Задание 10 Вариант 1

**Краткое описание работы**

В данной работе был реализован алгоритм сжатия данных с использованием методов RLE (Run-Length Encoding) и COBS (Consistent On-Bit Stuffing). В проекте были добавлены соответствующие заголовочные файлы и файлы кода для методов RLEDeflate и RLEInflate, которые отвечают за сжатие и разжатие данных, соответственно.

**1. Кодирование данных с использованием RLE**

Метод RLEDeflate: Этот метод принимает на вход указатель на массив байтов (входные данные), длину этих данных и указатель на выходной буфер, куда будут записаны закодированные данные. Алгоритм RLE работает следующим образом: он обходит входные данные и ищет последовательности одинаковых байтов. Каждая такая последовательность заменяется на пару: одно значение, указывающее на количество повторений, и само значение. Например, последовательность из 5 одинаковых символов 'A' будет заменена на '5A'.

**2. Декодирование данных с использованием RLE**

Метод RLEInflate: Этот метод выполняет обратную операцию — преобразует закодированные данные обратно в исходные. Если RLEDeflate записал, например, '5A', то RLEInflate вернет 5 символов 'A'. Метод принимает закодированные данные, их длину и выходной буфер для результатов.

**3. Применение кодирования COBS**

Кодирование COBS используется в данном проекте для устранения проблем с нулевыми байтами, которые могут присутствовать в данных. COBS добавляет дополнительную информацию в виде 'маркеров', чтобы избежать разбиения последовательности байтов на части, что важно для корректной передачи и хранения бинарных данных. Это гарантирует, что байты с нулевым значением не окажутся в закодированном виде, и тем самым облегчит последующую декомпрессию.

**4. Оценка эффективности сжатия методом RLE**

Эффективность алгоритма RLE зависит от структуры исходных данных. Если данные содержат много повторяющихся символов, сжатие будет эффективным. Например, текст типа "AAAAAABBBCCDAA" будет сильно сжат. Однако, если данные будут состоять из случайных байтов (как в файловых данных, например, в исполняемом файле), эффективность может оказаться низкой, так как повторяющихся символов будет недостаточно для значительного сжатия. В данном эксперименте RLE продемонстрировало свою способность уменьшать размер данных, хотя в случае файла data.exe, результаты могли варьироваться.

**Заключение**

Работа над проектом позволила изучить алгоритмы сжатия данных и их применение на практике. Реализация методов кодирования и декодирования данных с использованием RLE и COBS оказалась успешной, а конечный файл после сжатия и декомпрессии корректно выполняется, что подтверждает правильность работы алгоритмов. Проект был очищен от ненужных файлов и заархивирован.