Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen (LEA)
Prof. Dr. Harald Räcke
Moritz Fuchs, Philipp Hoffmann, Christian Müller
Chris Pinkau, Stefan Toman

Winter Term 2015/16
Problem Set 8
02.12.2015

Algorithms for Programming Contests

This problem set is due by

Wednesday, 09.12.2015, 6:00 a.m.

Try to solve all the problems and submit them at

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

This week's problems are:

A **Exam Preparation Revisited** 3

B **Making Change** 5

C **Packing Cases** 7

D **Potato Farmers** 9

E **Chocolate Tasting** 11

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

Problem	A	B	C	D	E
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.

A Exam Preparation Revisited

Author: Stefan Toman

Do you remember the problem "Exam Preparation" in which Lea tried to put as much information as possible on one sheet of paper that she is allowed to use during an exam? Well, Lea needs your help again since she has changed her mind on three things:

- Lea thought there was an unlimited supply of information for each topic, but that turns out to be wrong: For instance, there is no use writing down more than five pieces of information about the birth date of the university's patron Thomas Underwood.
- Since Lea surfs the internet a lot, she did not study as much as she planned to. Now, she needs the perfect cheat sheet with the maximum amount of information.
- The allowed size of the cheat sheets is not as big as expected.

Can you help her again?

Input

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

Each lecture starts with a line containing two integers: m, the number of characters that fit on the allowed cheat sheet, and n, the number of topics covered. n lines describing the topics follow. The i-th line contains three integers p_i, l_i and s_i where p_i is the number of pieces of information available, l_i is the length of a piece of information for this topic and s_i is its score.

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 topics to be added (topic i may appear at most p_i times in this list). The sum of their lengths should be at most m and the sum of their scores should be as big as possible.

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

```
7
10 2
1 3 5
6 1 1

10 3
2 3 7
1 2 8
3 7 5

10 3
1 5 10
3 1 1
3 2 3

6 2
4 6 10
3 3 5

1 6
4 6 10
3 3 5
2 3 2
7 7 23
9 8 17
4 10 8

7 2
2 2 7
5 3 3

8 4
1 4 7
1 2 3
1 2 1
1 2 2
```

Output

```
Case #1: 1 2 2 2 2 2 2
Case #2: 1 1 2
Case #3: 1 2 3 3
Case #4: 1
Case #5:
Case #6: 1 1 2
Case #7: 1 2 4
```

B 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

```
14
6 29
1 2 4 5 6 8

4 43
1 6 7 13

4 780
1 6 35 97

9 827
1 15 17 26 64 70 79 88 98

5 598
1 21 45 64 68

2 756
1 55

5 507
1 23 33 35 58

6 789
1 9 70 73 79 98

7 418
1 15 19 64 74 87 88

10 481
1 13 16 27 38 44 51 59 75 100

7 979
1 9 11 49 68 76 100

5 932
1 57 80 81 89

10 612
1 6 28 30 36 67 69 74 96 99

3 306
1 18 19
```

Output

```
Case #1: 0 0 0 1 0 3
Case #2: 0 5 0 1
Case #3: 4 0 0 8
Case #4: 1 0 0 0 0 2 0 0 7
Case #5: 0 1 1 3 5
Case #6: 41 13
Case #7: 0 0 2 1 7
Case #8: 1 0 2 0 2 5
Case #9: 0 0 1 1 1 3 0
Case #10: 0 1 0 0 0 0 0 2 2 2
Case #11: 0 0 1 0 1 0 9
Case #12: 1 1 0 2 8
Case #13: 0 0 0 1 0 0 0 0 4 2
Case #14: 0 17 0
```

C 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

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

Sample Data

Input

```
11
9 1
5 4 3

7 2
4 2 2
3 1 5

8 2
3 5 5
2 2 3

6 3
4 1 1
5 2 4
3 1 3

8 2
5 2 2
4 5 1

10 4
2 1 5
2 5 3
4 1 1
5 2 4

6 3
1 2 1
1 5 3
5 4 2

7 1
3 1 3

7 2
1 5 4
6 5 6

2 4
5 4 1
1 3 4
3 6 3
2 6 2

8 2
1 6 6
2 2 2
```

Output

```
Case #1: no
Case #2: yes
Case #3: yes
Case #4: yes
Case #5: no
Case #6: yes
Case #7: yes
Case #8: no
Case #9: yes
Case #10: yes
Case #11: no
```

D Potato Farmers

Author: Christian Müller

Lea's friend Nick is a biologist at UBOP (University for Biologists and Other People). Recently, the IMLRS (Institute for More or Less Reliable Statistics) published a report showing a strong correlation between a farmer's capacity for logical thinking and the dimensions of their produced agricultural products.

This study would make a great source for Nick's next paper on "Agricultural Production Techniques Through The Ages". However, there have been several incidents with studies by that particular institute, so Nick wants to verify their results for himself. Luckily for him, the IMLRS included their data in the report.

As a start, Nick wants to find examples for farmers where those with lower IQ produce higher average weight of their potatoes.

Input

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

Every test case starts with a line containing an integer n, the amount of farmers. n lines follow. The k-th line contains two integers iq, the IQ of farmer k (in hundredths of IQ points) and w, the average weight of his potatoes (in tenths of grams).

Output

Your task is for each test case to find a sequence of farmers $k_1, ...k_n$ with maximum length such that $iq(k_1) \leq iq(k_2) \leq ... \leq iq(k_n)$ and $w(k_1) > w(k_2) > ... > w(k_n)$. Note that you can reorder the farmers given in the input as you see fit.

For each test case, print a line containing "Case #i: x" where i is its number, starting at 1, and x is the maximum possible length of the sequence.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 20000$
- $1 \leq w \leq 10000$
- $1 \leq i \leq 20000$

Sample Data

Input

```
6
2
100 1000
150 800

3
120 1000
100 800
150 900

5
9 19
7 13
12 20
7 15
20 19

5
4 20
6 15
14 12
8 9
19 13

6
11 9
1 17
6 18
17 13
17 17
6 9

9
10 11
8 12
7 16
14 7
2 2
20 17
9 17
10 4
2 15
```

Output

```
Case #1: 2
Case #2: 2
Case #3: 2
Case #4: 3
Case #5: 3
Case #6: 4
```

E Chocolate Tasting

Author: Stefan Toman

Lea is a well-known chocolate gourmet: She loves all types of chocolate and has an impressive knowledge about them. Next week, she wants to organize a big chocolate tasting for all of her friends. They should try even the most exotic chocolate bars and feel the incomparable pleasure of melting high-quality chocolate on their tongues.

As Lea has many friends and maybe even more kinds of chocolate she likes, she needs to plan buying the chocolate in advance. She will need quite a few trips to the store to buy all the chocolate she needs. Since it is a little warmer at her home than in the chocolate store she knows in which order to buy the chocolate in a way that each chocolate bar has the perfect eating temperature when the big chocolate tasting starts. To find the perfect order, one needs to have a great knowledge about melting temperatures and the temperature curves for making chocolate, but Lea knows everything by heart and computed that order already. The store also prepared packages for Lea containing the chocolate bars.

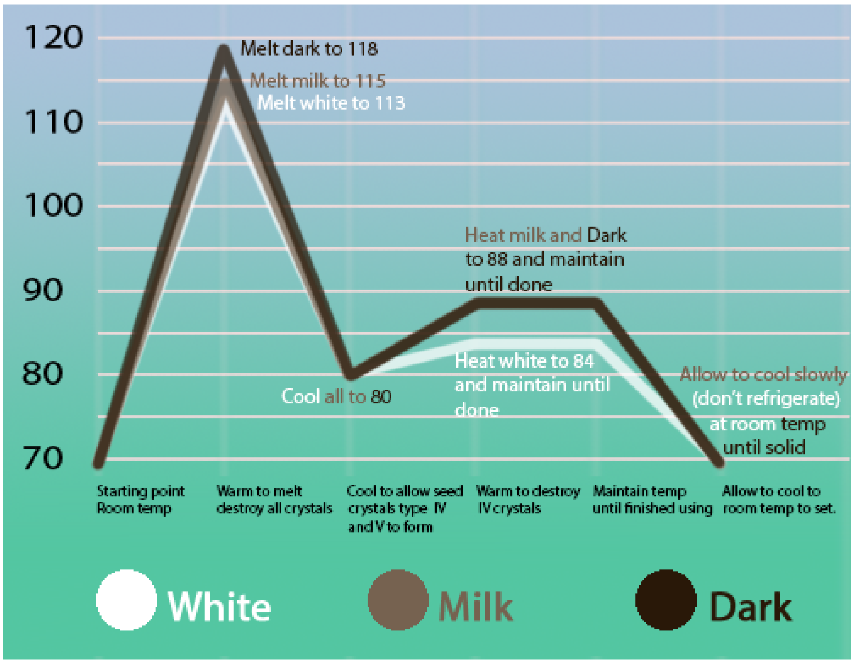

Figure 1: The perfect temperature curve for making chocolate.
Source: `http://www.thecookinggeek.com`

Given the order to buy the chocolate bars, she needs to decide which packages to take at each visit of the store. Lea wants to visit the store as few times as possible. If she reaches the minimum number of store visits, she wants to minimize her chocolate carrying coefficient CCC. The CCC of one trip to the store is computed as

$$CCC(x) = \begin{cases} 0 & \text{if } x \le a \\ (x-a)^2 & \text{if } a < x \le b \end{cases}$$

where x is the amount of chocolate she carries at the moment, a is the amount of chocolate that Lea can carry without problems and b is the maximum amount of chocolate Lea can carry at all (all given in kilograms). The sum of the CCC's of her trips should be as small as possible. Can you tell her what this sum will be?

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 three space-separated integers n, a and b where n is the number of chocolate packages and a and b are as described above. One line containing n space-separated integers follows describing the chocolate packages to buy in order. The i-th integer x_i describes the weight of the i-th package.

Output

For each test case, output one line containing "Case #i: y" where i is its number, starting at 1, and y is the minimum sum of all CCC's such that the number of store visits is also minimized. Print "impossible" for y if Lea is not able to carry the chocolate to her home.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 10^5$
- $1 \leq a \leq b \leq 100$
- $1 \leq x_i \leq 100$ for all $1 \leq i \leq n$

Sample Data

Input

```
14
6 1 5
1 1 1 1 1 1

5 2 10
4 7 1 9 3

3 2 5
4 8 5

4 1 2
1 2 1 1

5 1 1
1 1 1 1 1

5 4 4
3 4 1 3 2

2 3 4
4 3

6 3 6
3 5 5 2 1 6

8 3 4
4 4 4 1 3 3 1 3

5 2 4
3 2 3 2 4

8 3 7
3 6 5 2 1 5 4 2

6 4 4
1 4 3 1 4 2

5 1 3
2 1 2 1 2

6 4 4
3 4 4 1 1 3
```

Output

```
Case #1: 8
Case #2: 90
Case #3: impossible
Case #4: 2
Case #5: 0
Case #6: 0
Case #7: 1
Case #8: 17
Case #9: 5
Case #10: 6
Case #11: 43
Case #12: 0
Case #13: 9
Case #14: 0
```