Problem 1

```
import java.util.Random;
import java.util.*;
import java.time.LocalDate;
import java.util.HashMap;
import java.util.Map;

/**
 * Write a description of class CT437_HashFunction1 here.
 *
 * @gbadebo
 * @version (a version number or a date)
 */
public class CT437_HashFunction1 {
    //HashMap<Integer, String> hash_map = new HashMap<Integer, String>();
    //HashMap<String, Integer> hash_map = new HashMap<String, Integer>();
    public static void main(String[] args) {
        List<String> stringList = new ArrayList(); //List used to hold randomly generated
string
        List<Integer> testList = new ArrayList(); //List used to hold hash strings
        int res = 0; //Used to hold hash value
        boolean c = false; // Sets boolean value to false



        if (args != null && args.length > 0) { // Check for <input> value
            res = hashF1(args[0]); // call hash function with <input>
            if (res < 0) { // Error
                System.out.println("Error: <input> must be 1 to 64 characters long.");
            }
            else {
                System.out.println("input = " + args[0] + " : Hash = " + res);
```

```
            System.out.println("Start searching for collisions");
            // Your code starts here!
            // 77,000 is chosen as according to the Birthday Problem 77,000
            // strings are needed for a 32-bit integer to have a
            // probability of 50% of having a hash collision
            for(int i = 0; i <= 77000; i++) {
                stringList.add(RandomString()); //Randomly generated strings
                                                //are added to list
            }

        for (int j = 0; j < stringList.size(); j++) {
             testList.add(hashF1(stringList.get(j))); //Randomly generated strings are
hashed and added to list
            if (testList.get(j).equals(res)){ //IF-loop proceeds when hash value of
randomly generated string equals hash value of inputted string
            String inp = stringList.get(j); //Get value of randomly generated hash string
before it was hashed
            String ans = String.valueOf(testList.get(j)); //Get hash value of randomly
generated string
            System.out.println("String: " + inp + " has collision value " + ans + " !");
//Prints results of hash collision
            c = true; //c is equal to true if hash collision is found
            }
        }
            if (!c) {
        System.out.println("No collision found"); // Sets c is equal to false if no hash
collision is found
    }

    }}
    else { // No <input>
        System.out.println("Use: CT437_HashFunction1 <Input>");
```

```
        }
    }

   public static String RandomString() {
    int leftLimit = 48; // numeral '0'
    int rightLimit = 122; // letter 'z'
    Random randomGenerator=new Random();
    int targetStringLength = randomGenerator.nextInt(64) + 1; //Length of string is
randomly generated between 1-64
    Random random = new Random();

    //Random string is generated and returned
    String generatedString = random.ints(leftLimit, rightLimit + 1)
      .filter(i -> (i <= 57 || i >= 65) && (i <= 90 || i >= 97))
      .limit(targetStringLength)
      .collect(StringBuilder::new, StringBuilder::appendCodePoint,
StringBuilder::append)
      .toString();

      return generatedString;  //Random string is returned
    //System.out.println(generatedString);
}

    private static int hashF1(String s){
        int ret = -1, i;
        int[] hashA = new int[]{1, 1, 1, 1};
        String c =s;
        String filler, sIn;
        filler = new
String("ABCDEFGHABCDEFGHABCDEFGHABCDEFGHABCDEFGHABCDEFGHA
BCDEFGHABCDEFGH");
```

```
        if ((s.length() > 64) || (s.length() < 1)) { // String does not have required length
            ret = -1;
        }
        else {
            sIn = s + filler; // Add characters, now have "<input>HABCDEF..."
            sIn = sIn.substring(0, 64); // // Limit string to first 64 characters
            // System.out.println(sIn); // FYI
            for (i = 0; i < sIn.length(); i++){
                char byPos = sIn.charAt(i); // get i'th character
                hashA[0] += (byPos * 17); // Note: A += B means A = A + B
                hashA[1] += (byPos * 31);
                hashA[2] += (byPos * 101);
                hashA[3] += (byPos * 79);
            }

            hashA[0] %= 255;  // % is the modulus operation, i.e. division with rest
            hashA[1] %= 255;
            hashA[2] %= 255;
            hashA[3] %= 255;

            ret = hashA[0] + (hashA[1] * 256) + (hashA[2] * 256 * 256) + (hashA[3] * 256
* 256 * 256);
            if (ret < 0) ret *= -1;
        }
        return ret;
    }
}
```

List of up to 10 hash collisions found.

```
BlueJ: Terminal Window - Comp
Options
input = Bamb0 : Hash = 1079524045
Start searching for collisions
String: PAyp3sruLaS9CjPGDR has collision value 1079524045 !
String: 67ldo9 has collision value 1079524045 !
String: g0lHW has collision value 1079524045 !
String: yZ has collision value 1079524045 !
String: vEWuIc7PNuehOG5v has collision value 1079524045 !
String: O2heT has collision value 1079524045 !
String: rs6Ds02sdfpQKGWp has collision value 1079524045 !
String: XFMrPCwI1OdOCCToKkoT has collision value 1079524045 !
String: 1HyLhN3 has collision value 1079524045 !
String: EkVKc1 has collision value 1079524045 !
String: iFxpQWvH3YpexJFyhJyN0V70bAdvJ has collision value 1079524045 !
String: EMgsuXGePOeR6wb has collision value 1079524045 !
```

Problem 2

```
//XOR encryption is now used for hash function
    //By using XOR encryption it is impossible to decrypt the data using
    //brute force attack as to decrypt the data you need to know the
    //encryption key
    private static String hashF1(String s){
// Encryption Key is defined
        char encryptKey = 'X';

        //String to store encrypted/decrypted String is defined
        String res = "";

        //length of input string is calculated
        int inputLength = s.length();

        // XOR operation of key is performed
        // with every character in string
        for (int i = 0; i < inputLength; i++)
        {
            res = res +
            Character.toString((char) (s.charAt(i) ^ encryptKey));
        }
```

```
        return res; //return hash value
    }
}
```

Results before hashF1 is modified.

```
BlueJ: Terminal Window - Comp
Options
input = Search125 : Hash = 1978057711
Start searching for collisions
String: uDdNvQBrc509YnBbDQCDfQxaNxGUclZAo2FyKaGoF303tuPiwIBBoD9 has collision value 1978057711 !
String: ylbxH1ZH53ipTo5FNLwRYZbjmHI3azg7s has collision value 1978057711 !
String: PF82WPQRXUKIhC0d has collision value 1978057711 !
String: S3aUgBKQm has collision value 1978057711 !
String: i7siyi7lxBiyjrqeYaZOSSmcmFets4vjHsqSj1QVgKGU262FyL1P has collision value 1978057711 !
String: LozqEyHAsHyiRtUZBhgqYZWZFpJGcC6Rz4DA6z43nti5m has collision value 1978057711 !
String: NSaeA1fC9NzY7WXazwklQtEXRMTyEv5RW93l has collision value 1978057711 !
String: t02gxm1WowpkQMvnp6CYTD1u17B6DgRsFuoC has collision value 1978057711 !
String: vc7l7Kg has collision value 1978057711 !
String: VkWNRfkIiI037nTaSOdw5iOFNVvdQbSc70cLMzF9 has collision value 1978057711 !
String: BgP89uNEs3c has collision value 1978057711 !
```

```
BlueJ: Terminal Window - Comp
Options
input = hello : Hash = 1853942169
Start searching for collisions
String: algaNV3m331JaHgrQ4CDhsCOUQaIgkpV1rRz8daQuUia1QfSk6AFfu has collision value 1853942169 !
String: mBUVwWgJB3aE has collision value 1853942169 !
String: 7OP0pJtZOXGx has collision value 1853942169 !
String: XkddjtyBzsmtEcfRn9umlwUeigALlJjxBCBpj73TEQFi5KHvyJ6IW has collision value 1853942169 !
String: rBfdM6duH has collision value 1853942169 !
String: ux6pfH9p0s3sG6pTN4Mo3rYlpUEleAZNU3wSx2sIqOcaPxf has collision value 1853942169 !
String: Q7Yo1oH07MojIWWUXdTABnsd3CUYIpX6VGf10ym96cvdhW has collision value 1853942169 !
String: IGI34pDX2bYjXQk9wq2sLD22Cf3ivfqfFS77u3oluEtOuMZnN91AoJIxNah has collision value 1853942169 !
String: Br3bhHWX4izQS76NlA6XCfftQ3vnOY5GMQL9UFqpPezL has collision value 1853942169 !
String: eCthduhIkVdsIDBfjwUB6hSZIF50fsM8xk has collision value 1853942169 !
String: q8owrV has collision value 1853942169 !
String: LnJ28Mxu4SJh8zLRX2N7xiTpDSz has collision value 1853942169 !
String: qjuzJ has collision value 1853942169 !
String: FGpXPrZnJL4oGWW8iu5ysH223ji has collision value 1853942169 !
String: F1w92ubGuU85QdwOIbq6oA3GY0THy5AoVrArVIhv05mrvx721YRu9VNwF1VJmS3i has collision value 1853942169 !
String: g1UBebYEacoIejqRnvQQAxmSb4A5qyNFAg has collision value 1853942169 !
String: yxXmMT has collision value 1853942169 !
String: sBGVfmH3ZUQ2nE7wsidKmJN5Xq5 has collision value 1853942169 !
String: KXnmnD0sj8EwNCUc1EoyteA0WxTnameD1x has collision value 1853942169 !
String: 3Um5biY8UYni78sNAy005gdlsyxqKaKapwDv06YXdgVm5v5Nq has collision value 1853942169 !
String: 82UTAUl4wGhcX3xYvZ1KnKVeZTFSFyAZOTn930JuFtClo8459gM has collision value 1853942169 !
String: OMiCFBOtCB84EQxl has collision value 1853942169 !
```

Results after hashF1 is modified.

BlueJ: Terminal Window - Comp

Options

```
input = Search125 : Hash =  =9*;0ijm
Start searching for collisions
No collision found
```

Can only enter input while your programming is running

BlueJ: Terminal Window - Comp

Options

```
input = hello : Hash = 0=447
Start searching for collisions
No collision found
```

Can only enter input while your programming is running