Before the 2009 elections at the European Parliament, Bill and Ted have asked their friends to make guesses about the outcome of the ballot. Now, the results have been published, so Bill and Ted want to check who was right. But checking the results of their many friends would take a very long time, and they need the evaluation to be done by a computer. Since they are not so good at programming, they ask you for help.

Input

The data provided by Bill and Ted has the following format: The first line consists of the number p of parties followed by the number g of guesses (with $1 \leq p \leq 50$ and $1 \leq g \leq 10000$). Then follow p lines, each line consisting of a unique party name of length ≤ 20 (only containing letters 'a'-'z', 'A'-'Z' and digits '0'-'9') and the achieved vote percentage of this party with one digit after the decimal point. After the parties follow g lines, each consisting of a guess. A guess has the form $P_1 + P_2 + \ldots + P_k\ COMP\ n$, where P_1 to P_k are party names, $COMP$ is one of the comparison operators '<', '>', '<=', '>=' or '=' and n is an integer between 0 and 100, inclusively. Each party name occurs at most once in each guess.

Output

For each guess, sum up the vote percentages of the parties and compare them with the specified integer n. Then, print a line stating whether the guess was correct. See the sample output for details.

Note: Be careful with the comparison of floating point values, because some values in the input (like 0.1) do not have an exact representation as a floating point number.

Sample Input

```
6 5
CDU 30.7
SPD 20.8
Gruene 12.1
FDP 11.0
DIELINKE 7.5
CSU 7.2
FDP > 11
CDU + SPD < 50
SPD + CSU >= 28
FDP + SPD + CDU <= 42
CDU + FDP + SPD + DIELINKE = 70
```

Sample Output

```
Guess #1 was incorrect.
Guess #2 was incorrect.
Guess #3 was correct.
Guess #4 was incorrect.
Guess #5 was correct.
```