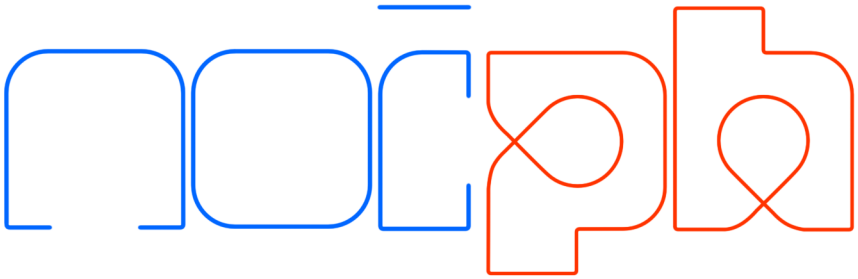


2024

National Olympiad in Informatics

Finals Round 1

Important! Read the following:

Hidden Test Cases. Your solution will be checked by running it against one or more (usually several) hidden test cases. You will not have access to these cases, but a correct solution is expected to handle them correctly.

Strict Output Format. The output checker is **strict**. Follow these guidelines strictly:

- It is **space sensitive**. Do not output extra leading or trailing spaces. Do not output extra blank lines unless explicitly stated.
- It is **case sensitive**. So, for example, if the problem asks for the output in lowercase, follow it.
- Do not print any tabs. (No tabs will be required in the output.)
- Do not output anything else aside from what's asked for in the Output section. So, do not print things like "`Please enter t`".

Not following the output format strictly and exactly will likely result in the verdict "*Output isn't correct*".

Use Standard I/O. Do not read from, or write to, a file. You must read from the standard input and write to the standard output.

Submit Code Only. Only include **one** file when submitting: the source code (.cpp, .py, etc.) and nothing else.

No Java Package. For Java submissions, do not include a `package` line.

No Weird Filenames. Only use letters, digits and underscores in your filename. Do not use spaces or other special symbols.

Use Fast I/O. Many problems have large input file sizes, so use fast I/O. For example:

- In C/C++, use `scanf` and `printf`.
- In Python, use `sys.stdin.readline()`

Flush On Interactive Problems. On interactive problems, make sure to **flush** your output stream after printing.

- In C++, use `fflush(stdout);` or `cout << endl;`
- In Python, use `sys.stdout.flush()` or `print` with `flush=True`
- For more details, including for other languages, ask a question/clarification through CMS.

Good luck and enjoy the contest! 😃

Contents

Notes

- Many problems have large input file sizes, so use fast I/O. For example:
  - In C/C++, use `scanf` and `printf`.
  - In Python, use `sys.stdin.readline()`
- On interactive problems, make sure to **flush** your output stream after printing.
  - In C++, use `fflush(stdout);` or `cout << endl;`
  - In Python, use `sys.stdout.flush()` or `print` with `flush=True`
  - For more details, including for other languages, ask a question/clarification through CMS.

Good luck and enjoy the problems!

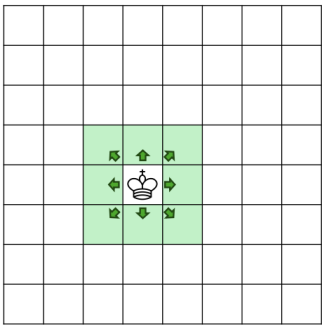
Problem A
Happy Na Birthday Mo Pa

Your thoughtful friends, Carlos and Sarina, want to sing you a Happy Birthday song. They have prepared and composed their own rendition with such poetic lyrics such as: "Ilang taon ka na? Happy birthday, ilang taon ka na?" and "Iiyak na 'yan'." Now they're ready to perform it in front of you. In a room with special acoustics even! This way, you can fully experience their unmatched singing talent.

The room can be thought of as a grid with r rows and c columns, and thus a total of rc squares. Label the rows 1 to r from top to bottom, and the columns 1 to c from left to right. Let (i, j) be the square in row i and column j, so for instance, the top-left-most square is $(1, 1)$, and the bottom-right-most square is (r, c).

Due to the weird acoustics of the room, the strength of the sound (hence the volume) in each square is determined by a very particular formula.

We must first define the **chess king distance**. Given two squares $A = (a_i, a_j)$ and $B = (b_i, b_j)$, the chess king distance from A to B, denoted by $\text{cd}(A, B)$, is the fewest number of moves that a chess king must make in order to move from square A to square B. For those who have never played chess, *where have you been all your life?* And second of all, one move of a chess king allows it to move to any of the eight squares that share an edge or corner with the king's current square.



If Carlos were singing in square $C = (c_i, c_j)$ and Sarina were singing in square $S = (s_i, s_j)$, then the volume $v(X)$ of their voices at a given square X would be computed using the following formula:

$$v(X) = \frac{1}{\text{cd}(C, X) + \text{cd}(S, X)}.$$

Your wish for this birthday is to know which squares in the grid your favorite singing couple could be in. However, you are blinded by Carlos's physique and cannot see! So you cannot easily solve this problem and make it... sheesh!

You do know, however, the value of $1/v(X)$ for each square X in the grid. Also, Carlos and Sarina are always in separate squares. Given this information, output two squares where Carlos and Sarina could possibly be.

We guarantee that for all test cases for this problem, at least one answer exists. After all, who would want to miss the sound of Carlos and Sarina's angelic voices and witty rap lyrics?

Input Format

The first line of input contains two space-separated integers r and c.

Then, r lines follow, each containing c space-separated integers, encoding the value of $1/v(X)$ for each square in the grid.

Output Format

Output two lines, each containing two space-separated integers. The first line should contain the values c_i and c_j, while the second line should contain the values s_i and s_j—these correspond to the squares C and S where Carlos and Sarina respectively are located.

If there are multiple possible answers, any will be accepted.

Constraints

For all subtasks
$1 \leq r, c$ $2 \leq rc \leq 200\,000$ $1 \leq 1/v(X) \leq 10^9$ for each square X. A solution always exists.

Subtask	Points	Constraints
1	**31**	$rc \leq 200$
2	**29**	$rc \leq 2000$
3	**12**	$r = 1$
4	**28**	No further constraints.

Sample I/O

Input	Output
3 4 4 2 2 3 4 3 2 2 5 4 3 3	1 2 2 4

Problem B
The Last Light of NOI

Dr Kakaiba and his wife, Dr Kakaibabe, are preparing for the looming fight versus Thanos. In the worst case scenario, Thanos acquires the power of the Infinity Stones and harnesses the power of the Quantum Realm, allowing him to snap more than once.

You are the final n champions of NOI.PH, arranged in a row and labeled from 1 to n. Each of you has been assigned a skill rating, represented by some positive integers $a_1, a_2, \ldots, a_n$.

Whenever Thanos snaps his fingers, the following happens:

- Let ℓ be the number of champions remaining.
- Consider all **contiguous** subarrays of length $\lfloor \ell/2 \rfloor$ ($\ell/2$ rounded down)
- One of these subarrays is independently selected uniformly at random.
- All champions in that subarray *vanish*, trapped inside the Soul Stone.
    - All remaining champions "move to close the gap".

Thanos repeatedly snaps his fingers until only one champion remains. This champion is the Last Light of NOI, and they alone must defeat Thanos in a one-on-one duel in order to secure the fate of the cosmos.

Dr Kakaiba and Dr Kakaibabe replay this scenario countless times (they've already examined more than 14,000,605 possibilities, and counting). After each simulation, they take note of the skill rating of the Last Light of NOI. The two of them keep track of the running average of the skill ratings of the Last Lights of NOI so far after each experiment. As they perform more and more experiments, this running average converges to a value, known as the expected value.[1]

What is the expected value of the skill rating of the Last Light of NOI?

Input Format

The first line of input contains a single integer n, the number of NOI's champions.

The second line of input contains n space-separated integers, the skill-ratings of the champions $a_1, a_2, a_3, \ldots, a_n$ in the order that they appear in the row.

[1] More precisely, it can be shown that there's a value e (the expected value) such that the running average converges to e with probability 1.

Output Format

Output a single decimal value, the expected value of the Last Light of NOI.

Your answer will be accepted if it has an absolute or relative error of at most 10^{-9} from the judge's answer. In symbols, let ans_{you} be your answer, and let ans_{judge} be the judge's answer. Your answer will be accepted if

$$\frac{|ans_{\text{you}} - ans_{\text{judge}}|}{\max(1, |ans_{\text{you}}|, |ans_{\text{judge}}|)} \leq 10^{-9}$$

In short, **make sure you print your answer to sufficiently many decimal places**. For example,

- In C++, `#include` the `<iomanip>` library, and use

  ```
  cout << fixed << setprecision(12) << ans << endl;
  ```

  to output `ans` to exactly 12 decimal places.
- In Python, use `print(f'{ans:.12f}')` to output `ans` to exactly 12 decimal places.

Constraints

For all subtasks
$1 \leq a_i \leq 10^9$

Subtask	Points	Constraints
1	**37**	$2 \leq n \leq 75$
2	**26**	$2 \leq n \leq 500$
3	**22**	$2 \leq n \leq 5000$
4	**15**	$2 \leq n \leq 200\,000$

Sample I/O

Input	Output
5 1 1 2 3 5	2.5000000000000000000

Explanation

Here is one possible result of a simulation on the sample input:

- $\lfloor 5/2 \rfloor = 2$, and so one subarray from among $[1, 1]$ or $[1, 2]$ or $[2, 3]$ or $[3, 5]$ is uniformly randomly selected to vanish. Suppose that $[1, 2]$ is chosen, so the champions now look like $[1, 3, 5]$.
- $\lfloor 3/2 \rfloor = 1$, and so one subarray from among $[1]$ or $[3]$ or $[5]$ is uniformly randomly selected to vanish. Suppose that $[5]$ is chosen, so the champions now look like $[1, 3]$.
- $\lfloor 2/2 \rfloor = 1$, and so one subarray from among $[1]$ or $[3]$ is uniformly randomly selected to vanish. Suppose that $[1]$ is chosen, so the remaining champion (the Last Light of NOI) has rating 3.

If the Doctors repeat many and many more simulations, the average value of the rating of the Last Light of NOI.PH can be shown to converge to exactly $5/2$.[2]

[2]More precisely, it converges to $5/2$ with probability 1.

Problem C
Volta

As avid TV viewers know, robots work best in groups of five called Voltes Groups. For example, the Super Rangers have five robot Swords, that can merge into a Megasword. In the past, these Voltes Groups have fought alien spaceships with merging and missiles. Froggy, the consultant to Japan's robotics department, wants to enlist all the robots in the upcoming fight against Thanos. But Froggy realized that he doesn't have enough power. He turns to his cellphone, scrolling through his contacts, trying to think of a solution.

Meanwhile, Perla Magtoto always knew she had the potential to make a difference. After being struck by lightning thrice, she gained the superpower of making electricity with her hands. With these powers, she became famous as the superhero Volta.

Meanwhile, Representative Darth Newton, after being elected to the Senate for his second law of motion campaign, decided to pass another law. You might have learned Ohm's law in physics class, which states that voltage, or the potential difference, equals current times resistance. Well, Darth Newton decided to repeal that law, and make a new law, saying:

The sum of the potentials equals current plus resistance.

Meanwhile, Froggy realizes that Volta could help, so he comes to Perla's residence, which doubles as a dress shop called *Perla ng Silangan*. Froggy asks Perla if she could use her powers, as Volta, to power the robots. "Of course, and it'll be easy," said Perla, "I've even powered robots before!" Indeed, Perla once used her electricity to power the robots named Oh-Vlading and Oh-Blah-Blah.

Suddenly, Darth Newton's law was published in the Official Galaxy Gazette! Volta felt the change in the ultraelectromagnetic field, and realized that it might not be so easy to power all of Froggy's robots. But Froggy said, "It's okay, let me explain the problem, and then we can figure out whether we can power all the robots."

There are n *robots*, numbered 1 to n. Froggy needs to power the robots with an integer **current** c. The robots are connected by w **wires**, each with some integer **resistance** r. If robot a is wired to robot b, that means electricity can flow from a to b. Wires only go in one direction: if a is wired to b, then electricity cannot travel from b to a.

Electricity can travel from robot a_1 to a_k if there exist robots $a_2, \ldots, a_{k-1}$ such that a_1 is wired to a_2, a_2 is wired to a_3, and so on, up to a_{k-1} being wired to a_k. The total resistance along such a route is the sum of the resistances of each wire along that route. The robots are connected such that there are no short-circuits; that is, electricity cannot travel from a robot to itself.

Before Volta turns on the current (of strength c), she needs to assign integer *potentials* to each robot. The ith robot has two potentials: the out-potential p_i, and the in-potential q_i. The potentials need to be assigned so that, if electricity can travel between two robots, it must follow *Darth Newton's law*. That is, if electricity can travel from robot a to b via some route, the sum of the out-potential of a and the in-potential of b, must equal the current c, plus the total resistance r it takes to travel from a to b along this route. In symbols: to follow Darth Newton's law, we must have

$$p_a + q_b = c + r.$$

If there are multiple ways to travel from robot a to b, they *all* need to follow Darth Newton's law.

Can you help Froggy and Volta assign *nonnegative-integer* potentials such that Darth Newton's law is always followed?

Input Format

The first line of input contains a single integer t, the number of test cases. Then, the descriptions of the t test cases follow.

The first line of each test case contains three space-separated integers n, w, and c, the number of robots, the number of wires, and the current.

The next w lines each contain three space-separated integers a_i, b_i, and r_i, meaning that robot a_i is wired to robot b_i by a wire with resistance r_i.

Output Format

For each test case, if it is not possible to assign potentials to follow Darth Newton's law, output a single line containing "`NO`" (without quotes).

Otherwise, output a single line containing "`YES`". Then output n lines. The ith line should contain two space-separated integers p_i and q_i, representing the out-potential and in-potential of the ith robot, such that $0 \leq p_i, q_i \leq 2 \cdot 10^{18}$.

If there are multiple possible answers, any one would be accepted. It can be shown that if a solution exists, then one exists with each p_i and q_i respecting the above bounds.

Constraints

Let N be the sum of the ns in a single test file. Let W be the sum of the ws in a single test file.

For all subtasks
$1 \le t \le 1000$ $1 \le n, N \le 500\,000$ $0 \le w, W \le 500\,000$ $0 \le c \le 10^{18}$ $1 \le a_i, b_i \le n$ $1 \le r_i \le 10^9$ It is guaranteed that electricity cannot travel from a robot to itself.

Subtask	Points	Constraints
1	**23**	$a_i = i$, $b_i = i + 1$ for all i
2	**21**	$n \le 20$, $t \le 100$
3	**16**	a_i is odd and b_i is even for all i
4	**31**	$N \le 3000$
5	**9**	No additional constraints.

Sample I/O

Input	Output
1 6 6 10 1 2 2 1 3 3 2 4 2 3 4 1 5 6 8 5 6 8	YES 12 500 10 0 9 1 6000 2 18 293814820823 0 0

Explanation

In this case, it is possible to assign nonnegative-integer potentials. For example, electricity can flow from robot 1, to robot 3, then to robot 4. Here, $p_1 = 12$, $q_4 = 2$, $c = 10$, and the total resistance along this route is 4, so this follows Darth Newton's law.

Problem D
Rock Collection

Thanos wants to collect k stones. We assume that it is for a rock collection of some sort, but it might be for a totally different purpose that we do not predict at all. Who's to say?

Anyway, the stones he wishes to collect are conveniently in a location that can be represented as a grid with r rows and c columns. We number the rows 1 to r from top to bottom, and number the columns 1 to c from left to right. Thanos insists that the stones must be collected in a certain order, perhaps for cataloguing purposes. For example, the Rak of Aegis must be collected before The Kidney Stone. And The Sharon Stone must be collected before Dwayne The Rock. It is a bit complicated.

For the purposes of this problem, we won't bore you with the names, and simply say that the xth stone he must collect is located in square S_x.

Thanos is very systematic in planning his journey. He chooses any starting square, teleports there, and then snaps his fingers to practice for something completely unrelated. He must collect the stones in order—that is, if $x < y$, then he must collect the stone at square S_x before the stone at square S_y. Note that Thanos is allowed to go through a square without collecting the stone there.

All squares start off as being **safe**. When Thanos collects a stone from a square, that square becomes unsafe as soon as he leaves it. In one second, Thanos can move from his current square to any safe square that is directly one square north, south, east, or west from his current square (if such a safe square exists). Picking up a stone takes negligible time.

Finally, Thanos is efficient. He knows the minimum amount of time needed in order to finish the job.

As soon as he snaps his fingers, Stephen, a doctor who is a little bit eccentric (no relation to Dr Kakaiba or Dr Kakaibabe), felt a disturbance. Consider the set of all possible paths that Thanos can take in order to finish the job in the minimum amount of time. For some reason, Stephen managed to see all these possibilities, but concluded that *only one of them* leads to Thanos failing in his overall plan.

How many different minimum-time paths are there? Compute this number modulo 998244353. Two paths are considered different if Thanos starts in a different starting location, or ever makes a different move at any point in time during his travels.

Input Format

The first line of input contains three space-separated integers r, c, and k.

The next k lines of input describe the squares containing the stones in the order

that Thanos wants to collect them. The xth of them contains two space-separated integers i_x and j_x denoting the row and column number of square S_x, respectively.

Output Format

Output one line containing an integer denoting the number of minimum-time paths, modulo 998244353.

Constraints

For all subtasks
$3 \leq r, c \leq 10^6$ $2 \leq k \leq 32$ $1 \leq i_x \leq r$ for each x $1 \leq j_x \leq c$ for each x No two stones are in the same square.

Subtask	Points	Constraints
1	**17**	$k \leq 3$
2	**29**	$r, c \leq 30$
3	**30**	$k \leq 12$
4	**18**	No two stones are in the same row or column.
5	**6**	No additional constraints.

Sample I/O

Input 1	Output 1
5 7 3 3 4 3 2 3 6	8

Input 2	Output 2
7 4 5 2 3 3 2 5 2 4 2 3 3	2

Explanation

In the first sample input, the minimum time is 8 seconds, and there are 8 paths that take 8 seconds, so the answer is $8 \bmod 998244353 = 8$.

In the second sample input, the minimum time is 7 seconds, and there are 2 paths that take 7 seconds, so the answer is $2 \bmod 998244353 = 2$. One of these paths is

$$\underbrace{(2,3)}_{\text{pick stone}} \to (3,3) \to \underbrace{(3,2)}_{\text{pick stone}} \to (4,2) \to \underbrace{(5,2)}_{\text{pick stone}} \to \underbrace{(4,2)}_{\text{pick stone}} \to (4,3) \to \underbrace{(3,3)}_{\text{pick stone}} .$$

The highlighted ones represent the times when Thanos picks up the respective stones. Note that Thanos doesn't pick up the stones at $(3,3)$ and $(4,2)$ the first time he visits them.