5125   A Smart Brain is a Tasty Brain

The zombies have cornered you and your team. There's no hope...

But wait! Suddenly, they all stop advancing and offer you a deal instead. As it turns out, the zombies are having a problem finding good tasting brains, which, to a zombie, are the smart brains (there is nothing more disgusting to a zombie than the brains of a communications major). While your brains would be really delicious, the zombies realize that they could be put to better use by helping them to find other smart brains. The deal is this: help the zombies determine whether a given brain is smart (and therefore tasty) or not, and they will let your team go (at least for now).

Being the big-brained team that you are, you quickly discover that brains can be determined to be smart or not if the brain can return a correct answer to a given Boolean expression. You quickly compile a list of Boolean expressions together and set off testing one expression on each brain.

Every Boolean expression is deterministic and is recursively defined as follows

1. *expression* = '(' + (*value* or *expression*) + *operation* + (*value* or *expression*) + ')'

2. *value* = 't' or 'f'

3. *operation* = '&' or '|'

4. All values and expressions can be preceded with an optional '!' symbol.

There are no characters in the expression other than what's listed below. Here is each character's definition:

1. '&' represents AND. ($a\&b$) evaluates to true if both a and b are true; false otherwise.

2. '|' represents OR. ($a|b$) evaluates to false if both a and b are false; true otherwise.

3. '!' represents NOT. !(a) evaluates to false if a is true; true otherwise.

4. '(' and ')' are the endpoints of expressions. Order of operations stipulates that all expressions inside parentheses must be evaluated first. There is always a corresponding end parenthesis ')' for every beginning parenthesis '('.

5. 't' and 'f' represent true and false, respectively.

Input

The first line contains an integer x such that $0 < x \le 10000$. This is followed by x lines each containing a complete Boolean expression (up to 50 characters long) followed by one space, the equals symbol '=', another space, and the test brain's evaluation of the expression (either 't' or 'f').

Output

On one line for each test brain, print the number of the brain followed by a colon and a space followed by either 'Good brain' if the expression was evaluated correctly or 'Bad brain' if the expression was not evaluated correctly. Once you are finished, I suggest you start running as fast as you can, as the zombies will only give you so much of a head start for your help.

Sample Input

```
4
(t|f) = f
((t|f)&((f|t)&f)) = f
((f&f)|(f|!(t&f))) = t
(f|(f|t)) = f
```

Sample Output

```
1: Bad brain
2: Good brain
3: Good brain
4: Bad brain
```