1. (6 points) Fill in the chart below, by giving the output of the code below. If a statement would result in an error, you may simply write “Error” as the output. Recall that the ASCII values for ‘A’ - ‘Z’ are 65-90 and ‘a’ - ‘z’ are 97-122.

```java
String s1 = "blue red";
System.out.println("1: " + s1.substring(2));
String s2 = "blue";
s2 += " red";
System.out.println("2: " + (s1 == s2));
s2 = "BLUE RED";
System.out.println("3: " + s2.compareTo(s1));
System.out.println("4: " + s2.compareTo(s2));
s1.toUpperCase();
System.out.println("5: " + s1);
System.out.println("6: " + s2.replaceAll("E", "B"));
```

**Solution:** Answers:

1: “ue red”

2: false (remember, must use .equals for String comparison)

3: -32 (half credit for +/- error)

4: 0 (another way of comparing equality in Strings)

5: “blue red” (remember, NO method in the String class will change a String)

6: “BLUB RBD”

2. (4 points) Consider the function mystery below. What would a call of mystery(8) return?

```java
public static int mystery(int x) {
    if (x < 1) {
        return x;
    } else {
        return x + mystery(x - 2);
    }
}
```

**Solution:** 20
3. We want to write a function which takes a String as input and returns a new String that is the reverse of the original String (so if the input is “hello” the returned String would be “olleh”). A recursive implementation of this function is:

```java
public static String reverse(String s) {
    if (s.length() == 0) {
        return "";
    }
    String delegated = reverse(s.substring(1, s.length()));
    return delegated + s.charAt(0);
}
```

(a) (2 points) Describe (in English, not code!) the base case(s) of this algorithm.

**Solution:** The reverse of an empty string will also be an empty string. (lines 2-4).

(b) (1 point) Describe (in English, not code!) how we reduce the problem in scope so that we will eventually reach the base case.

**Solution:** A smaller String made by removing the first character from the original String (line 6)

(c) (2 points) Describe (in English, not code!) the “small bit of work” that this function does to incorporate the recursive solution.

**Solution:** It appends the character in the first position to the last position by appending it to the end of the reversed String that is returned by our “clone”.