Kick Start 2021 - Round B

Increasing Substring

Problem

Your friend John just came back from vacation, and he would like to share with you a new property that he learned about strings.

John learned that a string C of length L consisting of uppercase English characters is strictly increasing if, for every pair of indices i and j such that $1 \leq i < j \leq L$ (1-based), the character at position i is smaller than the character at position j.

For example, the strings `ABC` and `ADF` are strictly increasing, however the strings `ACC` and `FDA` are not.

Now that he taught you this new exciting property, John decided to challenge you: given a string $\mathbf{S}$ of length $\mathbf{N}$, you have to find out, for every position $1 \leq i \leq \mathbf{N}$, what is the length of the longest strictly increasing substring that ends at position i.

Input

The first line of the input gives the number of test cases, $\mathbf{T}$. $\mathbf{T}$ test cases follow.

Each test case consists of two lines.

The first line contains an integer $\mathbf{N}$, representing the length of the string.

The second line contains a string $\mathbf{S}$ of length $\mathbf{N}$, consisting of uppercase English characters.

Output

For each test case, output one line containing `Case #`x`:` y_1 y_2 `...` y_n, where x is the test case number (starting from 1) and y_i is the length of the longest strictly increasing substring that ends at position i.

Limits

Memory limit: 1 GB.
$1 \leq \mathbf{T} \leq 100$.

Test Set 1

Time limit: 20 seconds.
$1 \leq \mathbf{N} \leq 100$.

Test Set 2

Time limit: 40 seconds.
$1 \leq \mathbf{N} \leq 2 \times 10^5$.

Sample

Sample Input

```
2
4
ABBC
6
ABACDA
```

Sample Output

```
Case #1: 1 2 1 2
Case #2: 1 2 1 2 3 1
```

In Sample Case #1, the longest strictly increasing substring ending at position 1 is `A`. The longest strictly increasing substrings ending at positions 2, 3 and 4 are `AB`, `B` and `BC`, respectively.

In Sample Case #2, the longest strictly increasing substrings for each position are `A`, `AB`, `A`, `AC`, `ACD` and `A`.