C Compressing Commands



Only true terminal-dwellers use green on black colours. CC BY-NC 2.0 by Kjetil Korslien on Flickr

Unlike your friends, you live in a terminal. You are cool. Your terminal is everything to you: you have optimized the font, colour scheme, keyboard shortcuts, and what not.

Once thing still annoys you though: all these commands you're typing involve so many file paths and are so long! Then it occurs to you: all this time you have been working from the *root directory* of the file system, but in fact you can *change directory* to anywhere you like! This should simplify your life a lot!

This way, if your *working directory* is `/a/b`, you can refer to the absolute file path `/a/b/x` using simply `x`. To *go up* a level, you can use `..`, so that you can refer to `/a/y/z` as `../y/z`, and to `/some/other/directory` as `../../some/other/directory`.

You being you, of course you overdo this and will now use relative paths *everywhere*!

Given the n absolute file paths in the command you want to run, find the working directory that minimizes the total number of relative path components. For example, `a/a/b/c`, and `../../a/b` both contain 4 path components. Note that you can only change the working directory to a directory and not to a file path. Filenames will never coincide with directory names in the same directory.

In the first sample it is best to set the working directory to `/home/jury/compressingcommands`, leading to 6 components: `secret`, `solutions`, and `../../hackerman/answers`.

In the second sample it is best to set the working directory to `/a`, leading to 19 path components: `b/a/a/b`, `a/a`, `../b`, `a/b`, `b/a/a/a`, `../c`, and `b/a/b`.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 10^5$), the number of absolute file paths.
- n lines, each with a string s, an absolute file path. Each path contains only lowercase English letters (`a-z`) and slashes ('`/`'), starts with '`/`', does not end with '`/`', and does not contain consecutive slashes ("`//`").

The total number of characters in the n strings is at most 10^6.

Output

Output the minimal number of relative path components that can be achieved by changing to a different working directory.

Sample Input 1

```
3
/home/jury/compressingcommands/secret
/home/jury/compressingcommands/solutions
/home/hackerman/answers
```

Sample Output 1

```
6
```

Sample Input 2

```
7
/a/b/a/a/b
/a/a/a
/b
/a/a/b
/a/b/a/a/a
/c
/a/b/a/b
```

Sample Output 2

```
19
```

Sample Input 3

```
3
/x/y/z
/x/y/z
/x/y/z
```

Sample Output 3

```
3
```