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.

Findlast:

for (int i=x.length-1; i > 0; i--) 沒有包含到 i = 0，應該改成 for (int i=x.length-1; i >= 0; i--)

lastZero:

for (int i = 0; i < x.length; i++) 會從前面開始找’0’，找到就終止，不一定可以找到最後一個’0’，應該改成 for (int i = x.length-1; i >=0; i--)

countPositive:

x[i] >= 0 會把'0'當成positive，應改成 x[i] > 0

oddOrPos:

(x[i]%2 == 1 || x[i] > 0) 會漏負數的奇數，e.g. -3，應改成 (x[i]%2 != 0 || x[i] > 0)

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

Findlast:

X = [], y =1 , Expected = -1

lastZero:

x = [], Expected = -1

countPositive:

x = [-4, 2, 1, 2], Expected = 3 (x[]沒有0，所以沒事)

oddOrPos:

x = [0,1,2,3], Expected = 2(x[]沒有負的奇數，所以沒事)

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

Findlast:

x = [2, 3, 2, 6], y = 2; (Expected = 2, Output = 2)

lastZero:

x = [1, 1, 0]; (Expected = 2, Output = 2)

countPositive:

Impossible, 若要出現faullt, x就要有0, 但是有0,就會有error

oddOrPos:

Impossible, 若要出現fault, x就要存在負奇數, 但是有負奇數, 就會有error

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

Findlast:

x = [7], y = 4; (Expected = -1, Output = -1)

lastZero:

x = [1, 1]; (Expected = -1, Output = -1)

countPositive:

Impossible, 若要出現error, x就要有0, 但是有0,就會有failure

oddOrPos:

Impossible, 若要出現error, x就要存在負奇數, 但是有負奇數, 就會有failure

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

Findlast:

X = [2,3,5], y = 2, Expect = 0

i = 2, no error

i = 1, no error

i = 0, first Error occur

lastZero:

x = [0, 1, 0]; (Expected = 2)

i = 0, fisrt error occur, 因為i=0就回傳0, 應該是在i=2回傳2

countPositive:

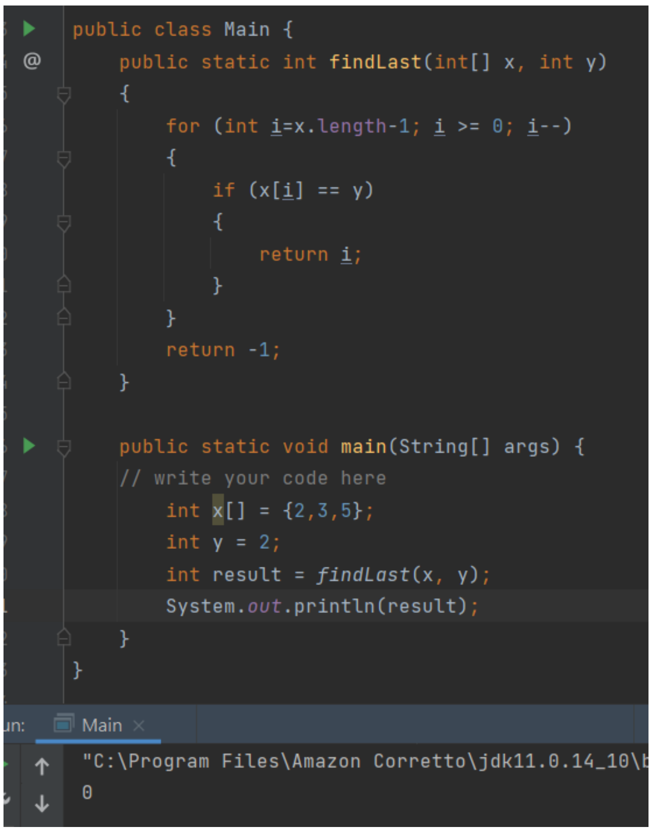
x = [-4,2,0,2], expected = 2
I = 0, no error, count=0
I = 1, no error, count=1
I = 2, first error, count =2, but should be 1
I = 3, error state, count=3, but should be 2

oddOrPos:

x = [-3,-2,0,1,4], expected = 3
I = 0, first error, count = 0, but should be 1
I = 1, error state, count = 0, but should be 1
I = 2, error state, count = 0, but should be 1
I = 3, error state, count = 1, but should be 2
I = 4, error state, count = 2, but should be 3

(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.

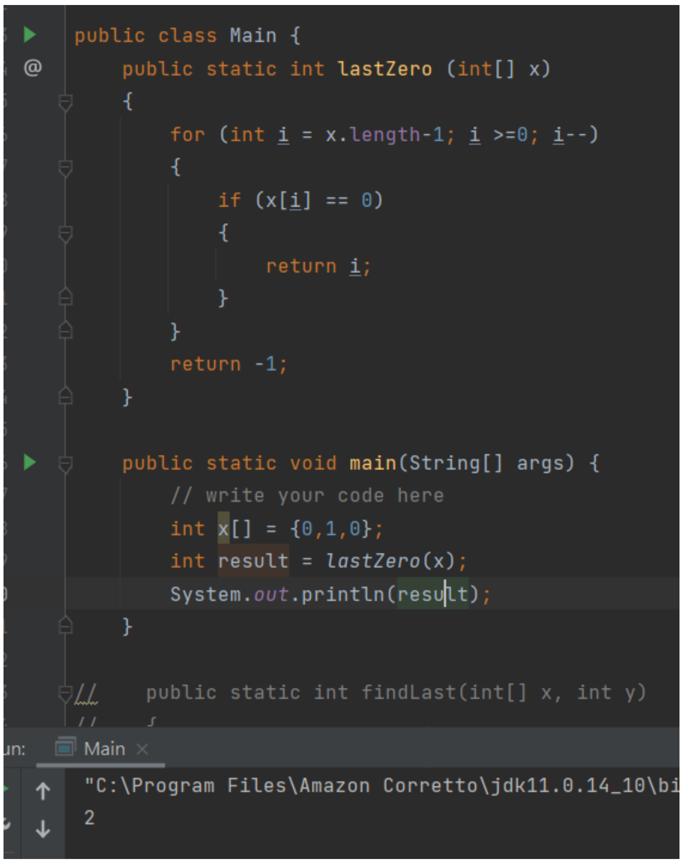
Findlast:



```
public class Main {
    public static int findLast(int[] x, int y)
    {
        for (int i=x.length-1; i >= 0; i--)
        {
            if (x[i] == y)
            {
                return i;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
    // write your code here
        int x[] = {2,3,5};
        int y = 2;
        int result = findLast(x, y);
        System.out.println(result);
    }
}
```

```
"C:\Program Files\Amazon Corretto\jdk11.0.14_10\
0
```

LastZero:

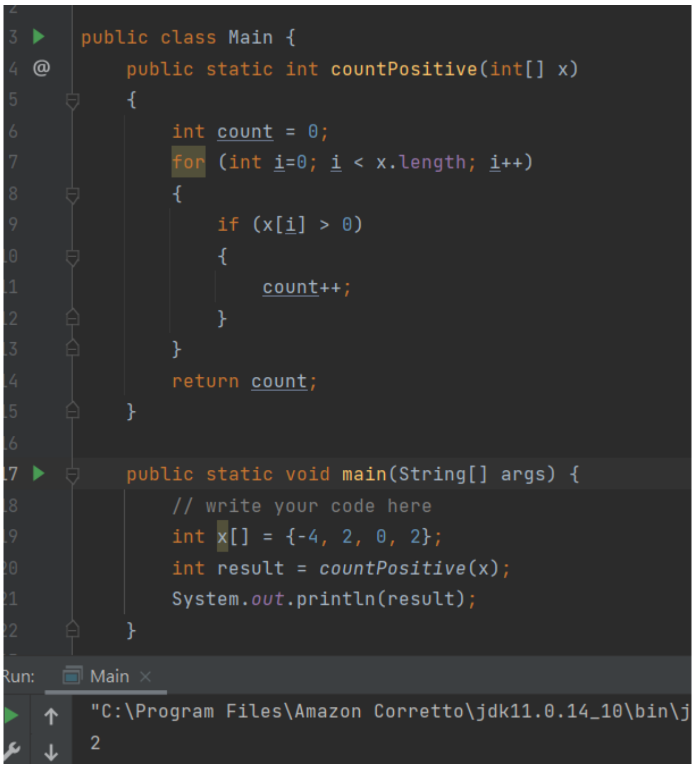


```
public class Main {
    public static int lastZero (int[] x)
    {
        for (int i = x.length-1; i >=0; i--)
        {
            if (x[i] == 0)
            {
                return i;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        // write your code here
        int x[] = {0,1,0};
        int result = lastZero(x);
        System.out.println(result);
    }

//    public static int findLast(int[] x, int y)
```

Run: Main

```
"C:\Program Files\Amazon Corretto\jdk11.0.14_10\bi
2
```

CountPositive:

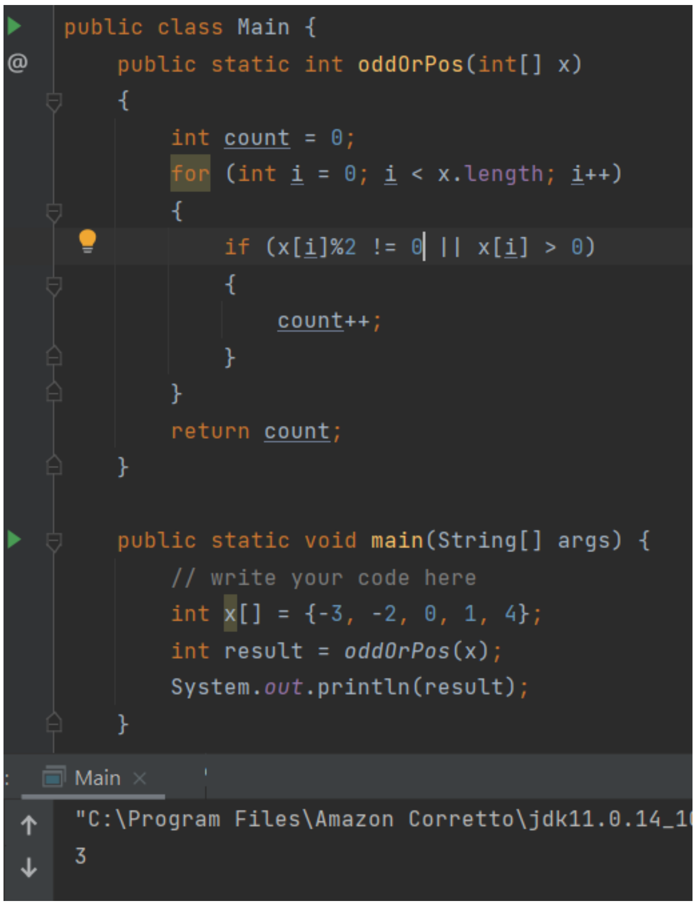


```
public class Main {
    public static int countPositive(int[] x)
    {
        int count = 0;
        for (int i=0; i < x.length; i++)
        {
            if (x[i] > 0)
            {
                count++;
            }
        }
        return count;
    }

    public static void main(String[] args) {
        // write your code here
        int x[] = {-4, 2, 0, 2};
        int result = countPositive(x);
        System.out.println(result);
    }
```

Run: Main

```
"C:\Program Files\Amazon Corretto\jdk11.0.14_10\bin\j
2
```

OddOrPos:



```
public class Main {
    public static int oddOrPos(int[] x)
    {
        int count = 0;
        for (int i = 0; i < x.length; i++)
        {
            if (x[i]%2 != 0 || x[i] > 0)
            {
                count++;
            }
        }
        return count;
    }

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

```
"C:\Program Files\Amazon Corretto\jdk11.0.14_1
3
```