Q1.Sample Input:{1,2,1,3,1}

Sample Output: true

Sample Input:{1,2,3,4,5}

Sample Output: false

#include <iostream>

#include <unordered\_set>

#include <vector>

bool hasDuplicate(const std::vector<int>& nums) {

std::unordered\_set<int> numSet;

for (int num : nums)

if (!numSet.insert(num).second)

return true;

return false;

}

int main() {

std::vector<int> input1 = {1, 2, 1, 3, 1};

std::vector<int> input2 = {1, 2, 3, 4, 5};

std::cout << "Sample Output 1: " << std::boolalpha << hasDuplicate(input1) << std::endl; // Should print true

std::cout << "Sample Output 2: " << std::boolalpha << hasDuplicate(input2) << std::endl; // Should print false

return 0;

}

Q2. Sample Input:{1,2,1,3,1}

Sample Output: 1

Sample Input:{1,2,3,4,2}

Sample Output: 2

#include <iostream>

#include <unordered\_map>

#include <vector>

int firstNonRepeating(const std::vector<int>& nums) {

std::unordered\_map<int, int> countMap;

for (int num : nums)

countMap[num]++;

for (int num : nums)

if (countMap[num] == 1)

return num;

return -1;

}

int main() {

std::vector<int> input1 = {1, 2, 1, 3, 1};

std::vector<int> input2 = {1, 2, 3, 4, 2};

std::cout << "Sample Output 1: " << firstNonRepeating(input1) << std::endl;

std::cout << "Sample Output 2: " << firstNonRepeating(input2) << std::endl;

return 0;

}