Debugging - Trace Tables

1. Kathryn creates an algorithm that will search the callTotal array, and returns the largest value present.

```
1. SET callTotal TO [21, 165, 102, 30, 177, 33]
2. SET maximum TO callTotal[0]
3. FOR counter FROM 1 TO 5 DO
4.     IF callTotal[counter] < maximum THEN
5.         SET maximum TO callTotal[counter]
6.     END IF
7. END FOR
```

a) A trace table is used to record the change to a variable at the corresponding line number. Complete the Trave Table below.

Line	**callTotal**	**maximum**	**counter**
1	21, 165, 102, 30, 177, 33		
2			
3			
5			
3	...	...	...

b) The algorithm is incorrect

i) Explain why the algorithm is incorrect

ii) Describe how to correct the algorithm

2. A program has been written to find the location of a requested value in a list, however the program does not return the correct location. The algorithm responsible is shown below.

```
1. SET dataset TO [72,34,19,73,52,28]
2. SET location TO -1
3. SET target TO 72
4. FOR counter = 0 TO len(dataset) DO
5.     IF dataset[counter] = target THEN
6.         SET counter TO location
7.     END IF
8. END FOR
```

a) A trace table is being used to record the changes to the variable when stepping through the code.

Complete the information in the table below, recording the value assigned to the variable for lines 2, 3, 4 and 6. (Line 5 does not change a variable's value and so it is not included)

line	**dataset**	**find**	**location**	**counter**
1	72, 34, 19, 73, 52, 28			
2				
3				
4				
6				

b) Explain why the location is never correctly given.

c) Explain how the algorithm should be correct

3. Michael uses the following algorithm to count the occurrences of a number in an array.
The number she is counting is 12.

```
1.   SET list to [12,65,32,17,65,32,17,17,12,89,20,10]
2.   SEND “Enter value to find: TO DISPLAY
3.   RECEIVE target FROM KEYBOARD
4.   FOR element FROM 0 to 11 DO
5.       SET counter TO 0
6.       IF list[element] = target THEN
7.           SET counter TO counter + 1
8.       END IF
9.   END FOR
10.  SEND "Total number found is", counter TO DISPLAY
```

a) A trace table is shown below which shows the line numbers where a variable has changed.
State the missing values at A, B, C & D.

line	**list**	**counter**	**element**	**target**
1	12, 65, 32, 17, 65, 32, 17, 17, 12, 78, 20, 10			
3				**A** -
4			0	
5		**B** -		
7		**C** -		
4				
5		**D** -		

b) The algorithm is incorrect and the correct number of occurrences is not displayed.

i) Explain why the algorithm is incorrect.

ii) Describe how to correct the algorithm

4. Patrick has created the following algorithm to search an array for a specific value.

```
1.    SET list TO ["A","D","M","S","A","U"]
2.    SET target TO "A"
3.    SET found TO false
4.    FOR counter FROM 0 to len(list) DO
5.        IF list[counter]=target THEN
6.            SET found to true
7.            SET foundlocation TO counter
8.        END IF
9.    END FOR
10.   IF found = true THEN
11.       SEND "Item found at " + foundlocation TO DISPLAY
12.   ELSE
13.       SEND "Not found" TO DISPLAY
14.   END IF
```

a) A trace table is shown below which shows line numbers where a variable has changed. State the missing value at A, B, C, D

line	list	target	counter	found	foundLocation
1	A, D, M, S, A, U				
2		**A** -			
3				**B** -	
4			0		
6				**C** -	
7					**D** -

b) The algorithm is incorrect, as it only reports the location of the last found item in the array.

i) Explain the error in the logic of the algorithm.

ii) Re-write the algorithm to correct this error so that the location of each occurrence of the target is reported.