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

Баулин Александр

Культиясов Ярослав

**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, структуру с заданными параметрами (фамилией, именем и т.д.).

# Листинг

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#include <conio.h>

#include <string.h>

int e1()

{

int max = 0, min, razn;

int a[5];

printf("\n");

for (int i = 0; i < 5; i++)

{

printf("vvedite chislo \n");

scanf\_s("%d", &a[i]);

printf("\n");

if (i == 0)

{

min = a[i];

max = a[i];

}

else

{

if (a[i] < min)

{

min = a[i];

}

if (a[i] > max)

{

max = a[i];

}

}

}

razn = max - min;

printf("min=%d max=%d raznica=%d", min, max, razn);

return 0;

}

int e2()

{

int mas[20];

printf("\n");

printf("rand mass\n");

for (int i = 0; i < 20; i++)

{

mas[i] = rand()%100;

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

}

printf("\n");

return 0;

}

int e3()

{

int\* a;

int i, n;

printf("\n");

printf("razmer massiva = ");

scanf\_s("%d", &n);

printf("\n");

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

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

{

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

scanf\_s("%d", &a[i]);

}

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

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

free(a);

return 0;

}

int e4()

{

int a[3][3];

int i, j, x, sum=0, st;

printf("\n");

printf("massiv\n");

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

{

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

{

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

scanf\_s("%d", &a[i][j]);

}

}

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

{

printf("\n");

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

{

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

}

}

printf("\n");

printf("1 dlya sum strok\n2 dlya sum stolbcov\n");

scanf\_s("%d", &x);

if (x == 1)

{

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

{

sum = 0;

printf("sum %d stroki ", i+1);

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

{

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

}

printf("= %d \n", sum);

}

}

if (x == 2)

{

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

{

sum = 0;

printf("sum %d stolbca ", j+1);

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

{

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

}

printf("= %d \n", sum);

}

}

return 0;

}

struct people {

char name[20], facult[20];

};

struct node {

node\* next;

people value;

};

struct list {

node\* first;

int size;

};

list\* create() {

list\* students = (list\*)malloc(sizeof(list));

students->first = NULL;

students->size = 0;

return students;

}

void add(list\* students, people value) {

node\* newStud = (node\*)malloc(sizeof(node));

newStud->next = NULL;

newStud->value = value;

if (students->size == 0) {

students->first = newStud;

students->size++;

return;

}

node\* last = students->first;

while (last->next != NULL) {

last = last->next;

}

last->next = newStud;

students->size++;

}

void print(list\* students) {

node\* current = students->first;

while (current != NULL) {

people value = current->value;

printf("%s %s\n", value.facult, value.name);

current = current->next;

}

}

void search(list\*students) {

printf("\n");

printf("vvedite facult and name\n");

people find;

scanf("%s %s", &find.facult, &find.name);

node\* current = students->first;

while (current != NULL) {

people search = current->value;

if ((strcmp(find.facult, search.facult)==0) && (strcmp(find.name, search.name)==0))

printf("%s %s", search.facult, search.name);

current = current->next;

}

}

int e5() {

list\* students = create();

printf("\n");

printf("vvedite facult and name\n\n");

char buffer[255];

do {

gets\_s(buffer);

if (buffer[0] == '\*')

break;

people stud;

sscanf(buffer, "%s %s", &stud.facult, &stud.name);

add(students, stud);

} while (true);

print(students);

search(students);

return 0;

}

int main()

{

int\* a;

int n;

char wr;

printf("nomer zadaniya\n");

wr = \_getche();

switch (wr) {

case '1':

e1();

break;

case '2':

e2();

break;

case '3':

e3();

break;

case '4':

e4();

break;

case '5':

e5();

break;

}

}

# Результат работы программы

**Задание 1.**

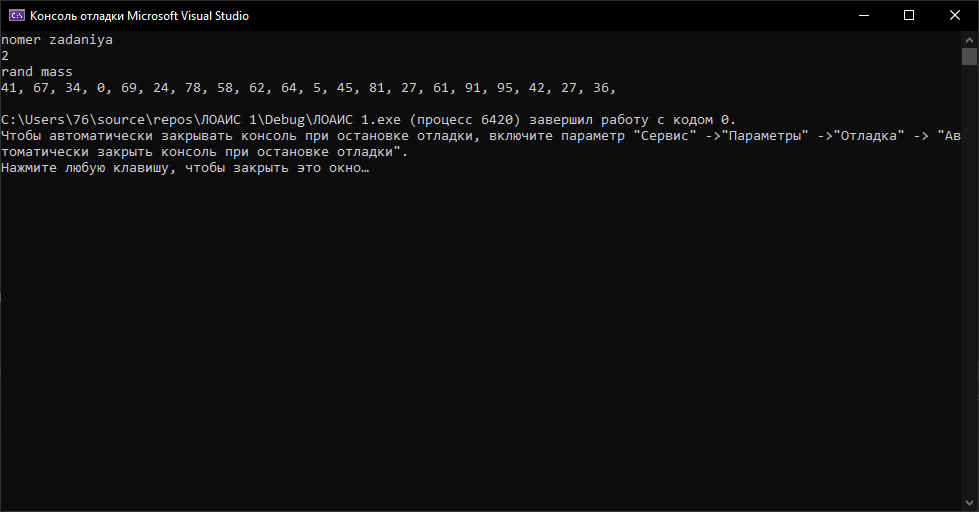
Результаты работы программы показаны на рисунке 1.



**Рисунок 1 – Результат работы программы**

**Задание 2.**

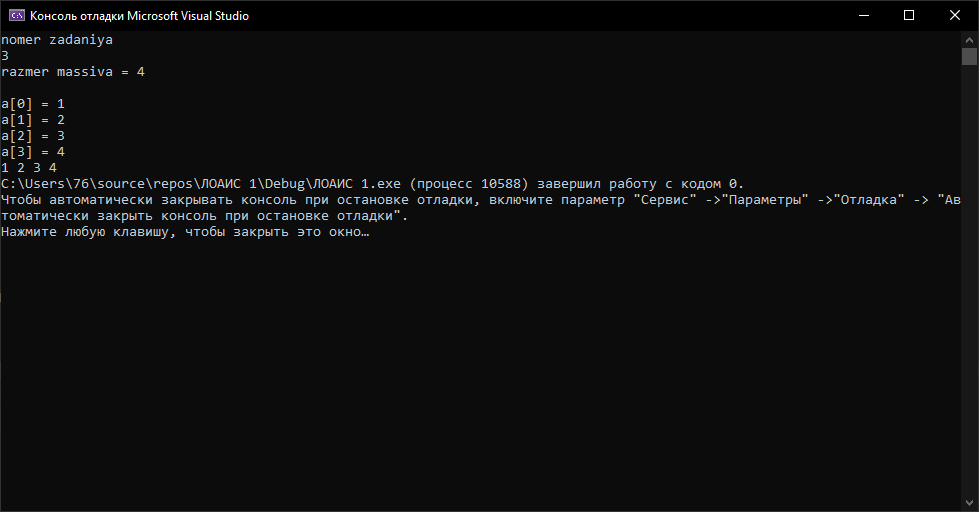
Результаты работы программы показаны на рисунке 2.



**Рисунок 2 – Результат работы программы**

**Задание 3.**

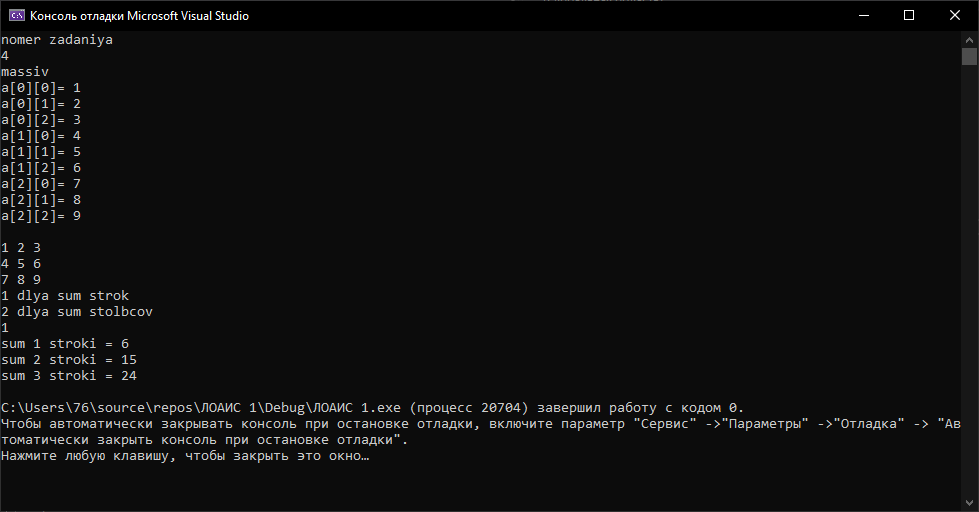
Результаты работы программы показаны на рисунке 3.



**Рисунок 3 – Результат работы программы**

**Задание 4.**

Результаты работы программы показаны на рисунке 4.



**Рисунок 4 – Результат работы программы**

**Задание 4.**

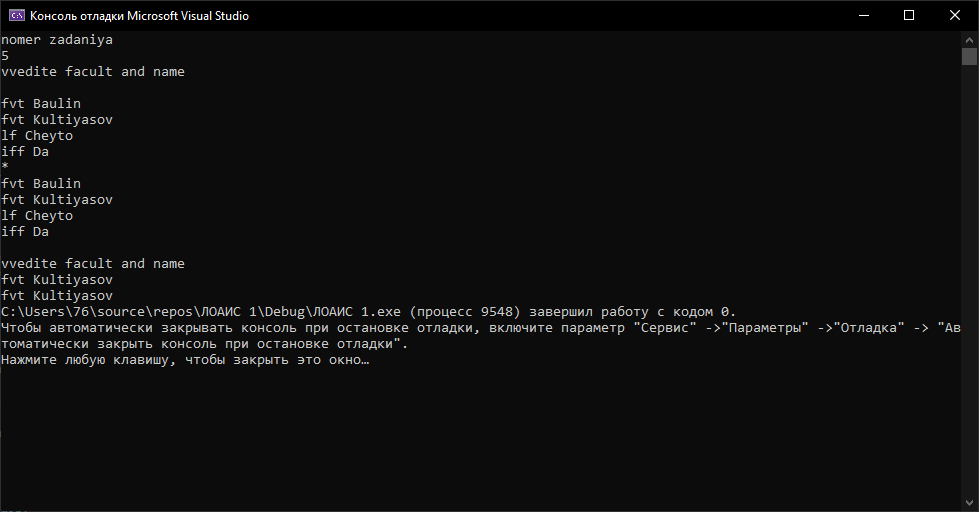
Результаты работы программы показаны на рисунке 5.



**Рисунок 5 – Результат работы программы**

**Задание 5.**

Результаты работы программы показаны на рисунке 6.



**Рисунок 6 – Результат работы программы**

# Вывод

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