

NOI2012 北京队选拔赛

竞赛时间：2012 年 6 月 3 日上午 8:00–13:00

题目名称	结界	符卡	晚宴
目录	boundary	card	dinner
可执行文件名	boundary	card	dinner
输入文件名	boundary.in	card.in	dinner.in
输出文件名	boundary.out	card.out	dinner.out
每个测试点时限	1 秒	2 秒	3 秒
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	有	无	无
题目类型	传统	传统	传统

提交源程序须加后缀

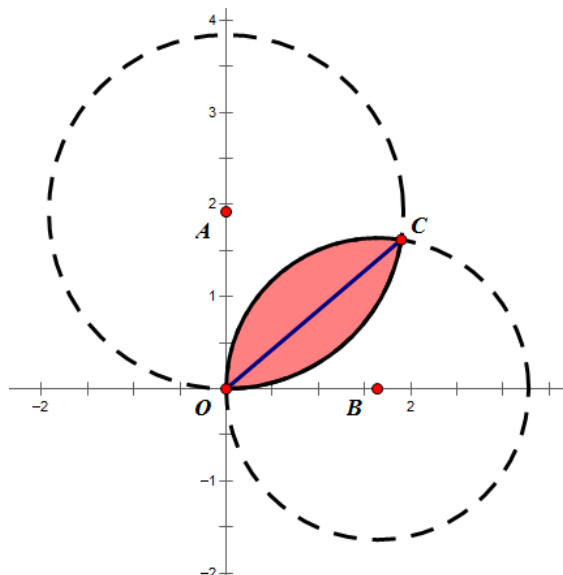
对于 Pascal 语言	boundary.pas	card.pas	dinner.pas
对于 C 语言	boundary.c	card.c	dinner.c
对于 C++ 语言	boundary.cpp	card.cpp	dinner.cpp

- 注意：1. 最终测试时，所有编译命令均不打开任何优化开关
2. 每道题目的空间限制为：314159 kb，代码限制为 20kb。
3. 测评软件为 Cena0.8.2。

结界

【问题描述】

魔法少女小薷准备去参加世界魔法大会，此时，她正在研究所需的交通工具。如果你是一位魔法爱好者，你一定见过如下形状的“传送门”（红色阴影部分）：



为什么它的形状酷似“眼睛”呢？对，看了上面的那张已经添加了辅助线的图片，你大致了解这是为什么了：因为它是两个圆形结界的重叠部分。

今天，我们不再深究魔法内部的机理，而转向应用。在描述你的任务之前，我们先明确“传送门”的相关概念：

设圆 A、圆 B 分别是以 A、B 为圆心的两圆，我们称两圆（均包含内部）的交集为“传送门”，当且仅当满足以下条件：

1. 圆 A 和圆 B 有且仅有两个交点：O 和 C。
2. 圆 A 和圆 B 正交，即：角 AOB 为直角（由对称性，此时角 ACB 亦为直角）。

并且此时我们称点 O 和 C 为该“传送门”的端点，O 与 C 的欧几里德距离为该传送门的“尺寸”，圆 A 和圆 B 为该“传送门”的两个“圆形结界”。

你的任务其实很简单：

给定平面上 N 个点，代表障碍，你需要放置一个“尺寸”尽可能大的“传送门”，需要满足：

1. 该“传送门”的一个端点在原点（即：坐标为(0, 0)的点）
2. 该“传送门”的任何一个圆形结界内部都不包含障碍点。

你只需输出这个最大尺寸即可。

你的答案将依照精度给分，具体细节见“评分标准”。

【输入文件】

输入文件为 *boundary.in*。

第一行包含一个正整数 N ，表示障碍点的数目。

接下来的 N 行，每行包含两个整数： x_i 、 y_i ，表示障碍点的坐标。

【输出文件】

输出文件为 *boundary.out*。

输出一个小数，表示“传送门”的最大尺寸。

【输入样例 1】

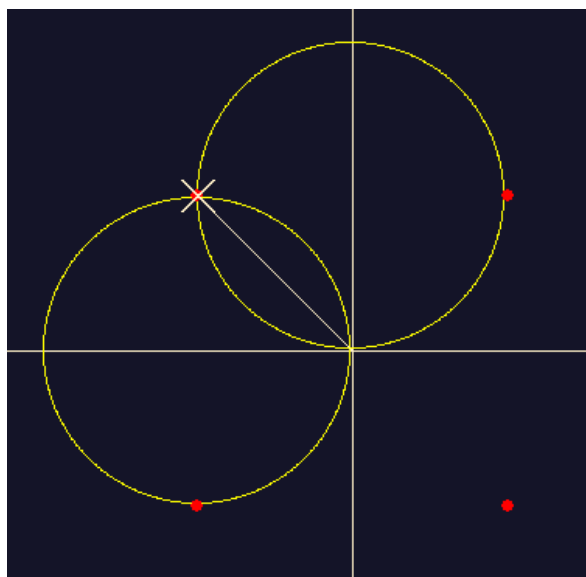
```
4
100 100
-100 100
-100 -100
100 -100
```

【输出样例 1】

```
141.42135623730950488016
```

【样例 1 解释】

障碍物的分布以及一组最优解如下图所示：



【输入样例 2】

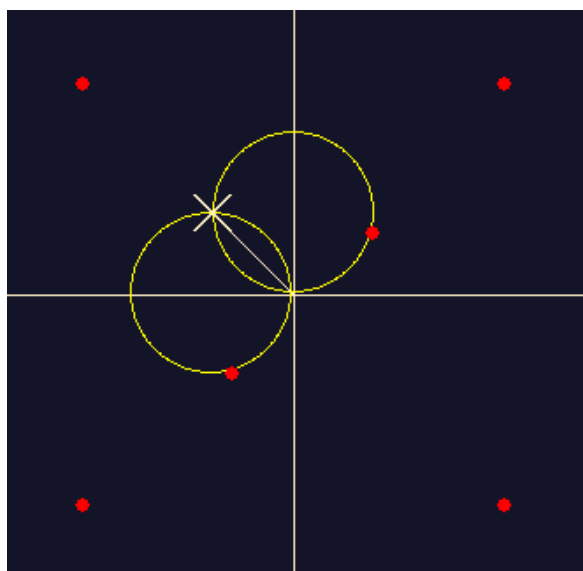
```
6
45 35
-35 -45
120 120
-120 120
-120 -120
120 -120
```

【输出样例 2】

```
65.65991539589369869436
```

【样例 2 解释】

障碍物的分布以及一组最优解如下图所示：



【数据规模和约定】

30%的数据中 $N \leq 6$;

100%的数据中 $4 \leq N \leq 50$;

对于所有数据：每个障碍点到原点的距离 d 满足： $50 \leq d \leq 200$ 。

对于所有数据：对于四个象限中的每一个象限，均存在至少一个障碍点在该象限。

可以证明，在此条件下，答案总是存在的。

注：第一象限包含 x 、 y 坐标均为正的点；第二象限包含 x 、 y 坐标分别为负、正的点；第三象限包含 x 、 y 坐标均为负的点；第四象限包含 x 、 y 坐标分别为正、负的点。

【评分方法】

对于每一组数据独立评分：

如果你的程序没有得到输出或输出的格式错误（包括但不限于：编译错误、超时、超内存限制、运行时错误、输出的不是一个正实数），得 0 分，否则：

设标准程序输出的实数为：S，你的程序输出的实数是 P：如果 S 和 P 的整数部分不同：得 0 分，否则：

设 match 为最大的 x 满足：S 和 P 的小数点前 x 位均相同，那么你的得分为：

若 $\text{match} \geq 12$ ，得 10 分，否则：

若 $\text{match} \geq 11$ ，得 8 分，否则：

若 $\text{match} \geq 10$ ，得 6 分，否则：

若 $\text{match} \geq 8$ ，得 5 分，否则：

若 $\text{match} \geq 6$ ，得 4 分，否则：

若 $\text{match} \geq 4$ ，得 3 分，否则：

若 $\text{match} \geq 2$ ，得 2 分，否则：

得 1 分。

例如，对于样例 1，输出“141.99999”会得到 1 分，输出“141.42”会得到 2 分，而输出“100.100”则会得 0 分。

需要注意：

输出的小数必须是形如“123.456”的格式，如果是例如“.456”、“0123.456”、“56e-3”、“123”的格式，均视为格式错误。

另外，如果你认为答案是形如“123.5”这样的实数，请输出：

“123.50000000000000”，即，补 0 使得小数点后至少有 12 位

否则，match 将不会超过小数点后的数字数目。

【提示】

1. 标准输出保证精确到小数点后 20 位，为向下截断，例如：“1.98”被近似为“1.9”而不是四舍五入到“2.0”。

2. 评分标准满足：未使用高精度整数、浮点数运算的参考程序能够获得满分，并且保证对于每一个数据： $\text{match} \geq 13$ 。当然，为了确保精度满足要求，你也可以使用高精度运算。

3. 如果你的程序所输出的精度与运行时间有很大关系，下面的信息也许对你有帮助：

测评机执行以下代码所需时间为 1.2 秒左右：

```
double ans = 0;
for(double i = 1; i <= 10000000; i++)
    ans += sin(i*i);
```

符卡

【问题描述】

在乘坐“传送门”仅仅 10^{-44} 秒之后，小薔到达了世界魔法大会的现场。

距离正式比赛还有一天时间，小薔已经认识了很多有名的参赛者，包括素有“教主”之称的小诚。每次参加世界大会，小诚都会和大家玩符卡游戏，这已经成为了惯例。而另一个大家都早已熟知的事情是：小诚的运气特别差，总是拿到很差的符卡。小薔打算从符卡游戏中最基本的操作：“切洗卡组”来展开研究。

一副卡组的状态可以用一个数列来表示：

$$[A_1, A_2, \dots, A_n]$$

这表示：最上面的一张为 A_1 ，下面一张是 A_2 ，……，最下面（底部）的是 A_n ，不同的符卡用不同的数字表示。

每一次切洗分两步完成：

第一步：从某一个位置将卡组分为上下两个卡组，具体地：选择整数 $0 \leq x \leq n$ ，该卡组变为两个卡组（某个卡组中的卡片数量可能是 0）：

$$[A_1, A_2, \dots, A_x] \quad [A_{x+1}, A_{x+2}, \dots, A_n]$$

第二步：将两个卡组 A 、 B 合并为一个卡组 C ，具体地：

while(A 、 B 中至少有一组非空)：

 选择一个非空的卡组，将底部的卡片拿出，放到 C 的底部。

（你可能已经发现：这个操作其实就是归并排序中的“归并”操作。）

为了确认切洗卡组是否能真正保证良好的随机性，小薔决定先解决这样一个问题：

给定卡组的初始状态 A 、目标状态 B ，以及切洗次数 K ，计算：进行 n 次切洗，卡组由状态 A 最终变为状态 B 的不同的方法数 s 。

具体地，我们规定：

1. 两个方法视为不同的方法，当且仅当存在 k ：两种方法的第 k 次切洗的方法不同。
2. 两个切洗方法不同，当且仅当第一步或第二步中至少有一个不同。
3. 第一步不同，当且仅当选择的整数 x 不同。
4. 第二步不同，当且仅当某一次选择的卡组不同。

为了方便，你只需要输出 s 模 314159 的值即可。

314159 和“圆”有一些关系，你可能会用到的性质是：它是一个素数。

【输入文件】

输入文件 *card.in*:

第一行包含 2 个由空格隔开的正整数 n 、 K 。

第二行包含 n 个由空格隔开的正整数: $A_1 \sim A_n$, 表示初始状态。

第三行包含 n 个由空格隔开的正整数: $B_1 \sim B_n$, 表示目标状态。

【输出文件】

输出文件为 *card.out*。

仅一行, 包含一个整数: s 模 314159 的值。

【输入样例 1】

```
4 2
1 2 3 4
4 3 2 1
```

【输出样例 1】

```
1
```

【样例 1 解释】

唯一的一种方法是:

第一次: $[1, 2, 3, 4] \Rightarrow [1, 2] [3, 4] \Rightarrow [3, 1, 4, 2]$

第二次: $[3, 1, 4, 2] \Rightarrow [3, 1] [4, 2] \Rightarrow [4, 3, 2, 1]$

【输入样例 2】

```
8 8
8 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8
```

【输出样例 2】

```
99070
```

【样例 2 解释】

s 的值是 2^{64} , 请注意在计算过程中使用取模运算, 以防溢出。

【输入样例 3】

```
4 2
1 2 3 4
1 2 3 4
```

【输出样例 3】

35

【样例 3 解释】

对比样例 1, 我们会发现: 当 $n=4$ 且每张符卡都不同时, 通过 2 次切洗卡组, 产生的随机效果非常差:

结果为 $[1, 2, 3, 4]$ 的概率是 $[4, 3, 2, 1]$ 的 35 倍。

【数据规模和约定】

数据编号	n 不超过	K 不超过	特别限定
1	5	10	无
2			
3			
4			
5	10	10^3	A、B 均为 1~n 的排列 （例如：样例 1 和 3）
6			
7	10^3		
8			
9	10^6	10^{12}	
10			

所有数据保证:

1. n 、 K 为正整数, 且 $1 \leq A_i, B_i \leq n$ 。
2. 初始状态和目标状态所包含的符卡是相同的, 或等价地: 对于 $1\sim n$ 中的任意整数, 在 A 及 B 中出现的次数是相同的。

晚宴

【问题描述】

一转眼，就到了颁奖晚宴的时间了。

在这次大会上，很多原本陌生的人成为了好朋友。现在，组织者正在考虑如何分配餐桌：

来参加这次大会的魔法少女一共有 n 位，我们将她们编号为 $1 \sim n$ 。她们之间有对称的二元关系：“互为朋友”。

主办方希望每一桌就座的魔法少女中的任何两位，都“互为朋友”。特别地，如果某一桌仅有一个人就座，或者是空的，也认为是可行的。

小蒨发现，这可能有非常多的可行方案。但是，她不清楚餐桌的数量 X ，当她询问了 q 个人后，得到了 q 个答案，她希望对每一个可能的数量，都计算出可行方案的数目。

需要注意的是：

1. 餐桌是彼此不同的。
2. 显然，这 n 位魔法少女是彼此不同的。
3. 显然，每位魔法少女仅仅且必须就座在一个圆桌旁。
4. 对于一个餐桌，我们仅关心就座的魔法少女的集合，只要集合相同，就认为是相同的方案。

为了方便，你只需要输出 s 模 314159 的值即可。

314159 和“圆”有一些关系，你可能会用到的性质是：它是一个素数。

【输入文件】

输入文件为 *dinner.in*。

每一个输入文件包含 T 组测试数据：

每组测试数据的第一行包含两个正整数： n 、 q 。

接下来 $n-1$ 行，第 i 行包含 i 个由空格隔开的整数： $f_{i,1} \sim f_{i,i}$ 。

如果编号为 $(i+1)$ 的魔法少女与编号为 j 的魔法少女互为朋友，则 $f_{i,j}$ 是 1，否则 $f_{i,j}$ 是 0。

接下来一行，包含由空格隔开的 q 个整数： $X_1 \sim X_q$ 。

【输出文件】

输出文件为 *dinner.out*。

一行，包含 q 个由空格隔开的整数：总方案数模 314159 的值。

【输入样例 1】

```
3 3
0
0 0
1 2 3
3 3
1
1 1
1 2 3
```

【输出样例 1】

```
0 0 6
1 8 27
```

【样例 1 解释】

如果三位魔法少女都互不为朋友，那么只好每一个人都单独就座，所以当餐桌的数目少于 3，是没有可行方案的。当餐桌数目为 3 时，方案数为 $3!=6$ 。

如果三位魔法少女都互为朋友，那么每一位都可以在 x 个餐桌中随意选择，所以答案是 x^3 。

【输入样例 2】

```
4 5
0
1 1
0 0 1
117 86 374 583 174
4 5
1
0 0
0 1 1
198 493 6 448 880
```

【输出样例 2】

```
84881 28728 205395 74314 198803
165792 271756 750 90069 12151
```

【数据规模和约定】

所有数据按以下方式随机生成：

每个测试点的参数为： T 、 N 、 M 、 Q ：

这个文件一共有 T 组数据，对于每一组数据：

$n = N$ ， $q = Q$ 。

在确定二元关系所用的“下三角矩阵”中，一共有 $\frac{n(n-1)}{2}$ 项，等概率随机选取其中 M 项，置为 1，其余为 0。

$X_1 \sim X_q$ 在 1~1000 中独立地以等概率随机选取。

例如，样例 2 就是由参数 $T=2$ ， $N=4$ ， $M=3$ ， $Q=5$ 所生成的。

每个数据点的参数如下：

编号	T	N	M	Q
1	10	10	20	10
2			30	
3			40	
4		15	30	1000
5			60	
6			90	
7		20	50	
8			150	
9		25	45	
10			280	

【提示】

测评机执行以下代码所需时间为 1.5 秒左右：

```
int f(int x)
{
    return x > 0? f(x-1) + f(x-2) : 1;
}

int main()
{
    cout << f(38) << endl;
    return 0;
}
```