# The 2nd Universal Cup

Uni
Cup

## Stage 18: Dolgoprudny

January 13-14, 2024

This problem set should contain 10 problems on 13 numbered pages.

**Based on**

45th Petrozavodsk Programming Camp
Summer 2023

Petrozavodsk Programming Camp

# Problem A. Anti-Plagiarism

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

As a homework, the teacher asked all the students of the art class to draw a beautiful, and most importantly original, tree. After everyone has submitted their work, the teacher began to suspect some students of cheating.

The teacher considers a tree $T_1$ to be copied from a tree $T_2$ if it is possible to add some (possibly zero) vertices and edges to $T_2$ and relabel its vertices so that it becomes the same as $T_1$.

In total, she suspects $t$ pairs of students. For each given pair of trees, check if first tree could be copied from the second tree.

## Input

The first line contains an integer $t$ ($1 \le t \le 10^4$): the number of suspicious pairs of students.

After that, there are $t$ descriptions of pairs of trees.

The first line of each description contains an integer $n$ ($2 \le n \le 10^5$). Each of the next $n-1$ lines contains two integers $u$ and $v$ ($1 \le u, v \le n$): the edges of the first student's tree.

The next line of each description contains an integer $m$ ($2 \le m \le n$). Each of the next $m-1$ lines contains two integers $u$ and $v$ ($1 \le u, v \le m$): the edges of the second student's tree.

It is guaranteed that the sum of $n$ over all pairs of students does not exceed $5 \cdot 10^5$, and the sum of $n \cdot m$ does not exceed $10^7$.

## Output

For each of the $t$ pairs of trees, print a line containing a single word (case-insensitive): "Yes" if the first tree could be copied from the second tree, or "No" otherwise.

## Example

| *standard input* | *standard output* |
|---|---|
| 2<br>5<br>1 2<br>1 5<br>2 3<br>2 4<br>4<br>1 2<br>1 3<br>1 4<br>6<br>1 2<br>1 3<br>1 4<br>5 1<br>6 1<br>4<br>1 2<br>2 3<br>3 4 | Yes<br>No |

## Problem B. Bit Component

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

Denis really likes binary representation of numbers. Once he wrote down binary representations of numbers from 1 to 7, in some order, one under the other, so that their rightmost digits were aligned. Then he realized that the ones in these numbers form a single connected region: if we consider the places for digits as squares on a grid, all squares containing ones are connected by sides. Now he wonders if the same thing can be done for numbers from 1 to $n$ for different $n$.

You are given an integer $n$. You should find a good permutation of all integers from 1 to $n$, or determine that there is no such permutation. A permutation is *good* if, when we write the numbers down as described above, the ones form a single connected region.

### Input

The only line contains a single integer $n$ ($1 \le n \le 2 \cdot 10^5$).

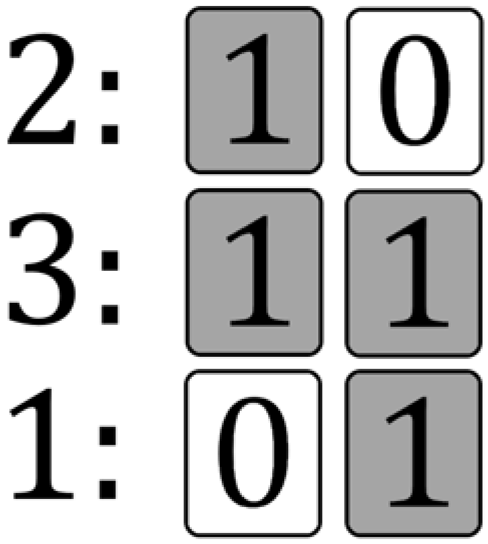
### Output

If there is no good permutation of numbers from 1 to $n$, print a single line with the word "NO" (uppercase). Otherwise, print a line with the word "YES" (uppercase), and then another line containing the good permutation you found. If there are several possible answers, print any one of them.

### Examples

| *standard input* | *standard output* |
|---|---|
| 1 | YES<br>1 |
| 2 | NO |
| 3 | YES<br>2 3 1 |

### Note



Third example

# Problem C. Crossing the Border

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 5 seconds |
| Memory limit: | 512 mebibytes |

Kostya F. is crossing the border of a certain country, carrying $n$ taxable items with him. Each item is characterized by two integers $w_i$ and $c_i$: the weight of the item and the amount of tax that must be paid for transporting this item across the border.

Kostya needs to distribute all his items among several knapsacks. He can use any number of knapsacks. According to the airline's rules, the weight of each knapsack is limited, so the total weight of items inside each knapsack cannot exceed $W$.

There are special rules for customs fees. When customs officers are checking the luggage, they open each knapsack, and set the tax for it equal to the **maximum** tax rate among the items inside this knapsack. The total tax is the sum of individual taxes for all knapsacks.

For purely practical reasons, Kostya wants to know what is the minimum total tax he can pay in order to cross the border with all his items. Also, out of pure curiosity, he wants to know in how many different ways this minimum can be achieved. Help him. Since the number of ways can be very large, you should find it modulo 998 244 353.

Two ways to put items into knapsacks are considered the same if there is a bijection between knapsacks such that the corresponding knapsacks have exactly the same sets of items.

## Input

The first line contains two integers: the number of items $n$ and the maximum weight of one knapsack $W$ ($1 \le n \le 22$; $1 \le W \le 5 \cdot 10^7$).

The next $n$ lines describe the items. The $i$-th of them contains two integers: the weight $w_i$ and tax $c_i$ for item $i$ ($1 \le w_i \le W$; $1 \le c_i \le 5 \cdot 10^7$).

## Output

Print a line with two integers. The first must be the minimum total tax Kostya can pay. The second must be the number of ways to achieve that minimum, taken modulo 998 244 353.

## Example

| *standard input* | *standard output* |
|---|---|
| 5 5<br>3 5<br>1 4<br>2 3<br>2 2<br>2 1 | 9 4 |

## Note

In the example, there are 4 different ways to distribute items among knapsacks with total tax equal to 9 (items are numbered from 1 to 5):

- $[1,3]$, $[2,4,5]$: tax $5+4=9$
- $[1,4]$, $[2,3,5]$: tax $5+4=9$
- $[1,5]$, $[2,3,4]$: tax $5+4=9$
- $[1,2]$, $[3,4]$, $[5]$: tax $5+3+1=9$

# Problem D. Dinosaur Bones Digging

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

Paleontologists are looking for dinosaur bones! They have already found a long line of $n$ square sectors, and numbered them by integers from 1 to $n$. Each square has a side of 1 meter. Preliminary measurements showed that, in sector $i$, the depth of the soil potentially containing dinosaur bones is $a_i$ meters. Below that depth lies solid bedrock. All the numbers $a_i$ turned out to be different integers.

Scientists have prepared $q$ different plans for their research. Each plan includes the construction of a research station on a subsegment of sectors numbered from $\ell_j$ to $r_j$. After picking a subsegment, they will pick one of its sectors $m$ ($\ell_j \le m \le r_j$) as the main sector.

A special device will be buried in the main sector at the depth of $a_m$ meters. This device allows the researchers to analyze the top $a_m$ meters of all the sectors under the research station that have depth **strictly greater** than $a_m$. In total, $a_m \cdot k$ cubic meters of soil will be analyzed, where $k$ is the number of sectors under the station (that is, between $\ell_j$ and $r_j$, inclusive) which are deeper than the main sector.

Paleontologists want to find as much dinosaur bones as possible, so they want to analyze as much soil as possible. Help them! Find the maximum volume of soil which can be analyzed if a subsegment is chosen from the plans, and its main sector is then chosen optimally.

## Input

The first line contains a single integer $n$, the number of sectors ($1 \le n \le 10^6$).

The second line contains $n$ distinct integers $a_1, a_2, \ldots, a_n$, the depths of sectors ($1 \le a_i \le 10^9$).

The next line contains a single integer $q$, the number of plans ($1 \le q \le 10^6$).

Each of the next $q$ lines describes a plan. The $j$-th of them contains two integers $\ell_j$ and $r_j$ which are the endpoints of the subsegment for the $j$-th plan ($1 \le \ell_j \le r_j \le n$).

## Output

Print a line with a single integer: the maximum volume of analyzed soil in cubic meters.

## Example

| *standard input* | *standard output* |
|---|---|
| 6<br>3 5 2 7 4 6<br>2<br>1 5<br>3 6 | 9 |

## Note

In the example, scientists should pick the first plan and the first sector as its main sector. Then $3 \cdot 3 = 9$ (since 5, 7, 4 are larger than 3) cubic meters of soil will be analyzed.

# Problem E. Egg Drop Challenge

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

Annual *Egg Drop Challenge* is about to start! There are $n$ people participating in the challenge. They are standing on different floors of the same building, and the $i$-th of them stands on the $i$-th floor, at a height of $h_i$ above the ground. The person on the $n$-th floor is holding an egg. The task of the participants is to safely lower this egg to the first floor as quickly as possible.

When person $i$ holds an egg in their hands, they can throw it down with any real-valued speed from 0 to $v_i$.

When the egg flies past person $i$, that is, when it is at a height of $h_i$, they can (but are not obliged to) catch it if it flies at a speed of at most $u_i$. Catching the egg means completely stopping it. The egg is considered to be safely lowered to the first floor when it was caught by person 1.

When the egg is falling, it falls with the acceleration of free fall $g$. For simplicity of calculations, we will assume that there is no air resistance, and we will also assume that $g = 1$. In the Note section below, you can find equations that describe the motion of the egg under such assumptions. Catching and throwing are assumed to happen instantly.

Help the participants find the shortest time possible to safely lower the egg to the first floor, or tell them that it is impossible to do so.

## Input

The first line contains a single integer $n$, the number of people participating in the challenge ($2 \le n \le 3 \cdot 10^5$).

The next $n$ lines describe the people participating in the challenge. The $i$-th such line describes person $i$ and contains three integers: $h_i$, $v_i$, and $u_i$ ($1 \le h_i \le 10^{18}$; $1 \le v_i, u_i \le 10^9$): the height of person $i$ above the ground and the maximum speed with which they can throw and catch an egg, respectively.

It is guaranteed that $h_i < h_{i+1}$ for all $1 \le i \le n - 1$.

## Output

If it is possible to lower an egg to the first floor by satisfying all the constraints, print a line with one real number: the minimum time it takes to do this. The answer will be considered correct if the absolute or relative error does not exceed $10^{-6}$.

Otherwise, print a line with the number $-1$.

## Examples

| *standard input* | *standard output* |
|---|---|
| 5<br>2 1 7<br>14 6 4<br>18 1 7<br>21 2 5<br>28 4 10 | 6.00000000000000000000 |
| 2<br>1 1 4<br>10 5 1 | -1 |

## Note

If the egg starts flying at a speed of $v$, then after flying for $t$ seconds, it will move at a speed of $v + g \cdot t$, or just $v + t$ in our case, and will cover the total distance of $v \cdot t + g \cdot \frac{t^2}{2}$, or just $v \cdot t + \frac{t^2}{2}$ in our case.

In the first example, the optimal solution looks as follows:

- Person 5 throws the egg at a speed of 4. After flying for 2 seconds, it will cover $4 \cdot 2 + \frac{2^2}{2} = 10$ meters and will be moving at a speed of 6, so person 3 will be able to catch it.
- Person 3 throws the egg at a speed of 1, and after 2 seconds, person 2 will catch it.
- Person 2 throws the egg at a speed of 5, and after 2 more seconds, person 1 will catch it, completing the challenge in 6 seconds.

In the second example, even if person 2 throws the egg at a speed of 0, it will still move faster than 4 when it reaches the first floor. So, safe lowering is impossible.

# Problem F. Fortune Wheel

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

A *Fortune Wheel* has $n$ sectors numbered from 0 to $n-1$ in clockwise order. It also has an arrow pointing at one of the sectors. Right now, it is pointing at sector $x$.

You are very good at spinning the Wheel. More specifically, you have learned $K$ distinct power spins, characterized by their power $k_1, k_2, \ldots, k_K$. A *power spin* with power $p$ means that you spin the Wheel with such power that the arrow would turn exactly $p$ sectors clockwise: formally, from sector $y$, it would turn to sector $(y+p) \bmod n$. Also, you can do a common spin: spin the Wheel so that the arrow would be pointing at a uniformly random sector. Your skills allow you to do any number of spins any number of times in any order.

You want the arrow to be pointing at sector 0 as soon as possible. Find the expected value of the number of spins required to do so in an optimal strategy. A strategy is considered optimal if it minimizes the said expected value.

## Input

The first line contains three integers: the number of sectors $n$, the starting sector of the arrow $x$, and the number of power spins $K$ ($1 \le n \le 10^5$; $0 \le x \le n-1$; $1 \le K \le 500$).

The second line contains $k$ distinct integers $k_1, k_2, \ldots, k_K$ ($1 \le k_i \le n$).

## Output

Print a line containing two integers $p$ and $q$ ($0 \le p$; $0 < q$): numerator and denominator of an irreducible fraction $p/q$ which is the expected value of the number of spins. It can be proved that the answer can be represented in this way.

## Examples

| *standard input* | *standard output* |
|---|---|
| 6 3 2<br>2 4 | 8 3 |
| 5 4 1<br>1 | 1 1 |

## Problem G. Growing Sequences

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

In scientific research, exponentially growing sequences appear quite often. Some researches are especially interested in integer arrays of length $n$ where each element is at least twice as large as the previous one: formally, $2 \cdot a_i \le a_{i+1}$ for $1 \le i \le n-1$. They want to calculate the number of different bounded arrays satisfying this condition.

Help them! Count the number of such arrays consisting of integers from 1 to $c$. Since this number can be very large, you should output it modulo 998 244 353.

### Input

The only line contains two integers $n$ and $c$ ($1 \le n \le 60$; $1 \le c \le 10^{18}$): the length of the arrays and the maximum value of their elements.

### Output

Output the number of different arrays modulo 998 244 353.

### Examples

| *standard input* | *standard output* |
|---|---|
| `1 5` | `5` |
| `3 6` | `4` |
| `15 179` | `0` |
| `35 1234567887654321` | `576695683` |

### Note

In the first example, there are 5 different arrays: [1], [2], [3], [4], [5].

In the second example, there are 4 different arrays: $[1, 2, 4]$, $[1, 2, 5]$, $[1, 2, 6]$, $[1, 3, 6]$.

In the third example, there are no arrays satisfying the conditions.

# Problem H. Hierarchies of Judges

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

In Treeland, a judicial reform is being carried out: the government needs to choose a new judicial hierarchy. The hierarchy will consist of $n$ judges conveniently labeled by integers from 1 to $n$.

The judicial hierarchy is a rooted labeled tree, and its vertices are the judges. Each judge has an ordered list of direct subordinates: child nodes in the rooted tree. Additionally, each judge in the judicial hierarchy is either reliable or unreliable.

The judicial hierarchy is called *fair* if the following is true for each judge: among him and all of his direct subordinates, at least half are reliable.

Two judicial hierarchies are considered different if at least one of three conditions holds true:

- some judge is reliable in one hierarchy, and unreliable in the other;
- for some judge, the set of his direct subordinates in one hierarchy differs from the set of his direct subordinates in the other;
- for some judge, the relative order of his unreliable subordinates in one hierarchy is different from the relative order of his unreliable subordinates in another hierarchy.

You can check the notes section to better understand these conditions.

Government wants to choose a fair judicial hierarchy by going through all the possible options. In order to assess the scale of the work, they need to know how many different fair judicial hierarchies exist. Calculate this number modulo 998 244 353.

## Input

The first line contains a single integer $n$ ($1 \le n \le 2 \cdot 10^5$): the number of judges in the hierarchy.

## Output

Print a single integer: the number of different fair judicial hierarchies with $n$ judges modulo 998 244 353.

## Examples

| *standard input* | *standard output* |
|---|---|
| 1 | 1 |
| 3 | 24 |
| 5 | 3190 |
| 100 | 413875584 |

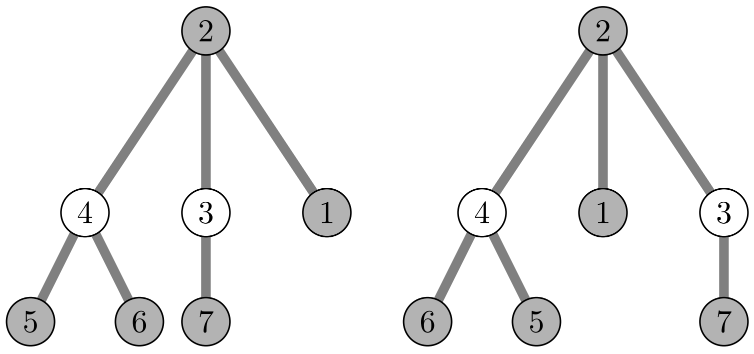
## Note

Below are some examples of judicial hierarchies to clarify the conditions.

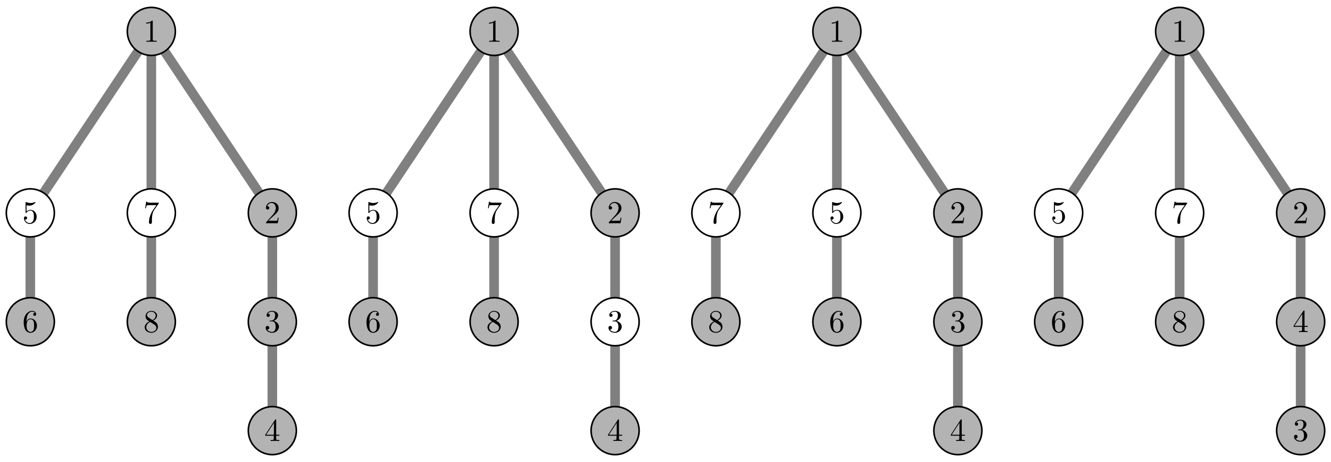
The direct subordinates of each judge are placed below the judge, from left to right.

Reliable judges are gray, and unreliable judges are white.

These two fair hierarchies are considered the same:



These four are all different:



## Problem I. Institute

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

Tikhon passed the entrance exams, and now studies at a research institute. However, he is afraid to walk around its campus. He is worried that he may forget his pass somewhere and permanently lose it, and then he will not be able to attend classes.

The campus of Tikhon's institute is a directed graph. Some of the edges of this graph can only be traversed with a pass.

To lose the pass permanently, Tikhon would need to start at his dormitory, which is located at the first vertex, then walk along zero or more edges of the campus, then leave his pass in some vertex, continue walking around the campus, and then not be able to return to the vertex where he has left the pass.

Tikhon would like to know if he is worried for no reason: please help him find out if it is possible to permanently lose the pass in the campus.

### Input

The first line contains two integers $n$ and $m$ ($1 \le n, m \le 3 \cdot 10^5$): the number of vertices and edges in the graph, respectively.

Each of the following $m$ lines contains three integers $u_i$, $v_i$, $t_i$ ($1 \le u_i, v_i \le n$; $1 \le t_i \le 2$) describing the directed edges of the graph. Edge $i$ allows passage from $u_i$ to $v_i$. If a pass is required to go through edge $i$, then $t_i = 1$, otherwise $t_i = 2$.

The given graph may contain loops and multiple edges between the same vertices.

### Output

Print a line with a single word (case-insensitive): "Yes" if Tikhon can permanently lose his pass, or "No" otherwise.

### Examples

| *standard input* | *standard output* |
|---|---|
| 3 4<br>1 2 1<br>2 3 2<br>3 2 1<br>3 1 2 | Yes |
| 6 8<br>1 2 1<br>2 3 2<br>3 2 2<br>3 4 1<br>4 1 2<br>1 5 2<br>5 4 2<br>6 1 2 | No |

### Note

In the first example, Tikhon can first traverse the edge $1 \to 2$ with a pass, then, leaving a pass at vertex 2, traverse the edge $2 \to 3$. After that, he will not be able to return to vertex 2.

# Problem J. Jumping Lights

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

You are given a tree: an undirected connected graph on $n$ vertices with $n-1$ edges. Initially, none of the vertices are marked. Your task is to process $q$ queries of the following three types:

- "`0` $w$": unmark vertex $w$; if $w$ is not marked, nothing happens.
- "`1` $w$": mark vertex $w$; if $w$ is marked, nothing happens.
- "2": simultaneously for all vertices in the tree: mark the vertex if it has at least one marked neighbor, otherwise unmark it.

After each query, find how many marked vertices are there in the tree.

## Input

The first line contains two integers $n$ and $q$ ($2 \le n \le 3 \cdot 10^5$; $1 \le q \le 10^6$): the number of vertices in the tree and the number of queries, respectively.

Each of the next $n-1$ lines describes an edge of the tree by two integers $u$ and $v$ ($1 \le u, v \le n$).

Each of the next $q$ lines represents a query in the format shown above ($1 \le w \le n$).

## Output

Print a single line with $q$ integers: the number of marked vertices in the tree after each query.

## Examples

| *standard input* | *standard output* |
|---|---|
| `8 8`<br>`1 2`<br>`2 3`<br>`2 4`<br>`1 5`<br>`5 6`<br>`5 7`<br>`5 8`<br>`1 1`<br>`2`<br>`2`<br>`0 1`<br>`0 3`<br>`0 4`<br>`0 5`<br>`2` | `1 2 6 5 4 3 3 1` |
| `4 5`<br>`1 2`<br>`1 3`<br>`2 4`<br>`1 2`<br>`2`<br>`0 4`<br>`2`<br>`2` | `1 2 1 2 2` |