



## 提高级

题目名称	糖豆人	考试	快速数字变换	吃饭!
题目类型	传统型	传统型	传统型	传统型
目录	guys	exam	fnt	launch
可执行文件名	guys	exam	fnt	launch
输入文件名	guys.in	exam.in	fnt.in	launch.in
输出文件名	guys.out	exam.out	fnt.out	launch.out
每个测试点时限	1000ms	1000ms	3000ms	1000ms
内存限制	256mb	512mb	512mb	512mb
子任务数目	10	10	20	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于C++语言	guys.cpp	exam.cpp	fnt.cpp	launch.cpp
---------	----------	----------	---------	------------

编译选项

对于C++语言	-O2 -lm
---------	---------

### 注意事项（请仔细阅读）

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. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
4. 程序可使用的栈空间内存限制与题目的内存限制一致。
5. 只提供Linux格式附加样例文件。

# 糖豆人 (guys)

## 【题目描述】

你最近特别沉迷于玩糖豆人。参加比赛已经无法满足你了，现在你想组织一场比赛。

你的手头一共有 $x$ 个糖豆人。你想按照以下方式让它们进行比赛。

如果 $x$ 是个偶数，那么首先进行淘汰赛：将所有糖豆人进行两两配对，每两个糖豆人进行一场比赛，输者淘汰。因此，一轮循环赛后，糖豆人数量会减半，即 $x$ 会被除以2。进行完一轮淘汰赛后，如果 $x$ 仍然为偶数，那么继续进行淘汰赛，直到 $x$ 变成奇数。

$x$ 变成奇数后，如果只剩下一支队伍，则比赛结束。否则，你进行一轮循环赛：每个糖豆人都要与其他 $x - 1$ 个糖豆人进行一场比赛，所以总共要进行 $\frac{x(x-1)}{2}$ 场循环赛。

现在，你知道了比赛总场数（比赛总场数=淘汰赛场数+循环赛场数），你想知道一个有多少个糖豆人。如果答案有多个，请按照从小到大的顺序输出所有的 $x$ 。

## 【输入格式】

第一行有一个整数 $n$ ，表示比赛场数。

## 【输出格式】

一行若干整数，按照从小到大的顺序输出所有的 $x$ 。  
如果不存在 $x$ ，则输出 $-1$ 。

## 【输入样例1】

```
1 | 3
```

## 【输出样例1】

```
1 | 3 4
```

## 【输入样例2】

```
1 | 2
```

## 【输出样例2】

```
1 | -1
```

## 【样例1解释】

如果 $x = 4$ ，会进行两轮淘汰赛：

第一轮4个糖豆人进行2场比赛，2个糖豆人淘汰，剩余2个晋级；

第二轮2个糖豆人进行1场比赛，1个糖豆人淘汰，剩余1个晋级；

只剩下一个糖豆人，无需进行循环赛。

所以一共有3场比赛。

如果 $x = 3$ ，会进行两轮淘汰赛：

$x$ 是奇数，不进行淘汰赛，只进行循环赛，一共 $\frac{3(3-1)}{2} = 3$ 场比赛。

#### 【数据范围】

对于前30%数据，保证答案是奇数；

对于前60%数据： $n \leq 10^6$ ；

对于100%的数据， $n \leq 10^{18}$ 。

# 考试 (exam)

## 题目描述

小 Z 是一个小学生，作为学生当然少不了考试。小 Z 的老师给了一本包含  $n$  张试卷的试题集，第  $i$  张试卷包含  $m_i$  道题目，题目用编号来表示，编号相同则题目相同。一张试卷不会有两道相同的题目，但相同的题目可能出现在不同的试卷中。老师说考试卷会随机从中抽取，但是由于时间有限，小 Z 不可能把整本试题集做完，他只能去随机做一些题。

具体地，小 Z 会先在  $[1, n]$  中等概率抽取一张试卷  $x$ ，再在  $[1, m_x]$  中等概率抽取一个题号，并做掉这道题。假设老师从  $[1, n]$  中随机抽取了一张试卷  $y$ ，小 Z 想知道他做的这道题在这张试卷中出现的概率是多少。

小 Z 只需要你告诉他答案模  $10^9 + 7$  意义下的结果。设答案可以表示为分数  $\frac{x}{y}$ ，你需要找到一个  $m$ ，满足  $x = m \cdot y \pmod{10^9 + 7}$ 。

## 输入格式

第一行包含一个整数  $n$ ，表示试卷的数量  
接下来  $n$  行，每行以一个数字  $m_i$  开头，表示第  $i$  张试卷的题目数量。接下来  $m_i$  个数字，表示代表题目的编号。

## 输出格式

输出一个整数表示答案

## 样例

【输入样例1】

1	2
2	1 2
3	2 2 1

【输出样例1】

1	875000007
---	-----------

【输入样例2】

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

【输出样例2】

1 | 480000004

## 约定

---

对于 20% 的测试数据,  $n \leq 100, m_i \leq 10$

对于 40% 的测试数据,  $n \leq 10^3, m_i \leq 50$

对于全部测试数据,  $n \leq 10^6, \sum m_i \leq 10^6$

题目编号范围  $\in [1, 10^6]$

# 快速数字变换 (fnt)

## 题目描述

Prof. Wang 是一位数学大师，他最近在研究快速数字变换（FNT, Fast Number Transform）。这天他将草稿纸上的一个正整数  $n$  给剪了下来，并将每一位分开重新排列，他很快就算出了重排列后所有不含前导零的数字种类数，他觉得这个问题太简单，于是他加上了一点限制：重排列出来的数字必须与给定的正整数  $m$  互质。

作为数学大师的 Prof. Wang 很快就算出来了结果，他觉得这个问题很有意思，所以他想考考你。换句话说，Prof. Wang 给你  $n, m$ ，你需要求出满足下列条件的 **正整数个数**：

- 不含前导 0
- 十进制下重排列后可以得到  $n$
- 与  $m$  互质

## 输入格式

一共一行两个正整数  $n, m$ 。分别表示数字  $n$  和数字  $m$ 。

## 输出格式

仅一行，表示正整数个数。

## 样例

#样例1

```
1 | 111 1
```

```
1 | 1
```

#样例2

```
1 | 123456 17
```

```
1 | 676
```

#样例3

```
1 | 9876543210 100
```

```
1 | 1290240
```

# 说明提示

---

对于 10% 的测试数据,  $1 \leq n \leq 10^{10}, 1 \leq m \leq 100$

对于另外 10% 的测试数据,  $1 \leq n \leq 10^{15}$ , 且  $m$  是质数

对于 40% 的测试数据,  $1 \leq n \leq 10^{15}, 1 \leq m \leq 100$

对于全部的测试数据,  $1 \leq n \leq 10^{20}, 1 \leq m \leq 100$

# 吃饭! (launch)

## 题目描述

小Y是一名高中生，他每天都要从自己上课的教学楼出发去食堂吃饭（不止一个食堂）。学校中有很多楼，楼之间由单向道路连接（因为学校的道路规划非常差，所以道路是单行线）。在一学期的开始，学校会宣布这学期每一天食堂午饭的价格。小Y必须保证自己饭卡里的钱大于等于食堂的午饭价格，才能吃顿饭。

小Y是一个很聪明的学生，他把许多个余额不同的饭卡藏在学校不同的楼里，这样他只要从教室出发，然后顺路拿到一个合适的饭卡，就可以吃到午饭，但是吃过饭后，小Y就会把饭卡不知道丢到哪里，再也找不到了。

但是小Y十分懒惰，他希望花尽可能少的时间到达食堂吃饭，不过身为一个很聪明的人，他已经提前得到了经过学校每条道路花费的时间，与自己在什么地方藏了一张余额为多少的饭卡。

换言之，我们可以认为每一栋楼在学校是一个有向图上的顶点，小Y每天的行程为一条由教室出发，找到一张足够支付饭钱的饭卡然后随意到达任一食堂的路径。在第 $i$ 天，小Y需要去一个饭卡还未被取走的顶点 $c_j$ ，保证卡内饭钱 $d_j > value_i$ ，然后前往任意一个食堂 $address_p$ 吃饭。

小Y想知道他在这个学期的 $t$ 天内怎么走才能保证吃饭总共花费的时间最短。

数据保证小Y每次都可以吃顿饭

## 输入格式

第一行2个整数 $t, n, m$ 分别表示楼的数量和学校内路的数量。

接下来 $m$ 行每行3个正整数 $u, v, l$ 表示从编号为 $u$ 的楼到编号为 $v$ 的楼存在一个花费时间为 $l$ 的单向道路。

接下来一行一个整数 $t$ ，表示本学期时间时间。

接下来一行 $t$ 个正整数代表每天需要的饭钱 $value_i$ 。

接下来一行三个整数 $s, p, q$ 分别表示教室所在楼编号，食堂数量和饭卡数量。

接下来 $p$ 行，每行两个正整数 $c_i, d_i$ 表示编号为 $c_i$ 的楼内有一张 $d_i$ 余额的饭卡。

接下来一行 $q$ 个整数，表示食堂所在的楼的编号 $address_i$ 。

## 输出格式

输出一个整数，表示花费最小总时间。

## 样例

1	6 8
2	1 2 2
3	2 3 2
4	2 4 1
5	1 3 5
6	3 4 3



7	1 4 4
8	4 5 3
9	2 6 2
10	2
11	5 6
12	1 3 2
13	3 7
14	2 6
15	4 8
16	5 6

1	10
---	----

# 约定

对于20%的数据：  $1 \leq t, n, p, q \leq 50, 1 \leq m \leq 500$

对于40%的数据：  $1 \leq t, n, p, q \leq 3000, 1 \leq m \leq 500$

对于100%的数据：  $1 \leq t, n, p, q \leq 10^5, 1 \leq m \leq 10^6$