МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

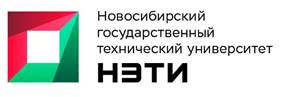
образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

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

Кафедра защиты информации

**

**ОТЧЁТ**

**По практической работе №3**

**«**Решение задач с помощью итераторов и алгоритмов»

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

Выполнил:Проверил:

Студент гр. «АБс-324», «АВТФ»                         *доцент кафедры ЗИ*

*Бурлаков И.Е. Архипова А. Б.*

«\_\_» \_\_\_\_\_\_ 2024 г.«\_\_\_» \_\_\_\_\_\_ 2024 г.

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

(подпись)                                                                       (подпись)

Новосибирск 2024

**Цели и задачи работы**: изучение алгоритмов и итераторов. Разработка программных решений, демонстрирующих применение итераторов и алгоритмов для решения задач.

**Задание к работе**: самостоятельно разработать и решить 20 задач используя итераторы и алгоритмы.

**Методика выполнения работы:**

1. Разработать 10 вариантов задач.

2. Написать и отладить программу решения задач.

3. Протестировать работу программы на различных исходных данных.

4. Ответить на вопросы по выполненным заданиям, по запросу преподавателя модифицировать код.

**Задача №1**

Задан список опоздавших нужно найти всех опоздавших

и отчислить  
#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

struct BD {

string fio;

string groupp;

int timeMin;

};

int main() {

setlocale(LC\_ALL, "Rus");

vector<BD> bd = {

{"Бурлаков", "АБС-324", -1},

{"Деревянкин", "АБС-324", 0},

{"Клаус", "АБС-324", -15},

{"Аксенов", "АБС-324", 7},

{"Тятюшкин", "АБС-324", 20},

{"Михайленко", "АБС-324", 13},

{"Рожкова", "АБС-324", 8},

{"Мокрушин", "АБС-324", 17},

};

vector<BD> result;

copy\_if(bd.begin(), bd.end(), back\_inserter(result),

[](const BD& mb) {

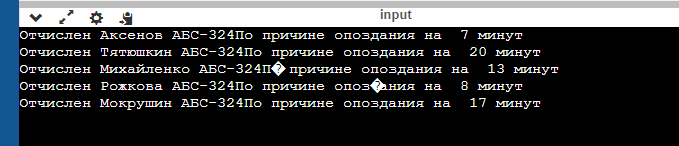
return mb.groupp == "АБС-324" and mb.timeMin > 0 ;

});

for (const auto& mb : result) {

cout << "Отчислен " << mb.fio << " " << mb.groupp << "По причине опоздания на " << " " << mb.timeMin << " минут" << endl;

}

}

**Задача №2**на паре по программированию были 2 гурппы

преподователь сказал "у какой группы больше всего попыток при сдаке коллоквиума то та группа будет расвормирована и отчислина"

найти ту группу которая больше всего потратила поппыток на сдачу коллоквиума

#include <vector>

#include <numeric>

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "Rus");

vector<int> abs324 = { 3, 5, 19, 0, 0, 11, 2, 1, 15, 20 };

vector<int> abs322 = {11, 4, 2, 5, 6, 9, 2, 7, 6, 8, 20 };

int resabs324 = accumulate(abs324.begin(), abs324.end(), 0);

int resabs322 = accumulate(abs322.begin(), abs322.end(), 0);

if (resabs324 > resabs322) {

cout << "АБс-324 проиграл с" << resabs324 << " попытками";

}

else if (resabs322 > resabs324) {

cout << "АБс-322 проиграл с " << resabs322 << " попытками";

}

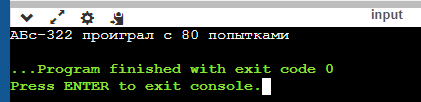
else {

cout << "У классов одно и то же количество баллов, победила дружба";

}

return 0;

}

****

**Задача №3**На лекции по теорие вероятности пришла проверка с военкомата

есть список с студентами

всех кто не достиг призывного возраста убрать из списка

найти самого молодого и самого старшего студента и вручить ему повестку

#include <iostream>

#include <algorithm>

#include <vector>

#include <string>

using namespace std;

struct Prizivnik {

string fio;

int age;

};

int main() {

setlocale(LC\_ALL, "Rus");

vector<Prizivnik> priziv =

{

{"Бурлаков И Е", 21},

{"Аксенов Н Е", 23,},

{"Клаус В Е", 18},

{"Тятюшкин М", 18},

{"Кошкин К О", 16},

{"Романенко В М", 20},

{"Святкой Д М", 15},

{"Рыгзынов Ч О", 17},

};

priziv.erase(remove\_if(priziv.begin(), priziv.end(),

[](const Prizivnik& p) {

return p.age < 16;

}));

if (!priziv.empty())

{

auto youngest = \*min\_element(priziv.begin(), priziv.end(),

[](const Prizivnik& a, const Prizivnik& b)

{ return a.age < b.age; });

auto oldest = \*max\_element(priziv.begin(), priziv.end(),

[](const Prizivnik& a, const Prizivnik& b)

{ return a.age < b.age; });

cout << "Самый молодой призывник: " << youngest.fio << " Его возраст: " << youngest.age << endl;

cout << "Самый старший призывник: " << oldest.fio << " Его возраст: " << oldest.age << "\nЗдравствуй юность в сопогах =)" << endl;

}

else

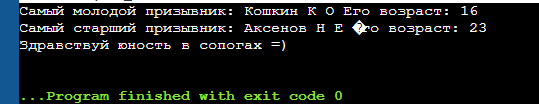
{

cout << "После фильтрации не осталось ни одного участника" << endl;

}

return 0;

}}

 **Задача №4**Вас отправили в магазин за стульями для комнат в общежитие , из за того, что вы плохо учили информатику пришлось позвать таких-же рабочих как и вы, но в процессе они сбились со счёта и сколько каких стульев вы купили вы не знаете. Необходимо посчитать сколько у вас стульев удобных и не удобных.  
#include <iostream>

#include <vector>

#include <algorithm>

#include <string>

#include <random>

using namespace std;

int main() {

setlocale(LC\_ALL, "Rus");

// Инициализация генератора случайных чисел

random\_device rd;

mt19937 gen(rd());

// Определение цветов

vector<string> chair = { "Удобный стул", "Не удобный стул" };

vector<string> chairs(200);

for (int i = 0; i < 200; i++) {

int index = gen() % 2; // Генерируем случайный индекс от 0 до 2

chairs.push\_back(chair[index]);

}

// Подсчет количества каждого цвета

int good\_chair = count(chairs.begin(), chairs.end(), "Удобный стул");

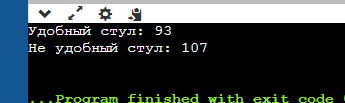
int not\_good\_chair = count(chairs.begin(), chairs.end(), "Не удобный стул");

// Вывод результатов

cout << "Удобный стул: " << good\_chair << endl;

cout << "Не удобный стул: " << not\_good\_chair << endl;

return 0;

} 

**Задача №5**Вова решил посмеяться над инфобезом и дал им загадку с реверсироваными словами Необходимо реверсировать слова в строке не изменяя порядок слов и ответить на загадку.  
#include <iostream>

#include <algorithm>

#include <string>

#include <sstream>

using namespace std;

int main() {

setlocale(LC\_ALL, "Rus");

string sentence = "атЭ ьщев ончыбо тисев в мотсуп екиньлидолох и юе язьлен ясьтавозьлоп ан 1 екиткарп";

istringstream iss(sentence);

string word;

string result;

string vopros = "Мышь";

while (iss >> word) {

reverse(word.begin(), word.end());

result += word + " ";

}

cout << "Реверсированные слова:\n " << result;

cout << "\nВашь ответ ? \n" ;

string otv;

cin >> otv;

if (vopros != otv)

{

cout << "Вы гений ";

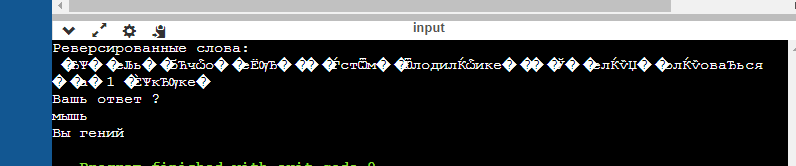
}

else

{

cout << " Вы пыталис (";

}

} 

**Задача №6**Есть группа студентов и их суммарные оценки за семестр по стобалльной системе. Необходимо преобразовать оценки в пятибальную систему.  
#include <iostream>

#include <string>

#include <vector>

using namespace std;

struct stud {

string fio;

int ball;

};

void pereobrozovanie(vector<stud>& students) {

for (auto b= students.begin(); b != students.end(); ++b) {

if (b->ball >= 84) {

b->ball = 5;

}

else if (b->ball >= 73) {

b->ball = 4;

}

else if (b->ball >= 50) {

b->ball = 3;

}

else {

b->ball = 0;

}

}

}

int main() {

vector<stud> students = {

{"Михайлов П.Ф", 90}, {"Бурлаков И.Е", 75}, {"Кошкин К.Е", 63}, {"Власова К.Е ", 44} };

pereobrozovanie(students);

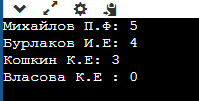
for (auto student : students) {

cout << student.fio << ": " << student.ball << endl;

}

return 0;

}



**Задача №7**Вы решили пойти в кино с друзьями, но двое из них придирчывые к выбору фильма.Они выдвинули два условия : фильм должен быть выпущен позже 1991 года и иметь рейтинг выше 8.2 на кинопоиске.Ваша задача — вывести список подходящих фильмов.

#include <iostream>

#include <vector>

#include <algorithm>

#include <string>

using namespace std;

struct kino {

string title;

int publication\_year;

double rating;

};

int main()

{

setlocale(LC\_ALL, "Rus");

vector<kino> movies = {

{"Джуманджи", 2011, 8.3},

{"Последний раз", 1999, 8.8},

{"Шрэк", 2001, 8.4},

{"Кто где и как ", 1994, 8.6},

{"Бойцовский клуб", 1999, 8.6},

};

int date = 1991;

double minR = 8.2;

vector<kino> sortedMovies;

// Скопировать фильмы в новый массив изходя из рейтинга и даты релиза.

copy\_if(movies.begin(), movies.end(), back\_inserter(sortedMovies), [date, minR](const

kino& movie) {

return movie.publication\_year > date && movie.rating > minR;

});

cout << "Фильмы, опубликованные после " << date << " и с рейтингом выше " << minR << ":" <<

endl;

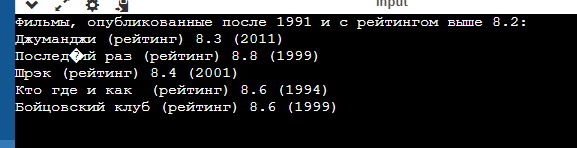
for (const auto& movie : sortedMovies) {

cout << movie.title << " (рейтинг) " << movie.rating << " (" << movie.publication\_year << ")" << endl;

}

return 0;

}

****

**Задача №8**Есть таблица баллов по предмету после завершения всех работ они умножаться на 3 найти максимальный балл

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

setlocale(LC\_ALL, "Rus");

vector<int> numbers = { 5, 8, 3, 21, 2 };

// Умножим каждый элемент вектора на 3

vector<int>::iterator it;

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

\*it \*= 3;

}

// Выведем измененные значения на экран

cout << "Все измененные баллы:" << endl;

for (const auto& num : numbers) {

cout << num << " ";

}

cout << endl;

// Найдем максимальный элемент вектора

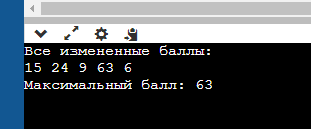
int max\_element = \*std::max\_element(numbers.begin(), numbers.end());

// Выведем максимальный элемент на экран

cout << "Максимальный балл: " << max\_element << endl;

return 0;

}

**  
Задача №9**Задан список баллов по предмету найти сумму всех баллов с учетом баллов по ргр

#include <iostream>

#include <vector>

#include <algorithm>

#include <numeric>

using namespace std;

int main() {

setlocale(LC\_ALL, "Rus");

vector<int> numbers = { 10, 10, 10, 10, 10, 10, 10};

// Найдем сумму всех чисел в векторе

int sum = 0;

sum = accumulate(numbers.begin(), numbers.end(), 0);

// Вывод сумму на экран

cout << "Сумма за все задания " << sum << endl;

// Умная все баллы на ргр

double otv = sum \* 0.3;

cout << "Сумма c учетом ргр " << otv << endl;

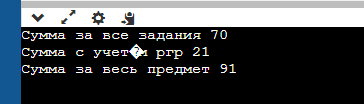
// Сумма за весь предмет

double ves\_predmet = sum + otv;

cout << "Сумма за весь предмет " << ves\_predmet << endl;

return 0;

}

**  
Задача №10**Есть список баллов по всем предметам нужно посчитать средний балл студента

#include <iostream>

#include <vector>

#include <algorithm>

#include <numeric>

using namespace std;

int main() {

setlocale(LC\_ALL, "Rus");

vector<int> grades = { 85, 92, 78, 88, 95 };

// Подсчитаем средний балл

double average = 0.0;

if (!grades.empty()) {

average = static\_cast<double>(accumulate(grades.begin(), grades.end(), 0)) / grades.size();

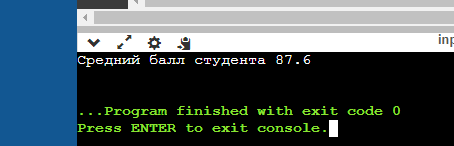
}

// Выведем средний балл на экран

cout << "Средний балл студента " << average << endl;

return 0;

}



**Вывод**

В ходе работы были изучены алгоритмы и итераторы, что позволило разработать и решить 10 задач с их использованием. Были созданы и отлажены программы, успешно протестированные на различных данных. Работа продемонстрировала эффективное применение итераторов и алгоритмов для решения практических задач.