Browser Games

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

In the upcoming n days, n browser games will be released on a new website. According to the plan, the administrator will release a new game per day. Users have to open the corresponding URL (Uniform Resource Locator) and get feedback from the server to download a game.

However, the setup of the server uses unreadable legacy codes. Once a user somehow finds the URL of an unreleased game, the data of the game would leak out. To temporarily fix the problem, the administrator decided to add a series of confirmation prefixes, which are non-empty strings, at the server-side. The server will respond with the correct game data when the requested URL does correspond to a game (no matter released or unreleased) and at least one confirmation prefix is a prefix of the URL; otherwise, the server will declare that the game is not found.

To make the work easier, the administrator asks you to find the minimum number of confirmation prefixes the server required to avoid data leaks every time after a new game release.

Input

The first line contains an integer n ($1 \leq n \leq 5 \times 10^4$), indicating the number of browser games to be released.

In the next n lines, the i-th line contains a non-empty string, consisting of only lowercase letters ('a' to 'z'), dots ('.') and forward slashes ('/'), indicating the URL of the browser game released on the i-th day.

It is guaranteed that the length of each given URL is at most 50, and no given URL is the prefix of any other given URL.

Output

Output n lines, the i-th of which contains an integer indicating the minimum number of required confirmation prefixes after the i-th new game released.

Example

standard input	standard output
3 ufoipv.ofu hsbocmvfgboubtz.kq hfotijo.njipzp.dpn/kb	1 2 2