

HW#1: Faults, Errors, and Failures

Below are two faulty programs. Each includes a test case that results in failure. Answer the following questions about each program.

```
/**
 * Find last index of element
 *
 * @param x array to search
 * @param y value to look for
 * @return last index of y in x; -1 if absent
 * @throws NullPointerException if x is null
 */
public int findLast (int[] x, int y)
{
    for (int i=x.length-1; i > 0; i--)
    {
        if (x[i] == y)
        {
            return i;
        }
    }
    return -1;
}
// test: x = [2, 3, 5]; y = 2; Expected = 0
// Book website: FindLast.java
// Book website: FindLastTest.java
```

```
/**
 * Count positive elements
 *
 * @param x array to search
 * @return count of positive elements in x
 * @throws NullPointerException if x is null
 */
public int countPositive (int[] x)
{
    int count = 0;
    for (int i=0; i < x.length; i++)
    {
        if (x[i] >= 0)
        {
            count++;
        }
    }
    return count;
}
// test: x = [-4, 2, 0, 2]; Expcted = 2
// Book website: CountPositive.java
// Book website: CountPositiveTest.java
```

- Identify the fault.
- If possible, identify a test case that does **not** execute the fault.
- If possible, identify a test case that executes the fault, but does **not** result in an error state.
- If possible identify a test case that results in an error, but **not** a failure. Hint: Don't forget about the program counter.
- For the given test case, identify the first error state. Be sure to describe the complete state.
- Fix the fault and verify that the given test now produces the expected output.

Submission Guidelines

- Assignment is individual. No group work is accepted.
 - Only PDF format is accepted.**
 -
-