

Name:

Jon

SID:

Section Number:

01 
  02 
  03 
  04 
  05 
  06 
  07 
  08 
  09 
  10 
  11 
  12

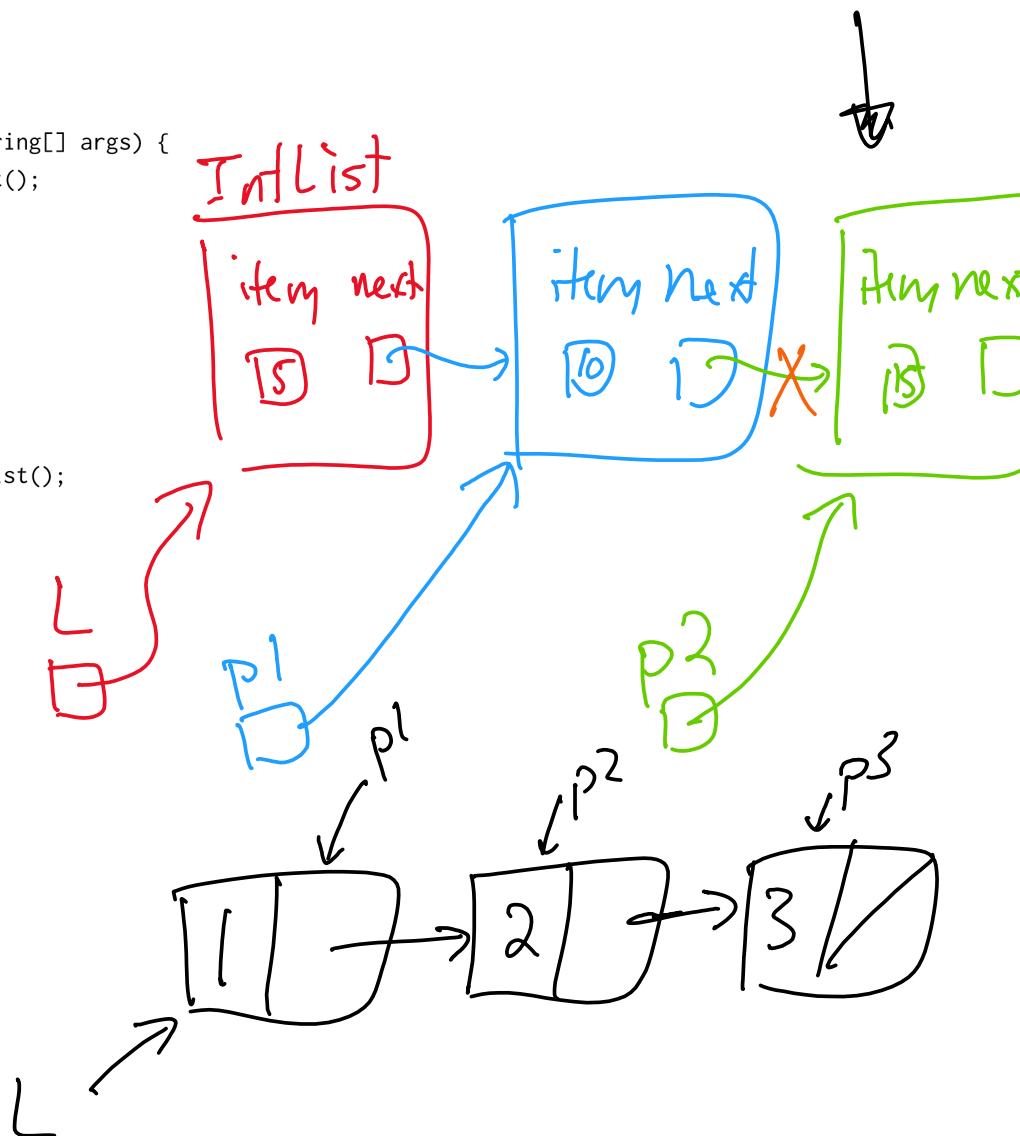
Write your name and login above. Please complete this worksheet during your lab, and turn it in to your TA by the end of your section. You are encouraged to work with your partners and neighbors collaboratively.

### 1 IntList Box and Pointer Diagram

1.1 Draw the box-and-pointer diagram that results from running the following code.

```

1 public class IntList {
2     public int item;
3     public IntList next;
4
5     public static void main(String[] args) {
6         IntList L = new IntList();
7         L.item = 5;
8         L.next = null;
9
10        L.next = new IntList();
11        L.next.item = 10;
12        IntList p1 = L.next;
13
14        L.next.next = new IntList();
15        L.next.next.item = 15;
16        IntList p2 = p1.next;
17        p1.next = null;
18    }
19 }
    
```



## 2 Iteration vs. Recursion

- 2.1 Consider the two different implementations of the `size` method provided in the lab spec. The method is first implemented using iteration (with a `while` loop) and the other method is implemented using recursion.

Answer the following questions with your partner and write down your conclusions.

- (a) How does the recursive `size` keep track of the pointer, `p`, that we initialize in `iterativeSize`?
  
- (b) How does the recursive `size` keep track of the `totalSize` which we had to create a variable for in `iterativeSize`?
  
- (c) Why does the base case for `size` terminate when `next == null`, rather than when `this == null`?