

En este caso no estamos ejecutando el código que tiene el fallo.

3. java LastZero 4 0 6

The last index of zero is: 1

En este caso estamos ejecutando el código que tiene el fallo pero no provoca error ya que al haber un solo 0 no provoca lo mencionado anteriormente.

4 y 5. No se puede provocar un error en el estado pero que no acabe provocando un failure, ya que el programa evalúa directamente los números que metemos por la terminal, entonces si metemos varios 0 provocará un failure y si no metemos ninguno o uno serán los ejemplos de los apartados 2 y 3.

Count Positive

1. En la función countPositive dentro del if está contando también el 0 y el 0 no debe ser tomado como número positivo, luego sustituimos el mayor o igual por mayor y el problema quedaría resuelto.

```
public static int countPositive (int[] x)
{
    int count = 0;

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

2. Como en los casos anteriores, ejecutamos el programa sin meter números.

java CountPositive

Usage: java CountPositive v1 [v2] [v3] ...

3. Para provocar que ejecute el fallo que hay en el código pero que no muestre error basta por meter en la terminal números que no sean 0.

java CountPositive -4 5 3 -2 7

Number of positive numbers is: 3

4 y 5. Al igual que nos pasaba en el caso de LastZero no es posible provocar un error de estado que no provoque un failure, ya que para provocar un error de estado hay que escribir 0 que derivarán en un failure.

OddOrPos

1. En la función que calcula los números impares tenemos que el módulo debe de dar 1, entonces si hacemos el módulo de los números impares cuyo resultado es -1 no se suman al contador. La solución es que el valor absoluto del módulo sea 1 tal y como nuestro a continuación:

```
for (int i = 0; i < x.length; i++)
{
    if (Math.abs(x[i]%2) == 1 || x[i] > 0)
    {
        count++;
    }
}
return count;
```

2. Como en los casos anteriores, si no metemos ningún número al ejecutar el programa la salida del mismo es:

```
java OddOrPos
```

```
Usage: java OddOrPos v1 [v2] [v3] ...
```

3. Si no ponemos un número impar negativo el programa no tiene un error en el estado pero está ejecutando el código que tiene el fallo.

```
java OddOrPos -2 0 1 4
```

```
Number of elements that are either odd or positive is: 2
```

4 y 5. Como en los dos casos anteriores , no hay posibilidad de provocar un error en el estado que no provoque un failure ya que hay que meter todos los números que deseemos desde la línea de comandos, y la única opción de provocar errores en el estado es metiendo números impares negativos que acabarán derivando en un failure.