



## International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

## 游戏

中文 — 1.1

Bazza 和 Shazza 正在玩游戏。游戏在一个  $R$  行  $C$  列的网格上进行。其中， $R$  行编号为  $0, \dots, R-1$ ， $C$  列编号为  $0, \dots, C-1$ 。我们用  $(P, Q)$  表示位于  $P$  行  $Q$  列的单元格。每个单元格包含一个非负整数，游戏开始时所有单元格内的整数均为零。

游戏如下进行：任意时刻，Bazza 可以做如下动作之一：

- 修改一个单元格  $(p, q)$  内包含的整数值；
- 要求 Shazza 计算一个给定子矩阵中所有单元格内数字的最大公约数 (GCD)，子矩阵的两个对角分别为  $(p, q)$  和  $(u, v)$ （子矩阵包含给定的两个对角点）。

Bazza 会做  $N_U + N_Q$  次动作(其中，修改单元格内数据  $N_U$  次，询问 GCD  $N_Q$  次)。

你的任务是对 Bazza 提出的问题给出正确答案。

## 样例

假设  $R = 2$  且  $C = 3$ ，Bazza 做了如下的修改动作：

- 将单元格  $(0, 0)$  中的值修改为 20；
- 将单元格  $(0, 2)$  中的值修改为 15；
- 将单元格  $(1, 1)$  中的值修改为 12。

20	0	15
0	12	0

上图显示了修改后的网格。Bazza可以提如下的问题：

- 对角点  $(0, 0)$  和  $(0, 2)$ ：该子矩阵中有3个整数，分别是 20, 0 和 15, 它们的最大公约数是 5。
- 对角点  $(0, 0)$  和  $(1, 1)$ ：该子矩阵中有3个整数，分别是 20, 0, 0 and 12, 它们的最大公约数是 4。

现在假设 Bazza 做了如下修改：

- 将单元格  $(0, 1)$  的值修改为 6；
- 将单元格  $(1, 1)$  的值修改为 14。

20	6	15
0	14	0

上图显示了网格的新状态。Bazza 于是就想知道如下子矩阵的GCD：

- 对角点  $(0, 0)$  和  $(0, 2)$ ：现在这个子矩阵中包含3个整数 20, 6 和 15, 它们的最大公约数是 1。
- 对角点  $(0, 0)$  和  $(1, 1)$ ：现在这个子矩阵包含4个整数 20, 6, 0 和 14, 它们的最大公约数是 2。

至此，Bazza 做了  $N_U = 5$  次更新和  $N_Q = 4$  询问。

## 实现

你需要提交一个文件实现函数 `init()`，`update()`，`calculate()`，说明如下：

你的电脑上提供了模板程序 (`game.c`，`game.cpp` 和 `game.pas`)，每个程序中都包含一个函数 `gcd2(X, Y)`，该函数计算两个给定非负整数  $X$  和  $Y$  的最大公约数。如果  $X = Y = 0$ ，那么 `gcd2(X, Y)` 会返回 0。

该函数很快，可以得到满分。具体说，在最坏情况下，该函数的运行时间与  $\log(X + Y)$  成线性关系。

## 你的函数: **init()**

C/C++ `void init(int R, int C);`

Pascal `procedure init(R, C : LongInt);`

### 描述

你的提交必须实现该函数。

该函数传给你网格的大小，允许你初始化任何全局变量和数据结构。它只能在你调用 `update()` 和 `calculate()` 之前被调用1次。

### 参数

- `R`: 行数。
- `C`: 列数。

## 你的函数 **update()**

C/C++ `void update(int P, int Q, long long K);`

Pascal `procedure update(P, Q : LongInt; K : Int64);`

### 描述

你的提交必须实现该函数。

当Bazza改变单元格中的整数时调用此函数。

### 参数

- `P`: 单元格的行号( $0 \leq P \leq R-1$ )。
- `Q`: 单元格的列号( $0 \leq Q \leq C-1$ )。
- `K`: 这个单元格中新的整数( $0 \leq K \leq 10^{18}$ )。这个新整数可能与原来的整数相同。

## 你的函数: `calculate()`

C/C++ `long long calculate(int P, int Q, int U, int V);`

Pascal `function calculate(P, Q, U, V : LongInt) : Int64;`

### 描述

你的提交必须实现该函数。

该函数计算以  $(P, Q)$  和  $(U, V)$  为对角点的子矩阵中所有整数的最大公约数。这个范围是包含单元格  $(P, Q)$  和  $(U, V)$  的。

如果这个子矩阵中的所有整数都是0，那么该函数返回0。

### 参数

- $P$ : 子矩阵左上角单元格的行号 ( $0 \leq P \leq R-1$ )。
- $Q$ : 子矩阵左上角单元格的列号 ( $0 \leq Q \leq C-1$ )。
- $U$ : 子矩阵右下角单元格的行号 ( $P \leq U \leq R-1$ )。
- $V$ : 子矩阵右下角单元格的列号 ( $Q \leq V \leq C-1$ )。
- *Returns*: 子矩阵中所有整数的GCD或者0（如果所有的整数都是0）。

## 样例

下面说明前面的例子：

函数调用	返回值
<code>init(2, 3)</code>	
<code>update(0, 0, 20)</code>	
<code>update(0, 2, 15)</code>	
<code>update(1, 1, 12)</code>	
<code>calculate(0, 0, 0, 2)</code>	5
<code>calculate(0, 0, 1, 1)</code>	4
<code>update(0, 1, 6)</code>	
<code>update(1, 1, 14)</code>	
<code>calculate(0, 0, 0, 2)</code>	1
<code>calculate(0, 0, 1, 1)</code>	2

## 限制

- 时间限制: 见子任务。
- 内存限制: 见子任务。
- $1 \leq R, C \leq 10^9$
- $0 \leq K \leq 10^{18}$ , K表示Bazza放到单元格中的数字。

## 子任务

关于子任务的参数，请查阅英文版本。

Subtask	Points	R	C	$N_U$	$N_Q$	Time limit	Memory limit

---

## 评测

你电脑上的样例评分程序从文件 `game.in` 中读入。该文件格式如下：

- 第1行: `R C N`
- 接下来的 `N` 行: 每行表示一个动作，以动作发生的先后顺序给出。

表示每个动作的一行的格式如下：

- `update(P, Q, K)` 表示为: `1 P Q K`
- `calculate(P, Q, U, V)` 表示为: `2 P Q U V`

例如，上面的例子的输入文件格式如下：

```
2 3 9
1 0 0 20
1 0 2 15
1 1 1 12
2 0 0 0 2
2 0 0 1 1
1 0 1 6
1 1 1 14
2 0 0 0 2
2 0 0 1 1
```

---

## 编程语言说明

C/C++ 你必须 `#include "game.h"` .

Pascal 你必须定义 `unit Game` . 所有数组从 `0` (而非 `1`) 开始。

因为单元格中的整数可能很大，建议C/C++用户 使用 `long long` 类型，Pascal用户 使用 `Int64` 类型。