Problem L – Lisp to Infix Translator.

Author: Ariel Ortiz Ramírez

Your good friend Progra María de la Codificación is taking an advanced undergraduate class calleda *Intriguing Computer Programming Curiosities* (ICPC) where she is learning Lisp (acronym for *List Processor*), a fairly ancient but highly respectable and intriguing programming language.

Lisp programs are built using a tree-structured nested list notation called *s-expressions* (symbolic expressions). For our purpose, an *s-expression* can be recursively defined as:

1. An *atom*, which is a single capital English letter from 'A' to 'Z'.
2. A *list* of the form '(op s1 s2 ...)' where 'op' is either the addition '+' or multiplication '*' operator, and 's1 s2 ...' are two or more *s-expressions* that serve as operands. A single space character is used to delimit the elements contained within the list (for example, to separate 'op' from 's1'). There are no spaces after an opening parenthesis '(' nor before a closing parenthesis ')'.

The following are examples of valid *s-expressions*:

- A
- (+ A B C)
- (* D D D D)
- (+ A B (* C (+ D (+ E F) G H) (* I J)) K L)

Given that '+' and '*' represent prefix operators, *s-expressions* can be converted into the ordinary infix notation that we typically use in other more conventional programming languages like C or Java. Thus, the previous examples are equivalent to:

- A
- A+B+C
- D*D*D*D
- A+B+C*(D+E+F+G+H)*I*J+K+L

As can be observed from the last example, we avoid using more parenthesis than those that are strictly required to preserve the semantics of the original *s-expression*. Remember that multiplication has a higher precedence than addition when using the ordinary infix notation.

Progra María is having trouble understanding programs written in Lisp. Because you want to help your friend, you have decided to write a program that automatically translates Lisp *s-expressions* into ordinary infix notation.

Input

The first line of the input contains an integer N that specifies the number of test cases that follow ($1 \leq N \leq 100$). Each of the following N lines contains a valid *s-expression* as previously described. The size of each individual line is no greater than 15000 characters. Also, lists can be deeply nested, but no more than 350 levels.

Output

For each test case, output in a line the translated expression in ordinary infix notation. Please note that the results should not contain spaces anywhere.

Sample input 1

```
4
A
(+ A B C)
(* D D D D)
(+ A B (* C (+ D (+ E F) G H) (* I J)) K L)
```

Sample output 1

```
A
A+B+C
D*D*D*D
A+B+C*(D+E+F+G+H)*I*J+K+L
```

Sample input 2

```
3
(* (+ N O P) (+ (+ (* (+ Q R) (+ S T)) U) (* (+ V W X) (+ Y Z))))
(+ T (* T (+ T (* T (+ T (* T (+ T T)))))))
(+ (* (+ (* (+ (* (+ T T) T) T) T) T) T) T)
```

Sample output 2

```
(N+O+P)*((Q+R)*(S+T)+U+(V+W+X)*(Y+Z))
T+T*(T+T*(T+T*(T+T)))
(((T+T)*T+T)*T+T)*T+T
```