

GDOI 练习 @CF

Set II

Duration: 4(.5)hrs

Carefully selected by @pwecar

| | | | |
|---------|-----------|-------------------------------------|-------------------------|
| 题目名称 | Two Sets | Have you ever heard about the word? | Sequence Transformation |
| 原题 | | | |
| 可执行文件名 | a.cpp/pas | b.cpp/pas | c.cpp/pas |
| 输入文件名 | a.in | b.in | c.in |
| 输出文件名 | a.out | b.out | c.out |
| 每个测试点时限 | 1S | 1S | 1S |
| 测试点数目 | 20 | 20 | 10 |
| 每个测试点分值 | 5 | 5 | 10 |
| 是否有部分分 | 无 | 无 | 无 |
| 题目类型 | 传统型 | 传统型 | 传统型 |

注意：最终测试时，所有编译命令均不打开任何优化开关。

Two Sets

(a.*)

Description

Philips 非常喜欢数字。最近 pwecar 送给他了一个由 n 个非负整数组成的集合。Philips 非常喜欢和 Bing 玩。他立即决定把他 n 个数字中的一部分送给 Bing。为了让游戏更加有趣，Philips 决定使得给她的数字集合满足如下条件：

我们用 x_1 表示 Philips 的数字集合的 xor 值，用 x_2 表示 Bing 的数字集合的 xor 值。要使得 $x_1 + x_2$ 尽可能地大。假如有多种划分集合的方法使得集合满足上述条件，Philips 就要让 x_1 尽可能地小。

Xor 运算是“异或”，在 Pascal 语言中用“xor”表示，在 C/C++/Java 语言中用“^”表示。

帮助 Philips 按照上述方法划分集合。如果有多种合适的方法，就找出其中任意一种方法。请注意，Philips 将一部分数字给了 Bing 之后，他可能就没有任何剩余的数字了。反之亦然，Philips 也可以不给 Bing 任何数字。在这两种情况下，我们都假定空集的 xor 值为 0。

Input format

第一行包含一个整数 n ，表示 pwecar 给了他多少个数字。第二行包含 n 个用空格隔开的数字，保证它们都是不超过 10^{18} 的非负整数。

Output format

输出一行，包含 n 个用空格隔开的整数，如果第 i 个数是 1，则表示 Philips 保留给出的第 i 个数字，如果第 i 个数是 2，则表示 Philips 把给出的第 i 个数字给了 Bing。

Sample Inputs

Input 1:

6
1 2 3 4 5 6

Input 2:

3
1000000000000 1000000000000 1000000000000

Input 3:

8
1 1 2 2 3 3 4 4

Sample Outputs

Output 1:
2 2 2 2 2 2
Output 2:
2 2 2
Output 3:
1 2 1 2 2 2 1 2

Constraints

对于 30%的数据 $n \leq 10$ 。
对于 60%的数据 $n \leq 1000$ 。
对于 100%的数据 $n \leq 100000$ 。

Attention

本题有 SPJ 喔~所以当有多个最优解时，输出任意一个即可。

Have You Ever Heard About the Word?

(b.*)

Description

String 专题又出现啦~(\(\cong \nabla \cong\))/~ 这次的题很水的啦 0.0

一个字符串的子串是该字符串的一段连续子序列,如 `bca` 是 `abcabc` 的子串,而 `cc` 不是。

一个重复块(repeating block)由一个字符串与自身连接而成,如 `abcabc` 是一个重复块,而 `abcabd`, `ababab` 不是。

你有一个由拉丁字符组成的字符串。每一步你要找到它的子串中最短的重复块,如果有多于一个,你必须选择最左边的那个。你要将那个形如 `xx(x - 某个字符串)` 的重复块替换成 `x`, 换句话说你要删除其中的一个 `x`。重复以上步骤直到字符串中不存在重复块。

最终的字符串会是怎样的? 看样例解释来更清楚地理解问题描述。

Input format

一行, 一个由小写拉丁字符组成的字符串 `s`。

Output format

输出做完所有操作后的字符串。

Sample input

```
aaaabaaab
```

Sample output

```
ab
```

Constrains

对于 10%的数据 $|S| \leq 10$ 。

对于 30%的数据 $|S| \leq 1000$ 。

这之间还有几个没那么大的点喔(◕‿◕)

对于 100%的数据 $1 \leq |S| \leq 50000$ 。

Sequensce Transformation

(c.*)

Description

你有一个非减的序列 x_1, x_2, \dots, x_n ($1 \leq x_1 \leq x_2 \leq \dots \leq x_n \leq q$)。你还有两个整数 a 和 b ($a \leq b; a(n-1) < q$)。

你要把序列 x_1, x_2, \dots, x_n 变换成 y_1, y_2, \dots, y_n ($1 \leq y_i \leq q, a \leq y_{i+1} - y_i \leq b$)，变换的代价为 $\sum_{i=1}^n (y_i - x_i)^2$ 。你的任务是选择一个序列 y 使得变换代价最小。

Input format

第一行有四个整数 n, q, a, b ($1 \leq q, a, b \leq 10^9; a(n-1) < q; a \leq b$)。

第二行有非减的序列 x_1, x_2, \dots, x_n ($1 \leq x_1 \leq x_2 \leq \dots \leq x_n \leq q$)。

Output format

在第一行输出 n 个实数——被找到的序列 y_1, y_2, \dots, y_n ($1 \leq y_i \leq q, a \leq y_{i+1} - y_i \leq b$)。

在第二行输出最小变换代价。

如果有多个最优答案，可以输出任意一个。

如果你的输出结果和参考结果相对误差的绝对值不超过 10^{-6} ，就会被认为是正确的。

Sample input

Input 1:

3 6 2 2

1 4 6

Input 2:

10 100000 8714 9344

3378 14705 17588 22672 32405 34309 37446 51327 81228 94982

Sample output

Output 1:

1.666667 3.666667 5.666667

0.666667

Output 2:

1.000000 8715.000000 17429.000000 26143.000000 34857.000000 43571.000000
52285.000000 61629.000000 70973.000000 80317.000000
797708674.000000

Constraints

对于 10%的数据, $1 \leq n \leq 10$

对于 40%的数据, $1 \leq n \leq 5900$

对于 70%的数据, $1 \leq n \leq 100000$

对于 100%的数据, $1 \leq n \leq 300000$