558 Wormholes

In the year 2163, wormholes were discovered. A wormhole is a subspace tunnel through space and time connecting two star systems. Wormholes have a few peculiar properties:

- Wormholes are *one-way* only.
- The time it takes to travel through a wormhole is negligible.
- A wormhole has two end points, each situated in a star system.
- A star system may have more than one wormhole end point within its boundaries.
- For some unknown reason, starting from our solar system, it is always possible to end up in any star system by following a sequence of wormholes (maybe Earth is the centre of the universe).
- Between any pair of star systems, there is at most one wormhole in either direction.
- There are no wormholes with both end points in the same star system.

All wormholes have a constant time difference between their end points. For example, a specific wormhole may cause the person travelling through it to end up 15 years in the future. Another wormhole may cause the person to end up 42 years in the past.

A brilliant physicist, living on earth, wants to use wormholes to study the Big Bang. Since warp drive has not been invented yet, it is not possible for her to travel from one star system to another one directly. This *can* be done using wormholes, of course.

The scientist wants to reach a cycle of wormholes somewhere in the universe that causes her to end up in the past. By travelling along this cycle a lot of times, the scientist is able to go back as far in time as necessary to reach the beginning of the universe and see the Big Bang with her own eyes. Write a program to find out whether such a cycle exists.

Input

The input file starts with a line containing the number of cases c to be analysed. Each case starts with a line with two numbers n and m. These indicate the number of star systems ($1 \leq n \leq 1000$) and the number of wormholes ($0 \leq m \leq 2000$). The star systems are numbered from 0 (our solar system) through $n-1$. For each wormhole a line containing three integer numbers x, y and t is given. These numbers indicate that this wormhole allows someone to travel from the star system numbered x to the star system numbered y, thereby ending up t ($-1000 \leq t \leq 1000$) years in the future.

Output

The output consists of c lines, one line for each case, containing the word '`possible`' if it is indeed possible to go back in time indefinitely, or '`not possible`' if this is not possible with the given set of star systems and wormholes.

Sample Input

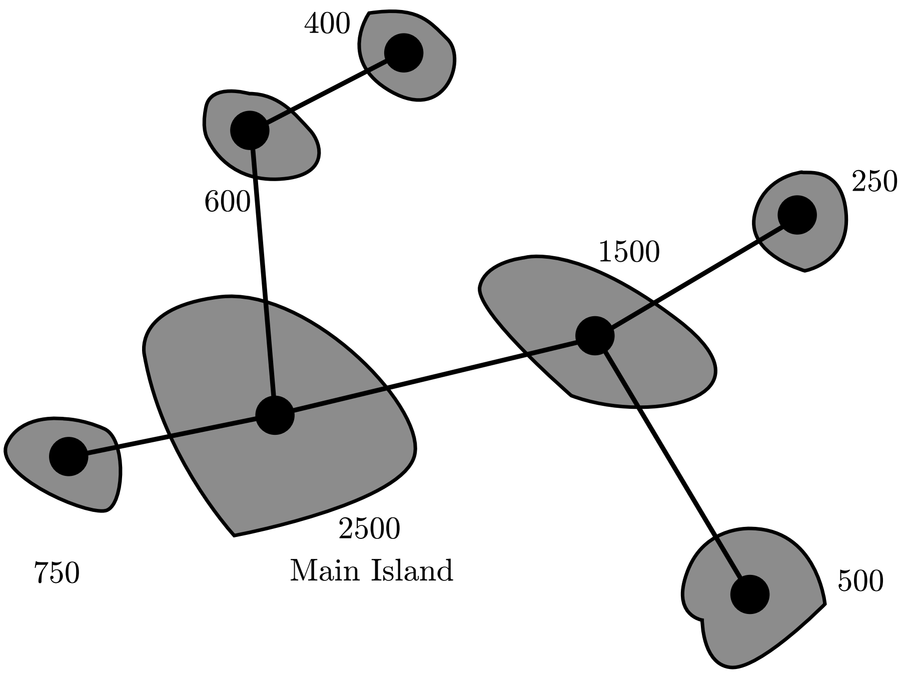
```
2
3 3
0 1 1000
1 2 15
2 1 -42
4 4
0 1 10
1 2 20
2 3 30
3 0 -60
```

Sample Output

```
possible
not possible
```

1013 Island Hopping

The company Pacific Island Net (PIN) has identified several small island groups in the Pacific that do not have a fast internet connection. PIN plans to tap this potential market by offering internet service to the island inhabitants. Each groups of islands already has a deep-sea cable that connects the main island to the closest internet hub on the mainland (be it America, Australia or Asia). All that remains to be done is to connect the islands in a group to each other. You must write a program to help them determine a connection procedure.



For each island, you are given the position of its router and the number of island inhabitants. In the figure, the dark dots are the routers and the numbers are the numbers of inhabitants. PIN will build connections between pairs of routers such that every router has a path to the main island. PIN has decided to build the network such that the total amount of cable used is minimal. Under this restriction, there may be several optimal networks. However, it does not matter to PIN which of the optimal networks is built.

PIN is interested in the average time required for new customers to access the internet, based on the assumption that construction on all cable links in the network begins at the same time. Cable links can be constructed at a rate of one kilometer of cable per day. As a result, shorter cable links are completed before the longer links. An island will have internet access as soon as there is a path from the island to the main island along completed cable links. If m_i is the number of inhabitants of the i^{th} island and t_i is the time when the island is connected to the internet, then the average connection time is:

$$\frac{\sum t_i * m_i}{\sum m_i}$$

Input

The input consists of several descriptions of groups of islands. The first line of each description contains a single positive integer n, the number of islands in the group ($n \leq 50$). Each of the next n lines has three integers x_i, y_i, m_i, giving the position of the router (x_i, y_i) and number of inhabitants m_i ($m_i > 0$) of the islands. Coordinates are measured in kilometers. The first island in this sequence is the main island.

The input is terminated by the number zero on a line by itself.

Output

For each group of islands in the input, output the sequence number of the group and the average number of days until the inhabitants are connected to the internet. The number of days should have two digits to the right of the decimal point. Use the output format in the sample given below.

Place a blank line after the output of each test case.

Sample Input

```
7
11 12 2500
14 17 1500
9 9 750
7 15 600
19 16 500
8 18 400
15 21 250
0
```

Sample Output

```
Island Group: 1  Average 3.20
```

929 Number Maze

Consider a number maze represented as a two dimensional array of numbers comprehended between 0 and 9, as exemplified below. The maze can be traversed following any orthogonal direction (i.e., north, south, east and west). Considering that each cell represents a cost, then finding the minimum cost to travel the maze from one entry point to an exit point may pose you a reasonable challenge.

0	3	1	2	9
7	3	4	9	9
1	7	5	5	3
2	3	4	2	5

Your task is to find the minimum cost value to go from the top-left corner to the bottom-right corner of a given number maze of size $N \times M$ where $1 \leq N, M \leq 999$. Note that the solution for the given example is 24.

Input

The input file contains several mazes. The first input line contains a positive integer defining the number of mazes that follow. Each maze is defined by: one line with the number of rows, N; one line with the number of columns, M; and N lines, one per each row of the maze, containing the maze numbers separated by spaces.

Output

For each maze, output one line with the required minimum value.

Sample Input

```
2
4
5
0 3 1 2 9
7 3 4 9 9
1 7 5 5 3
2 3 4 2 5
1
6
0 1 2 3 4 5
```

Sample Output

```
24
15
```

10397 Connect the Campus

Many new buildings are under construction on the campus of the University of Waterloo. The university has hired bricklayers, electricians, plumbers, and a computer programmer. A computer programmer? Yes, you have been hired to ensure that each building is connected to every other building (directly or indirectly) through the campus network of communication cables.

We will treat each building as a point specified by an x-coordinate and a y-coordinate. Each communication cable connects exactly two buildings, following a straight line between the buildings. Information travels along a cable in both directions. Cables can freely cross each other, but they are only connected together at their endpoints (at buildings).

You have been given a campus map which shows the locations of all buildings and existing communication cables. You must not alter the existing cables. Determine where to install new communication cables so that all buildings are connected. Of course, the university wants you to minimize the amount of new cable that you use.

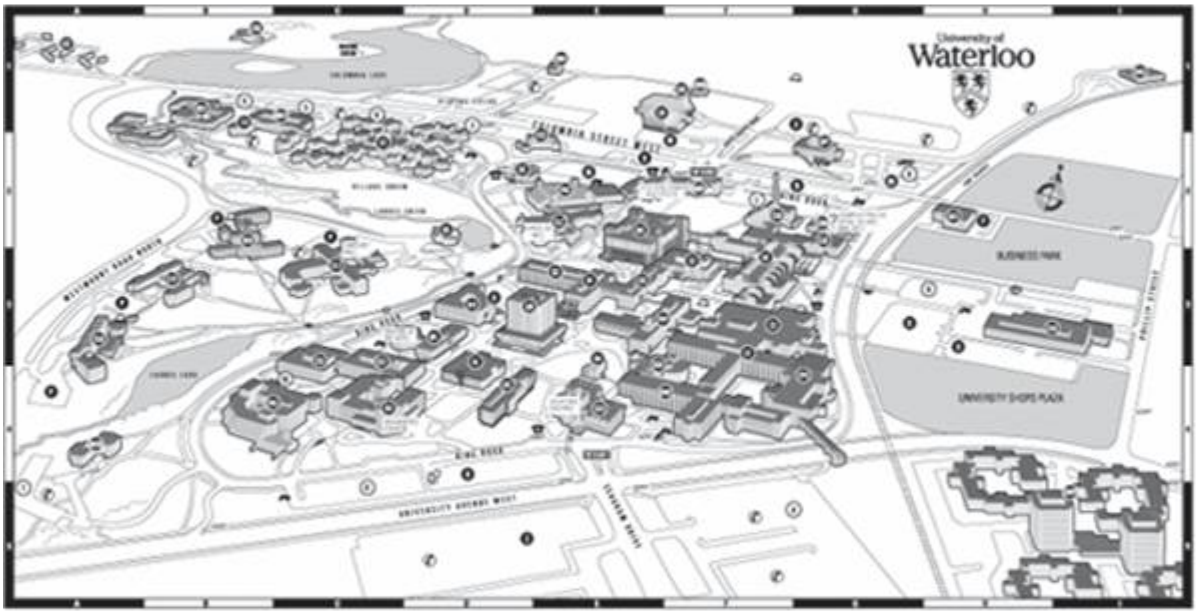


Fig: University of Waterloo Campus

Input

The input file describes several test cases. The description of each test case is given below:

The first line of each test case contains the number of buildings N ($1 \leq N \leq 750$). The buildings are labeled from 1 to N. The next N lines give the x and y coordinates of the buildings. These coordinates are integers with absolute values at most 10000. No two buildings occupy the same point. After that there is a line containing the number of existing cables M ($0 \leq M \leq 1000$) followed by M lines describing the existing cables. Each cable is represented by two integers: the building numbers which are directly connected by the cable. There is at most one cable directly connecting each pair of buildings.

Output

For each set of input, output in a single line the total length of the new cables that you plan to use rounded to two decimal places.

Sample Input

```
4
103 104
104 100
104 103
100 100
1
4 2
4
103 104
104 100
104 103
100 100
1
4 2
```

Sample Output

```
4.41
4.41
```

11367 Full Tank?

After going through the receipts from your car trip through Europe this summer, you realised that the gas prices varied between the cities you visited. Maybe you could have saved some money if you were a bit more clever about where you filled your fuel?

To help other tourists (and save money yourself next time), you want to write a program for finding the cheapest way to travel between cities, filling your tank on the way. We assume that all cars use one unit of fuel per unit of distance, and start with an empty gas tank.

Input

The first line of input gives $1 \leq n \leq 1000$ and $0 \leq m \leq 10000$, the number of cities and roads. Then follows a line with n integers $1 \leq p_i \leq 100$, where p_i is the fuel price in the ith city. Then follow m lines with three integers $0 \leq u$, $v < n$ and $1 \leq d \leq 100$, telling that there is a road between u and v with length d. Then comes a line with the number $1 \leq q \leq 100$, giving the number of queries, and q lines with three integers $1 \leq c \leq 100$, s and e, where c is the fuel capacity of the vehicle, s is the starting city, and e is the goal.

Output

For each query, output the price of the cheapest trip from s to e using a car with the given capacity, or '`impossible`' if there is no way of getting from s to e with the given car.

Sample Input

```
5 5
10 10 20 12 13
0 1 9
0 2 8
1 2 1
1 3 11
2 3 7
2
10 0 3
20 1 4
```

Sample Output

```
170
impossible
```

Reachable Roads

After the major earthquake and fires to your town, planners are assessing the damages to the roads in the city. Some roads are just gone, and others are impassable. The result is that it is impossible to get from some places to others. The planners want to get the city running quickly. Therefore they want to add as few roads as possible to get a city where everyone can reach everyone else.

We'll make the simplifying assumptions that building any road is the same cost as building any other road, and that all roads are bidirectional.

The planners have asked you to find the minimum number of roads that they can add to the current grid so that everyone can get everywhere.

Input

Input starts with a line containing an integer $1 \le n \le 100$ indicating the number of cities that follow. Each city's description starts with a line describing the number of road endpoints $1 \le m \le 1000$. Each endpoint is in the range 0 to m - 1. This is followed by a number $0 \le r \le \min(m(m-1)/2, 10m)$, then a list of r pairs of endpoints which are directly connected by a usable road. Each pair appears on its own line, all pairs are distinct, and all roads connect to different endpoints.

Output

For each city description, output the minimum number of roads that must be added.

Sample Input

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

Sample Output 1

```
1
0
```