

## CCF 全国信息学奥林匹克联赛（NOIP2014）复赛

## 普及组

（请选手务必仔细阅读本页内容）

## 一. 题目概况

中文题目名称	珠心算测试	比例简化	螺旋矩阵	子矩阵
英文题目与子目录名	count	ratio	matrix	submatrix
可执行文件名	count	ratio	matrix	submatrix
输入文件名	count.in	ratio.in	matrix.in	submatrix.in
输出文件名	count.out	ratio.out	matrix.out	submatrix.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
测试点数目	10	10	10	20
每个测试点分值	10	10	10	5
附加样例文件	有	有	有	有
结果比较方式	全文比较（过滤行末空格及文末回车）			
题目类型	传统	传统	传统	传统
运行内存上限	128M	128M	128M	128M

## 二. 提交源程序文件名

对于 C++ 语言	count.cpp	ratio.cpp	matrix.cpp	submatrix.cpp
对于 C 语言	count.c	ratio.c	matrix.c	submatrix.c
对于 pascal 语言	count.pas	ratio.pas	matrix.pas	submatrix.pas

## 三. 编译命令（不包含任何优化开关）

对于 C++ 语言	g++ -o count count.cpp -lm	g++ -o ratio ratio.cpp -lm	g++ -o matrix matrix.cpp -lm	g++ -o submatrix submatrix.cpp -lm
对于 C 语言	gcc -o count count.c -lm	gcc -o ratio ratio.c -lm	gcc -o matrix matrix.c -lm	gcc -o submatrix submatrix.c -lm
对于 pascal 语言	fpc count.pas	fpc ratio.pas	fpc matrix.pas	fpc submatrix.pas

## 注意事项：

- 1、文件名（程序名和输入输出文件名）必须使用英文小写。
- 2、C/C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
- 3、全国统一评测时采用的机器配置为：CPU AMD Athlon(tm) 64x2 Dual Core CPU 5200+，2.71GHz，内存 2G，上述时限以此配置为准。
- 4、只提供 Linux 格式附加样例文件。
- 5、特别提醒：评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以其为准。

## 1. 珠心算测验

(count.cpp/c/pas)

### 【问题描述】

珠心算是一种通过在脑中模拟算盘变化来完成快速运算的一种计算技术。珠心算训练，既能够开发智力，又能够为日常生活带来很多便利，因而在很多学校得到普及。

某学校的珠心算老师采用一种快速考察珠心算加法能力的测验方法。他随机生成一个正整数集合，集合中的数各不相同，然后要求学生回答：其中有多少个数，恰好等于集合中另外两个（不同的）数之和？

最近老师出了一些测验题，请你帮忙求出答案。

### 【输入】

输入文件名为 `count.in`。

输入共两行，第一行包含一个整数  $n$ ，表示测试题中给出的正整数个数。

第二行有  $n$  个正整数，每两个正整数之间用一个空格隔开，表示测试题中给出的正整数。

### 【输出】

输出文件名为 `count.out`。

输出共一行，包含一个整数，表示测验题答案。

### 【输入输出样例】

<code>count.in</code>	<code>count.out</code>
4 1 2 3 4	2

### 【样例说明】

由  $1+2=3$ ， $1+3=4$ ，故满足测试要求的答案为 2。注意，加数和被加数必须是集合中的两个不同的数。

### 【数据说明】

对于 100% 的数据， $3 \leq n \leq 100$ ，测验题给出的正整数大小不超过 10,000。

2. 比例简化

(ratio.cpp/c/pas)

【问题描述】

在社交媒体上，经常会看到针对某一个观点同意与否的民意调查以及结果。例如，对某一观点表示支持的有 1498 人，反对的有 902 人，那么赞同与反对的比例可以简单的记为 1498:902。

不过，如果把调查结果就以这种方式呈现出来，大多数人肯定不会满意。因为这个比例的数值太大，难以一眼看出它们的关系。对于上面这个例子，如果把比例记为 5:3，虽然与真实结果有一定的误差，但依然能够较为准确地反映调查结果，同时也显得比较直观。

现给出支持人数  $A$ ，反对人数  $B$ ，以及一个上限  $L$ ，请你将  $A$  比  $B$  化简为  $A'$  比  $B'$ ，要求在  $A'$  和  $B'$  均不大于  $L$  且  $A'$  和  $B'$  互质（两个整数的最大公约数是 1）的前提下， $A'/B' \geq A/B$  且  $A'/B' - A/B$  的值尽可能小。

【输入】

输入文件名为 ratio.in。

输入共一行，包含三个整数  $A, B, L$ ，每两个整数之间用一个空格隔开，分别表示支持人数、反对人数以及上限。

【输出】

输出文件名为 ratio.out。

输出共一行，包含两个整数  $A', B'$ ，中间用一个空格隔开，表示化简后的比例。

【输入输出样例】

ratio.in	ratio.out
1498 902 10	5 3

【数据说明】

对于 100% 的数据， $1 \leq A \leq 1,000,000, 1 \leq B \leq 1,000,000, 1 \leq L \leq 100, A/B \leq L$ 。

### 3. 螺旋矩阵

(**matrix.cpp/c/pas**)

#### 【问题描述】

一个  $n$  行  $n$  列的螺旋矩阵可由如下方法生成：

从矩阵的左上角（第 1 行第 1 列）出发，初始时向右移动；如果前方是未曾经过的格子，则继续前进，否则右转；重复上述操作直至经过矩阵中所有格子。根据经过顺序，在格子中依次填入  $1, 2, 3, \dots, n^2$ ，便构成了一个螺旋矩阵。

下图是一个  $n = 4$  时的螺旋矩阵。

1	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

现给出矩阵大小  $n$  以及  $i$  和  $j$ ，请你求出该矩阵中第  $i$  行第  $j$  列的数是多少。

#### 【输入】

输入文件名为 **matrix.in**。

输入共一行，包含三个整数  $n, i, j$ ，每两个整数之间用一个空格隔开，分别表示矩阵大小、待求的数所在的行号和列号。

#### 【输出】

输出文件名为 **matrix.out**。

输出共一行，包含一个整数，表示相应矩阵中第  $i$  行第  $j$  列的数。

#### 【输入输出样例】

<b>matrix.in</b>	<b>matrix.out</b>
4 2 3	14

#### 【数据说明】

对于 50% 的数据， $1 \leq n \leq 100$ ；

对于 100% 的数据， $1 \leq n \leq 30,000, 1 \leq i \leq n, 1 \leq j \leq n$ 。

4. 子矩阵

(submatrix.cpp/c/pas)

【问题描述】

给出如下定义：

1. 子矩阵：从一个矩阵当中选取某些行和某些列交叉位置所组成的新矩阵（保持行与列的相对顺序）被称为原矩阵的一个子矩阵。

例如，下面左图中选取第 2、4 行和第 2、4、5 列交叉位置的元素得到一个 2\*3 的子矩阵如右图所示。

9	3	3	3	9
9	4	8	7	4
1	7	4	6	6
6	8	5	6	9
7	4	5	6	1

的其中一个 2\*3 的子矩阵是

4	7	4
8	6	9

2. 相邻的元素：矩阵中的某个元素与其上下左右四个元素（如果存在的话）是相邻的。
3. 矩阵的分值：矩阵中每一对相邻元素之差的绝对值之和。

本题任务：给定一个 n 行 m 列的正整数矩阵，请你从这个矩阵中选出一个 r 行 c 列的子矩阵，使得这个子矩阵的分值最小，并输出这个分值。

【输入】

输入文件名为 submatrix.in。

第一行包含用空格隔开的四个整数 n，m，r，c，意义如问题描述中所述，每两个整数之间用一个空格隔开。

接下来的 n 行，每行包含 m 个用空格隔开的整数，用来表示问题描述中那个 n 行 m 列的矩阵。

【输出】

输出文件名为 submatrix.out。

输出共 1 行，包含 1 个整数，表示满足题目描述的子矩阵的最小分值。

【输入输出样例 1】

submatrix.in	submatrix.out
5 5 2 3 9 3 3 3 9 9 4 8 7 4 1 7 4 6 6 6 8 5 6 9 7 4 5 6 1	6

【输入输出样例 1 说明】

该矩阵中分值最小的 2 行 3 列的子矩阵由原矩阵的第 4 行、第 5 行与第 1 列、第 3 列、

第 4 列交叉位置的元素组成，为 $\begin{smallmatrix} 6 & 5 & 6 \\ 7 & 5 & 6 \end{smallmatrix}$ ，其分值为 $|6 - 5| + |5 - 6| + |7 - 5| + |5 - 6| + |6 - 7| + |5 - 5| + |6 - 6| = 6$ 。

【输入输出样例 2】

submatrix.in	submatrix.out
7 7 3 3 7 7 7 6 2 10 5 5 8 8 2 1 6 2 2 9 5 5 6 1 7 7 9 3 6 1 7 8 1 9 1 4 7 8 8 10 5 9 1 1 8 10 1 3 1 5 4 8 6	16

【输入输出样例 2 说明】

该矩阵中分值最小的 3 行 3 列的子矩阵由原矩阵的第 4 行、第 5 行、第 6 行与第 2 列、第 6 列、第 7 列交叉位置的元素组成，选取的分值最小的子矩阵为

9 7 8

9 8 8

5 8 10

【数据说明】

对于 50%的数据， $1 \leq n \leq 12, 1 \leq m \leq 12$ , 矩阵中的每个元素  $1 \leq a[i][j] \leq 20$ ;  
对于 100%的数据， $1 \leq n \leq 16, 1 \leq m \leq 16$ , 矩阵中的每个元素  $1 \leq a[i][j] \leq 1000$ ,  
 $1 \leq r \leq n, 1 \leq c \leq m$ 。