER)

'2) The objective function (e.g., Function SOLVE\_CASH\_OBJECTIVE\_FUNCTION)

'3) The optimization function (e.g., Function MULLER\_ZERO\_FUNC)

'The main body's purpose is to take parameters, initialize the optimization process, process singular

'calculations, and eventually return the equilibrium solution. The objective function is used to call

'the calculation process portion of the main body. The optimization function calculates the equilibrium

'solution by utilizing a numerical optimization method which will call the objective function as part

'of the iterative process.

'----------------------------------------------------------------------------------------------

'iii. Structure of a Main Body Function

'----------------------------------------------------------------------------------------------

'1) Parameter Intake

'There are three kinds of parameters required for the Main Body Function: constants required to

'perform one iterative process, the value of the sought after element as optional to determine

'whether the Main Body Function is called for the first time, and miscellaneous optional parameters

'to determine the output and optimization set-ups.

'2) Calculation process

'When the main body function is called for the first time (i.e. when there is no value assigned to the

'sought after element), the function will proceed to a subroutine to load the constants into a private

'global array (the subroutine is usually called LOAD\_LINE). Then, the function will proceed into another

'subroutine (normally named SOLVE\_LINE) to call the Optimization function which will return the equilibrium

'solution.

'When the Main Body Function is called with a starting value for the sought after element, the Main Body

'Function will perform one calculation process and produce an ending value of the sought after element. When

'the absolute difference between the starting value and the ending value is less than an error term (normally

'less than 10^-4), the ending value is the equilibrium solution to the circular reference.

'3) Structure of an Objective Function

'Parameter Intake: The Objective Function only takes in a value assigned to the sought after element.

'Calculation process: The Objective Function will call the Main Body Function by passing the element

'value as the starting value of one calculation process.

'4) Structure of an Optimization Function

'Parameter Intake: The Optimization Function takes in the name of the Objective Function in string format,

'and the initial guessing values.

'Calculation process: The calculation process varies depending on the optimization method it utilizes;

'nevertheless, it always involves calling the Objective Function iteratively until there is convergence

'among the ending values of the sought after element.

'Choice of Optimization Method

'Normally, there are two families of optimization methods: the Newton family and the Bisection family. A member

'of the Newton family is the Muller's method, and a member for the Bisection family is the Nelder-Mead simplex

'method. The Newton family typically tries to estimate the equilibrium solution via polynomial curve fitting

'(area for 3-d, etc.) and has faster convergence speeds than the Bisection family. However, lack of convergence

'will occur if the epsilon is too small, too many calculation steps that increases the machine rounding error,

'or the function exhibits small variations with large input perturbation. The Bisection family searches the

'equilibrium solution through zoning. Therefore, once an equilibrium solution is located in a range, the

'optimization process will continuously narrow that range until the equilibrium solution is found; a method from

'the Newton's family may exhibit explosive behavior if the objective function is not structured 'nicely'.

'In practice, Muller's method is always preferred if it can produce viable results with a small enough error

'term; otherwise, the Nelder-Mead simplex method could be utilized to solve for the equilibrium point. In this course

'we will only cover the Newton family.

'----------------------------------------------------------------------------------------------

Sub Section4\_Solve\_Cash()

'------------------------------------------------------------------------------------------

'Previous Fiscal Year

'------------------------------------------------------------------------------------------

Dim PREV\_CASH\_VAL As Double

Dim PREV\_LT\_DEBT\_VAL As Double

Dim PREV\_TOTAL\_ASSETS\_VAL As Double

Dim PREV\_RE\_VAL As Double

'------------------------------------------------------------------------------------------

'------------------------------------------------------------------------------------------

'Current Fiscal Year INPUTS

'------------------------------------------------------------------------------------------

'------------------------------------------------------------------------------------------

Dim REVENUE\_VAL As Double

Dim EBIT\_VAL As Double

'------------------------------------------------------------------------------------------

Dim ASSETS\_EXCLUDING\_CASH\_VAL As Double

'------------------------------------------------------------------------------------------

'Total Accounts Receivable

'Inventories

'Other Current Assets

'Net Property, Plant & Equipment

'Intangible Assets (Including Goodwill)

'Other Non-Current Assets

'------------------------------------------------------------------------------------------

'------------------------------------------------------------------------------------------

Dim LIABILITIES\_EQUITIES\_EXCLUDING\_LT\_DEBT\_AND\_ADDITIONS\_RE\_VAL\_AND\_CURRENT\_PORT\_DEBT As Double

'------------------------------------------------------------------------------------------

'Equity and Liability

'Accounts Payable

'Other Current Liabilities

'Deferred Taxes

'Capital Lease Obligations

'Minority Interest Liability

'Other Non- Current Liabilities

'Par + APIC - Treasury (& Other Adjustments)

'Preferred Stock

'------------------------------------------------------------------------------------------

Dim LTD\_RATIO\_VAL As Double

Dim EFFECTIVE\_INTEREST\_RATE\_VAL As Double

Dim MONEY\_MARKET\_INTEREST\_RATE\_VAL As Double

Dim EFFECTIVE\_TAX\_RATE\_VAL As Double

Dim DIVIDENDS\_PAYOUT\_RATE\_VAL As Double

Dim AFTER\_TAX\_ADJUSTMENTS\_RATE\_VAL As Double

'------------------------------------------------------------------------------------------

Dim TOTAL\_ASSETS\_VAL As Double

Dim LT\_DEBT\_VAL As Double

Dim INTEREST\_EXPENSE\_VAL As Double

Dim PRE\_TAX\_INCOME\_VAL As Double

Dim EFFECTIVE\_TAX\_VAL As Double

Dim AFTER\_TAX\_ADJUSTMENTS\_VAL As Double

Dim NET\_INCOME\_VAL As Double

Dim DIVIDENDS\_VAL As Double

Dim ADDITIONS\_RE\_VAL As Double

Dim TOTAL\_LIABILITIES\_EQUITIES\_VAL As Double

'------------------------------------------------------------------------------------------

Dim j As Long

Dim SCOLUMN As Long

Dim NCOLUMNS As Long

Dim CASH\_VAL As Double

'------------------------------------------------------------------------------------------

With ActiveSheet

SCOLUMN = 12

NCOLUMNS = 20

For j = 1 To NCOLUMNS

'PREVIOUS FISCAL YEAR

PREV\_CASH\_VAL = .Cells(169, SCOLUMN + j - 1).Value

PREV\_LT\_DEBT\_VAL = .Cells(186, SCOLUMN + j - 1).Value

PREV\_TOTAL\_ASSETS\_VAL = .Cells(179, SCOLUMN + j - 1).Value

PREV\_RE\_VAL = .Cells(193, SCOLUMN + j - 1).Value

'CURRENT\_FISCAL\_YEAR

REVENUE\_VAL = .Cells(147, SCOLUMN + j).Value

EBIT\_VAL = .Cells(154, SCOLUMN + j).Value

ASSETS\_EXCLUDING\_CASH\_VAL = \_

.Cells(170, SCOLUMN + j) + \_

.Cells(171, SCOLUMN + j) + \_

.Cells(172, SCOLUMN + j) + \_

.Cells(176, SCOLUMN + j) + \_

.Cells(177, SCOLUMN + j) + \_

.Cells(178, SCOLUMN + j)

LIABILITIES\_EQUITIES\_EXCLUDING\_LT\_DEBT\_AND\_ADDITIONS\_RE\_VAL\_AND\_CURRENT\_PORT\_DEBT = \_

.Cells(182, SCOLUMN + j) + \_

.Cells(184, SCOLUMN + j) + \_

.Cells(187, SCOLUMN + j) + \_

.Cells(188, SCOLUMN + j) + \_

.Cells(189, SCOLUMN + j) + \_

.Cells(190, SCOLUMN + j) + \_

.Cells(192, SCOLUMN + j) + \_

.Cells(194, SCOLUMN + j)

'---------------------------------------------------------------------------

LTD\_RATIO\_VAL = .Cells(141, SCOLUMN + j).Value

'Remember that this is used to calculate current LT Debt Val

EFFECTIVE\_INTEREST\_RATE\_VAL = .Cells(140, SCOLUMN + j).Value

MONEY\_MARKET\_INTEREST\_RATE\_VAL = .Cells(142, SCOLUMN + j).Value

DIVIDENDS\_PAYOUT\_RATE\_VAL = .Cells(143, SCOLUMN + j).Value

EFFECTIVE\_TAX\_RATE\_VAL = .Cells(139, SCOLUMN + j).Value

AFTER\_TAX\_ADJUSTMENTS\_RATE\_VAL = .Cells(124, SCOLUMN + j).Value

CASH\_VAL = FINAL\_CASH\_SOLVER(PREV\_CASH\_VAL, \_

PREV\_LT\_DEBT\_VAL, \_

PREV\_TOTAL\_ASSETS\_VAL, \_

PREV\_RE\_VAL, \_

REVENUE\_VAL, \_

EBIT\_VAL, \_

ASSETS\_EXCLUDING\_CASH\_VAL, \_

LIABILITIES\_EQUITIES\_EXCLUDING\_LT\_DEBT\_AND\_ADDITIONS\_RE\_VAL\_AND\_CURRENT\_PORT\_DEBT, \_

LTD\_RATIO\_VAL, \_

EFFECTIVE\_INTEREST\_RATE\_VAL, \_

MONEY\_MARKET\_INTEREST\_RATE\_VAL, \_

DIVIDENDS\_PAYOUT\_RATE\_VAL, \_

EFFECTIVE\_TAX\_RATE\_VAL, \_

AFTER\_TAX\_ADJUSTMENTS\_RATE\_VAL, "", 0)

Debug.Print .Cells(164, SCOLUMN + j), Format(Abs(CASH\_VAL - .Cells(169, SCOLUMN + j)), "0.0000")

Next j

End With

End Sub

'Circular Reference Function for solving Cash Balance (DCF): Function FINAL\_CASH\_SOLVER

'The parameter-intakes are segregated into three arrays - asset items, liability and equity

'items, net income items, and miscellaneous items - previous year total asset, effective

'tax rate, effective interest rate, LTD/last year assets, and money market interest rate.

'Description of a Single Calculation Process:

'1. Calculate the value of specified asset - all asset items except cash

'2. Calculate the value of long term debt based on current year and previous year total asset

'3. Calculate the interest expense based on long term debt

'4. Calculate the value of net income and addition to retained earnings, and then the updated

' value of retained earnings

'5. Calculate the value of specified liabilities - all liability and shareholder's equity

' item except short term debt

'6. Calculate the value of net required financing

'7. Finally derive an updated value for cash based on the net required financing.

'Choice of Optimization Methods

'Upon testing, Muller 's method can produce an equilibrium solution with an error term of

'10^-5, which is sufficient.

Function FINAL\_CASH\_SOLVER(ByVal PREV\_CASH\_VAL As Double, \_

ByVal PREV\_LT\_DEBT\_VAL As Double, \_

ByVal PREV\_TOTAL\_ASSETS\_VAL As Double, \_

ByVal PREV\_RE\_VAL As Double, \_

ByVal REVENUE\_VAL As Double, \_

ByVal EBIT\_VAL As Double, \_

ByVal ASSETS\_EXCLUDING\_CASH\_VAL As Double, \_

ByVal LIABILITIES\_EQUITIES\_EXCLUDING\_LT\_DEBT\_AND\_ADDITIONS\_RE\_VAL\_AND\_CURRENT\_PORT\_DEBT As Double, \_

ByVal LTD\_RATIO\_VAL As Double, \_

ByVal EFFECTIVE\_INTEREST\_RATE\_VAL As Double, \_

ByVal MONEY\_MARKET\_INTEREST\_RATE\_VAL As Double, \_

ByVal DIVIDENDS\_PAYOUT\_RATE\_VAL As Double, \_

ByVal EFFECTIVE\_TAX\_RATE\_VAL As Double, \_

ByVal AFTER\_TAX\_ADJUSTMENTS\_RATE\_VAL As Double, \_

Optional ByVal CASH\_VAL As Variant = "", \_

Optional ByVal OUTPUT As Integer = 0)

'------------------------------------------------------------------------------------------

'ASSETS\_EXCLUDING\_CASH\_VAL:

'------------------------------------------------------------------------------------------

'Total Accounts Receivable

'Inventories

'Other Current Assets

'Net Property, Plant & Equipment

'Intangible Assets (Including Goodwill)

'Other Non-Current Assets

'------------------------------------------------------------------------------------------

'------------------------------------------------------------------------------------------

'LIABILITIES\_EQUITIES\_EXCLUDING\_LT\_DEBT\_AND\_ADDITIONS\_RE\_VAL\_AND\_CURRENT\_PORT\_DEBT:

'------------------------------------------------------------------------------------------

'Equity and Liability

'Accounts Payable

'Other Current Liabilities

'Deferred Taxes

'Capital Lease Obligations

'Minority Interest Liability

'Other Non- Current Liabilities

'Par + APIC - Treasury (& Other Adjustments)

'Preferred Stock

'---------------------------------------------------------------------------

Dim X\_VAL As Double ' solving for cash

Dim TOTAL\_ASSETS\_VAL As Double

Dim LT\_DEBT\_VAL As Double

Dim INTEREST\_EXPENSE\_VAL As Double

Dim PRE\_TAX\_INCOME\_VAL As Double

Dim EFFECTIVE\_TAX\_VAL As Double

Dim AFTER\_TAX\_ADJUSTMENTS\_VAL As Double

Dim NET\_INCOME\_VAL As Double

Dim DIVIDENDS\_VAL As Double

Dim ADDITIONS\_RE\_VAL As Double

Dim TOTAL\_LIABILITIES\_EQUITIES\_VAL As Double

Dim CURRENT\_PORT\_DEBT As Double

Dim SPECIFIED\_LIABILITY\_VAL As Double

'Optimizer-----------------------------------------------------------------

Dim CONVERG\_VAL As Integer

Dim COUNTER As Long

Dim nLOOPS As Long: nLOOPS = 1000

Dim tolerance As Double: tolerance = 0.001

Dim LOWER\_VAL As Double: LOWER\_VAL = 1e-06

Dim UPPER\_VAL As Double: UPPER\_VAL = 10000

'---------------------------------------------------------------------------------

'On Error GoTo ERROR\_LABEL

If CASH\_VAL = "" Then 'Solve for Cash

GoSub LOAD\_PUBLIC\_LINE

GoSub SOLVE\_LINE

End If

TOTAL\_ASSETS\_VAL = ASSETS\_EXCLUDING\_CASH\_VAL + CASH\_VAL

LT\_DEBT\_VAL = ((PREV\_TOTAL\_ASSETS\_VAL + TOTAL\_ASSETS\_VAL) / 2) \* LTD\_RATIO\_VAL

INTEREST\_EXPENSE\_VAL = (EFFECTIVE\_INTEREST\_RATE\_VAL \* ((LT\_DEBT\_VAL + PREV\_LT\_DEBT\_VAL) / 2)) - (MONEY\_MARKET\_INTEREST\_RATE\_VAL \* PREV\_CASH\_VAL) 'This function could change depending on how you want to calculate interest payment for each type of debt

PRE\_TAX\_INCOME\_VAL = EBIT\_VAL - INTEREST\_EXPENSE\_VAL

EFFECTIVE\_TAX\_VAL = PRE\_TAX\_INCOME\_VAL \* EFFECTIVE\_TAX\_RATE\_VAL

AFTER\_TAX\_ADJUSTMENTS\_VAL = REVENUE\_VAL \* AFTER\_TAX\_ADJUSTMENTS\_RATE\_VAL

NET\_INCOME\_VAL = PRE\_TAX\_INCOME\_VAL - EFFECTIVE\_TAX\_VAL - AFTER\_TAX\_ADJUSTMENTS\_VAL

DIVIDENDS\_VAL = NET\_INCOME\_VAL \* DIVIDENDS\_PAYOUT\_RATE\_VAL

ADDITIONS\_RE\_VAL = NET\_INCOME\_VAL - DIVIDENDS\_VAL

SPECIFIED\_LIABILITY\_VAL = LIABILITIES\_EQUITIES\_EXCLUDING\_LT\_DEBT\_AND\_ADDITIONS\_RE\_VAL\_AND\_CURRENT\_PORT\_DEBT + \_

PREV\_RE\_VAL + ADDITIONS\_RE\_VAL + LT\_DEBT\_VAL

If ASSETS\_EXCLUDING\_CASH\_VAL - SPECIFIED\_LIABILITY\_VAL < 0 Then

CURRENT\_PORT\_DEBT = 0

Else

CURRENT\_PORT\_DEBT = ASSETS\_EXCLUDING\_CASH\_VAL - SPECIFIED\_LIABILITY\_VAL

End If

TOTAL\_LIABILITIES\_EQUITIES\_VAL = SPECIFIED\_LIABILITY\_VAL + CURRENT\_PORT\_DEBT

Select Case OUTPUT

Case 0

FINAL\_CASH\_SOLVER = CASH\_VAL

Case 1 'Objective Function

FINAL\_CASH\_SOLVER = Abs(TOTAL\_ASSETS\_VAL - TOTAL\_LIABILITIES\_EQUITIES\_VAL)

End Select

Exit Function

'--------------------------------------------------------------------

LOAD\_PUBLIC\_LINE:

'--------------------------------------------------------------------

PUB\_ARR(1) = PREV\_CASH\_VAL

PUB\_ARR(2) = PREV\_LT\_DEBT\_VAL

PUB\_ARR(3) = PREV\_TOTAL\_ASSETS\_VAL

PUB\_ARR(4) = PREV\_RE\_VAL

PUB\_ARR(5) = REVENUE\_VAL

PUB\_ARR(6) = EBIT\_VAL

PUB\_ARR(7) = ASSETS\_EXCLUDING\_CASH\_VAL

PUB\_ARR(8) = LIABILITIES\_EQUITIES\_EXCLUDING\_LT\_DEBT\_AND\_ADDITIONS\_RE\_VAL\_AND\_CURRENT\_PORT\_DEBT

PUB\_ARR(9) = LTD\_RATIO\_VAL

PUB\_ARR(10) = EFFECTIVE\_INTEREST\_RATE\_VAL

PUB\_ARR(11) = MONEY\_MARKET\_INTEREST\_RATE\_VAL

PUB\_ARR(12) = DIVIDENDS\_PAYOUT\_RATE\_VAL

PUB\_ARR(13) = EFFECTIVE\_TAX\_RATE\_VAL

PUB\_ARR(14) = AFTER\_TAX\_ADJUSTMENTS\_RATE\_VAL

Return

'--------------------------------------------------------------------

SOLVE\_LINE:

'--------------------------------------------------------------------

X\_VAL = MULLER\_ZERO\_FUNC(LOWER\_VAL, UPPER\_VAL, \_

"SOLVE\_CASH\_OBJECTIVE\_FUNCTION", CONVERG\_VAL, COUNTER, nLOOPS, tolerance)

If CONVERG\_VAL <> 0 Then

CASH\_VAL = PUB\_TOL\_VAL

Else

CASH\_VAL = X\_VAL

End If

'--------------------------------------------------------------------

Return

'--------------------------------------------------------------------

ERROR\_LABEL:

FINAL\_CASH\_SOLVER = PUB\_TOL\_VAL

End Function

Function SOLVE\_CASH\_OBJECTIVE\_FUNCTION(ByVal X\_VAL As Double)

SOLVE\_CASH\_OBJECTIVE\_FUNCTION = FINAL\_CASH\_SOLVER(PUB\_ARR(1), \_

PUB\_ARR(2), \_

PUB\_ARR(3), \_

PUB\_ARR(4), \_

PUB\_ARR(5), \_

PUB\_ARR(6), \_

PUB\_ARR(7), \_

PUB\_ARR(8), \_

PUB\_ARR(9), \_

PUB\_ARR(10), \_

PUB\_ARR(11), \_

PUB\_ARR(12), \_

PUB\_ARR(13), \_

PUB\_ARR(14), \_

X\_VAL, 1)

End Function

Sub TEST\_FINAL\_CASH\_SOLVER()

'Previous Fiscal Year INPUTS!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Dim PREV\_CASH\_VAL As Double

Dim PREV\_LT\_DEBT\_VAL As Double

Dim PREV\_TOTAL\_ASSETS\_VAL As Double

Dim PREV\_RE\_VAL As Double

'Current Fiscal Year INPUTS!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Dim REVENUE\_VAL As Double

Dim EBIT\_VAL As Double

Dim ASSETS\_EXCLUDING\_CASH\_VAL As Double

Dim LIABILITIES\_EQUITIES\_EXCLUDING\_LT\_DEBT\_AND\_ADDITIONS\_RE\_VAL\_AND\_CURRENT\_PORT\_DEBT As Double

Dim LTD\_RATIO\_VAL As Double

Dim EFFECTIVE\_INTEREST\_RATE\_VAL As Double

Dim MONEY\_MARKET\_INTEREST\_RATE\_VAL As Double

Dim EFFECTIVE\_TAX\_RATE\_VAL As Double

Dim DIVIDENDS\_PAYOUT\_RATE\_VAL As Double

Dim AFTER\_TAX\_ADJUSTMENTS\_RATE\_VAL As Double

'Current Fiscal Year Calculations!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Dim X\_VAL As Double ' solving for cash

Dim TOTAL\_ASSETS\_VAL As Double

Dim LT\_DEBT\_VAL As Double

Dim INTEREST\_EXPENSE\_VAL As Double

Dim PRE\_TAX\_INCOME\_VAL As Double

Dim EFFECTIVE\_TAX\_VAL As Double

Dim AFTER\_TAX\_ADJUSTMENTS\_VAL As Double

Dim NET\_INCOME\_VAL As Double

Dim DIVIDENDS\_VAL As Double

Dim ADDITIONS\_RE\_VAL As Double

Dim TOTAL\_LIABILITIES\_EQUITIES\_VAL As Double

'PREVIOUS FISCAL YEAR

PREV\_CASH\_VAL = 12777

PREV\_LT\_DEBT\_VAL = 5242

PREV\_TOTAL\_ASSETS\_VAL = 47540

PREV\_RE\_VAL = 30330

'CURRENT\_FISCAL\_YEAR

REVENUE\_VAL = 56370.6

EBIT\_VAL = 1474.605

ASSETS\_EXCLUDING\_CASH\_VAL = 34415.37

LIABILITIES\_EQUITIES\_EXCLUDING\_LT\_DEBT\_AND\_ADDITIONS\_RE\_VAL\_AND\_CURRENT\_PORT\_DEBT = 7847.25

LTD\_RATIO\_VAL = 0.110265039966344 'Remember that this is used to calculate current LT Debt Val

EFFECTIVE\_INTEREST\_RATE\_VAL = 0.0262853832951543

MONEY\_MARKET\_INTEREST\_RATE\_VAL = 0.005

DIVIDENDS\_PAYOUT\_RATE\_VAL = 0

EFFECTIVE\_TAX\_RATE\_VAL = 0.165082717353045

AFTER\_TAX\_ADJUSTMENTS\_RATE\_VAL = 0

Debug.Print FINAL\_CASH\_SOLVER(PREV\_CASH\_VAL, \_

PREV\_LT\_DEBT\_VAL, \_

PREV\_TOTAL\_ASSETS\_VAL, \_

PREV\_RE\_VAL, \_

REVENUE\_VAL, \_

EBIT\_VAL, \_

ASSETS\_EXCLUDING\_CASH\_VAL, \_

LIABILITIES\_EQUITIES\_EXCLUDING\_LT\_DEBT\_AND\_ADDITIONS\_RE\_VAL\_AND\_CURRENT\_PORT\_DEBT, \_

LTD\_RATIO\_VAL, \_

EFFECTIVE\_INTEREST\_RATE\_VAL, \_

MONEY\_MARKET\_INTEREST\_RATE\_VAL, \_

DIVIDENDS\_PAYOUT\_RATE\_VAL, \_

EFFECTIVE\_TAX\_RATE\_VAL, \_

AFTER\_TAX\_ADJUSTMENTS\_RATE\_VAL, "", 0) '= 10003.1433112492

End Sub