МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХ™НИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

**

**ПРАКТИЧЕСКАЯ РАБОТА №3**

**по дисциплине: «*Программирование*»**

|  |  |
| --- | --- |
| Выполнил:  Студент гр. «АБс-323», «АВТФ»  *Баранов Михаил Олегович*  «29» апреля 2024г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | Проверил:  *Доцент кафедры ЗИ*  *Архипова Анастасия Борисовна*  «\_\_\_» \_\_\_\_\_\_ 2024г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Новосибирск 2024

**Цели и задачи работы:** изучение функционала библиотек algorithm и iterator на примере собственных задач.

**Задание к работе:** придумать и реализовать 10 задач с использованием алгоритмов и итераторов.

1. **Задание №1.**

Преподаватель расставил баллы за семестр, однако сайт университета перестал работать. Деканат срочно требует от преподавателя в бумажном виде передать переведенные в оценки баллы. Необходимо подсчитать количество каждой из оценок (перевод в баллы осуществляется на примере системы оценивания НГТУ).

**Реализация на C++**

#include <iostream>

#include <algorithm>

#include <iterator>

#include <vector>

using namespace std;

int pointsToGrade(int points) {

if (points >= 0 && points <= 24) {

return 1;

}

else if (points >= 25 && points <= 49) {

return 2;

}

else if (points >= 50 && points <= 72) {

return 3;

}

else if (points >= 73 && points <= 86) {

return 4;

}

else if (points >= 87 && points <= 100) {

return 5;

}

}

int main() {

system("chcp 1251");

vector<int>points = { 55, 91, 76, 74, 67, 65, 100, 95, 42, 24, 88, 26 , 10, 43, 56, 37, 73, 64, 38, 56, 74, 45, 73, 90};

vector<int> grades(5, 0);

for (auto it = points.begin(); it != points.end(); ++it) {

int grade = pointsToGrade(\*it);

grades[grade - 1]++;

}

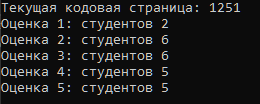
for (int i = 0; i < grades.size(); ++i) {

cout << "Оценка " << i + 1 << ": студентов " << grades[i] << "\n";

}

}

**Результат работы программы**



1. **Задание №2.**

Отец семьи из четырех человек собирается в магазин за продуктами. Ему надоело, что каждый член семьи составляет свой список продуктов, ему нужен один большой список, из которого будут удалены повторяющиеся продукты. Необходимо реализовать слияние четырех векторов с удалением повторяющихся элементов.

**Реализация на C++**

#include <iostream>

#include <algorithm>

#include <iterator>

#include <vector>

using namespace std;

int main() {

system("chcp 1251");

vector<string> dad = { "яблоки", "молоко", "хлеб", "сыр", "лапша"};

vector<string> mom = { "молоко", "яйца", "масло", "хлеб", "говядина"};

vector<string> son = { "яблоки", "масло", "кефир", "сыр", "чипсы"};

vector<string> daughter = { "яблоки", "яйца", "колбаса", "кефир", "творог"};

vector<string> family;

family.reserve(dad.size() + mom.size() + son.size() + daughter.size());

family.insert(family.end(), dad.begin(), dad.end());

family.insert(family.end(), mom.begin(), mom.end());

family.insert(family.end(), son.begin(), son.end());

family.insert(family.end(), daughter.begin(), daughter.end());

sort(family.begin(), family.end());

family.erase(unique(family.begin(), family.end()), family.end());

cout << "Общий список покупок без повторов:\n";

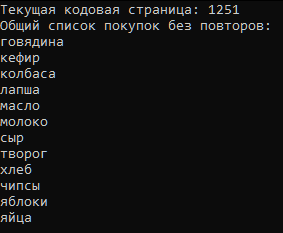
for (auto& product : family) {

cout << product << endl;

}

}

**Результат работы программы**



1. **Задание №3.**

В городе будущего у каждого военнослужащего вместо имени и звания используется свой уникальный восьмизначный номер. Этот номер строится следующим образом: имеется большой контейнер из всех двузначных чисел. Эти числа разбросаны случайным образом. Перед генерацией номера контейнер делится на 4 равные части. В первой и третьей частях выбирается по максимальному элементу, во второй и четвертой - по минимальному. Числа соединяются и получается номер. После этого номера удаляются из списка. Необходимо сгенерировать номер для военнослужащего при данном векторе двузначных чисел.

**Реализация на C++**

#include <iostream>

#include <algorithm>

#include <iterator>

#include <vector>

using namespace std;

int main() {

system("chcp 1251");

vector<int> numbers = { 14, 45, 34, 26, 75, 82, 13, 64, 86, 12, 99, 44 };

vector<int> unNumbers;

int quarterSize = numbers.size() / 4;

auto max1\_it = max\_element(numbers.begin(), numbers.begin() + quarterSize);

unNumbers.push\_back(\*max1\_it);

auto min2\_it = min\_element(numbers.begin() + quarterSize, numbers.begin() + 2 \* quarterSize);

unNumbers.push\_back(\*min2\_it);

auto max3\_it = max\_element(numbers.begin() + 2 \* quarterSize, numbers.begin() + 3 \* quarterSize);

unNumbers.push\_back(\*max3\_it);

auto min4\_it = min\_element(numbers.begin() + 3 \* quarterSize, numbers.end());

unNumbers.push\_back(\*min4\_it);

vector<int> elementsToRemove = { \*max1\_it, \*min2\_it, \*max3\_it, \*min4\_it };

for (int element : elementsToRemove) {

numbers.erase(remove(numbers.begin(), numbers.end(), element), numbers.end());

}

cout << "Уникальный номер: ";

for (auto& item : unNumbers) {

cout << item;

}

cout << "\n" << "Оставшиеся элементы: ";

for (auto& item : numbers) {

cout << item << ' ';

}

}

**Результат работы программы**



1. **Задание №4.**

Владелец ресторана попросил шеф-повара составить новое меню. Увидев новое меню, владелец ужаснулся, когда увидел количество позиций, содержащих лук, ведь он его ненавидит. Перед шефом новая задача – убрать все блюда, содержащие лук в названии, из меню. Необходимо убрать из вектора все элементы, содержащие в названии “onion”.

**Реализация на C++**

#include <iostream>

#include <vector>

#include <algorithm>

#include <string>

using namespace std;

bool containsOnion(string& dish) {

return !(dish.find("onion"));

}

int main() {

system("chcp 1251");

vector<string> menu = {"onion rings", "mushroom soup", "beef steak", "caesar salad", "onion soup"};

menu.erase(remove\_if(menu.begin(), menu.end(), containsOnion), menu.end());

cout << "Обновленное меню:\n";

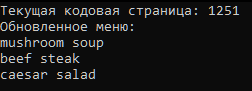
for (const auto& dish : menu) {

cout << dish << std::endl;

}

}

**Результат работы программы**



1. **Задание №5.**

У мальчика есть коллекция карточек Ninja Turtles, каждая из карточек имеет свой цвет: red – злодеи, grey – нейтральные герои или мутанты, green – мутанты-герои, blue – люди. Он хочет посчитать сколько у него мутантов-героев, но карточек слишком много, и он постоянно сбивается со счета. Однако у него есть список всех его карточек. Необходимо найти число карточек green.

**Реализация на C++**

#include <iostream>

#include <vector>

#include <algorithm>

#include <string>

using namespace std;

int main() {

system("chcp 1251");

vector<string> cards = { "red", "grey", "green", "blue", "green", "grey", "green", "red", "red", "red", "green", "grey", "green", "blue", "green", "green"};

int greenCards = count\_if(cards.begin(), cards.end(), [](const string& card) {

return card == "green";

});

cout << "Количество карточек мутантов-героев (цвет green): " << greenCards << endl;

}

**Результат работы программы**



1. **Задание №6.**

В честь своего дня рождения магазин «Пятерочка» объявил о необычной акции: в этот день на все продукты, начинающиеся с буквы «п», действует скидка 55%. Необходимо помочь администратору найти все подходящие товары из множества других.

**Реализация на C++**

#include <iostream>

#include <vector>

#include <algorithm>

#include <string>

using namespace std;

int main() {

vector<string> words = { "apple", "banana", "orange", "peach", "kiwi", "grape", "paprika", "pineapple"};

char filterLetter = 'p';

vector<string> filteredWords;

copy\_if(words.begin(), words.end(), back\_inserter(filteredWords),

[filterLetter](const string& word) {

return word.front() == filterLetter;

});

cout << filterLetter << ":\n";

for (auto& word : filteredWords) {

cout << word << endl;

}

}

**Результат работы программы**



1. **Задание №7.**

Есть список имен ваших друзей, но вы случайно забыли добавить некоторых из них. Нужно обновить список и добавить недостающих друзей в определенные позиции, по принципу важности.

**Реализация на C++**

#include <iostream>

#include <vector>

#include <algorithm>

#include <string>

using namespace std;

int main() {

// исходный список

vector<string> contacts = { "Alice", "Charlie", "Eve" };

// недостающие контакты

vector<pair<string, int>> missingContacts = { {"Bob", 1}, {"Dave", 3} };

for (const auto& contact : missingContacts) {

contacts.insert(contacts.begin() + contact.second - 1, contact.first);

}

cout << "Обновленный список контактов:\n";

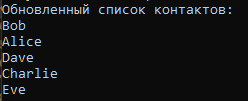
for (const auto& name : contacts) {

cout << name << endl;

}

}

**Результат работы программы**



1. **Задание №8.**

У вас есть список чисел. Ваша задача — разделить этот список так, чтобы все четные числа находились перед нечетными числами.

**Реализация на C++**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

bool isEven(int n) {

return n % 2 == 0;

}

int main() {

vector<int> numbers = { 3, 8, 5, 12, 7, 6, 9 };

partition(numbers.begin(), numbers.end(), isEven);

cout << "Обновленный список чисел:\n";

for (const auto& num : numbers) {

cout << num << " ";

}

cout << endl;

}

**Результат работы программы**



1. **Задание №9.**

Школьник Степан идет домой. Ему позвонила мама и сказала, что нужно купить хлеб. Чем ниже цена хлеба, тем менее качественным он окажется. Степану совсем негде нести мелочь, поэтому он рассматривает варианты без сдачи в первую очередь. Необходимо узнать, есть ли в каталоге хлеб с заданной ценой.

**Реализация на C++**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

vector<int> productPrices = { 10, 20, 30, 40, 50, 60, 70 };

int targetPrice = 30;

bool found = binary\_search(productPrices.begin(), productPrices.end(), targetPrice);

if (found) {

cout << "Цена " << targetPrice << " найдена в каталоге." << endl;

}

else {

cout << "Цена " << targetPrice << " не найдена в каталоге." << endl;

}

}

**Результат работы программы**



1. **Задание №10.**

У нас есть список оценок студентов, отсортированный по возрастанию. Нужно найти диапазон студентов, которые получили определенную оценку.

**Реализация на C++**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

system("chcp 1251");

vector<int> grades = { 50, 60, 70, 70, 70, 80, 90 };

int targetGrade = 70;

auto range = equal\_range(grades.begin(), grades.end(), targetGrade);

auto start = range.first - grades.begin();

auto end = range.second - grades.begin();

cout << "Диапазон оценок " << targetGrade << ": [" << start+1 << ", " << end << "]" << endl;

return 0;

}

**Результат работы программы**



**Вывод**

Таким образом, у меня получилось реализовать задачи с использованием алгоритмов и итераторов.