Homework 1

Below are four faulty programs. Each includes test inputs that result 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
```

```
/**
* Find last index of zero
*
* @param x array to search
*
* @return last index of 0 in x; -1 if absent
* @throws NullPointerException if x is null
*/
public static int lastZero (int[] x)
{
   for (int i = 0; i < x.length; i++)
   {
        if (x[i] == 0)
        {
             return i;
        }
    }
    return -1;
}
// test: x = [0, 1, 0]; Expected = 2
// Book website: LastZero.java
// Book website: LastZeroTest.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
```

```
/**
* Count odd or postive elements
*
* @param x array to search
* @return count of odd/positive values in x
* @throws NullPointerException if x is null
*/
public static int oddOrPos(int[] x)
{
    int count = 0;
    for (int i = 0; i < x.length; i++)
    {
        if (x[i]%2 == 1 || x[i] > 0)
        {
            count++;
        }
     }
     return count;
}
// test: x = [-3, -2, 0, 1, 4]; Expected = 3
// Book website: OddOrPos.java
// Book website: OddOrPosTest.java
```

(a) Explain what is wrong with the given code. Describe the fault precisely by proposing a modification to the code.

(b) If possible, give a test case that does not execute the fault. If not, briefly explain why not.

(c) If possible, give a test case that executes the fault, but does not result in an error state. If not, briefly explain why not.

(d) If possible, give a test case that results in an error state, but not a failure. Hint: Don't forget about the program counter. If not, briefly explain why not.

(e) For the given test case, describe the first error state. Be sure to describe the complete

state.

(f) Implement your repair and verify that the given test now produces the expected output. Submit a screen printout or other evidence that your new program works.

310551019 王璽禎

Answer

- **Find last index of element**
  - A. Fault: `for (int i=x.length-1; i > 0; i--)`
    for-loop 應包含陣列起始位置 0，應更正為 `for (int i=x.length-1; i >= 0; i--)`。
  - B. Test: x = [], y = 2
    讓 x 為空，null 會導致 NullPointerException，因此不會運行錯誤。
  - C. Test: x = [1, 2, 3], y = 2
    Expect: 1
    Actual: 1
    只要答案不在 index = 0 的位置，都會對。
  - D. Test: x = [1, 2, 3], y = 4
    Expect: -1
    Actual: -1
    x 陣列中本來就沒有 y，即使沒經過 i = 0 答案也是對的。
  - E. Test: x = [2, 3, 5], y = 2
    Expect: 0
    Actual: -1
    First Error State: i = 0, before return -1
    應該要比對到 i = 0，但沒有。
  - F. `for (int i=x.length-1; i >= 0; i--)`

- **Find last index of zero**
  - A. Fault: `for (int i = 0; i < x.length; i++)`
    找最後一個 0 需要從後面往前面找第一個出現的 0，應更正為 `for (int i=x.length-1; i >= 0; i--)`。
  - B. 所有輸入都會運行錯誤。
  - C. Test: x = [1]
    Expect: -1
    Actual: -1
    若陣列長度唯一，則低到高或高到低的運行結果是一樣的。
  - D. Test: x = [0, 1]
    Expect: 0
    Actual: 0

陣列中只有一個 0，低到高或高到低運行都只會有一個答案。

E. Test: x = [0, 1, 0]

Expect: 2

Actual: 0

First Error State: i = 0, after i = 0

在 i = 0 時遇到第一個錯誤狀態，在 length-1 的位置有另一個 0。若 x 中有多個 0，則 for-loop 一開始就會遇到第一個錯誤狀態。

F. for (int i=x.length-1; i >= 0; i--)

- **Count positive elements**

A. Fault: if (x[i] >= 0)

0 不是正數，應更正為 if (x[i] > 0)

B. Test: x = []

Expect: 0

Actual: 0

陣列為空就不會執行到錯誤程式碼。

C. Test: x = [1, 2]

Expect: 2

Actual: 2

x 陣列中不包含 0 答案會是對的。

D. 只要陣列包含 0 就一定會錯。

E. Test: x = [-4, 2, 0, 2]

Expect: 2

Actual: 3

First Error State: i = 2, count = 1, before count++

F. if (x[i] > 0)

- **Count odd or postive elements**

A. Fault: if (x[i]%2 == 1 || x[i] > 0)

沒有考慮到奇數為負的狀況，應改為 if (x[i]%2 == -1 || x[i] > 0)

B. Test: x = []

Expect: 0

Actual: 0

陣列為空就不會執行到錯誤程式碼。

C. Test: x = [1, 2]

Expect: 2

Actual: 2

x 陣列中不包含奇數為負的值，答案會是對的。

D. 只要陣列包含奇數為負的值就一定會錯。

E. Test: x = [-3, -2, 0, 1, 4]

Expect: 3

Actual: 2

First Error State: i = 0, count = 0, after if-statement

F. `if (x[i]%2 == -1 || x[i] > 0)`