Sanith shetty 4nm20is123   HACKERRANK PROBLEMS

1.

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

    /*
     * Complete the 'rotateLeft' function below.
     *
     * The function is expected to return an INTEGER_ARRAY.
     * The function accepts following parameters:
     *  1. INTEGER d
     *  2. INTEGER_ARRAY arr
     */

    public static List<Integer> rotateLeft(int d, List<Integer> a
rr) {
    // Write your code here
    int n=d%arr.size();
    List<Integer> o=new ArrayList<Integer>();
    for(int i=n;i<arr.size();i++){
        o.add(arr.get(i));
    }
    for(int i=0;i<n;i++)o.add(arr.get(i));
    return o;

    }


}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new In
putStreamReader(System.in));
```

```
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT_PATH")));

        String[] firstMultipleInput = bufferedReader.readLine().replaceAll("\\s+$", "").split(" ");

        int n = Integer.parseInt(firstMultipleInput[0]);

        int d = Integer.parseInt(firstMultipleInput[1]);

        List<Integer> arr = Stream.of(bufferedReader.readLine().replaceAll("\\s+$", "").split(" "))
            .map(Integer::parseInt)
            .collect(toList());

        List<Integer> result = Result.rotateLeft(d, arr);

        bufferedWriter.write(
            result.stream()
                .map(Object::toString)
                .collect(joining(" "))
            + "\n"
        );

        bufferedReader.close();
        bufferedWriter.close();
    }
}
```

2.

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

    /*
     * Complete the 'camelcase' function below.
     *
     * The function is expected to return an INTEGER.
     * The function accepts STRING s as parameter.
     */

    public static int camelcase(String s) {
    // Write your code here
    int count=1;
    for(int i=0;i<s.length();i++){
        if(Character.isUpperCase(s.charAt(i))){
            count++;

        }
    }
    return count;

    }

}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new In
putStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new Fi
leWriter(System.getenv("OUTPUT_PATH")));

        String s = bufferedReader.readLine();

        int result = Result.camelcase(s);

        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();

        bufferedReader.close();
        bufferedWriter.close();
    }
}
```

3.

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

    /*
     * Complete the 'staircase' function below.
     *
     * The function accepts INTEGER n as parameter.
     */

    public static void staircase(int n) {
    // Write your code here
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if(j>=n-i-1)
            System.out.print("#");
            else
            System.out.print(" ");
    }
    System.out.println();

    }

}
}
public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new In
putStreamReader(System.in));
```

```
        int n = Integer.parseInt(bufferedReader.readLine().trim()
);

        Result.staircase(n);

        bufferedReader.close();
    }
}
```