Consider this correctness proof for countZeros work (Exercise 1 from Chapter 19 Program Correctness Proofs reading)?

```
1:    int countZeros(int array L) {
2:         // postcondition: returns number of zero elements in L
3:         @          0 == # 0 in L[0..-1] AND 0 <= L.length
4:         int count = 0;
5:         @          count == # 0 in L[0..-1] AND 0 <= L.length
6:         int index = 0;
7:         @          count == # 0 in L[0..index-1] AND index <= L.length
8:         while (index < L.length()) {
9:             @              index < L.length AND count == # 0 in L[0..index-1]
10:            if (L[index] == 0) {
11:               @ CASE T:       L[index] == 0 AND count == # 0 in L[0..index-1]
12:                  count = count + 1;
13:             } else {  // CASE F: L[index] != 0 AND count == # 0 in L[0..index]
14:              @          AND index +1 <= L.length
15:             } // else {
16:            index = index + 1;
17:            @              count == # 0 in L[0..index-1] AND index <= L.length
18:         } // while
19:         @ NOTE: (NOT (index < L.length)) == index >= L.length
20:        @          index >= L.length AND
21:         @          count == # 0 in L[0..index-1] AND index <= L.length
22:         @          count == # 0 in L[0..L.length-1]
23:         @          return count;
24:    }
```

1a) What happens if index = index + 1 is replaced by index = index + 2 and how is this reflected in the correctness proof?

1b) How would you prove that countZeros never goes into an infinite loop?
2) a) Using pseudo-code and assuming the function ‘random()’ returns a random integer, sketch a program that would randomly test the countZeros method.
b) How are you handling the ‘test oracle’ issue?

3) If the line:
   5:   while (index < L.length()) {
were miswritten as:
   5:   while (index <= L.length()) {
What test case would catch this error (assume the program behaves like a C program)?