KAIST 10th ICPC Mock Competition

Sponsored By:

Rules

- This contest is KAIST 10th ICPC Mock Competition.
- This contest is sponsored by DEVSISTERS, Jane Street, MOLOCO, NAVER D2, and STARTLINK.
- This contest starts at 11:30 and ends at 16:30, Oct 9th 2020, Korean Standard Time (GMT +9).
- You can only participate in a team of three students.
- Unlike our previous contests, you are allowed to use the Internet and any reference materials.
- This contest consists of 12 problems.
- Problems are NOT sorted by difficulty.
- Solutions to problems submitted for judging are called *run*s. Each run is judged as accepted or rejected, and the team is notified of the results.
- Teams are ranked according to the most problems solved. Teams who solve the same number of problems are ranked first by least total time and, if need be, by the earliest time of submission of the last accepted run.
- The *total time* is the sum of the time *consumed* for each problem solved. The time consumed for a solved problem is the time elapsed from the beginning of the contest to the submission of the first accepted run plus 20 penalty minutes for every previously rejected run for that problem. There is no time consumed for a problem that is not solved.
- All problems have the same memory limit, which is language-dependent and is as follows:
  - `C11/C++17      : 1024MB`
  - `Java/Kotlin    : 1536MB`
  - `PyPy 2/Python 3: 2048MB`
- Problems may have different time limits.

Problem list

#	Problem Name	Time limit (All Languages)
A	Advertisement Matching	2000 ms
B	Bombs In My Deck	1000 ms
C	Economic One-way Roads	5000 ms
D	Fix Wiring	1000 ms
E	LCS 8	3000 ms
F	Min-hashing	1000 ms
G	Mock Competition Marketing	1000 ms
H	Query On A Tree 17	2000 ms
I	Remote Control	2000 ms
J	Sewing Graph	2000 ms
K	Square, Not Rectangle	1500 ms
L	Steel Slicing 2	1000 ms

Problem A. Advertisement Matching

Time limit: 2 seconds

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 recipient, 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 ($1 \le N, M \le 250\,000$).

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

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

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

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

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

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. Bombs In My Deck

Time limit: 1 second

Donghyun is playing a digital card game. In this game, two players start with a deck (stack of cards) of 30 cards and 30 HP (health points). They alternate turns drawing and playing their cards. The winner is the first person who makes their opponent's HP less than or equal to 0.

This time, Donghyun met a tough opponent; the opponent mixed some **bombs** into Donghyun's deck! Now, there are A cards in Donghyun's deck, and B of them are **bombs**. Each card in the deck is equally likely to be a bomb.

Donghyun starts his next turn with C HP. On his next turn, he will remove cards from the top of his deck one at a time until he removes a card that is not a bomb or until his HP becomes less than or equal to 0. For each bomb that he removes, he loses 5 HP. Donghyun's deck is guaranteed to contain at least one card which is not a bomb, so this process is guaranteed to terminate.

Donghyun is worried he may lose the game because of the bombs. Specifically, he will lose the game if and only if his HP becomes less than or equal to 0. Donghyun asks you to calculate the probability that he does not lose the game in his next turn.

Input

On the first and only line, three space-separated integers A, B, and C are given. ($1 \leq B < A \leq 30$, $1 \leq C \leq 30$)

Donghyun has A cards in his deck, B of them are bombs, and his current HP is C.

Output

Output the probability that Donghyun survives after his next turn.

Your output will be considered correct if the absolute error between your answer and the jury's answer does not exceed 10^{-6}.

Example

standard input	standard output
4 2 5	0.5
4 2 6	0.833333333
4 2 20	1

Problem C. Economic One-way Roads

Time limit: 5 seconds

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 $2 \leq N \leq 18$ 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 $1 \leq i \leq N$, $a_{ii} = -1$. For all pairs of distinct integers $1 \leq i, j \leq N$, either $a_{ij} = a_{ji} = -1$ or $0 \leq a_{ij}, a_{ji} \leq 10^6$.

Output

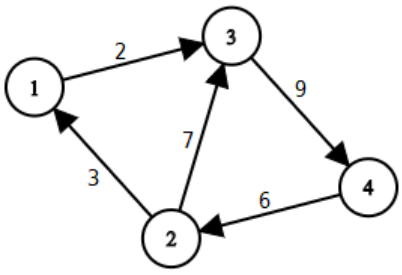
Output the minimum cost needed to orient all roads to satisfy the mayor's condition. If it is impossible, output `-1`.

Example

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

standard input	standard output
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

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



Problem D. Fix Wiring

Time limit: 1 second

You are in the spaceship *The Skeld* with your fellow crewmates. However, while checking the central power system, you found out that one crucial wiring installation of the central power system has been sabotaged. To prevent an engine failure, you should quickly fix the installation.

The installation consists of N nodes and $M = \frac{N(N-1)}{2}$ wires. All possible pairs of distinct nodes in the installation are connected with a wire. Originally, each of the M wires had exactly one tag attached to it. Each tag has a positive integer value, and different tags may have the same value. However, due to the sabotage, all the tags were removed from the wires and left on the floor. Fortunately, you gathered all M tags from the ground. Now, in order to fix the installation, you have to trigger the reboot sequence by re-attaching all the tags back to the wires twice.

For an installation where all wires have a tag on them, let's define **the cost of an installation** as the cost of its minimum spanning tree, that is, the minimum cost of a subset of wires such that all nodes are connected using only the given subset of wires. Here, the cost of the set of wires is defined as the sum of the tag values of all wires in the set.

The reboot sequence is triggered in two steps:
First, you should attach the tags to **minimize** the cost of the installation.
Then, after detaching all tags, you should attach the tags to **maximize** the cost of the installation.

Please compute the cost of each installation.

Input

The first line of the input contains an integer N, representing the number of nodes. ($2 \leq N \leq 100$)

The second line of the input contains $M = \frac{N(N-1)}{2}$ positive integers $C_1, C_2, \cdots C_M$, representing the integer values of M tags. ($1 \leq C_i \leq 2 \cdot 10^9$)

Output

Output a single line containing two integers. The first should be the **minimum** possible cost of the installation, the second should be the **maximum** possible cost of the installation.

Example

standard input	standard output
4 5 3 8 8 5 9	13 16

Problem E. LCS 8

Time limit: 3 seconds

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 $1 \leq i_1 < i_2 < \dots < i_n \leq m$ such that $s_x = t_{i_x}$ for all $1 \leq x \leq n$.

Input

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

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

Output

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

Example

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

Problem F. Min-hashing

Time limit: 1 second

Consider an undirected simple graph $G = (V, E)$. The problem of finding a node with *similar connectivity* is a well-researched topic, because it acts as a good metric to determine which nodes are relevant to other nodes. Services such as "friend recommendation" in Facebook is a good example of its applications. To formalize the notion of similarity, the concept of *Jaccard similarity* can be used, which is defined as $|N(v_1) \cap N(v_2)|/|N(v_1) \cup N(v_2)|$, where $N(v) = \{u|(u, v) \in E\}$.

Here, we will instead discuss the *min-hashing* method. Assume each node v has the label l_v. The *shingle value* s_v of node v is defined as $s_v = \min\{l_u|u \in N(v)\}$. This method is efficient enough to keep up with industrial needs, and it is also a great metric for similarity: the Jaccard similarity between the set of neighbors $N(v_1)$ and $N(v_2)$ is an unbiased estimator of the probability that nodes v_1 and v_2 have the same shingle values, for random unique labels.

Let's think about a variant of min-hashing: we repeatedly perform min-hashing by taking the label as the previous iteration's shingle value. In this variant, for each node v and the number of iterations k, the value $h_v^{(k)}$ is defined as

$$h_v^{(k)} = \begin{cases} s_v, & \text{if } k = 1 \\ \min\{h_u^{(k-1)}|u \in N(v)\}, & \text{if } k \geq 2 \end{cases}$$

For each k, let c_k be the number of unordered pairs of distinct vertices $\{u, v\}$ such that $h_u^{(k)} = h_v^{(k)}$. Then, how does the value c_k change as k increases? In this problem, your task is to compute $\max_{k \in \mathbb{N}} c_k$.

Input

The first line contains two positive integers n and m ($1 \leq n \leq 100\,000, 1 \leq m \leq 250\,000$) representing the number of nodes and the number of edges, respectively. The nodes are numbered from 1 to n. Note that these are **not** the labels of the nodes.

The second line contains n integers comprising a permutation of the first n positive integers, where the i-th number in the line represents the initial label of node i.

Each of the next m lines contains two integers. The i-th of these lines contains two distinct integers u_i and v_i ($1 \leq u_i, v_i \leq n$), which means $\{u_i, v_i\} \in E$.

The input will be set in a way such that there are no self-loops, parallel edges, or nodes with a degree of zero.

Output

Print the maximum value of c_k over all positive integers k.

Example

standard input	standard output
5 5 1 2 3 4 5 1 2 2 3 3 4 4 5 5 1	10
4 3 1 2 3 4 1 2 2 3 3 4	2

Problem G. Mock Competition Marketing

Time limit: 1 second

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

Whenever the application has available space for ads, the app requests the *AdExchange* platform to determine which ad to show. Then, the *AdExchange* holds an auction, in which bidders like MOLOCO bid for the opportunity to show their advertisement.

RUN wants to advertise its 2020 ICPC Mock Competition. They have asked MOLOCO for the advertisement. RUN has a total of K dollars and wants to display the ad for internal apps used by KAISTians. The ads in those apps are in one of six types, and the bidding price depends only on the type of ad. The first bidder to bid on the ad gets to show their ad.

AdExchange has already determined the costs of the six ad types and the N auctions it will run today. In the i-th auction, *AdExchange* will run an auction for an ad of type c_i. *AdExchange* never runs two auctions at the same time, more specifically auction $i+1$ cannot start until after auction i ends.

MOLOCO will bid during auctions using the following strategy - before the auctions begin, RUN selects a set of ad types. During the i-th auction, if the ad type for that auction is in RUN's set, and RUN has enough money to bid on an ad of that type, MOLOCO will submit a bid. Otherwise, they will ignore it.

MOLOCO is very fast at bidding, so it will always be the first bidder if it bids on the ad. Determine the maximum number of ads they can bid on if RUN selects the set of ad types optimally.

Input

The first line contains two space-separated integers N, K. ($1 \le N \le 100\,000, 0 \le K \le 10^9$)

The next line contains 6 integers $b_1, b_2, \ldots, b_6$. b_i indicates the cost for ad type i. ($1 \le b_i \le 10^9$)

The next line contains N integers $c_1, c_2, \ldots, c_N$. c_i indicates the ad type of the i-th auction. ($1 \le c_i \le 6$)

Output

Print the maximum number of ads they can bid on.

Example

standard input	standard output
6 10 1 2 3 4 5 6 6 5 4 3 2 1	4
12 10 1 1 2 2 3 3 6 5 4 3 2 1 1 2 3 4 5 6	7

For both samples, it is optimal for RUN to select the set $\{1, 2, 3, 4\}$.

Problem H. Query On A Tree 17

Time limit: 2 seconds

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 I. Remote Control

Time limit: 2 seconds

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 $1 \leq N \leq 300\,000$.

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 $1 \leq Q \leq 300\,000$.

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 $-300\,000 \leq x, y \leq 300\,000$, and $(x, y) \neq (0, 0)$.

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 J. Sewing Graph

Time limit: 2 seconds

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 4 3 2 3 5 3 1

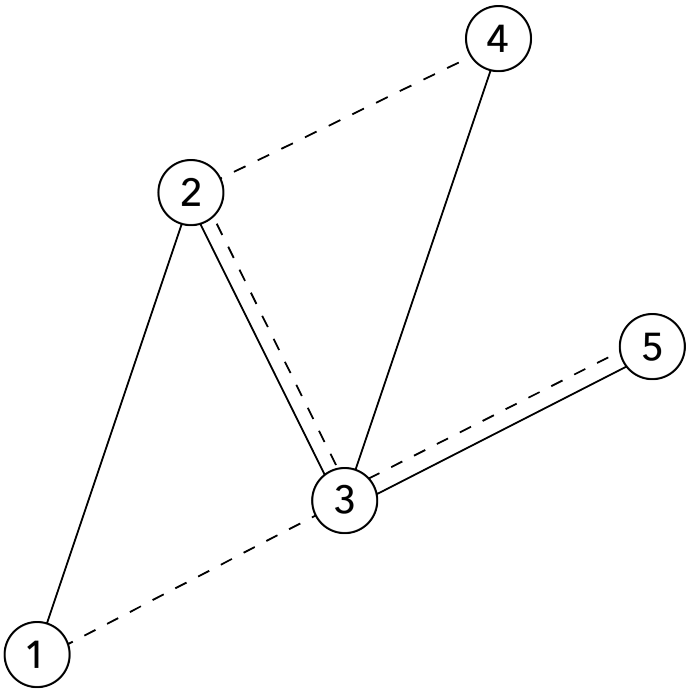


Figure 1. Visualization of sample output. Solid lines represent for edges on front side, dashed lines represent for back side.

Problem K. Square, Not Rectangle

Time limit: 1.5 seconds

A histogram is a polygon made by aligning N adjacent rectangles that share a common base line. Each rectangle is called a bar. The i-th bar from the left has width 1 and height H_i.

Your goal is to find the area of the largest rectangle contained in the given histogram, such that one of the sides is parallel to the base line.

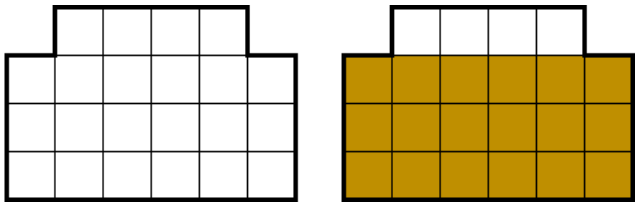


Figure 1. The histogram given in the example, with the largest rectangle shown on the right.

Actually, no, you have to find the largest **square**. Since the area of a square is determined by its side length, you are required to output the **side length** instead of the area.

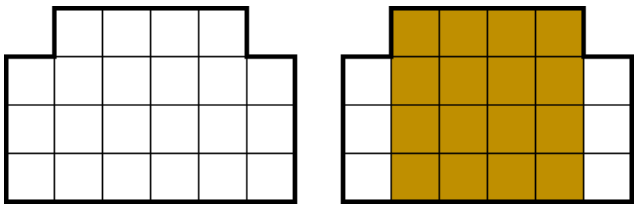


*Figure 2. The histogram given in the example, with the largest **square** shown on the right.*

Input

On the first line, a single integer N is given, where $1 \le N \le 300\,000$.

On the next line, N space-separated integers $H_1, H_2, \cdots, H_N$, are given. H_i $(1 \le H_i \le 10^9)$ is the height of the i-th bar.

Output

Output the side length of the largest square in the histogram, such that one of the sides is parallel to the base line.

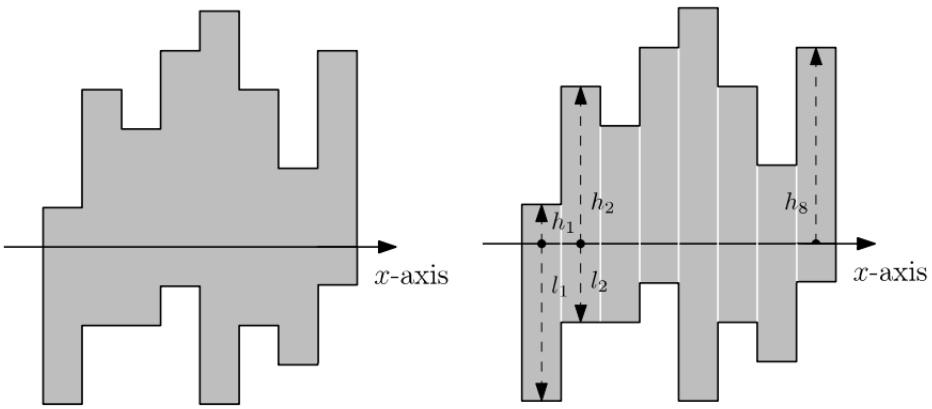
Example

standard input	standard output
6 3 4 4 4 4 3	4

Problem L. Steel Slicing 2

Time limit: 1 second

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. ($1 \leq N \leq 250\,000$)

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. ($1 \leq h_i, l_i \leq 1\,000\,000$)

Output

Print a single integer denoting the minimum operations needed.

Example

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