1830 - Again Making Queries I

Description

There are **N** contiguous cells numbered from 1 to **N**, identified as the sequence **A**.
Initially, each cell contains the values **A[1], A[2], ..., A[N]** (**A[i]** = 0 for 1 <= i <= **N**): this mean that initially, each cell contains a 0 in it.

A sub-contiguous group of cells (always one, two or three cells) can be updated in this way --- **Update(i,k)**:
•The cell numbered **i + 1** is added **k**, if the cell exist.
•The cell numbered **i - 1** is added **k**, if the cell exist.
•The cell numbered **i** is added **k+1**.

For example, if **N** = 6 and the updates **[3, 6]** and **[4, 7]** were performed, this is what would happen.
Initially: {0, 0, 0, 0, 0, 0}
Update [3, 6]: {0, 6, 7, 6, 0, 0}
Update [4, 7]: {0, 6, 14, 14, 7, 0}

After performing some update operations, it would be amazing to answer questions like the following:
1) A range **[u, v]** is defined such that **u <= v**.
2) The answer is the sum of every cell in the range **[u, v]** (both **u** and **v** are included) modulus 10^4.

Given **N** and **U** updates ranges. You have to write a program capable of answering **Q** questions.

Input specification

The first line contains three integers: **N**, **U**, and **Q (1 <= N, U, Q <= 10^6)**, representing the number of cells, the number of update operations, and the number of questions respectively.
Each of the following **U** lines contains two integers **(1 <= i <= N)** and **(1 <= k <= 10^6)** separated by a single space indicating an update operation.
Each of the following **Q** lines contains two integers **u** and **v (1 <= u <= v <= N)** separated by a single space indicating a question.

Output specification

For each question **[u, v]** you must print the sum of all contiguous cells starting at **u** and ending at **v** modulus 10^4.

Sample input

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

Sample output

```
21
41
```

Hint(s)

Source	Yonny Mondelo Hernández
Added by	**ymondelo20**
Addition date	2012-05-18
Time limit (ms)	30000
Test limit (ms)	3000
Memory limit (kb)	130000
Output limit (mb)	64
Size limit (bytes)	30000
Enabled languages	C C# C++ Java Pascal Perl PHP Python Ruby Text