P=NP

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

Busy Beaver has a string A made up of characters `P` and `N`. He can perform two types of operations on A:

- Pick a substring* `P` and replace it with `NP`.
- Pick a substring `NP` and replace it with `P`.

Busy Beaver has a target string B. Determine if he can turn A into B after some number of operations.

Input

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

The only line of each test case contains the strings A and B $(1 \leq |A|, |B| \leq 10^5)$, consisting of characters `P` and `N`.

The sum of $|A| + |B|$ across all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output `YES` if Busy Beaver can turn A into B using the operations above, and `NO` otherwise.

You can output the answer in any case (upper or lower). For example, the strings "`yEs`", "`yes`", "`Yes`", and "`YES`" will be recognized as positive responses.

Example

standard input	standard output
7	YES
P NP	YES
PNPN NPPN	NO
PP NP	NO
NPN PPNP	YES
PNPP PPNNNNNNNNNNNNNNNNNNP	NO
PPNNPPNNPP NNPPNNPPNN	NO
NPNNNNNPN PPPN	

Note

In the first test case, we can perform one operation to turn `P` into `NP`.

In the second test case, we can do two operations: `PNPN` → `PPN` and then `PPN` → `NPPN`.

In the third test case, we can show that it is not possible to turn `PP` into `NP` using any number of operations.

*A substring of length k is a contiguous sequence of k adjacent characters of a string.