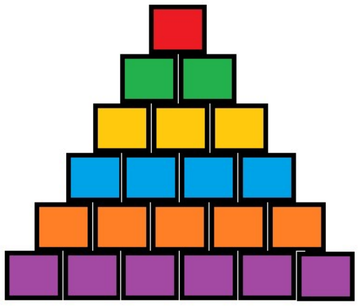
Problem A. Admiral

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Suppose that you are an admiral of a famous naval troop. The troop consists of 21 battleships. There are 6 kinds of battleships. First, there is one flagship in which the admiral must be, and it is denoted by the integer 0. Other battleship kinds are denoted by integers from 1 to 5, and there are 2, 3, 4, 5, and 6 ships of these kinds, respectively.

The troop is going to a battle against the enemy. Hence, the correct arrangement of the battleships is very important. The shape of the battlefield is shown on the picture to the right. For simplicity, we consider all battleships to have the same rectangular shape.



Fortunately, the optimal arrangement of the battleships is already known. As you can see, the battlefield consists of 6 rows. In the optimal arrangement, all the battleships denoted by number i must be located in the i-th row (rows are numbered starting from 0).

You are given the initial state of the battlefield as input. The battleships occupy the 21 required positions, but some of them may be in a wrong row. You can change the state of battlefield by swapping the position of **the flagship** with an adjacent battleship. Two battleships are considered adjacent if and only if they are not in the same row, but share parts of their edges. For example, if we denote a cell in i-th row and j-th position from the left as (i, j), then the cell $(2, 1)$ is adjacent to the cells $(1, 0)$, $(1, 1)$, $(3, 1)$, and $(3, 2)$ (here, rows and positions are numbered starting from 0). Your task is to make the minimum possible number of swaps so as to reach the optimal arrangement, or to report that more than 20 swaps are required.

Input

The first line of input contains an integer T, the number of test cases ($1 \leq T \leq 10$).

Each test case consists of 6 lines. The i-th line of each test case contains i space-separated integers denoting the kinds of battleships at the i-th row of the battlefield, listed from left to right.

Output

For each test case, if you can't reach the goal in 20 or less swaps, print "`too difficult`" on a single line. Otherwise, print one integer: the minimum possible number of swaps.

Example

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

Problem B. Array Challenge

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Consider an array h such that its elements are defined as follows:

$$\begin{aligned} h_0 &= 2, \\ h_1 &= 3, \\ h_2 &= 6, \\ h_n &= 4h_{n-1} + 17h_{n-2} - 12h_{n-3} - 16 \text{ for } n \geq 3. \end{aligned}$$

Additionally, let us define two arrays b and a as shown below:

$$\begin{aligned} b_n &= 3h_{n+1}h_n + 9h_{n+1}h_{n-1} + 9h_n^2 + 27h_nh_{n-1} - 18h_{n+1} - 126h_n - 81h_{n-1} + 192 \text{ for } n > 0, \text{ and} \\ a_n &= b_n + 4^n \text{ for } n > 0. \end{aligned}$$

Your task is to find the value $\lfloor \sqrt{a_n} \rfloor$ for a given integer n. As the answer could be very large, print it modulo $10^9 + 7$.

Input

The first line of input contains an integer T, the number of test cases ($1 \leq T \leq 1000$).

Each test case consists of a single line containing an integer n ($2 \leq n \leq 10^{15}$).

Output

For each test case, print a single line with a single integer: the answer to the problem.

Example

standard input	standard output
3 4 7 9	1255 324725 13185773

Problem C. Boring Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

Yong Chol is playing a game with his brother. The game board is a $N \times N$ grid. Each cell of the grid contains a coin. Each coin is either heads up or tails up. The players take turns to flip coins. In each turn, the player selects a cell (x, y) $(1 \le x, y \le N)$ with a coin heads up, and also two integers w and h $(1 \le w \le x,\ 1 \le h \le y)$. After that, the player looks at the rectangle of cells with its opposite corners containing cells $(x - w + 1, y - h + 1)$ and (x, y), and flips all coins in this rectangle, changing heads to tails and vice versa.

The player who cannot make a move loses the game.

The game with his little brother is so boring for Yong Chol that he wants to finish it immediately. But before doing that, Yong Chol wants to know who will win if the two players play optimally from now on. The board may be rather large, so its current state is given as a list of rectangles such that their union represents the cells with a coin heads up, and all other cells contain a coin tails up. Yong Chol is to take the next move. Can you help him find out who will win?

Input

The first line of input contains an integer T, the number of test cases $(1 \le T \le 200)$.

Each test case starts with a line containing two integers: N, the size of the board, and M, the number of rectangles $(1 \le N \le 10^9,\ 1 \le M \le 10^5)$.

Each of the next M lines contains four integers x_1, y_1, x_2, and y_2: the coordinates of the two opposite corners of the rectangle $(1 \le x_i, y_i \le N,\ x_1 \le x_2,\ y_1 \le y_2)$.

The total sum of M over all test cases will not exceed $6 \cdot 10^5$. For at least 90 percent of the test cases, M will be smaller than 600.

Output

For each test case, if Yong Chol wins the game, print "`Yong Chol`", otherwise print "`Brother`".

Example

standard input	standard output

`2`
`3 2`
`1 2 1 3`
`2 1 3 1`
`2 1`
`1 1 2 2`
	`Brother`
	`Yong Chol`

Problem D. Brother and Sister

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Jin Song and Jin Hui are brother and sister. They go to the same university. Jin Song is a member of ICPC training club, and Jin Hui is a member of English club.

Today is Jin Hui's 17-th birthday, and it is also the first birthday at university. So, as a good brother, Jin Song prepared an amazing present for his pretty younger sister. But the mischievous sister went to university before he got up, so he decided to go to the English club and give his birthday present to her.

Jin Hui is a very clever girl, so she thought that her brother will surely come to see her and decided to prepare a funny trick for him. She has n fellows at the English club, and they all agreed with her idea. First, Jin Hui assigned numbers to herself and all her fellows: Jin Hui has number 0, and fellows are numbered by distinct integers from 1 to n. Then, she asked her fellows to wear the same uniform and thick glasses just like her, so that the $n+1$ girls look exactly the same. As a result, Jin Song will get confused and try to find his sister. The trick then goes as follows:

1. He randomly and uniformly chooses a girl i who wasn't asked yet.
2. He asks the chosen girl "Are you my sister?". If the current girl i is Jin Hui (that is, $i = 0$), she will immediately end this trick. Otherwise, girl i will say that Jin Hui is the girl p_i. The numbers p_i are agreed upon before Jin Song comes.
3. If $p_i = i$ (hence, a fellow girl says that she is Jin Hui) or Jin Song had already asked the girl p_i, he will notice that the girl told him a lie and go to Step 1. Otherwise, he will continue by asking girl p_i, going to Step 2.

Jin Song wants to find his sister as soon as possible, and he wants to know how long it will take to find his clever younger sister.

Your task is to find the expected value of the number of girls Jin Song will have to ask (including his younger sister) before the trick ends.

Input

The first line of input contains an integer T, the number of test cases ($1 \le T \le 10$).

Each test case consists of two lines. The first line contains an integer n, the number of Jin Hui's fellows ($1 \le n \le 10^5$). The second line contains n space-separated integers $p_1, p_2, \ldots, p_n$, where p_i is the number of the fellow to which fellow i redirects Jin Song ($1 \le p_i \le n$).

Output

It can be shown that the expected value can be represented as an irreducible fraction of the form p/q (that is, p and q are coprime integers). So, for each test case, print a single line containing a single integer: the value $(p \cdot q^{-1}) \bmod (10^9 + 7)$.

Example

standard input	standard output
1 3 2 3 1	250000005

Note

In the example, there are 4 ways to choose the first girl. And after Jin Song chooses the first girl, what happens next is uniquely determined. One way is to choose his sister first, and the others are to choose 1, 2, or 3.

If he chooses his sister first, the trick ends immediately, and he asks 1 girl in total.

If he chooses girl 1 first, then he will ask three girls: 1, 2, and 3. Then he will notice that they told him a lie and directly go to his sister. So he asks 4 girls in total, in the order $1, 2, 3, 0$.

The cases of choosing girl 2 or 3 first are similar to choosing girl 1: the sequences of girls are $2, 3, 1, 0$ and $3, 1, 2, 0$, respectively.

Each case occurs with the same probability of $1/4$.

So, the answer is $((1 + 4 + 4 + 4) \cdot 4^{-1}) \bmod (10^9 + 7) = 250\,000\,005$.

Problem E. Cube Summation

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

Given an integer N, consider all multi-sets of positive integers such that their sum is N.

For example, if $N = 3$, there are three possible multi-sets: $\{1, 1, 1\}$, $\{1, 2\}$, and $\{3\}$.

For each multi-set, calculate the cube of its size, and output the sum of all these values modulo 998 244 353.

Input

The first line of input contains an integer T, the number of test cases ($1 \le T \le 10^5$).

Each test case consists of a single line containing a single integer N ($1 \le N \le 10^5$).

Output

For each test case, output a single line with a single integer: the answer to the problem.

Example

standard input	standard output
4 1 2 3 100000	1 9 36 513842114

Note

For the first case, the only possible multi-set is $\{1\}$. So the answer is $1^3 = 1$.

For the second case, there are two possible multi-sets: $\{1, 1\}$ and $\{2\}$. So the answer is $2^3 + 1^3 = 9$.

For the third case, there are three possible multi-sets: $\{1, 1, 1\}$, $\{1, 2\}$, and $\{3\}$. So the answer is $3^3 + 2^3 + 1^3 = 36$.

Problem F. Function Counting

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Let M be the set of integers at most n by absolute value, that is, $M = \{x \in \mathbb{Z} \colon |x| \le n\}$.

Let $f_k(x)$ be the function f applied k times to an initial value x, that is, $f_0(x) = x$ and $f_i(x) = f(f_{i-1}(x))$ for any $i \ge 1$.

Given the integers n and k, count the number of functions $f(x)$ satisfying the following conditions:

1. $f \colon M \to M$,
2. $\forall x \in M \colon f_k(x) = -x$,
3. $\forall x \in M \colon |(|f(x)| - |x|)| \le 2$.

As the answer may be very large, print it modulo $10^9 + 7$.

Input

The first line of input contains an integer T, the number of test cases ($1 \le T \le 100$).

Each test case contains a pair of positive integers n and k ($n \cdot k \le 10^9$).

The total sum of $n \cdot k$ over all test cases does not exceed $4 \cdot 10^9$.

Output

For each test case, output the answer modulo $10^9 + 7$ on a separate line.

Example

standard input	standard output
7 1 1 2 1 100 1 1 2 2 2 3 2 20 4	1 1 1 0 2 0 1048576

Note

If $k = 1$, only the function $f(x) = -x$ satisfies all requirements.

If $n = k = 2$, two functions exist:
$(-2, -1, 0, 1, 2) \to (1, -2, 0, 2, -1)$ and
$(-2, -1, 0, 1, 2) \to (-1, 2, 0, -2, 1)$.

Problem G. Jacana Numbers

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

Let us define *Jacana numbers* in the following way:

$$\begin{aligned} J(n,1) &= n, \\ J(n,k) &= n^{J(n,k-1)} \text{ for } k > 1. \end{aligned}$$

We have two Jacana numbers: $J(n,a)$ and $J(m,b)$. Your task is to compare them.

Input

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

Each test case contains four integers: n, a, m, and b ($1 \le n, a, m, b \le 10^9$).

Output

For each test case, print a single line containing a single character:

- if $J(n,a) > J(m,b)$, output "`>`",
- if $J(n,a) = J(m,b)$, output "`=`", and
- if $J(n,a) < J(m,b)$, output "`<`".

Example

standard input	standard output
3 2 2 3 1 2 3 3 2 1 2 1 4	> < =

Problem H. Monkeys

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	512 mebibytes

Given is a tree with N vertices. There are K monkeys in the tree. The monkeys want to occupy some vertices of the tree so that each monkey is in some vertex and each vertex contains at most one monkey. Then, they want to remove some edges of the tree so that each monkey can still move to at least one other monkey using only the remaining edges.

Your task is to find the minimum possible number of remaining edges.

Input

The first line of input contains an integer T, the number of test cases ($1 \leq T \leq 100$).

Each test case begins with a line containing two integers N and K ($2 \leq K \leq N \leq 10^5$). The next line contains $N-1$ space-separated integers $a_1, a_2, \ldots, a_{N-1}$ ($1 \leq a_i \leq i$). They mean that, for each i, there is an edge between vertex a_i and vertex $i+1$.

Output

For each test case, print the minimum possible number of remaining edges.

Example

standard input	standard output
2 4 4 1 2 3 4 3 1 1 1	2 2

Problem I. Rotating Line

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 mebibytes

Consider an infinite two-dimensional plane. There are n fixed points and one rotating line given on the plane. One of the points is $(0, 0)$, the origin, and the line initially coincides with the y axis. The points are distinct, but it is **not guaranteed** that no three points lie on a same line.

The line is being rotated in counter-clockwise direction. The centre of rotation is always one of the n given points, initially the origin, but it may change. Specifically, as long as the line contains only one point, the rotation centre is this point. As soon as the line meets another point, consider all given points currently lying on our line, ordered as pairs of coordinates (x, y). If there are k points on the line, and the current rotation centre is the p-th of them, counting from 1, the next rotation centre will be the $(k - p + 1)$-th of these points. If $p = k - p + 1$, the rotation centre does not change. Initially, the line starts rotating even if there are several given points on it.

Your task is to find the q-th rotation centre. The initial centre is numbered by 0, and as long as the centre does not change, the counter does not increase. Nevertheless, it is guaranteed that, for the given set of points, the q-th rotation centre exists.

Input

The first line of input contains an integer n, the number of points ($2 \le n \le 3000$).

The i-th line of the following n lines contains two integers x and y, the coordinates of the i-th point ($-100 \le x, y \le 100$). It is guaranteed that one of these points is $(0, 0)$. The points are distinct, but it is **not guaranteed** that no three points lie on a same line.

The next line contains an integer Q, the number of queries ($1 \le Q \le 2700$). Then follow Q lines, each contains a single integer q_i ($0 < q_i \le 10^9$). It is guaranteed that the queries are given in strictly ascending order.

Output

Prine Q lines, one for each query. Each line must contain a pair of integers: the coordinates of q_i-th rotation centre.

Example

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

Note

The initial rotation centre (numbered as 0) is $(0, 0)$. The rotation centre number 1 is $(-1, 2)$, because this is the point which the line meets first.

Then the line meets two points $(0, 1)$ and $(1, 0)$ simultaneously, so that there are three points on the line. $(-1, 2)$ is the first of them in sorted order, so the centre has to change to the third point, which is $(1, 0)$. Thus, $(1, 0)$ is the rotation centre number 2.

Then the line meets three points $(-1, 0)$, $(0, 0)$, and $(2, 0)$, so that there are four points on the line. The centre is the third of them in sorted order, so the centre number 3 will be the second of the four points, $(0, 0)$.

Similarly, we can show that the 4-th rotation centre will be $(0, 1)$.

After that, the line meets two points $(-1, 2)$ and $(1, 0)$. The centre is the second point out of three in sorted order, so it does not change.

After that, the line meets $(2, 0)$, so the 5-th rotation centre is $(2, 0)$.

Problem J. Schedule

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 mebibytes

There are N tasks: the i-th task has to start at moment s_i and finish at moment e_i. There is also a potentially infinite supply of machines. We want to assign tasks to machines. Each task will be assigned to one machine. On the other hand, each machine may handle an arbitrary number of tasks as long as no two of them overlap. Tasks i and j are said to overlap if the intersection of the open intervals (s_i, e_i) and (s_j, e_j) is non-empty.

A machine is turned on at the moment when the earliest of its assigned tasks has to start, and turned off at the moment when the latest of them has to finish. The working time of a machine is the length of the time period between these two moments: we cannot turn a single machine on and off more than once.

Your task is to find the minimum possible number of machines K such that we can use only K machines to perform all tasks. Additionally, when using K machines, find the minimum possible sum of all their working times.

Input

The first line of input contains an integer T, the number of test cases ($1 \le T \le 100$).

Each test case begins with a line containing one integer N ($0 < N \le 10^5$). Each of the next N lines contains two integers s_i and e_i ($0 \le s_i < e_i \le 10^9$).

It is guaranteed that $N > 50$ for no more than 10 test cases.

Output

For each test case, print two integers in one line: the minimum possible number of machines K to perform all tasks and the minimum sum of all working times when using K machines.

Example

standard input	standard output
1 3 1 3 4 6 2 5	2 8

Problem K. Two Paths

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

You are given an undirected graph with n nodes (numbered from 1 to n) and m edges. Each edge has a length. The graph contains neither multiple edges nor self-loops.

Alice and Bob are now trying to play a game. Each player has to pick a path from 1 to n (not necessary a simple path). The paths have to be different.

Alice always moves first, and she is so clever that she took one of the shortest paths from 1 to n. Now is Bob's turn. Bob wants to pick the shortest possible path from 1 to n which is different from Alice's path. Your task is to find the length of such path.

Two paths S and T are considered different if and only if they have different number of edges or there is an integer i such that the i-th edge of S differs from the i-th edge of T.

Input

The first line of input contains two integers: the number of nodes n and the number of edges m ($2 \le n \le 10^5$, $1 \le m \le 10^5$). Each of the next m lines contains three integers a, b, and w which mean that there is an edge between node a and node b, and its length is w ($1 \le a, b \le n$, $1 \le w \le 10^9$). It is guaranteed that there is at least one path from 1 to n.

Output

Print a single line with a single integer: the length of a valid shortest path for Bob.

Examples

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