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:**

студенты группы 20ВВ3

Пантюшов Егор

Шмелёв Данила

**Приняли:**

Юрова О.В.

Митрохин М.А.

Пенза 2021

# Название

Простые структуры данных

**Цель работы** – изучение простых структур данных, получение навыков работы с динамическими массивами и генерацией случайных чисел.

# Методические указания

К простым структурам данных языка Си относятся массивы, строки (массивы символов). Структуры (struct) – простейшая составная структура данных.

**Массив** – это конечная совокупность данных одного типа.

Массивы могут состоять из целых чисел, чисел с плавающей запятой, символов и других типов данных. Существуют даже массивы массивов, – многомерные массивы.

Для использования функций динамического распределения памяти необходимо подключение библиотеки &lt;malloc.h&gt; или &lt;stdlib.h&gt; (в зависимости от компилятора и используемого стандарта языка).

Структура - это совокупность переменных одного или нескольких типов,

сгруппированных в один элемент.

Объявление структуры осуществляется с помощью ключевого слова struct, за которым идет ее тип и далее список элементов, заключенных в фигурные скобки:

# Лабораторное задание

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.).

# Листинг

#define \_CRT\_NONSTDC\_NO\_WARNINGS

#define \_CRT\_SECURE\_NO\_WARNINGS

#define HEADER ("Лабораторная работа №1\nВыполнили: Шмелёв Д.В., Пантюшов Е.И.\n")

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include <conio.h>

#include <locale.h>

#include <math.h>

#include <string.h>

#define n 10

int main()

{

setlocale(LC\_ALL, "RUS");

printf(HEADER);

srand(time(NULL));

//Задание 1

/\*int arr[n] = {0,1,2,3,4,5,6,7,8,9};

int min = INT\_MAX, max = INT\_MIN;

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

{

if (arr[i] < min)

{

min = arr[i];

}

if (arr[i] > max)

{

max = arr[i];

}

}

printf("Разница между максимальным и минимальным элементов массива равна %d\n", max - min);\*/

//Задание 2

/\*int arr[n];

printf("Массив, составленный из рандомных чисел:");

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

{

arr[i] = rand() % 100;

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

}\*/

//Задание 3

/\*int size = 0;

printf("Введите размерность массива: ");

scanf("%d", &size);

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

printf("Заполните массив:\n");

for (int i = 0; i < size; i++)

{

printf("%d элемент: ", i + 1);

scanf("%d", &arr[i]);

}

printf("Полученный массив:");

for (int i = 0; i < size; i++)

{

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

}

free(arr);\*/

//Задание 4

/\*int arr[n][n];

int sum = 0;

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

{

for (int j = 0; j < n; j++)

{

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

printf(" %-3d ", arr[i][j]);

}

printf("\n");

}

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

{

for (int j = 0; j < n; j++)

{

sum = sum + arr[i][j];

}

printf("Сумма %d-ой строки равна %d\n", i + 1, sum);

sum = 0;

}\*/

//Задание 5//

setvbuf(stdin, NULL, \_IONBF, 0);

setvbuf(stdout, NULL, \_IONBF, 0);

int i ;

struct student

{

char famil[20];

char name[20], facult[20];

int Nomzach;

} stud[3],vvod;

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

{

printf("Введите фамилию студента\n"); scanf("%20s", stud[i].famil);

}

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

{

printf("Введите имя студента %s\n", stud[i].famil); scanf("%20s", stud[i].name);

}

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

{

printf("Введите название факультета студента %s %s\n", stud[i].famil, stud[i].name); scanf("%20s", stud[i].facult);

}

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

{

printf("Введите номер зачётной книжки студента %s %s\n", stud[i].famil, stud[i].name); scanf("%d", &stud[i].Nomzach);

}

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

{

printf("Cтудент %s %s обучается на факультете %s, номер зачётной книжки %d \n", stud[i].famil, stud[i].name,

stud[i].facult, stud[i].Nomzach);

};

printf("Введите фамилию студента\n"); scanf("%20s", vvod.famil);

printf("Введите имя студента %s\n", vvod.famil); scanf("%20s", vvod.name);

printf("Введите название факультета студента %s %s\n", vvod.famil, vvod.name); scanf("%20s", vvod.facult);

printf("Введите номер зачётной книжки студента %s %s\n", vvod.famil, vvod.name); scanf("%d", &vvod.Nomzach);

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

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

if (strcmp(stud[i].name, vvod.name) == 0) {

if (strcmp(stud[i].facult, vvod.facult) == 0) {

if (stud[i].Nomzach == vvod.Nomzach)

printf("Cтудент %s %s обучается на факультете %s, номер зачётной книжки %d \n", stud[i].famil, stud[i].name,

stud[i].facult, stud[i].Nomzach);

}

}

}

}

getch();

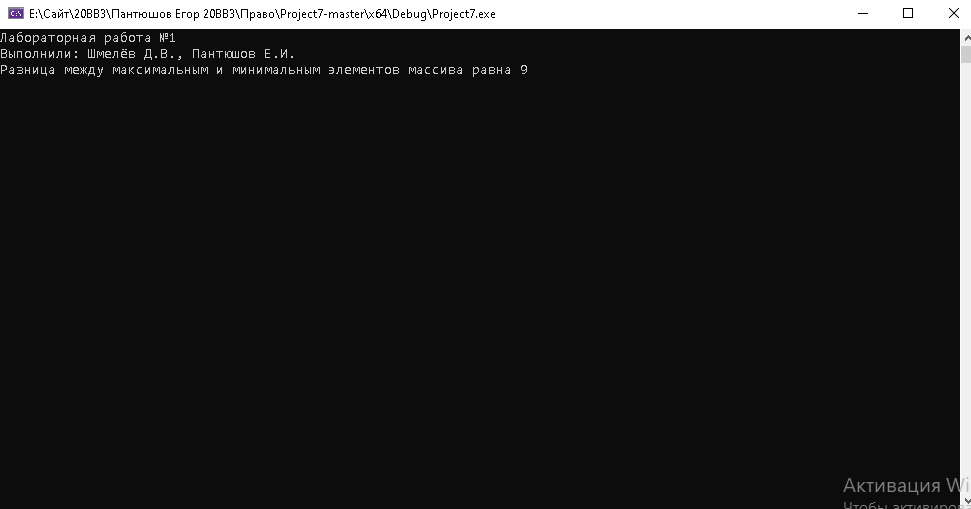
return 0;

}

# Результат работы программы

**Задание 1.**

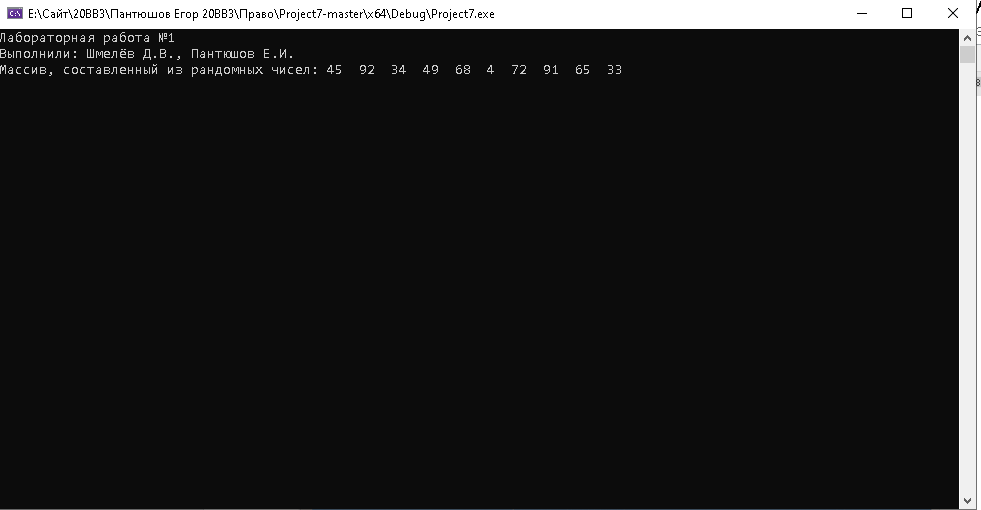
Результаты работы программы показаны на рисунке 1.



**Рисунок 1 – Результат работы программы**

**Задание 2.**

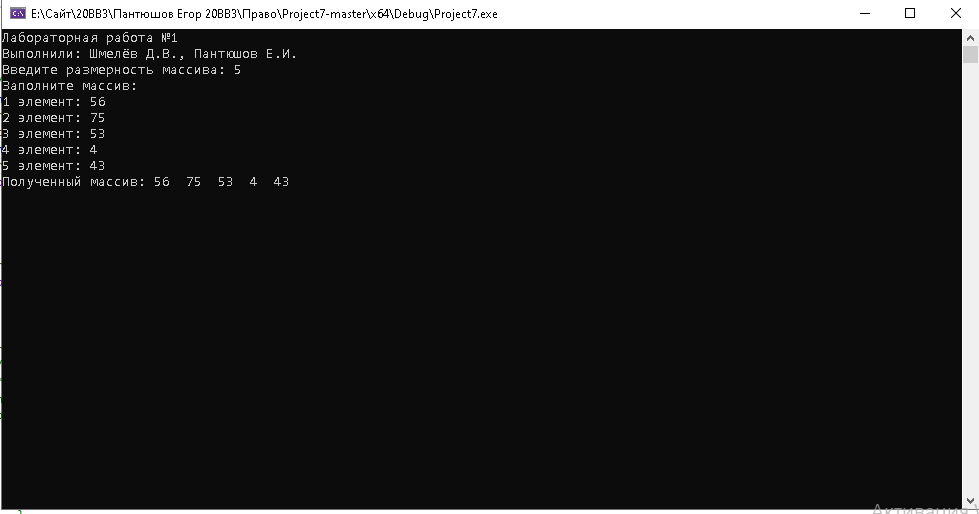
Результаты работы программы показаны на рисунке 2.



**Рисунок 2 – Результат работы программы**

**Задание 3.**

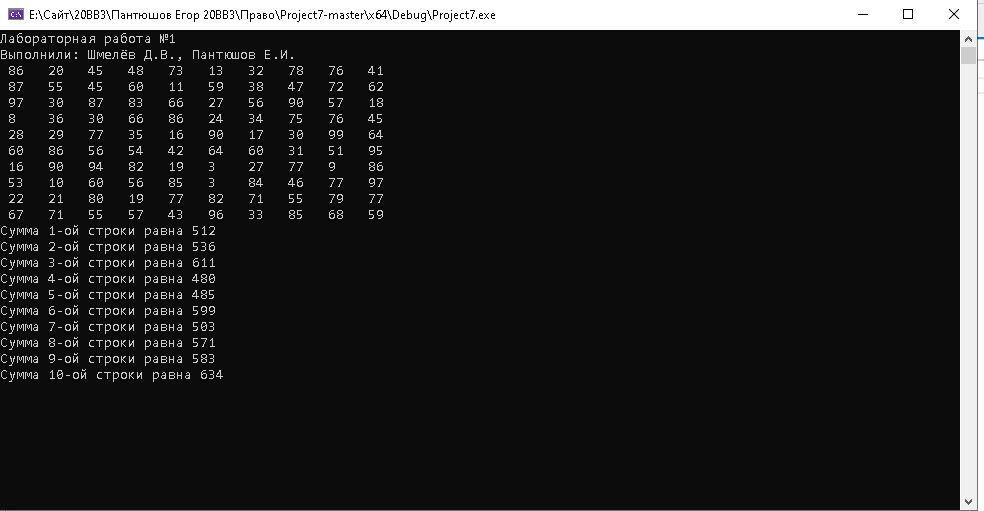
Результаты работы программы показаны на рисунке 3.



**Рисунок 3 – Результат работы программы**

**Задание 4.**

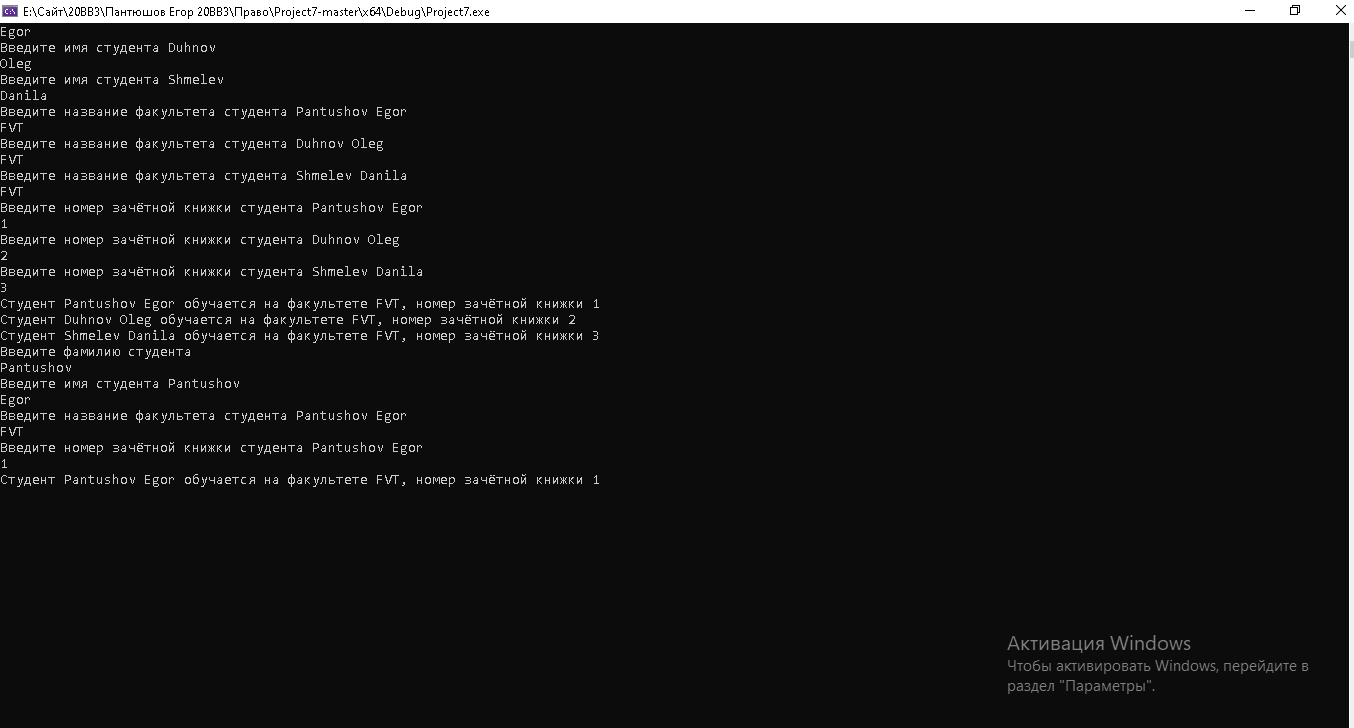
Результаты работы программы показаны на рисунке 4.



**Рисунок 4 – Результат работы программы**

**Задание 5.**

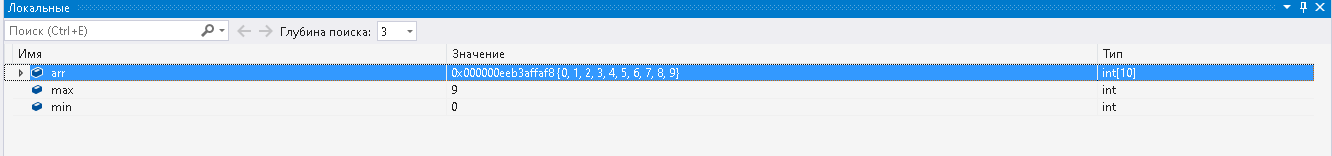
Результаты работы программы показаны на рисунке 5.



**Рисунок 5 – Результат работы программы**

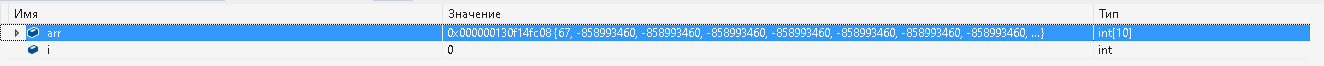
# Протокол трассировка программы

На рисунке 6 показан протокол трассировки после выполнения задания 1



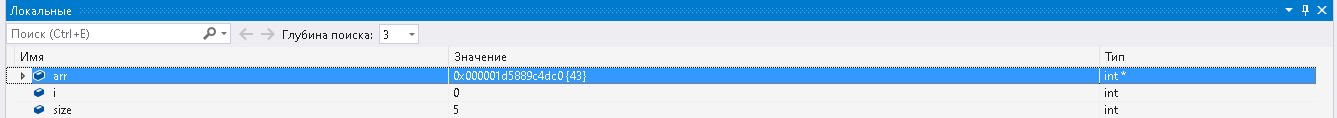
**Рисунок 6 – Протокол трассировки**

На рисунке 7 показан протокол трассировки после выполнения задания 2



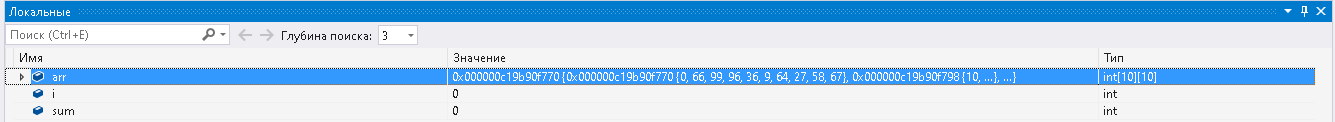
**Рисунок 7– Протокол трассировки**

На рисунке 8 показан протокол трассировки после выполнения задания 3



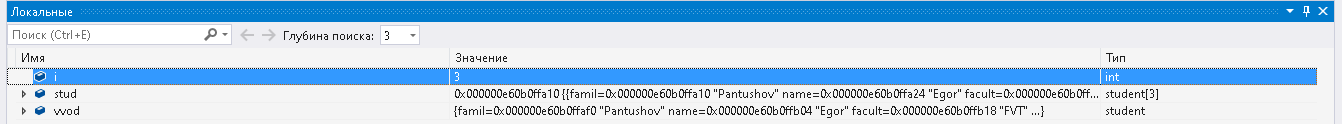
**Рисунок 8– Протокол трассировки**

На рисунке 9 показан протокол трассировки после выполнения задания 4



**Рисунок 9– Протокол трассировки**

На рисунке 10 показан протокол трассировки после выполнения задания 5



**Рисунок 10– Протокол трассировки**

# Вывод

В ходе выполнения лабораторной работы была разработана программа, в ходе работы над которой были изучены простые структуры данных, навыки работы с динамическими массивами и генерацией случайных чисел.