1630 Folding

Bill is trying to compactly represent sequences of capital alphabetic characters from 'A' to 'Z' by folding repeating subsequences inside them. For example, one way to represent a sequence 'AAAAAAAAAABABABCCD' is '10(A)2(BA)B2(C)D'. He formally defines folded sequences of characters along with the unfolding transformation for them in the following way:

- A sequence that contains a single character from 'A' to 'Z' is considered to be a folded sequence. Unfolding of this sequence produces the same sequence of a single character itself.

- If S and Q are folded sequences, then SQ is also a folded sequence. If S unfolds to S' and Q unfolds to Q', then SQ unfolds to $S'Q'$.

- If S is a folded sequence, then $X(S)$ is also a folded sequence, where X is a decimal representation of an integer number greater than 1. If S unfolds to S', then $X(S)$ unfolds to S' repeated X times.

According to this definition it is easy to unfold any given folded sequence. However, Bill is much more interested in the reverse transformation. He wants to fold the given sequence in such a way that the resulting folded sequence contains the least possible number of characters.

Input

Input file contains several test cases, one per line. Each of them contains a single line of characters from 'A' to 'Z' with at least 1 and at most 100 characters.

Output

For each input case write a different output line. This must be a single line that contains the shortest possible folded sequence that unfolds to the sequence that is given in the input file. If there are many such sequences then write any one of them.

Sample Input

```
AAAAAAAAAABABABCCD
NEERCYESYESYESNEERCYESYESYES
```

Sample Output

```
9(A)3(AB)CCD
2(NEERC3(YES))
```