Problem A. Ants

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

Your are given a tree with vertices numbered from 1 to n. There are ants in the vertices of the tree. Initially, at each vertex i, there is one ant having index i.

You will be given q queries. Each query j contains an index a_j of an ant that needs help. During query j, each ant goes to an adjacent vertex that is closest to ant a_j, or does not move if they are located at the same vertex. After each query, print the total number of pairs of ants which are located at the same vertex.

Note that the changes persist between queries: for example, when the ants have to move to ant a_2, they are already in the positions after moving to ant a_1. Also note that each query requires to move to **ant** a_j, which was at vertex a_j initially, but can be in some other vertex at the time of the query.

Input

The first line of input contains an integer n, the size of the tree ($2 \le n \le 10^5$).

Each of the next $n - 1$ lines contains two integers u and v describing an edge of the tree ($1 \le u, v \le n$). It is guaranteed that the edges form a tree.

The next line contains an integer q, the number of queries ($1 \le q \le 10^5$).

The next q lines contain integers $a_1, a_2, \ldots, a_q$, one per line: the numbers of ants in the queries ($1 \le a_j \le n$).

Output

For each query, print a single line with the answer to it: the number of pairs of ants which are located at the same vertex after this query.

Examples

standard input	*standard output*
5 1 2 1 3 2 4 2 5 5 1 1 3 3 5	4 10 10 10 10
8 1 2 1 3 2 4 2 5 3 6 3 7 7 8 6 1 3 4 2 5 6	5 21 28 28 28 28

Problem B. Interesting Subsegments

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

A subsegment (contiguous subarray) of an array is *interesting* if the sum of values on this subsegment is divisible by 3.

You are given two integers n and k. Your goal is to construct the lexicographically minimal array of length n such that it consists only of integers 0, 1, and 2, and has exactly k distinct interesting subsegments.

Array a of length n is lexicographically smaller than array b of the same length if there is $1 \le i \le n$ such that $a_j = b_j$ for $j < i$ and $a_i < b_i$. Two subsegments are distinct if some element of the array belongs to one subsegment but not to the other.

Input

The only line of input contains two integers n and k ($1 \le n \le 10^6$, $0 \le k \le 10^{18}$).

Output

Output `-1` if there is no such array. Otherwise, output the lexicographically smallest array of size n which satisfies the constraints.

Examples

standard input	*standard output*
5 3	0 1 0 1 0
5 5	-1

Problem C. Even Forest

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

An undirected tree is called *even* if there is no simple path of odd length connecting two of its leaves. In particular, a tree with just one vertex is considered even.

You are given an undirected tree G with vertices numbered from 1 to n. A graph obtained by removing some (possibly none) of the edges of G is called a *forest*: it consists of one or more disjoint trees. Determine the minimum possible number k such that we can remove k edges of G in such a way that the resulting forest consists only of even trees.

Input

The first line contains one integer n ($1 \le n \le 10^6$).

Each of the next $n-1$ lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$) denoting an edge connecting vertex u_i and vertex v_i.

The graph is guaranteed to be a tree.

Output

Output the minimum number of edges k such that we can remove k edges of G in such a way that each tree in the resulting forest is even.

Examples

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

Problem D. Yellow Blue Bus

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

You are given n blue points and m yellow points. You need to find a circle such that all blue points are located outside or on the boundary of this circle, and all yellow are located inside or on the boundary of this circle.

The tests for this problem were generated in such a way that, for each test case, there exists a circle with the following properties:

- the radius is $r \le 10^9$,
- the center is at a point (x, y) where $|x| \le 10^9$ and $|y| \le 10^9$,
- for each blue point P, the distance from the center to P is at least $r + 10^{-7} \cdot \max(1, r)$, and
- for each yellow point Q, the distance from the center to Q is at most $r - 10^{-7} \cdot \max(1, r)$.

To check when *your* answer will be considered correct, see output format.

Input

The first line contains t, the number of test cases you need to solve ($1 \le t \le 10^5$).

The first line of each test case contains an integer n, the number of blue points ($1 \le n \le 50\,000$).

Each of the next n lines contains two integers x and y: the coordinates of a blue point ($-10^9 \le x, y \le 10^9$).

The next line contains an integer m, the number of yellow points ($1 \le m \le 50\,000$).

Each of the next m lines contains two integers x and y: the coordinates of a yellow point ($-10^9 \le x, y \le 10^9$).

It is guaranteed that, in each test case, all points (blue **and** yellow) are pairwise distinct. Additionally, the sum of n over all test cases does not exceed $50\,000$, and the sum of m over all test cases does not exceed $50\,000$.

Output

For each test case, output two lines.

On the first line, output a **real** number r, the radius of your circle.

On the second line output two **real** numbers x and y, the coordinates of the center of your circle.

Your answer will be considered correct if:

- $0 \le r \le 10^{10}$,
- $|x| \le 10^{10}$ and $|y| \le 10^{10}$,
- for each blue point P, the distance from center to P is at least $r - 10^{-7} \cdot \max(1, r)$, and
- for each yellow point Q, the distance from center to Q is at most $r + 10^{-7} \cdot \max(1, r)$.

Example

standard input	*standard output*
3 3 3 1 1 3 4 5 3 3 3 4 3 3 4 2 0 0 -1 -1 2 1 1 2 2 4 2 0 0 2 -2 0 0 -2 4 1 0 0 1 -1 0 0 -1	1.001 3 3 1.000 1.5 1.5 1.001 0 0

Problem E. Permutation Matrix

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

You are given a positive integer n. Construct such matrix $2^n \times 2^n$ that:

- The matrix contains distinct positive integers from 1 to 2^{2n}.
- The sums of elements all each submatrices of size $2^{n-1} \times 2^{n-1}$ are equal.

A submatrix is a contiguous rectangle of elements in the original matrix.

Input

The first line contains an integer n ($1 \le n \le 10$).

Output

On the very first line, print "`YES`" if the answer exists, or "`NO`" if not.

If the answer exists, print any such matrix on the next 2^n lines, with each line containing 2^n space-separated integers.

Example

standard input	*standard output*
1	NO

Problem F. Anti-stress

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

A huge number of people are stressed nowadays. Scientists and psychologists try to explain this phenomenon and find ways to resist stress. Businessmen, on the other hand, try to make money. They invent and sell an enormous number of different anti-stress and relaxation toys. You are stressed sometimes, as other people, so you decided to buy one of these toys.

The toy you've bought looks like an infinite bamboo desk with $2n$ pins located on it: n of the pins are blue, the rest are yellow. There are also a red pin and n rubber bands included in the kit. You can drive the red pin wherever you want, and then put the rubber bands over the pins. Nine of ten psychologists assure that you won't be stressed anymore if you put all n rubber bands in such a way that each band is put over one blue pin, one yellow pin and the single red pin in the middle. In other words, each band should be attached to blue and yellow pins by its ends, and then stretched by the red pin. It is also forbidden to attach two different bands to one pin (blue or yellow).

However, the kit you've bought turned out to be defective. All the rubber bands are old, and may tear if you stretch them too much. Imagine you attached a band by its ends to some blue and some yellow pin located at points A and B correspondingly. Let C be the point where you've driven the red pin. Then the band will tear if and only if angle the ACB is acute (less than 90 degrees). Note that you can put the red pin **wherever** you want, even at some point occupied by a blue or a yellow pin. If A or B is equal to C then the angle ACB is assumed to be 180 degrees.

You are very stressed today (aren't you?). That's why you decided to seek help from this toy. Can you drive the red pin somewhere on the desk and then put all n rubber bands so that no band will tear?

Input

The first line of input contains a single integer q, the number of test cases ($1 \le q \le 2 \cdot 10^5$).

Each test case is described as follows. The first line contains a single integer n, the number of pins of each color ($1 \le n \le 2 \cdot 10^5$). Each of the next $2n$ lines contains two integers x_i and y_i: the coordinates of pins on the desk ($-10^6 \le x_i, y_i \le 10^6$). The first n of these lines correspond to blue pins, and the latter n lines to yellow pins.

It is guaranteed that, in each test case, no two pins are located at the same point. It is also guaranteed that the total sum of n over all test cases is less or equal to $2 \cdot 10^5$.

Output

The output for each test case should consist of one or two lines.

If there is no way to put red pin and stretch bands so that no band will tear, print "`impossible`" (without quotes) on a single line.

Otherwise, print two lines for this test case:

- The first line should contain two real numbers X and Y: the chosen coordinates for the red pin ($|X|$ and $|Y|$ should not exceed 10^9).
- The second line should contain n integers m_i. Let us enumerate blue and yellow pins from 1 to n in the order they go in the input. Then m_i means that you decided to stretch the band between i-th blue pin and m_i-th yellow pin. In other words, for each blue pin, you have to print the number of the yellow pin matched to it. Note that m_i should form a permutation of integers from 1 to n.

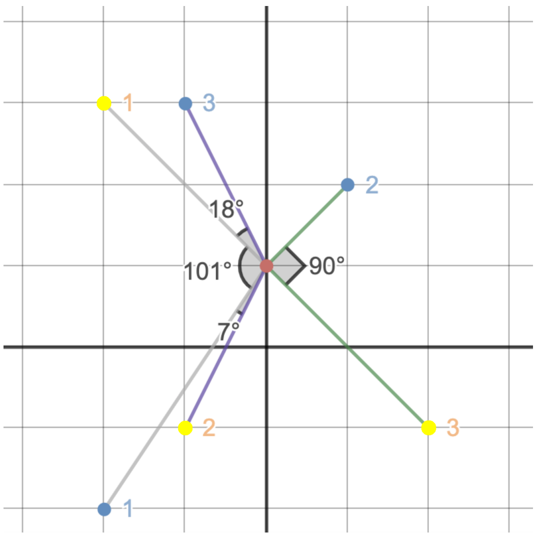
If there are several possible solutions, print any one of them. Here is how the angles are checked:

Let A, B and C be the points with some blue pin, the yellow pin paired to the blue one, and the red pin, correspondingly. Let $r = |AC|^2 + |CB|^2$ and $d = |AB|^2$, where $|AC|$, $|CB|$ and $|AB|$ are the distances between corresponding pins. Then the checker will accept your solution if $r \le d$ with absolute or relative error less than 10^{-6}.

Example

standard input	standard output
3 3 -2 -2 1 2 -1 3 -2 3 -1 -1 2 -1 2 1 2 -1 0 2 -1 0 0 2 3 2 -2 -1 -3 -1 1 -3	0.0 1.0 1 3 2 0.0 0.0 2 1 -0.5 -1.0 1 2

Note



Consider the first test case.

We can choose the red pin to be located at point $(0, 1)$, and pair the points in the following way:

1. First blue and first yellow.
2. Second blue and third yellow.
3. Third blue and second yellow.

All three angles are not acute, thus the pairing is valid.

Problem G. Mismatch

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

You are given an array $a_1, a_2, \ldots, a_n$ of n nonnegative integers. For each k from 1 to n, find the number of subsequences of size k ($a_{i_1}, a_{i_2}, \ldots, a_{i_k}$; $1 \le i_1 < \ldots < i_k \le n$) such that their bitwise AND is equal to zero ($a_{i_1} \wedge a_{i_2} \wedge \ldots \wedge a_{i_k} = 0$). Since the answers can be very large, compute them modulo 998 244 353.

Two subsequences are considered distinct if there is an index i such that the element a_i is included in one of the subsequences but not the other.

Input

The first line contains an integer n ($1 \le n \le 2^{19}$). The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 2^{19}$).

Output

Print n space-separated integers $b_1, b_2, \ldots, b_n$, where b_i is the answer for $k = i$ modulo 998 244 353.

Examples

standard input	*standard output*
3 0 1 2	1 3 1
6 1 2 2 7 6 7	0 3 9 10 5 1

Problem H. Lucky Tickets

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

The government plans a transport reform which will change ticket numbers. The new ticket numbers will consist of q digits in n-ary number system, where q is a prime number. Leading zeros are **allowed**.

A ticket is considered lucky if the product of all its digits added to the sum of all its digits, taken modulo n, equals s. Additionally, every lucky ticket has a degree of luckiness: the degree of luckiness of the ticket with digits $a_1a_2\ldots a_q$ equals

$$(a_1+1)(a_2+2)\ldots(a_q+q)+2^0a_1+2^1a_2+\ldots+2^{q-1}a_q.$$

For reform report, it is needed to calculate the sum of luckiness of all lucky tickets modulo q.

Input

The first line contains three integers n, s and q ($2 \le n \le 10^6$, $0 \le s < n$, $2 \le q \le 10^6$, q is prime).

Output

Print the sum of luckiness of all lucky tickets modulo q.

Examples

standard input	*standard output*
2 0 2	0
10 9 2	1
3 2 3	2

Problem I. Diversity Street

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

A building company wants to build n houses on a new street. Due to requirements of increasing diversity of the city skyline, all houses must have different heights from 1 to n floors. Moreover, there are m additional restrictions on minimum height of buildings on certain intervals of the street. Due to importance of the new street, city administration is ready to cancel one of the restrictions. Determine whether it is possible to create a plan for the new street under the given constraints.

Input

The first line contains two integers: the length of the street n and the number of height restrictions m ($2 \le n \le 3 \cdot 10^5$, $1 \le m \le 3 \cdot 10^5$).

The next m lines contain restrictions. Each restriction is described by three integers h, s, and f which mean that houses from s to f inclusively must have height at least h ($2 \le h \le n$, $1 \le s \le f \le n$).

Output

If it is not possible to create a street plan, print "NO" on a single line.

Otherwise, on the first line, print "YES". On the second line, print a street plan consisting of n house heights. The plan must satisfy all restrictions except possibly one. If there are several possible plans, print any one of them.

Examples

standard input	*standard output*
2 2 2 1 1 2 1 2	YES 2 1
3 2 2 1 2 2 2 3	YES 1 3 2
4 2 4 1 2 3 2 4	NO

Problem J. Disbalance

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

Scientists discovered a new bacteria species that reproduces in a peculiar way. When there's x bacteria in one room, each minute they perform a telepathic communication, upon which one of them is selected to divide. The probability of each particular bacteria being selected is equal to $1/x$.

Scientists became interested in how well this division strategy is balanced. They placed n Petri dishes in a room, each dish with exactly 1 bacteria. After each divide, coefficient d was calculated in the following way. If the number of bacteria in one of the dishes was higher than in all other dishes combined, d was set to the difference between these two quantities. Otherwise, d was set to 0. Formally, if there are $a_1 \geq a_2 \geq \ldots \geq a_n$ bacteria in the dishes, then $d = \max(a_1 - a_2 - \ldots - a_n, 0)$.

Find the expected value of the sum of k numbers: the values of d after the first, second, …, k-th minute of this study. It is possible to write the answer in the form $\frac{p}{q}$, where p and q are relatively prime integers and $q \not\equiv 0 \pmod{998\,244\,353}$. Output such integer r that $r \cdot q \equiv p \pmod{998\,244\,353}$.

Input

The first line contains an integer t, the number of test cases ($1 \leq t \leq 3 \cdot 10^5$).

Each of the following t lines describes one test case and contains two integers n and k ($1 \leq n, k \leq 10^6$).

It is guaranteed that the sum of all n and all k in all test cases is at most $2 \cdot 10^6$.

Output

For each test case, print a single line with a single integer r such that $r \cdot q \equiv p \pmod{998\,244\,353}$, where $\frac{p}{q}$ is the expected value of the sum of k numbers: the values of d after the first, second, …, k-th minute of the study.

Example

standard input	*standard output*
8 1 1 1 2 2 1 2 2 3 1 3 2 3 3 4 3	2 5 1 332748120 0 499122177 299473307 598946612

Problem K. Spiral Matrix

Input file: *standard input*
Output file: *standard output*
Time limit: 4 seconds
Memory limit: 512 mebibytes

Define a *neighbouring* pair of cells in a matrix as a pair of cells (r_a, c_a) and (r_b, c_b) such that:

- either $r_a = r_b$ and $|c_a - c_b| = 1$,
- or $c_a = c_b$ and $|r_a - r_b| = 1$.

Define a *spiral matrix* as a matrix which satisfies the following conditions:

- The matrix contains only distinct positive integers.
- One can start from some cell (i, j) and arrange all other cells in a path, so that every two consecutive cells in the path are a neighbouring pair, and by following the path from (i, j) and considering the values in the matrix, we form a continuous integer interval $[l..r]$ in the order of visiting them.

Given is a matrix of size $n \times m$ consisting of distinct positive integers. We are also given q queries. Each query defines a submatrix with corners (r_1, c_1) and (r_2, c_2). For each query, determine whether this submatrix is spiral.

Input

The first line contains three integers n, m and q ($1 \le n, m \le 2000$, $1 \le q \le 10^6$), denoting the size of the matrix and the number of queries, respectively.

Each of the next n lines contains m integers. The j-th integer on the i-th of these lines denotes the element $a_{i,j}$ located in the i-th row and j-th column of the matrix ($1 \le a_{i,j} \le 10^9$). It is guaranteed that all elements are distinct.

Each of the next q lines contains four integers r_1, c_1, r_2, c_2 ($1 \le r_1 \le r_2 \le n$, $1 \le c_1 \le c_2 \le m$), denoting the corners of a submatrix.

Output

For each query, print the answer on a separate line. Print "`YES`" if the submatrix is spiral, or "`NO`" otherwise.

Example

standard input	*standard output*
5 7 10	NO
10 11 12 13 14 15 16	YES
9 2 3 32 31 30 17	NO
8 1 4 25 26 29 18	YES
7 6 5 24 27 28 19	YES
52 51 50 23 22 21 20	NO
1 1 5 7	YES
1 1 4 1	YES
2 2 5 3	YES
1 4 5 7	NO
1 1 4 3	
1 1 5 3	
2 2 2 2	
2 2 2 3	
3 4 5 7	
3 3 4 4	