**findLast**

1- The program doesn’t check the x[0] element because it uses i > 0. It should use I >= 0 instead.

2- x = [3, 2, 5]; y = 2; Expected = 1 does not execute the fault because the program starts from the 22th index and then executes for the 1th index and the returns the method.

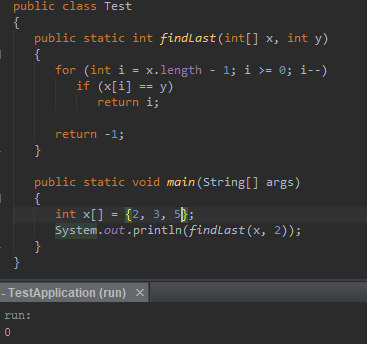
3- x = [1, 5, 3]; y = 2; Expected = -1 would execute the fault but not result in an error state because we do not run the program for the 0th index but we don’t need to anyway since it is not 2.

4- I don’t think this is possible. In order for an incorrect state to occur, we need to have an internal or external failure. We either need to have bad arguments or have wrong code in the method.

But in the textbook it defines failure as external. Does this mean failure only consists of parameter related ones and not code mistakes in the method?

5- We have two parameters, an array and an int. We start at the end of the array and start looking for the second parameter one by one. But we do not test against the 0th element. We only test against 2nd and 1st elements and then leave the for loop because of i > 0.

6-



**Countpositive**

1- The description says it counts positive numbers but it also counts zeros because of the x[i] >= 0. We need to use x[i] > 0 instead.

2- In order to never execute the faulty code, we need to never enter the foor loop. The only way to do that is making x.length 0. So if we pass an empty array we will not execute the faulty code.

3- If we use the argument [-4, 2, 1, 2], we will execute the fault but not end up with an error state because we don’t have an zeros.

4- Same answer as other program.

5- The parameter array includes 1 negative 1 zero 2 positive numbers. The faulty code x[i] >= 0 counts zero as positive so the method ends up returning 3 as in there are 3 positive numbers in this array even though there are only 2.

6-

