Censorship

Mr. Panda wants to read a short story to his kids. However, he realises that the story contains several bad words that he wants to censor from his kids. Can you help Mr. Panda do so?

Input
The input contains 3 lines.
The first line contains a single integer **N**, the number of bad words that Mr. Panda wants to censor.

The second line contains **N** words that Mr. Panda wants to censor. All the words will only contain English letters and there will be **exactly 1 space** in between these words.

The third line contains the story that Mr. Panda wants to censor. They will consist of a sequence of words that only contain English letters. There will also be **exactly 1 space** in between words.

Output
Print the entire story with the bad words censored.
The censored words should be replaced with '*' characters, with the number of characters equal to the length of the word.
You only need to censor entire words, so for example if "bad" is a bad word, you should not censor "Sinbad".
However, you must censor words regardless of case so for example if "bad" is a bad word, you must censor "BaD".

Limits
- $1 \leq \mathbf{N} \leq 10$
- Each censored word will not be more than 30 characters long.
- The story will contain between 1 and 10000 characters.
- All the words will only contain English letters and there will be **exactly 1 space** in between words.

Sample Input (**censorship1.in**)	Sample Output (**censorship1.out**)
4 code debug cat panda Mr Panda and Rar the Cat like to code	Mr ***** and Rar the *** like to ****
Sample Input (**censorship2.in**)	Sample Output (**censorship2.out**)
4 code debug cat panda but they do not like debugging	but they do not like debugging

Notes:

1. You should develop your program in the subdirectory `ex1` and use the skeleton java file provided. You should not create a new file or rename the file provided.
2. You are free to define your own helper methods and classes (or remove existing ones) if it is suitable.
3. Please be reminded that the marking scheme is:
   a. Public Test Cases (1%)
      i. 1% for passing **all** test cases, 0% otherwise
   b. Hidden Test Cases (1%)
      i. Partial scoring depending on test cases passed
   c. Manual Grading (1%)
      i. Overall Correctness (correctness of algorithm, severity of bugs)
      ii. Coding Style (meaningful comments, modularity, proper indentation, meaningful method and variable names)

Skeleton File – Censorship.java

You are given the skeleton file `Censorship.java`. You should see a non-empty file when you open the skeleton file. Otherwise, you might be in the wrong working directory.
You should see the following contents when you open the skeleton file:

```
/**
 * Name         :
 * Matric. No   :
 * PLab Acct.   :
 */

import java.util.*;

public class Censorship {
    private void run() {
        //implement your "main" method here
    }

    public static void main(String[] args) {
        Censorship newCensorship = new Censorship();
        newCensorship.run();
    }
}
```