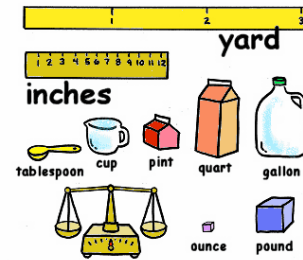


Problem H

Dimensions

You and your friend Christian have decided to take a vacation year and travel all around the world to see magnificent places, meet wonderful people, and experience new cultures. Unfortunately, with different cultures come difficult differences. And the hardest differences engineers like you know of are unit differences. Why are people using miles, firkins, microfortnights, candlepowers, boisseaux, foes, pints, kWh and not to mention degrees Celsius when they instead would be much happier using the standard and beloved SI units of metres, kilograms, seconds, amperes, kelvins and candelas?



Quantity	Name	Symbol
length	metre	m
mass	kilogram	kg
time	second	s
electric current	ampere	A
temperature	kelvin	K
luminous intensity	candela	cd

Table 1: List of SI units

In fact, you love the SI units so much that you refuse to use any other units. Derived units like the joule (J), the newton (N), and the ohm (Ω) are perfectly expressible in their equivalent SI units of $\text{kg m}^2 / \text{s}^2$, $\text{kg m} / \text{s}^2$, and $\text{kg m}^2 / \text{s}^3 \text{ A}^2$, respectively. So during your travel, you record all units you come across, along with their definitions. Of course, some definitions are depending on other definitions, like $\text{Pa} = \text{N} / \text{m}^2$.

With your definitions ready at hand, you don't have to put up with such nonsense as 60 firkins / microfortnights or 63 km / h anymore, since you can always convert them SI units. Even calculations like $100 \text{ m} + 1.3 \text{ km}$ and $7 \text{ N} \cdot 8 \Omega$ become a breeze to you.

Input specifications

The following syntax is given ('?' denotes "zero or one", '+' denotes "one or more", and '*' denotes "zero or more"):

```

power ::= { integer larger than 1 }
unit ::= { upper or lower case English letter }+
dimension ::= unit ['^' power]?
size ::= { any floating-point number } [' ' dimension]* ['/' [' ' dimension]+]?
operator ::= '+' OR '-' OR '*'

```

expression ::= *size* ' ' *operator* ' ' *size*
unit definition ::= *unit* '=' *size*

All *units* have lengths less than 10, and in a *size*, no *units* are repeated.

The first line of the input consists of a single integer U , the number of new units. Then follow U lines with new *unit definitions*. After the unit definitions follows a line with a single integer N , and then N lines with either an *expression* or a *size*.

In the input, all *powers* are less than 5, and all *units* are SI unit symbols or previously defined units.

Output specifications

For each computation, output one line with the answer to the expression, or the size itself, converted to SI units. If the answer cannot be computed, output “Incompatible” (without the quotes). The answer should be formatted as a *size*, with the following clarifications:

- Units can be written in any order, but keep them on the correct side of the division sign.
- Do not output a unit if its exponent is 0, or its exponent if it is equal to 1.
- Output one space between each unit and between units and the division sign, \.
- Do not output any spaces before or after the exponent sign, ^.

Notes and Constraints

- $0 < U \leq 100$
- $0 < N \leq 1000$
- Units are case sensitive.
- No lines are longer than 140 characters.
- No input, output or part of any computation have size of absolute value above 10^{100} .
- No part of any computation will yield division by 0.
- Any output with a relative or absolute error of 10^{-6} is accepted.

Sample input

```
4
km = 1000 m
h = 3600 s
J = 1 kg m^2 / s^2
X = 3 m^2 kg s / A K cd^4
4
100 m + 1.3 km
63 km / h
1E5 J * 0.003 h^2 / km^2
1 J - 2 X
```

Output for sample input

```
1400.0 m
17.5 m / s
3888.0 kg
Incompatible
```