Лабораторная работа 2

**Цель**: освоить принципы работы с текстовыми файлами в языке С.

**Задача**: написать программу, реализующую простой алгоритм сжатия текста.

**Условия:**

Написать программу сжатия текстового файла по алгоритму:

1. Подсчет частоты встречи каждого слова в файле.
2. Поиск самого популярного среди длинных слов (А).
3. Поиск самого непопулярного среди коротких слов (В).
4. Замена всех слов А и В друг на друга.
5. Повтор пунктов 2-4 до тех пор, пока это имеет смысл.

**На 4-6**

- Программа должна сжать и разжать текстовый файл по приведенному выше алгоритму.

- В результате работы должно быть создано два файла: сжатый и разжатый. Последний должен полностью совпадать с исходным. Сжатый файл должен быть меньше исходного.

- При выборе слова А и В понятия «длинное слово» и «короткое слово» можно трактовать гибко. То есть, «длинное слово» не обязательно «самое длинное слово», главное, чтобы его использование имело положительный эффект.

ПРЕДУПРЕЖДЕНИЕ: чтение файла посимвольно (например, через fgetc) – быдлокод, так работает очень долго. Работайте со строками или словами!

ПРЕДУПРЕЖДЕНИЕ: Вычитывание ВСЕГО файла в строчку и работа только со строчкой – тоже быдлокод, так как убивает идеологию работы с файлами. Грамотно используйте функции из библиотеки stdio.

ПРЕДУПРЕЖДЕНИЕ: незакрытые хэндлы, отсутствие простейших проверок успешности файловых операций будут трактоваться как БАГИ.

**На 7-8**

- Сжатие и разжатие файла реализовать двумя разными программами (два отдельных проекта).

- Следовательно, в сжатый файл должны быть упакованы данные, необходимые для точного разжатия.

**На 9-10**

- Для реализации пункта 1 алгоритма использовать собственный связный список (стек или очередь).

Условия приема:

- **Плохое оформление** (названия файлов, функций, переменных, отступы, структура проекта и т.п.) и небольшие баги должны быть исправлены. При этом студент теряет лишь свое место в очереди.

- **Баги** (быдлокод, нерегулярное падение программы, выпадание ошибок) караются снижением категории (следовательно, и максимального балла). При этом лаба принимается. Так что тестируйте свое ПО заранее.

- **Серьезные ошибки**, влияющие на способность программы выполнять свою задачу (условия, описанные в задании) в принципе должны быт исправлены, при этом студент теряет и свое место в очереди и категорию.