



The 2024 ICPC European Championship Official Problem Set



icpc global sponsor
programming tools



icpc diamond
multi-regional sponsor



icpc europe
regional sponsor

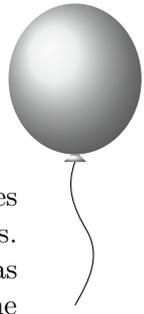


icpc
sponsor



CITADEL | CITADEL Securities

icpc
sponsor



A Grove

You want to plant trees in a square lawn of size $n \times n$ whose corners have Cartesian coordinates $(0, 0)$, $(n, 0)$, $(0, n)$, and (n, n) . Trees can only be planted at locations with integer coordinates. Every tree will grow roots within a disk of radius r centered at the location where the tree was planted; such disks must be fully contained in the lawn (possibly touching the boundary of the lawn) and can only intersect each other on their boundaries.

Find a configuration that maximizes the number of trees.

INPUT

The first and only line contains an integer n ($1 \leq n \leq 20$) and a real number r ($0 < r \leq n/2$) — the length of the sides of the lawn, and the radius of the disks where each tree will grow roots. The real number r is given in decimal notation with at least 1 and at most 3 digits after the decimal point.

OUTPUT

In the first line, print the maximum number m of trees that can be planted.

In the next m lines, print a configuration that maximizes the number of trees. Specifically, in the $(i + 1)$ -th line, print two integers x and y — the coordinates of the location where the i -th tree should be planted. You can print the trees in any order.

If there are multiple solutions, print any of them.

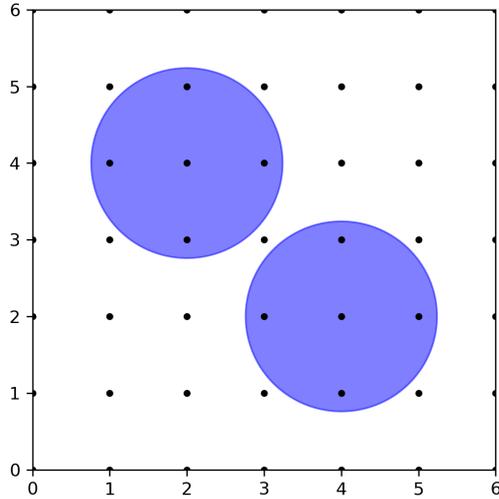
SAMPLES

Sample input 1	Sample output 1
6 1.241	2 4 2 2 4

Explanation of sample 1.

The sample output is shown in the following figure. Note that this is not the only configuration that maximizes the number of trees.

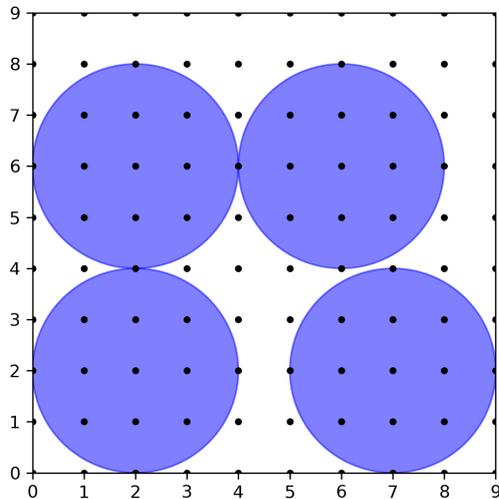
The 2024 ICPC European Championship



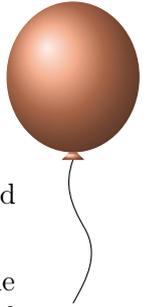
Sample input 2	Sample output 2
9 2.0	4 2 2 7 2 2 6 6 6

Explanation of sample 2.

The sample output is shown in the following figure. Note that this is not the only configuration that maximizes the number of trees.



B Charming Meals



The Czech cuisine features n appetizers and n main dishes. The i -th appetizer has spiciness a_i , and the i -th main dish has spiciness b_i .

A typical Czech meal consists of exactly one appetizer and one main dish. You want to pair up the n appetizers and n main dishes into n meals with each appetizer and each main dish being included in exactly one meal.

Your meals shall surprise the diners, so you want the spiciness levels of the two parts of the same meal to be as different as possible. The charm of a meal is the difference (in absolute value) between the spiciness of the appetizer and the spiciness of the main dish. So, a meal consisting of an appetizer with spiciness x and a main dish with spiciness y has charm equal to $|x - y|$.

You want to maximize the minimum charm of the resulting n meals. What is the largest possible value of the minimum charm that you can achieve?

INPUT

Each test contains multiple test cases. The first line contains an integer t ($1 \leq t \leq 1000$) — the number of test cases. The descriptions of the t test cases follow.

The first line of each test case contains a single integer n ($1 \leq n \leq 5000$) — the number of appetizers and main dishes.

The second line of each test case contains n integers a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$) — the spicinesses of the n appetizers.

The third line of each test case contains n integers b_1, b_2, \dots, b_n ($0 \leq b_i \leq 10^9$) — the spicinesses of the n main dishes.

It is guaranteed that the sum of n^2 over all test cases does not exceed $25 \cdot 10^6$.

OUTPUT

For each test case, print the largest possible value of the minimum charm you can achieve.

The 2024 ICPC European Championship

SAMPLES

Sample input 1	Sample output 1
<pre> 4 3 0 0 0 1000000000 1000000000 1000000000 5 1 2 3 4 5 1 2 3 4 5 6 0 0 0 100 100 100 100 100 100 0 0 0 7 14 25 62 74 86 95 12 51 62 71 72 92 20 84 </pre>	<pre> 1000000000 2 100 30 </pre>

Explanation of sample 1.

In the **first test case**, no matter how you pair up the appetizers with the main dishes, each meal will have an appetizer with spiciness 0 and a main dish with spiciness 1000000000, so the charm of each meal will be 1000000000.

In the **second test case**, one optimal way to pair up appetizers and main dishes is: (1, 5), (2, 4), (3, 1), (4, 2), (5, 3). The corresponding meals have charms: 4, 2, 2, 2, 2. The resulting minimum charm is 2.

In the **third test case**, one way to maximize the minimum charm is to pair up the three appetizers with spiciness 0 with the three main dishes with spiciness 100, and the three appetizers with spiciness 100 with the three main dishes with spiciness 0. Doing so, the charm of each meal will be exactly 100.

C Annual Ants' Gathering

Deep within a forest lies an ancient tree, home to n ants living in n tiny houses, indexed from 1 to n , connected by the branches of the tree.



Once a year, all the ants need to gather to watch the EUC. For this, all ants move along the $n - 1$ branches of the tree they live on to meet at the home of one ant.

However, this year the ants could not agree on where to meet and need your help to gather up. You can tell all the ants currently at house u to move to house v if there is a branch directly connecting those two houses. However, the ants ignore your command if there are fewer ants gathered in house v than in house u , i.e., if it would be easier for the ants from house v to move. This even holds true if no ant at all is currently in house v . You can give this kind of commands as many times as you want.

Is it possible for you to gather all the ants in a single house?

INPUT

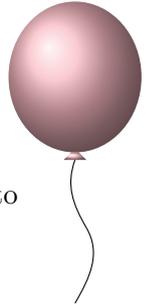
The first line contains one integer n ($1 \leq n \leq 200\,000$) — the number of ant homes.

Each of the following $n - 1$ lines contains two integers u and v ($1 \leq u, v \leq n$) — there is a branch directly connecting the house u and house v .

It is guaranteed that every ant can reach the house of any other ant just by following the branches of the tree.

OUTPUT

Print YES if it is possible to gather all the ants in a single house. Otherwise, print NO.



The 2024 ICPC European Championship

SAMPLES

Sample input 1	Sample output 1
7 5 1 3 2 4 6 3 6 7 1 1 3	YES

Explanation of sample 1.

You can gather all the ants at house 3 as follows:

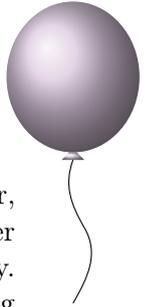
- You tell to the ant at house 4 to move to house 6.
- You tell to the ant at house 2 to move to house 3.
- You tell to the two ants at house 6 to move to house 3 (which already contains two ants).
- You tell to the ant at house 5 to move to house 1.
- You tell to the ant at house 7 to move to house 1 (which already contains two ants).
- You tell to the three ants at house 1 to move to house 3 (which already contains four ants).

Sample input 2	Sample output 2
5 1 4 4 2 3 2 5 3	NO

Explanation of sample 2.

It is impossible to gather all the ants in a single house.

Sample input 3	Sample output 3
6 4 5 5 6 6 1 2 6 3 2	YES



D Funny or Scary?

You are designing a new video game. It has n scenarios, which the player may play in any order, but each scenario must be played exactly once. When a player switches from a scenario to another scenario, the game shows a specially crafted transition video to make it all feel part of one big story. This video is specific to a pair of scenarios, but not to their order, in other words, the video playing when switching from scenario a to scenario b is the same as the video playing when switching from scenario b to scenario a . Therefore, you need to create $\frac{n(n-1)}{2}$ different transition videos, one for each possible pair of different scenarios.

Each transition video can be either *funny* or *scary*. It is boring to see too many funny videos or too many scary videos in a row. Therefore, your goal is to create the videos in such a way that no matter in which order does the player approach the scenarios, they will never see more than $\lceil \frac{3n}{4} \rceil$ transition videos of the same type in a row.

You have already come up with ideas for at most $\lfloor \frac{n}{2} \rfloor$ of the transition videos, and therefore already know if those will be funny or scary. Now you need to choose funny or scary for all other transition videos in such a way that the above requirement is satisfied.

INPUT

The first line contains a single integer n ($2 \leq n \leq 24$) — the number of scenarios in the game.

The next n lines describe the partial transition video plan. Each of those lines contains n characters. The j -th character of the i -th line corresponds to the transition video between the i -th and the j -th scenarios. It will be **F** if the corresponding transition video will be funny, **S** if the corresponding transition video will be scary, **?** if the corresponding transition video is still undecided, or **.** if $i = j$.

It is guaranteed that the i -th character of the j -th line and the j -th character of the i -th line will be the same for all i and j . It is guaranteed that at most $\lfloor \frac{n}{2} \rfloor$ (n divided by 2, rounded down) transition videos will already be decided, in other words, that at most $2\lfloor \frac{n}{2} \rfloor$ characters in the input will be **F** or **S**.

OUTPUT

Print n lines describing the full transition video plan in the same format as the input. Each of those lines must contain n characters. The j -th character of the i -th line must be **F** if the corresponding transition video is funny, **S** if the corresponding transition video is scary, or **.** if $i = j$.

Each **?** character from the input must be replaced with either **F** or **S**, and all other characters from the input must remain unchanged. It must still hold that the i -th character of the j -th line and the j -th character of the i -th line are the same for all i and j .

For each permutation of the n scenarios, it must hold that the transition videos corresponding to playing the scenarios in this order do not have more than $\lceil \frac{3n}{4} \rceil$ ($3n$ divided by 4, rounded up) videos of the same type consecutively.

If there are multiple solutions, print any of them. It can be proven that for all inputs satisfying the

The 2024 ICPC European Championship

constraints of this problem a solution always exists.

SAMPLES

Sample input 1	Sample output 1
<pre>5 .FF?? ?.??? F?.S? ??S.? ????.</pre>	<pre>.FFFF F.FFF FF.SF FFS.F FFFF.</pre>

Explanation of sample 1.

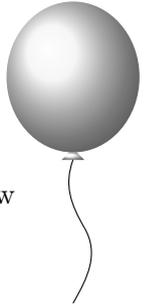
We are allowed $\lceil \frac{3 \cdot 5}{4} \rceil = 4$ transition videos of the same type in a row, but for any permutation of the 5 scenarios the player will see only 4 transition videos in total, therefore we can choose funny or scary freely. We must still respect the already chosen types.

Sample input 2	Sample output 2
<pre>12 .???????????? ?.???????????? ?? .?????????? ??? .????????? ???? .???????? ????? .??????? ?????? .?????? ??????? .????? ???????? .??? ????????? .?? ?????????? .? ????????????.</pre>	<pre>.SSSFFSSSSFS S.SFFSFSFFFS SS.SFFFSSSFS SFS.FFSSSSSFS FFFF.FFFFFSF FSFFF.SFFSFF SFFSFS.SSSFS SSSSFFS.SSFS SFSSFFSS.SFS SFSSFSSSS.FS FFFFSFFFFF.F SSSSFFSSSSF.</pre>

Explanation of sample 2.

One of the 479001600 possible permutations of scenarios is 1, 7, 4, 12, 9, 8, 2, 6, 10, 3, 11, 5. The player will get the following sequence of transition videos for this permutation: SSSSSSSSFS. Even though this sequence has 10 scary transition videos in total, it has only 9 scary transition videos in a row, which is the maximum allowed amount ($\lceil \frac{3 \cdot 12}{4} \rceil = 9$).

E Damage per Second



You just created a new character in your favourite role-playing game and now have to decide how to skill him.



The two skill attributes to be chosen are: *damage per hit* and *hits per second*. Damage per hit is the amount of damage you deal with a single hit, while hits per second is the number of hits you can make in one second. Initially, both skill attributes are set at 0. You have k skill points to distribute as you want; in other words, you can choose the values of the two skills so that they are positive integers with sum at most k .

The tutorial of the game (the boring part you want to finish as soon as possible) consists of n monsters to be killed one after the other. The i -th monster has h_i health points, i.e., it dies after you have inflicted at least h_i damage.

How can you assign the two skill attributes to minimize the time necessary to kill all the n monsters?

INPUT

The first line contains two integers n and k ($1 \leq n \leq 200\,000$, $2 \leq k \leq 200\,000$) — the number of enemies and the number of skill points.

The second line contains n integers h_i ($1 \leq h_i \leq 10^{13}$) — the health of the i th enemy.

OUTPUT

Print two positive integers x and y ($1 \leq x, y$ and $x + y \leq k$) — the number of skill points you want to invest in damage per hit and hits per second. If there are multiple optimal solutions, print any of them.

The 2024 ICPC European Championship

SAMPLES

Sample input 1	Sample output 1
1 7 14	3 4

Explanation of sample 1.

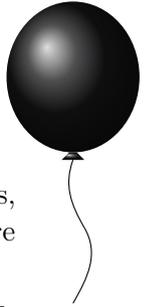
There is only one monster and you have 7 skill points to distribute. If you make 3 damage per hit, you will need 5 hits to kill it. If you do 4 hits per second, you will need 1.25 seconds to beat the monster. There is no way to beat the monster faster than this.

Sample input 2	Sample output 2
4 9 1 2 3 4	4 5

Explanation of sample 2.

You will need one hit for each monster and a total time of 0.8 seconds if you distribute 4 skill points on damage per hit and the remaining 5 points on hits per second.

Sample input 3	Sample output 3
5 13 3 4 5 6 7	7 6



F Dating

You are the developer of a dating app which ignores gender completely. The app has n users, indexed from 1 to n . Each user's profile features a list of the activities they enjoy doing. There are m possible activities, indexed from 1 to m .

A match between two users is good if they share at least one activity and, at the same time, both of them like at least one activity that the other user does not like.

Find a good match if it exists.

INPUT

The first line contains two integers n and m ($2 \leq n \leq 200\,000$, $1 \leq m \leq 10^6$) — the number of users and the number of activities.

Each of the following n lines contains a number k_i ($0 \leq k_i \leq m$) — the number of activities that user i likes — followed by k_i distinct integers from 1 to m — the activities user i likes.

It is guaranteed that $k_1 + k_2 + \dots + k_n$ does not exceed 10^6 .

OUTPUT

Print YES if a good match exists. Otherwise, print NO.

If a good match exists, on the next line print two integers — the indexes of two users that make a match.

SAMPLES

Sample input 1	Sample output 1
3 5 3 1 2 4 5 1 2 3 4 5 2 1 5	YES 3 1

Explanation of sample 1.

Users 1 and 3 form a match, because they share activity 1, and, furthermore, user 3 likes activity 5 (which user 1 does not like) and user 1 likes activity 4 (which user 3 does not like). Note that users 1 and 2, as well as users 2 and 3, do not form a match, as there is no activity that users 1 or 3 like, and user 2 doesn't like.



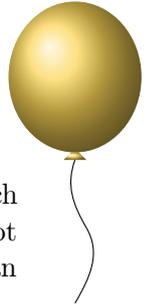
icpc | europe contests



The 2024 ICPC European Championship

Sample input 2	Sample output 2
3 3 1 1 1 2 3 2 3 1	NO

G Scooter



The Czech Technical University campus consists of n buildings, indexed from 1 to n . In each building, there can be a math class scheduled, or a computer science class, or neither (but not both). Additionally, in each building, there is at most one professor, and each professor is either an expert in mathematics or in computer science.

As an intern at University Express Inc., your job is to quickly transport the professors to their classes. For this, you have been granted a brand new two-person scooter, able to accommodate yourself, plus at most one passenger.

Initially, you are the only person on the scooter. When you arrive at a building, you may drop off or pick up professors to/from that building. However, in order to improve the efficiency of your task, you are allowed to drive to each of the n buildings **at most once**, in the order of your choice (you can also decide where to start the itinerary).

After the end of your itinerary, in each building where a math class is scheduled, there must be a professor expert in math, and in each building where a computer science class is scheduled, there must be a professor expert in computer science.

Devise an itinerary that makes it possible to teach all classes.

INPUT

The first line contains an integer n ($1 \leq n \leq 2000$) — the number of buildings in the campus.

The second line contains a string c of length n consisting of the characters $-$, C , M — the i -th character denotes the subject of the class scheduled in the i -th building. C stands for computer science, M stands for mathematics, while $-$ means that there is no class scheduled in the i -th building.

The third line contains a string p of length n consisting of the characters $-$, C , M — the i -th character denotes the expertise of the professor in the i -th building (if there is a professor). C stands for computer science, M stands for mathematics, while $-$ means that there is no professor in the i -th building.

It is guaranteed that, for all tests given to your program, there exists a valid itinerary.

OUTPUT

In the first line print an integer l — the number of operations in your chosen itinerary.

The i -th ($1 \leq i \leq l$) of the next l lines must contain one of three commands:

1. **DRIVE** x — go to the building with the number x ($1 \leq x \leq n$);
2. **PICKUP** — pick up the professor which was *initially* at the current building;
3. **DROPOFF** — drop off the passenger professor at the current building.

In order for the itinerary to be valid, the following conditions must hold:

The 2024 ICPC European Championship

1. No two DRIVE instructions should go to the same building;
2. At most one DROPOFF and one PICKUP instruction **in this order** should be issued at each specific building;
3. For all PICKUP instructions, there must be a professor *initially* at the building, as well as no one already riding along on the scooter;
4. For all DROPOFF instructions, there must be a professor riding along at the time of the command;
5. After the itinerary, in each building, if there is a class in that building, there must be a professor expert in the class' subject (either initially, or because they were dropped off there).

Note that, in particular, you cannot pick up a professor that you just dropped off for an itinerary to be valid.

SAMPLES

Sample input 1	Sample output 1
<pre>3 CM- -CM</pre>	<pre>7 DRIVE 3 PICKUP DRIVE 2 DROPOFF PICKUP DRIVE 1 DROPOFF</pre>

Explanation of sample 1.

You start by driving to building number 3. You then pick up the mathematics professor. After dropping him off at building number 2, where a mathematics class is being held, you pick up the computer science professor from there, and drop her off at building number 1, finishing your itinerary.

Sample input 2	Sample output 2
<pre>1 C C</pre>	<pre>0</pre>



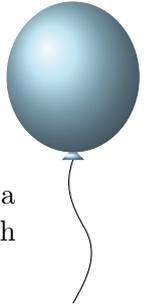
icpc | europe contests



The 2024 ICPC European Championship

Sample input 3	Sample output 3
2 -M MC	4 DRIVE 1 PICKUP DRIVE 2 DROPOFF

BLANK PAGE



H Division Avoidance

A newly discovered organism can be represented as a set of cells on an infinite grid. There is a coordinate system on the grid such that each cell has two integer coordinates x and y . A cell with coordinates $x = a$ and $y = b$ will be denoted as (a, b) .

Initially, the organism consists of a single cell $(0, 0)$. Then zero or more *divisions* can happen. In one division, a cell (a, b) is removed and replaced by two cells $(a + 1, b)$ and $(a, b + 1)$.

For example, after the first division, the organism always consists of two cells $(1, 0)$ and $(0, 1)$, and after the second division, it is either the three cells $(2, 0)$, $(1, 1)$ and $(0, 1)$, or the three cells $(1, 0)$, $(1, 1)$ and $(0, 2)$.

A division of a cell (a, b) can only happen if the cells $(a + 1, b)$ and $(a, b + 1)$ are not yet part of the organism. For example, the cell $(1, 0)$ cannot divide if the organism currently consists of the three cells $(1, 0)$, $(1, 1)$ and $(0, 2)$, since the cell $(1, 1)$ that would be one of the results of this division is already part of the organism.

You are given a set of forbidden cells (c_i, d_i) . Is it possible for the organism to contain none of those cells after zero or more divisions?

INPUT

Each test contains multiple test cases. The first line contains an integer t ($1 \leq t \leq 10\,000$) — the number of test cases. The descriptions of the t test cases follow.

The first line of each test case contains an integer n ($1 \leq n \leq 10^6$) — the number of forbidden cells.

The next n lines contain two integers each. The i -th of such lines contains c_i and d_i ($0 \leq c_i, d_i \leq 10^9$) — the coordinates of the i -th forbidden cell. It is guaranteed that all forbidden cells are distinct.

It is guaranteed that the sum of values of n over all test cases does not exceed 10^6 .

OUTPUT

For each test case, print **YES** if it is possible for the organism to contain no forbidden cells after zero or more divisions. Otherwise, print **NO**.

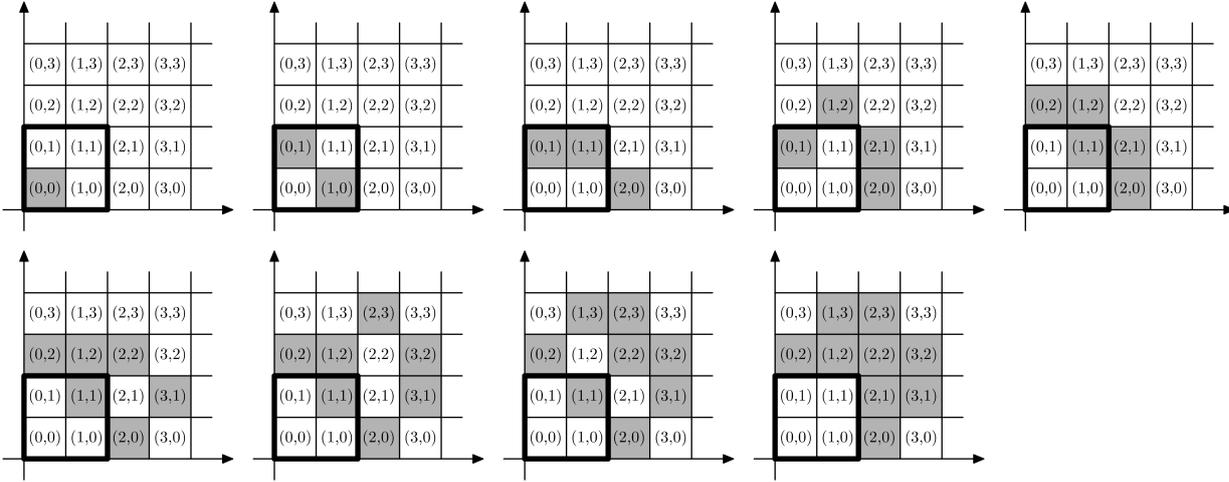
The 2024 ICPC European Championship

SAMPLES

Sample input 1	Sample output 1
2	YES
4	NO
0 0	
1 0	
0 1	
1 1	
16	
0 0	
0 1	
0 2	
0 3	
1 0	
1 1	
1 2	
1 3	
2 0	
2 1	
2 2	
2 3	
3 0	
3 1	
3 2	
3 3	

Explanation of sample 1.

In the **first test case**, dividing the following cells in the following order creates an organism without any forbidden cells: (0,0), (1,0), (1,1), (0,1), (2,1), (2,2), (1,2), (1,1). The following picture demonstrates how the organism changes during this process:





icpc global sponsor
programming tools



icpc diamond
multi-regional sponsor



icpc europe
regional sponsor



icpc
sponsor

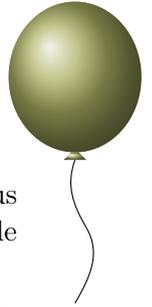


icpc
sponsor

The 2024 ICPC European Championship

In the **second test case**, you can see that, surprisingly, any organism always has at least one cell in the $0 \leq x, y \leq 3$ square, no matter how many divisions we do.

BLANK PAGE



I Disks

You are given n disks in the plane. The center of each disk has integer coordinates, and the radius of each disk is a positive integer. No two disks overlap in a region of positive area, but it is possible for disks to be tangent to each other.

Your task is to determine whether it is possible to change the radii of the disks in such a way that:

- Disks that were tangent to each other remain tangent to each other.
- No two disks overlap in a region of positive area.
- The sum of all radii strictly decreases.

The new radii are allowed to be arbitrary positive real numbers. The centers of the disks cannot be changed.

INPUT

The first line contains an integer n ($1 \leq n \leq 1000$) — the number of disks.

The next n lines contain three integers each. The i -th of such lines contains x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$), and r_i ($1 \leq r_i \leq 10^9$) — the coordinates of the center, and the radius, of the i -th disk.

OUTPUT

Print YES if it is possible to change the radii in the desired manner. Otherwise, print NO.

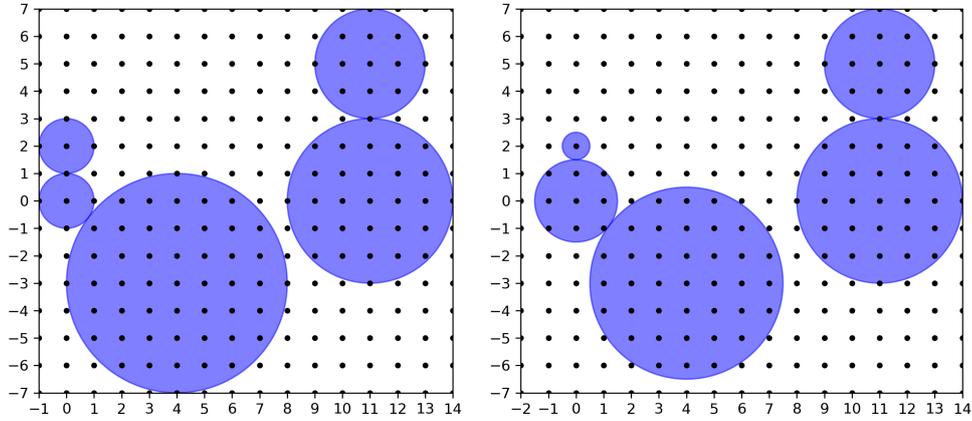
SAMPLES

Sample input 1	Sample output 1
5 0 2 1 0 0 1 4 -3 4 11 0 3 11 5 2	YES

Explanation of sample 1.

One can decrease the radii of the first and third disk by 0.5, and increase the radius of the second disk by 0.5. This way, the sum of all radii decreases by 0.5. The situation before and after changing the radii is depicted below.

The 2024 ICPC European Championship

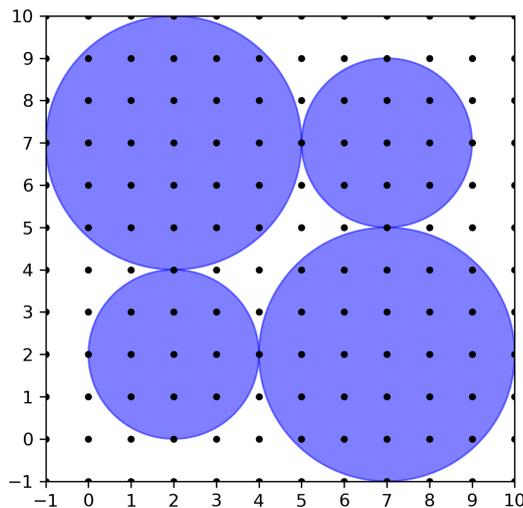


First sample (left) and a valid way to change the radii of the disks (right).

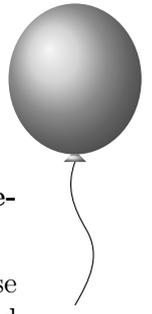
Sample input 2	Sample output 2
4 2 2 2 7 2 3 7 7 2 2 7 3	NO

Explanation of sample 2.

Depicted below, there is no way to change the radii of the disks in the desired manner.



Second sample.



J Amanda the Amoeba

This problem has an attachment. You can use it to simulate and visualize the movements of the amoeba.

Amoeba Amanda lives inside a rectangular grid of square pixels. Her body occupies some of these pixels. Other pixels may be either free or blocked. Amanda moves across the grid using the so-called amoeboid movement. In each step of such a movement, her body first shrinks by one pixel (one pixel of the body is removed and becomes free), and then grows at a different place (one previously-free pixel is added to the body).

To prevent structural damage, Amanda's body always occupies a connected region of pixels, which means that any pair of pixels forming the body can be connected by a sequence of adjacent pixels without ever leaving the body. Two pixels are considered adjacent if they share a common side (each pixel has at most 4 neighbours). The body remains connected even during the movement, including the moment after removing a pixel and before adding another one.

Your task is to help Amanda find her way around. Given her initial position and desired final position, suggest a sequence of valid moves leading from the former to the latter.

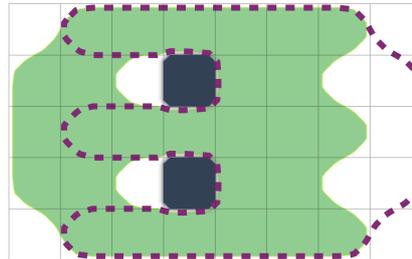


Illustration of sample 1: The filled shape is the initial position, the dotted region is the final position.

INPUT

The first line contains two integers r and c ($1 \leq r, c \leq 50$) — the size of the rectangular grid in pixels.

The next r lines contain c characters each, describing the initial position of Amanda. Each of those characters is either a dot `.` denoting a free pixel, an asterisk `*` denoting Amanda's body, or an `X` denoting a blocked pixel which may never be occupied.

The next line is empty.

The next r lines describe the desired final position in the same format as the initial position.

It is guaranteed that:

- The number of pixels forming Amanda's body is the same in both positions, and it is at least 2.
- The body of Amanda is connected in the initial position.
- The body of Amanda is connected in the final position.

The 2024 ICPC European Championship

- The blocked pixels do not change between the descriptions of the initial and final position, their placement is exactly the same in both positions.

OUTPUT

Print YES if it is possible for Amanda to go from the initial position to the final one. Otherwise, print NO.

If it is possible, on the next line print one integer m ($0 \leq m \leq 10\,000$) — the number of moves to execute.

The following m lines must contain four integer coordinates each: i_1, j_1, i_2, j_2 ($1 \leq i_1, i_2 \leq r$, $1 \leq j_1, j_2 \leq c$). These four coordinates specify one move, meaning that the pixel at i_1 -th row and j_1 -th column is first removed from the body. Then, (i_2, j_2) must designate a **different** location where one pixel is added.

The sequence should consist only of valid moves and after the last move, Amanda's body should occupy the desired final position.

If there are multiple solutions, print any of them.

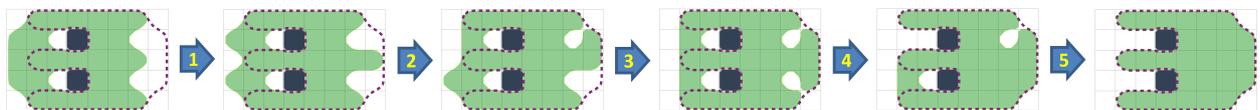
Under the assumptions of this problem, it can be proven that if it is possible for Amanda to go from the initial position to the desired final one, then it is possible to do it with at most 10 000 moves.

SAMPLES

Sample input 1	Sample output 1
<pre> 5 8 .*****. **.X**.. *****. **.X**.. .*****. .*****. ...X**** .***** ...X**** .*****. </pre>	<pre> YES 5 3 1 3 8 2 1 2 8 4 1 4 8 2 2 4 7 4 2 2 7 </pre>

Explanation of sample 1.

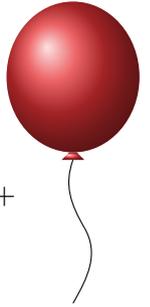
Amanda executes 5 moves to reach the final position, as shown in the figure below.



The 2024 ICPC European Championship

Sample input 2	Sample output 2
2 5 *.X.. **X.. ..X** ..X*.	NO

BLANK PAGE



K Make Triangle

You are given n positive integers x_1, x_2, \dots, x_n and three positive integers n_a, n_b, n_c satisfying $n_a + n_b + n_c = n$.

You want to split the n positive integers into three groups, so that:

- The first group contains n_a numbers, the second group contains n_b numbers, the third group contains n_c numbers.
- Let s_a be the sum of the numbers in the first group, s_b be the sum in the second group, and s_c be the sum in the third group. Then s_a, s_b, s_c are the sides of a triangle with positive area.

Determine if this is possible. If this is possible, find one way to do so.

INPUT

Each test contains multiple test cases. The first line contains an integer t ($1 \leq t \leq 100\,000$) — the number of test cases. The descriptions of the t test cases follow.

The first line of each test case contains the integers n, n_a, n_b, n_c ($3 \leq n \leq 200\,000, 1 \leq n_a, n_b, n_c \leq n - 2, n_a + n_b + n_c = n$) — the number of integers to split into three groups, and the desired sizes of the three groups.

The second line of each test case contains n integers x_1, x_2, \dots, x_n ($1 \leq x_i \leq 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed 200 000.

OUTPUT

For each test case, print **YES** if it is possible to split the numbers into three groups satisfying all the conditions. Otherwise, print **NO**.

If such a split exists, then describe the three groups as follows.

On the next line, print n_a integers a_1, a_2, \dots, a_{n_a} — the numbers in the first group.

On the next line, print n_b integers b_1, b_2, \dots, b_{n_b} — the numbers in the second group.

On the next line, print n_c integers c_1, c_2, \dots, c_{n_c} — the numbers in the third group.

These $n_a + n_b + n_c = n$ integers should be a permutation of x_1, x_2, \dots, x_n , and they should satisfy the conditions from the statement.

If there are multiple solutions, print any of them.

The 2024 ICPC European Championship

SAMPLES

Sample input 1	Sample output 1
4	YES
6 2 2 2	1 1
1 1 1 1 1 1	1 1
5 3 1 1	1 1
1 1 1 1 1	NO
6 2 2 2	NO
1 1 1 1 1 3	YES
8 1 2 5	16
16 1 1 1 1 1 1 12	12 1
	1 1 1 1 1

Explanation of sample 1.

In the **first test case**, we can put two 1s into each group: the sum in each group would be 2, and there exists a triangle with positive area and sides 2, 2, 2.

In the **second and third test cases**, it can be shown that there is no such way to split numbers into groups.

In the **fourth test case**, we can put number 16 into the first group, with sum 16, numbers 12 and 1 into the second group, with sum 13, and the remaining five 1s into the third group, with sum 5, as there exists a triangle with positive area and sides 16, 13, 5.