Using the following grammar:

$< expression > := < component > \mid < component >$ `+` $< expression >$
$< component > := < factor > \mid < factor >$ `*` $< component >$
$< factor > := < positiveinteger > \mid$ `(`$< expression >$`)`

write a program which analyses expressions conforming to the rules of this grammar and evaluates them, if the analysis has been successfully completed. It may be assumed that there is no overflow of float(C)/real(Pascal) numbers range.

Input

A integer n stating the number of expressions, then consecutive n lines, each containingan expression given as a character string.

Output

For each line value of the expression or output message '`ERROR`' if the expression does not follow the grammar.

Sample Input

```
5
32
12+34
1*(2+3)+3
1(2+3)+3
qwe323
```

Sample Output

```
32
46
8
ERROR
ERROR
```