Let us define a regular brackets sequence in the following way:

1. Empty sequence is a regular sequence.
2. If S is a regular sequence, then (S) and [S] are both regular sequences.
3. If A and B are regular sequences, then AB is a regular sequence.

For example, all of the following sequences of characters are regular brackets sequences:

`()`, `[]`, `(())`, `([])`, `()[]`, `()[()]`

And all of the following character sequences are not:

`(`, `[`, `)`, `)(`, `([)]`, `([]`

Some sequence of characters '`(`', '`)`', '`[`', and '`]`' is given. You are to find the shortest possible regular brackets sequence, that contains the given character sequence as a subsequence. Here, a string $a_1a_2\dots a_n$ is called a subsequence of the string $b_1b_2\dots b_m$, if there exist such indices $1 \le i_1 < i_2 < \dots < i_n \le m$, that $a_j = b_{i_j}$ for all $1 \le j \le n$.

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The input file contains at most 100 brackets (characters '`(`', '`)`', '`[`' and '`]`') that are situated on a single line without any other characters among them.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

Write to the output file a single line that contains some regular brackets sequence that has the minimal possible length and contains the given sequence as a subsequence.

Sample Input

```
1

([(]
```

Sample Output

```
()[()]
```