

2168 Convert

The importance of postfix and prefix notation in parsing arithmetic expressions is that these notations are completely free of parentheses. Consequently, an expression in postfix (or prefix) form is crucial because having a unique form for an expression greatly simplifies its evaluation. But we humans prefer to read and evaluate an infix expression especially when it is parenthesized.

We would like to try the conversion from the postfix format to the parenthesized infix format for expression written in some functional language. The language will consist of the unary function 'INV' and the binary functions 'ADD' and 'MUL'.

Input

The input shall consist of several cases. Each case will be represented on a separate line. The number of cases will be given in the first line of the input.

Output

For each input case, the converted expression should be on a separate line, and should have a space after each comma.

Sample Input

```
2
2 -3 Add Inv 6 10 Mul Mul
100 1000 MUL 10 ADD INV
```

Sample Output

```
Mul ( Inv (Add (2, -3)), Mul (6, 10))
Inv ( Add (10, Mul (100, 1000)))
```