CODEFORCES
Sponsored by TON

HOME   TOP   CATALOG   CONTESTS   GYM   PROBLEMSET   GROUPS   RATING   EDU   API   CALENDAR   HELP

PROBLEMS   SUBMIT CODE   MY SUBMISSIONS   STATUS   HACKS   STANDINGS   CUSTOM INVOCATION

C. Arrow Path

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There is a grid, consisting of 2 rows and n columns. The rows are numbered from 1 to 2 from top to bottom. The columns are numbered from 1 to n from left to right. Each cell of the grid contains an arrow pointing either to the left or to the right. No arrow points outside the grid.

There is a robot that starts in a cell $(1, 1)$. Every second, the following two actions happen one after another:

1. Firstly, the robot moves left, right, down or up (**it can't try to go outside the grid, and can't skip a move**);
2. then it moves along the arrow that is placed in the current cell (the cell it ends up after its move).

Your task is to determine whether the robot can reach the cell $(2, n)$.

Input

The first line 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 ($2 \le n \le 2 \cdot 10^5$).

The second line contains a string consisting of exactly n characters < and/or > — the first row of the grid.

The third line contains a string consisting of exactly n characters < and/or > — the second row of the grid.

Additional constraints on the input:

- n is even;
- there are no arrows pointing outside the grid;
- the sum of n over all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case, print YES if the robot can reach the cell $(2, n)$; otherwise, print NO.

You can print each letter in any case. For example, yes, Yes, YeS will all be recognized as positive answer.
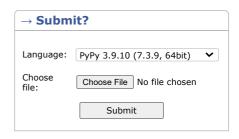
Example

input   Copy

```
4
4
>><<
>>><
2
><
><
4
>>><
>><<
6
>><<><
><>><
```

output   Copy

```
YES
YES
```

→ **Virtual participation**   >

→ **Clone Contest to Mashup**   ⌄

You can clone this contest to a mashup.

[Clone Contest]

→ **Submit?**

Language: PyPy 3.9.10 (7.3.9, 64bit) ⌄

Choose file: [Choose File] No file chosen

[Submit]

→ **Last submissions**

Submission	Time	Verdict
251529749	Mar/15/2024 19:21	Accepted

→ **Problem tags**

brute force   constructive algorithms
dfs and similar   dp   graphs
shortest paths

No tag edit access

→ **Contest materials**

- Announcement   ✕
- Tutorial   ✕

NO
YES

Note

In the first example, one of the possible paths looks as follows:
$(1,1) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (2,4)$.

In the second example, one of the possible paths looks as follows:
$(1,1) \rightarrow (2,1) \rightarrow (2,2)$.

In the third example, there is no way to reach the cell $(2,4)$.

In the fourth example, one of the possible paths looks as follows:
$(1,1) \rightarrow (2,1) \rightarrow (2,2) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (2,4) \rightarrow (2,5) \rightarrow (2,6)$.

Supported by

ITMO UNIVERSITY