



西安交通大学
XI'AN JIAOTONG UNIVERSITY

2025 Xi'an Jiaotong University Programming Contest

Problem Set

- A Card Master
- B Nobody Can Log In
- C Quantum Mechanics
- D Symmetrical Isolation: The Battle of Black and White
- E Pythagorean Cup
- F Beat Verdict: Precision Strike
- G Primal Core Optimization: Attribute Balance
- H 9-Nine
- I Ori
- J Bastion
- K Seele Vollerei
- L Gros-Phi
- M History of Terra
- N Kessoku Band
- O Discrete Logarithm

May 18, 2025

Do not open the problem set before the contest begins!



Problem A. Card Master

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

CuteWhite has recently become fascinated with a multiplayer card game.

The rules of the game are as follows: each player draws three cards, and the sum of the face values of the three cards is calculated as the score for that round. The player with the highest score wins. Specifically, if all three cards have the same face value, it is called a “bomb,” and the score is increased by an additional 100 points.

However, CuteWhite is not very good at calculations. Therefore, he hopes you can write a program that can quickly calculate the score of the three cards he draws.

Input

The input consists of a single line with three integers a_1 , a_2 , and a_3 ($1 \leq a_1, a_2, a_3 \leq 13$), separated by a space, representing the face values of the three cards drawn by CuteWhite in this round of the game.

Output

Output a single line containing just one integer, which represents CuteWhite's score.

Examples

standard input	standard output
6 6 6	118
1 1 4	6

Note

For the first sample, since all three cards have the same face value, the score should be increased by an additional 100 points, resulting in a total score of $6 + 6 + 6 + 100 = 118$.

For the second sample, since the face values of the three cards don't have the same face value, the total score is $1 + 1 + 4 = 6$.



Problem B. Nobody Can Log In

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Chapter One: Cultivation

Chapter Two: That heavy rain ruined my OI dream

Chapter Three: The first prize in the programming competition is still a first prize

Chapter Four: What does it mean when the password is incorrect?

Chapter Five: That 1i1l ruined my dream of a first prize

Chapter Six: Cultivation, preparing for the 2026 programming competition

— Shirost

Senior J is a renowned cryptography master in the Kingdom of Trees.

The annual programming competition in the Kingdom of Trees is about to begin, and Prime Minister S has entrusted Senior J with the important task of maintaining the competition system. Senior J generated accounts and passwords for n participating teams using `uuid4().hex` to create hexadecimal strings as passwords (containing only the digits 0-9 and a-f), but he felt that the strength of the passwords generated this way was somewhat insufficient. Therefore, he switched to randomly generating byte sequences and using `base64` encoding to create more complex, high-strength passwords (containing lowercase letters, uppercase letters, numbers, underscores `_`, and hyphens `-`).

However, during the warm-up match, many contestants frequently entered incorrect passwords due to difficulty distinguishing the characters `1`, `i`, `I`, and `l`, making it impossible for them to log into the system smoothly. Prime Minister S was very angry about this and demanded that Senior J immediately rectify the passwords for the official competition by removing all easily confused characters. Senior J now wants to know how to modify the passwords for the official competition.

Specifically, given the original password for each team during the official competition login, you need to remove all occurrences of `1` (the digit 1), `i` (the lowercase letter i), `I` (the uppercase letter i), and `l` (the lowercase letter L) from the original password and output the modified password.



Input

The first line of input contains a positive integer n ($1 \leq n \leq 5 \times 10^4$), indicating that there are n teams participating in the competition.

The next n lines each contain a non-empty string s_i (the length of the string $|s_i|$ satisfies $1 \leq |s_i| \leq 5 \times 10^4$; input characters consist only of lowercase letters, uppercase letters, numbers, underscores `_`, and hyphens `-`), representing the original password for each participating team's login to the system.

It is guaranteed that the sum of the lengths of the original passwords for the n teams does not exceed 5×10^4 .

Output

Output a total of n lines, where the i -th line represents the password of the i -th team after removing `1iI1`. If the password is empty, output **empty string**.

Examples

standard input	standard output
5 mCj_m3sYshA 1liil1ili Itboe1Vcig4 lsj5lFx1Nrs -j1ioFlIiI4	mCj_m3sYshA tboeVcg4 sj5FxNrs -joF4
6 Ave_mujica Doloris Mortis Timoris Amoris Oblivionis	Ave_mujca Doors Morts Tmors Amors Obvons



Problem C. Quantum Mechanics

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

MCPlayer542 has recently been studying “Quantum Computation and Quantum Information.” He is very interested in the representation of quantum states.

Unlike classical bits, a qubit can not only be in the state $|0\rangle$ or the state $|1\rangle$, but can also be in a superposition state between $|0\rangle$ and $|1\rangle$.

The state of a quantum system composed of n qubits is usually represented by a 2^n -dimensional complex vector $|\psi\rangle$, where the product of the i -th complex number and its conjugate represents the probability of the state of these n quantum bits being equal to the binary representation of i , after a quantum measurement on the computational basis.

For example, when you perform a quantum measurement on a two-qubit system represented by $|\phi\rangle = [\frac{1}{2}, \frac{1}{2} - \frac{i}{2}, 0, \frac{i}{2}]^T$, it will be in the state $|00\rangle$ with a probability of $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$, in the state $|01\rangle$ with a probability of $(\frac{1}{2} - \frac{i}{2}) \times (\frac{1}{2} + \frac{i}{2}) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$, in the state $|10\rangle$ with a probability of $0 \times 0 = 0$, and in the state $|11\rangle$ with a probability of $\frac{i}{2} \times (-\frac{i}{2}) = \frac{1}{4}$.

Now he has a quantum state $|\psi\rangle$ with n quantum bits. Please help him calculate the probabilities of each qubit being in the states $|0\rangle$ and $|1\rangle$ after a measurement on the computational basis, respectively.

Input

The first line contains a positive integer n ($1 \leq n \leq 8$), representing the number of quantum bits in the system.

The next 2^n lines contain the i -th line with two real numbers a_i and b_i , separated by a space, representing the real part and the imaginary part of the i -th component of the quantum system vector, respectively.

The data guarantees that the sum of the probabilities of the quantum system being in all states does not exceed an absolute or relative error of 10^{-6} from 1.

Output

Output n lines, each containing two real numbers p_0 and p_1 , separated by a space, representing the probabilities of each quantum bit being in the state $|0\rangle$ and the state $|1\rangle$ after the measurement, respectively.

Your output will be considered correct if and only if the absolute or relative error from the correct answer does not exceed 10^{-6} .



Example

standard input	standard output
2	0.250000000 0.750000000
0.5 0	0.750000000 0.250000000
0.5 -0.5	
0 0	
0 0.5	



Problem D. Symmetrical Isolation: The Battle of Black and White

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 512 megabytes

You are playing a game called “Your Craft”. In this game, you can freely create and destroy different types of blocks in a three-dimensional space filled with blocks.

Now, you have a grid composed of $N \times M$ cells beneath your feet. At the beginning, you covered these cells with white carpets, and then you went to sleep.

Unfortunately, while you were sleeping, the problem setter secretly replaced some of the white carpets with black carpets.

When you woke up and opened the game again, you found that your masterpiece had been tainted, and you were very angry! So you decided to make the best of the situation by keeping the existing black carpets unchanged and dyeing some of the white carpets black to satisfy the following two conditions:

1. No two cells with black carpets can share a common edge, which means:
 - If the cell in the i -th row and j -th column contains a black carpet, then the cells in the $(i - 1)$ -th row and j -th column, $(i + 1)$ -th row and j -th column, i -th row and $(j - 1)$ -th column, and i -th row and $(j + 1)$ -th column cannot contain black carpets if they have carpets.
2. The final pattern must have at least one horizontal or vertical line of symmetry, which means at least one of the following two propositions must be satisfied:
 - For any $i \in [1, N]$ and $j \in [1, M]$, the color of the carpet in the i -th row and j -th column must be the same as the color of the carpet in the $(N - i + 1)$ -th row and j -th column;
 - For any $i \in [1, N]$ and $j \in [1, M]$, the color of the carpet in the i -th row and j -th column must be the same as the color of the carpet in the i -th row and $(M - j + 1)$ -th column.

You need to determine whether you can achieve these two conditions.

Note: You cannot destroy the existing black carpets.

Input

The first line contains two integers N and M ($1 \leq N, M \leq 100$), separated by a space, representing the size of the grid.

The next N lines each contain a string S of length M . In the string S , a character `W` indicates that this position has a white carpet, while a character `B` indicates that this position has a black carpet. No other characters will appear in the string.



Output

If you can satisfy the conditions given in the problem, output the string **Yes**; otherwise, output the string **No**.

The answer is case insensitive. For example, the strings **yEs**, **Yes**, **yes**, and **YES** will all be recognized as you can satisfy the conditions.

Examples

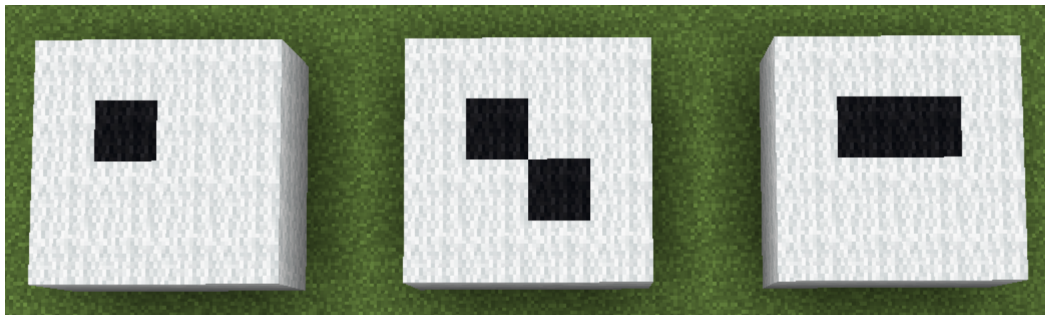
standard input	standard output
4 4 WWWW WBWW WWWW WWWW	No
4 4 BWVB WWWW WWWW WWVB	Yes
4 4 BWWV WBWW WWBV WWWW	No
5 5 BWBWB WBWBW BWBWB WBWBW BWBWB	Yes
2 2 BB WW	No
2 2 WW WW	Yes



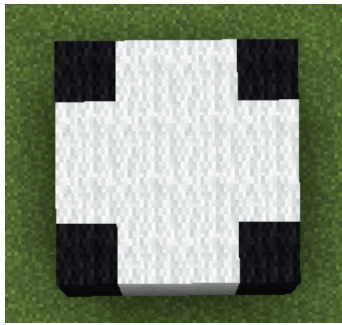
standard input	standard output
8 6 W B W W W B W W W W W B W W W W W W B W B W W W W W B W W W W B W W W B W W W W	No

Note

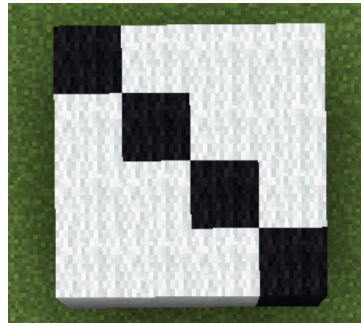
For the first sample:



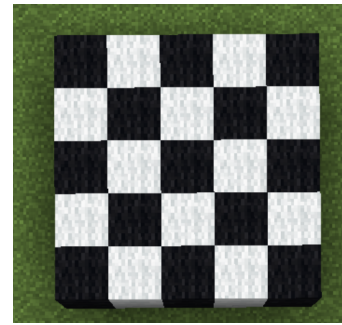
The leftmost image shows the situation you saw when you woke up. You cannot achieve the conditions by changing white carpets to black carpets. The pictures in the middle and on the rightmost show two possible ways to replace the carpet, but they cannot satisfy the two requirements in the question (the middle image does not have a horizontal or vertical line of symmetry; in the rightmost image, the black carpets share a common edge). It can be proven that any way of changing the carpets does not satisfy the requirements, so output No.



(a) The image of 2nd sample



(b) The image of 3rd sample



(c) The image of 4th sample

For the second sample: The image shows a valid way to change the carpets. You can change the white carpet in the lower left corner to black, which satisfies both requirements in the problem. Therefore, you should output **Yes**.

For the third sample: The image shows one possible way to change the carpets, but it is not a valid one. Remember: only configurations with a horizontal or vertical line of symmetry are considered valid.

For the fourth sample: As shown in the image, the situation you saw when you woke up is already valid, so you do not need to make any changes.

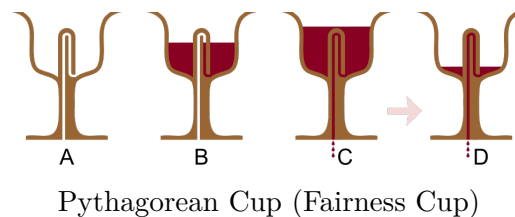
For the fifth sample: When you woke up, there were already two black carpets sharing a common edge, so you cannot change the white carpets to satisfy the requirements.

For the sixth sample: A pattern with all white carpets is also considered valid because it has a horizontal or vertical line of symmetry and does not have two black carpets sharing a common edge.

Problem E. Pythagorean Cup

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Alice and Bob are playing a game with n Pythagorean cups (known as Fairness cups in Chinese). When drinking from a Pythagorean cup, it behaves like a regular cup as long as it is not filled to the top. If it is filled to the top, the liquid will completely flow out through the bottom hole. The structure of the Pythagorean cup is shown in the figure below.



In the initial phase, wine is poured into the n empty Pythagorean cups in sequence. There are three possible amounts of wine that can be poured: no wine, represented by the uppercase letter E; half a cup of wine, represented by the uppercase letter H; and a full cup of wine, represented by the uppercase letter F.

Then, each time Alice and Bob can choose a **non-empty** Pythagorean cup and pour all the wine from it into the cup to its right. If they choose the rightmost cup, they can simply discard it.

At any moment, if a Pythagorean cup is full, it will immediately flow out and become empty. If a cup is half full and half a cup of wine is poured into it, it will overflow and become empty.

Alice and Bob take turns making moves, with Alice going first. If Alice or Bob has no Pythagorean cups to choose from, the person who cannot choose Pythagorean cups loses the game. Assuming both Alice and Bob play optimally, determine who wins the game.

Input

The input consists of a single line containing a string S ($1 \leq |S| \leq 10^6$). The length of the string $|S|$ represents the number of Pythagorean cups n .

S_i is one of the uppercase letters EHF, indicating how much wine to pour into the i -th Pythagorean cup initially. No wine is represented by the uppercase letter E; half a cup of wine is represented by the uppercase letter H; and a full cup of wine is represented by the uppercase letter F.

Output

Output a single line containing a string.

If Alice wins, output Alice; if Bob wins, output Bob.



Examples

standard input	standard output
HEFH	Alice
HEHEFHFEHFE	Bob

Note

For the first sample: Initially, half a cup of wine is poured into the 1-st Pythagorean cup; no wine is poured into the 2-nd cup; a full cup of wine is poured into the 3-rd cup, which immediately flows out; half a cup of wine is poured into the 4-th cup. After the start, the states of the 4 cups are half full, empty, empty, and half full.

At the beginning, Alice can choose the 4-th cup and discard it; then, only the 1-st cup is half full, and Bob can only choose the 1-st cup to pour into the 2-nd cup; similarly, at this point, only the 2-nd cup is half full, and Alice can only choose the 2-nd cup to pour into the 3-rd cup; Bob can only choose the 3-rd cup to pour into the 4-th cup; Alice can only choose the 4-th cup and discard it. At this point, Bob cannot choose any Pythagorean cup anymore and loses the game.



Problem F. Beat Verdict: Precision Strike

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

This is an interactive problem.

You are playing a music game. In expert mode, you need to accurately calibrate the timing parameter for hitting notes, which is a mysterious **integer** x within the range $[1, n]$.

To determine this parameter, you can perform at most 4 calibration tests. In each test, you choose an **integer** test value y ($1 \leq y \leq n$), and the system will inform you whether $y > x$ is true or not. Ultimately, you need to provide an **integer** estimated parameter value x' ($1 \leq x' \leq n$), such that $x' \in [0.5x, 2x]$.

Note: The final estimated value x' does not count towards the aforementioned maximum of 4 tests.

Input

The input contains multiple test cases. The first line of the data contains an integer t ($1 \leq t \leq 500$) indicating the number of test cases. The interaction process for each test case is described below.

Interaction Protocol

In each test case, the first line of input contains a positive integer n ($1 \leq n \leq 5 \times 10^9$), representing the possible range of the parameter.

If you want to perform a calibration test, output `? y` ($1 \leq y \leq n$). Then you will read the response to that test, which will be an integer a ($a \in \{0, 1\}$), where 1 indicates that $y > x$ is true, and 0 indicates that $y > x$ is false.

If you want to submit the calibration parameter x' , output `! x'` ($1 \leq x' \leq n$). You will then immediately end the interaction for this test case and prepare for the interaction of the next test case. This interaction does not count towards the maximum of 4 tests.

Note that at the end of each round of output in your program (i.e., after each output of `? y` or `! x'`), **you need to output a newline and flush the output buffer**; otherwise, you will receive Time Limit.

You can use:

- C's `fflush(stdout);`
- C++'s `cout.flush();`
- Java's `System.out.flush();`
- Python's `stdout.flush();`

to flush the output buffer.

Please note: The interactive library is adaptive, meaning that the positive integer x ($1 \leq x \leq n$) **may change with your inputs** in each test case, but it will always satisfy all previously made inquiries.



If the answer you output at the end is correct, you will receive **Correct**;

If your inquiries do not conform to the problem's range requirements, or if the final answer you output is incorrect, you will receive **Wrong Answer**;

Additionally, other evaluation results will still be returned during the evaluation process based on normal circumstances.

Example

standard input	standard output
2	
1	
	! 1
8	
	? 6
1	
	? 4
1	
	? 2
0	
	? 3
0	
	! 3

Note

In the first test case, it can be uniquely determined that $x = 1$, so we directly submit $x' = 1$.

In the second test case, the hidden parameter $x = 3$, and the interaction process is as follows:

1. Test $y = 6$, the response is that $y > x$ is true;
2. Test $y = 4$, the response is that $y > x$ is true;
3. Test $y = 2$, the response is that $y > x$ is false;
4. Test $y = 3$, the response is that $y > x$ is false;
5. Submit $x' = 3$.

Please note that this example only serves to demonstrate the interaction format. It is not guaranteed that the queries provided are optimal or uniquely determine the answer.



Problem G. Primal Core Optimization: Attribute Balance

Input file: standard input
Output file: standard output
Time limit: 9 seconds
Memory limit: 512 megabytes

You are playing a game called “Rhythm Sin”. In this game, you control partners to explore the “World”. Each partner has three attributes: Stop (S), Flag (F), and Ever (E). When all partners’ attributes are completely consistent, they can successfully fuse.

Your goal is to fuse all partners, meaning you must make every pair of partners have identical attributes. To achieve this, you need to modify their attributes. Each modification consists of the following steps:

1. Select any partner as the target (you can only select one partner).
2. Choose any subset of her attributes.
3. Increment or decrement all selected attributes by 1 (the same operation must be carried out simultaneously, i.e., all selected attributes increment or all selected attributes decrement).

Due to limited magical power, each modification costs 1 primal stone. Your task is to determine the minimum number of primal stones required to achieve your goal.

Input

The first line contains an integer N ($1 \leq N \leq 10^5$), the number of partners.

The next N lines each contain three integers S_i , F_i , and E_i ($1 \leq S_i, F_i, E_i \leq 10^5$), separated by a space, representing the values of the three attributes for the i -th partner.

Output

A single integer representing the minimum number of primal stones required.



Examples

standard input	standard output
3 2 2 2 2 2 2 1 2 3	2
3 2 3 1 3 5 3 3 2 1	4
1 201 502 10	0
4 1 2 3 1 2 3 4 5 6 4 5 6	6

Note

For the first sample, one possible sequence of modifications (costing 2 primal stones) is:

1. Select partner 3 and S attribute, modify her attributes to 2, 2, 3;
2. Select partner 3 and E attribute, modify her attributes to 2, 2, 2.

It can be proved that you cannot achieve the goal using less than 2 primal stones, so the answer is 2.

For the second sample, one possible sequence of modifications (costing 4 primal stones) is:

1. Select partner 1 and S and E attributes, modify her attributes to 3, 3, 2;
2. Select partner 2 and F and E attributes, modify her attributes to 3, 4, 2;
3. Select partner 2 again and F attribute, modify her attributes to 3, 3, 2;
4. Select partner 3 and F and E attributes, modify her attributes to 3, 3, 2.

It can be proved that you cannot achieve the goal using less than 4 primal stones, so the answer is 4.

For the third sample, because there is only one partner, you don't need to perform any modifications, so the answer is 0.



Problem H. 9-Nine

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

*“Kokonotsu Kokonoka Kokonoiro,
Sorairo Sorauta Soranooto,
Haruiro Harukoi Harunokaze,
Yukiuro Yukihana Yukinoato.”*

—9-nine—

Artifact, a kind of magic item from anime called *Rinnetense No Moubiusu Ring*, written by City Shiromitsukawa, comes true unpredictably! The User can get his or her own special ability. All of this story begins from that earthquake.

“Thousands of years ago, the earth and the magic world were connected. The Artifact is just a magic item from the magic world. Because of lots of things, there was set up an organization called *Sefiruto* (Tree of Kabbalah), retrieving all of the Artifacts, using magic to prevent the connection of two worlds. However, attributing to the earthquake, the gate was broken, and the Artifact came to this world again.”
—— the muppet named Sophitia said so.

With the help of Sophitia, our male lead KakeruNimi leads the four female leads in a fierce struggle against the villains, ultimately successfully preventing them from making unforgivable mistakes using Artifacts. After that, KakeruNimi and Sophitia decide to disconnect the two worlds, reusing the magic gate.

The lock of the gate can be seen as two black and white matrices with size 9×9 , and either KakeruNimi or Sophitia can control the matrix in his or her own world. They can:

1. KakeruNimi turns the matrix in his world to rotate by 90 degrees to the left or right;
2. Sophitia turns the matrix in her world to rotate by 90 degrees to the left or right (two matrices can be rotated independently);
3. KakeruNimi and Sophitia choose the same column of the matrices in two worlds and then swap them;

When KakeruNimi's matrix is all white, as soon as Sophitia's matrix is all black, the gate will be locked, and then everyone can return to normal life.

Time is limited, please YOU — our “Nine” — help them complete this task within 999 steps.

That being said, the question maker was warned that the question was too difficult, so he decided to narrow down the data range!

Specifically, reduce the side length of the square matrix to 3 and the number of steps to $9 \times 9 = 81$!



Formally speaking, there are two 3×3 01 matrices A and B , and you can do the following:

1. Operation 1: Select matrix A or B and rotate it 90 degrees to the left or right;
2. Operation 2: Select the integer k ($1 \leq k \leq 3$) and swap the k -th column of the two matrices.

Please convert A into a full 0 matrix within 81 operations, ensuring that B is always a full 1 matrix at this time.

Input

Line 1 ~ 3, each line is a 01 string of length 3, representing the initial state of matrix A .

Line 4 ~ 6, each line is a 01 string of length 3, representing the initial state of matrix B .

The data guarantees that within a finite number of operations, A can be transformed into a full 0 matrix and B into a full 1 matrix.

Output

The first line should be an integer N ($0 \leq N \leq 81$), indicating that you need to perform a total of N operations.

The next N lines should be in the following format:

- If you choose to use operation 1 for an item, please output the matrix name and rotation direction. Specifically:
 - If you want to rotate the A matrix to the left by 90 degrees, output AL;
 - If you want to rotate the A matrix to the right by 90 degrees, output AR;
 - If you want to rotate the B matrix to the left by 90 degrees, output BL;
 - If you want to rotate the B matrix to the right by 90 degrees, output BR.
- If you choose to use operation 2 for an item, please output C and column number. For example, if you want to swap the 2nd column of two matrices, please output C2.

Where, rotating a matrix to the left means rotating it counterclockwise; Rotating a matrix to the right means rotating it clockwise.

Please note: **Do not output extra spaces at the end of the line or carriage returns at the end of the text.**



Example

standard input	standard output
100	9
000	BL
000	BL
111	BL
111	AL
011	C1
	AL
	C1
	AR
	C1

Note

It can be proven that, under the condition of valid data, there must be a solution to the original question within step 999, and the weakened version of the question must have a solution within step 81.

The most important thing is that 9-nine- will release a new work this year and is expected to be launched on Steam in the winter



Problem I. Ori

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

The Spirit Tree is the source of life on Nibel Mountain, with its roots flowing with light that maintains the ecological balance of the entire forest. The body of the ancient tree is composed of n cores of light, which are interconnected by $n-1$ energy branches to form an acyclic tree structure. Between any two cores, there is a unique energy branch path winding between them, as if woven by the threads of fate.

Ori, originally the guardian spirit nurtured by the ancient tree, was swept into the abyss by a violent storm on a rainy night when thunder tore the sky apart. The ancient tree, having lost Ori, attempted to call him back through a light ritual, but the uncontrolled energy backlash plunged it into eternal night, and the once clear cores are now covered with dark patterns.

Now, the returning Ori must activate all the eroded cores: when he first touches a core, pure energy will dispel the darkness; however, upon repeated passes, chaotic energy will accumulate to form overload waves. The ancient tree's laws strictly limit that the total number of repeated triggered waves along the entire path must not exceed k .

At this moment, Ori is suspended in the void woven by the star network. He can start from any core and traverse along the trajectories of the energy branches. Ori needs to plan a path among the winding energy branches to light up the maximum number of cores within the limits.

Only by resonating with as many light cores as possible can the true power of the ancient tree be awakened, allowing the healing light to surge again through every leaf vein of Nibel Mountain! Nibel Mountain will welcome dawn, and darkness will eventually be reduced to ashes in the resonance of this star network...

Formally, given a non-negative integer k , provide an unrooted tree $T = (V, E)$, where $V = \{1, 2, \dots, n\}$.

Define a path $l = (u_1, u_2, u_3, \dots, u_m)$ composed of m vertices such that for any $1 \leq i \leq m-1$, $(u_i, u_{i+1}) \in E$. If there exists $1 \leq i < j \leq m$ such that $u_i = u_j$, the path l is still considered valid.

Define the set of essential distinct points on path l as $V' = \{v \mid v = u_i, \exists i \in [1, m]\}$. Let $|V'|$ be the size of the set V' . Let the number of repeated points on path l be s , where $s = m - |V'|$.

You need to find a path l that maximizes $|V'|$ while satisfying $s \leq k$.

You need to output this path. If there are multiple paths that satisfy the conditions, output any one valid path, and it will be considered correct.



Input

The input contains multiple test cases. The first line of data contains an integer t ($1 \leq t \leq 2 \times 10^3$) indicating the number of test cases. The following describes each test case.

The first line of each test case contains two integers n ($2 \leq n \leq 2 \times 10^5$) and k ($0 \leq k \leq n$), separated by a space, representing the number of points on the tree and the maximum number of repeatable points. Note that k can be 0.

The next $n - 1$ lines each contain two positive integers u and v ($1 \leq u, v \leq n$), separated by a space, indicating that there is an edge (u, v) on the tree. It is guaranteed that these $n - 1$ edges form a tree.

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

Output

For each test case, output two lines.

The first line contains a positive integer m , indicating the number of vertices in path l . The path l must maximize the size of the essential distinct point set $|V'|$ while satisfying the number of repeated points $s \leq k$.

The second line contains m positive integers $u_1, u_2, u_3, \dots, u_m$, separated by spaces, indicating that the path l is $(u_1, u_2, u_3, \dots, u_m)$.

If there are multiple paths that satisfy the conditions, output any one valid path, and it will be considered correct.

It can be proven that for a valid path, m must satisfy $m \leq n + k$. The output m must satisfy $m \leq n + k$, otherwise it will directly return **Wrong Answer**.

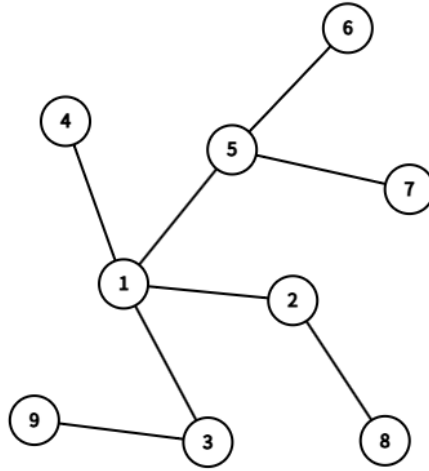
Example

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



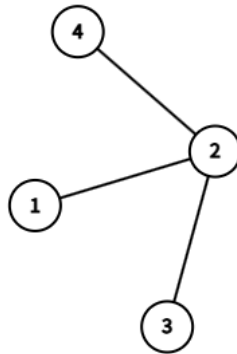
Note

For the first test case, $n = 9$ and $k = 3$, the tree formed is as follows:



One feasible solution is: $l = (9, 3, 1, 5, 7, 5, 6, 5, 1, 2, 8)$. It can be proven that the size of the essential distinct point set $|V'|$ reaches the maximum value of 8, and the number of repeated points $s = 3 \leq k$.

For the second test case, $n = 4$ and $k = 4$, the tree formed is as follows:



One feasible solution is: $l = (3, 2, 1, 2, 4)$. It can be proven that the size of the essential distinct point set $|V'|$ reaches the maximum value of 4, and the number of repeated points $s = 1 \leq k$.

Due to the large input and output data size for this problem, it is recommended to use faster input and output methods.



Problem J. Bastion

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

A bastion is the first type of fortress that relies solely on direct firepower and has no attack blind spots.

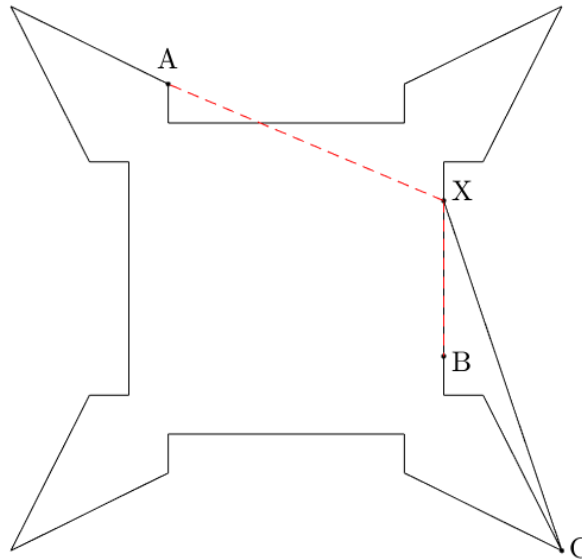


A simple non-degenerate polygon is a closed polygonal **region** composed of a sequence of n ($n \geq 3$) vertices, satisfying the following conditions:

- The vertex sequence connects end to end, forming n edges, constituting a compact closed set in the plane (including all boundary and interior points);
- The n edges only meet at the vertices and do not intersect each other (adjacent edges have no other intersection points outside the common vertex);
- The n vertices are distinct, not located inside any non-adjacent edges, and any three consecutive vertices are not collinear.



The bastion can be viewed as a simple non-degenerate polygon composed of n points and n edges. For two distinct points P and Q on the edges of the polygon, we define that point P can directly fire at point Q if and only if the line segment PQ intersects the polygon only at points P and Q . As shown in the figure below, points A and B cannot directly fire at point X , but point C can. If there is a point P on the edge of the polygon such that there is no other point Q on the edge of the polygon for which point Q can directly fire at P , then we call point P a fire blind spot of the polygon.



We call a simple non-degenerate polygon a bastion if and only if it has at most a finite number of points that are fire blind spots. Given a simple non-degenerate polygon, please determine whether it is a bastion.

Formally, given a simple non-degenerate polygon, please determine whether there are only a finite number of points P located on the edges of the polygon such that there is no other point Q on the edges of the polygon for which the line segment PQ intersects the polygon only at points P and Q .

Input

The input contains multiple test cases. The first line of the data contains an integer t ($1 \leq t \leq 10^4$) indicating the number of test cases. The following describes each test case.

The first line of each test case contains an integer n ($3 \leq n \leq 10^5$), which is the number of edges of the polygon.

The next n lines each contain two integers x_i and y_i ($-10^9 \leq x_i, y_i \leq 10^9$), describing a vertex of the polygon.

The input data guarantees that a simple non-degenerate polygon is formed. The vertices are given in counterclockwise order, that is, after traversing all the vertices in order, it rotates exactly 360° counterclockwise. Connect edges in sequence according to the input order of the vertices (the i -th vertex is connected to the $(i + 1)$ -th vertex, and the n -th vertex is connected back to the 1-st vertex).

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



Output

For each test case, output a single line containing a string. If the polygon is a bastion, output YES; otherwise, output NO. The answer is case insensitive.

Example

standard input	standard output
2	YES
20	NO
7 5	
9 5	
13 13	
5 9	
5 7	
-5 7	
-5 9	
-13 13	
-9 5	
-7 5	
-7 -5	
-9 -5	
-13 -13	
-5 -9	
-5 -7	
5 -7	
5 -9	
13 -13	
9 -5	
7 -5	
4	
1 1	
-1 1	
-1 -1	
1 -1	



Problem K. Seele Vollerei

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 32 megabytes

Please note the unusual space constraints for this problem.

Given an **operation sequence** A and a rooted tree T , each operation in A is like “add x to all point weights on the simple path from u to v in T .” The simple path from u to v is defined as a path starting from u and ending at v that does not repeatedly visit any nodes or edges.

Initially, the root of T is 1, and all point weights are 0.

You need to perform three operations:

1. Given an interval $[l, r]$, perform operations from $A_l, A_{l+1} \dots$ to A_r in sequence;
2. Query the sum of the subtrees of T with x as the root;
3. Change the root of T to y (If the current root is y , no operation will be performed).

Input

The first line contains three positive integers n_1 , n_2 , and m ($1 \leq n_1, n_2, m \leq 1 \times 10^5$), separated by a space, representing the length of A , the total number of points in T , and the number of inquiries and operations.

The next n_1 lines, the i -th line contains three integers u_i , v_i , and x_i ($1 \leq u_i, v_i \leq n_2, 1 \leq x_i \leq 1 \times 10^2$), separated by a space, representing the operation A_i as “add x_i to all point weights on the simple path from u_i to v_i in T .”

Next, there are $n_2 - 1$ lines, each containing two positive integers u and v ($1 \leq u, v \leq n_2$), separated by a space, representing an edge of the tree. It is guaranteed that the $n_2 - 1$ edges form a tree.

Next, there are m lines. The first number op ($op \in \{1, 2, 3\}$) in each line indicates the operation number, followed by:

- If op is 1, it is followed by two positive integers l and r ($1 \leq l \leq r \leq n_1$), separated by a space, indicating that the operations of $A_l, A_{l+1} \dots A_r$ are executed in sequence;
- If op is 2, it is followed by a positive integer x ($1 \leq x \leq n_2$), indicating that the sum of the subtrees of T with x as the root is queried;
- If op is 3, it is followed by a positive integer y ($1 \leq y \leq n_2$), indicating that the root of T is changed to y (If the current root is y , no operation will be performed).



Output

For each operation 2, output one line of one integer, representing the sum of the subtrees of T with x as the root.

The data ensures that the answer is within the range expressed by a signed long long integer.

Example

standard input	standard output
4 5 7	29
3 5 2	25
5 4 3	63
3 1 2	
2 5 1	
2 1	
4 2	
1 5	
3 2	
1 1 4	
2 1	
1 2 3	
2 2	
3 3	
1 1 3	
2 2	



Problem L. Gros-Phi

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Awa is participating in a live music game called Gros-Phi. In the event, Awa must appear at a specified location at a specified time.

Specifically, Gros-Phi's activity area is an infinitely long straight line. Gros-Phi has a total of n scoring points. The i -th scoring point requires Awa to appear at position x_i at time t_i .

Awa's maximum running speed is v units per moment. At time 0, Awa can choose any position and then start playing Gros-Phi.

Awa wants to know how many scoring points she can reach at most.

Input

The first line contains a positive integer T ($1 \leq T \leq 5 \times 10^5$), indicating that Awa played a total of T games.

For each game, the first line contains two positive integers n and v ($1 \leq n \leq 5 \times 10^5$, $1 \leq v \leq 10^9$), separated by a space, indicating the number of decision points and the maximum speed of Awa.

The next n lines, each containing two integers t_i and x_i ($0 \leq t_i, |x_i| \leq 10^9$), separated by a space, describe a scoring point. It is guaranteed that there are no two identical scoring points in a game.

It is guaranteed that the sum of n in T rounds of games does not exceed 5×10^5 .

Output

There should be T lines in total, each line contains one integer, indicating how many scoring points Awa can reach at most in the corresponding game.



Example

standard input	standard output
3	3
6 1	2
8 7	2
8 -6	
10 -8	
2 5	
7 -9	
1 0	
6 1	
0 -6	
0 0	
8 2	
10 -8	
9 -5	
2 -9	
6 1	
7 4	
8 -4	
8 9	
3 -9	
1 0	
7 2	

Note

Since the input and output data of this question are large, it is recommended to use a faster input and output method, such as `scanf` and `printf`.



Problem M. History of Terra

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

“Doctor, the connection between us will transcend even time and space. I am sure of it.”

“Even if the ocean completely boils away and the atmosphere vanishes, even if all the moons in the sky are pulled into the vortex of our planet’s gravity, and even if the sun continues to mercilessly expand, eventually swallowing up all its children until none can be heard... I am sure we will meet again.”

“At the far end of our civilization, once adorned by darkness and the brilliance of the stars up above, we will meet once again.”

“I will wait until that day comes. I promise you I will. Wait for me. You must wait for me, too.”

“It’s so quiet here... Too quiet. Don’t leave me behind.”

You are the Doctor of Rhodes Island, the Babel’s Omen. But above all, you are the Prophet of Priestess. Civilization teeters on the brink — yet, against all odds, you remembered her before it was too late. Now go, pull her back from the darkness. Do not betray the one who waited through time itself.



This is the first step to solve the mystery, a counting problem.

There is a sequence $a_1, a_2, \dots, a_n$ of length n , **elements in the sequence are different**, and an operation is defined as follows:

- Select a subscript i ($1 \leq i \leq n - 1$) that satisfies $|a_i - a_{i+1}| = 1$, and swap a_i with a_{i+1} in the sequence.

You can perform **any number of** operations. How many different sequences can you get? Since the answer is large, please output the answer modulo 998244353.

Two sequences $a_1, a_2, \dots, a_n$ and $b_1, b_2, \dots, b_n$ are different if and only if there exists $i \in [1, n]$ such that $a_i \neq b_i$.



Input

There are two lines of input.

The first line contains a positive integer n ($1 \leq n \leq 10^6$), which indicates the length of the sequence.

The second line contains n positive integers a_i ($1 \leq a_i \leq 10^9$), separated by a space, which indicates that the initial sequence is $a_1, a_2, \dots, a_n$. The data guarantees that a_i in the sequence are different.

Output

Output a single positive integer, representing the number of possible sequences modulo 998244353.

Examples

standard input	standard output
4 1 4 2 3	3
9 11 4 5 14 19 1 9 8 10	6
1 1	1
12 4 3 7 6 8 11 9 10 12 14 13 15	90

Note

For the first sample, the initial sequence is 1, 4, 2, 3. Note that the initial sequence is also a possible sequence and needs to be counted.

For the sequence 1, 4, 2, 3, since $|a_3 - a_4| = 1$, a_3, a_4 can be swapped, and the sequence after swapping is 1, 4, 3, 2.

For the sequence 1, 4, 3, 2, since $|a_2 - a_3| = 1$, a_2, a_3 can be swapped, and the sequence after swapping is 1, 3, 4, 2.

It can be proved that for this sequence, there are only the above 3 possibilities for different sequences after any number of operations.

Since the input and output data scale of this question is large, it is recommended to use a faster input and output method.



Problem N. Kessoku Band

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

"I'm here to end the band"

A girl band is taking a group photo, but the photographer has bad intentions. She plans to process the photos and then post them on social media, causing dissatisfaction among fans, which will cause the popularity of some members of the band to rise or fall. She thinks this will end the band. Her plan is as follows:

It is known that the girl band has n members, who are lined up from left to right for the photo. The initial popularity ranking of each member is different, ranging from 1 to n . The photographer plans to take k photos and publish them on social media after each photo. Each of her photos only contains a continuous section of the band member permutation, so that the fans of the most popular member who does not appear in this photo will be extremely angry and quit following her, causing the ranking of this member to drop by one (the original ranking of the member who was one lower will rise). Through k operations, she successfully disrupted the ranking of the band members, and her goal was achieved.

But now she has a problem because there are too many members in the band; for each captured photo, she can't judge which member who doesn't appear ranks first, so she turned to you for help, hoping you can help her solve the problem.

Formally, given a permutation $a_1, a_2, \dots, a_n$ from 1 to n , perform k operations, each operation:

- Given an interval $[l, r]$, find the **smallest positive integer that does not appear in the interval $[l, r]$ in the current permutation**, and set this number to x (i.e. $x = \min\{p \mid \forall i \in [l, r], p \neq a_i, p \geq 1\}$);
- If $x < n$, output x , and swap the positions of the numbers x and $x + 1$ in the current permutation $a_1, a_2, \dots, a_n$;
- If $x \geq n$, output the string **peace**.

Wherein, the permutation $a_1, a_2, \dots, a_n$ from 1 to n refers to a set of numbers that satisfies $\{a_1, a_2, \dots, a_n\} = \{1, 2, \dots, n\}$.

Input

The first line contains a positive integer n ($1 \leq n \leq 5 \times 10^5$), indicating the size of the permutation.

The second line contains n positive integers a_i , separated by a space, indicating the initial permutation $a_1, a_2, \dots, a_n$ ($\{a_1, a_2, \dots, a_n\} = \{1, 2, \dots, n\}$).

The third line contains a positive integer k ($1 \leq k \leq 10^5$), indicating the number of operations.

The next k lines, each containing two positive integers l and r ($1 \leq l \leq r \leq n$), separated by a space, indicate the range of operations.



Output

For each operation in each test data, give the answer and perform the modification.

Remember that the smallest positive integer that does not appear in the interval $[l, r]$ in the permutation $a_1, a_2, \dots, a_n$ is x :

- If $x < n$, output x and swap the positions of the numbers x and $x + 1$ in the current permutation $a_1, a_2, \dots, a_n$;
- If $x \geq n$, output the string **peace**.

Examples

standard input	standard output
5 4 3 1 2 5 5 2 4 2 5 1 3 1 3 1 5	4 peace 2 3 peace
10 10 9 4 3 2 1 6 5 8 7 3 1 10 2 9 3 9	peace 7 8

Note

For the first sample:

The first operation is $[2, 4]$, and the current permutation is 4, 3, 1, 2, 5.

The smallest positive integer that does not appear in the interval $[2, 4]$ is 4. Output 4 and swap the positions of 4 and 5 in the permutation.

The second operation is $[2, 5]$, and the current permutation is 5, 3, 1, 2, 4.

The smallest positive integer that does not appear in the interval $[2, 5]$ is 5. Output **peace**, and the permutation remains unchanged.

The third operation is $[1, 3]$, and the current permutation is 5, 3, 1, 2, 4.

The smallest positive integer that does not appear in the interval $[1, 3]$ is 2. Output 2 and swap the positions of 2 and 3 in the permutation.



The fourth operation is $[1, 3]$, and the current permutation is 5, 2, 1, 3, 4.

The smallest positive integer that does not appear in the interval $[1, 3]$ is 3. Output 3 and swap the positions of 3 and 4 in the permutation.

The fifth operation $[1, 5]$, the current permutation is 5, 2, 1, 4, 3.

The smallest positive integer that does not appear in the interval $[1, 5]$ is 6. Output **peace**, the permutation remains unchanged.



Problem O. Discrete Logarithm

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Given positive integers a, c, p , ensure that p is **prime**, find b such that:

$$a^b \equiv b^c \pmod{p}$$

We say that integers A, B, C have $A \equiv B \pmod{C}$ if and only if there exists an integer k such that $A - B = C \times k$.

Input

Input is just one line of three integers a, c , and p ($1 \leq a, c < p \leq 10^9$), separated by a space, with the meaning as described in the question.

The data guarantees that p is a prime.

Output

Output only one integer b ($1 \leq b \leq 10^{18}$). If there are multiple legal answers, you can output any one.

It can be proved that there is at least one solution in the range.

Examples

standard input	standard output
3 5 7	16
14530529 19260817 19491001	5660025

Note

For the first sample, we have:

$$3^{16} \equiv 16^5 \pmod{7}$$

Because:

$$3^{16} \bmod 7 = 43046721 \bmod 7 = 4$$

$$16^5 \bmod 7 = 1048576 \bmod 7 = 4$$