1) What does the following code print when i = 1?

```java
public static void rec1 (int i) {
    if (i==0)
        System.out.print(i + " ");
    for (int j=0; j<2; j++) {
        rec1(i-1);
        rec1(i-1);
    }
}
```

2) What does the following code print when i = 1?

```java
public static void rec2 (int i) {
    if (i==0) {
        System.out.print(i + " ");
    } else {
        for (int j=0; j<2; j++) {
            rec2 (i-1);
            rec2 (i-1);
        }
    }
}
```

3) What does rec return when list = {1,3,5,7,9}?

```java
public int rec(int [] list){
    return rec3(list, 0);
}

public int rec3(int [] list, int start){
    if (start == list.length - 1) {
        return list[start];
    } else {
        return Math.max(list[start], rec3(list, start + 1));
    }
}
```
Use this definition for 4 & 5. In this recursive definition with two cases:

- A **list of names** is
  1. A name **(no spaces)**
  or
  2. A name followed by a **semicolon and a space** followed by a **list of names**

4) Which of the following match the definition of a list of names below?
   A. Bob Alice
   B. Bob; Alice;
   C. Alice; Mary; Ted
   D. Owen;
   E. None of the above

5) How about this list?
   A. Bob Alice
   B. Bob; Alice;
   C. Alice; Mary; Ted
   D. Owen;
   E. None of the above

6) Given the following code snippet, how many times does the call tree include a base case?

```java
System.out.println(combRec(4,2));

public long combRec(long n, long k){
    if (n==k || k==0)
        return 1;
    else
        return combRec(n-1,k-1) +
               combRec(n-1,k);
}
```