CEOI '16 P4 - Match

Time limit: 2.0s   **Memory limit:** 512M

We define a valid bracket sequence as a string that is either:

- The empty string;
- A string `(B)`, where `B` is a valid bracket sequence.
- `LR`, the concatenation of two strings `L` and `R` which are both valid bracket sequences.

Let B be a valid bracket sequence of length N. We define B_i to be the i-th character of sequence B. For two indices i and j, $1 \leq i < j \leq N$, we say that B_i and B_j are matching brackets if:

- $B_i = $ `(` and $B_j = $ `)`;
- $i = j - 1$, or the subsequence $C = B_{i+1}B_{i+2}\ldots B_{j-1}$ is a valid bracket sequence.

Let S be a string of lowercase English letters. We define S_i to be the i-th character of string S. We say that a valid bracket sequence B matches S if:

- B has the same length as S;
- for any pair of indices i and j, $i < j$, if B_i and B_j are matching brackets, then $S_i = S_j$.

For a given string S consisting of N lowercase letters, find the lexicographically smallest valid bracket sequence that matches S, or print `-1` if no such bracket sequence exists.

Input

The input contains a string S of N lowercase letters on the first line.

Output

In the output you should write either a string B with N characters that represents the lexicographically smallest bracket sequence that matches the input string, or `-1` if no such bracket sequence exists.

Notes and constraints

- $2 \leq N \leq 100\,000$
- For test cases worth 10 points $N \leq 18$.
- For test cases worth another 27 points $N \leq 2\,000$.
- We say that a bracket sequence A is lexicographically smaller than a bracket sequence B if there is an index $i, 1 \leq i \leq N$, such that $A_j = B_j$ for each $j < i$, and $A_i < B_i$.
- Character `(` is considered lexicographically smaller than character `)`.

Sample Input 1

```
abbaaa
```

Sample Output 1

```
(()())
```

Note for Sample Input 1

Another valid bracket sequence is `(())()`, but this is not the smallest lexicographic solution.

Sample Input 2

```
abab
```

Sample Output 2

```
-1
```

Note for Sample Input 2

There is no valid bracket sequence that matches the given string.