

# Super-Zigzag-YGY Round #1

(SZY Project Version)

出题人:王梓祺

## 声明:

本次题目以 SZY 为主角。

事件均为虚构。

故事情节如有雷同纯属巧合:)

# 1、WZQ 的游戏

(game.c/.cpp/.pas, game.in, game.out)

时间限制:2s 空间限制:256M

题目满分:100pts 测试点数目:10

## 【问题描述】：

WZQ 与**神犇 SZY** 在玩一个游戏。

WZQ 在  $1 \sim n$  中任意选一个数  $x$ ，并且要求**神犇 SZY** 来猜这个数  $x$ 。

**神犇 SZY** 可以进行诸如此类的询问：“ $x$  能否被  $y$  整除”，而 WZQ 只会回答 yes 或者 no。

**神犇 SZY** 想要知道他至少需要询问多少次才能确定数  $x$ ，并且想要知道他询问的数都是什么。

(HINT:我们称数  $x$  是确定的当且仅当**神犇 SZY** 可以区分  $1 \sim n$  的任意两个数。并且询问的结果会在**神犇 SZY** 问完所有的问题之后统一返回。)

**神犇 SZY** 当然会这个问题了，但是他的键盘不知道被什么人更改了键子，在他键盘复原之前他是没办法编程解决这个问题的，所以他找到你来帮助他。

## 【输入格式】：

第一行包括一个正整数  $n$ 。

## 【输出格式】：

第一行包括一个正整数，代表**神犇 SZY** 最少需要询问几次。

接下来一行输出**神犇 SZY** 需要询问的数，按从小到大顺序输出。

## 【样例数据】：

game1.in	game1.out
4	3 2 3 4

## 【数据范围】：

对于 30% 的数据满足  $n \leq 10^3$ 。

对于 100% 的数据满足  $n \leq 10^7$ 。

## 【样例解释】：

如果  $x$  不可以被 2,3,4 整除，那么  $x$  是 1。

如果  $x$  被 4 整除，那么  $x$  是 4。

如果  $x$  被 3 整除，那么  $x$  是 3。

其他情况  $x$  是 2，可以发现没有询问两次的可行解。

## 2、SZY 的城市

(city.c/.cpp/.pas, city.in, city.out)

时间限制:1s 空间限制:256M

题目满分:100pts 测试点数目:10

【问题描述】：

**神犇 SZY** 在一款内测的模拟游戏中拥有  $n$  个城市，这些城市之间一共有  $n-1$  条双向的道路，任意两个城市之间都可互相到达。城市是从  $1\sim n$  标号的，并且拥有一个首都，首都可以设置在任意一个城市。

在这个游戏之初，所有的道路都是坏的，但是**神犇 SZY** 想要修复某些道路使他的子民开心，我们假设第  $x$  号城市中的子民是开心的当且仅当从首都出发到第  $x$  号城市的路上至多只有一条坏的道路。

**神犇 SZY** 想要知道对于任意一个首都，他有多少种不同的修复道路的方式使得所有城市中的子民都能够开心，

**神犇 SZY** 当然会这个问题了，但**神犇 SZY** 深深地被这款游戏吸引，根本没有时间切屏出去解这个问题，所以他找到了你，让你来帮他解决这个问题。**神犇 SZY** 答应你，如果你能够帮他解决这个问题，那么他会把这个游戏的邀请码发给你。

【输入格式】：

第一行一个正整数  $n$ ，代表一共有  $n$  个城市。

第二行一共  $n-1$  个正整数  $p_2, p_3, \dots, p_n$ ，其中  $p_i$  代表有一条道路连接  $i$  与  $p_i$ 。

【输出格式】：

第一行  $n$  个正整数分别代表以第  $i$  个城市作为首都的道路修建的方案数。

由于答案可能很大，所以要求方案数  $\text{mod } 1000000007$ 。

【样例数据】：

city.in	city.out
3	4 3 3
1 1	
city.in	city.out
5	5 8 9 8 5
1 2 3 4	

【数据范围】：

对于 10% 的数据满足  $n \leq 10^2$

对于 50% 的数据满足  $n \leq 10^3$

对于 100% 的数据满足  $n \leq 2 \cdot 10^5$

### 3、SZY 的旅行

(travel.c/.cpp/.pas, travel.in, travel.out)

时间限制:4s 空间限制:256M

题目满分:100pts 测试点数目:10

【问题描述】：

**神犇 SZY** 买彩票中了一大笔钱，他决定用剩下的钱去神♂秘国度旅游。

这个神♂秘国度恰好有从 1 到  $n$  标号的  $n$  个城市，并且每个城市都有一个飞机场。在这个神♂秘国度中，有且仅有一个航空公司，并且这个航空公司只提供  $m$  条航线。但是这个航空公司的制度十分的畸形，简单的说，对于一个航线  $i$ ，你只有先前已经乘坐过  $d_i$  次航班，才能通过这趟航班从  $a_i$  飞到  $b_i$ ，并且任意一个航班不能够从  $b_i$  飞到  $a_i$ 。

**神犇 SZY** 在来到这个神♂秘国度的 1 号城市后听到别人说第  $n$  号城市的景色最为美好，所以他极其想要从他所在的 1 号城市去第  $n$  号城市，但是为了节省时间与金钱，**神犇 SZY** 想要乘坐最少次数的航班到达第  $n$  号城市。

**神犇 SZY** 当然会这个问题了，但不幸的是，**神犇 SZY** 这趟旅行并没有携带他的笔记本电脑，所以他在调查了所有航线后给你打电话进行求助，**神犇 SZY** 答应你，如果你能够帮他解决这个问题，那么他会给你带回去一些纪念品。

【输入格式】：

第一行包含两个正整数  $n, m$ 。

接下来  $m$  行，每行包含三个整数  $a_i, b_i, d_i (1 \leq a_i, b_i \leq n)$ 。

输入数据可能有重边或自环。

【输出格式】：

输出**神犇 SZY** 最少需要乘坐多少趟航班。

数据保证可以从 1 号城市到达  $n$  号城市。

【样例数据】：

travel.in	travel.out
3 2 1 2 0 2 3 1	2
travel.in	travel.out
3 3 2 1 0 2 3 6 1 2 0	8

【数据范围】：

测试点 1~3  $2 \leq n \leq 150, 1 \leq m \leq 150, 0 \leq d_i \leq 10^5$

测试点 4~7  $2 \leq n \leq 100, 1 \leq m \leq 100, 0 \leq d_i \leq 2 \cdot 10^9$

测试点 8~10  $n=150, m=150, 0 \leq d_i \leq 2 \cdot 10^9$

# Good Luck to You:)

