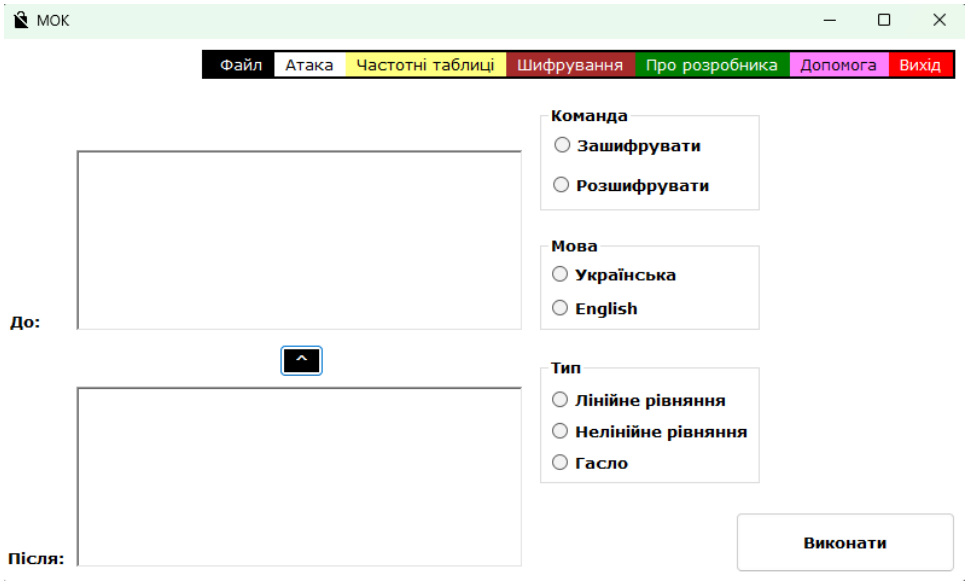
Лабораторна робота 2

Шифр Тритеміуса

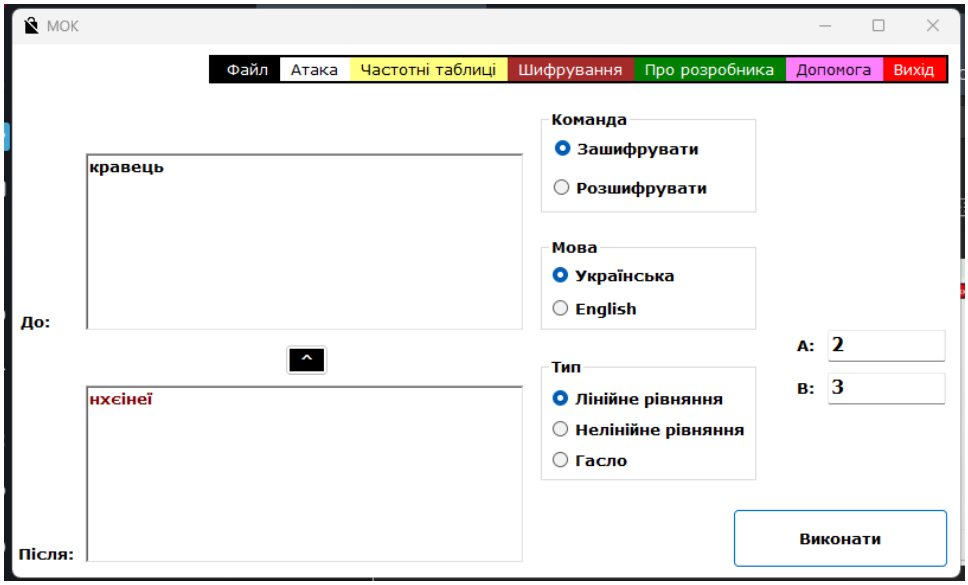
Кравець Ольга

ПМО-31

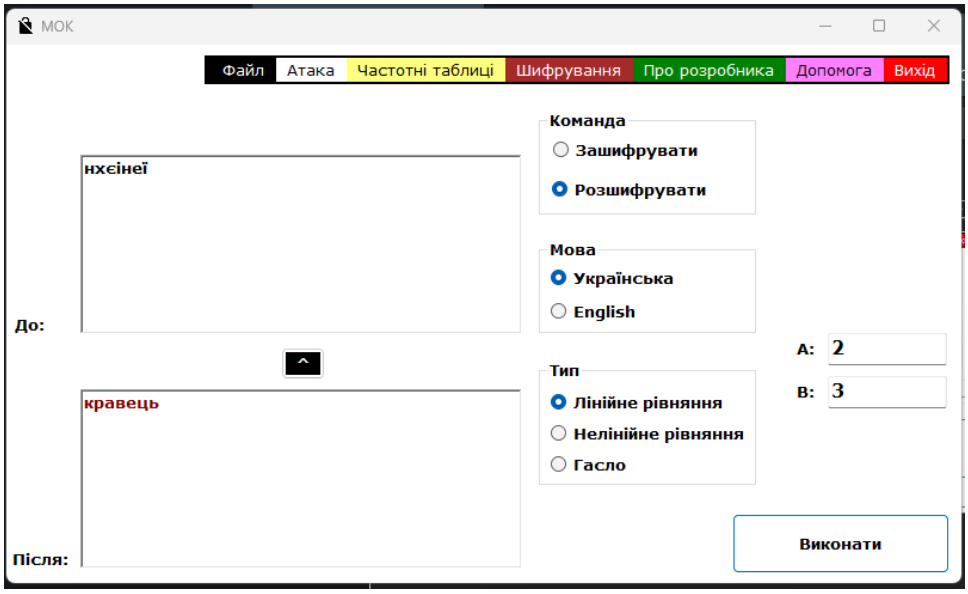
Інтерфейс криптографічної системи:



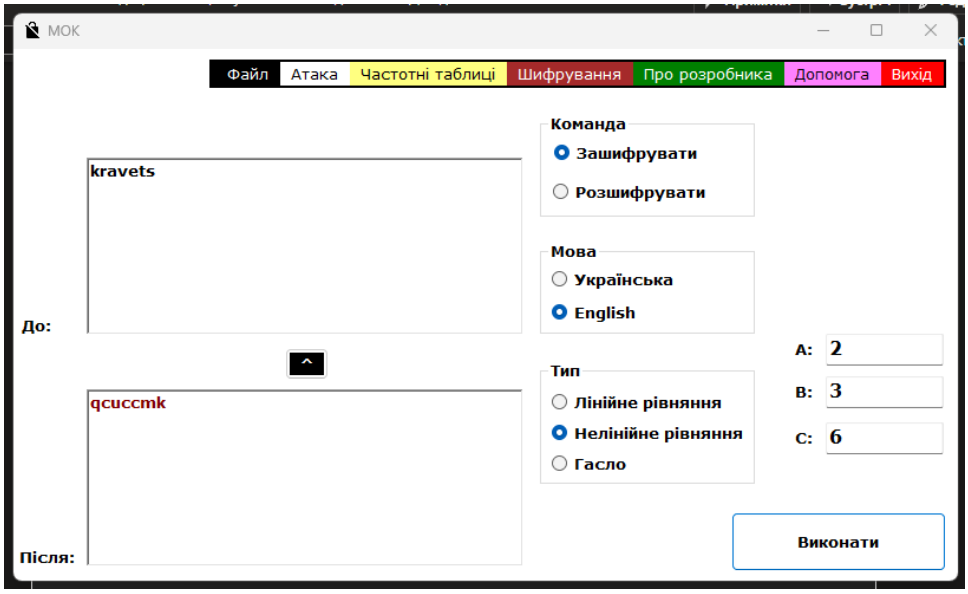
Лінійне шифрування українською мовою:



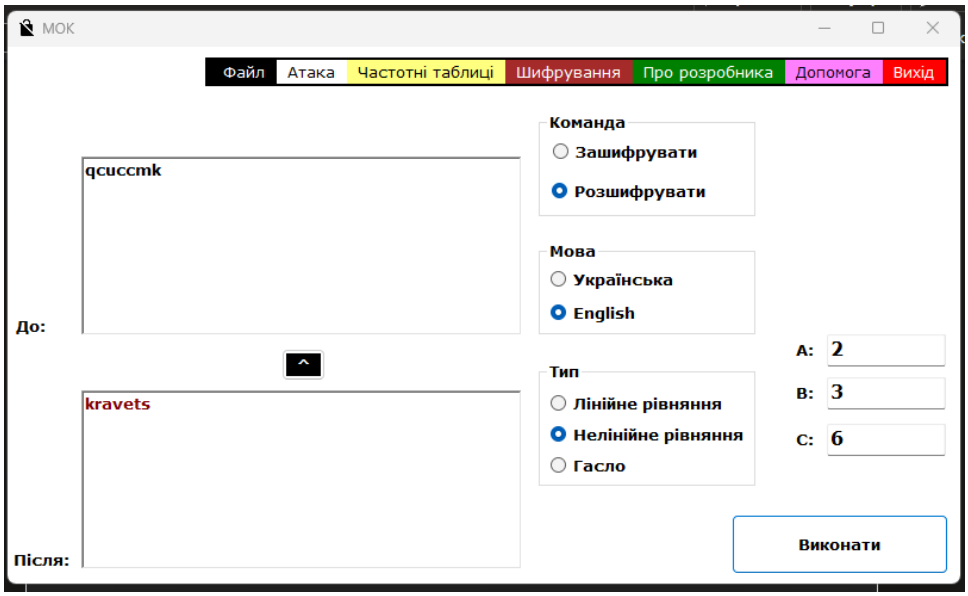
Лінійне дешифрування українською мовою:



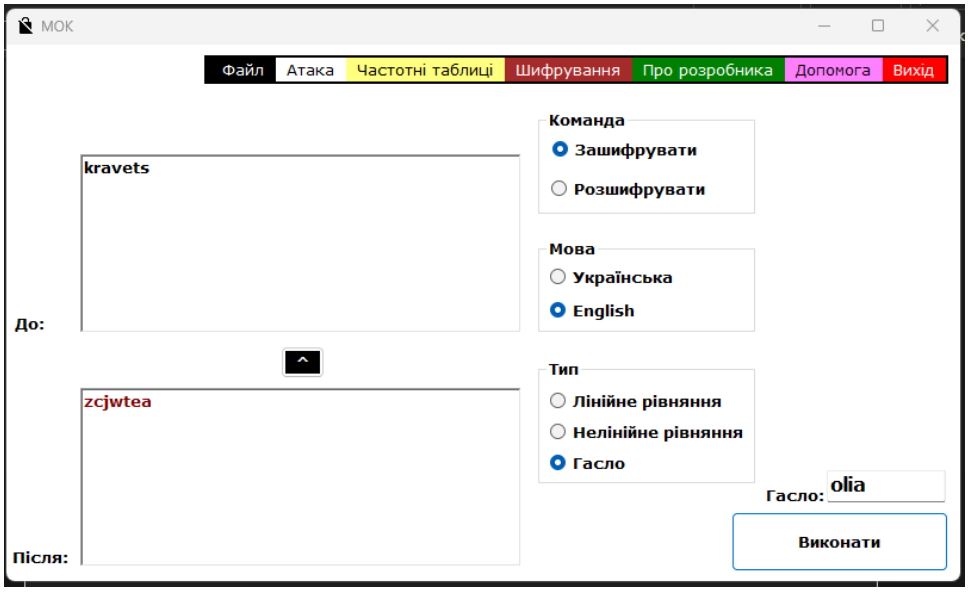
Лінійне шифрування англійською мовою:



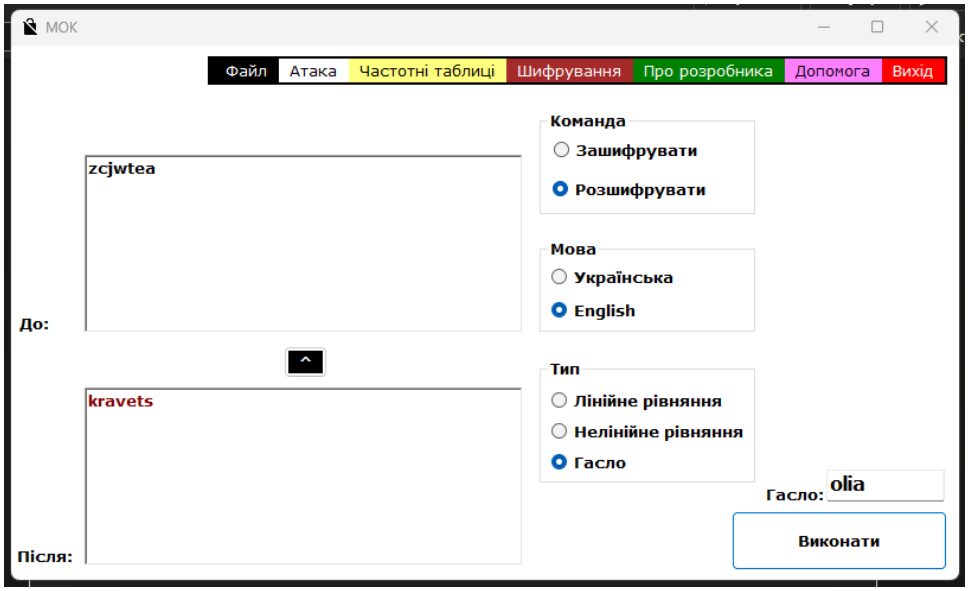
Лінійне дешифрування англійською мовою:



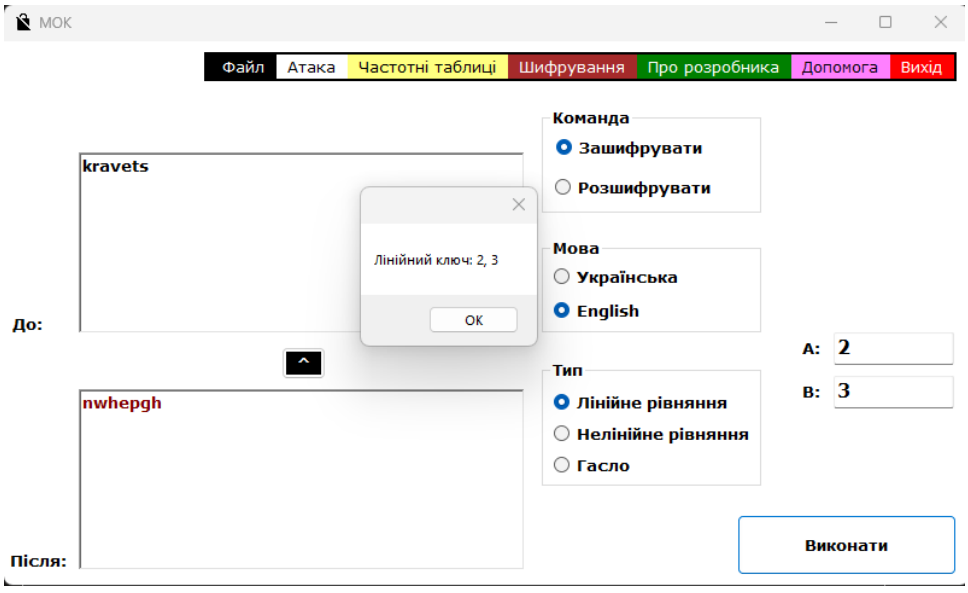
Шифрування гаслом:



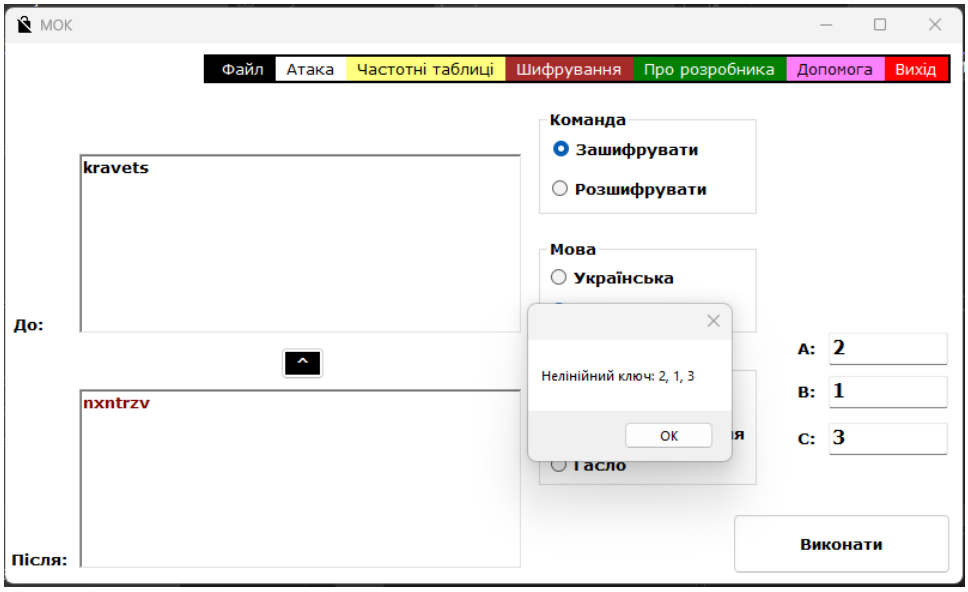
Дешифрування гаслом:



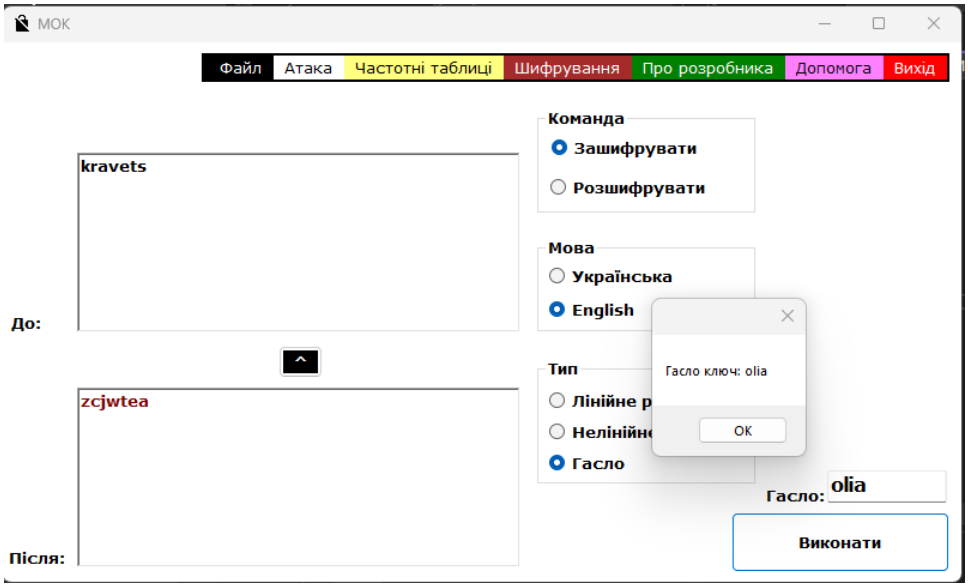
Лінійна атака:



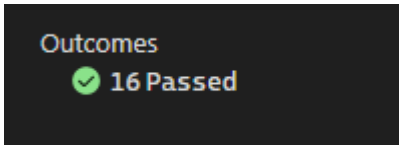
Нелінійна атака:



Атака гаслом:



Тестування програми юніт тестами:



```
[TestClass()]
0 references
public class TrithemiusTests
{
    Trithemius t = new Trithemius();

    [TestMethod()]
    | 0 references
    public void TestEncryptLinearAB()
    {
        string input = "kravets";
        string expected = "nwhepgh";

        int step1 = 2;
        int step2 = 3;
        string output = t.EncryptLinearAB(input, step1, step2);
        Assert.AreEqual(expected, output);
    }

    [TestMethod()]
    | 0 references
    public void TestUKREncryptLinearAB()
    {
        string input = "кравець";
        string expected = "нхєінеї";

        int step1 = 2;
        int step2 = 3;
        string output = t.UKREncryptLinearAB(input, step1, step2);
        Assert.AreEqual(expected, output);
    }
```

```
    [TestMethod()]
    | 0 references
    public void TestDecryptLinearAB()
    {
        string input = "nwhepgh";
        string expected = "kravets";

        int step1 = 2;
        int step2 = 3;
        string output = t.DecryptLinearAB(input, step1, step2);
        Assert.AreEqual(expected, output);
    }

    [TestMethod()]
    | 0 references
    public void TestUKRDecryptLinearAB()
    {
        string input = "нхєінеї";
        string expected = "кравець";

        int step1 = 2;
        int step2 = 3;
        string output = t.UKRDecryptLinearAB(input, step1, step2);
        Assert.AreEqual(expected, output);
    }
```

```
[TestMethod()]
0 references
public void TestEncryptNonLinearABC()
{
    string input = "kravets";
    string expected = "oasaaki";

    int step1 = 2;
    int step2 = 3;
    int step3 = 4;

    string output = t.EncryptNonLinearABC(input, step1, step2, step3);
    Assert.AreEqual(expected, output);
}

[TestMethod()]
0 references
public void TestUKREncryptNonLinearABC()
{
    string input = "кравець";
    string expected = "ощоасцх";

    int step1 = 2;
    int step2 = 3;
    int step3 = 4;

    string output = t.UKREncryptNonLinearABC(input, step1, step2, step3);
    Assert.AreEqual(expected, output);
}
```

```
[TestMethod()]
0 references
public void TestDecryptNonLinearABC()
{
    string input = "oasaaki";
    string expected = "kravets";

    int step1 = 2;
    int step2 = 3;
    int step3 = 4;

    string output = t.DecryptNonLinearABC(input, step1, step2, step3);
    Assert.AreEqual(expected, output);
}

[TestMethod()]
0 references
public void TestUKRDecryptNonLinearABC()
{
    string input = "ощоасцх";
    string expected = "кравець";

    int step1 = 2;
    int step2 = 3;
    int step3 = 4;

    string output = t.UKRDecryptNonLinearABC(input, step1, step2, step3);
    Assert.AreEqual(expected, output);
}
```

```
    }

    [TestMethod()]
    0 references
    public void TestEncryptMotto()
    {
        string input = "kravets";
        string pass = "olia";
        string expected = "zcjwtea";

        string output = t.EncryptMotto(input, pass);
        Assert.AreEqual(expected, output);
    }

    [TestMethod()]
    0 references
    public void TestDecryptMotto()
    {
        string input = "zcjwtea";
        string pass = "olia";
        string expected = "kravets";

        string output = t.DecryptMotto(input, pass);
        Assert.AreEqual(expected, output);
    }
}
```