UCPC 2025

Union of Clubs for Programming Contests
Summer Contest 2025

Preliminaries

Official Problemset

July 12th, 2025, 14:00 – 17:00

Hosted and Organized by

Union of Clubs for Programming Contests

Sponsored by

나정휘 (jhnah917)

정재헌(gravekper)

UCPC 2025

Preliminaries

Problems

This problem set contains 11 problems.

A Physical Education is a Math Subject 2

B Domino Game

C Beam

D Adding Operations

E Concert

F Market-Making

G Smart Warehouse

H Coastline

I Robotic Vaccum

J XOR Graph

K Distance Multiplication Maximization

The memory limit for all problems is 1GB.

Problem A. Physical Education is a Math Subject 2

Time Limit 1 second Memory Limit 1024 MB

It's time for PE class at UCPC Elementary School. Today, there is a five-lap running assessment around the field.

Wookyung was not used to running that much, so he was filled with worries. To encourage him, the PE teacher smiled at Wookyung and said, "Don't worry too much. You may run slowly. You only need to complete 5 laps in at most 1800 seconds."

Thanks to the coach's encouragement, Wookyung gained courage and slowly began running. He started to get tired, but he pushed forward without giving up and eventually managed to complete 4 laps.

Finally, the last lap. While catching his breath for a short bit, Wookyung started to get worried. "Did I run too slowly...? Can I still pass the assessment if I complete this last lap well enough...?" Being so exhausted, Wookyung felt it would take more than 300 seconds to finish his last lap, but he made up his mind and decided to complete it exactly in 300 seconds.

Given Wookyung's records of the 4 laps he has already completed, find out whether he can pass the assessment if he completes his last lap in exactly 300 seconds.

Input

Four lines of input are given. Each line contains t_i, denoting the time Wookyung took to complete each lap, measured in seconds. $(1 \le t_i \le 1800)$

Output

Print `Yes` if Wookyung can pass the assessment, and `No` otherwise.

Examples

standard input	standard output
375 375 375 375	Yes
375 375 375 400	No
300 300 300 300	Yes

Problem B. Domino Game

Time Limit 1 second Memory Limit 1024 MB

There is a grid of size $N \times M$. The cell at row i, column j contains a non-negative integer a_{ij}. $(1 \le i \le N;\ 1 \le j \le M)$

Each turn, Minji performs the following actions.

1. Minji selects two horizontally or vertically adjacent grid cells. At least one of the two cells must contain a positive integer.
2. Minji decreases the number in each selected cell by 1. However, if some cell contains 0, she does not decrease that number.

The game ends when every cell contains 0. Minji wishes to play this game as long as she can. Find out the maximum number of turns the game can last when Minji does her best.

Input

The first line of input contains two space-separated positive integers N and M. $(2 \le N, M \le 1\,000)$

Then, N lines follow. The i-th of them contains M space-separated integers $a_{i1}, a_{i2}, \cdots, a_{iM}$. $(0 \le a_{ij} \le 10^9)$

Output

In the first line, print the maximum number of turns the game can last.

Examples

standard input	standard output
2 2 0 1 0 0	1
2 2 1 0 1 1	3

Problem C. Beam

Time Limit 2 seconds Memory Limit 1024 MB

You are running an interval storage service. The storehouse currently contains N intervals, where the i-th interval is represented as $[l_i, r_i]$ on the number line.

Q upcoming laser strikes on the storehouse have been announced! The damage range of the j-th laser can be represented as the interval $[s_j, e_j]$. Each time a laser strike occurs, your job is to move the stored intervals such that no interval is hit by the laser.

Specifically, for each stored interval $[l_i, r_i]$, a suitable integer x_{ij} is selected. Then, for all (i, j), the length of the intersection between $[l_i+x_{ij}, r_i+x_{ij}]$ and $[s_j, e_j]$ must be 0. Here, the length of the intersection between two intervals $[a, b]$ and $[c, d]$ is defined as $\max(0, \min(b, d) - \max(a, c))$.

Intervals are heavy and require special machines when moving them. Moving an interval costs an amount of (Distance moved) × (Length of interval) in electricity bills. Thus, the fee to be paid before the j-th laser beam strikes is $\sum_{i=1}^{N}(r_i - l_i)|x_{ij}|$.

Note that all intervals are moved back to their initial position after each laser strike, and this also costs electricity.

Your task is to pay the minimum amount in electricity bills while keeping all stored intervals safe. Calculate the cost required to do so.

Input

The first line of input contains N, denoting the number of intervals, and Q, denoting the number of laser strikes. $(1 \le N, Q \le 250\,000)$

The i-th of the following N lines contains l_i and r_i, denoting the endpoints of the i-th interval. $(1 \le l_i < r_i \le 1\,000\,000)$

Then, the j-th of the following Q lines contains s_j and e_j, denoting the endpoints of the range damaged by the j-th laser strike. $(1 \le s_j < e_j \le 1\,000\,000)$

All input values are integers.

Output

Print Q lines in total: the minimum amount to pay in electricity bills when moving all intervals to a safe position and back to the original state, for each laser strike in a single line.

Example

standard input	standard output
2 2 1 5 4 8 3 5 8 9	24 0

Problem D. Adding Operations

Time Limit 1 second Memory Limit 1024 MB

FuriosaAI develops AI semiconductors for data centers and high-performance edge computing. RNGD is FuriosaAI's second-generation AI accelerator based on a proprietary architecture called TCP(Tensor Contraction Processor), and provides an SDK compatible with major frameworks such as PyTorch.

For RNGD's AI model to carry out an inference task, the model first needs to be **compiled** into a sequence of RNGD operations. Let's assume the following situation, which is a simplification of the actual compilation process.

The basic unit of time for an RNGD operation is a **cycle**, and the model's execution needs to be completed within H cycles. Let the H cycles be labeled cycle 1, cycle 2, $\cdots$, cycle H sequentially.

There are N operations in the compiled AI model. The execution of the i-th operation starts at cycle A_i and ends at cycle B_i. Note that there may be operations whose cycles overlap.

While executing the compiled model, RNGD has been requested to execute one additional operation. The additional operation must not have overlapping cycles with any of the N other operations and must be executed within the H cycles. Specifically,

- Let T be the number of cycles the additional operation takes to complete. In other words, if the additional operation begins executing at cycle S, it terminates at cycle $S+T-1$.
- While the additional operation is being executed, no other operations should be executed.
- $1 \le S \le S+T-1 \le H$ should hold.

The value of T may differ based on the type of operation to be added. To test various cases, we will consider Q different values of T. Given T_i, denoting the number of cycles required for the additional operation at the i-th case, write a program that calculates the number of possible starting cycles S quickly.

Note that each case is handled independently. In other words, each of the additional operations is only effective for its corresponding case.

Input

The first line of input contains two space-separated integers N, denoting the number of operations of the AI model, and H, denoting the total number of cycles. $(0 \le N \le 200\,000;\ 1 \le H \le 10^9)$

The i-th of the following N lines contains two space-separated integers A_i and B_i, denoting the beginning and ending cycles of the i-th operation. $(1 \le A_i \le B_i \le H)$

The next line contains Q, denoting the number of cases to be tested. $(1 \le Q \le 200\,000)$

Each of the following Q lines contains T_i, denoting the number of cycles required for the additional operation. $(1 \le T_i \le H)$

Output

Print Q lines in total; the answer for each case in a single line.

Examples

standard input	standard output
3 30 5 8 12 19 14 23 3 2 4 6	11 5 2
0 4 2 1 2	4 3

Problem E. Concert

Time Limit 1 second Memory Limit 1024 MB

In the city where Hyunbin lives, a concert is held every day.

Hyunbin hates loud noises, so he installed N noise barriers, which are sequentially numbered from 1 to N. The i-th noise barrier can absorb D_i units of noise. After installing the noise barriers, he made a rule that all future concerts can only be held between two neighboring noise barriers. In other words, all future concerts can only be held between noise barrier c and noise barrier $c+1$ for some integer c. $(1 \le c < N)$

Unlike Hyunbin's plan to absorb all the noise and enjoy the quiet days, sometimes noise barriers were unable to withstand the whole noise from a concert. Suppose a concert is held between noise barriers c and $c+1$, and emits x units of noise. Noise barrier c only absorbs $\min(D_c, x)$ units of noise, and if there is some noise left unabsorbed, it is spread to barrier $c-1$. Likewise, noise barrier $c+1$ only absorbs $\min(D_{c+1}, x)$ units of noise, and the rest of the noise spreads to barrier $c+2$. This procedure is repeated until there is no noise left, or some noise is unabsorbed, but there is no noise barrier to absorb it.

After each concert, Hyunbin has decided to enhance each of the N noise barriers' absorption capacity by the amount of noise it has absorbed. Thus, if the noise barrier k has absorbed x units of noise, its absorption capacity becomes $D_k + x$ right after the concert.

Hyunbin performs one of the two following operations Q times.

- $1\ c\ x$: A concert which emits x units of noise is held between noise barriers c and $c+1$, and after that, the noise barriers are enhanced subsequently. $(1 \le c < N;\ 1 \le x \le 10^9)$
- $2\ c$: Hyunbin measures the absorption capacity of barrier c. $(1 \le c \le N)$

Print the absorption capacity of the noise barrier each time an operation of type 2 is performed.

Input

The first line of input contains an integer N, denoting the number of noise barriers Hyunbin has installed. $(2 \le N \le 200\,000)$

The second line of input contains N space-separated integers, each denoting D_i, the absorption capacity of each noise barrier. $(1 \le D_i \le 10^9)$

The third line of input contains Q, denoting the number of operations Hyunbin has carried out. $(1 \le Q \le 200\,000)$

Each of the following Q lines contains an operation in the same format as written in the problem statement. It is guaranteed that at least one type 2 operation will be given in the input.

All input values are integers.

Output

Print the results of all type 2 operations carried out by Hyunbin, in order, one per line.

Example

standard input	standard output
6 5 1 2 4 7 3 5 1 2 1 2 3 1 4 7 2 3 2 5	3 6 14

Problem F. Market-Making

Time Limit 4 seconds Memory Limit 1024 MB

HRT is first and foremost a math and technology company. We are engineers and researchers working as one team to solve difficult problems, and trading millions of shares a day on the world's financial markets.

A new employee at HRT has developed a market-making engine based on the newly designed trading scheme. The developer wishes to analyze the stability of the new engine. The engine uses a virtual account (initially holding no shares) for trading and will follow the strategy below for N contiguous transaction periods (ticks).

- At the i-th transaction tick, the engine selects an integer between a_i and b_i, inclusive. If the selected integer is positive, it buys an equal number of shares as the integer. If it is negative, the engine sells an equal number of shares as the absolute value of the integer. If it is 0, the engine does not make a transaction. Shares can be sold even if the engine does not hold a sufficient amount, and in this scenario, the number of shares held becomes negative.
- If the number of shares held right after the i-th transaction is 0, the trading scheme is considered to have minimized position exposure and thus contributed to market stability. In this case, the stability of the engine increases by x_i.
- All other factors, such as fees or slippage, are ignored.

Find the maximum stability the new market-making engine can achieve after all N transactions have been completed.

Input

The first line of input contains the number of transactions N. $(1 \le N \le 1\,000\,000)$

The i-th of the following N lines contains three space-separated integers a_i, b_i, x_i, denoting the information about the i-th transaction. $(-10^9 \le a_i \le b_i \le 10^9;\ 1 \le x_i \le 10^9)$

Output

Print the maximum stability the new market-making engine can achieve after all N transactions have been completed.

Examples

standard input	standard output
3 -1 0 3 1 1 2 -1 0 5	8
5 1 1 1000 -2 -1 7 1 1 5 -1 -1 4 1 1 8	13
8 -1 1 5 -4 2 7 3 4 4 -6 4 8 -2 -1 6 -5 7 1 4 6 9 -7 7 5	34

Problem G. Smart Warehouse

Time Limit 3 seconds Memory Limit 1024 MB

Hyundai AutoEver is a company leading the future of mobility innovation, transforming the paradigm of smart logistics in the Fourth Industrial Revolution.

Hyundai AutoEver is testing a smart warehouse in which items are placed in the form of an $N \times M$ grid. Let's denote the cell in the r-th row from the top and the c-th column from the left as (r, c). Each cell (r, c) has an item whose value is A_{rc}. The value of an item can be negative.

A robot in the warehouse is sometimes instructed to transport an item outside. For efficient transportation, the robot can also transport nearby items together as a rectangular group. Specifically, suppose the robot is instructed to transport an item at cell (r, c). In that case, the robot selects four integers r_1, r_2, c_1, c_2 such that $1 \le r_1 \le r \le r_2 \le N$, $1 \le c_1 \le c \le c_2 \le M$, and then transports all items inside the rectangular area $[r_1, r_2] \times [c_1, c_2]$.

For each of the NM items, calculate the maximum sum of values of the items that can be transported when the robot is instructed to transport the item.

Input

The first line of input contains two space-separated integers N and M, denoting the size of the warehouse. $(2 \le N \le 500;\ 2 \le M \le 500)$

The r-th of the following N lines contains M space-separated integers $A_{r1}, A_{r2}, \cdots, A_{rM}$, which correspond to the value of each item in the r-th row of the grid. $(-10^3 \le A_{rc} \le 10^3)$

Output

Print N lines in total. In the r-th line of output, print M integers, which denote the maximum sum of values of the items that can be transported when the robot is instructed to transport the item at each of the cells $(r, 1)$, $(r, 2)$, $\cdots$, (r, M).

Examples

standard input	standard output
3 4 2 2 2 2 2 -8 0 -12 2 -8 7 5	8 8 9 8 6 1 9 4 6 6 12 12
2 2 -1 -3 2 -1	1 -3 2 1

Problem H. Coastline

Time Limit 1 second Memory Limit 1024 MB

There are N cities on a circular coastline. The cities are numbered from 1 to N in clockwise order. For all pairs of cities, there exists a bidirectional road connecting the two cities.

Among all these roads, we call the road connecting two cities a and b the **special road**. The special road connects two cities, neither of which is city 1, and is known to be the road with the most beautiful scenery.

Jeongseo plans to travel through the cities following the method below, only by road.

- His travel begins at city 1. City 1 is considered to be already visited at the moment he begins his travel.
- During his travel, he should not visit a city that he has already visited, including city 1, and every city must be visited exactly once.
- None of the roads used in the travel course may intersect with each other.
- The **special road** must be used exactly once during his travel. It does not matter in which direction he travels through the special road.

Find the number of travel paths that satisfy all of the conditions above.

Input

The first line of input contains a positive integer N, denoting the number of cities. $(3 \le N \le 1\,000\,000)$

The second line of input contains two space-separated integers a and b, denoting the two cities connected by the special road. $(2 \le a, b \le N;\ a \ne b)$

Output

Print the total number of paths that satisfy the conditions given in the problem statement modulo $1\,000\,000\,007$.

Examples

standard input	standard output
4 2 4	2
6 5 4	11

Problem I. Robotic Vaccum

Time Limit 1 second Memory Limit 1024 MB

LG ThinQ is LG Electronics' AI platform, which monitors home appliances in real time and provides users with personalized services.

A robotic vacuum with LG ThinQ vision AI is placed in a square-shaped room. This robot can split the room into a grid of size $10^9 \times 10^9$ and determine whether each cell is contaminated or not. Let's call the cell in the y-th row from the top and x-th column from the left as cell (x, y). The robot compresses the information of N distinct contaminated cells and sends it to the server as follows.

1. The (x, y) coordinates of contaminated cells are sorted in the increasing order of x coordinates (for equal x coordinates, in increasing order of y coordinates).
2. The x coordinates are removed after sorting, and only the y coordinates are sent to the server in their order.

When two contaminated cells are adjacent horizontally or vertically, we consider the two cells to be in the same contamination zone. The server should estimate the number of contamination zones from the data it received from the robot. Given the data that was sent to the server, find the minimum and maximum possible number of contamination zones.

Input

The first line of input contains an integer N, denoting the number of contaminated cells. $(1 \le N \le 200\,000)$

The following N lines contain N integers $y_1, \cdots, y_N$ representing the y coordinates of the contaminated cells in the order received by the server, one number per line. $(1 \le y_i \le 10^9)$

Output

Print the minimum possible number of contamination zones in the first line.

Print the maximum possible number of contamination zones in the second line.

Example

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

Problem J. XOR Graph

Time Limit 1 second Memory Limit 1024 MB

An XOR graph with N vertices is defined as follows.

- The graph has N vertices numbered from 0 to $N-1$.
- For each vertex i, there are directed edges from vertex i to vertex $i \oplus t$ and vertex $(i \oplus t)+1$.
- However, if the destination vertex exceeds $N-1$, the edge does not exist.

Here, $\oplus$ denotes the bitwise XOR operation.

Given the number of vertices N, the starting vertex x, the destination vertex y, and a non-negative integer t, find the minimum number of edges needed to move from vertex x to vertex y.

Input

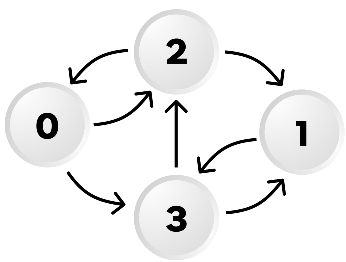
The first line of input contains four space-separated integers N, x, y, and t. ($2 \le N \le 10^{18}$; $0 \le x, y < N$; $x \ne y$; $0 \le t < 2^{20}$)

Output

Print the minimum number of edges needed to move from vertex x to vertex y. If no such path exists, print -1 instead.

Examples

standard input	standard output
`4 2 3 2`	`2`



standard input	standard output
`10 7 9 6`	`-1`
`165321961421 998244353 5029393147 98207`	`8242633`

Problem K. Distance Multiplication Maximization

Time Limit 7 seconds   Memory Limit 1024 MB

There is a tree (a connected undirected graph with no cycles) consisting of N vertices. The vertices are numbered from 1 to N, and the edges are numbered from 1 to $(N-1)$.

Write a program that performs the following query.

- u v: For vertex x $(1 \le x \le N)$, print the maximum value of $\text{dist}(x,u) \times \text{dist}(x,v)$. $(1 \le u, v \le N)$

Here, $\text{dist}(x,y)$ is defined as the number of edges on the shortest path from vertex x to vertex y. For any vertex x, $\text{dist}(x,x) = 0$.

Input

The first line of input contains N, denoting the number of vertices of the tree. $(2 \le N \le 300\,000)$

The following $(N-1)$ lines contain the information of the tree. The i-th of them contains two space-separated integers denoting the two vertices connected by edge i.

The next line contains Q, denoting the number of queries. $(2 \le Q \le 300\,000)$

Each of the following Q lines contains the information for each query in the same format as written in the problem statement.

Output

Print Q lines containing the answer to each query, one per line.

Example

standard input	standard output
5 1 2 2 3 2 4 4 5 3 1 2 2 5 3 3	6 3 9