

## Day5

默认空间 512MB，时间 2s，开 O2，开栈，64 位 win10 测评，数据有梯度。

### 幸运数字(lucky.cpp)

#### 题目描述

定义一个 6 位数是 lucky 的，当且仅当它的前 3 位之和等于后 3 位之和，比如 165912  
给出数  $x$ ，找出最小的、大于  $x$  的幸运数字。

无解输出 -1.

#### 格式

输入：一个数，表示  $x$

输出：一个数，表示下一个幸运数字

#### 范围：

- $10^5 \leq x \leq 10^6 - 2$

#### Sample Input 0

555555

#### Sample Output 0

555564

### 售货员(salesman.cpp)

#### 题目描述

街道上有  $n$  个点  $x_1, x_2, \dots, x_n$ 。你可以在其中一个点开始行程，在任何一个点结束。要求遍历所有点。从点  $i$  走到点  $j$ ，需要花费  $\text{abs}(x_i - x_j)$  的时间。请问最少花费多少时间。

#### 格式

输入第一行一个数  $t$ ，表示数据组数。

对于每组数据，第一行一个数  $n$ ，第二行  $n$  个数  $x_i$

输出一共  $t$  行，每行一个数表示答案。

#### 范围

##### Constraints

- $1 \leq t \leq 500$
- $1 \leq n \leq 50$  (So there may be only one house)
- $1 \leq x_i \leq 1000$
- The houses' locations are distinct
- For 20% of the maximum score,  $n \leq 3$
- For 50% of the maximum score,  $n \leq 6$

#### Sample Input 0

```
1
3
11 6 9
```

#### Sample Output 0

```
5
```

## 好大的 Gcd(gcd.cpp)

### 题目描述

有两个数组 A,B, 大小均为 N。请在 A 中选择一个数 x, B 中选择一个数 y, 使得 gcd(x,y) 最大；如果有多组数 gcd 相同，找出 x+y 最大的。

### 格式

输入：第一行一个数 n, 第二行 n 个数表示  $A_i$ , 第三行 n 个数表示  $B_i$ 。

输出：输出 gcd 最大的那对数 (gcd 相同则和最大) 的 x+y 的值。

### 范围

- $1 \leq n \leq 5 \times 10^5$
- $1 \leq A_i \leq 10^6$
- $1 \leq B_i \leq 10^6$

#### Sample Input 0

```
5
3 1 4 2 8
5 2 12 8 3
```

#### Sample Output 0

```
16
```

Thus, maximum sum of pair (8, 8) is equal to  $8 + 8 = 16$

## 买买买(purchase.cpp)

### 题目描述

蓝月商场有 n 件宝贝，每件宝贝有两个属性：价钱 price 和品牌 brand。其中 brand 是 1-5 之间某个整数。**每件宝贝价钱两两不同。**

贪玩蓝月有 Q 个代言人，每个代言人拍完戏之后，希望能从蓝月商场免费顺走一样宝贝。但是每个代言人有自己的喜好，例如古天乐只喜欢品牌 1, 陈小春喜欢品牌 1 或品牌 2, 渣渣辉喜欢品牌 3 和 5……具体来说，代言人会有 d 个喜欢的品牌 ( $1 \leq d \leq 5$ )，同时他最喜欢这些品牌中，价钱第 k 便宜的宝贝。

请你求出每个代言人最喜欢的宝贝的价钱是多少！**如果不存在这件宝贝，请输出 -1.**

### 格式

输入：第一行一个整数 n, 第二行 n 个整数描述每件宝贝的品牌，第三行 n 个数描述每

件宝贝的价钱。第四行一个整数  $Q$ ，接下来  $Q \times 3$  行，每 3 行描述一个代言人的信息。其中第一行一个整数  $d$ ，第二行  $d$  个数表示喜欢的品牌，第三行一个数表示  $k$ 。

输出：一共  $Q$  行，每行一个数。

范围：

- $1 \leq n \leq 10^5$
- $1 \leq q \leq 10^5$
- $1 \leq prices_i \leq 10^9$
- $1 \leq brands_i \leq 5$
- $1 \leq d \leq 5$
- $1 \leq l_j \leq 5$
- $1 \leq k \leq 5 \cdot 10^5$
- all  $prices_i$  are distinct

### Sample Input 0

```
3
1 1 2
1 3 2
3
1
1
2
2
1 2
2
1
3
1
```

### Sample Output 0

```
3
2
-1
```

## 干爆字符串(string.cpp)

### 题目描述

有两个仅包含小写字母的字符串  $x, y$ ，长度分别为  $n, m$ 。你需要修改  $x$  串中某些字符，是新的  $x$  串和  $y$  串的最长公共子序列长度至少为  $k$ 。

我们将'a'...'z'对应成  $0 \cdots 25$ ，修改两个字符付出的代价，为它们对应的数的 xor 值。比如说，将'a'修改成'z'代价为  $0 \text{ xor } 25 = 25$ 。

请问最少付出多少代价，使得新的  $x$  串和  $y$  串的 LCS 大于等于  $k$ 。

格式

输入：第一行三个数  $n, m, k$ , 第二行长度为  $n$  的字符串  $x$ , 第三行长度为  $m$  的字符串  $y$ 。

输出：最小花费，如果不能请输出 -1。

范围：

**Constraints**

- $1 \leq n \leq 350$
- $1 \leq m \leq 350$
- $1 \leq k \leq 350$
- (All the characters in both  $x$  and  $y$  are lowercase English letters)

**Sample Input 0**

7 4 2

merging

pair

**Sample Output 0**

3

**Explanation 0**

You can convert the string "merging" to "perging" by replacing 'm' with 'p'. Now the longest common subsequence between them is "pi" whose length is 2. Thus, the pair of strings can be merged using LCS.  $\text{Cost} = f(m) \oplus f(p) = 12 \oplus 15 = 3$ .

## 阶乘数组(array.cpp)

**题目描述**

给出长度为  $n$  的序列  $A_1, A_2, \dots, A_n$ 。进行  $m$  次操作，有三种类型：

- 1、给出  $l, r$ ，将区间  $[l, r]$  的  $A_i$  都加一。
- 2、给出  $l, r$ ，询问区间  $[l, r]$  的  $A_i!$  的和，对  $10^9$  取模。
- 3、给出  $l, v$ ，将  $A_i$  单点修改为  $v$ 。

**格式**

输入第一行两个数  $n, m$ ，第二行  $n$  个数  $A_i$ 。

接下来  $m$  行，每行三个数，第一个数  $k$ ，表示第几类操作，后面两个数如题所述。

输出：对于每个操作 2 输出解。

**范围**

- $1 \leq n, m \leq 10^5$
- $1 \leq A_i, v \leq 10^9$
- $1 \leq l \leq r \leq n$
- $1 \leq i \leq n$

**Sample Input 0**

5 7

1 1 1 1 1  
2 1 5  
1 1 5  
2 1 5  
1 1 3  
2 1 5  
3 1 15  
2 1 5

### Sample Output 0

5  
10  
22  
674368016

	Array	Answer					
Before 1 <sup>st</sup> operation	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	5
1	1	1	1	1			
Before 3 <sup>rd</sup> operation	<table><tr><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td></tr></table>	2	2	2	2	2	10
2	2	2	2	2			
Before 5 <sup>th</sup> operation	<table><tr><td>3</td><td>3</td><td>3</td><td>2</td><td>2</td></tr></table>	3	3	3	2	2	22
3	3	3	2	2			
Before 7 <sup>th</sup> operation	<table><tr><td>15</td><td>3</td><td>3</td><td>2</td><td>2</td></tr></table>	15	3	3	2	2	674368016
15	3	3	2	2			