**Задача 1**. Дубликаты

Коллекционер-авантюрист собрал целых N карточек с изображением супергероев MARVEL. Карточки в коллекции могут повторяться больше двух раз или не повторяться вообще. Каждой карточке присвоен свой уникальный номер, выраженный натуральным числом. Зная номера всех карточек, необходимо найти среди них все дубликаты, если они есть, и исключить их из последовательности.

**Входные данные.**

На вход подаётся число N () – количество чисел в наборе. Далее через пробел вводится последовательность чисел из N значений (.

**Выходные данные.**

На первой строке через пробел необходимо вывести количество различных номеров дубликатов, на второй строке – последовательность чисел без повторов элементов. Если дубликатов нет, вывести 0 на первой строке, а на второй – последовательность без изменений.

**Пример 1.**

|  |  |
| --- | --- |
| input 8 5 7 1 1 8 1 6 5 | output 2 5 7 1 8 6 |

**Решение.**

Для организации данных необходимо завести контейнер unordered\_set in для входных данных.

На вход будут подаваться числа – номера карточек x, и наличие повторов будет определяться следующим условием: если метод find(x) не находит в последовательности введённое число x (то есть указывает на конец контейнера), значит число уникальное и оно вводится в последовательность. В противном случае число является дубликатом, и его номер заносится в контейнер duplicate, причем искомое число повторов будет равняться числу элементов этого контейнера. Таким образом за один проход мы избавляемся от дубликатов карточек.#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> duplicate;

int n, x;

testinput >> n;

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

{

testinput >> x;

if (in.find(x) == in.end()) in.insert(x);

else

{

duplicate.insert(x);

}

}

if (!duplicate.empty())

{

testoutput << duplicate.size();

}

else testoutput << 0;

testoutput << "\n";

for (auto z : in) testoutput << z << " ";

testinput.close();

testoutput.close();

return 0;

}