Problem A : Operations (blue balloon)

Due to heavy snow, George Dantzig was late for his class. When he arrived, his elementary school teacher, Jerzy Neyman had written a homework assignment on the blackboard. It seemed a little harder than usual, but George wrote it down anyway. The assignment was a complex mixture of additions, multiplications and divisions (shown by fractions) on positive numbers; something like the expression below:

$$5 + \frac{4 \times \frac{\frac{5}{7}}{6}}{2 + 3 \times 5} \times 2 + 1 + \frac{1}{1}$$

Every expression or sub-expression has a baseline on which all its elements lie, including operators, numbers, and fraction lines. Obviously, the numerator and denominator of a fraction are respectively placed on top and bottom of its fraction line and they themselves are smaller (possibly complex) sub-expressions. Note that multiplication has a higher priority than addition in evaluating mathematical expressions. Your task is to help George to evaluate the expression as a simple fraction.

Standard Input

There are several test cases in the input. Each test case starts with a line containing a single integer *n* as the height of the complex expression ($1 \leqslant n \leqslant 60$). The expression is presented in the next *n* lines. Each of these lines has at most 200 characters and consists of space characters, consecutive digits as positive numbers, "*" characters as operators for multiplication, "+" characters as operators for addition, and consecutive sequences of "-" characters as fraction lines. If a fraction line is made of t characters ($t \geqslant 3$), its corresponding numerator and denominator are horizontally aligned within its t - 2 middle characters. Note that there might be some vertical space between a fraction line and its corresponding numerator and/or denominator. Also, the elements of a baseline might be separated by some space characters. Furthermore, some unnecessary spaces at the end of each line may be omitted. The input terminates with a line containing 0 which should not be processed.

Standard Output

For each test case, print a single line containing the value of the expression in the form of "numerator/denominator". Note that the numerator and the denominator of each fraction must be coprime, i.e., their greatest common divisor should be 1.

Sample Input	Sample Output
(see below)	(see below)

Sample Input

```
7
       5
      ---
       7            1
   4*-----
       6
5+---------*2+1+ -----
   2+3 *5         1
0
```

Sample Output

```
2519/357
```

Problem B: Vote

(red balloon)

Jenabkhan who has become billionaire from his Laboo bussiness, is now running for president. His country uses a strange mechanism, so-called electoral college, to select the president. There are several states in the country, and each state counts the votes independently. Depending on the population, each state has some members in the electoral college, and all of those members will vote the candidate with the majority of votes in their state. In the case of ties, each state has some tie-break rule to announce the clear winner. The president will be the candidate who receives more than half of votes in the electoral college.

Given the chance of Jenabkhan to win in each state, compute his winning probability in the electoral college.

Standard Input

The input consists of several test cases. Each test case starts with a line containing a single integer n denoting the number of states ($1 \leqslant n \leqslant 1000$). Each of the next n lines contains a real value p_i with at most 4 digits after the decimal point ($0 \leqslant p_i \leqslant 1$) and a positive integer e_i, specifying the winning probability of Jenabkhan in the i-th state and the number of electoral votes associated with that state, respectively. The total number of members in the electoral college is an odd number and is no more than 2000. The input terminates with a line containing 0 which should not be processed.

Standard Output

For each test case, output in a single line containing the winning probability of Jenabkhan, rounded to exactly four digits after the decimal point (e.g., 0.3000 is correct while 0.3 is not).

Sample Input	Sample Output

1
0.4 1
3
0.5 1
0.5 2
0.5 10
3
0.5 1
0.5 2
0.5 2
2
0.2 1
0.8 10
2
0.25 1
0.751 10
0
	0.4000
	0.5000
	0.5000
	0.8000
	0.7510

Problem C : Soldier

(yellow balloon)

You have just started your military service with the border guards. Since you are a Computer Science graduate, they asked you to implement a schedule for the soldiers. Initially no soldier is on duty and you need the schedule to satisfy 2 requirements:

- There are **N** soldiers, each soldier can be on duty for at most **K** continuous months and then he must take one month of vacation.

- The schedule needs to make sure that the guaranteed minimum number of soldiers on duty at any given time is maximized.

Given **N** and **K**, the system will calculate the maximum guaranteed number of soldiers to be on duty at any given time.

Standard Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer **T**, the number of test cases ($1 \leq T \leq 100$).

Each test case consists of a line containing 2 space separated integers:

- **N**: The number of soldiers ($0 \leq N \leq 10,000,000$)

- **K**: The number of continuous months a soldier can be on duty before they have to take a month of vacation ($0 \leq K \leq 10,000,000$).

Standard Output

For each test case, print a single line containing the maximum guaranteed number of soldiers on duty at any given time.

Sample Input	Sample Output

3
4 1
9 3
21 3
	2
	6
	15

Problem D : Candy (Orange balloon)

Pokemon Go just released the Buddy update. It lets you select a Pokemon to appear alongside your trainer's avatar on your profile screen. As you walk with your buddy, it will find candy that can be used to evolve the Pokemon.

The Buddy system divides the Pokemons into 3 groups. Each group gives one candy upon walking for 1, 3, and 5 kilometers respectively.

In this problem you will be given the Pokemon group **G**, the number of candies **C** you initially have, and the number of candies **E** required to evolve the Pokemon. You should calculate the number of Kilometers required to walk in order to evolve the Pokemon.

Standard Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer **T**, the number of test cases ($1 \le T \le 100$).

Each test case consists of a line containing three space separated integers:

- **G**: The group of the Pokemon ($1 \le G \le 3$)
- **C**: The initial candies you have ($0 \le C \le 100$)
- **E**: The candies required to evolve the Pokemon ($1 \le E \le 100$)

Standard Output

For each test case, print a single line containing the number of Kilometers of walking required to Evolve the Pokemon.

Sample Input	Sample Output

2
1 15 51
1 18 21
	36
	3

Problem E : Sisters (Purple balloon)

Shika and Bika are two cute little sisters. They like to play together all the time. These days, they are learning about integers. They developed the following game. Shika chooses a certain range, without telling Bika about it, then the game goes in rounds. In each round, Shika calls out one integer from the range she chose, then Bika replies with another integer (it needn't be in the range, just any integer). Shika ends the game only after making sure that she called out each integer in the range at least once. This means Shika can call out an integer more than once and Bika can reply with different integer each time. For example, Shika might call out 3 in a round and Bika replies with 4, then later in another round, Shika calls out 3 again but Bika replies with 3 this time. After every round, they write down a pair consisting of the integer that Shika called out in the round and the integer Bika replied with. The write up has the following problems:

- They do not write the pair in a certain order. So, if Shika says x, and Bika replies with y, they sometimes write this pair as (x, y) and other times as (y, x).
- In some rounds, they forget to write the pair at all.

Next day after the game, Shika asks Bika questions like: "Did I call out integer q yesterday?". Bika has the write up and she is allowed to answer with one of the following answers. "YES", "NO", or "NOT SURE". Given the written pairs, can you help Bika answer Shika's questions?

Standard Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer **T** $(1 \leq T \leq 100)$.

Each test case represents a game and starts with a line that contains two space separated integers:

- **N**: The number of written pairs $(1 \leq N \leq 1000)$

- **Q**: The number of queries ($1 \le Q \le 1000$)

Followed by **N** lines each containing 2 space separated integers **a** and **b** representing the i^{th} pair ($-1,000,000 \le a, b \le 1,000,000$)

Followed by **Q** lines each containing a single integer **A** representing the j^{th} query ($-1,000,000 \le A \le 1,000,000$)

Standard Output

For each test case, print Q lines, where the ith line answers the ith query with "YES", "NO", or "NOT SURE".

Sample Input	Sample Output

1
2 3
10 7
20 14
3
8
11
	NOT SURE
	NOT SURE
	YES

Note

In the test case, we can see that we have the following possibilities:

1. - Round 1: Shika called out 10, Bika replied with 7
   - Round 2: Shika called out 20, Bika replied with 14
2. - Round 1: Shika called out 10, Bika replied with 7
   - Round 2: Shika called out 14, Bika replied with 20
3. - Round 1: Shika called out 7, Bika replied with 10
   - Round 2: Shika called out 20, Bika replied with 14
4. - Round 1: Shika called out 7, Bika replied with 10
   - Round 2: Shika called out 14, Bika replied with 20

For the first query, Bika would never be able to decide if 3 was called out by Shika or not, since sorne rounds' pairs rnay be rnissing. For the second query, in possibilities 1 and 2, Bika can't guarantee that Shika called out 8. For the third query, 11 was for sure called out by Shika, as it is between the two integers called out by Shika in all possibilities.