暑假集训模拟赛（一）

(时间：8:30-11:30)

难度：普及组

一. 题目概览

中文题目名称	ISBN 号码	排座椅	传球游戏	立体图
英文题目名称	isbn	seat	ball	drawing
可执行文件名	isbn	seat	ball	drawing
输入文件名	isbn.in	seat.in	ball.in	drawing.in
输出文件名	isbn.out	seat.out	ball.out	drawing.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
比较方式	全文比较	全文比较	全文比较	全文比较
题目类型	传统	传统	传统	传统

二. 提交原程序文件名

对于 pascal 语言	isbn.pas	seat.pas	ball.pas	drawing.pas
对于 C 语言	isbn.c	seat.c	ball.c	drawing.c
对于 C++语言	isbn.cpp	seat.cpp	ball.cpp	drawing.cpp

三. 编译命令（不包含任何优化开关）

对于 pascal 语言	fpc isbn.pas	fpc seat.pas	fpc ball.pas	fpc drawing.pas
对于 C 语言	gcc –o isbn isbn.c	gcc –o seat seat.c	gcc –o ball ball.c	gcc –o drawing drawing.c
对于 C++语言	g++ -o isbn isbn.cpp	g++ -o seat seat.cpp	g++ -o ball ball.cpp	g++ -o drawing drawing.cpp

四. 运行内存限制

运行内存上限	50M	50M	50M	50M

注意事项：

1、 文件名（程序名和输入输出文件名）必须使用小写。

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 1.9GHz，内存 512M，上述时限以此配置为准。各省在自测时可根据具体配置调整时限。

1. ISBN 号码

(isbn.c/cpp)

【问题描述】

每一本正式出版的图书都有一个 ISBN 号码与之对应，ISBN 码包括 9 位数字、1 位识别码和 3 位分隔符，其规定格式如“x-xxx-xxxxx-x”，其中符号“-”就是分隔符（键盘上的减号），最后一位是识别码，例如 0-670-82162-4 就是一个标准的 ISBN 码。ISBN 码的首位数字表示书籍的出版语言，例如 0 代表英语；第一个分隔符“-”之后的三位数字代表出版社，例如 670 代表维京出版社；第二个分隔符后的五位数字代表该书在该出版社的编号；最后一位为识别码。

识别码的计算方法如下：

首位数字乘以 1 加上次位数字乘以 2……以此类推，用所得的结果 mod 11，所得的余数即为识别码，如果余数为 10，则识别码为大写字母 X。例如 ISBN 号码 0-670-82162-4 中的识别码 4 是这样得到的：对 067082162 这 9 个数字，从左至右，分别乘以 1，2，...,9,再求和，即 0×1+6×2+……+2×9=158，然后取 158 mod 11 的结果 4 作为识别码。

你的任务是编写程序判断输入的 ISBN 号码中识别码是否正确，如果正确，则仅输出“Right”；如果错误，则输出你认为是正确的 ISBN 号码。

4

【输入】

输入文件 isbn.in 只有一行，是一个字符序列，表示一本书的 ISBN 号码（保证输入符合 ISBN 号码的格式要求）。

【输出】

输出文件 isbn.out 共一行，假如输入的 ISBN 号码的识别码正确，那么输出“Right”，否则，按照规定的格式，输出正确的 ISBN 号码（包括分隔符“-”）。

【输入输出样例 1】

isbn.in	isbn.out
0-670-82162-4	Right

【输入输出样例 2】

isbn.in	isbn.out
0-670-82162-0	0-670-82162-4

2. 排座椅

(seat.c/cpp)

【问题描述】

上课的时候总会有一些同学和前后左右的人交头接耳，这是令小学班主任十分头疼的一件事情。不过，班主任小雪发现了一些有趣的现象，当同学们的座次确定下来之后，只有有限的 D 对同学上课时会交头接耳。同学们在教室中坐成了 M 行 N 列，坐在第 i 行第 j 列的

同学的位置是（i,j），为了方便同学们进出，在教室中设置了 K 条横向的通道，L 条纵向的通道。于是，聪明的小雪想到了一个办法，或许可以减少上课时学生交头接耳的问题：她打算重新摆放桌椅，改变同学们桌椅间通道的位置，因为如果一条通道隔开了两个会交头接耳的同学，那么他们就不会交头接耳了。

请你帮忙给小雪编写一个程序，给出最好的通道划分方案。在该方案下，上课时交头接耳的学生的对数最少。

【输入】

输入文件 seat.in 的第一行，有 5 个用空格隔开的整数，分别是 M，N，K，L，D（2<=N，M<=1000，0<=K<M，0<=L<N，D<=2000）。

接下来的 D 行，每行有 4 个用空格隔开的整数。第 i 行的 4 个整数 X_i，Y_i，P_i，Q_i，表示坐在位置(X_i, Y_i)与(P_i, Q_i)的两个同学会交头接耳（输入保证他们前后相邻或者左右相邻）。

输入数据保证最优方案的唯一性。

【输出】

输出文件 seat.out 共两行。

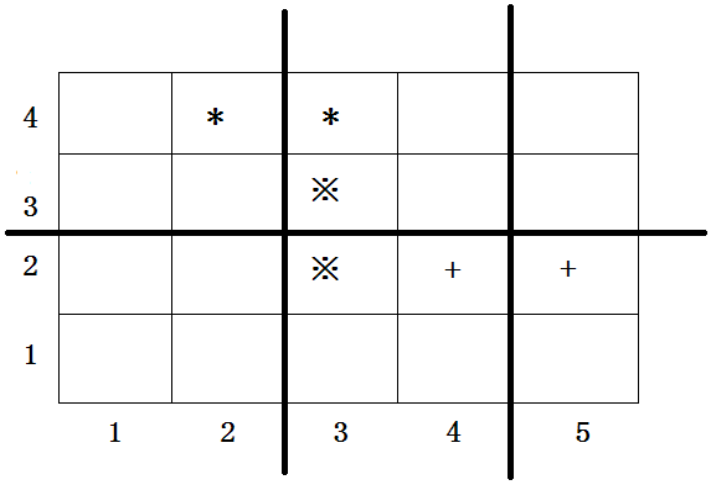
第一行包含 K 个整数，a_1，a_2……a_K，表示第 a_1 行和 a_1+1 行之间、第 a_2 行和 a_2+1 行之间、…、第 a_K 行和第 a_K+1 行之间要开辟通道，其中 a_i< a_{i+1}，每两个整数之间用空格隔开（行尾没有空格）。

第二行包含 L 个整数，b_1，b_2……b_L，表示第 b_1 列和 b_1+1 列之间、第 b_2 列和 b_2+1 列之间、…、第 b_L 列和第 b_L+1 列之间要开辟通道，其中 b_i< b_{i+1}，每两个整数之间用空格隔开（列尾没有空格）。

【输入输出样例】

seat.in	seat.out
4 5 1 2 3 4 2 4 3 2 3 3 3 2 5 2 4	2 2 4

【输入输出样例解释】



上图中用符号*、※、+标出了 3 对会交头接耳的学生的位置，图中 3 条粗线的位置表示通道，图示的通道划分方案是唯一的最佳方案。

3．传球游戏

(ball.c/cpp)

【问题描述】

上体育课的时候，小蛮的老师经常带着同学们一起做游戏。这次，老师带着同学们一起做传球游戏。

游戏规则是这样的：n 个同学站成一个圆圈，其中的一个同学手里拿着一个球，当老师吹哨子时开始传球，每个同学可以把球传给自己左右的两个同学中的一个（左右任意），当老师在此吹哨子时，传球停止，此时，拿着球没有传出去的那个同学就是败者，要给大家表演一个节目。

聪明的小蛮提出一个有趣的问题：有多少种不同的传球方法可以使得从小蛮手里开始传的球，传了 m 次以后，又回到小蛮手里。两种传球方法被视作不同的方法，当且仅当这两种方法中，接到球的同学按接球顺序组成的序列是不同的。比如有三个同学 1 号、2 号、3 号，并假设小蛮为 1 号，球传了 3 次回到小蛮手里的方式有 1->2->3->1 和 1->3->2->1，共 2 种。

【输入】

输入文件 ball.in 共一行，有两个用空格隔开的整数 n，m（3<=n<=30，1<=m<=30）。

【输出】

输出文件 ball.out 共一行，有一个整数，表示符合题意的方法数。

【输入输出样例】

ball.in	ball.out
3 3	2

【限制】

40%的数据满足：3<=n<=30，1<=m<=20

100%的数据满足：3<=n<=30，1<=m<=30

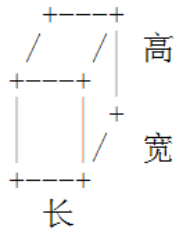
4．立体图

(drawing.c/cpp)

【问题描述】

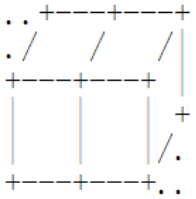
小渊是个聪明的孩子，他经常会给周围的小朋友们将写自己认为有趣的内容。最近，他准备给小朋友们讲解立体图，请你帮他画出立体图。

小渊有一块面积为 m*n 的矩形区域，上面有 m*n 个边长为 1 的格子，每个格子上堆了一些同样大小的积木（积木的长宽高都是 1），小渊想请你打印出这些格子的立体图。我们定义每个积木为如下格式，并且不会做任何翻转旋转，只会严格以这一种形式摆放：

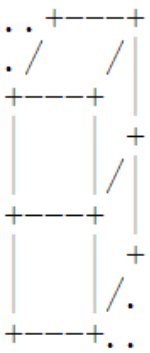


每个顶点用 1 个加号'+'表示，长用 3 个"-"表示，宽用 1 个"/"，高用两个"|"表示。字符'+','-','/','|'的 ASCII 码分别为 43，45，47，124。字符'.'(ASCII 码 46)需要作为背景输出，即立体图里的空白部分需要用'.'来代替。立体图的画法如下面的规则：

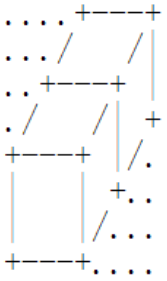
若两块积木左右相邻，图示为：



若两块积木上下相邻，图示为：



若两块积木前后相邻，图示为：



立体图中，定义位于第(m,1)的格子（即第 m 行第 1 列的格子）上面自底向上的第一块积木（即最下面的一块积木）的左下角顶点为整张图最左下角的点。

【输入】

输入文件 drawing.in 第一行有用空格隔开的 2 个整数 m 和 n，表示有 m*n 个格子（1<=m,n<=50）。

接下来的 m 行，是一个 m*n 的矩阵，每行有 n 个用空格隔开的整数，其中第 i 行第 j 列上的整数表示第 i 行第 j 列的个子上摞有多少个积木（1<=每个格子上的积木数<=100）。

【输出】

输出文件 drawing.out 中包含题目要求的立体图，是一个 K 行 L 列的字符串矩阵，其中 K 和 L 表示最少需要 K 行 L 列才能按规定输出立体图。

【输入输出样例】

drawing.in	drawing.out
3 4 2 2 1 2 2 2 1 1 3 2 1 2