The 15th Shandong CCPC Provincial Collegiate Programming Contest

Contest Session

May 25, 2025

Problem List

A	Project Management
B	Pinball
C	Bracket Integer
D	Distributed System
E	Greatest Common Divisor
F	ACE String
G	Assembly Line
H	Minimum Spanning Tree
I	Square Puzzle
J	Useful Algorithm
K	Path Planning 2
L	Stella
M	Triangulation

This problem set should contain 13 (thirteen) problems on 19 (nineteen) numbered pages. Please inform a runner immediately if something is missing from your problem set.

Prepared by SUA Programming Contest Problem Setter Team.
https://sua.ac/

Hosted by

Problem Set Prepared by

It's against the rules to open non-contest websites during the contest.
If you're interested (which is our pleasure),
please scan the QR code only after the contest.

Problem A. Project Management

There are n employees in a company, where the i-th employee is of rank a_i. The company is launching a big project, which requires as many employees to join as possible. However, the employees are not fond of their colleagues with higher ranks. To make sure that all people in the project are willing to work together, for all $1 \le i \le n$, if the i-th employee is in the project, then there can only be at most b_i people in the project with a rank larger than a_i.

As the boss of the company, you need to choose as many people as possible to join the project.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 2 \times 10^5$), indicating the number of employees.

For the following n lines, the i-th line contains two integers a_i and b_i ($1 \le a_i \le n$, $0 \le b_i < n$), indicating the rank of the i-th employee, and at most how many people with a higher rank he/she would like to work with.

It's guaranteed that the sum of n of all test cases does not exceed 2×10^5.

Output

For each test case, first output one line containing one integer c, indicating the maximum number of people who can join the project. Then output one line containing c distinct integers $p_1, p_2, \cdots, p_c$ separated by a space, indicating the indices of the employees to join the project. If there are multiple optimal answers, you can output any of them.

Example

standard input	standard output
2 6 3 0 4 0 3 1 5 3 1 2 3 1 2 1 1 1 0	3 6 3 4 2 1 2

Problem B. Pinball

There are two horizontal straight lines $y = 0$ and $y = H$ on the 2-dimensional plane. Between the two lines there are initially n tiny wooden boards which can be regarded as single points. The i-th wooden board is located at (x_i, y_i).

Maintain q operations of the following three types.

- $+\ x\ y$: Add a wooden board located at (x, y) to the plane.
- $-\ x\ y$: Remove the wooden board located at (x, y) from the plane.
- $?\ x\ y\ v_y\ g$: A pinball is released from (x, y).

  Let $\overrightarrow{v} = (v_x, v_y)$ be the velocity of the ball (that is to say, if the ball is currently located at (x, y) it will move to $(x + v_x\epsilon, y + v_y\epsilon)$ after ϵ seconds). If $g \ge x$ then $v_x = 1$, otherwise $v_x = -1$, while v_y is given by the query. The value of v_y is either 1 or -1.

  When the ball hits a wooden board or one of the two horizontal straight lines, v_y will be reversed (that is, v_y becomes $-v_y$) while v_x remains unchanged.

  Calculate the y coordinate of the pinball when its x coordinate equals to g.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$) indicating the number of test cases. For each test case:

The first line contains three integers H, n and q ($2 \le H \le 10^9$, $1 \le n, q \le 10^5$) indicating the position of the upper horizontal straight line, the number of initial wooden boards and the number of operations.

For the following n lines, the i-th line contains two integers x_i and y_i ($1 \le x_i \le 10^9$, $1 \le y_i < H$) indicating the position of the i-th wooden board.

For the following q lines, the i-th line first contains a character op ($op \in \{+, -, ?\}$) indicating the type of the i-th operation.

- If $op = +$ then two integers x and y ($1 \le x \le 10^9$, $1 \le y < H$) follow, indicating the position of the wooden board to be added. It's guaranteed that there is currently no board located at (x, y).
- If $op = -$ then two integers x and y ($1 \le x \le 10^9$, $1 \le y < H$) follow, indicating the position of the wooden board to be removed. It's guaranteed that there is currently a board located at (x, y).
- If $op =?$ then four integers x, y, v_y and g ($1 \le x, g \le 10^9$, $1 \le y < H$, $v_y \in \{-1, 1\}$) follow, indicating the initial position of the pinball, the initial velocity along the y-axis of the pinball and the target x-coordinate. It's guaranteed that there is currently no board located at (x, y).

It's guaranteed that neither the sum of n nor the sum of q of all test cases will exceed 2×10^5.

Output

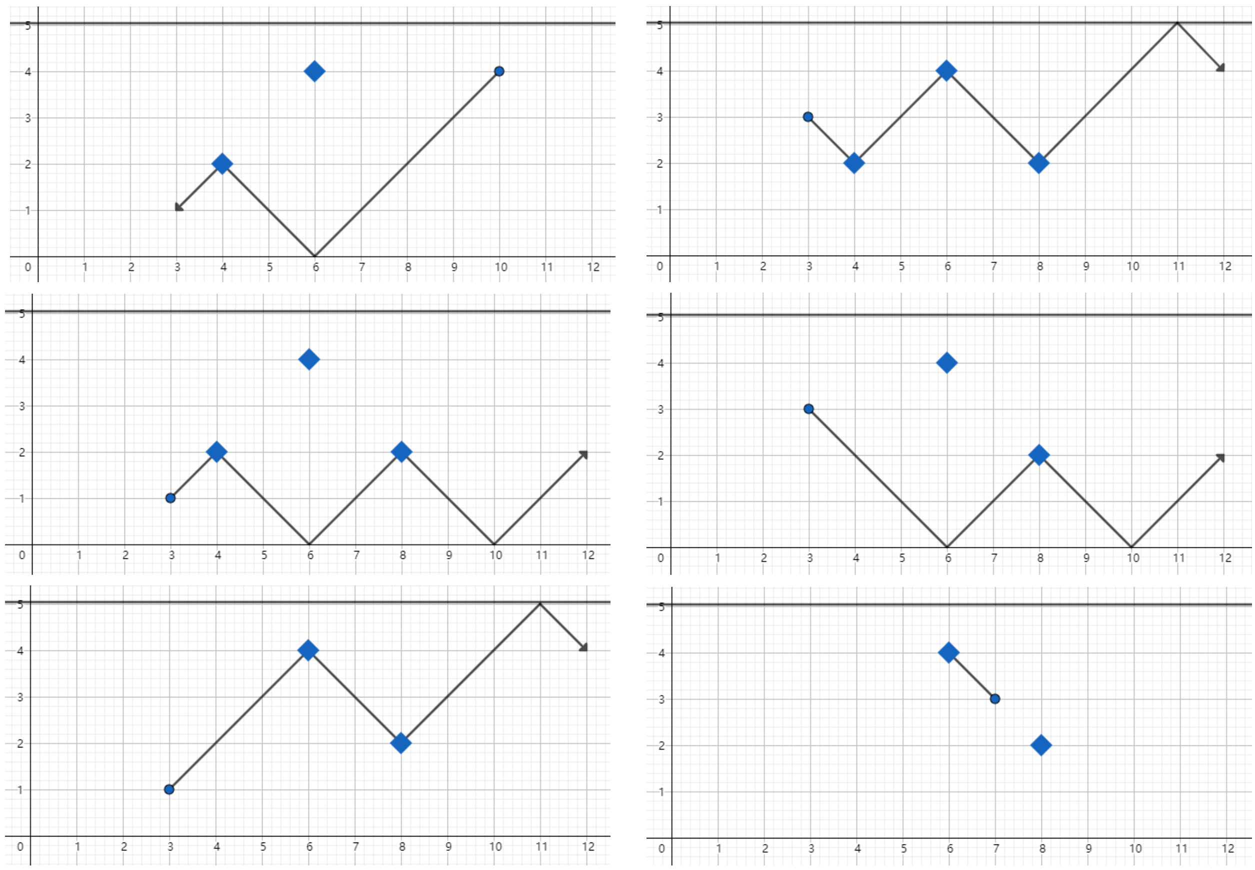
For each operation of the third type output one line containing one integer indicating the answer.

Example

standard input	standard output
2 5 2 8 4 2 6 4 ? 10 4 -1 3 + 8 2 ? 3 3 -1 12 ? 3 1 1 12 - 4 2 ? 3 3 -1 12 ? 3 1 1 12 ? 7 3 1 6 10 1 2 5 5 ? 9 2 1 9 ? 9 2 -1 9	1 4 2 2 4 4 2 2

Note

We illustrate the queries of the first sample test case as below. Wooden boards are represented as diamonds.



Problem C. Bracket Integer

Let $(_w$ and $)_w$ be a round bracket of weight w. Define the balanced weighted bracket sequence (BWBS) as follows:

- An empty sequence is a BWBS.
- If P is a BCBS, then $(_w P)_w$ is also a BCBS. That is, the two brackets at the beginning and the end have the same weight.
- If P and Q are both BWBS, then PQ is also a BWBS.

For example, $(_1(_3)_3)_1$ and $(_5(_7)_7(_2)_2)_5$ are both BWBS, while $(_1(_3)_1)_3$ and $)_5(_5(_7)_7$ are not.

Consider a positive integer with n digits without leading zeros, where its i-th most significant digit is d_i (that is, the integer is equal to $\sum_{i=1}^{n} d_i \times 10^{n-i}$). We say the integer is a bracket integer, if there exists a balanced weighted bracket sequence where the i-th bracket has weight d_i. For example, 1000022122 is a bracket integer, as there exists such BWBS $(_1(_0(_0)_0)_0(_2)_2)_1(_2)_2$.

Given an integer A, find the largest bracket integer smaller than or equal to A.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases. For each test case:

The first and only line contains an integer A ($11 \le A < 10^{2\times 10^5}$).

It's guaranteed that the total number of digits of A of all test cases does not exceed 2×10^5.

Output

For each test case output one line containing one integer, indicating the largest bracket integer smaller than or equal to A. As 11 is a bracket integer, the answer always exists. Note that bracket integers must not have leading zeros.

Example

standard input	standard output
4 20250525 11451419198100 1001 1000	20244202 11451418814154 1001 99

Problem D. Distributed System

A distributed system has n worker nodes numbered from 0 to $(n-1)$. The system is going to process q tasks, where the i-th task can be represented by two integers a_i and b_i, meaning that the task has a_i subtasks numbered from 0 to $(a_i - 1)$, and subtask j will be assigned to worker $(b_i + j) \bmod n$.

For each $0 \le k < n$, calculate the total number of subtasks assigned to worker k after all tasks are processed.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases. For each test case:

The first line contains two integers n and q ($1 \le n, q \le 2 \times 10^5$), indicating the number of worker nodes and the number of tasks.

For the following q lines, the i-th line contains two integers a_i and b_i ($1 \le a_i \le 10^9$, $0 \le b_i < n$), indicating the i-th task.

It's guaranteed that neither the sum of n nor the sum of q of all test cases will exceed 2×10^5.

Output

For each test case output one line containing n integers $v_0, v_1, \cdots, v_{n-1}$ separated by a space, where v_k is the total number of subtasks assigned to worker k after all tasks are processed.

Example

standard input	standard output
2 7 3 10 0 4 2 21 1 1 2 200 0 100 0	5 5 6 5 5 5 4 300

Problem E. Greatest Common Divisor

Given a positive integer sequence $a_1, a_2, \cdots, a_n$ of length n, you need to perform exactly k operations. For each operation, you need to choose one element and increase its value by 1.

Maximize the greatest common divisor of all elements after the operations.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^3$), indicating the number of test cases. For each test case:

The first line contains two integers n and k ($1 \le n \le 10^6$, $1 \le k \le 10^{12}$) indicating the length of the sequence and the number of operations.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le 10^6$) indicating the sequence.

It's guaranteed that the sum of n of all test cases will not exceed 10^6, the sum of $\max a_i$ of all test cases will not exceed 10^6, and the sum of k of all test cases will not exceed 10^{12}.

Output

For each test case output one line containing one integer, indicating the maximum possible greatest common divisor of all elements after exactly k operations.

Example

standard input	standard output
2 3 6 2 9 8 3 7 2 9 8	5 2

Note

For the first sample test case, we can change the sequence to $5, 10, 10$, and the greatest common divisor is 5.

For the second sample test case, we can change the sequence to $6, 10, 10$, and the greatest common divisor is 2.

Problem F. ACE String

We say a string $s_1 s_2 \cdots s_k$ of length k is an ACE string, if it can be divided into five non-empty substrings, such that the first one, the third one and the fifth one are equal.

More formally, the string is an ACE string if there exists two positive integers p and q satisfying all the following constraints:

- $3p + 2 \le k$.
- $p + 2 \le q \le k - 2p$.
- For all $1 \le i \le p$, $s_i = s_{q+i-1} = s_{k-p+i}$.

Given a string of length n, find the length of the longest ACE substring, or report that such substring does not exist.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 3 \times 10^5$), indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 3 \times 10^5$), indicating the length of the string.

The second line contains a string $s_1 s_2 \cdots s_n$ of length n consisting of lowercase English letters.

It's guaranteed that the sum of n of all test cases does not exceed 3×10^5.

Output

For each test case, output one line containing one integer, indicating the length of the longest ACE substring. If such substring does not exist, output `0` instead.

Example

standard input	standard output
3 9 abcabcabc 6 abaaaa 1 a	8 6 0

Note

For the first sample test case, the longest ACE substring is `abcabcab`, as it can be divided into `ab`, `c`, `ab`, `c`, and `ab`, where the first, third, and fifth substrings are equal.

For the second sample test case, the longest ACE substring is `abaaaa`, as it can be divided into `a`, `a`, `a`, `aa`, and `a`, where the first, third, and fifth substrings are equal.

Problem G. Assembly Line

There are k workers along an assembly line, numbered from 1 to k, each with an inbox. They're going to process n artifacts, where the i-th artifact will be added into the inbox of the w_i-th worker at the beginning of the t_i-th minute. Each minute, the following events happen in order:

1. New artifacts (if any) are added to the inbox of some workers.
2. If the inbox of an worker is not empty, he/she grabs an artifact from the box and processes it. This event happens simultaneously for all workers.
3. If a worker has just processed an artifact:
   - For worker $1 \le i < k$, he/she puts the artifact into the inbox of worker $(i + 1)$.
   - For worker k, he/she puts the artifact into the shipping box, and this artifact is completed.

   This event also happens simultaneously for all workers.

How many minutes do these workers need to complete all n artifacts?

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases. For each test case:

The first line contains two integers n and k ($1 \le n \le 2 \times 10^5$, $1 \le k \le 10^9$), indicating the number of artifacts and the number of workers.

For the following n lines, the i-th line contains two integers w_i and t_i ($1 \le w_i \le k$, $1 \le t_i \le 10^9$), indicating that the i-th artifact will be put into the inbox of the w_i-th worker at the beginning of the t_i-th minute.

It's guaranteed that the sum of n of all test cases will not exceed 2×10^5.

Output

For each test case output one line containing one integer, indicating the number of minutes needed to complete all n artifacts.

Example

standard input	standard output
2 4 3 3 2 2 1 3 2 1 2 2 10 1 7 4 20	5 26

Note

We now explain the first sample test case. The following chart shows the number of artifacts in all inboxes after the 1-st and the 3-rd event in each minute. The numbers are given by a sequence $\{c_1, c_2, c_3\}$, where c_i is the number of artifacts in the inbox of the i-th worker.

Minute	**Event** 1	**Event** 3
1	$\{0, 1, 0\}$	$\{0, 0, 1\}$
2	$\{1, 0, 3\}$	$\{0, 1, 2\}$
3	$\{0, 1, 2\}$	$\{0, 0, 2\}$
4	$\{0, 0, 2\}$	$\{0, 0, 1\}$
5	$\{0, 0, 1\}$	$\{0, 0, 0\}$

Problem H. Minimum Spanning Tree

Given an undirected connected graph with n vertices and m weighted edges, you can add at most k more edges to the graph. If the edge you add connects vertices u and v, then it will have a weight of $|u - v|$.

Minimize the total weight of edges in the minimum spanning tree.

Note that the graph may contain self loops or multiple edges, and it is allowed to add an edge between two vertices when there is already one or more edges between them.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$) indicating the number of test cases. For each test case:

The first line contains three integers n, m, and k ($2 \le n \le 2 \times 10^5$, $n - 1 \le m \le 2 \times 10^5$, $0 \le k \le 2 \times 10^5$) indicating the number of vertices in the graph, the number of edges in the graph, and the number of edges you can add.

For the following m lines, the i-th line contains three integers u_i, v_i, and w_i ($1 \le u_i, v_i \le n$, $0 \le w_i \le 10^9$) indicating an edge connecting vertices u_i and v_i with weight w_i.

It is guaranteed that the given graph is connected. It is also guaranteed that the sum of n, the sum of m, and the sum of k of all test cases will not exceed 2×10^5.

Output

For each test case:

First, output one line containing one integer c ($0 \le c \le k$), indicating the number of edges you want to add.

Then output c lines, where the i-th line contains two integers u_i and v_i ($1 \le u_i, v_i \le n$), indicating that the i-th edge you add connects vertices u_i and v_i, and has a weight of $|u_i - v_i|$.

Then output one line containing one integer, indicating the minimum total weight of the minimum spanning tree.

Then output one line containing $(n-1)$ distinct integers $e_1, e_2, \cdots, e_{n-1}$ ($1 \le e_i \le m + c$) separated by a space, indicating the indices of the edges on the minimum spanning tree. Note that $1 \le e_i \le m$ indicates an edge in the original graph, and $e_i > m$ indicates the $(e_i - m)$-th edge you add.

Example

standard input	standard output
3 5 6 2 1 2 3 2 3 5 1 4 7 4 2 4 5 4 8 3 5 1 4 5 100 1 2 2 2 3 0 2 4 0 4 1 1 3 4 3 3 2 0 1 2 100 2 3 100	2 2 3 3 4 6 8 1 6 7 0 1 2 3 4 0 200 1 2

Problem I. Square Puzzle

Consider a grid with 3 rows and 3 columns. Each cell of the grid has an integer in it. Each integer from 1 to 9 (both inclusive) appears exactly once in the grid.

You can perform the following three types of operations on the grid.

Operation	Description	Example
Right-shift one row	Choose one row and move its right-most element to the left-most position.	Right-shift the third row 2 9 3 2 9 3 6 7 8 → 6 7 8 5 1 4 4 5 1
Down-shift one column	Choose one column and move its bottom-most element to the top-most position.	Down-shift the first column 2 5 7 8 5 7 4 9 1 → 2 9 1 8 6 3 4 6 3
Rotate clockwise	Rotate the whole grid 90 degrees clockwise.	Rotate clockwise 5 7 3 8 1 5 1 4 9 → 2 4 7 8 2 6 6 9 3

Given two such grids, calculate the minimum number of operations needed to transform the first grid to the second one.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 2 \times 10^5$), indicating the number of test cases. For each test case:

For the first three lines, the i-th line contains a string $a_{i,1}a_{i,2}a_{i,3}$ ($1 \le a_{i,j} \le 9$), indicating the i-th line of the first grid.

For the following three lines, the i-th line contains a string $b_{i,1}b_{i,2}b_{i,3}$ ($1 \le b_{i,j} \le 9$), indicating the i-th line of the second grid.

Output

For each test case output one line. If it is possible to transform the first grid to the second one, output an integer indicating the minimum number of operations needed; If it is impossible to do so, output `-1` instead.

Example

standard input	standard output
4 293 678 514 624 579 183 624 579 183 293 678 514 123 456 789 321 456 789 123 456 789 123 456 789	3 5 -1 0

Note

For the first sample test case, as shown in the description, we can first right-shift the third row, then down-shift the first column, and finally rotate clockwise.

Problem J. Useful Algorithm

"Stop learning useless algorithms, go and solve some problems, learn how to use binary search."

– Um_nik

Binary search is a very useful algorithm that has applications in various problems. The following pseudocode demonstrates one implementation of binary search. The function `BINARYSEARCH` takes two parameters A and k, where $A = a_1, a_2, \cdots, a_n$ is a strictly increasing integer sequence of length n, and k is an integer that appears in sequence A. The function returns the index of integer k in sequence A. Note that in the following pseudocode, $\lfloor x \rfloor$ is the largest integer smaller than or equal to x.

Algorithm 1 The Binary Search Algorithm

```
function BinarySearch(A, k)
    l ← 1
    r ← n
    while l < r do
        m ← ⌊(l+r)/2⌋
        if a_m ≥ k then
            r ← m
        else
            l ← m + 1
        end if
    end while
    return l
end function
```

The use of binary search has a prerequisite, which is, the sequence being searched must be sorted. However, in this problem, we will temporarily ignore this constraint and study the performance of binary search on arbitrary sequences.

Given two integers n and k ($1 \le k \le n$), a permutation $P = p_1, p_2, \cdots, p_n$ of n is good if we can correctly find the index of integer k in permutation P with binary search. In other words, let $i =$ `BINARYSEARCH(`P`,` k`)`, then P is good if $p_i = k$.

We now randomly picks a permutation of n with equal probability. Calculate the probability to pick a good permutation.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$) indicating the number of test cases. For each test case:

The first and only line contains two integers n and k ($1 \le k \le n \le 10^9$).

Output

For each test case, output one line containing one integer, indicating the probability to pick a good permutation modulo $(10^9 + 7)$.

It can be proven that the answer is a rational number $\frac{P}{Q}$. You need to output the value of $PQ^{-1} \bmod (10^9 + 7)$, where Q^{-1} is the integer that satisfies $QQ^{-1} \bmod (10^9 + 7) = 1$.

Example

standard input	standard output
4 3 2 3 1 3 3 1 1	500000004 333333336 666666672 1

Note

For the first sample test case, permutations $\{1, 2, 3\}$, $\{2, 3, 1\}$, and $\{3, 1, 2\}$ are all good. So the answer is $\frac{3}{3!} = \frac{1}{2}$. Since $2 \times 500000004 \bmod (10^9 + 7) = 1$, you should output $1 \times 500000004 \bmod (10^9 + 7) = 500000004$.

Problem K. Path Planning 2

There is a grid with n rows and m columns. Each cell of the grid has an integer in it, where $a_{i,j}$ indicates the integer in the cell located at the i-th row and the j-th column.

Let (i, j) be the cell located at the i-th row and the j-th column. You now start from $(1, 1)$ and need to reach (n, m). When you are in cell (i, j), you can either move to its right cell $(i, j + 1)$ if $j < m$ or move to its bottom cell $(i + 1, j)$ if $i < n$.

Let $\mathbb{S}$ be the set consisting of integers in each cell on your path, including $a_{1,1}$ and $a_{n,m}$. Let $\text{mex}(\mathbb{S})$ be the smallest non-negative integer which does not belong to $\mathbb{S}$. Find a path to minimize $\text{mex}(\mathbb{S})$ and calculate this minimum possible value.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$) indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n, m \le 10^6$, $1 \le n \times m \le 10^6$) indicating the number of rows and columns of the grid.

For the following n lines, the i-th line contains m integers $a_{i,1}, a_{i,2}, \cdots, a_{i,m}$ ($0 \le a_{i,j} \le 10^9$) where $a_{i,j}$ indicates the integer in cell (i, j).

It's guaranteed that the sum of $n \times m$ of all test cases will not exceed 10^6.

Output

For each test case output one line containing one integer indicating the minimum possible value of $\text{mex}(\mathbb{S})$.

Example

standard input	standard output
2 2 3 2 0 1 0 3 4 1 5 100 0 2 0 1	1 3

Note

For the first sample test case there are 3 possible paths.

- The first path is $(1, 1) \to (1, 2) \to (1, 3) \to (2, 3)$. $\mathbb{S} = \{2, 0, 1, 4\}$ so $\text{mex}(\mathbb{S}) = 3$.
- The second path is $(1, 1) \to (1, 2) \to (2, 2) \to (2, 3)$. $\mathbb{S} = \{2, 0, 3, 4\}$ so $\text{mex}(\mathbb{S}) = 1$.
- The third path is $(1, 1) \to (2, 1) \to (2, 2) \to (2, 3)$. $\mathbb{S} = \{2, 0, 3, 4\}$ so $\text{mex}(\mathbb{S}) = 1$.

So the answer is $\min(3, 1, 1) = 1$.

For the second sample test case there is only 1 possible path, which is $(1, 1) \to (1, 2) \to (1, 3) \to (1, 4) \to (1, 5)$. $\mathbb{S} = \{100, 0, 2, 1\}$ so $\text{mex}(\mathbb{S}) = 3$.

Problem L. Stella

In astronomy, stellar classification is the classification of stars based on their spectral characteristics. Most stars are currently classified under the Morgan-Keenan (MK) system using the letters O, B, A, F, G, K, and M, a sequence from the hottest (O type) to the coolest (M type). Each letter class is then subdivided using a numeric digit with 0 being hottest and 9 being coolest. For example, A8, A9, F0, and F1 form a sequence from hotter to cooler.

It obvious that 7 letters and 10 digits give us a total of 70 different classes, where O0 is the hottest and M9 is the coolest. Given two classes, determine if the first class is hotter than, cooler than, or the same with the second class.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \leq T \leq 10^3$), indicating the number of test cases. For each test case:

The first and only line contains two strings of length 2, indicating the two classes.

Output

For each test case output one line.

- If the first class is hotter than the second class, output `hotter`.
- If the first class is cooler than the second class, output `cooler`.
- If the first class is the same with the second class, output `same`.

Example

standard input	standard output
5 O2 O7 M9 M2 G2 G2 A0 B9 F8 K1	hotter cooler same cooler hotter

Problem M. Triangulation

There are n points placed at equal distances on a circle, numbered from 1 to n in clockwise direction starting from a certain point. Let d be the length of arc between two neighboring points.

BaoBao draws $(2n-3)$ chords on the circle, each connecting two of the n points. It's guaranteed that no two chords connect the same pair of points, and no two chords intersect inside the circle (excluding the border). It's easy to see that, these chords form $(n-2)$ triangles. These triangles are called a triangulation of the circle.

BaoBao likes the triangulation a lot, so he records the $(n-2)$ triangles on a piece of paper. The i-th triangle is recorded as three integers $k_{i,1}, k_{i,2}, k_{i,3}$, where $k_{i,j} \times d$ is the length of the shortest arc between the j-th vertex and its next vertex on the i-th triangle. For $j = 1$ and $j = 2$, its next vertex is the $(j+1)$-th vertex. For $j = 3$, its next vertex is the 1-st vertex.

Several days later, when BaoBao wants to admire the triangulation again, he finds with dismay that all chords are erased by his roommate DreamGrid, so he has to recover the triangulation according to the paper. Your task is to help BaoBao determine $v_{i,j}$ for all $1 \le i \le n$ and $1 \le j \le 3$, meaning that the j-th vertex of the i-th triangle is the $v_{i,j}$-th point on the circle. It is possible that BaoBao recorded the wrong information, in this case you should tell BaoBao that such triangulation does not exist.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases. For each test case:

The first line contains an integer n ($3 \le n \le 2 \times 10^5$), indicating the number of points on the circle.

For the following $(n-2)$ lines, the i-th line contains three integers $k_{i,1}, k_{i,2}, k_{i,3}$ ($1 \le k_{i,j} \le \lfloor \frac{n}{2} \rfloor$), where $k_{i,j} \times d$ is the length of the shortest arc between the j-th vertex and its next vertex on the i-th triangle.

It's guaranteed that the sum of n of all test cases will not exceed 2×10^5.

Output

For each test case:

- If such triangulation exists, first output `Yes` in one line. Then output $(n-2)$ lines, where the i-th line contains three integers $v_{i,1}, v_{i,2}, v_{i,3}$ separated by a space, meaning that the j-th vertex of the i-th triangle is the $v_{i,j}$-th point on the circle. If there are multiple valid answers, you can output any of them.
- If such triangulaton does not exist, just output `No` in one line.

Example

standard input	standard output
3 3 1 1 1 4 1 1 1 1 2 1 6 3 1 2 1 2 1 1 3 2 1 2 1	Yes 3 2 1 No Yes 2 5 4 1 2 6 6 5 2 3 2 4

Note

The first and the third sample test cases are illustrated below.