

[136 Single Number \(link\)](#)

Description

Given a **non-empty** array of integers `nums`, every element appears *twice* except for one. Find that single one.

You must implement a solution with a linear runtime complexity and use only constant extra space.

Example 1:

Input: `nums = [2,2,1]`
Output: `1`

Example 2:

Input: `nums = [4,1,2,1,2]`
Output: `4`

Example 3:

Input: `nums = [1]`
Output: `1`

Constraints:

- $1 \leq \text{nums.length} \leq 3 * 10^4$
- $-3 * 10^4 \leq \text{nums}[i] \leq 3 * 10^4$
- Each element in the array appears twice except for one element which appears only once.

(scroll down for solution)

Solution

Language: cpp

Status: Accepted

```
#include <vector>
using namespace std;

class Solution {
public:
    int singleNumber(vector<int>& nums) {
        int result = 0;
        for (int num : nums) {
            result ^= num;
        }
        return result;
    }
};
```