☆ Array Burst problem

Given an String array and burst length as input, the output should be an array which is a shrunk input array such that the sequentially repeating elements more than or equal to the burst length should be removed. This has to be repeated till the array cannot be shrunk any further. Use single characters in the string Array for simplicity as shown in sample test cases.

Sample Case 1:

input : 8
a
b
c
c
c
d
e
e
3
output :
a
b
d
e
e

Sample Case 2:

input :
15
a
b
c
d
e
e
e
e
d
d
c
b
f
g
f,
3
output :
a
b
c
c
b
f
g
f

YOUR ANSWER

Goldman Sachs Online Assessment - Sep 1 | 38m : 19s to test end | 0/2 Attempted | Toran Sahu

Original code | Java 7

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {


    /*
     * Complete the function below.
     */
    static String[] getShrunkArray(String[] inputArray, int burstLength) {
        int count = 0;
        //int MAX_INTEGER = 200;
CASE_INSENSITIVE_ORDER



        for( int i=0; i< inputArray.length; i++){
            repeatedArray.add(inputArray[i]);
        }


        for( int i=0; i< inputArray.length; i++){
            for( int j=i+1; j< inputArray.length; j++){
                if (inputArray[i].equals(inputArray[j])){
                    count++;

                }}
            if (count>=3){
                repeatedArray.remove(inputArray[i]);
            }
        }

        Integer res_size = repeatedArray.size();
        String[] res = new String[res_size];
        Integer idx=0;
        for (String e:repeatedArray){
                //System.out.println(e);
                res[idx]=e;
                idx++;

            }
        //System.out.println(repeatedArray);
    return res;
    }


    public static void main(String[] args) throws IOException {
        Scanner in = new Scanner(System.in);
        final String fileName = System.getenv("OUTPUT_PATH");
        BufferedWriter bw = null;
        if (fileName != null) {
            bw = new BufferedWriter(new FileWriter(fileName));
        }
        else {
            bw = new BufferedWriter(new OutputStreamWriter(System.out));
        }
```

```
        inputArray_size = Integer.parseInt(in.nextLine().trim());

        String[] inputArray = new String[inputArray_size];
        for(int i = 0; i < inputArray_size; i++) {
            String inputArray_item;
            try {
                inputArray_item = in.nextLine();
            } catch (Exception e) {
                inputArray_item = null;
            }
            inputArray[i] = inputArray_item;
        }

        int burstLength;
        burstLength = Integer.parseInt(in.nextLine().trim());

        res = getShrunkArray(inputArray, burstLength);
        for(int res_i = 0; res_i < res.length; res_i++) {
            bw.write(String.valueOf(res[res_i]));
            bw.newLine();
        }

        bw.close();
    }
}

```

Line: 66 Col: 21

Test against custom input

Run Code | Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*