Name (print): __________________________________________________________

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.

   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") );

   Answers:
   1:  
   2:  
   3:  
   4:  
   5:  
   6:  

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

   public static int mystery(int x) {
      if (x < 1) {
         return x;
      }
   else {
         return x + mystery(x - 2);
      }
   }
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.
(b) (1 point) Describe (in English, not code!) how we reduce the problem in scope so that we will eventually reach the base case.
(c) (2 points) Describe (in English, not code!) the “small bit of work” that this function does to incorporate the recursive solution.