

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем

Лабораторная работа № 11

по дисциплине: Основы программирования

тема: «Использование указателей на функции и создание программы из
нескольких файлов»

Выполнил: ст. группы ПВ202

Аладиб язан

Проверил:

Валентина Станиславовна

Притчин Иван Сергеевич

Белгород 2021

Лабораторная работа № 11

«Использование указателей на функции и создание программы из нескольких файлов»

Цель работы: получение навыков передачи функций функциям в качестве параметров, отдельной компиляции файлов программы и создания модульной программы.

Задания для подготовки к работе :

1. Изучить правила описания и использования указателей на функции в языке Си.
2. Разработать алгоритм и описать функцию для универсальной сортировки произвольного массива с произвольным базовым типом. Функции передается массив как нетипизованный параметр, его длина, размер элемента и логическая функция сравнения двух элементов массива.
3. С использованием этой функции решить следующую задачу. Имеется информация о студентах группы: Ф.И.О., результаты последней экзаменационной сессии. Требуется получить список студентов с указанием среднего балла по итогам сессии, упорядоченный по указанию пользователя либо лексикографически, либо по невозрастанию среднего балла.
4. Функцию универсальной сортировки и необходимые для нее программные объекты описать в отдельном модуле.
5. Подобрать наборы тестовых данных.

Выполнение работы:

Тестовые данные:

Исходные данные	результаты
<p>Введите количество студентов: 3 Введите Ф.И.О. студента 1: Иван Иванов Введите средний балл студента 1: 4.5 Введите Ф.И.О. студента 2: Анна Петрова Введите средний балл студента 2: 4.8 Введите Ф.И.О. студента 3: Сергей Смирн Введите средний балл студента 3: 3.9</p> <p>Сортировать по: 1. Имени 2. Среднему баллу</p> <p>1</p>	<p>Отсортированные студенты: Ф.И.О.: Анна Петрова, Средний балл: 4.80 Ф.И.О.: Иван Иванов, Средний балл: 4.50 Ф.И.О.: Сергей Смирнов, Средний балл: 3.90</p>
<p>Введите количество студентов: 3 Введите Ф.И.О. студента 1: Иван Иванов Введите средний балл студента 1: 4.5 Введите Ф.И.О. студента 2: Анна Петрова Введите средний балл студента 2: 4.8 Введите Ф.И.О. студента 3: Сергей Смирн Введите средний балл студента 3: 3.9</p> <p>Сортировать по: 1. Имени 2. Среднему баллу</p> <p>2</p>	<p>Отсортированные студенты: Ф.И.О.: Анна Петрова, Средний балл: 4.80 Ф.И.О.: Иван Иванов, Средний балл: 4.50 Ф.И.О.: Сергей Смирнов, Средний балл: 3.90</p>

Текст программы:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
// Определяем структуру Student
typedef struct {
```

```

    char name[100];        // Ф.И.О. студента
    float averageScore;    // Средний балл
} Student;

// Универсальная функция сортировки
void universalSort(void* base, size_t nitems, size_t size, int (*compar)(const void*,
const void*)) {
    // Реализация простой сортировки пузырьком для демонстрации
    for (size_t i = 0; i < nitems - 1; i++) {
        for (size_t j = 0; j < nitems - i - 1; j++) {
            void* a = (char*)base + j * size;
            void* b = (char*)base + (j + 1) * size;

            if (compar(a, b) > 0) {
                // Обмен
                char temp[size];
                memcpy(temp, a, size);
                memcpy(a, b, size);
                memcpy(b, temp, size);
            }
        }
    }
}

// Функция сравнения для сортировки по имени
int compareByName(const void* a, const void* b) {
    const Student* studentA = (const Student*)a;
    const Student* studentB = (const Student*)b;
    return strcmp(studentA->name, studentB->name);
}

// Функция сравнения для сортировки по среднему баллу
int compareByAverageScore(const void* a, const void* b) {
    const Student* studentA = (const Student*)a;
    const Student* studentB = (const Student*)b;
    if (studentA->averageScore < studentB->averageScore) return 1; // Сортировка по
невозрастанию
    if (studentA->averageScore > studentB->averageScore) return -1;
    return 0;
}

// Главная функция
int main() {
    int n;
    printf("Введите количество студентов: ");
    scanf("%d", &n);

    Student* students = malloc(n * sizeof(Student));

    // Ввод данных студентов
    for (int i = 0; i < n; i++) {

```

```

        printf("Введите Ф.И.О. студента %d: ", i + 1);
        scanf(" %[^\\n]", students[i].name); // Чтение строки с пробелами
        printf("Введите средний балл студента %d: ", i + 1);
        scanf("%f", &students[i].averageScore);
    }

    int choice;
    printf("Сортировать по:\\n1. Имени\\n2. Среднему баллу\\n");
    scanf("%d", &choice);

    // Сортировка в зависимости от выбора пользователя
    if (choice == 1) {
        universalSort(students, n, sizeof(Student), compareByName);
    } else if (choice == 2) {
        universalSort(students, n, sizeof(Student), compareByAverageScore);
    } else {
        printf("Неверный выбор!\\n");
        free(students);
        return 1;
    }

    // Вывод отсортированных студентов
    printf("\\nОтсортированные студенты:\\n");
    for (int i = 0; i < n; i++) {
        printf("Ф.И.О.: %s, Средний балл: %.2f\\n", students[i].name,
students[i].averageScore);
    }

    free(students);
    return 0;
}

```