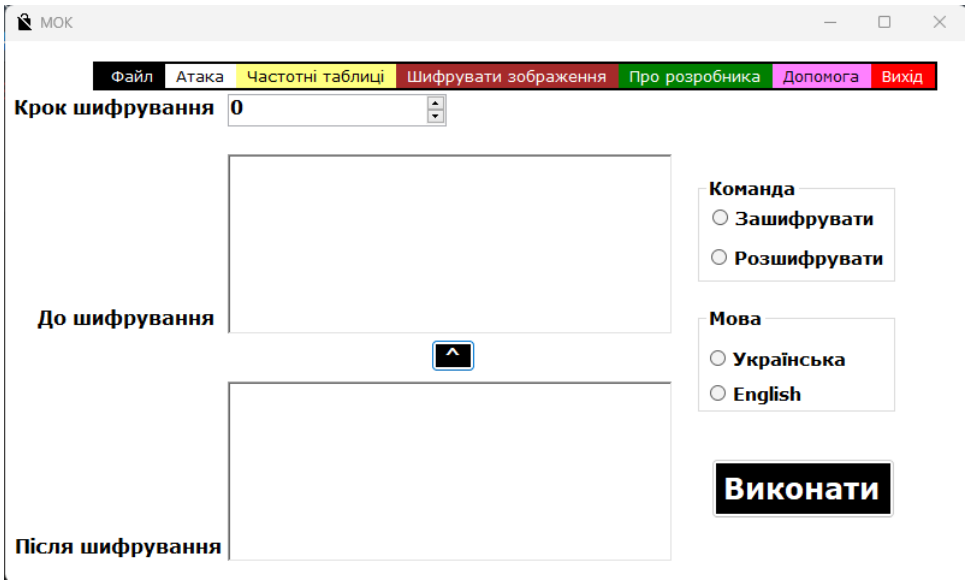
Лабораторна робота 1

Шифр зсуву

Кравець Ольга

ПМО-31

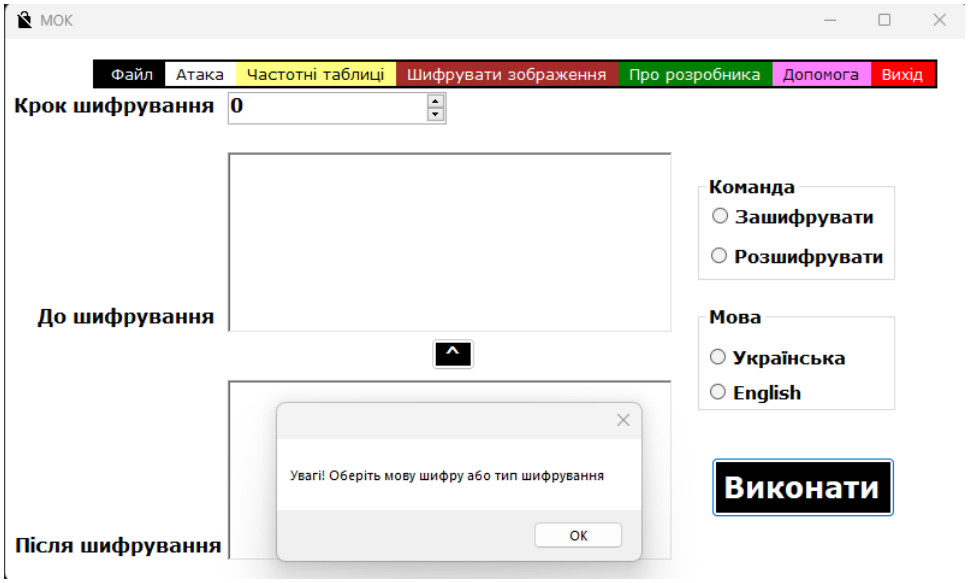
Інтерфейс криптографічної системи:



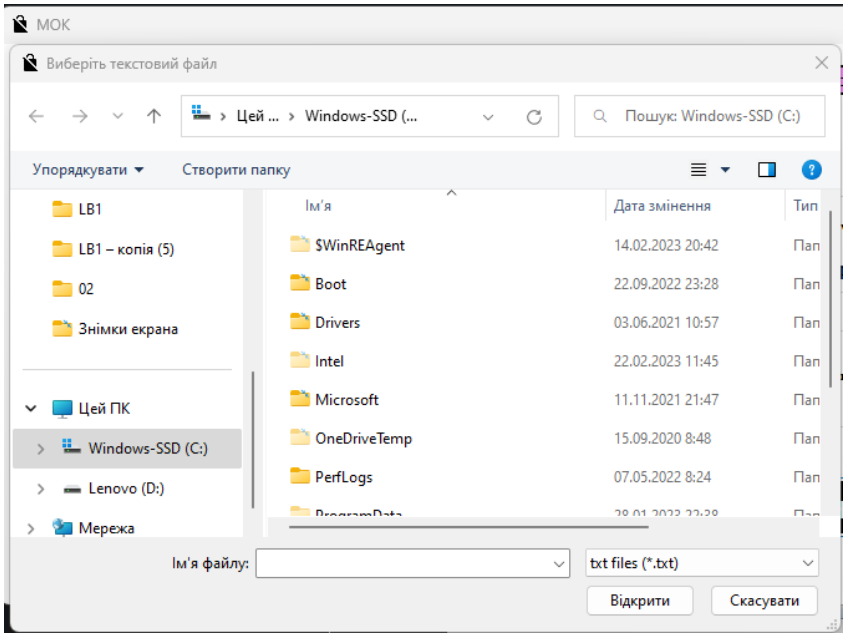
Валідація ключа: стоїть обмеження numericUpDown, яке не дозволяє писати не число в комірку

```
private void numericUpDown1_ValueChanged(object sender, EventArgs e)
{
    numericUpDownINT = (int)numericUpDown1.Value;
}
```

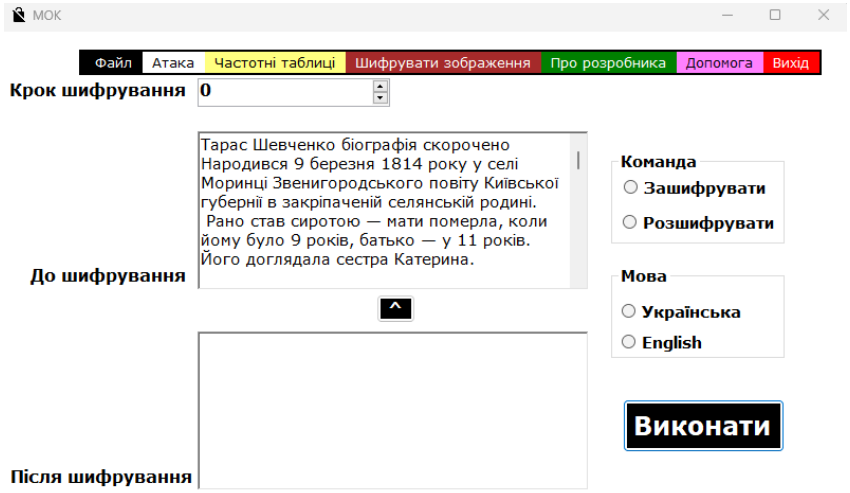
При спробі зашифрувати пусте поле видає помилку



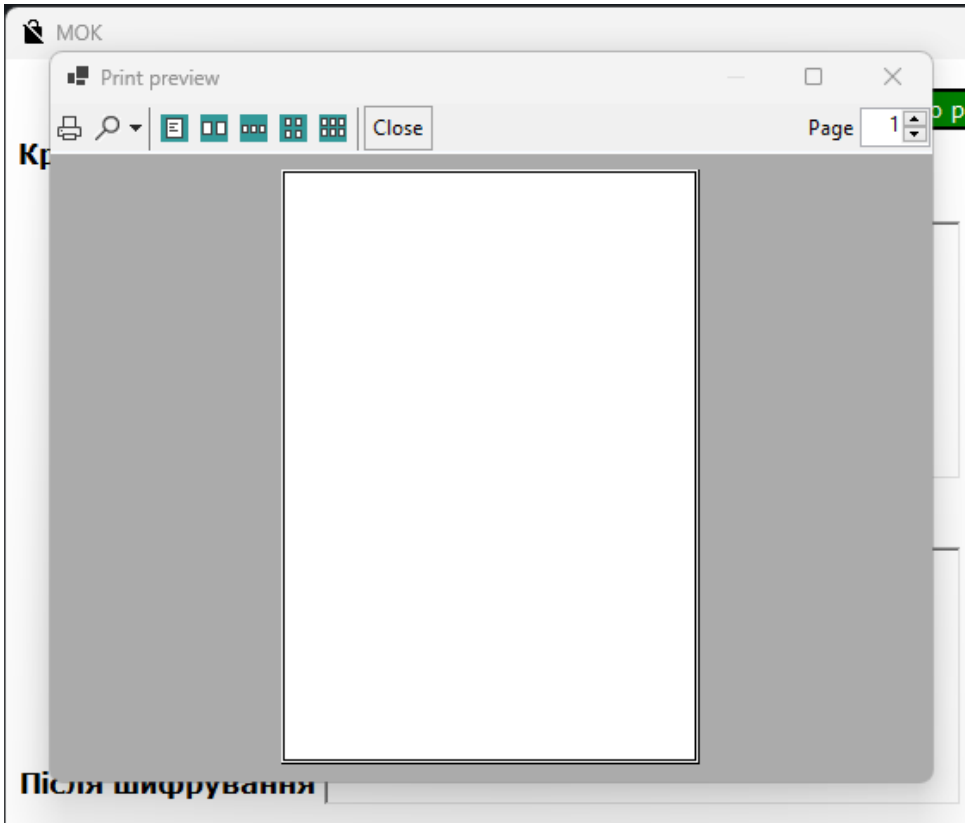
Відкриття файлу:



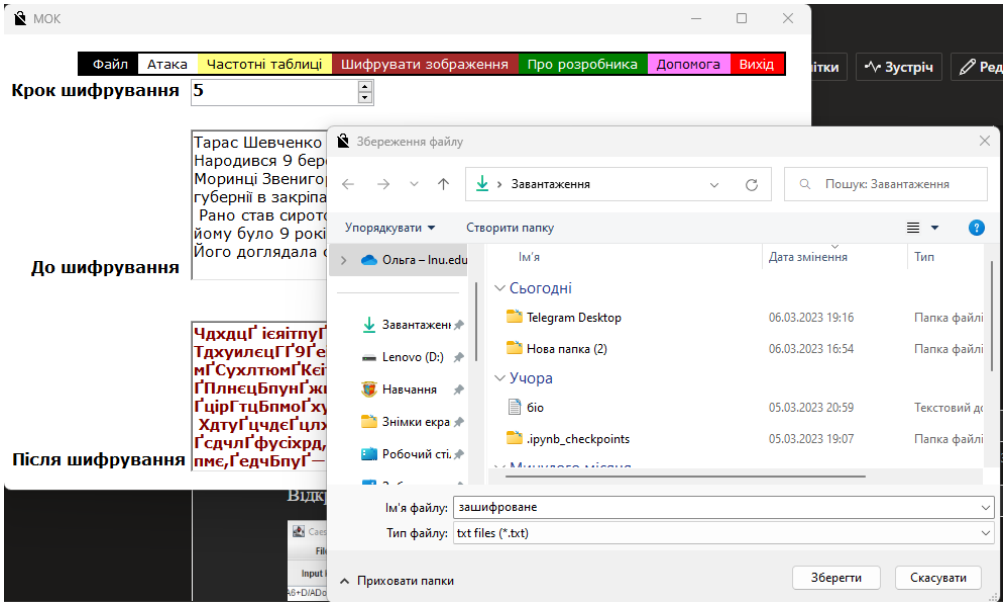
У випадку текстового файлу текст переноситься у вибрану комірку



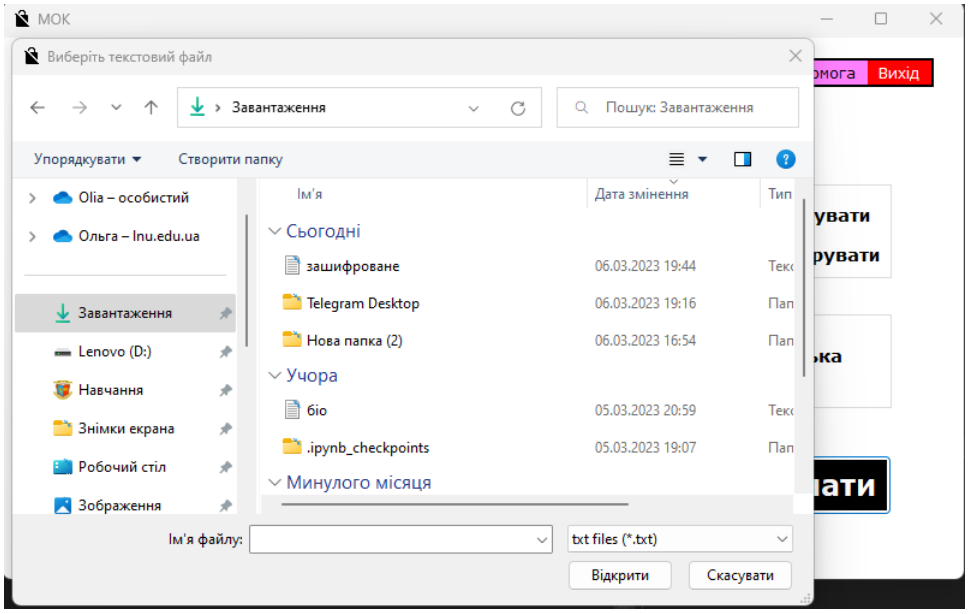
Друк файлу:

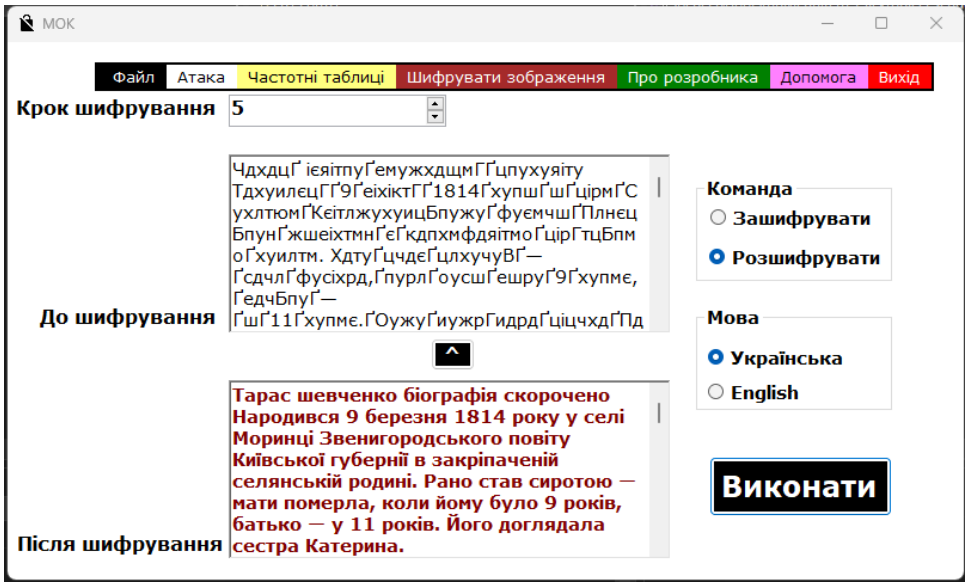


Збереження зашифрованого тексту:

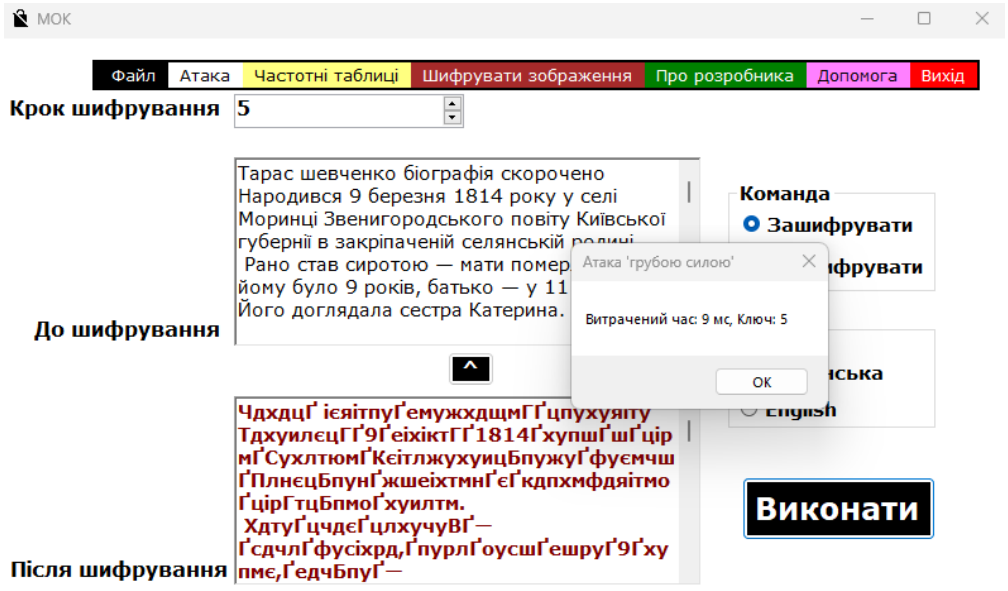


Відкриття та дешифрація:



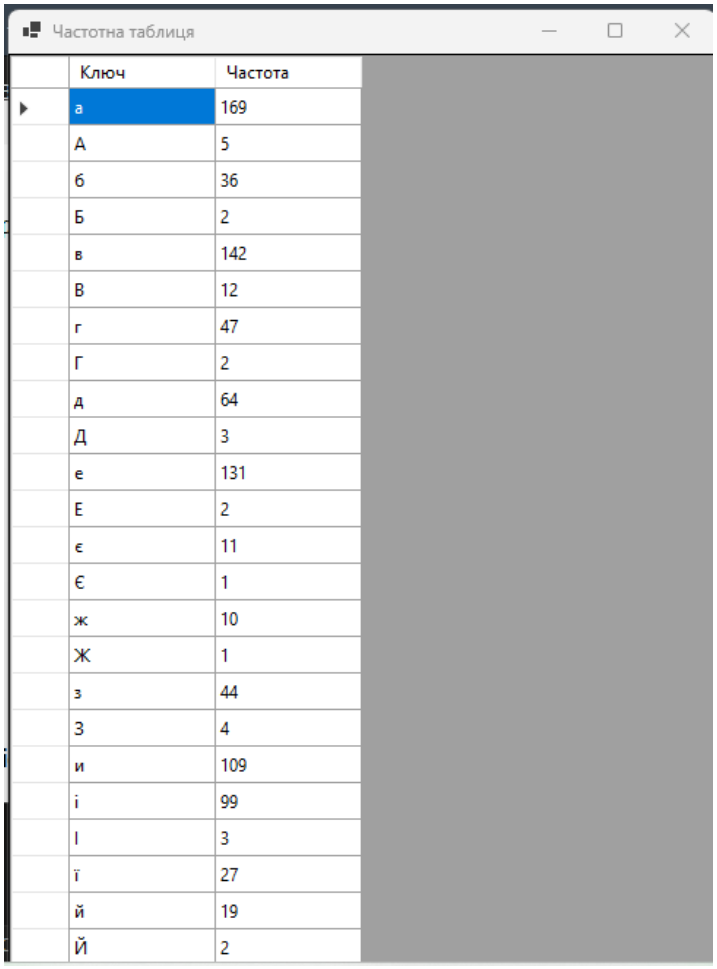


Атака методом “грубої сили”:



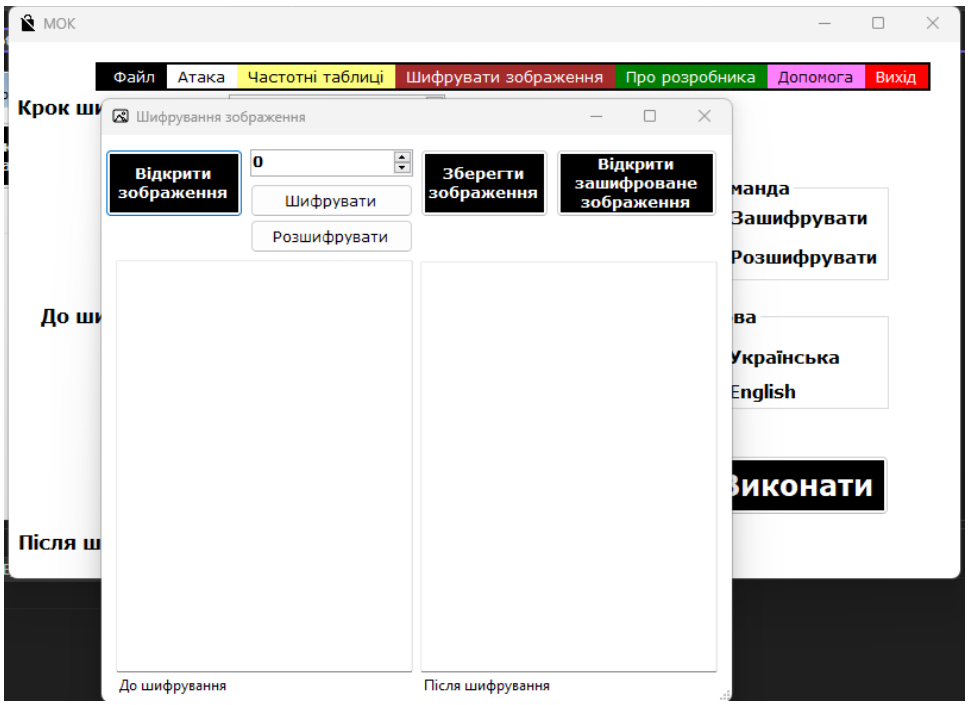
Бачимо час та підібраний ключ

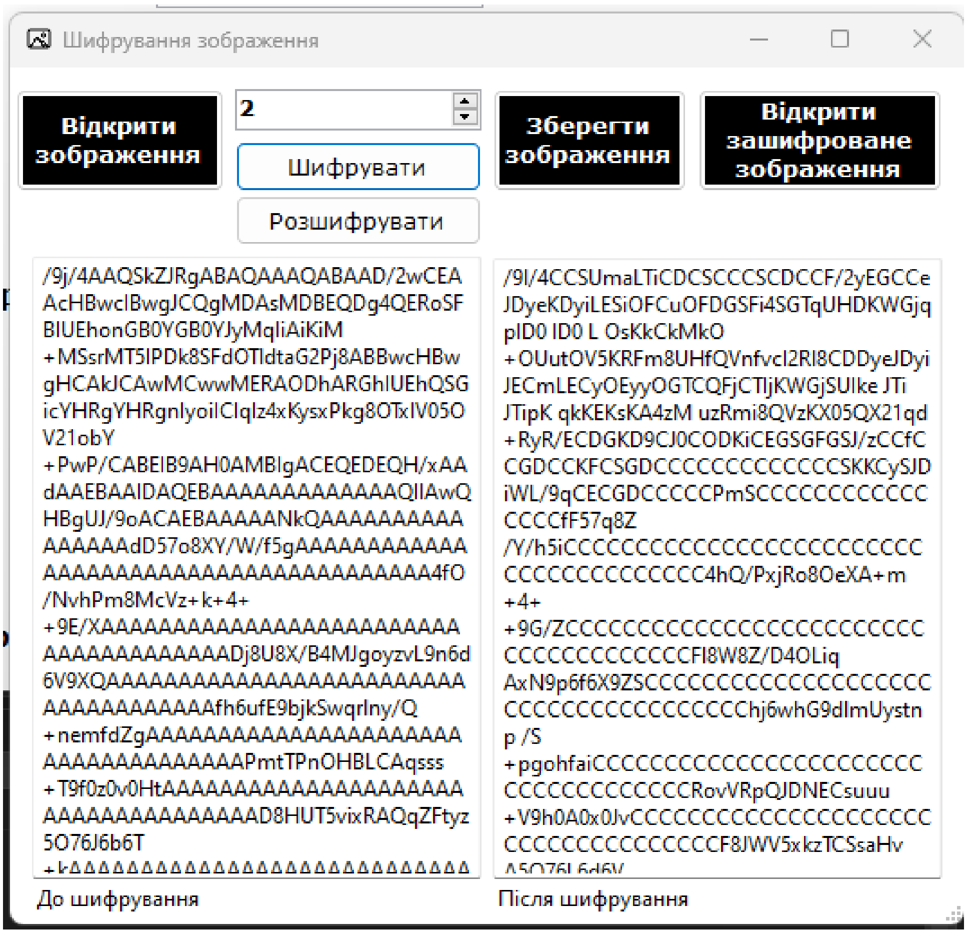
Частотні таблиці:



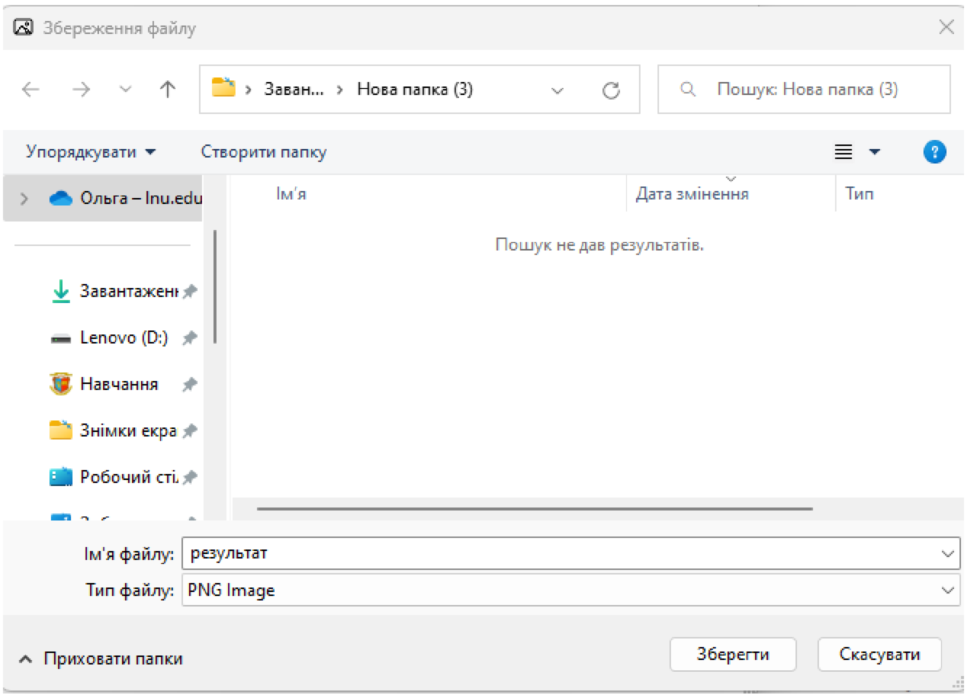
Ключ	Частота
а	169
А	5
б	36
Б	2
в	142
В	12
г	47
Г	2
д	64
Д	3
е	131
Е	2
є	11
Є	1
ж	10
Ж	1
з	44
З	4
и	109
і	99
І	3
ї	27
й	19
Й	2

Шифрування зображення

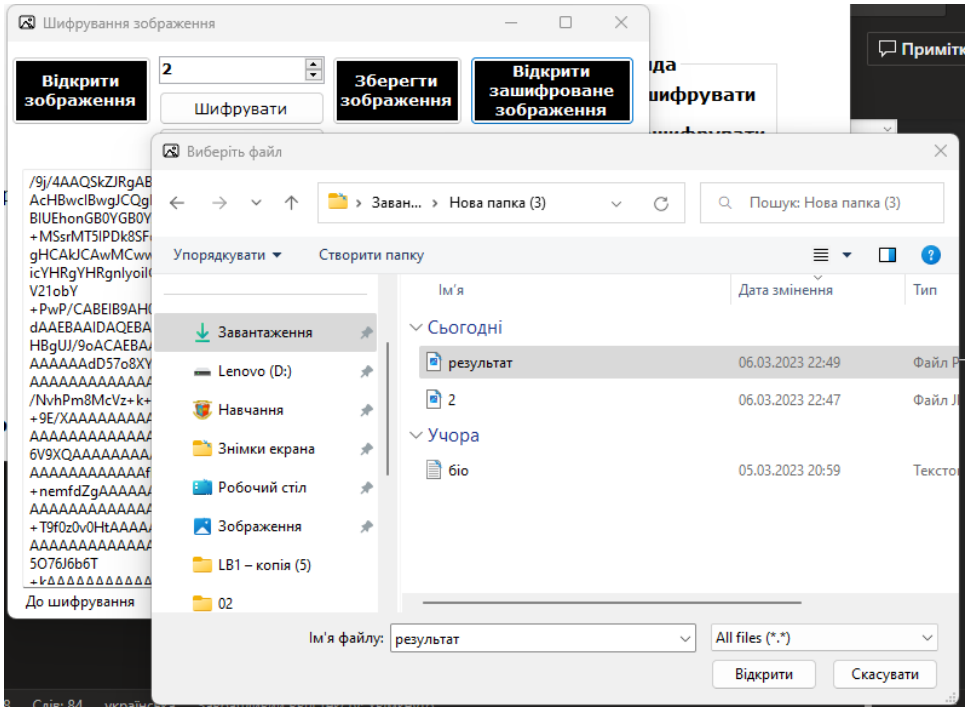


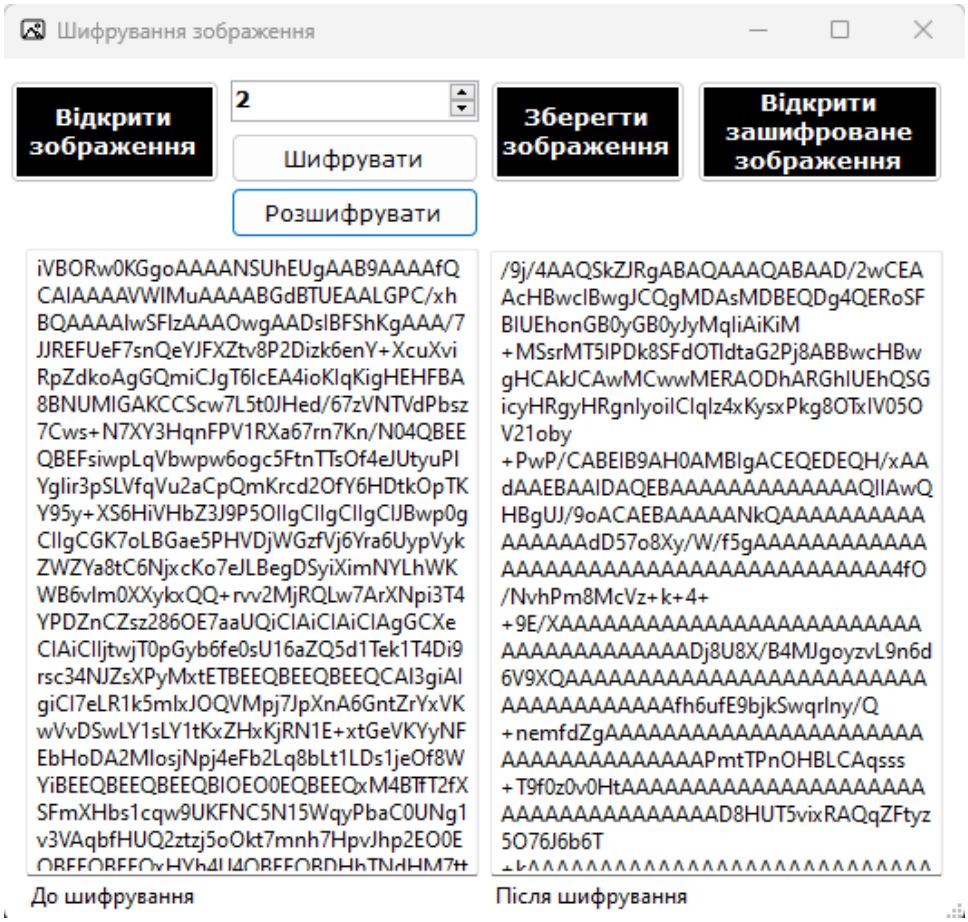


Збереження зашифрованого зображення

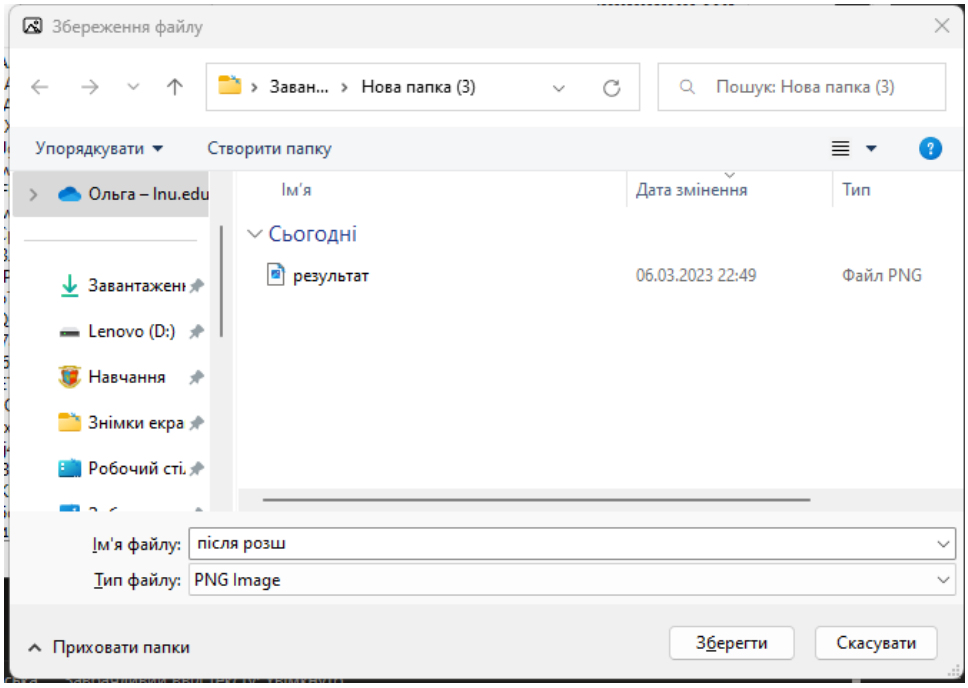


Розшифрування зображення





Збереження зображення після розшифрування

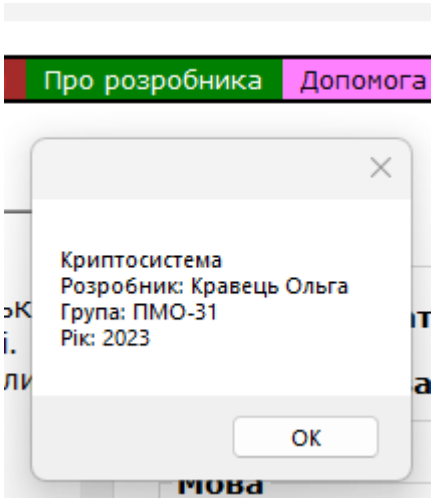


До шифрування

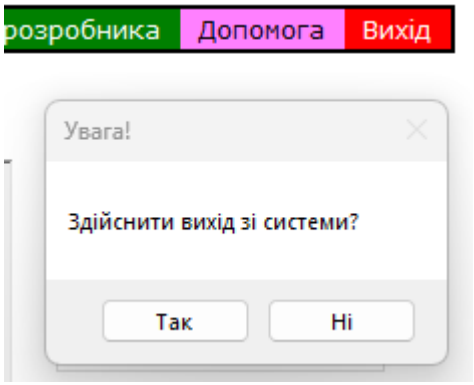
Після розшифрування

Картинка відновилася

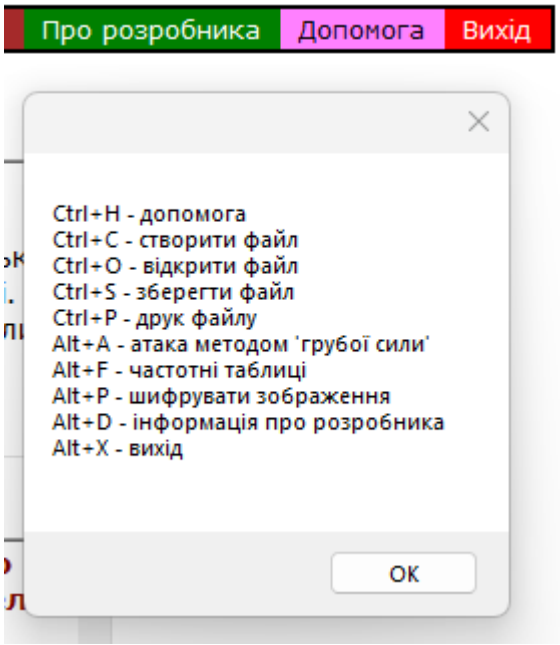
Натискання кнопки Про розробника



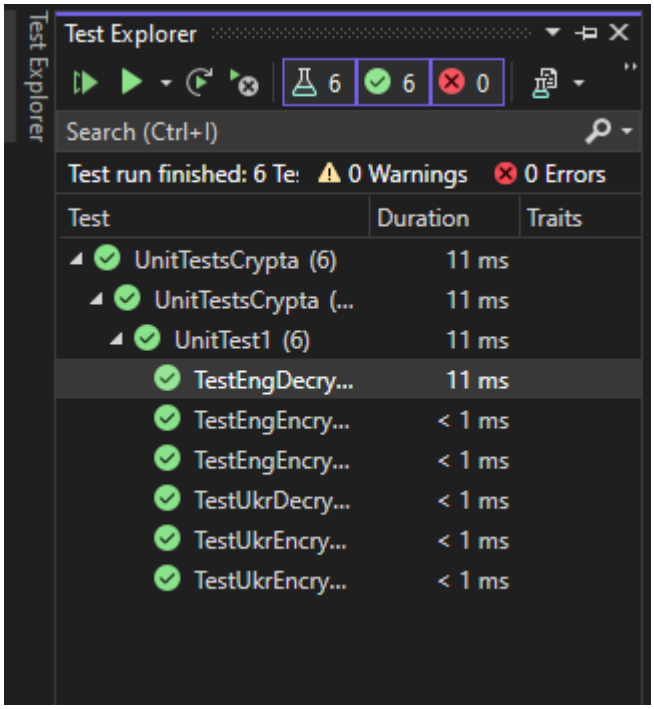
Кнопка Вихід



Кнопка Допомога



Тестування програми юніт тестами



```
namespace UnitTestsCrypta
{
    [TestClass]
    0 references
    public class UnitTest1
    {
        [TestMethod]
        | 0 references
        public void TestEngEncryption()
        {
            string input = "Olia";
            string expected = "Rold";
            int step = 3;
            string output = EngEncryp(input, step);
            Assert.AreEqual(expected, output);
        }

        [TestMethod]
        | 0 references
        public void TestEngEncryptionFail()
        {
            string input = "Olia";
            string expected = "old";
            int step = 3;
            string output = EngEncryp(input, step);
            Assert.AreNotEqual(expected, output);
        }

        [TestMethod]
        | 0 references
        public void TestUkrEncryption()
        {
            string input = "Кравець";
            string expected = "ФьиїмВЕ";
            int step = 10;
            string output = UkrEncryp(input, step);
            Assert.AreEqual(expected, output);
        }

        [TestMethod]
        | 0 references
        public void TestUkrEncryptionFail()
        {
            string input = "Кравець";
            string expected = "їмВЕ";
            int step = 10;
            string output = UkrEncryp(input, step);
            Assert.AreNotEqual(expected, output);
        }

        [TestMethod]
        | 0 references
        public void TestEngDecryption()
        {
            string input = "Rold";
            string expected = "Olia";
            int step = 3;
            string output = EngDecryp(input, step);
            Assert.AreEqual(expected, output);
        }

        [TestMethod]
        | 0 references
        public void TestUkrDecryption()
        {
            string input = "ФьиїмВЕ";
            string expected = "Кравець";
            int step = 10;
            string output = UkrDecryp(input, step);
            Assert.AreEqual(expected, output);
        }
```