Problem A. Space Opera

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

The Space Team Fan Club has decided to perform the opera "Space Fights: The Power Gets Up". Not content with performing in any ordinary opera house, they have decided to host an outdoor performance on the moon!

The Club has acquired N stands and need to figure out how to place them to make as much profit as possible. They must place the stands in some order from closest to the stage to furthest away. The i-th stand consists of M_i tiers of seats. Each tier of seats they manage to sell in the stand is worth P_i Star Bucks (or \$ for short).

The catch is that a stand with M_i tiers of seats will stop anyone in the lowest M_i tiers of every stand placed further away from the stage from being able to see the opera. Opera fans are aware of this, so they won't buy these seats; the Space Team Fan Club will only be able to profit from those tiers of seats that can see the stage. Help them figure out the maximum profit they can make if they arrange the stands correctly.

Input

The first line of input contains a single integer N. Each of the next N lines contains two space-separated integers. The i-th of these lines contains the integers M_i and P_i.

Output

A single integer, the maximum profit the Space Team Fan Club can make.

Subtask 1 (points: 3)

The input will be exactly the same as one of the sample inputs.

Subtask 2 (points: 21)

$1 \leq N \leq 8$

For all i, $1 \leq M_i, P_i \leq 100$.

Subtask 3 (points: 17)

$1 \leq N \leq 1\,000$

For all i, $1 \leq M_i, P_i \leq 10\,000$.

Subtask 4 (points: 59)

$1 \leq N \leq 200\,000$

For all i, $1 \leq M_i, P_i \leq 10^8$

Examples

standard input	standard output
3 3 3 2 5 1 1	13
5 1 3 1 8 1 9 2 4 2 5	14

Problem B. Space Robots

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

The Space Team is having a Space Bike race across Space City. Help them find the optimal route.

Because Space City is in space, it has a three dimensional cubic grid of roads. To be more precise, Space City has N layers of intersections in the vertical direction, M in the North-South direction and P in the East-West direction. The intersections are labelled by coordinates (x, y, z), where $1 \le x \le N$, $1 \le y \le M$ and $1 \le z \le P$. There is a road joining two intersections if and only if their coordinates are the same in two dimensions and differ by exactly 1 in the third.

The Space Team knows that it takes exactly T seconds to cycle between two adjacent intersections along a road. Their problem is that they need to get past the robots. There is a robot at every intersection. At a particular intersection, the robot will be green in a particular direction (up-down, North-South or East-West), and red in the other directions. While the robot is green in the up-down direction, the Space Team can pass through the intersection as long as they enter the intersection along a vertical road. The Space Team may then leave the intersection along any road. However, if the Space Team comes from another direction, they will have to wait for the robot to go green for it.

A particular robot will be green in the up-down direction for U seconds at the start of the race. It will then be green in the North-South direction for S seconds, and then green in the East-West direction for W seconds. After this, the cycle will repeat, with the robot becoming green in the up-down direction for another U seconds, and so on.

The race starts at the intersection $(1, 1, 1)$ (the racers may immediately leave the intersection at the start of the race) and ends at intersection (N, M, P) (the racers must pass through the intersection to finish). What is the minimum time in which the Space Team can complete the race?

Input

The first line contains the four integers N, M, P, T. The next $N \times M \times P$ lines describe the intersections, one per line. The intersections are described in lexicographical order (see below): the first line describes intersection $(1, 1, 1)$, the second describes intersection $(1, 1, 2)$ and so on. Each line contains the three integers U, S and W describing the robot at that intersection.

Output

A single integer, the minimum time needed to complete the race, in seconds.

Constraints

$T \le 10^9$.

For all robots, $U, S, W \le 10^9$.

Subtask 1 (points: 21)

$N = M = 1$

$P \le 100$

$T \le 10^7$

For all robots, $U, S, W \le 10^7$.

Subtask 2 (points: 17)

$N = 1$

$M \times P \leq 100\,000$

$T = 1\,000$

For all robots, $U, S, W \leq 20$

Subtask 3 (points: 13)

$N \times M \times P \leq 20$

Subtask 4 (points: 49)

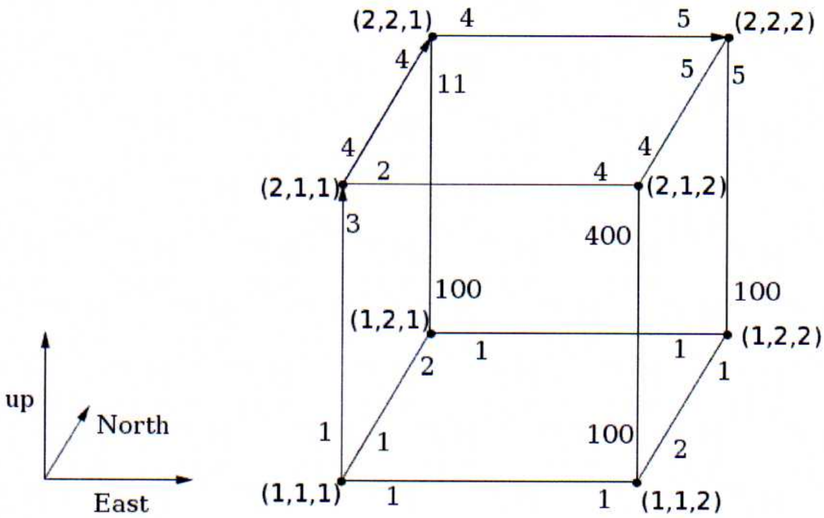
$N \times M \times P \leq 100\,000$

Examples

standard input	standard output
2 2 2 5 1 1 1 100 2 1 100 2 1 100 1 1 3 4 2 400 4 4 11 4 4 5 5 5	25

Note

In the sample input, the city looks like this:



The Team heads up out of intersection $(1, 1, 1)$. It takes them 5 seconds to reach intersection $(2, 1, 1)$. They wait 4 seconds for the light to turn green, then head North, arriving at intersection $(2, 2, 1)$ after a total of 14 seconds. The light is green when they arrive, so they immediately head East, arriving at intersection $(2, 2, 2)$ after a total of 19 seconds. They must wait 6 more seconds for the light to turn green for them, at which time they pass through the intersection and complete the race.

Intersections in lexicographical order are sorted primarily by their first coordinate, secondarily by their second coordinate, and only by their third coordinate if the first two are tied.

Problem C. Towers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

It is rumoured that when the universe was only 5 billion years old, the planet Sycerian orbited a white dwarf near the center of what would become the Milky Way. On this planet there existed a grand civilization of Lombaxes that built high and mighty xel'naga towers across the land. However their time of prosperity was short lived, and now it is not known what has become them. In his travels, Spock has come across Sycerian and wishes to discover the answer to the mystery of the Lombaxes. He knows their records are saved in a central mainframe on the planet, but the only way to access the mainframe is to know intricate details about the network of the xel'naga towers that have been left behind.

Each tower is indexed from 0 to $N-1$ and has a number in the range $[-1000\,000, 1000\,000]$ that levitates above it. While the numbers above each tower can change at any time, the underlying network of the towers is fixed. Each tower, excluding tower 0, is connected to a singe tower a level above it, and tower 0 is at the highest level in the network. A path through this network is a sequence of connected towers where each successive tower in the path is a level below the previous tower. The number of a path is the sum of the numbers on each of the towers in the path.

The mainframe's AI asks Spock questions of the form: how many paths of number W start at tower v in the network? If Spock can convince the mainframe he must be a Lombax by answering all the questions successfully, we will gain access to the secret records. Help Spock solve the Lombax mystery.

Input

The first line contains two space separated integers, N Q. Q is the total amount of questions the AI asks Spock, plus the total amount of times a number levitating above a tower changes.

The next line contains N integers, indexed from 0 to $N-1$. The ith number is T_i, where T_i is the index of the tower above and connected to tower i. T_0 is always -1 since tower 0 has no tower connected above it.

The next line contains N integers. The ith integer is L_i, the number currently levitating above tower i.

The next Q lines each contain 3 integers: $mode$ v W. If $mode$ is 0, this means you should output the number of paths of number W that start at tower v in the network on a single line. If $mode$ is 1, this means that the number that levitates above tower v now changes to W.

Output

For every query with $mode=0$, output the required value on a single line.

Constraints

In all cases, the number L that levitates above any tower satisfies $-1000\,000 \leq L \leq 1000\,000$.

In queries where mode=1, $-10^{16} \leq W \leq 10^{16}$.

Subtask 1 (points: 15)

$N \leq 1000$, $Q \leq 1000$

Subtask 2 (points: 20)

$N \leq 150\,000$, $Q \leq 10\,000$

For all i, $T_i = i - 1$.

At any time, the number levitating above any tower is larger or equal to 1.

All queries will have mode=0.

Subtask 3 (points: 20)

$N \leq 150\ 000$, $Q \leq 10\ 000$.

For all i, $T_i = i - 1$.

At any time, the number levitating above any tower is larger or equal to 1.

Subtask 4 (points: 45)

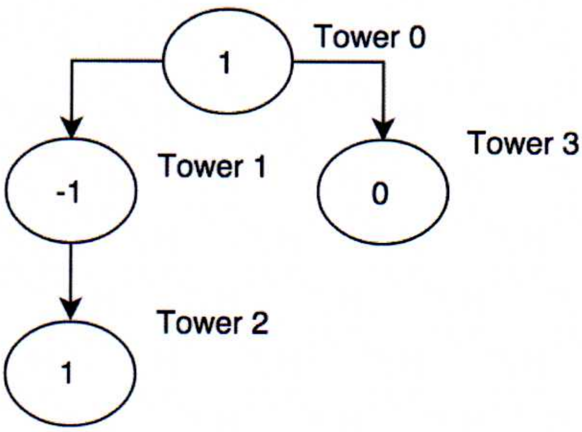
$N \leq 150\ 000$, $Q \leq 10\ 000$

Examples

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

Note

In the sample input, the tower network initially looks like this:



For the first query, there are 3 paths starting at tower 0 with path sum 1: The paths $0 \rightarrow 1 \rightarrow 2$ (1+-1+1=1), $0 \rightarrow 3$ (1+0=1), and the path consisting of the single tower 0 (1=1).

For the second query, the number above tower 3 changes to 7.

For the third query, the number above tower 1 changes to 1

For the fourth query, there is one path starting from tower 1 with path sum 2: $1 \rightarrow 2$ (1+1=2).

For the fifth query, there is one path starting from tower 0 with path sum 8: $0 \rightarrow 3$ (1+7=8).