Reward:  **1000**

A number is lonely if it only appears in a list once. If a number appears in the list twice it is not lonely. Given a list of numbers where every number in the list appears twice except one number that appears only once. You need to find the lonely number.

[Show Less](javascript:void(0);)

**Input 1 (nums)** → array.integer :

A list of numbers containing one lonely number

**Output** → integer :

The lonely number

**Mi solución ACEPTADA:**

#include <stdio.h>

#include <iostream>

#include <map>

#include <vector>

using namespace std;

int LonelyNumber(std::vector< int > nums) {

std::map<int, int> mapa;

for(int i = 0; i < nums.size(); i++) {

int key = nums[i];

if(mapa.count(key)){

mapa[key]++;

} else {

mapa[key] = 1;

}

}

int lonely = 0;

for(std::map<int, int>::iterator it = mapa.begin(); it != mapa.end(); it++) {

if(it->second == 1){

lonely = it->first;

break;

}

}

return lonely;

}

int main() {

/\*

std::string s1 = "aabcc";

std::string s2 = "adcaa";

int n = commonCharacterCount(s1, s2);

printf("Cant iguales: %d ", n);

\*/

vector<int> vec;

vec.push\_back(1);

vec.push\_back(6);

vec.push\_back(5);

vec.push\_back(5);

vec.push\_back(1);

vec.push\_back(6);

vec.push\_back(7);

vec.push\_back(4);

vec.push\_back(7);

int n = LonelyNumber(vec);

printf("\nLonely %d ", n);

return 0;

}

Gassa 's solution

int r,i, LonelyNumber(std::vector<int> a) {

for(i = a.size(); i--; ) {

r^=a[i];

}

int t=r;

}

