GROUP 3: Algorithms

a)

```
void a(int[] numbers) {
    for(int i = 0; i < numbers.length; i++) {
        int target = numbers[i];
        // the following function call uses sequential search to count all
        // of the occurrences of the given target in the given array of
        // values
        if (sequentialSearchCount(numbers, target) == 1) {
            System.out.println("only one " + target + " found in array");
        }
    }
}
```

b)

```
void b(int[] numbers) {
    // "enhanced for loop" used to go through all values in the array
    // much like how Python's for loop works. The equivalent Python code
    // would be "for target in numbers:" (without the quotes)
    for(int target : numbers) {
        // the following function call uses binary search to search for
        // the given target in the array. Since all values come from the
        // array, we can use this to test to see if our binary search
        // function is working correctly
        if (binarySearch(numbers, target) == false) {
            System.out.println("Binary search failed to find target");
        }
    }
}
```

c)

```
void c(float[] values) {
    System.out.println("Original values:" );
    for(int i = 0; i < values.length; i++) {
        System.out.print(values[i] + " ");
    }

    bubbleSort(values);

    System.out.println("Sorted values:" );
    for(int i = 0; i < values.length; i++) {
        System.out.print(values[i] + " ");
    }
}
```

d)

```
void d(double[] values, double target) {
    sequentialSearch(values, target);
    selectionSort(values);
    binarySearch(values, target);
}
```

e)

```
void e(String[] names, String[] ranks) {
    String[] namesAndRanks = new String[names.length];
    for(int i = 0; i < names.length; i++) {
        namesAndRanks[i] = names[i] + " " + ranks[i];
    }

    String combined = "";
    for(int i = 0; i < namesAndRanks.length; i++) {
        combined += namesAndRanks[i] + ", ";
    }

    System.out.println(combined);
}
```