УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №3

по предмету

Теория Информации

Вариант 1

Выполнил:

Лащенко Е.Н.

Проверил:

Болтак С.В.

Группа 351002

Минск 2025

1. **Пример работы алгоритма быстрого возведения в степень с использованием модульной арифметики.**

**517 mod 13 = ?  
a – основание(5)  
n – показатель степени(17)**

**e – четное n или нет(да/нет)**

**r – результат(вначале равен 1)**

**x – значение mod(13)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Шаг | a | n | e | Действие | r mod x |
| 1 | 5 | 17 | нет | r = 1 \* 5 mod 13 = 5 n = 17 - 1 = 16 | 5 |
| 2 | 5 | 16 | да | a = 5^2 mod 13 = 12 n = 16 / 2 = 8 | 5 |
| 3 | 12 | 8 | да | a = 12^2 mod 13 = 1 n = 8 / 2 = 4 | 5 |
| 4 | 1 | 4 | да | а = 1^2 mod 13 = 1 n =4 /2 = 2 | 5 |
| 5 | 1 | 2 | да | a = 1^2 mod 13 = 1 n = 2 / 2 = 1 | 5 |
| 6 | 1 | 1 | нет | r = 5 \* 1 mod 13 = 5 n = 1 - 1= 0 | 5 |

1. **Пример поиска всех первообразных корней для p и заданного модуля.**

Задано простое p = 29

Ищем простые делители p-1 = 28 = 2^2\*7

Проверяем является ли число 2 первообразным корнем по модулю 29:

2^(28/2) mod 29 = 28; 2^(28/7) mod 29 = 16. Число 2 является первообразным по модулю 29.

Проверяем является ли число 3 первообразным корнем по модулю 29:

3^(28/2) mod 29 = 28; 3^(28/7) mod 29 = 23. Число 3 является первообразным по модулю 29.

Если найден один первообразный корень g по модулю p, остальные корни имеют вид g^k, где НОД(k, p-1) = 1. Для p = 29, допустимые k: 1, 3, 5, 9, 11, 13, 15, 17, 19, 23, 25, 27:

1) 2^1 mod 29 = 2

2) 2^3 mod 29 = 8

3) 2^5 mod 29 = 3

4) 2^9 mod 29 = 19

5) 2^11 mod 29 = 18

6) 2^13 mod 29 = 14

7) 2^15 mod 29 = 27

8) 2^17 mod 29 = 21

9) 2^19 mod 29 = 26

10) 2^23 mod 29 = 10

11) 2^25 mod 29 = 11

12) 2^27 mod 29 = 15

Тогда все первообразные корни для модуля p = 29 это 2, 8, 3, 19, 18, 14, 27, 21, 26, 10, 11, 15

1. **Пример работы расширенного алгоритма Евклида с взаимно простыми числами**

**x1\*a + y1\*b = НОД(a,b), a = 543, b = 307, НОД(a,b) = 1, т.к a и b взаимно простые.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **итерация** | **Делимое** | **Делитель** | **Частное** | **Остаток** |
| 1 | 725 | 302 | 2 | 121 |
| 2 | 302 | 121 | 2 | 60 |
| 3 | 121 | 60 | 2 | 1 |
| 4 | 60 | 1 | 60 | 0 |

Начинаем обратный ход с предпоследней операции(60/1):  
1 = 121 –60\*2;  
1 = 121 – (302 – 121\*2)\*2;(подставляем выражение для 60)

1 = (725 – 302\*2)\*5 – 302\*2 (подставляем выражение для 121)

1 = 725\*5 − 302\*(-12)  
Получаем ответ: x = 5,y = −12

**Задание:**

Разработать программное средство, выполняющее вычисление открытого ключа **(*KO*)** алгоритма ***RSA*** и побайтовое шифрование данным ключом по алгоритму ***RSA*** произвольного файла. Значения параметров ***p,*** ***q*** и ***KС*,** а также имя входного файла задаются пользователем. Программа должна осуществлять проверку ограничений на вводимые пользователем значения параметров алгоритма. Организовать вывод содержимого зашифрованного файла на экран в виде чисел в 10 системе счисления.

Разработать программное средство, выполняющее расшифрование файла, каждый 16-битный блок которого представляет собой зашифрованное по алгоритму ***RSA*** 8-битное значение. Значения модуля ***r*** и ***закрытого*** ключа ***KС***задаются пользователем.

Использовать алгоритм быстрого возведения в степень и расширенный алгоритм Евклида.

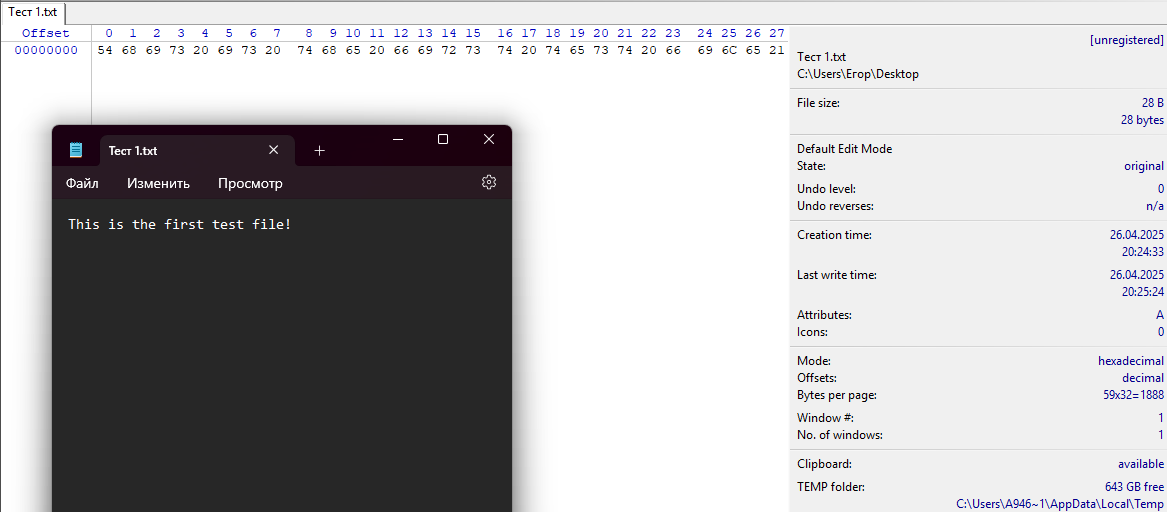
При использовании длинной арифметики для определения простоты числа использовать один из вероятностных тестов: тест Ферма или тест Миллера-Рабина.

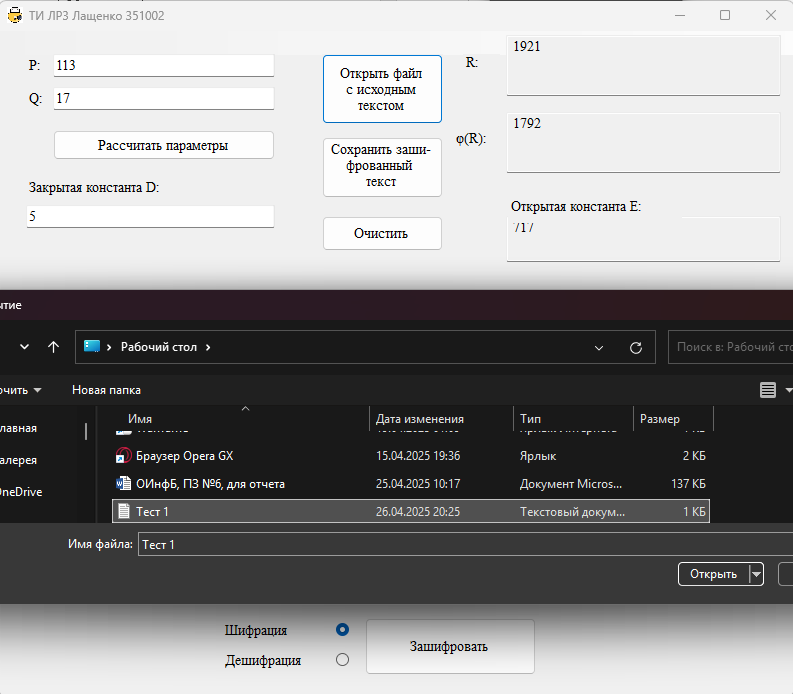
Результат работы программы – зашифрованный/расшифрованный файл/ы.

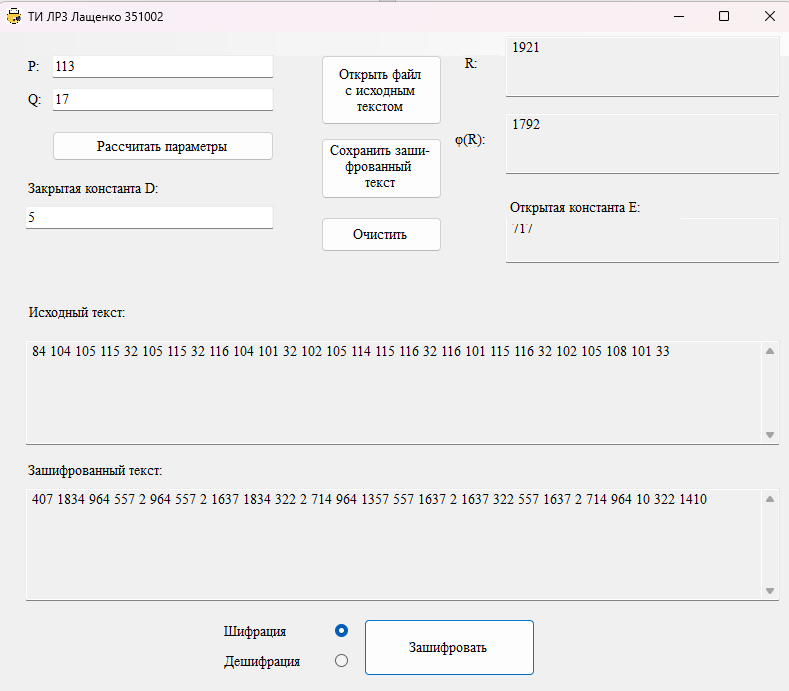
**Тесты на разные типы файлов**

1. Текстовый файл:

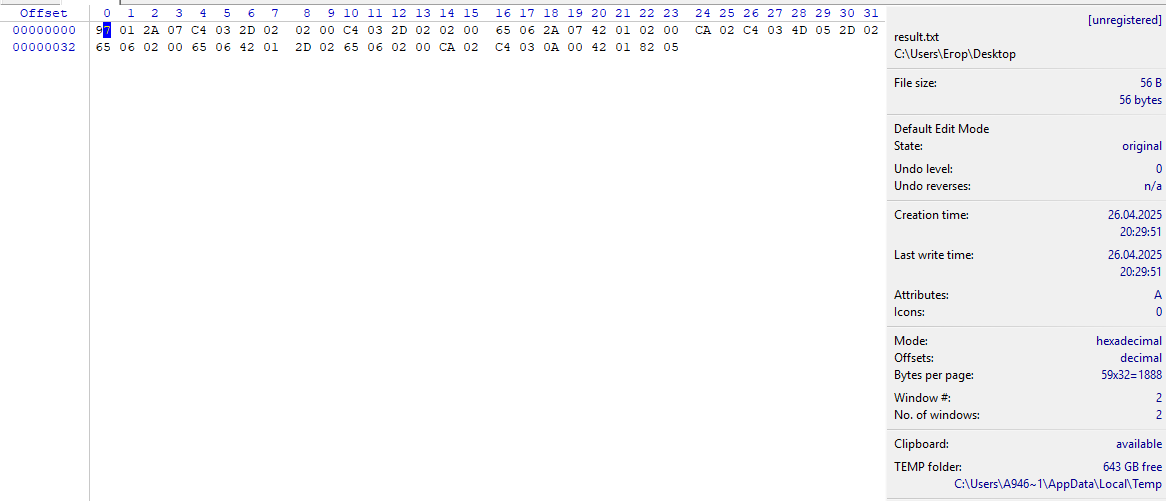
Исходное содержимое:

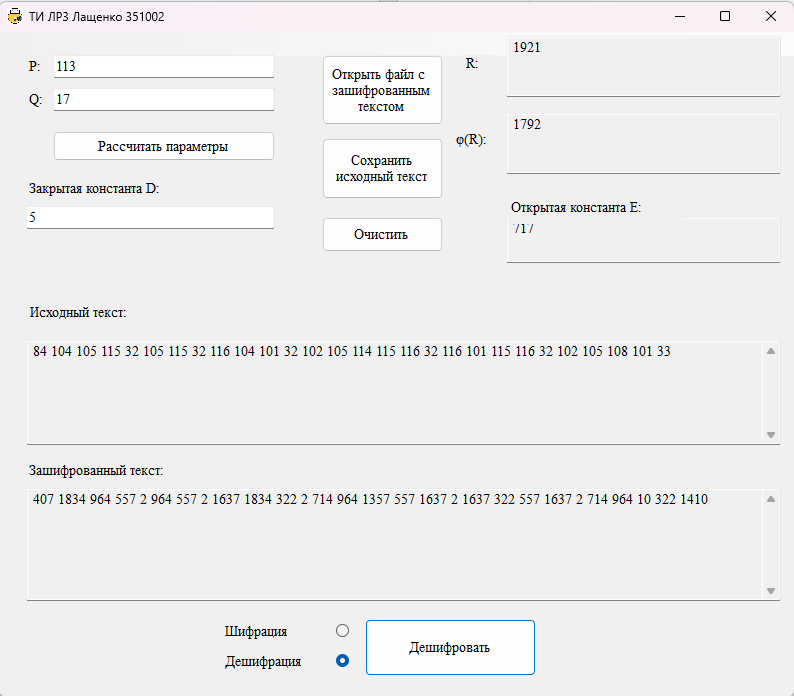


Отображение в программе:  


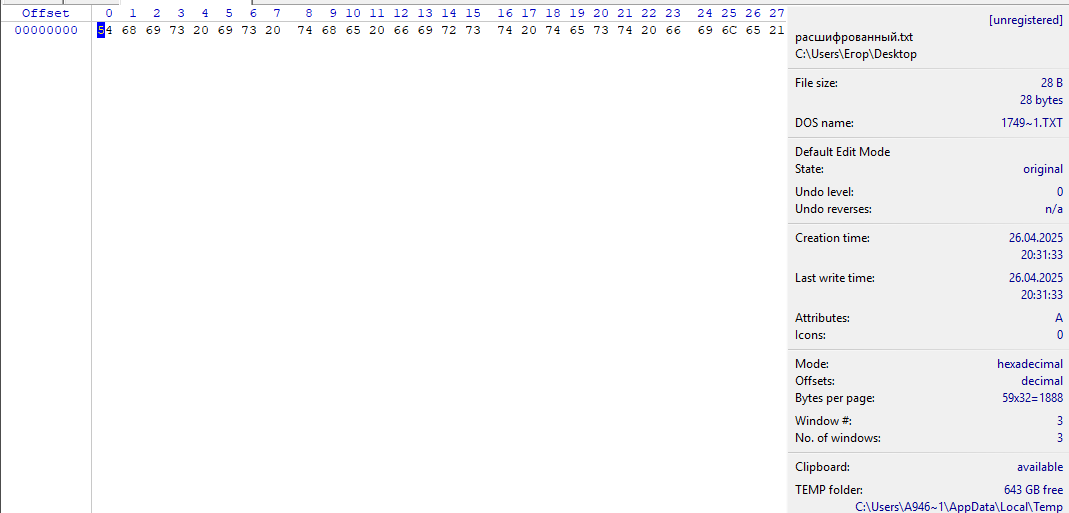
Результат шифрования:  


Отображение зашифрованного содержимого в файле:



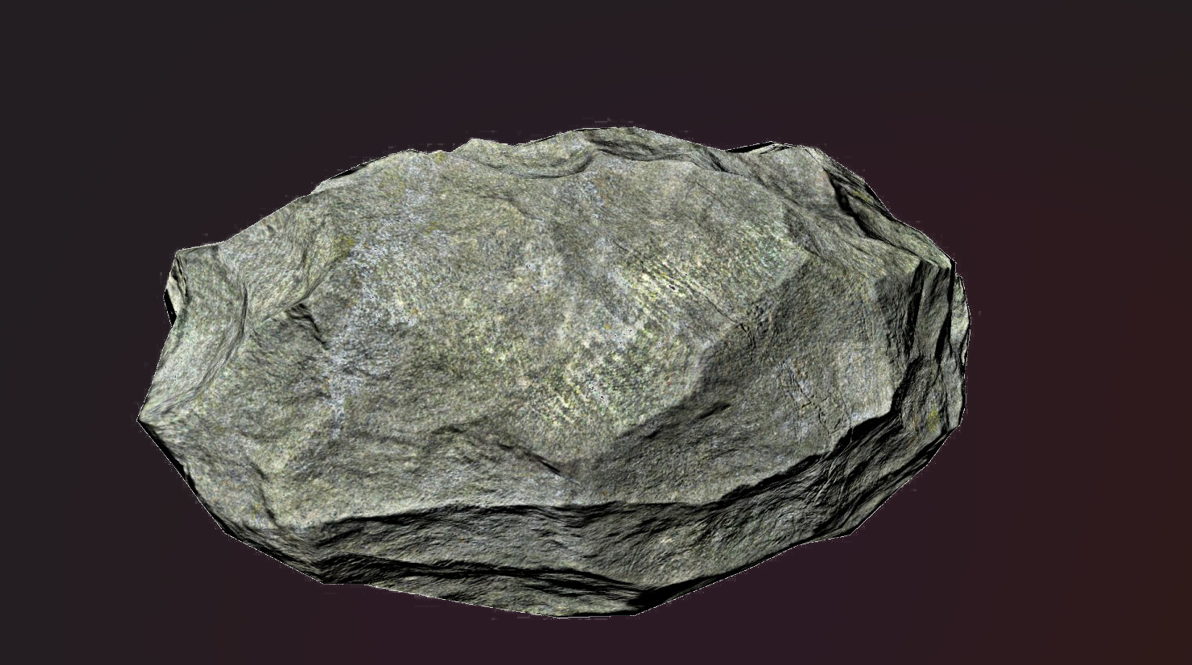
Результат расшифровки:  


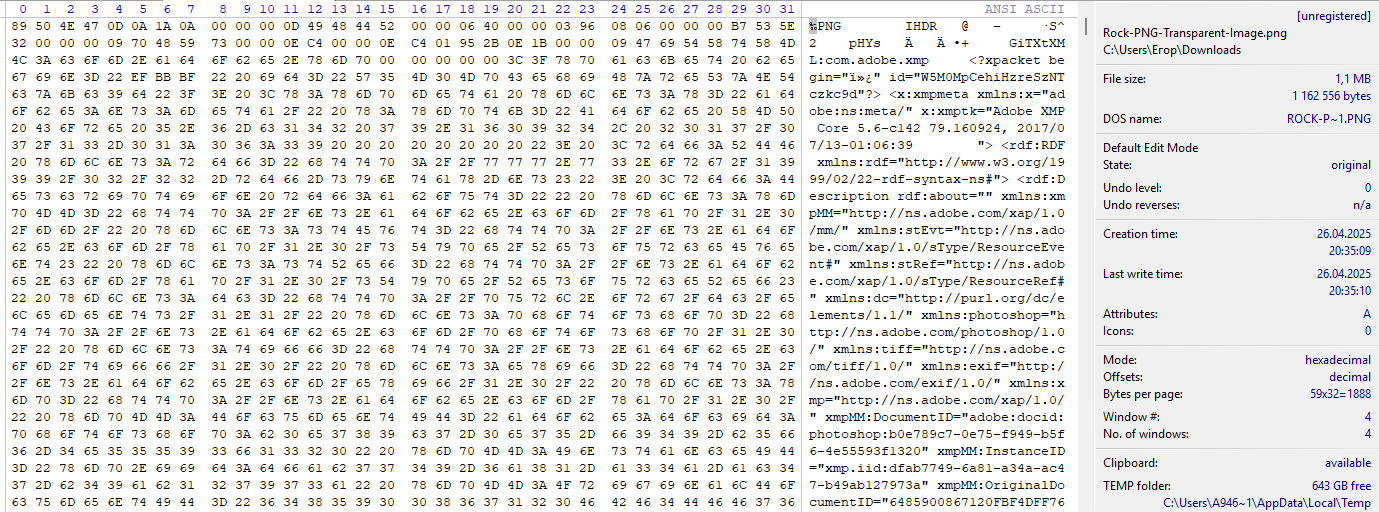
Результат расшифровки записанный в файл:

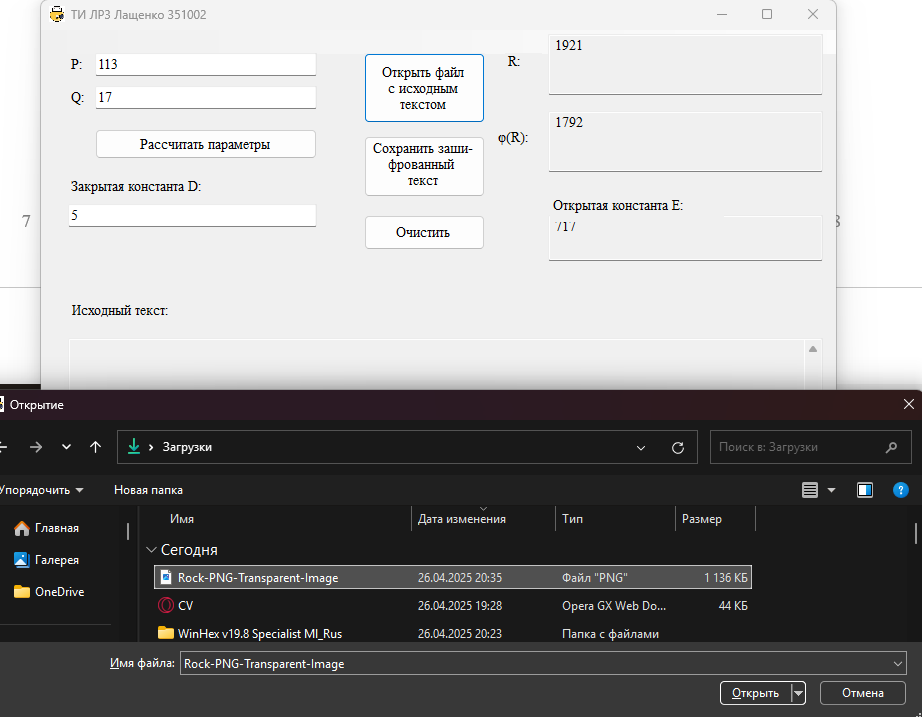


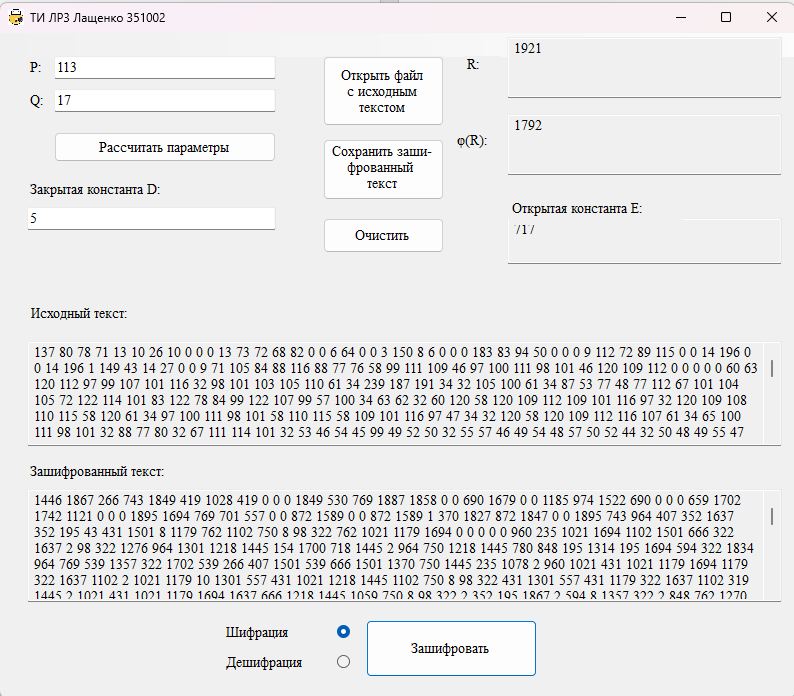
1. Изображение(png-файл):

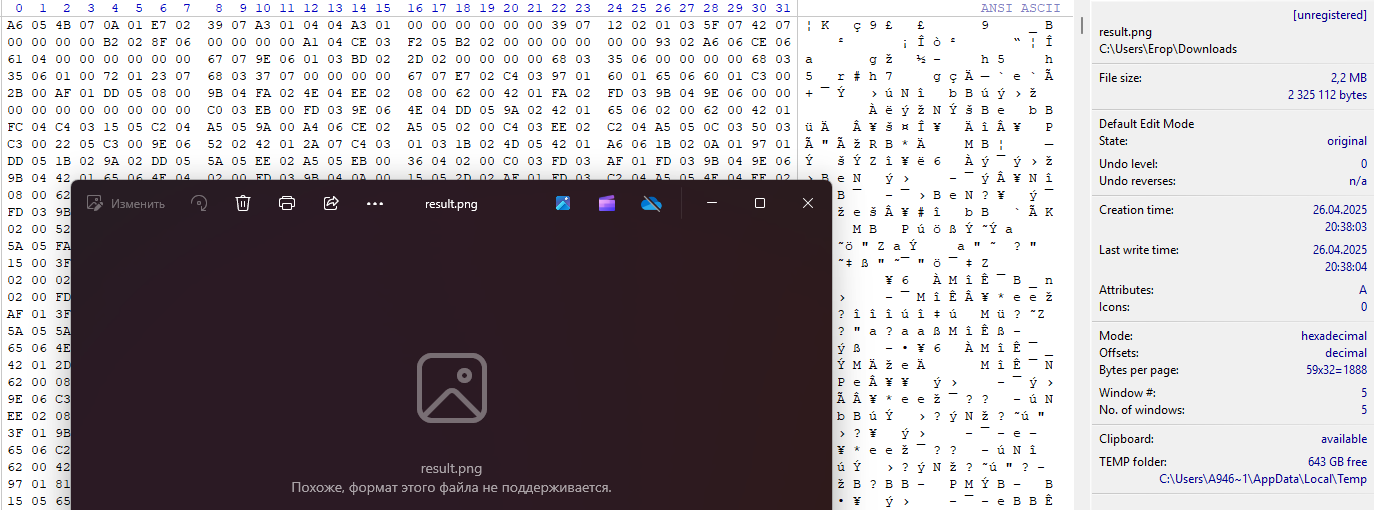
Исходное содержимое:

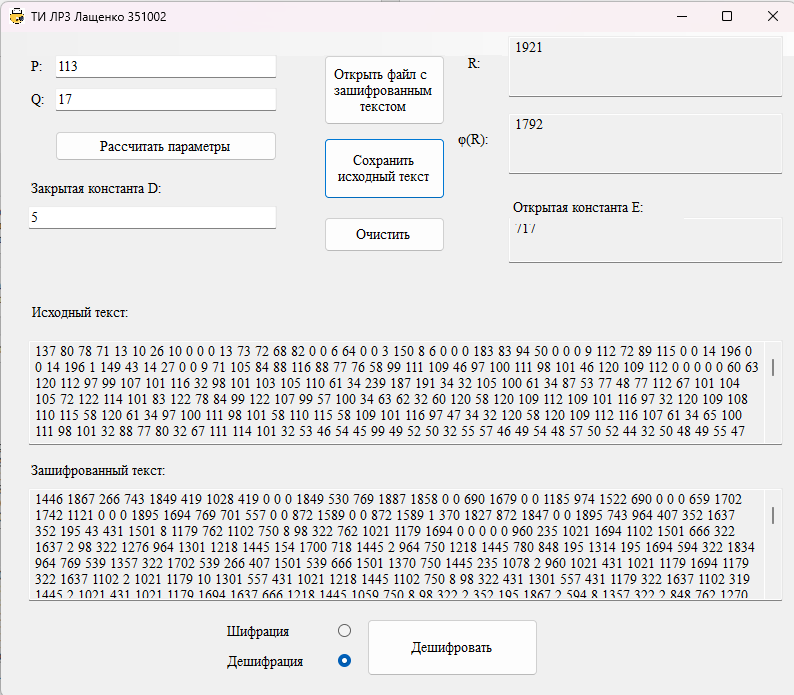


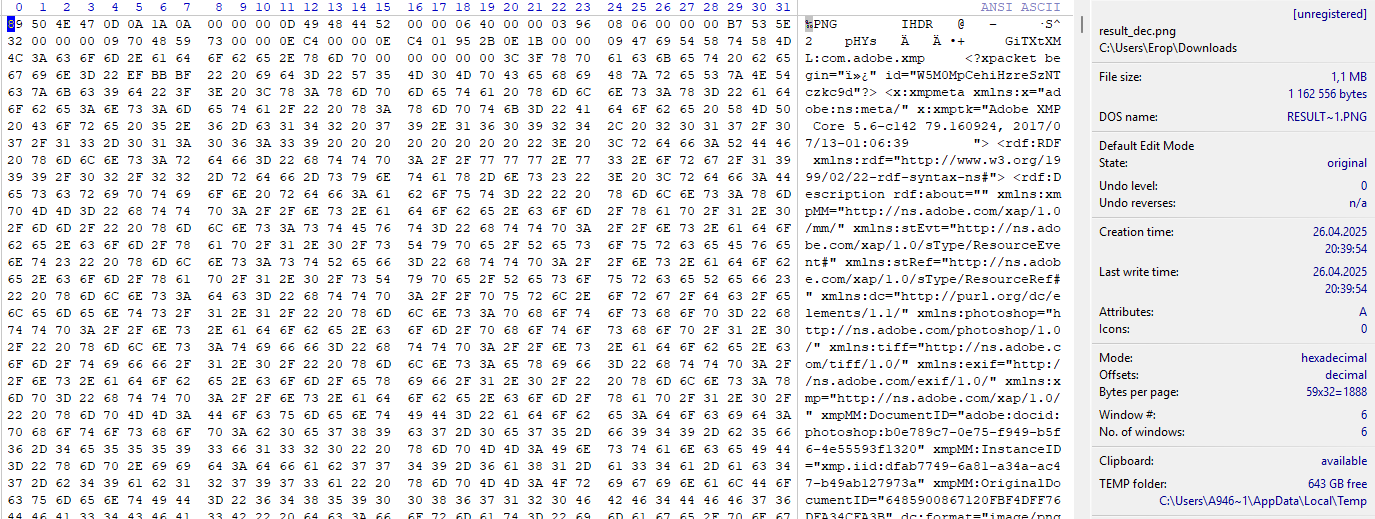


Отображение в программе:  


Результат шифрования:  


Отображение зашифрованного содержимого в файле:

Результат расшифровки:  


Результат расшифровки записанный в файл:  


Отображение расшифрованного файла:  
