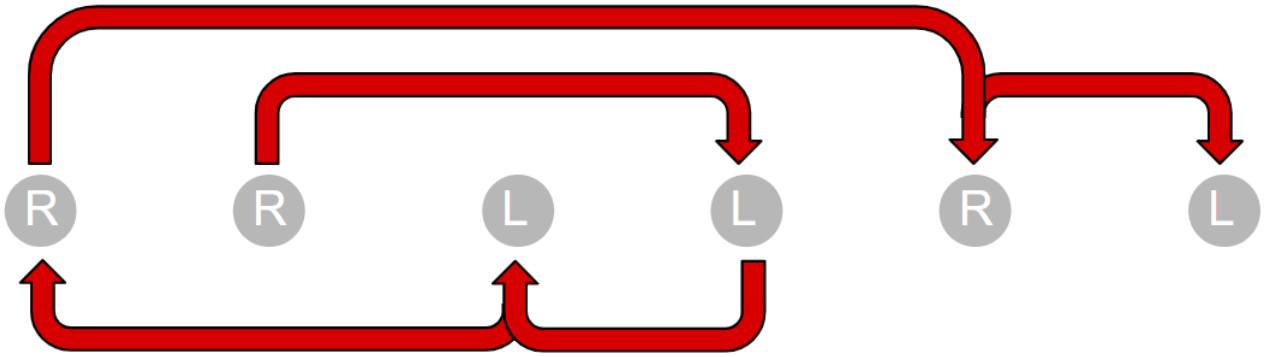
Stone Jump

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There are n stones in a row, each marked with L or R. You can jump from an L stone to any stone to its left, and from an R stone to any stone to its right.

More formally, for a stone at position $1 \le i \le n$, if it is marked with L, you can jump to any stone j where $1 \le j < i$, and if it is marked with R, you can jump to any stone j where $i < j \le n$.

Your goal is to find a sequence of jumps such that you visit every stone exactly once. If you can start at any stone of your choice, is this possible?



One example of a valid path visiting each stone once, starting from the second stone

Input

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

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

The next line of each test case contains a string s consisting of n characters `L` and `R`, indicating whether stone i is marked with L or R.

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

Output

For each test case print "YES" (without quotes) if it is possible, and "NO" (without quotes) otherwise.

You may print every letter in any case you want (so, for example, the strings "yEs", "yes", "Yes" and "YES" will all be recognized as positive answers).

Example

standard input	standard output
7 6 LRRLRL 1 L 2 RL 2 LR 5 RRRRL 10 LRLRLRLRLR 31 LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL	YES YES YES NO YES NO YES

Note

In the second sample case, $n = 1$, so you can start on the first stone, at which point you have visited all stones and completed your path.

In the fourth sample case, it can be shown that no valid path exists.