Question 12 Max. Marks 100.00

Longest Increasing Subsequence

You are given a string S of length N. Now, you are given Q queries. Each query is one of the following types:
$1\ X\ Y$ Change the character at position X to Y.
2 Find the length of the longest increasing subsequence of the string S.

Notes

1. In the longest increasing subsequence, the characters are in strictly increasing order.
2. A subsequence of a string not necessarily contains contiguous characters. For example, the string "**ale**" is a subsequence of string "**apple**".
3. Read the constraints carefully.

Input Format

The first line contains an integer N as input denoting the length of the string S.
The next line contains a string S of lowercase English alphabet characters. The length of this string is N.
The next line contains an integer Q as input denoting the total number of the queries.
Each query is one of the two types as described in the problem statement above.

Output Format

For each query of type 2, you need to print the answer in a new line.

Constraints

$1 \leq L \leq R \leq N \leq 10^5$
$1 \leq Q \leq 10^5$
The string S will contain at most 8 distinct characters at every point of time. Each character belongs to $[a, b, c, d, e, f, g, h]$.

Sample Input

```
7
aabccae
3
2
1 6 d
2
```

Sample Output

```
4
5
```

Explanation

In the sample for the first query of type 2 the length of the longest increasing subsequence of characters is 4 **abce**.

After the string is updated the length of the longest increasing subsequence of characters is 5 **abcde**.

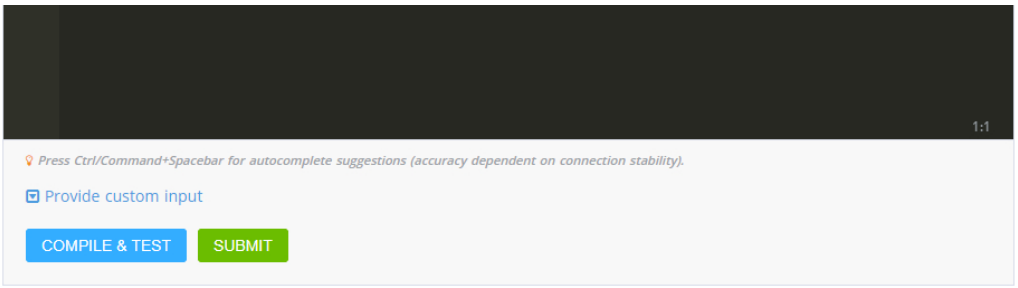
Note: Your code should be able to convert the sample input into the sample output. However, this is not enough to pass the challenge, because the code will be run on multiple test cases. Therefore, your code must solve this problem statement.

Time Limit: 1.0 sec(s) for each input file
Memory Limit: 256 MB
Source Limit: 1024 KB
Marking Scheme: Marks are awarded if any testcase passes
Allowed Languages: Java, Java 8

New Submission | All Submissions

Java (openjdk 1.7.0_95) Save

```java
import java.io.BufferedReader;
import java.io.InputStreamReader;

class TestClass {
    public static void main(String args[] ) throws Exception {

        //BufferedReader
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String name = br.readLine();                // Reading input from STDIN
        System.out.println("Hi, " + name + ".");    // Writing output to STDOUT



    }
}

```



Press Ctrl/Command+Spacebar for autocomplete suggestions (accuracy dependent on connection stability).

Provide custom input

COMPILE & TEST SUBMIT