Choosing Test Cases
Let’s talk about what to test.

In Lab2 you create a class MyArrayList\<E\> with methods
  a) 2 constructors
  b) int size()
  c) boolean add(E item) void add(int index, e item)
  d) E get()
  e) E set(int index, E item)
  f) E remove(int index)
  g) boolean isEmpty()
  h) void clear()
There are 3 ways a method can go wrong:
   A. its basic functionality could be wrong,
   B. it might fail at the extremes (What if the list has 0 or 1 entry? What if it is full?)
   C. it could fail by interacting badly with another method.

To test our code we want to write test cases that cover these possibilities as thoroughly as possible.
Suppose we are testing boolean add( E item ) Here are some things to test:

A) Basic functionality: If we add some data, can we get it from the end? Can we do a sequence of adds? Does add always return true?

B) Extreme cases: Can we add to the empty list? If we start with a list of capacity N, can we put more than N items in it?

C) Interactions with other methods: Is the size correct after an add?
All of our test cases are covered by the following method:

```java
void testAddE() {
    myArrayList<Integer> L = new myArrayList<Integer>(2);
    Boolean returnVal = true;
    for (int i = 0; i < 5; i++) {
        returnVal = returnVal && L.add(i*i);
        AssertEquals(L.size(), i+1);
    }
    AssertTrue(returnVal);
    for (int i = 0; i < 5; i++) {
        int j = L.get(i);
        AssertEquals( j, i*i);
    }
}
```
Now suppose we are testing the two-argument add: void add(int index, E item). Here are issues to test:

A) Basic functionality: If we add some data, can we retrieve it with get()? Can we do a sequence of adds?

B) Extreme cases: Can we add to the empty list?
   - If size == N, can we add at index 0, and index N?
   - If we start with a list of capacity N, can we put more than N items into it?
   - Do we throw the right exceptions if the index is too small or too big?

C) Interactions with other methods: Is the size correct after an add?
In addition to the tests for the 1-argument add, you might have this in testing the 2-argument add:

```java
MyArrayList<Integer> L = new MyArrayList<Integer>();
int [] A = {5, 10, 25};
for (int x: A)
    L.add(x);
L.add(3, 500);
L.add(0, 100);
AssertEquals( L.get(0), 100);
AssertEquals( L.get(1), 5);
AssertEquals( L.get(2), 10);
AssertEquals( L.get(3), 25);
AssertEquals( L.get(4), 500);
```