**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10

по дисциплине: «Основы алгоритмизации и программирования»

**Студент группы № 219/2**

**Клопова Ульяна**

**Вариант 15**

**Преподаватель: Рябова В.В.**

Санкт-Петербург

2025 г.

[**Задание 1.**](../программы/Не%20лже%20Лупозар/Не%20лже%20Лупозар.sln)Определите вектор для хранения чисел типа int. Заполнить вектор рандомными числами в интервале от -21 до +19. Вывести вектор на экран. Удалить все отрицательные числа из вектора. Вывести полученный вектор

Отсортировать получившийся вектор по возрастанию. Найти макс и мин элемент в отсортированном массиве.

Удалить все дубликаты, оставляя только уникальные значения в векторе.

#include <iostream>

#include <vector>

#include <cstdlib>

#include <ctime>

#include <algorithm>

using namespace std;

bool is\_even(int i) {

return i < 0;

}

int main()

{

setlocale(0, "");

vector<int> vec;

srand(time(0));

cout << "Изначальный вектор" << endl;

for (int i = 0; i < 15; i++) {

vec.push\_back(rand() % 41 - 21);

cout << vec[i] << " ";

}

cout << endl;

cout << endl;

cout << "Удаление отрицательных значений" << endl;

auto iter = remove\_if(vec.begin(), vec.end(), is\_even);

vec.erase(iter, vec.end());

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

cout << vec[i] << " ";

}

cout << endl;

cout << endl;

cout << "Удаление дубликатов (если есть)" << endl;

auto last = unique(vec.begin(), vec.end());

vec.erase(last, vec.end());

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

cout << vec[i] << " ";

}

}

**Запуск 1.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Результат при первом запуске

**Запуск 2.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Результат при втором запуске

**Запуск 3.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Результат при третьем запуске

[**Задание 2.**](../программы/Синяя%20птица/Синяя%20птица.sln)Создать одномерный пустой вектор. Заполнить его **n** элементами (число n задает пользователь). Заполнение осуществлять рандомными числами через push\_back(). Вывести вектор в консоль. Найдите сумму элементов массива, стоящих после первого нечётного по значению элемента.

#include <iostream>

#include <vector>

#include <cstdlib>

#include <ctime>

#include <algorithm>

using namespace std;

int main()

{

setlocale(0, "");

vector<int> vec;

int n,sum=0;

cout << "Введите позицию для вставки (от 0 до 20 ):" <<endl;

cin >> n;

if (n < 1 || n > 20) {

cout << "Некорректное количество элементов." << endl;

return 1;

}

srand(time(0));

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

vec.push\_back(rand() % 61 - 30);

cout << vec[i] << " ";

}

bool a = false;

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

if (!a && vec[i] % 2 != 0) {

a = true;

}

else if (a) {

sum += vec[i];

}

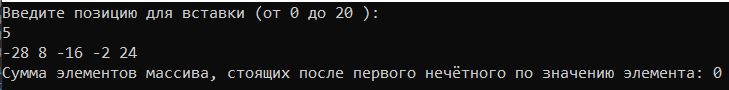
}

cout << endl;

cout << "Сумма элементов массива, стоящих после первого нечётного по значению элемента: " << sum << endl;

}

**Запуск 1.**

****

Результат при первом запуске

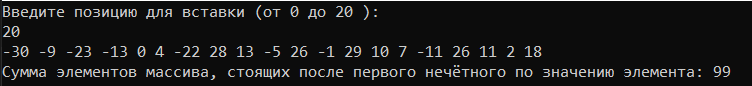
**Запуск 2.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Результат при втором запуске

**Запуск 3.**

****

Результат при третьем запуске

[**Задание 3.**](../программы/Реквием/Реквием.sln)Создать одномерный вектор из n элементов и заполнить его рандомными вещественными числами. Изменить размер вектора: увеличить его размер на 5 элементов (n + 5). Вывести вектор после изменения размера. Добавить элементы в вектор: два произвольных элемента идущих после 3 элемента. Вывести вектор после изменения.

#include <iostream>

#include <vector>

#include <cstdlib>

#include <iomanip>

#include <ctime>

using namespace std;

double rand\_plus(double a, double b) {

return a + (double)(rand()) / ((double)(RAND\_MAX / (a - b)));

}

int main() {

setlocale(0, "");

vector<double> vec;

srand(time(0));

cout << "Введите размер вектора: " << endl;

int n;

cin >> n;

if (n < 1 ) {

cout << "Некорректное количество элементов." << endl;

return 1;

}

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

vec.push\_back(rand\_plus(10, -5));

cout << setprecision(4) <<vec[i] << " ";

}

cout << endl;

vec.resize(n + 5);

cout << "Вектор после добавления 5 элементов: " << endl;

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

cout << setprecision(4) << vec[i] << " ";

}

cout << endl;

cout << "Вектор после добавления 2 произвольных чисел после 3 элемента вектора: " << endl;

vec.insert(vec.begin() + 3, rand\_plus(10,-5));

vec.insert(vec.begin() + 3, rand\_plus(10, -5));

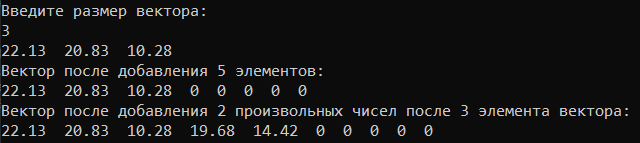
for (int i = 0; i < vec.size(); i++) {

cout << setprecision(4) << vec[i] << " ";

}

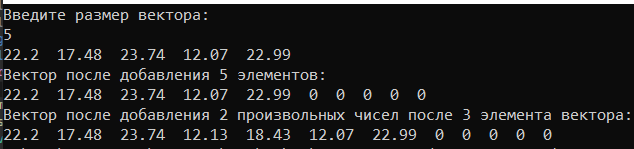
}

**Запуск 1.**

****

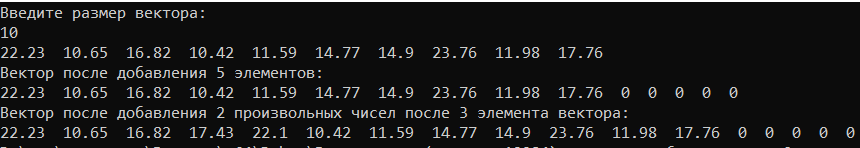
Результат при первом запуске

**Запуск 2.**

****

Результат при втором запуске

**Запуск 3.**

****

Результат при третьем запуске

[**Задание 4.**](../программы/Young%20girl%20A/Young%20girl%20A.sln)Создать одномерный вектор из **n** элементов и заполнить его рандомными целыми числами. Удалить следующие позиции в векторе: первые три элемента в векторе. Вывести вектор после удаления.

Создать еще один одномерный вектор из **n** элементов и заполнить его рандомными целыми числами. Удалить все элементы, попадающие под условие: все отрицательные двузначные числа. Вывести вектор после удаления.

#include <iostream>

#include <vector>

#include <cstdlib>

#include <ctime>

#include <algorithm>

using namespace std;

bool isNegative(int num) {

return (num < 0 && num >= -99 && num <= -10);

}

int main() {

setlocale(0, "");

int n;

cout << "Введите размер вектора: " << endl;

cin >> n;

if (n < 1) {

cout << "Некорректное количество элементов." << endl;

return 1;

}

vector<int> vec1;

srand(time(0));

cout << "Первый вектор:" << endl;

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

vec1.push\_back(rand() % 61 - 30);

cout << vec1[i] << " ";

}

cout << endl;

cout << "Первый вектор после удаления первых трех элементов:" << endl;

if (vec1.size() >= 3) {

vec1.erase(vec1.begin(), vec1.begin() + 3);

for (int i = 0; i < vec1.size(); i++) {

cout << vec1[i] << " ";

}

cout << endl;

}

else {

cout << "В первом векторе недостаточно элементов для удаления." << endl;

}

cout << endl;

vector<int> vec2;

cout << "Второй вектор:" << endl;

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

vec2.push\_back(rand() % 201 - 100);

cout << vec2[i] << " ";

}

cout << endl;

cout << "Второй вектор после удаления отрицательных двузначных чисел:" << endl;

vec2.erase(remove\_if(vec2.begin(), vec2.end(), isNegative), vec2.end());

for (int i = 0; i < vec2.size(); i++) {

cout << vec2[i] << " ";

}

cout << endl;

return 0;

}

**Запуск 1.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Результат при первом запуске

**Запуск 2.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Результат при втором запуске

**Запуск 3.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Результат при третьем запуске

[**Задание 5.**](../программы/Революция/Революция.sln)Создать одномерный вектор размерностью **n** элементов. Заполнить вектор целыми случайными числами. Вывести вектор на экран. Сделать меню с пунктами:

1. «Найти максимальное число»;

2. «Найти минимальное число»;

3. «Найти позицию максимального числа» (вывести все индексы если таких чисел несколько);

4. «Найти позицию минимального числа» (вывести все индексы если таких чисел несколько);

5. «Поиск в массиве заданного числа» (вывести все индексы если таких чисел несколько);

6. «Повторить с новыми числами» (пересобрать массив с новыми числами);

7. «Выход».

Реализацию пунктов меню осуществить отдельными функциями. После вызова 1-5 пункта – в консоль выводится результат операции и снова открывается меню с выбором действий с массивом. После вызова 6 пункта меню, произведите очистку консольного окна с выводом нового массива и отображением списка меню. Для очистки можно использовать system("cls"). Пункт 7 – завершает работу программы.

#include <iostream>

#include <vector>

#include <cstdlib>

#include <ctime>

#include <algorithm>

#include <limits>

#ifdef \_WIN32

#define CLEAR "cls"

#else

#define CLEAR "clear"

#endif

using namespace std;

void fillVector(vector<int>& vec, int n) {

vec.clear();

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

vec.push\_back(rand() % 61 - 30);

}

}

void printVector(const vector<int>& vec) {

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

cout << vec[i] << " ";

}

cout << endl;

}

int fMax(const vector<int>& vec) {

if (vec.empty()) {

cout << "Вектор пуст." << endl;

return numeric\_limits<int>::min();

}

return \*max\_element(vec.begin(), vec.end());

}

int fMin(const vector<int>& vec) {

if (vec.empty()) {

cout << "Вектор пуст." << endl;

return numeric\_limits<int>::max();

}

return \*min\_element(vec.begin(), vec.end());

}

void fMaP(const vector<int>& vec) {

if (vec.empty()) {

cout << "Вектор пуст." << endl;

return;

}

int maxVal = fMax(vec);

cout << "Индексы максимального числа (" << maxVal << "): ";

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

if (vec[i] == maxVal) {

cout << i + 1 << " ";

}

}

cout << endl;

}

void fMiP(const vector<int>& vec) {

if (vec.empty()) {

cout << "Вектор пуст." << endl;

return;

}

int minVal = fMin(vec);

cout << "Индексы минимального числа (" << minVal << "): ";

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

if (vec[i] == minVal) {

cout << i + 1 << " ";

}

}

cout << endl;

}

void fNum(const vector<int>& vec, int target) {

if (vec.empty()) {

cout << "Вектор пуст." << endl;

return;

}

cout << "Индексы числа (" << target << "): ";

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

if (vec[i] == target) {

cout << i + 1 << " ";

}

}

cout << endl;

}

int main() {

setlocale(0, "");

vector<int> vec;

int n, cho;

cout << "Введите размер вектора: " << endl;

cin >> n;

if (n < 1) {

cout << "Некорректное количество элементов." << endl;

return 1;

}

srand(time(0));

fillVector(vec, n);

do {

system(CLEAR);

cout << "Вектор: ";

printVector(vec);

cout << "Выберете действие: " << endl;

cout << "1. Найти максимальное число" << endl;

cout << "2. Найти минимальное число" << endl;

cout << "3. Найти позицию максимального числа" << endl;

cout << "4. Найти позицию минимального числа" << endl;

cout << "5. Поиск в массиве заданного числа" << endl;

cout << "6. Повторить с новыми числами" << endl;

cout << "7. Выход" << endl;

cout << "Ваш выбор: ";

cin >> cho;

switch (cho) {

case 1: {

int maxVal = fMax(vec);

cout << "Максимальное число: " << maxVal << endl;

break;

}

case 2: {

int minVal = fMin(vec);

cout << "Минимальное число: " << minVal << endl;

break;

}

case 3: {

fMaP(vec);

break;

}

case 4: {

fMiP(vec);

break;

}

case 5: {

int a;

cout << "Введите число для поиска: ";

cin >> a;

fNum(vec, a);

break;

}

case 6: {

fillVector(vec, n);

break;

}

case 7: {

cout << "Бааай" << endl;

break;

}

default:

cout << "Нет такого действия: " << endl;

break;

}

cout << "\nНажмите Enter, чтобы продолжить...";

cin.ignore();

cin.get();

} while (cho != 7);

return 0;

}