**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   1. 當x array只有一個element時，findLast一定會回傳   -1，不管x裡有沒有y的值。  For的條件應為 : for (int i=x.length-1; i >= 0; i--)。  (b) 執行NullPointerException時則不會執行fault。  (c) test: x = [1, 2, 5]; y = 2; Expected = 1  (d) test: x=[3, 2, 5]; y = 1 Expected = -1 沒有執行到x[0]  (e) test: x = [2, 3, 5]; y = 2; Expected = 0 Actual = -1 | /\*\*  \* 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   1. For的條件是應為 : for(int i = x.length-1 ;i>=0;i--)   ，否則function找到array裡第一個0就會回傳index。   1. 執行NullPointerException時則不會執行fault。 2. test: x = [2, 1, 0]; Expected = 2 3. test: x = [2, 1, 3]; 執行完I =3，但回傳-1。 4. test: x = [0, 1, 0]; Expected = 2，actual=0。 |
| /\*\*  \* 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   1. if條件式應為(x[i]>0)不然0也會算positive elements 2. 執行NullPointerException時則不會執行fault 3. test: x = [-4, 2, 1, 2]; Expcted = 2 4. test: x = [-4, 2, 0, 2],x[2]為0但會使count++。 5. test: x = [-4, 2, 0, 2]; Expcted = 2 actual=3 | /\*\*  \* 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   1. -3%2為-1，if條件式應為(x[i]%2 == 1 || x[i]%2 == -1 || x[i] > 0) 2. 執行NullPointerException時則不會執行faul 3. test: x = [ 0, 1, 4]; Expected = 2 4. test: x = [-3, -2, 0, 1, 4]; x[0]不會使count++ 5. test: x = [-3, -2, 0, 1, 4]; Expected = 3 actual=2 |

(a) Explain what is wrong with the given code. Describe the fault precisely by proposing a modification to the code.

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

(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.

(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.

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

(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.

1