

# NOI2016 模拟题

Chloe\_fan 613

## 题目概览:

题目名称		Zbox loves ants	Zbox loves stack	Zbox loves meizi
可执行文件名		ants	stack	meizi
输入文件名		ants.in	stack.in	meizi.in
输出文件名		ants.out	stack.out	meizi.out
时间限制		1s	4s	60s
空间限制		1024MiB	1024MiB	1024MiB
测试点数目		10	10	1
测试点分值		10	10	100
比较方式		全文比较	全文比较	Special Judge
题目类型		传统	传统	<b>交互</b>
编译命令	c++	g++ ants.cpp	g++ stack.cpp	<b>g++ grader.cpp meizi.cpp</b>
	pascal	fpc ants.pas	fpc stack.pas	<b>fpc grader.pas</b>

## 注意事项:

- 1、文件名(程序名和输入输出文件名)必须使用小写。
- 2、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、评测时采用的机器配置为: cpu 酷睿™ i7 ,内存 4G,上述时限以此配置为准。
- 4.保证标准程序在 cpu 酷睿™ i5 配置下用时不超过时限的一半。

# Zbox loves ants

## 题目描述

从小热爱生物的 Zbox 开始观察蚂蚁了.她有一根长度为  $m$  的绳子,在最初的时刻,上面分布着  $n$  只蚂蚁,她发现,每一只蚂蚁在最初都可能选择任意一个方向爬,爬行的速度始终为 1,当有两只蚂蚁相遇时,它们会各自调转方向,以原有的速度继续爬行.Zbox 知道,每只蚂蚁随意选择一种方向一共有  $2^n$  种方案,每一种方案都存在一个时间使得最后一只蚂蚁恰好爬离绳子.现在她想问问你,每一个这样的时间对应的方案数分别是多少.

## 输入描述

第一行一个整数  $n$ ,表示蚂蚁数量.

第二行输入为 4 个 32 位无符号整数  $x, a_0, b_0, c_0$  用于生成(伪)随机数.

蚂蚁的初始位置以及绳子的长度的生成方式,我们将以文件形式下发并且有使用该数据生成方式的有正确性保证的代码在选手目录下(包含 `cpp` 和 `pascal` 两个版本).

代码中  $m$  表示绳子的长度,数组  $A$  最终将储存编号为 1 到  $n$  的蚂蚁的初始位置到绳子左端点的距离,且保证  $A$  数组中的数单调递增.

数据保证所有的蚂蚁到绳子两端的距离(共  $2n$  个数)都不相同.

## 输出描述

一个整数  $Ans$ ,其中  $Ans = \sum (i * Cnt[i])$ ,  $Cnt[i]$  指的是在  $i$  时刻所有蚂蚁能恰好爬离绳子的方案数.答案对  $10^9+7$  取模.

## 样例输入 1

2

124780488 197083903 101694583 1000000007

## 样例输出 1

981038651

## 样例输入 2

10

822573732 69646737 10463453 1000000009

## 样例输出 2

140196847

## 样例输入 3

100000

161625489 395725869 274204143 1000000007

## 样例输出 3

806646457

## 数据规模

对于 20%的数据  $n \leq 10$

对于 40%的数据  $n \leq 10^3$

对于 60%的数据  $n \leq 10^5$

对于 100%的数据  $2 \leq n \leq 5 \cdot 10^6; 0 \leq A_i, m < 2^{63}; 1 \leq x, a_0, b_0, c_0 < 2^{31}$

# Zbox loves stack

## 题目描述

从小热爱计算机的 Zbox 开始玩栈了.Zbox 有  $n$  个栈,他命令你支持如下操作:

- \*.把第  $l$  个栈到第  $r$  个栈都压入一个元素  $x$
- \*.把第  $l$  个栈到第  $r$  个栈都弹出栈顶(若栈为空则无视)
- \*.询问第  $s$  个栈的第  $k$  个元素是多少(栈顶为第一个元素)

## 输入描述

第一行 2 个数  $n, q$

接下来  $q$  行,每行第一个数为  $t$

若  $t$  为 0,输入  $l, r, x$ ,把第  $l$  个栈到第  $r$  个栈都压入一个元素  $x$

若  $t$  为 1,输入  $l, r$ ,把第  $l$  个栈到第  $r$  个栈都弹出栈顶

若  $t$  为 2,输入  $s, k$ ,询问第  $s$  个栈的第  $k$  个元素是多少,若栈中没有  $k$  个元素则输出 "Error"

## 输出描述

对于每一个  $t=2$  的操作,输出一行 "Error" 或一个数表示答案

## 样例输入

```
10 12
2 6 1
0 7 8 8143
0 6 9 13209
2 7 1
2 7 2
1 5 6
1 2 7
2 5 2
1 1 8
0 2 5 22920
1 5 7
2 7 1
```

## 样例输出

Error

13209

8143

Error

Error

## 数据规模

对于 10%的数据,  $n \leq 5000, q \leq 5000$

对于 30%的数据,  $n \leq 100000, q \leq 30000$

对于另外 10%的数据, 保证没有  $t=1$  的操作

对于另外 10%的数据, 所有  $t=0$  的操作满足  $l=r$

对于另外 10%的数据, 所有  $t=1$  的操作满足  $l=r$

对于另外 10%的数据, 所有  $t=2$  的操作满足  $k=1$

对于另外 10%的数据, 保证不会输出 "Error"

对于 100%的数据,  $n \leq 1000000, q \leq 100000$ , 所有输入的数在  $[0, 2^{31}-1]$  之间

# Zbox loves meizi

## 题目描述

从小热爱妹子的 Zbox 开始玩妹子了.Zbox 是校草,必然有大量妹子来追 Zbox.Zbox 发现有  $2000 \times 2000$  个妹子正在向他靠近,排成一个  $2000 \times 2000$  的矩阵.每个妹子有一个魅力值,在茫茫人海中不同的妹子的魅力值是不同的.Zbox 认为,一个妹子是他看得上的妹子,当且仅当她的魅力值高于他周围所有妹子的任何一个的魅力值,比如图中红色格子的妹子是 Zbox 看得上的.

1	4	2	5
8	3	19	7
10	15	18	17
12	16	9	22
14	11	10	21

Zbox 患有密集恐惧症,他想尽快找到一个他看得上的妹子以便于哔~

## 实现细节

**如果你使用的是 cpp :**

你需要完成一个函数 `FINDMXPOINT()` 来返回一个 Zbox 看得上的妹子的位置,格式为:

```
pair<int,int> FINDMXPOINT();
```

其中,first 表示行数.second 表示列数

你需要在你的程序中包含头文件 "MXPOINT.h"

你在函数中可以调用以下函数:

```
int ASK(int x,int y);
```

`ASK(x,y)` 的返回值为第  $x$  行第  $y$  列的妹子的魅力值.

如果  $x < 1$  或  $x > 2000$  或  $y < 1$  或  $y > 2000$  函数将返回 -1.保证所有妹子的魅力值在 1 到

$2 \times 10^9$  之间.

有一个样例程序 sample.cpp 在选手目录下.

程序 grader.cpp 在选手目录下.

头文件 MXPOINT.h 在选手目录下.

样例数据 sampletest1~3.in 在选手目录下.

你可以用如下的命令编译程序:

```
g++ -o code grader.cpp meizi.cpp
```

用如下的命令运行程序:

```
code
```

这个程序将返回你 3 个样例数据的得分

你提交的文件为 meizi.cpp

### **如果你使用的是 pascal :**

你需要完成一个库,包括一个函数 FINDMXPOINT()来返回一个 Zbox 看得上的妹子的位置,格式为:

```
function FINDMXPOINT:longint;
```

假设你找到的点在第  $i$  行第  $j$  列则返回  $i+j \times 10000$

你需要在你的程序中包含库"graderlib"

你在函数中可以调用以下函数:

```
Function ASK(x:longint;y:longint);
```

ASK(x,y)的返回值为第  $x$  行第  $y$  列的妹子的魅力值.

如果  $x < 1$  或  $x > 2000$  或  $y < 1$  或  $y > 2000$  函数将返回-1.保证所有妹子的魅力值在 1 到  $2 \times 10^9$  之间.

有一个样例程序 sample.pas 在选手目录下.

程序 grader.pas 在选手目录下.

库 graderlib 在选手目录下.

样例数据 sampletest1~3.in 在选手目录下.

你可以用如下的命令编译程序:

```
fpc grader.pas
```

用如下的命令运行程序:

```
grader
```

这个程序将返回你 3 个样例数据是否正确

你提交的文件为 meizi.pas

## 注意事项

- \*.请勿在你的程序中输出任何信息,这会导致不可预料的错误
- \*.在评测时 **FINDMXPOINT()** 会被调用多次,意味着你可能需要手动清空数组
  - 重要的话说第二遍: 你可能需要手动清空数组**
  - 重要的话说第三遍: 你可能需要手动清空数组**
- \*.请不要卡评测,请用 RE 或 WA 来代替 TLE,谢谢
- \*.测试数据有 20 组,最终得分为每组数据的最低得分,请尽量优化你的算法
- \*.测试数据有 20 组,程序运行时间为 20 组数据的总时间,交互库大约会使用 20s 左右的时间,这意味着你需要尽量在 1s 内回答一个数据

## 评分方式

- 如果对于任意一组数据你的程序能在 6050 次询问及以内返回正确答案得 100 分
- 如果对于任意一组数据你的程序能在 16050 次询问及以内返回正确答案得 70 分
- 如果对于任意一组数据你的程序能在 48050 次询问及以内返回正确答案得 50 分
- 如果对于任意一组数据你的程序能在 180050 次询问及以内返回正确答案得 30 分
- 如果对于任意一组数据你的程序能在 4000050 次询问及以内返回正确答案得 10 分
- 如果对于任意一组数据你的程序返回错误答案得 0 分
- 如果你的程序运行时发生错误得 0 分
- 如果你的程序运行时间超出限制得 -10 分(骗你的)