Problem A. Advertisement Matching

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

MOLOCO is a company that matches advertisers with potential users using their high-performance ad platform.

MOLOCO is in contact with N advertisers, where the i-th advertiser has paid for a_i advertisements to deliver. Our advanced prediction algorithm has picked M potential recipients, which we will deliver the advertisements to. For the j-th audience, we can deliver up to b_j advertisements.

Jaehyun is testing several hypotheses to increase engagement in the advertisements. One day, Jaehyun thought that all advertisements received by a single recipient should come from different advertisers: it is boring to watch the same advertisement multiple times.

Jaehyun wants to estimate the profitability of his hypotheses. He will perform the following kinds of updates.

- `1 i`: Increase a_i by one.
- `2 i`: Decrease a_i by one.
- `3 j`: Increase b_j by one.
- `4 j`: Decrease b_j by one.

All updates are cumulative. Jaehyun wants to check if the system can deliver all advertisements of our advertisers given the changing landscape of the advertisers and recipients.

Input

The first line contains two integers, N and M ($1 \leq N, M \leq 250\,000$).

The next line contains N integers $a_1, a_2, \ldots, a_N$ ($0 \leq a_i \leq 250\,000$).

The next line contains M integers $b_1, b_2, \ldots, b_M$ ($0 \leq b_j \leq 250\,000$).

The next line contains a single integer Q ($1 \leq Q \leq 250\,000$).

The next Q lines contain two integers in one of the following forms:

- `1 i` ($1 \leq i \leq N$)
- `2 i` ($1 \leq i \leq N$)
- `3 j` ($1 \leq j \leq M$)
- `4 j` ($1 \leq j \leq M$)

The input will be set in a way such that all a_i and b_j values are always nonnegative.

Output

Print Q lines. On the i-th line, print 1 if all advertisements can be delivered given the first i updates, and 0 otherwise.

Example

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

Problem B. Cactus Competition

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

In Byteland, there is a school with an enormously large playground. The playground is an $N \times M$ grid. Since the playground is so large, the weather may differ in different squares of the grid. For $1 \le i \le N, 1 \le j \le M$, cell (i, j) has a temperature of $A_i + B_j$. The sequences A and B are known in advance.

Sunghyeon wants to organize a relay race competition in the playground. The race will start in cell $(S, 1)$ and end in cell (T, M), where S and T are integers that are to be determined. In the race, students will only run to the right and down. In other words, a student in cell (i, j) can move directly to cells $(i, j + 1)$ or $(i + 1, j)$. From this, it is clear that $1 \le S \le T \le N$ must hold for the race to be valid.

The people of Byteland love cacti. Therefore, all students will hold a cactus while in the race. Since cacti can't endure cold weather, all cells in the racetrack should satisfy $A_i + B_j \ge 0$.

There are $\frac{N(N+1)}{2}$ candidate races over all candidate pairs of S and T. Please find the number of valid pairs - a pair is valid if the corresponding race can be completed given the constraints due to the cacti.

Input

The first line contains two space-separated integers N, M $(1 \le N, M \le 200\,000)$.

The second line contains N space-separated integers $A_1, A_2, \cdots, A_N$ $(-10^9 \le A_i \le 10^9)$.

The third and final line contains M space-separated integers $B_1, B_2, \cdots, B_M$ $(-10^9 \le B_i \le 10^9)$.

Output

Print the number of valid pairs of S and T.

Examples

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

Problem C. Economic One-way Roads

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

The country of RUN has N cities, some of which are connected by two-way roads. Each road connects two different cities, and no two roads connect the same pair of cities. It is not guaranteed that every city is reachable from every other city by traveling along some roads.

Due to traffic issues, the mayor of RUN decided to make all roads one-way. After doing so, it must be possible to move from any city to any other city using one or more roads. To save as much money as possible, over all possible road orientations that satisfy this condition, the mayor will pick the cheapest one. Note that the cost of orienting a road depends on both the specific road and the direction it is oriented in.

Input

On the first line, the number of cities $2 \le N \le 18$ is given.

On each of the next N lines, N space-separated integers are given. The j-th integer in the $i+1$-th line, a_{ij}, is the cost of orienting the road from city i to j, or -1 if there is no road connecting these two cities.

For all integers $1 \le i \le N$, $a_{ii} = -1$. For all pairs of distinct integers $1 \le i, j \le N$, either $a_{ij} = a_{ji} = -1$ or $0 \le a_{ij}, a_{ji} \le 10^6$.

Output

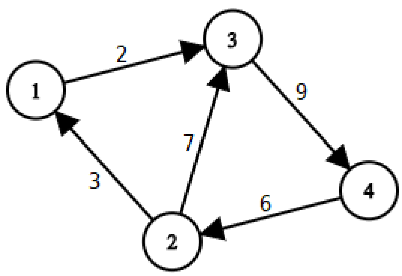
Output the minimum cost needed to orient all roads to satisfy the mayor's condition. If it is impossible, output `-1`.

Examples

standard input	standard output
4 -1 3 2 -1 3 -1 7 7 5 9 -1 9 -1 6 7 -1	27
6 -1 1 2 -1 -1 -1 3 -1 4 -1 -1 -1 5 6 -1 0 -1 -1 -1 -1 0 -1 6 5 -1 -1 -1 4 -1 3 -1 -1 -1 2 1 -1	-1

Note

For the first sample, this is the cheapest way to orient the roads to satisfy the mayor's condition:



Problem D. Just Meeting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

SNUPS is the programming contest club of Seoul National University. There are N members in the club, and for every two distinct members $1 \leq i, j \leq N, i \neq j$, we define the *friend distance* $C(i, j)$. The friend distance is symmetric ($C(i, j) = C(j, i)$ for all $i \neq j$), and is an integer in range $[1, 10^7]$. The smaller the friend distance is, the more friendly two members are toe each other.

When there is an agenda to discuss in SNUPS, they select three distinct members uniformly at random and the three members hold a discussion. However, if $C(i, j) < C(j, k)$ and $C(i, j) < C(i, k)$ for the three members i, j, k in discussion, then members i and j may collude and ignore member k. We call this an *unjust meeting*.

Currently, there are M pairs of friends, and their friend distance **can not be changed**. All other pairs of members can have their friend distances changed. Sunghyeon will hold a party for the SNUPS members and change the friend distances for all of these pairs. After the party, the friend distance should satisfy the symmetry condition and should be an integer in range $[1, 10^7]$.

Sunghyeon wants to know if he can set the friend distance in such a way that it is impossible to have an unjust meeting. If this is possible, he also wants to minimize $\sum_{i=1}^{N} \sum_{j=i+1}^{N} C(i, j)$ in order to make the club as friendly as possible.

Input

The first line contains two space-separated integers N, M. ($3 \leq N \leq 300\,000, 0 \leq M \leq 300\,000$).

The next M lines contains three space-separated integers A_i, B_i, D_i. This means that the member A_i and B_i are friends, and have a friend distance of D_i. ($1 \leq A_i, B_i \leq N, 1 \leq D_i \leq 10^7, A_i \neq B_i$).

These M lines describe exactly all friends in SNUPS, and no pair of friends is described more than once. Formally, for all $1 \leq i \neq j \leq M$, $\{A_i, B_i\} \neq \{A_j, B_j\}$.

Output

Print `-1` if Sunghyeon's plan is not satisfiable. Otherwise, print the minimum possible value of $\sum_{i=1}^{N} \sum_{j=i+1}^{N} C(i, j)$.

Examples

standard input	standard output
4 2 1 2 5 2 4 3	14
4 4 1 2 10 1 3 20 2 4 30 3 4 40	-1

Problem E. Chemistry

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

Changki Yun is a professor in the Department of Chemistry, Seoul National University. To fight the ongoing COVID-19 pandemic, Changki conducts research on a certain molecule.

The molecule consists of N atoms with M chemical bonds, which we will simply consider as an undirected graph without self-loops or parallel edges. Note that, unlike real-life chemistry, it is not guaranteed that the molecule is connected, or that atoms in the molecule have degree at most 4.

Changki can use a machine to change the molecule. When Changki enters two integers $1 \leq L \leq R \leq N$, the machine will keep only atoms in the range $L, L+1, \cdots, R$, as well as any bonds that were only between kept atoms.

Changki thinks that molecules which form **chains** are crucial to his research. A molecule forms a chain if you can place the atoms in a line such that a chemical bond between two atoms exists if and only if they are adjacent on the line. Please count the number of pairs (L, R) which can be put in the machine to form a chain.

Input

The first line contains two space-separated integers N, M $(1 \leq N \leq 250\,000, 0 \leq M \leq 250\,000)$.

The next M lines contain two space-separated integers u, v denoting that there is a chemical bond connecting vertices u and v. $(1 \leq u, v \leq N, u \neq v)$ There are no parallel edges.

Output

Print the single integer denoting the answer.

Examples

standard input	standard output
3 3 1 2 2 3 3 1	5
8 7 2 1 1 4 4 3 3 8 8 7 7 5 5 6	17
12 12 1 2 3 4 5 6 7 8 9 10 11 12 2 4 4 6 6 8 8 10 10 12 12 2	28

Problem F. Interval Graph

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

You are given N closed intervals. The i-th interval has the range $[s_i, e_i]$, and has a positive integer weight w_i. Consider the undirected graph of N vertices, where each vertex corresponds to an interval, and there exists an edge between two vertices if and only if the corresponding pair of intervals has a nonempty intersection. For a given list of intervals, we call this graph the *interval graph*.

Jaehyun is addicted to problems about trees and queries, so he wants the graph to be acyclic. To accomplish this, he can delete zero or more intervals. Jaehyun is lazy, so over all possible ways to delete intervals, he will select the way which minimizes the weight of the intervals he has to delete. Print the weight of the remaining intervals after he has made the interval graph acyclic.

Input

The first line contains the single integer N $(1 \leq N \leq 250\,000)$.

The next N lines contain three space-separated integers s_i, e_i, w_i $(1 \leq s_i \leq e_i \leq 500\,000,\ 1 \leq w_i \leq 10^{12})$.

Output

Print the weight of the remaining intervals after Jaehyun's deletions.

Examples

standard input	standard output
3 1 3 10 3 5 20 5 7 30	60
3 1 3 1 2 3 2 3 3 3	5

Problem G. LCS 8

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

You are given a string S of length N, consisting of uppercase letters, and a small nonnegative integer K.

Please compute the number of strings T of length N, consisting of only uppercase letters, such that the longest common subsequence of S and T has length at least $N - K$. As the number could be large, print the number of such strings modulo $10^9 + 7$.

A string $S = s_1 s_2 \dots s_n$ is a subsequence of a string $T = t_1 t_2 \dots t_m$ if there exists an increasing sequence of indices $1 \le i_1 < i_2 < \dots < i_n \le m$ such that $s_x = t_{i_x}$ for all $1 \le x \le n$.

Input

The first line of the input contains the length-N string S ($1 \le |S| \le 50\,000$). All characters of S are uppercase letters.

The next line of the input contains the single integer K ($0 \le K \le 3$).

Output

Print the number of such strings modulo $10^9 + 7$.

Examples

standard input	standard output
ACAYKP 0	1
CAPCAK 1	896
WEDONTNEEDNOEDUCATION 2	24651976
WEDONTNEEDNOTHOUGHTCONTROL 3	224129308

Problem H. Alchemy

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Jeyeon the alchemist has obtained the *philosopher's stone* which can transform one or more materials into a single material of a different type. Now he is trying to obtain the most valuable material in the universe. Every material in the world has a value corresponding to some nonnegative integer, and two materials are different in type if and only if they differ in value.

The philosopher's stone can consume k ($k \geq 1$) materials, the ith having value x_i, and generates a material that was not among the consumed values. Over all such materials, the philosopher's stone will generate the material with minimum value, also known as the *mex* of the original values. It is allowed to consume multiple materials of the same value.

Jeyeon starts out with some elements with values varying from 0 to $N-1$. Specifically, he has c_i materials with value i. He will use the philosopher's stone zero or more times. In the end, Jeyeon wants to own exactly one material. Jeyeon wants to maximize the value of this single material. Can you calculate this value for him? Note that the value of this material may be greater than or equal to N.

Input

The first line contains the single integer N. ($1 \leq N \leq 100\,000$)

The next line contains N space-separated integers $c_0, c_1, \cdots, c_{N-1}$. ($0 \leq c_i \leq 10^9$, $c_0 + c_1 + \cdots + c_{N-1} \geq 1$)

Output

Print the maximum possible value of the single remaining material.

Examples

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

Problem I. Query On A Tree 17

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

On *Baekjoon Online Judge*, there are a series of problems related to processing queries on a tree. We KAISTians are proud to offer the seventeenth edition in this series.

You are given a tree with N vertices. Each vertex is numbered from 1 to N. The tree is rooted at vertex 1.

People can live in each vertex. Let $A[i]$ be the number of people living in vertex i. Initially, $A[i] = 0$ for all $1 \le i \le N$.

Write a program that processes the Q following queries:

- `1 u`: Add 1 to $A[i]$ for all vertices i in the subtree rooted at vertex u.
- `2 u v`: Add 1 to $A[i]$ for all vertices i on the unique shortest path between the two vertices u and v. Note that u and v might be equal.

After each query, print the vertex x that minimizes the quantity $\sum_{y=1}^{N} A[y] \times dist(x, y)$, where $dist(x, y)$ is the number of edges on the path between x and y. If there is more than one such vertex, print the vertex with minimum distance from the root (vertex 1). It can be proven that such vertex is unique. In other words, we should find a vertex that minimizes the total distance needed for everyone to gather.

Input

The first line contains the single integer N $(2 \le N \le 100\,000)$.

The next $N - 1$ lines describe all edges of the tree. Each line contains two space-separated integers u, v denoting that there is an edge connecting vertices u and v. $(1 \le u, v \le N, u \ne v)$

The next line contains the single integer Q $(1 \le Q \le 100\,000)$.

The next Q lines contains several integers in one of the following forms:

- `1 u` $(1 \le u \le N)$
- `2 u v` $(1 \le u, v \le N)$

Output

Print Q lines, denoting the answer after each update.

Example

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

Problem J. Remote Control

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Jaemin lives on an infinitely large rectangular grid. Recently, he moved to a new grid, so currently there is only one wall in the grid, which occupies one of the cells.

Each cell has a coordinate determined by the following: the wall is at the coordinate $(0, 0)$. If a cell has coordinate (x, y), then the cell immediately to the right is $(x + 1, y)$, the cell immediately to the left is $(x - 1, y)$, the cell immediately above is $(x, y + 1)$, and the cell immediately below is $(x, y - 1)$.

Today, he brought his favorite remote-controlled toy car to the grid. The car occupies exactly one cell in the grid. Unfortunately, as this grid is extremely large, he forgot where he put the car. In this case, the only way to move the car is to press a button on the remote control. Upon pressing it, the car will try to execute a predetermined path of N moves, where each move involves the car trying to move to an adjacent cell in one of the four directions. Note that the car cannot go into the wall; if there is a wall at the cell the car is trying to move to, the car will ignore that command and not move, but will continue to try to execute the moves afterwards.

Since you decided to help him, he will ask you Q questions of the form: "if my car starts at (x, y), and I press the button once, where will it end up at?" Can you answer his questions?

Input

On the first line, the length of the command N is given, where $1 \le N \le 300\,000$.

On the next line, the car's predetermined path is given. It is a string of length N, and each character is either `L`, `R`, `U`, or `D`, meaning that the car moves left, right, up, or down, respectively. When he presses the button, the car follows each character of the command one by one as described above.

On the next line, the number of questions Q is given, where $1 \le Q \le 300\,000$.

On each of the next Q lines, two space-separated integers x and y are given. The starting point of the car is (x, y). The input satisfies $-300\,000 \le x, y \le 300\,000$, and $(x, y) \ne (0, 0)$.

Output

For each question, output x and y on one line, separated by a space, where (x, y) is the answer to the question.

Example

standard input	standard output
8 RRDRUULL 5 -2 1 -2 2 -2 -1 -3 -1 1 1	-1 3 -1 3 1 0 -2 -1 2 2

Problem K. Sewing Graph

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Donghyun recently bought a square-shaped tablecloth. There are N dots on the cloth, and the dots can be seen from both sides of the cloth. Donghyun thought that the tablecloth can be made more beautiful, so he decided to decorate the cloth with sewing.

For convenience, let's assume that each dot is a point on the xy-plane and the dots are numbered from 1 to N. Dot i ($1 \le i \le N$) is placed at coordinate (x_i, y_i). No two dots have the same coordinates. A **sewing sequence** is an integer sequence $\{s_i\}$ of length $k \ge 2$ satisfying $1 \le s_i \le N$ ($1 \le i \le k$) and $s_i \ne s_{i+1}$ ($1 \le i \le k-1$). The sequence draws edges on the cloth per the following rules:

- Draw an edge connecting dot s_{2i-1} and dot s_{2i} on the front side of the cloth for all $1 \le i \le \left\lfloor \frac{k}{2} \right\rfloor$.
- Draw an edge connecting dot s_{2j} and dot s_{2j+1} on the back side of the cloth for all $1 \le j \le \left\lfloor \frac{k-1}{2} \right\rfloor$.

Donghyun wants to make a **beautiful pattern** on the tablecloth, which is defined as the following:

- For both sides of the cloth, all N dots are connected by the edges on that side.
- Two edges on the same side of the cloth can intersect only at a common endpoint.

Donghyun is very busy, so he wants to finish his sewing job as quickly as possible. In other words, over all sewing sequences that produces a beautiful pattern, Donghyun decides to choose the shortest such sequence. Your job is to find such a sequence.

Note that Donghyun wants to minimize the length of the sewing sequence itself, not the sum of the lengths of the edges he draws.

Input

On the first line, a single integer N is given. ($2 \le N \le 1\,000$)

For each of the next N lines, two integers x_i and y_i are given, which means dot i is placed at coordinate (x_i, y_i). ($1 \le x_i, y_i \le 10^9$)

No two dots are at the same coordinates.

Output

On the first line, output a positive integer k, the length of the shortest sewing sequence that produces a beautiful pattern.

On the next line, output s_1, s_2, $\cdots$, s_k, the actual sewing sequence.

It can be proven that, for every possible input, there exists a sewing sequence that produces a beautiful pattern.

Example

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

Note

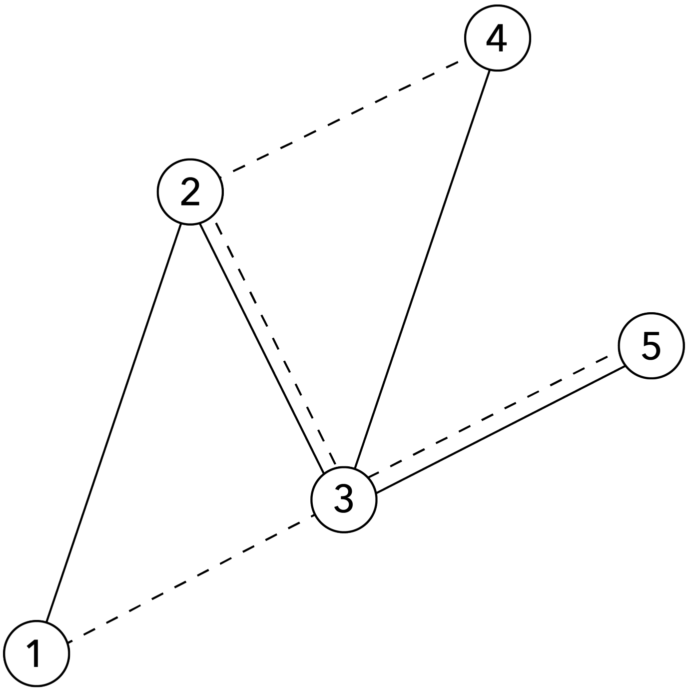
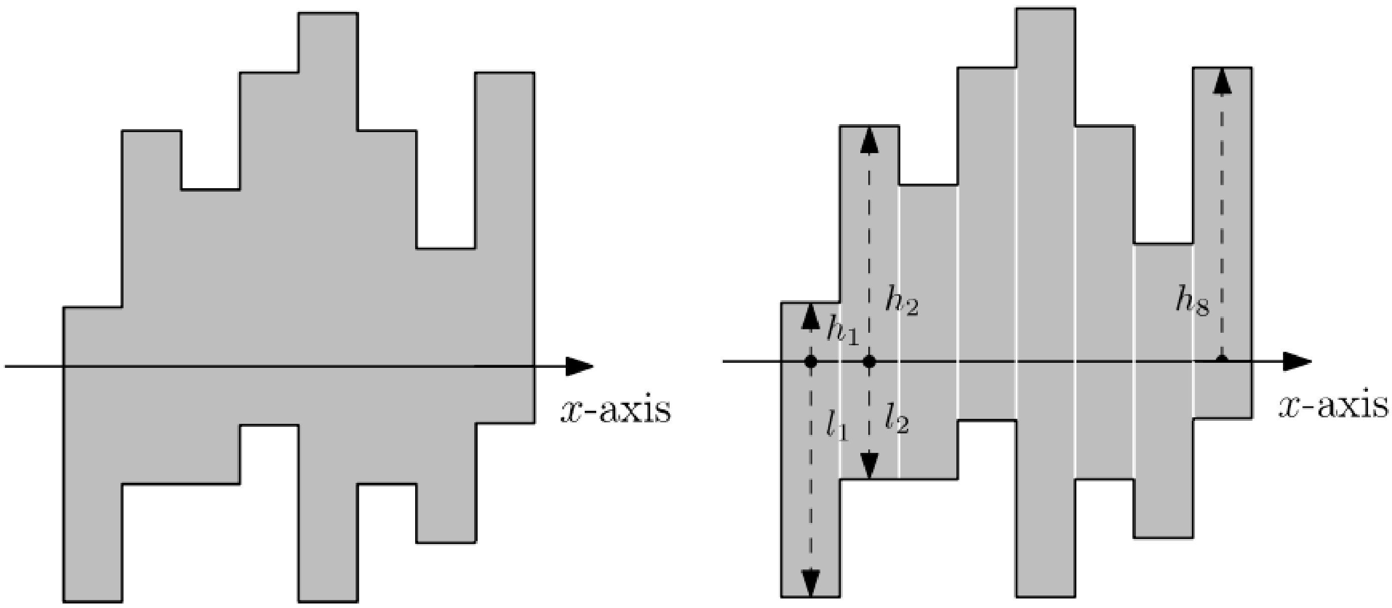


Figure 1. Visualization of sample output. Solid lines represent for edges on front side, dashed lines represent for back side.

Problem L. Steel Slicing 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

ISCO(ICPC Steel Company) is a company that buys steel sheets of a certain shape, cuts them into pieces, and sells them in the industry market. Every steel sheet that ISCO buys has a shape of two histograms of equal width, where one histogram is reflected vertically and soldered to the bottom of the other histogram. This process forms a polygon without holes such that each side is either horizontal or vertical. We call such a polygon a histogon. See the below figure for a histogon.



Since the market price of a piece becomes much higher if the steel piece is rectangular, it is desirable to cut a steel sheet into several rectangles. To achieve this, you will use a laser cutter.

In a single operation, the laser cutter can trace either a horizontal segment or a vertical segment through a polygon that touches the border of the polygon at exactly two points, the two endpoints of the segment. After the move, the polygon will be cut into two smaller polygons per the path the laser cutter traced. Note that the laser cutter can only operate on a single polygon in one operation.

The laser cutter is expensive to use, so your task is to find the minimum number of laser cutter operations needed such that after all the operations, all of the resulting polygons are rectangles.

Input

The first line of the input contains the number N, denoting the width of the histogon. $(1 \le N \le 250\,000)$

The next N lines contain two integers h_i, l_i, denoting the height of the first histogram and second histogram, respectively, for the i-th column. h_i denotes the height of the i-th column of the histogram which is not reflected, and l_i denotes the height of the i-th column of the histogram which is reflected. $(1 \le h_i, l_i \le 1\,000\,000)$

Output

Print a single integer denoting the minimum operations needed.

Examples

standard input	standard output
8 1 4 4 2 3 2 5 1 6 4 4 2 2 3 5 1	7
5 23 15 23 17 3 22 15 3 5 1	4