# Verbum Profundum

## Verbum Profundum

On a regular day, Emre and Alp are walking around the Golet (one of the greatest beauties of ITU) as Emre talks about his super power to Alp: his word memory. He knows more words than average. But how he can do that? With a simple technique, he can memorize words much easier than usual: If the reverse of the first half (including median at words with odd length) of any word can be found as a suffix of any word in his memory, then he can remember a whole set of words much more easly.

Emre defines the number of words that hold this property as **Verbum Profundum**. After this conversation Alp gets curious about his own **Verbum Profundum** and he tries calculating it as he continues his morning walk. Can you help him find his **Verbum Profundum**?

Submit Solution

**Points:** 1

**Time limit:** 2.0s
Java: 4.0s
Java 8: 4.0s
Javascript v8: 6.0s
Mono C#: 4.0s
Python: 6.0s

All submissions

Best submissions

My submissions

### Input Format

First line consists of one integer $n$ which is the number of words Alp knows.

Next $n$ lines consist of one string each, $s$ , which consists of lowercase English letters.

### Constraints

$1 \leq n$, $(length\ of\ s)$

$(max\ length\ of\ s) * n \leq 10^6$

### Output Format

Print one integer denoting the **Verbum Profundum** of Alp.

#### Sample Input

```
3
kmnbbb
bbcnmk
bbbxx
```

Copy

#### Sample Output

```
2
```

Copy

#### Sample Explanation

kmnbbb and bbbxx satisfy the rule.

Request clarification