The 2nd Universal Cup

Finals

February 22, 2025

This problem set should contain 13 problems on 23 numbered pages.

- A. Traveling in Cells 2
- B. Exchanging Kubic 3
- C. Longest Increasing Subsequence
- D. Puzzle: Nurikabe
- E. Quad Kingdoms Chess 2
- F. World of Rains
- G. Circular Parterre
- H. Homeward Glance
- I. Deciding Game
- J. Divide the String
- K. Master of Modular Arithmetic
- L. Not Another Constructive Problem
- M. Dividing Chains

Problem A. Traveling in Cells 2

Little Cyan Fish (小青鱼) has a friend called Wanwan (绾绾). Wanwan is on a large grid. The coordinates of the lower left corner are $(0, 0)$, and the coordinates of the upper right corner are $(10^6, 10^6)$.

Initially, there are n obstacles on the grid, where the i-th obstacle is located in $(x_i^{(O)}, y_i^{(O)})$. All the obstacles are not on the boundary of the grid, i.e. $0 < x_i^{(O)}, y_i^{(O)} < 10^6$.

Wanwan can only move one step up or one step right at a time. At any time, Wanwan cannot occupy a cell that contains an obstacle.

Formally, a *path* from the cell (x_s, y_s) to the cell (x_t, y_t) can be represented by a series of cells $P : (a_1, b_1), (a_2, b_2), \ldots, (a_\ell, b_\ell)$, where:

- $a_1 = x_s, b_1 = y_s$ and $a_\ell = x_t, b_\ell = y_t$.
- For all $1 \le i < \ell$, exactly one of the following condition applies:
  1. $a_{i+1} = a_i$ and $b_{i+1} = b_i + 1$;
  2. $a_{i+1} = a_i + 1$ and $b_{i+1} = b_i$.
- For all $1 \le i \le \ell$, (a_i, b_i) is not an obstacle.

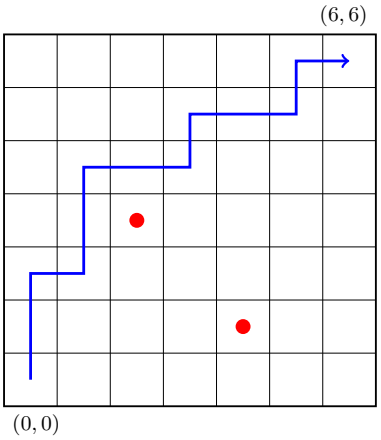

Figure 1: A valid path. Obstacles are marked as red dots.

Of course, Wanwan cannot visit any of cells with the obstacles. However, there were several more cells that was not covered by any valid path. Any cell that cannot be part of **any** valid path from $(0, 0)$ to $(10^6, 10^6)$ becomes a new obstacle. This process repeats until every free cell (i.e., a cell not containing an obstacle) can be included in some valid path P from $(0, 0)$ to $(10^6, 10^6)$.

For example, in the figure below, obstacles that appeared initially are marked as red dots, and the obstacles added in the process are marked as orange dots.

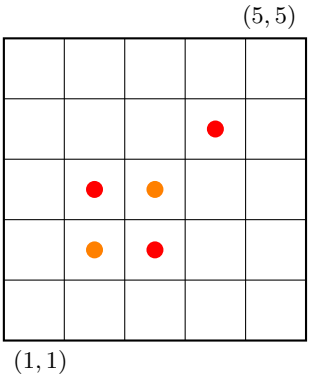

Figure 2: A grid after adding all additional obstacles.

Next, after adding all additional obstacles, Little Cyan Fish needs to complete q random walk queries. For the i-th walk query, Little Cyan Fish takes all the path from $(x_i^{(s)}, y_i^{(s)})$ to $(x_i^{(t)}, y_i^{(t)})$. Little Cyan Fish wants to know how many cells are accessible by at least one of the path.

Formally, a cell (x, y) should be counted if there exists a path $P : (a_1, b_1), \cdots, (a_\ell, b_\ell)$, such that $(a_1, b_1) = (x_i^{(s)}, y_i^{(s)})$, $(a_\ell, b_\ell) = (x_i^{(t)}, y_i^{(t)})$, and $(a_i, b_i) = (x, y)$ for some $1 \le i \le \ell$.

Input

The first line of the input contains two integers n and q ($1 \le n, q \le 10^5$).

The next n lines of the input describe all the obstacles **existed initially**. The i-th line of these lines contains two integers $x_i^{(O)}$ and $y_i^{(O)}$ ($1 \le a_i, b_i < 10^6$) , indicating the coordinates of the i-th obstacle.

The next q lines of the input describe all the queries. The i-th line of these lines contains four integers $x_i^{(s)}$, $y_i^{(s)}$, $x_i^{(t)}$, and $y_i^{(t)}$ ($1 \le x_i^{(s)} \le x_i^{(t)} < 10^6$, $1 \le y_i^{(s)} \le y_i^{(t)} < 10^6$) , indicating the i-th task.

Output

For each walk task, output a single line with a single integer, indicating the answer.

Examples

standard input	standard output
3 3 2 3 3 2 4 4 1 1 5 5 3 4 5 5 3 3 4 5	20 4 0
8 2 2 2 2 5 3 5 4 1 5 4 6 1 7 3 8 2 1 1 9 5 5 1 8 5	28 11

Note

In the first example, there are three obstacles initially (marked as red dots), and two more obstacles added after Wanwan's first trip (marked as orange dots).

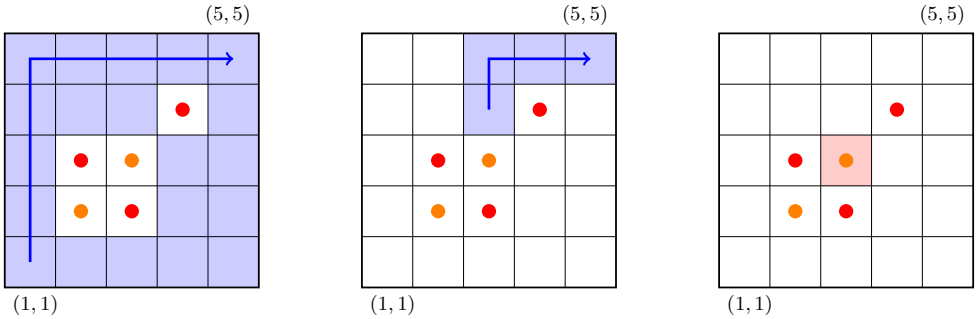

Figure 3: The figure corresponding the first example.

For each of the query, all the accessible cells are marked as blue. In the third query, as $(x_i^{(s)}, y_i^{(s)})$ has been occupied with an obstacle, there were no cells can be accessible by any of the path.

Problem B. Exchanging Kubic 3

Little Cyan Fish is conducting a social experiment with Prof. Kubic. In the experiment, there is a row of n cells numbered from 1 to n. An integer array a of length n describes the distribution of soldiers in these cells. For each cell i:

- If $a_i = 0$, cell i is empty.
- If $a_i > 0$, cell i contains a_i *good* soldiers.
- If $a_i < 0$, cell i contains $-a_i$ *bad* soldiers.

Little Cyan Fish can apply several operations. In an operation, Little Cyan Fish may choose two indices i and j satisfying $1 \le i, j \le n$ such that $a_i > 0$ and j is adjacent to i (that is, $j \in \{i-1, i+1\}$). Then, the operation moves all soldiers from cell i to cell j by performing the following updates:

$$a_j \leftarrow a_j + a_i, \quad a_i \leftarrow 0.$$

Little Cyan Fish hates the bad soldiers, so he wants to eliminate them all. In other words, he needs to achieve

$$a_i \ge 0 \quad \text{for all } 1 \le i \le n.$$

Determine the minimum number of operations required to reach his goal, or report if it is impossible.

Input

There are multiple test cases. The first line of the input contains a single integer T ($T \ge 1$), indicating the number of the test cases. For each test case:

The first line of the input contains a single integer n ($1 \le n \le 5 \times 10^5$), the number of cells.

The next line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$), representing the soldier distribution in the cells.

It is guaranteed that the sum of n over all test cases does not exceed 5×10^5.

Output

For each test case:

If there is no way to reach Little Cyan Fish's goal, output a single line with a single word "`No`".

Otherwise, the first line of the output should contain the word "`Yes`".

The next line of the output should contain a single integer, indicating the minimum number of operations required to ensure that $a_i \ge 0$ for all $1 \le i \le n$.

Example

standard input	standard output
4 2 -2 1 3 1 0 -1 5 -1 4 -1 -1 -1 6 -1 2 -1 -1 3 -1	No Yes 2 Yes 5 Yes 5

Problem C. Longest Increasing Subsequence

Little Cyan Fish loves the concept of LIS, and he wants you to construct many, many different LIS!

Before our story begins, recall that *subsequence* is a sequence obtained by removing any number of elements (probably zero) from the original sequence. For example, 4 3 5 is a subsequence of 2 4 1 3 5. An LIS, or *longest increasing subsequence*, is the longest **monotonically increasing** subsequence of a given sequence. We use $\mathrm{LIS}(a)$ to indicate the **length** of the LIS of the sequence a. For example, $\mathrm{LIS}(2\ 4\ 1\ 3\ 5) = 3$.

For two integer sequences $p_1, p_2, \ldots, p_k$ and $q_1, q_2, \ldots, q_l$, the sequence $p+q$ is another sequence r obtained by concatenating the sequence p and q. More formally, the length of the sequence r will be $k+l$, and the elements of r are defined by:

$$r_i = \begin{cases} p_i & i \le k \\ q_{i-k} & \text{otherwise} \end{cases}, \text{ for all } 1 \le i \le k+l$$

Now, Little Cyan Fish has a sequence $a_1, a_2, \ldots, a_n$ of n integers, such that $1 \le a_i \le m$ for all $1 \le i \le n$. Now, he wants you to find another sequence b of length $m-n$, such that:

- $1 \le b_i \le m$.
- All the integers in a and b are distinct. In other words, $a+b$ is a permutation of length m.
- $\mathrm{LIS}(a+b) = \mathrm{LIS}(b+a)$.

Input

There are multiple test cases. The first line of the input contains a single integer T ($T \ge 1$), indicating the number of the test cases. For each test case:

The first line of the input contains two integers n and m ($1 \le n < m \le 10^6$).

The next line of the input contains n distinct integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le m$).

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

Output

For each test case:

If it is impossible to get any possible plan, output a single line "No".

Otherwise, the first line of the output should contain the word "Yes".

The next line of the output should contain $m-n$ integers, representing the constructed sequence $b_1, b_2, \ldots, b_{m-n}$.

Example

standard input	standard output
3 3 6 2 6 5 7 12 3 7 6 9 10 2 1 3 6 1 2 3	Yes 3 1 4 Yes 4 5 8 12 11 No

Problem D. Puzzle: Nurikabe

"*In my next contest, I want you to solve a nurikabe puzzle with only two clues*"
"*Sounds too easy. Let me write a solution!*"
"*Ah, are you sure? Have you solved a problem called Fillomino by Yuhao?*"
"*Hmm, wait...*"
— Little Cyan Fish and a random judge

Little Cyan Fish is a fan of logic puzzles. Today, he is playing a special version of the classic puzzle "Nurikabe" (ぬりかべ).

The puzzle is played on an $n \times m$ grid. We refer to the cell in the x-th row and y-th column as (x, y), where $1 \le x \le n$ and $1 \le y \le m$.

The goal is to shade some (possibly zero) cells *black*, leaving the remaining cells *white*. The black and white cells form connected *regions*, where cells are considered connected if they share a side (horizontally or vertically, not diagonally). Formally, two cells with the same color $C_1(x_1, y_1)$ and $C_2(x_2, y_2)$ are considered in the same region if and only if at least one of the following condition holds:

- $|x_1 - x_2| + |y_1 - y_2| = 1$
- There exists another cell $C_3(x_3, y_3)$ with the same color, such that C_1 and C_3 are in the same region, and C_2 and C_3 are in the same region.

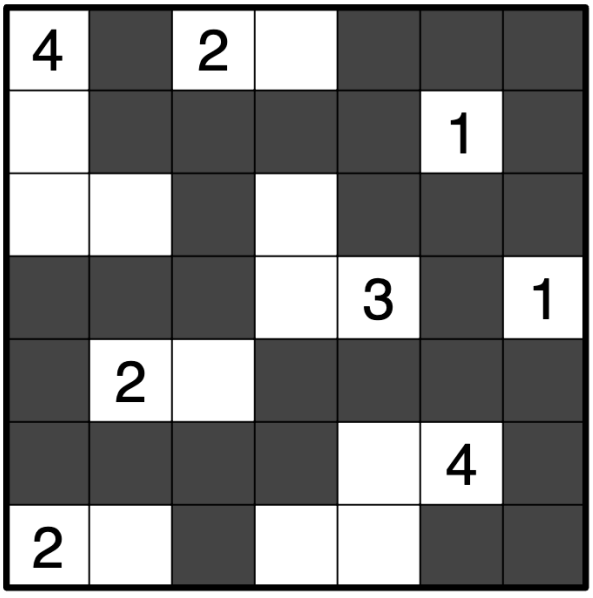

Figure 4: A sample grid for $n = 7$ and $m = 7$.

Clues are those numbers placed in a specific cell. Cells with a *clue* **cannot** be shaded black.

The shaded (black) cells must satisfy the following two conditions:

- **(Connectivity)**: All black cells must belong to a **single** black region.
- **(2×2-free)**: No 2×2 block of cells can be entirely black.

The unshaded (white) cells must satisfy the following two conditions:

- **(One Clue a Region)**: Each white region must contain **exactly one** clue.
- **(Size Constraint)**: The area (number of cells) in each white region must equal the value of the clue within that region.

For example, Little Cyan Fish considers the following four solutions incorrect because:

- In the first solution, there is a 2×2 group of cells shaded in black.
- In the second solution, there is more than one black region.
- In the third solution, the white region in the top-left corner contains more than one clue.
- In the fourth solution, the white region in the bottom-left does not contain any clues, and the white region in the bottom-right corner does not meet the area requirement.

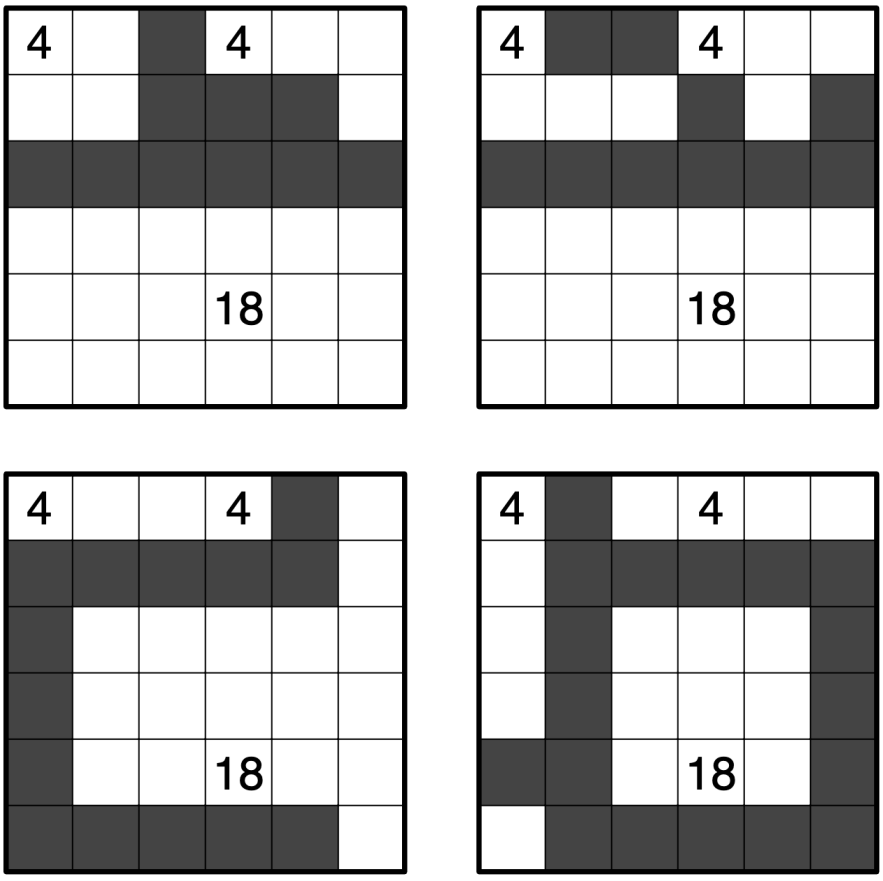

Figure 5: Wrong solutions for a given puzzle

And the following solution is a correct solution.

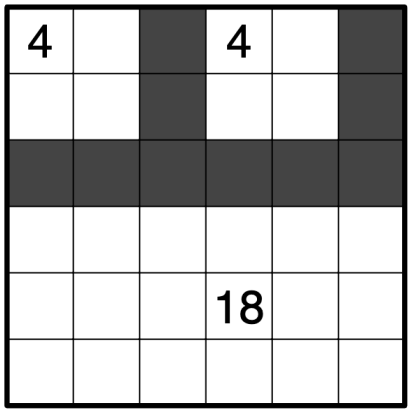

Figure 6: A correct solution for a given puzzle

Now, you are given a nurikabe puzzle with **exactly one clue**. You need to give a solution to this puzzle.

Input

There are multiple test cases. The first line of the input contains a single integer T ($T \geq 1$), indicating the number of the test cases. For each test case:

The first line of the input contains two integers n and m ($n, m \geq 1$), indicating the length and the width of the grid.

The next line of the input contains three integers i, j, x ($1 \leq i \leq n$, $1 \leq j \leq m$, $1 \leq x \leq n \cdot m$), indicating the only one clue located in cell (i, j) with the number x.

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

Output

For each test case:

If it is impossible to shade some cells to satisfy all the requirements, output a single line "`No`".

Otherwise, the first line of the output should contain the word "`Yes`".

The next n lines of the output describe your solution. The i-th line of these lines should contain exactly m characters, either "`.`" (a dot) or "`#`" (a number sign) . The j-th character describes the status of the cell (i, j).

- If the character is "`.`", it means that the cell is unshaded (white).
- If the character is "`#`", it means that the cell is shaded (black).

Example

standard input	standard output
4 3 3 2 2 1 5 5 2 2 1 4 6 1 1 20 2 5 2 5 10	Yes ### #.# ### No Yes .... #..... ###... Yes ....

Note

Want to find some joy after solving all the problems? Here it comes.

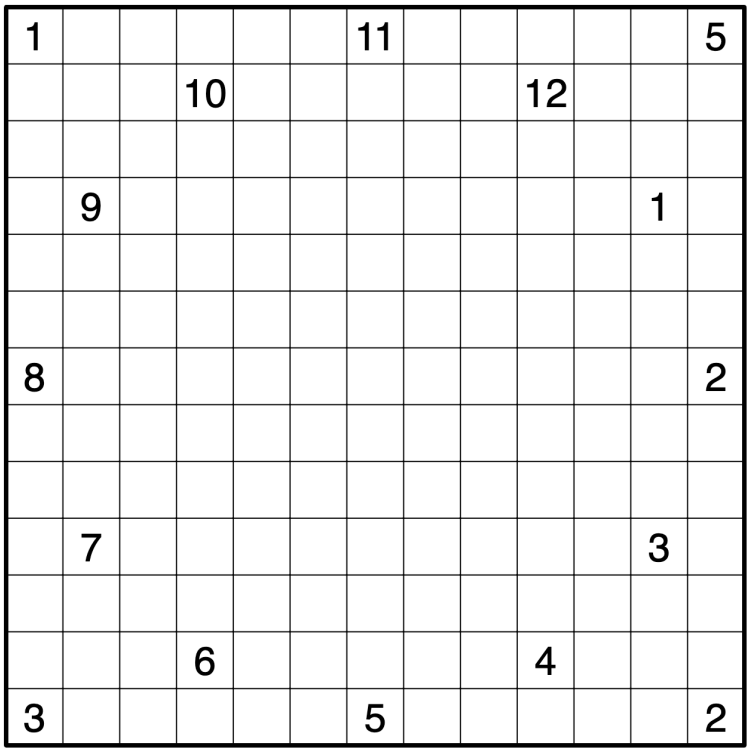

Figure 7: The 2nd Universal Cup Puzzle Contest: Nurikabe (credit to apiad)

Problem E. Quad Kingdoms Chess 2

Quad Kingdoms Chess, or QKC, is very popular in the 605A dormitory of THU (Technical Hammer University).

Little Cyan Fish has many friends in that great university. Today, after a long day of classes, Little X and Little Y decided to play a game of QKC.

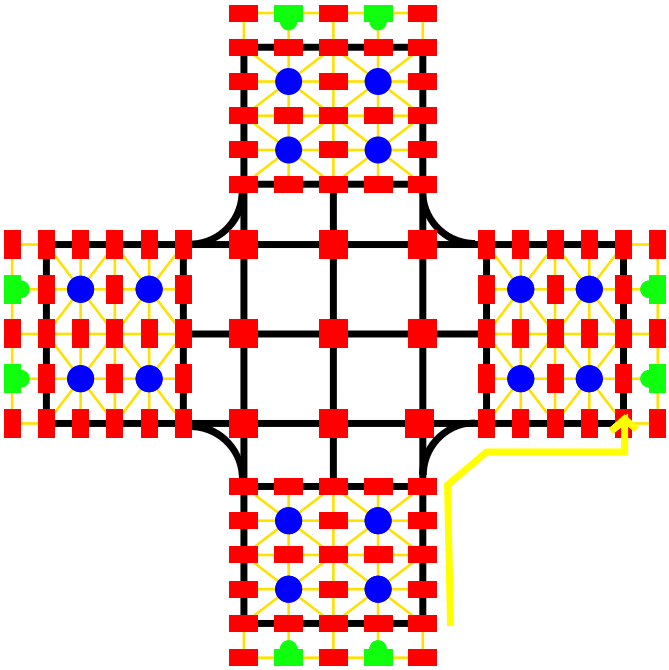

Figure 8: Prepared to play the Quad Kingdom Chess

Unfortunately, they could not gather four players — Little Cyan Fish does not know how to play it! So, to find some joy, they each randomly grabbed a set of pieces and began a trivial game of comparing piece strengths. The simplified game proceeds as follows:

Each piece is represented by an integer $s_i \in [1, 10^5]$. Little X and Little Y each have a collection of pieces. Then, the contest is conducted according to the following rules:

1. Initially, both players arrange their pieces into a sequence, and then shuffle the sequence **uniformly at random**.
2. Then, both players will deploy the first piece in their sequences, denoted as x for Little X and y for Little Y.
3. If $x > y$, Little Y's piece is eliminated, and he must deploy the next piece in his sequence.
4. If $x < y$, Little X's piece is eliminated, and he must deploy the next piece in his sequence.
5. If $x = y$, both pieces are eliminated, and both players deploy the next piece in their sequences.

If at any moment one player cannot deploy a piece while the other still has at least one remaining, the player with remaining pieces wins; if both players run out of pieces simultaneously, the game is declared a draw.

Little Cyan Fish found the game was too simple! So before shuffling the collection, he secretly inserted k **additional** *bombs* into Little X's collection (that is, Little X now has $n + k$ pieces, with n normal pieces and k bombs). When playing the game, a *bomb* always results in mutual elimination when it battles any piece.

Given the collections of pieces for both players, Little Cyan Fish wants to know the probabilities of the game ending with Little X winning, Little Y winning, or drawing, modulo 998 244 353.

Input

The first line of the input contains three integers n, m, and k ($1 \le n, m \le 1000$, $0 \le k \le 20$), indicating the number of non-bomb pieces Little X has, the number of pieces Little Y has, and the number of bombs Little Cyan Fish inserted, respectively.

The next line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^5$), indicating the sizes of Little X's non-bomb pieces.

The next line of the input contains m integers $b_1, b_2, \ldots, b_m$ ($1 \le b_j \le 10^5$), indicating the sizes of Little Y's pieces.

Output

The first line of the output contains a single integer P_X, indicating the probability of Little X winning, modulo 998 244 353.

The next line of the output contains a single integer P_Y, indicating the probability of Little Y winning, modulo 998 244 353.

The next line of the output contains a single integer $P_=$, indicating the probability of a draw, modulo 998 244 353.

Examples

standard input	standard output
5 5 1 11 4 2 11 9 20 1 6 20 1	0 1 0
8 7 5 58 42 34 58 12 12 9 1 28 59 59 1 36 14 7	695057094 239873545 63313715
8 7 5 9 1 2 12 7 7 7 6 15 4 15 1 15 11 13	673575784 13961460 310707110

Problem F. World of Rains

Little Cyan Fish is exploring a world of rain! A two-dimensional world is represented as an $N \times M$ grid. We denote the cell in the i-th row and j-th column as (i, j), where $1 \le i \le N$ and $1 \le j \le M$.

A magical cat controls the rain in this world. The rainfall process spans $S+1$ seconds, numbered from 1 to $S+1$. At the beginning (time 0, before the first second), no cell contains any water.

Each second i $(1 \le i \le S+1)$, the following happens:

1. **Rainfall:** For each cell that **does not** currently contain a water droplet, the cat independently and randomly decides whether to create a new water droplet in that cell.
2. **Movement (except at time $S+1$):** If $i \le S$, all existing water droplets move simultaneously due to gravity and wind. A droplet at cell (x, y) moves to cell $(x+1, y+d_i)$.
3. **Disappearance:** Any droplet that moves outside the bounds of the $N \times M$ grid disappears permanently.

Your task is to calculate the number of distinct possible rainfall scenarios. Two scenarios are considered different if and only if there exists at least one cell (i, j) and one time t $(1 \le t \le S+1)$ where a droplet is created in that cell at that time in one scenario, but not in the other.

Since the number of scenarios can be very large, output the result modulo 998 244 353.

Input

There are multiple test cases. The first line of the input contains a single integer T $(T \ge 1)$, indicating the number of the test cases. For each test case:

The first line of the input contains three integers N, M, and S $(1 \le N, M, S \le 5 \times 10^5)$.

The next line of the input contains S integers: $d_1, d_2, \ldots, d_S$ $(-10^9 \le d_i \le 10^9)$.

It is guaranteed that the sum of S over all test cases does not exceed 5×10^5.

Output

For each test case:

Output a single line containing a single integer, indicating the answer modulo 998 244 353.

Example

standard input	standard output
3 2 2 1 1 3 3 5 1 0 0 0 -1 9 10 7 1 4 -2 -8 5 -7 142857	192 536867776 736446321

Problem G. Circular Parterre

There are n sprinklers on a two-dimensional Cartesian plane, each of which can water an area inside or on the boundary of a circle.

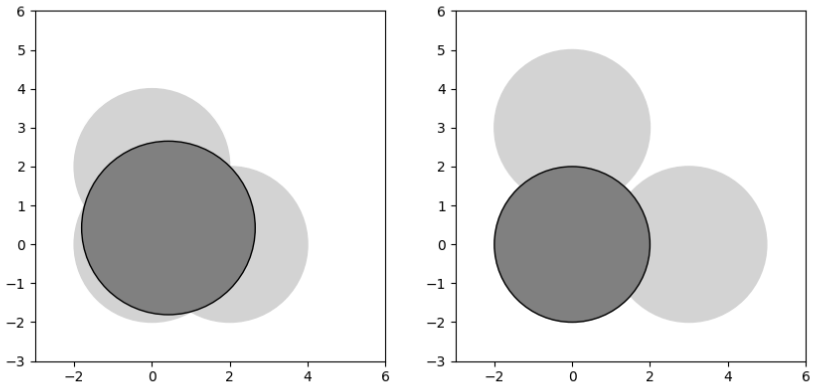

Figure 9: The largest circular parterres in the sample cases.

You need to build the largest circular parterre, such that every point inside or on the boundary of the circular parterre can be watered by some sprinkler.

Input

The first line of the input contains an integer T ($1 \le T \le 300$), indicating the number of test cases. For each test case:

- The first line contains an integer n ($1 \le n \le 300$), indicating the number of sprinklers.
- Then n lines follow, the i-th of which contains three integers x_i, y_i ($-1\,000 \le x_i, y_i \le 1\,000$) and r_i ($1 \le r_i \le 1\,000$), indicating that the i-th sprinkler waters an area inside or on the boundary of the circle centered at (x_i, y_i) with radius r_i.
- It is guaranteed that no two circles are identical or tangent to each other, and the closest pair of points in the (multi)set consisting of the intersection points between any two circles has a distance of no less than 0.01.

It is guaranteed that the sum of n over all test cases does not exceed 300.

Output

For each test case, output a line containing a real number, indicating the radius of the largest circular parterre.

Your answer is acceptable if its absolute or relative error does not exceed 10^{-6}. Formally speaking, suppose that your output is x and the jury's answer is y, your output is accepted if and only if $\frac{|x-y|}{\max(1,|y|)} \le 10^{-6}$.

Example

standard input	standard output
2 3 0 0 2 0 2 2 2 0 2 3 0 0 2 0 3 2 3 0 2	2.230710143300821420 2.000000000000000000

Problem H. Homeward Glance

Oblivionis took one final look at home before departing, resolutely deciding to leave the past behind. She wished that the magic of time could reshape her now chaotic heartbeat into a healthy, harmonious rhythm once again. But before she could truly forget everything, a question lingered in her mind: What if the events of the past had unfolded in a different order? Would the outcome have been different, and would her understanding of it have changed as well?

Given an $n \times n$ matrix A over $\mathbb{F}_{998244353}$, consider how many matrices B commute with A, i.e., how many matrices satisfy $AB = BA$. It can be proven that there exists a positive integer k such that the number of such matrices is 998244353^k. Please determine the value of k.

Input

The first line of the input contains a single integer n ($1 \le n \le 500$).

The next n lines describe the matrix A. The i-th line of these lines contains n integers A_{ij} ($0 \le A_{ij} < 998244353$), indicating the matrix A.

Output

Output a single line with a single integer, indicating the answer, k.

Examples

standard input	standard output
3 1 2 3 4 5 6 7 8 9	3
3 1 1 0 0 1 0 0 0 1	5

Problem I. Deciding Game

Little Cyan Fish is preparing the schedule of the *Table Tennis Night* in The 2nd Universal Cup Finals. "*Aha, it's time to host a Universal Cup Table Tennis Championship!*"

Little Cyan Fish wants to set a rule that determines the winner after multiple games. Each table tennis game is played with two players, called A and B. A winner is always decided in each game — therefore, there is no draw in table tennis. The winner of each game will score one point, and the other player will not get any points. The championship between two players will end immediately when the following conditions are both met:

- At least one player has scored m points;
- One player is ahead by at least two points.

Little Cyan Fish has a magic that could accurately predict the results of all the matches. He provided a string of length n, $s_1 s_2 \ldots s_n$, where each character is either 'A' or 'B'. Here, 'A' represents player A and 'B' represents player B, indicating that the i-th point in the match will be scored by the player represented by $s_{((i-1) \bmod n)+1}$.

You need to calculate how many points will be scored before the match ends or indicate that the match will never end.

Input

There are multiple test cases. The first line of the input contains a single integer T ($T \geq 1$), indicating the number of the test cases. For each test case:

The first line of the input contains two integers n and m ($1 \leq n \leq 2 \times 10^5$, $1 \leq m \leq 10^{18}$).

The second line of the input contains a string $s_1 s_2 \ldots s_n$ containing only 'A's and 'B's.

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

Output

For each test case:

If the championship will never end, output a single line with a single word "`No`".

Otherwise, the first line of the output should contain the word "`Yes`".

The next line of the output should contain a single integer, indicating how many points will be scored before the championship ends.

Example

standard input	standard output
3 1 11 A 2 11 AB 3 11 ABB	Yes 11 No Yes 17

Problem J. Divide the String

Little Cyan Fish has two binary strings s and t. A *binary string* is a string where each character is either 0 or 1.

For a binary string s, let $c_0(s)$ be the number of zeroes in s and $c_1(s)$ be the number of ones in s. Of course, we will have $c_0(s) + c_1(s) = |s|$, as s contains only zeroes and ones.

Little Cyan Fish wants to divide s into $k = |t|$ substrings $\sigma_1, \sigma_2, \ldots, \sigma_k$, in other words, $s = \sigma_1 + \sigma_2 + \ldots + \sigma_k$. For each i, σ_i satisfies the following conditions:

- if $t_i = 0$, then $c_0(\sigma_i) > c_1(\sigma_i)$.
- if $t_i = 1$, then $c_0(\sigma_i) < c_1(\sigma_i)$.

Help Little Cyan Fish find a way to divide the string, or report that it is impossible to meet his requirements!

Input

There are multiple test cases. The first line of the input contains a single integer T ($T \geq 1$), indicating the number of the test cases. For each test case:

The first line of the input contains a single string s ($1 \leq |s| \leq 10^6$).

The next line of the input contains a single string t ($1 \leq |t| \leq |s|$).

It is guaranteed that the sum of $|s|$ over all test cases does not exceed 10^6.

Output

For each test case:

If it is impossible to meet Little Cyan Fish's requirements, output a single line with a single word "`No`".

Otherwise, the first line of the output should contain the word "`Yes`".

To reduce the size of the output, you should output the solution in the following format.

The next line of the output contains a binary string A with length $|A| = |s|$, describing the solution. If you divide the string S between the i-th digit and the $(i+1)$-th digit, then the i-th digit of A shall be marked as 1. Otherwise, the i-th digit of A should be marked 0. Specifically, A_n should always be equal to 1, as the string always ends at the last character.

For example, if you would like to divide the string 00101010 into three segments 00, 10101, 0, then you should output 01000011.

Example

standard input	standard output
3 1001 0 111 1 011011010011101 110111	No Yes 001 Yes 000011000110011

Problem K. Master of Modular Arithmetic

"*But, come on...*"
"*I'm sick of solving yet another crazy problem on counting some, ugh, non-sensical things*"
"*And then take the answer, modulo a large prime number.*"
"*Can we get something different?*"

Little Cyan Fish loves modular arithmetic and is the one who mastered the skills of handling modular arithmetic.

To test if you have truly understood the beauty of modular arithmetic, Little Cyan Fish gives you a sequence $a_1, a_2, \ldots a_n$ of n positive integers.

Then, you are allowed to perform the following operations at most $2n + 10$ times:

1. Choose an integer $1 \leq x \leq 10^9$.
2. Choose two indices i and j, such that $1 \leq i, j \leq n$ and $i \neq j$.
3. Update the sequence:
   - $a_i \leftarrow a_i \bmod x$
   - $a_j \leftarrow a_j \cdot x$

Little Cyan Fish wants you to apply some operations on the sequence a, such that in the end, a becomes another sequence $b_1, b_2, \ldots, b_n$. Can you figure out if it is possible?

Input

There are multiple test cases. The first line of the input contains a single integer T ($T \geq 1$), indicating the number of test cases. For each test case:

The first line of the input contains a single integer n ($1 \leq n \leq 5 \times 10^5$).

The next line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^8$), indicating the initial sequence.

The next line of the input contains n integers $b_1, b_2, \ldots, b_n$ ($1 \leq b_i \leq 10^8$), indicating the final sequence.

It is guaranteed that the sum of n over all test cases does not exceed 5×10^5.

Output

For each test case:

If it is impossible to convert the sequence a to the sequence b within $2n + 10$ operations, output a single line "`No`".

Otherwise, the first line of the output should contain the word "`Yes`".

The next line of the input contains a single integer m ($0 \leq m \leq 2n + 10$), indicating the number of operations you would like to apply.

The next m lines describe all your operations. Each line of these lines contains three integers i, j, and x, indicating an operation.

Example

standard input	standard output
3 2 2 2 1 2 4 4 4 4 4 1 1 1 1 5 1 4 3 2 5 2 4 5 4 1	Yes 5 1 2 10 2 1 19 1 2 7 2 1 3 1 2 2 No Yes 3 3 4 2 1 3 5 5 1 2

Problem L. Not Another Constructive Problem

Consider a tree $T = (V, E)$ with n vertices, labeled from 1 to n. The i-th vertex initially has a value p_i. All the values of p_i are distinct integers from 1 to n. In other words, p_i is a permutation of length n.

You are asked to perform the following operations $n - 1$ times:

- Select an edge $e = (x, y) \in E$.
- Swap the value of p_x and p_y.
- Remove the edge e from E.

In the end, T will become a graph without any edges. Such a tree T is called *yummy* if there exists a way to apply these operations, such that p_i becomes q_i for all $1 \le i \le n$.

You may think Little Cyan Fish would like to find a plan to prove T is *yummy*. But we have too many constructive problems in this contest! Therefore, he gives you an undirected graph G with n vertices and many, many edges. In fact, there are $c_{i,j}$ distinct edges between the vertices i and j.

Little Cyan Fish would like you to count how many *yummy* spanning trees this graph contains. Two spanning trees are considered different if they contain different sets of edges. As the answer can be huge, output the answer modulo $10^9 + 7$.

Input

The first line of the input contains a single integer n ($1 \le n \le 500$), indicating the number of nodes of the graph.

The next line of the input contains a permutation $p_1, p_2, \ldots, p_n$, indicating the value on each node initially.

The next line of the input contains a permutation $q_1, q_2, \ldots, q_n$, indicating the value on each node eventually.

The next n lines describe the matrix c, each line with n integers. The j-th integer of the i-th line of these lines describes the value of $c_{i,j}$ ($0 \le c_{i,j} \le 10^9 + 6$), indicating the number of edges between nodes i and j.

It is guaranteed that $c_{i,i} = 0$, and $c_{i,j} = c_{j,i}$ for all $1 \le i, j \le n$.

Output

Output a single line that contains a single integer, indicating the number of *yummy* spanning trees of this graph modulo $10^9 + 7$.

Examples

standard input	standard output
4 1 2 3 4 3 4 2 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0	12
6 1 2 3 4 5 6 5 3 4 1 6 2 0 3 3 2 1 2 3 0 2 2 2 1 3 2 0 3 2 3 2 2 3 0 1 2 1 2 2 1 0 1 2 1 3 2 1 0	5050

Problem M. Dividing Chains

For a sequence $z_1, z_2, \cdots, z_n$, consider a set $S = \{1, n+1\}$. Little Cyan Fish may perform the following operations **exactly** $n-1$ times:

- Choose an integer x s.t. $x \notin S$ and $1 < x \leq n$.
- Let l be the **predecessor** of x in S, i.e., the largest integer in S such that $l < x$.
- Let r be the **successor** of x in S, i.e., the smallest integer in S such that $x < r$.
- Do nothing **or** apply the following update to the sequence z
  - Reverse the subsegment $z_l, z_{l+1}, \ldots, z_{r-1}$, i.e., z_i becomes $z_{r-1-(i-l)}$ for all $l \leq i \leq r-1$.
- Update $S \leftarrow S \cup \{x\}$.

Obviously, in the end, S will become $\{1, 2, \ldots, n, n+1\}$. For a given **monotonically non-decreasing** sequence $a_1, a_2, \ldots, a_n$, you need to count the number of sequences $z_1, z_2, \ldots, z_n$, modulo 998 244 353, that satisfy the following condition:

- There exists a way to apply the operations above to convert the sequence $z_1, z_2, \cdots, z_n$ to $a_1, a_2, \cdots, a_n$.

This problem sounds non-sensical to Little Cyan Fish. Therefore, he asked you to solve this problem.

Input

The first line of the input contains a single integer n ($1 \leq n \leq 500$).

The next line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_1 \leq a_2 \leq \ldots \leq a_n \leq n$).

Output

Output a single line containing a single integer, indicating the answer modulo 998 244 353.

Examples

standard input	standard output
3 2 3 3	3
5 1 1 3 3 5	29
9 1 2 3 5 5 6 7 8 9	26276