Pattern Compression

SEASON 10 – FINAL

The Bulgarian language and literature exam is over, the students' grades are already known and Ms. Lalka Dencheva is relieved that she will not write (many) explanations because of certain individuals who have failed miserably. However, after a detailed analysis of the results, she found that writing their own argumentative texts was quite a challenge for many of her students. Until now, she had secretly suspected that it was not possible for the entire class she taught to create such masterpieces of the interpretive genre as she had read day and night all year. Now her suspicions were confirmed, and she decided she could no longer allow her students to deceive her.

Ms. Lalka Dencheva copied one of the essays received during the e-learning, pasted the text into the Internet search engine and … received error 414. After numerous attempts to solve this problem, including the purchase of a new laptop, the teacher asked one computer-literate student in her class who can be trusted. He was happy to explain that the search engine could not process such long search queries and suggested that his teacher use only part of the essay as a pattern. However, Ms. Dencheva did not like this option and the student started looking for another solution.

For the purpose of this problem, we can assume that the text of the essay consists of N lowercase Latin letters. You need to find a suitable pattern to use to search the entire text in the search engine. Fortunately, the search engine provides several options for abbreviating the pattern:

1) A valid pattern is any sequence of lowercase Latin letters
   - ❖ For example: "`aab`" is a pattern for searching the text "`aab`"

2) If P is a valid pattern for the text T, then $P?(l_1, l_2, …, l_k)$ is a valid pattern for searching the results of the concatenation of T and each of the letters $l_1, l_2, …, l_k$
   - ❖ For example: "`aab?(c,d)`" is a pattern for searching of the texts "`aabc`" and "`aabd`"

3) If P is a valid pattern for searching the all the texts in the set S, then $[P, k]$ is a valid pattern for searching the results of the concatenation of the (optionally not different) texts $T_1, T_2, …, T_k$, each of which belongs to S, where k greater than 1
   - ❖ For example: "`[aab?(c,d),2]`" is a pattern for searching the texts "`aabcaabc`", "`aabcaabd`" and "`aabdaabd`"

4) If P is a valid pattern for searching the text T, then such is also the pattern obtained from the concatenation of: '#', '=', Q, '.' and P', where Q is a sequence of lowercase Latin letters and P' is obtained by replacing (part of) the appearances of Q as a substring with the symbol '#'. This type of abbreviation can be used only in the beginning of the pattern.
   - ❖ For example: "`#=aab.#c#d`" is a pattern for searching the text "`aabcaabd`"

Of course, there are many different patterns that can be used to search for specific text through a search engine. It is clear that the pattern should not be too long, but it should also be relatively accurate so that the text you are looking for can be found.

Let's define the following two functions:

- ❖ $length(P)$ – the number of symbols in P
- ❖ $uncertainty(T, P)$ – for every lowercase Latin letter in the text T, which is represented by a '?' in the pattern P, it is added the count of different letters, which this '?' can replace in the text

Your task is to find a valid pattern P for a given text T, so that the sum of the two functions $length(P) + uncertainty(T, P)$ is as small as possible.

Input
The first line of the input file `pattern.in` contains the string T, composed of N lowercase Latin letters.

Output
On the only line of the output file `pattern.out` print a string that is a valid pattern to search for the specified text. It should consist only of lowercase Latin letters, numbers and the symbols '[', ']', '(', ')', '.', ',', '#', '=' and '?'. The length of the pattern must not exceed N.

Constraints
$1 \leq N \leq 10^5$

Examples:

Input	Output
aaaaaaaaaab	[a,10]b
ababxababyababzababz	[[ab,2]?(x,y,z),4]
dingdangdongdangdingdongdang	[d?(i,a,o)ng,7]
dingdangdongdangdingdongdang	#=ngd.di#a#o#a#i#o#ang

Explanation:

In the first testcase $length(P) = 7$ and $uncertainty(T, P) = 0$, because in the patterns there are not question mark.

In the second testcase $length(P) = 18$ and $uncertainty(T, P) = 4 * 3 = 12$, because 4 of the letters in the text are represented by a '?', which can replace 3 different letters.

In the third testcase $length(P) = 15$ and $uncertainty(T, P) = 7 * 3 = 21$.

In the fourth testcase $length(P) = 22$ and $uncertainty(T, P) = 0$.

Scoring:

For each testcase let *minScore* is the minimal result among all participants' solutions and *yourScore* is your result, calculated as a sum of the values of the functions $length(P) + uncertainty(T, P)$. You will receive $\frac{minScore + 1}{yourScore + 1} \times 100$ % of the points for that testcase.

Subtasks

#	Number of tests	Constraint for N	Additional constraints
1	10%	$N \leq 100$	None
2	20%	$N \leq 1000$	T is a concatenation of strings with equal length of at most 4 letters. Among them there are at most 4 different strings.
3	30%	$N \leq 10000$	T consists of at most 4 different letters
4	40%	$N \leq 100000$	None