**Лабораторная работа №4 (Указатели)**

**Задача №1**

Написать программу, которая позволяет пользователю ввести несколько

строк с клавиатуры, а затем выводящую их в порядке возрастания длины строки.

**Пояснение**

Строки вводятся до появления пустой строки и записываются в двумерный символьный массив. Одновременно происходит создание и заполнение массива указателей на char. После окончания ввода программа сортирует укзатели в массиве и выводит строки в соответствии с отсортированными указателями.

**Состав**

Программа должна состоять из функций:

- void lineSort(char \*str[],int size) - функция, сортирующая массив указателей

- void printLines(const char \*str[],int size) - функция, печатающая строки в порядке массива str

- main() - организация ввода строк в двумерный символьный массив

Создаются три файла: task1.h,task1.c,main1.c.

**Задача №2**

Написать программу, которая с помощью массива указателей выводит слова

строки в обратном порядке. Пример:

"буря мглою небо кроет" -> "кроет небо мглою буря"

**Пояснение**

Для входной строки создается массив указателей на char, в который заносятся адреса первых символов каждого слова. Затем организуется новая строка, при использовании этого массива из указателей. Массив из указателей должен объявляться внутри функции **reverseWords**.

**Состав**

Программа должна состоять из функций:

- char \* reverseWords(char \* in, char \*out) - функция, переворачивающая слова из in и записывающую их в out

- main() - организация ввода строки

Создаются три файла: task2.h,task2.c,main2.c.

**Задача №3**

Написать программу, которая запрашивает строку и определяет, не является

ли строка палиндромом (одинаково читается и слева направо и справа налево)

**Пояснение**

Цель задачи - применить указатели для быстрого сканирования строки с двух концов

**Состав**

Программа должна состоять из функций:

- int isPalindrome(char \* str) - функция, проверяющая str (ответ либо 0, либо 1)

- main() - организация ввода строки

Создаются три файла: task3.h,task3.c,main3.c.

**Задача №4**

Написать программу, сортирующую строки (см. задачу 1), но использующую

строки, прочитанные из текстового файла. Результат работы программы также

записывается в файл.

**Пояснение**

Программа должна использовать **task1.c**, в котором уже определена функция **lineSort**, а в **task4.c** еще одного определения быть не должно.

**Состав**

Программа должна состоять из функций:

- void lineSort(char \*str[],int size) - функция, сортирующая массив указателей

- void printLinesToFile(const char \*str[],int size,FILE \*fp) - функция, печатающая строки в порядке массива str

- main() - организация ввода строк в двумерный символьный массив

Создаются три файла: task4.h,task4.c,main4.c.

**Задача №5**

Написать программу, которая запрашивает количество родственников в семье,

а потом позволяет ввести имя родственника и его возраст. Программа должна

определить самого молодого и самого старого родственника и вывести их имена

**Пояснение**

Нужно завести массив строк для хранения имён и два указателя: **young** и **old**, которые по мере ввода, связывать с нужными строками. Также необходимы три числовые переменные для:

* Ввода текущего возраста
* Хранения максимального возраста
* Хранения минимального возраста

**Состав**

Программа должна состоять из функций:

- main()

Файлы: main5.c.