1 Boxes and Pointers II

Draw a box and pointer diagram for each code block.

(a) \[
\text{int}[n] x = \{1, 2, 3\};
\text{int}[n] y = x;
y[2] = 7;
\]

(b) \[
\text{IntList} l = \text{IntList.list}(1, 2, 3);
\text{IntList} l2 = l;
l.\text{tail}.\text{tail}.\text{head} = 7;
\]

(c) \[
\text{IntList}[n] ll = \text{new IntList}[3];
ll[0] = \text{IntList.list}(1, 2);
ll[1] = \text{IntList.list}(2);
\]

2 Debugging is good for your health

The following code is broken. Please identify and fix the errors.

```java
/** Returns the sum of squares of numbers in nums. */
public int sumOfSquares(int[] nums) {
    int total = 0;
    for (int i = 0; i < nums.size; i += 1) {
        total += (nums + i) * (nums + i);
    }
    return total;
}
```

3 Fun with arrays

Complete the following methods according to their specifications.

```java
/** Given an array A (size > 1), return the avg. of all items in A. */
public static double average(double[] A) {
    // Your code here
}
```
import static java.lang.Math.max; // max(a, b) returns max of a, b
import static java.lang.Math.min; // min(a, b) returns min of a, b

/** Given an array A, return a 2 element array B where B[0] is the
 * minimum element of A and B[1] is the maximum element of A. */
public static int[] minMax(int[] A) {

4 Bonus for Bosses: LinkedFaceitteroogle Interview

Welcome to LinkedFaceitteroogle! I hear you’re interested in an engineering position here. First,
let’s see how well you can program.

Given an integer \( k \) and an array \( A \) of \( n \) integers, design an algorithm to move \( A[k] \) to the left-most
index such that all elements up to index \( k \) are sorted in increasing order. You may assume that prior
to moving \( A[k] \), all elements up to index \( k - 1 \) are sorted in increasing order and that \( k < n \).

(a) public static void moveInt(int[] A, int k) {

(b) How can we extend this algorithm to sort an entire array?