Problem A. Ice Hockey World Championship

Time limit: 1 second
Memory limit: 256 MB

This year the Ice Hockey World Championship took place in the Czech Republic. Bobek has arrived in Prague and he would like to visit some of the matches. He does not have any team preferences and he does not have any time restrictions. If he had enough money, he would be able to visit all of the matches. Unfortunately, he has only a limited number of Czech crowns, all of which can be spent on tickets. Knowing how much a ticket costs for each match, calculate the number of ways he can attend a set of matches without exceeding his budget. Two ways are considered different if there exists a match which is visited in one of the ways, but not visited in the other.

Input
A description of Bobek’s situation is read from the standard input. The first line of input contains two positive integers \(N\) and \(M\) \((1 \leq N \leq 40, 1 \leq M \leq 10^{18})\), denoting the number of matches and the number of Czech crowns Bobek can spend. The second line contains \(N\) space-separated positive integers, none of them exceeding \(10^{16}\), representing costs of the matches in Czech crowns.

Output
Output a single line with the number of ways Bobek can visit the matches. Please note that due to the limit on \(N\), this number will be at most \(2^{40}\).

Example

<table>
<thead>
<tr>
<th>standard input</th>
<th>standard output</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 1000</td>
<td>8</td>
</tr>
<tr>
<td>100 1500 500 500 1000</td>
<td></td>
</tr>
</tbody>
</table>

Note
The eight possible ways are:

- no matches visited
- the match worth 100
- the first match worth 500
- the second match worth 500
- the match worth 100 and the first match worth 500
- the match worth 100 and the second match worth 500
- both matches worth 500
- the match worth 1000.

Scoring
There are 10 groups of tests, each of them worth 10 points. The upper bounds on \(N\) and \(M\) in each group of tests are as follows.

<table>
<thead>
<tr>
<th>Group</th>
<th>1–2</th>
<th>3–4</th>
<th>5–7</th>
<th>8–10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limit on (N)</td>
<td>10</td>
<td>20</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>Limit on (M)</td>
<td>(10^6)</td>
<td>(10^{15})</td>
<td>(10^6)</td>
<td>(10^{18})</td>
</tr>
</tbody>
</table>
Problem B. Nuclearia

Time limit: 1 second
Memory limit: 256 MB

Long ago, the people of Nuclearia decided to build several nuclear plants. They prospered for many years, but then a terrible misfortune befell them. The land was hit by an extremely strong earthquake, which caused all the nuclear plants to explode, and radiation began to spread throughout the country. When the people had made necessary steps so that no more radiation would emanate, the Ministry of Environment started to find out how much individual regions were polluted by the radiation. Your task is to write a program that will answer the queries of the Ministry.

How radiation spreads

Nuclearia can be viewed as a rectangle consisting of \( W \times H \) cells. Each nuclear plant occupies one cell and is parametrized by two positive integers: \( a \), which is the amount of radiation caused to the cell where the plant was, and \( b \), which describes how rapidly the caused radiation decreases as we go farther from the plant.

More precisely, the amount of radiation caused to cell \( C = [x_C, y_C] \) by explosion of a plant in cell \( P = [x_P, y_P] \) is \( \max(0, a - b \cdot d(P, C)) \), where \( d(P, C) \) is the distance of the two cells, defined by \( d(P, C) = \max(|x_P - x_C|, |y_P - y_C|) \) (i.e., the minimum number of moves a chess king would travel between them).

The total radiation in a cell is simply the sum of the amounts that individual explosions caused to it. As an example, consider a plant with \( a = 7 \) and \( b = 3 \). Its explosion causes 7 units of radiation to the cell it occupies, 4 units of radiation to the 8 adjacent cells, and 1 unit of radiation to the 16 cells whose distance is 2. Note that if this plant is situated on the edge of Nuclearia or one cell away from the edge, then the explosion also affects some cells outside Nuclearia. An explosion that affects cells outside Nuclearia is called boundary. (Actually, we are never interested in what happens outside Nuclearia. We just need this definition in the Grading section below.)

Queries

The Ministry of Environment makes several queries about the average-per-cell radiation in a given rectangular region. As great chaos rules in the Ministry, you may make no further assumptions about the queried regions — they may overlap or even repeat.

Input

The description of Nuclearia is read from the standard input. The first line contains two space-separated positive integers \( W \) and \( H \) (where \( W \cdot H \leq 2500000 \)) which stand for the width and height of Nuclearia, respectively. The second line contains a positive integer \( N \), which is the number of exploded plants (\( 1 \leq N \leq 200000 \)). Each of the following \( N \) lines contains four positive integers \( x_i, y_i, a_i, b_i \) (\( 1 \leq x_i \leq W, 1 \leq y_i \leq H, 1 \leq a_i, b_i \leq 10^9 \)), which describe a plant in cell \([x_i, y_i]\) with parameters \( a_i, b_i \). Each cell contains at most one plant.

The following line contains a positive integer \( Q \), which is the number of queries (\( 1 \leq Q \leq 200000 \)). Each of the following \( Q \) lines contains four positive integers \( x_{1j}, y_{1j}, x_{2j}, y_{2j} \) (\( 1 \leq x_{1j} \leq x_{2j} \leq W \) and \( 1 \leq y_{1j} \leq y_{2j} \leq H \)), which describe a query about the rectangle whose upper-left corner is the cell \([x_{1j}, y_{1j}]\) and lower-right corner is the cell \([x_{2j}, y_{2j}]\).

You can assume that the total radiation in Nuclearia is less than \( 2^{63} \).

Output

For each query, output a line which contains the average-per-cell radiation in the queried region, rounded
to the nearest integer (half-integral values are rounded up).

Example

<table>
<thead>
<tr>
<th>standard input</th>
<th>standard output</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>1 1 7 3</td>
<td>2</td>
</tr>
<tr>
<td>3 2 4 2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>1 2 2 3</td>
<td></td>
</tr>
<tr>
<td>1 1 4 3</td>
<td></td>
</tr>
<tr>
<td>4 2 4 2</td>
<td></td>
</tr>
<tr>
<td>1 3 4 3</td>
<td></td>
</tr>
</tbody>
</table>

Note

The radiation in Nuclearia after the two explosions is as follows.

```
7 6 3 2
4 6 5 2
1 3 3 2
```

Note that the first explosion is boundary, while the second one is not. As for the queries:

1. The total radiation in the 2-by-2 square is 14, so the average is $14/4 = 3.5$, rounded to 4.
2. The total radiation in Nuclearia is 44, so the average is $44/12 \approx 3.67$, rounded to 4.
3. The average in a single cell is simply the amount of radiation therein.
4. The average radiation in the last row is $9/4 = 2.25$, rounded to 2.

Scoring

There are 14 groups of tests. The groups of tests with odd numbers only contain plants for which $a$ is a multiple of $b$. Further constraints on groups of tests and their grading are as follows.

<table>
<thead>
<tr>
<th>Group</th>
<th>Further constraints</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$H = 1$, $N \cdot W \leq 10^8$, $Q \cdot W \leq 10^8$</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>$H = 1$, $N \cdot W \leq 10^8$, $Q \cdot W \leq 10^8$</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>$N \cdot W \cdot H \leq 10^8$, $Q \cdot W \cdot H \leq 10^8$</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>$N \cdot W \cdot H \leq 10^8$, $Q \cdot W \cdot H \leq 10^8$</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>$H = 1$, $N \cdot W \leq 10^8$</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>$H = 1$, $N \cdot W \leq 10^8$</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>$N \cdot W \cdot H \leq 10^8$</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>$N \cdot W \cdot H \leq 10^8$</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>$H = 1$</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>$H = 1$</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>no boundary explosions</td>
<td>15</td>
</tr>
<tr>
<td>12</td>
<td>no boundary explosions</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>none</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>none</td>
<td>8</td>
</tr>
</tbody>
</table>
Problem C. Calvinball championship, again

Recall that a Calvinball championship is being held in Czech Republic this year. A game of Calvinball is played by \( n \) players with distinct names, divided into any number of non-empty teams. Some players dislike each other; disliking is symmetric: if player \( a \) dislikes player \( b \), then also \( b \) dislikes \( a \).

The International Calvinball Disorganization decided to make a last-minute change to the team selection procedure: no two people who dislike each other may be on the same team, and subject to that, the number of teams must be as small as possible.

For example, if Calvin, Hobbes, Susie, Tom, Jerry, and Batman play in the game, Batman dislikes everyone else and Tom does not like Jerry and Hobbes, it is possible to play the game with three teams (for example, Batman alone, Tom with Susie, and Calvin with Hobbes and Jerry), but not with two teams (since Batman, Tom, and Jerry dislike each other and must be in different teams), and not with four teams (since a smaller number of teams is possible).

Given the description of which players dislike each other, determine some allowed division of the players into teams (an arbitrary one, if several exist).

Input

This is an output-only task. In "Files", you will find zip-archive with 10 files called input1.txt, ..., input10.txt. Each of the files has the following format.

The first line contains two non-negative integers \( n \) and \( m \), giving the number of players and the number of distinct pairs of players that dislike each other, respectively. The players are numbered from 1 to \( n \). The \( i \)-th of the \( m \) following lines contains two distinct positive integers \( a_i \) and \( b_i \) (\( 1 \leq a_i, b_i \leq n \)), showing that the players \( a_i \) and \( b_i \) dislike each other.

Output

For the input file input\( k \).txt (with \( k = 1, \ldots, 10 \)), prepare the output file output\( k \).txt with the following format. The first line contains a non-negative integer \( t \), giving the number of teams to which the players are divided. The \( i \)-th of the following \( t \) lines contains a space-separated list of numbers of players belonging to the \( i \)-th team. The teams as well as the players in each team can be listed in any order.

The output files should be submitted through the contest interface. In case your submission lacks some of the output files, these are copied from the previous submission (if there is any). It is therefore also possible to submit the output files one by one.

Example

<table>
<thead>
<tr>
<th>standard input</th>
<th>standard output</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 7</td>
<td>3</td>
</tr>
<tr>
<td>1 6</td>
<td>6</td>
</tr>
<tr>
<td>2 6</td>
<td>4 3</td>
</tr>
<tr>
<td>3 6</td>
<td>1 2 5</td>
</tr>
<tr>
<td>4 6</td>
<td></td>
</tr>
<tr>
<td>5 6</td>
<td></td>
</tr>
<tr>
<td>5 4</td>
<td></td>
</tr>
<tr>
<td>2 4</td>
<td></td>
</tr>
</tbody>
</table>

Note

The example corresponds to the situation described in the task statement, with players numbered as follows:
**Scoring**

10 points will be awarded for each of the 10 input files for which a correct output is submitted.