Problem K
Space Alignment
Time Limit: 1 Second, Memory Limit: 2G

You are collaborating with a few other programmers on a coding project. To your horror, you discover that some people have been using tabs to indent code while other people have been using spaces to indent code within the same file. Apparently these people didn't notice that anything was amiss when they did so.

You wish to replace the tabs with spaces so that the file has consistent indenting. Consistent indenting means that for every line at a nesting depth of k, $k \geq 0$, the first non-whitespace character in the line should be preceded by exactly $k \cdot i$ spaces, where $i > 0$.

Is it possible to replace each tabs with a fixed number of spaces to repair the file and have consistent indenting?

Input

The first line of input contains a single integer n ($2 \leq n \leq 100$), which is the number of lines of code.

Each of the next n lines contains a string, consisting of a sequence of the characters '**s**' (representing a space) and/or '**t**' (representing a tab), followed by a single '**{**' or '**}**'. Each line will have at most $1{,}000$ characters.

The first line is guaranteed to be a single '**{**', the last line is guaranteed to be a single '**}**', and the braces throughout the data are guaranteed to match; that is, looking at only those characters as a single string, it is always possible to repeatedly remove a substring "**{}**" until you are left with the empty string.

Output

Output a single integer, which is the smallest number of spaces greater than zero to replace each tab with and achieve consistent indentation, or -1 if it isn't possible.

Sample Input 1	Sample Output 1

10
{
ss{
sts{
tt}
t}
t{
ss}
}
{
}
	2