Problem A. August

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

"Remember when I pulled up and said get in the car, and then cancelled my plans just in case you'd call, back when I was living for the hope of it all."

After death by a thousand cuts, he found his love. He has two parameter a and b, and he wants to show his lover a piece of drawing bounded by the following math curves.

$$\begin{cases} y = \sqrt{a^2 - (x-a)^2} \\ y = \sqrt{a^2 - (x+a)^2} \\ y = \frac{2b}{\pi}\left(\arccos\left(\frac{x+a}{a}\right) - \pi\right) \\ y = \frac{2b}{\pi}\left(\arcsin\left(\frac{x-a}{a}\right) - \frac{\pi}{2}\right) \end{cases}$$

Now his lover wants to know the area bounded by the close curve. Can you tell him?

Input

The first line contains T ($1 \le T \le 1000$), the count of testcases.

Then the next T lines, each of them contains two integer a and b ($1 \le a, b \le 1000$).

Output

For each test case, output a number for the answer with an absolute or relative error of at most 10^{-4}. Precisely speaking, assume that your answer is a and and the jury's answer is b, your answer will be considered correct if $\frac{|a-b|}{max\{1,|b|\}} \le 10^{-4}$, where $\max\{x, y\}$ means the maximum of x and y and $|x|$ means the absolute value of x.

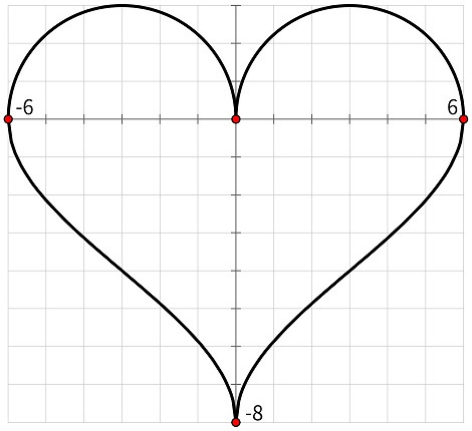
Example

standard input	standard output
2 3 4 1000 1000	76.27433388 7141592.65358979

Note

August sipped away like a bottle of wine.

This is the rendered picture for the example.



Problem B. Bills of Paradise

Input file:	standard input
Output file:	standard output
Time limit:	9 seconds
Memory limit:	256 megabytes

Little Y lost the racing game and signed his name on the contract. He has to pay for n bills! And the i-th bill is of a_i dollars.

However, the owner of contract was devil. The owner rolled eyes and gave him a random number generator with given random seeds denoted by two integers k_1 and k_2 to get the sequence of a_i. The following code in C++ tells you how to generate the sequence. You can use the code directly in your submissions.

```
unsigned long long k1, k2;

unsigned long long xorShift128Plus() {
    unsigned long long k3 = k1, k4 = k2;
    k1 = k4;
    k3 ^= k3 << 23;
    k2 = k3 ^ k4 ^ (k3 >> 17) ^ (k4 >> 26);
    return k2 + k4;
}

int n;
long long a[1000001];

void gen() {
    scanf("%d %llu %llu", &n, &k1, &k2);
    for (int i = 1; i <= n; i++) {
        a[i] = xorShift128Plus() % 999999999999 + 1;
    }
}
```

Q commands are sent from the owner. There are four types of commands.

- `F` x: Query the smallest unpaid bill a_i where $a_i \geq x$ holds. If such a_i doesn't exist, print 10^{12}.
- `D` x: Pay the smallest unpaid bill a_i where $a_i \geq x$ holds. If such a_i doesn't exist, do nothing.
- `C` x: Query the summation of all unpaid bills a_i where $a_i \leq x$ holds. If such a_i doesn't exist, print 0.
- `R` x: Refund (退款) all paid bills a_i and mark them unpaid where $a_i \leq x$ holds. If such a_i doesn't exist, do nothing.

You are required to process these commands now.

Input

The first line contains one integer T ($1 \le T \le 5$) denoting the count of testcase.

For each testcase,

The first line contains integers n, k_1, k_2 ($1 \le n \le 10^6, 0 \le k_1, k_2 \le 2^{64} - 1$), which is used in function `gen()`.

The second line contains integer Q ($1 \le Q \le 5 \cdot 10^5$).

Then follow Q lines, each containing one command $op\ x$ ($op \in \{\texttt{F}, \texttt{D}, \texttt{C}, \texttt{R}\}, 1 \le x \le 10^{12} - 1$).

It is guaranteed that the count of type `R` does not exceed 10, and $a_i \neq a_j$ holds for each pair of (i, j) where $i \neq j$.

It is guaranteed that $\sum n \le 3.2 \times 10^6$ and $\sum Q \le 1.7 \times 10^6$.

Output

For each command `F` and command `C`, output a line containing the answer.

Example

standard input	standard output
1 8 1234567890987 654321234567 8 F 234567891234 D 345678912345 C 456789123456 R 567891234567 C 100000000000 D 100000000000 C 654321234567 F 987654321234	245682167348 680270154870 0 1552964262449 1000000000000

Note

For this example, the generated sequence is $\{565040138009, 102855093402, 245682167348, 331732894120, 987193824410, 699419056191, 780922390772, 410509062972\}$

Problem C. Cornelia Street

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

"We bless the rain on Cornelia Street, memorize the creaks in the floor."

Creak, creak, creak... The drum beats regularly. To keep the number of syllables in harmony, one sentence in chorus is as long as one in verse. However, drumbeats in the chorus is much stronger.

Jessica wants to memorize the music, piano and drumbeat. She records the music and encodes it into a string. So, the recorded string is so strange.

Let A be the pattern of verse and B be the pattern of chorus. So $A \neq B$ and $|A| = |B|$. The recorded string S is in the pattern of $AAA\cdots A\ BB\cdots B\ AAAAA\cdots Aa$. Formally, it is formed as n times repeat of A, m times repeat of B, k times repeat of A, and the broken piece a at the end is one prefix of A. And note that $n, m, k > 0, |a| \geq 0$.

Now Jessica wants to know the pattern of A and B. Can you help her?

Input

The first line contains a number T ($1 \leq T \leq 10$) denoting the count of testcases.

For each testcase, one line containing a string S ($7 \leq |S| \leq 8 \times 10^5$). Note that S consists of only lowercase letters and uppercase letters.

It is guaranteed that $\sum |S| \leq 4.5 \times 10^6$. The patterns in all testcases are generated randomly, but the character sets are different. There are 30% testcases generated in full character set, and the rest 70% are generated in large number of single character and small number of other characters.

Output

For each testcase, output one line containing string A and B respectively, separated by one space. If there are multiple answers, choose the shortest one.

Example

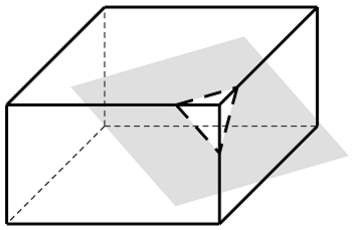
standard input	standard output
3 aaaaaaaaaabaabaaaaaa ABABABABZXCVZXCVABABABA abcababcacabcacabcababc	aaa aba ABAB ZXCV abcab abcac

Problem D. Death by Thousand Cuts

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

While Jonathan was doing his math homework, he found a multiple-choice question:

Given a cuboid with each edge parallel to the coordinate axis, two vertexes of which are $(0, 0, 0)$ and (a, b, c), and a plane $Ax + By + Cz + D = 0$ which intersects the cuboid. You are asked to calculate the number of edges of the cross section. Four options are provided: A. three, B. four, C. five and D. six.



Unfortunately, the number D has been covered by black ink so Jonathan didn't know what exactly it is. Therefore, D could be the any real number satisfying the plane intersects the cuboid in equal probability.

Then Jonathan found some options had significantly greater probability to be the correct answer in some cases. For example, if the cuboid is $(0, 0, 0), (1, 1, 1)$ and the plane is $x + y + D = 0$, then the option B has the 100% probability to be the right answer. Therefore, he wanted to know each option's probability to be the correct answer in normal cases. As the problem is too hard for him, please help him to calculate the answer.

We can prove that the probability of each option is a rational number. Let's denote it by irreducible fraction Q/P, you are only to print the value $Q \times P^{-1} \bmod 10^9 + 7$. Here P^{-1} is multiplicative inverse of P modulo $10^9 + 7$.

Input

The first line of the input gives the number of test cases, T ($1 \le T \le 10^4$). T test cases follow.

For each test case, only one line contains 6 integers a, b, c, A, B, C ($1 \le a, b, c \le 10^4, 0 \le |A|, |B|, |C| \le 10^4$, $|A| + |B| + |C| > 0$), representing the diagonal point (a, b, c) and the parameters of the plane equation.

Output

For each test case, print a line with four integers P_1, P_2, P_3, P_4, representing the probability of options A, B, C and D respectively.

Example

standard input	standard output
3 1 2 3 1 1 1 2 2 2 1 1 1 2 2 2 2 -1 1	333333336 333333336 333333336 0 666666672 0 0 333333336 500000004 0 500000004 0

Problem E. Everybody Lost Somebody

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

"But there's nothing I wouldn't do to wake up and remember it."

Jonathan is fan of string problems. He is learning lexicographic order and suffix array these days.

String x is lexicographically less than string y, if either x is a prefix of y (and $x \neq y$), or there exists such i ($1 \leq i \leq \min(|x|, |y|)$), that $x_i < y_i$, and for any j ($1 \leq j < i$), $x_j = y_j$. Here $|a|$ denotes the length of the string a. The lexicographic comparison of strings is implemented by operator `<` in modern programming languages. For example, `everybody` is lexicographically smaller than `somebody`.

Let $\mathrm{suf}\, i$ be $x_i x_{i+1} \cdots x_n$ for string x of length n. In suffix array problems, there are two commonly used arrays: sa of length n and $height$ of length $n-1$. Formally, sa_i ($1 \leq i \leq n$) is the starting position of the i-th lexicographically smallest suffix $\mathrm{suf}\, j$, which means $sa_i = j$. And $height_i$ ($2 \leq i \leq n$) is the length of longest common prefix between $\mathrm{suf}\, sa_{i-1}$ and $\mathrm{suf}\, sa_i$. For example, the sa and $height$ for `remember` is $\{6, 4, 2, 7, 5, 3, 8, 1\}$ and $\{0, 2, 1, 0, 1, 0, 1\}$ respectively.

As we all know, Little Y is a Riddler. One day, Little Y got a string S of length n consisting of only lowercase letters. He used suffix-array algorithms to get the array sa and $height$. He erased several numbers in $height$ and gave the two modified array to Jonathan.

Curiously, Jonathan wants to know what the string S is. Please help him to figure out a possible answer. Since there may be multiple answers, you only need to print the lexicographically smallest one. It is guaranteed that the answer exists.

Input

The first line contains one integer T ($1 \leq T \leq 15$) denoting the count of testcase.

For each test case,

The first line contains one integer n ($1 \leq n \leq 5000$), indicating the length of the string S.

The second line contains n integers $a_1, a_2, ..., a_n$, indicating the array sa.

The third line contains $n-1$ integers $b_2, b_3, ..., b_n$, indicating the modified array $height$. $b_i = -1$ means that $height_i$ is erased by Little Y. Otherwise, $height_i = b_i$.

It is guaranteed that $\sum n \leq 4.1 \times 10^4$.

Output

For each testcase, one line with one string S of length n, indicating the lexicographically smallest answer.

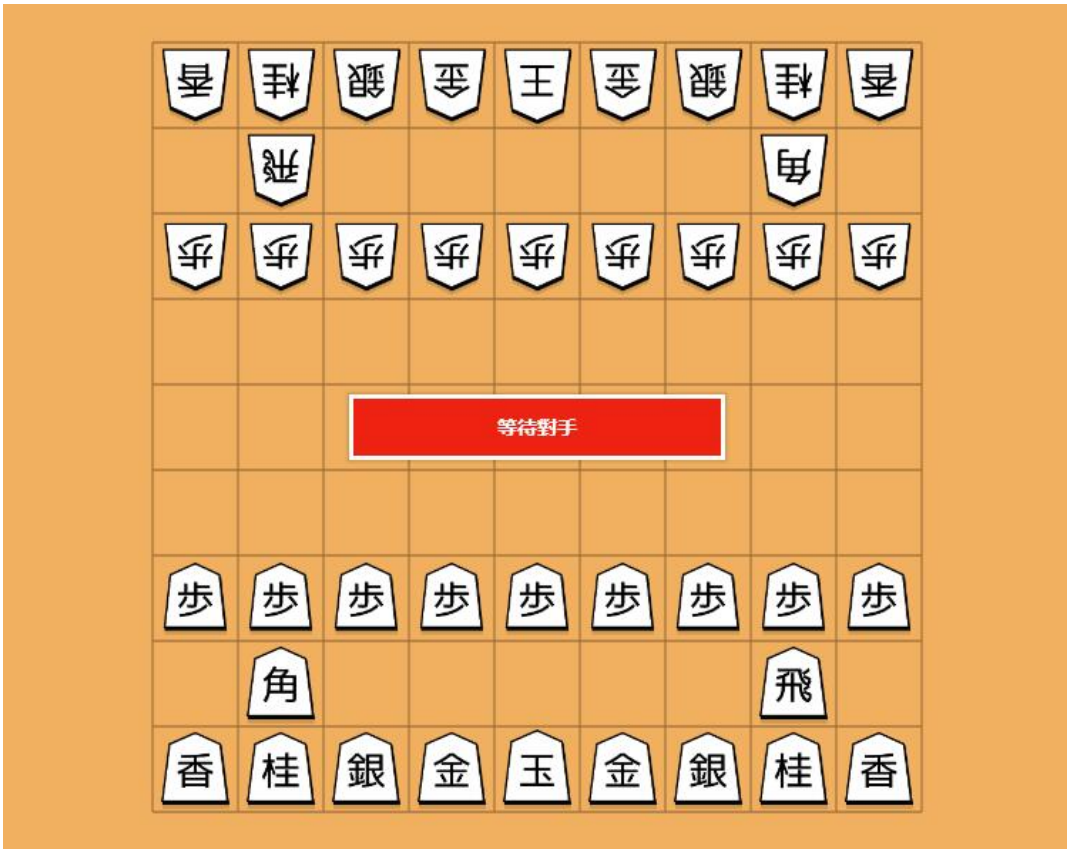
Example

standard input	standard output
3 4 4 1 2 3 1 0 0 4 4 1 2 3 1 -1 0 11 11 4 8 1 5 9 2 6 10 3 7 -1 -1 -1 0 -1 -1 -1 0 -1 3	abca aaba abcabbcabca

Problem F. False God

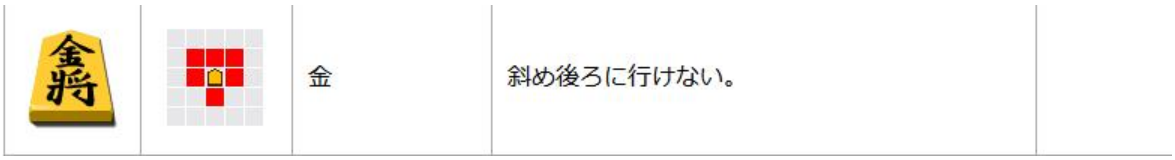
Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are facing an endgame of Shogi.

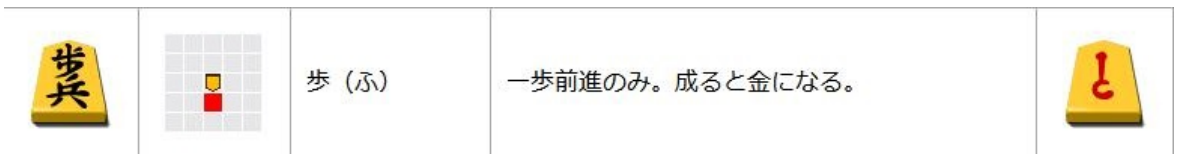


You have only one Kin(金将). However, your opponent has n Fu(歩兵).

The rule of this game is as following.



If Kin is at (x, y), it can move to one of the following positions in one step: $(x+1, y)$, $(x+1, y+1)$, $(x, y+1)$, $(x-1, y+1)$, $(x-1, y)$, $(x, y-1)$.



If Fu is at (x, y), it can only move to $(x, y-1)$ in one step.

Now you know the initial position of your Kin and your opponent's Fu. Each turn, you move your Kin first and then all Fu will move one step.

When you are moving Kin, you can defeat the Fu at that position. If one Fu moves one step and meets Kin, your opponent is also defeated.

To win this game, you want to defeat Fu as many as possible. Now Chino wonders the maximum count of Fu that will be defeated. Can you help cute Chino?

Input

The first line contains one integer T ($1 \le T \le 10$) denoting the number of testcases.

For each testcase,

The first line contains two integers x_0, y_0 ($|x_0|, |y_0| \le 1\,000\,000\,000$) denoting the initial position for Kin.

The second line contains one integer n ($1 \le n \le 1\,000$) denoting the number of Fus your opponent has.

The next n lines each contains two integers x_i, y_i ($|x_i|, |y_i| \le 1\,000\,000\,000$) denoting the position of Fu i ($1 \le i \le n$).

It is guaranteed that $(x_i, y_i) \neq (x_j, y_j)$ for all pairs of i, j satisfying $0 \le i < j \le n$.

It is guaranteed that $\sum n \le 5000$.

Output

One line for each testcase with one integer denoting the answer.

Example

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

Note

For the second testcase, your Kin can defeat the last 4 Fu.

Problem G. Goodbye

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

"Goodbye to the friends I've had, goodbye to my upstairs neighbor, goodbye to the kids downstairs, and anybody who lend me a favor."

"Or, goodbye to the positive proper divisor?"

That game is popular these days. Two player take turns to choose a specific number and claim it. The player who takes the first turn chooses one positive proper divisor of given n. In the next rounds, player should choose one positive proper divisor (真因数) of the number claimed in the previous round. The player who can't make a choice will win.

Formally, a positive proper divisor of n is a divisor of n, and it cannot be either 1 or n.

Now Chino is going to play this game with David. Chino always wants to win and meanwhile she wants to keep the chosen number as big as possible. If with the given n, Chino will win directly or fail anyway, you should also tell her.

Can you help Chino to choose her answer in the first round?

Input

The first line contains one integer T ($1 \leq T \leq 10^3$) denoting the count of testcases.

For each testcase, one line with one integer n ($1 \leq n \leq 10^5$) denoting the given n in one game.

Output

For each testcase, you should output the biggest choice, or 0 when Chino will win directly, or -1 when Chino will fail anyway.

Example

standard input	standard output
4 3 4 8 666	0 -1 4 111

Problem H. Hate That You Know Me

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Little Y can't help himself learning number theory. However, math is too hard for him. He was beaten by `SPOJ AFS3` and felt down. So here is a simple math problem.

Let $\sigma_k(n)$ be the following definition:

$$\sigma_k(n) = \sum_{d|n} d^k$$

For example, when $k = 0$, this function is known as the count of divisors of n. And when $k = 1$, this function is known as the sum of divisors of n.

Now he wants to calculate the following fomula for given a and b.

$$\left(\left(\sum_{i=1}^{n} \sigma_a(i)\right) \oplus \left(\sum_{i=1}^{n} \sigma_b(i)\right)\right) \mod 2^{64}$$

where $\oplus$ means the bitwise exclusive or.

Input

The first line contains an integer T ($1 \le T \le 15$) denoting the count of testcase.

For each testcase, one line containing three integer a, b and n.

To be much simpler, it is guaranteed that $0 \le a, b < 4$ and $1 \le n \le 10^{12}$. Then you can solve this problem without either interpolation or `SPOJ DIVCNT1`.

Output

For each testcase, one line containing the value.

Example

standard input	standard output
2 0 1 4 2 3 2	7 12

Problem I. InkBall FX

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

Recently, Jonathan is addicted to a game called InkBall FX.

The game goes on a plane coordinate system. On the first quadrant, there are some segments. The segments are all parallel to x-axis and they are pairwise disjoint. Two segments are disjoint if and only if they don't have mutual point, including the end points. Therefore, we can use (L_i, R_i, Y_i) to note the i^{th} segment, which means the segment starts from the position (L_i, Y_i) and ends in the position (R_i, Y_i).

There is an extremely small ball on the position $(0, 0)$. At the beginning, the speed of the ball in the horizontal direction is 1 per second, and that in the vertical direction is also 1 per second. If the ball is now at the position (x, y), then it will move to the position $(x + t, y + t)$ after t seconds, if it does not touch the segment during this process.

When the ball hits any segment, a completely elastic collision occurs, which means the speed in vertical direction will be reversed and that in the horizontal direction will remain the same, since the segments are parallel to x-axis. After the collision, the segment will disappear, which means it cannot be collided twice.

If the ball hits the ends of the segment, the collision still occurs like what's mentioned before.

Jonathan is curious about how many segments the ball could hit, but the ball moves too slow. Please write a program to help him to get the answer quickly.

Input

The first line contains one integer T ($1 \le T \le 10$) denoting the count of testcase.

For each testcase,

The first line contains one integer n ($1 \le n \le 10^5$), representing the number of segments.

Then n lines follow. The i-th line contains three integers L_i, R_i, Y_i ($1 \le L_i, R_i, Y_i \le 10^9, L_i < R_i$), representing the i-th segment.

Output

For each testcase, one line with an integer x – times that collision occurred.

Example

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

Problem J. Jingle Bells

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

Chino has n jingle bells, and she plans to decorate them on the Christmas tree one by one.

However, this Christmas tree is strange. This tree has n nodes, numbered from 1 to n. Each node has two value a_i, b_i. When Chino puts a jingle bell on node i, she will gain beauty value. Formally, **after** putting one jingle bell, let S be the set of nodes which contains at least one jingle bell, she will get $b_i \times \sum_{j \notin S} a_j$ points.

At the beginning, Chino can only put bells on the root node of the Christmas tree and get 0 points. Then Chino can put jingle bells on any node v satisfying $(u, v) \in E(G), u \in S, v \notin S$ and get its beauty value.

Chino want to make Christmas tree the most beautiful, but she don't know the maximum beauty value she can get. Can you help her?

Input

The first line contains an integer T ($1 \leq T \leq 5$) denoting the count of testcase.

For each testcase,

The first line contains an integer n ($1 \leq n \leq 100\,000$) denoting the numbers of nodes and jingle bells.

The second line contains $n - 1$ integers $f_2, f_3, \cdots, f_n$, and f_i represents the parent of node i is node f_i ($1 \leq f_i < i$).

The next n lines each contains 2 integers a_i, b_i ($0 < a_i, b_i \leq 10\,000$), which is for the node value a_i, b_i. It is guaranteed that $a_1 = b_1 = 0$.

It is guaranteed that $\sum n \leq 2.1 \times 10^5$.

Output

For each testcase, one line with one integer denoting the maximum beauty value.

Example

standard input	standard output
2 4 1 1 2 0 0 3 1 5 1 4 1 10 1 1 2 2 3 3 6 6 6 0 0 4 1 5000 1 3 1 6 1 200 1 1 1 1 1 1 1 1 1	14 16040

Note

For the first sample, we can put the jingle bells in the order of $1 - 2 - 4 - 3$.

Problem K. Keeping A Secret

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

Little Y is a Riddler. He always says that this is a secret.

One day, he got a undirected tree. He met Jessica and say, "Hey! I've got a new tree. Do you want to guess? This is a secret."

"This is a tree with n nodes. Each node has three restrictions and two parameters d_i and x_i."

- The depth of node i in the tree is d_i.
- The position of i in the BFS sequence of tree should be less than or equal to x_i.
- If x is before y in BFS sequence, either the parent node of x is also the parent node of y, or the parent node of x is before the parent node of y in BFS sequence.

However, Jessica quickly finds that too many trees satisfy his conditions. To give him a lesson, she wants to figure out the number of the trees satisfying all these restrictions. Can you help her?

Input

The first line contains one integer T ($1 \le T \le 10$) denoting the count of testcase.

For each testcase,

The first line contains one integer n ($1 \le n \le 100\,000$) denoting the number of nodes in the tree.

The next n lines each contains two integer d_i, x_i ($1 \le d_i, x_i \le n$) denoting the restriction of node i.

It is guaranteed that $\sum n \le 5.1 \times 10^5$.

Output

For each testcase, output one line containing the number of trees modulo $10^9 + 7$.

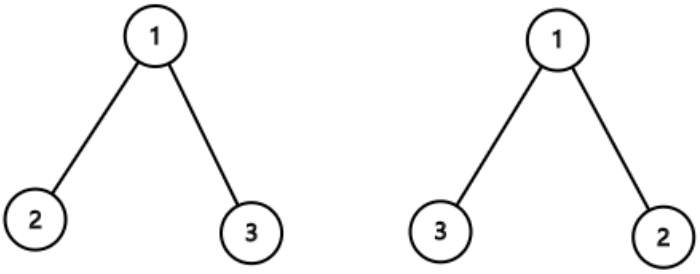
Example

standard input	standard output
1 9 1 1 2 3 2 3 2 5 3 6 3 7 3 8 3 8 3 9	336

Note

The depth of root is 1.

Note that the BFS sequence of following two trees are different. They are $1-2-3$ and $1-3-2$ respectively.



Problem L. Let's Get Married

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

"Don't wanna walk alone. So, let's get married."

Little Y lives in a strange town, which can be represented by a finite 2D plane. Each grid point on the 2D plane has a unique number. The unique number is generated by breadth first search. The origin point $(0, 0)$ is numbered as 0. We search for the points which is not accessed yet (in the order of top, right, bottom, left) and number them with the smallest available positive number.

$$\begin{array}{cccccc} & & 5 & & & \\ & 7 & 1 & 6 & & \\ \cdots & 4 & 0 & 2 & 8 & \\ & \cdots & 3 & 9 & & \\ & & \cdots & & & \end{array}$$

For example, the point $(1, -1)$ is for 9 and the point 6 is at $(1, 1)$.

One day, Little Y met Little K and quickly fell in love.

Little Y wants to follow Little K's steps. Surprisingly, Little K will publish his location.

Once noticing the location, Little Y will go to that point. However, Little Y is not clever and need your help.

Input

The first line contains T ($1 \le T \le 10^4$), denoting the number of Little K's position.

The next T lines contains the messages in the following two ways.

- `1 id` for the unique number
- `2 x y` for the position

It is guaranteed that all points satisfy $|x| + |y| \le 10^8$

Output

Initially, $(x_{current},\ y_{current})$ is set to $(0, 0)$.

For each message, you should help Little Y in the following instructions.

For `1 id`, you should output the position difference $(x - x_{current},\ y - y_{current})$ of point id at (x, y).

For `2 x y`, you should output the unique number of the point at this position.

After processing the current message, $(x_{current},\ y_{current})$ will be set to (x, y).

Example

standard input	standard output
6 1 8 2 3 3 1 14 2 5 1 1 2 1 0	2 0 66 -2 -1 70 -4 -1 -1 0