NOI 2017 TASKS

Tasks
Task 1: Best Place
Task 2: Roadside Advertisements
Task 3: Hotspot
Task 4: RMQ
Task 5: I want to be the very best too!

Notes:

1. This document consists of 16 pages excluding this page.
2. Each task is worth 100 marks.
3. For each task, your program will be tested on a few sets of input instances. We call each set a subtask. Each subtask is worth a few marks.
4. The subtasks vary in size and complexity, leading to different levels of difficulty. If you find solving the task completely difficult, you may want to focus on the easier subtasks. The first subtask is expected to be very easy.
5. The maximum execution time on each input instance in Tasks 1, 2, 3, 4 & 5 are 1.0, 1.0, 2.5, 1.0 & 5.0 seconds, respectively, and the memory size is limited to 512MB for all tasks.

HAPPY PROGRAMMING!

Task 1: `Best Place`

IOI 2020 will be held in Singapore, hosted by School of Computing of the National University of Singapore (NUS), supported by Singapore Ministry of Education (MOE) and Singapore Exhibition & Convention Bureau (SECB).

One of the challenges of hosting such a large event is to find a suitable location for the event. A common suggestion would be to select a location that reduces the travelling distance for the participants. You are thus tasked to write a program to find one such location.

Suppose there are N participants for IOI 2020. Assume the i^{th} participant's house is located on (X_i, Y_i) on the cartesian plane. You aim to find a location to hold IOI 2020 such that the total sum of travelling distance is minimized. Assume IOI 2020 is chosen to be at (X, Y), the travelling distance for the i^{th} participant is the Manhattan distance between (X, Y) and (X_i, Y_i), i.e., $|X - X_i| + |Y - Y_i|$.

If there are multiple locations where the total sum of travelling distance is minimized, you are to output any one of these coordinates. It is possible for the location of the event be the same as the location of a participant's house. However, the event must be held at a position (X, Y) where X and Y are integers. It is also possible that more than 1 participant lives at the same coordinates. In that case, you are to sum their travelling distance separately. (Assume they will travel separately.)

Input

Your program must read from standard input.

The input will start with a single integer, N, in a single line. N denotes the total number of participants.

N lines will then follow with 2 integers each, the i^{th} line will contain X_i and Y_i. This indicates that the coordinate of the i^{th} participant's house is at (X_i, Y_i).

Output

Your program must output to standard output only.
Output 2 integers (X, Y) on the same line, space separated. (X, Y) will denote the coordinates of the event where by the total sum of travelling distance would be minimized.

Subtasks

The maximum execution time on each instance is 1.0s. Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	N	X_i, Y_i
1	3	$N = 2$	$0 \leq X_i, Y_i \leq 10^9$
2	20	$2 \leq N \leq 1\,000$	$0 \leq X_i \leq 1\,000, Y_i = 0$
3	28	$2 \leq N \leq 100\,000$	$0 \leq X_i \leq 10^9, Y_i = 0$
4	13	$2 \leq N \leq 100$	$0 \leq X_i, Y_i \leq 100$
5	17	$2 \leq N \leq 1\,000$	$0 \leq X_i, Y_i \leq 10^9$
6	19	$2 \leq N \leq 100\,000$	$0 \leq X_i, Y_i \leq 10^9$

Sample Testcase 1

This testcase is valid for all subtasks.

Input	Output
2 1 0 4 0	3 0

Note that this is not the only output that would be accepted. In particular, the following alternative outputs will also be accepted for Sample Testcase 1.

- 1 0
- 2 0
- 4 0

Sample Testcase 1 Explanation

Regardless of whether we choose $(1, 0), (2, 0), (3, 0)$ or $(4, 0)$, the sum of total travelling distance will still be 3.

Sample Testcase 2

This testcase is only valid for subtasks 2 to 6.

Input	Output
6 1 0 3 0 5 0 7 0 9 0 11 0	7 0

Note that this is not the only output that would be accepted. In particular, the following alternative outputs will also be accepted for Sample Testcase 2.

- `5 0`
- `6 0`

Sample Testcase 3

This testcase is only valid for subtasks 4 to 6.

Input	Output
9 1 16 3 12 5 6 7 10 9 8 11 4 13 14 15 2 17 18	9 10

Note that this is the only output that is accepted for Sample Testcase 3.

Task 2: `Roadside Advertisements`

Singapore National Olympiad in Informatics (SG NOI) celebrates its 20-th version in year 2017 (this year). This year, we are grateful to have FIVE (5) different sponsors (in alphabetical order): Garena, IMDA, Lee Foundation, Micron, and the organizer+sponsor: School of Computing (SoC).

Associate Professor Tan Sun Teck, the SG NOI chairman, wants to put roadside advertisements on roads connecting these 5 sponsors. He would like to ensure that any person affiliated with SG NOI will see an SG NOI-related advertisement on every road they traverse whenever they travel between their offices (e.g. for the important weekly NOI meetings). Thus, any road on the shortest path connecting any pair of the 5 sponsors should have a roadside advertisement installed.

For each road in Singapore, the cost of putting an advertisement on that road is known in advance. The overall cost of this advertising campaign is the sum of these costs. Prof Tan wants to find the minimum cost required to achieve his objective.

By the way, the roads in *this version of Singapore* have an interesting property where there is exactly one possible path between any two landmarks in Singapore.

Input format

Your program must read from standard input.
The first line of input contains one positive integer V.
The next $V-1$ lines of input will each contain 3 integers: u, v, w that denotes that landmark u and landmark v in Singapore are connected with a road and installing a roadside advert in this road costs w SGD.
It is guaranteed that $0 \leq u, v < V$ and $1 \leq w \leq 1\,000$.
Then, there will be another line with one positive integer Q, denoting the number of queries.
Afterwards, there will be Q lines with 5 integers each, denoting the location of the 5 sponsors of SG NOI: landmarks $\{a, b, c, d, e\}$.
It is guaranteed that a, b, c, d and e are pairwise different.
A possible input would be:

```
6
4 0 4
0 1 2
1 3 9
3 5 1
3 2 5
2
4 0 3 5 2
0 4 1 3 5
```

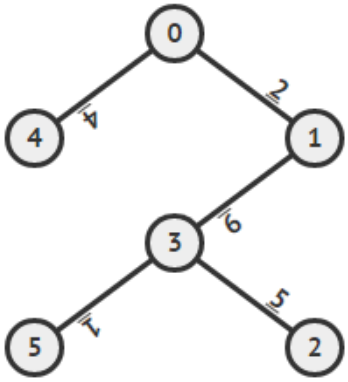

Figure 1: Sample Fictional Singapore with $V = 6$

Output format

For each query, your program must output one line with a single integer to standard output: the minimum cost for Prof Tan's roadside advertisement campaign. On the above example, the answer is:

```
21
16
```

Explanation

For the first query, the 5 sponsors are located at landmarks $\{4, 0, 3, 5, 2\}$ (notice that the landmarks are not necessarily sorted). To connect all pairs of the 5 sponsors in this fictional Singapore, we need to use all roads. Therefore we need to place a roadside advertisement on every road. The minimum cost is thus the sum of costs for all roads: $4 + 2 + 9 + 1 + 5 = 21$.

For the second query, the 5 sponsors are located at landmarks $\{0, 4, 1, 3, 5\}$. This time, Prof Tan can choose not to place a roadside advertsement along road $3 - 2$ with cost 5 (as landmark 2 is NOT an SG NOI sponsor). So the total cost is $21 - 5 = 16$.

Subtasks

The maximum execution time on each instance is 1.0s. Your program will be tested on sets of input instances as follows:

Subtask	Marks	V	Others
1	7	$V = 5, Q = 1$	
2	23	$5 \leq V \leq 50\,000, 1 \leq Q \leq 10\,000$	A landmark connects at most two other landmarks.
3	40	$5 \leq V \leq 50\,000, 1 \leq Q \leq 100$	
4	30	$5 \leq V \leq 50\,000, 1 \leq Q \leq 10\,000$	All the best!

Sample Testcase 1

This testcase is valid for all subtasks.

Input	Output
5 0 1 1 1 2 2 2 3 3 3 4 4 1 4 0 3 1 2	10

Sample Testcase 2

This testcase is valid for subtasks 3 and 4.

Input	Output
6 4 0 4 0 1 2 1 3 9 3 5 1 3 2 5 2 4 0 3 5 2 0 4 1 3 5	21 16

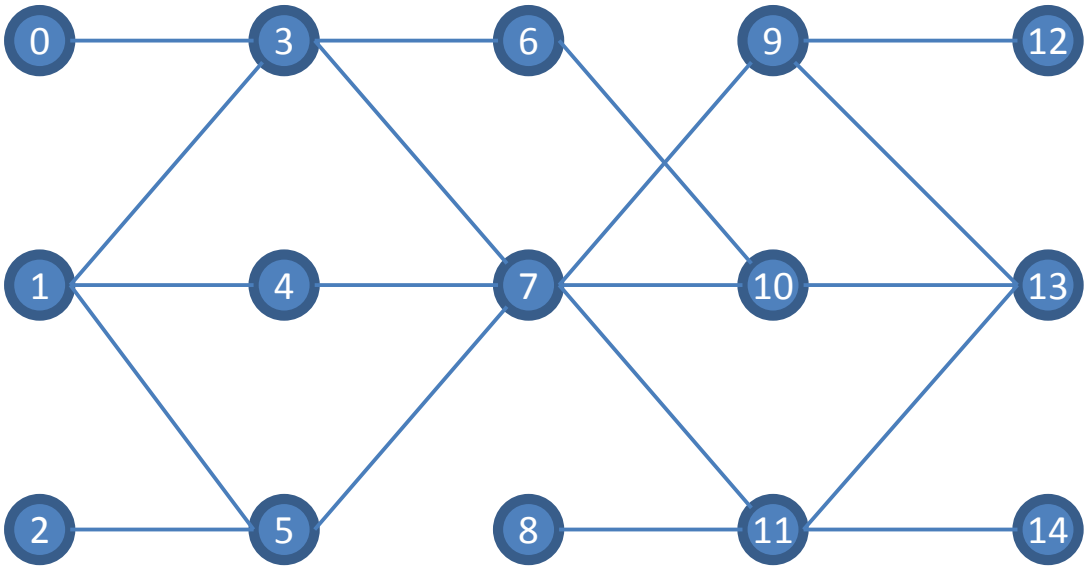

Figure 2: An example map with 15 towns and 19 roads.

Task 3: `Hotspot`

Consider a country with n towns. The towns are connected by m roads, all with the same length. (See Figure 2 for an example.)

This country has k citizens. Interestingly, the home and office of the ith citizen are located in different towns A_i and B_i, respectively. Hence, every day, the ith citizen moves back and forth between two fixed towns A_i and B_i (since the ith citizen needs to work).

To save time, the ith citizen will choose a path with the shortest length. If there are several shortest paths between A_i and B_i, the ith citizen chooses by random one shortest path which Donald does not know. The expected chance that the ith citizen passing through town w equals

$$E_i(w) = \frac{\text{Number of shortest paths between } A_i \text{ and } B_i \text{ passing through town } w}{\text{Number of shortest paths between } A_i \text{ and } B_i}.$$

Donald is the president of the country and he wants to understand the needs of his citizens. He wants to setup a meeting office in a town so that he can meet as many citizens as possible. Precisely, he aims to setup the meeting office in a town w that maximizes $\sum_{i=0}^{k-1} E_i(w)$.

Your task is to help Donald identify the town w. When there are multiple towns w that maximize $\sum_{i=0}^{k-1} E_i(w)$, report any one of them. In addition, due to certain geological constraints during the construction of the towns and roads, it is found that the number of shortest paths between any two towns will not exceed 2^{15}. Note that **double-precision floating-point** is required for this problem.

Example 1: Suppose $k = 1$ where $(A_0, B_0) = (4, 10)$. Then, there is exactly one shortest path of length 2, which is $(4, 7, 10)$. Donald can build the meeting office in either town 4, town 7, or town 10.

Example 2: Suppose $k = 2$ where $(A_0, B_0) = (4, 10)$ and $(A_1, B_1) = (3, 8)$. Then, there is exactly one shortest path between town 4 and town 10 of length 2, which is $(4, 7, 10)$. Also, there is exactly one shortest path between town 3 and town 8 of length 3, which is $(3, 7, 11, 8)$. If Donald builds the meeting office in town 7, the expected number of citizens passing through town 7 is 2.

Example 3: Suppose $k = 2$ where $(A_0, B_0) = (1, 13)$ and $(A_1, B_1) = (6, 2)$. Then, we have: (1) 10 shortest paths of lenght 4 between town 1 and town 13. (2) 3 shotest paths of lenght 3 between town 6 and town 2. If Donald builds the meeting office in town 7, we have: (1) For the 0th citizen, 9 shortest paths between town 1 and town 13 passing through town 7. (2) For the 1st citizen, 2 shortest paths between town 6 and town 2 passing through town 7. Then, the expected number of citizens passing through town 7 is $E_0(7) + E_1(7) = 9/10 + 2/3 = 1.567$. In fact, this is the best solution. Donald needs to build the meeting office in town 7.

Input

Your program must read from standard input.

The input will start with two integers n and m in a single line. n denotes the number of towns while m denotes the number of edges. Then, the next m lines are the roads, each consisting of two integers representing the two towns connected by the road.
Afterwards, the next line contains an integer k, which denotes the number of citizens. It is followed by k lines. The ith line stores two integers A_i and B_i, for $i = 0, \ldots, k - 1$.
For example 2, the input is as follows:

```
15 19
0 3
1 3
1 4
1 5
2 5
3 6
3 7
4 7
5 7
6 10
7 9
7 10
7 11
8 11
9 12
9 13
10 13
11 13
```

```
11 14
2
4 10
3 8
```

Output

Your program must output to standard output only. Output a single line with exactly one integer, representing the town w that maximises $\sum_{i=0}^{k-1} E_i(w)$. When there are multiple possible towns, output any one of them.
For example 2, the output is

```
7
```

Subtasks

The maximum execution time on each instance is 2.5s. Your program will be tested on sets of input instances that satisfy the following restrictions:

Subtask	**Marks**	Criteria
1	4	The map is a straight line, $n \leq 1000$, $m = n - 1$, $k = 1$
2	5	The map is a tree, $n \leq 1000$, $m = n - 1$, $k = 1$
3	11	The map is a straight line, $n \leq 1000$, $m = n - 1$, $k \leq 200$
4	18	The map is a tree, $n \leq 1000$, $m = n - 1$, $k \leq 200$
5	26	$n \leq 1000$, $m \leq 8000$, $k \leq 20$
6	36	$n \leq 5000$, $m \leq 40000$, $k \leq 2000$

Sample Testcase 1

This testcase is only valid for subtasks 5 and 6.

Input	Output
5 5 0 1 1 2 2 3 3 4 4 0 2 1 3 2 4	2 or 3

Sample Testcase 2

This testcase is only valid for subtasks 3, 4, 5 and 6.

Input	Output
5 4 0 1 1 2 2 3 3 4 3 0 2 1 3 2 4	2

Sample Testcase 3

This testcase is only valid for subtasks 4, 5 and 6.

Input	Output
6 5 0 2 1 2 2 3 3 4 3 5 2 0 5 1 4	2 or 3

Task 4: RMQ

Kraw the Krow has N cows. Each cow wears a tag containing a **unique** number from 0 to $N-1$. The cows are lined up in a row in some unknown order. The positions of the cows are labelled from 0 to $N-1$ in order.

For reasons that we will probably never know, Kraw found the need to answer an embarrassingly large number of Range Minimum Queries (RMQs). RMQs are questions of the form, "What is the smallest tag number of the cows standing at or between positions L and R?"

Kraw is very lazy, so he paid Coco the Code Monkey less than minimum wage to solve the RMQs for him. Coco completed all of them four minutes before the deadline, and Kraw was very pleased.

That was in 2013. Now, Kraw is making a big loss in his Cow Tagging conglomerate and is starting to doubt the authenticity of Coco's work. For all he knew, Coco could have just randomly generated answers to all the RMQs.

Unfortunately, after so many years, all the cows have dispersed and Coco is suspisciously uncontactable. Help Kraw check if there exists any ordering of cows such that all of Coco's answers are valid.

Input format

Your program should read the input from standard input. The input consists of:

- one line with two integers N ($1 \leq N \leq 100\,000$), the number of cows, and Q ($1 \leq Q \leq 100\,000$), the number of RMQs;
- Q lines, with the i-th line containing three integers: L_i and R_i ($0 \leq L_i \leq R_i < N$), the left and right bounds of the i-th RMQ, and A_i ($0 \leq A_i < N$), Coco's answer to the RMQ.

Output format

Output one line containing N space-separated integers: any possible ordering of cows such that A_i is the correct answer to the RMQ $[L_i, R_i]$ for all i, and no two cows share the same tag number.

If there does not exist any such ordering of cows, fill all N integers with -1.

Subtasks

The maximum execution time on each instance is 1.0s. Your program will be tested on sets of input instances that satisfy the following restrictions:

Subtask	Marks	N	Q
1	23	$N \leq 10$	$Q \leq 10$
2	44	$N \leq 1\,000$	$Q \leq 1\,000$
3	33	No additional constraints.	

For each test case, if your program's output is not a permutation (that is, it contains repeated tag numbers) but satisfies all the RMQs, you will still be eligible for **30%** of that subtask's score.

Sample Testcase 1

Input	Output
5 3 0 2 1 1 3 0 1 4 0	1 4 3 0 2

Note that this is not the only output that would be accepted.

Sample Testcase 1 Explanation

First, we note that our proposed ordering $[1, 4, 3, 0, 2]$ is indeed a permutation (that is, every integer from 0 to 4 appears exactly once). Then, we check the RMQs:

- The first RMQ ($L = 0$, $R = 2$) refers to the subarray $[1, 4, 3]$. The smallest tag number within that range is 1.
- The second RMQ refers to the subarray $[4, 3, 0]$. The smallest tag number within that range is 0.
- The third RMQ refers to the subarray $[4, 3, 0, 2]$. The smallest tag number within that range is 0.

Since all answers coincide with Coco's answers, $[1, 4, 3, 0, 2]$ is indeed a valid solution.

Sample Testcase 2

Input	Output
3 2 0 1 1 1 2 1	-1 -1 -1

Sample Testcase 2 Explanation

If the 0-th cow were in position 0 or 1, the answer to the first RMQ would be 0 instead of 1. If the 0-th cow were in position 2, the answer to the second RMQ would be 0 instead of 1. Thus, it is impossible to order the cows such that all RMQs are satisfied.

Note that the following output would still be valid for 30% of the subtask's score:

```
1 1 1
```

Task 5: `I want to be the very best too!`

Having seen Ash Ketchum's success in becoming the very best, Mr. Panda wants to follow his footsteps and becomes the very best too! To do so, he wants to catch as many different types of Pokémon as possible to fill up his Pokédex and become the very best. However, there are many other Pokémon trainers in his way and he has to defeat them in order to reach his goal.

Mr. Panda and the Pokémon live in a world which can be represented by a $R \times C$ grid with R rows and C columns and Mr. Panda can only go from one square to another if they are adjacent. The top left square is labelled as $(1, 1)$ and the bottom right square is labelled as (C, R). Each grid square has a Pokémon trainer that Mr. Panda will have to battle in order to pass through that grid square and obtain the Pokémon that is owned by that trainer. Each Pokémon trainer has his/her own level relative to the other trainers which determines which Pokémon trainer is the best. We assume all Pokémon trainers have different levels (to ensure a clear winner for every battle). The Pokémon trainer in square (j, i) has level L_{ij} and he/she uses a Pokémon of type P_{ij} for the battle.

To make things even more difficult, the world is constantly changing so the Pokémon trainers will sometimes change the type of Pokémon they battle with in order to increase their chances of winning. Because of this, Mr. Panda wants to plan when is the best time to begin his expedition. At certain points in time, he wants to know how many types of Pokémon he will be able to obtain if he starts off at square (X_q, Y_q) and is only able to defeat Pokémon trainers of level at most L_q. Remember, he cannot pass through a grid square if he cannot defeat the Pokémon trainer there. Defeating a Pokémon type more than once with different Pokémon trainers will only count as one type of Pokémon.

Input

Your program must read from standard input.
The first line of input contains three positive integers R, C and Q. It is guaranteed that $R \times C \leq 50000$.
It is followed by $2R + Q$ lines.

- The first R lines of input will each contain C integers. The j-th integer on the i-th line represents L_{ij}. It is guaranteed that $1 \leq L_{ij} \leq 10^9$ and that all the values will be distinct.
- The next R lines of input will each contain C integers. The j-th integer on the i-th line represents P_{ij}. It is guaranteed that $1 \leq P_{ij} \leq 50000$.
- The last Q lines contain 4 integers each, representing a query.
  - If the first integer of the line is 1, it represents a type 1 query and the next 3 integers represent X_q, Y_q, P_q. This means that the trainer at (X_q, Y_q) changed to use Pokémon of type P_q instead. It is guaranteed that $1 \leq P_q \leq 50000$.

– If the first integer of the line is 2, it represents a type 2 query and the next 3 integers represent X_q, Y_q, L_q. This means that Mr. Panda wants to know, if he starts at square (X_q, Y_q) and is only able to defeat Pokémon trainers of level at most L_q, how many different types of Pokémon he will be able to obtain. It is guaranteed that $1 \leq L_q \leq 10^9$.

Output

Your program must output to standard output only.
For each query of type 2, output one line with a single integer, the number of types of Pokémon Mr. Panda will be able to obtain.

Subtasks

The maximum execution time on each instance is 5.0s. Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	R	Q	Others
1	11	$R = 1$	$1 \leq Q \leq 1000$	The trainer at $(X, 1)$ will be of level X.
2	16	$1 \leq R \leq 50000$	$1 \leq Q \leq 10$	-
3	20	$1 \leq R \leq 50000$	$1 \leq Q \leq 100000$	$1 \leq P_{ij}, P_q \leq 2$
4	24	$R = 1$	$1 \leq Q \leq 100000$	The trainer at $(X, 1)$ will be of level X.
5	29	$1 \leq R \leq 50000$	$1 \leq Q \leq 100000$	-

Sample Testcase 1

This testcase is valid for all subtasks.

Input	Output
1 5 5 1 2 3 4 5 1 1 2 1 2 2 2 1 2 2 1 1 4 2 4 1 3 1 3 1 1 2 3 1 4	1 2 0 1

Sample Testcase 2

This testcase is only valid for subtasks 1, 2, 4 and 5.

Input	Output
1 5 5 1 2 3 4 5 3 4 3 2 5 2 3 1 3 2 1 1 5 2 4 1 2 1 4 1 4 2 3 1 5	2 4 0 3

Sample Testcase 3

This testcase is only valid for subtasks 2, 3 and 5.

Input	Output
3 3 5 1 4 3 11 2 7 5 10 6 1 1 1 2 1 2 1 2 1 2 2 1 6 2 2 3 10 2 3 2 3 1 2 2 2 2 2 1 4	1 2 0 2

Sample Testcase 4

This testcase is only valid for subtasks 2 and 5.

Input	Output
3 3 5 1 4 3 11 2 7 5 10 6 6 3 3 4 6 4 9 4 9 2 2 1 6 2 2 3 10 2 3 2 3 1 2 2 7 2 2 1 4	2 4 0 3

Explanation for Sample Testcase 4

For the first type 2 query, Mr. Panda can only catch Pokemon of types 3 and 6 by defeating the Pokemon trainers with levels 1, 2, 3 and 4.
For the second type 2 query, Mr. Panda can catch all types of Pokemon as he can defeat all Pokemon trainers except the one with level 11.
For the third type 2 query, Mr. Panda cannot defeat the trainer at the starting square so he cannot go anywhere and thus cannot catch any Pokemon.
For the fourth type 2 query, Mr. Panda can catch 3 types of Pokemon since the trainer at $(2, 2)$ is now using a Pokemon of type 7.