# Inc Match Two

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 0.25 seconds |
| Memory limit: | 4 megabytes |

You are given a sequence $s$ of lowercase Latin letters, on which you can perform the following operation as many times as you please:

- Choose any occurrence of a letter which is not '`z`' and replace it with the subsequent letter of the alphabet*; then, as long as there are two or more adjacent identical characters, delete them.

Determine the minimum number of operations required to make the string empty.

## Input

The first line of the input contains a single integer, $t$ ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains a single integer, $n$ ($1 \le n \le 2 \cdot 10^5$) — the length of the string.

The second line of each test case contains the string, $s$, consisting of lowercase Latin letters. It is guaranteed that no two adjacent characters in the string are equal.

It is guaranteed that the sum of $n$ over all test cases does not exceed $2 \cdot 10^5$.

## Output

For each test case, output a single integer — the minimum number of operations required to make the string empty, or $-1$ if it's impossible.

## Example

| standard input | standard output |
|---|---|
| 7<br>1<br>a<br>2<br>ab<br>3<br>abc<br>4<br>abac<br>5<br>abcab<br>6<br>abcbad<br>5<br>azabc | NO<br>YES<br>NO<br>YES<br>YES<br>YES<br>NO |

## Note

In the first test case, the string cannot become empty.

In the second test case, the first character can be replaced with '`b`' to match its neighbor.

In the third test case, it can be proven that one character will always remain in the end, regardless of the operations performed.

*For example, '`a`' becomes '`b`', '`b`' becomes '`c`', etc.