Problem A. Bipartite graph

Input file:	`bipartite.in`
Output file:	`bipartite.out`
Time limit:	1 second
Memory limit:	256 megabytes

Bipartite graph is an undirected graph $\langle V, E \rangle$ whose vertices can be divided into two disjoint sets L and R ($L \cap R = \emptyset$ and $L \cup R = V$), such that for every edge $(u, v) \in E$, either $u \in L, v \in R$ or $v \in L, u \in R$.

You are given undirected graph. Check if this graph is bipartite.

Input

First line contains two integers n and m — number of vertices and edges in graph, respectively ($1 \leq n \leq 100\,000$; $0 \leq m \leq 200\,000$). Next m lines contain edges; i-th of these lines contains two integers u_i and v_i — vertices connected by i-th edge ($1 \leq u_i, v_i \leq n$).

Output

Output "`YES`", if given graph is bipartite, or "`NO`" otherwise.

If the graph is bipartite output the proper division into two sets: for each vertex output 1, if this vertex in set L, otherwise output 2.

Examples

bipartite.in	bipartite.out
4 4 1 2 1 3 2 4 4 2	YES 1 2 2 1
3 3 1 2 2 3 3 1	NO

Problem B. Number of inversions

Input file:	`inverse.in`
Output file:	`inverse.out`
Time limit:	1 second
Memory limit:	256 megabytes

Given an array $A = \langle a_1, a_2, \ldots, a_n \rangle$ of distinct integers. You are to find number of pairs of indices (i, j) such that $i < j$ и $a_i > a_j$.

Input

First line of input contains integer n ($1 \le n \le 50\,000$) — number of elements in array A.

Second line contains array itself. No two elements of array coincide.

Output

Output one integer — the number of inversions in given array.

Examples

inverse.in	inverse.out
4 1 2 4 5	0
4 5 4 2 1	6

Problem C. Three lines

Input file:	`3lines.in`
Output file:	`3lines.out`
Time limit:	1 second
Memory limit:	256 megabytes

Farmer John wants to monitor his n cows ($1 \le n \le 50\,000$) using a new surveillance system he has purchased.

The ith cow is located at position (x_i, y_i) with integer coordinates; no two cows occupy the same position. FJ's surveillance system contains three special cameras, each of which is capable of observing all the cows along either a vertical or horizontal line. Please determine if it is possible for FJ to set up these three cameras so that he can monitor all n cows. That is, please determine if the n locations of the cows can all be simultaneously "covered" by some set of three lines, each of which is oriented either horizontally or vertically.

Input

The first line contains integer n ($1 \le n \le 50\,000$).

The i-th of the next n lines contains (x_i, y_i) — location of cow i ($0 \le x_i, y_i \le 10^9$).

Output

Output 1 if it is possible to monitor all n cows with three cameras, or 0 if not.

Examples

3lines.in	3lines.out
6 1 7 0 0 1 2 2 0 1 4 3 4	1

Note

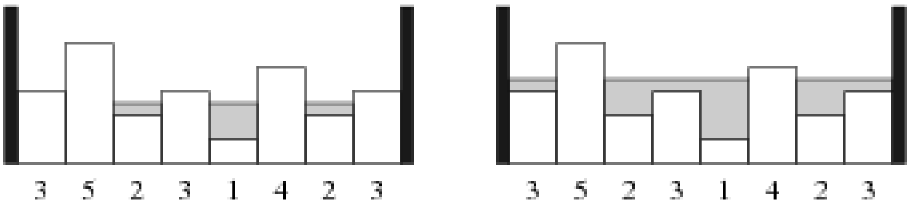
The lines $y = 0$, $x = 1$, and $y = 4$ are each either horizontal or vertical, and collectively they contain all n of the cow locations.

Problem D. Islands

Input file:	`islands.in`
Output file:	`islands.out`
Time limit:	1 second
Memory limit:	256 megabytes

Whenever it rains, Farmer John's field always ends up flooding. However, since the field isn't perfectly level, it fills up with water in a non-uniform fashion, leaving a number of "islands" separated by expanses of water.

FJ's field is described as a one-dimensional landscape specified by n ($1 \leq n \leq 100\,000$) consecutive height values $h_1, h_2 \ldots h_n$. Assuming that the landscape is surrounded by tall fences of effectively infinite height, consider what happens during a rainstorm: the lowest regions are covered by water first, giving a number of disjoint "islands", which eventually will all be covered up as the water continues to rise. The instant the water level become equal to the height of a piece of land, that piece of land is considered to be underwater.



An example is shown above: on the left, we have added just over 1 unit of water, which leaves 4 islands (the maximum we will ever see). Later on, after adding a total of 7 units of water, we reach the figure on the right with only two islands exposed. Please compute the maximum number of islands we will ever see at a single point in time during the storm, as the water rises all the way to the point where the entire field is underwater.

Input

The first line contains integer n ($1 \leq n \leq 100\,000$).

The next n lines contain h ($1 \leq h_i \leq 10^9$).

Output

Output integer giving the maximum number of islands that appear at any one point in time over the course of the rainstorm.

Examples

islands.in	islands.out
8 3 5 2 3 1 4 2 3	4

Problem E. Bale Share

Input file:	`baleshare.in`
Output file:	`baleshare.out`
Time limit:	1 second
Memory limit:	256 megabytes

Farmer John has just received a new shipment of n ($1 \le n \le 40$) bales of hay, where bale i has size s_i ($1 \le s_i \le 100$). He wants to divide the bales between his three barns as fairly as possible.

After some careful thought, FJ decides that a "fair" division of the hay bales should make the largest share as small as possible. That is, if b_1, b_2, and b_3 are the total sizes of all the bales placed in barns 1, 2, and 3, respectively (where $b_1 \ge b_2 \ge b_3$), then FJ wants to make b_1 as small as possible.

For example, if there are 8 bales in these sizes:

2 4 5 8 9 14 15 20

A fair solution is

Barn 1: 2 9 15 $b_1 = 26$
Barn 2: 4 8 14 $b_2 = 26$
Barn 3: 5 20 $b_3 = 25$

Please help FJ determine the value of b_1 for a fair division of the hay bales.

Input

The first line contains n ($1 \le n \le 40$), the number of bails

Each of the next n lines contains s_i ($1 \le s_i \le 100$), the size of the i-th bale.

Output

Please output in the first line the value of b_1 in a fair division of the hay bales.

Examples

baleshare.in	baleshare.out
8 14 2 5 15 8 9 20 4	26

Problem F. Mountain Climbing

Input file:	`climb.in`
Output file:	`climb.out`
Time limit:	1 second
Memory limit:	256 megabytes

Farmer John has discovered that his cows produce higher quality milk when they are subject to strenuous exercise. He therefore decides to send his n cows ($1 \leq n \leq 25\,000$) to climb up and then back down a nearby mountain!

Cow i takes u_i time to climb up the mountain and then d_i time to climb down the mountain. Being domesticated cows, each cow needs the help of a farmer for each leg of the climb, but due to the poor economy, there are only two farmers available, Farmer John and his cousin Farmer Don. FJ plans to guide cows for the upward climb, and FD will then guide the cows for the downward climb. Since every cow needs a guide, and there is only one farmer for each part of the voyage, at most one cow may be climbing upward at any point in time (assisted by FJ), and at most one cow may be climbing down at any point in time (assisted by FD). A group of cows may temporarily accumulate at the top of the mountain if they climb up and then need to wait for FD's assistance before climbing down. Cows may climb down in a different order than they climbed up.

Please determine the least possible amount of time for all n cows to make the entire journey.

Input

First line contains integer n ($1 \leq n \leq 25\,000$).

Next n lines contain two space-separated integers: u_i and d_i ($1 \leq u_i, d_i \leq 50\,000$).

Output

Output a single integer representing the least amount of time for all the cows to cross the mountain.

Examples

climb.in	climb.out
3 6 4 8 1 2 3	17

Note

If cow 3 goes first, then cow 1, and then cow 2 (and this same order is used for both the ascent and descent), this gives a total time of 17.

Problem G. Grass planting

Input file:	`grassplant.in`
Output file:	`grassplant.out`
Time limit:	1 second
Memory limit:	256 megabytes

Farmer John has n barren pastures ($2 \le n \le 100\,000$) connected by $n-1$ bidirectional roads, such that there is exactly one path between any two pastures. Bessie, a cow who loves her grazing time, often complains about how there is no grass on the roads between pastures. Farmer John loves Bessie very much, and today he is finally going to plant grass on the roads. He will do so using a procedure consisting of m steps ($1 \le m \le 100\,000$).

At each step one of two things will happen:

- FJ will choose two pastures, and plant a patch of grass along each road in between the two pastures, or,
- Bessie will ask about how many patches of grass on a particular road, and Farmer John must answer her question.

Farmer John is a very poor counter — help him answer Bessie's questions!

Input

First line contains two integers n and m ($2 \le n \le 10^5$; $1 \le m \le 10^5$).

Next $n-1$ lines describe the roads, each of them contains two integers specifying the endpoints of a road.

Next m lines describe steps. The first character of the line is either `P` or `Q`, which describes whether or not FJ is planting grass or simply querying. This is followed by two space-separated integers a_i and b_i ($1 \le a_i, b_i \le n$) which describe FJ's action or query.

Output

Output the answers to queries in the same order as the queries appear in the input, one answer per line.

Examples

grassplant.in	grassplant.out
4 6 1 4 2 4 3 4 P 2 3 P 1 3 Q 3 4 P 1 4 Q 2 4 Q 1 4	2 1 2

Problem H. Simplifying the Farm

Input file:	`simplify.in`
Output file:	`simplify.out`
Time limit:	1 second
Memory limit:	256 megabytes

Farmer John has been taking an evening algorithms course at his local university, and he has just learned about minimum spanning trees. However, Farmer John now realizes that the design of his farm is not as efficient as it could be, and he wants to simplify the layout of his farm.

The farm is currently arranged like a graph, with vertices representing fields and edges representing pathways between these fields, each having an associated length. Farmer John notes that for each distinct length, at most three pathways on his farm share this length. FJ would like to remove some of the pathways on his farm so that it becomes a tree — that is, so that there is one unique route between any pair of fields. Moreover, Farmer John would like this to be a minimum spanning tree — a tree having the smallest possible sum of edge lengths.

Help Farmer John compute not only the sum of edge lengths in a minimum spanning tree derived from his farm graph, but also the number of different possible minimum spanning trees he can create.

Input

The first line contains two integers n and m ($1 \le n \le 40\,000$; $1 \le m \le 100\,000$), representing the number of vertices and edges in the farm graph, respectively. Vertices are numbered from 1 to n.

The next m lines contain three integers a_i, b_i and l_i ($1 \le a_i$, $b_i \le n$; $1 \le l_i \le 1\,000\,000$) representing an edge from vertex a_i to b_i with length l_i. No edge length l_i will occur more than three times.

Output

Output two integers representing the length of the minimal spanning tree and the number of minimal spanning trees, modulo 1 000 000 007.

Examples

simplify.in	simplify.out
4 5 1 2 1 3 4 1 1 3 2 1 4 2 2 3 2	4 3

Note

Picking both edges with length 1 and any edge with length 2 yields a minimum spanning tree of length 4.