**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное

учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Направление информатика и вычислительная техника

Отчет

по лабораторной работе №3

по дисциплине

«ТЕОРИЯ ИНФОРМАЦИИ»

## Архиватор

Выполнил:

Студент группы 8В32 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Г Усачев

Проверил:

Ассистент ОИТ ИШИТР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Кузнецов

Томск 2024

# Цель работы

разработать программу, архивирующую входной файл.

# Задание

Разработать программу, архивирующую и разархивирующую входной файл с помощью алгоритмов Шеннона-Фано и Хаффмана.

# Демонстрация построчной отладки кода

В файл «main.cpp» записываем код алгоритмов Шеннона-Фано и Хаффмена из лабораторной работы №2

// Структура узла дерева Хаффмана

struct Node {

std::string symbol; // Символ

double probability; // Вероятность

Node\* left; // Указатель на левое поддерево

Node\* right; // Указатель на правое поддерево

Node(std::string symbol, double probability) : symbol(symbol),

probability(probability),

left(nullptr),

right(nullptr){};

};

// структура для кода Шеннона-Фано

struct Symbol

{

std::string symbol;

double probability;

std::string code;

Symbol (std::string symbol, double probability): symbol(symbol), probability(probability){};

};

Листинг 1 – Программа «main.cpp»

В листинге 2 представлено расчёт максимальной энтропии для определения оптимальной кобминации.

for (int i = 0; i < max\_block\_length; i++)

{

double entropy = 0;

for (const auto pair : blocks[i]) {

double probability = static\_cast<double>(pair.second) / total[i];

entropy += probability \* log2(probability);

}

entropy \*= -1;

std::cout << "entropy blok length " << i+1 << " = " << entropy << std::endl;

// по максимальной энтропии определяем оптимальную длину блока

if (max\_entropy < entropy)

{

optimal\_block\_length = i+1;

max\_entropy = entropy;

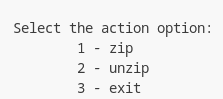
total\_blocks = total[i];

}

}

Листинг 2 – Программа «main.cpp»

На рисунке 1 представлено меню, с которым взаимодействует пользователь.

Рисунок 1 – Вывод меню



На листинге 3 представлено кодирование текста из файла.

/\* кодировка\*/

std::string zip\_file\_content = "";

// проходим по всем блоками и проверяем, если есть код записываем

for (int i\_line = 0; i\_line <= file\_content.size(); i\_line += optimal\_block\_length)

{

std::string block = file\_content.substr(i\_line, optimal\_block\_length); // Извлечение блока

for (const auto& pair : huffman\_blocks\_code)

{

if (pair.first == block)

{

zip\_file\_content += pair.second;

}

}

}

Листинг 3 — кодирование текста

На листинге 4 представлено раскодирование текста из закодированного файла.

for (char bit : file\_content)

{

buffer += bit;

if (codes.count(buffer))

{

unzip\_file\_content += codes[buffer];

buffer.clear();

}

}

if (!buffer.empty())

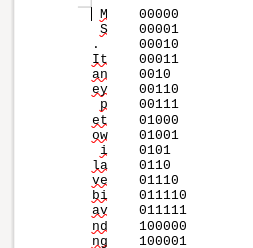
{

std::cerr << "Error: buffer not clear";

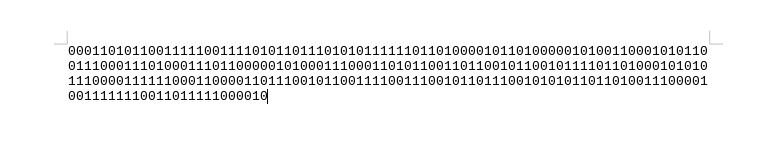
}

Листинг 4 — раскодирование текста

На рисунке 2 представлен файл с записанной кодировкой.

Рисунок 2 — файл с кодами

На рисунке 3 представлен закодированный файл.

Рисунок 3 — закодированный файл

**Выводы**

В результате лабораторной работы получены навыки сборки проекта на языке программирования С++ с применением программного средства автоматизации сборки программного обеспечения CMake и компилятора GCC в редакторе кода Visual Studio Code. Разработана программа архивирующая и разархивирующая входной файл с помощью кодирования алгоритмом Шеннона – Фано и Хаффмана. Написана и отлажена программа.

# Приложение 1 – Код программы

https://github.com/oldFiSt/teor\_inf\_03.git