文件名 code.cpp/in/out xor.cpp/in/out gcd.cpp/in/out wave.cpp/in/out

题目名称	反码	异或	最大公约数	波浪
题目类型	传统型	传统型	传统型	传统型
每个测试点时限	C/C++ 1 秒，其他语言 2 秒	C/C++ 1 秒，其他语言 2 秒	C/C++ 1 秒，其他语言 2 秒	C/C++ 1 秒，其他语言 2 秒
内存限制	C/C++ 256MB，其他语言 512MB	C/C++ 256MB，其他语言 512MB	C/C++ 256MB，其他语言 512MB	C/C++ 256MB，其他语言 512MB
子任务数目	4	10	10	10
测试点是否等分	是	是	是	是

反码

【题目描述】

鸡尾酒今天学习了原码反码补码的概念，现在他想要设计一个程序，能够自动把原码转换成反码。

原码转换成反码的规则：原码的第一位为符号位，若符号位为 0，则反码与原码相同。若符号位为 1，则符号位不变，将其他位全部取反。

但是写代码太累了，于是鸡尾酒将这个任务交给了你。

【输入格式】

输入一行字符串表示一个原码。

【输出格式】

输出一行一个字符串表示该原码对应的反码。

【样例 1 输入】

00011101

【样例 1 输出】

00011101

【样例 2 输入】

1001

【样例 2 输出】

1110

【数据范围】

对于 50% 的数据，满足符号位为 0。

对于 100% 的数据，满足字符串长度不超过 32，且字符串仅由 0 和 1 构成。

异或

【题目描述】

鸡尾酒获得了一个数组，他定义数组的价值为每个相邻元素异或之和。例如数组 [3,5,9] 的价值为 $(3\text{^}5) + (5\text{^}9) = 18$，其中 ^ 为二进制中的异或运算。

现在他将数组向右进行 m 次**向右循环移动**，每次移动 b_i位，循环移动都建立在上一次移动的基础上进行。现在他想考考你，每次循环移动过后，这个数组的价值为多少?

向右循环移动的定义：向右循环移动一位，数组中最右边的元素变成最左边的元素，其余元素全部向右移动一位。

例如数组 [1,2,3,4,5] 向右循环移动两位的结果是 [4,5,1,2,3]。

【输入格式】

输入第一行包含两个正整数 n和m，分别代表数组的长度和循环移动的次数。

接下来一行包含n个由空格隔开的正整数，分别表示数组中每个数字的长度。

接下来一行包含m个正整数，分别表示每次循环移动要移动的位数 b_i。

【输出格式】

先输出一个答案表示原来数组的价值，然后输出m个整数，表示对于每一次循环移位后，数组当前的价值。所有输出的整数用空格隔开。

【样例 1 输入】

5 3

1 2 3 4 5

1 1 2

【样例 1 输出】

12 15 9 13

【样例 1 说明】

初始时数组价值为 (1^2) + (2^3) + (3^4) + (4^5) = 12。

第一次移位之后，数组变为 [5,1,2,3,4] ，其价值为 (5^1) + (1^2) + (2^3) + (3^4) = 15。

第二次移位之后，数组再次向右循环移动一次，数组变为 [4,5,1,2,3]，价值为 9。

第三次移位之后，数组向右移动两次，变为 [2,3,4,5,1]，价值为 13。

【样例 2 输入】

5 5

10 13 2 5 6

4

1

5

1

2

【样例 1 输出】

32 37 32 32 41 29

【数据范围】

对于 20% 的数据，满足 $n \leq 10^5, m = 0$。

对于另外 40% 的数据，满足 $b_i \leq n \leq 10^3, m \leq 5$。

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

最大公约数

【题目描述】

鸡尾酒的数学很差，他学了很长时间的最大公约数，终于有一天他会求最大公约数了。

于是他迫不及待地向你提问——给定数轴上的区间 $[l, r]$，你可以从中任选两个不相同的整数，求它们的最大公约数。请问它们的最大公约数最大为多少？

【输入格式】

输入两个正整数 l，r，意义如题面所示。

【输出格式】

输出一行一个正整数表示答案。

【样例 1 输入】

6 10

【样例 1 输出】

3

【样例 1 说明】

可以从中选出两个正整数 6 和 9，它们的最大公约数为 3。除此之外，没有一种选法可以使得两个数字的最大公约数更大。

【样例 2 输入】

10000 19999

【样例 2 输出】

6666

【数据范围】

对于 20% 的数据，满足 $l < r \leq 1000$。

对于另外 20% 的数据，满足 $r \% l == 0$。

对于 100% 的数据，满足 $1 \leq l < r \leq 10^7$。

波浪

【题目描述】

在 DOTA2 第八届国际邀请赛（TI8）中，选手 AME 操刀的水人在关键时刻释放了一个波浪的技能，扭转局势，帮助 OG 战队获得冠军。

为了纪念这一技能，我们引出一个“波浪数组”的概念，其定义如下：

对于除首尾位置之外的元素，每一个位置要么比两侧相邻的数字小，要么比两侧相邻的数字大。

例如 [1,3,2,5,3,4] 就是一个波浪数组，而 [2,3,4,1,2] 则不是，因为第二个位置 3 比左边的数字 2 大，比右边的数字 4 小。

注意：根据定义，长度小于等于 2 的数组一定是波浪数组。

现在有一个长度为n数组，你每次操作可以将任意一个位置的数字修改成任意一个新数字。我们想要将其变成一个波浪数组，请问最小的修改次数是几次？

【输入格式】

输入第一行包含一个正整数n，代表数组长度。

接下来一行包含n个正整数，其中第i个数字表示数组的第 i 个元素 a_i 。

【输出格式】

对于每一组询问，输出一行一个整数表示最少修改次数。

【样例 1 输入】

6

1 1 2 2 3 3

【样例 1 输出】

3

【样例 1 说明】

可以将数组改为 [1,5,2,4,1,3] 或者 [1,0,2,-1,5,3] 等，都是波浪数组，其中加粗的数字表示被修改的数字。我们可以发现至少修改三个数字才可以将原来的数组变为波浪数组，所以答案为 3。

【样例 2 输入】

11

703 702 703 703 702 703 702 702 702 700 702

【样例 2 输出】

3

【数据范围】

对于 10% 的数据，有 $1 \leq n \leq 20$。

对于 30% 的数据，有 $1 \leq n \leq 1000$。

对于另外 10% 的数据，有 $1 \leq n \leq 10^5$ ，且数组元素各不相同。

对于另外 10% 的数据，有 $1 \leq n \leq 10^5$ ，且数组元素全部相同。

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