

Cycle Game

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 256 megabytes

There's a board with $n \times m$ white squares. Jeroen plays a single-player game on this grid. Each turn he colors one square black. Two squares are adjacent if they share an edge. The squares, together with these adjacency relations form a planar graph. The game ends when a simple cycle of black squares is formed that has a non-empty interior. By drawing the edges of the graph with straight lines, between the centres of the squares we get a drawing of the planar graph. A square that does not lie on the cycle, is in the interior of the cycle, if it lies in the interior of the polygon that the edges of the cycle form, when drawn as straight line segments. Jeroen wants to produce a pretty picture on the playing board when he's done so he already thought of the moves he is going to play. If a move happens to end the game, he doesn't play it. In the input there are k distinct positions (r_i, c_i) , the moves that Jeroen has planned out.

Your job is to figure out for each move, if it would end the game or not, and print a bitstring with 1 if the i -th move ends up being played, and 0 if this move would end the game, and does not get played.

Input

The first line of the input contains three integers n, m, k ($1 \leq n \cdot m \leq 300\,000$, $1 \leq k \leq n \cdot m$) The next k lines each consist of two integers r_i, c_i ($1 \leq r_i \leq n$, $1 \leq c_i \leq m$), which are the row and column of the i -th move that Jeroen is planning to play.

Output

Output a single line with a string of length k with a 1 on the i -th position if the i -th move will be played, and 0 if this move would end up losing, so it will not be played.

Examples

standard input	standard output
4 3 7 2 1 2 2 2 3 3 1 3 2 4 1 4 2	1111111
3 3 8 1 1 1 2 1 3 2 3 3 3 3 2 3 1 2 1	11111110