

C3 - Statement

A 摩卡与补码

题目描述

补码是一种用来表示有符号整数的编码方式，广泛用于计算机系统中。与原码和反码不同，补码能够统一处理正数和负数，并且简化了计算机中的加减运算。在 C 语言中，有符号整数常是以 32 位补码表示的。

现在，输入一个 `int` 范围内的整数，请你输出它的 32 位补码表示形式

输入

多行输入，每行一个正整数 n ，保证 n 在 `int` 范围内。

保证输入不超过 10000 行。

输出

对于每行输入，输出一行，为 n 的 32 位补码表示。

输入样例

```
5
-5
0
```

输出样例

```
00000000000000000000000000000101
11111111111111111111111111111011
00000000000000000000000000000000
```

Hint

本题希望同学们能够对 整数在计算机中存储的即为二进制补码形式 有更深刻的了解，如果你感到这个任务很困难的话，不妨试试直接输出一个整数的二进制位。

```
for(int i = 31 ; i >= 0 ;i--) {
    printf("%d", (n >> i) & 1);
}
```

B Unsigned medium int

题目描述

Xhesica不小心摔坏了自己电脑 $\frac{1}{4}$ 的CPU，现在，他发现自己电脑上的 `unsigned int` 只有 $32 \times \frac{3}{4} = 24$ 位了。Xhesica给这种 `unsigned int` 命名为 `unsigned medium int`。

`unsigned medium int` 实在是太有意思了，所以Xhesica将会给出你一些整数，你能告诉他这些整数能否用 `unsigned medium int` 存储，以及存储之后的二进制表示吗？

输入

第一行一个整数 t , $1 \leq t \leq 100$ 表示数据组数。

接下来 t 行，每行一个整数 a , $1 \leq |a| \leq 10^9$ 表示Xhesica的提问的整数。

输出

共 t 行，对于每一个整数 a ,如果 `unsigned medium int` 能够表示，则输出 a 的用 `unsigned medium int` 存储的二进制表示，即一个长度为24的 01 串。如果不能表示，输出 `we need a new cpu`

输入样例

```
3
1
1000000000
-1
```

输出样例

```
000000000000000000000001
we need a new cpu
we need a new cpu
```

Hint

只是迷你版的 `unsigned int`

C 摩卡与小水獭密码

题目描述

水獭们最近收到了一封神秘信件，信件中包含了一段看似杂乱无章的字符串。为了破解这封信的内容，水獭需要对字符串进行一些变换：把所有的小写字母变成大写字母，大写字母变成小写字母。同时，水獭还想知道，信件中出现次数最多的字符是什么。

Moca 决定编写一段程序帮助水獭完成这个任务。

输入

第一行一个正整数 n , $1 \leq n \leq 10000$ 。

第二行 n 个字符, 字符的 ASCII 值只可能在 33 到 126 之间 (包括 33 和 126)

输出

第一行 n 个字符, 对应着原来的 n 个字符: 如果原字符是个小写字母, 对应字符是其对应的大写字母; 如果原字符是个大写字母; 对应字符是其对应的小写字母; 否则对应字符是原字符。

第二行 1 个字符, 表示在**原字符串**中出现次数最多的字符。

第三行 1 个整数, 表示在**原字符串**中出现次数最多的字符出现的次数。

保证只有一个出现次数最多的字符!

输入样例

```
30
mITAKEmOCAlOVESbREADVERYmUCH~!
```

输出样例

```
MitakeMocaLovesBreadVeryMuch~!
E
4
```

Hint

可以参考 P3 - 例 3-11

D 置 0 置 1

题目描述

给定一个 `unsigned int` 范围内的自然数 n 。已知 `unsigned int` 的二进制由 32 位 0 和 1 构成, 记为 $s_{31}s_{30}\dots s_1s_0$ 。

现在, 你需要对自然数 n 连续进行 t 次操作。每次操作都需要选择一个位置 k 进行操作, 以及选择操作类型, 操作类型为 0 表示将第 k 位 s_k 设置为 0, 操作类型为 1 表示将第 k 位 s_k 设置为 1。

输入

- 第一行包含一个整数 n , 表示初始整数。保证 n 为 `unsigned int` 范围内的整数。
- 第二行包含一个整数 t , 表示操作次数。保证 $0 \leq t \leq 1000$
- 接下来 t 行, 每行包含两个整数, 分别为位数 k 和操作类型 α (0 或 1), 表示每次操作的位置和类型。其中保证 $0 \leq k \leq 31$ 。

输出

- 先输出 t 行，每行一个整数，表示每次操作后的结果。
- 最后一行输出一个整数，表示最终的结果。

输入样例

```
10
3
5 0
4 1
3 0
```

输出样例

```
10
26
18
18
```

样例解释：

初始整数是 10（二进制表示为 001010）。根据操作指令，分别执行以下操作：

1. 将第 5 位设置为 0，结果为 001010（二进制表示）。
2. 将第 4 位设置为 1，结果为 011010（二进制表示）。
3. 将第 3 位设置为 0，结果为 010010（二进制表示）。

最终的结果是 010010（十进制表示为 18）。

Hint

参考 P3 - 3.3 节按位与、按位或符号处的例题

E 点分十进制

题目描述

点分十进制（Dotted Decimal Notation）全称为点分（点式）十进制表示法，是 IPv4 的 IP 地址标识方法。IPv4 中用四个字节表示一个 IP 地址，每个字节按照十进制表示为 $0 \sim 255$ 。点分十进制就是用 4 组从 $0 \sim 255$ 的数字，来表示一个 IP 地址，其中四个字节之间用一个点分隔开，如 192.168.1.1。

现在给出一组点分十进制格式的数，请你编写程序将每个这样的 4 字节二进制串转换为**无符号**十进制数。

输入

第一行输入一个整数 T ($1 \leq T \leq 10^5$)。

接下来 T 行，每行一个满足题目描述部分所述限制的点分十进制数。

输出

对于每组数据，输出一行，表示点分十进制数对应的无符号十进制数。

输入样例

```
1
192.168.1.1
```

输出样例

```
3232235777
```

样例解释

192.168.1.1 对应的32位为 11000000 10101000 00000001 00000001，这32位二进制数转换为十进制数（无符号）对应为 3232235777。

F Baymax 的整数变换

题目描述

给定初始整数 0，你可以对其进行以下两种操作：

- 操作 1：将当前的整数加 1
- 操作 2：将当前的整数乘 2

Baymax 想知道，至少需要进行多少次操作，才能使初始整数 0 变为给定的目标整数 n 呢？

输入

第一行一个整数 T ，表示数据的组数， $0 < T \leq 100$ 。

接下来的 T 行，每行一个整数 n ，表示目标整数， $0 \leq n \leq 10^8$ 。

输出

对于每组数据，输出一行，表示需要进行的最小操作次数。

输入样例

```
3
4
114514
1919810
```

输出样例

```
3
27
30
```

样例解释

对于目标数 4，可以先对 0 进行一次加 1，再进行两次乘 2，最少需要 3 次操作。

G 小 p 与异或

题目背景

小 p 最近沉迷于异或运算的世界，他发现异或运算有很多有趣的性质，并且可以用来解决一些看似复杂的问题。

题目描述

小 p 给了你一个包含从 1 到 n 的连续自然数的数组，他希望你能迅速求出指定区间 $[l, r]$ 中所有数的异或和。

输入

第一行两个整数 n, T ，分别表示数组的大小和小 p 的询问次数。

接下来 T 行，每行 2 个整数 l, r ，表示每次询问的区间。

输出

共 T 行，每行输出一个整数，表示 $l \oplus (l + 1) \oplus \cdots \oplus r$ 。（其中 \oplus 表示异或运算）

输入样例

```
5 3
1 2
3 5
4 4
```

输出样例

```
3
2
4
```

数据范围

对于 40% 的数据, $1 \leq n \leq 10^6$ 。

对于 100% 的数据, $1 \leq n \leq 10^{18}, 1 \leq T \leq 10^5$ 。

Hint

我确信有一种绝妙的方法, 可惜这里太小写不下。

如果无从下手, 不妨想想 C1 的 J 题给过你什么样的启发?

H ddz 的 g^c^d

题目描述

ddz 有一个正整数 x , ddz 想找到一个严格小于 x 的正整数 y , 满足 $gcd(x, y) = x \oplus y$, $gcd(x, y)$ 表示 x 和 y 的最大公因数, \oplus 表示按位异或。

输入

第一行给出整数 n , 代表数据组数 ($0 < n < 10^5$)

接下来输入 n 行, 每行一个正整数为 x ($0 < x < 10^9$)

输出

对于每组数据, 输出一行一个正整数 y , 满足 $0 < y < x$ 且 $gcd(x, y) = x \oplus y$, 如果不存在满足条件的 y 则输出 -1 , 如果存在多个满足条件的 y 输出任意一个即可。

输入样例

```
2
5
4
```

输出样例

```
4
-1
```

样例解释

- 对于 5 来说, 因为 $gcd(5, 4) = 5 \oplus 4 = 1$, 所以 4 满足条件
- 对于 4 来说, 可以验证 1, 2, 3 都不满足条件, 所以输出 -1

I 丢失的号码牌

请注意本题的空间范围限制

题目描述

小 P 正在准备一场大型活动。在这场活动中，小 P 为每一位参与者准备了两张一模一样的号码牌，一张将交给参与者，另一张将用于最后的抽奖活动。但由于一个失误，所有的号码牌都混在了一起。在这次失误后，小 P 重新清点了号码牌数量，发现有号码牌出现丢失，且丢失的数量 **不超过两张**。

经过小 P 的确认，所有的号码都**至少保留一张**。因此小 P 将为那些丢失的号码重新制作一张该号码的号码牌。

现在，将给你未丢失的 n 张号码牌的号码 a_1, a_2, \dots, a_n ，请你编写一段程序，帮助小 P 统计出即将重新制作的号码。

输入

输入共两行。

第一行输入一个整数 n 。其中， $0 < n \leq 5 \times 10^6$ 。

第二行输入 n 个整数，为 a_1, a_2, \dots, a_n 。其中， $0 < a_i \leq 10^9$ 。

输出

若只需要重新制作一张号码牌，直接输出该张号码牌的号码。

若需要重新制作两张号码牌，请按从小到大的顺序依次输出两张号码牌的号码，中间请用一个空格间隔。

样例输入1

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

样例输出1

```
1
```

样例输入2

```
6
1 9 6 7 6 9
```

样例输出2

```
1 7
```


J 完全数

题目描述

我们定义，如果一个正整数的各个数位包含了 0-9 的所有数字，那么称其为「完全数」。例如，1023456789，990732184650 都是完全数，而 123456789，998244353 不是完全数。

Gino 很喜欢完全数。现在他有 n 个正整数 $a_1, a_2 \cdots a_n$ ，他想从中选择两个数 a_i, a_j ， $1 \leq i < j \leq n$ ，并将它们首尾相接，合并成一个大数 m 。请问，能使得 m 成为完全数的数对 (i, j) 总共有多少对？

输入

第一行一个正整数 n ，保证 $2 \leq n \leq 2 \times 10^5$ 。

第二行 n 个正整数 $a_1, a_2, \cdots a_n$ ，保证 $1 \leq a_i \leq 10^{18}$

输出

一个整数，表示能使得大数 m 成为完全数的数对 (i, j) 的数量。

输入样例 1

```
5
1234567890 1234567890 1234567890 1 2
```

输出样例 1

```
9
```

输入样例 2

```
5
14209 3867950 124389 114514 371852606
```

输出样例 2

```
4
```

样例解释

对于第一组样例，有且仅有 $(4, 5)$ 是不满足的，因此答案为 9。

对于第二组样例，满足条件的数对有 $(1, 2), (1, 5), (2, 3), (3, 5)$ ，因此答案为 4。