

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 группы

24BVB1

Куничкина В.А.

Суркова Д.А.

Приняли:

к.т.н. Деев М.В.

к.т.н. доцент Юрова О.В.

Цель работы

Освоить основы работы с простыми структурами данных в языке Си: статическими и динамическими массивами, а также структурами (struct). Научиться выполнять базовые операции с массивами (инициализация, обращение, изменение, работа с памятью) и структурами (объявление, инициализация, доступ к полям, поиск). Получить практические навыки использования функций для работы с динамической памятью и генерации случайных чисел.

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

1-3

```
#include "stdafx.h"
#include <conio.h>
#include <locale.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    setlocale(LC_ALL, "Russian");

    printf("Залание 1-3\n");
    int size;
    printf("Введите размер массива: ");
    scanf("%d", &size);
    printf("\n");

    int *a = (int *)malloc(size * sizeof(int));
    srand(time(0));

    for (int i = 0; i < size; i++) {
        a[i] = rand();
        printf("Элемент %d: %d\n", i + 1, a[i]);
    }

    int min=a[0], max=a[0];
    for (int i = 1; i < size; i++){
        if (max < a[i])
            max = a[i];
        if (min > a[i])
            min = a[i];
    }

    printf("\nМаксимальное: %d\nМинимальное: %d\n", max, min);

    free(a);
    getch();
    return 0;
}
```

4

```
#include "stdafx.h"
#include <conio.h>
#include <locale.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main() {
    int rows, cols;

    setlocale(LC_ALL, "Russian");
    printf("Введите количество строк: ");
    scanf("%d", &rows);
    printf("Введите количество столбцов: ");
    scanf("%d", &cols);

    int** mtr = (int**)malloc(rows * sizeof(int*));
    for (int i = 0; i < rows; i++) {
        mtr[i] = (int*)malloc(cols * sizeof(int));
    }

    int* rS = (int*)malloc(rows * sizeof(int));
    int* cS = (int*)malloc(cols * sizeof(int));

    srand((unsigned int)time(NULL));

    for (int i = 0; i < rows; i++) {
        rS[i] = 0;
        for (int j = 0; j < cols; j++) {
            mtr[i][j] = rand() % 100; // числа от 0 до 99
        }
    }

    for (int j = 0; j < cols; j++) {
        cS[j] = 0;
    }
    // Вычисление сумм
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            rS[i] += mtr[i][j];
            cS[j] += mtr[i][j];
        }
    }
    // Вывод матрицы
    printf("Матрица %dx%d:\n", rows, cols);
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%4d", mtr[i][j]);
        }
        printf(" | Сумма строки: %d\n", rS[i]);
    }

    for (int j = 0; j < cols; j++) {
        printf("----");
    }
    printf("\n");
}
```

```

    for (int j = 0; j < cols; j++) {
        printf("%4d", cS[j]);
    }
    printf("\n");

    for (int i = 0; i < rows; i++) {
        free(mtr[i]);
    }
    free(mtr);
    free(rS);
    free(cS);

    getch ();
    return 0;
}

```

5

```

#include "stdafx.h"
#include <conio.h>
#include <Windows.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int main() {
    struct Student {
        char lastName[50];
        char firstName[50];
        int age;
        char group[20];
    };
    SetConsoleOutputCP(1251); //подключение русс яз
    SetConsoleCP(1251);

    struct Student students[5] = {
        {"Иванов", "Иван", 20, "24BBB1"},
        {"Петров", "Петр", 21, "24BBB2"},
        {"Сидорова", "Мария", 19, "24BBB1"},
        {"Кузнецов", "Алексей", 20, "24BBB2"},
        {"Смирнова", "Елена", 22, "24BBB3"}
    };

    int choice = 1;
    int count = 4;

    while (choice != 0) {
        printf("\n=== Система поиска студентов ===\n");
        printf("1. Поиск по фамилии\n");
        printf("2. Поиск по имени\n");
        printf("3. Поиск по возрасту\n");
        printf("4. Поиск по группе\n");
        printf("5. Показать всех студентов\n");
        printf("0. Выход\n");
        printf("Выберите действие: ");
        scanf("%d", &choice);

        char sName[50];
    }
}

```

```

char sFName[50];
int sAge;
char sGroup[20];
int found;
int i;

if (choice == 1) { // По фамилии
    printf("Введите фамилию для поиска: ");
    scanf("%s", sName);

    printf("\nРезультаты поиска по фамилии '%s':\n", sName);
    found = 0;
    for (i = 0; i < count; i++) {
        if (strcmp(students[i].lastName, sName) == 0) {
            printf("Студент %d: %s %s, возраст: %d, группа: %s, средний балл:
%.2f\n",
                    i + 1, students[i].lastName, students[i].firstName,
                    students[i].age, students[i].group);
            found = 1;
        }
    }
}
else if (choice == 2) { // По имени
    printf("Введите имя для поиска: ");
    scanf("%s", sFName);

    printf("\nРезультаты поиска по имени '%s':\n", sFName);
    found = 0;
    for (i = 0; i < count; i++) {
        if (strcmp(students[i].firstName, sFName) == 0) {
            printf("Студент %d: %s %s, возраст: %d, группа: %s, средний балл:
%.2f\n",
                    i + 1, students[i].lastName, students[i].firstName,
                    students[i].age, students[i].group);
            found = 1;
        }
    }
    if (!found) {
        printf("Студенты не найдены!\n");
    }
}
else if (choice == 3) { // По возрасту
    printf("Введите возраст для поиска: ");
    scanf("%d", &sAge);

    printf("\nРезультаты поиска по возрасту '%d':\n", sAge);
    found = 0;
    for (i = 0; i < count; i++) {
        if (students[i].age == sAge) {
            printf("Студент %d: %s %s, возраст: %d, группа: %s, средний балл:
%.2f\n",
                    i + 1, students[i].lastName, students[i].firstName,
                    students[i].age, students[i].group);
            found = 1;
        }
    }
}
else if (choice == 4) { // По группе
    printf("Введите группу для поиска: ");

```

```

scanf("%s", sGroup);

printf("\nРезультаты поиска по группе '%s':\n", sGroup);
found = 0;
for (i = 0; i < count; i++) {
    if (strcmp(students[i].group, sGroup) == 0) {
        printf("Студент %d: %s %s, возраст: %d, группа: %s, средний балл:
%.2f\n",
                i + 1, students[i].lastName, students[i].firstName,
                students[i].age, students[i].group);
        found = 1;
    }
}
}
else if (choice == 5) { // Показать всех студентов
    printf("\nСписок всех студентов:\n");
    for (i = 0; i < count; i++) {
        printf("Студент %d: %s %s, возраст: %d, группа: %s, средний балл:
%.2f\n",
                i + 1, students[i].lastName, students[i].firstName,
                students[i].age, students[i].group);
    }
}
else if (choice == 0) { // Выход
    printf("Выход из программы...\n");
}
}

getch ();
return 0;
}

```

Результаты работы программ

```

D:\Рабочий стол\24vvvv1\2 к
Задание 1-3
Введите размер массива: 14

Элемент 1: 22470
Элемент 2: 25108
Элемент 3: 9541
Элемент 4: 15570
Элемент 5: 31871
Элемент 6: 28013
Элемент 7: 501
Элемент 8: 26834
Элемент 9: 30606
Элемент 10: 12113
Элемент 11: 4688
Элемент 12: 26603
Элемент 13: 21256
Элемент 14: 22232

Максимальное: 31871
Минимальное: 501

```

Задание 4

Введите количество строк: 5

Введите количество столбцов: 6

Матрица 5x6:

41	52	91	55	16	25	Сумма строки: 280
80	7	17	81	16	82	Сумма строки: 283
63	42	66	24	84	66	Сумма строки: 345
57	74	16	50	66	92	Сумма строки: 355
67	70	8	26	15	59	Сумма строки: 245

308 245 198 236 197 324

Задание 5

=== Система поиска студентов ===

1. Поиск по фамилии
2. Поиск по имени
3. Поиск по возрасту
4. Поиск по группе
5. Показать всех студентов
0. Выход

Выберите действие: 2

Введите имя для поиска: Мария

Результаты поиска по имени 'Мария':

Студент 3: Сидорова Мария, возраст: 19, группа: 24BBB1

=== Система поиска студентов ===

1. Поиск по фамилии
2. Поиск по имени
3. Поиск по возрасту
4. Поиск по группе
5. Показать всех студентов
0. Выход

Выберите действие: 0

Выход из программы...

Протокол трассировки программ

```
int *a = (int *)malloc(size * sizeof(int));
srand(time(0));

for (int i = 0; i < size; i++) {
    a[i] = rand();
    printf("Элемент %d: %d\n", i + 1, a[i]);
}

int min=a[0], max=a[0];
for (int i = 1; i < size; i++){
    if (max < a[i])
        max = a[i];
    if (min > a[i])
        min = a[i];
}

printf("\nМаксимальное: %d\nМинимальное: %d\n", max, min);

free(a);
getch();
return 0;
}
```

100 % <

Локальные

Имя	Значение
i	0
max	-858993460
size	6
min	-858993460
a	0x013d78d0
	6460

Заполнение массива в 1-3 задании.

Вывод

В ходе выполнения лабораторной работы были изучены и практически применены основные принципы работы с простыми структурами данных в языке Си:

- Освоены методы объявления, инициализации и обработки статических и динамических массивов.
- Приобретены навыки работы с динамической памятью с использованием функций *malloc*, *calloc* и *free*.
- Изучены основы генерации псевдослучайных чисел с помощью функций *rand* и *srand*.
- Получен опыт работы со структурами: объявление, инициализация, доступ к полям, организация поиска по заданным критериям.

Работа заложила фундамент для дальнейшего изучения более сложных структур данных и алгоритмов, а также продемонстрировала важность управления памятью и организации данных в программировании.

GitHub

Куничкина Вероника

Суркова Дарья