**NOIP** 模拟赛

比赛时间：**2018** 年 **2** 月 **8** 日 **9:00-12:00**

|  |  |  |  |
| --- | --- | --- | --- |
| 题目名称 | Qizy 的指数 | Qizy 的函数 | Qizy 的∑ |
|  |  |  |  |
| 输入文件名 | exp.in | function.in | sum.in |
| 输出文件名 | exp.out | function.out | sum.out |
| 每个测试点时限 | 1 秒 | 2 秒 | 1 秒 |
| 内存限制 | 256MB | 256MB | 256MB |
| 测试点数目 | 10 | 10 | 10 |
| 每个测试点分值 | 10 | 10 | 10 |
| 是否有部分分 | 否 | 否 | 否 |
| 题目类型 | 传统型 | 传统型 | 传统型 |
| 是否有样例文件 | 是 | 是 | 是 |
| 是否有附加文件 | 否 | 否 | 否 |

提交源程序须加后缀

|  |  |  |  |
| --- | --- | --- | --- |
| 对于 C++ 语言 | exp.cpp | function.cpp | sum.cpp |
| 对于 C 语言 | exp.c | function.c | sum.c |
| 对于 Pascal 语言 | exp.pas | function.pas | sum.pas |

编译开关

|  |  |  |  |
| --- | --- | --- | --- |
| 对于 C++ 语言 | -lm | -lm | -lm |
| 对于 C 语言 | -lm | -lm | -lm |
| 对于 Pascal 语言 | N/A | N/A | N/A |

**Qizy** 的指数

题目描述

# 求 *an*! (mod *p*) 输入描述

输入文件为 ***exp.in***

第一行，一个整数 *T*，表述数据组数

接下来 *T* 行，每行三个正整数，表示 *ai,ni,pi* 输出描述

输出文件为 ***exp.out***

*T* 行，每行一个整数表示第 *i* 组数据的答案样例 1 输入

2

1. 1 2
2. 3 2

# 样例 1 输出

0

1

样例 2 输入输出见选手目录下的 ***exp/exp2.in***与 ***exp/exp2.ans***

# 数据范围及约定

对于前 30% 的数据：1 ≤ *ai,ni,pi* ≤ 10 对于 90% 的数据：1 ≤ *ai,ni,pi* ≤ 105 对于 100% 的数据：1 ≤ *ai,ni,pi* ≤ 107;*T* ≤ 10

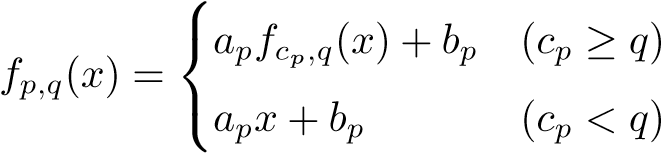
**Qizy** 的函数

# 题目描述

Qizy 最近很闲，于是补起了文化课。他一翻开数学教材就被各种初中没见过的函数给吸引住了，于是 Qizy 决定自己也要造出优美的函数。

但他冥思苦想了很久很久，发现自己并没有妙妙的想法，因为此时他能想起来的函数只有一次函数了。但他发现：一个一次函数虽然简单。但如果是一些一次函数层层嵌套，其函数图像也能变得很优雅。

于是他先找了三个数列 {*ai*}*,*{*bi*}*,*{*ci*}，然后定义了以下函数：



我们闭上眼睛感受一下，能想象出这个函数的图像一定非常优雅。但荒废文化课已久的 Qizy 已经画不出其具体图像了。于是他降低了要求：只要单点求值就好。

但 Qizy 他冥思苦想了很久很久，发现自己连单点求值都不会了。于是只能请你帮忙了。不过这个请求不是无偿的：只要你能帮 Qizy 解决这个问题，

Qizy 就会在这次比赛中奖励你 100 分！

具体来讲：给定三个长度为 *n* 的序列 {*ai*}*,*{*bi*}*,*{*ci*}。然后再定义函数 *fp,q*，其具体定义前文已经给出。最后再给定你 *m* 个询问，第 *i* 个询问会给定三个数 *pi,qi,xi*，你需要输出所有 *fpi,qi*(*xi*) mod 1000000007 异或起来的结果。

# 输入描述

输入文件为 ***function.in***

第一行，两个整数分别表示 *n,m*

接下来 *n* 行，每行三个数，第 *i* + 1 行的三个数分别表示 *ai,bi,ci*

接下来 *m* 行，每行三个数，第 *i* + *n* + 1 行的三个数分别表示第 *i* 个询问的 *pi,qi,xi* 输出描述输出文件为 ***function.out***

设 *ansi* = *fpi,qi*(*xi*) mod 1000000007

一行，一个整数，即所有 *ansi* 异或起来的结果

# 样例 1 输入

5 6

9 2 0

1. 0 1
2. 2 2
3. 1 1

4 3 2

1 1 2

4 4 5

1. 1 3
2. 1 7

3 1 9

3 2 11

样例 1 输出

2751

样例 1 解释

每个询问的结果依次为：20,41,233,1563,3488,464

样例 2 输入输出

见选手目录下的 ***function/function2.in***与 ***function/function2.ans***

# 数据范围及约定

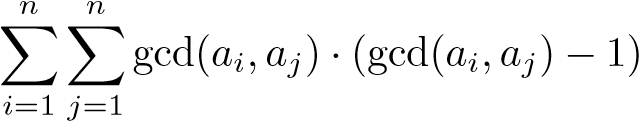
对于前 20% 的数据：*n,m* ≤ 3000 对于前 40% 的数据：*n,m* ≤ 100000 另有 20% 的数据：*ci* = *i* − 1

对于 100% 的数据：*n,m* ≤ 1000000;1 ≤ *pi,qi* ≤ *n*;0 ≤ *ai,bi,xi* ≤ 109;0 ≤ *ci < i*

**Qizy** 的∑

# 题目描述

给定序列 {*ai*}，求



答案对 1000000007(109 + 7) 取模

# 输入描述

输入文件为 ***sum.in*** 第一行，一个整数，表示 *n*

第二行，*n* 个整数，表示 *a*1*,a*2*,...,an*

# 输出描述

输出文件为 ***sum.out***

一行，一个整数，表示答案答案对 1000000007 取模

# 样例 1 输入

5

1 3 4 2 4

样例 1 输出

64

样例 2 输入输出

见选手目录下的 ***sum/sum2.in***与 ***sum/sum2.ans***

# 数据范围及约定

对于前 20% 的数据：1 ≤ *n* ≤ 103 对于前 40% 的数据：1 ≤ *n* ≤ 105 对于 100% 的数据：1 ≤ *n* ≤ 5000000*,*1 ≤ *ai* ≤ 500000