算法基础（三十）：数学基础 - 博弈论 - 台阶Nim

直接上样例：

> 现在，有一个 n 级台阶的楼梯，每级台阶上都有若干个石子，其中第 i 级台阶上有 a_i 个石子($i \geq 1$)。
>
> 两位玩家轮流操作，每次操作可以从任意一级台阶上拿若干个石子放到下一级台阶中（不能不拿）。
>
> 已经拿到地面上的石子不能再拿，最后无法进行操作的人视为失败。
>
> 问如果两人都采用最优策略，先手是否必胜。
>
> **输入格式**
>
> 第一行包含整数 n。
>
> 第二行包含 n 个整数，其中第 i 个整数表示第 i 级台阶上的石子数 a_i。
>
> **输出格式**
>
> 如果先手方必胜，则输出 `Yes`。
>
> 否则，输出 `No`。
>
> **数据范围**
>
> $1 \leq n \leq 10^5$,
> $1 \leq a_i \leq 10^9$
>
> **输入样例：**
>
> ```
> 3
> 2 1 3
> ```
>
> **输出样例：**
>
> ```
> Yes
> ```

先说结论：如果先手遇到的奇数台阶上的石子$a_{2k-1} \oplus a_{2k-3} \oplus a_{2k-5} \ldots\ldots \oplus a_1 = x \neq 0$的话先手必赢，否则先手必输，下面来证明这个结论

- 假设先手遇到的奇数台阶上的石子满足：$a_{2k-1} \oplus a_{2k-3} \oplus a_{2k-5} \ldots\ldots \oplus a_1 = x \neq 0$，那么先手一定可以通过一个操作使得$a_{2k-1} \oplus a_{2k-3} \oplus a_{2k-5} \ldots\ldots \oplus a_1 = 0$

> 对于某一奇数台阶上的石子a_{2i+1}来说，一定有$a_{2i+1} \oplus x < a_{2i+1}$，所以先手可以移动$a_{2i+1} - a_{2i+1} \oplus x$个石子到下面的偶数级台阶上，使得这一级的奇数台阶上的石子个数变为$a_{2i+1} \oplus x$，这样操作就就可以使得留给后手的局面为所有的奇数级台阶的石子的异或和为`0`

- 若后手遇到$a_{2k-1} \oplus a_{2k-3} \oplus a_{2k-5} \ldots\ldots \oplus a_1 = 0$这种情况那么有两种可能：后手移动偶数台阶上的石子到奇数台阶上；后手移动奇数台阶上的石子到偶数台阶上，可以证明不管是哪一种情况，后手进行操作后最后只能遇到奇数级台阶上的石子个数异或和为`0`

> 若后手是从偶数级台阶移动石子到奇数级台阶上，假如移动第四级台阶上的`k`个石子到第三级台阶上，那么轮到先手操作的时候，可以把第三级台阶上来自第四级台阶的`k`个石子再移动到第二级台阶上，这样留给后手的局面没有发生变化，后手再次遇到的奇数级台阶的石子的个数的异或和为`0`

> 若后手从奇数级台阶移动石子到偶数级台阶上，那么他留给先手的局面一定是所有奇数级台阶上的石子个数异或和不为`0`的局面，这个与前面的`nim`游戏一样，可以用反证法证明，于是先手经过操作，又留给后手异或和为`0`的局面

- 所以到游戏终止的时候，先手遇到的局面总是奇数级台阶上的石子个数异或为`0`这种局面，这就说明奇数级台阶上始终有石子，先手始终有石子可以移动，所以到最后无法移动的局面一定是后手面对的，后手必输

代码实现极其简单：

```cpp
#include<iostream>
#include<algorithm>
using namespace std;

int main()
{
    int n;
    int res = 0;

    cin >> n;

    for(int i = 1; i <= n; i ++)
    {
        int x;
        scanf("%d", &x);
        if(i % 2) res ^= x;
    }

    if(res) puts("Yes");
    else puts("No");
    return 0;
}
```