**Министерство науки и высшего образования Российской Федерации**

**Пензенский государственный университет**

**Кафедра «Вычислительная техника»**

**Отчет**

по лабораторной работе №1

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Простые структуры данных».

Выполнили студ. группы 24ВВВ4:

Агуреев А.П.

Невежин И.С.

Приняли:

к.э.н. доцент Акифьев И.В.\_\_\_\_

к.т.н. доцент Юрова О.В.\_\_\_\_

Пенза 2025

**Цель работы:**Вспомнить и закрепить навыки работы с простыми структурами данных.

**Лабораторные задания:**

**Задание 1:**Написать программу, вычисляющую разницу между максимальным и минимальным элементами массива.

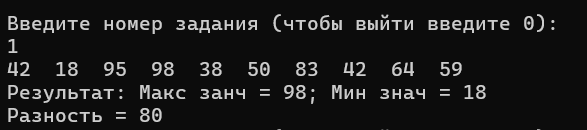
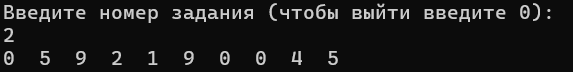
**Задание 2:**Написать программу, реализующую инициализацию массива случайными числами.

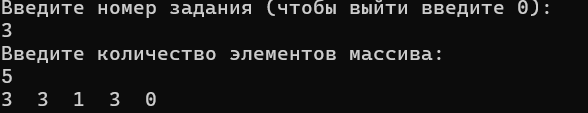
**Задание 3:**Написать программу, реализующую создание массива произвольного размера, вводимого с клавиатуры.

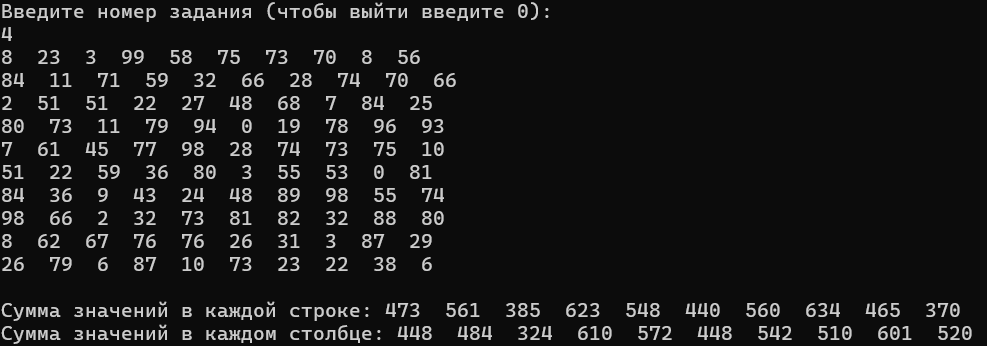
**Задание 4:**Написать программу, вычисляющую сумму значений в каждом столбце (или строке) двумерного массива.

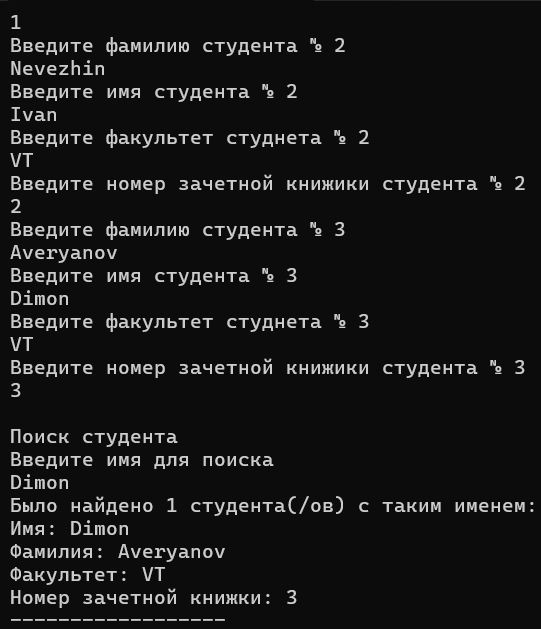
**Задание 5:**Написать программу, осуществляющую поиск среди структур student структуру с заданными параметрами (фамилией, именем и т.д.).

**Результат работы программы:**

**Задание 1:** **Задание 2:**

**Задание 3:**

**Задание 4:**

**Задание 5:**

**Листинг:**

#include <iostream>

#include <stdio.h>

#include <vector>

using namespace std;

void first\_task() {

int arr[10], max\_el = -1, min\_el = 101, raznost;

for (int i = 0; i < 10; i++) {

arr[i] = rand() % 100;

if (max\_el < arr[i]) {

max\_el = arr[i];

}

if (min\_el > arr[i]) {

min\_el = arr[i];

}

cout << arr[i] << " ";

}

raznost = max\_el - min\_el;

cout << "\nРезультат: Макс занч = " << max\_el << "; Мин знач = " << min\_el << "\nРазность = " << raznost;

}

void second\_task() {

int arr[10];

for (int i = 0; i < 10; i++) {

arr[i] = rand() % 10;

cout << arr[i] << " ";

}

}

int error\_fix(int arg) {

while (!(cin >> arg)) {

cin.clear(); // Сброс флага ошибки

cin.ignore(numeric\_limits <streamsize>::max(), '\n'); //Очищение от необработанных символов в cin

cout << "Вы ввели некорректный символ!\n";

cout << "Введите число:\n";

}

return arg;

}

void third\_task() {

int count = 0;

cout << "Введите количество элементов массива:\n";

count = error\_fix(count);

int\* arr = (int\*)malloc(count \* (sizeof(int)));

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

arr[i] = rand() % count;

cout << arr[i] << " ";

}

free(arr);

}

void fourth\_task() {

int arr[10][10], sum\_string[10], sum\_column[10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

arr[i][j] = rand() % 100;

cout << arr[i][j] << " ";

}

cout << "\n";

}

for (int i = 0; i < 10; i++) {

int tmp = 0, tmp\_2 = 0;

for (int j = 0; j < 10; j++) {

tmp += arr[i][j];

tmp\_2 += arr[j][i];

}

sum\_string[i] = tmp;

sum\_column[i] = tmp\_2;

}

cout << "\nСумма значений в каждой строке: ";

for (int i = 0; i < 10; i++) {

cout << sum\_string[i] << " ";

}

cout << "\nСумма значений в каждом столбце: ";

for (int i = 0; i < 10; i++) {

cout << sum\_column[i] << " ";

}

}

struct Student {

string surname, name, faculty;

int Nomzach;

} stud[3];

void print\_students(const vector<int>&students) {

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

int id\_student = students[i];

cout << "Имя: " << stud[id\_student].name << '\n';

cout << "Фамилия: " << stud[id\_student].surname << '\n';

cout << "Факультет: " << stud[id\_student].faculty << '\n';

cout << "Номер зачетной книжки: " << stud[id\_student].Nomzach << '\n';

cout << "------------------\n";

}

}

int last\_task() {

for (int i = 0; i < 3; i++) {

int tmp = i + 1;

cout << "Введите фамилию студента № " << tmp << '\n';

cin >> stud[i].surname;

cout << "Введите имя студента № " << tmp << '\n';

cin >> stud[i].name;

cout << "Введите факультет студнета № " << tmp << '\n';

cin >> stud[i].faculty;

cout << "Введите номер зачетной книжики студента № " << tmp << '\n';

int tmp\_2 = 0;

stud[i].Nomzach = error\_fix(tmp\_2);

}

cout << "\nПоиск студента\n";

bool find\_student = 1;

vector<int> students;

vector<int> filter;

string SearchName, SearchSurname, SearchFaculty;

int SearchNomzach = 0;

cout << "Введите имя для поиска\n";

cin >> SearchName;

for (int i = 0; i < 3; i++) {

if (stud[i].name == SearchName) {

students.push\_back(i);

}

}

if (students.empty()) {

cout << "Совпадений не было найдено\n";

return 0;

}

cout << "Было найдено " << students.size() << " студента(/ов) с таким именем:\n";

print\_students(students);

if (students.size() == 1) {

return 0;

}

cout << "\nВведите фамилию студента\n";

cin >> SearchSurname;

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

int id\_student = students[i];

if (stud[id\_student].surname == SearchSurname) {

filter.push\_back(id\_student);

}

}

students = move(filter);

if (students.empty()) {

cout << "Совпадений не было найдено\n";

return 0;

}

cout << "С такими параметрами найдено " << students.size() << " совпадений:\n";

print\_students(students);

if (students.size() == 1) {

return 0;

}

cout << "\nВведите факультет студента\n";

cin >> SearchFaculty;

filter.clear();

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

int id\_student = students[i];

if (stud[id\_student].faculty == SearchFaculty) {

filter.push\_back(id\_student);

}

}

students = move(filter);

if (students.empty()) {

cout << "Совпадений не было найдено\n";

return 0;

}

cout << "С такими параметрами найдено " << students.size() << " совпадений:\n";

print\_students(students);

if (students.size() == 1) {

return 0;

}

cout << "\nВведите номер зачетной книжики студента\n";

SearchNomzach = error\_fix(SearchNomzach);

filter.clear();

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

int id\_student = students[i];

if (stud[id\_student].Nomzach == SearchNomzach) {

filter.push\_back(id\_student);

}

}

students = move(filter);

if (students.empty()) {

cout << "Совпадений не было найдено\n";

return 0;

}

cout << "С такими параметрами найдено " << students.size() << " совпаден(ия/ие):\n";

print\_students(students);

}

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(nullptr));

int tasks\_number = 0;

do{

cout << "\nВведите номер задания (чтобы выйти введите 0):\n";

tasks\_number = error\_fix(tasks\_number);

if (tasks\_number > 5) {

cout << "Такого задания нет";

continue;

}

switch (tasks\_number) {

case 1:

first\_task();

break;

case 2:

second\_task();

break;

case 3:

third\_task();

break;

case 4:

fourth\_task();

break;

case 5:

last\_task();

break;

}

} while(tasks\_number != 0);

}

**Вывод:**В ходе данной лабораторной работы вспомнили и закрепили навыки работы с простыми структурами данных.