143 15wi midterm key

1.

```
a) [1, 3, 22, 42]
b) [3, 20, 7, 44, 80]
c) [20, 1, 30, 62, 46, 88]
```

2.

```
mystery(22, 1)            3
mystery(12, 22)          1
mystery(35, 331)          4
mystery(7162, 7567)       2
mystery(45331, 321)       46
```

3.

```
public static boolean isReverseAlphabetical(Stack<Character> s) {
       if (s.size() < 2) {
           return true;
       }

       boolean sorted = true;
       char prev = s.pop();
       Queue<Character> backup = new LinkedList<Character>();
       backup.add(prev);
       while (!s.isEmpty()) {
           char curr = s.pop();
           backup.add(curr);
           if (prev - 1 != curr) {
               sorted = false;
           }
           prev = curr;
       }

       while (!backup.isEmpty()) {
           s.push(backup.remove());
       }

       while (!s.isEmpty()) {
           backup.add(s.pop());
       }

       while (!backup.isEmpty()) {
           s.push(backup.remove());
       }

       return sorted;
   }
```

4.

a)
```
list2 = list1;
list1 = list1.next;
list2.next = null;
```

b)
```
list2 = list1.next;
list1.next = list1.next.next;
list2.next = null;
```

c)
```
ListNode temp = list2;
list2 = list2.next;
temp.next = list1.next;
temp.next.next = list1;
list1 = temp;
list1.next.next.next = null;
```

d)
```
list2.next.next = list1.next.next;
list1.next.next = list2.next;
list2.next = list1;
list1 = list1.next;
list2.next.next = null;
```

5.

```
public static String switchCase(String s) {
    if (s.length() == 0) {
        return "";
    } else {
        String rest = switchCase(s.substring(1));
        if (Character.isUpperCase(s.charAt(0))) {
            return Character.toLowerCase(s.charAt(0)) + rest;
        }
        else {
            return Character.toUpperCase(s.charAt(0)) + rest;
        }
    }
}
```

6.

```
public static Set<String> popularDepartments(List<String> classes) {
    // count the classes offered by each department
    Map<String, Integer> m = new TreeMap<String, Integer>();
    for (int i = 0; i < classes.size(); i++) {
       String dept = classes.get(i).substring(0, 3).toLowerCase();
       if (m.containsKey(dept)) {
          m.put(dept, m.get(dept) + 1);
       } else {
          m.put(dept, 1);
       }
    }

    // put the desired members of the map into a set
    Set<String> set = new TreeSet<String>();
    for (String s : m.keySet()) {
       if (m.get(s) >= 4) {
          set.add(s);
       }
    }
    return set;
 }
```