

Problem I. Squares

Time limit: 2 seconds
Memory limit: 512 megabytes

Consider an infinite grid. An infinite set of 2×2 squares is *covering set* if each cell of the plane is covered by exactly one square and they cover all the cells of the plane.

A set of squares is *good* if it is a subset of some *covering set*.

You have an initially empty set of squares S and n queries for adding and removing squares (x_i, y_i) , where the pair of numbers (x_i, y_i) describes a square that covers the cells (x_i, y_i) , $(x_i + 1, y_i)$, $(x_i, y_i + 1)$, and $(x_i + 1, y_i + 1)$.

After each query, you are required to output a single number —the size of the largest *good* subset of the set S .

Input

The first line contains a single number n ($1 \leq n \leq 200\,000$) — the number of queries.

The following n lines contain two integers x_i, y_i ($1 \leq x_i, y_i \leq 10^9$). If at the moment of the i -th query the square defined by the pair (x_i, y_i) was contained in S , then it is removed from the set, otherwise—it is added.

Output

Output n lines, in the i -th line output the size of the largest good subset of S after executing the first i queries.

Example

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