

Problem J. Road Manager

Input file: *standard input*
Output file: *standard output*
Time limit: 4 seconds
Memory limit: 512 mebibytes

There are $n \times m$ cells on a grid, the top left cell is at $(1, 1)$ while the bottom right cell is at (n, m) .

Two cells are considered adjacent if they share a common side. Note that the grid is circular: (i, m) is also adjacent to $(i, 1)$. However, there is no such rule for connecting the first row and the last row.

There is exactly one bidirectional road between two adjacent cells:

- For every cell (i, j) such that $1 \leq i \leq n$ and $1 \leq j \leq m$, there is a road between it and $(i, j \bmod m + 1)$. Note that the grid is circular, so when $j = m$, there is a road between (i, m) and $(i, 1)$.
- For every cell (i, j) such that $1 \leq i < n$ and $1 \leq j \leq m$, there is a road between it and $(i + 1, j)$.

The roads form a weighted graph, every road has its weight. You will be given q queries. In the i -th query, you will be given two integers, l_i and r_i . Consider the graph that remains if we remove all cells (x, y) such that $1 \leq x \leq n$ and $l_i \leq y \leq r_i$, together with the roads linked to them. You need to find the total weight of the minimum spanning tree (MST) of the resulting graph.

Note that different queries are independent. In other words, the removed cells and roads come back after you answer a query.

Input

There is only one test case in each test.

The first line of this test case contains six integers n, m, SA, SB, SC , and lim ($1 \leq n \leq 100, 4 \leq m \leq 10\,000$, and $1 \leq SA, SB, SC, lim \leq 10^9$).

Since the grid might be rather large, you have to generate the input using the following C/C++ code, where `addege(a, b, c, d, w)` means that we add a road between (a, b) and (c, d) with weight w :

```
1 unsigned int SA, SB, SC; int lim;
2 int getweight() {
3     SA ^= SA << 16;
4     SA ^= SA >> 5;
5     SA ^= SA << 1;
6     unsigned int t = SA;
7     SA = SB;
8     SB = SC;
9     SC ^= t ^ SA;
10    return SC % lim + 1;
11 }
12 void gen() {
13     scanf("%d%d%u%u%u%d", &n, &m, &SA, &SB, &SC, &lim);
14     int i, j, w;
15     for (i = 1; i <= n; i++)
16         for (j = 1; j <= m; j++) {
17             w = getweight();
18             if (j < m) {
19                 addege(i, j, i, j + 1, w);
20             } else {
21                 addege(i, j, i, 1, w);
22             }
23         }
24     for (i = 1; i < n; i++)
25         for (j = 1; j <= m; j++) {
26             w = getweight();
27             addege(i, j, i + 1, j, w);
28         }
29 }
```

A note for coders in other languages: `int` is a 32-bit signed integer type, whereas `unsigned int` is a 32-bit unsigned integer type. The result of every operation with `unsigned int` is truncated to the last 32 bits. This effectively means each result is taken modulo 2^{32} .

On the second line, there is one integer q ($1 \leq q \leq 10\,000$), denoting the number of queries.

Each of the next q lines contains two integers l_i and r_i ($1 < l_i \leq r_i < m$) describing a query.

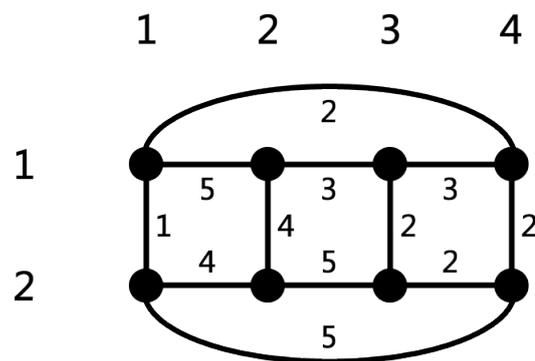
Output

For each query, print a single line containing an integer: the total weight of the MST of the respective graph.

Example

standard input	standard output
2 4 1 2 3 5	9
3	5
2 2	13
2 3	
3 3	

Note



The generated graph in the example.