

Problema M

Metralhadora

Fulanito foi jogar um arcade das antigas. No jogo, ele pode colocar uma metralhadora em qualquer lugar da sua base, que consiste de todos os pontos (x, y) com coordenadas inteiras e $x < 0$. Há N inimigos no campo de batalha. O i -ésimo inimigo ($1 \leq i \leq N$) está na posição (x_i, y_i) com $x_i > 0$. Todas as posições são dadas de antemão.

Uma metralhadora posicionada em (x_m, y_m) cobre um ângulo de visão para a direita centrado na reta $y = y_m$, cujos limites são dados pelas retas $y = y_m \pm \frac{x - x_m}{2}$. Quando colocada, ela atinge todos os inimigos na região delimitada por esse ângulo, incluindo os localizados nas retas-limite.

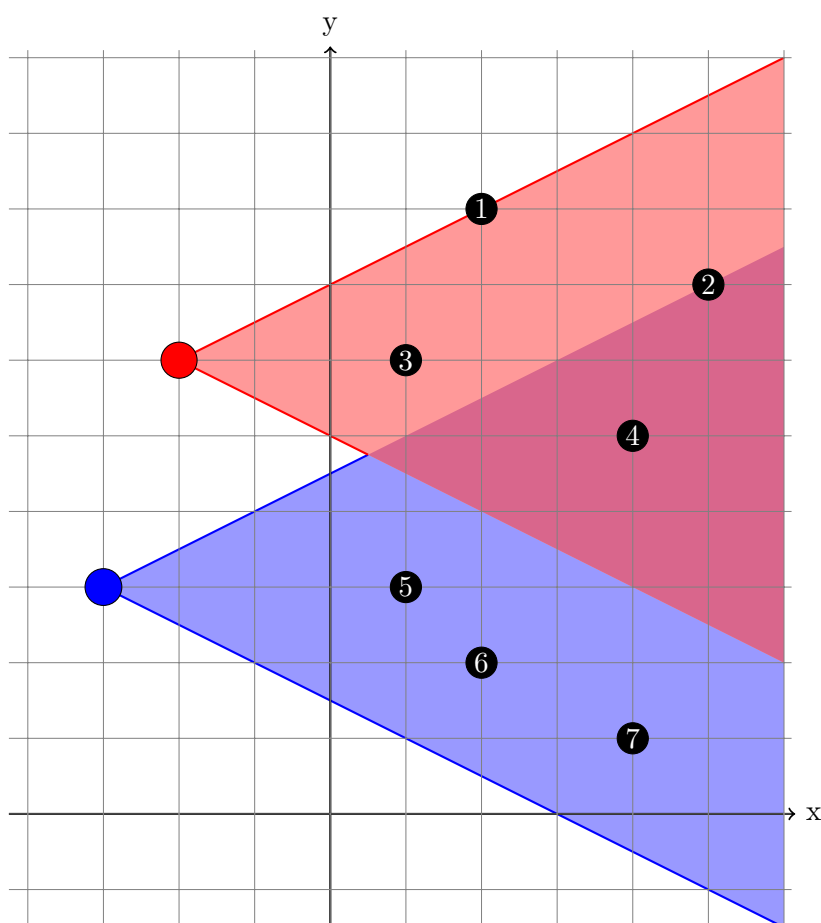


Figura 1: Representação pictórica da entrada de exemplo

O sistema de pontuação usado por esse jogo é desnecessariamente complicado; muitos acreditam que tal sistema foi um grande erro dos desenvolvedores (que, em resposta, afirmam com convicção que “não é um bug, é um recurso!”). Especificamente, a pontuação obtida por um dado posicionamento da metralhadora é calculada executando os seguintes passos:

- Liste os índices (i entre 1 e N) de **todos** os inimigos que a metralhadora atinge.
- Ordene os índices em ordem crescente, e chame os valores ordenados de $i_0 < i_1 < \dots < i_{k-1}$.
- Compute a pontuação usando a fórmula $\left(\sum_{j=0}^{k-1} i_j \cdot 5782344^j \right) \bmod (10^9 + 7)$, onde $a \bmod b$ denota o resto da divisão de a por b .

- Nota: Uma metralhadora que não atinge inimigos recebe uma pontuação exatamente igual a 0.

Para melhorar nesse jogo, Fulanito te faz Q perguntas: cada consulta pede o placar que seria obtido se posicionássemos a metralhadora numa certa posição (x_m, y_m) . Para tornar o problema mais desafiador, os valores de (x_m, y_m) não são dados diretamente. Ao invés disso, são dados valores a e b que podem ser usados para calcular x_m e y_m através das fórmulas $x_m = -1 - ((p + a) \bmod (10^9 + 7))$ e $y_m = (p + b) \bmod (10^9 + 7)$, onde p é a resposta da consulta anterior ($p = 0$ ao processar a primeira consulta).

NOTA: É garantido que a soma do número de inimigos atingidos em todas as consultas é no máximo 10^6 .

Entrada

A entrada consiste de várias linhas. A primeira linha da entrada contém dois inteiros N , Q ($1 \leq N, Q \leq 10^5$), o número de inimigos e o número de consultas.

As próximas N linhas da entrada contém dois inteiros cada: x_i e y_i ($1 \leq x_i, y_i \leq 10^9$), as coordenadas da posição do i -ésimo inimigo.

As próximas Q linhas contém dois inteiros cada: Os valores a e b ($0 \leq a, b < 10^9 + 7$) que especificam cada consulta, como explicado no enunciado.

Saída

Para cada consulta, imprima um único inteiro contendo a resposta para a consulta.

Exemplo de entrada 1	Exemplo de saída 1
7 2	626214369
2 8	981053491
5 7	
1 6	
4 5	
1 3	
2 2	
4 1	
2 3	
373785639 373785644	

Problem M

Machine Gun

Fulanito plays an old-school arcade game. In the game, he can place a machine gun anywhere within his base, which contains all (x, y) points with $x < 0$. There are N enemies on the battlefield. The i -th enemy ($1 \leq i \leq N$) is at position (x_i, y_i) with $x_i > 0$. All these positions are given in advance.

A machine gun at (x_m, y_m) covers an angle of vision to the right, centered on the line $y = y_m$, and whose borders are given by lines $y = y_m \pm \frac{x - x_m}{2}$. When placed, it kills all enemies within such angle, including its border lines.

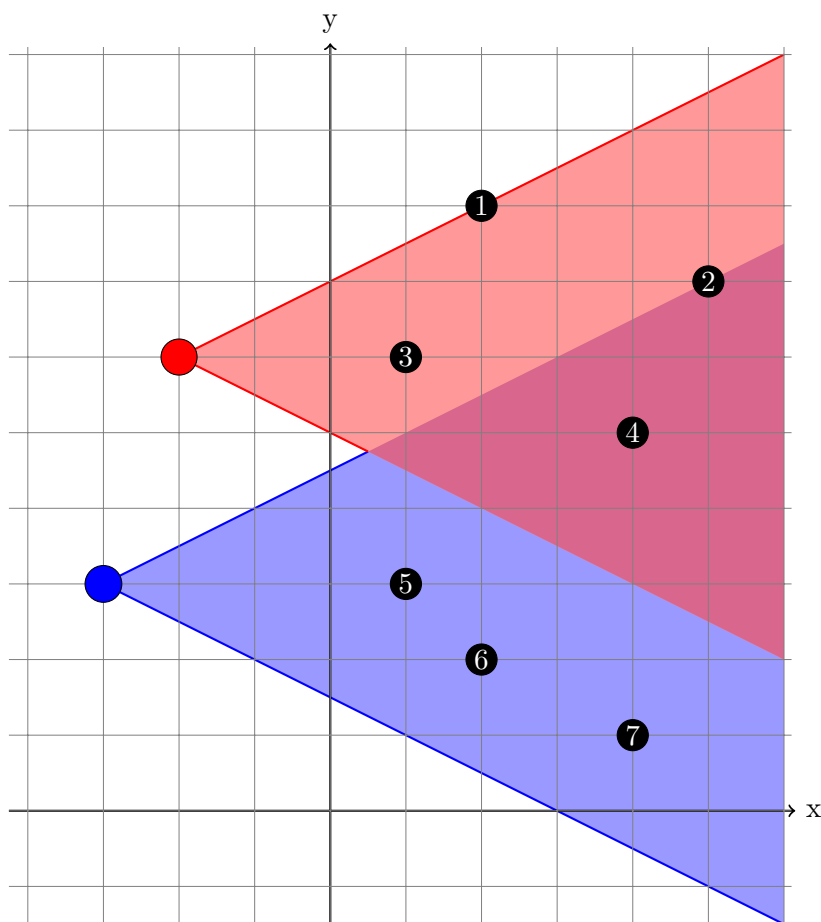


Figure 1: Pictorial representation of the sample input

The scoring system used by this videogame is extremely awkward: many believe that it was in fact a horrible bug by the game developers, who then proceeded to just yell “it’s not a bug, it’s a feature!” to anyone asking about it. Specifically, the score achieved by a certain placement of the machine gun is calculated by executing the following steps:

- Take the ids (i between 1 and N) of **all** the enemies that it kills.
- Sort all those ids in increasing order.
- Let the sorted values be $i_0 < i_1 < \dots < i_{k-1}$
- Compute the score as $\left(\sum_{j=0}^{k-1} i_j \cdot 5782344^j \right) \bmod (10^9 + 7)$

- Note: A machine gun that kills no enemies gets a score of exactly 0 points.

To get better at this game, Fulanito is going to ask you q questions: each question asks the score that would be achieved by placing the machine gun at a certain (x_m, y_m) .

To make the problem more challenging, each (x_m, y_m) is not given directly. Instead, values a and b are given, such that $x_m = -1 - ((p + a) \bmod (10^9 + 7))$ and $y_m = (p + b) \bmod (10^9 + 7)$, where p is the answer to the previous query ($p = 0$ when processing the first query).

NOTE: It is guaranteed that in total, taking all queries together, the number of killed enemies is at most 10^6 .

$$1 \leq x_i, y_i \leq 10^9$$

$$1 \leq N, q \leq 10^5$$

$$0 \leq a, b < 10^9 + 7$$

Input

The first line contains the two integers N, q .

The next N lines contains two integers each: x_i and y_i .

The next q lines contains two integers each: The values a and b that specify each query as explained in the problem statement.

Output

For each query, output a single line with a single integer containing the answer to that query.

Input example 1	Output example 1
7 2	626214369
2 8	981053491
5 7	
1 6	
4 5	
1 3	
2 2	
4 1	
2 3	
373785639 373785644	

Problema M

Metralleta

Fulanito juega un juego de arcade de la vieja escuela. En el juego, él puede poner una metralleta en cualquier lugar dentro de su base, la cual contiene todos los puntos (x, y) , donde $x < 0$. Hay N enemigos en el campo de batalla. El i -ésimo enemigo ($1 \leq i \leq N$) está en la posición (x_i, y_i) con $x_i > 0$. Todas estas posiciones se entregan por adelantado.

Una metralleta en el punto (x_m, y_m) cubre un ángulo de visión hacia la derecha, centrado en la línea $y = y_m$, y sus bordes están dados por las líneas $y = y_m \pm \frac{x - x_m}{2}$. Al ser posicionada, elimina a todos los enemigos en ese ángulo, incluyendo sus bordes.

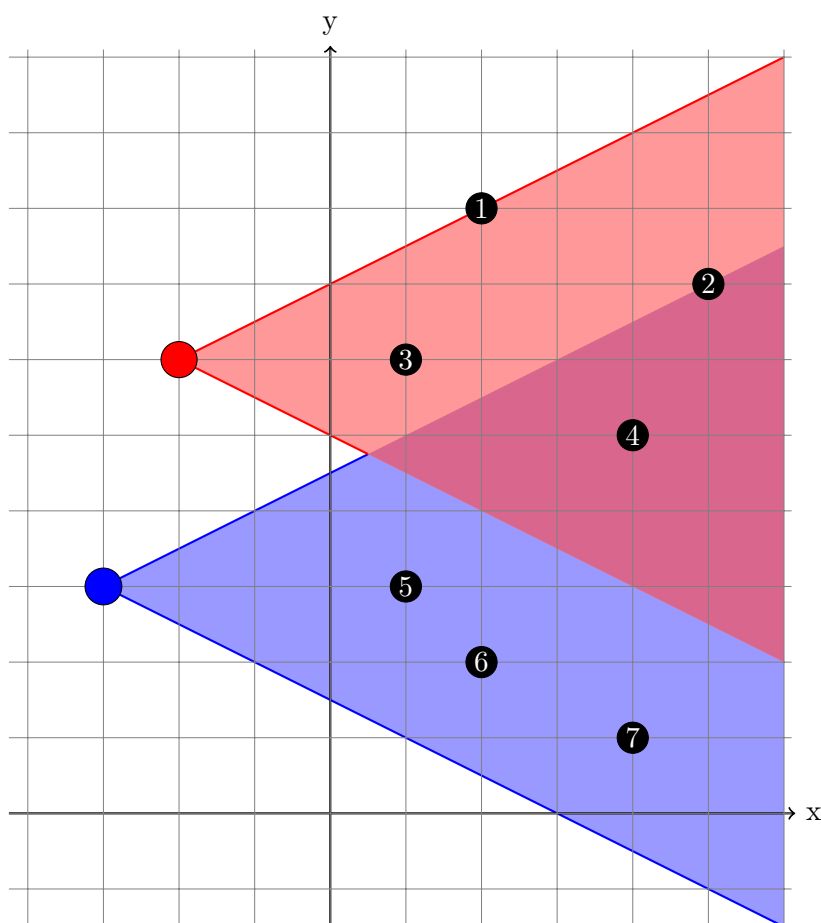


Figura 1: Pictorial representation of the sample input

El sistema de puntuación usado en este video juego es extremadamente extraño: muchos creen que de hecho era un horrible bug de los desarrolladores del juego, quienes responden gritando “¡no es un bug, es un feature!” a cualquiera que les pregunte sobre el tema. Específicamente, el puntaje que se obtiene por cierto posicionamiento de la metralleta se calcula de la siguiente manera::

- Se toman los ids (i entre 1 y N) de **todos** los enemigos que elimina.
- Ordenar esos ids en orden ascendente.
- Sean los valores ordenados $i_0 < i_1 < \dots < i_{k-1}$
- Calcular el puntaje como $\left(\sum_{j=0}^{k-1} i_j \cdot 5782344^j \right) \bmod (10^9 + 7)$

- Nota: Una metralleta que no elimina a ningún enemigo obtiene un puntaje de exáctamente 0 puntos.

Para mejorar en este juego, Fulanito te hará q consultas: cada consulta pregunta por el puntaje que obtendría al posicionar una metralleta en algún (x_m, y_m) .

Para hacer el problema más retador, cada (x_m, y_m) no es entregado por adelantado. En su lugar, se entregan los valores a , y b , de modo que $x_m = -1 - ((p + a) \bmod (10^9 + 7))$ y $y_m = (p + b) \bmod (10^9 + 7)$, donde p es la respuesta a la consulta anterior ($p = 0$ cuando se procesa la primer consulta).

NOTA: Se garantiza que en total, tomando todas las consultas juntas, el total de enemigos eliminados es a lo más 10^6 .

$$1 \leq x_i, y_i \leq 10^9$$

$$1 \leq N, q \leq 10^5$$

$$0 \leq a, b < 10^9 + 7$$

Entrada

La primer línea de entrada contiene dos enteros N , y q .

Cada una de las siguientes N líneas contiene dos enteros: x_i , y y_i .

Cada una de las siguientes q líneas contienen dos enteros: a , y b que especifican cada una de las consultas como se explica en el enunciado.

Salida

Para cada consulta, imprima una línea con un número entero, representando la respuesta a esa consulta.

Ejemplo de entrada 1	Ejemplo de salida 1
7 2	626214369
2 8	981053491
5 7	
1 6	
4 5	
1 3	
2 2	
4 1	
2 3	
373785639 373785644	