

# CT255 Assignment 2

## Rainbow Tables

### Problem 1 Source Code:

```
else {
    //Passed value passed into the program is assigned as the first value
    System.out.println("Start value pair: "+start);
    for (i=0; i<10000; i++) {
        //The reduction function is run 10000 times, and the rounding value is
        also incremented (Starts at 0)
        start = reductionFunction(hashFunction(start), i);
    }
    //End value is printed to the screen
    System.out.println("End value pair after 10000 iterations: "+start);
}
```

### Problem 2 Source Code:

```
else {
    //Long array with given hash values in problem 2.
    long[] matches = new long[]{895210601874431214L, 750105908431234638L,
111111111115664932L,
        977984261343652499L};
    long hashGen;
    //Passed value passed into the program is assigned as the first value
    String stringGen = start;
    //For loop is used to search and iterate over each password hash.
    for(int matchIndex=0; matchIndex<4; matchIndex++) {
        System.out.println("\nStart value pair: "+stringGen);
        System.out.println("Searching for "+matches[matchIndex]);
        for (i = 0; i < 10000; i++) {
            hashGen = (hashFunction(stringGen));
            //If a generated hash is found to be the same as the given hash
            value thats being searched for, inform
            //the user
            if (hashGen == matches[matchIndex]) {
                //The password that matches the given hash value is printed to
                the screen.
                System.out.println("\nHash: " + matches[matchIndex] + "
found!\nChain element match " + stringGen+"\n");
            }
            //The loop continues searching for more matches
            //The reduction function is run 10000 times, and the rounding
            value is also incremented (Starts at 0)
            stringGen = reductionFunction(hashGen, i);
        }
        //String chain is reset and loop starts again to look for matches with
        the next hash value
        //End value is printed to the screen
        System.out.println("End value pair after 10000 iterations:
"+stringGen);
        stringGen = start;
    }
}
```

### Problem 2 Password Finds / Solution:

**1<sup>st</sup> Hash Match:** Password: Pigtail1. Hash Found: 977984261343652499. Chain Element Match: hPef!9lq

**2<sup>nd</sup> Hash Match:** Password: aaaaaaaa. Hash Found: 895210601874431214. Chain Element Match: BAG94NPY