### 

КОМИТЕТ ПО ОБРАЗОВАНИЮ САНКТ-ПЕТЕРБУРГА

**САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

**ОТЧЕТ**

**Основы алгоритмизации и программирования**

**Практическая работа 8. Сортировка структур по указателям**

Работу выполнил: Танасов Евгений Эдуардович

Группа:324 Специальность: 09.02.07

Преподаватель: Смирнова Ирина Петровна

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«22» ноября 2024 г.

Преподаватель: Смирнова Ирина Петровна

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» сентября 2021 г.

Санкт-Петербург 2021

Санкт-Петербург 2015 г.

Оглавление

[**Цель практической работы:** 3](#_Toc181912877)

[**Исходный текст программы:** 3](#_Toc181912878)

[Результат работы программы**:** 4](#_Toc181912879)

# **Цель практической работы:**

Разобраться в построении структуры и в коде объяснить как она работает.

# **Исходный текст программы:**

#include <stdio.h> // Библиотека для ввода-вывода (printf, fopen, fread).

#include <string.h> // Библиотека для работы со строками (strcmp используется для сравнения строк).

#include <locale.h> // Библиотека для настройки локализации (setlocale).

#include <cstdlib> // Библиотека для системных вызовов (std::system).

/\*

Программа № 3. Сортировка структур по указателям

\*/

typedef struct { // Определение структуры для хранения информации о книге.

char author[40]; // Автор книги (строка до 39 символов + '\0').

char title[80]; // Название книги (строка до 79 символов + '\0').

int count; // Количество экземпляров книги.

} TBook;

typedef TBook\* PBook; // Определяется тип указателя на структуру `TBook`.

int main() {

const int N = 100; // Задаётся максимальный размер массива книг.

TBook Books[N]; // Массив структур для хранения информации о книгах.

PBook p[N], p1; // Массив указателей на структуры и временный указатель для сортировки.

int i, j, M; // Счётчики циклов и переменная для хранения количества книг.

FILE\* F; // Указатель на файл для работы с бинарными данными.

for (i = 0; i < N; i++) p[i] = &Books[i]; // Инициализация массива указателей, каждый указывает на соответствующий элемент массива `Books`.

F = fopen("books.dat", "rb"); // Открытие файла "books.dat" в бинарном режиме для чтения.

M = fread(&Books[0], sizeof(TBook), N, F); // Считывание до N структур `TBook` из файла в массив `Books`. Возвращается количество прочитанных структур.

fclose(F); // Закрытие файла.

printf("Имеющиеся данные о книгах:\n");

for (i = 0; i < M; i++) // Вывод всех книг, считанных из файла.

printf("%s, %s, %d\n", Books[i].author, Books[i].title, Books[i].count);

// Сортировка массива указателей на структуры книг по имени автора.

for (i = 0; i < M - 1; i++) { // Внешний цикл сортировки (метод пузырька).

for (j = M - 2; j >= i; j--) { // Внутренний цикл сортировки (проход по неотсортированным элементам).

if (strcmp(p[j]->author, p[j + 1]->author) > 0) { // Сравнение имён авторов книг через указатели.

p1 = p[j]; // Временное сохранение указателя на меньший элемент.

p[j] = p[j + 1]; // Перестановка указателей.

p[j + 1] = p1; // Завершение обмена указателей.

}

}

}

printf("\nПосле сортировки через указатели:\n");

for (i = 0; i < M; i++) // Вывод книг после сортировки (через массив указателей).

printf("%s, %s, %d\n", p[i]->author, p[i]->title, p[i]->count);

getchar(); // Ожидание ввода пользователя перед завершением программы.

}

# Результат работы программы**:**

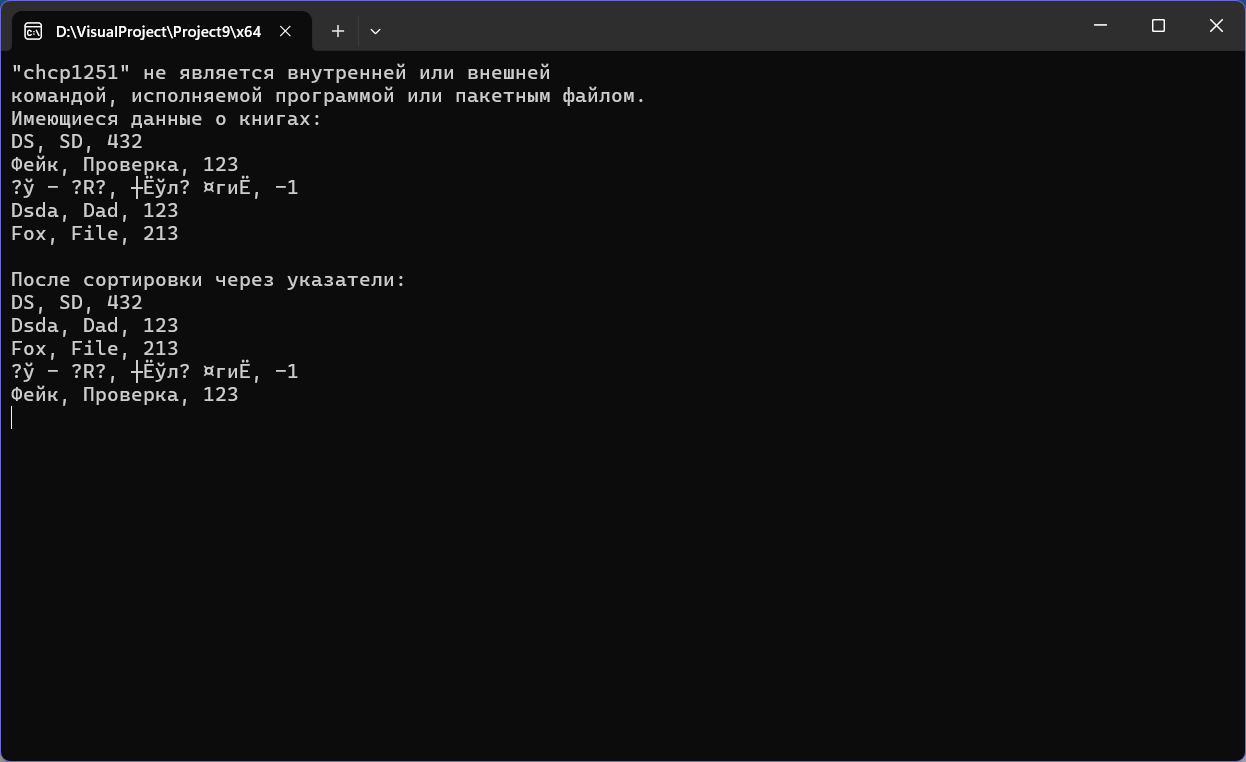


Рисунок 1 – Тест 1