Homework 1

Below are four faulty programs. Each includes test inputs that result in failure. Answer the following questions about each program.

```
/**    (Code_1)
* 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
```

```
/**    (Code_2)
* 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
```

```
/**    (Code_3)
* 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
```

```
/**    (Code_4)
* 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.

- Code_1: test data輸入後會得到的output為-1；而非預期的0。for loop的終止條件應改為***i >= 0***，才會檢查陣列所有值。

Beautified Java Code

```java
/**  (Code_1)
 * 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
```

- Code_2: test data輸入後會得到的output為0；而非預期的2。for loop的起始與條件應該改為*int i = x.length-1*；終止條件應改為*i >= 0*，才會從末端往前端尋找最後一個0的元素。

Beautified Java Code

```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 = x.length-1; x >=0; i++) {
        if (x[i] == 0) {
            return i;
        }
    }
    return -1;
}
// test: x = [0, 1, 0]; Expected = 2
// Book website: LastZero.java
// Book website: LastZeroTest.java
```

- Code_3: test data輸入後會得到output為3；而非預期的2。原因是判斷positive number的條件為*x[i] >= 0*，應改為*x[i] > 0*。

Beautified Java Code

```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
```

- Code_4: test data輸入後會得到output為2；而非預期的3。原因是判斷odd的條件為*x[i] % 2 == 1*，應改為*x % 2 == -1*，讓負奇數可以被判斷到。

Beautified Java Code

```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
```

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

- Code_1: 一定會引起fault，i>0判斷式必定會被執行。
- Code_2: 一定會引起fault，for loop內容必定會被執行。
- Code_3: x=[]，不會引起fault (x[i]>=0)。
- Code_4: x=[]，不會引起fault (x[i]%2==1)

(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.

- Code_1: x = [2, 4, 6, 8] ; y = 6。會引起fault，但6並非在第一個，不會觸發Error
- Code_2: 會引起fault，也會觸發Error
- Code_3: x = [3, 5, 7, 9]。會引起fault，但全部數字都>0，不會觸發Error
- Code_4: x = [3, 4, 5, 6]。會引起fault，但全部數字都非負奇數，不會觸發Error

(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.

- Code_1: x = [2, 4, 6] ; y = 3。從尾端開始往前端拜訪，i只會拜訪到索引1的地方，但因為還沒找到3，也應該要拜訪索引0，此時卻沒有拜訪，導致Error發生但沒有failure。
- Code_2: x = [2, 3, 4, 0]。此時掃描方向錯誤，會引起Error，但**0只出現一次**，不會出現failure。
- Code_3: 一定會觸發failure。引起Error的情況會在**x裡面出現0**，而只要有0出現，count必定加一，會有非預期的輸出，導致failure必定發生。
- Code_4: 一定會觸發failure。引起Error的情況會在**x裡面出現負奇數**，而只要有負奇數出現，count必定加一，會有非預期的輸出，導致failure必定發生。

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

- Code_1: 首次發生Error在for loop存取陣列第二個數字結束時，由於for loop終止條件設定錯誤，讓陣列第一個數無法被存取到，導致Error發生。
- Code_2: 首次發生Error在for loop存取陣列第一個元素”0”時，由於搜尋方向錯

誤，讓存取第一個數獲得0就回傳，導致Error發生。

- Code_3: 首次發生Error在for loop存取到陣列第三個數”0”時，由於if條件錯誤讓count在此時加一，出現不正確的狀態，導致Error發生。
- Code_4: 首次發生Error在for loop存取到陣列第一個數”-3”時，由於if條件前半段敘述錯誤，讓-3雖為奇數卻不被count所計數，導致Error發生。

(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.

- Code_1:

Hello.java

```
2  public class Hello {
3
4      /**
5       * Find last index of element
6       *
7       * @param x array to search
8       * @param y value to look for
9       * @return last index of y in x; -1 if absent
10      * @throws NullPointerException if x is null
11      */
12     public static int findLast(int[] x, int y) {
13         for (int i = x.length - 1; i >= 0; i--) {
14             if (x[i] == y) {
15                 return i;
16             }
17         }
18         return -1;
19     }
20     // test: x = [2, 3, 5]; y = 2; Expected = 0
21     // Book website: FindLast.java
22     // Book website: FindLastTest.java
23
24     public static void main(String[] args) {
25         int[] x = {2, 3, 5};
26         int y = 2;
27         System.out.println(findLast(x, y));
28
29     }
30
```

Problems @ Javadoc Declaration Search Console

```
<terminated> Hello (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_171
0
```

- Code_2:

Hello.java

```
2  public class Hello {
3
4      /**
5       * Find last index of zero
6       *
7       * @param x array to search
8       *
9       * @return last index of 0 in x; -1 if absent
10      * @throws NullPointerException if x is null
11      */
12     public static int lastZero(int[] x) {
13         for (int i = x.length-1; i >= 0; i++) {
14             if (x[i] == 0) {
15                 return i;
16             }
17         }
18         return -1;
19     }
20     // test: x = [0, 1, 0]; Expected = 2
21     // Book website: LastZero.java
22     // Book website: LastZeroTest.java
23
24     public static void main(String[] args) {
25         int[] x = {0, 1, 0};
26         System.out.println(lastZero(x));
27
28     }
29 }
30
```

Problems @ Javadoc Declaration Search Console

<terminated> Hello (1) [Java Application] /Library/Java/JavaVirtualMachines/j

```
2
```

- Code_3:

Hello.java

```
2  public class Hello {
3
4      /**
5       * Count positive elements
6       *
7       * @param x array to search
8       * @return count of positive elements in x
9       * @throws NullPointerException if x is null
10      */
11     public static int countPositive(int[] x) {
12         int count = 0;
13         for (int i = 0; i < x.length; i++) {
14             if (x[i] > 0) {
15                 count++;
16             }
17         }
18         return count;
19     }
20     // test: x = [-4, 2, 0, 2]; Expcted = 2
21     // Book website: CountPositive.java
22     // Book website: CountPositiveTest.java
23
24     public static void main(String[] args) {
25         int[] x = {-4, 2, 0, 2};
26         System.out.println(countPositive(x));
27
28     }
29 }
30
```

Problems @ Javadoc Declaration Search Console

<terminated> Hello (1) [Java Application] /Library/Java/JavaVirtualMachines/j

2

- Code_4:

Hello.java

```java

public class Hello {

    /**
     * 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

    public static void main(String[] args) {
        int[] x = {-3, -2, 0, 1, 4};
        System.out.println(oddOrPos(x));

    }
}
```

Problems | Javadoc | Declaration | Search | Console

```
<terminated> Hello (1) [Java Application] /Library/Java/JavaVirtualMachines/jdl
3
```