

## 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  $d_i, s_i, t_i$ ，表示某租借者需要从第  $s_i$  天到第  $t_i$  天租借教室（包括第  $s_i$  天和第  $t_i$  天），每天需要租借  $d_i$  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  $d_i$  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  $s_i$  天到第  $t_i$  天中有至少一天剩余的教室数量不足  $d_i$  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】**

<code>classroom.in</code>	<code>classroom.out</code>
4 3	-1
2 5 4 3	2
2 1 3	
3 2 4	
4 2 4	

**【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  $1 \leq n, m \leq 10$ ；

对于 30% 的数据，有  $1 \leq n, m \leq 1000$ ；

对于 70% 的数据，有  $1 \leq n, m \leq 10^5$ ；

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

### 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。

接下来一行一个整数  $m$ ，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】**

<code>blockade.in</code>	<code>blockade.out</code>
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， $2 \leq n \leq 10$ ；

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

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

对于 80% 的数据， $2 \leq n \leq 10,000$ ；

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