Árboles y Grafos, 2025-1

Para entregar el viernes 11 de abril de 2025
A las 23:59 en la arena de programación

Instrucciones para la entrega

- Para esta tarea y todas las tareas futuras en la arena de programación, la entrega de soluciones es *individual*. Por favor escriba claramente su nombre, código de estudiante y sección en cada archivo de código (a modo de comentario). Adicionalmente, cite cualquier fuente de información que utilizó. Los códigos fuente que suba a la arena de programación deben der de su completa autoría.
- En cada problema debe leer los datos de entrada de la forma en la que se indica en el enunciado y debe imprimir los resultados con el formato allí indicado. No debe agregar mensajes ni agregar o eliminar datos en el proceso de lectura. La omisión de esta indicación puede generar que su programa no sea aceptado en la arena de programación.
- Puede resolver los ejercicios en C/C++ y Python. Sin embargo, deben haber por los menos soluciones a dos problemas en cada lenguaje.
- Debe enviar sus soluciones a través de la arena. Antes de subir sus soluciones asegurese de realizar pruebas con los casos de pruebas proporcionados para verificar que el programa es correcto, que finaliza y no se quede en un ciclo infinito.
- El primer criterío de evaluación será la aceptación del problema en la arena cumpliendo los requisitos indicados en los enunciados de los ejercicios y en este documento. El segundo criterio de evaluación será la complejidad computacional de la solución y el uso de los temas vistos en clase. Es necesario incluir en la cabecera del archivo comentarios que expliquen la complejidad de la solución del problema para cada caso.

  En adición a lo anterior, para efectos de la calificación se tendrán en cuenta aspectos de estilo como no usar `break` ni `continue` y que las funciones deben tener únicamente una instrucción `return` que debe estar en la última línea.

Problemas prácticos

Hay cinco problemas prácticos cuyos enunciados aparecen a partir de la siguiente página. El problema restante se calificará con **0.5 décimas de bonificación**. En caso de que la nota total de la tarea supere 5.0 el excedente será puesto en otra tarea del curso.

A - Problem A

Source file name: `cards.cpp`
Time limit: 3 seconds

In the age of television, not many people attend theater performances. Antique Comedians of Malidinesia are aware of this fact. They want to propagate theater and, most of all, Antique Comedies. They have printed invitation cards with all the necessary information and with the programme. A lot of students were hired to distribute these invitations among the people. Each student volunteer has assigned exactly one bus stop and he or she stays there the whole day and gives invitation to people travelling by bus. A special course was taken where students learned how to influence people and what is the difference between influencing and robbery.

The transport system is very special: all lines are unidirectional and connect exactly two stops. Buses leave the originating stop with passengers each half an hour. After reaching the destination stop they return empty to the originating stop, where they wait until the next full half an hour, e.g. X:00 or X:30, where 'X' denotes the hour. The fee for transport between two stops is given by special tables and is payable on the spot. The lines are planned in such a way, that each round trip (i.e., a journey starting and finishing at the same stop) passes through a Central Checkpoint Stop (CCS) where each passenger has to pass a thorough check including body scan.

All the ACM student members leave the CCS each morning. Each volunteer is to move to one predetermined stop to invite passengers. There are as many volunteers as stops. At the end of the day, all students travel back to CCS. You are to write a computer program that helps ACM to minimize the amount of money to pay every day for the transport of their employees.

Input

The input consists of N cases. The first line of the input contains only positive integer N. Then follow the cases. Each case begins with a line containing exactly two integers P and Q, $1 \leq P, Q \leq 100000$. P is the number of stops including CCS and Q the number of bus lines. Then there are Q lines, each describing one bus line. Each of the lines contains exactly three numbers —the originating stop, the destination stop, and the price. The CCS is designated by number 1. Prices are positive integers the sum of which is smaller than 1000000000. You can also assume it is always possible to get from any stop to any other stop.

The input must be read from standard input.

Output

For each case, print one line containing the minimum amount of money to be paid each day by ACM for the travel costs of its volunteers.

The output must be written to standard output.

Sample Input	**Sample Output**
`2`	`46`
`2 2`	`210`
`1 2 13`	
`2 1 33`	
`4 6`	
`1 2 10`	
`2 1 60`	
`1 3 20`	
`3 4 10`	
`2 4 5`	
`4 1 50`	

B - Problem B

Source file name: `escape.cpp`
Time limit: 1 second

Poor Kianoosh! 6 weeks since he was officially accepted as a citizen of Barareh and during this time he has had nothing but Nochophskew for his breakfast, lunch and dinner. And nowadays, during the nights, he dreams of nothing but a small piece of bread, some cheese and a glass of milk to have for his breakfast. And during the days, he thinks of nothing but a great escape from this weird city in order to turn his night-time dreams into reality.

After carefully checking all the potential ways out of the city, he concluded that there are only three such possible ways. The first one is through the city gate which is of course guarded by fully-armed guards, ready to kill any creature trying to go through the gate without permission. The second way is passing through the northern walls of the city and going through a jungle which contains nothing but those deadly snakes called NeshimanGazes (And be sure that Kianoosh does not like to repeat his creepy experience with them again).

Finally, the third and the last (and of course the only feasible) way is passing through a secret door in the famous so called HezarDaroon Castle (means a castle with thousands of doors). Unfortunately, the problem with this castle is that no one actually knows the way to reach this secret door.

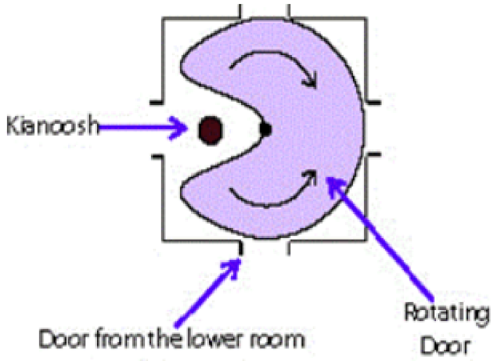


Figure 1. A Room containing a Rotating Door

After spending some time gathering information about the castle, Kianoosh found out that HezarDaroon can actually be modeled as an m×n grid with each cell representing a room in the castle. There is always a door between two neighboring rooms but there are some rooms which contain a NeshimanGaz ready to serve anyone trying to enter the room with its poisonous bites.

There are also some rooms which contain a rotating door. A rotating door is actually a cylinder positioned at the center of the room with a slice removed from it. In Figure 1, you can see a room with a rotating door with its removed slice facing towards west.

As it can be seen, one can only enter the room from the neighbor to which the removed slice is facing. And once Kianoosh enters the room, he can rotate the door clockwise or counter-clockwise in order to reach the other doors in the room. For example, in the configuration of figure 1, if he rotates the door clockwise by 90 degrees, he can reach the door to the north and if he continues rotating the door by another 90 degrees in the same direction, he can reach the door to east.

Another important fact about the rotating doors is that associated to each door is a number *d* which is the amount of time, measured in seconds, that it takes for a normal human-being(like Kianoosh) to rotate the door by 90 degrees. You may also assume that going from a room to its neighbor room takes exactly 1 second.

Today, after spending a noticeable amount of money, Kianoosh managed to buy a map of the castle from underground shops of Barareh. Taking a look at the map, he found that the entrance to the castle is located at the lower-left corner and the secret door is located at the upper-right corner of the grid. He can also see the position and the initial configuration of the rotating doors and the coordinates of rooms containing the deadly NeshimanGazes to which Kianoosh does not like to enter.

Kianoosh is happy. He is only one step away from his dreams coming true. But how can he find his way to the secret door?

Yes! He needs help! And as a genius problem-solver, you should help him find the path with the minimum required time to reach the secret passage from the lower-left corner of the castle.

Input

The first line of the input contains an integer T which is the number of test cases. Each test case begins with a line containing $1 \leq n \leq 150$ (the number of rows) and $1 \leq m \leq 150$ (the number of columns) followed by n lines, with the i-th line containing m characters, the j-th character of which representing the status of the j-th cell in the i-th row of the grid (counted from north). Each character may have six different values. A `'.'` character represents a normal room while a `'#'` represents a room containing a NeshimanGaz and `'N'`, `'W'`, `'E'`, `'S'` represent a room with a rotating door facing north, west, east, and south respectively. The map is followed by D integers, the k-th of which is equal to d_k for the k-th door, counting the doors from the north-west corner in a row major order. You can assume that D (which is the number of rotating doors in the castle) is no more than 500, and the entrance (room at lower-left corner) and the room containing the secret door (room at upper-right corner) does not contain NeshimanGazes or rotating doors.

The input must be read from standard input.

Output

For each test case, your program should output a line containing a single integer which is the minimum amount of time required for Kianoosh to reach the secret door. In the case the secret door is not reachable with respect to the given map and rules, a line containing the phrase `'Poor Kianoosh'` should be printed.

The output must be written to standard output.

Sample Input	**Sample Output**

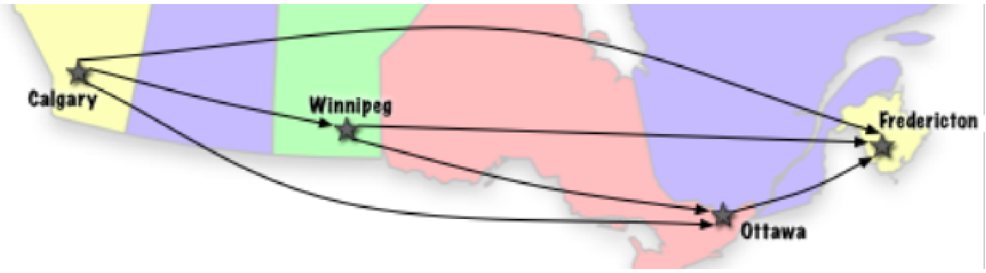
`1`
`3 3`
`.#.`
`..W`
`...`
`10`
	`14`

C - Problem C

Source file name: `fredericton.cpp`
Time limit: 3 seconds

After being inspired by Salvador Dali's artwork, Brett decided he would like to travel to Fredericton, New Brunswick to visit the Beaverbrook Art Gallery.

Brett lives in Calgary, and would like to find the cheapest flight or flights that would take him to Fredericton. He knows that a direct flight from Calgary to Fredericton, if one exists, would be absurdly expensive, so he is willing to tolerate a certain number of stopovers. However, there are multiple airlines in Canada with so many different flights between different cities now, which makes it very difficult for Brett to find the least expensive way to Fredericton! Can you write a program to help Brett plan his route?



Map showing a sample of the flights that would take Brett to Fredericton.

You will be given a list of cities between and including Calgary and Fredericton. The cities will be given in order of "nearest" to "farthest". The first city will always be "Calgary" and the last "Fredericton".

You will also be given a list of flights between pairs of cities, and the associated cost for each flight, taxes included. There will never be a flight from a farther city to a nearer city — Brett has already discarded those flights, deeming them to be a waste of time and money. Bear in mind, however, that there may be more than one flight between any two cities, as Brett is considering flights from all airlines.

Finally, you are presented with a number of queries. Each query is a single integer indicating the maximum number of stopovers Brett will tolerate. For each query, your program must calculate the least total cost of flights that would take Brett from Calgary to Fredericton with no more than the requested number of stopovers.

Input

The first line of input contains a single number indicating the number of scenarios to process. A blank line precedes each scenario.

Each scenario begins with a number N ($2 \leq N \leq 250$), the number of cities, followed by N lines containing the names of the cities. A city name is a string of up to 20 uppercase and lowercase letters. Next is a number M ($0 \leq M \leq 10000$), the number of flights available, followed by M lines describing the flights. Each flight is described by its departure city, its destination city, and an integer representing its cost in dollars. The final line starts with Q ($1 \leq Q \leq 50$), the number of queries, followed by Q numbers indicating the maximum number of stopovers.

The input must be read from standard input.

Output

For each scenario, your program should output the scenario number, followed by the least total cost of the flights for each query. Follow the format of the sample output. If no flights can satisfy a query, write `'No satisfactory flights'`. Output a blank line between scenarios.

The output must be written to standard output.

Sample Input

```
2

4
Calgary
Winnipeg
Ottawa
Fredericton
6
Calgary Winnipeg 125
Calgary Ottawa 300
Winnipeg Fredericton 325
Winnipeg Ottawa 100
Calgary Fredericton 875
Ottawa Fredericton 175
3 2 1 0

3
Calgary
Montreal
Fredericton
2
Calgary Montreal 300
Montreal Fredericton 325
1 0
```

Sample Output

```
Scenario #1
Total cost of flight(s) is $400
Total cost of flight(s) is $450
Total cost of flight(s) is $875

Scenario #2
No satisfactory flights
```

D - Problem D

Source file name: `krochanska.cpp`
Time limit: 15 seconds

This is a problem about spies and counter-spies in the old days of the Iron Curtain. CONTROL, the secret intelligence agency of the free world, must fight against KAOS, the international organization of evil.

CONTROL agents 82 through 85 have all diedattempting to deliver the payroll for Control's freelance agents located behind the Iron Curtain. They all were killed by the mysterious and sinister KAOS agent Cirilo Krochanska, while traveling aboard the Orient Express. You are Maxwell Smart, agent 86, and you must board the train, make contact with agent B-12, give him the encrypted message "tnih sevig cilc thgir", deliver the payroll, and avoid becoming Krochanska's fifth victim!

But, where is Krochanska? We know he always travel by train; he is in some important train station in Europe, and is ready to reach immediately any destination where he is required.

The railway network of Europe consists of a number of lines. Each line goes between two different stations; and there are also some intermediate stations uniformly distributed in each line. For example, if we enumerate the stations from 1 to 13, we can have the following three lines:

- **Line 1**. 1 - 2 - 3 - 4 - 5 - 6 - 7.
- **Line 2**. 8 - 9 - 4 - 10 - 13.
- **Line 3**. 11 - 2 - 12 - 9 - 6 - 7.

Trains travel in both directions of the lines. The time to travel from a station to the following (or the previous) station of the line is always 2 hours. As you can observe, some stations are used by different lines —we call them the *important stations*—, while other stations are only used by one line —*the secondary stations*—.

We believe Krochanska is situated in an important station where he can travel faster to any other station. In particular, he tries to minimize the average of the minimum times from his station to the rest of important stations. You can assume that there is no loss of time to switch from one line to another at a station.

You have to find out where Cirilo Krochanska is.

Input

The input will contain several test cases. The first line indicates the number of test cases.

For each test case, the first line contains two integers: N and S. N is the total number of existing stations, and S is the number of lines. Stations are numbered from 1 to N; N is not greater than 10000; and S is not greater than 100. Next, we have S lines, one for each train line. These lines consist of a list of stations, separated by blank spaces, ending with a `'0'`.

There will be between 1 and 100 important stations, inclusive. There is always a path between any pair of stations.

The input must be read from standard input.

Output

For each test case, you have to output the number of the resulting station, in the following format:

`Krochanska is in:` X

where X is the number of the station. If there are more than one important station with the minimum distance, then you have to output the one with the smallest number.

The output must be written to standard output.

Sample Input	Sample Output

4
13 3
1 2 3 4 5 6 7 0
8 9 4 10 13 0
11 2 12 9 6 7 0
6 2
2 5 3 6 1 4 0
4 1 6 3 5 2 0
5 2
1 2 3 4 5 0
3 5 1 4 2 0
7 2
3 5 1 2 4 7 6 0
3 6 1 0
	Krochanska is in: 9
	Krochanska is in: 3
	Krochanska is in: 4
	Krochanska is in: 6

E - Problem E

Source file name: `namek.cpp`
Time limit: 3 seconds

The year is 3471, and Earth's space technology has advanced significantly over the past centuries. Zlatan XV, better known simply as Zlatan, led the team that developed the technology that made it possible to achieve the first space travels at speeds close to the speed of light. These journeys have allowed humanity to explore planets and even galaxies that once seemed unreachable.

During a recent expedition, a group of astronauts was sent to a planet called Namekusei, where they met friendly and kind humanoids with green skin. These beings allowed the humans to explore their planet and conduct studies on some of their plants and materials. During this exploration, the astronauts discovered a strange pink, viscous substance that seemed to have a life of its own, as it moved from one place to another. After isolating a small sample of this substance, the astronauts decided it would be best to conduct further analysis on Earth. They asked the Namekuseijin for permission to take the substance back to Earth. The Namekuseijin agreed but requested that the astronauts promise not to use their discoveries for evil purposes and warned them that the substance could be dangerous, so they had to transport it at their own risk.

After traveling for several months at a speed close to that of light, the sample of the pink substance was brought to Earth. Scientists conducted various analyses and experiments, which allowed them to better understand the physical and chemical composition of this strange material. They made several important discoveries. Scientists found that the substance consists of atoms that are bonded together to form molecules. The atoms of the pink substance exert forces of attraction or repulsion on other atoms. These forces cause atoms to transmit or receive energy. A bond forms between two atoms that either attract or repel each other directly. It is possible for an atom a to exert an attractive force on an atom b while atom b exerts a repulsive force on atom a.

For energy movement to occur between two atoms, they must attract or repel each other either directly (meaning there is a bond between them) or indirectly through other atoms that are bonded to them. Two atoms a and b are said to be connected if there is energy movement from a to b and vice versa. A molecule is defined as a group of atoms such that all the atoms in the molecule are connected. It is impossible for two atoms in different molecules to be connected, although energy may move from an atom in one molecule to an atom in another molecule. A molecular chain is a sequence of molecules $M_1, M_2, \ldots, M_k$ such that there is a movement of energy from M_1 to M_2, from M_2 to M_3, and so on.

Scientists have determined that it is impossible to separate the atoms within a molecule, but it is possible to isolate groups of molecules. The separation of molecules depends on the energy each molecule can store. The energy a molecule can store is the maximum amount of energy that can move between any pair of atoms in the molecule. The amount of energy that can move between any pair of atoms is determined by the sequence of bonds between them, in which the minimum energy transfer occurs. To separate two molecules, it is necessary to alter the bond between them by applying an amount of energy greater than the sum of the energies that both molecules can store.

To continue their experiments, scientists need to isolate the molecular sequence that starts from the molecule containing a given atom a and ends in the molecule containing a given atom b, as they suspect that this molecular sequence will reveal many insights about the functioning of the pink substance.

Your task is to determine the minimum amount of energy required to isolate the molecular sequence that the scientists are interested in.

Input

The input consists of T test cases. The first line contains the integer T, representing the number of test cases. For each test case:

The first line of each test case contains an integer n ($1 \leq n \leq 2000$), which represents the number of atoms. The atoms are numbered from 0 to $n - 1$. The second line contains two integers a and b, which correspond to the atoms in the molecules where the molecular sequence to be isolated begins and ends. The following lines contain three integers a, b, c ($0 \leq a, b < n, -10000 \leq c \leq 10000$), indicating that there is a bond between atoms a and b, and the energy flowing between them is c.

- If $c > 0$, the atoms a, b attract each other.
- If $c < 0$, the atoms a, b repel each other.

The input for a test case ends when the values $a = b = -1$ appear.

The input must be read from standard input.

Output

For each test case, print the amount of energy required to isolate the molecular sequence associated with the given atoms.

The output must be written to standard output.

Sample Input	Sample Output

1
18
10 15
16 14 3
0 1 5
1 4 -2
10 7 1
1 2 2
14 17 -1
8 6 3
2 3 1
6 5 2
9 10 6
15 16 8
6 14 1
3 4 1
2 0 -3
17 15 4
4 0 -1
11 13 1
4 5 4
13 12 -2
5 6 6
11 6 4
7 8 3
14 15 5
12 11 5
8 9 2
-1 -1 -1
	26