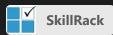


▼ LeaderBoard & Prev Day Solution

**DAILY CHALLENGE** 

ProgramID- 6048



**String Triangle Pattern** 

Accept an integer  $\mathbf{N}$  and a string  $\mathbf{S}$  as input. The program must print the last  $\mathbf{N}$  characters of each word of  $\mathbf{S}$  in the desired pattern as shown in the Example Input/Output section.

#### Note:

Each word in **S** contains more than **N** characters.

If there is no word to fill the pattern then use \*.

## **Boundary Condition(s):**

2 <= N <= 10

Length of the string **S** is between 1 and 1000.

## **Input Format:**

The first line contains the integer  ${\bf N}$ .

The second line contains the string **S**.

## **Output Format:**

The list of lines contain the last  ${\bf N}$  characters in the desired pattern.

# **Example Input/Output 1:**

Input:

2

apple mango orange

Output:

ple

ngo nge

#### **Example Input/Output 2:**

Input:

2

ask happy word letter king

```
Output:
sk
py rd
er ng **
```

#### **Explanation:**

The last 2 characters of the first word in S is "sk".

The first line of the output must contain one word. So, "sk" is printed.

The last 2 characters of the second and third word are "py" and "rd".

The second line of the output must contain two words separated by space(s). So, "py" and "rd" printed.

The last 2 characters of the fourth and fifth word are "er" and "ng".

The third line of the output must contain three words separated by space(s). But here, there are two words only. So, "er" and "ng" is printed. Then, for the third word "\*\*" is printed.

**Max Execution Time Limit: 5000 millisecs** 



```
import java.util.*;
public class Hello {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int spaces = sc.nextInt();
        int linesPerWords = 1;
        while(sc.hasNext()){
            for(int i=0;i<linesPerWords;i++){</pre>
                 String word = sc.hasNext() ? sc.next() : "*".repeat(space
                System.out.print(word.substring(word.length()-spaces)+" "
            }
            System.out.println();
            linesPerWords++;
        }
        System.out.println(123);
    }
```

