

Problem A Smart Agent

There is a very secret base in Potatoland where potato mash is made according to a special recipe. The neighbors from Porridgia decided to seize this recipe and to sell it to Pilauland. For this mission they have been preparing special agent Pearlo for many years. When, finally, Pearlo learned all secrets of espionage, he penetrated into the Potatoland territory and reached the secret base.

Now he is standing at the entrance, but to get inside he needs to pass combination lock. A minute ago, one of the workers entered the password on the terminal and opened the door. The terminal is a square digital keyboard 3×3 with digits from 1 to 9.

Pearlo knows that the password consists from **distinct digits** and is probably symmetric with respect to the central button of the terminal. He has heat sensor which allowed him to detect the digits which the worker pressed. Now he wants to check whether the password entered by the worker is symmetric with respect to the central button of the terminal. This fact can Help Pearlo to reduce the number of different possible password combinations.

Input

First line of the input, contains a single integer t , denoting number of test cases.

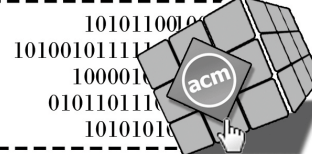
Input for each test case, contains the matrix of three rows of three symbols each. Symbol «X» means that the corresponding button was pressed, and «.» means that it was not pressed. The matrix may contain no «X», also it may contain no «.».

Output

For each test case, print YES if the password is symmetric with respect to the central button of the terminal and NO otherwise, each in a separate line.

Sample Test

Input	Output
2	YES
XX.	NO
...	
.XX	
X.X	
X..	
...	



Problem B Palindromic Times

Hassan is asleep if and only if Hassan is attending a lecture. This is a well-known formula among Hassan's colleagues.

On a Wednesday afternoon, Hassan was attending Dr. Ardeshir's lecture. At 12:21, right before falling asleep, he was staring at the digital watch around Masoud's wrist. He noticed that the digits on the clock were the same when read from both directions i.e. a palindrome.

In his sleep, he started dreaming about such rare moments of the day when the time displayed on a digital clock is a palindrome. As soon as he woke up, he felt destined to write a program that finds the next such moment.

However, he still hasn't mastered the skill of programming while sleeping, so your task is to help him.

Input

First line of the input, contains a single integer t , denoting number of test cases.

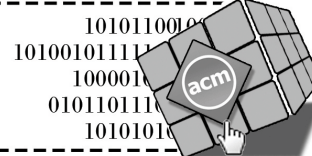
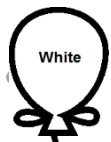
For each test case, there is a line starts with a string with the format "HH:MM" where "HH" is from "00" to "23" and "MM" is from "00" to "59". Both "HH" and "MM" have exactly two digits.

Output

For each test case, print a line contains the palindromic time of day that comes soonest after the time given in the corresponding input. If the input time is palindromic, output the soonest palindromic time after the input time.

Sample Test

Input	Output
2	13:31
12:21	22:22
21:59	



Problem C

Fruit Ninja

Almost all of you are familiar with the famous game “Fruit Ninja”. This time a ninja comes to buy some fruits to play with. He has two bags with capacity of a and b fruits (first bag can be loaded with at most a fruits and the second bag with at most b fruits). There are n fruits with different weights of w_i ($1 \leq i \leq n$) and $a + b = n$. As ninjas must be so fast in moving, our ninja **wants to load all of these n fruits** in his bags so that the sum of average weights of two bags minimizes.

Input

For each test case, first line contains an integer n ($2 \leq n \leq 1005$), total number of fruits. The second line contains two positive integers a, b ($1 \leq a, b \leq n - 1, a + b = n$). The third line contains a sequence of integers w_1, w_2, \dots, w_n ($1 \leq w_i \leq 10^6$), weight of each fruit. Input ends with 0 for parameter n .

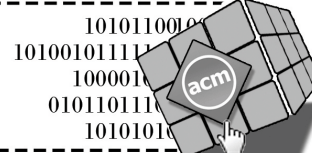
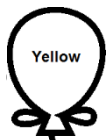
Output

For each test case, print the sequence of integers f_1, f_2, \dots, f_n , where f_i ($1 \leq f_i \leq 2$) is the number of the bag in which the i th fruit should be loaded in a line. If there are several possible solutions, then print such that the sequence f_1, f_2, \dots, f_n is the smallest lexicographically.

The sequence p_1, p_2, \dots, p_n is lexicographically less than q_1, q_2, \dots, q_n if there exists such j ($1 \leq j \leq n$) that $p_i = q_i$ for all $1 \leq i < j$, and $p_j < q_j$.

Sample Test

Input	Output
5	1 1 1 2 2
3 2	1 1 2 2
4 4 5 4 4	2 1 1 2 2 2
4	
2 2	
3 5 4 5	
6	
2 4	
5 4 4 5 4 4	
0	

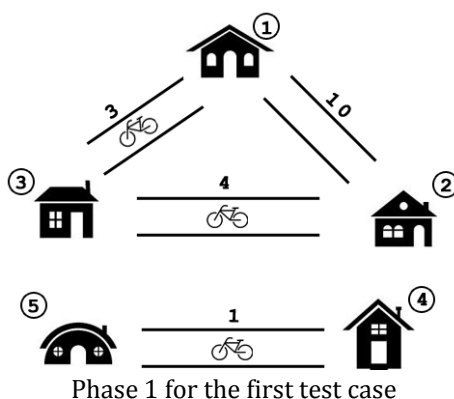


Problem D MiliTown

Construction of MiliTown (a new town in MiliCity) is just finished and it's time to provide some services for ease of living in town. There are lots of MiliBuildings in MiliTown and according to the plan of the project, each MiliBuilding must be accessible from any other MiliBuildings. This access may be direct or indirect (an indirect access between two MiliBuildings is a kind of access that there are some MiliBuildings between them). Government concerns a lot about air pollution and has chosen bicycle-exclusive line project as the first choice, but it seems there is a big problem with this town – a construction engineering mistake – there are some MiliBuildings that are not accessible from some other MiliBuildings!

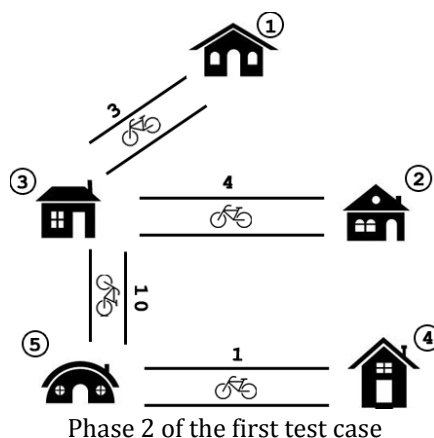
Milad is invited to solve the problem in two phases. In the first phase, he must make a plan for a bicycle-exclusive line between all connected MiliBuildings with constraints below at the lowest possible cost:

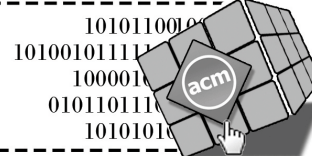
- Bicycles can only navigate on roads.
- Each road has a specific cost to be facilitated with bicycle-exclusive line.



Phase two will begin one year after accomplishment of phase one. In this phase, Milad must make a plan to bring the access between all MiliBuildings in MiliTown using bicycles. Again it should cost as less as possible and obey the following rule:

- If there is no road-access between some MiliBuildings, it is allowed to destroy a road – which is not facilitated with bicycle-exclusive line in phase one – and reconstruct it between any other two MiliBuildings. Facilitation cost remains the same as it was before the destruction.





Input

There is a single integer t on the first line of the input denoting number of test cases.

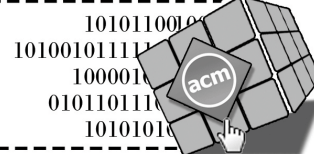
The first line of each test case, contains two integers n and m , with $1 \leq n \leq 200000$ and $1 \leq m \leq 200000$, which, n indicates the total MiliBuildings and m is the number roads. This is followed by m lines. In the i th line, there are three integers a_i , b_i and c_i . a_i and b_i are numbers of two different MiliBuildings and c_i with $1 \leq c_i \leq 1000$ is the cost of bicycle-exclusive line facilities in the road between a_i and b_i .

Output

For each test case, print a single line contains total bicycle facility cost (both phase one and phase two) and total number of road destructions separated by single space and print "Who were the engineers?!" (Without quotes) if there is no possible solution for the town.

Sample Test

Input	Output
2 5 4 1 2 10 1 3 3 2 3 4 4 5 1 6 4 1 2 10 1 3 3 2 3 4 4 5 1	18 1 Who were the engineers?!



Problem E Delicious Food

Your university has changed its food providing system and ran a one-on-one elimination online voting system to find out the most preferred food for each day. In this system, each time, two foods are chosen from a set of different kind of foods and students vote on which they prefer. The loser (less preferred) food is eliminated and the winner is returned to the food set. This process continues until only one food remains (It must be so delicious!).

One day, when you are going to vote for your food, the security module crashes and now you have access to all the information kept in the voting system such as students' food preferences and the set of foods that is going to be voted. Surprisingly, you have access to control which two foods to show each time for voting. How such an access! It's time to eat your most preferred food for lunch in the next day. Is there any way to control food selection so your beloved food be the winner?

Input

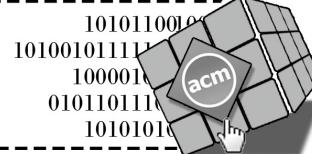
The first line of each test case, contains three integers n , m , and c with $1 \leq n \leq 100$, $1 \leq m \leq 100$ and $1 \leq c \leq n$, which, n indicates the total number of foods in the food set, m is the number of students and c is the number of the food you love to eat (foods are numbered from 1 to n). This is followed by m lines, each containing a permutation of the numbers 1 through n . The i 'th line should be interpreted as a ranking of the foods in the set by the student i . If two foods are pitted against each other, then student i will vote for the one appears before the other one in their rank (preference) list. You may also assume m is always odd. The last line of input contains three zeros and should not be processed.

Output

There is a single line of output for each test case with either the message yes or no indicating if it is possible for you to eat your preferred food next day or not.

Sample Test

Input	Output
4 3 1 1 2 3 4 2 3 1 4 3 4 1 2 4 3 4 1 2 3 4 2 3 1 4 3 4 1 2 0 0 0	yes no



Problem F

Jumpers and Commanders

This year, Milad won the cup of “Jumpers and Commanders” AI Contest. His smart jumper agent (the agent that he has programmed) did all the jumping tasks correctly.

A jumper agent is a robot that automatically and continuously jumps 1 unit (column) forward in constant time intervals from the time it is turned on. We call this kind of jumps “**Forward Jump**”. These jumpers can be programmed to jump upward or downward by special commands “up” and “down” respectively (1 unit per command). Special commands can only be executed between two forward jumps, i.e. after each forward jump, the agent checks its command list and executes (if any) “up” and “down” commands, and then performs the next forward jump.

A jumping task is defined as follow:

- There is a $n \times m$ (n rows and m columns) grid map with $2 \leq n \leq 50, 2 \leq m \leq 10000$.
- Some grid cells are blocked and no jumper can jump there. Also jumpers cannot jump over a blocked cell.
- A jumper should be placed on any of the cells in the first column of the map (contestant can choose any of the cells in the first column) and begin jumping to the last column without any jumps on blocked cells (It is clear that the number of forward jump intervals is equal to the number of map columns).
- Total number of commands in all command lists must be minimum.

Each jumper has a commander IC in its circuit. This IC is responsible for generating special commands (up and down) list. It is a programmable IC and must be programed by contestants in a way that it produces the correct special command list in each forward jump interval according to the map (it is also possible to generate an empty command list for a specific interval).

Intervals:	1	2	3	4	5	6	7
			x	x	x		
			x		x	x	x
Initial Position	x	x	x				

Commander						
Command list in each interval						
1	2	3	4	5	6	7
		up, up		down		

An Example of a Jumper and Its Commander (right table) for a Jumping Task (left table)

Input

The first line of the input contains an integer t , number of test cases.

In the first line of each test case, there are two integers n and m , denoting number of rows and columns respectively, followed by n strings. The i th string specifies the state of all cells of the i th row. Strings are consist of two kind of characters; ‘.’ for empty cell and ‘#’ for blocked cells. It is guaranteed that there is at least one way to start from first column and end with the last column.

Output

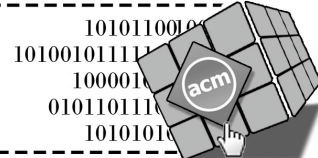
For each test, print a line containing the minimum total number of up and down commands, the commander generates, if Milad places his agent on the map.

Sample Test

Look at the next page.

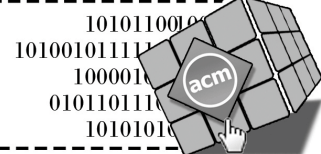
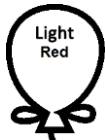


Shahed University
The 4th acm Programming Contest
چهارمین دوره مسابقات acm دانشگاه شاهد



Sample Test

Input	Output
3	1
3 5	6
.#...	0
...#.	
.###.	
5 10	
#...#...	
..##.#.#..	
#...#...#	
..##.#.#.#	
#..#...#..	
3 5	
####	
.....	
####	



Problem G Prime Calculation

In Shahed University, Computer Science students had wanted to create a very complex device. For creating this device they had to use several small chip-sets that just can do simple calculations. They had started to build the device part by part and used all the chip-sets they had, but one. In the last part they had to calculate all the prime numbers less than number n . However, it was impossible because they had only one small chip-set to implement that part and all that chip-set could do, was to calculate sum of three integer numbers. They tried several different methods to calculate at least k prime number of all of them. One of the students suggested that they use sum of two neighboring prime numbers and 1 to calculate a prime number (e.g. $7 + 11 + 1 = 19$), and claimed that this method works for at least k prime numbers which is exactly what they wanted, so now they want to know if this claim is right or not.

Write a program which given numbers n and k , determine if there is at least k prime numbers less than n that can be calculated from sum of two neighboring prime numbers and 1.

Note that two prime numbers are called neighboring if there are no other prime numbers between them.

Input

First line of input contains an integer t , number of test cases.

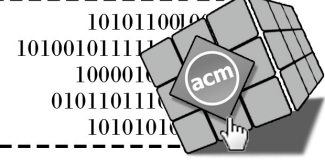
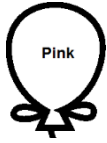
For each test case there is a single line containing two integers n ($2 \leq n \leq 1000$) and k ($0 \leq k \leq 1000$), as described in the problem statement.

Output

For each test, print YES if at least k prime numbers from 2 to n inclusively can be expressed as it was described above. Otherwise print NO. Print a blank line between the answers of two consecutive test cases.

Sample Test

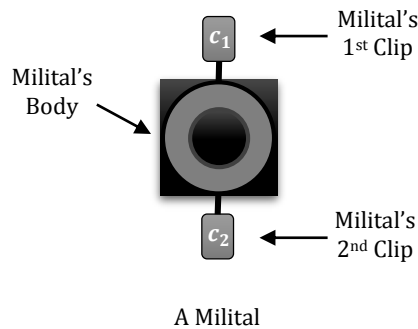
Input	Output
3	YES
61 5	
4 1	NO
4 0	YES



Problem H

Longest Possible Milital Chain

You are invited to the Milital Contest to create the longest Milital chain. Milital is a metal shape that has two clips on two opposite sides (c_1 and c_2 in the picture below). They are in different weights and Militals with different weights have different shapes. There are n different shapes and 10^5 different clips in this contest. Clips in one Milital may not be similar. Two clips can be attached to each other, if they are of the same kind.



You are given n boxes of Militals in each there are **exactly 3 Militals** of the same weight but may be **different in their clips** and It is guaranteed that **there is no pair of boxes with same weights**. You must create the longest possible chain of given Militals and hang it to the counter machine. This machine lifts up your chain and counts number of its Militals. Here are contest rules:

- You cannot use more than one Milital of the same weight (in other words you can use at most one Milital from each box).
- You are not allowed to use lighter Militals under heavier ones.

Write a program to compute number of Militals in the maximum possible chain.

Input

The first line of each test case, contains integer n , with $1 \leq n \leq 10^5$, denoting number of boxes. Boxes are numbered from 1 to n in order of increasing weight. Militals in boxes with lower number are lighter than those in boxes with higher number.

Following n lines, each contains 3 pairs of integers denoting type of clips for Militals in each box. Types are numbered from 1 to 10^5 . All integers are separated by a single space.

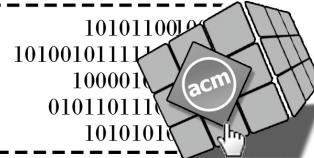
The last line of input is single zero and should not be processed.

Output

There is a single line of output for each test case with maximum possible number of Militals in a chain.

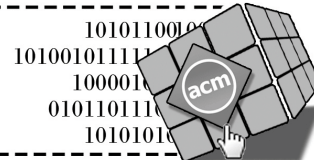
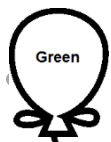
Sample Test

Look at the next page.



Sample Test

Input	Output
3	2
1 2 2 2 1 2	8
3 3 3 3 3 3	
3 2 1 1 1 1	
10	
1 5 10 3 6 5	
2 6 7 3 6 9	
5 7 3 2 1 9	
1 3 3 5 8 10	
6 6 2 2 4 4	
1 2 3 4 5 6	
10 9 8 7 6 5	
6 1 2 3 4 7	
1 2 3 3 2 1	
3 2 1 1 2 3	
0	



Problem I Portal in Desert

Lately some strange observations are reported from deserts around Shahed University. The report is mainly about teleportation (the ability to move somewhere else instantly). When astronomy students found the report, they quickly went there to analyze what is happening. They found that, there are some teleportation portals appearing in different moments of a day. Students said that portals have two features x and y denoting some kind of distance that objects can teleport through. An object can travel from portal r to portal q if and only if $x_q < x_r < y_q$ OR $x_q < y_r < y_q$.

After analysis students made, they could prepare a pattern for births of portals. A new astronomy student has found these pattern interesting and wants to do something more interesting. He wants to know if there is any way to start traveling from a portal and end with another portal **in a specific moment of a day**, but he is so young to find his answer and asks you to write a program that can provide him the answer. He has prepared a list that specifies births of portals and queries for traveling possibility between portals. Each item in the list consists of 3 space separated sub items; a character and two integers. 2 different kind of items which may appear in his list are as follows:

- $b \ x \ y$: A new portal with distance x to y is created (portal birth). $x < y$
- $q \ a \ b$: A query to answer – “Is there any way to start traveling from a th portal and end with b th portal according to current structure of portals”. Portals are indexed starting from 1. At birth of each portal, the minimum positive integer which is not used yet, is assigned to it as its index. $a \neq b$

Input

First line of input consists of a single integer t , number of test cases.

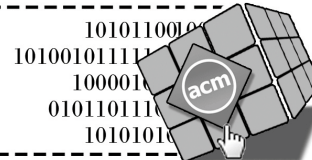
In the second line of each test, there is single integer n , ($1 \leq n \leq 100$) number of items in the list, and the following n lines, denote each list's item in the format mentioned above.

Output

For each test, give the answer of queries in the list, each in a line by printing “YES” if traveling is possible and “NO” otherwise. Print a blank line after answering all the queries of a test.

Sample Test

Input	Output
2	NO
5	YES
b 1 10	
b 11 15	YES
q 1 2	YES
b 5 14	YES
q 1 2	
5	
b 3 5	
b 4 6	
b 5 7	
q 1 3	
q 3 1	



Problem J

Drown in Programming

Programming is one of the most interesting activities for Milad. He has done lots of programming projects till now (Specially, .Net Projects). Each project p_i takes him d_i days to be finished using a specific algorithm. He may be asked to write it in a_i different algorithms (Each different algorithm takes the same amount of days d_i). Milad never works on two projects in a day.

Now he wants to know what project he had been working on, in a certain day.

Input

In the first line of each test case, there are two numbers n and m which are number of projects and number of questions Milad wants to know its answer, respectively ($1 \leq n, m \leq 10^5$).

The i th line of the following n lines specifies d_i and a_i for project p_i ($1 \leq d_i, a_i \leq 10^9$). It is guaranteed that total number of days for all projects does not exceeds 10^9 .

In the last line of each test case, there are m space separated numbers each denotes the day's number in which Milad wants to know which project he had been working on.

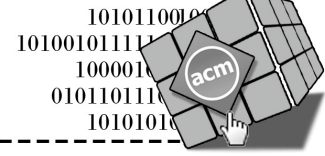
The line "0 0" (Without quotes) indicates end of test cases.

Output

In each test case, print the project's number he had been working on for day(s) he asks, each in a separate line.

Sample Test

Input	Output
3 3	1
1 1	2
1 1	3
1 1	1
1 2 3	2
3 5	2
1 1	2
2 2	3
3 3	
1 3 2 4 9	
0 0	



Problem K Letters

Averell Dalton decided to write an anonymous letter cutting the letters out of a newspaper heading. He knows heading s_1 and text s_2 that he wants to send. Averell can use every single heading letter no more than once. Averell doesn't have to cut the spaces out of the heading — he just leaves some blank space to mark them. Help him; find out if he will manage to compose the needed text.

Input

First line of the input, contains a single integer t , denoting number of test cases.

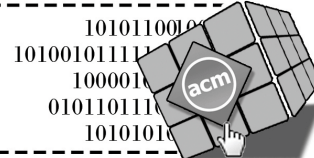
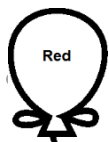
For each test case, the first line contains a newspaper heading s_1 . The second line contains the letter text s_2 . s_1 and s_2 are non-empty lines consisting of **spaces, uppercase and lowercase Latin letters**, whose lengths do not exceed 200 symbols. The uppercase and lowercase letters should be differentiated. Averell does not cut spaces out of the heading.

Output

For each test case, If Averell can write the given anonymous letter, print YES, otherwise print NO, each in a separate line.

Sample Test

Input	Output
4	YES
Welcome to the fourth acm contest of Shahed University	NO
We are here	NO
Welcome to the fourth acm contest of Shahed University	YES
We have stress	
Welcome to the fourth acm contest of Shahed University	
Can we leave the contest	
abcdefghijkl	
k j i h g f e d c b a	



Problem L Dam and Pumps

There is a dam with n pumps aligned in a row to disgorge the water behind the dam. These pumps are numbered 1 to n from left to right. Initially some of the pumps are switched on. A pump can be switched on (the pump should be switched off at that moment) if there is at least one adjacent pump which is already switched on. A worker wants to switch all the pumps on. He knows the initial state of pumps and he's wondering how many different ways there exist to switch all the pumps on. Please find the required number of ways modulo 1000000007 ($10^9 + 7$).

Input

The first line of the input, contains a single integer t , the number of test cases, followed by the input data for each test case. In the second line there are two integers n and m , where n is the number of pumps in the sequence and m is the number of pumps which are initially switched on, ($1 \leq n \leq 1000$, $1 \leq m \leq n$). The third line contains m distinct integers, each between 1 to n inclusively, denoting the indices of pumps which are initially switched on.

Output

For each test case, print a single line containing the number of different possible ways to switch on all the pumps modulo 1000000007 ($10^9 + 7$).

Sample Test

Input	Output
3	1
3 2	6
1 3	2520
5 1	
3	
10 3	
8 2 5	