Trace table 1

Challenge 1

C C

Read through the program and then complete the trace table. The `for` loop uses a step value of `-2`.

Pseudocode

```
FOR val = 6 TO 0 STEP -2
    PRINT(val+2)
NEXT val
```

Drag and drop the given values in the correct spaces to complete the trace table. Each value can be used more than once.

val	Output:
6	

Items:

6 8 0 2 4

Trace table 3

Challenge 2

Read through the following pseudocode program:

Pseudocode

```
word = "computer"

FOR index = 0 TO LEN(word)-1 STEP 2
    PRINT(word[index])
NEXT index
```

Fill in the trace table by dragging and dropping the correct values from the options given.

index	OUTPUT
0	
	c

Items:

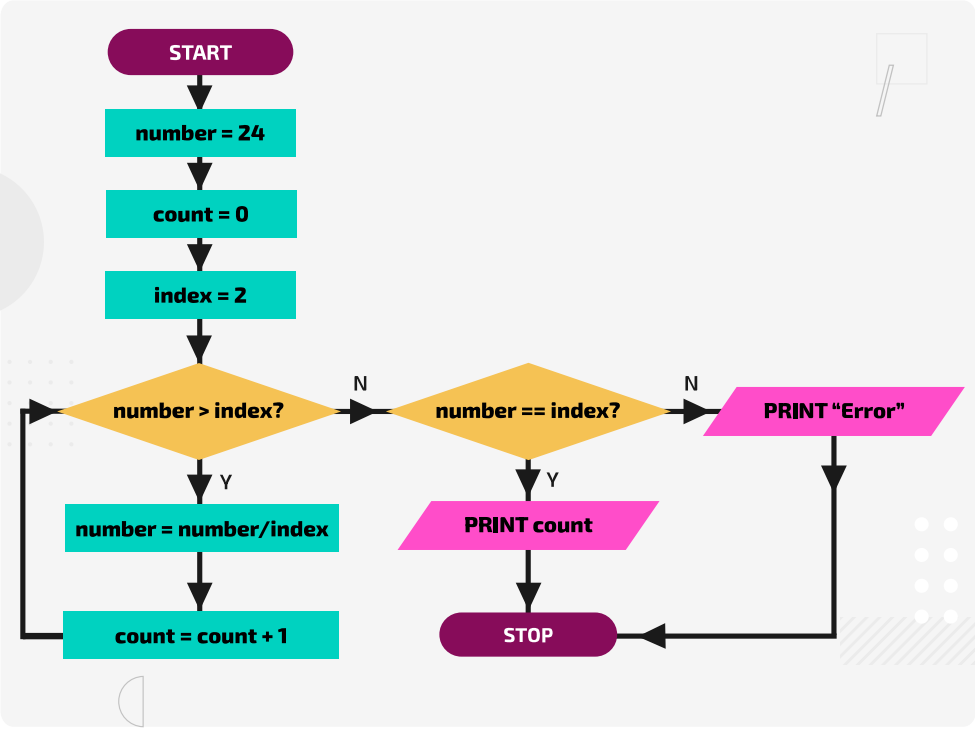
1 2 3 4 5 6 7 8 o m p u t e r

Quiz:
STEM SMART Computer Science Week 36

Trace a flow chart 2

Challenge 1

Study the flowchart below that represents an algorithm:



Trace the algorithm and fill in the trace table by dragging and dropping the correct values from the options given.

number	count	OUTPUT
	0	
12		
	3	
1.5		

	4	

Items:

2 1 24 4 6 Error 3

Quiz:
STEM SMART Computer Science Week 36

Iteration: trace code 1

Challenge 1

C C

A student has designed an algorithm that is shown in pseudocode below. The algorithm uses a condition-controlled loop.

Pseudocode

```
PROCEDURE main()
    input_value = INPUT("Enter a number: ")
    num = INT(input_value)
    count = 0
    WHILE num < 500
        num = num * 2
        count = count + 1
    ENDWHILE
    PRINT(count - 1)
ENDPROCEDURE
```

The student traces the algorithm for a user input of 16.

Which of the following options shows a trace table with the correct values?

○

Trace table			
num	**count**	**num < 500**	**OUTPUT**
16	0	TRUE	
32	1	TRUE	
64	2	TRUE	
128	3	TRUE	
256	4	TRUE	
		FALSE	3

○

Trace table			
num	**count**	**num < 500**	**OUTPUT**
16	0	TRUE	
32	1	TRUE	
64	2	TRUE	

128	3	TRUE	
256	4	TRUE	
512	5	FALSE	
			4

○

Trace table			
num	**count**	**num < 500**	**OUTPUT**
16	0	TRUE	
32	1	TRUE	
64	2	TRUE	
128	3	TRUE	
256	4	TRUE	
512	5	FALSE	
			511

○

Trace table			
num	**count**	**num < 500**	**OUTPUT**
16	1	TRUE	
32	2	TRUE	
64	3	TRUE	
128	4	TRUE	
256	5	TRUE	
512	6	FALSE	
			5

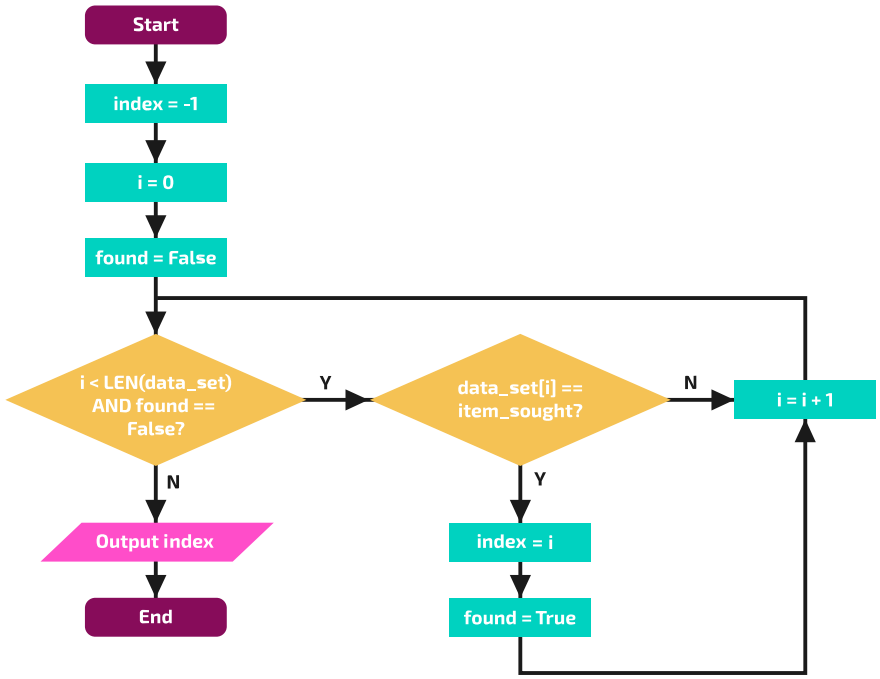
Quiz:
STEM SMART Computer Science Week 36

Flow chart to pseudocode

Challenge 1

C C

A student has specified an algorithm using a flowchart. She wants to specify the same algorithm using pseudocode. Drag and drop the pseudocode statements into the same order as that expressed by the flowchart. Make sure that you use appropriate indentation.



Flowchart

Available items

```
i = i + 1
```

```
i = 0
```

```
found = True
```

```
index = i
```

```
PRINT (index)
```

```
ENDWHILE
```

```
WHILE i < LEN(data_set) AND found == False
```

```
index = -1
```

```
found = False
```

```
ENDIF
```

```
IF data_set[i] == item_sought THEN
```

Quiz:
STEM SMART Computer Science Week 36

Trace table 4

Challenge 1

Read through the pseudocode program. A trace table has been partially filled in below showing the state of the variables and condition, alongside the output value.

Type the correct value into each of the missing spaces in the trace table.

Pseudocode

```
1  PROCEDURE myprocedure(num)
2      value = 0
3      WHILE num > -1
4          value = value + 1
5          num = num - value
6      END WHILE
7      PRINT num
8  ENDPROCEDURE
9
10 myprocedure(3)
```

num	value	num greater than -1?	Output
3			
	0		
		True	

Recursion: trace the code 3

Practice 2

A recursive subroutine has been written as follows:

Pseudocode

```
1 FUNCTION do_something(n)
2     IF n == 1 THEN
3         RETURN 0
4     ELSE
5         RETURN 1 + do_something(n DIV 2)
6     ENDIF
7 ENDFUNCTION
```

`DIV` performs an integer division. For example, `42 DIV 10` will return 4 since this is the whole number of times that 10 goes into 42.

When the subroutine is run with the value 18 specified as the argument, i.e. `do_something(18)`, the subroutine returns the value 4. This is because the return statement on line 5 is `1 +` the return value of the recursive call.

What value is returned when the subroutine is run with the value 100 specified as an argument, i.e. `do_something(100)`?

Quiz:
STEM SMART Computer Science Week 36

Recursion: trace the code 5

Challenge 1

`scores` is an array that contains the test scores of each student from a recent class test. Each element of the array contains a record containing the student's surname, first name, and score. These values are accessed using the syntax:

```
student.surname
student.forename
student.score
```

The array is ordered by surname and contains records for these students:

Ahmed, Atkinson, Bashir, Bell, Chesworth, Cooper, Endover, Faujdar, James, Khan, Mohammad, Murray, Singh, Williams, Young

The teacher wants to know what score was achieved in the test by the student with the surname James. The search is carried out using the recursive subroutine defined below.

Pseudocode

```
PROCEDURE search(surname, scores, first, last)
    IF first > last THEN
        PRINT("Not found")
    ELSE
        mid = (first + last) DIV 2
        IF scores[mid].surname == surname THEN
            PRINT(scores[mid].score)
        ELSEIF surname > scores[mid].surname THEN
            first = mid + 1
            search(surname, scores, first, last)
        ELSE
            last = mid - 1
            search(surname, scores, first, last)
        ENDIF
    ENDIF
ENDPROCEDURE
```

`DIV` performs an integer division. For example, `42 DIV 10` will return 4 since this is the whole number of times that 10 goes into 42.

When the subroutine is called for the first time, the arguments specified are 0 for `first` and 14 for `last`, i.e.

```
search("James", scores, 0, 14)
```

How many times in total will the subroutine be called (including the first call) so that the score for the student with the surname James can be retrieved?

Recursion: trace the code 2

Challenge 2

A recursive subroutine has been written as follows:

Pseudocode

```
FUNCTION do_something(x, y)
    IF x == 1 THEN
        RETURN y
    ELSE IF y == 1 THEN
        RETURN x
    ELSE
        RETURN do_something(x-1, y-2)
    ENDIF
ENDFUNCTION
```

Trace the subroutine to determine what the final return value will be when the following call is made:

```
do_something(4, 8)
```

Quiz:
STEM SMART Computer Science Week 36

Recursion: trace table

Challenge 2

A recursive subroutine has been written below. A trace table has been partially filled in, showing the state of the variables and condition, alongside the number of calls to the `make_num` subroutine and the return value.

Write the correct value in the correct location in the trace table:

Pseudocode

```
1  FUNCTION make_num(x, y)
2      IF x == 0 THEN
3          RETURN y
4      ELSE
5          RETURN make_num(x-2, y-1)
6      ENDIF
7  ENDFUNCTION
8
9  make_num(6, 7)
```

Call #	x	y	x equals 0?	Return
1	6	7	False	make_num(4, 6)
2				
3				
4				