The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – Abduction 2

Abduction 2

On a sunny day, there was an abduction incident at a crossing in a city. It is suspected that the criminals were Anna and Bruno, and they escaped from the abduction scene by a car. The car was not yet found. The police is searching it even now.

The urban area where the criminals was driving the car is a city of rectangular grid with H streets running in the east-west direction and W streets running in the south-north direction. The distance between two adjacent crossings is 1 kilometer.

Each street has an integer, the *congestion degree*. The congestion degree of the i-th street ($1 \le i \le H$) from north running in the east-west direction is A_i, and the congestion degree of the j-th street ($1 \le j \le W$) from west running in the south-north direction is B_j. These $H + W$ values are different from each other. For each street, the congestion degree has the same value at any points on it.

The investigation by the police revealed that the criminals moved in the city in the following way:

- They did not go out of the city, and they did not go out of the streets.

- In the beginning, the criminals chose one of the movable directions from the abduction scene, and moved toward that direction.

- When they came to a crossing, if the congestion degree of the transversing street is larger than that of the current street, they turned at that crossing. If it was possible to turn to the both directions, they could choose any one of them.

- When they came to a crossing, if the congestion degree of the current street is larger than that of the transversing street, they kept going straight. However, if they were in the boundary of the city and could not go straight, they stopped moving at that place.

There are Q candidates of crossings for the abduction scene. These Q candidates are different from each other. To determine the number of investigators, the police wants to know, for each candidate of crossings, the maximum possible travel distance for the criminals assuming the abduction incident was occurred at that place.

For each of Q queries, calculate the maximum possible travel distance from the given candidate of crossings.

Task

Given the congestion degrees of the streets in the city and Q candidates of crossings for the abduction scene, write a program which calculates the maximum possible travel distance from each candidate of crossings.

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – Abduction 2

Input

Read the following data from the standard input.

- The first line of input contains three space separated integers H, W, Q. This means the city is a rectangular grid with H streets running in the east-west direction and W streets running in the south-north direction, and there are Q candidates of crossings for the abduction scene.

- The second line of input contains H space separated integers $A_1, A_2, \ldots, A_H$. This means the congestion degree of the i-th street ($1 \leq i \leq H$) from north running in the east-west direction is A_i.

- The third line of input contains W space separated integers $B_1, B_2, \ldots, B_W$. This means the congestion degree of the j-th street ($1 \leq j \leq W$) from west running in the south-north direction is B_j.

- The k-th line ($1 \leq k \leq Q$) of the following Q lines contains two space separated integers S_k, T_k. This means the k-th candidate of crossings for the abduction scene is the crossing of the S_k-th street from north running in the east-west direction and the T_k-th street from west running in the south-north direction.

Output

Write Q lines to the standard output. The k-th line of output should contain an integer, the maximum possible travel distance (in kilometers) from the k-th candidate of crossings for the abduction scene.

Constraints

All input data satisfy the following conditions.

- $2 \leq H \leq 50\,000$.

- $2 \leq W \leq 50\,000$.

- $1 \leq Q \leq 100$.

- $1 \leq A_i \leq 1\,000\,000\,000$ ($1 \leq i \leq H$).

- $1 \leq B_j \leq 1\,000\,000\,000$ ($1 \leq j \leq W$).

- The $H + W$ integers $A_1, A_2, \ldots, A_H, B_1, B_2, \ldots, B_W$ are different from each other.

- $1 \leq S_k \leq H$ ($1 \leq k \leq Q$).

- $1 \leq T_k \leq W$ ($1 \leq k \leq Q$).

- $(S_k, T_k) \neq (S_l, T_l)$ ($1 \leq k < l \leq Q$).

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – Abduction 2

Subtask

There are 5 subtasks. The score and additional constraints of each subtask are as follows:

Subtask 1 [13 points]

- $H \leq 8$.

- $W \leq 8$.

- $Q = 1$.

Subtask 2 [10 points]

- $H \leq 2\,000$.

- $W \leq 2\,000$.

- $Q = 1$.

Subtask 3 [17 points]

- $Q = 1$.

Subtask 4 [4 points]

- $H \leq 2\,000$.

- $W \leq 2\,000$.

Subtask 5 [56 points]

There are no additional constraints.

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – Abduction 2

Sample Input and Output

Sample Input 1	Sample Output 1
3 3 5	4
3 2 6	5
1 4 5	4
1 1	4
1 2	2
2 2	
3 1	
3 3	

For example, for the third query, the travel distance achieves the maximum value if the criminals moved in the following way:

- They moved east for 1 kilometer from the crossing of the second street from north and the second street from west.

- They could move south or north from the crossing of the second street from north and the third street from west. They chose to move south for 1 kilometer.

- They could move west only from the crossing of the third street from north and the third street from west. They moved west for 1 kilometer.

- They could move west only from the crossing of the third street from north and the second street from west. They moved west for 1 kilometer.

- They could not move from the crossing of the third street from north and the first street from west. They stopped moving at that place.

If they moved as above, the travel distance is 4 kilometers.

Sample Input 2	Sample Output 2
4 5 6	7
30 10 40 20	6
15 55 25 35 45	9
1 3	4
4 3	6
2 2	9
4 1	
2 5	
3 3	

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – City

City

There are many cities in the Kingdom of JOI. The road network satisfies the following conditions:

- (Condition 1)   The cities are numbered from 0 to $N-1$. Here, N is the number of cities in the Kingdom of JOI.

- (Condition 2)   The cities are connected by $N-1$ roads. We can pass each road in both directions. We can travel from every city to any other cities if we pass through several roads.

- (Condition 3)   We can travel from the city 0 to every city by passing at most 18 roads.

Every day, in the Kingdom of JOI, many people depart from the city 0 to other cities. Because many people have two destinations, they sometimes ask the following queries: for two different cities X, Y, which one of (0), (1), (2) is satisfied?

(0) We necessarily pass the city Y when we travel from the city 0 to the city X.

(1) We necessarily pass the city X when we travel from the city 0 to the city Y.

(2) Neither (0) nor (1).

Note that, in the above situation, exactly one of (0), (1), (2) is satisfied. When $X = 0$, we consider (1) is satisfied regardless of the value of Y. Similarly, when $Y = 0$, we consider (0) is satisfied regardless of the value of X.

It is known that, just like the road network in the Kingdom of JOI, the above Conditions 1–3 are satisfied in other countries as well. In the Kingdom of JOI, people plan to develop the following two machines so that they can be used also in other countries.

- (Machine 1)   Given the number of cities N and information of the road network, this machine assigns a code to each city. A code is an integer between 0 and $2^{60} - 1$, inclusive.

- (Machine 2)   Given the codes of two different cities X, Y assigned by Machine 1, this machine answers the query.

If large integers are assigned as codes, it is difficult to treat them. We want to develop machines so that smaller values are assigned as codes.

Note that, when Machine 2 is used, neither the number of cities N nor information of the road network is directly given to the machine.

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – City

Task

In order to develop two machines as above, write the following two programs:

- Given the number of cities N in the Kingdom of JOI and information of the road network, the first program assigns a code to each city.

- Given the codes of two different cities assigned by the first program, the second program answers the query for these cities.

Implementation Details

You need to submit two files written by the *same programming language*.

The first file is either `Encoder.c` or `Encoder.cpp`. This file implements Machine 1. It should implement the following function. The program should include `Encoder.h`.

- `void Encode(int N, int A[], int B[])`

  For each test case, this function is called once.

  - The parameter `N` is the number of cities N.
  - The parameters `A[]`, `B[]` are integer sequences of length $N - 1$ describing the road network. The elements `A[i]`, `B[i]` ($0 \le i \le N - 2$) mean there is a road connecting the city `A[i]` and the city `B[i]` directly.

  Your program should call the following function.

  ★ `void Code(int city, long long code)`
  This function assigns codes to the cities.

    ◇ The parameter `city` is the city to which a code is assigned. The parameter `city` should be an integer between 0 and $N - 1$, inclusive. If the call to this function has parameters outside this range, your program is considered as **Wrong Answer[1]**. It should not be called twice with the same parameters. If it is called twice with the same parameters, your program is considered as **Wrong Answer[2]**.
    ◇ The parameter `code` is the code assigned to the city whose number is `city`. The parameter `code` should be an integer between 0 and $2^{60} - 1$, inclusive. If the call to this function has parameters outside this range, your program is considered as **Wrong Answer[3]**.

  Your program should call the function `Code` exactly N times. If the number of calls to `Code` is not N when the function `Encode` is terminated, your program is considered as **Wrong Answer[4]**.

  If the call to the function `Encode` is considered Wrong Answer, your program is terminated immediately.

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – City

The second file is either `Device.c` or `Device.cpp`. This file implements Machine 2. It should implement the following function. The program should include `Device.h`.

- `void InitDevice()`

  This function initializes Machine 2. The function `InitDevice` is called once before calls to the following function `Answer`.

- `int Answer(long long S, long long T)`

  This function answers each query. For each query, the function `Answer` is called once.

    ◦ The parameters `S`, `T` are the codes assigned to two different cities X, Y by the function `Encode`.

    ◦ To answer the query, the return value of the function `Answer` should satisfy the following conditions:

      ⋄ The return value should be `0` if we necessarily pass the city Y when we travel from the city 0 to the city X.

      ⋄ The return value should be `1` if we necessarily pass the city X when we travel from the city 0 to the city Y.

      ⋄ Otherwise, the return value should be `2`.

    Hence the return value of the function `Answer` should be an integer between 0 and 2, inclusive. If the return value is outside this range, your program is considered as **Wrong Answer[5]**. If the return value is inside this range but it does not satisfy the above conditions, your program is considered as **Wrong Answer[6]**.

Grading Procedure

The grading is done in the following way. If your program is considered as Wrong Answer, it is terminated immediately.

(1) The function `Encode` is called once.

(2) The function `InitDevice` is called once.

(3) For each test case, there are Q queries given to Machine 2. For the j-th query ($1 \leq j \leq Q$), the function `Answer` is called whose parameter `S` is S_j and parameter `T` is T_j, where S_j, T_j are the codes assigned to the cities X_j, Y_j by the function `Encode`, respectively.

(4) Your program is considered as correct.

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – City

Important Notices

- Running time and memory usage are calculated for (1), (2), (3) of Grading Procedure. Note that, in the procedure (3), the function `Answer` is called Q times in total.

- Your program must not be considered as Wrong Answer in the function call to `Encode` in the procedure (1), `InitDevice` in the procedure (2), or `Answer` in the procedure (3). Your program must be executed without runtime error.

- Your program can implement other functions for internal use, or use global variables. Submitted programs will be compiled with the grader, and become a single executable file. All global variables and internal functions should be declared `static` to avoid confliction with other files. The programs of Machine 1 and Machine 2 will be executed as two distinct processes when they are graded. They can not share global variables.

- Your program should not use standard input and standard output. Your program should not communicate with other files by any methods.

Compilation and Test Run

You can download an archive file from the contest webpage which contains a sample grader to test your program. The archive file also contains a sample source file of your program.

A sample grader consists of one source file which is either `grader.c` or `grader.cpp`. For example, if your programs are `Encoder.c` and `Device.c`, or `Encoder.cpp` and `Device.cpp`, you run the following commands to compile your programs.

- C

  ```
  gcc -std=c11 -O2 -o grader grader.c Encoder.c Device.c -lm
  ```

- C++

  ```
  g++ -std=c++14 -O2 -o grader grader.cpp Encoder.cpp Device.cpp
  ```

When the compilation succeeds, the executable file `grader` is generated. Note that the actual grader is different from the sample grader. The sample grader will be executed as a single process, which will read input data from the standard input and write the results to the standard output.

Input for the Sample Grader

The sample grader reads the following data from the standard input.

- The first line contains two space separated integers N, Q. This means there are N cities, and Q queries are given.

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – City

- The $(i + 1)$-th line $(0 \le i \le N - 2)$ of the following $N - 1$ lines contains two space separated integers A_i, B_i. This means there is a road connecting the city A_i and the city B_i directly.

- The j-th line $(1 \le j \le Q)$ of the following Q lines contains three space separated integers X_j, Y_j, E_j. This means $X = X_j$ and $Y = Y_j$ in the j-th query, and your program will be considered Wrong Answer by the sample grader if the answer to this query is different from E_j.

Output of the Sample Grader

When the program terminates successfully, the sample grader writes the following information to the standard output. (The quotation mark is not written actually.)

- If your program is considered as correct, the sample grader writes the maximum of the codes assigned to the cities in the following form "`Accepted : max_code=123456`."

- If your program is considered as Wrong Answer, the sample grader writes its type in the following form "Wrong Answer [1]."

If your program is considered as several types of Wrong Answer, the sample grader reports only one of them.

Constraints

All input data satisfy the following conditions. For the meaning of N, Q, A_i, B_i, X_j, Y_j, see Input for the Sample Grader.

- $2 \le N \le 250\,000$.

- $1 \le Q \le 250\,000$.

- $0 \le A_i \le N - 1$ $(0 \le i \le N - 2)$.

- $0 \le B_i \le N - 1$ $(0 \le i \le N - 2)$.

- $A_i \ne B_i$ $(0 \le i \le N - 2)$.

- We can travel from the city 0 to every city by passing at most 18 roads.

- $0 \le X_j \le N - 1$ $(1 \le j \le Q)$.

- $0 \le Y_j \le N - 1$ $(1 \le j \le Q)$.

- $X_j \ne Y_j$ $(1 \le j \le Q)$.

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – City

Subtask

There are 2 subtasks. The score and additional constraints of each subtask are as follows:

Subtask 1 [8 points]

- $N \leq 10$.

Subtask 2 [92 points]

There are no additional constraints. In this subtask, the score is calculated as follows:

- Let L be the maximum of the codes assigned to the cities in all test cases for this subtask.

- The score for this subtask is:

  - 0 point if $2^{38} \leq L$.
  - 10 points if $2^{36} \leq L \leq 2^{38} - 1$.
  - 14 points if $2^{35} \leq L \leq 2^{36} - 1$.
  - 22 points if $2^{34} \leq L \leq 2^{35} - 1$.
  - $\lfloor 372 - 10 \log_2(L + 1) \rfloor$ points if $2^{28} \leq L \leq 2^{34} - 1$. Here, $\lfloor x \rfloor$ is the largest integer not exceeding x.
  - 92 points if $L \leq 2^{28} - 1$.

  Note that, if $2^{38} \leq L$, the result for this subtask may be displayed as "Accepted : 0 points" or "Wrong Answer" in the contest system,

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – City

Sample Communication

Here is a sample input for sample grader and corresponding function calls.

Sample Input	Sample Calls	
6 5	Machine 1	Machine 2
4 1	Encode(...)	
0 3	Code(0,0)	
4 5	Code(2,4)	
3 2	Code(4,16)	
3 4	Code(1,1)	
2 4 2	Code(3,9)	
1 0 0	Code(5,25)	
5 1 2		InitDevice()
5 3 0		Answer(4,16)
4 1 1		Answer(1,0)
		Answer(25,1)
		Answer(25,9)
		Answer(16,1)

The function `Encode(...)` is called with the following parameters:

Parameter	Encode(...)
N	6
A	{4, 0, 4, 3, 3}
B	{1, 3, 5, 2, 4}

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – Dragon 2

Dragon 2

In the JOI plain, people are living with dragons.

The JOI plain is a wide coordinate plane with X and Y coordinates. The point with X-coordinate x and Y-coordinate y is denoted by (x, y).

There are N dragons in the JOI plain, numbered from 1 to N. There are M tribes of dragons. The tribes are numbered from 1 to M. The dragon i ($1 \leq i \leq N$) is always staying at the point (A_i, B_i) in the JOI plain, and its tribe is C_i. It is not always the case that dragons of all kinds live in the JOI plain.

In the JOI plain, there are two villages of human beings at the points $(D_1, E_1), (D_2, E_2)$. The two villages are connected by a road, which is a line segment connecting the two points of the villages.

The points $(A_1, B_1), \ldots, (A_N, B_N)$ and the points $(D_1, E_1), (D_2, E_2)$ are different from each other. No three of them lie on a straight line.

Sometimes, there are conflicts between tribes of dragons. If the tribe a ($1 \leq a \leq M$) gets hostile to the tribe b ($1 \leq b \leq M, a \neq b$), every dragon of tribe a launches balls of fire toward all dragons of tribe b. A ball of fire goes straight toward the target. After it gets to the target, it keeps going toward the same direction. Hence the track of a ball of fire is a half-line.

When conflicts between tribes occur, the road must be damaged if a ball of fire from a dragon comes across it. You have a list of Q possible conflicts between tribes of dragons which can occur in the near future. For each possible conflict, you want to calculate the number of balls of fire coming across the road.

Task

Given the information of the dragons and the villages of human beings and a list of possible conflicts between tribes of dragons, write a program which calculates, for each conflict, the number of balls of fire coming across the road.

Input

Read the following data from the standard input.

- The first line of input contains two space separated integers N, M. This means there are N dragons in the JOI plain, and there are M tribes of dragons.

- The i-th line ($1 \leq i \leq N$) of the following N lines contains three space separated integers A_i, B_i, C_i. This means the dragon i ($1 \leq i \leq N$) is staying at the point (A_i, B_i) in the JOI plain, and its tribe is C_i.

- The following line contains four space separated integers D_1, E_1, D_2, E_2. This means there are two villages of human beings at the points $(D_1, E_1), (D_2, E_2)$ in the JOI plain.

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – Dragon 2

- The following line contains an integer Q, the number of possible conflicts between tribes of dragons.

- The j-th line ($1 \leq j \leq Q$) of the following Q lines contains two space separated integers F_j, G_j. This means, in the j-th possible conflict, the tribe F_j gets hostile to the tribe G_j.

Output

Write Q lines to the standard output. The j-th line ($1 \leq j \leq Q$) of output contains the number of balls of fire coming across the road in the j-th conflict between tribes.

Constraints

All input data satisfy the following conditions.

- $2 \leq N \leq 30\,000$.

- $2 \leq M \leq N$.

- $-1\,000\,000\,000 \leq A_i \leq 1\,000\,000\,000$ ($1 \leq i \leq N$).

- $-1\,000\,000\,000 \leq B_i \leq 1\,000\,000\,000$ ($1 \leq i \leq N$).

- $1 \leq C_i \leq M$ ($1 \leq i \leq N$).

- $-1\,000\,000\,000 \leq D_1 \leq 1\,000\,000\,000$.

- $-1\,000\,000\,000 \leq E_1 \leq 1\,000\,000\,000$.

- $-1\,000\,000\,000 \leq D_2 \leq 1\,000\,000\,000$.

- $-1\,000\,000\,000 \leq E_2 \leq 1\,000\,000\,000$.

- The $N + 2$ points $(A_1, B_1), \ldots, (A_N, B_N), (D_1, E_1), (D_2, E_2)$ are different from each other. No three of them lie on a straight line.

- $1 \leq Q \leq 100\,000$.

- $1 \leq F_j \leq M$ ($1 \leq j \leq Q$).

- $1 \leq G_j \leq M$ ($1 \leq j \leq Q$).

- $F_j \neq G_j$ ($1 \leq j \leq Q$).

- $(F_j, G_j) \neq (F_k, G_k)$ ($1 \leq j < k \leq Q$).

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – Dragon 2

Subtask

There are 3 subtasks. The score and additional constraints of each subtask are as follows:

Subtask 1 [15 points]

- $N \le 3\,000$.

Subtask 2 [45 points]

- $Q \le 100$.

Subtask 3 [40 points]

There are no additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1
4 2	1
0 1 1	2
0 -1 1	
1 2 2	
-6 1 2	
-2 0 2 0	
2	
1 2	
2 1	

In the first conflict between tribes, the following are satisfied:

- The ball of fire launched by the dragon 1 toward the dragon 3 does not come across the road.

- The ball of fire launched by the dragon 1 toward the dragon 4 does not come across the road.

- The ball of fire launched by the dragon 2 toward the dragon 3 comes across the road.

- The ball of fire launched by the dragon 2 toward the dragon 4 does not come across the road.

Hence one ball of fire comes across the road.

In the second conflict between tribes, the following are satisfied:

- The ball of fire launched by the dragon 3 toward the dragon 1 comes across the road.

- The ball of fire launched by the dragon 3 toward the dragon 2 comes across the road.

The 16th Japanese Olympiad in Informatics (JOI 2016/2017)
Spring Training Camp/Qualifying Trial
March 19–25, 2017 (Komaba/Yoyogi, Tokyo)

Contest Day 4 – Dragon 2

- The ball of fire launched by the dragon 4 toward the dragon 1 does not come across the road.

- The ball of fire launched by the dragon 4 toward the dragon 2 does not come across the road.

Hence two balls of fire come across the road.

Sample Input 2	Sample Output 2
3 2	1
-1000000000 -1 1	
-999999998 -1 1	
0 0 2	
999999997 1 999999999 1	
1	
1 2	

Sample Input 3	Sample Output 3
6 3	4
2 -1 1	2
1 0 1	4
0 3 2	0
2 4 2	2
5 4 3	1
3 9 3	
0 0 3 3	
6	
1 2	
1 3	
2 1	
2 3	
3 1	
3 2	