TLE '17 Contest 2 P1 - Cadadr

Time limit: 2.0s **Memory limit:** 256M

Some computer science courses at the University of Fireloo teach a programming language called Bracket.

Two of the functions in Bracket are `(car x)` and `(cdr x)`. These functions are used a lot, so the Bracket developers allowed programmers to "combine" multiple uses of `(car x)` and `(cdr x)` into one name.

Suppose that the function is `(cijk...r x)`, where `i`, `j`, `k`,... are characters each representing either `a` or `d`. This function is equivalent to `(cir (cjk...r x))`, which is equivalent to `(cir (cjr (ck...r x)))`, and so on. Note that the placement of the brackets is important. A full expansion only contains `car` and `cdr` functions.

Sample ending logic of a program in Bracket.

For example, `(cadadr x)` can be fully expanded to become `(car (cdr (car (cdr x))))`.

Given a function in the form `(cijk...r x)`, please output the full expansion.

Input Specification

The only line of input will contain a string in the form of `(cijk...r x)`. It will contain no more than $100\,000$ characters.

For 50% of the points, the string will contain no more than $1\,000$ characters.

Output Specification

Output a single line, the full expansion of the given function. Ensure that brackets are proper and that there is a space between the last `cdr` or `car` and the following `x`. Other spacing will not matter.

Sample Input 1

```
(cadadr x)
```

Sample Output 1

```
(car (cdr (car (cdr x))))
```

Sample Input 2

```
(cdadaddr x)
```

Sample Output 2

```
(cdr (car (cdr (car (cdr (cdr x))))))
```