Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen (LEA)
Prof. Dr. Ernst W. Mayr
Christian Müller, Philipp Hoffmann, Chris Pinkau, Stefan Toman

Summer Term 2015
Problem Set 7
11.06.2015

Algorithms for Programming Contests

This problem set is due by

Thursday, 18.06.2015, 6:00 a.m.

Try to solve all the problems and submit them at

`https://judge.in.tum.de/`

This week's problems are:

The following amount of points will be awarded for solving the problems.

Problem	SS15N07A	SS15N07B	SS15N07C	SS15N07D	SS15N07E
Difficulty	easy	easy	medium	medium	hard
Points	4	4	6	6	8

If the judge does not accept your solution but you are sure you solved it correctly, use the "request clarification" option. In your request, include:

- the name of the problem (by selecting it in the subject field)
- a verbose description of your approach to solve the problem
- the time you submitted the solution we should judge

We will check your submission and award you half the points if there is only a minor flaw in your code.

If you have any questions please ask by using the judge's clarification form.

SS15N07A Making Change

Author: Chris Pinkau

Templonia is a very strange place. On one hand, there are beautiful temples, great ancient cities and many nice people. On the other hand, the Templonians have acquired some very peculiar customs. The one which causes the biggest problem for Lea is that Templonian merchants always want their money in as few coins or notes as possible. Even worse, there are many different coins and notes for the Templonian Column, and not every merchant takes all values. Surely, the Templonian people are accustomed to this behaviour and are very good in calculating these things. But as a foreign tourist, Lea has some difficulties to say the least. A tolerant person as she is, she does not want to anger the local merchants and tries to adopt Templonian customs. Whenever she has an unusually hard problem at finding the right coins, she takes out her phone and calls you. Can you please help her not get beaten up by angry merchants?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a single line containing two integers n and c. n is the number of coin and note values and c is the amount of money that must be spent. The next line consists of n distinct integers $v_1, \ldots, v_n$ describing the coin/note values in increasing order. You may assume that a coin of value 1 is always available, and that Lea has as many of all the coins/notes as she needs.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is a space-separated sequence of n integers $a_1, \ldots, a_n$, where a_i means that the optimal solution uses exactly a_i coins/notes of value v_i.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 100$
- $1 \leq c \leq 100000$
- $1 \leq v_i \leq 10000$ for all $1 \leq i \leq n$
- $1 \leq a_i \leq c$ for all $1 \leq i \leq n$

Sample Data

Input

```
2
6 29
1 2 4 5 6 8

4 43
1 6 7 13
```

Output

```
Case #1: 0 0 0 1 0 3
Case #2: 0 5 0 1
```

SS15N07B Zombie Apocalypse

Author: Stefan Toman

Do not watch too many horror movies! If you are asking why, read what happened to Lea last week:

Lea loves horror movies and watches vast numbers of them. But last week she exaggerated by watching a 24 hours marathon of zombie apocalypse movies. Watching all these movies without sleeping left her dazed. After the last movie ended (there was no happy end...) she decided to prepare for the zombie apocalypse since it must be coming if there are that many movies about it. She rushed to the closest supermarket and began to buy huge amounts of food.

Now, she realises that she does not know what to buy to be well-prepared. Although in confusion, she still wants to do this right, but the weight she may carry is limited and she fears to go to the supermarket a second time since it may be too late by then. Can you help her once more?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing two integers m and n, where m is the weight Lea may carry in grams and n is the number of groceries available. n lines describing the groceries follow. The i-th line contains three integers p_i, l_i and s_i where p_i is the number of packets of this grocery available in the supermarket, l_i is the weight of a packet of this grocery in grams and s_i is the amount of calories per packet.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is a space-separated list of groceries to buy (grocery i may appear at most p_i times in this list). The sum of their weights should be at most m and the sum of their calories should be as big as possible. If there are multiple optimal solutions, any of them will be accepted.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 100$
- $1 \leq m \leq 3000$
- $1 \leq p_i \leq 100$ for all $1 \leq i \leq n$

- $1 \leq l_i \leq 100$ for all $1 \leq i \leq n$
- $1 \leq s_i \leq 10000$ for all $1 \leq i \leq n$

Sample Data

Input

```
2
10 2
1 3 5
6 1 1

10 3
2 3 7
1 2 8
3 7 5
```

Output

```
Case #1: 1 2 2 2 2 2 2
Case #2: 1 1 2
```

SS15N07C Vaults & Vampires

Author: Stefan Toman

Grunkh, the brutal troll, defeated the good human mage Gregor McHexroy in a long and exhausting battle. Swords and clubs went into splinters, a forest burned down and even the mountain where all squirrels from the forest ran for shelter exploded during the epic battle. Nevertheless, Grunkh survived albeit badly injured. He collects all the gold McHexroy had in his pockets and trudges back to his cave to heal his wounds. Suddenly, a wild rat appears and dares to attack Grunkh, who is 10 times as big and 100 times as strong as the rat. Normally, this would be an easy fight, but now Grunk is heavily injured and can barely move.

"I fought more than one hour to defeat this mage and now a rat tries to kill me and get all the loot? This is ridiculous, I need to find a new GM (game master)..." Lea thinks, who is playing Grunkh at the latest gathering of her friends testing the new RPG "Vaults & Vampires". Nevertheless, she has to roll the dice now and see whether she can beat this tiny rat. At least, she wants to know the exact probability to win before she does so. Can you help her?

Input

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

Each test case consists of a line containing an integer n and a string x. n is the least number of points Lea has to get when rolling the dice and x is a string describing the dice. A set of a dice with b sides each (labelled 1 to b will be described as "adb". Multiple sets of dice may be concatenated by "+" signs.

Output

For each test case, output one line containing "Case #i: y" where i is its number, starting at 1, and y is the probability to roll at least n points. The probability should be printed as a simplified rational number in the format "numerator/denominator". Simplified means that the numerator and denominator should not have a common divisor bigger than one and should not be negative. 0 should always be printed as "0/1".

Constraints

- $1 \leq t \leq 20$
- $0 \leq n \leq 1000$
- There will be at most 50 dice with at least 3 and at most 20 sides each.

Sample Data

Input

```
3
3 1d6
15 1d6+2d20
75 25d6
```

Output

```
Case #1: 2/3
Case #2: 523/600
Case #3: 1478174426405911253/1579460446107205632
```

SS15N07D Packing Cases

Author: Philipp Hoffmann

Just recently, during Lea's visit at her uncle's house, she was reminded that while some people are quite tall, sadly she is not. She could not even reach the glasses that were stored in the topmost shelf in the kitchen. Luckily for her, there were a lot of packing cases lying around and she could use them to build a tower and then climb on it to reach the glasses.

Building such a tower is of course a very shaky endeavour, and Lea does not want to fall. So she imposed the following restriction on the tower: Given two packing cases a and b with dimensions x_a, y_a, z_a and x_b, y_b, z_b, case a may only be stacked onto case b if $x_a < x_b$ and $y_a < y_b$. Please remember that a case can be rotated to fit that restriction.

Lea now has to figure out whether it is possible to reach the desired height if she stacks the cases optimally, or not.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with two integers, h and n, the height the tower should reach and the number of case types, n lines follow. The i-th line describes the i-th case layout and contains three integers x_i, y_i, z_i. Lea has exactly 5 Boxes of each type at her disposal.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is either "yes" if Lea can build a tower of height at least h according to the constraints, or "no" if it is not possible to do so. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 1000$
- $1 \leq x_i, y_i, z_i \leq 40000$
- $1 \leq h \leq 5 * 10^6$

Sample Data

Input

```
2
9 1
5 4 3

7 2
4 2 2
3 1 5
```

Output

```
Case #1: no
Case #2: yes
```

SS15N07E Birthday Cake

Author: Stefan Toman

Lea and Bea are best friends, but they do not see each other often since Bea moved to another city far away from Lea's home. They still call each other often and meet from time to time. They also send gifts for birthdays and christmas regularly. Bea's birthday is next week and this year Lea wants to bake a birthday cake for her. The cake should be sent to Bea, so Lea needs to finish the cake today to bring it to the post office tomorrow morning.

Unfortunately, Lea's favourite TV show is also on today: the famous Baking Tray World Championships! In this show some C-list celebrities and old professional athletes who need money slide down an ice track on baking trays. The event is moderated by a famous guy who was a baker when he was young and switched to TV shows later on. He always does the craziest TV shows and most people are never sure whether he is sober. All in all, it is a great show!

Lea prepared the dough for the cake and wants to bake it, but she does not know how much time it will need to bake. On the other hand, she wants to leave the TV in her living room as rarely as possible to check whether the cake is ready. If she leaves to check the cake there are three possibilities: The cake is not ready yet (Lea will leave it in the oven), the cake is perfect (Lea will switch off everything and go back to the TV show), or the cake is burned (Lea will throw the cake away). Since Lea wants to make the perfect cake and expects that for some tries the cake will be in the oven for too long, she prepared dough for several cakes.

Lea needs a strategy for checking the cake satisfying her constraints. She knows that all cakes are the same and will always look the same after the same time in the oven. The temperature in the oven is constant and cannot be changed. The perfect baking time will be an integer number of minutes. Lea does not care how many of the cakes she prepared she has to throw away, she just needs one perfect cake. Also, she does not care how long it will take to bake the cakes, the TV show is quite long. Her strategy may change after she checks the cake depending on what she sees. Lea wants to minimize the number of times she has to go to the kitchen in the worst case. What number can she achieve?

Input

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

Each test case consists of a line containing two integers n and m where n is the number of cakes Lea prepared and m is the maximum baking time in minutes.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the numbe rof times Lea has to go to the kitchen in the worst case for her best strategy. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 500$
- $1 \leq n \leq 1000$
- $1 \leq m \leq 1000$

Sample Data

Input

```
3
3 5
1 5
5 2
```

Output

```
Case #1: 3
Case #2: 5
Case #3: 2
```