Hamburg Collegiate Programming Contest

at Hamburg University of Technology

July 2010

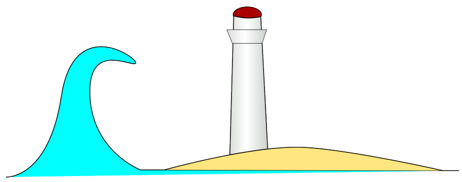
Problem Set

Prof. Dr. Sibylle Schupp
Prof. Dr. Volker Turau
Gustav Munkby
Christian Renner

(intentionally left blank page)

Contents

A :. The King of Atlantis



The people of the prosperous and wealthy city of Atlantis have ever been afraid of the downfall of their empire. Due to their city being situated on a tiny and rather flat island, even a small flood wave could instantaneously wipe everything from the surface. The only way of escape would be to seek shelter on top of the enormous lighthouse. The selfish king of Atlantis would therefore like to know which places were safe to stay on, in case a flood wave hit the island within a given number of arbitrary time units T.

For this purpose, he has offered you a sack of gold to determine these safe places. For the sake of simplicity, you may assume that all places of the island will be hit by the flood wave at the same time.

Input Specification

The first line of input contains the number of testcases, at most 100.

Each test case starts with a line holding $1 \le N \le 500$, the number of places, $L \le N$, the place of the lighthouse, and $0 \le T \le 1\,000$, the time between sighting of a flood wave and its arrival at the island.

Next comes a line with the single number M, the number of ways between places. Then follow M lines with three integers each: a, b, and t, where a is the starting place of a way and b is its end ($1 \le a, b \le N$, $a \ne b$), while $t \le 1\,000$ is the number of time units needed by the king to move from a to b. Note that ways are directed, i.e., a way from a to b does not imply that there is a way from b to a. In addition, there is no more than one way with the same direction between two places.

Output Specification

Per test case one line with the ascendingly ordered set of safe places, separated by a single blank.

Sample Data

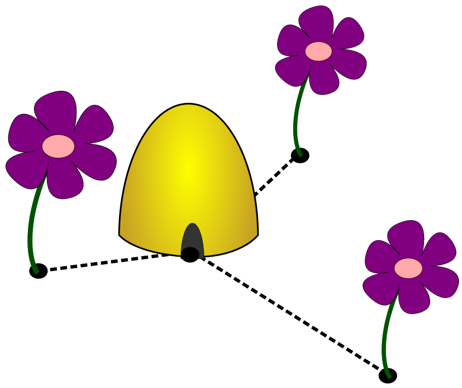
Input

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

Output

```
1 2
```

B :. New Bees in Town



Her highness, bee queen Beatrix has decided to move her colony to a new region with a huge amount of fresh flower fields. For Beatrix is a gentle queen, she has decided to instruct her best engineer—you—to find the best possible construction position for the beeyard.

Given the center positions of the flower fields and the number of flowers of these fields, find the position of the beeyard to minimize the effort of the colony to visit all flowers once! The effort of visiting a field of flowers grows quadratically with the line-of-sight distance to that field, and a bee cannot visit more than 10 flowers before it returns to the beeyard. Since the bees are new to the region and thus afraid of getting lost, they do not travel from one flower field to another one directly, i.e., they always visit the beeyard before flying to the next flower field. *Due to the high number of flowers per field and the lack of precise knowledge, you may assume that all flowers of a field are situated in its center. You may also model the beeyard as a single point.*

Input Specification

The first line of input contains the number of testcases, at most 100.

Each test case starts with a line holding the number F of flower fields ($1 \leq F \leq 10\,000$). Then follow F lines with three values each: Two floating point numbers x and y with the center position of the field in the region ($-10\,000 \leq x, y \leq 10\,000$) and the number f of flowers belonging to the field ($1 \leq f \leq 10\,000\,000$).

Output Specification

Per test case one line with the optimal position of the beeyard, where the two coordinates are separated with a single blank space. An absolute or relative error of at most 10^{-6} is acceptable for your solution.

Sample Data

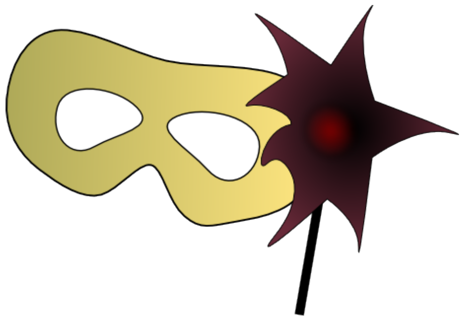
Input

```
2
4
0 0 1
0 1 1
1 0 1
1 1 1
4
0 0 1
0 1 1
1 0 1
1 1 11
```

Output

```
0.5 0.5
0.6 0.6
```

C :. Carnival Planner



The world-renowned yearly carnival Teohaha is held in an unnamed city. In previous years the carnival has been located in a rather remote part of the city, but this year, the idea is to bring the carnival closer to the town central. However, the carnival would require several streets to be closed. Not only because of the carnival itself, but also to give room for all spectators and merchants. As a traffic planner, you are responsible for the city's network of streets, and you are concerned about the suggested route for the carnival. The city consists of several intersections, connected by one- or two-way roads. Obviously, the carnival will prevent any traffic in the whole area enclosed by the carnival route, but you are worried that it might not only have a local impact. You want to make sure that there are no isolated parts in the city. A part of the city is isolated either if it cannot be reached from all other parts of the city, or if you cannot reach all parts of the city starting there.

Input Specification

The first line of input contains the number of testcases, at most 100. Each test case starts with a line containing first the number of intersections I, second the number of roads R, third the length of the carnival route C. The maximum size of each I and C is 1000 and for R 10 000. The following R lines consist of the three numbers i_1, i_2 and w in that order, which describe a road between the intersections i_1 and i_2. The intersections are numbered from 1 to I inclusive. When w is 1 then traffic goes only from i_1 to i_2, but otherwise traffic goes in both directions. The following C lines list the ordering of the intersections visited by the carnival. Note that the start and end position need not be the same.

Output Specification

Per test case, one line listing the size of each isolated part of the city in increasing sizes, separated by a single blank. The size of an isolated part is given by the number of intersections it contains.

Sample Data

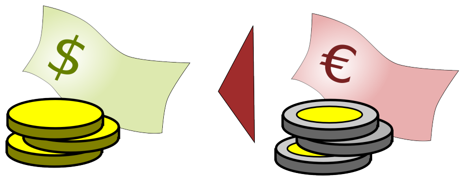
Input

```
2
3 3 1
3 1 0
1 2 0
2 3 1
2
5 5 2
5 1 0
5 2 0
5 3 0
5 4 0
3 4 0
5
1
```

Output

```
2
1 2
```

D :. No Dollars at Hand



During your stay in the United States, you spontaneously travel to the idyllic Babler State Park. Well within the park, you enter a souvenir shop and find a very nice piece of woodcraft that you would like to bring home. At this instant you recall that you completely forgot to exchange money in the morning. As there is no bank close-by and the shop owner only accepts cash, your credit card is of absolutely no help. Yet, you still have some euros left in your wallet, so that your only chance is to annoy other tourists passing by. As most of them are helpful, they will exchange your money; yet, depending on their degree of helpfulness, the exchange rate varies. Since you're anxious to save your money, you ask a whole bunch of people from an European tourist group about exchanging conditions. You write down all exchange rates on a sheet of paper and finally pick those people for exchanging your money that make you spend as few euros as possible. Since the woodcraft souvenir is rather expensive, you will have to exchange money using the offers of various people. In addition, don't end up having more dollars than needed for your purchase—you may exchange less money with a person than offered to you!

Input Specification

The first line of input contains the number of testcases, at most $1\,000$.

Each test case starts with the cost C of the souvenir you want to buy (in dollars) and the budget B of euros you have in your pocket; B and C are non-negative decimals. Then follows the number P of persons in the tourist group ($1 \leq P \leq 200$). The next P lines consist of two decimals: first, the amount of dollars d offered to you for exchange; second, the amount of euros e that the person demands for the exchange. The resulting rate of exchange is also valid, if you exchange less money with a person than is offered to you.

Output Specification

Per test case one line with the minimum amount of euros spent to buy the souvenir. The format needed for your output is always $e.cc$, where $e \geq 0$ are euros and cc are cents (always two digits). If you cannot exchange enough money, print `impossible`.

Sample Data

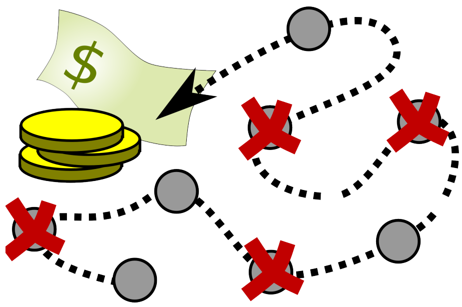
Input

```
2
14.95 20.00
3
5.00 4.00
5.00 5.00
4.00 10.00
14.95 20.00
4
5.00 4.00
5.00 5.00
4.00 10.00
6.00 5.00
```

Output

```
impossible
12.95
```

E :. Easy Cash



A long time ago in the wild west, two brothers decided upon robbing a bank. In fact, they had the idea of robbing a lot of them. To optimize what banks to rob, the two brothers wrote down a long list and estimated how much money there was in each bank. However, the brothers strongly disliked traveling, and therefore decided on an order in which to rob the banks which would incur the minimum amount of travel. This would not have been a problem, unless the two brothers mistakingly published their list in the state-wide newspaper. Thus, if they robbed a bank, the adjacent banks on the list would get suspicious, and the brothers did not want to take any unnecessary chances. What is the optimal set of banks that the brothers could rob, assuming that they cannot rob two banks adjacent on their list?

Input Specification

The first line contains the number of test cases, at most 100.

Each test case follows with two lines. The first line contains the number of banks, no more than 10^5. The second line contains the amount of money in each bank, no more than 10^8.

Output Specification

For each test case, print the maximum amount of money the two brothers can "safely" get away with.

Sample Data

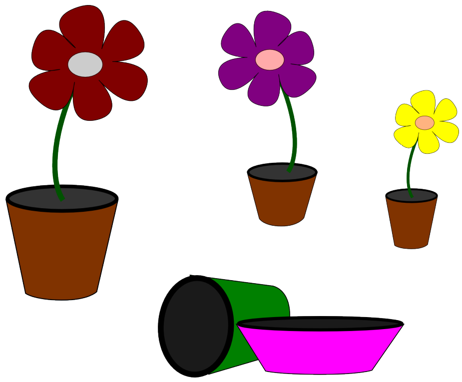
Input	Output

<pre>2
3
100 150 100
9
60 57 5 38 71 87 75 78 45</pre>
	<pre>200
	263</pre>

(intentionally left blank page)

F :. Flower Power

Christian has bought new flowers for his apartment. Unfortunately, he has not thought about the different sizes and colors of the available flower pots and is therefore in an unconvenient situation. If he does not devise a suitable plan for combining flowers and pots, he might end up with flowers that cannot be planted at all. However, he might need to buy new flower pots, even if he finds a perfect assignment of flowers and pots. As he has no clue how to solve his problem and wants to buy as few new pots as possible, he asks you to figure out which flower to put into which pot and how many pots to buy.



Input Specification

The first line of input contains the number of testcases, at most 100.

For each testcase, the number F of flowers and the number P of pots are contained in the first line ($F, P \leq 200$) along with the number C of possible color combinations ($1 \leq C \leq 100$, there are at most 10 different colors). Then follow F lines of flowers with the color c_f of the flower plus the diameter d_f and height h_f of its cachepot (a cachepot is the ugly plastic pot that flowers are sold in). Next come P lines with the color c_p, the diameter d_p and height h_p of each available flower pot. The colors are strings and the dimensions of the cachepots and pots are positive integers not larger than 100. Finally, there are C lines with color combinations that Christian will accept. Each line consists of two colors, where the first one is the color of the flower and the second one is the color of the pot. You may assume that a flower fits into a given pot, if the diameter and height of its cachepot are not larger than the corresponding dimensions of that pot.

Output Specification

Per test case, print one line with the number P_b of pots that Christian has to buy.

Sample Data

Input

```
2
2 2 3
red 10 10
green 10 10
red 12 10
green 15 10
red red
red green
green red
3 2 3
red 10 10
green 10 10
blue 16 10
green 12 10
green 15 10
red green
green red
blue green
```

Output

```
0
2
```

LAST PAGE