Quiz 4 solutions and explanations

IMPORTANT: Even if you do not intend/need to look through the solutions to Quiz 4, you still need to mark this quiz as completed using the blue "Mark as Completed" button in the lower right of this page. By doing so, you will unlock Assignment 4!

This document is meant to provide clear explanations for the Quiz 4 questions (not the in-video quizzes since they have explanations already). I do NOT provide feedback during the quiz (like I do for the screencasts) because a learner could just guess, obtain the correct answers, then put them back into the quiz and get 100%!

This document is purely for you to learn more and to correct your misconceptions about the material. If you view this document soon after you take the quiz to see why you missed a certain question, it will serve as a great learning tool!

PLEASE DO NOT SHARE THIS DOCUMENT WITH ANYONE! Using this document to complete Quiz 4 is a violation of Coursera's Honor Code (a.k.a. cheating).

Question 1:

What will be displayed in the message box when this sub is run?

```
Option Explicit

Sub WhatDoesThisDo()
Dim i As Integer, j As Integer
Dim k As Integer
For i = 1 To 3
    For j = 1 To 3
        k = i + j
    Next j
    If k = 5 Then Exit For
Next i
MsgBox j / i
End Sub
```

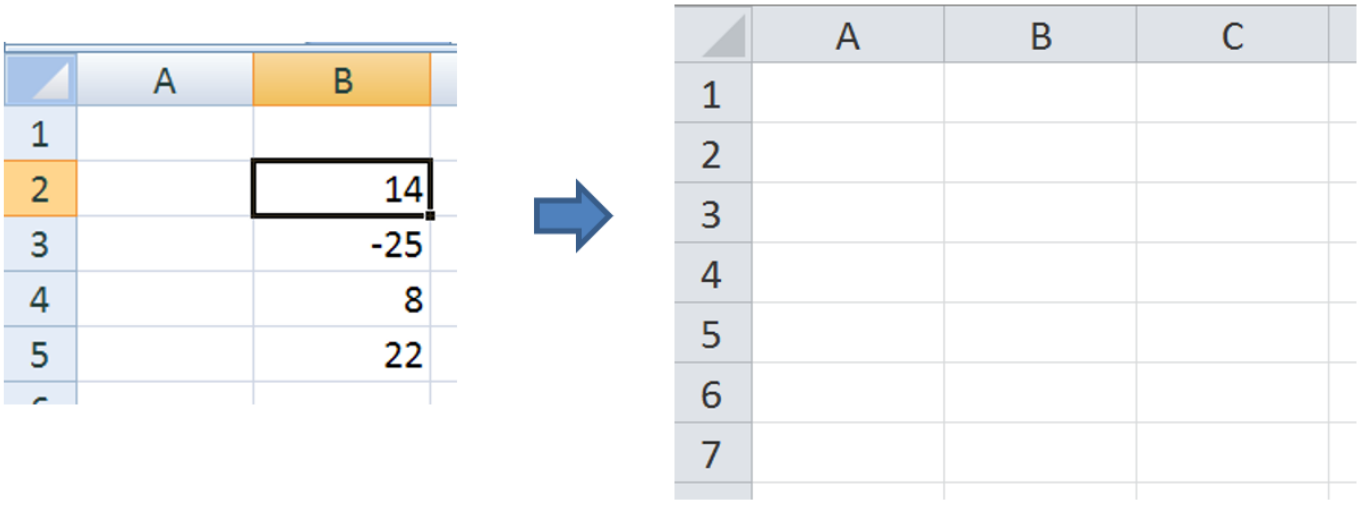
Answer: For questions like this, I like to set up a table of the various variables in the subroutine.

Iteration/step	i	j	k
1	1	1	2
2	1	2	3
3	1	3	4
4	2	1	3
5	2	2	4
6	2	3	5 → Exit For, but at this point j = 4 since indices exit at 1 greater than their last value; at this point, j is 4 and i is 2 so j/i = 2.

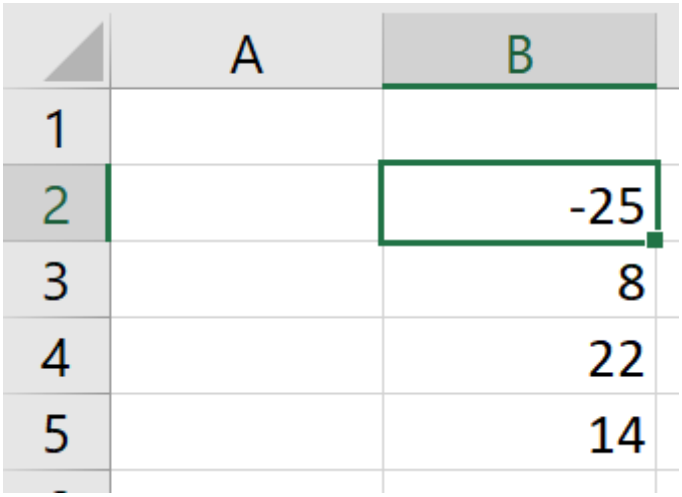
Question 2:

How will the appearance of the spreadsheet below change when the MassiveConfusion sub is run?

```
Option Explicit
Sub MassiveConfusion()
Dim WhatsIt As Double
Dim i As Integer
WhatsIt = ActiveCell.Value
For i = 1 To 3
    ActiveCell.Offset(i - 1, 0) = ActiveCell.Offset(i, 0)
Next i
ActiveCell.Offset(3, 0) = WhatsIt
End Sub
```

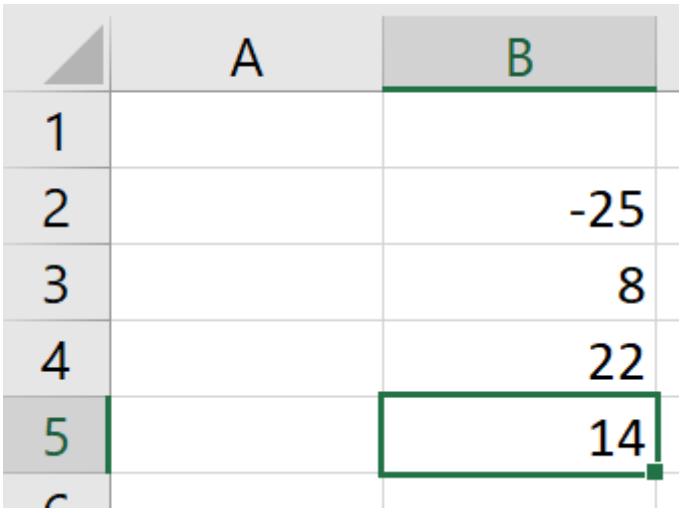


A.



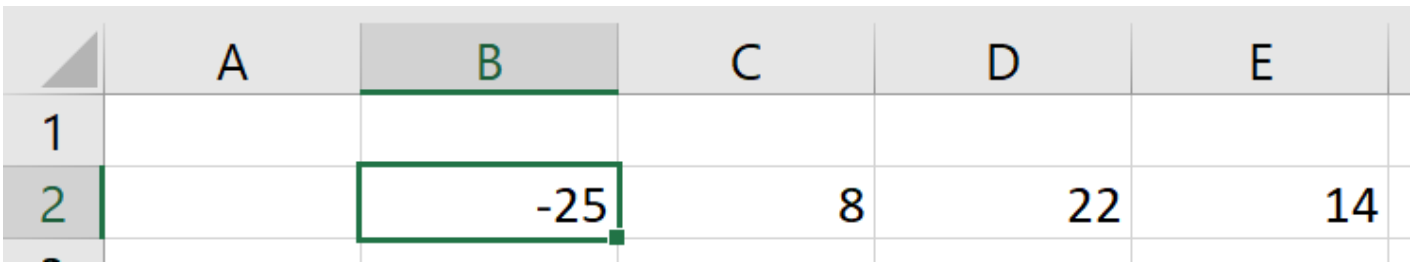
Correct! **WhatsIt** will be set to 14 (active cell value) in the first step. Next, in the For…Next loop we go one at a time for i ranging from 1 to 3. For i = 1: ActiveCell.Offset(0,0) = ActiveCell.Offset(1,0). This places the value in B3 (-25) into cell B2. For i = 2: ActiveCell.Offset(1,0) = ActiveCell.Offset(2,0). This places the value in B4 (8) into cell B3. For i =3: ActiveCell.Offset(2,0) = ActiveCell.Offset(3,0). This places the value in B5 (22) into cell B4. Outside the For…Next loop, we take WhatsIt (equal to 14) and offset it 3 rows and 0 columns from the active cell (B2) into cell B5.

B.



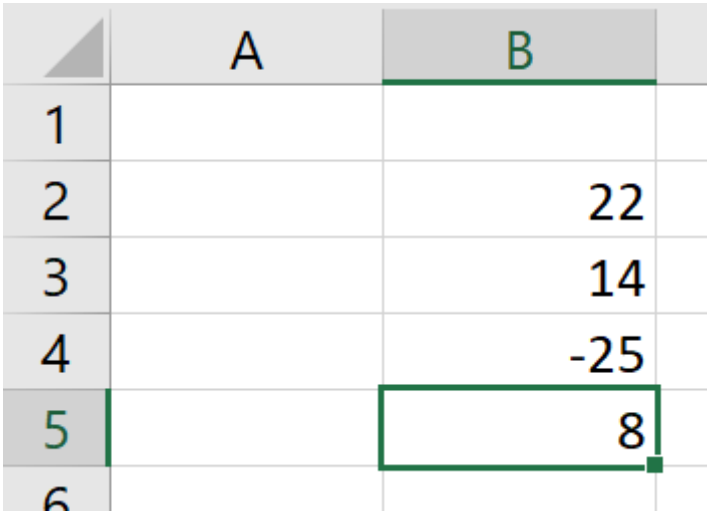
Incorrect. See the explanation for option A above.

C.



Incorrect. See the explanation for option A above.

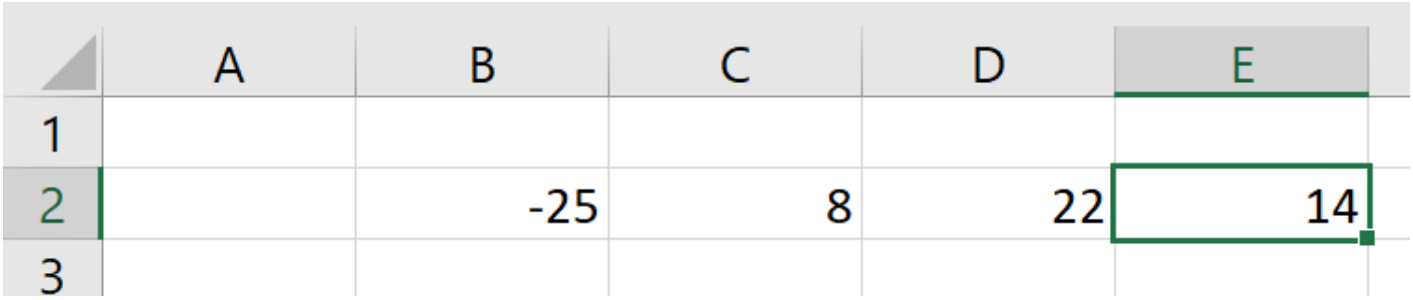
D.



	A	B
1		
2		22
3		14
4		-25
5		8
6		

Incorrect. See the explanation for option A above.

E.



	A	B	C	D	E
1					
2		-25	8	22	14
3					

Incorrect. See the explanation for option A above.

Question 3:

We wish to create a VBA subroutine that will ask the user for an initial number N then the subroutine will tell the user how many doublings (nd) are required to exceed 1,000,000. Which of the following subroutines would work for this scenario? Select all that apply. (HINT: 2 are correct!)

A.

```
Option Explicit

Sub Invest()
Dim N As Double, nd As Integer
N = InputBox("Enter initial number.")
Do
    If N > 1000000 Then Exit Do
    N = 2 * N
    nd = nd + 1
Loop
MsgBox nd & " doublings are required to exceed 1,000,000!"
End Sub
```

Correct! If N comes in greater than 1,000,000 then the loop will be exited, which is important. Otherwise, N is doubled and nd is counted for each iteration. Once N exceeds 1,000,000 then the loop is exited and the number of doublings are displayed.

B.

```
Option Explicit

Sub Invest()
Dim N As Double, nd As Integer
N = InputBox("Enter initial number.")
Do While N <= 1000000
    N = 2 * N
    nd = nd + 1
Loop
MsgBox nd & " doublings are required to exceed 1,000,000!"
End Sub
```

Correct! While N is less than or equal to 1,000,000 the loop is entered. If N comes in greater than 1,000,000 then the loop will never be entered, which is important. Otherwise, N is doubled and nd is counted for each iteration. Once N exceeds 1,000,000 then the loop is not entered and the number of doublings are displayed.

C.

```
Option Explicit

Sub Invest()
Dim N As Double, nd As Integer
N = InputBox("Enter initial number.")
Do While N > 1000000
    N = 2 * N
    nd = nd + 1
Loop
MsgBox nd & " doublings are required to exceed 1,000,000!"
End Sub
```

Incorrect. The conditional statement on the “Do While … “ part is incorrect. The code makes it such that while N is greater than 1,000,000 we double N, which is not what we want to do. If N enters at 1, for example, then we won’t double it at all!

D.

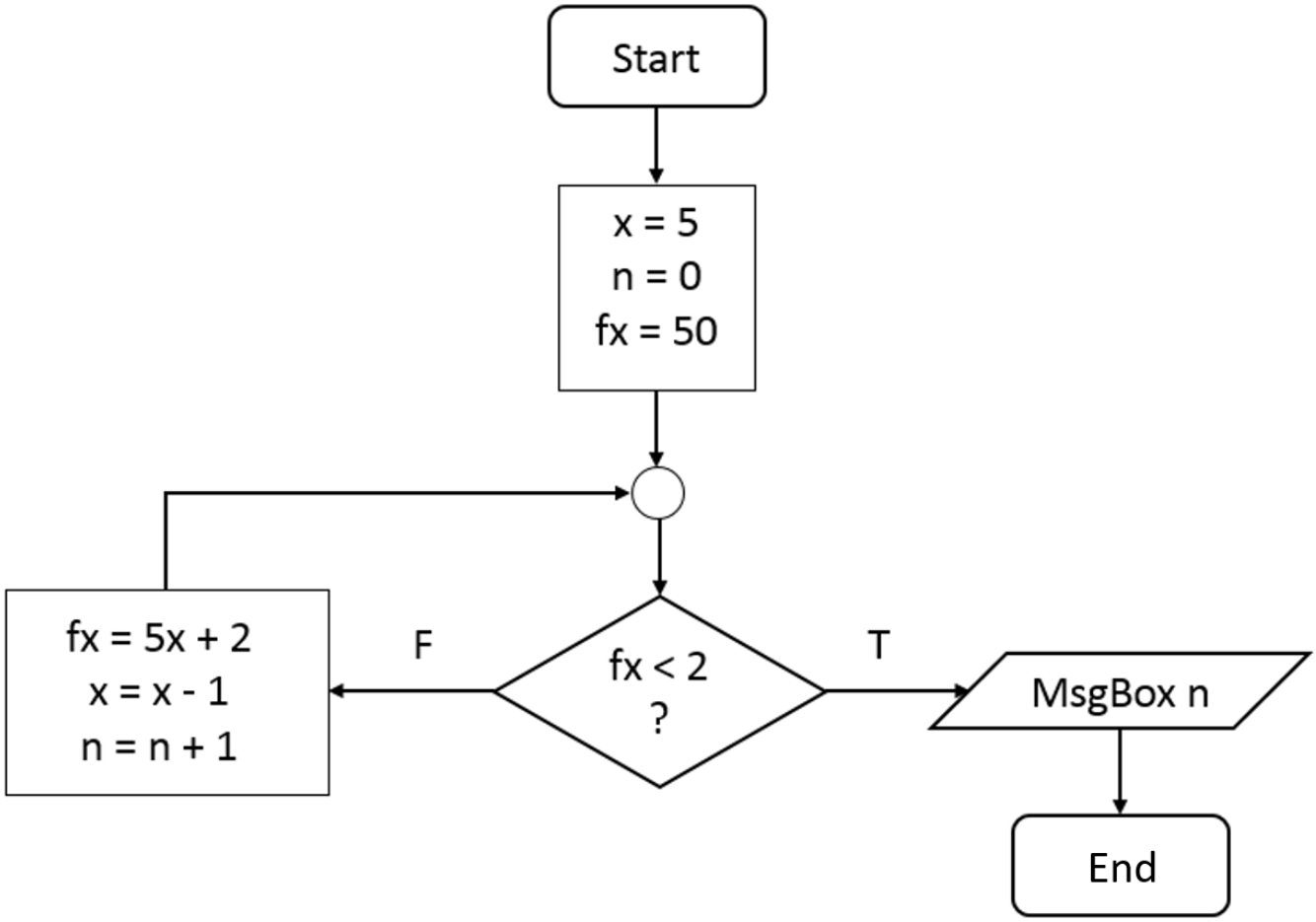
```
Option Explicit

Sub Invest()
Dim N As Double, nd As Integer
N = InputBox("Enter initial number.")
Do
    N = 2 * N
    If N > 1000000 Then Exit Do
    nd = nd + 1
Loop
MsgBox nd & " doublings are required to exceed 1,000,000!"
End Sub
```

Incorrect. This is mostly correct, but it undercounts the number of doublings by 1. The “nd = nd + 1” line is in the wrong spot – for example, if N enters at a value of 600,000 then N will double to 1,200,000 then exit the loop, but nd will not be incremented to 1 (it will show 0 doublings in the message box). In general, the number of doublings required will always be 1 fewer than the actual number of doublings required.

Question 4:

What value will be displayed in the message box when the subroutine shown is executed?



Answer: After x is set to 5, n is set to 0, and fx is set to 50, the Do…Loop is entered. In the first iteration, since fx is not less than 2 (50 > 2) the loop is resumed and the following are calculated: fx is set to 27, x is set to 4, and n is set to 1. In the next iteration of the Do…Loop, since fx (27) is not less than 2 the loop is resumed once again: fx is set to 22, x is set to 3, and n is set to 2. The loop goes again since fx (22) is not less than 2: fx is set to 17, x is set to 2, and n is set to 3. Ditto, fx (17) is not less than 2 so the loop resumes: fx is set to 12, x is set to 1, and n is set to 4. Ditto, fx (12) is not less than 2 so the loop resumes: fx is set to 7, x is set to 0, and n is set to 5. Ditto! fx (7) is not less than 2 so the loop resumes: fx is set to 2, x is set to -1, and n is set to 6. One more time since fx (2) is not less than 2: fx is set to -3, x is set to -2, and n is set to 7. Finally! fx = -3 *is* less than 2, so we exit the Do…Loop and display n (equal to 7) in the message box. So, the answer is 7.

Question 5:

What does the subroutine below do?

```
Option Explicit
```

```
Sub WhatWillHappen()
Dim b As Integer, i As Integer
Dim nr As Integer
nr = Selection.Rows.Count
For i = 1 To nr
    If Selection.Cells(i, 1) > 7 Or Selection.Cells(i, 1) < -7 Then
        b = b + 1
    End If
Next i
MsgBox b
End Sub
```

A. This sub counts the number of cells in a column selection that are greater than 7 or less than -7.

Correct! First, the number of rows of a Selection (this is just a column) are counted. Next, a For…Next loop iterates through all rows (i goes from 1 to nr). If an individual item of the Selection [Selection.Cells(i,1)] is greater than 7 or less than -7, b is incremented by 1. So, b represents the number of items that are greater than 7 or less than -7. At the end of the subroutine, b is displayed in a message box.

B. This sub counts the number of cells in a column selection that are between -7 and 7.

Incorrect. See explanation above for Part A.

C. This sub sums all cells in a column selection that are greater than 7 or less than -7.

Incorrect. See explanation above for Part A.

D. This sub sums all cells in a column selection that are between -7 and 7.

Incorrect. See explanation above for Part A.