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:

студенты групп 24ВВВ3

Масюк А. Р.

Мамонтов Н. П.

Приняли:

Юрова О. В.

Деев М. В.

Пенза 2025

**Цель работы**

Повторить на практике базовые принципы работы с простыми структурами данных в языке С: статическими и динамическими массивами, а также структурами. Закрепить навыки управления памятью, обработки данных и организации программного кода.

**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.).

**Работа программы**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.**

Выполнение заданий 1–3

**Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.**

Выполнение задания 4

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Выполнение задания 5

**Вывод**

В ходе выполнения данной лабораторной работы мы повторили на практике базовые принципы работы с простыми структурами данных в языке С: статическими и динамическими массивами, а также структурами. Закрепили навыки управления памятью, обработки данных и организации программного кода.Написали рабочий код в соответствии с заданием на языке С++

**Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <locale.h>

#include <string.h>

#define ROWS 3

#define COLS 4

struct student

{

char famil[20];

char name[20];

char facult[20];

int Nomzach;

};

int main(void)

{

setlocale(LC\_ALL, "Russian");

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

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

printf("ЗАДАНИЕ 1:\n\n");

int n = 10;

int a[10];

int i;

srand(time(NULL));

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

{

a[i] = rand() % 100;

}

printf("Сгенерированный массив: ");

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

{

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

}

printf("\n");

int min = a[0];

int max = a[0];

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

{

if (a[i] < min)

min = a[i];

if (a[i] > max)

max = a[i];

}

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

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

printf("Разница между максимальным и минимальным: %d\n", max - min);

printf("\nЗАДАНИЕ 2:\n\n");

int b[10];

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

{

b[i] = rand() % 100;

}

printf("Массив случайных чисел:\n");

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

{

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

}

printf("\n");

printf("\nЗАДАНИЕ 3:\n\n");

int size;

int\* dynamic\_array;

printf("Введите размер массива: ");

scanf("%d", &size);

dynamic\_array = (int\*)malloc(size \* sizeof(int));

if (dynamic\_array == NULL)

{

printf("Ошибка выделения памяти!\n");

}

else

{

printf("Введите %d элементов массива:\n", size);

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

{

printf("Элемент %d: ", i + 1);

scanf("%d", &dynamic\_array[i]);

}

printf("Введенный массив:\n");

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

{

printf("%d ", dynamic\_array[i]);

}

printf("\n");

free(dynamic\_array);

}

while (getchar() != '\n');

printf("\nЗАДАНИЕ 4:\n\n");

int matrix[ROWS][COLS] = {

{1, 2, 3, 4},

{5, 6, 7, 8},

{9, 10, 11, 12}

};

int col\_sums[COLS] = { 0 };

int j;

printf("Исходная матрица:\n");

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

{

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

{

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

}

printf("\n");

}

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

{

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

{

col\_sums[j] += matrix[i][j];

}

}

printf("\nСуммы по столбцам:\n");

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

{

printf("Столбец %d: %d\n", j + 1, col\_sums[j]);

}

printf("\nЗАДАНИЕ 5:\n\n");

struct student stud[3];

int found = 0;

char search\_famil[20];

char search\_name[20];

int search\_nomzach;

int search\_choice;

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

{

printf("Введите фамилию студента %d: ", i + 1);

scanf("%19s", stud[i].famil);

}

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

{

printf("Введите имя студента %s: ", stud[i].famil);

scanf("%19s", stud[i].name);

}

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

{

printf("Введите название факультета студента %s %s: ", stud[i].famil, stud[i].name);

scanf("%19s", stud[i].facult);

}

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

{

printf("Введите номер зачётной книжки студента %s %s: ", stud[i].famil, stud[i].name);

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

}

printf("\nСписок всех студентов:\n");

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

}

printf("\nПоиск студента:\n");

printf("1 - По фамилии\n");

printf("2 - По имени\n");

printf("3 - По номеру зачетной книжки\n");

printf("Выберите вариант поиска: ");

scanf("%d", &search\_choice);

switch (search\_choice)

{

case 1:

printf("Введите фамилию для поиска: ");

scanf("%19s", search\_famil);

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

{

if (strcmp(stud[i].famil, search\_famil) == 0)

{

printf("Найден студент: %s %s, факультет: %s, номер: %d\n",

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

found = 1;

}

}

break;

case 2:

printf("Введите имя для поиска: ");

scanf("%19s", search\_name);

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

{

if (strcmp(stud[i].name, search\_name) == 0)

{

printf("Найден студент: %s %s, факультет: %s, номер: %d\n",

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

found = 1;

}

}

break;

case 3:

printf("Введите номер зачетной книжки: ");

scanf("%d", &search\_nomzach);

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

{

if (stud[i].Nomzach == search\_nomzach)

{

printf("Найден студент: %s %s, факультет: %s, номер: %d\n",

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

found = 1;

}

}

break;

default:

printf("Неверный выбор!\n");

break;

}

if (!found && search\_choice >= 1 && search\_choice <= 3)

{

printf("Студент не найден!\n");

}

system("pause");

return 0;

}