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.

i.

```
/**
* 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
```

- 原程式漏檢查index為0的element，因此需修改迴圈的條件式為 i>=0。

ii.

```
/**
* 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; i >= 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
```

- 原程式的return結果為第一個element值為0的index，須從array x中最後一個元素往前檢查才符合需求。

iii.

```
/**
* 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
```

- 0非正數，須將判斷條件中的等號移除。

iv.

```
/**
* 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
```

- 負奇數mod 2的結果為-1，因此須將判斷式x[i]%2 == 1修改為x[i]%2 == -1 (因為x[i]%2 == 1已包含於x[i]>0)。

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

在所有code中，只要x為null則不會執行到fault，而是進入NullPointerException。

(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.

i. x = [2, 3, 5] ; y = 3, Result = 1

ii. 必定發生error state。Error: i的初始值應為x.length-1，其值卻為0

iii. x = [-4, 2, -1, 2], Result = 2

iv. x = [-2, 2, 0, 1, 4], Result = 3

(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.

i. x = [2, 3, 5] ; y = 0, Result = -1。Error: 出迴圈時i的值應為-1，其值卻為0

ii. x = [1, 1, 0], Result = 2。Error: i的初始值應為x.length-1

iii. error state必定發生在count，而count為return結果，因此必定造成failure。

iv. error state必定發生在count，而count為return結果，因此必定造成failure。

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

i. 出迴圈時i的值應為-1，其值卻為0

ii. i的初始值應為x.length-1，其值卻為0

iii. 執行完x[2]的判斷時，count應為1，其值卻為2

iv. 執行完x[0]的判斷時，count應為1，其值卻為0

(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.