Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Теория информации (ТИ)

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

Тема работы:

Потоковое шифрование

Выполнила

студентка: гр. 351002 Хмель А.А.

Проверила: Болтак С.В.

Минск 2025

Тестовые наборы для метода потокового шифрования с помощью LFSR

Тест 1

Тестовая ситуация: проверка на корректность длины ключа.

Полученный результат: предупреждение о том, что ключ должен быть 26 бит.

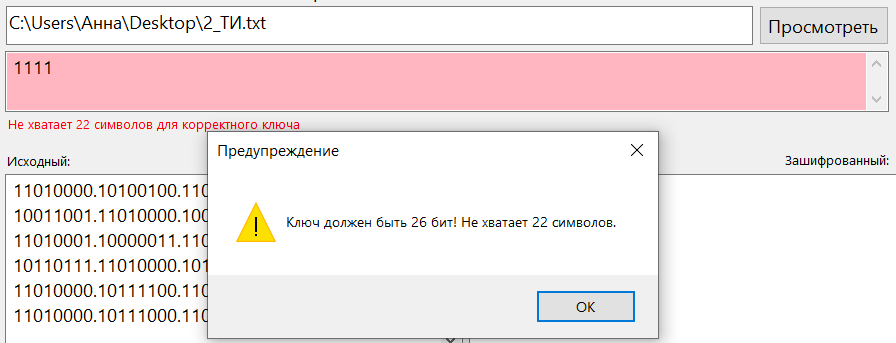


Рисунок 1 – Полученный результат. Тест 1

Тест 2

Тестовая ситуация: проверка на корректность работы программы при пустом поле для файла.

Полученный результат: ошибка о том, что файл не открыт.

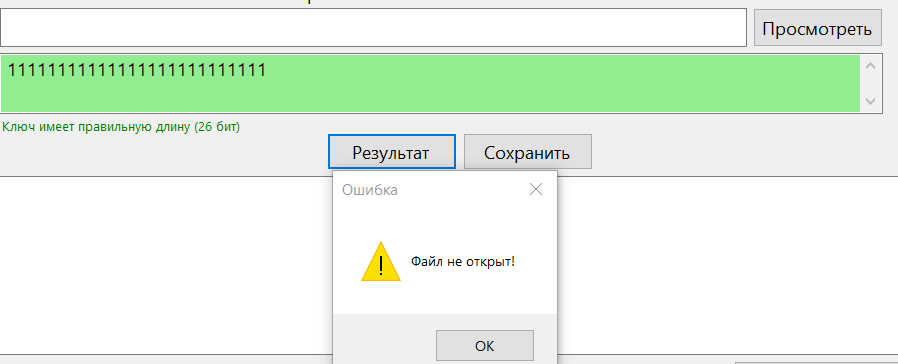


Рисунок 2 – Полученный результат. Тест 2

Тест 3

Тестовая ситуация: проверка на корректность работы программы для пустого файла.

Полученный результат: Предупреждение о том, что файл пуст.

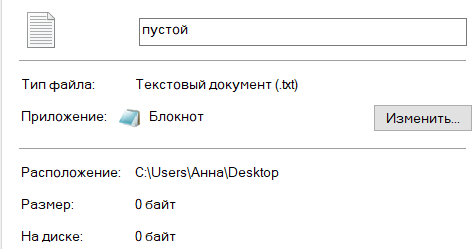
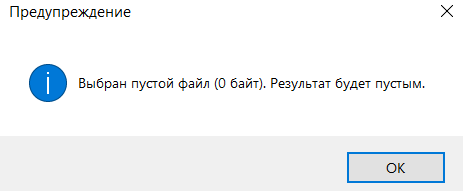
 

Рисунок 3 – Полученный результат. Тест 3

Тест 4

Тестовая ситуация: проверка на корректность работы программы для маленького размера файла.

Исходные данные:

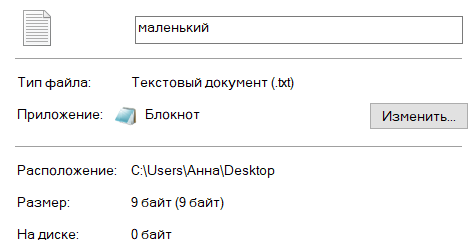
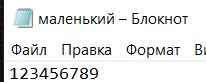


Рисунок 4 – Свойства и содержимое исходного файла. Тест 4

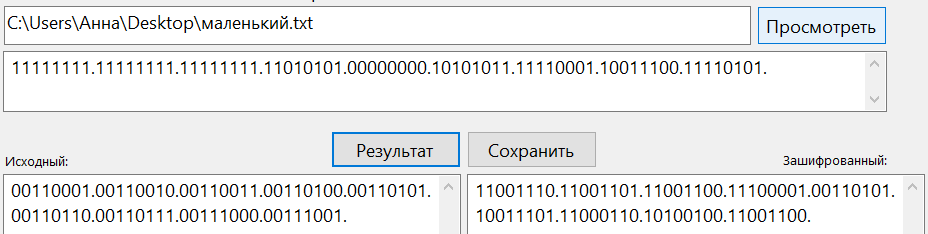


Рисунок 5 – Шифрация. Тест 4

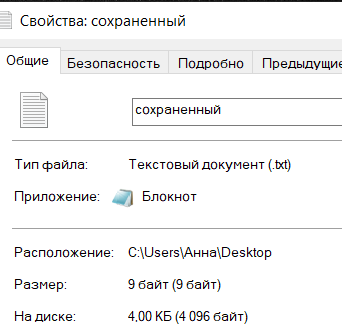
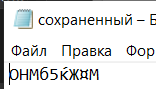


Рисунок 6 – Свойства и содержимое зашифрованного файла. Тест 4

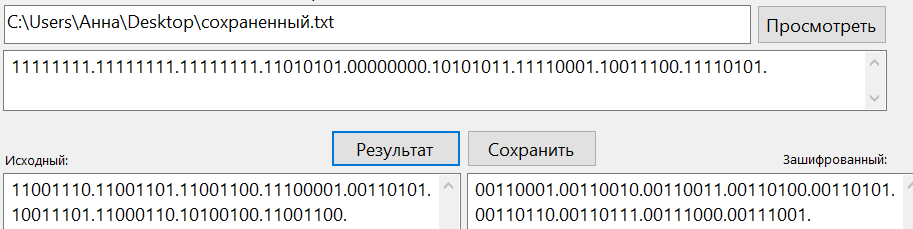


Рисунок 7 – Дешифрация. Тест 4

Объяснения смотрите на странице 13.

Тест 5

Тестовая ситуация: проверка на корректность работы программы для большого размера файла.

Исходные данные:

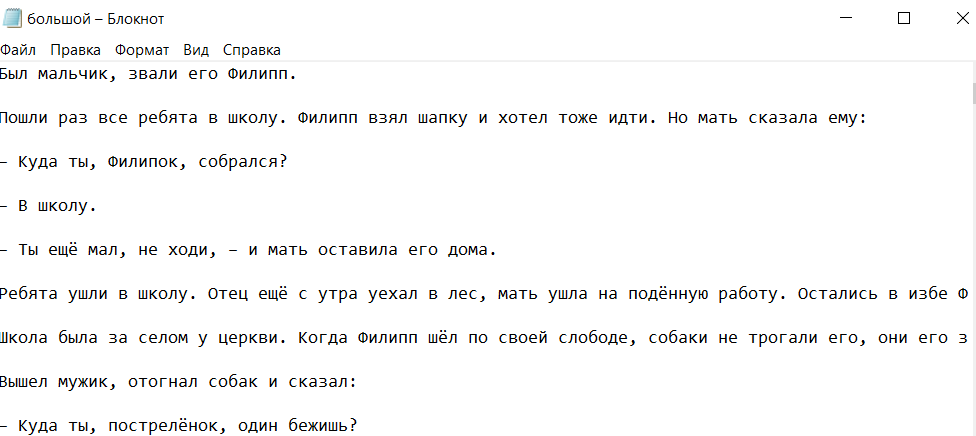
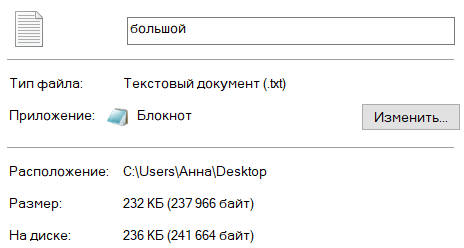


Рисунок 8 – Свойства и содержимое исходного файла. Тест 5

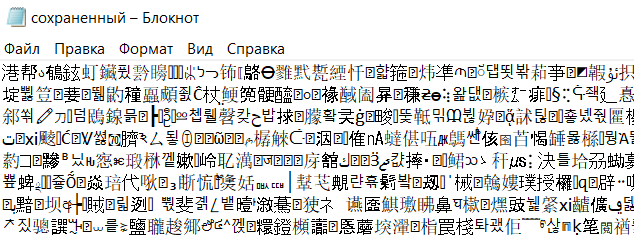
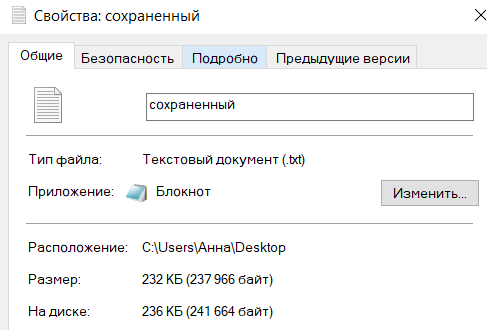
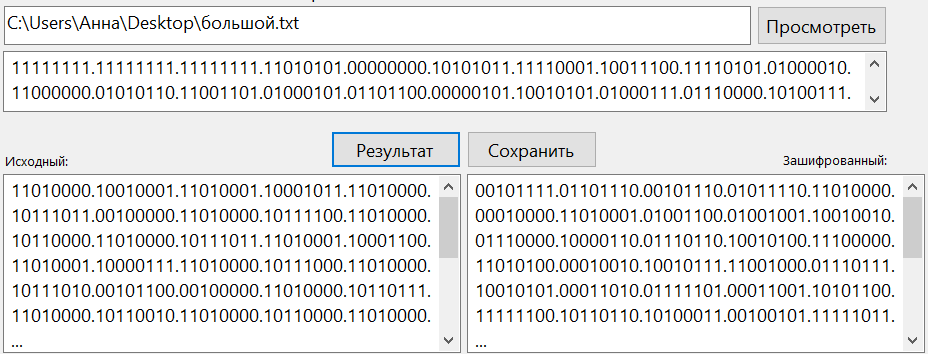


Рисунок 9 – Свойства и содержимое зашифрованного файла. Тест 5



ПРИМЕЧАНИЕ: разделитель **…** . Программа выводит первые и посл 30 бит.

Рисунок 10 – Шифрация. Тест 5

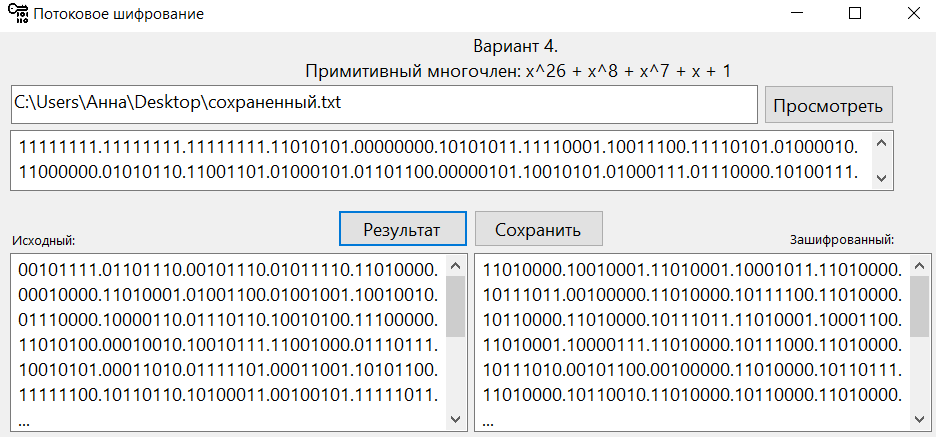


Рисунок 11 – Дешифрация. Тест 5

Тест 6

Тестовая ситуация: проверка на корректность работы программы для файла музыки.

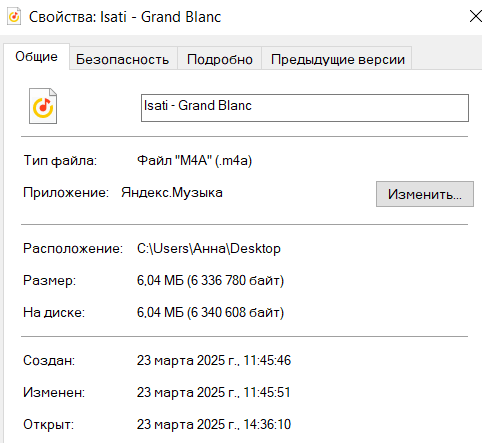


Рисунок 12 – Свойства исходного файла. Тест 6

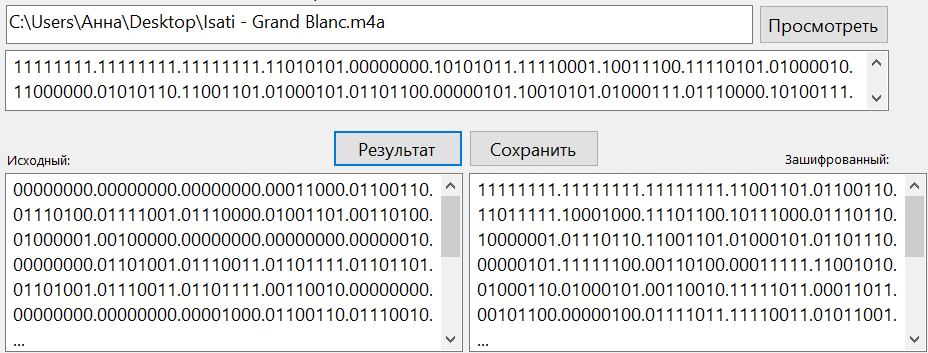
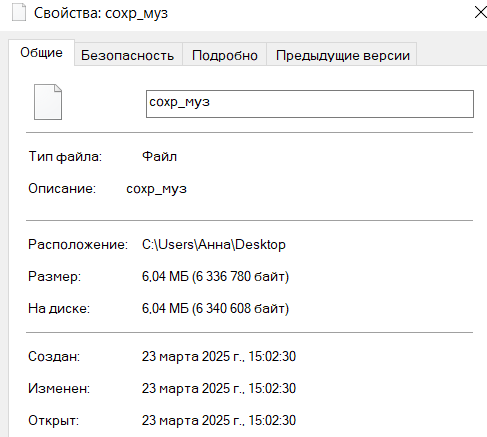


Рисунок 13 – Шифрация. Тест 6



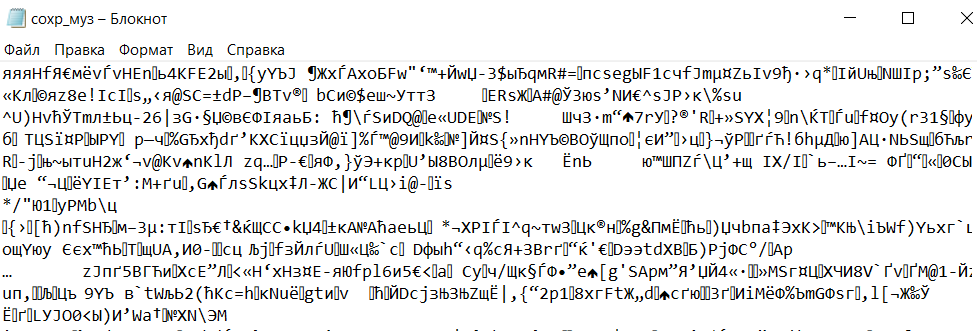


Рисунок 14 – Свойства и содержимое зашифрованного файла. Тест 6

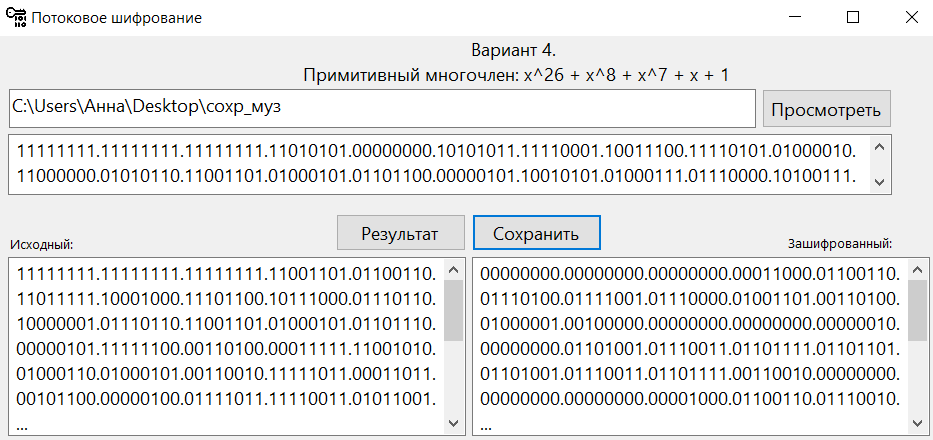


Рисунок 15 – Дешифрация. Тест 6

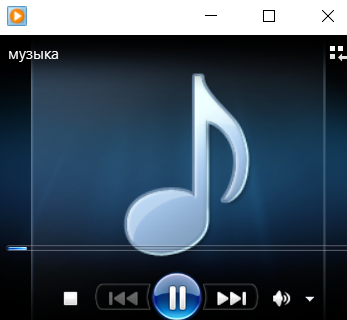


Рисунок 16 – Полученный результат. Тест 6

Тест 7

Тестовая ситуация: Проверка на корректность работы программы для файла формата изображения.

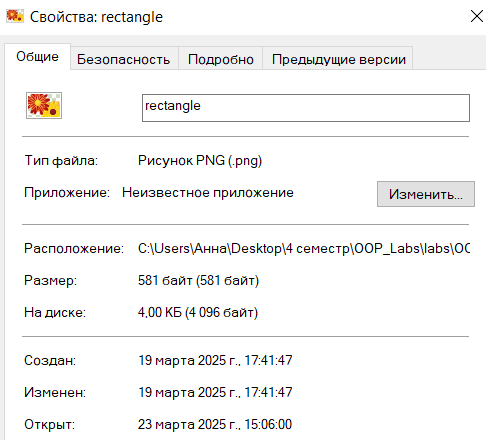
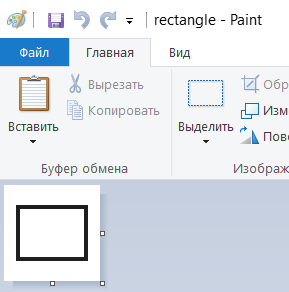
 

Рисунок 17 – Свойства и содержимое исходного файла. Тест 7

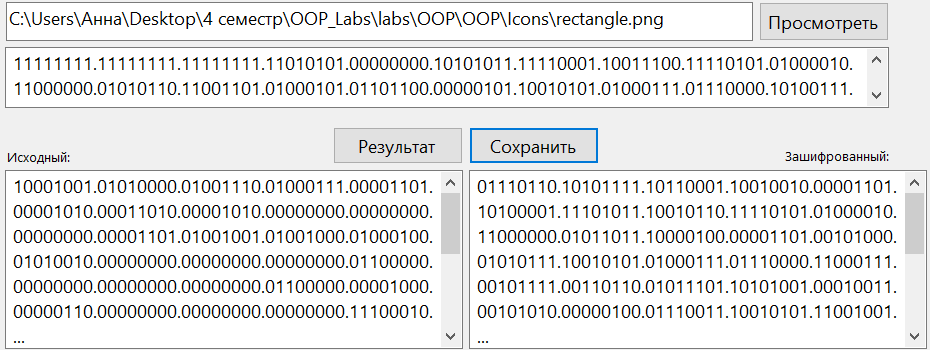
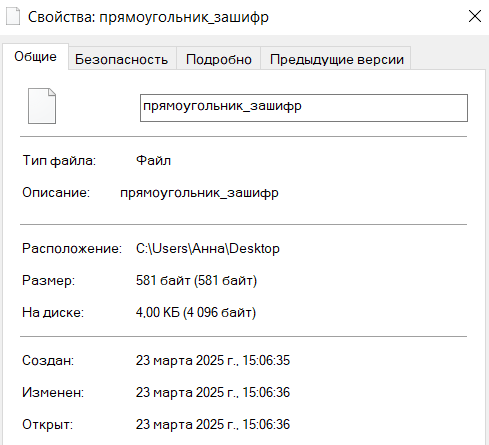


Рисунок 18 – Шифрация. Тест 7



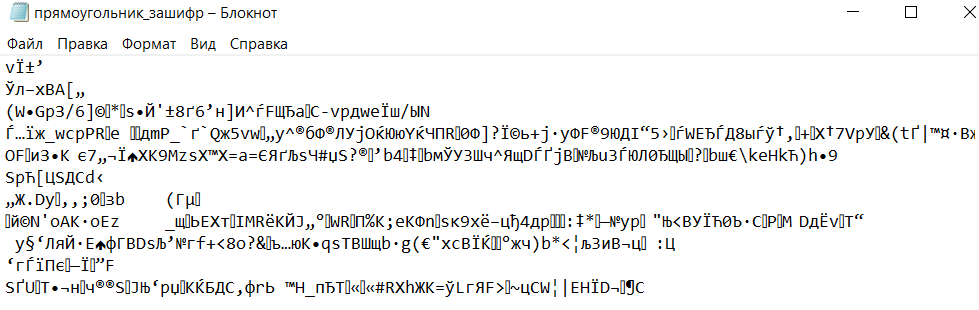


Рисунок 19 – Свойства и содержимое зашифрованного файла. Тест 7

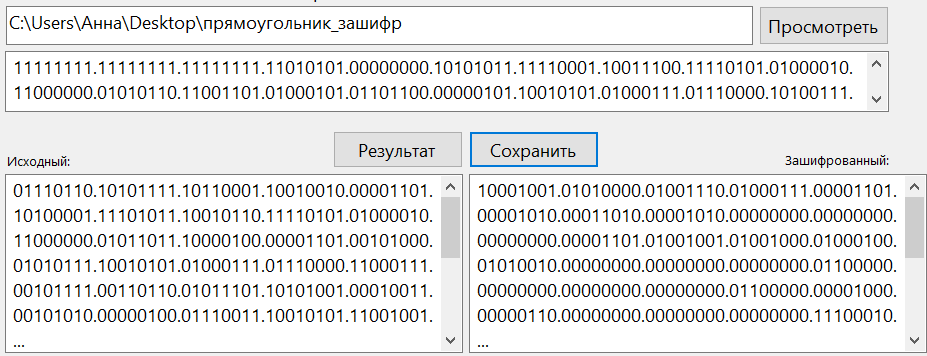


Рисунок 20 – Дешифрация. Тест 7

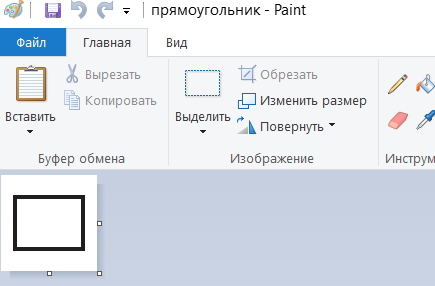


Рисунок 21 – Полученный результат. Тест 7

Тест 8

Тестовая ситуация: проверка на корректность работы программы для файла формата видео.

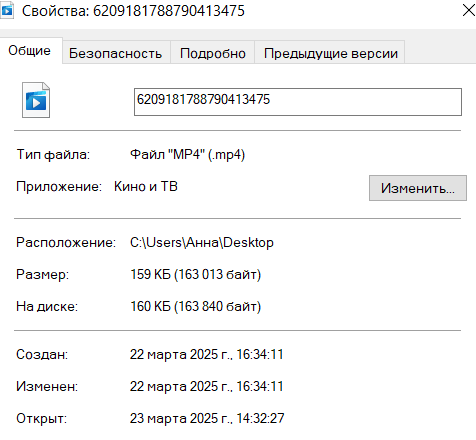
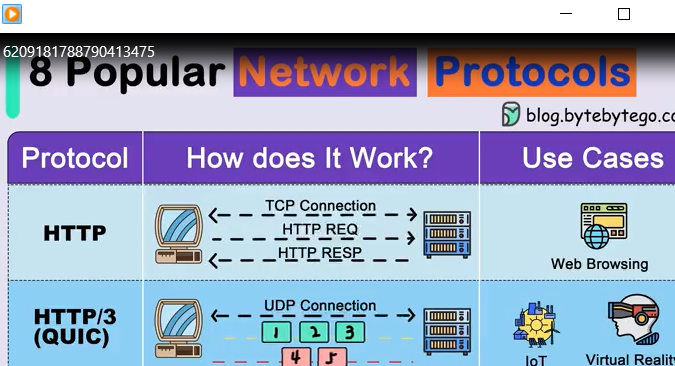
 

Рисунок 22 – Свойства и содержимое исходного файла. Тест 8

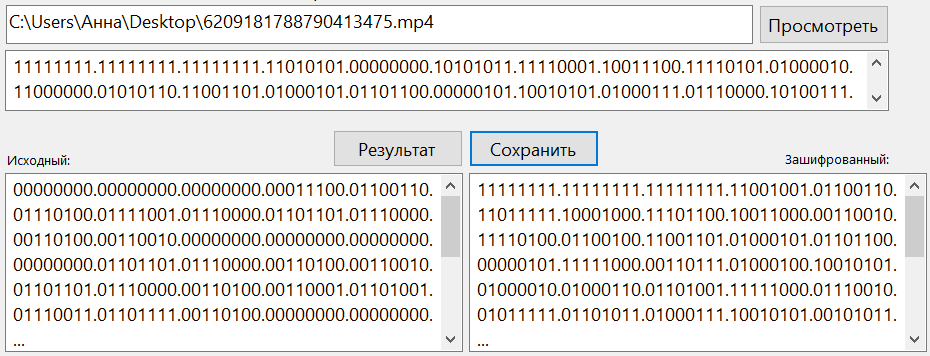


Рисунок 23 – Шифрация. Тест 8

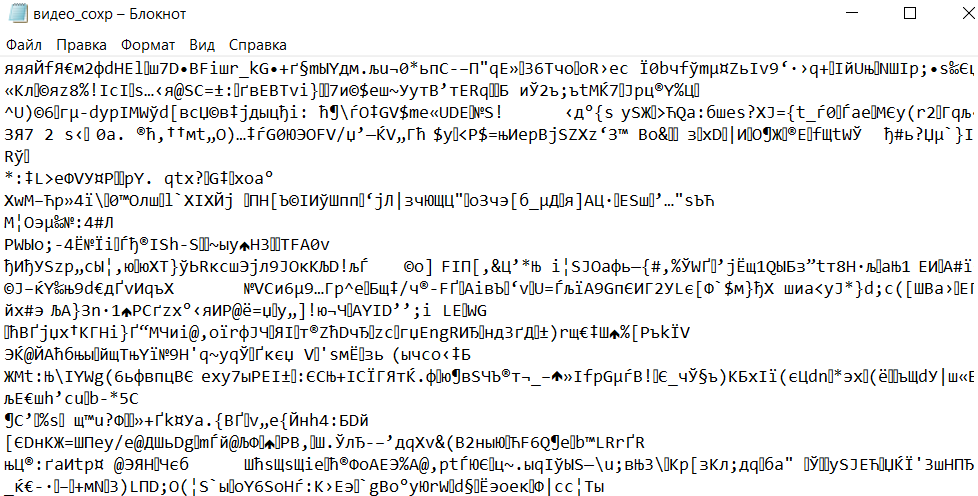


Рисунок 24 – Содержимое зашифрованного файла. Тест 8

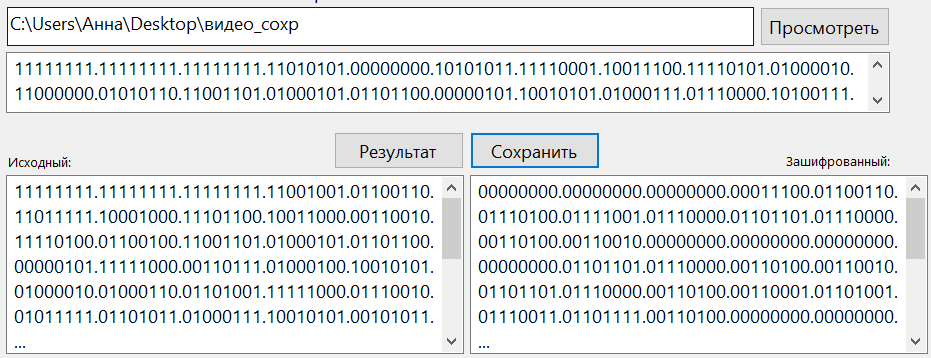
****

Рисунок 25 – Дешифрация. Тест 8

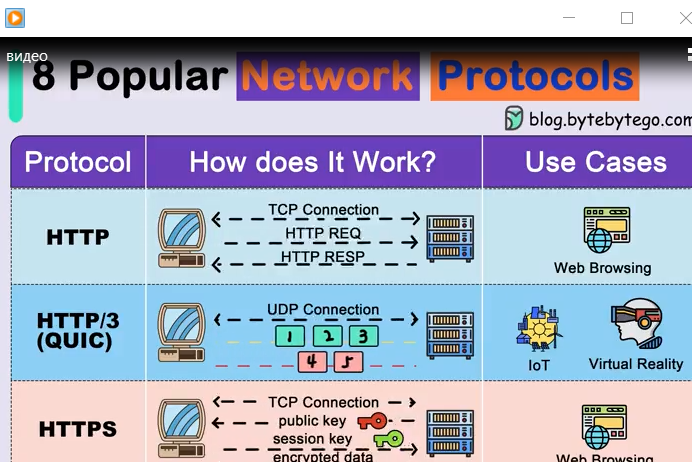
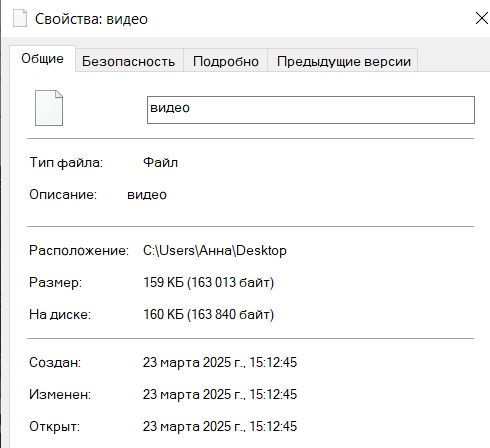
****

Рисунок 26 – Свойства и содержание полученного результата. Тест 8

Тест 9

Тестовая ситуация: проверка на корректность работы программы для файла формата видео.

Исходные данные: см. Тест 8

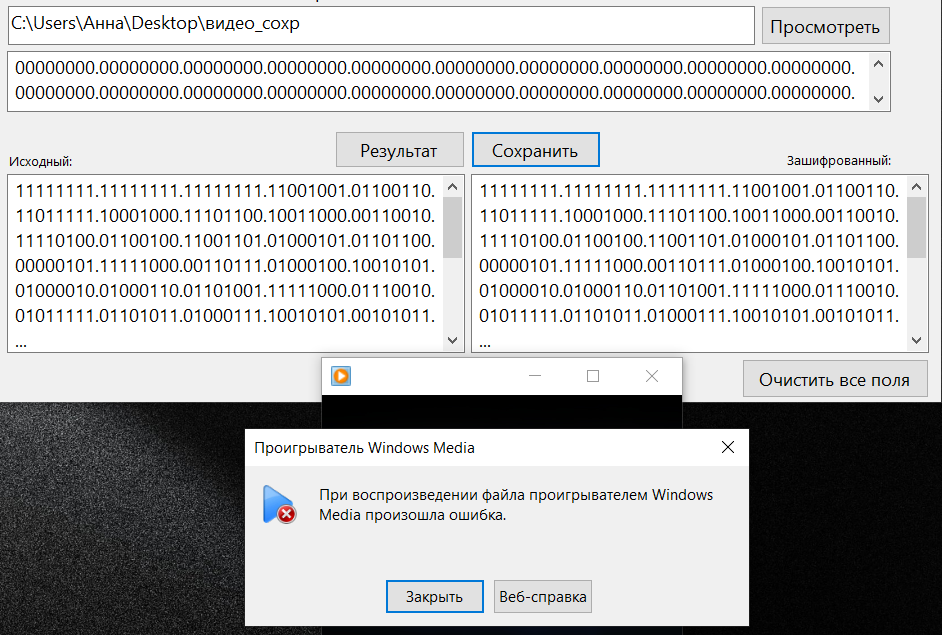
****

Рисунок 27 –Полученный результат. Тест 9

Тест 10

Тестовая ситуация: проверка на корректность работы программы для

относительно большоого файла, содержащего только нули.

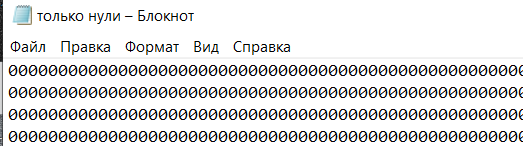


Рисунок 28 – Содержимое исходного файла. Тест 10

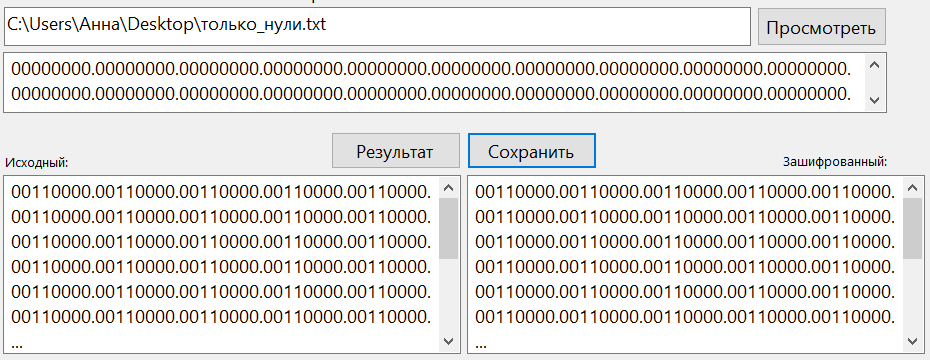


Рисунок 29 – Шифрация. Тест 10

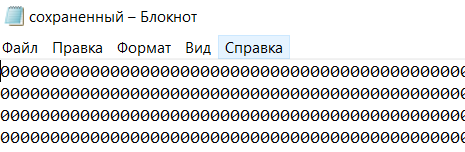


Рисунок 30 – Содержимое зашифрованного файла. Тест 10

Процесс построения сдвигового регистра с линейными обратными связями, задаваемый многочленом ***P*(*x*) = *x*26 +*x*8 + *x*7 + *x* + 1** для теста №4

Начальное состояние: 111111111111111111111111

**1**11111111111111111**11**11111**1** → **1**11111111111111111**11**11111**0** → **1**11111111111111111**11**11110**1** → **1**11111111111111111**11**11101**0** → **1**11111111111111111**11**11010**1** → **1**11111111111111111**11**10101**0** → **1**11111111111111111**11**01010**1** → **1**11111111111111111**10**10101**0** → **1**11111111111111111**01**01010**0** → **1**11111111111111110**10**10100**0** → **1**11111111111111101**01**01000**0** → **1**11111111111111010**10**10000**0** → **1**11111111111110101**01**00000**0** → **1**11111111111101010**10**00000**0** → **1**11111111111010101**00**00000**0** → **1**11111111110101010**00**00000**1** → **1**11111111101010100**00**00001**0** → **1**11111111010101000**00**00010**1** → **1**11111110101010000**00**00101**0** → **1**11111101010100000**00**01010**1** → **1**11111010101000000**00**10101**0** → **1**11110101010000000**01**01010**1** → **1**11101010100000000**10**10101**1** → **1**11010101000000001**01**01011**1** → **1**10101010000000010**10**10111**1** → **1**01010100000000101**01**01111**1** → **0**10101000000001010**10**11111**1** → **1**01010000000010101**01**11111**0** → **0**10100000000101010**11**11110**0** → **1**01000000001010101**11**11100**0** → **0**10000000010101011**11**11000**1** → **1**00000000101010111**11**100011 → **0**00000001010101111**11**00011**0** → **0**00000010101011111**10**00110**0** → **0**00000101010111111**00**01100**1** → **0**00001010101111110**00**11001**1** → **0**00010101011111100**01**10011**1** → **0**00101010111111000**11**00111**0** → **0**01010101111110001**10**01110**0** → **0**10101011111100011**00**11100**1** → **1**01010111111000110**01**11001**1** → **0**10101111110001100**11**10011**1** → **1**01011111100011001**11**00111**1** → **0**10111111000110011**10**01111**0** → **1**01111110001100111**00**11110**1** → **0**11111100011001110**01**11101**0** → **1**11111000110011100**11**11010**1** → **1**11110001100111001**11**10101**0** → **1**11100011001110011**11**01010**1** → **1**11000110011100111**10**10101**0** → **1**10001100111001111**01**01010**0** → **1**00011001110011110**10**10100**0** → **0**00110011100111101**01**01000**0** → **0**01100111001111010**10**10000**1** → **0**11001110011110101**01**00001**0** → **1**10011100111101010**10**00010**1** → **1**00111001111010101**00**00101**1** → **0**01110011110101010**00**01011**0** → **0**11100111101010100**00**10110**0** → **1**11001111010101000**01**01100**0** → **1**10011110101010000**10**11000**0** → **1**00111101010100001**01**10000**0** → **0**01111010101000010**11**00000**0** → **0**11110101010000101**10**00000**0** → **1**11101010100001011**00**00000**1** → **1**11010101000010110**00**00001**0** → **1**10101010000101100**00**00010**1** → **1**01010100001011000**00**00101**0** → **0**10101000010110000**00**01010**1** → **1**01010000101100000**00**10101**1** → **0**10100001011000000**01**01011**0** → **1**01000010110000000**10**10110**1.**

Последовательность сгенерированных битов ключа: 11111111.11111111.11111111.11010101.00000000.10101011.11110001.10011100.11110101.

11111111.11111111.11111111.11010101.00000000.10101011.11110001.10011100.11110101.

00110001.00110010.00110011.00110100.00110101.00110110.00110111.00111000.00111001.

11001110.11001101.11001100.11100001.00110101.10011101.11000110.10100100.11001100.

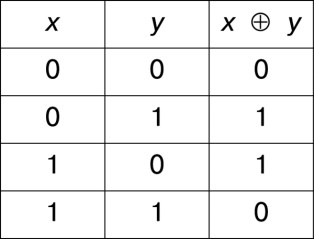


Рисунок 3 – Логическая операция XOR.

*P*(*x*) = *x*26 +*x*8 + *x*7 + *x* + 1

