Problem I. Midpoint

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	256 mebibytes

One day, you found $L + M + N$ points on a 2D plane, which you named $A_1, \ldots, A_L$, $B_1, \ldots, B_M$, $C_1, \ldots, C_N$. Note that two or more points of them can be at the same coordinate. These were named after the following properties:

- the points $A_1, \ldots, A_L$ were located on a single straight line,
- the points $B_1, \ldots, B_M$ were located on a single straight line, and
- the points $C_1, \ldots, C_N$ were located on a single straight line.

Now, you are interested in a triplet (i, j, k) such that C_k is the midpoint between A_i and B_j. Your task is counting such triplets.

Input

The first line contains three space-separated positive integers L, M, and N ($1 \leq L, M, N \leq 10^5$). The next L lines describe A. The i-th of them contains two space-separated integers representing the x-coordinate and the y-coordinate of A_i. The next M lines describe B. The j-th of them contains two space-separated integers representing the x-coordinate and the y-coordinate of B_j. The next N lines describe C. The k-th of them contains two space-separated integers representing the x-coordinate and the y-coordinate of C_k.

It is guaranteed that the absolute values of all the coordinates do not exceed 10^5.

Output

Print the number of the triplets which fulfill the constraint.

Examples

standard input	standard output
2 2 3 0 0 2 0 0 0 0 2 0 0 1 1 1 1	3
4 4 4 3 5 0 4 6 6 9 7 8 2 11 3 2 0 5 1 4 3 7 4 10 5 1 2	8
4 4 4 0 0 3 2 6 4 9 6 7 14 9 10 10 8 13 2 4 2 5 4 6 6 8 10	3