Given an integer array nums of length n where all the integers of nums are in the range [1, n] and each integer appears once or twice, return an array of all the integers that appears twice.You must write an algorithm that runs in O(n) time and uses only constant extra space.Example 1:Input: nums = [4,3,2,7,8,2,3,1]Output: [2,3]Example 2:Input: nums = [1,1,2]Output: [1]Example 3:Input: nums = [1]Output: [] Constraints:n == nums.length1 <= nums[i] <= nEach element in nums appears once or twice.

Solution

class Solution {

public:

vector<int> findDuplicates(vector<int>& nums) {

vector<int> res;

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

if(nums[abs(nums[i]) - 1] < 0) {

res.push\_back(abs(nums[i]));

}

nums[abs(nums[i]) - 1] = -1 \* nums[abs(nums[i]) - 1];

}

return res; }

};