A. Equal or Not Equal

time limit per test: 2 seconds
memory limit per test: 256 megabytes

You had n positive integers $a_1, a_2, \ldots, a_n$ arranged *in a circle*. For each pair of neighboring numbers (a_1 and a_2, a_2 and a_3, ..., a_{n-1} and a_n, and a_n and a_1), you wrote down: are the numbers in the pair equal or not.

Unfortunately, you've lost a piece of paper with the array a. Moreover, you are afraid that even information about equality of neighboring elements may be inconsistent. So, you are wondering: is there any array a which is consistent with information you have about equality or non-equality of corresponding pairs?

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of test cases. Next t cases follow.

The first and only line of each test case contains a non-empty string s consisting of characters E and/or N. The length of s is equal to the size of array n and $2 \le n \le 50$. For each i from 1 to n:

- if $s_i =$ E then a_i is equal to a_{i+1} ($a_n = a_1$ for $i = n$);
- if $s_i =$ N then a_i is not equal to a_{i+1} ($a_n \neq a_1$ for $i = n$).

Output

For each test case, print YES if it's possible to choose array a that are consistent with information from s you know. Otherwise, print NO.

It can be proved, that if there exists some array a, then there exists an array a of positive integers with values less or equal to 10^9.

Example

input Copy

```
4
EEE
EN
ENNEENE
NENN
```

output Copy

```
YES
NO
YES
YES
```

Note

In the first test case, you can choose, for example, $a_1 = a_2 = a_3 = 5$.

In the second test case, there is no array a, since, according to s_1, a_1 is equal to a_2, but, according to s_2, a_2 is not equal to a_1.

In the third test case, you can, for example, choose array $a = [20, 20, 4, 50, 50, 50, 20]$.

In the fourth test case, you can, for example, choose $a = [1, 3, 3, 7]$.