Inc Match Two (Easy)

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

This is the easy version of the problem. In this version, you are asked to answer Yes or No. You can only make hacks if both versions of the problem are solved.

Su is eager to play a game with her beloved students. She will give them a sequence $s_1, s_2, ..., s_n$ of lowercase Latin letters and ask them to perform operations on their copy of the string, according to the following rules:

- A student can choose any **occurrence** of a letter which is not '`z`' and replace it with the subsequent letter of the alphabet;
- After each operation, while two or more **adjacent** characters are equal, they are immediately deleted from the string;
- Whoever makes their string empty with the fewest operations wins; if it's not possible, no one wins.

In anticipation of the game, she wants to know if there is a winning strategy for the string.

Input

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

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

The second line of each test case contains a string, s, consisting of lowercase Latin letters.

Output

For each test case, output a single string — "`YES`" if it is possible to make the string empty according to the game rules, or "`NO`" if it's impossible.

You can output the letters in any case. For example, the strings "`yEs`", "`yes`", and "`Yes`" will also be recognized as positive responses.

Example

standard input	standard output
7 1 a 2 ab 3 abc 4 abac 5 abcab 6 abcbad 5 azabc	NO YES NO YES YES YES 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.