

第 2 题：转盘(circle)，运行时限 2s，内存上限 512M，100 分。

【问题描述】

一次小 G 和小 H 原本准备去聚餐，但由于太麻烦了于是题面简化如下：

一个转盘上有摆成一圈的 n 个物品（编号 1 至 n ），其中第 i 个物品会在 T_i 时刻出现。

在0时刻时，小 G 可以任选 n 个物品中的一个，我们将其编号记为 s_0 。并且如果 i 时刻选择了物品 s_i ，那么 $i + 1$ 时刻可以继续选择当前物品或者选择下一个物品。当 s_i 为 n 时，下一个物品为物品 1，否则下一个物品为 $s_i + 1$ 。在每一时刻（包括0时刻）如果小 G 所选择的物品已经出现了，那么小 G 将会标记它。小 H 想知道，在物品选择的最优策略下，小 G 什么时候能标记所有物品？

但麻烦的是，物品的出现时间会不时修改。我们将其描述为 m 次修改，每次修改将改变其中一个物品的出现时间。每次修改之后，你也要求出当前局面的答案。对于其中部分测试点，小 H 还追加了强制在线的要求。

【程序文件名】

源程序文件名为`circle.cpp/c/pas`。

【输入格式】

输入文件名为`circle.in`。

第一行三个非负整数 n 、 m 、 p ，代表一共有 n 个物品， m 次修改。 p 只有0或1两种取值，强制在线时 p 为1，否则为0。本节后面将解释如何使用 p 。

接下来一行，有 n 个用空格隔开的非负整数，第 i 个数 T_i 代表物品 i 的出现时间。

接下来 m 行，每行两个非负整数 x 、 y ，代表一次修改及询问。修改方式如下：

（1）如果 $p = 0$ ，则表示物品 x 的出现时间 T_x 修改为 y 。

（2）如果 $p = 1$ ，则先将 x 和 y 分别异或 $LastAns$ 得到 x' 和 y' ：即 $x' = x \text{ xor } LastAns$ ， $y' = y \text{ xor } LastAns$ 。然后将物品 x' 的出现时间 $T_{x'}$ 修改为 y' 。其中的 $LastAns$ 是前一个询问的答案；特别的，第一次修改时的 $LastAns$ 为初始局面的答案。其中的 xor 为按位异或运算，例如 $1 \text{ xor } 2 = 3$ ， $4 \text{ xor } 5 = 1$ ， $6 \text{ xor } 11 = 13$ 。

保证输入合法。

【输出格式】

输出文件名为`circle.out`。

第一行一个整数代表初始局面的答案。

接下来 $m + 1$ 行每行一个整数分别代表每次修改后的答案。

【输入输出样例】

circle1.in	circle1.out
5 3 0	5
1 2 3 4 5	7
3 5	6
5 0	7
1 4	

【样例解释】

第1次询问为：对于5个物品的出现时间分别是1、2、3、4、5时，最早什么时候标记完所有物品？其中一个最优策略为：在时刻0和1时都选择物品1，而在时刻2~5都选择下一物品。其中 0 时刻选择物品 1 尚未出现，无法标记。

第3次询问为：对于5个物品的出现时间分别是1、2、5、4、0时，最早什么时候标记完所有物品？。其中一个最优策略为：从时刻0至时刻6分别选择物品4、5、1、2、3、3、4。其中时刻0和时刻4时，所选物品尚未出现，其余时刻的所选物品都能被标记。

【数据范围】

测试点编号	n	m	T_i/T_x	p
1	≤ 10	≤ 10	≤ 10	$= 0$
2	≤ 1000	$= 0$	≤ 1000	
3	≤ 100000		≤ 100000	
4	≤ 5000	≤ 5000		
5	≤ 80000	≤ 80000		
6		$= 1$		
7	≤ 90000	≤ 90000		$= 0$
8		$= 1$		
9	≤ 100000	≤ 100000	$= 0$	
10			$= 1$	

对于所有数据，保证 $3 \leq n \leq 10^5$ ， $0 \leq m \leq 10^5$ ， $0 \leq T_i/T_x \leq 10^5$ 。

【编译命令】

对于 c++语言： `g++ -o circle circle.cpp -lm -O2`

对于 c 语言： `gcc -o circle circle.c -lm -O2`

对于 pascal 语言： `fpc circle.pas -O2`

提示：本题将开启 O2 优化指令。