NOIP提高组复赛模拟

一、题目概况

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文题目名称 | 五子棋 | | 迷宫 | 雪人 |
| 英文题目与子目录名 | FIR | | maze | snowman |
| 可执行文件名 | FIR | | maze | snowman |
| 输入文件名 | FIR.in | | maze.in | snowman.in |
| 输出文件名 | FIR.out | | maze.out | snowman.out |
| 每个测试点时限 | 1秒 | | 2秒 | 2秒 |
| 附加样例文件 | 无 | | 无 | 无 |
| 题目类型 | 传统 | | 传统 | 传统 |
| 运行内存上限 | 512M | | 512M | 512M |
| 结果比较方式 | | 全文比较（过滤行末空格及文末回车） | | |
| 编译优化 | | 无 | | |

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、只提供 Linux 格式附加样例文件。

4、提交的程序代码文件的放置位置请参照各省的具体要求。

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. **五子棋**

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

**【题目背景】**

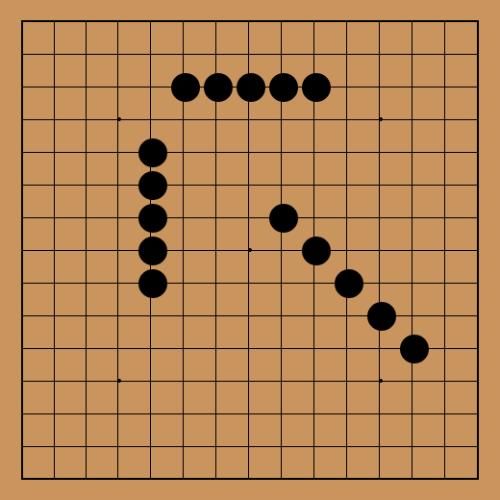
ITer是一头大象，作为一个象生赢家，ITer当然有很多母象，这天，ITer和他的其中一个母象Kitty van ♂ 游戏。

**【问题描述】**

就像题目名字一样，他们在玩一种类似于五子棋的游戏，只是规则和五子棋有一些不同。

就像五子棋一样，ITer和Kitty依次在棋盘上放置棋子，ITer放置黑子，Kitty放置白子。与五子棋不同的是，当一种颜色的X个棋子组成一行或一列或一个对角线的时候（X不一定等于5），执该颜色的子的玩家胜利。

如图，当X=5时，ITer胜利的三种方法：



下了一会儿后，ITer和Kitty觉得无聊了。于是，他们又加了一个类似围棋的规则：当一个连通分量（四连通）的棋子的四周被另一种棋子围住时，这些棋子就消失了，我们称这些子被“吃掉”了。



如图，三颗白棋的四周都被黑棋围住，所以这三颗棋子就应该消失。成为下图：



现在ITer和Kitty下了N步棋，ITer先手，每次下棋的位置用坐标表示。

现在ITer想知道，下了N步棋后，是否有人赢了，在第几步赢了，是否有不合法的走法。

**【输入格式】**

输入文件名为FIR.in。

输入文件第一行两个正整数N，X，表示下了N步棋，当一种颜色的X个棋子组成一行或一列或一个对角线的时候，执该颜色的子的玩家胜利。

接下来的行，每行2个正整数Xi，Yi，当i为奇数时，表示ITer在第 i步下在了（Xi，Yi）上，当i为偶数时，表示Kitty在第 i步下在了（Xi，Yi）上。

**【输出格式】**

输出文件名为FIR.out。

如果ITer在前N步赢了，则输出“ITer X”（中间用空格连接），X表示ITer在第几步赢了。

如果Kitty在前N步赢了，则输出“Kitty X”（中间用空格连接），X表示Kitty在第几步赢了。

如果在N步以内有不合法的步，则输出“illegal”。

一步不合法有以下几种情况：

1. 这步下在了已经有棋子的格子上。
2. 在没有吃掉对方棋子的情况下，该步子所在的联通分量的棋子被对方的棋子吃掉。如图1，白子走在了红叉的位置上是不合法的，因为这样会使该联通分量被吃掉。



图1

如图2，如果白子走在红叉的位置上是合法的，因为它吃掉了右边的两个黑子。走完这步后白子不会消失，棋盘上的子变成图3。



图2



图3

请忽视某一个人赢了以后出现的不合法的步。

如果既没有出现不合法的步，又没有决出胜负，则输出“draw”。

假设棋盘是无限的。

**【输入输出样例1】**

|  |  |
| --- | --- |
| **FIR1.in** | **FIR1.out** |
| 20 5  2 3  3 3  3 2  2 2  4 3  2 4  5 3  1 3  3 4  5 5  2 3  3 3  2 3  2 5  6 3  1 5  3 3  1 5  1 5  1 5 | ITer 17 |

**【输入输出样例说明1】**

尽管第18,19,20步是不合法的，但是ITer在第17步就赢了，所以不会输出“illegal”。

**【输入输出样例2】**

|  |  |
| --- | --- |
| **FIR2.in** | **FIR2.out** |
| 20 6  2 3  3 3  3 2  2 2  4 3  2 4  5 3  1 3  3 4  5 5  2 3  3 3  2 3  2 5  6 3  1 5  3 3  6 1  6 2  5 1 | draw |

**【数据规模与约定】**

Subtask 1 30%：。

Subtask 2 70%：。

**2. 迷宫**

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

**【题目背景】**

小K是一个探险家，这天，他遇到了一个迷宫……

**【问题描述】**

迷宫可以抽象成一个矩阵，小K要从（1，1）走到（N，M），而且只能往下和往右走，即小K只能从（X，Y）走到（X，Y+1）和（X+1，Y）。小K不能走出迷宫（即X>N或Y>M）。当然，迷宫有一些格子是被堵住的，小K不能从这些格子经过。

每个没被堵住的格子都有一个权值，小K十分喜欢X这个数字和异或这个运算。所以他希望所有他经过的格子的异或和为X。现在小K想知道他有多少种走法，听说你是一位大佬，于是他向你求助。

**【输入格式】**

输入文件名为maze.in。

输入文件第一行为三个正整数，表示迷宫有N行M列，小K喜欢X。

接下来的行，每行个正整数，其中第行列的正整数代表。

输入保证A11>0且Anm>0。

**【输出格式】**

输出文件名为maze.out。

输出文件包含一行一个正整数，表示小K有多少种走法使得他经过的格子的异或和为X。

**【输入输出样例】**

|  |  |
| --- | --- |
| **maze.in** | **maze.out** |
| 3 3 1  1 1 5  2 3 1  0 4 5 | 2 |

**【输入输出样例说明】**

小K有2种走法，分别为：

（1，1）->（2，1）->（2，2）->（3，2）->（3，3） 1^2^3^4^5=1

（1，1）->（1，2）->（1，3）->（2，3）->（3，3） 1^1^5^1^5=1

**【数据规模与约定】**

Subtask 1 30%：

Subtask 2 20% : ，

Subtask 3 50%：

1. **雪人**

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

**【题目背景】**

大佬WZY在AK NOIP 2018后，决定去冰天雪地的Y城堆雪人。

**【问题描述】**

WZY堆了N个雪人，每个雪人都有一个可爱度Xi，WZY认为两串雪人a1,a2…an与b1,,b2…bm和谐当且仅当

1. n=m
2. a1-b1=a2­-b2=…=an-bn

WZY现在要从一堆雪人中选择两串雪人A=[l1,r1],B=[l2,r2]（两串可以重叠，即若l1≤l2，l2可以小于等于r1），使得A和B和谐，现在WZY想知道对于所有的方案中，min（|l1-l2|,len(A)）的最大值。

len（A）为A中所含雪人的个数。因为WZY还要准备AK JSOI 2019，所以他把这个问题交给了你。

**【输入格式】**

输入文件名为snowman.in。

输入文件第一行为一个正整数，表示雪人的个数。

输入文件第二行为N个整数：X1，X2，…XN。

**【输出格式】**

输出文件名为snowman.out。

输出文件仅一行一个整数，为min（|l1-l2|,len(A)）的最大值。

**【输入输出样例1】**

|  |  |
| --- | --- |
| **snowman.in** | **snowman.out** |
| 10  1 2 3 4 5 6 7 8 9 10 | 5 |

**【输入输出样例1说明】**

应选A=[1,5],B=[6,10]，此时，min(|6-1|,5)=5;

**【数据规模与约定】**

Subtask 1 10%：。

Subtask 2 20%：。

Subtask 3 50%：

Subtask 4 20%：