

МИНОБРНАУКИ РОСИИ

*Федеральное государственное бюджетное образовательное учреждение высшего образования*

***«МИРЭА – Российский технологический университет»***

**РТУ МИРЭА**



Отчет по выполнению практического задания №8.1

**Тема:**

Алгоритмы кодирования и сжатия данныхДисциплина: «Структуры и алгоритмы обработки данных»

|  |  |  |
| --- | --- | --- |
| Выполнил студент: Васильев Б.А.  Группа: ИКБО-20-23 |  |  |

Москва 2024

**СОДЕРЖАНИЕ**

[ЦЕЛЬ РАБОТЫ 3](#_Toc181896201)

[ХОД РАБОТЫ 3](#_Toc181896202)

[Задание 1 3](#_Toc181896203)

[Формулировка задачи 3](#_Toc181896204)

[Метод Шеннона-Фано 4](#_Toc181896205)

[Метод Лемпеля-Зива LZ78 5](#_Toc181896206)

[Задание 2 7](#_Toc181896207)

[МЕТОД ШЕННОНА-ФАНО 7](#_Toc181896208)

[Формулировка задачи 7](#_Toc181896209)

[Описание подхода к решению 7](#_Toc181896210)

[Код программы 8](#_Toc181896211)

[Результаты тестирования 11](#_Toc181896212)

[МЕТОД ХАФФМАНА 12](#_Toc181896213)

[Формулировка задачи 12](#_Toc181896214)

[Описание подхода к решению 12](#_Toc181896215)

[Код программы 14](#_Toc181896216)

[Результаты тестирования 17](#_Toc181896217)

[ВЫВОД 18](#_Toc181896218)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 18](#_Toc181896219)

# ЦЕЛЬ РАБОТЫ

Познакомиться с алгоритмами кодирования и сжатия данных.

# ХОД РАБОТЫ

## Задание 1

### Формулировка задачи

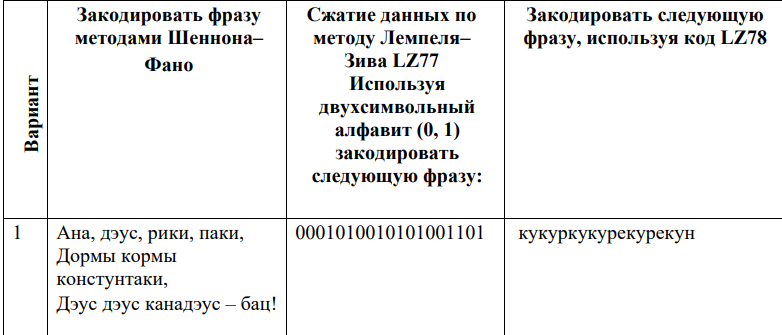


Рисунок 1 – Индивидуальный вариант задания

Выполнить каждую задачу варианта, представив алгоритм решения в виде таблицы и указав результат сжатия.

### Метод Шеннона-Фано

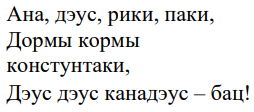


Рисунок 2 – Текст для кодирования

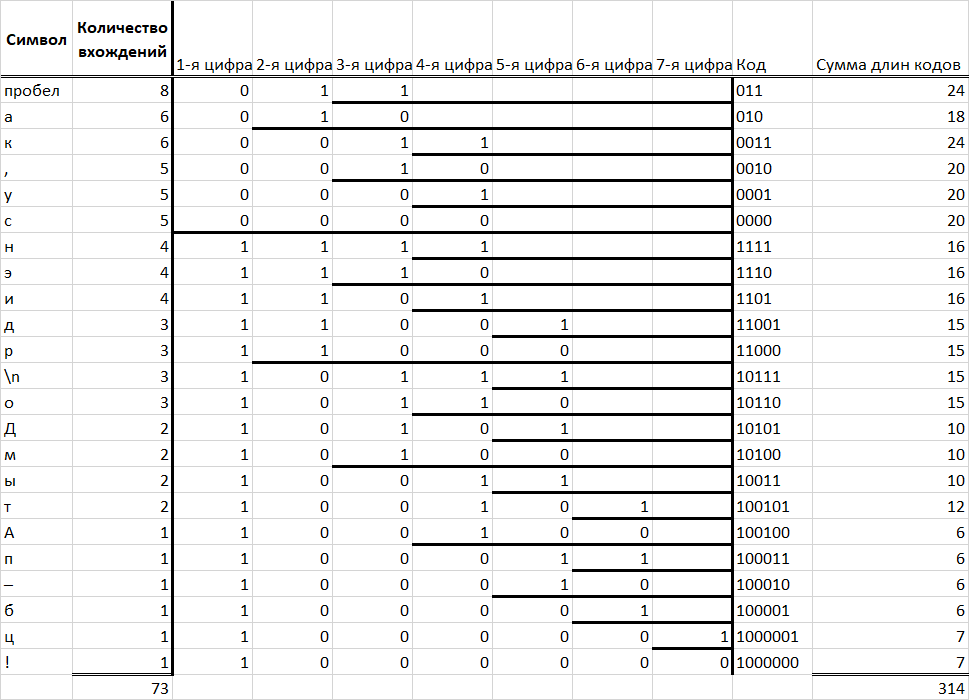


Рисунок 3 – Индивидуальный вариант задания

Незакодированная фраза – 73\*8 бит = 584 бит

Закодированная фраза – 314 бит

### Метод Лемпеля-Зива LZ77



Рисунок 4 – Последовательность для кодирования

Таблица 1 – Последовательное кодирование подпоследовательностей

|  |  |  |
| --- | --- | --- |
| Содержимое окна (сжимаемый текст) | Содержимое упреждающего буфера | Код назначенный последовательности |
|  | 0100101010010000101 |  |
| 0 | 100101010010000101 | 0 |
| 10 | 0101010010000101 | 10 |
| 01 | 01010010000101 | 11 |
| 010 | 10010000101 | 100 |
| 100 | 10000101 | 101 |
| 1000 | 0101 | 110 |
| 0101 |  | 111 |

Таблица 2 — Итоги кодирования

|  |  |
| --- | --- |
| Исходный текст | 0100101010010000101  0.10.01.010.100.1000.0101 |
| LZ-код | 0.10.11.110.100.1010.1001 |
| R | 2 3 |
| Вводимые коды | - 10 11 100 101 110 111 |

### Метод Лемпеля-Зива LZ78



Рисунок 5 – Индивидуальный вариант задания

Таблица 3 – Кодирование с помощью метода LZ78

|  |  |  |
| --- | --- | --- |
| Словарь | Считываемое содержимое | Код |
|  | к | <0, к> |
| к = 1 | у | <0, у> |
| к = 1, у = 2 | ку | <1, у> |
| к = 1, у = 2, ку = 3 | р | <0, р> |
| к = 1, у = 2, ку = 3, р = 4 | кук | <3, к> |
| к = 1, у = 2, ку = 3, р = 4, кук = 5 | ур | <2, р> |
| к = 1, у = 2, ку = 3, р = 4, кук = 5, ур = 6 | е | <0, е> |
| к = 1, у = 2, ку = 3, р = 4, кук = 5, ур = 6, е = 7 | кур | <3, р> |
| к = 1, у = 2, ку = 3, р = 4, кук = 5, ур = 6, е = 7,  кур = 8 | ек | <7, к> |
| к = 1, у = 2, ку = 3, р = 4, кук = 5, ур = 6, е = 7,  кур = 8, ек = 9 | ун | <2, н> |
| к = 1, у = 2, ку = 3, р = 4, кук = 5, ур = 6, е = 7,  кур = 8, ек = 9, ун = 10 | EOF | <0, EOF> |

Результат:

<0, к><0, у><1, у><0, р><3, к><2, р><0, е><3, р><7, к><2, н><0, EOF>

## Задание 2

### МЕТОД ШЕННОНА-ФАНО

#### Формулировка задачи

Разработать алгоритм и реализовать программу сжатия текста алгоритмом Шеннона-Фано. Разработать алгоритм и программу восстановления сжатого текста. Выполнить тестирование программы на текстовом файле. Определить процент сжатия.

#### Описание подхода к решению

Метод Шеннона-Фано представляет собой алгоритм для построения префиксного кода, направленного на минимизацию средней длины кодов для кодируемых сообщений. В основе метода лежит принцип разделения символов на две группы, примерно равные по суммарной вероятности или частоте появления. Этот процесс начинается с сортировки всех символов в порядке убывания частот, после чего множество символов делится на две группы так, чтобы суммы частот в этих группах были как можно ближе друг к другу.

Далее каждой группе присваивается свой префикс: обычно группе с более высокой суммарной частотой назначается префикс "0", а группе с меньшей — "1". Затем процесс повторяется для каждой из полученных групп: каждая группа снова делится на две подгруппы с равными суммами частот, и каждой подгруппе назначается следующий бит в коде. Этот рекурсивный процесс продолжается до тех пор, пока не будут достигнуты отдельные символы, для которых таким образом и будет определен уникальный префиксный код.

Метод Шеннона-Фано позволяет получить эффективное представление символов за счет того, что более частым символам назначаются более короткие коды, а менее частым — более длинные. Однако, в отличие от метода Хаффмана, метод Шеннона-Фано не всегда гарантирует минимально возможную длину кода для всех случаев, так как разделение на группы по частотам может быть неточным, что иногда приводит к незначительно большему объему данных.

#### Код программы

Реализуем код приложения на языке программирования C++ (рис. 6-8).

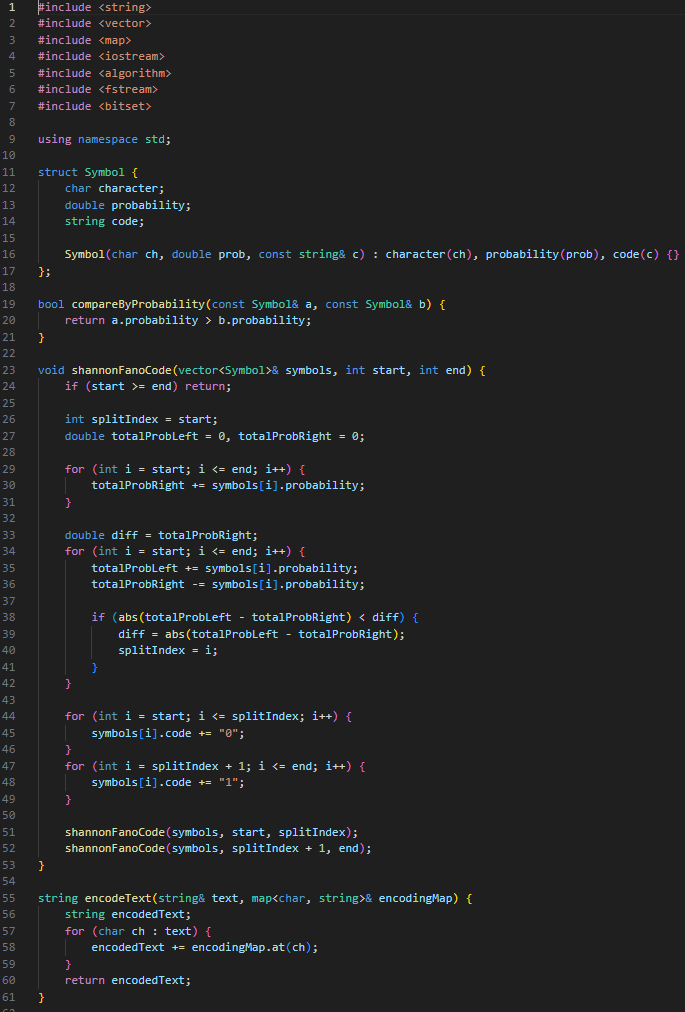


Рисунок 6 – код программы (часть 1)

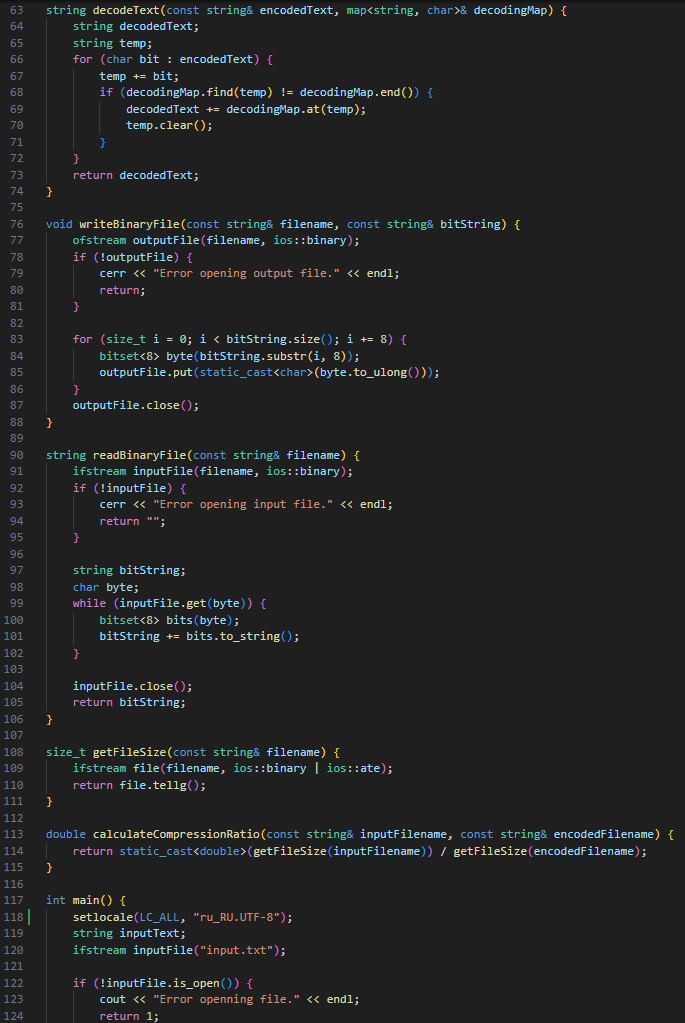


Рисунок 7 – код программы (часть 2)

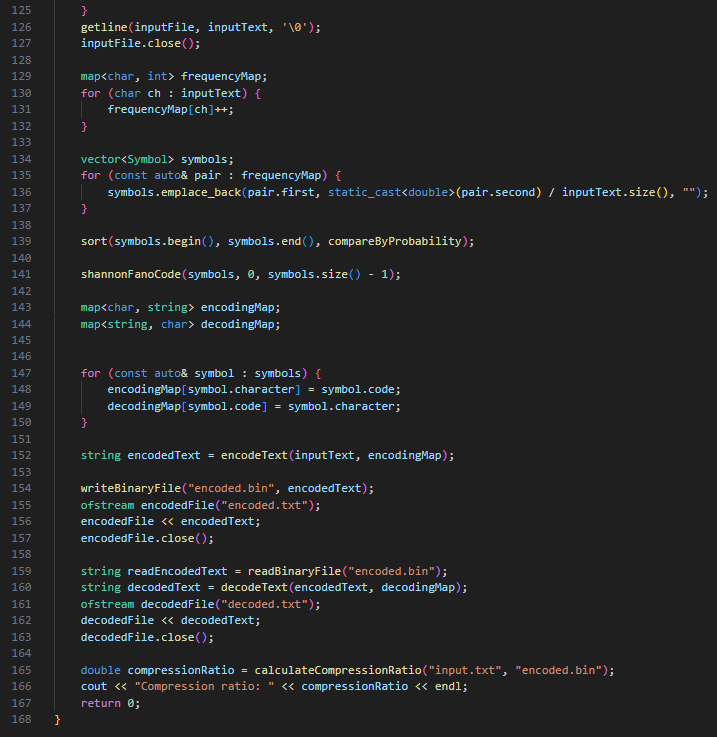
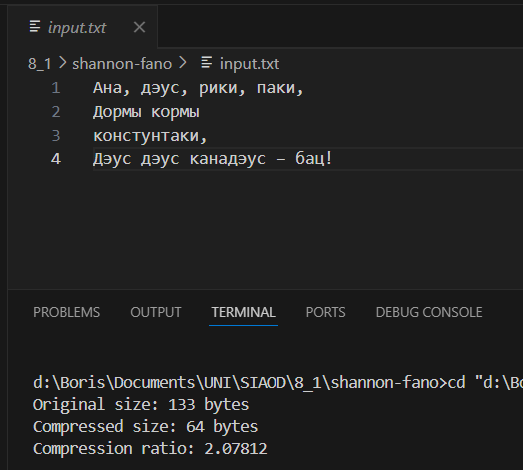
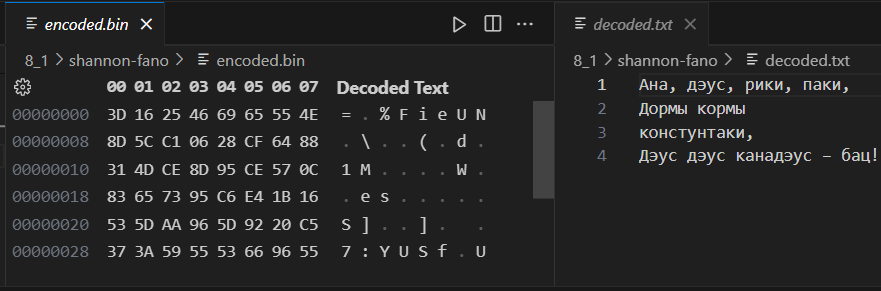


Рисунок 8 – код программы (часть 3)

#### Результаты тестирования

Выполним тестирование программы (рис. 9-10).

  
Рисунок 9 – Пример текстового файла

  
Рисунок 10 – Пример текстового файла

Тестирование показало, что программа работает корректно.

### МЕТОД ХАФФМАНА

#### Формулировка задачи

Провести кодирование(сжатие) исходной строки символов «Фамилия Имя Отчество» с использованием алгоритма Хаффмана. Исходная строка символов, таким образом, определяет индивидуальный вариант задания для каждого студента.

#### Описание подхода к решению

Алгоритм кодирования методом Хаффмана представляет собой жадный алгоритм для построения префиксного кода, минимизирующего среднюю длину кодируемых сообщений. Алгоритм начинается с анализа исходных данных и вычисления частоты каждого символа в тексте. На основании этих частот строится бинарное дерево, где каждый узел представляет символ и его частоту, а каждый путь от корня к листу определяет уникальный префиксный код для символа.

Для построения дерева Хаффмана все символы добавляются в приоритетную очередь в виде узлов с весами, соответствующими их частотам. Затем из очереди последовательно извлекаются два узла с наименьшими частотами, и для них создается новый родительский узел с частотой, равной сумме частот этих узлов. Новый узел возвращается в очередь, и процесс продолжается до тех пор, пока в очереди не останется один узел, который становится корнем дерева. В результате получается дерево, в котором более часто встречающиеся символы находятся ближе к корню, а редкие — дальше от корня, что позволяет получить короткие коды для часто встречающихся символов.

После построения дерева каждому символу присваивается код путем прохождения от корня к листьям, добавляя "0" при переходе в левое поддерево и "1" — в правое. Эти коды и используются для сжатия исходного текста. Кодировка Хаффмана обеспечивает компактное представление данных, так как более короткие коды назначаются более частым символам, а редкие символы получают более длинные коды, что позволяет эффективно сжимать данные.

#### Ход решения

На основе данных о количестве вхождений каждого символа (рис. 10а), построим дерево кодирования Хаффмана (рис. 10б).

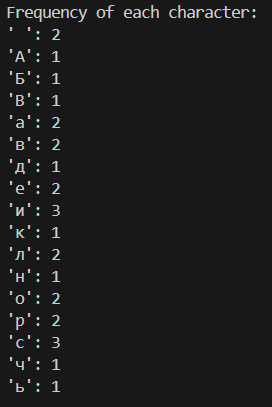


Рисунок 10а – количество вхождений каждого символа

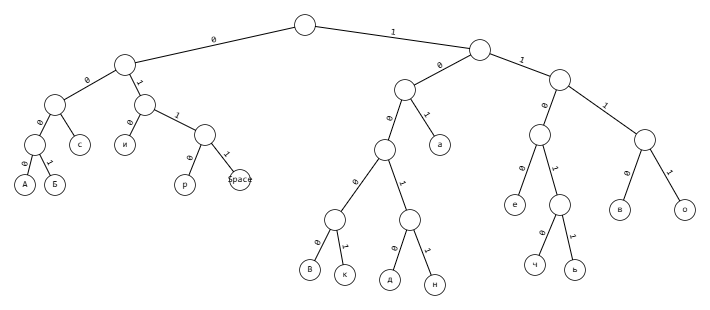


Рисунок 10б – дерево кодирования Хаффмана

#### Код программы

Реализуем код приложения на языке программирования C++ (рис. 11-13).

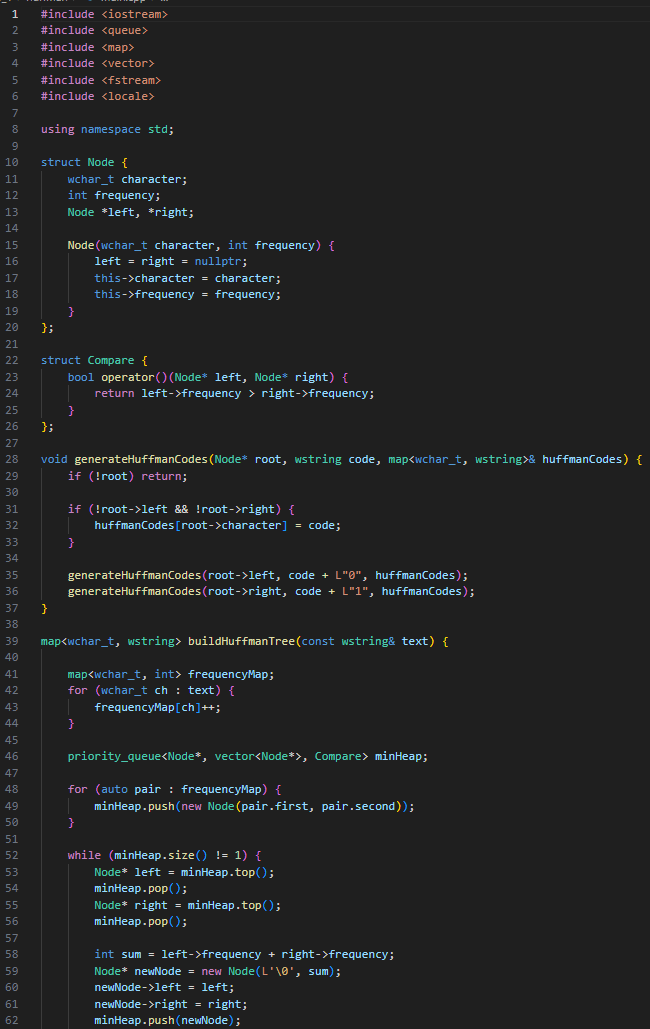


Рисунок 11 – код программы (часть 1)



Рисунок 12 – код программы (часть 2)

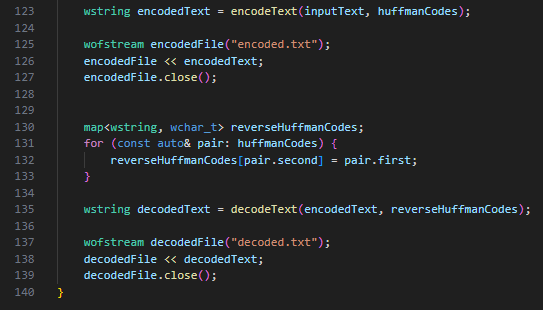
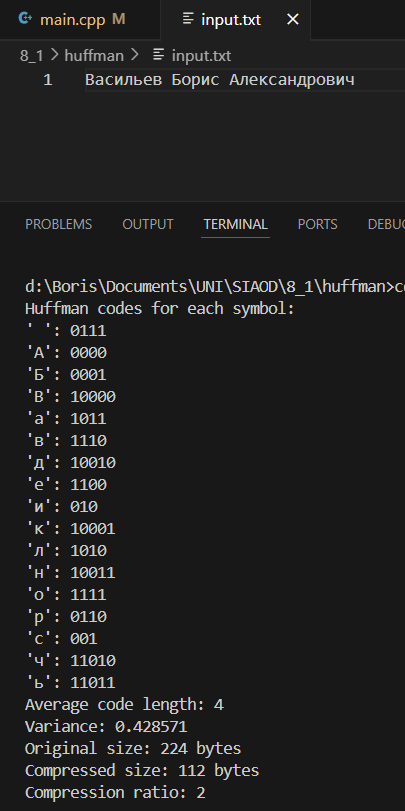
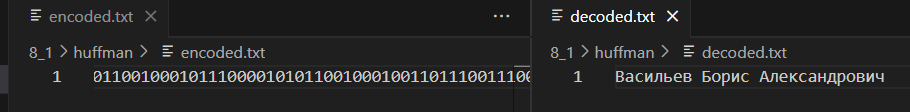


Рисунок 13 – код программы (часть 3)

#### Результаты тестирования

Выполним тестирование программы на различных значениях n. (рис. 14-15)

  
Рисунок 14 – исходный файл и запуск программы

  
Рисунок 15 – закодированный и декодированный файл

Тестирование показало, что программа работает корректно.

# ВЫВОД

В результате выполнения работы были освоены различные приёмы кодирования и сжатия данных.

# СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Рысин, М. Л. Введение в структуры и алгоритмы обработки данных : учебное пособие / М. Л. Рысин, М. В. Сартаков, М. Б. Туманова. — Москва : РТУ МИРЭА, 2022 — Часть 2 : Поиск в тексте. Нелинейные структуры данных. Кодирование информации. Алгоритмические стратегии — 2022. — 111 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/310826 (дата обращения: 02.11.2024).
2. Документация по языку С++ [Электронный ресурс]. URL: https://docs.microsoft.com/ruru/cpp/cpp/ (дата обращения 07.11.2024).