→ Swapping 2 numbers using XOR

```
INT A = 2 , B = 3;

A = A^B;
B = B^A;   // Here, A is (A^B)
```
→ So, B = B^(A^B) → B^B^A → 0^A → A

```
A = A^B;   // Here, B is A and A is (A^B)
```
→ So, A = (A^B)^A → A^A^B → 0^B → B

→ Given array a of n integers. All integers are present in even count except one. Find that one integer which has odd count in: O(N) time complexity and O(1) space complexity.

$N < 10^{*5}$

$a[i] < 10^5$

Input:

9

2 3 3 3 7 7 2 8 8

Output:

3

→ Approach → We will use XOR property.
i) XOR of two same numbers is 0
ii) XOR of a number with 0 is the same number.

→ All the even count will be converted in 0 in the end and one odd count will remain.
eg: A^A^B^C^B^A^C
We can re-arrange that in:
A^A^A^B^B^C^C ⟶ 0^0^A ⟶ A

```
INT N;
CIN >> N;          // 9
INT x;
INT ANS = 0;

FOR (INT i = 0; i < n; i++)
{
    CIN >> x;      // 2 3 3 3 7 7 2 8 8
    ANS ^= x;
}

COUT << ANS << "\n"   // 3
```