

Problem B. 差 (diff.c/cpp/pas)

Input file: diff.in
Output file: diff.out
Time limit: 1 seconds
Memory limit: 1 gigabytes

这是一道交互题。

老虎和蒜头是好朋友，蒜头和栋栋修修抽抽鼠鼠都是好朋友。蒜头居住的城市可以看成是一个由 $R \times C$ 的网格组成的二维平面，蒜头和他的朋友的家都坐落在平面的格点上，同一个格点上可以居住任意多的人。蒜头认为，好朋友的距离一定不能太远，因此蒜头的家一定满足到所有好朋友家的距离和最短。平面上两点 (a, b) 和 (c, d) 之间的距离定义为 $|a - c| + |b - d|$ 。

然而，老虎并不知道蒜头的家在哪，老虎也不知道蒜头的朋友们的家在哪，他甚至忘记了自己的家在哪里，他只想知道，蒜头的家到蒜头所有朋友的家们的距离之和是多少。当然，为了达成这个目标，老虎可以向蒜头提出若干次询问。

Interaction Protocol

你的任务是实现一个函数：

```
long long getMiniumDistance(int R, int C);
```

其中 R 和 C 是蒜头居住城市的大小，为了向蒜头发起询问，你的程序可以使用下列函数：

```
long long getDistance(int x, int y);
```

该函数会回答蒜头的所有朋友的居所到平面上的点 (x, y) 的距离和。你的程序可以在得到答案后返回结果。若你的返回结果和标准答案相同，那么我们将根据你调用 `getDistance` 函数的次数来计算你的得分。

注意你必须保证 $1 \leq x \leq R, 1 \leq y \leq C$ ，否则交互库会立即终止你的程序，并返回 “Wrong Answer”。

Examples

以下是一次成功的交互，在该例中蒜头有 3 个朋友，分别住在 $(1, 2), (1, 4), (1, 10)$ ，而蒜头的家住在 $(1, 4)$ 位置。在该例中，程序共进行了三次询问。

```
getMiniumDistance(1, 10): // 网格大小为1×10
getDistance(1, 3)    // 返回值为9
getDistance(1, 7)    // 返回值为11
getDistance(1, 4)    // 返回值为8
answer = 8
return answer
```

Notes

我们保证蒜头所有朋友的位置与你所提出的询问无关，即这些信息在进行询问前已经是确定的。

本题共有两个子任务，第一个子任务占 20%，满足 $R, C \leq 100$ ；第二个子任务占 80%，满足 $R, C \leq 10^7$ 。同时，对于两个子任务来说，蒜头的朋友数量都不会超过 1000。

对于两个子任务，我们都有如下评分标准，不妨假设你的程序调用 `getDistance` 函数的次数为 T ，你的得分占子任务总分的比例为 p ：

$T \in$	$p =$
$(10000, \infty)$	0
$(200, 10000]$	0.1
$(100, 200]$	0.3
$(80, 100]$	0.5
$(72, 80]$	0.7
$(67, 72]$	$1 - 0.05(T - 67)$
$[0, 67]$	1.0

本题交互库的运行时间不会超过 0.5s，占用空间大小不会超过 128MB，也就是说选手程序的可用时间至少为 0.5s，可用空间至少为 896MB。

在下发文件中，我们提供了 `diff.h`, `grader.cpp`, `diff.cpp`, `sample.cpp` 三个文件，你最终提交的代码应当包含在 `diff.cpp` 中。

如果你想要命令行编译你的代码，应当执行：`g++ diff.cpp grader.cpp -o diff -O2 -std=c++14`。如果你并不想使用命令行编译，或是不熟悉如何处理，可以直接在 `sample.cpp` 下的函数内进行编辑，并修改文件名后提交该文件。在最终评测的头文件中，不会包括 `#define CONTESTANT` 一行。