

收割小麦(psenica)

时间限制: 1 Sec 内存限制: 128 MB

题目描述

小 C 和 Lay 博士正在麦田里玩游戏！他们分别有一个代号,小 C 叫 Mirko,Lay 博士叫 Slavko。他们制定了以下的游戏规则：

- 当轮到小 C 时，他选择最矮的一只小麦，把它提高到第一个比它高的小麦长度。
 - 轮到 Lay 博士时，他选择最高的一只小麦，剪到第一个比它矮的的小麦长度。
 - 若游戏轮到某个玩家 x,但小麦只有两种或两种以下不同的高度时，游戏结束,x 就输了。
- 现在假设游戏从小 C 开始，请帮他们判定谁能获胜和游戏结束时最高和最矮的麦子高度。

输入

第一行包括整数 N,表示麦子的个数.
接下来 N 个数，表示每个麦子的高度。高度不超过 105。

输出

第一行,输出获胜的一方,如果小 C 获胜,输出"Mirko";如果 Lay 博士获胜,输出"Slavko".
第二行两个整数,表示游戏结束时最矮和最高的麦子高度.

样例

3 3 3 3	Slavko 3 3
4 3 1 2 1	Slavko 1 2
7 2 1 3 3 5 4 1	Slavko 2 3

50%的数据 N<=500
80%的数据 N<=3000.
100%的数据 N<=100000.

序列游戏(rotiraj)

Time Limit:1 S. Memory Limit:128M.

Description

小 C 和小 T 设计了一个新游戏！小 T 有一个长度为 N 的序列，并把它分成许多子序列，每个子序列都有 K 个数， N 是 K 的倍数。第一个子序列是第 $1 \sim K$ 个数，第二个子序列是第 $K+1 \sim 2K$ 个数，以此类推。

小 C 让小 T 对序列进行一些操作，包括以下两种类型：

操作 1. 将每一个子序列中的数向左/向右轮换 x 个位置。（只在子序列里轮换位置）

操作 2. 将整个序列上的数向左/向右轮换 x 个位置。

注意操作 2 可以改变每个数所属的子序列。进行完所有操作之后，小 T 把最终序列给小 C，小 C 的任务就是找出小 T 原来的的序列。小 C 现在需要你的帮助！

Input

第一行包括三个正整数：

N 为序列的长度， K 为每个子序列的长度， Q 为操作的个数。

下面 Q 行，每行两个整数：

A ($1 \leq A \leq 2$) 指操作种类。

X 表示移动的步数，负数表示向左移动，正数表示向右移动。

最后一行 N 个用空格分开的整数 Z_i 表示所有操作后的序列。

Output

输出 N 个整数, 表示操作前的序列.

Sample

Input 4 2 2 2 2 1 1 3 2 1 0	Input 8 4 4 1 3 1 15 1 -5 2 -1 6 10 14 19 2 16 17 1	Input 9 3 5 1 1 2 -8 2 9 1 1 2 -4 3 1 8 7 4 5 2 6 9
Output 0 1 2 3	Output 6 10 14 1 2 16 17 19	Output 5 3 6 9 7 1 8 2 4

数据范围

测试点	n 的范围	K 的范围	Q 的范围	Zi 和 X 的范围
1	$1 \leq n \leq 100$	$1 \leq K \leq n$	$1 \leq Q \leq 100$	$0 \leq Zi \leq 10^5,$ $-10^5 \leq X \leq 10^5$
2			$1 \leq Q \leq 1000$	
3			$1 \leq Q \leq 10000$	
4			$1 \leq Q \leq 10^5$	
5	$1 \leq K \leq 100,$ $1 \leq K \leq n$			
6				
7				
8	$1 \leq K \leq n$			
9				
10				

XOR 寄存器(xor)

时间限制: 1 Sec 内存限制: 128 MB

题目描述

小 C 需要完成一个有趣的作业：设计一个小数据处理器。这个处理器共有 N 个寄存器，分别用 1 到 N 进行编号。每一个寄存器可以储存一个 32 位的 unsigned int（无符号的 32 位二进制整数，可能的取值为 0 到 $2^{32}-1$ ）。

这个数据处理器有以下两种功能：

指令	描述	例子
1 K M	在二进制下将寄存器 K 中的数的右端的 M 位移动到最左端	0000 0000 0000 0000 0010 0011 1111 1011 →(M=1010) → 1111 1110 1100 0000 0000 0000 0000 1000 (十进制: 9211→M=10→4 273 995 784)
2 K L	将寄存器 K 和寄存器 L 中的数进行 XOR 计算，并输出计算结果	0000 0000 0000 0000 0000 0011 1100 0111 XOR 0000 0000 0000 0111 1100 0000 0000 0111 = 0000 0000 0000 0111 1100 0011 1100 0000 (十进制: 967 XOR 507 911 = 508 864)

机智的小 C 完美地很快了作业，但是他突然发现他忘了写如何读取寄存器中原本寄存的数的操作。现在，小 C 唯一能做的就是用处理器中操作 1 和操作 2 得到答案来推断原本寄存器中所寄存的数。小 C 已经很累了，接下来就由你来帮助他得到每个寄存器中原本存放的数吧。

如果寄存器中的数大小有多种可能性，就找出字典序最小的一个。（对于任意两种可能性，如果寄存器 K-1 中储存的数相同，就比较寄存器 K 中储存的数的大小，选择较小的可能性。）

输入

输入第一行，两个整数， N ， E ，分别表示寄存器的数量和操作的次数；

接下来的按照操作进行的时间顺序输入对于操作的描述，格式同表格所述，每个操作占一行。

所有输入的格式都是合法的($1 \leq K, L \leq N$ ， $0 \leq M < 32$)。对于每一个 2 操作，会另外再输入一个计算结果 X ，单独占一行。所有的输入数据都是十进制的。

输出

输出仅一行，共 N 个整数表示各个寄存器中原本储存的数，用一个空格隔开。

如果根据输入数据推断出没有可行的方案，输出 -1，表示小 C 的处理器存在错误。

样例

3 3 2 1 2 1 2 1 3 2 2 2 3 3	0 1 2
4 6 2 4 2 3 2 4 1 6 1 3 1 2 3 1 2 1 2 2 2 2 3 7	5 0 14 3
5 6 2 4 2 10 2 5 3 2 2 2 3 1 2 1 4 3 1 3 1 2 3 4 2147483663	15 6 7 12 5

提示

对于 30%的数据，N，E 都不超过 1000，X 不超过 1023。

对于 60%的数据，不存在操作 1。

对于 100%的数据 $2 \leq N \leq 100\,000$ ， $1 \leq E \leq 100\,000$ ， $0 \leq X \leq 2^{32}-1$