

# NOIP 模拟赛

day3

题目名称	排序	上帝造题的七分钟	游戏
题目类型	传统型	传统型	传统型
可执行文件名	sort.exe	god.exe	game.exe
输入文件名	sort.in	god.in	game.in
输出文件名	sort.out	god.out	game.out
每个测试点时限	5.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
测试点/包数目	10	10	10
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	sort.cpp	god.cpp	game.cpp
对于 C 语言	sort.c	god.c	game.c
对于 Pascal 语言	sort.pas	god.pas	game.pas

编译选项

对于 C++ 语言	-O2 -std=c++14 -Wl,--stack=536870912
对于 C 语言	-O2 -std=c14 -Wl,--stack=536870912
对于 Pascal 语言	-O2

## 排序 (sort)

### 【题目描述】

在 2016 年，佳媛姐姐喜欢上了数字序列。因而他经常研究关于序列的一些奇奇怪怪的问题，现在他在研究一个难题，需要你来帮助他。这个难题是这样子的：

给出一个 1 到  $n$  的全排列，现在对这个全排列序列进行  $m$  次局部排序，排序分为两种：

1:(0,l,r) 表示将区间  $[l,r]$  的数字升序排序

2:(1,l,r) 表示将区间  $[l,r]$  的数字降序排序

最后询问第  $q$  位置上的数字。

### 【输入格式】

从文件 *sort.in* 中读入数据。

输入数据的第一行为两个整数  $n$  和  $m$ 。 $n$  表示序列的长度， $m$  表示局部排序的次数。

第二行为  $n$  个整数，表示 1 到  $n$  的一个全排列。

接下来输入  $m$  行，每一行有三个整数  $op, l, r$ ， $op$  为 0 代表升序排序， $op$  为 1 代表降序排序， $l, r$  表示排序的区间。

最后输入一个整数  $q$ ， $q$  表示排序完之后询问的位置。

### 【输出格式】

输出到文件 *sort.out* 中。

输出数据仅有一行，一个整数，表示按照顺序将全部的部分排序结束后第  $q$  位置上的数字。

### 【样例 1 输入】

```
6 3
1 6 2 5 3 4
0 1 4
1 3 6
0 2 4
3
```

### 【样例 1 输出】

```
5
```

**【子任务】**

对于 30% 数据,  $n, m \leq 100$ .

对于 50% 数据,  $n \leq 2000, m \leq 10000$ .

对于 70% 数据,  $n, m \leq 10000$ .

对于 100% 数据,  $1 \leq n, m \leq 100000, 1 \leq q \leq n$ .

## 上帝造题的七分钟 (god)

### 【题目描述】

XLk 觉得《上帝造题的七分钟》不太过瘾，于是有了第二部。

" 第一分钟，X 说，要有数列，于是便给定了一个正整数数列。

第二分钟，L 说，要能修改，于是便有了对一段数中每个数都开平方 (下取整) 的操作。

第三分钟，k 说，要能查询，于是便有了求一段数的和的操作。

第四分钟，彩虹喵说，要是 noip 难度，于是便有了数据范围。

第五分钟，诗人说，要有韵律，于是便有了时间限制和内存限制。

第六分钟，和雪说，要省点事，于是便有了保证运算过程中及最终结果均不超过 64 位有符号整数类型的表示范围的限制。

第七分钟，这道题终于造完了，然而，造题的神牛们再也不想写这道题的程序了。"

——《上帝造题的七分钟·第二部》

所以这个神圣的任务就交给你了。

### 【输入格式】

从文件 *god.in* 中读入数据。

第一行一个整数  $n$ ，代表数列中数的个数。

第二行  $n$  个正整数，表示初始状态下数列中的数。

第三行一个整数  $m$ ，表示有  $m$  次操作。

接下来  $m$  行每行三个整数  $k, l, r$ ， $k=0$  表示给  $[l, r]$  中的每个数开平方 (下取整)， $k=1$  表示询问  $[l, r]$  中各个数的和。

### 【输出格式】

输出到文件 *god.out* 中。

对于询问操作，每行输出一个回答。

### 【样例 1 输入】

```
10
1 2 3 4 5 6 7 8 9 10
5
0 1 10
1 1 10
1 1 5
```

0 5 8

1 4 8

**【样例 1 输出】**

19

7

6

**【子任务】**

对于 10% 数据,  $n, m \leq 10$ .

对于 20% 数据,  $n, m \leq 100$ .

对于 30% 数据,  $n, m \leq 1000$ .

对于 40% 数据,  $n, m \leq 10000$ .

对于 50% 数据,  $n, m \leq 50000$ .

对于 100% 的数据,  $1 \leq n, m \leq 100000$ ,  $1 \leq l \leq r \leq n$ , 数列中的数大于 0, 且不超过  $1e12$ 。

**【提示】**

数据不保证  $L \leq R$  若  $L > R$ , 请自行交换  $L, R$ , 谢谢!

## 游戏 (game)

### 【题目描述】

everlasting 觉得太无聊了，于是决定和蒟蒻 yyc 玩游戏！

他们会玩  $T$  轮游戏，每轮游戏中有若干局，他们的初始积分为 1，每局的分值为  $k$ ，输的人的得分乘  $k$ ，赢的人得分乘  $k^2$ 。每轮游戏后，everlasting 都会询问这次游戏双方的得分，但 yyc 记不住得分只能随口说两个得分。但他不知道这两个数最终会不会成为两个人的得分。于是 yyc 决定向你求助！

### 【输入格式】

从文件 *game.in* 中读入数据。

第一行一个正整数  $T$ ，表示游戏轮数。

接下来  $T$  行，每行两个数  $x,y$ ，表示 yyc 随口说出的两个人的得分。

### 【输出格式】

输出到文件 *game.out* 中。

$T$  行，每行一个 “Yes” 或 “No”。表示这两个数是否合法。

### 【样例 1 输入】

```
5
3 9
8 8
245 175
233 666
9 9
```

### 【样例 1 输出】

```
Yes
Yes
Yes
No
No
```

【样例 1 解释】

第一轮中，一局游戏  $k=3$ 。  
第二轮中，两局游戏  $k=2$ ，每人赢一局。  
第三轮中  $k=5$  和  $7$ ，两个人分别赢一局。

【样例 2】

见选手目录下的 `game/game2.in` 与 `game/game2.ans`。

【提示】

请选手用较快的输入输出方式进行数据处理。

【子任务】

测试点编号↵	$T$ ↵	$x$ ↵	$y$ ↵	其他性质↵
1↵	$= 1$ ↵	$= 2$ ↵	$\leq 10^9$ ↵	$x$ 是质数↵
2↵	$= 1$ ↵	$\leq 10$ ↵	$\leq 10$ ↵	↵
3↵	$\leq 10$ ↵	$\leq 10^3$ ↵	$\leq 10^3$ ↵	↵
4↵	$\leq 10^3$ ↵	$\leq 10^3$ ↵	$\leq 10^3$ ↵	↵
5↵	$= 1$ ↵	$\leq 10^9$ ↵	$\leq 10^9$ ↵	↵
6↵	$\leq 5 \times 10^5$ ↵	$\leq 10^9$ ↵	$\leq 10^9$ ↵	$x$ 是质数↵
7↵	$\leq 5 \times 10^5$ ↵	$\leq 10^9$ ↵	$\leq 10^9$ ↵	$x,y=2^k$ ↵
8↵	$\leq 5 \times 10^5$ ↵	$\leq 10^3$ ↵	$\leq 10^3$ ↵	↵
9↵	$\leq 5 \times 10^5$ ↵	$\leq 10^3$ ↵	$\leq 10^3$ ↵	↵
10↵	$\leq 5 \times 10^5$ ↵	$\leq 10^3$ ↵	$\leq 10^3$ ↵	↵
11↵	$\leq 10^3$ ↵	$\leq 10^9$ ↵	$\leq 10^9$ ↵	↵
12↵	$\leq 10^3$ ↵	$\leq 10^9$ ↵	$\leq 10^9$ ↵	↵

↵

13↵	$\leq 10^5$ ↵	$\leq 10^9$ ↵	$\leq 10^9$ ↵	↵
14↵	$\leq 10^5$ ↵	$\leq 10^9$ ↵	$\leq 10^9$ ↵	↵
15↵	$\leq 10^6$ ↵	$\leq 10^9$ ↵	$\leq 10^9$ ↵	x 是质数↵
16↵	$\leq 10^6$ ↵	$\leq 10^5$ ↵	$\leq 10^5$ ↵	↵
17↵	$\leq 10^6$ ↵	$\leq 10^9$ ↵	$\leq 10^9$ ↵	↵
18↵	$\leq 10^6$ ↵	$\leq 10^9$ ↵	$\leq 10^9$ ↵	↵
19↵	$\leq 10^6$ ↵	$\leq 10^9$ ↵	$\leq 10^9$ ↵	↵
20↵	$\leq 10^6$ ↵	$\leq 10^9$ ↵	$\leq 10^9$ ↵	↵