

A 序列异或

时空限制：1s 256MB

文件名

xor.in/xor.out/xor.cpp

题意

给一个长度为 n 的数组 a_1, a_2, \dots, a_n , 问有多少组满足条件的 b_1, b_2, b_3, b_4 ($1 \leq b_1 < b_2 < b_3 < b_4 \leq n$) 满足 $a_{b_1} \text{ xor } a_{b_2} \text{ xor } a_{b_3} \text{ xor } a_{b_4} = 0$ 。

输入格式

第一行一个整数 n 表示数组长度。接下来一行 n 个整数, 第 i 个整数 a_i 表示数组中第 i 个元素的值。

输出格式

输出一个数，表示答案。

输入样例

```
6
1 3 0 0 2 2
```

输出样例

```
5
```

样例解释

$b = \{1, 2, 3, 5\}, \{1, 2, 3, 6\}, \{1, 2, 4, 5\}, \{1, 2, 4, 6\}, \{3, 4, 5, 6\}$

数据范围

共 10 组数据,
测试点 1, 2 满足, $n \leq 100, a_i \leq 1000$ 。
测试点 3, 4 满足, $n \leq 1000, a_i \leq 1000$ 。
测试点 5, 6 满足, $n \leq 5000, a_i \leq 1000$ 。
对于 100% 的数据, 满足 $4 \leq n \leq 5000, 0 \leq a_i \leq 1000000$ 。

B 乘法破译

时空限制：1s 256MB

文件名

multiplication.in/multiplication.out/multiplication.cpp

题意

你获得了一张加密的 p 进制下的乘法表, 每个字母代表了一个 $0 \sim p - 1$ 之间的不同的整数。下面是一个 $p = 4$ 的例子。

\times	A	B	C	D
A	CD	BB	DC	BA
B	BB	BB	BB	BB
C	DC	BB	DB	BC
D	BA	BB	BC	BD

这里的 CD 表示 p 进制下的数字, 实际上也就是 $C \cdot p + D$ 。上面的乘法表中, 我们带入 $A = 3, B = 0, C = 2, D = 1$ 就成立。现在给你加密的乘法表, 希望你能找到每个字符表示的数。

输入格式

第一行, 一个整数 p 。接下来 p 行, 每行 $2p$ 个整数。第 i 行的 $2j - 1$ 和 $2j$ 个数表示 i 这个字符和 j 这个字符的乘积的高位和低位。这里我们用数字 $0 \sim p - 1$ 表示字符第 1 到 p 个字符。也就可以认为样例中的 0, 1, 2, 3 分别表示 ABCD。

输出格式

输出一行, 一共 p 个数字, 分别表示数字 0 到 $p - 1$ 对应的数字是什么。可以证明, 在题目限制下, 一定存在唯一解。

输入样例

```
4
2 3 1 1 3 2 1 0
1 1 1 1 1 1 1 1
3 2 1 1 3 1 1 2
1 0 1 1 1 2 1 3
```

输出样例

```
5
```

数据范围

共 10 组数据,

测试点 1, 2, 3 满足, $p \leq 10$ 。

测试点 4, 5 满足, $p \leq 50$ 。

测试点 6, 7 满足, $p \leq 500$ 。

对于 100% 的数据, 满足 $3 \leq p \leq 2000$, 保证乘法表一定合法。

C 锦标赛

时空限制：1s 256MB

文件名

tournament.in/tournament.out/tournament.cpp

题意

有 n 个玩家参加比赛, 他们分别有能力值 a_1, a_2, \dots, a_n 。

需要进行 $n - 1$ 轮比赛, 每一轮在剩下的玩家里任选两个玩家 i, j 。如果 $|a_i - a_j| > K$, 那么其中能力值高的玩家会获胜, 能力值低的玩家会被淘汰。如果 $|a_i - a_j| \leq K$, 那么两个玩家都有可能获胜, 另一个玩家被淘汰。

$n - 1$ 轮比赛之后, 只剩下一个玩家。问有多少个玩家可能是最后获胜的玩家。

输入格式

第一行, 两个整数 n, K , 表示玩家的总人数, 和获胜条件中的参数。

接下来一行 n 个整数 a_1, a_2, \dots, a_n , 表示玩家的能力值。

输出格式

一个整数, 表示最后可能获胜的玩家个数。

输入样例

```
5 3 1 5 9 6 3
```

输出样例

```
5
```

数据范围

共 10 组数据。

测试点 1 满足 $n \leq 5$ 。

测试点 2 满足 $n \leq 10$ 。

测试点 3, 4, 5 满足 $n \leq 1000$ 。

对于 100% 的数据, 满足 $n \leq 10^5, 1 \leq a_i, K \leq 10^9$ 。

D 括号序列

时空限制：1s 256MB

文件名

bracket.in/bracket.out/bracket.cpp

题意

给一个由左右括号构成的字符串 s , 对于每一个位置 i , 输出有多少个子串, 满足这个子串是一个合法的括号序列, 并且 i 这个位置在子串中。其中合法的括号序列定义如下:

- 空串是合法的。
- 如果 S 是合法的, 那么 (S) 也是合法的。
- 如果 U, V 是合法的, 那么 UV 也是合法的。

输入格式

一行, 一个由左右括号构成的字符串 S 。

输出格式

由于答案可能很大, 输出 $\sum (i \cdot ans_i \bmod (10^9 + 7))$ 即可, 其中 ans_i 表示第 i 个位置 的答案。注意这里我们要先取模, 再相加。

输入样例

```
((())
```

输出样例

```
49
```

样例解释

这里的 ans 分别为 1, 3, 3, 3, 3, 1。

数据范围

共十组数据。

对于 30% 的数据, 保证 $|S| \leq 5000$ 。

对于 70% 的数据, 保证 $|S| \leq 10^6$ 。

对于 100% 的数据, 保证 $|S| \leq 10^7$ 。