

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】

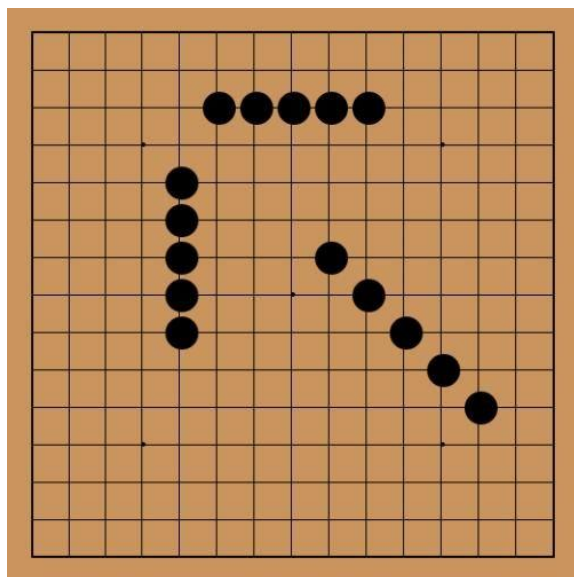
ITer 是一头大象，作为一个象生赢家，ITer 当然有很多母象，这天，ITer 和他的其中一个母象 Kitty van ♂ 游戏。

【问题描述】

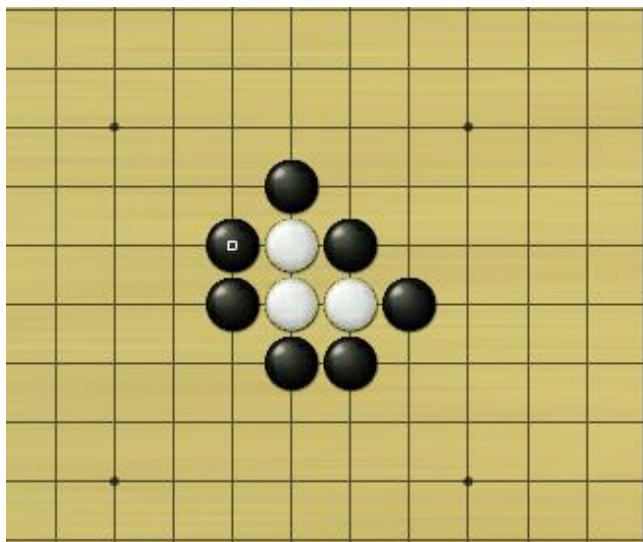
就像题目名字一样，他们在玩一种类似于五子棋的游戏，只是规则和五子棋有一些不同。

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。

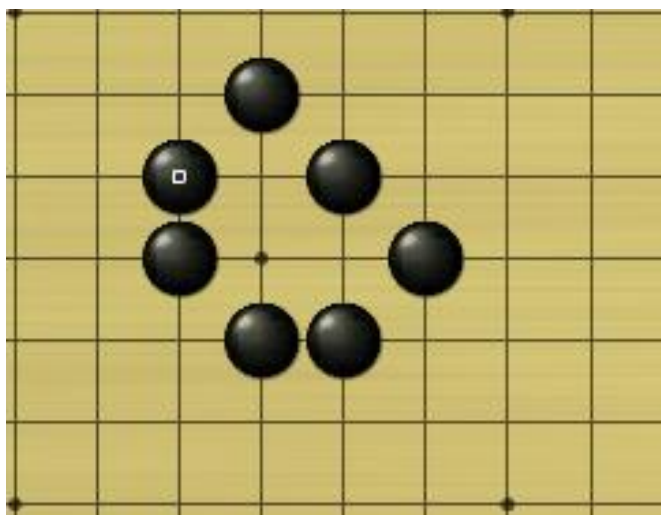
如图，当 $x=5$ 时，ITer 胜利的三种方法：



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

现在 I Ter 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 N 行，每行 2 个正整数 X_i, Y_i ，当 i 为奇数时，表示 I Ter 在第 i 步下在了 (X_i, Y_i) 上，当 i 为偶数时，表示 Kitty 在第 i 步下在了 (X_i, Y_i) 上。

【输出格式】

输出文件名为 `FIR.out`。

如果 I Ter 在前 N 步赢了，则输出“I Ter X”（中间用空格连接）， X 表示 I Ter 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。

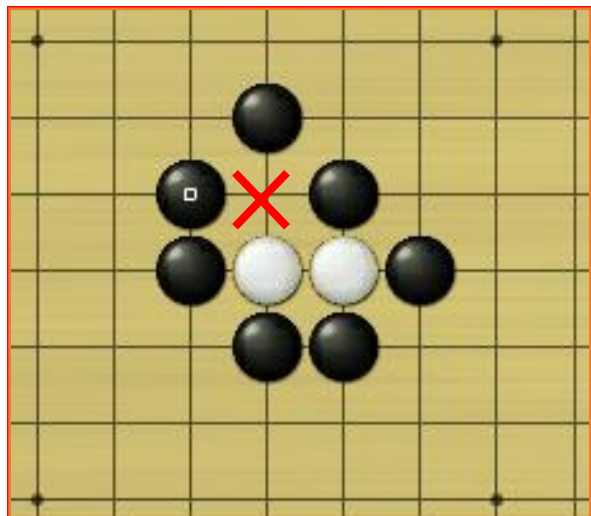


图 1

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。

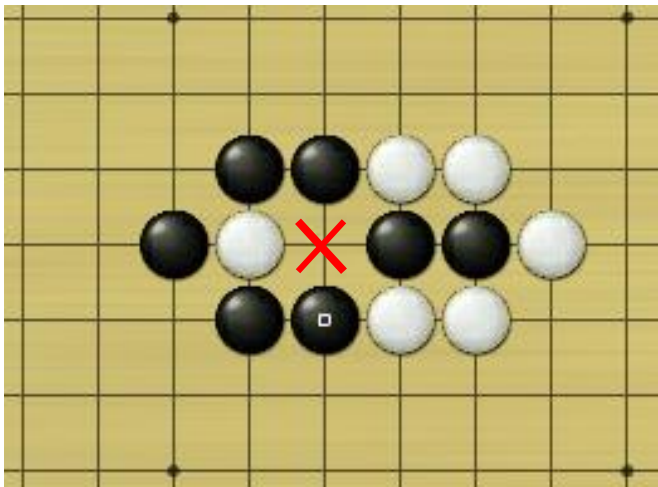


图 2

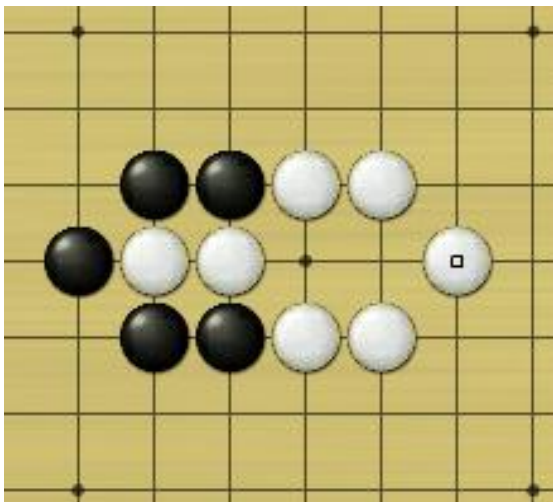


图 3

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	ITer 17

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	

【输入输出样例说明 1】

尽管第 18,19,20 步是不合法的，但是 I Ter 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	draw
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	

【数据规模与约定】

Subtask 1 30% : $N \leq 20, X \leq 5, X_i, Y_i \leq 10$ 。

Subtask 2 70% : $N \leq 1000, X \leq 10, 1 \leq X_i, Y_i \leq 1000$ 。

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 N, M, X ，表示迷宫有 N 行 M 列，小 K 喜欢 X 。

接下来的 N 行，每行 M 个正整数，其中第 i 行 j 列的正整数代表 A_{ij} ，若 $A_{ij} = 0$ ，则表示第 i 行 j 列被堵住了，若 $A_{ij} > 0$ ，这表示第 i 行 j 列没有被堵住，且权值为 A_{ij} 。

输入保证 $A_{11} > 0$ 且 $A_{nm} > 0$ 。

【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：

$(1, 1) \rightarrow (2, 1) \rightarrow (2, 2) \rightarrow (3, 2) \rightarrow (3, 3)$ $1^2 3^4 5^1 = 1$

$(1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3) \rightarrow (3, 3)$ $1^1 1^5 1^5 = 1$

【数据规模与约定】

Subtask 1 30% : $N, M \leq 10$

Subtask 2 20% : $N, M \leq 20$, $0 \leq A_{ij} \leq 10,000$

Subtask 3 50% : $N, M \leq 20, 0 \leq A_{ij} \leq 1,000,000,000$

3. 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 x_i ，WZY 认为两串雪人 a_1, a_2, \dots, a_n 与 b_1, b_2, \dots, b_m 和谐当且仅当

1. $n=m$
2. $a_1-b_1=a_2-b_2=\dots=a_n-b_n$

WZY 现在要从一堆雪人中选择两串雪人 $A=[l_1, r_1], B=[l_2, r_2]$ (两串可以重叠，即若 $l_1 \leq l_2$ ， l_2 可以小于等于 r_1)，使得 A 和 B 和谐，现在 WZY 想知道对于所有的方案中， $\min(|l_1-l_2|, \text{len}(A))$ 的最大值。

$\text{len}(A)$ 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 N ，表示雪人的个数。

输入文件第二行为 N 个整数： x_1, x_2, \dots, x_N 。

【输出格式】

输出文件名为 snowman.out。

输出文件仅一行一个整数，为 $\min(|l_1-l_2|, \text{len}(A))$ 的最大值。

【输入输出样例 1】

snowman.in	snowman.out
------------	-------------

10	5
1 2 3 4 5 6 7 8 9 10	

【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% : $N \leq 500$ 。

Subtask 2 20% : $N \leq 5000$ 。

Subtask 3 50% : $N \leq 100000$

Subtask 4 20% : $N \leq 500000, 0 \leq X_i \leq 100000000$