Assignment 2 - Rainbow Functions

```
public class RainbowTable {

    public static void main(String[] args) {
        long res = 0;
        int i;
        String start;

        // Creating two arrays. One for the start values and the other for the end values

        ArrayList<String> sValues = new ArrayList<String>();
        // Populating array
        sValues.add("Kermit12");
        sValues.add("Modulus!");
        sValues.add("Pigtail1");
        sValues.add("GalwayNo");
        sValues.add("Trumpets");
        sValues.add("HelloPat");
        sValues.add("pinky##!");
        sValues.add("01!19!56");
        sValues.add("aaaaaaaa");
        sValues.add("036abgH");

        ArrayList<String> eValues = new ArrayList<String>();
        // populating array
        eValues.add("lsXcRAuN");
        eValues.add("L2rEsY8h");
        eValues.add("R0NoLf0w");
        eValues.add("9PZjwF5c");
        eValues.add("!oeHRZpK");
        eValues.add("dkMPG7!U");
        eValues.add("eDx58HRq");
        eValues.add("vJ90ePjV");
        eValues.add("rLtVvpQS");
        eValues.add("klQ6IeQJ");


        if (args != null && args.length > 0) { // Check for <input> value
            start = args[0];

            if (start.length() != 8) {
                System.out.println("Input " + start + " must be 8 characters long - Exi
t");
            } else {
                // hash code // problem 1
```

```
                System.out.println("+--------------------+\n|\tProblem 1\n+---------------
-----+");
                String str = sValues.get(0);
                long hash = 0;
                for(int k = 0;k<10000;k++){
                    hash = hashFunction(str);
                    str = reductionFunction(hash, k);
                }

                System.out.println(sValues.get(0) + " -> " + str);

                // adding the hash values for problem 2 into an array
                ArrayList<String> hashValues = new ArrayList<>();
                hashValues.add("895210601874431214");
                hashValues.add("750105908431234638");
                hashValues.add("111111111115664932");
                hashValues.add("977984261343652499");

                System.out.println("+--------------------+\n|\tProblem 2\n+---------------
-----+");

                // Searching through each starting element
                sValues.forEach((element) -> {
                    String str_ = element;  // keeping track of the original string \\
                    long hash_ = 0;
                    // for loop to iterate over hash and reduction function 10,000 times
                    for(int k = 0; k<10000; k++){
                        // hashFunction returns long from starting value
                        hash_ = hashFunction(str_);
                        // reductionFunction converts it to a string value
                        str_ = reductionFunction(hash_, k);

                        // We check if any of the iterations collide with our 4 hash value
s
                        if(hashValues.contains(Long.toString(hash_))){
                            System.out.println("Hash Collision found! " + "Starting value:
" + element +
                            ", current value: "  + str_ +  " and its hash: " + hash_);
                        }
                    }
                    //System.out.println("Starting value: " + element + " -> End value: "
 + str);
                });


            }
        } else { // No <input>
            System.out.println("Use: RainbowTable <Input>");
        }
    }
```