Watson

Lea is amazed by Watson, the artificially intelligent computer system that won Jeopardy! in 2011. Watson's victory proves how fast the field of artificial intelligence evolves and showcases the impressive capabilities of current systems.

Lea wanted to understand how these systems work and therefore decided to build her own Watson which she called "LEAtron 3000"! After some days spent on the project she realized that she will not be able to beat Watson. This was, of course, not due to her knowledge, but due to the limited amount of computing power she had access to (namely, her old Strawberry Tau).

Since she still wanted to build something, Lea changed the game to an easier one. Instead of answering general questions, "LEAtron 3000" should solve easy equations, which were still given in natural language. She wrote the code and tested it several times, the machine always won against her!

Before showing it to the world, she applied some last performance tweaks and somehow managed to crash the machine. As Lea is lazy, she never made any backups and does not want to write everything from scratch again. Still, she wants to take her machine to several quiz shows, defeat all the human beings and win all the prize pools. Can you help her and rewrite the code?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of a single line containing a string s. The string will consist of digits 0 to 9 and the words "plus", "minus", "times" and "tothepowerof". There will always be at least one digit between operator strings. No number other than 0 will begin with 0.

Output

For each test case, print a line containing "Case #i: y" where y is the result of the computation given in s. In contrast to usual notation, all operations should be evaluated from left to right.

Constraints

- $1 \leq t \leq 20$.
- s will have at least 1 and at most 1000 characters.
- y and all intermediate results will be between -10^9 and 10^9.
- All numbers appearing in s will be between 0 and 10^9 (inclusive).

Sample Input 1

```
3
1plus12minus3
5tothepowerof5minus3
1minus8times5tothepowerof3
```

Sample Output 1

```
Case #1: 10
Case #2: 3122
Case #3: -42875
```