TASK	AUTOCESTA	MULTI
input data	standard input	
output data	standard output	
time limit	1 sec	5 sec
memory limit	256 MB	256 MB
points	**100**	**100**
	200	

One well known wealthy trader has figured out a way to save on road tolls which his trucks pay on the state highway. Simply put - he will buy the highway. After some thinking, he came up with an even better plan - he will only buy those parts of the highway which are most profitable. That way the trucks whose routes are completely owned by him won't have to pay any toll.

The highway is split into sections with length of **one kilometer**. For each kilometer of the highway we know its **purchase price**.

To determine which sections he wants to buy, the trader will check out the ride plan for the next year. He knows all the routes his trucks will ride. Each **route** is determined with three numbers **A**, **B** and **C** in the following manner:

- On that route the truck **will enter the highway A kilometers** from the start of the highway and **leave the highway B kilometers** from the start, thus traveling |B-A| kilometers on the highway. For instance, if the truck needs to drive L kilometers from the start of the highway, A=0 and B=L.
- If the whole route from A to B is owned by the trader, the truck won't pay any toll. Otherwise, if the truck has to pass through at least one section **not owned by the trader**, he will have to pay total of C (regardless of the number of state owned sections of the highway that the truck has to pass through).

Additionally, the state has set a simple law - on each section of the highway, at most K trucks can pass **in one direction** and **K** in the other direction. However, this law only holds for the sections of the highway owned by the state (i.e. not owned by the trader).

Write a program that calculates he **minimal total expense** for the wealthy trader, in such a way that all his trucks can take complete their routes. The total expense is defined as the expense for **buying some parts** of the highway plus **all the tolls paid** by every truck going by a route not owned by the trader.

INPUT DATA

In the first line you can read **L** ($1 \leq \mathbf{L} \leq 100\,000$), total length of the highway in kilometers.

In the second line there are **L** integers $\mathbf{X_i}$ ($0 \leq \mathbf{X_i} \leq 1\,000\,000\,000$), purchase price of i-th section of the highway.

Third line contains **N** ($1 \leq \mathbf{N} \leq 100\,000$), the number of routes the trucks will take.

In each of the following **N** lines there are three integers $\mathbf{A_i}$, $\mathbf{B_i}$ and $\mathbf{C_i}$ ($0 \leq \mathbf{A_i}, \mathbf{B_i} \leq \mathbf{L}$, $\mathbf{A_i} \neq \mathbf{B_i}$, $0 \leq \mathbf{C_i} \leq 1\,000\,000\,000$), description of **i**-th route.

Last line contains number **K** ($1 \leq \mathbf{K} \leq 100$), maximal number of trucks that can go in each direction on a state owned section of the highway.

OUTPUT DATA

Write only one integer - minimal total expense for the trader if he optimally selects and buys some sections of the highway.

SCORING

In test cases worth 50% of total points points K is 1.

In test cases worth 34% of total points points N is at most 10.

In test cases worth 68% of total points points at least one of the last two conditions will hold.

In test cases worth 50% of total points points L is at most 1000.

In test cases worth 66% of total points points N is at most 1000.

TEST EXAMPLES

ulaz 3 300 300 300 2 0 3 400 2 1 400 99 **izlaz** 700	**ulaz** 10 1 3 3 1 1 1 2 2 2 3 5 0 10 2 1 5 4 1 4 4 9 0 2 10 9 4 2 **izlaz** 15

Description of the sample #1: If the trader doesn't buy any section, he must pay a total of 800 - 400 of toll for each route. If he buys the whole highway, he will pay 900. But if he buys only second section, he will pay 300 for it plus 400 for toll for first route (second route goes only through second section).

One infamous online multiplayer video game is currently being played by **N** people. The goal of the game is to move around the map and shoot other players with water guns of all sorts. Every player can move with certain **speed** and has a single gun of determined **range**. We can say that player A is **much better** than player B if **both of these parameters** (speed and gun range) **are strictly greater** for player A than for player B.

To make the game more interesting, the programmers will add **K bots** into the game. Bots are controlled by a computer and participate in the game in the same manner as normal players. Each bot is determined by same two parameters as normal players, but to make them distinguishable, they all must have **distinct speeds and distinct gun ranges** (this doesn't include players). In order for bots not to be overpowered, **for each bot** there must be a **normal player** that is **much better** than that bot.

Before the bots are added, the players are waiting for **one more normal player** to join the game. Your task is that for each of **Q** candidate new players determine the following: if that player joins the **N** existing players, on how many distinct ways can you select and add K bots to the game? Two bot selections are considered different if there is a single bot in one selection that doesn't exist in the other selection. This number can be quite huge, so write it modulo 10 009.

INPUT DATA

First line contains two integers **N** and **K** ($2 \leq \mathbf{N} \leq 100\,000$, $1 \leq \mathbf{K} \leq 30$), the number of players and the number of bots.

The following **N** lines contain parameters for the players, $\mathbf{V}_i$ and $\mathbf{R}_i$ ($1 \leq \mathbf{V}_i, \mathbf{R}_i \leq 100\,000$), speed and gun range for player i.

Next line contains one integer **Q** ($1 \leq \mathbf{Q} \leq 100\,000$), number of candidate players that could join the game.

Last **Q** lines contain game parameters for the candidate players, $\mathbf{V}_i$ and $\mathbf{R}_i$ ($1 \leq \mathbf{V}_i, \mathbf{R}_i \leq 100\,000$), speed and gun range for candidate player i.

OUTPUT DATA

You need to output a total of **Q** lines, one for each candidate player.

For each candidate player **J** output one line containing the number of ways (**modulo 10009**) to select **K** bots if that candidate player would join the other **N** players.

SCORING

In test cases worth 44% of total points **K** is 2.

In test cases worth 55% of total points **Q** is 1.

In test cases worth 77% of total points, at least one of these two conditions will hold.

TEST EXAMPLES

ulaz	ulaz
2 1 2 5 3 3 1 4 2 izlaz 7	2 2 5 2 3 5 2 5 5 1 3 izlaz 72 24

Explanation of sample #1: Speed and gun range of the bot can be any of the following: (1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (2, 2) and (3, 1).