全国信息学奥林匹克联赛（NOIP2011）复赛

提高组 day1

（请选手务必仔细阅读本页内容）

一．题目概况

中文题目名称	铺地毯	选择客栈	mayan 游戏
英文题目与子目录名	carpet	hotel	mayan
可执行文件名	carpet	hotel	mayan
输入文件名	carpet.in	hotel.in	mayan.in
输出文件名	carpet.out	hotel.out	mayan.out
每个测试点时限	1 秒	1 秒	3 秒
测试点数目	10	10	10
每个测试点分值	10	10	10
附加样例文件	有	有	有
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

二．提交源程序文件名

对于 C++语言	carpet.cpp	hotel.cpp	mayan.cpp
对于 C 语言	carpet.c	hotel.c	mayan.c
对于 pascal 语言	carpet.pas	hotel. pas	mayan. pas

三．编译命令（不包含任何优化开关）

对于 C++语言	g++ -o carpet carpet.cpp -lm	g++ -o hotel hotel.cpp -lm	g++ -o mayan mayan.cpp -lm
对于 C 语言	gcc -o carpet carpet.c -lm	gcc -o hotel hotel.c -lm	gcc -o mayan mayan.c -lm
对于 pascal 语言	fpc carpet.pas	fpc hotel.pas	fpc mayan.pas

四．运行内存限制

内存上限	128M	128M	128M

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、全国统一评测时采用的机器配置为：CPU P4 3.0GHz，内存 1G，上述时限以此配置为准。

4、特别提醒：评测在 NOI Linux 下进行。

1．铺地毯

(carpet.cpp/c/pas)

【问题描述】

为了准备一个独特的颁奖典礼，组织者在会场的一片矩形区域（可看做是平面直角坐标系的第一象限）铺上一些矩形地毯。一共有 n 张地毯，编号从 1 到 n。现在将这些地毯按照编号从小到大的顺序平行于坐标轴先后铺设，后铺的地毯覆盖在前面已经铺好的地毯之上。地毯铺设完成后，组织者想知道覆盖地面某个点的最上面的那张地毯的编号。注意：在矩形地毯边界和四个顶点上的点也算被地毯覆盖。

【输入】

输入文件名为 carpet.in。

输入共 n+2 行。

第一行，一个整数 n，表示总共有 n 张地毯。

接下来的 n 行中，第 i+1 行表示编号 i 的地毯的信息，包含四个正整数 a，b，g，k，每两个整数之间用一个空格隔开，分别表示铺设地毯的左下角的坐标（a，b）以及地毯在 x 轴和 y 轴方向的长度。

第 n+2 行包含两个正整数 x 和 y，表示所求的地面的点的坐标（x，y）。

【输出】

输出文件名为 carpet.out。

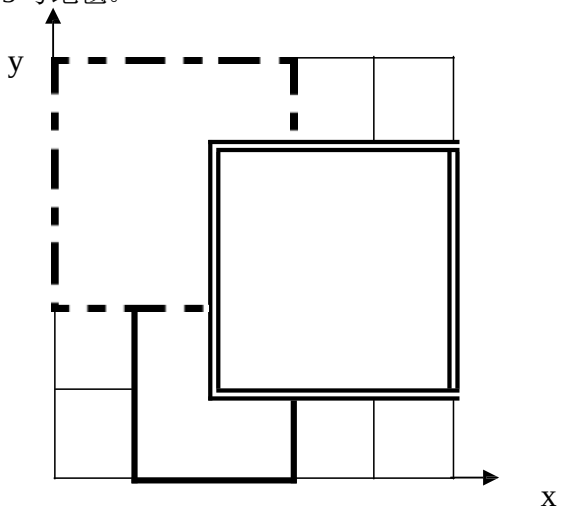
输出共 1 行，一个整数，表示所求的地毯的编号；若此处没有被地毯覆盖则输出-1。

【输入输出样例 1】

carpet.in	carpet.out
3 1 0 2 3 0 2 3 3 2 1 3 3 2 2	3

【输入输出样例说明】

如下图，1 号地毯用实线表示，2 号地毯用虚线表示，3 号用双实线表示，覆盖点（2，2）的最上面一张地毯是 3 号地毯。



【输入输出样例 2】

carpet.in	carpet.out
3 1 0 2 3 0 2 3 3 2 1 3 3 4 5	-1

【输入输出样例说明】

如上图，1 号地毯用实线表示，2 号地毯用虚线表示，3 号用双实线表示，点（4，5）没有被地毯覆盖，所以输出-1。

【数据范围】

对于 30%的数据，有 n≤2；

对于 50%的数据，0≤a, b, g, k≤100；

对于 100%的数据，有 0≤n≤10,000，0≤a, b, g, k≤100,000。

2．选择客栈

(hotel.cpp/c/pas)

【问题描述】

丽江河边有 n 家很有特色的客栈，客栈按照其位置顺序从 1 到 n 编号。每家客栈都按照某一种色调进行装饰（总共 k 种，用整数 0 ~ k-1 表示），且每家客栈都设有一家咖啡店，每家咖啡店均有各自的最低消费。

两位游客一起去丽江旅游，他们喜欢相同的色调，又想尝试两个不同的客栈，因此决定分别住在色调相同的两家客栈中。晚上，他们打算选择一家咖啡店喝咖啡，要求咖啡店位于两人住的两家客栈之间（包括他们住的客栈），且咖啡店的最低消费不超过 p。

他们想知道总共有多少种选择住宿的方案，保证晚上可以找到一家最低消费不超过 p 元的咖啡店小聚。

【输入】

输入文件 hotel.in，共 n+1 行。

第一行三个整数 n，k，p，每两个整数之间用一个空格隔开，分别表示客栈的个数，色调的数目和能接受的最低消费的最高值；

接下来的 n 行，第 i+1 行两个整数，之间用一个空格隔开，分别表示 i 号客栈的装饰色调和 i 号客栈的咖啡店的最低消费。

【输出】

输出文件名为 hotel.out。

输出只有一行，一个整数，表示可选的住宿方案的总数。

【输入输出样例 1】

hotel.in	hotel.out
5 2 3 0 5 1 3 0 2 1 4 1 5	3

【输入输出样例说明】

客栈编号	①	②	③	④	⑤
色调	0	1	0	1	1
最低消费	5	3	2	4	5

2 人要住同样色调的客栈，所有可选的住宿方案包括：住客栈①③，②④，②⑤，④⑤，但是若选择住 4、5 号客栈的话，4、5 号客栈之间的咖啡店的最低消费是 4，而两人能承受的最低消费是 3 元，所以不满足要求。因此只有前 3 种方案可选。

【数据范围】

对于 30%的数据，有 n≤100；

对于 50%的数据，有 n≤1,000；

对于 100%的数据，有 2≤n≤200,000，0<k≤50，0≤p≤100， 0≤最低消费≤100。

3．Mayan 游戏

(mayan.cpp/c/pas)

【问题描述】

Mayan puzzle 是最近流行起来的一个游戏。游戏界面是一个 7 行 5 列的棋盘，上面堆放着一些方块，方块不能悬空堆放，即方块必须放在最下面一行，或者放在其他方块之上。游戏通关是指在规定的步数内消除所有的方块，消除方块的规则如下：

1、 每步移动可以且仅可以沿横向（即向左或向右）拖动某一方块一格：当拖动这一方块时，如果拖动后到达的位置（以下称目标位置）也有方块，那么这两个方块将交换位置（参见输入输出样例说明中的图 6 到图 7）；如果目标位置上没有方块，那么被拖动的方块将从原来的竖列中抽出，并从目标位置上掉落（直到不悬空，参见下面图 1 和图 2）；

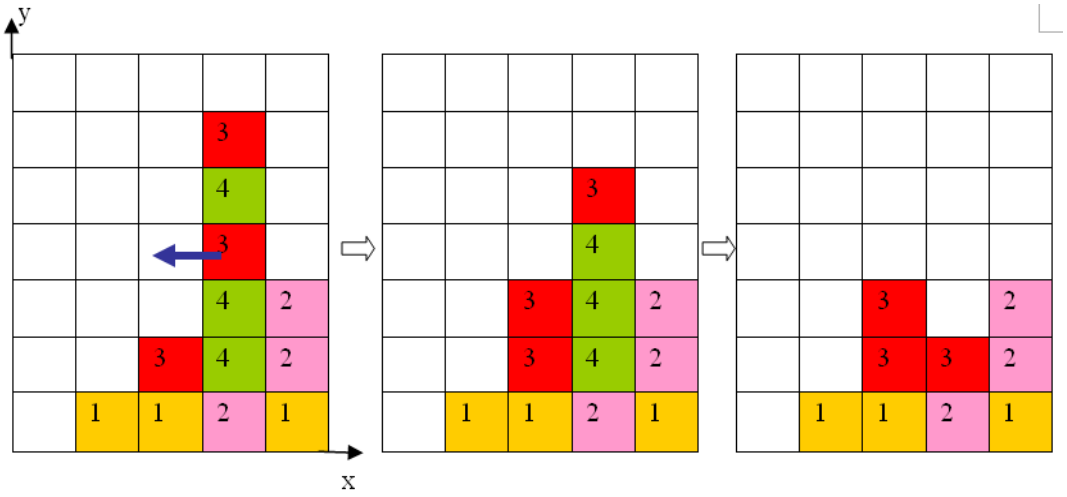


图 1　　图 2　　图 3

2、 任一时刻，如果在一横行或者竖列上有连续三个或者三个以上相同颜色的方块，则它们将立即被消除（参见图 1 到图 3）。

注意：

a) 如果同时有多组方块满足消除条件，几组方块会同时被消除（例如下面图 4，三个颜色为 1 的方块和三个颜色为 2 的方块会同时被消除，最后剩下一个颜色为 2 的方块）。

b) 当出现行和列都满足消除条件且行列共享某个方块时，行和列上满足消除条件的所有方块会被同时消除（例如下面图 5 所示的情形，5 个方块会同时被消除）。

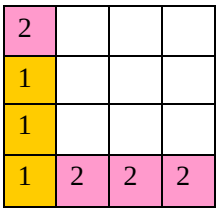


图 4

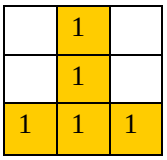


图 5

3、 方块消除之后，消除位置之上的方块将掉落，掉落后可能会引起新的方块消除。注意：掉落的过程中将不会有方块的消除。

上面图 1 到图 3 给出了在棋盘上移动一块方块之后棋盘的变化。棋盘的左下角方块的坐标为（0, 0），将位于（3, 3）的方块向左移动之后，游戏界面从图 1 变成图 2 所示的状态，此时在一竖列上有连续三块颜色为 4 的方块，满足消除条件，消除连续 3 块颜色为 4 的方块后，上方的颜色为 3 的方块掉落，形成图 3 所示的局面。

【输入】

输入文件 mayan.in，共 6 行。

第一行为一个正整数 n，表示要求游戏通关的步数。

接下来的 5 行，描述 7*5 的游戏界面。每行若干个整数，每两个整数之间用一个空格隔开，每行以一个 0 结束，自下向上表示每竖列方块的颜色编号（颜色不多于 10 种，从 1 开始顺序编号，相同数字表示相同颜色）。

输入数据保证初始棋盘中没有可以消除的方块。

【输出】

输出文件名为 mayan.out。

如果有解决方案，输出 n 行，每行包含 3 个整数 x，y，g，表示一次移动，每两个整数之间用一个空格隔开，其中（x，y）表示要移动的方块的坐标，g 表示移动的方向，1 表示向右移动，-1 表示向左移动。注意：多组解时，按照 **x** 为第一关健字，**y** 为第二关健字，**1** 优先于**-1**，给出一组字典序最小的解。游戏界面左下角的坐标为（**0**，**0**）。

如果没有解决方案，输出一行，包含一个整数-1。

【输入输出样例 1】

mayan.in	**mayan.out**
3 1 0 2 1 0 2 3 4 0 3 1 0 2 4 3 4 0	2 1 1 3 1 1 3 0 1

【输入输出样例说明】

按箭头方向的顺序分别为图 6 到图 11

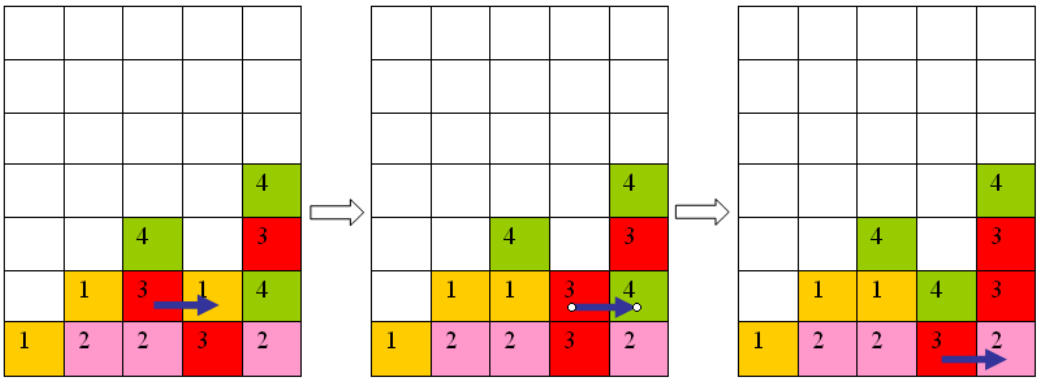


图 6　　图 7　　图 8

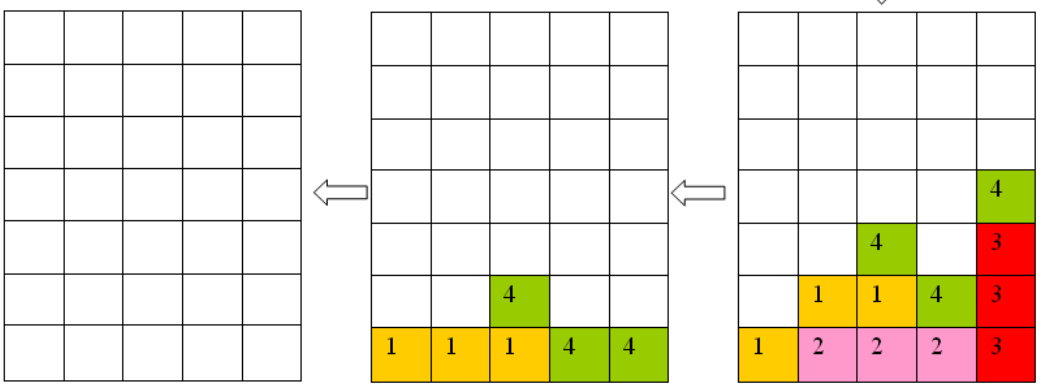


图 11　　图 10　　图 9

样例输入的游戏局面如上面第一个图片所示，依次移动的三步是：（2，1）处的方格向右移动，（3，1）处的方格向右移动，（3，0）处的方格向右移动，最后可以将棋盘上所有方块消除。

【数据范围】

对于 30%的数据，初始棋盘上的方块都在棋盘的最下面一行；

对于 100%的数据，0 < n≤5。

全国信息学奥林匹克联赛（　**NOIP2011**）复赛

提高组 day2

（请选手务必仔细阅读本页内容）

一．题目概况

中文题目名称	计算系数	聪明的质监员	观光公交
英文题目与子目录名	factor	qc	bus
可执行文件名	factor	qc	bus
输入文件名	factor.in	qc.in	bus.in
输出文件名	factor.out	qc.out	bus.out
每个测试点时限	1 秒	1 秒	1 秒
测试点数目	10	20	20
每个测试点分值	10	5	5
附加样例文件	有	有	有
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

二．提交源程序文件名

对于 C++语言	factor.cpp	qc.cpp	bus.cpp
对于 C 语言	factor.c	qc.c	bus.c
对于 pascal 语言	factor.pas	qc. Pas	bus. pas

三．编译命令（不包含任何优化开关）

对于 C++语言	g++ - o factor factor.cpp -lm	g++ - o qc qc.cpp –lm	g++ - o bus bus.cpp -lm
对于 C 语言	gcc - o factor factor.c -lm	gcc - o qc qc.c –lm	gcc - o bus bus.c -lm
对于 pascal 语言	fpc factor.pas	fpc qc.pas	fpc bus.pas

四．运行内存限制

内存上限	128M	128M	128M

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、全国统一评测时采用的机器配置为： CPU P4 3.0GHz，内存 1G，上述时限以此配置为准。

4、特别提醒：评测在 **NOI Linux** 下进行。

1．计算系数

(factor.cpp/c/pas)

【问题描述】

给定一个多项式 $(ax+by)^k$，请求出多项式展开后 $x^n y^m$ 项的系数。

【输入】

输入文件名为 factor.in。

共一行，包含 5 个整数，分别为 a，b，k，n，m，每两个整数之间用一个空格隔开。

【输出】

输出文件名为 factor.out。

输出共 1 行，包含一个整数，表示所求的系数，这个系数可能很大，输出对 **10007** 取模后的结果。

【输入输出样例】

factor.in	factor.out
1 1 3 1 2	3

【数据范围】

对于 30%的数据，有 $0 \le k \le 10$；

对于 50%的数据，有 $a = 1$，$b = 1$；

对于 100%的数据，有 $0 \le k \le 1{,}000$，$0 \le n, m \le k$，且 $n + m = k$，$0 \le a$，$b \le 1{,}000{,}000$。

2．聪明的质监员

(qc.cpp/c/pas)

【问题描述】

小 T 是一名质量监督员，最近负责检验一批矿产的质量。这批矿产共有 n 个矿石，从 1 到 n 逐一编号，每个矿石都有自己的重量 w_i 以及价值 v_i。检验矿产的流程是：

1、给定 m 个区间 $[L_i, R_i]$；

2、选出一个参数 W；

3、对于一个区间 $[L_i, R_i]$，计算矿石在这个区间上的检验值 Y_i：

$$Y_i = \sum_j 1 * \sum_j v_j, \quad j \in [L_i, R_i] \text{ 且 } w_j \ge W, \ j \text{ 是矿石编号}$$

这批矿产的检验结果 **Y** 为各个区间的检验值之和。即：$Y = \sum_{i=1}^{m} Y_i$

若这批矿产的检验结果与所给标准值 S 相差太多，就需要再去检验另一批矿产。小 T 不想费时间去检验另一批矿产，所以他想通过调整参数 W 的值，让检验结果尽可能的靠近标准值 S，即使得 **S-Y** 的绝对值最小。请你帮忙求出这个最小值。

【输入】

输入文件 qc.in。

第一行包含三个整数 n，m，S，分别表示矿石的个数、区间的个数和标准值。

接下来的 n 行，每行 2 个整数，中间用空格隔开，第 i+1 行表示 i 号矿石的重量 w_i 和价值 v_i 。

接下来的 m 行，表示区间，每行 2 个整数，中间用空格隔开，第 i+n+1 行表示区间 $[L_i, R_i]$ 的两个端点 L_i 和 R_i。注意：不同区间可能重合或相互重叠。

【输出】

输出文件名为 qc.out。输出只有一行，包含一个整数，

表示所求的最小值。

【输入输出样例】

qc.in	qc.out
5 3 15 1 5 2 5 3 5 4 5 5 5 1 5 2 4 3 3	10

【输入输出样例说明】

当 W 选 4 的时候，三个区间上检验值分别为 20、5、0，这批矿产的检验结果为 25，此时与标准值 S 相差最小为 10。

【数据范围】

对于 10%的数据，有 $1 \le n, m \le 10$；对于 30%的数据，有 $1 \le n, m \le 500$；对于 50%的数据，有 $1 \le n, m \le 5{,}000$；对于 70%的数据，有 $1 \le n, m \le 10{,}000$；

对于 100%的数据，有 $1 \le n, m \le 200{,}000$，$0 < w_i, v_i \le 10^6$，$0 < S \le 10^{12}$，$1 \le L_i \le R_i \le n$。

3． 观光公交

（ bus.cpp/c/pas ）

【问题描述】

风景迷人的小城 Y 市，拥有 n 个美丽的景点。由于慕名而来的游客越来越多，Y 市特意安排了一辆观光公交车，为游客提供更便捷的交通服务。观光公交车在第 0 分钟出现在 1 号景点，随后依次前往 2、3、4…… n 号景点。从第 i 号景点开到第 i+1 号景点需要 D_i 分钟。任意时刻，公交车只能往前开，或在景点处等待。

设共有 m 个游客，每位游客需要乘车 1 次从一个景点到达另一个景点，第 i 位游客在 T_i 分钟来到景点 A_i，希望乘车前往景点 B_i（$A_i < B_i$）。为了使所有乘客都能顺利到达目的地，

公交车在每站都必须等待需要从该景点出发的所有乘客都上车后才能出发开往下一景点。假设乘客上下车不需要时间。 一个乘客的旅行时间， 等于他到达目的地的时刻减去他来到出发地的时刻。 因为只有一辆观光车， 有时候还要停下来等其他乘客， 乘客们纷纷抱怨旅行时间太长了。 于是聪明的司机 ZZ 给公交车安装了 k 个氮气加速器，每使用一个加速器，可以使其中一个 D_i 减 1。对于同一个 D_i 可以重复使用加速器，但是必须保证使用后 $\mathbf{D_i}$ 大于等于 **0**。

那么 ZZ 该如何安排使用加速器，才能使所有乘客的旅行时间总和最小？

【输入】

输入文件名为 bus.in 。

第 1 行是 3 个整数 n, m, k，每两个整数之间用一个空格隔开。 分别表示景点数、 乘客数和氮气加速器个数。

第 2 行是 n-1 个整数，每两个整数之间用一个空格隔开，第 i 个数表示从第 i 个景点开往第 i+1 个景点所需要的时间，即 D_i。

第 3 行至 m+2 行每行 3 个整数 T_i, A_i, B_i，每两个整数之间用一个空格隔开。 第 i+2 行表示第 i 位乘客来到出发景点的时刻，出发的景点编号和到达的景点编号。

【输出】

输出文件名为 bus.out 。共一行，包含一个整数，表示最小的总旅行时间。

【输入输出样例】

bus.in	bus.out
3 3 2 1 4 0 1 3 1 1 2 5 2 3	10

【输入输出样例说明】

对 D_2 使用 2 个加速器，从 2 号景点到 3 号景点时间变为 2 分钟。

公交车在第 1 分钟从 1 号景点出发， 第 2 分钟到达 2 号景点， 第 5 分钟从 2 号景点出发，第 7 分钟到达 3 号景点。

第 1 个旅客旅行时间 7-0 = 7 分钟。

第 2 个旅客旅行时间 2-1 = 1 分钟。

第 3 个旅客旅行时间 7-5 = 2 分钟。

总时间 7+1+2 = 10 分钟。

【数据范围】

对于 10%的数据， k=0 ；

对于 20%的数据， k=1 ；

对于 40%的数据， $2 \le n \le 50$， $1 \le m \le 1,000$， $0 \le k \le 20$， $0 \le D_i \le 10$， $0 \le T_i \le 500$ ；

对于 60%的数据， $1 \le n \le 100$， $1 \le m \le 1,000$， $0 \le k \le 100$， $0 \le D_i \le 100$， $0 \le T_i \le 10,000$ ；

对于 100%的数据， $1 \le n \le 1,000$， $1 \le m \le 10,000$， $0 \le k \le 100,000$， $0 \le D_i \le 100$， $0 \le T_i \le 100,000$。

CCF 全国信息学奥林匹克联赛（ NOIP2012 ）复赛

提高组 day1

（请选手务必仔细阅读本页内容）

一．题目概况

中文题目名称	Vigen ère 密码	国王游戏	开车旅行
英文题目与子目录名	vigenere	game	drive
可执行文件名	vigenere	game	drive
输入文件名	vigenere.in	game.in	drive.in
输出文件名	vigenere.out	game.out	drive.out
每个测试点时限	1 秒	1 秒	1 秒
测试点数目	10	10	20
每个测试点分值	10	10	5
附加样例文件	有	有	有
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

二．提交源程序文件名

对于 C++语言	vigenere.cpp	game.cpp	drive.cpp
对于 C 语言	vigenere.c	game.c	drive.c
对于 pascal 语言	vigenere.pas	game.pas	drive.pas

三．编译命令（不包含任何优化开关）

对于 C++语言	g++ - o vigenere vigenere.cpp -lm	g++ - o game game.cpp -lm	g++ - o drive drive.cpp -lm
对于 C 语言	gcc- o vigenere vigenere.c -lm	gcc- o game game.c -lm	gcc- o drive drive.c -lm
对于 pascal 语言	fpc vigenere.pas	fpc game.pas	fpc drive.pas

四．运行内存限制

内存上限	128M	128M	128M

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、全国统一评测时采用的机器配置为： CPU Intel Core2 Quad Q8200 2.33GHz， 内存 2G，上述时限以此配置为准。

4、特别提醒：评测在 **NOI Linux** 下进行。

1. Vigen ère 密码

(vigenere.cpp/c/pas)

【问题描述】

16 世纪法国外交家 Blaise de Vigen ère 设计了一种多表密码加密算法——Vigen ère 密码。Vigen ère 密码的加密解密算法简单易用，且破译难度比较高，曾在美国南北战争中为南军所广泛使用。

在密码学中，我们称需要加密的信息为明文，用 M 表示；称加密后的信息为密文，用 C 表示；而密钥是一种参数， 是将明文转换为密文或将密文转换为明文的算法中输入的数据，记为 k。 在 Vigen ère 密码中，密钥 k 是一个字母串， $k=k_1k_2\cdots k_n$。当明文 $M=m_1m_2\cdots m_n$ 时，得到的密文 $C=c_1c_2\cdots c_n$，其中 $c_i=m_i?k_i$，运算 ? 的规则如下表所示：

?	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Vigen ère 加密在操作时需要注意：

1. ? 运算忽略参与运算的字母的大小写，并保持字母在明文 M 中的大小写形式；
2. 当明文 M 的长度大于密钥 k 的长度时，将密钥 k 重复使用。

例如，明文 M=Helloworld ，密钥 k=abc 时，密文 C=Hfnlpyosnd 。

明文	H	e	l	l	o	w	o	r	l	d
密钥	a	b	c	a	b	c	a	b	c	a
密文	H	f	n	l	p	y	o	s	n	d

【输入】

输入文件名为 vigenere.in。

输入共 2 行。

第一行为一个字符串，表示密钥 k，长度不超过 100，其中仅包含大小写字母。第二行为一个字符串，表示经加密后的密文，长度不超过 1000，其中仅包含大小写字母。

【输出】

输出文件名为 vigenere.out。

输出共 1 行，一个字符串，表示输入密钥和密文所对应的明文。

【输入输出样例】

vigenere.in	vigenere.out
CompleteVictory Yvqgpxaimmklongnzfwpvxmniytm	Wherethereisawillthereisaway

【数据说明】

对于 100%的数据，输入的密钥的长度不超过 100，输入的密文的长度不超过 1000，且都仅包含英文字母。

2．国王游戏

(game.cpp/c/pas)

【问题描述】

恰逢 H 国国庆，国王邀请 n 位大臣来玩一个有奖游戏。首先，他让每个大臣在左、右手上面分别写下一个整数，国王自己也在左、右手上各写一个整数。然后，让这 n 位大臣排成一排，国王站在队伍的最前面。排好队后，所有的大臣都会获得国王奖赏的若干金币，每位大臣获得的金币数分别是：排在该大臣前面的所有人的左手上的数的乘积除以他自己右手上的数，然后向下取整得到的结果。

国王不希望某一个大臣获得特别多的奖赏，所以他想请你帮他重新安排一下队伍的顺序，使得获得奖赏最多的大臣，所获奖赏尽可能的少。注意，国王的位置始终在队伍的最前面。

【输入】

输入文件为 game.in。

第一行包含一个整数 n，表示大臣的人数。

第二行包含两个整数 a 和 b，之间用一个空格隔开，分别表示国王左手和右手上的整数。

接下来 n 行，每行包含两个整数 a 和 b，之间用一个空格隔开，分别表示每个大臣左手和右手上的整数。

【输出】

输出文件名为 game.out。

输出只有一行，包含一个整数，表示重新排列后的队伍中获奖赏最多的大臣所获得的金币数。

【输入输出样例】

game.in	game.out
3 1 1 2 3 7 4 4 6	2

【输入输出样例说明】

按 1、2、3 号大臣这样排列队伍，获得奖赏最多的大臣所获得金币数为 2；
按 1、3、2 这样排列队伍，获得奖赏最多的大臣所获得金币数为 2；
按 2、1、3 这样排列队伍，获得奖赏最多的大臣所获得金币数为 2；
按 2、3、1 这样排列队伍，获得奖赏最多的大臣所获得金币数为 9；
按 3、1、2 这样排列队伍，获得奖赏最多的大臣所获得金币数为 2；
按 3、2、1 这样排列队伍，获得奖赏最多的大臣所获得金币数为 9。
因此，奖赏最多的大臣最少获得 2 个金币，答案输出 2。

【数据范围】

对于 20%的数据，有 1≤n≤10， 0 < a、 b < 8；
对于 40%的数据，有 1≤n≤20， 0 < a、 b < 8；
对于 60%的数据，有 1≤n≤100；
对于 60%的数据，保证答案不超过 10^9；
对于 100%的数据，有 1 ≤n ≤1,000， 0 < a、 b < 10000。

3．开车旅行

(drive.cpp/c/pas)

【问题描述】

小 A 和小 B 决定利用假期外出旅行，他们将想去的城市从 1 到 N 编号， 且编号较小的城市在编号较大的城市的西边 ，已知各个城市的海拔高度互不相同， 记城市 i 的海拔高度为 H_i，城市 i 和城市 j 之间的距离 d[i,j] 恰好是这两个城市海拔高度之差的绝对值，即 d[i, j] = |??? ??|。

旅行过程中，小 A 和小 B 轮流开车，第一天小 A 开车，之后每天轮换一次。他们计划选择一个城市 S 作为起点， 一直向东行驶 ，并且最多行驶 X 公里就结束旅行。小 A 和小 B 的驾驶风格不同，小 B 总是沿着前进方向选择 一个最近的城市作为目的地，而小 A 总是 沿着前进方向选择 第二近的城市作为目的地（注意： 本题中如果当前城市到两个城市的距离相同，则认为离海拔低的那个城市更近 ）。如果其中任何一人无法按照自己的原则选择目的城市，或者到达目的地会使行驶的总距离超出 X 公里，他们就会结束旅行。

在启程之前，小 A 想知道两个问题：

1. 对于一个给定的 $X=X_0$，从哪一个城市出发，小 A 开车行驶的路程总数与小 B 行驶的路程总数的比值最小 （如果小 B 的行驶路程为 0，此时的比值可视为无穷大， 且两个无穷

大视为相等）。如果从多个城市出发， 小 A 开车行驶的路程总数与小 B 行驶的路程总数的比值都最小，则输出海拔最高的那个城市。

2. 对任意给定的 $X=X_i$ 和出发城市 S_i，小 A 开车行驶的路程总数以及小 B 行驶的路程总数。

【输入】

输入文件为 drive.in。

第一行包含一个整数 N，表示城市的数目。

第二行有 N 个整数，每两个整数之间用一个空格隔开，依次表示城市 1 到城市 N 的海拔高度，即 H_1，H_2，……，H_n，且每个 H_i 都是不同的。

第三行包含一个整数 X_0。

第四行为一个整数 M，表示给定 M 组 S_i 和 X_i。

接下来的 M 行，每行包含 2 个整数 S_i 和 X_i，表示从城市 S_i 出发，最多行驶 X_i 公里。

【输出】

输出文件为 drive.out。

输出共 M+1 行。

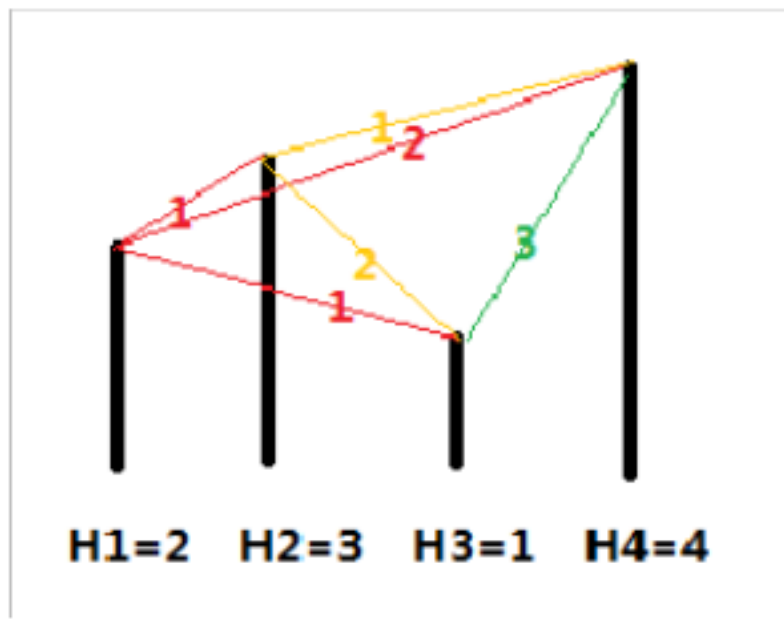
第一行包含一个整数 S_0，表示对于给定的 X_0，从编号为 S_0 的城市出发， 小 A 开车行驶的路程总数与小 B 行驶的路程总数的比值最小。

接下来的 M 行，每行包含 2 个整数，之间用一个空格隔开，依次表示在给定的 S_i 和 X_i 下小 A 行驶的里程总数和小 B 行驶的里程总数。

【输入输出样例 1】

drive.in	drive.out
4 2 3 1 4 3 4 1 3 2 3 3 3 4 3	1 1 1 2 0 0 0 0 0

【输入输出样例 1 说明】



各个城市的海拔高度以及两个城市间的距离如上图所示。

如果从城市 1 出发，可以到达的城市为 2,3,4，这几个城市与城市 1 的距离分别为 1,1,2，但是由于城市 3 的海拔高度低于城市 2，所以我们认为城市 3 离城市 1 最近，城市 2 离城市 1 第二近，所以小 A 会走到城市 2。到达城市 2 后，前面可以到达的城市为 3,4，这两个城市与城市 2 的距离分别为 2,1，所以城市 4 离城市 2 最近，因此小 B 会走到城市 4。到达城市 4 后，前面已没有可到达的城市，所以旅行结束。

如果从城市 2 出发，可以到达的城市为 3,4，这两个城市与城市 2 的距离分别为 2,1，由于城市 3 离城市 2 第二近，所以小 A 会走到城市 3。到达城市 3 后，前面尚未旅行的城市为 4，所以城市 4 离城市 3 最近，但是如果要到达城市 4，则总路程为 2+3=5>3，所以小 B 会直接在城市 3 结束旅行。

如果从城市 3 出发，可以到达的城市为 4，由于没有离城市 3 第二近的城市，因此旅行还未开始就结束了。

如果从城市 4 出发，没有可以到达的城市，因此旅行还未开始就结束了。

【输入输出样例 2】

drive.in	drive.out
10 4 5 6 1 2 3 7 8 9 10 7 10 1 7 2 7 3 7 4 7 5 7 6 7 7 7 8 7 9 7 10 7	2 3 2 2 4 2 1 2 4 5 1 5 1 2 1 2 0 0 0 0 0

【输入输出样例 2 说明】

当 X=7 时，

如果从城市 1 出发，则路线为 1 -> 2 -> 3 -> 8 -> 9，小 A 走的距离为 1+2=3，小 B 走的距离为 1+1=2。（在城市 1 时，距离小 A 最近的城市是 2 和 6，但是城市 2 的海拔更高，视为与城市 1 第二近的城市，所以小 A 最终选择城市 2；走到 9 后，小 A 只有城市 10 可以走，没有第 2 选择可以选，所以没法做出选择，结束旅行）

如果从城市 2 出发，则路线为 2 -> 6 -> 7，小 A 和小 B 走的距离分别为 2，4。

如果从城市 3 出发，则路线为 3 -> 8 -> 9，小 A 和小 B 走的距离分别为 2，1。

如果从城市 4 出发，则路线为 4 -> 6 -> 7，小 A 和小 B 走的距离分别为 2，4。

如果从城市 5 出发，则路线为 5 -> 7 -> 8，小 A 和小 B 走的距离分别为 5，1。

如果从城市 6 出发，则路线为 6 -> 8 -> 9，小 A 和小 B 走的距离分别为 5，1。

如果从城市 7 出发，则路线为 7 -> 9 -> 10，小 A 和小 B 走的距离分别为 2，1。

如果从城市 8 出发，则路线为 8 -> 10，小 A 和小 B 走的距离分别为 2，0。

如果从城市 9 出发，则路线为 9，小 A 和小 B 走的距离分别为 0，0（旅行一开始就结束了）。

如果从城市 10 出发，则路线为 10，小 A 和小 B 走的距离分别为 0，0。

从城市 2 或者城市 4 出发小 A 行驶的路程总数与小 B 行驶的路程总数的比值都最小，但是城市 2 的海拔更高，所以输出第一行为 2。

【数据范围】

对于 30%的数据，有 $1 \leq N \leq 20$，$1 \leq M \leq 20$；

对于 40%的数据，有 $1 \leq N \leq 100$，$1 \leq M \leq 100$；

对于 50%的数据，有 $1 \leq N \leq 100$，$1 \leq M \leq 1,000$；

对于 70%的数据，有 $1 \leq N \leq 1,000$，$1 \leq M \leq 10,000$；

对于 100%的数据，有 $1 \leq N \leq 100,000$, $1 \leq M \leq 10,000$, $-1,000,000,000 \leq H_i \leq 1,000,000,000$, $0 \leq X_0 \leq 1,000,000,000$，$1 \leq S_i \leq N$，$0 \leq X_i \leq 1,000,000,000$，数据保证 H_i 互不相同。

CCF 全国信息学奥林匹克联赛（ NOIP2012 ）复赛

提高组 day2

（请选手务必仔细阅读本页内容）

一．题目概况

中文题目名称	同余方程	借教室	疫情控制
英文题目与子目录名	mod	classroom	blockade
可执行文件名	mod	classroom	blockade
输入文件名	mod.in	classroom.in	blockade.in
输出文件名	mod.out	classroom.out	blockade.out
每个测试点时限	1 秒	1 秒	2 秒
测试点数目	10	20	10
每个测试点分值	10	5	10
附加样例文件	有	有	有
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

二．提交源程序文件名

对于 C++语言	mod.cpp	classroom.cpp	blockade.cpp
对于 C 语言	mod.c	classroom.c	blockade.c
对于 pascal 语言	mod.pas	classroom.pas	blockade.pas

三．编译命令（不包含任何优化开关）

对于 C++语言	g++ -o mod mod.cpp -lm	g++ -o classroom classroom.cpp -lm	g++ -o blockade blockade.cpp -lm
对于 C 语言	gcc -o mod mod.c -lm	gcc -o classroom classroom.c -lm	gcc -o blockade blockade.c -lm
对于 pascal 语言	fpc mod.pas	fpc classroom.pas	fpc blockade.pas

四．运行内存限制

内存上限	128M	128M	128M

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、全国统一评测时采用的机器配置为： CPU Intel Core2 Quad Q8200 2.33GHz ，内存 2G，上述时限以此配置为准。

4、特别提醒：评测在 **NOI Linux** 下进行。

1．同余方程

(mod.cpp/c/pas)

【问题描述】

求关于 x 的同余方程 $ax \equiv 1 \pmod{b}$的最小正整数解。

【输入】

输入文件为 mod.in。

输入只有一行，包含两个正整数 a, b，用一个空格隔开。

【输出】

输出文件为 mod.out。

输出只有一行，包含一个正整数 x_0，即最小正整数解。输入数据保证一定有解。

【输入输出样例】

mod.in	mod.out
3 10	7

【数据范围】

对于 40%的数据， $2 \leq b \leq 1,000$；

对于 60%的数据， $2 \leq b \leq 50,000,000$；

对于 100%的数据， $2 \leq a, b \leq 2,000,000,000$。

2．借教室

(classroom.cpp/c/pas)

【问题描述】

在大学期间，经常需要租借教室。大到院系举办活动，小到学习小组自习讨论，都需要向学校申请借教室。教室的大小功能不同，借教室人的身份不同，借教室的手续也不一样。

面对海量租借教室的信息，我们自然希望编程解决这个问题。

我们需要处理接下来 n天的借教室信息，其中第 i天学校有 r_i个教室可供租借。共有 m份订单，每份订单用三个正整数描述，分别为 d_j, s_j, t_j，表示某租借者需要从第 s_j天到第 t_j天租借教室（包括第 s_j天和第 t_j天），每天需要租借 d_j个教室。

我们假定，租借者对教室的大小、地点没有要求。即对于每份订单，我们只需要每天提供d_j个教室，而它们具体是哪些教室，每天是否是相同的教室则不用考虑。

借教室的原则是先到先得，也就是说我们要按照订单的先后顺序依次为每份订单分配教室。如果在分配的过程中遇到一份订单无法完全满足，则需要停止教室的分配，通知当前申请人修改订单。这里的无法满足指从第 s_j天到第 t_j天中有至少一天剩余的教室数量不足 d_j个。

现在我们需要知道，是否会有订单无法完全满足。如果有，需要通知哪一个申请人修改订单。

【输入】

输入文件为 classroom.in。

第一行包含两个正整数 n, m，表示天数和订单的数量。

第二行包含 n 个正整数，其中第 i 个数为 r_i，表示第 i 天可用于租借的教室数量。

接下来有 m 行，每行包含三个正整数 d_j, s_j, t_j，表示租借的数量，租借开始、结束分别在第几天。

每行相邻的两个数之间均用一个空格隔开。天数与订单均用从 1开始的整数编号。

【输出】

输出文件为 classroom.out。

如果所有订单均可满足，则输出只有一行，包含一个整数 0。否则（订单无法完全满足）输出两行，第一行输出一个负整数 -1，第二行输出需要修改订单的申请人编号。

【输入输出样例】

classroom.in	classroom.out
4 3 2 5 4 3 2 1 3 3 2 4 4 2 4	-1 2

【输入输出样例说明】

第 1 份订单满足后，4 天剩余的教室数分别为 0，3，2，3。第 2 份订单要求第 2 天到第 4 天每天提供 3 个教室，而第 3 天剩余的教室数为 2，因此无法满足。分配停止，通知第 2 个申请人修改订单。

【数据范围】

对于 10%的数据，有 $1 \le n, m \le 10$；

对于 30%的数据，有 $1 \le n, m \le 1000$；

对于 70%的数据，有 $1 \le n, m \le 10^5$；

对于 100%的数据，有 $1 \le n, m \le 10^6, 0 \le r_i, d_j \le 10^9, 1 \le s_j \le t_j \le n$。

3．疫情控制

（blockade.cpp/c/pas ）

【问题描述】

H 国有 n 个城市，这 n 个城市用 n-1 条双向道路相互连通构成一棵树，1 号城市是首都，也是树中的根节点。

H 国的首都爆发了一种危害性极高的传染病。当局为了控制疫情，不让疫情扩散到边境城市（叶子节点所表示的城市），决定动用军队在一些城市建立检查点，使得从首都到边境城市的每一条路径上都至少有一个检查点，边境城市也可以建立检查点。但特别要注意的是，首都是不能建立检查点的。

现在，在 H 国的一些城市中已经驻扎有军队，且一个城市可以驻扎多个军队。一支军

队可以在有道路连接的城市间移动，并在除首都以外的任意一个城市建立检查点，且只能在一个城市建立检查点。一支军队经过一条道路从一个城市移动到另一个城市所需要的时间等于道路的长度（单位：小时）。

请问最少需要多少个小时才能控制疫情。注意：不同的军队可以同时移动。

【输入】

输入文件名为 blockade.in。

第一行一个整数 n，表示城市个数。

接下来的 n-1 行，每行 3 个整数，u、v、w，每两个整数之间用一个空格隔开，表示从城市 u 到城市 v 有一条长为 w 的道路。数据保证输入的是一棵树，且根节点编号为 1。

接下来一行 m，表示军队个数。

接下来一行 m 个整数，每两个整数之间用一个空格隔开，分别表示这 m 个军队所驻扎的城市的编号。

【输出】

输出文件为 blockade.out。

共一行，包含一个整数，表示控制疫情所需要的最少时间。如果无法控制疫情则输出 -1。

【输入输出样例】

blockade.in	blockade.out
4 1 2 1 1 3 2 3 4 3 2 2 2	3

【输入输出样例说明】

第一支军队在 2 号点设立检查点，第二支军队从 2 号点移动到 3 号点设立检查点，所需时间为 3 个小时。

【数据范围】

保证军队不会驻扎在首都。

对于 20%的数据，$2 \le n \le 10$；

对于 40%的数据，$2 \le n \le 50$，$0 < w < 10^5$；

对于 60%的数据，$2 \le n \le 1000$，$0 < w < 10^6$；

对于 80%的数据，$2 \le n \le 10,000$；

对于 100%的数据，$2 \le m \le n \le 50,000$，$0 < w < 10^9$。

CCF 全国信息学奥林匹克联赛（ NOIP2013 ）复赛

提高组 day1

（请选手务必仔细阅读本页内容）

一．题目概况

中文题目名称	转圈游戏	火柴排队	货车运输
英文题目与子目录名	circle	match	truck
可执行文件名	circle	match	truck
输入文件名	circle.in	match.in	truck.in
输出文件名	circle.out	match.out	truck.out
每个测试点时限	1 秒	1 秒	1 秒
测试点数目	10	10	20
每个测试点分值	10	10	5
附加样例文件	有	有	有
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
运行内存上限	128M	128M	128M

二．提交源程序文件名

对于 C++ 语言	circle.cpp	match.cpp	truck.cpp
对于 C 语言	circle.c	match.c	truck.c
对于 pascal 语言	circle.pas	match.pas	truck.pas

三．编译命令（不包含任何优化开关）

对于 C++ 语言	g++ -o circle circle.cpp -lm	g++ -o match match.cpp -lm	g++ -o truck truck.cpp -lm
对于 C 语言	gcc- o circle circle.c -lm	gcc- o match match.c –lm	gcc- o truck truck.c -lm
对于 pascal 语言	fpc circle.pas	fpc match.pas	fpc truck.pas

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、全国统一评测时采用的机器配置为： CPU AMD Athlon(tm) 64x2 Dual Core CPU 5200+ ，2.71GHz，内存 2G，上述时限以此配置为准。

4、只提供 Linux 格式附加样例文件。

5、特别提醒：评测在 **NOI Linux** 下进行。

1．转圈游戏

(circle.cpp/c/pas)

【问题描述】

n 个小伙伴（编号从 0 到 n-1）围坐一圈玩游戏。按照顺时针方向给 n 个位置编号，从 0 到 n-1。最初，第 0 号小伙伴在第 0 号位置，第 1 号小伙伴在第 1 号位置，……，依此类推。

游戏规则如下：每一轮第 0 号位置上的小伙伴顺时针走到第 m 号位置，第 1 号位置小伙伴走到第 m+1 号位置，……，依此类推，第 n - m号位置上的小伙伴走到第 0 号位置，第 n-m+1 号位置上的小伙伴走到第 1 号位置，……，第 n-1 号位置上的小伙伴顺时针走到第 m-1 号位置。

现在，一共进行了 10^k 轮，请问 x 号小伙伴最后走到了第几号位置。

【输入】

输入文件名为 circle.in 。

输入共 1 行，包含 4 个整数 n、m、k、x，每两个整数之间用一个空格隔开。

【输出】

输出文件名为 circle.out 。

输出共 1 行，包含 1 个整数，表示 10^k 轮后 x 号小伙伴所在的位置编号。

【输入输出样例】

circle.in	circle.out
10 3 4 5	5

【数据说明】

对于 30%的数据， 0 < ??< 7 ；

对于 80%的数据， 0 < ??< 10^7 ；

对于 100%的数据， 1 < ??< 1,000,000 ， 0 < ??< ?? 1 ≤x ≤n， 0 < ??< 10^9 。

2．火柴排队

(match.cpp/c/pas)

【问题描述】

涵涵有两盒火柴， 每盒装有 n 根火柴， 每根火柴都有一个高度。 现在将每盒中的火柴各自排成一列， 同一列火柴的高度互不相同， 两列火柴之间的距离定义为： $\sum_{i=1}^{n}(??_{??}-??_{??})^2$ ，其中 a_i 表示第一列火柴中第 i 个火柴的高度， b_i 表示第二列火柴中第 i 个火柴的高度。

每列火柴中相邻两根火柴的位置都可以交换， 请你通过交换使得两列火柴之间的距离最小。请问得到这个最小的距离，最少需要交换多少次？ 如果这个数字太大，请输出这个最小交换次数对 **99,999,997** 取模的结果。

【输入】

输入文件为 match.in。

共三行，第一行包含一个整数 n，表示每盒中火柴的数目。

第二行有 n 个整数，每两个整数之间用一个空格隔开，表示第一列火柴的高度。

第三行有 n 个整数，每两个整数之间用一个空格隔开，表示第二列火柴的高度。

【输出】

输出文件为 match.out。

输出共一行，包含一个整数，表示最少交换次数对 **99,999,997** 取模的结果 。

【输入输出样例 1】

match.in	match.out
4 2 3 1 4 3 2 1 4	1

【输入输出样例说明】

最小距离是 0，最少需要交换 1 次，比如：交换第 1 列的前 2 根火柴或者交换第 2 列的前 2 根火柴。

【输入输出样例 2】

match.in	match.out
4 1 3 4 2 1 7 2 4	2

【输入输出样例说明】

最小距离是 10，最少需要交换 2 次，比如：交换第 1 列的中间 2 根火柴的位置，再交换第 2 列中后 2 根火柴的位置。

【数据范围】

对于 10%的数据， $1 \leq n \leq 10$ ；

对于 30%的数据， $1 \leq n \leq 100$ ；

对于 60%的数据， $1 \leq n \leq 1{,}000$ ；

对于 100%的数据， $1 \leq n \leq 100{,}000$ ， $0 \leq$ 火柴高度 $\leq 2^{31} - 1$。

3．货车运输

(truck.cpp/c/pas)

【问题描述】

A 国有 n 座城市， 编号从 1 到 n，城市之间有 m 条双向道路。 每一条道路对车辆都有重量限制， 简称限重。 现在有 q 辆货车在运输货物， 司机们想知道每辆车在不超过车辆限重的情况下，最多能运多重的货物。

【输入】

输入文件名为 truck.in。

输入文件第一行有两个用一个空格隔开的整数 n，m，表示 A 国有 n 座城市和 m 条道路。

接下来 m 行每行 3 个整数 x、y、z，每两个整数之间用一个空格隔开， 表示从 x 号城市到 y 号城市有一条限重为 z 的道路。注意： x 不等于 y，两座城市之间可能有多条道路。

接下来一行有一个整数 q，表示有 q 辆货车需要运货。

接下来 q 行，每行两个整数 x、y，之间用一个空格隔开，表示一辆货车需要从 x 城市运输货物到 y 城市，注意： x 不等于 y。

【输出】

输出文件名为 truck.out。

输出共有 q 行，每行一个整数，表示对于每一辆货车，它的最大载重是多少。如果货车不能到达目的地，输出 -1。

【输入输出样例】

truck.in	truck.out
4 3 1 2 4 2 3 3 3 1 1 3 1 3 1 4 1 3	3 -1 3

【数据说明】

对于 30%的数据， 0 < ??< 1,000 ， 0 < ??< 10,000 ， 0 < ??< 1,000 ；

对于 60%的数据， 0 < ??< 1,000 ， 0 < ??< 50,000 ， 0 < ??< 1,000 ；

对于 100%的数据， 0 < ??< 10,000 ， 0 < ??< 50,000 ， 0 < ??< 30,000 ，0 ≤z ≤100,000 。

CCF 全国信息学奥林匹克联赛（ NOIP2013 ）复赛

提高组 day2

（请选手务必仔细阅读本页内容）

一. 题目概况

中文题目名称	积木大赛	花匠	华容道
英文题目与子目录名	block	flower	puzzle
可执行文件名	block	flower	puzzle
输入文件名	block.in	flower.in	puzzle.in
输出文件名	block.out	flower.out	puzzle.out
每个测试点时限	1 秒	1 秒	1 秒
测试点数目	10	10	20
每个测试点分值	10	10	5
附加样例文件	有	有	有
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
运行内存上限	128M	128M	128M

二. 提交源程序文件名

对于 C++语言	block.cpp	flower.cpp	puzzle.cpp
对于 C 语言	block.c	flower.c	puzzle.c
对于 pascal 语言	block.pas	flower.pas	puzzle.pas

三. 编译命令（不包含任何优化开关）

对于 C++语言	g++ -o block block.cpp -lm	g++ -o flower flower.cpp -lm	g++ -o puzzle puzzle.cpp -lm
对于 C 语言	gcc -o block block.c -lm	gcc -o flower flower.c -lm	gcc- o puzzle puzzle.c -lm
对于 pascal 语言	fpc block.pas	fpc flower.pas	fpc puzzle.pas

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、全国统一评测时采用的机器配置为： CPU AMD Athlon(tm) 64x2 Dual Core CPU 5200+ ，2.71GHz，内存 2G，上述时限以此配置为准。

4、只提供 Linux 格式附加样例文件。

5、特别提醒：评测在 **NOI Linux** 下进行。

1. 积木大赛

(block.cpp/c/pas)

【题目描述】

春春幼儿园举办了一年一度的“积木大赛”。今年比赛的内容是搭建一座宽度为 ?? 的大厦，大厦可以看成由 ?? 块宽度为 1 的积木组成，第 ?? 块积木的最终高度需要是 ??。

在搭建开始之前，没有任何积木（可以看成 ?? 块高度为 0 的积木）。接下来每次操作，小朋友们可以选择一段连续区间 [??,??]，然后将第 ?? 块到第 ?? 块之间（含第 L 块和第 R 块）所有积木的高度分别增加 1。

小 ??是个聪明的小朋友，她很快想出了建造大厦的最佳策略，使得建造所需的操作次数最少。但她不是一个勤于动手的孩子，所以想请你帮忙实现这个策略，并求出最少的操作次数。

【输入】

输入文件为 block.in

输入包含两行，第一行包含一个整数 ??，表示大厦的宽度。

第二行包含 ?? 个整数，第 ?? 个整数为 ??。

【输出】

输出文件为 block.out

仅一行，即建造所需的最少操作数。

【输入输出样例】

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

【样例解释】

其中一种可行的最佳方案，依次选择

[1,5] [1,3] [2,3] [3,3] [5,5]

【数据范围】

对于 30%的数据，有 1 ≤??≤10 ；

对于 70%的数据，有 1 ≤??≤1000 ；

对于 100%的数据，有 1 ≤??≤100000 ， 0 ≤??≤10000 。

2．花匠

(flower.cpp/c/pas)

【问题描述】

花匠栋栋种了一排花，每株花都有自己的高度。花儿越长越大，也越来越挤。栋栋决定把这排中的一部分花移走，将剩下的留在原地，使得剩下的花能有空间长大，同时，栋栋希望剩下的花排列得比较别致。

具体而言，栋栋的花的高度可以看成一列整数 $?_1,?_2,\cdots,?_{??}$。设当一部分花被移走后，剩下的花的高度依次为 $??_1,??_2,\cdots,??_{??}$，则栋栋希望下面两个条件中至少有一个满足：

条件 A：对于所有的 $1 \leq ?? \leq \frac{??}{2}$，有 $??_{2??} > ??_{2??-1}$，同时对于所有的 $1 \leq ?? < \frac{??}{2}$，有 $??_{2??} > ??_{2??+1}$；

条件 B：对于所有的 $1 \leq ?? \leq \frac{??}{2}$，有 $??_{2??} < ??_{2??-1}$，同时对于所有的 $1 \leq ?? < \frac{??}{2}$，有 $??_{2??} < ??_{2??+1}$。

注意上面两个条件在 $?? = 1$ 时同时满足，当 $?? > 1$ 时最多有一个能满足。

请问，栋栋最多能将多少株花留在原地。

【输入】

输入文件为 flower.in。

输入的第一行包含一个整数 $??$，表示开始时花的株数。

第二行包含 $??$ 个整数，依次为 $?_1,?_2,\cdots,?_{??}$，表示每株花的高度。

【输出】

输出文件为 flower.out。

输出一行，包含一个整数 $??$，表示最多能留在原地的花的株数。

【输入输出样例】

flower.in	flower.out
5 5 3 2 1 2	3

【输入输出样例说明】

有多种方法可以正好保留 3 株花，例如，留下第 1、4、5 株，高度分别为 5、1、2，满足条件 B。

【数据范围】

对于 20%的数据，$?? \leq 10$；

对于 30%的数据，$?? \leq 25$；

对于 70%的数据，$?? \leq 1000$，$0 \leq ?_{??} \leq 1000$；

对于 100%的数据，$1 \leq ?? \leq 100{,}000$，$0 \leq ?_{??} \leq 1{,}000{,}000$，所有的 $?_{??}$ 随机生成，所有随机数服从某区间内的均匀分布。

3. 华容道

(puzzle.cpp/c/pas)

【问题描述】

小 B 最近迷上了华容道，可是他总是要花很长的时间才能完成一次。于是，他想到用编程来完成华容道： 给定一种局面， 华容道是否根本就无法完成， 如果能完成，最少需要多少时间。

小 B 玩的华容道与经典的华容道游戏略有不同，游戏规则是这样的：

1. 在一个 n*m 棋盘上有 n*m 个格子，其中有且只有一个格子是空白的，其余 n*m-1 个格子上每个格子上有一个棋子，每个棋子的大小都是 1*1 的；
2. 有些棋子是固定的，有些棋子则是可以移动的；
3. 任何与空白的格子相邻（有公共的边）的格子上的棋子都可以移动到空白格子上。

游戏的目的是把某个指定位置可以活动的棋子移动到目标位置。

给定一个棋盘， 游戏可以玩 q 次，当然， 每次棋盘上固定的格子是不会变的，但是棋盘上空白的格子的初始位置、指定的可移动的棋子的初始位置和目标位置却可能不同。第 i 次玩的时候， 空白的格子在第 EX_i 行第 EY_i 列，指定的可移动棋子的初始位置为第 SX_i 行第 SY_i 列，目标位置为第 TX_i 行第 TY_i 列。

假设小 B 每秒钟能进行一次移动棋子的操作，而其他操作的时间都可以忽略不计。请你告诉小 B 每一次游戏所需要的最少时间，或者告诉他不可能完成游戏。

【输入】

输入文件为 puzzle.in。

第一行有 3 个整数，每两个整数之间用一个空格隔开，依次表示 n、m 和 q；

接下来的 n 行描述一个 n*m 的棋盘，每行有 m 个整数，每两个整数之间用一个空格隔开，每个整数描述棋盘上一个格子的状态， 0 表示该格子上的棋子是固定的， 1 表示该格子上的棋子可以移动或者该格子是空白的。

接下来的 q 行，每行包含 6 个整数依次是 EX_i、EY_i、SX_i、SY_i、TX_i、TY_i，每两个整数之间用一个空格隔开，表示每次游戏空白格子的位置，指定棋子的初始位置和目标位置。

【输出】

输出文件名为 puzzle.out。

输出有 q 行， 每行包含 1 个整数， 表示每次游戏所需要的最少时间， 如果某次游戏无法完成目标则输出 -1 。

【输入输出样例】

puzzle.in	puzzle.out
3 4 2 0 1 1 1 0 1 1 0 0 1 0 0 3 2 1 2 2 2 1 2 2 2 3 2	2 -1

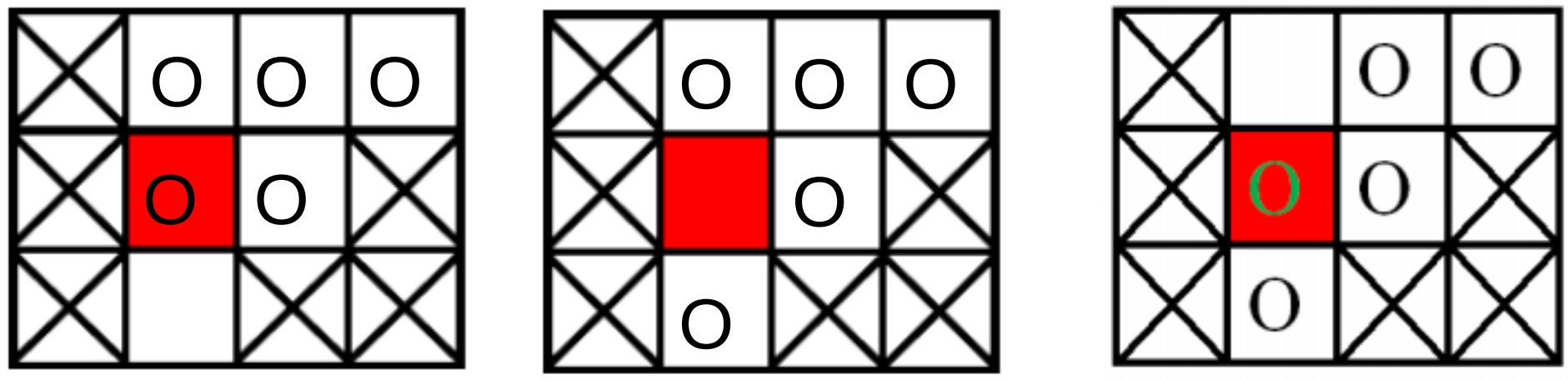
【输入输出样例说明】

棋盘上划叉的格子是固定的, 红色格子是目标位置, 圆圈表示棋子, 其中绿色圆圈表示目标棋子。

1. 第一次游戏，空白格子的初始位置是 (3, 2)（图中空白所示），游戏的目标是将初始位置在 (1, 2) 上的棋子（图中绿色圆圈所代表的棋子）移动到目标位置 (2, 2) （图中红色的格子）上。

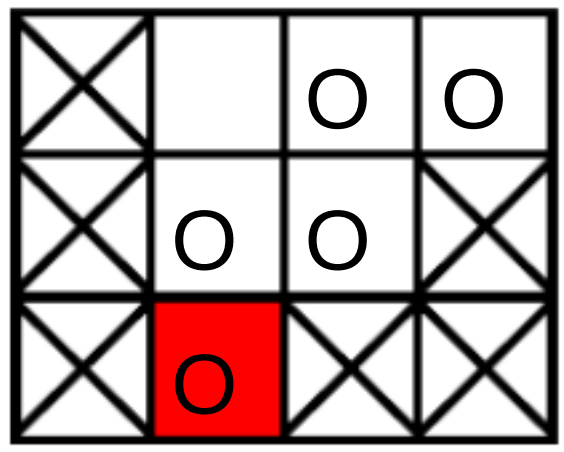
移动过程如下：

初始状态 第一步之后 第二步之后



2. 第二次游戏，空白格子的初始位置是（ 1, 2）（图中空白所示），游戏的目标是将初始位置在（ 2, 2）上的棋子（图中绿色圆圈所示）移动到目标位置 (3, 2) 上。

初始状态



要将指定块移入目标位置，必须先将空白块移入目标位置，空白块要移动到目标位置，必然是从位置（ 2，2）上与当前图中目标位置上的棋子交换位置，之后能与空白块交换位置的只有当前图中目标位置上的那个棋子, 因此目标棋子永远无法走到它的目标位置, 游戏无法完成。

【数据范围】

对于 30%的数据, $1 \le n, m \le 10$, $q = 1$；

对于 60%的数据, $1 \le n, m \le 30$, $q \le 10$；

对于 100%的数据, $1 \le n, m \le 30$, $q \le 500$。

CCF全国信息学奥林匹克联赛（ NOIP2014）复赛

提高组 day1

1. 生活大爆炸版石头剪刀布

(rps.cpp/c/pas)

【问题描述】

石头剪刀布是常见的猜拳游戏： 石头胜剪刀， 剪刀胜布， 布胜石头。如果两个人出拳一样，则不分胜负。在《生活大爆炸》第二季第 8 集中出现了一种石头剪刀布的升级版游戏。升级版游戏在传统的石头剪刀布游戏的基础上，增加了两个新手势：

斯波克：《星际迷航》主角之一。

蜥蜴人：《星际迷航》中的反面角色。

这五种手势的胜负关系如表一所示，表中列出的是甲对乙的游戏结果。

表一 石头剪刀布升级版胜负关系

甲对乙的结果 \ 乙	剪刀	石头	布	蜥蜴人	斯波克
剪刀	平	输	赢	赢	输
石头		平	输	赢	输
布			平	输	赢
蜥蜴人				平	赢
斯波克					平

现在，小 A和小 B尝试玩这种升级版的猜拳游戏。 已知他们的出拳都是有周期性规律的，但周期长度不一定相等。例如：如果小 A以“石头 - 布 - 石头 - 剪刀 - 蜥蜴人 - 斯波克”长度为 6的周期出拳，那么他的出拳序列就是“石头 - 布 - 石头 - 剪刀 - 蜥蜴人 - 斯波克 - 石头 - 布 - 石头 - 剪刀 - 蜥蜴人 - 斯波克 - ,, ”，而如果小 B以“剪刀 - 石头 - 布 - 斯波克 - 蜥蜴人”长度为 5的周期出拳，那么他出拳的序列就是“剪刀 - 石头 - 布 - 斯波克 - 蜥蜴人 - 剪刀 - 石头 - 布 - 斯波克 - 蜥蜴人 - ,, ”

已知小 A和小 B一共进行 N次猜拳。 每一次赢的人得 1分，输的得 0分；平局两人都得 0分。现请你统计 N次猜拳结束之后两人的得分。

【输入】

输入文件名为 rps.in 。

第一行包含三个整数： N, NA, NB, 分别表示共进行 N次猜拳、小 A出拳的周期

长度，小 B 出拳的周期长度。数与数之间以一个空格分隔。

第二行包含 NA个整数， 表示小 A 出拳的规律， 第三行包含 NB个整数， 表示小 B 出拳的规律。 其中， 0 表示 “ 剪刀 ”， 1 表示 “ 石头 ”， 2 表示 “ 布 ”， 3 表示 “ 蜥蜴人 ”， 4 表示 “ 斯波克 ”。数与数之间以一个空格分隔。

【输出】

输出文件名为 rps.out 。

输出一行， 包含两个整数，以一个空格分隔，分别表示小 A、小 B 的得分。

【输入输出样例 1】

rps.in	rps.out
10 5 6 0 1 2 3 4 0 3 4 2 1 0	6 2

【输入输出样例 2】

rps.in	rps.out
9 5 5 0 1 2 3 4 1 0 3 2 4	4 4

【数据说明】

对于 100%的数据， 0 < N ≤ 200 ， 0 < NA ≤ 200 ， 0 < NB ≤ 200 。

2. 联合权值

(link.cpp/c/pas)

【问题描述】

无向连通图 G有 n 个点， n-1 条边。点从 1到 n 依次编号， 编号为 i 的点的权值为 Wi ，每条边的长度均为 1。图上两点 (u， v) 的距离定义为 u 点到 v 点的最短距离。对于图 G上的点对 (u, v) ，若它们的距离为 2，则它们之间会产生 Wu× Wv 的联合权值。

请问图 G上所有可产生联合权值的 有序点对 中，联合权值最大的是多少？所有联合权值之和是多少？

【输入】

输入文件名为 link.in 。

第一行包含 1 个整数 n。

接下来 n-1 行，每行包含 2 个用空格隔开的正整数 u、v，表示编号为 u 和编号为 v 的点之间有边相连。

最后 1 行，包含 n 个正整数，每两个正整数之间用一个空格隔开，其中第 i 个整数表示图 G上编号为 i 的点的权值为 Wi。

【输出】

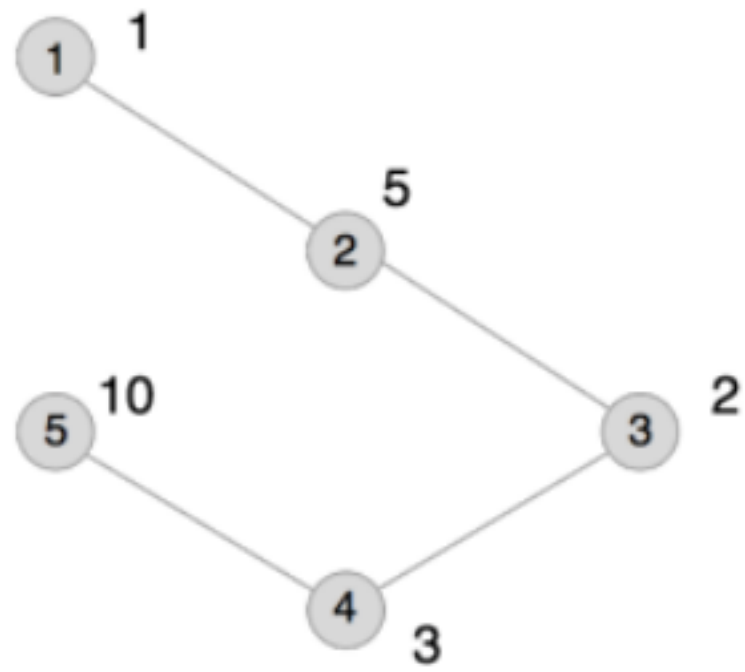
输出文件名为 link.out 。

输出共 1 行，包含 2 个整数，之间用一个空格隔开，依次为图 G上联合权值的最大值和所有联合权值之和。由于所有联合权值之和可能很大，输出它时要对 10007 取余。

【输入输出样例】

link.in	link.out
5 1 2 2 3 3 4 4 5 1 5 2 3 10	20 74

【样例说明】



本例输入的图如上所示，距离为 2 的有序点对有 (1,3) 、(2,4) 、(3,1) 、(3,5) 、(4,2) 、(5,3) 。其联合权值分别为 2、15、2、20、15、20。其中最大的是 20，总和为 74。

【数据说明】

对于 30%的数据，1<≤100；

对于 60%的数据，1<≤2000；

对于 100%的数据，1<≤200,000，0<Wi ≤10,000。

3. 飞扬的小鸟

（bird.cpp/c/pas)

【问题描述】

Flappy Bird 是一款风靡一时的休闲手机游戏。玩家需要不断控制点击手机屏幕的频率来调节小鸟的飞行高度，让小鸟顺利通过画面右方的管道缝隙。如果小鸟一不小心撞到了水管或者掉在地上的话，便宣告失败。

为了简化问题，我们对游戏规则进行了简化和改编：

1. 游戏界面是一个长为 n，高 为 m的二维平面，其中有 k 个管道（忽略管道的宽度） 。
2. 小鸟始终在游戏界面内移动。小鸟从游戏界面最左边任意整数高度位置出发， 到达游戏界面最右边时，游戏完成。
3. 小鸟每个单位时间沿横坐标方向右移的距离为 1，竖直移动的距离由玩家控制。如果点击屏幕，小鸟就会上升一定高度 X，每个单位时间可以点击多次，效果叠加；如果不点击屏幕，小鸟就会下降一定高度 Y。小鸟位于横坐标方向不同位置时，上升的高度 X和下降的高度 Y可能互不相同。
4. 小鸟高度等于 0 或者小鸟碰到管道时， 游 戏 失 败 。小 鸟 高 度 为 m时，无法再上升。

现在，请你判断是否可以完成游戏。如果可以，输出最少点击屏幕数；否则，输出小鸟最多可以通过多少个管道缝隙。

【输入】

输入文件名为 bird.in 。

第 1 行有 3 个整数 n，m，k，分别表示游戏界面的长度，高度和水管的数量，每两个整数之间用一个空格隔开；

接下来的 n 行，每行 2 个用一个空格隔开的整数 X和 Y，依次表示在横坐标位置 0~n-1 上玩家点击屏幕后，小鸟在下一位置上升的高度 X，以及在这个位置上玩家不点击屏幕时，小鸟在下一位置下降的高度 Y。

接下来 k 行，每行 3 个整数 P，L，H，每两个整数之间用一个空格隔开。每行表示一个管道，其中 P 表示管道的横坐标， L 表示此管道缝隙的下边沿高度为 L，H表示管道缝隙上边沿的高度（输入数据保证 P 各不相同，但不保证按照大小顺序给出） 。

【输出】

输出文件名为 bird.out 。

共两行。

第一行，包含一个整数，如果可以成功完成游戏，则输出 1，否则输出 0。

第二行，包含一个整数，如果第一行为 1，则输出成功完成游戏需要最少点击屏幕数，

否则，输出小鸟最多可以通过多少个管道缝隙。

【输入输出样例　1】

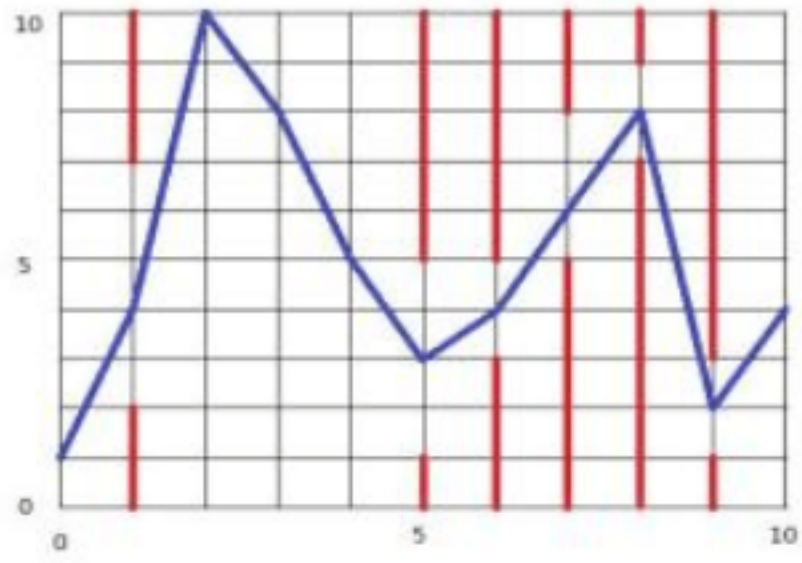
bird .in	bird .out
10 10 6 3 9 9 9 1 2 1 3 1 2 1 1 2 1 2 1 1 6 2 2 1 2 7 5 1 5 6 3 5 7 5 8 8 7 9 9 1 3	1 6

【输入输出样例　2】

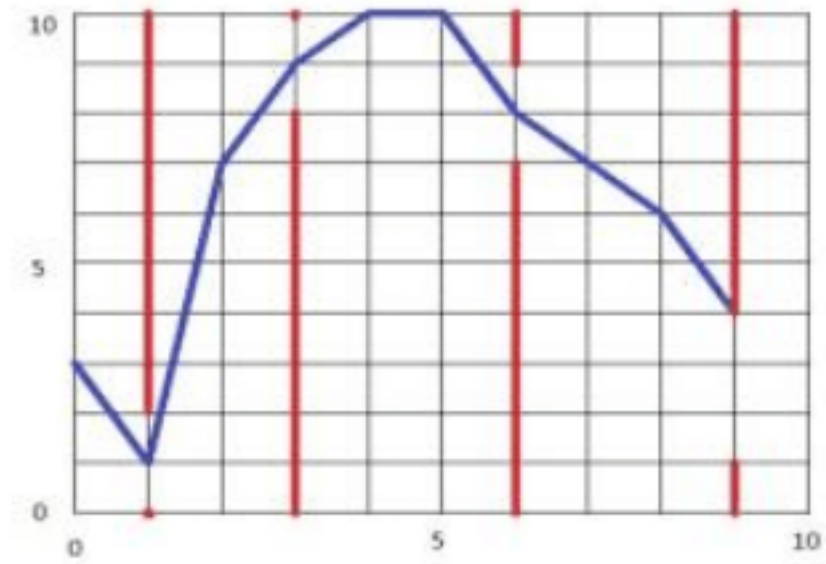
bird .in	bird .out
10 10 4 1 2 3 1 2 2 1 8 1 8 3 2 2 1 2 1 2 2 1 2 1 0 2 6 7 9 9 1 4 3 8 10	0 3

【输入输出样例说明】

如下图所示，蓝色直线表示小鸟的飞行轨迹，红色直线表示管道。



输入输出样例1说明



输入输出样例2说明

【数据范围】

对于 30%的数据： 5≤n≤10， 5≤m≤10， k=0，保证存在一组最优解使得同一单位时间最多点击屏幕 3 次；

对于 50%的数据： 5≤n≤20，5≤m≤10，保证存在一组最优解使得同一单位时间最多点击屏幕 3 次；

对于 70%的数据： 5≤n≤1000， 5≤m≤100；

对于 100%的数据： 5≤n≤10000，5≤m≤1000，0≤k<n， 0<X<n，0<Y<m，0<P<n， 0≤L<H≤m， L+1<H。

CCF全国信息学奥林匹克联赛（ NOIP2014）复赛

提高组 day2

1. 无线网络发射器选址

(wireless.cpp/c/pas)

【问题描述】

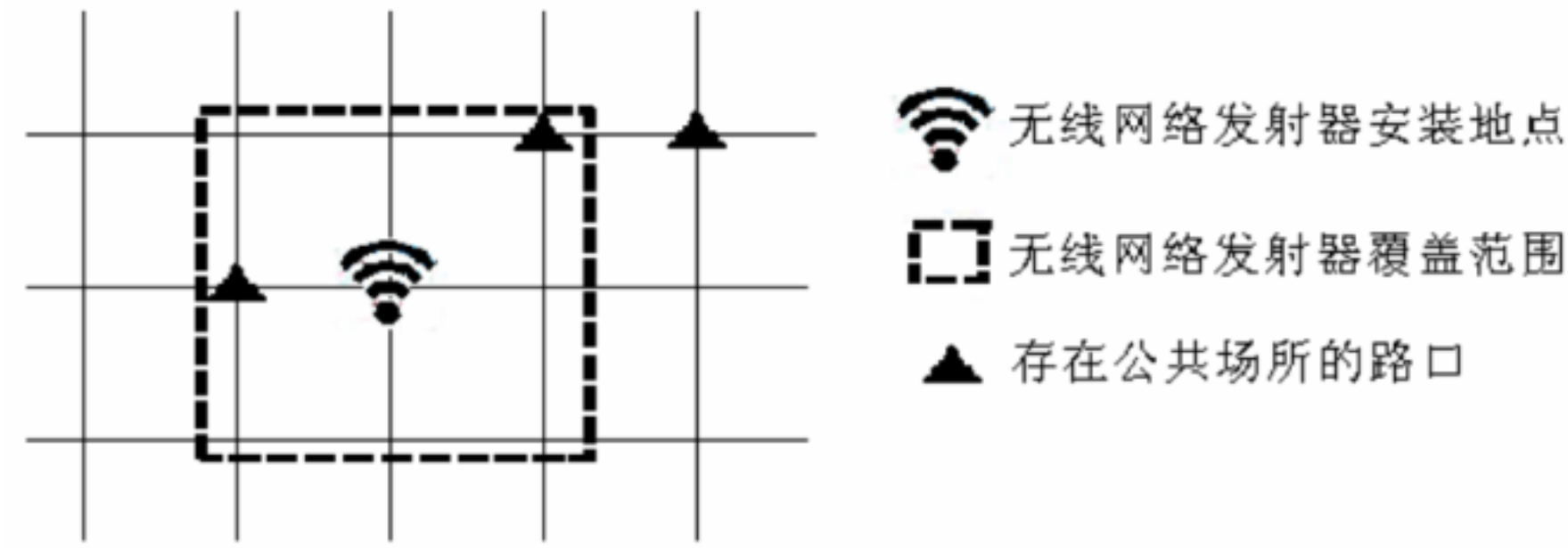
随着智能手机的日益普及， 人们对无线网的需求日益增大。 某城市决定对城市内的公共场所覆盖无线网。

假设该城市的布局为由严格平行的 129 条东西向街道和 129 条南北向街道所形成的网格状，并且相邻的平行街道之间的距离都是恒定值 1。东西向街道从北到南依次编号为 0,1,2 ，128，南北向街道从西到东依次编号为 0,1,2 ， 128。

东西向街道和南北向街道相交形成路口， 规定编号为 x 的南北向街道和编号为 y 的东西向街道形成的路口的坐标是（ x, y ）。 在某些路口存在一定数量的公共场所 。

由于政府财政问题， 只能安装一个大型无线网络发射器。 该无线网络发射器的传播范围是一个以该点为中心，边长为 2*d 的正方形。传播范围包括正方形边界。

例如下图是一个 d = 1 的无线网络发射器的覆盖范围示意图。



现在政府有关部门准备安装一个传播参数为 d 的无线网络发射器，希望你帮助他们在城市内找出合适的安装地点，使得覆盖的公共场所最多。

【输入】

输入文件名为 wireless.in 。

第一行包含一个整数 d，表示无线网络发射器的传播距离。

第二行包含一个整数 n，表示有公共场所的路口数目。

接下来 n 行，每行给出三个整数 x, y, k, 中间用一个空格隔开，分别代表路口的坐标(x, y) 以及该路口公共场所的数量。同一坐标只会给出一次。

【输出】

输出文件名为 wireless.out 。

输出一行，包含两个整数，用一个空格隔开，分别表示能覆盖最多公共场所的安装地点方案数，以及能覆盖的最多公共场所的数量。

【输入输出样例】

wireless.in	wireless.out
1 2 4 4 10 6 6 20	1 30

【数据说明】

对于 100%的数据，$1 \le d \le 20$，$1 \le n \le 20$，$0 \le x \le 128$，$0 \le y \le 128$，$0 < k \le 1{,}000{,}000$ 。

2. 寻找道路

(road.cpp/c/pas)

【问题描述】

在有向图 G 中，每条边的长度均为 1，现给定起点和终点，请你在图中找一条从起点到终点的路径，该路径满足以下条件：

1. 路径上的所有点的出边所指向的点都直接或间接与终点连通。
2. 在满足条件 1 的情况下使路径最短。

注意：图 G中可能存在重边和自环，题目保证终点没有出边。

请你输出符合条件的路径的长度。

【输入】

输入文件名为 road.in 。

第一行有两个用一个空格隔开的整数 n 和 m，表示图有 n 个点和 m条边。

接下来的 m行每行 2 个整数 x、y，之间用一个空格隔开，表示有一条边从点 x 指向点 y。

最后一行有两个用一个空格隔开的整数 s、t，表示起点为 s，终点为 t。

【输出】

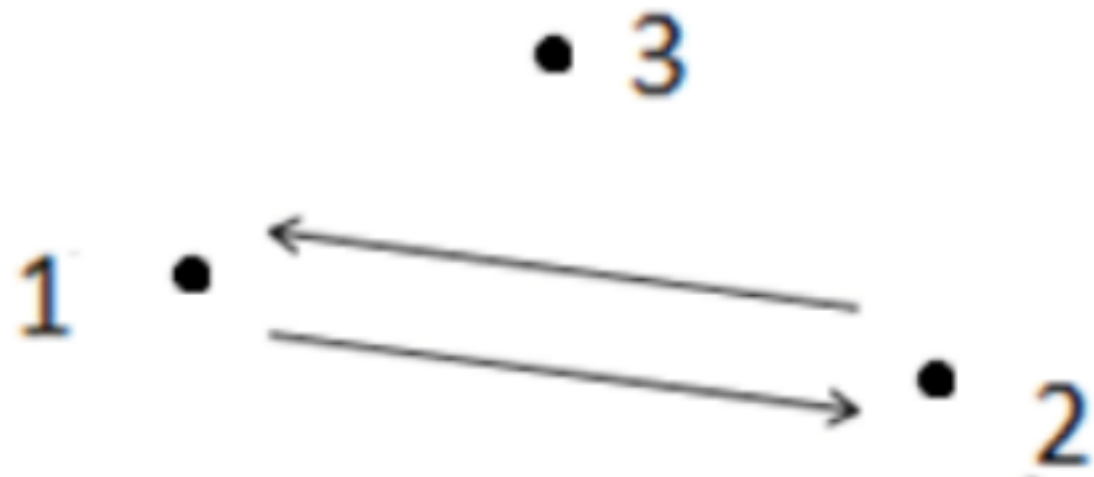
输出文件名为 road.out 。

输出只有一行，包含一个整数，表示满足题目描述的最短路径的长度。如果这样的路径不存在，输出 -1。

【输入输出样例 1】

road.in	road.out
3 2 1 2 2 1 1 3	－1

【输入输出样例说明】



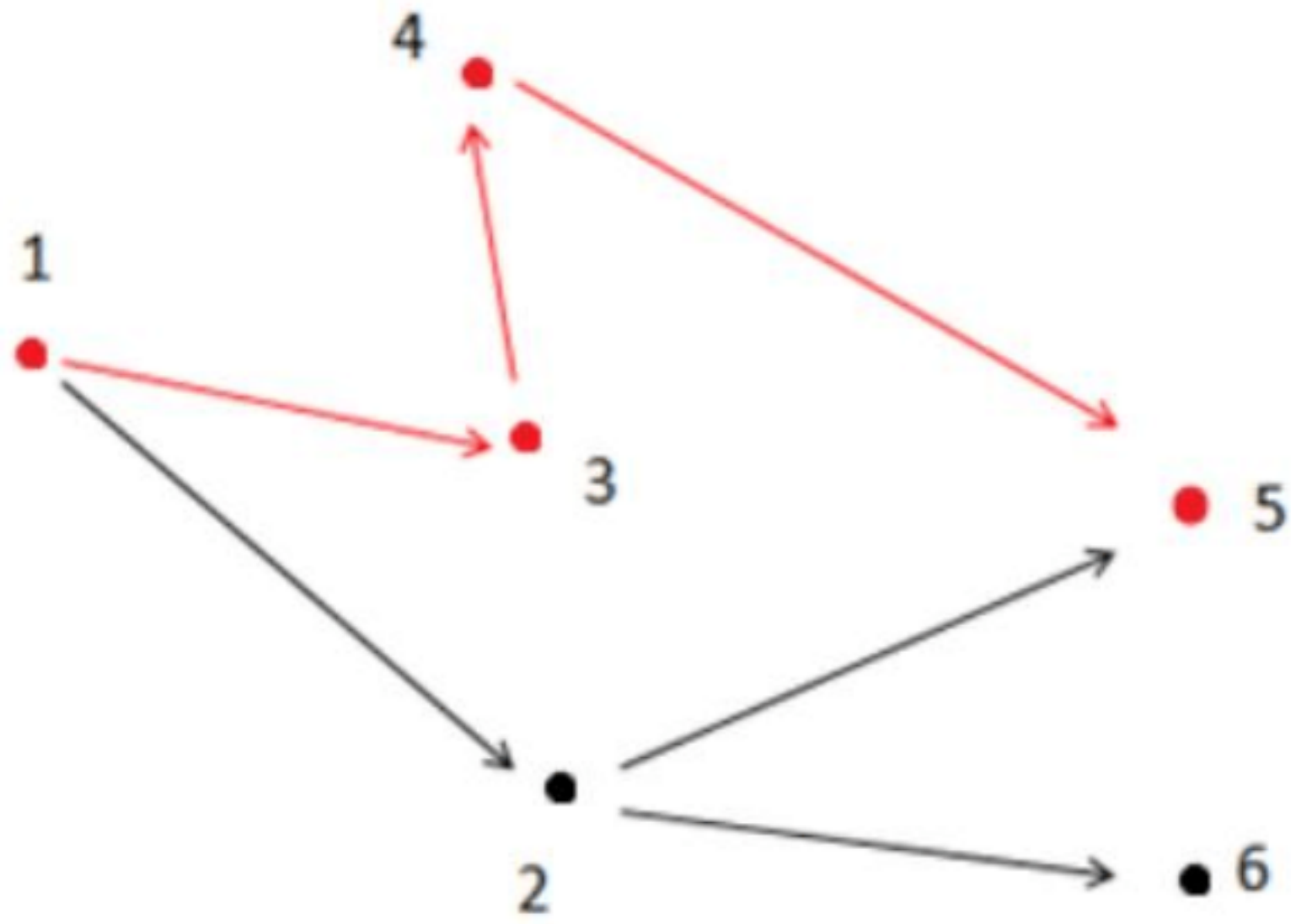
如上图所示，箭头表示有向道路，圆点表示城市。起点 1 与终点 3 不连通，所以满足题目描述的路径不存在，故输出 -1。

【输入输出样例 2】

road.in	road.out

6 6 1 2 1 3 2 6 2 5 4 5 3 4 1 5	3

【输入输出样例说明】



如上图所示，满足条件的路径为 1->3->4->5 。注意点 2 不能在答案路径中，因为点 2 连了一条边到点 6，而点 6 不与终点 5 连通。

【数据说明】

对于 30%的数据， 0< n ≤10， 0< m ≤20；

对于 60%的数据， 0< n ≤100， 0< m ≤2000；

对于 100%的数据， 0< n ≤10,000 ， 0< m ≤200,000 ， 0< x,y,s,t ≤n， x≠t 。

3. 解方程

(equation.cpp/c/pas)

【问题描述】

已知多项式方程：

$$a_0 + a_1x + a_2x^2 + \cdots + a_nx^n = 0$$

求这个方程在 [1, m] 内的整数解（ n 和 m均为正整数）。

【输入】

输入文件名为 equation.in 。

输入共 n+2 行。

第一行包含 2 个整数 n、m，每两个整数之间用一个空格隔开。

接下来的 n+1 行每行包含一个整数，依次为 a0,a1,a2，,, ,an。

【输出】

输出文件名为 equation.out 。

第一行输出方程在 [1, m] 内的整数解的个数。

接下来每行一个整数，按照从小到大的顺序依次输出方程在 [1, m] 内的一个整数解。

【输入输出样例 1】

equation.in	equation.out
2 10 1 -2 1	1 1

【输入输出样例 2】

equation.in	equation.out
2 10 2 -3 1	2 1 2

【输入输出样例 3】

equation.in	equation.out
2 10 1 3 2	0

【数据说明】

对于 30%的数据， 0<n≤2， |ai| ≤100， an≠0， m≤100；

对于 50%的数据， $0<n\leq 100$， $|ai|\leq 10^{100}$， $an\neq 0$， $m\leq 100$；

对于 70%的数据， $0<n\leq 100$， $|ai|\leq 10^{10000}$， $an\neq 0$， $m\leq 10000$；

对于 100%的数据， $0<n\leq 100$， $|ai|\leq 10^{10000}$， $an\neq 0$， $m\leq 1000000$。

CCF 全国信息学奥林匹克联赛 （NOIP2015 ） 复赛

提高组 day1

（请选手务必仔细阅读本页内容 ）

一. 题目概况

中文题目名称	神奇的幻方	信息传递	斗地主
英文题目与子目录名	magic	message	landlords
可执行文件名	magic	message	landlords
输入文件名	magic.in	message.in	landlords.in
输出文件名	magic.out	message.out	landlords.out
每个测试点时限	1 秒	1 秒	2 秒
测试点数目	10	10	20
每个测试点分值	10	10	5
附加样例文件	有	有	有
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
运行内存上限	128M	128M	1G

二. 提交源程序文件名

对于C++语言	magic.cpp	message.cpp	landlords.cpp
对于C语言	magic.c	message.c	landlords.c
对于pascal语言	magic.pas	message.pas	landlords.pas

三. 编译命令（不包含任何优化开关）

对于C++语言	g++ -o magic magic.cpp -lm	g++ -o message message.cpp -lm	g++ -o landlords landlords.cpp -lm
对于C语言	gcc -o magic magic.c -lm	gcc -o message message.c -lm	gcc -o landlords landlords.c -lm
对于pascal语言	fpcmagic.pas	fpcmessage.pas	fpclandlords.pas

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、全国统一评测时采用的机器配置为： CPU AMD Athlon(tm) II x2 240 processor ， 2.8GHz，内存 4G，上述时限以此配置为准。

4、只提供 Linux 格式附加样例文件。

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. 神奇的幻方
(magic.cpp/c/pas)

【问题描述】

幻方是一种很神奇的 N*N矩阵：它由数字 1,2,3, … … , N*N 构成，且每行、每列及两条对角线上的数字之和都相同。

当N为奇数时，我们可以通过以下方法构建一个幻方：

首先将 1写在第一行的中间。

之后，按如下方式从小到大依次填写每个数 K(K= 2,3, … , N*N) ：

1. 若(K- 1) 在第一行但不在最后一列，则将 K填在最后一行， (K- 1) 所在列的右一列；
2. 若(K- 1) 在最后一列但不在第一行，则将 K填在第一列， (K- 1) 所在行的上一行；
3. 若(K- 1) 在第一行最后一列，则将 K填在 (K- 1) 的正下方；
4. 若(K- 1) 既不在第一行，也不在最后一列，如果 (K- 1) 的右上方还未填数，则将 K填在(K- 1) 的右上方，否则将 K填在 (K- 1) 的正下方

现给定 N, 请按上述方法构造 N*N 的幻方。

【输入格式】

输入文件名为 magic.in 。

输入文件只有一行，包含一个整数 N, 即幻方的大小。

【输出格式】

输出文件名为 magic.out 。

输出文件包含 N行，每行 N个整数，即按上述方法构造出的 N*N的幻方。相邻两个整数之间用单个空格隔开。

【输入输出样例 1】

magic.in	magic.out
3	8 1 6 3 5 7 4 9 2

见选手目录下的 magic/magic1.in 和magic/magic1.ans 。

【输入输出样例 2】

见选手目录下的 magic/magic2.in 和magic/magic2.ans 。

【数据规模与约定】

对于100%的数据， 1 ≤N≤ 39且N为奇数。

2. 信息传递
(message.cpp/c/pas)

【问题描述】

有n个同学（编号为 1到n）正在玩一个信息传递的游戏。在游戏里每人都有一个固定的信息传递对象，其中，编号为 i 的同学的信息传递对象是编号为 Ti 的同学。游戏开始时，每人都只知道自己的生日。之后每一轮中，所有人会同时将自己当前所知的生日信息告诉各自的信息传递对象（注意：可能有人可以从若干人那里获取信息，但是每人只会把信息告诉一个人，即自己的信息传递对象）。当有人从别人口中得知自己的生日时，游戏结束。请问该游戏一共可以进行几轮？

【输入格式】

输入文件名为 message.in 。

输入共 2行。

第 1 行包含 1 个正整数 n，表示 n个人。

第 2 行包含 n个用空格隔开的正整数 T1,T2, … ，Tn，其中第 Ti 个整数表示编号为i 的同学的信息传递对象是编号为 Ti 的同学， Ti ≤n且Ti ≠i 。

数据保证游戏一定会结束。

【输出格式】

输出文件名为 message.out。

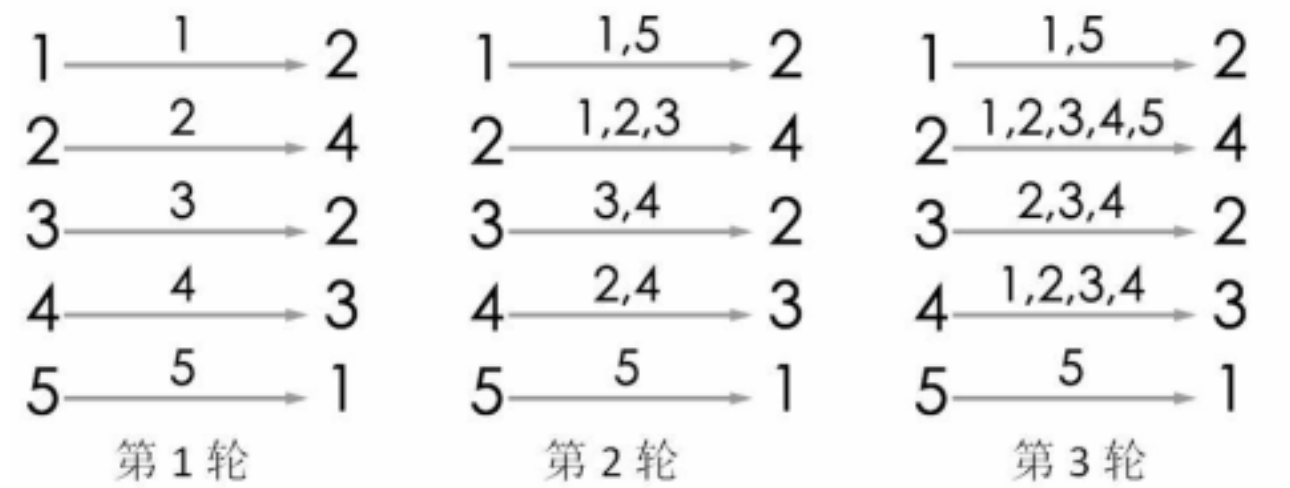
输出共 1行，包含 1个整数，表示游戏一共可以进行多少轮。

【输入输出样例 1】

message.in	message.out
5 24 2 3 1	3

见选手目录下的 message/message1.in 与message/message1.ans。

【输入输出样例 1说明】



游戏的流程如图所示。当进行完第 3轮游戏后， 4号玩家会听到 2号玩家告诉他自己的生日，所以答案为 3。当然，第 3轮游戏后， 2号玩家、 3号玩家都能从自己的消息来源得知自己的生日，同样符合游戏结束的条件。

【样例输入输出 2】

见选手目录下的 message/message2.in 与message/message2.ans。

【数据规模与约定】

对于 30%的数据 n≤200；

对于 60%的数据， n≤ 2500 ；

对于 100%的数据， n≤ 200000。

3. 斗地主
（landlords.cpp/c/pas)

【问题描述】

牛牛最近迷上了一种叫斗地主的扑克游戏。斗地主是一种使用黑桃、红心、梅花、方片的 A到K加上大小王的共 54张牌来进行的扑克牌游戏。在斗地主中，牌的大小关系根据 牌的数码 表示如下： 3<4<5<6<7<8<9<10<J<Q<K<A<2<小王 <大王，而花色并不对牌的大小产生影响 。每一局游戏中，一副 手牌 由n张牌组成。游戏者每次可以根据规定的 牌型 进行出牌，首先打光自己的手牌一方取得游戏的胜利。

现在，牛牛只想知道，对于自己的若干组 手牌，分别最少需要多少次出牌可以将它们打光。请你帮他解决这个问题。

需要注意的是，本题中游戏者每次可以出手的 牌型 与一般的斗地主相似而略有不同。具体规则如下：

牌型	牌型说明	牌型举例照片
火箭	即双王（双鬼牌）。	
炸弹	四张同点牌。如四个 A。	
单张牌	单张牌，比如 3。	
对子牌	两张码数相同的牌。	
三张牌	三张码数相同的牌。	
三带一	三张码数相同的牌 +一张单牌。例如：三张 3+单4	
三带二	三张码数相同的牌 +一对牌。例如：三张 3+对 4	
单顺子	五张或更多码数连续的单牌（不包括 2点和双王）例如：单 7+ 单 8+ 单 9+ 单 10+ 单 J 。另外，在顺牌（单顺子、双顺子、三顺子）中，牌的花色不要求相同。	
双顺子	三对或更多码数连续的对牌（不包括 2 点和双王）。例如：对 3+对4+对5。	
三顺子	二个或更多码数连续的三张牌（不能包括 2 点和双王）。例如：三张 3+ 三张 4+ 三张 5 。	
四带二	四张码数相同的牌 +任意两张单牌（或任意两对牌）例如：四张 5+ 单 3+ 单 8 或四张 4+ 对 5+ 对 7	

【输入格式】

输入文件名为 landlords.in 。

第一行包含用空格隔开的 2个正整数 T,n，表示手牌的组数以及每组手牌的张数。

接下来 T组数据，每组数据 n行，每行一个非负整数对 ai,bi ，表示一张牌，其中 ai 表示牌的数码， bi 表示牌的花色，中间用空格隔开。特别的，我们用 1来表示数码 A， 11表示数码 J，12表示数码 Q 13表示数码 K；黑桃、红心、梅花、方片分别用 1-4 来表示；小王的表示方法为 0 1，大王的表示方法为 0 2。

【输出格式】

输出文件名为 landlords.out 。

共T行，每行一个整数，表示打光第 i 组手牌的最少次数。

【输入输出样例 1】

landlords.in	landlords.out
18 74 84 91 10 4 11 1 51 14 11	3

见选手目录下的 landlords/landlords1.in 与 landlords/landlords1.ans 。

【输入输出样例 1说明】

共有 1组手牌，包含 8张牌：方片 7，方片 8，黑桃 9，方片 10，黑桃 J，黑桃 5，方片 A以及黑桃 A。可以通过打单顺子（方片 7，方片 8，黑桃 9，方片 10，黑桃 J），单张牌（黑桃 5）以及对子牌（黑桃 A以及方片 A）在 3次内打光。

【输入输出样例 2】

landlords.in	landlords.out
117 12 3 43 23 54 10 2 33 12 2 01 13 10 1 62 12 1 11 3	6

5 2 12 4 2 2 7 2		

见选手目录下的 landlords/landlords2.in　　与 landlords/landlords2.ans　　。

【样例输入输出 3】

见选手目录下的 landlords/landlords3.in　　与 landlords/landlords3.ans　　。

【数据规模与约定】

对于不同的测试点，我们约定手牌组数与张数的规模如下：

测试点编号	T	n	测试点编号	T	n
1	100	2	11	100	14
2	100	2	12	100	15
3	100	3	13	10	16
4	100	3	14	10	17
5	100	4	15	10	18
6	100	4	16	10	19
7	100	10	17	10	20
8	100	11	18	10	21
9	100	12	19	10	22
10	100	13	20	10	23

数据保证：所有的手牌都是随机生成的。

CCF 全国信息学奥林匹克联赛 （NOIP2015 ） 复赛

提高组 day2

（请选手务必仔细阅读本页内容 ）

一. 题目概况

中文题目名称	跳石头	子串	运输计划
英文题目与子目录名	stone	substring	transport
可执行文件名	stone	substring	transport
输入文件名	stone.in	substring.in	transport.in
输出文件名	stone.out	substring.out	transport.out
每个测试点时限	1 秒	1 秒	1 秒
测试点数目	10	10	20
每个测试点分值	10	10	5
附加样例文件	有	有	有
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
运行内存上限	128M	128M	256M

二. 提交源程序文件名

对于 C++语言	stone.cpp	substring.cpp	transport.cpp
对于C语言	stone.c	substring.c	transport.c
对于pascal语言	stone.pas	substring.pas	transport.pas

三. 编译命令 （不包含任何优化开关 ）

对于 C++语言	g++ -o stone stone.cpp -lm	g++ -o substring substring.cpp -lm	g++ -o transport transport.cpp -lm
对于C语言	gcc -o stone stone.c -lm	gcc -o substring substring.c -lm	gcc -o transport transport.c -lm
对于pascal语言	fpcstone.pas	fpcsubstring.pas	fpctransport.pas

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、全国统一评测时采用的机器配置为： CPU AMD Athlon(tm) II x2 240 processor ， 2.8GHz，内存 4G，上述时限以此配置为准。

4、只提供 Linux 格式附加样例文件。

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. 跳石头

(stone.cpp/c/pas)

【问题描述】

一年一度的“跳石头”比赛又要开始了！

这项比赛将在一条笔直的河道中进行，河道中分布着一些巨大岩石。组委会已经选择好了两块岩石作为比赛起点和终点。在起点和终点之间，有 N 块岩石（不含起点和终点的岩石）。在比赛过程中，选手们将从起点出发，每一步跳向相邻的岩石，直至到达终点。

为了提高比赛难度，组委会计划移走一些岩石，使得选手们在比赛过程中的最短跳跃距离尽可能长。由于预算限制，组委会至多从起点和终点之间移走 M 块岩石（不能移走起点和终点的岩石）。

【输入格式】

输入文件名为 stone.in 。

输入文件第一行包含三个整数 L，N，M，分别表示起点到终点的距离，起点和终点之间的岩石数，以及组委会至多移走的岩石数。

接下来 N 行，每行一个整数，第 i 行的整数 Di（0 < Di < L ）表示第 i 块岩石与起点的距离。这些岩石按与起点距离从小到大的顺序给出，且不会有两个岩石出现在同一个位置。

【输出格式】

输出文件名为 stone.out 。

输出文件只包含一个整数，即最短跳跃距离的最大值。

【输入输出样例 1】

stone.in	stone.out
2552 2 11 14 17 21	4

见选手目录下的 stone/stone1.in 和stone/stone1.ans 。

【输入输出样例 1说明】

将与起点距离为 2和14的两个岩石移走后，最短的跳跃距离为 4（从与起点距离17 的岩石跳到距离 21 的岩石，或者从距离 21 的岩石跳到终点）。

【输入输出样例 2】

见选手目录下的 stone/stone2.in 和stone/stone2.ans 。

【数据规模与约定】

对于 20%的数据， 0≤M≤N≤10。

对于 50%的数据， 0≤M≤N≤100。

对于 100%的数据， 0≤M≤N≤50,000， 1≤L≤1,000,000,000 。

2. 子串
(substring.cpp/c/pas)

【问题描述】

有两个仅包含小写英文字母的字符串 A和 B。现在要从字符串 A中取出 k个互不重叠的非空子串，然后把这 k个子串按照其在字符串 A中出现的顺序依次连接起来得到一个新的字符串，请问有多少种方案可以使得这个新串与字符串 B相等？注意：子串取出的位置不同也认为是不同的方案。

【输入格式】

输入文件名为 substring.in 。

第一行是三个正整数 n，m，k，分别表示字符串 A的长度，字符串 B的长度，以及问题描述中所提到的 k，每两个整数之间用一个空格隔开。

第二行包含一个长度为 n的字符串，表示字符串 A。

第三行包含一个长度为 m的字符串，表示字符串 B。

【输出格式】

输出文件名为 substring.out 。

输出共一行，包含一个整数，表示所求方案数。 由于答案可能很大，所以这里要求输出答案对 1,000,000,007 取模的结果。

【输入输出样例 1】

substring.in	substring.out
6 3 1 aabaab aab	2

见选手目录下 substring/substring1.in 与 substring/substring1.ans 。

【输入输出样例 2】

substring.in	substring.out
6 3 2 aabaab aab	7

见选手目录下 substring/substring2.in 与 substring/substring2.ans 。

【输入输出样例 3】

substring.in	substring.out
6 3 3 aabaab aab	7

见选手目录下 substring/substring3.in 与 substring/substring3.ans 。

【输入输出样例说明】

所有合法方案如下：（加下划线的部分表示取出的子串）

样例1：aabaab / aab aab

样例2：aabaab / aabaab/ a abaab/ aab aab

aabaab / aabaab/ aab aab

样例3：aabaab / aabaab/ aabaab/ aabaab

aa b aab / a abaab / aab aab

【输入输出样例 4】

见选手目录下 substring/substring4.in 与substring/substring4.ans 。

【数据规模与约定】

对于第 1组数据： 1≤n≤500, 1≤m≤50, k=1；

对于第 2组至第 3组数据： 1≤n≤500, 1≤m≤50, k=2；

对于第 4组至第 5组数据： 1≤n≤500, 1≤m≤50, k=m;

对于第 1组至第 7组数据： 1≤n≤500, 1≤m≤50, 1≤k≤m;

对于第 1组至第 9组数据： 1≤n≤1000, 1≤m≤100, 1≤k≤m;

对于所有 10组数据： 1≤n≤1000, 1≤m≤200, 1≤k≤m。

3. 运输计划
（transport.cpp/c/pas)

【问题描述】

公元2044年，人类进入了宇宙纪元。

L 国有 n 个星球，还有 n-1 条双向航道，每条航道建立在两个星球之间，这 n-1 条航道连通了L国的所有星球。

小P掌管一家物流公司，该公司有很多个运输计划，每个运输计划形如：有一艘物流飞船需要从 ui 号星球沿最快的宇航路径飞行到 vi 号星球去。显然，飞船驶过一条航道是需要时间的，对于航道 j，任意飞船驶过它所花费的时间为 tj，并且任意两艘飞船之间不会产生任何干扰。

为了鼓励科技创新， L国国王同意小 P的物流公司参与 L国的航道建设，即允许小P 把某一条航道改造成虫洞，飞船驶过虫洞 不消耗时间。

在虫洞的建设完成前小 P的物流公司就预接了 m个运输计划。在虫洞建设完成后，这 m个运输计划会 同时开始，所有飞船 一起出发。当这 m个运输计划 都完成时，小P的物流公司的阶段性工作就完成了。

如果小 P可以自由选择将哪一条航道改造成虫洞，试求出小 P的物流公司完成阶段性工作所需要的最短时间是多少？

【输入格式】

输入文件名为 transport.in 。

第一行包括两个正整数 n、m，表示 L国中星球的数量及小 P公司预接的运输计划的数量，星球从 1到n编号。

接下来 n-1 行描述航道的建设情况，其中第 i 行包含三个整数 ai, bi 和ti，表示第i 条双向航道修建在 ai 与 bi 两个星球之间，任意飞船驶过它所花费的时间为ti 。

接下来 m行描述运输计划的情况，其中第 j 行包含两个正整数 uj 和vj，表示第 j 个运输计划是从 uj 号星球飞往 vj 号星球。

【输出格式】

输出文件名为 transport.out 。

共1行，包含 1个整数，表示小 P的物流公司完成阶段性工作所需要的最短时间。

【输入输出样例 1】

transport.in	transport.out
6 3 1 23 1 64 3 17 4 36 3 55 3 6 2 5 4 5	11

见选手目录下的 transport/transport1.in 与transport/transport1.ans 。

【输入输出样例 1说明】

将第 1条航道改造成虫洞：则三个计划耗时分别为： 11、12、11，故需要花费的时间为 12。

将第 2条航道改造成虫洞：则三个计划耗时分别为： 7、15、11，故需要花费的时间为 15。

将第 3条航道改造成虫洞：则三个计划耗时分别为： 4、8、11，故需要花费的时间为 11。

将第 4条航道改造成虫洞：则三个计划耗时分别为： 11、15、5，故需要花费的时间为 15。

将第 5条航道改造成虫洞：则三个计划耗时分别为： 11、10、6，故需要花费的时间为 11。

故将第 3条或第 5条航道改造成虫洞均可使得完成阶段性工作的耗时最短，需要花费的时间为 11。

【样例输入输出 2】

见选手目录下的 transport/transport2.in 与transport/transport2.ans 。

【数据规模与约定】

所有测试数据的范围和特点如下表所示

测试点编号	n=	m=	约定
1	100	1	
2		100	第i 条航道连接i 号星球与i+1 号星球
3			
4	2000	1	
5	1000	1000	第i 条航道连接i 号星球与i+1 号星球
6	2000	2000	
7	3000	3000	
8	1000	1000	
9	2000	2000	
10	3000	3000	
11	80000	1	
12	100000		
13	70000	70000	第i 条航道连接i 号星球与i+1 号星球
14	80000	80000	
15	90000	90000	
16	100000	100000	
17	80000	80000	
18	90000	90000	
19	100000	100000	
20	300000	300000	
所有数据			$1 \le a,b_i,u_i,v_i \le n, 0 \le t \le 1000$

请注意常数因子带来的程序效率上的影响。

第 22 届全国青少年信息学奥林匹克联赛

CCF-NOIP-2016

提高组（复赛）　　第一试

竞赛时间： 2016 年 11 月 19 日 8:30 ～12:00

题目名称	玩具谜题	天天爱跑步	换教室
题目类型	传统型	传统型	传统型
目录	toy	running	classroom
可执行文件名	toy	running	classroom
输入文件名	toy.in	running.in	classroom.in
输出文件名	toy.out	running.out	classroom.out
每个测试点时限	1.0 秒	2.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	20	20	25
每个测试点分值	5	5	4

提交源程序文件名

对于 C++　语言	toy.cpp	running.cpp	classroom.cpp
对于 C　　语言	toy.c	running.c	classroom.c
对于 Pascal 语言	toy.pas	running.pas	classroom.pas

编译选项

对于 C++　语言	-lm	-lm	-lm
对于 C　　语言	-lm	-lm	-lm
对于 Pascal 语言			

注意事项：

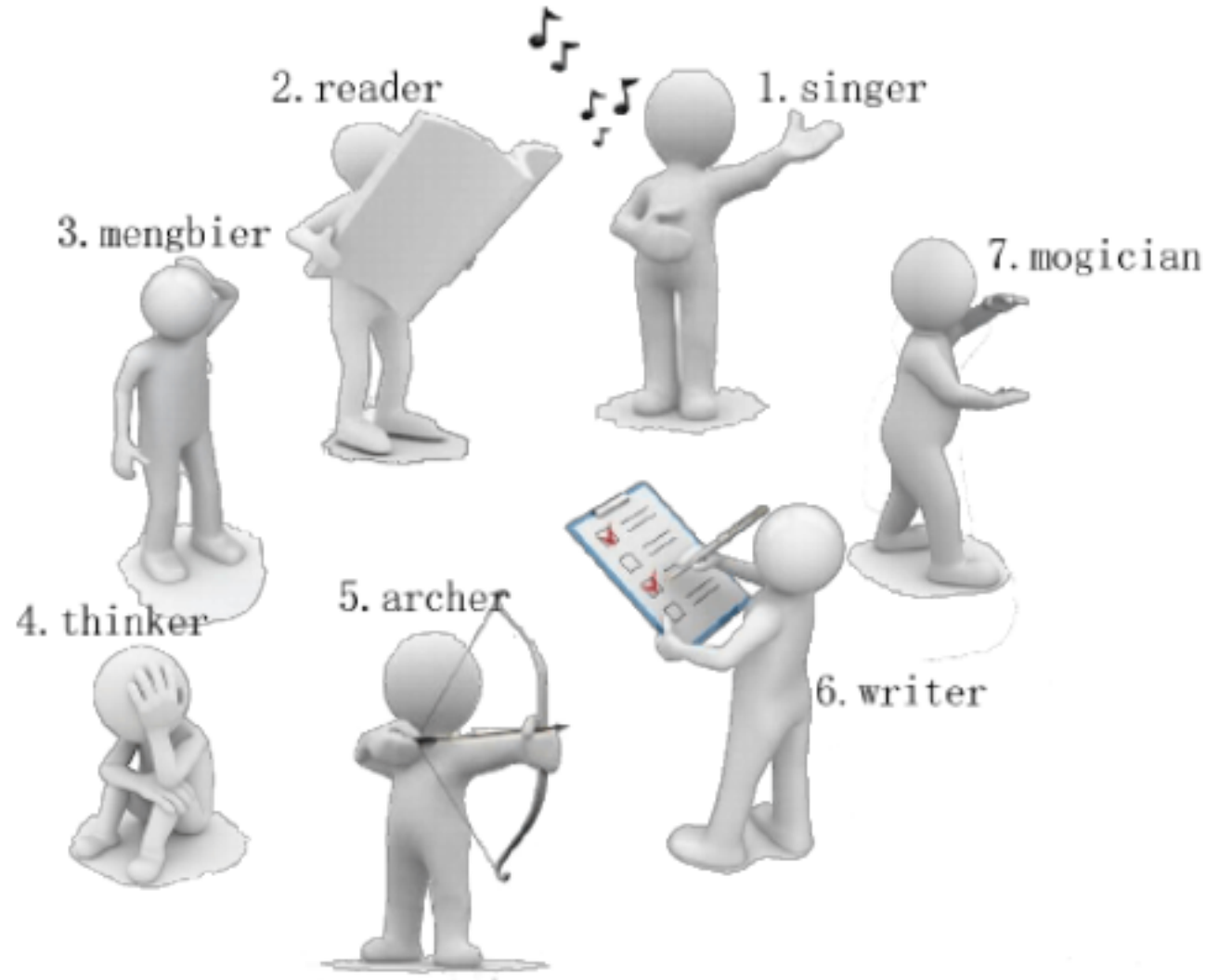
1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. 除非特殊说明，结果比较方式均为忽略行末空格及文末回车的全文比较。
3. C/C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
4. 全国统一评测时采用的机器配置为： CPU AMD Athlon(tm) II x2 240 processor，2.8GHz，内存 4G，上述时限以此配置为准。
5. 只提供 Linux 格式附加样例文件。
6. 评测在 NOI Linux 下进行。
7. 编译时不打开任何优化选项。

玩具谜题（ toy)

【问题描述】

小南有一套可爱的玩具小人，它们各有不同的职业。

有一天，这些玩具小人把小南的眼镜藏了起来。小南发现玩具小人们围成了一个圈，它们有的面朝圈内，有的面朝圈外。如下图：



这时 singer 告诉小南一个谜题： ‘眼镜藏在我左数第 3 个玩具小人的右数第 1 个玩具小人的左数第 2 个玩具小人那里。 ”

小南发现，这个谜题中玩具小人的朝向非常关键，因为朝内和朝外的玩具小人的左右方向是相反的：面朝圈内的玩具小人，它的左边是顺时针方向，右边是逆时针方向；而面向圈外的玩具小人，它的左边是逆时针方向，右边是顺时针方向。

小南一边艰难地辨认着玩具小人，一边数着：

‘singer 朝内，左数第 3 个是 archer 。

‘archer 朝外，右数第 1 个是 thinker 。

‘thinker 朝外，左数第 2 个是 writer 。

‘所以眼镜藏在 writer 这里！ ”

虽然成功找回了眼镜，但小南并没有放心。如果下次有更多的玩具小人藏他的眼镜，或是谜题的长度更长，他可能就无法找到眼镜了。所以小南希望你写程序帮他解决类似的谜题。这样的谜题具体可以描述为：

有 n 个玩具小人围成一圈，已知它们的职业和朝向。现在第 1 个玩具小人告诉小南一个包含 m 条指令的谜题，其中第 i 条指令形如“左数/右数第 s_i 个玩具小人 ”。你需要输出依次数完这些指令后，到达的玩具小人的职业。

【输入格式】

从文件 **toy.in**中读入数据。

输入的第一行包含两个正整数 n,m，表示玩具小人的个数和指令的条数。

接下来 n 行，每行包含一个整数和一个字符串，以 逆时针 为顺序给出每个玩具小人的朝向和职业。其中 0 表示朝向圈内， 1 表示朝向圈外。保证不会出现其他的数。字符串长度不超过 10 且仅由小写字母构成，字符串不为空，并且字符串两两不同。整数和字符串之间用一个空格隔开。

接下来 m 行，其中第 i 行包含两个整数 a_i, s_i，表示第 i 条指令。若 $a_i = 0$，表示向左数 s_i 个人；若 $a_i = 1$，表示向右数 s_i 个人。保证 a_i 不会出现其他的数， $1 \leq s_i < n$。

【输出格式】

输出到文件 **toy.out**中。输出一个字符串，表示从第一个读入的小人开始，依次数完m 条指令后到达的小人的职业。

【样例 1 输入】

```
7 3
0 singer
0 reader
0 mengbier
1 thinker
1 archer
0 writer
1 mogician
0 3
1 1
0 2
```

【样例 1 输出】

```
writer
```

【样例 1 说明】

这组数据就是【题目描述】中提到的例子。

【样例 2 输入】

```
10 10
1 c
0 r
0 p
1 d
1 e
1 m
1 t
1 y
1 u
0 v
1 7
1 1
1 4
0 5
0 3
0 1
1 6
1 2
0 8
0 4
```

【样例 2 输出】

```
y
```

【子任务】

子任务会给出部分测试数据的特点。如果你在解决题目中遇到了困难，可以尝试只解决一部分测试数据。

每个测试点的数据规模及特点如下表：

测试点	n	m	全朝内	全左数	$s_i = 1$	职业长度为 1
1	$= 20$	$= 10^3$	√	√	√	√
2			×			
3			√	×		
4			×			
5			√	√	×	
6			×			
7			√	×		
8			×			
9			√	√	√	×
10			×			
11			√	×		
12			×			
13			√	√	×	
14			×			
15			√	×		
16			×			
17	$= 10^5$	$= 10^5$	√	√		
18			×			
19			√	×		
20			×			

其中一些简写的列意义如下：

全朝内：若为“ √”， 表示该测试点保证所有的玩具小人都朝向圈内；

全左数：若为“ √”，表示该测试点保证所有的指令都向左数，即对任意的 $1 \le i \le m$，$a_i = 0$ ；

$s_i = 1$ ：若为“ √”，表示该测试点保证所有的指令都只数 1 个，即对任意的 $1 \le i \le m$，$s_i = 1$ ；

职业长度为 1：若为“ √”，表示该测试点保证所有玩具小人的职业一定是一个长度为 1 的字符串。

天天爱跑步（ running)

【问题描述】

小 C 同学认为跑步非常有趣，于是决定制作一款叫做《天天爱跑步》的游戏。《天 天爱跑步》是一个养成类游戏，需要玩家每天按时上线，完成打卡任务。

这个游戏的地图可以看作一棵包含 n 个结点和 n - 1 条边的树，每条边连接两个结点，且任意两个结点存在一条路径互相可达。树上结点编号为从 1到 n的连续正整数。

现在有 m 个玩家，第 i 个玩家的起点为 S_i，终点为 T_i。每天打卡任务开始时，所有玩家 在第 0 秒同时从 自己的起点 出发，以 每秒跑一条边 的速度，不间断地沿着 最短 路径向着 自己的终点 跑去，跑到终点后该玩家就算完成了打卡任务。（由于地图是一棵树，所以每个人的路径是唯一的）

小 C 想知道游戏的活跃度，所以在每个结点上都放置了一个观察员。在结点 j 的观 察员会选择在第 W_j 秒观察玩家，一个玩家能被这个观察员观察到当且仅当该玩家在第 W_j 秒也正好到达了结点 j。小 C 想知道每个观察员会观察到多少人？

注意：我们认为一个玩家到达自己的终点后该玩家就会结束游戏，他不能等待一段时间后再被观察员观察到。即对于把结点 j 作为终点的玩家：若他在第 W_j 秒前到达终点，则在结点 j 的观察员 不能观察到 该玩家；若他 正好 在第 W_j 秒到达终点，则在结点 j 的观察员 可以观察到 这个玩家。

【输入格式】

从文件 **running.in**中读入数据。

第一行有两个整数 n 和 m。其中 n代表树的结点数量，同时也是观察员的数量，m 代表玩家的数量。

接下来 n - 1 行每行两个整数 u 和 v，表示结点 u 到结点 v 有一条边。

接下来一行 n 个整数，其中第 j 个整数为 W_j，表示结点 j 出现观察员的时间。

接下来 m 行，每行两个整数 S_i 和 T_i，表示一个玩家的起点和终点。

对于所有的数据，保证 $1 \le S_i, T_i \le n$，$0 \le W_j \le n$。

【输出格式】

输出到文件 **running.out**中。

输出 1 行 n个整数，第 j 个整数表示结点 j 的观察员可以观察到多少人。

【样例 1 输入】

```
6 3
2 3
```

1 2

1 4

4 5

4 6

0 2 5 1 2 3

1 5

1 3

2 6

【样例 1 输出】

2 0 0 1 1 1

【样例 1 说明】

对于 1 号点， $W_1 = 0$，故只有起点为 1 号点的玩家才会被观察到，所以玩家 1 和玩家 2 被观察到，共 2 人被观察到。

对于 2 号点，没有玩家在第 2 秒时在此结点，共 0 人被观察到。

对于 3 号点，没有玩家在第 5 秒时在此结点，共 0 人被观察到。

对于 4 号点，玩家 1 被观察到，共 1 人被观察到。

对于 5 号点，玩家 1 被观察到，共 1 人被观察到。

对于 6 号点，玩家 3 被观察到，共 1 人被观察到。

【样例 2 输入】

5 3

1 2

2 3

2 4

1 5

0 1 0 3 0

3 1

1 4

5 5

【样例 2 输出】

1 2 1 0 1

【子任务】

每个测试点的数据规模及特点如下表所示。提示：数据范围的个位上的数字可以帮助判断是哪一种数据类型。

测试点编号	n	m	约定
1	= 991	= 991	所有人的起点等于自己的终点，即 $S_i = T_i$
2			
3	= 992	= 992	$W_j = 0$
4			
5	= 993	= 993	无
6	= 99994	= 99994	树退化成一条链，其中 1 与 2 有边，2 与 3 有边，...，n - 1 与 n 有边
7			
8			
9	= 99995	= 99995	所有的 $S_i = 1$
10			
11			
12			
13	= 99996	= 99996	所有的 $T_i = 1$
14			
15			
16			
17	= 99997	= 99997	无
18			
19			
20	= 299998	= 299998	

【提示】

如果你的程序需要用到较大的栈空间（这通常意味着需要较深层数的递归），请务必仔细阅读选手目录下的文档 **running/stack.pdf**，以了解在最终评测时栈空间的限制与在当前工作环境下调整栈空间限制的方法。

换教室（ classroom)

【问题描述】

对于刚上大学的牛牛来说，他面临的第一个问题是如何根据实际情况申请合适的课程。

在可以选择的课程中，有 2n 节课程安排在 n 个时间段上。在第 i (1 ≤i ≤n) 个时间段上，两节内容相同的课程同时在不同的地点进行，其中，牛牛预先被安排在教室 c_i 上课，而另一节课程在教室 d_i 进行。

在不提交任何申请的情况下，学生们需要按时间段的顺序依次完成所有的 n 节安排好的课程。如果学生想更换第 i 节课程的教室，则需要提出申请。若申请通过，学生就可以在第 i 个时间段去教室 d_i 上课，否则仍然在教室 d_i 上课。

由于更换教 室的需求太多，申请不一定能获得通过。通过计算，牛牛发现申请更换第 i 节课程的教室时，申请被通过的概率是一个已知的实数 k_i，并且对于不同课程的申请，被通过的概率是互相独立的。

学校规定，所有的申请只能在学期开始前一次性提交，并且每个人只能选择至多m节课程进行申请。这意味着牛牛必须一次性决定是否申请更换每节课的教室，而不能根据某些课程的申请结果来决定其他课程是否申请；牛牛可以申请自己最希望更换教室的 m门课程，也可以 不用完 这 m 个申请的机会，甚至可以一门课程都不申请。

因为不同的课程可能会被安排在不同的教室进行，所以牛牛需要利用课间时间从一间教室赶到另一间教室。

牛牛所在的大学有 v 个教室，有 e条道路。每条道路连接两间教室，并且是可以双向通行 的。

由于道路的长度和拥堵程度不同，通过不同的道路耗费的体力可能会有所不同。当第 i （ 1 ≤i ≤n - 1 ）节课结束后，牛牛就会从这节课的教室出发，选择一条耗费体力最少的 路径 前往下一节课的教室。现在牛牛想知道，申请哪几门课程可以使他因在教室间移动耗费的体力值的总和的期望值最小，请你帮他求出这个最小值。

现在牛牛想知道，申请哪几门课程可以使他因在教室间移动耗费的体力值的总和的期望值 最小，请你帮他求出这个最小值。

【输入格式】

从文件 **classroom.in** 中读入数据。

第一行四个整数 n, m, v, e。 n 表示这个学期内的时间段的数量； m 表示牛牛最多可以申请更换多少节课程的教室； v 表示牛牛学校里教室的数量； e 表示牛牛的学校里道路的数量。

第二行 n 个正整数，第 i （ 1 ≤i ≤n ）个正整数表示 c_i，即第 i 个时间段牛牛被安排上课的教室；保证 $1 \le c_i \le v$。

第三行 n 个正整数，第 i （ 1 ≤i ≤n ）个正整数表示 d_i，即第 i 个时间段另一间上同样课程的教室；保证 $1 \le d_i \le v$。

第四行 n 个实数，第 i （ 1 ≤i ≤n ）个实数表示 k_i ，即牛牛申请在第 i 个时间段更换教室获得通过的概率。保证 $0 \le k_i \le 1$ 。

接下来 e 行，每行三个正整数 a_j ， b_j ， w_j ，表示有一条双向道路连接教室 a_j ， b_j ，通过这条道路需要耗费的体力值是 w_j ；保证 $1 \le a_j, b_j \le v$ ， $1 \le w_j \le 100$ 。

保证 $1 \le n \le 2000$ ， $0 \le m \le 2000$ ， $1 \le v \le 300$ ， $0 \le e \le 90000$ 。

保证通过学校里的道路，从任何一间教室出发，都能到达其他所有的教室。

保证 输入的实数最多包含 3 位小数。

【输出格式】

输出到文件 **classroom.out** 中。

输出一行，包含一个实数，四舍五入精确到小数点后 恰好 2 位，表示答案。你的输出必须和标准输出 完全一样 才算正确。

测试数据保证四舍五入后的答案和准确答案的差的绝对值不大于 4×10^{-3} 。（如果你不知道什么是浮点误差，这段话可以理解为：对于大多数的算法，你可以正常地使用浮点数类型而不用对它进行特殊的处理）

【样例 1 输入】

```
3 2 3 3
2 1 2
1 2 1
0.8 0.2 0.5
1 2 5
1 3 3
2 3 1
```

【样例 1 输出】

```
2.80
```

【样例 1 说明】

所有可行的申请方案和期望收益如下表：

申请更换教室的时间段	申请通过的时间段	出现的概率	耗费的体力值	耗费的体力值的期望
无	无	1.0	8	8.0
1	1	0.8	4	4.8
	无	0.2	8	
2	2	0.2	0	6.4
	无	0.8	8	
3	3	0.5	4	6.0
	无	0.5	8	
1、2	1、2	0.16	4	4.48
	1	0.64	4	
	2	0.04	0	
	无	0.16	8	
1、3	1、3	0.4	0	2.8
	1	0.4	4	
	3	0.1	4	
	无	0.1	8	
2、3	2、3	0.1	4	5.2
	2	0.1	0	
	3	0.4	4	
	无	0.4	8	

【样例 2】

见选手目录下的 **classroom/classroom2.in** 与 **classroom/classroom2.ans**。

【提示】

1. 道路中可能会有 多条 双向道路连接相同的两间教室。也有可能有道路两端连接的是 同一间 教室。
2. 请注意区分 n, m, v, e 的意义， n 不是 教室的数量， m 不是 道路的数量。

【子任务】

测试点	n	m	v	特殊性质 1	特殊性质 2
1	≤1	≤1	≤300	×	×
2	≤2	≤0	≤20		
3		≤1	≤100		
4		≤2	≤300		
5	≤3	≤0	≤20	√	√
6		≤1	≤100		×
7		≤2	≤300	×	
8	≤10	≤0		√	√
9		≤1	≤20		×
10		≤2	≤100	×	
11		≤10	≤300		√
12	≤20	≤0	≤20	√	×
13		≤1	≤100	×	
14		≤2	≤300	√	
15		≤20		×	√
16	≤300	≤0	≤20		×
17		≤1	≤100		
18		≤2	≤300	√	√
19		≤300		×	
20	≤2000	≤0	≤20		×
21		≤1			
22		≤2	≤100		
23		≤2000			
24			≤300		
25					

特殊性质 1：图上任意两点 a_i，b_i，$a_i \neq b_i$间，存在一条耗费体力最少的路径只包含一条道路。

特殊性质 2：对于所有的 $1 \leq i \leq n$，$k_i = 1$。

第 22 届全国青少年信息学奥林匹克联赛

CCF-NOIP-2016

提高组（复赛）　　第二试

竞赛时间： 2016 年 11 月 20 日　8:30～12:00

题目名称	组合数问题	蚯蚓	愤怒的小鸟
题目类型	传统型	传统型	传统型
目录	problem	earthworm	angrybirds
可执行文件名	problem	earthworm	angrybirds
输入文件名	problem.in	earthworm.in	angrybirds.in
输出文件名	problem.out	earthworm.out	angrybirds.out
每个测试点时限	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	20	20	20
每个测试点分值	5	5	5

提交源程序文件名

对于 C++　语言	problem.cpp	earthworm.cpp	angrybirds.cpp
对于 C　　语言	problem.c	earthworm.c	angrybirds.c
对于 Pascal 语言	problem.pas	earthworm.pas	angrybirds.pas

编译选项

对于 C++　语言	-lm	-lm	-lm
对于 C　　语言	-lm	-lm	-lm
对于 Pascal 语言			

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. 除非特殊说明，结果比较方式均为忽略行末空格及文末回车的全文比较。
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. 全国统一评测时采用的机器配置为： CPU AMD Athlon(tm) II x2 240 processor，2.8GHz，内存 4G，上述时限以此配置为准。
5. 只提供 Linux 格式附加样例文件。
6. 评测在 NOI Linux 下进行。
7. 编译时不打开任何优化选项。

组合数问题（ problem)

【问题描述】

组合数 C_n^m 表示的是从 n 个物品中选出 m 个物品的方案数。举个例子，从 (1, 2, 3) 三个物品中选择两个物品可以有 (1, 2),(1, 3),(2, 3)这三种选择方法。根据组合数的定义，我们可以给出计算组合数 C_n^m 的一般公式：

$$C_n^m = \frac{n!}{m!(n-m)!}$$

其中 n! = 1 ×2 × ···×n。

小葱想知道如果给定 n,m 和 k，对于所有的 0 ≤i ≤n,0 ≤j ≤min (i, m) 有多少对 (i, j) 满足 C_i^j 是 k 的倍数。

【输入格式】

从文件 **problem.in**中读入数据。

第一行有两个整数 t,k，其中 t 代表该测试点总共有多少组测试数据， k 的意义见【问题描述】。

接下来 t 行每行两个整数 n,m，其中 n,m 的意义见【问题描述】。

【输出格式】

输出到文件 **problem.out**中。

t 行，每行一个整数代表所有的 0 ≤i ≤n, 0 ≤j ≤min (i, m) 中有多少对 (i , j) 满足 C_i^j 是 k 的倍数。

【样例 1 输入】

```
1 2
3 3
```

【样例 1 输出】

```
1
```

【样例 1 说明】

在所有可能的情况中，只有 C_2^1 = 2 是 2 的倍数。

【样例 2 输入】

```
2 5
4 5
6 7
```

【样例 2 输出】

```
0
7
```

【子任务】

测试点	n	m	k	t
1	≤3	≤3	= 2	= 1
2			= 3	$\leq 10^4$
3	≤7	≤7	= 4	= 1
4			= 5	$\leq 10^4$
5	≤10	≤10	= 6	= 1
6			= 7	$\leq 10^4$
7	≤20	≤100	= 8	= 1
8			= 9	$\leq 10^4$
9	≤25	≤2000	= 10	= 1
10			= 11	$\leq 10^4$
11	≤60	≤20	= 12	= 1
12			= 13	$\leq 10^4$
13	≤100	≤25	= 14	= 1
14			= 15	$\leq 10^4$
15		≤60	= 16	= 1
16			= 17	$\leq 10^4$
17	≤2000	≤100	= 18	= 1
18			= 19	$\leq 10^4$
19		≤2000	= 20	= 1
20			= 21	$\leq 10^4$

蚯蚓（ earthworm)

【问题描述】

本题中，我们将用符号 LcJ 表示对 c向下取整，例如： L3.0J = L3.1J = L3.9J = 3。

蛐蛐国最近蚯蚓成灾了！隔壁跳蚤国的跳蚤也拿蚯蚓们没办法，蛐蛐国王只好去请神刀手来帮他们消灭蚯蚓。

蛐蛐国里现在共有 n 只蚯蚓（ n 为正整数）。每只蚯蚓拥有长度，我们设第 i 只蚯蚓的长度为 a_i (i = 1, 2, . . . , n)，并保证所有的长度都是 非负整数（即：可能存在长度为 0 的蚯蚓）。

每一秒，神刀手会在所有的蚯蚓中，准确地找到最长的那一只（如有多个则任选一个）将其切成两半。神刀手切开蚯蚓的位置由常数 p （是满足 0 < p < 1 的有理数）决定，设这只蚯蚓长度为 x，神刀手会将其切成两只长度分别为 LpxJ 和 x- LpxJ 的蚯蚓。特殊地，如果这两个数的其中一个等于 0，则这个长度为 0的蚯蚓也会被保留。此 外，除了刚刚产生的两只新蚯蚓，其余蚯蚓的长度都会增加 q（是一个非负整常数）。

蛐蛐国王知道这样不是长久之计，因为蚯蚓不仅会越来越多，还会越来越长。蛐 蛐国王决定求助于一位有着洪荒之力的神秘人物，但是救兵还需要 m秒才能到来 ..... （m 为非负整数）蛐蛐国王希望知道这 m秒内的战况。

具体来说，他希望知道：

m秒内，每一秒被切断的蚯蚓被切断前的长度（有 m 个数）；

m 秒后，所有蚯蚓的长度（有 n+ m 个数）。

蛐蛐国王当然知道怎么做啦 ！但是他想考考你 .....

【输入格式】

从文件 **earthworm.in**中读入数据。

第一行包含六个整数 n, m, q, u, v, t ，其中： n, m, q 的意义见【问题描述】 ； u, v, t 均 为正整数；你需要自己计算 p = u/ v （保证 0 < u < v ）；t 是输出参数，其含义将会在【输出格式】中解释。

第二行包含 n 个非负整数，为 $a_1, a_2, \ldots, a_n$ ，即初始时 n只蚯蚓的长度。

同一行中相邻的两个数之间，恰好用一个空格隔开。

保证 $1 \le n \le 10^5$ ， $0 \le m \le 7 \times 10^6$ ， $0 < u < v \le 10^9$ ， $0 \le q \le 200$ ， $1 \le t \le 71$ ，$0 \le a_i \le 10^8$ 。

【输出格式】

输出到文件 **earthworm.out**

第一行输出 $\left\lfloor \frac{m}{t} \right\rfloor$ 个整数，按时间顺序，依次输出第 t 秒，第 2t 秒，第 3t 秒， ..... 被切断蚯蚓（在被切断前）的长度。

第二行输出 $\left\lfloor \frac{n+m}{t} \right\rfloor$ 个整数，输出 m 秒后蚯蚓的长度：需要按从大到小的顺序，依次输出排名第 t，第 2t，第 3t，....的长度。

同一行中相邻的两个数之间，恰好用一个空格隔开。即使某一行没有任何数需要输出，你也应输出一个空行。

请阅读样例来更好地理解这个格式。

【样例 1 输入】

```
3 7 1 1 3 1
3 3 2
```

【样例 1 输出】

```
3 4 4 4 5 5 6
6 6 6 5 5 4 4 3 2 2
```

【样例 1 说明】

在神刀手到来前： 3 只蚯蚓的长度为 3,3,2。

1 秒后：一只长度为 3 的蚯蚓被切成了两只长度分别为 1 和 2 的蚯蚓，其余蚯蚓的长度增加了 1。最终 4 只蚯蚓的长度分别为 (1,2),4,3。括号表示这个位置刚刚有一只蚯蚓被切断。

2 秒后：一只长度为 4 的蚯蚓被切成了 1 和 3。5 只蚯蚓的长度分别为： 2,3,(1,3),4。

3 秒后：一只长度为 4 的蚯蚓被切断。 6 只蚯蚓的长度分别为： 3,4,2,4,(1,3)。

4 秒后：一只长度为 4 的蚯蚓被切断。 7 只蚯蚓的长度分别为： 4,(1,3),3,5,2,4。

5 秒后：一只长度为 5 的蚯蚓被切断。 8 只蚯蚓的长度分别为： 5,2,4,4,(1,4),3,5。

6 秒后：一只长度为 5 的蚯蚓被切断。 9 只蚯蚓的长度分别为： (1,4),3,5,5,2,5,4,6。

7 秒后：一只长度为 6 的蚯蚓被切断。 10 只蚯蚓的长度分别为： 2,5,4,6,6,3,6,5,(2,4)。

所以，7 秒内被切断的蚯蚓的长度依次为 3,4,4,4,5,5,6。7 秒后，所有蚯蚓长度从大到小排序为 6,6,6,5,5,4,4,3,2,2。

【样例 2 输入】

```
3 7 1 1 3 2
3 3 2
```

【样例 2 输出】

```
4 4 5
6 5 4 3 2
```

【样例 2 说明】

这个数据中只有 $t = 2$ 与上个数据不同。只需在每行都改为每两个数输出一个数即可。

虽然第一行最后有一个 6 没有被输出，但是第二行仍然要重新从第二个数再开始输出。

【样例 3 输入】

```
3 7 1 1 3 9
3 3 2
```

【样例 3 输出】

```
2
```

【样例 3 说明】

这个数据中只有 $t = 9$ 与上个数据不同。注意第一行没有数要输出，但也要输出一个空行。

【子任务】

测试点 1 ~ 3 满足 $m = 0$。

测试点 4 ~ 7 满足 $n, m \leq 1,000$。

测试点 8 ~ 14 满足 $q = 0$，其中测试点 8 ～9 还满足 $m \leq 10^5$ 。

测试点 15 ~ 18 满足 $m \leq 3 \times 10^5$ 。

测试点 19 ~ 20 没有特殊的约定，参见原始的数据范围。

测试点 1 ~ 12，15 ~ 16 还满足 $v \leq 2$，这意味着 u, v 的唯一可能的取值是 $u = 1, v = 2$，即 $p = 0.5$。这可能会对解决问题有特殊的帮助。

每个测试点的详细数据范围见下表。

测试点	n	m	t	a_i	v	q
1	$= 1$	$= 0$	$= 1$	$\leq 10^6$	≤ 2	$= 0$
2	$= 10^3$					
3	$= 10^5$					
4	$= 1$	$= 10^3$				≤ 200
5	$= 10^3$					
6	$= 1$					
7	$= 10^3$					
8	$= 5 \times 10^4$	$= 5 \times 10^4$				$= 0$
9	$= 10^5$	$= 10^5$	$= 2$			
10		$= 2 \times 10^6$	$= 21$			
11		$= 2.5 \times 10^6$	$= 26$			
12		$= 3.5 \times 10^6$	$= 36$	$\leq 10^7$		
13		$= 5 \times 10^6$	$= 51$		$\leq 10^9$	
14		$= 7 \times 10^6$	$= 71$			
15	$= 5 \times 10^4$	$= 5 \times 10^6$	$= 1$	$\leq 10^8$	≤ 2	≤ 200
16		$= 1.5 \times 10^6$	$= 2$			
17	$= 10^5$	$= 10^5$	$= 3$		$\leq 10^9$	
18		$= 3 \times 10^5$	$= 4$			
19		$= 3.5 \times 10^6$	$= 36$			
20		$= 7 \times 10^6$	$= 71$			

愤怒的小鸟（ angrybirds)

【问题描述】

Kiana 最近沉迷于一款神奇的游戏无法自拔。

简单来说，这款游戏是在一个平面上进行的。

有一架弹弓位于 (0, 0)处，每次 Kiana 可以用它向第一象限发射一只红色的小鸟，小鸟们的飞行轨迹均为形如 $y = ax^2 + bx$的曲线，其中 a, b 是 Kiana 指定的参数，且必须满足 $a < 0$。

当小鸟落回地面（即 x 轴）时，它就会瞬间消失。

在游戏的某个关卡里，平面的第一象限中有 n 只绿色的小猪，其中第 i 只小猪所在的坐标为 (xi, yi)。

如果某只小鸟的飞行轨迹经过了 (xi, yi)，那么第 i 只小猪就会被消灭掉，同时小鸟将会沿着原先的轨迹继续飞行；

如果一只小鸟的飞行轨迹没有经过 (xi, yi)，那么这只小鸟飞行的全过程就不会对第 i 只小猪产生任何影响。

例如，若两只小猪分别位于 (1, 3)和 (3, 3)，Kiana 可以选择发射一只飞行轨迹为 $y = -x^2 + 4x$的小鸟，这样两只小猪就会被这只小鸟一起消灭。

而这个游戏的目的，就是通过发射小鸟消灭所有的小猪。

这款神奇游戏的每个关卡对 Kiana 来说都很难，所以 Kiana 还输入了一些神秘的指令，使得自己能更轻松地完成这个游戏。这些指令将在【输入格式】中详述。

假设这款游戏一共有 T 个关卡，现在 Kiana 想知道，对于每一个关卡，至少需要发射多少只小鸟才能消灭所有的小猪。由于她不会算，所以希望由你告诉她。

【输入格式】

从文件 **angrybirds.in**中读入数据。

第一行包含一个正整数 T，表示游戏的关卡总数。

下面依次输入这 T 个关卡的信息。每个关卡第一行包含两个非负整数 n，m，分别表示该关卡中的小猪数量和 Kiana 输入的神秘指令类型。接下来的 n 行中，第 i 行包含两个正实数 x_i，y_i，表示第 i 只小猪坐标为 (x_i, y_i)。数据保证同一个关卡中不存在两只坐标完全相同的小猪。

如果 m = 0，表示 Kiana 输入了一个没有任何作用的指令。

如果 m = 1，则这个关卡将会满足：至多用「 n/3+1?只小鸟即可消灭所有小猪。

如果 m = 2，则这个关卡将会满足：一定存在一种最优解，其中有一只小鸟消灭了至少 Ln/3J 只小猪。

保证 $1 \le n \le 18$，$0 \le m \le 2$，$0 < x_i, y_i < 10$，输入中的实数均保留到小数点后两位。

上文中，符号 「c? 和 LcJ 分别表示对 c 向上取整和向下取整，例如： 「2.1? = 「2.9? =「3.0? = L3.0J = L3.1J = L3.9J = 3 。

【输出格式】

输出到文件 **angrybirds.out**中。

对每个关卡依次输出一行答案。

输出的每一行包含一个正整数，表示相应的关卡中，消灭所有小猪最少需要的小鸟数量。

【样例 1 输入】

```
2
2 0
1.00 3.00
3.00 3.00
5 2
1.00 5.00
2.00 8.00
3.00 9.00
4.00 8.00
5.00 5.00
```

【样例 1 输出】

```
1
1
```

【样例 1 说明】

这组数据中一共有两个关卡。

第一个关卡与【问题描述】中的情形相同， 2 只小猪分别位于 (1.00,3.00)和(3.00,3.00)，只需发射一只飞行轨迹为 $y = -x^2 + 4x$ 的小鸟即可消灭它们。

第二个关卡中有 5 只小猪，但经过观察我们可以发现它们的坐标都在抛物线 $y = -x^2 + 6x$ 上，故 Kiana 只需要发射一只小鸟即可消灭所有小猪。

【样例 2 输入】

```
3
2 0
1.41 2.00
1.73 3.00
3 0
1.11 1.41
2.34 1.79
2.98 1.49
5 0
2.72 2.72
2.72 3.14
3.14 2.72
3.14 3.14
5.00 5.00
```

【样例 2 输出】

```
2
2
3
```

【样例 3 输入】

```
1
10 0
7.16 6.28
2.02 0.38
8.33 7.78
7.68 2.09
7.46 7.86
5.77 7.44
8.24 6.72
4.42 5.11
5.42 7.79
8.15 4.99
```

【样例 3 输出】

```
6
```

【子任务】

数据的一些特殊规定如下表：

测试点编号	n	m	T
1	≤2	= 0	≤10
2			≤30
3	≤3		≤10
4			≤30
5	≤4		≤10
6			≤30
7	≤5		≤10
8	≤6		
9	≤7		
10	≤8		
11	≤9		≤30
12	≤10		
13	≤12	= 1	
14		= 2	
15	≤15	= 0	≤15
16		= 1	
17		= 2	
18	≤18	= 0	≤5
19		= 1	
20		= 2	