全国青少年信息学奥林匹克联赛

模拟试题

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文题目名称 | 平方 | 翻转 | 冲锋 | 蛋糕 |
| 英文题目名称 | square | flip | go | cake |
| 输入文件名 | square.in | flip.in | go.in | cake.in |
| 输出文件名 | square.out | flip.out | go.out | cake.out |
| 测试点时限 | 1秒 | 3秒 | 1秒 | 5秒 |
| 内存限制 | 1024MB | 1024MB | 1024MB | 1024MB |
| 测试点数目 | 20 | 20 | 20 | 20 |
| 测试点分值 | 5 | 5 | 5 | 5 |
| 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 |

提交源程序须加后缀

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 对于C语言 | square.c | flip.c | go.c | cake.c |
| 对于C++ 语言 | square.cpp | flip.cpp | go.cpp | cake.cpp |

注意：最终测试时，所有编译命令均不打开任何优化开关。

平方

【问题描述】

给定序列，我们想算出另一个序列满足

1. 对于所有的；
2. 对于所有的是完全平方数；
3. 最小。

请输出最小的。

【输入格式】

第一行一个整数。

第二行个整数。

【输出格式】

一行一个数表示答案。请输出答案。

【样例输入1】

3

2 3 6

【样例输出1】

6

【样例输入2】

见下发文件 square/square2.in

【样例输出2】

见下发文件square/square2.ans

【数据规模与约定】

所有测试点的数据规模与约定如下：

|  |  |
| --- | --- |
| 测试点编号 | *ai*的规模 |
| 1 | *ai* ≤ 10 |
| 2 |
| 3 |
| 4 |
| 5 | *ai* ≤ 106 |
| 6 |
| 7 |
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |

对于所有数据，,。

翻转

【问题描述】

有一个的棋盘。初始每个格子都是好的，之后每一秒会坏一个之前没坏的格子。秒过后所有格子都会坏完。

我们想知道，每一秒过后，有多少子矩形满足其中的所有格子都是好的。

【输入格式】

第一行两个整数。

接下来行，每行表示一个坏掉的格子。

数据保证每个格子只会坏一次。

【输出格式】

输出行。第行表示前个格子坏掉后有多少子矩形是好的。

【样例输入1】

2 2

1 1

2 1

1 2

2 2

【样例输出1】

5

3

1

0

【样例输入2】

见下发文件 flip/flip2.in

【样例输出2】

见下发文件flip/flip2.ans

【数据规模与约定】

所有测试点的数据规模与约定如下：

|  |  |
| --- | --- |
| 测试点编号 | *n,m*的规模 |
| 1 | *n,m* ≤ 10 |
| 2 |
| 3 |
| 4 |
| 5 | *n,m* ≤ 100 |
| 6 |
| 7 |
| 8 |
| 9 | *n,m* ≤ 500 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |

对于所有数据，。

冲锋

【问题描述】

有一个的网格图，第行第列的格子的坐标为，也就是说左上角的格子的坐标为*，*右下角的坐标为。

现在有个兄弟要从走到，每次他只能走上下左右四个方向，这个兄弟永远不能走到网格图外面。

假设当前位置为，

向南走一步（加1）的代价为；

向北走一步（减1）的代价为；

向东走一步（加1）的代价为；

向西走一步（减1）的代价为；

请问从走到的最小代价是多少？

【输入格式】

输入第一行一个正整数表示数据组数。

接下来行，每行四个整数。

【输出格式】

对于每组数据，一行一个整数表示答案。请输出答案。

【样例输入1】

4

1 1 1 1

1 3 2 1

2 1 1 3

10 2 20 6

【样例输出1】

0

12

17

16999

【样例输入2】

见下发文件 go/go2.in

【样例输出2】

见下发文件 go/go2.ans

【数据规模与约定】

所有测试点的数据规模与约定如下：

|  |  |  |
| --- | --- | --- |
| 测试点编号 | *n*的规模 | *x,y*的规模 |
| 1 | *n* ≤ 100 | *x,y* ≤ 500 |
| 2 |
| 3 |
| 4 |
| 5 | *n* ≤ 50000 |
| 6 |
| 7 |
| 8 |
| 9 | *x,y* ≤ 109 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |

对于所有数据，。

蛋糕

【问题描述】

今天是杜老师的生日，蜗蜗给杜老师买了个大蛋糕。蛋糕里一共有块巧克力，巧克力的形状都是长方体，第块巧克力包含了所有点满足。巧克力之间可能会有重叠。

杜老师想在上面切三刀。第一刀必须切一个的平面，第二刀必须切一个的平面，第三刀必须切一个的平面，其中为3个整数。与此同时，需要满足每一块巧克力都至少被其中一刀切到（切到的定义是巧克力和至少一个平面的交非空）。

请帮杜老师切这3刀。

【输入格式】

第一行一个整数。

接下来行，每行6个整数，，，，，。

【输出格式】

如果无解，输出NO。

否则第一行输出YES，第二行输出3个整数。

如果存在多解，输出任意一组即可。

【样例输入1】

3

0 1 0 1 0 1

10 11 10 11 10 11

999999999 1000000000 999999999 1000000000 999999999 1000000000

【样例输出1】

YES

0 10 999999999

【样例输入2】

4

0 1 0 1 0 1

999999999 1000000000 0 1 0 1

0 1 999999999 1000000000 0 1

0 1 0 1 999999999 1000000000

【样例输出2】

YES

0 0 0

【样例输入3】

见下发文件 cake/cake3.in

【样例输出2】

见下发文件 cake/cake3.ans

【数据规模与约定】

所有测试点的数据规模与约定如下：

|  |  |
| --- | --- |
| 测试点编号 | *n*的规模 |
| 1 | *n* ≤ 100 |
| 2 |
| 3 | *n* ≤ 1000 |
| 4 |
| 5 |
| 6 |
| 7 |
| 8 |
| 9 | *n* ≤ 100000 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |

对于所有数据，,,。