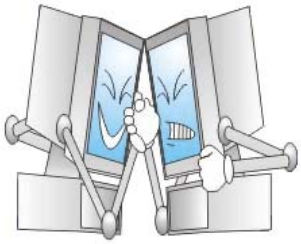

Pripreme za IOI 2008
7. avgust 2008. godine
predavač: Stevan Jončić
e-mail: stevanjoncic@yahoo.com

Razni zadaci

Zadatak 1. Jingjing the panda lives in a forest containing n pieces of bamboo land. Each bamboo land is very small and can be regarded as a single point. Bamboo land i contains L_i bamboos and is associated with a ``deliciousness'' W_i .

Jingjing eats all bamboos in a selected bamboo land every day. He has a bad habit: the deliciousness of the bamboo land he selects must be strictly larger than that of the day before.

Moving from one land to another is very tiring. The longer Jingjing walks before arriving a bamboo land i , the more bamboo he is expecting. If the distance he walked from the last bamboo land is strictly larger than the number of bamboos he finds in the current land (i.e L_i), he will die of sadness.

Distance of two points (x_0, y_0) and (x_1, y_1) equals to $|x_0 - x_1| + |y_0 - y_1|$, since Jingjing only moves north, south, east and west.

When you send Jingjing in one bamboo land someday, how many days can Jingjing survive (Jingjing is clever enough to find out the optimal way of living)?

We need this information so that we can bring him out before he dies.

Input

The first line contains a single integer t ($1 \leq t \leq 10$), the number of test cases. Each test case contains several lines. The first line contains a single integer n ($1 \leq n \leq 100,000$), the number of bamboo lands. The next n lines each contains 4 integers X_i , Y_i , W_i ,L_i , indicating the coordinate of i -th bamboo land, its deliciousness and number of bamboos.

You may assume that $0 \leq X_i, Y_i, W_i, L_i \leq 1,000,000$. No two lands have the same deliciousness. Two bamboo lands can be so close that they can be regarded as at the same point.

Output

For each test case, print the case number followed by the number of days Jingjing can survive. Look at the output for sample input for details.

Input	**Output**
2 3 0 0 3 4 2 2 2 3 5 5 1 3 3 0 0 3 4	Case 1: 2 Case 2: 2

2 2 2 3 5 5 1 3	

Zadatak 2. The terrorist group leaded by a well known international terrorist Ben Bladen is buliding a nuclear reactor to produce plutonium for the nuclear bomb they are planning to create. Being the wicked computer genius of this group, you are responsible for developing the cooling system for the reactor.

The cooling system of the reactor consists of the number of pipes that special cooling liquid flows by. Pipes are connected at special points, called nodes, each pipe has the starting node and the end point. The liquid must flow by the pipe from its start point to its end point and not in the opposite direction.

Let the nodes be numbered from 1 to N. The cooling system must be designed so that the liquid is circulating by the pipes and the amount of the liquid coming to each node (in the unit of time) is equal to the amount of liquid leaving the node. That is, if we designate the amount of liquid going by the pipe from i-th node to j-th as f_{ij}, (put $f_{ij} = 0$ if there is no pipe from node i to node j), for each i the following condition must hold:

$$\text{sum}(j=1..N, f_{ij}) = \text{sum}(j=1..N, f_{ji})$$

Each pipe has some finite capacity, therefore for each i and j connected by the pipe must be $f_{ij} \le c_{ij}$ where c_{ij} is the capacity of the pipe. To provide sufficient cooling, the amount of the liquid flowing by the pipe going from i-th to j-th nodes must be at least l_{ij}, thus it must be $f_{ij} \ge l_{ij}$.

Given c_{ij} and l_{ij} for all pipes, find the amount f_{ij}, satisfying the conditions specified above.

Input

The first line of the input file contains the number N ($1 \le N \le 200$) - the number of nodes and and M — the number of pipes. The following M lines contain four integer number each - i, j, l_{ij} and c_{ij} each. There is at most one pipe connecting any two nodes and $0 \le l_{ij} \le c_{ij} \le 10^5$ for all pipes. No pipe connects a node to itself. If there is a pipe from i-th node to j-th, there is no pipe from j-th node to i-th.

Output

On the first line of the output file print YES if there is the way to carry out reactor cooling and NO if there is none. In the first case M integers must follow, k-th number being the amount of liquid flowing by the k-th pipe. Pipes are numbered as they are given in the input file.

Input	**Output**
4 6 1 2 1 2 2 3 1 2 3 4 1 2 4 1 1 2 1 3 1 2 4 2 1 2	NO

Input	**Output**
4 6 1 2 1 3 2 3 1 3 3 4 1 3 4 1 1 3 1 3 1 3	YES 1 2 3 2 1

4 2 1 3	1

Zadatak 3. Almost all the TJU ACMers are good at computer games. During the long and boring summer, they want to play some games to relax. But their levels are quite different, for example, wtommy can beat others easily in the game StarCraft. They want the game to be more attractive, so every time they choose two members, whose levels are nearest, to fight. For there are so many people join and leave the team, this task turned to be very difficult. Now they are asking you for help.

Input

The first line of each test case contain a number M ($1 \le M \le 10^5$) indicating the number of commands. Each of The following M lines is one command. There are three types of commands:

- Join SS P : A new member called SS has joined the TJU ACM team. His game level is P. ($0 < P < 10^8$)
- Leave SS : The member SS has left the team.
- Play : oh yeah , they want to choose two guys to fight.

You can assume all the names is made up of less than 20 letters. All the people have different levels and different names. All the people will join and leave the team at most once. The input is terminated by a test case starting with $M = 0$. This test case should not be processed.

Output

Your program should respond to each 'Play' command , output one line containing the two names chosen to fight or "Poor Mr.Yu" if there are less than two members in the team. Please note the one with the higher level will come first. If there are more than one pair whose levels are nearest, you should output the pair with highest level, because people think the fight between higher levels will be more attractive. You should print a blank line after each test case.

Input	**Output**
8 Join WTommy 80 Join RoBa 50 Play Join Washington 65 Play Leave WTommy Leave RoBa Play 0	WTommy RoBa WTommy Washington Poor Mr.Yu

Zadatak 4. Dat je niz od n prirodnih brojeva P tako da je svaki broj veći ili jednak X a manji ili jednak Y. Potrebno je napraviti novi niz G koji je strogo rastući tako da suma abs(P[1]-G[1])+abs(P[2]-G[2])+...+(P[n]-G[n]) bude minimalna. Složenost algoritma je O(n*(Y-X))

Zadatak 5. Data je neki plac u obliku matrice n * m (n <= 50 , m <= 50) . Neka polja su prekrivena travom a neka blatom. Date su i daske sirine 1 i proizvoljne duzine. Potrebno je postaviti minimalni broj dasaka paralelno sa stranama placa taka da sva polja koja su blatnjava budu prekrivema. Jedno polje moze da bude prekriveno od strane vise dasaka.

Zadatak 6. Mirko radi na svinjogojskoj farmi koja se sastoji od M **zaključanih** svinjaca, a Mirko **nema** niti jedan ključ. Kupci dolaze na farmu jedan za drugim. Svaki od njih ima ključeve

nekih svinjaca i želi kupiti određeni broj svinja. Mirko ima na raspologanju sve podatke o svim kupcima koji će toga dana doći na farmu i rano ujutroradi raspored prodaje svinja kako bi ih taj dan prodao **što više**.

Preciznije, procedura kupnje svinja je sljedeća: kupac dolazi, otvara sve svinjce od kojih ima ključ, Mirko mu prodaje određeni broj svinja iz otključanih svinjaca, eventualno **preraspoređuje** preostale svinje po tim otključanim svinjcima, kupac zaključava svinjce i odlazi sa svojim svinjama.Broj svinja koje se mogu nalaziti u nekom svinjcu **nije** ograničen. Napišite program koji će odrediti **maksimalni** broj svinja koje Mirko može prodati.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se dva cijela broja M i N, $1 \leq M \leq 1000$, $1 \leq N \leq 100$, broj svinjaca i broj kupaca. Svinjci su označeni brojevima od 1 do M, a kupci brojevima od 1 do N. U sljedećem retku nalazi se M cijelih brojeva, za svaki svinjac redom broj svinja u njemu na početku radnog dana. Broj svinja u svakom svinjcu je veći ili jednak od 0 i manji ili jednak od 1000. U svakom od sljedećih N redaka redom su zapisani podaci o kupcima u sljedećem obliku (podaci o itom kupcu su zapisani u (i+2)-om retku): A K1 K2 ... KA B

To znači da taj kupac ima ključeve od svinjaca označenih brojevima K1, K2, ..., KA (koji su uzlazno sortirani) i da želi kupiti B svinja. Brojevi A i B mogu biti jednaki 0.

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba zapisati ukupni broj prodanih svinja.

Input	**Output**
3 3 3 1 10 2 1 2 2 2 1 3 3 1 2 6	7

Input	**Output**
6 6 6 3 2 0 1 3 2 1 2 0 1 3 3 1 1 1 2 2 3 8 2 4 5 2 2 4 6 6	15

Zadatak 7. The Interpeninsular Olympiad in Informatics is coming and the leaders of the Balkan Peninsula Team have to choose the best contestants on the Balkans. Fortunately, the leaders could choose the members of the team among N very good contestants, numbered from 1 to N ($3 \leq N \leq 500000$). In order to select the best contestants the leaders organized three competitions. Each of the N contestants took part in all three competitions and there were no two contestants with equal results on any of the competitions. We say that contestant A is *better* than another contestant B when A is ranked before B in all of the competitions. A contestant A is said to be *excellent* if no other contestant is better than A. The leaders of the Balkan Peninsula Team would like to know the number of excellent contestants.

Write a program named **TEAM**, which for given *N* and the three competitions results, computes the number of excellent contestants.

The input data are given on the **standard input** as four lines. The first line contains the number *N*. The next three lines show the rankings for the three competitions. Each of these lines contains the identification numbers of the contestants, separated by single spaces, in the order of their ranking from first to last place.

The **standard output** should contain one line with a single number written on it: the number of the excellent.

Input	**Output**
3 2 3 1 3 1 2 1 2 3 *Note: No contestant is better than any other contestant, hence all three are excellent.*	3

Input	**Output**
10 2 5 3 8 10 7 1 6 9 4 1 2 3 4 5 6 7 8 9 10 3 8 7 10 5 4 1 2 6 9 *Note: The excellent contestants are those numbered with 1, 2, 3 and 5.*	4

Zadatak 8. People of country T-land lived on the big plain many years ago. It is happened so that they started to quarrel, so they began to build walls to separate from each other.

One day they realized that walls surround some part of the country. Your task is to determine which wall was build first to surround a part of the T-land.

Input

The first line of input contains one number M (1<=M<=200000) - number of walls. Each of the following M lines contains four integer numbers: Cartesian coordinates of two ends of each wall. Walls are rectilinear segments with positive length, two walls can cross only by ends, and walls can't coincide. All coordinates do not exceed 10^9 by its absolute values.

Output

Write the answer in the single line of output. If all the walls leave the territory opened, write 0.

Input	**Output**
4 0 0 1 0 0 1 0 0 1 0 0 1 2 2 5 7	3

Zadatak 9. All programmers of *Mocrosoft* software company are organized in a strict subordination hierarchy. Every programmer has exactly one chief, except Bill Hates who is also the head of the company and has no chief.

Due to the celebration of the new 2003 year, chief accountant of *Mocrosoft* decided to pay a New Year Bonus Grant of 1000 dollars to some programmers. However being extremely concerned of the company wealth she would like to designate the least possible amount of money for these grants. On the other hand she didn't want to be accused of being too greedy or of giving preferences to some programmers. To do this, she developed the following scheme of grants appointment:

Each programmer may either assign a grant to one of his subordinates or have a grant assigned to him by his chief or none of the above. No programmer can simultaneously receive a grant and assign a grant to one of his subordinates. No programmer can assign a grant to more than one of his subordinates

The scheme seemed to be designed perfectly — nobody would like to assign a grant to anybody since in this case he himself would not receive money. But programmers somehow discovered the plan of chief accountant and decided to make a trick to get the most money possible and share them fairly afterwards. The idea was to make such grant assignments that the total amount of grant money received is maximum possible.

You were selected to write the program which will find the optimal grants appointment.

Input

The first line of the input file contains integer N — the number of programmers in *Mocrosoft* company ($2 \le N \le 500\ 000$). Each programmer is assigned his unique identifier — integer number ranging from 1 to N. Bill Hates has number 1 and each programmer has the number greater then the number of his chief. The second line of the input file contains N-1 integers, i-th of which being the number of the chief of the worker whose number is (i + 1).

Output

On the first line of the output file print the maximum possible amount of money workers can get. On the second line output the numbers of programmers that will receive grant in ascending order.

Input	**Output**
4 1 1 2	2000 3 4