PRÁCTICA INTRODUCCIÓN A PRUEBAS DE SW (TESTING)
TOMÁS GALINDO ENCINAS

Programa FindLast:

El fallo que se produce es cuando el elemento que se quiere buscar está en la primera posición, es decir en la posicion 0 del array. Esto ocurre porque en el bucle for recorre todo el array menos la posicion cero (desde i=x.length-1 hasta i>0); por lo tanto el bucle no entra en la posición cero.

Para corregir bien el codigo hay que cambiar la linea 16 por ----------> for (int i=x.length-1; i >= 0; i--)

El código erróneo siempre va a ser ejecutado, porque siempre va a entran en el array sin entrar en la primera posición; sólo que no nos daremos cuenta del error hasta que el elemento que buscamos esté en la posición 0.

Sin embargo, siempre que busquemos un número que se encuentre en cualquier posición, menos en la posición cero, este programa se va a ejecutar correctamente.

Programa LastZero:

El fallo está cuando hay varios elementos en el array que tienen el valor cero; éste programa va a devolver la posición del primer cero que encuentre y nos deveríamos quedar con el último.

Para arreglar el código hay dos opciones:

1º Opción: darle la vuelta al array, para que en este caso nos quedaremos con el primero del array dado la vuelta, es decir, el último del array original.

cambio línea 13 por for (int i = x.length-1; i >=0; i--)

2º Opción: cambiando el código de esta manera, guardándome en una variable el índice del elemento que sea igual a 0, (dentro del bucle) y si no está que me devuelva -1.

```
public static int lastZero (int[] x){
   int index = -1;
   for (int i = 0; i < x.length; i++){
      if (x[i] == 0){
        index = i;
      }
   }
   return index;
 }
```

El código erróneo va a ser siempre ejecutado, porque siempre va a entrar en el array y dentro del bucle va a actuar de la misma manera (guardando el primero).
Sin embargo, siempre que pongamos un único elemento que sea 0, el programa va a funcionar correctamente.

Programa CountPositive:

El fallo está en que el programa cuenta el valor 0 como número positivo.
Para arreglar el código, hay que cambiar la línea 16 por if (x[i] > 0) para que sólo considere como positivos los números mayores que 0.
El código erróneo siempre va a ser ejecutado, porque se utiliza ese if para realizar la comparación. Pero sólo va a actuar mal cuando se compare el número 0.
El programa siempre va a ser ejecutado correctamente siempre que no haya 0.

Programa OddOrPos:

El problema está cuando hay números negativos, que el módulo de la operación da -1 y nosotros tenemos puesto que debería dar 1 → entonces no entra en el bucle.
Para arreglarlo hay dos opciones:

1º Opción: poner otro Or para que entre en el bucle cuando el módulo es -1
linea 19: if (x[i]%2 == 1 || x[i]%2 == -1 || x[i] > 0)

2º Opción: Usar el valor absoluto del módulo:
linea 19: if (Math.abs(x[i]%2) == 1 || x[i] > 0)

El código erróneo siempre se va a ejecutar para la comparación del if, pero va a actuar mal sólo cuando los números sean negativos e impares.
El programa se va a ejecutar bien, siempre que no haya números negativos impares.