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

Китаев Андрей

Новиков Иван

Халилов Алмаз

**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 элементом 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, инициализирующую двумерный массив.

Задание 5: написать программу, осуществляющую поиск среди структур student структуру с заданным параметром.

**Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

//#include "stdafx.h"

#include <Windows.h>

#include <locale.h>

#include <time.h>

#include <malloc.h>

#include <stdlib.h>

#include <string.h>

//Git test :)

void task1()

{

int min = 100;

int max = 0;

int mas[10] = { 10, 20, 25, 23, 12, 11, 25, 14, 9, 6 };

printf("Задание 1:\n");

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

{

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

if (mas[i] > max)

max = mas[i];

if (mas[i] < min)

min = mas[i];

}

printf("\nРазница между max и min = %d\n", max - min);

}

void task2()

{

srand(time(NULL));

printf("Задание 2:\n");

int mas[10] = { 0 };

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

{

mas[i] = rand() % 100;

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

}

printf("\n");

}

void task3()

{

srand(time(NULL));

printf("Задание 3:\n");

int\* mas2;

int n = 0;

printf("Введите кол-во элементов: ");

scanf("%d", &n);

mas2 = (int\*)malloc(n \* sizeof(int));

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

{

mas2[i] = rand() % 100;

printf("%d ", mas2[i]);

}

free(mas2);

printf("\n");

}

void task4()

{

srand(time(NULL));

printf("Задание 4:\n");

printf("Динам. матрица\n");

int\*\* mas;

int strok, stolb, i, j;

printf("Введите кол-во строк: ");

scanf("%d", &strok);

printf("Введите кол-во столбцов: ");

scanf("%d", &stolb);

mas = (int\*\*)malloc(strok \* sizeof(int\*));

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

{

mas[i] = (int\*)malloc(stolb \* sizeof(int));

}

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

{

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

{

mas[i][j] = rand() % 99;

}

}

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

{

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

{

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

}

printf("\n");

}

}

void task5()

{

printf("Задание 5:\n");

int i;

struct student

{

char famil[40];

char name[40];

char facult[40];

int Nomzach;

} stud[3];

int kolv = 0;

printf("Введите количество студентов, которое хотите добавить\n»>\n");

scanf("%d", &kolv);

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

{

printf("Введите фамилию студента\n");

scanf("%20s", &stud[i].famil);

OemToCharA(stud[i].famil, stud[i].famil);

printf("Введите имя студента\n");

scanf("%20s", &stud[i].name);

OemToCharA(stud[i].name, stud[i].name);

printf("Введите название факультета студента\n");

scanf("%20s", &stud[i].facult);

OemToCharA(stud[i].facult, stud[i].facult);

printf("Введите номер зачетной книжки студента\n");

scanf("%20d", &stud[i].Nomzach);

}

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

{

printf("Студент %s %s обучается на факультете %s, номер зачётной книжки %d\n", stud[i].famil, stud[i].name, stud[i].facult, stud[i].Nomzach);

}

char sl[20];

printf("Поиск \n");

printf(">\a");

scanf("%s", sl);

OemToCharA(sl, sl);

printf("\n");

short chet = 0;

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

{

if (strcmp(sl, stud[i].famil) == 0 || strcmp(sl, stud[i].name) == 0 || strcmp(sl, stud[i].facult) == 0)

{

printf("Студент %s %s обучается на факультете %s, номер зачётной книжки %d\n", stud[i].famil, stud[i].name, stud[i].facult, stud[i].Nomzach);

chet++;

}

}

if (chet == 0)

printf("|\_\_\_\_\_\_\_Ничего не найдено\_\_\_\_\_\_\_|\n");

system("pause");

}

int main(void)

{

setlocale(LC\_ALL, "Russian");

task1();

printf("\n");

task2();

printf("\n");

task3();

printf("\n");

task4();

printf("\n");

task5();

printf("\n");

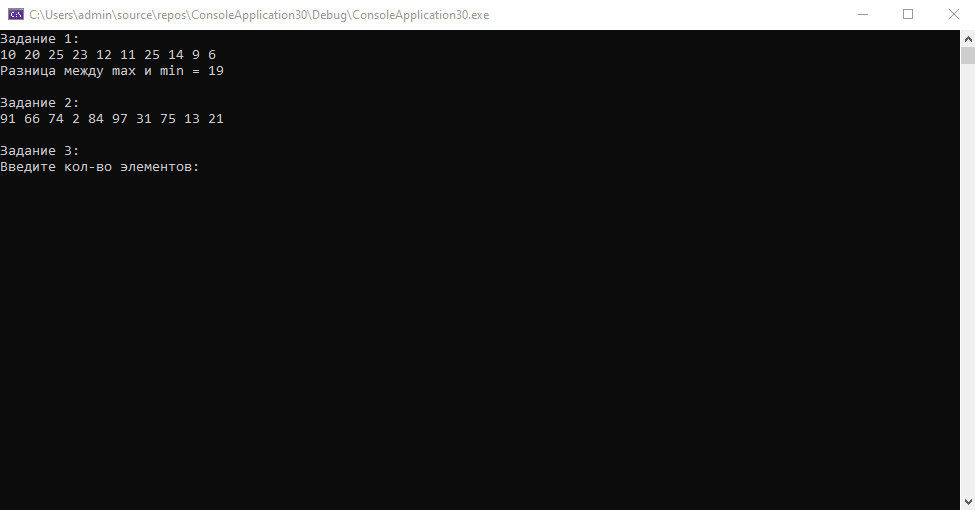
system("pause");

}

# **Результат работы программы**

**Задание 1 и 2.**

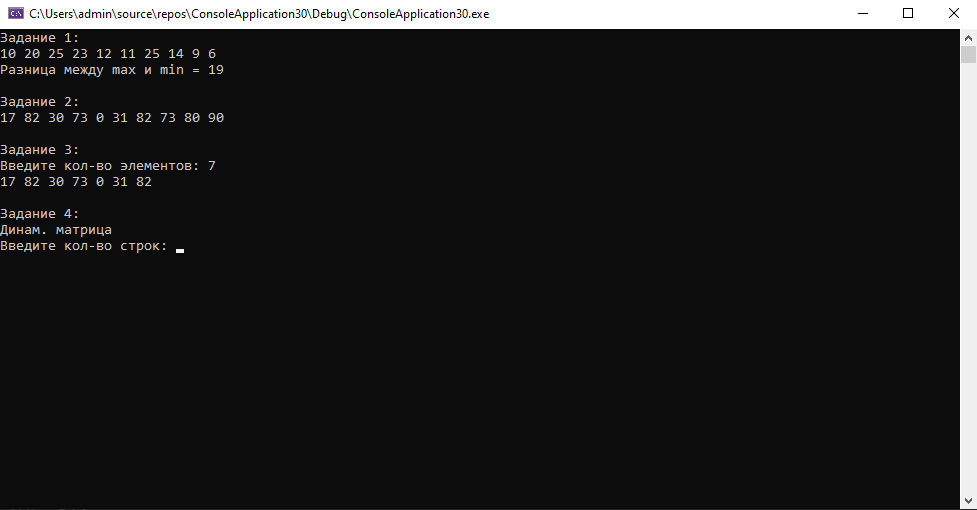
Результаты работы программы показаны на рисунке 1.



*Рисунок 1 – Результат работы программы*

**Задание 3.**

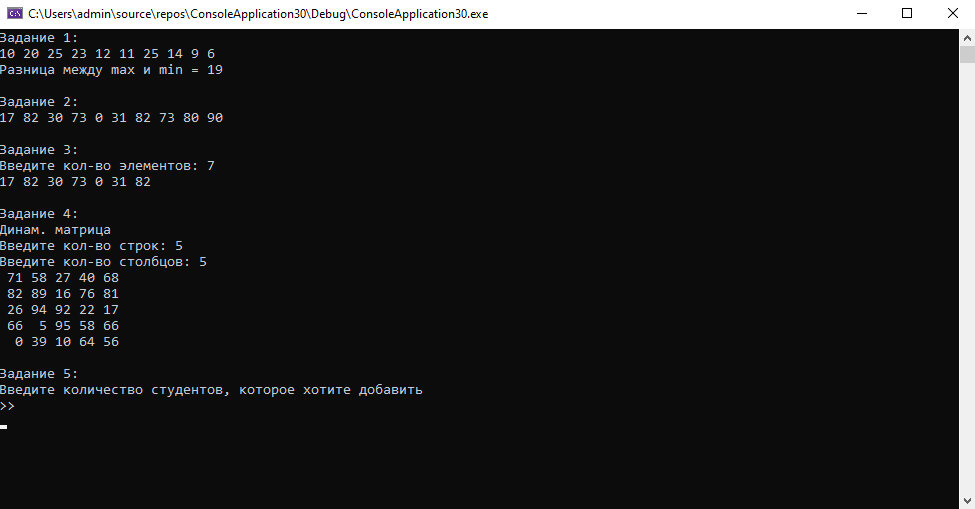
Результаты работы программы показаны на рисунке 2.



*Рисунок 2 – Результат работы программы*

**Задание 4.**

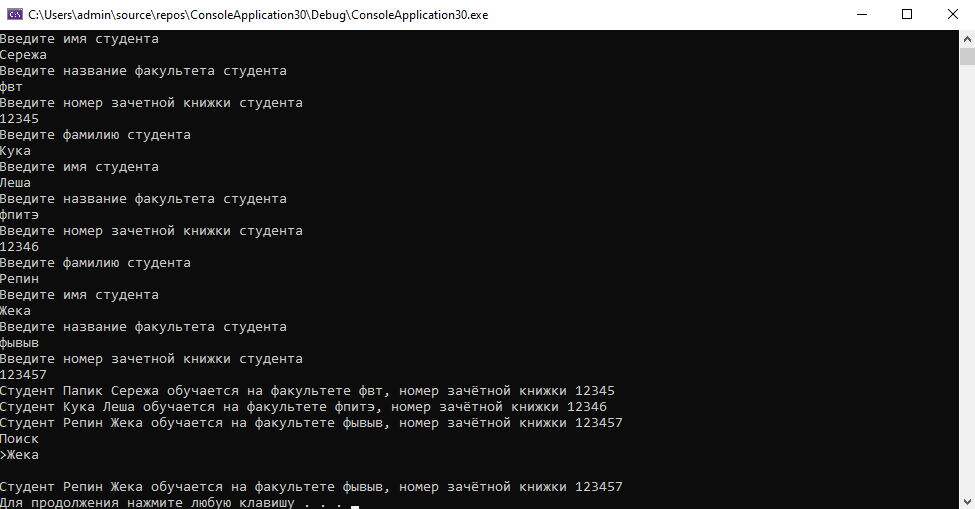
Результаты работы программы показаны на рисунке 3.



*Рисунок 3 – Результат работы программы*

**Задание 5.**

Результаты работы программы показаны на рисунке 4.



*Рисунок 4 – Результат работы программы*

**Вывод**

В ходе выполнения лабораторной работы изучили простые структуры данных, динамические массивы, генератор случайных чисел.