THICC '17 P4 - Kenta's Neural Networks

Time Limit: 1.0s
Python: 10.0s
Memory Limit: 256M

Kenta has a lot of neural networks which are trained to output a specific string. He has a copious amount of neural networks, as he is very good at building neural networks and loves them very much. Unfortunately, he has so many that he needs help finding out which of his neural networks is the best!

Each output for the neural network has a value that represents how accurate his neural network is. The accuracy value represents how many changes the neural network needs to make to the string to get the correct string. A change can be adding a character, removing one or editing one. The smaller the accuracy value, the better the neural network.

For example, to go from "horse" to "hose" you'd need one change, and from "almond" to "nut" you'd need six. You always want to use the minimum number of changes.

You'll be given the correct string which all neural networks wish to achieve, n number of neural networks, each accompanied by t different outputs from that same neural network. You'll need to output the results of the best (most accurate) neural network.

Input Specification

The first line will contain the correct string containing only lowercase letters.

The second line will contain n, the number of neural networks.

The following n lines will start with t, followed by t space separated strings, which are the neural networks' outputs.

Output Specification

A real number, the average accuracy value of the best neural network. An absolute or relative error of up to 10^{-6} will be accepted.

Constraints

Subtask 1 [20%]:

$1 \leq n, t \leq 10$

$1 \leq$ Length of any String ≤ 10

Subtask 2 [30%]:

$1 \leq n, t \leq 50$

$1 \leq$ Length of any String ≤ 50

Subtask 3[50%]:

$1 \leq n, t \leq 100$

$1 \leq$ Length of any String ≤ 100

Sample Input

```
memes
3
2 memes meemes
2 memes same
3 netuhiuoeou eunhouenht oentonhtet
```

Sample Output

```
0.5
```

Explanation

The first neural network has one perfect output. `meemes` needs the `e` removed to make `memes`. $(0+1)/2 = 0.5$.

The second neural network has a perfect output, and `same`, which would take 3 moves to turn into `memes`.

The third is a very bad neural network and need not be considered.