Matrix

You wake up as usual and, during the day, you start noticing certain glitches in the world. By the end of the day, you are sure you are in the Matrix and you want to get out. The Architect finds you and offers you a way out if you manage to aid him in a problem.

The Architect has an $N \times M$ matrix, where initially all of the cells are 0. During the day, parts of the matrix are modified and sometimes the Architect wants to find something about a certain cell in the matrix.
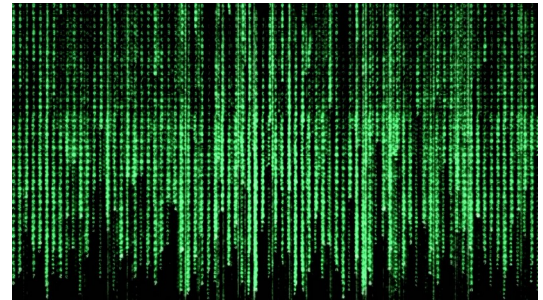
The Architect tells you that the matrix has N rows and M columns, then he sends you K operations as follows:

- UPDATE X Y_1 Y_2: all of the cells on row X between the columns Y_1 and Y_2 inclusively have their value increased by 1.
- QUERY X Y: find the value of the cell on row X and column Y.

Task

Because the Architect doesn't trust you completely, this problem is **NOT going to be judged the usual batch way**. Instead, you have to implement the following functions:

- `void initialize(long long N, long long M)`
- `void update(long long X, long long Y1, long long Y2)`
- `int query(long long X, long long Y)`

in your source file, with these exact prototypes. You may implement other functions and declare local and global variables. Your submissions must not interact in any way with standard input/output, nor with any other file. Do not declare a main function. Your source file will be compiled and linked together with a source file provided by the committee, which will handle the input/output and will call the functions your source file has declared **(grader.cpp).**

Limits and constraints

- $1 \leq N, M \leq 10^{18}$
- $0 \leq K \leq 500{,}000$
- Rows and columns are numbered starting from 1.
- Time limit: 1.5 seconds
- Memory limit: 128 MB

Subtasks

Subtask	Percent of points	Additional input constraints
1	20	**N, M** $\leq$ 1000; **K** $\leq$ 100
2	30	**N, M** $\leq$ 100,000; **K** $\leq$ 100,000
3	50	none

Sample grader

The problem's web page includes a sample grader, **sample_grader.{c,cpp}**. You can use the sample grader to compile and test your implementation of the functions. The sample grader reads the input file **matrix.in** in the following format:

line 1:	**N M**	
lines 2 until EOF:	0 X Y_1 Y_2	for `update()` operations or
	1 X Y	for `query()` operations

For each `query()` operation, the sample grader prints the return value to the output file **matrix.out.**

Example for the sample grader

matrix.in	matrix.out	Explanation
2 5 0 2 2 4 1 2 3 0 2 1 5 1 2 2 0 1 2 2 1 1 2	1 2 1	After the last operation, the matrix looks like this: 0 1 0 0 0 1 2 2 2 1