

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， $0 < k < 7$ ；

对于 80% 的数据， $0 < k < 10^7$ ；

对于 100% 的数据， $1 < n < 1,000,000$ ， $0 < m < n$ ， $1 \leq x \leq n$ ， $0 < k < 10^9$ 。

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： $\sum_{i=1}^n (a_i - b_i)^2$ ，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】

<code>match.in</code>	<code>match.out</code>
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】

<code>match.in</code>	<code>match.out</code>
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， $1 \leq n \leq 100,000$ ， $0 \leq \text{火柴高度} \leq 2^{31} - 1$ 。

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	3
1 2 4	-1
2 3 3	3
3 1 1	
3	
1 3	
1 4	
1 3	

【数据说明】

对于 30% 的数据， $0 < n < 1,000$ ， $0 < m < 10,000$ ， $0 < q < 1,000$ ；

对于 60% 的数据， $0 < n < 1,000$ ， $0 < m < 50,000$ ， $0 < q < 1,000$ ；

对于 100% 的数据， $0 < n < 10,000$ ， $0 < m < 50,000$ ， $0 < q < 30,000$ ， $0 \leq z \leq 100,000$ 。