模拟测试题

（请选手务必仔细阅读本页内容）

一、题目概况

中文题目名称	下棋	抓老鼠	观光电梯	采集
英文题目名称	chess	mouse	elevator	find
可执行文件名	chess	mouse	elevator	find
输入文件名	chess.in	mouse.in	elevator.in	find.in
输出文件名	chess.out	mouse.out	elevator.out	find.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
结果比较方式	逐行比较(忽略多余空格和制表符)			
附加样例文件	无	无	无	无
题目类型	传统	传统	传统	传统

二、提交源代码文件名

对于 C++语言	chess.cpp	mouse.cpp	elevator.cpp	find.cpp

三、编译命令（不包含任何优化开关）

对于C++语言	g++ -o matrix matrix.cpp -lm	g++ -o mouse mouse.cpp -lm	g++ -o elevator elevator.cpp -lm	g++ -o find find.cpp -lm

四、允许内存限制

内存上限	512M	512M	512M	512M

五、注意事项

1、 每位选手提交一个以自己编号命名的文件夹，在该文件夹下放4个源程序，然后再在该文件夹下建立四个子目录，名称分别为：chess、mouse、elevator、find，再把源代码放入对应的子目录中。

2、 文件夹名、文件名(程序名和输入输出文件名)必须使用英文小写。

3、 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。

4、 统一评测时采用的机器配置为:windows下lemon评测和全国评测系统下评测。

5、 最终测试时，所有编译命令均不打开任何优化开关。

6、 请尽力优化，会收获更多的部分得分。

7、 暴力出奇迹，打表拿省一！

1．下棋(chess.cpp)

【题目背景】

小 W 在家上网课，很无聊。这天他在家里发现了一个中国象棋的棋盘，他忽然想到了一个问题。

【问题描述】

在一个 n 行 m 列的棋盘上，放若干个炮（可以是 0 个），使得没有一个炮可以攻击到另一个炮。小 W 轻松地就构造出了可行的放法，但他还想知道一共有多少种方案。由于小 W 是一个蒟蒻，请你帮帮他。

小 W 的友情提示：在中国象棋中炮的行走方式是：一个炮攻击到另一个炮，当且仅当它们在同一行或同一列中，且它们之间恰好有一个棋子。

【输入】

输入文件 chess.in。

一行包含两个整数 n,m，之间由一个空格隔开。

【输出】

输出文件 chess.out。

总共的方案数，由于该值可能很大，只需给出方案数模 9999973 的结果。

【输入输出样例】

chess.in	chess.out
1 3	7

chess1.in	chess1.out
82 50	4920469

【样例解释】

对于第一个样例，除了放满的时候其他情况都是合法的，因此有 2^3-1=7 种情况。

【数据说明】

对于 30% 的数据，n 和 m 均不超过 6。

对于 50% 的数据，n 和 m 至少有一个数不超过 8。

对于 100% 的数据，100≤n,m≤100。

2. 抓老鼠(mouse.cpp)

【题目背景】

小 A 和小 B 在家里抓到了很多老鼠，其中有黑老鼠和白老鼠。小 A 和小 B 认为，白老鼠是幸运的老鼠，而他们想看看谁是幸运的人。所以，他们玩了一个游戏。

【问题描述】

袋子里有 w 只白鼠和 b 只黑鼠，小 A 和小 B 轮流从袋子里抓，谁先抓到白色谁就是幸运的。小 A 每次随机抓一只，小 B 每次随机抓完一只之后会有另一只随机老鼠跑出来。如果两个人都没有抓到白色则小 B 赢。小 A 先抓，问小 A 是幸运的人的概率。

【输入】

输入文件 mouse.in。

一行两个数 w, b。

【输出】

输出文件 mouse.out。

A 赢的概率，保留九位小数。

【输入输出样例】

mouse1.in	mouse1.out
1 3	0.500000000

mouse2.in	mouse2.out
5 5	0.658730159

mouse3.in	mouse3.out
200 200	0.666481173

【样例解释】

让我们看一下第一个样例。小 A 在第一回合中抓到一只白老鼠并立即获胜的概率是 1/4。小 B 在第一回合时抓到一只黑老鼠但没有获胜的概率为 3/4*2/3=1/2。之后，袋子里还有两只老鼠——一只黑一只白；其中一个跳了出来，另一个在第二回合被小 A 抓到。如果小 A 的老鼠是白色的，他获胜（概率为 1/2*1/2=1/4），否则没有人得到白色的老鼠。

因此，小 A 的胜率为 0.5

【数据说明】

对于 30%的数据：w,b<=10

对于 100%的数据：w,b<=1000

3. 观光电梯

(elevator.cpp)

【问题描述】

一个 9 层的楼有一个可以容纳 4 个人的电梯，你要管理这个电梯。

现在各层楼上有一些在排队的人，你知道他们在哪层要到哪层去。你也知道他们到电梯门口的顺序。根据公司的规定，如果一个人比其他人早到，他必须先进电梯（无论楼层，只凭时间）。注意人们离开电梯的时间不受限制。

电梯有两个命令：

1. 上楼或者下楼，代价为 1

2. 打开当前楼层的门，所有到目的地的人会从电梯里出来，当前楼层排队的人会在不违反公司规定的情况下一个一个进（在电梯还有空间的情况下）每个人用 1s 时间来出入电梯。

最初电梯是空的，在 1 楼。你需要求出最少用多长时间来把所有人送回到目的地。最后电梯可以停在任意位置

【输入】

输入文件 elevator.in。

第一行一个整数:人数量 N

之后 n 行：第 i 行包含两个整数 Ai,Bi。Ai 表示最初楼层,Bi 表示目的楼层。到达电梯门口的顺序按输入顺序排序

【输出】

输出文件 elevator.out。

一行一个数表示最小时间。

【输入输出样例】

elevator1.in	**elevator1.out**
2 3 5 5 3	10

elevator2.in	**elevator2.out**
2 5 3 3 5	12

【样例解释】

对于样例 2：

1.先到 5 楼，花费时间 5

2.1 号人进入，花费时间 1

3.再到 3 楼，花费时间 2

4.2 号人进入，花费时间 1。

5.1 号人出去，花费时间 1

6.到 5 楼，花费时间 2

【数据说明】

30%：n<=15

100%:$n \leq 2000$

4. 采集

(find.cpp)

【问题描述】

游戏的世界是一个二维坐标。X 轴、Y 轴坐标范围均为 1…N。初始的时候，所有的整数坐标点上均有一块金子，共 N^2 块。

一阵风吹过，金子的位置发生了一些变化。初始在（i,j）坐标处的金子会变到(f(i),f(j))坐标。其中 f(x)表示 x 各位数字的乘积，例如 f(99)=81,f(12)=2,f(10)=0。

如果金子变化后的坐标不在 1…N 的范围内，我们认为这块金子已经被移出游戏。同时可以发现，对于变化之后的游戏局面，某些坐标上的金子数量可能不止一块，而另外一些坐标上可能已经没有金子。这次变化之后，游戏将不会再对金子的位置和数量进行改变，玩家可以开始进行采集工作。

小 Z 是一名玩家。他打算进行 K 次采集。每次采集可以得到某一个坐标上的所有金子，采集之后，该坐标上的金子数变为 0。

现在小 Z 希望知道，对于变化之后的游戏局面，在采集次数为 K 的前提下，最多可以采集到多少块金子？ 答案可能很大，小 Z 希望得到对 10^9+7 取模之后的答案。

【输入】

输入文件 find.in。

共一行，包含两个正整数 N, K。

【输出】

输出文件 find.out。

一个整数，表示最多可以采集到的金子数量。

【输入输出样例】

find.in	**find.out**
12 5	18

【样例解释】

采集(1,2):从(1,2)(11,12) (1,12) (11, 2)来的四个金子

(1,1):从(1,1)(11,1) (1,11)(11,11)来的四个金子

(2,1):从(2,1)(12,11) (12,1)(2,11)来的四个金子

(2,2):从(2,2)(12,12) (12,2)(2,12)来的四个金子

(2,3):从(2,3)(12,3)来的两个金子

一共八个金子

【数据说明】

对于 30%的数据：N<=10^3

对于 100%的数据：N<=10^12,K<=min(N^2,10^5)