Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Исследование криптографических шифров на основе подстановки символов**

Студент: Димитриади А.В.

ФИТ 3 курс 5 группа

Преподаватель:

Савельева Маргарита Геннадьевна

Минск 2023

1. **Требования к приложению**

Приложение должно выполнять следующие функции:

* осуществлять шифрование на основе алгоритма Цезаря;
* осуществлять шифрование на основе алгоритма Трисемуса с ключевым словом;
* осуществлять оценку времени выполнения операций шифрования;
* осуществлять формирование появления символов для исходного и зашифрованных сообщений.

1. **Методика выполнения поставленных задач**

Шифрование сообщений выполняется с помощью функции EncryptCaesar. Код функции расположен на рисунке 2.1.

Эта функция использует алгоритм шифрования Цезаря, который является моноалфавитным. Смысл заключается в том, что исходный символ заменяется на символ со смещением. Формула шифрования символов сообщения: , где *N* – мощность алфавита, *k* – на сколько символов мы хотим сместить относительно начала алфавита, *x* – позиция нашего начального символа (нумерация алфавита начинается с нуля).

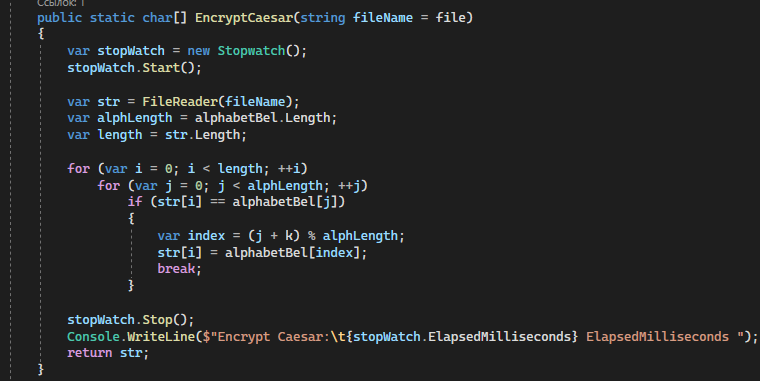


Рисунок 2.1 – Функция EncryptCaesar

Для расшифрования сообщения использовалась функция DecryptCaesar, алгоритм которой представлен на рисунке 2.2. Формула по которой расшифровывается сообщение:

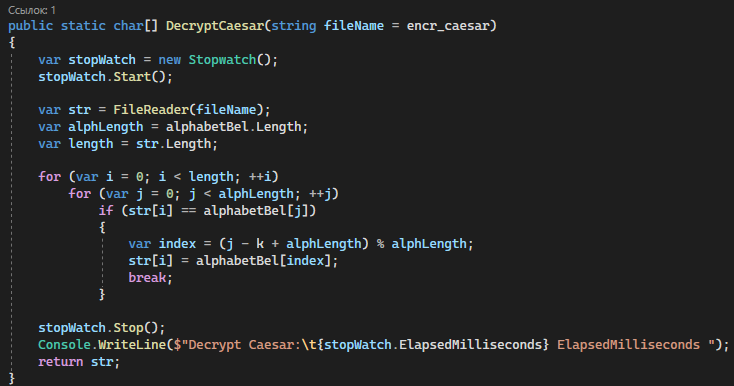


Рисунок 2.2 – Функция DecrytCaesar

Проверялась работоспособность данного алгоритма на белорусском тексте длинной в пятьсот символов (рисунок 2.3), считывали сообщения через функцию FileReader (рисунок 2.4) и далее с помощью функции FileWriter (рисунок 2.4) записываются данные в файл.

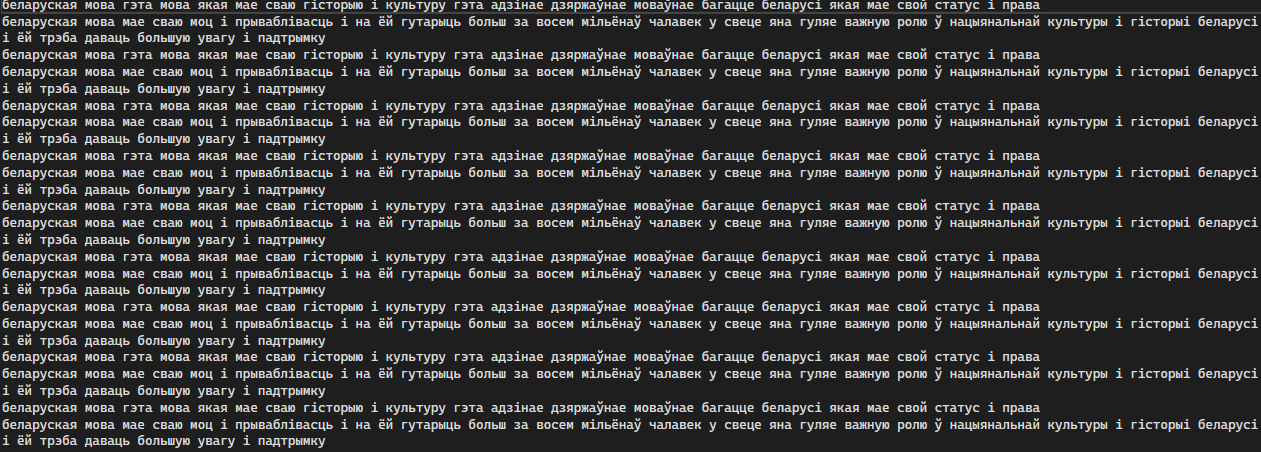


Рисунок 2.3 – Сообщение для зашифрования

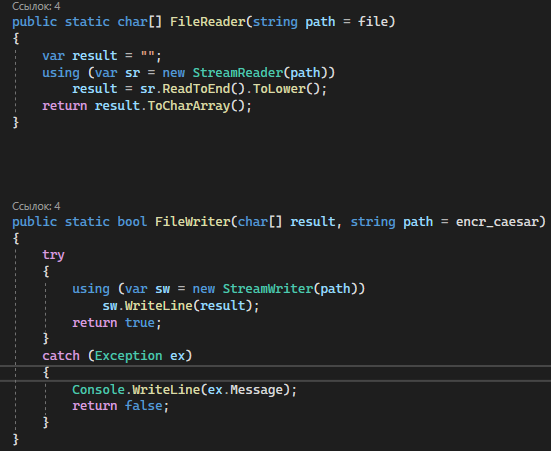


Рисунок 2.4 – Функци FileReader и FileWriter

Так же на рисунке 2.5 представлено зашифрованное сообщение. Покажем что функция работает правильно, путем зашифрования первого символа: б. Его порядковый номер равен единице, смещение у нас 21 (по варианту), мощность алфавита 32. Следовательно зашифрованный символ будет равен 1 + 21 = 22 mod 32 = 22 порядковому символу, а это символ ф, что доказывает правильность данного алгоритма.

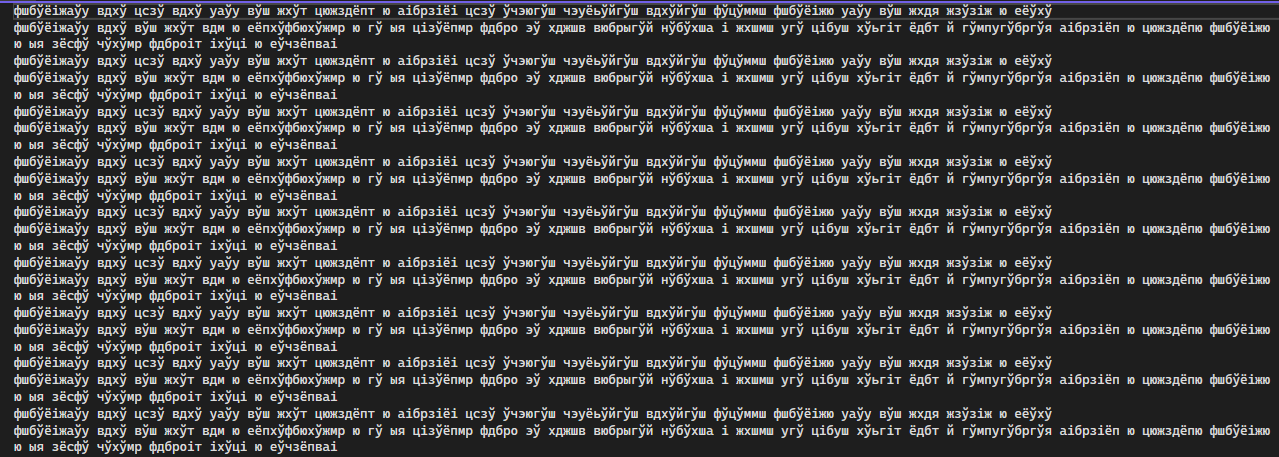


Рисунок 2.5 – Зашифрованное сообщение

Чтобы реализовать алгоритм Трисемуса используется таблица, инициализация которой представлен на рисунке 2.6.

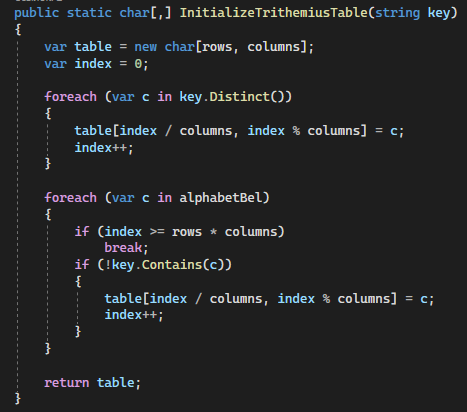


Рисунок 2.6 – Инициализация таблицы

Первоначально выбирается размер таблицы, количество строк и столбцов, которые в свою очередь зависят от мощности алфавита и длины кодового слова. Сначала записывается кодовое слово, причем повторяющиеся символы не пишутся, а дальше символы алфавита по порядку, которые еще не писались. При шифровании буква открытого текста заменяется буквой, расположенной ниже нее в том же столбце. Если буква текста оказывается в нижней строке таблицы, тогда для шифртекста берут самую верхнюю букву из того же столбца.

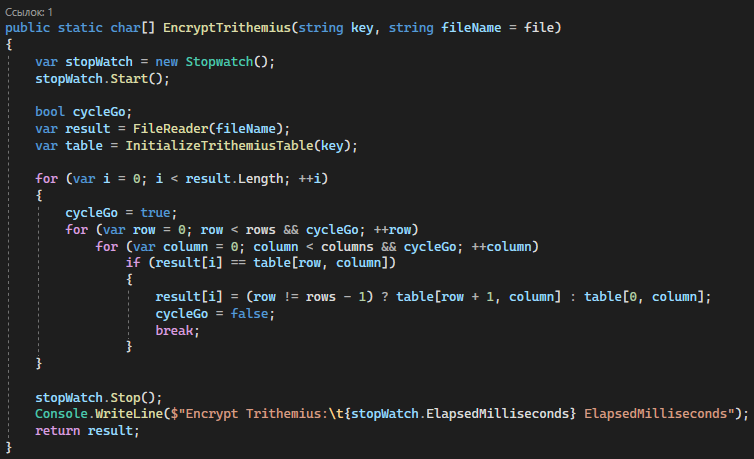


Рисунок 2.7 – Функция для шифрования

Результат работы алгоритма Трисемуса на основании ключа представлен на рисунке 2.8.

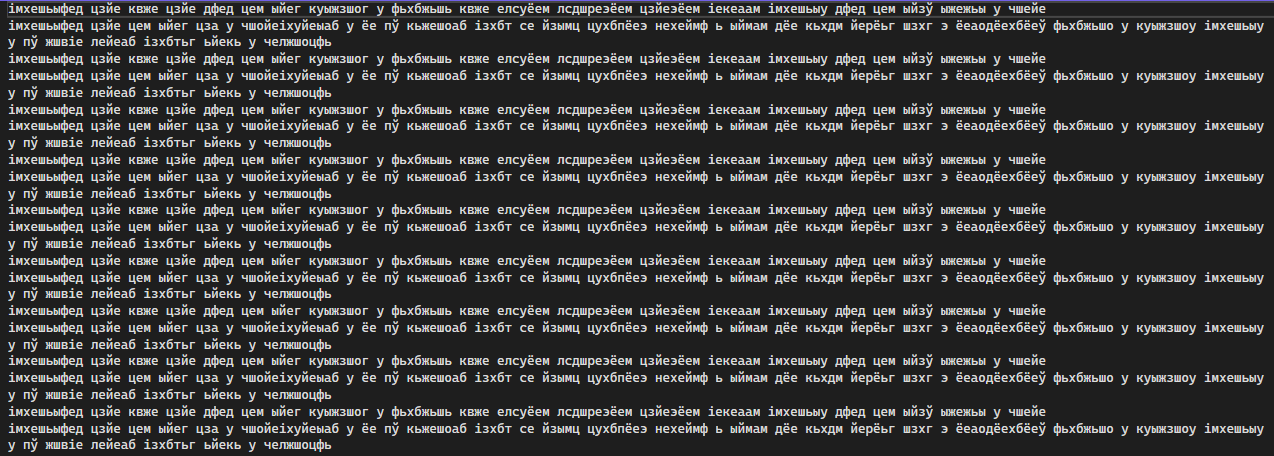


Рисунок 2.8 – Результат работы алгоритма Трисемуса

Были проведены исследования зависимости времени выполнения алгоритмов от количества символов в сообщении (рисунок 2.9).

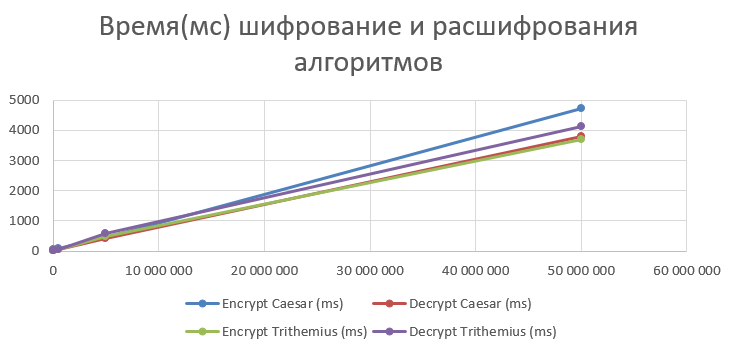


Рисунок 2.9 – Зависимость времени выполнения алгоритмов от количества символов в сообщении

Также сформированы гистограммы частот появления символов белорусского алфавита в сообщении (рисунок 2.10).

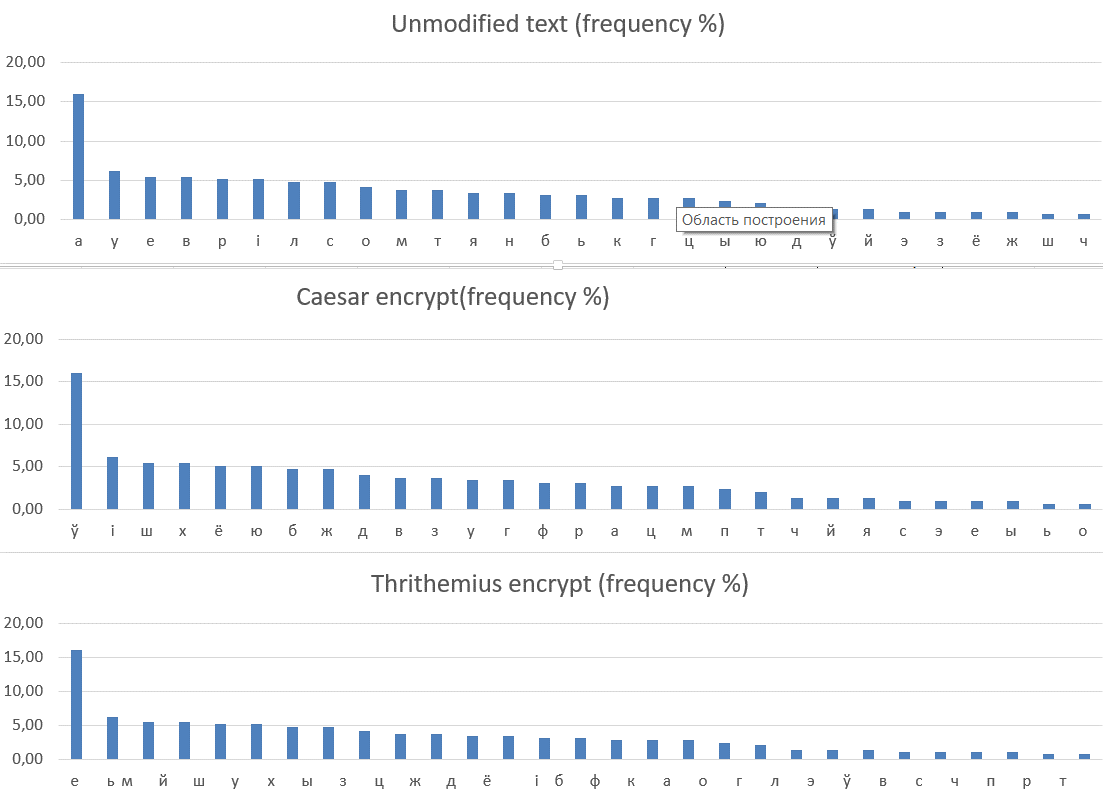


Рисунок 2.10 – Гистограммы частот появления символов белорусского алфавита в сообщении

**Вывод**

В ходе лабораторной работы были приобретены навыки разработки и использования шифров подстановки. Изучены методы шифрования алгоритмом Цезаря и Трисемуса. Проанализировав гистограммы, можно убедиться и сделать выводы, что их частота (графики) неизменны, только заменяется частота для символов, так как эти методы шифрования заменяют один символ на другой, то есть одному символу незашифрованного текста соответствует один символ зашифрованного текста.

Разработано приложение, на языке программирования C#,для реализации поставленных задач.