УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №3

по предмету

Теория Информации

Вариант 1

Выполнил:

Жигимонт А.Н.

Проверил:

Болтак С.В.

Группа 351003

Минск 2025

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

513 mod 7 = ?

* a (основание): 5
* n (показатель степени): 13
* e (чётность n): да/нет
* r (результат): изначально 1
* x – значение mod(7)

x (модуль): 7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Шаг | a | n | e | Действие | r mod x |
| 1 | 5 | 13 | нет | r = 1 \* 5 mod 7 = 5 n = 13 - 1 = 12 | 5 |
| 2 | 5 | 12 | да | a = 5^2 mod 7 = 4 n = 12 / 2 = 6 | 5 |
| 3 | 4 | 6 | да | a = 4^2 mod 7 = 2 n = 6 / 2 = 3 | 5 |
| 4 | 2 | 3 | нет | r = 5 \* 2 mod 7 = 3 n =3 – 1 = 2 | 3 |
| 5 | 2 | 2 | да | a = 2^2 mod 7 = 4 n = 2 / 2 = 1 | 3 |
| 6 | 4 | 1 | нет | r = 3 \* 4 mod 7 = 5 n =1 – 1 = 0 | 5 |

513 mod 7 = 5

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

1. Дано простое p = 47
2. Ищем простые делители p-1 = 46 = 2 \* 23
3. Проверяем, является ли число 5 первообразным корнем по модулю 47:

5 ^ 23 mod 47= 46; 5 ^ 2 mod 47 = 25. Число 5 является первообразным по модулю 43.

1. Если найден один первообразный корень g по модулю p, остальные корни имеют вид g^k, где НОД(k, p-1) = 1. Для p - 1 = 46, g = 5, допустимые k: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45:

5 ^ 1 mod 47 = 5

5 ^ 3 mod 47 = 31

5 ^ 5 mod 47 = 23

5 ^ 7 mod 47 = 11

5 ^ 9 mod 47 = 40

5 ^ 11 mod 47 = 13

5 ^ 13 mod 47 = 43

5 ^ 15 mod 47 = 41

5 ^ 17 mod 47 = 38

5 ^ 19 mod 47 = 10

5 ^ 21 mod 47 = 15

5 ^ 25 mod 47 = 22

5 ^ 27 mod 47 = 33

5 ^ 29 mod 47 = 26

5 ^ 31 mod 47 = 39

5 ^ 33 mod 47 = 35

5 ^ 35 mod 47 = 29

5 ^ 37 mod 47 = 20

5 ^ 39 mod 47 = 30

5 ^ 41 mod 47 = 45

5 ^ 43 mod 47 = 44

5 ^ 45 mod 47 = 19

Тогда все первообразные корни для модуля p = 47 это 5, 10, 11, 13, 15, 19, 20, 22, 23, 26, 29, 30, 31, 33, 35, 38, 39, 40, 41, 43, 44, 45.

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

x1\*a + y1\*b = НОД(a,b), a = 619, b = 432, НОД(a,b) = 1, т.к a и b взаимно простые.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **итерация** | **Делимое** | **Делитель** | **Частное** | **Остаток** |
| 1 | 619 | 432 | 1 | 187 |
| 2 | 432 | 187 | 2 | 58 |
| 3 | 187 | 58 | 3 | 13 |
| 4 | 58 | 13 | 4 | 6 |
| 5 | 13 | 6 | 2 | 1 |
| 6 | 6 | 1 | 6 | 0 |

Начинаем обратный ход с предпоследней операции:  
1 = 13 – 6\*2;  
1 = 13 – (58 – 13\*4) \* 2 = 13 \* 9 – 58 \* 2; (подставляем выражение для 6)

1 = (187 – 58 \* 3) \* 9 – 58 \* 2 = 187 \* 9 – 58 \* 29 (подставляем выражение для 13)

1 = 187 \* 9 – (432 – 187 \* 2) \* 29 = 187 \* 67 – 432 \* 29 (подставляем выражение для 58)

1 = (619 - 432) \* 67 – 432 \* 29 = 619 \* 67 – 432 \* 96 (подставляем выражение для 187)  
Получаем ответ: x = 67,y = −96

**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.

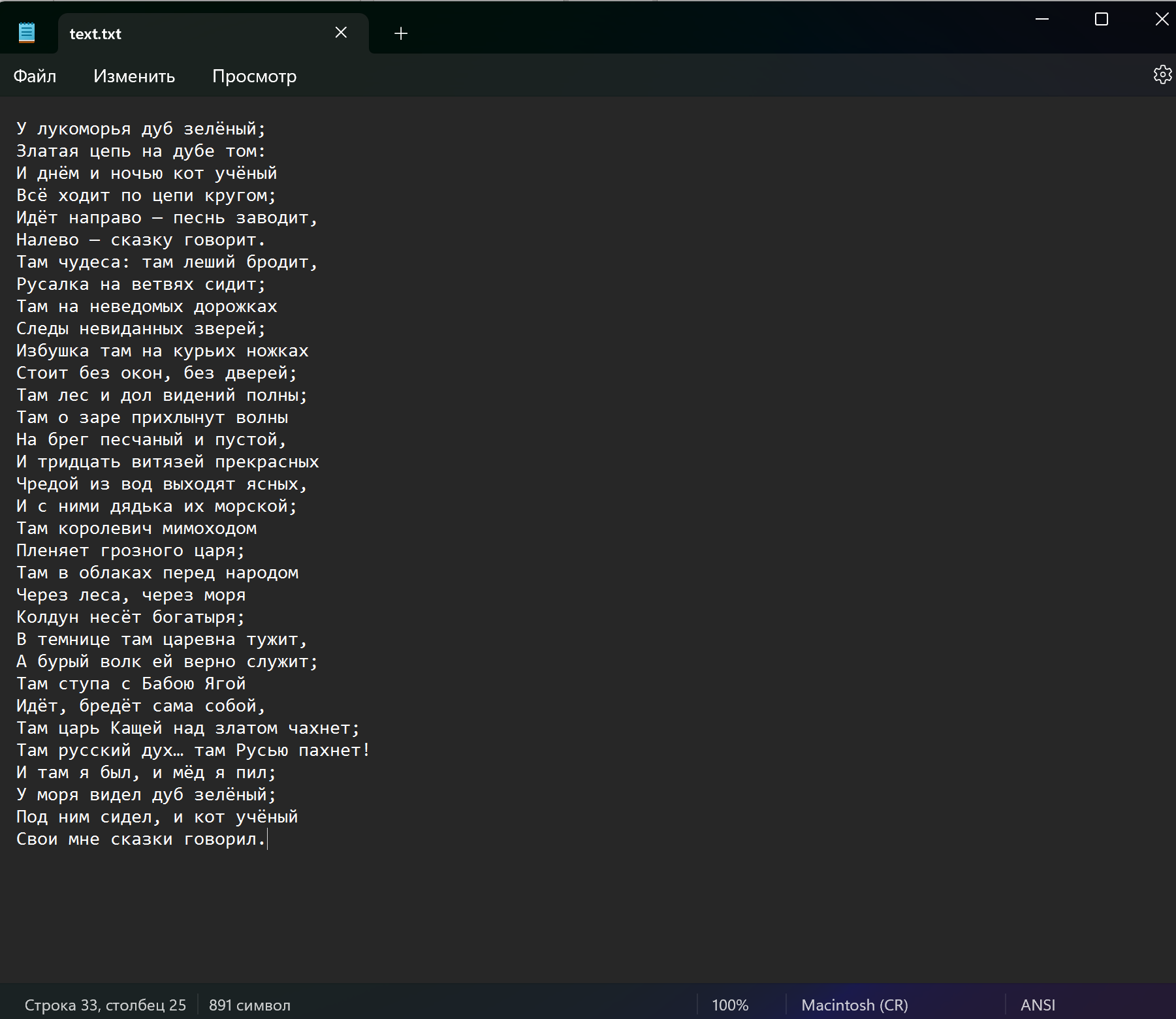
Использовать алгоритм быстрого возведения в степень и расширенный алгоритм Евклида.

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

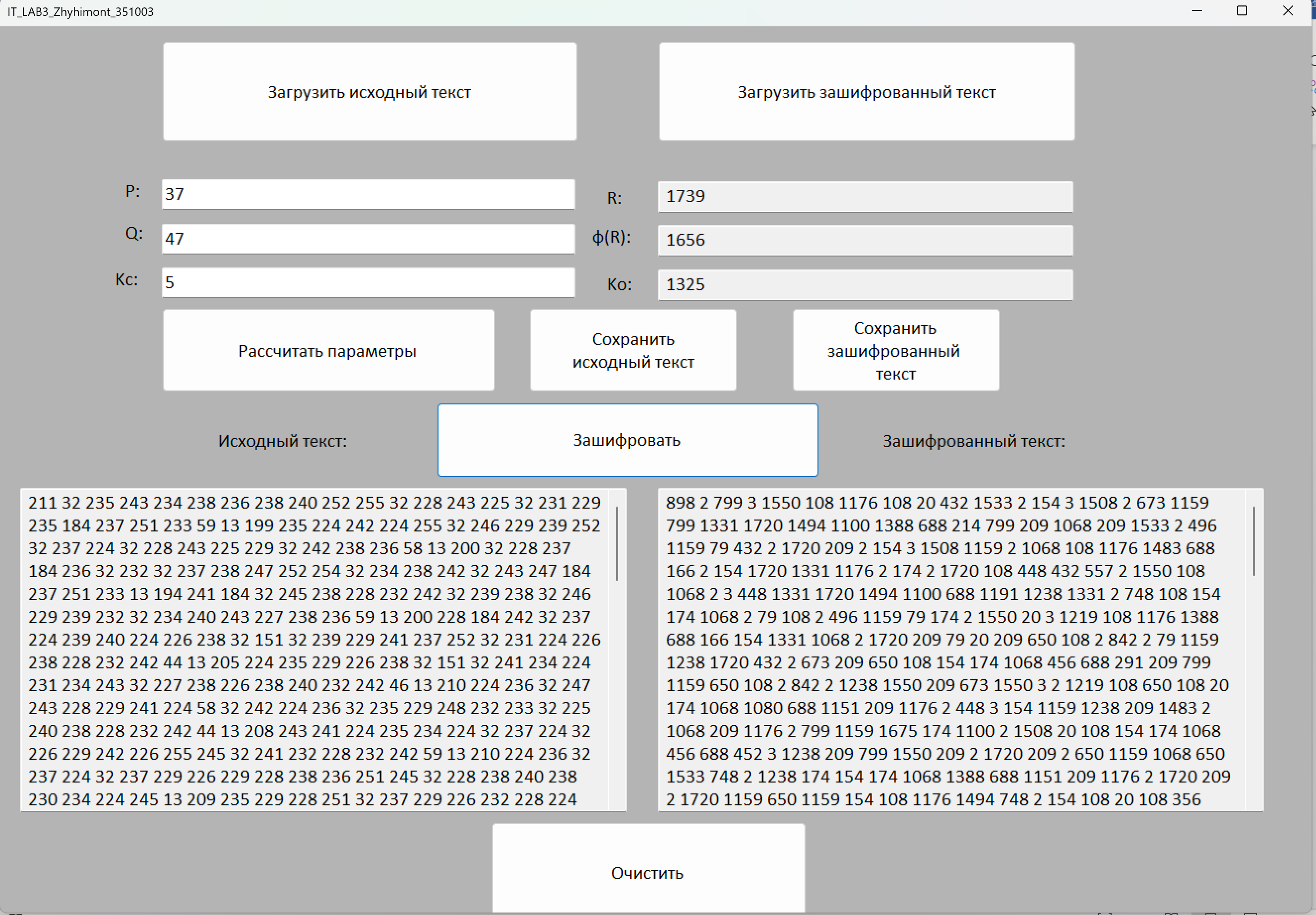
Результат работы программы – зашифрованный/расшифрованный файл/ы.

Тесты:

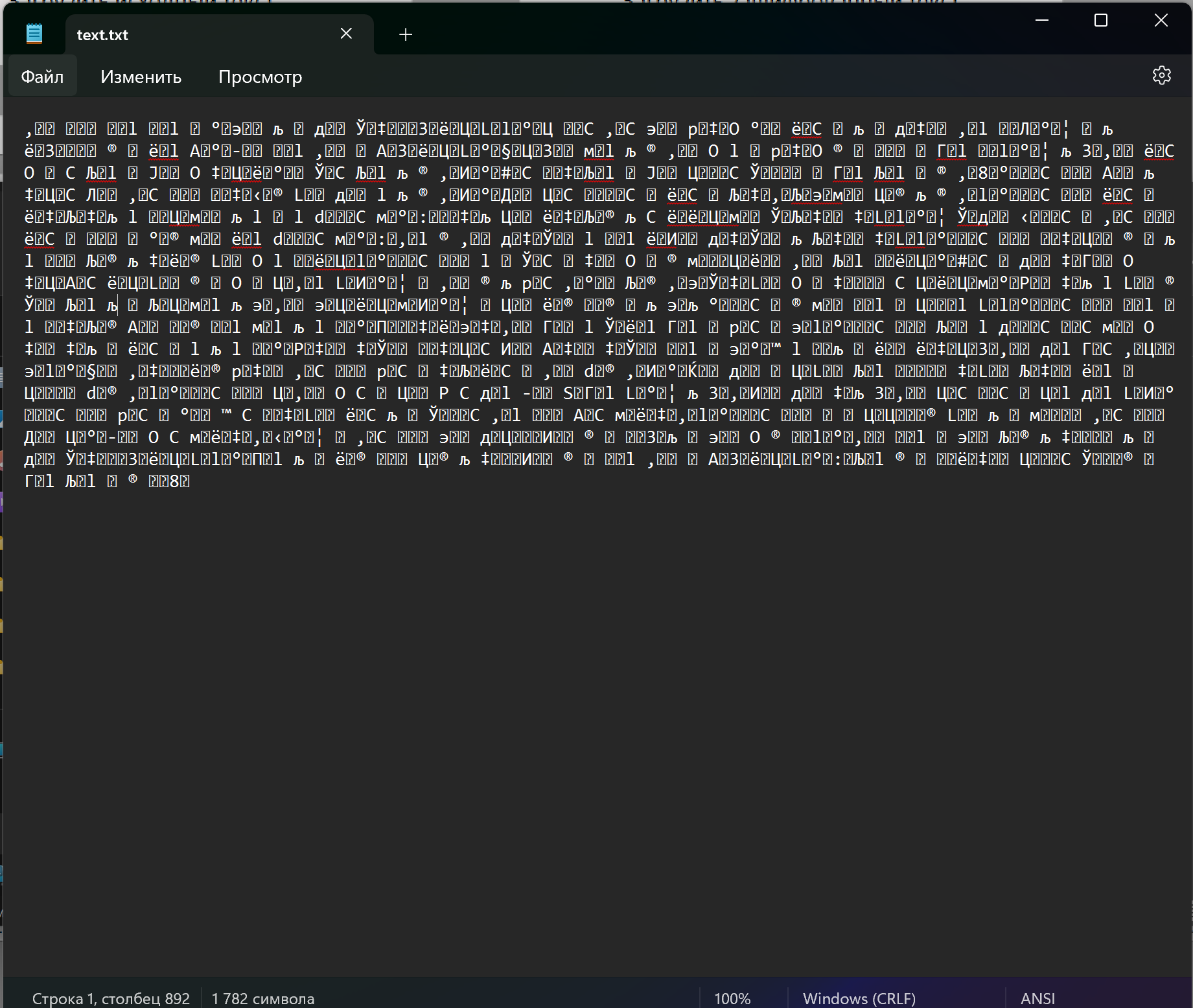
Шифрование текста:



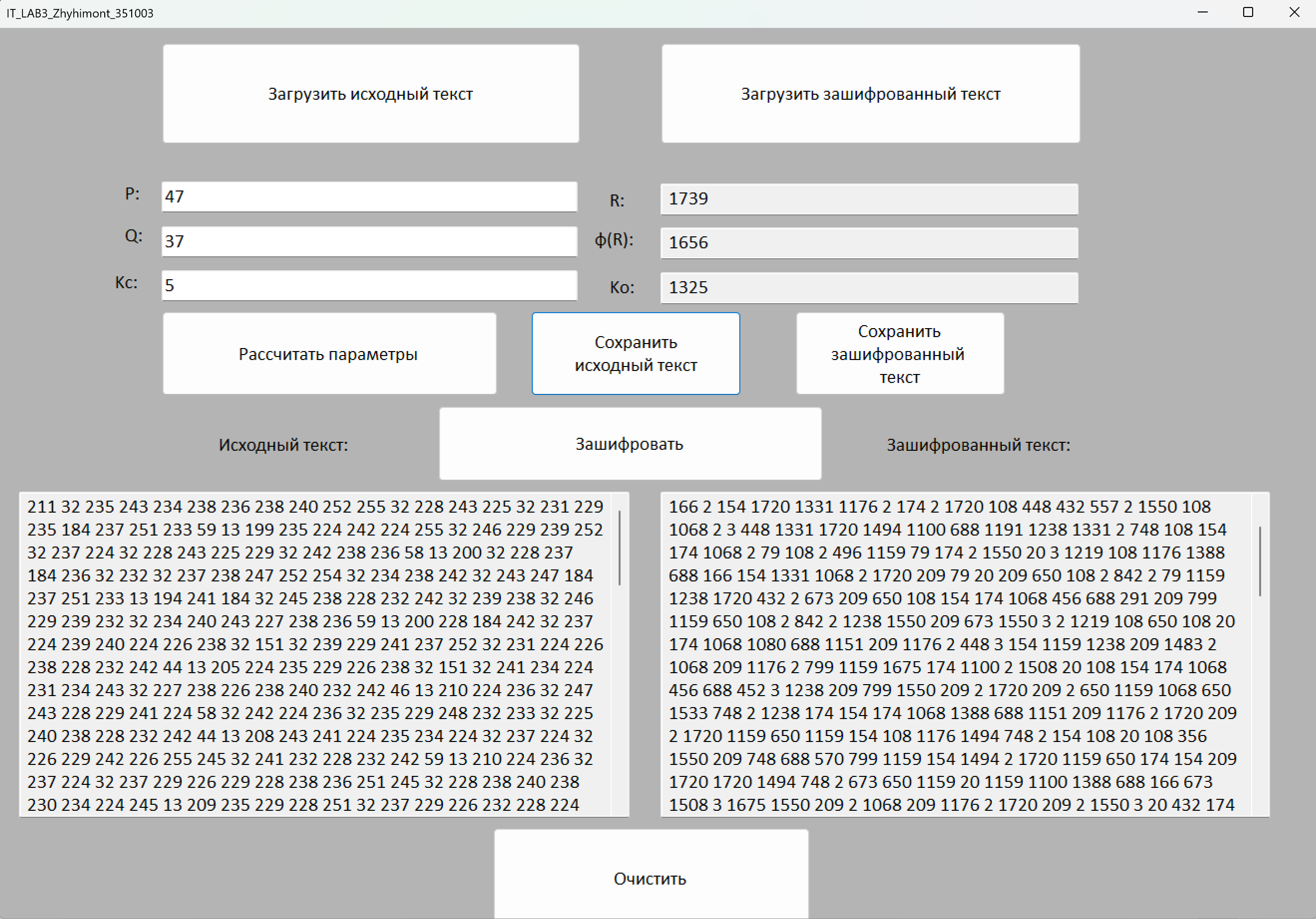
Результат работы шифратора:



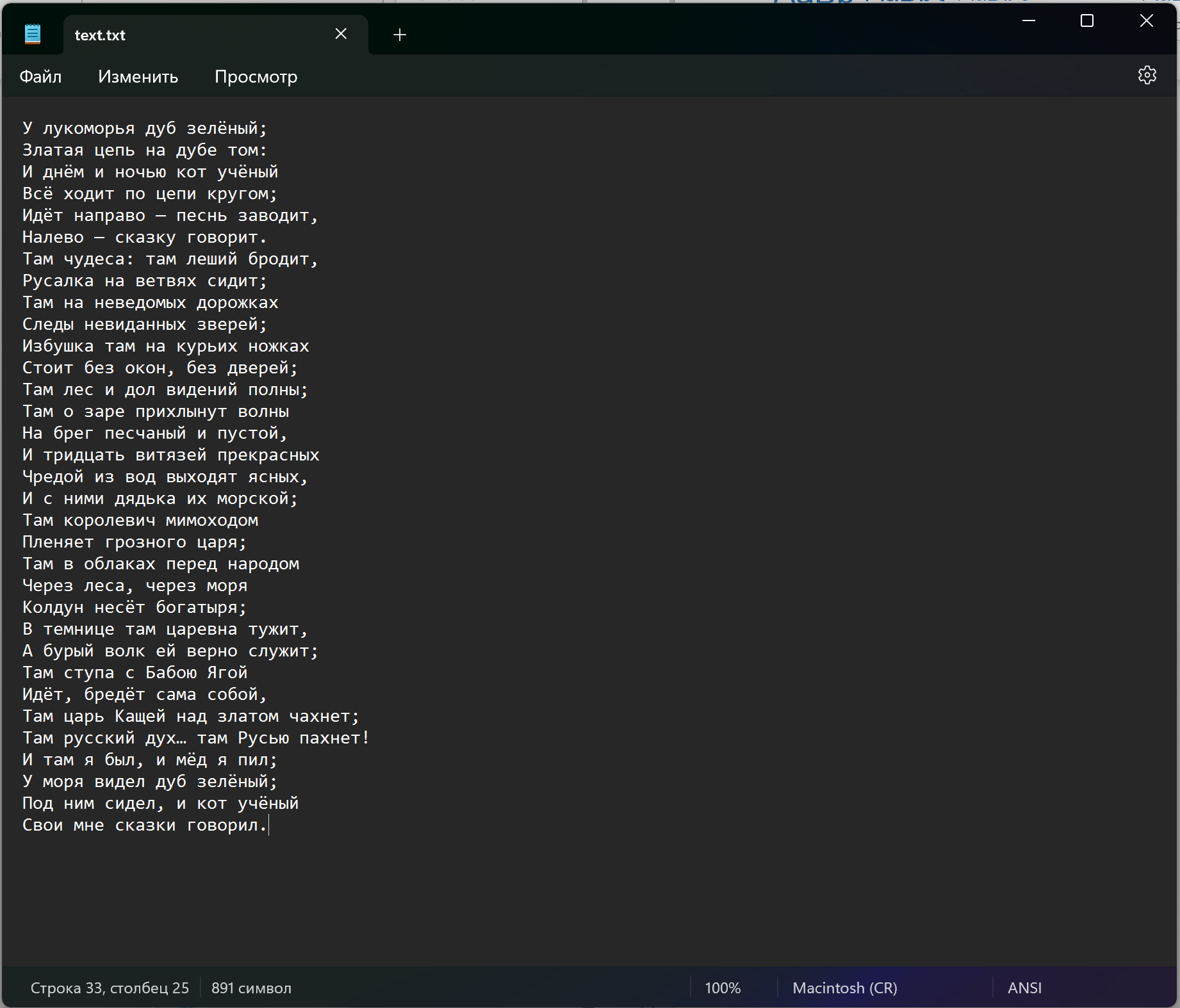
Зашифрованный текст:



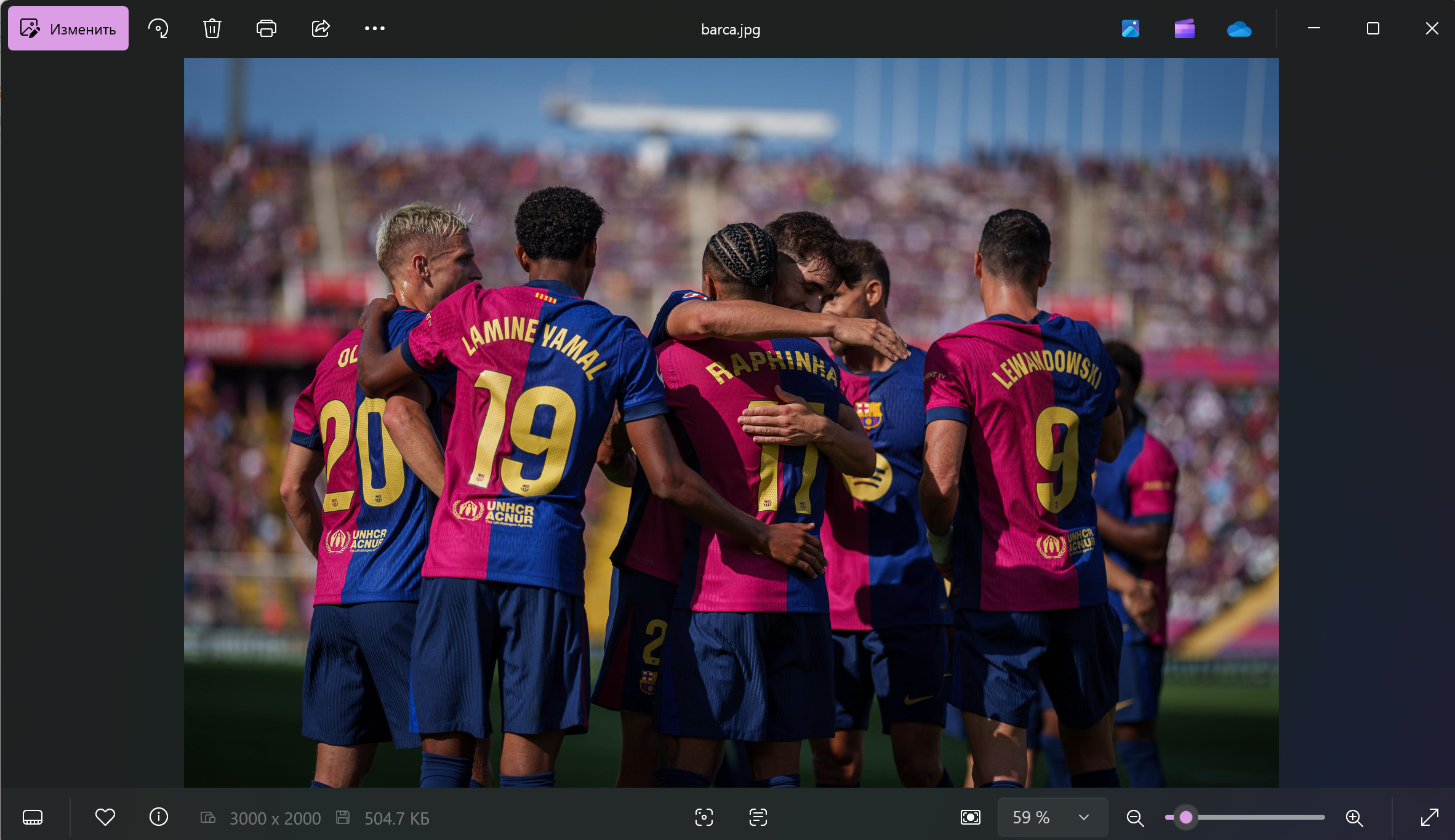
Результат работы дешифратора:



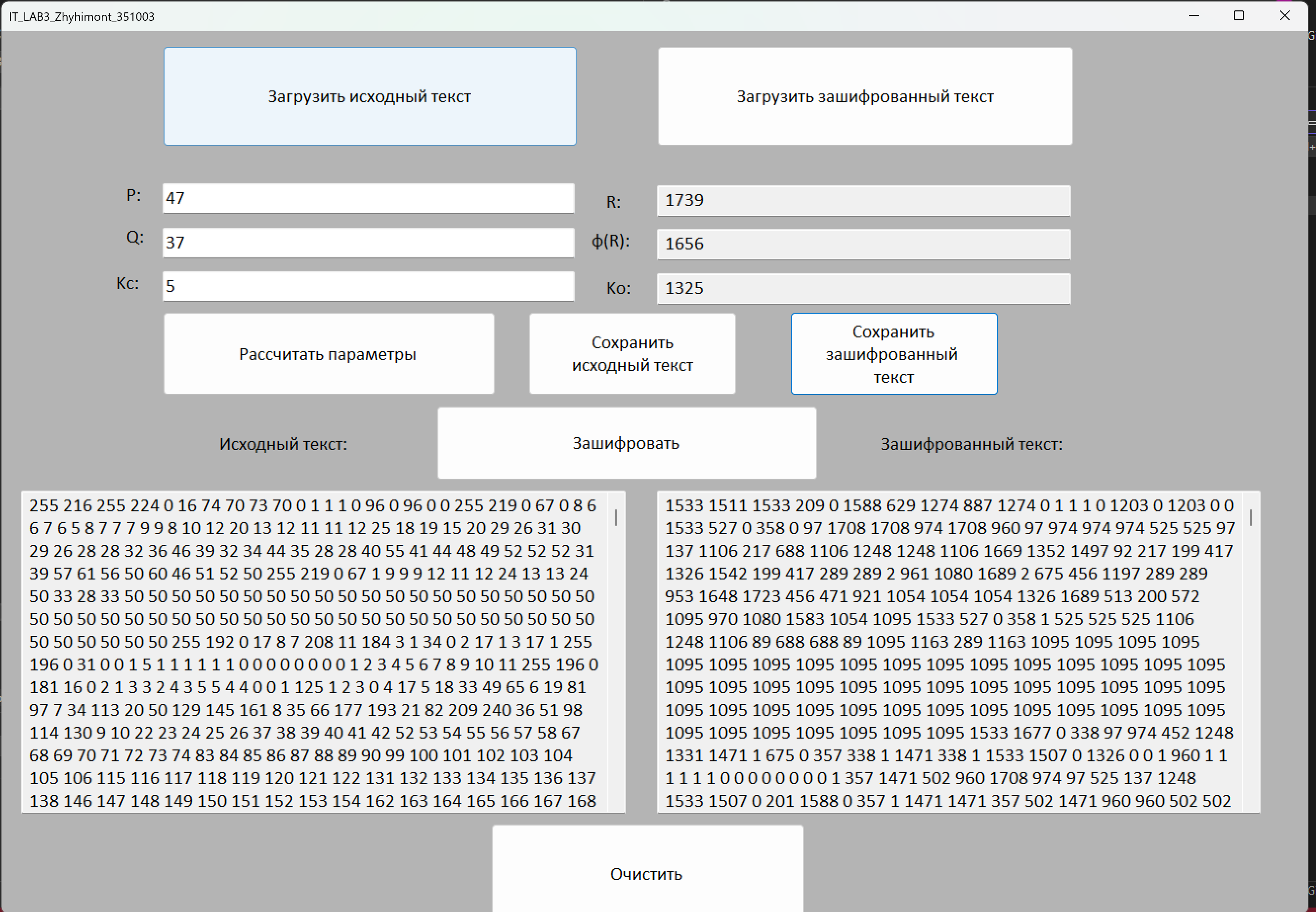
Расшифрованный текст:



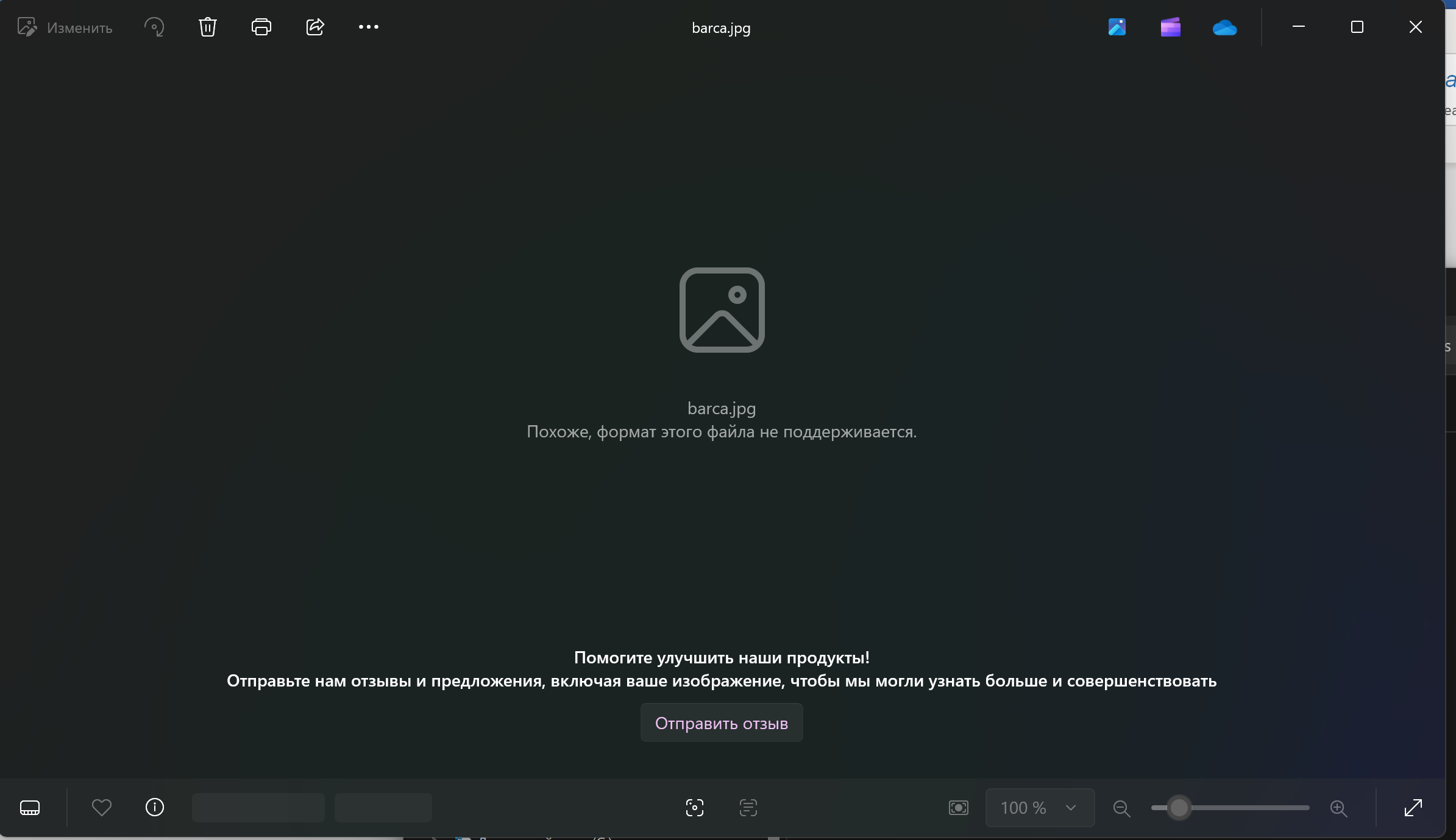
Шифрование фото:

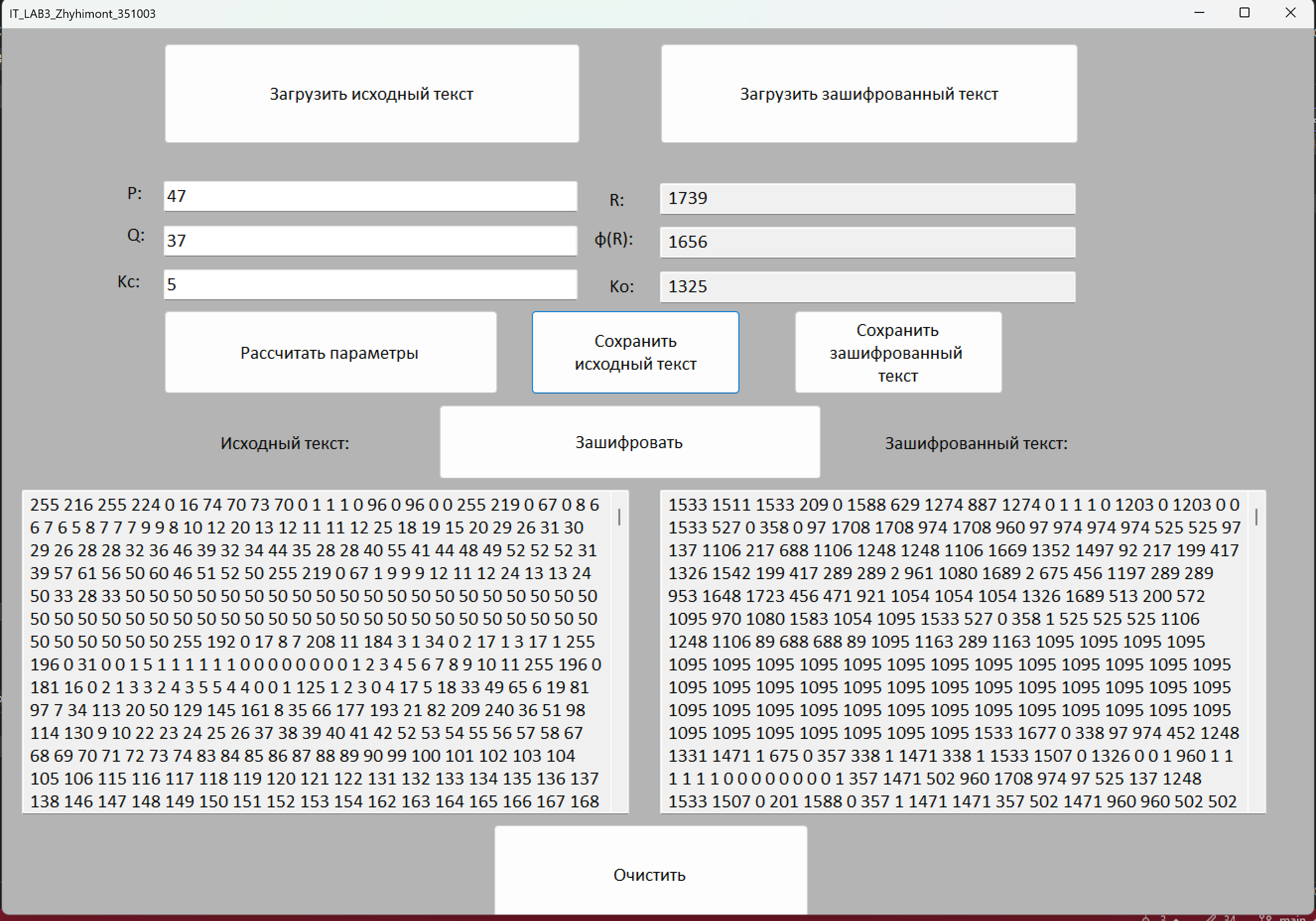


Результат работы шифратора:



Зашифрованная фотография



Результат работы дешифратора:

Расшифрованное фото:

