Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЁТ**

по лабораторной работе №1

по курсу «Логика и основы алгоритмизации в ИЗ»

на тему: «Простые структуры данных»

Выполнил:

студент группы 23ВВВ4

Соснин Глеб

Проверил:

доцент, Юрова О. В.

Пенза, 2024

**Цель** – познакомиться с простыми структурами данных языка Си, получить навыки работы с массивами и структурами.

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

Максимальный элемент массива — это элемент, который имеет самое большое числовое значение, а минимальный элемент массива — это элемент, имеющий самое маленькое значение.

**Листинг программы:**

#include <stdio.h>

#include <clocale>

int main(void)

{

setlocale(LC\_ALL, "Russian");

int a[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

int max = a[0]; // инициализируем максимальный элемент первым элементом массива

int min = a[0]; // инициализируем минимальный элемент первым элементом массива

int n = 10;

int i;

// Поиск максимального и минимального элементов массива

for (i = 1; i < n; i++) {

if (a[i] > max) {

max = a[i]; // обновляем max, если текущий элемент больше

}

if (a[i] < min) {

min = a[i]; // обновляем min, если текущий элемент меньше

}

}

// Вычисление разницы между максимальным и минимальным элементами

int difference = max - min;

// Вывод результата

printf("Максимальный элемент: %d\n", max);

printf("Минимальный элемент: %d\n", min);

printf("Разница между максимальным и минимальным элементами: %d\n", difference);

return 0;

}

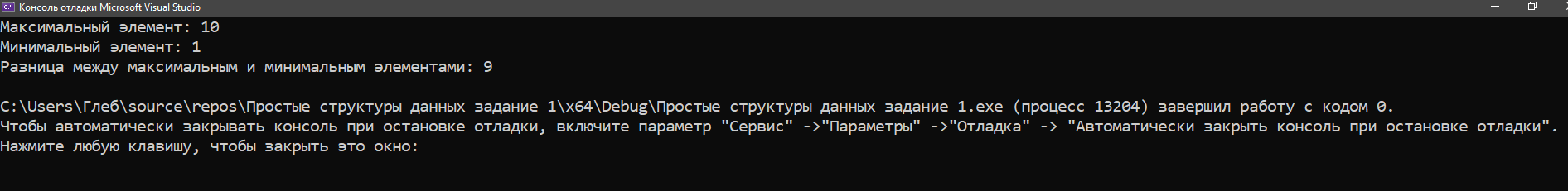


Рисунок 1 – Результат работы программы.

Пробегаемся по элементам массива глазами и видим, что минимальное значение — 1, а максимальное — 10. Переменные min и max имеют эти же значения соответственно, следовательно, программа работает.

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

**Листинг программы:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <locale.h>

int main(void)

{

setlocale(LC\_ALL, "Russian");

int a[10];

int n = 10; // количество элементов массива

int i = 0;

// Создание генератора случайных чисел

srand(time(NULL));

// Заполнение массива случайными числами

while (i < n)

{

a[i] = rand() % 100; // присваиваем каждому элементу случайное число от 0 до 99

i++;

}

// Вывод массива на экран

i = 0;

while (i < n)

{

printf("%d ", a[i]);

i++;

}

printf("\n");

return 0;

}

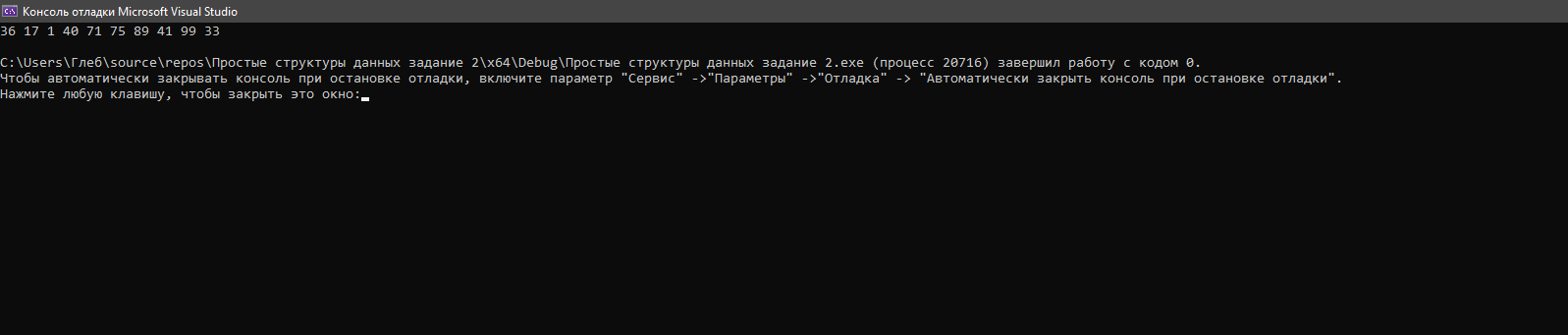


Рисунок 2 – Результат работы программы

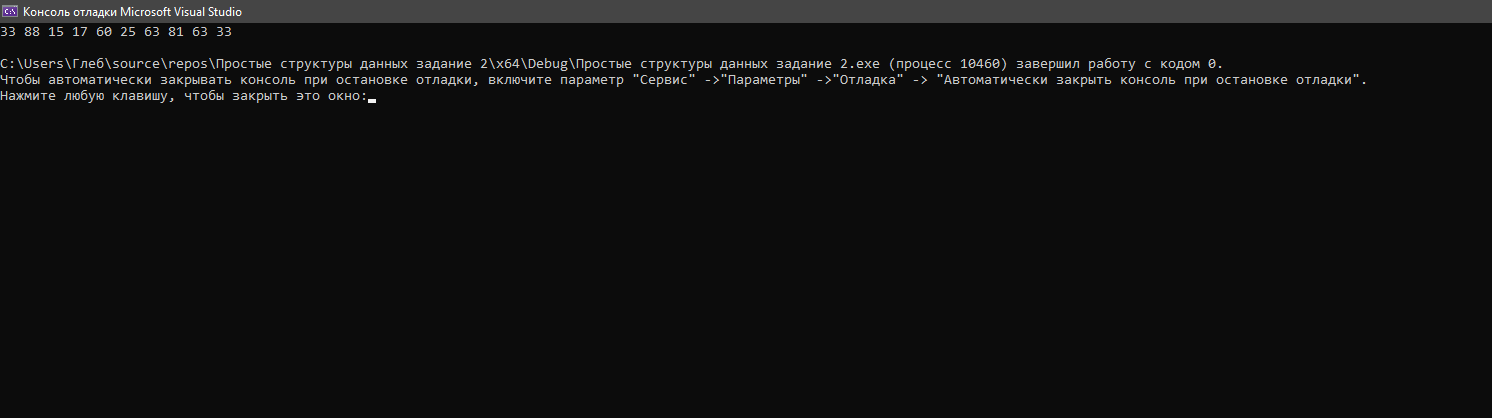


Рисунок 3 – Результат работы программы

При повторном запуске программы, видим, что произошла генерация других случайных чисел. Следовательно, программа работает.

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

**Листинг программы:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main(void)

{

setlocale(LC\_ALL, "Russian");

int n; // переменная для хранения размера массива

int i = 0; // переменная для индексации элементов массива

int\* a; // указатель на массив

printf("Введите размер массива:");

scanf("%d", &n);

// Выделение памяти для массива

a = (int\*)malloc(n \* sizeof(int)); // выделяем память для n элементов типа int

// Проверка выделения памяти

if (a == NULL) {

printf("Ошибка выделения памяти!\n");

return 1; // завершаем программу, если память не была выделена

}

// Заполнение массива числами от 1 до n

while (i < n)

{

a[i] = i + 1; // каждый элемент массива будет равен (индекс + 1)

i++;

}

// Вывод массива на экран

i = 0;

printf("Массив: ");

while (i < n)

{

printf("%d ", a[i]);

i++;

}

printf("\n");

// Освобождение выделенной памяти

free(a);

return 0;

}

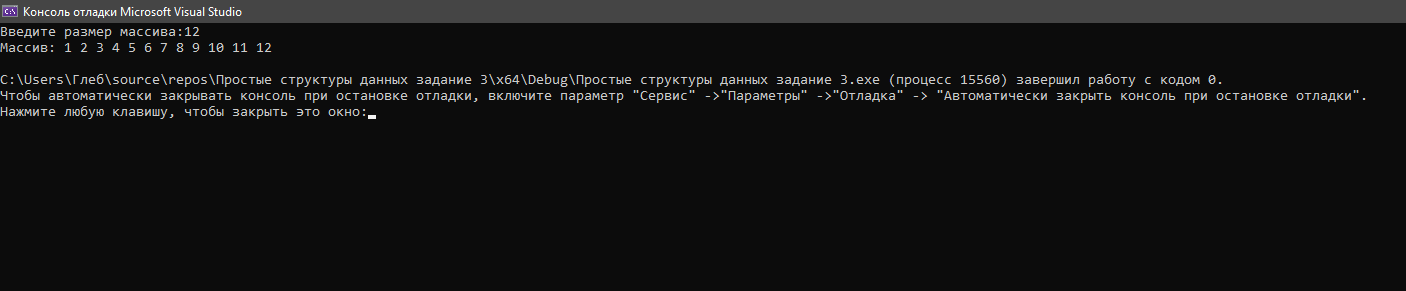


Рисунок 4 – Результат работы программы

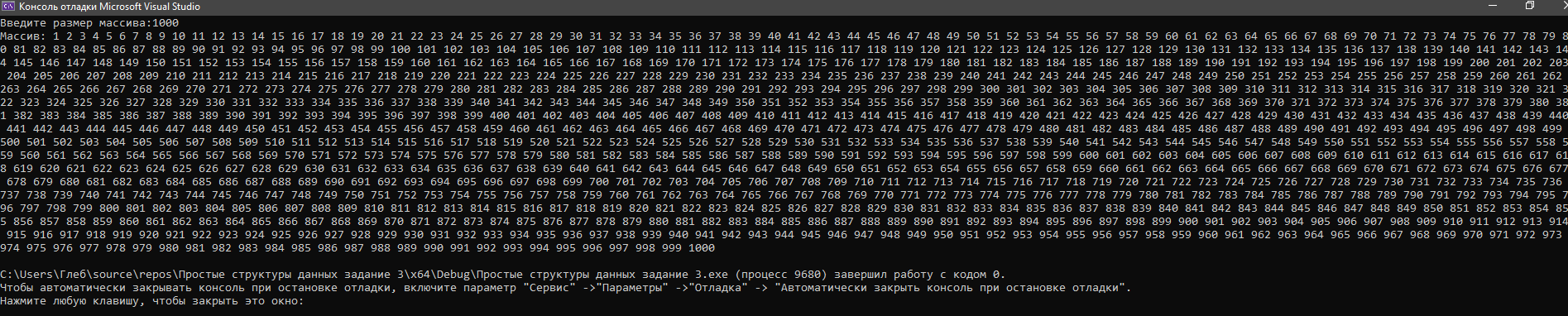


Рисунок 5 – Результат работы программы

Каждый раз задавая новый размер массива, видим, что программа выводит произвольное значение массива. Следовательно, программа работает.

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

**Листинг программы:**

#include <stdio.h>

#include <locale.h>

#define cols 8

int main(void)

{

setlocale(LC\_ALL, "Russian");

int rows = 3; // количество строк

int a[3][4] = { // двумерный массив 3x4

{1, 2, 3, 4},

{5, 6, 7, 8},

{9, 10, 11, 12}

};

int colSum[cols]; // массив для хранения суммы по каждому столбцу

int i, j; // индексы для циклов

// Создание массива для суммы столбцов нулями

for (i = 0; i < cols; i++)

{

colSum[i] = 0;

}

// Вычисление суммы элементов по каждому столбцу

for (i = 0; i < rows; i++)

{

for (j = 0; j < cols; j++)

{

colSum[j] += a[i][j]; // добавляем элемент к сумме соответствующего столбца

}

}

// Вывод суммы по каждому столбцу

for (j = 0; j < cols; j++)

{

printf("Сумма элементов в столбце %d: %d\n", j + 1, colSum[j]);

}

return 0;

}

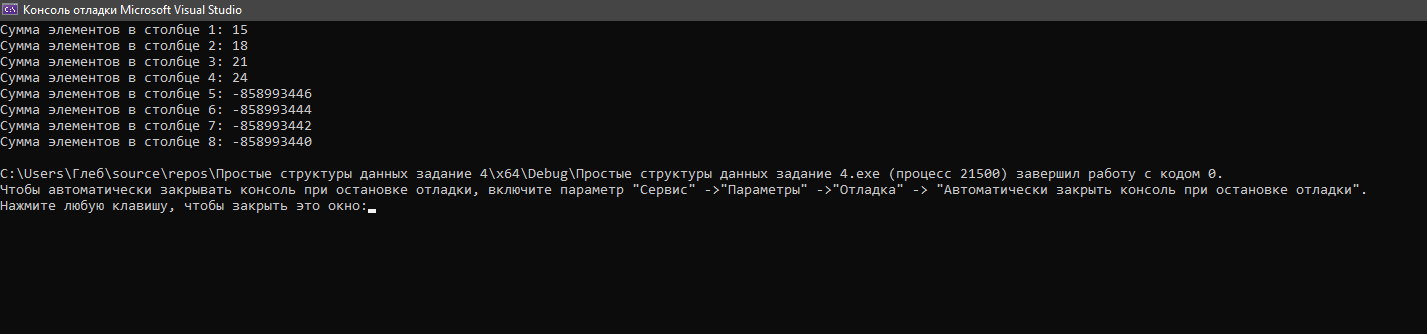


Рисунок 6 – Результат работы программы

Программа работает.

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

**Листинг программы:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <cstring>

using namespace std;

struct student {

char famil[20]; // фамилия

char name[20]; // имя

char facult[20]; // факультет

int Nomzach; // номер зачетной книжки

};

int main() {

setlocale(LC\_ALL, "Russian");

student stud[3];

int i;

// Ввод данных о студентах

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

cout << "Введите фамилию студента: ";

cin >> stud[i].famil;

cout << "Введите имя студента: ";

cin >> stud[i].name;

cout << "Введите название факультета студента: ";

cin >> stud[i].facult;

cout << "Введите номер зачетной книжки студента: ";

cin >> stud[i].Nomzach;

}

// Поиск студента по фамилии

char searchFamil[20];

cout << "\nВведите фамилию для поиска: ";

cin >> searchFamil;

bool found = false;

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

if (strcmp(stud[i].famil, searchFamil) == 0) {

cout << "\nНайден студент: " << stud[i].famil << " " << stud[i].name << "\n";

cout << "Факультет: " << stud[i].facult << "\n";

cout << "Номер зачетной книжки: " << stud[i].Nomzach << "\n";

found = true;

break;

}

}

if (!found) {

cout << "Студент с фамилией " << searchFamil << " не найден.\n";

}

return 0;

}

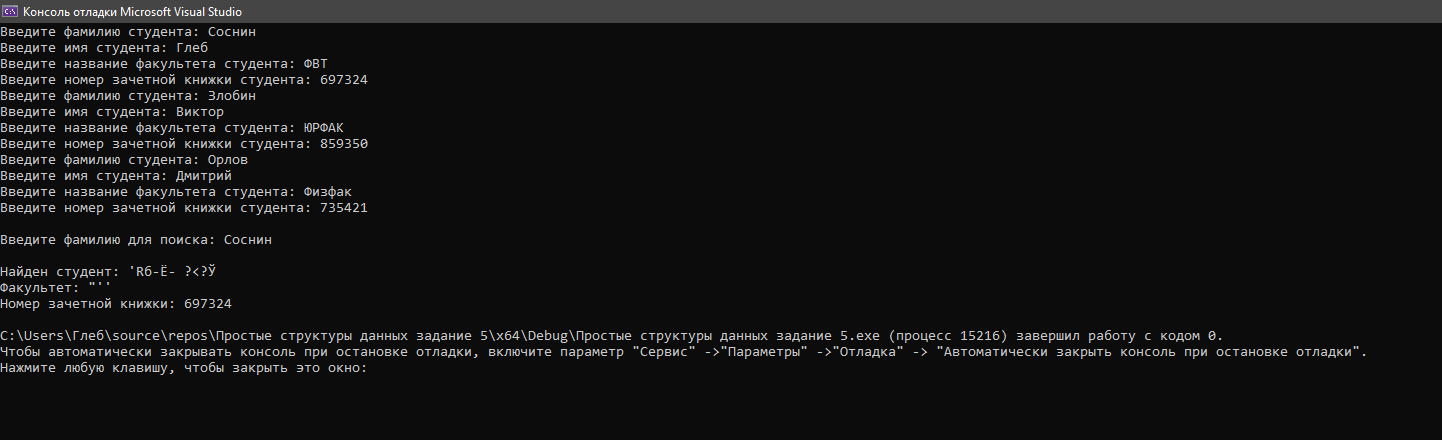


Рисунок 7 – Результат работы программы

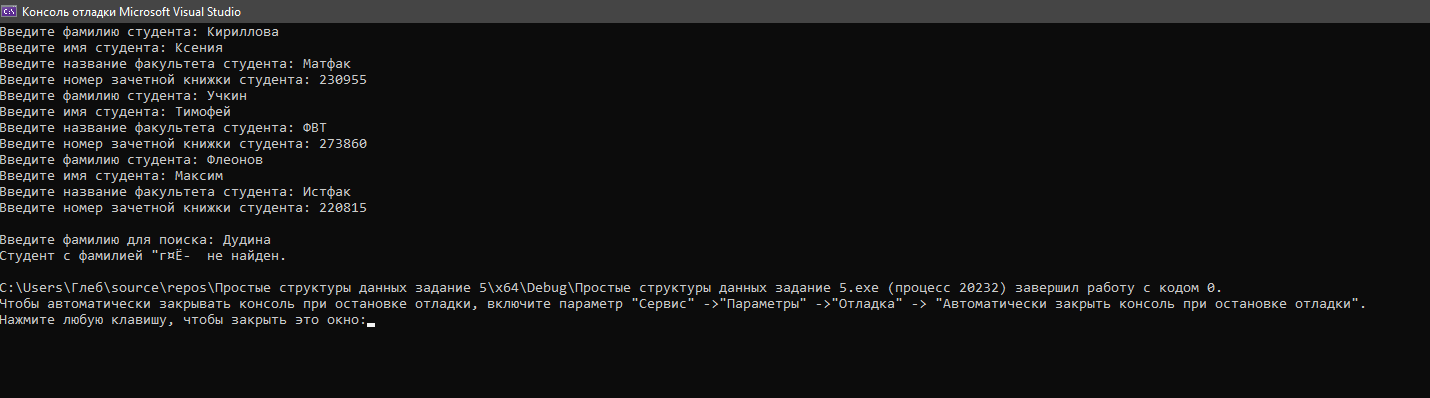


Рисунок 8 – Результат работы программы

Структура *student* содержит поля для хранения фамилии, имени, факультета и номера зачетной книжки.

В массиве *stud* хранятся данные трёх студентов.

Сначала пользователь вводит информацию о студентах.

После этого программа запрашивает фамилию для поиска.

Используется функция *strcmp* для сравнения введенной фамилии с фамилиями студентов в массиве.

Если студент найден, выводится информация о нем. Если нет, выводится сообщение об отсутствии студента с указанной фамилией.

**Вывод:** в ходе выполнения лабораторной работы познакомились с простыми структурами данных языка Си, получили навыки работы с массивами и структурами.