# 范围查询(Range)

**Descriptioin**

Let S be a set of n integral points on the x-axis. For each given interval [a, b], you are asked to count the points lying inside.

**Input**

The first line contains two integers: n (size of S) and m (the number of queries).

The second line enumerates all the n points in S.

Each of the following m lines consists of two integers a and b and defines an query interval [a, b].

**Output**

The number of points in S lying inside each of the m query intervals.

**Example**

Input

5 2

1 3 7 9 11

4 6

7 12

Output

0

3

**Restrictions**

0 <= n, m <= 5 \* 10^5

For each query interval [a, b], it is guaranteed that a <= b.

Points in S are distinct from each other.

Coordinates of each point as well as the query interval boundaries a and b are non-negative integers not greater than 10^7.

Time: 2 sec

Memory: 256 MB

**描述**

数轴上有n个点，对于任一闭区间 [a, b]，试计算落在其内的点数。

**输入**

第一行包括两个整数：点的总数n，查询的次数m。

第二行包含n个数，为各个点的坐标。

以下m行，各包含两个整数：查询区间的左、右边界a和b。

**输出**

对每次查询，输出落在闭区间[a, b]内点的个数。

**样例**

见英文题面

**限制**

0 ≤ n, m ≤ 5×105

对于次查询的区间[a, b]，都有a ≤ b

各点的坐标互异

各点的坐标、查询区间的边界a、b，均为不超过10^7的非负整数

时间：2 sec

内存：256 MB

# 祖玛(Zuma)

**Description**

Let's play the game Zuma!

There are a sequence of beads on a track at the right beginning. All the beads are colored but no three adjacent ones are allowed to be with a same color. You can then insert beads one by one into the sequence. Once three (or more) beads with a same color become adjacent due to an insertion, they will vanish immediately.



Note that it is possible for such a case to happen for more than once for a single insertion. You can't insert the next bead until all the eliminations have been done.

Given both the initial sequence and the insertion series, you are now asked by the fans to provide a playback tool for replaying their games. In other words, the sequence of beads after all possible eliminations as a result of each insertion should be calculated.

**Input**

The first line gives the initial bead sequence. Namely, it is a string of capital letters from 'A' to 'Z', where different letters correspond to beads with different colors.

The second line just consists of a single interger n, i.e., the number of insertions.

The following n lines tell all the insertions in turn. Each contains an integer k and a capital letter Σ, giving the rank and the color of the next bead to be inserted respectively. Specifically, k ranges from 0 to m when there are currently m beads on the track.

**Output**

n lines of capital letters, i.e., the evolutionary history of the bead sequence.

Specially, "-" stands for an empty sequence.

**Example**

Input

ACCBA

5

1 B

0 A

2 B

4 C

0 A

Output

ABCCBA

AABCCBA

AABBCCBA

-

A

**Restrictions**

0 <= n <= 10^4

0 <= length of the initial sequence <= 10^4

Time: 2 sec

Memory: 256 MB

**Hints**

List

**描述**

祖玛是一款曾经风靡全球的游戏，其玩法是：在一条轨道上初始排列着若干个彩色珠子，其中任意三个相邻的珠子不会完全同色。此后，你可以发射珠子到轨 道上并加入原有序列中。一旦有三个或更多同色的珠子变成相邻，它们就会立即消失。这类消除现象可能会连锁式发生，其间你将暂时不能发射珠子。

开发商最近准备为玩家写一个游戏过程的回放工具。他们已经在游戏内完成了过程记录的功能，而回放功能的实现则委托你来完成。

游戏过程的记录中，首先是轨道上初始的珠子序列，然后是玩家接下来所做的一系列操作。你的任务是，在各次操作之后及时计算出新的珠子序列。

**输入**

第一行是一个由大写字母'A'~'Z'组成的字符串，表示轨道上初始的珠子序列，不同的字母表示不同的颜色。

第二行是一个数字n，表示整个回放过程共有n次操作。

接下来的n行依次对应于各次操作。每次操作由一个数字k和一个大写字母Σ描述，以空格分隔。其中，Σ为新珠子的颜色。若插入前共有m颗珠子，则k ∈ [0, m]表示新珠子嵌入之后（尚未发生消除之前）在轨道上的位序。

**输出**

输出共n行，依次给出各次操作（及可能随即发生的消除现象）之后轨道上的珠子序列。

如果轨道上已没有珠子，则以“-”表示。

**样例**

见英文题面

**限制**

0 ≤ n ≤ 10^4

0 ≤ 初始珠子数量 ≤ 10^4

时间：2 sec

内存：256 MB

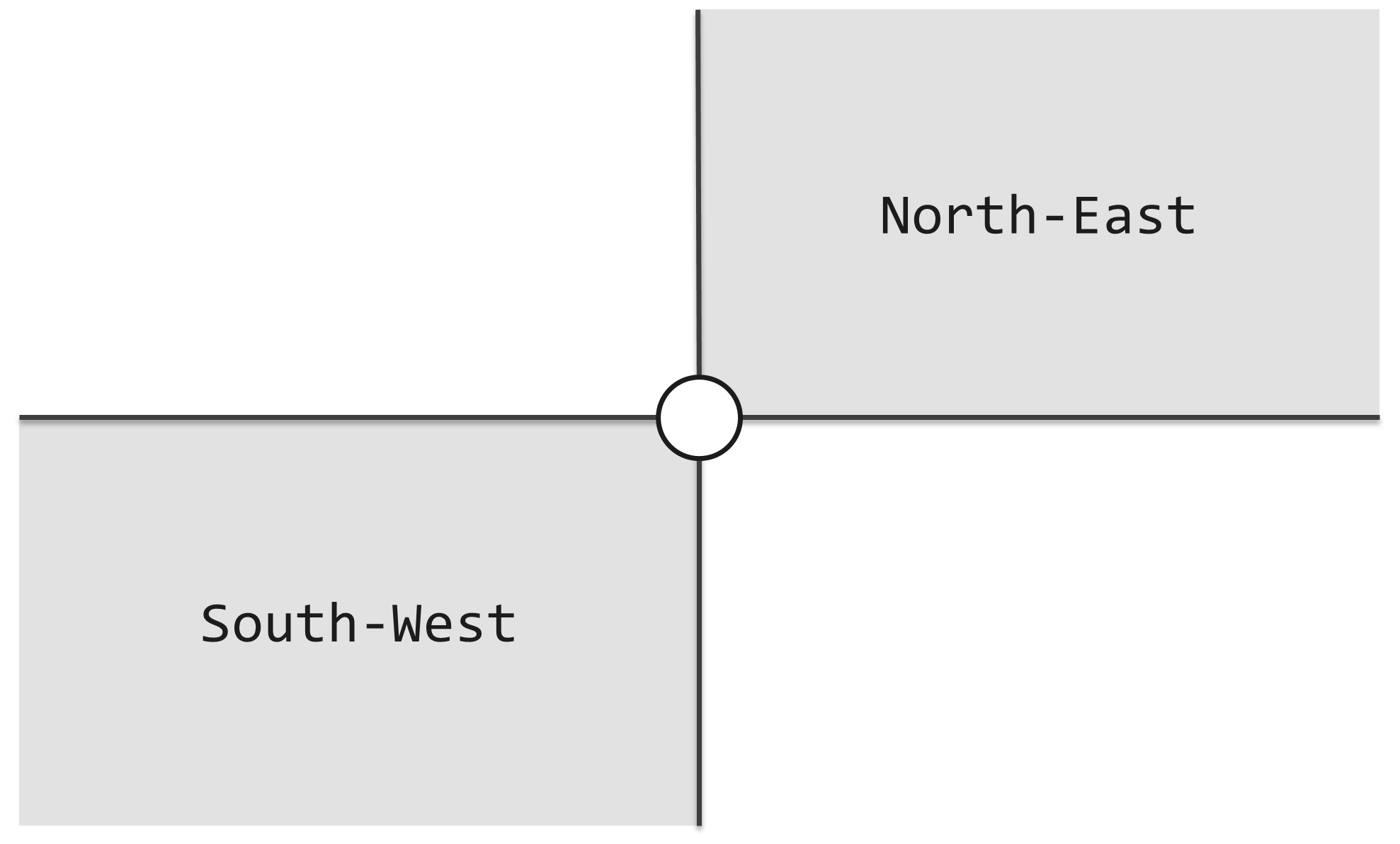
**提示**

列表

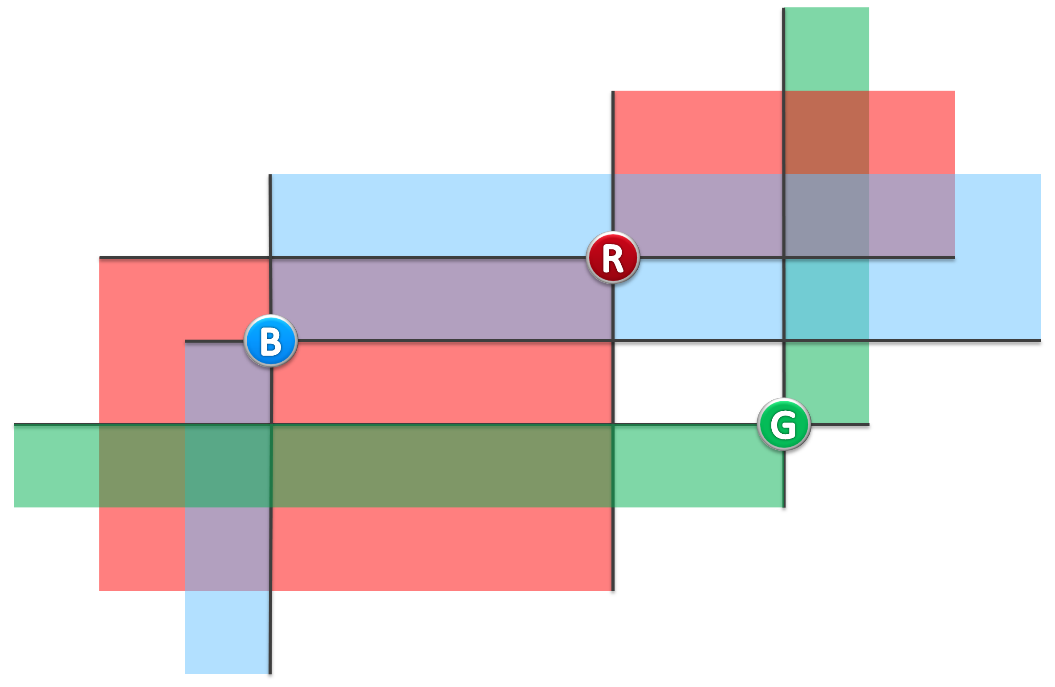
# 灯塔(LightHouse)

**Description**

As shown in the following figure, If another lighthouse is in gray area, they can beacon each other.



For example, in following figure, (B, R) is a pair of lighthouse which can beacon each other, while (B, G), (R, G) are NOT.



**Input**

1st line: N

2nd ~ (N + 1)th line: each line is X Y, means a lighthouse is on the point (X, Y).

**Output**

How many pairs of lighthourses can beacon each other

( For every lighthouses, X coordinates won't be the same , Y coordinates won't be the same )

**Example**

Input

3

2 2

4 3

5 1

Output

1

**Restrictions**

For 90% test cases: 1 <= n <= 3 \* 105

For 95% test cases: 1 <= n <= 106

For all test cases: 1 <= n <= 4 \* 106

For every lighthouses, X coordinates won't be the same , Y coordinates won't be the same.

1 <= x, y <= 10^8

Time: 2 sec

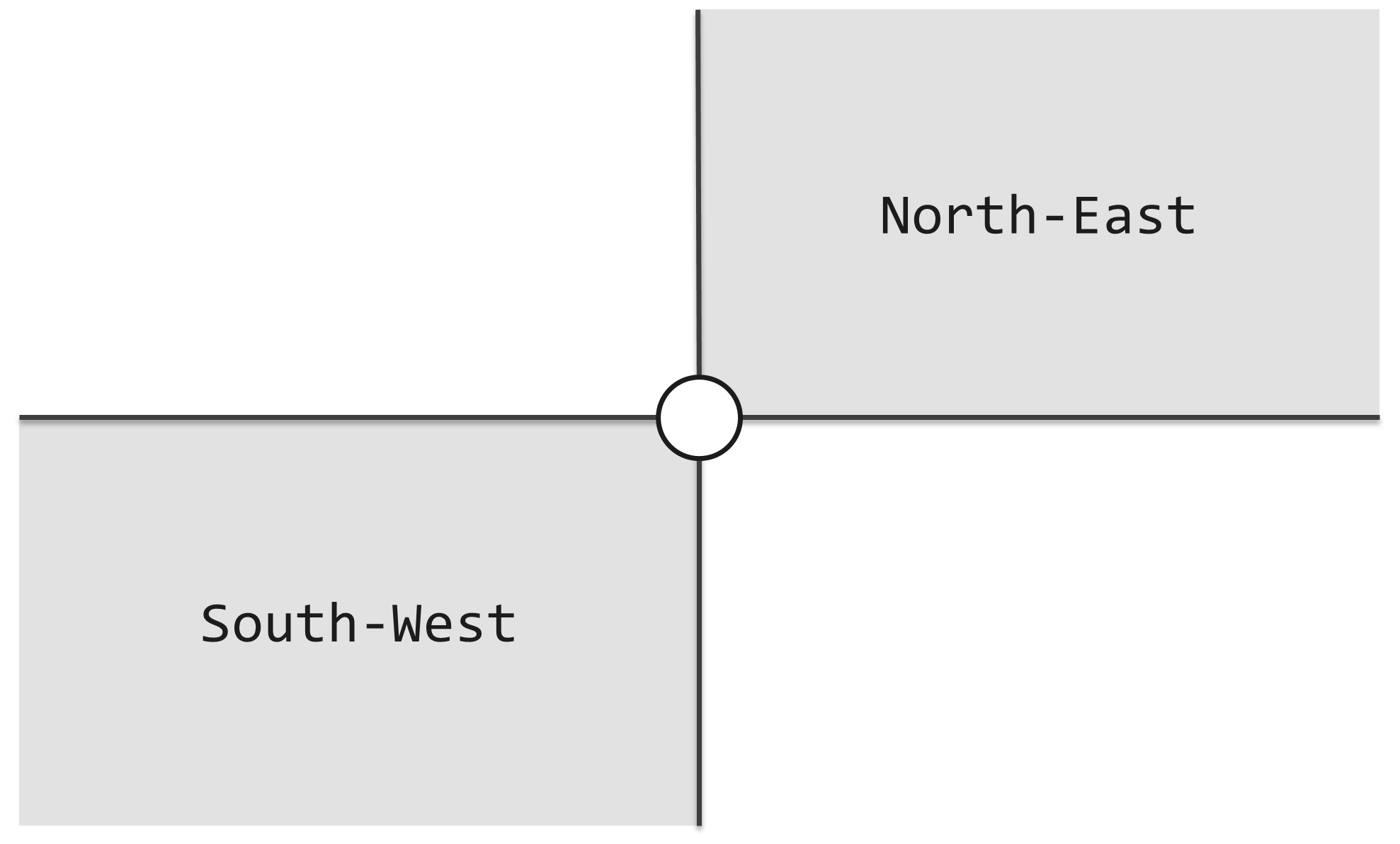
Memory: 256 MB

**Hints**

The range of **int** is usually [-231, 231 - 1], it may be too small.

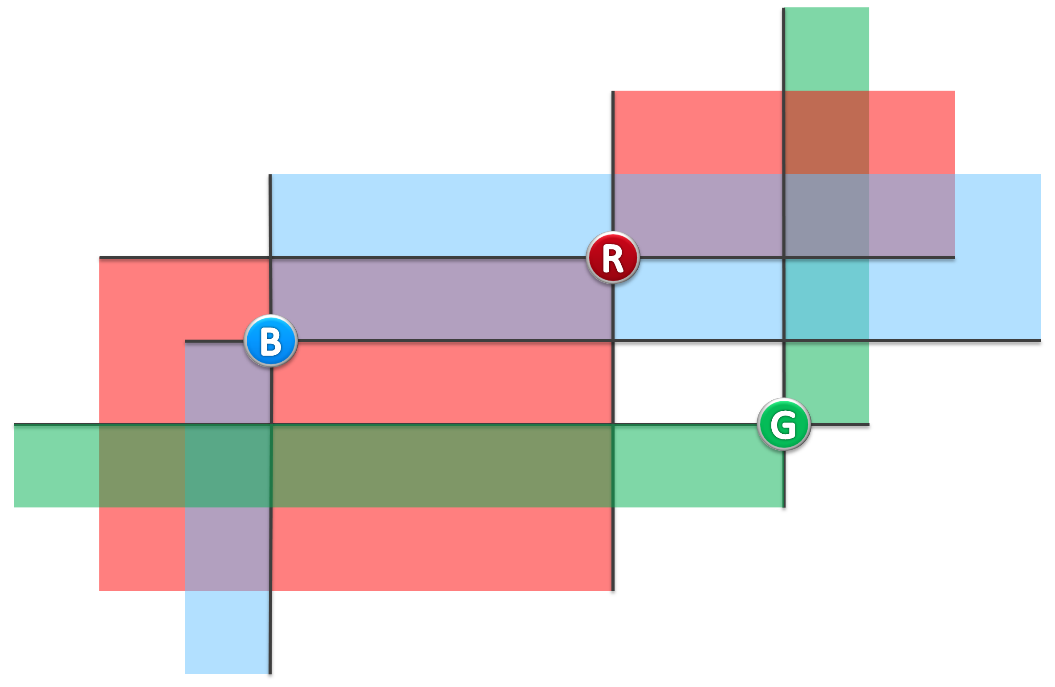
**描述**

海上有许多灯塔，为过路船只照明。



（图一）

如图一所示，每个灯塔都配有一盏探照灯，照亮其东北、西南两个对顶的直角区域。探照灯的功率之大，足以覆盖任何距离。灯塔本身是如此之小，可以假定它们不会彼此遮挡。



（图二）

若灯塔A、B均在对方的照亮范围内，则称它们能够照亮彼此。比如在图二的实例中，蓝、红灯塔可照亮彼此，蓝、绿灯塔则不是，红、绿灯塔也不是。

现在，对于任何一组给定的灯塔，请计算出其中有多少对灯塔能够照亮彼此。

**输入**

共n+1行。

第1行为1个整数n，表示灯塔的总数。

第2到n+1行每行包含2个整数x, y，分别表示各灯塔的横、纵坐标。

**输出**

1个整数，表示可照亮彼此的灯塔对的数量。

**样例**

见英文题面

**限制**

对于90%的测例：1 ≤ n ≤ 3×105

对于95%的测例：1 ≤ n ≤ 106

全部测例：1 ≤ n ≤ 4×106

灯塔的坐标x, y是整数，且不同灯塔的x, y坐标均互异

1 ≤ x, y ≤ 10^8

时间：2 sec

内存：256 MB

**提示**

注意机器中整型变量的范围，C/C++中的int类型通常被编译成32位整数，其范围为[-231, 231 - 1]，不一定足够容纳本题的输出。