**Задача 2.** Карточки

Коллекционер-авантюрист собрал всю коллекцию с героями MARVEL и, недолго думая, начал собирать новую. Теперь у него имеется много-много (целых N штук!) карточек с изображением супергероев DC. Каждой карточке присвоен свой уникальный номер, выраженный натуральным числом, однако сами карточки в коллекции могут повторяться. Коллекционер хочет быстро узнать, имеется ли в его коллекции карточка с определённым номером. Необходимо написать программу, которая будет отвечать на M таких запросов за O(1).  
  
**Входные данные.**

На вход подаётся число N () – количество чисел в наборе. Далее через пробел вводится последовательность чисел из N значений (. Числа в последовательности могут повторяться. На следующей строке вводится число M () – число запросов. Далее вводится M строк, на каждой – натуральное число x () – номер карточки, наличие которой в коллекции нужно проверить.

**Выходные данные.**

Необходимо вывести ответ на каждый вопрос в следующем формате:

А) Вывести «yes» (без кавычек), если запрошенный номер карточки встретился в коллекции хотя бы один раз;

Б) Вывести «no» (без кавычек) в случае, если запрошенный номер в коллекции отсутствует.

**Пример 1.**

|  |  |
| --- | --- |
| input 8 5 7 1 1 8 1 6 5 3 5 7 10 | output yes yes no |

**Решение.**

Для хранения входных данных необходимо завести контейнер unordered\_set in. Обработка поступающих запросов происходит в цикле for и осуществляется с помощью метода find(x). Если метод не находит в последовательности введённое число x (то есть указывает на конец контейнера), тогда выводится «no», в противном случае в консоль выводится «yes».

#include <iostream>

#include <unordered\_set>

#include <vector>

#include <algorithm>

#include <set>

#include <fstream>

using namespace std;

int main()

{

ifstream testinput("01");

ofstream testoutput("01.a");

unordered\_set <int> in;

unordered\_set <int>::iterator init = in.begin();

int n, m, x;

testinput >> n;

for (int i = 0; i < n; i++)

{

testinput >> x;

in.insert(x);

}

testinput >> m;

for (int i = 0; i < m; i++)

{

testinput >> x;

if (in.find(x) == in.end()) testoutput << "no" << "\n";

else testoutput << "yes" << "\n";

}

testinput.close();

testoutput.close();

return 0;

}