SAT: Functional Programming

1. What side effect does the following function have? How could it be rewritten to avoid side effects?

```
// Doubles the values of all elements in an array.
public static void doubleAll(int[] a) {
    for (int i = 0; i < a.length; i++) {
        a[i] = 2 * a[i];
    }
}
```

```
public static int[] doubleAll(int[] a) {
    int[] aCopy = Arrays.copyOf(a, a.length);
    for (int i = 0; i < a.length; i++) {
        aCopy[i] = 2 * a[i];
    }
    return aCopy;
}
```

2. Write a lambda expression that converts an integer into the square of that integer. For example, 4 would become 16.

```
x -> x * x
```

3. Write a lambda expression that accepts two strings representing a first and last name and concatenates them together into a string in "Last, First" format. For example, if passed "Cynthia" and "Lee", it would return "Lee, Cynthia".

```
(f,l) -> f + ", " + l
```

4. Write a method `sumAbsVals` that uses stream operations to compute the sum of the absolute values of an array of integers. For example, the sum of {-1, 2, -4, 6, -9} is 22.

```
public static int sumAbsVals(int[] a){
    return Arrays.stream(a)
                .map(x -> Math.abs(x))
                .sum();
}
```

5. Write a method `largestEven` that uses stream operations to find and return the largest even number from an array of integers. For example, if the array is {5, -1, 12, 10, 29, 2, 8}, your method should return 12. You may assume that the array contains at least one even integer. You must use stream operations to solve this problem. Do not use any loops or recursion.

```java
public static int largestEven(int[] a){
    return Arrays.stream(a)
                .filter(x -> x % 2 == 0)
                .max()
                .getAsInt();
}
```

6. Write a method `countNegatives` that uses stream operations to count how many numbers in a given array of integers are negative. For example, if the array is {5, -1, -3, 20, 47, -10, -8, -4, 0, -6, -6}, return 7. You must use stream operations to solve this problem. Do not use any loops or recursion.

```java
public static long countNegatives(int[] a){
    return Arrays.stream(a)
                .filter(x -> x < 0)
                .count();
}
```

7. Write a method `printDoubled` that uses stream operations to print twice the value of each element of array of integers. For example, if the array passed is {2, -1, 4, 16}, print:

```
4
-2
8
32
```

You must use stream operations to solve this problem. Do not use any loops or recursion.

```java
public static void printDouble(int[] a){
    Arrays.stream(a)
            .map(x -> x * 2)
            .forEach(System.out::println);
}
```