Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

АЛГОРИТМЫ ДОНАУЧНОЙ КРИПТОГРАФИИ

Отчет по лабораторной работе №1

По дисциплине

«Информационная безопасность»

Студент гр. 431-3

_____ Д.П. Андреев

«___»_____ 2024 г.

Проверил: старший преподаватель кафедры АСУ.

_____Я.В.Яблонский

«___»_____ 2024 г.

Томск 2024

1 Цель работы

Познакомиться и научиться работать с алгоритмами донаучной криптографии.

2 Задание на лабораторную работу

Вариант 2. Шифр сдвига. Напишите программу, позволяющую зашифровать и расшифровать сообщения с использованием с помощью шифра сдвига. Входные и выходные данные запишите в файл типа .txt.

3 Описание алгоритма шифрования

Шифр Цезаря — это простой тип подстановочного шифра, где каждая буква обычного текста заменяется буквой с фиксированным числом позиций вниз по алфавиту.

4 Листинг программы

```
using System;
using System.IO;

namespace lab_1
{
   class Program
   {
      static void Encryption(char[] textIn,int key)//Зашифровка
      {
         char[] ABC = new char[] { 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У',
'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я' };
         char[] EncryptABC = new char[] { 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С',
'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я' };
         //Преобразование алфавита
         if (key < 0)
         {
            for (int i = Math.Abs(key); i < ABC.Length; i++)
            {
               EncryptABC[i] = ABC[i + key];
            }
            for (int i = 0; i < Math.Abs(key); i++)
            {
               EncryptABC[i] = ABC[ABC.Length + key + i];
            }
         }
         else
         {
            for (int i = 0; i < ABC.Length-key; i++)
            {
               EncryptABC[i] = ABC[i + key];
            }
            for (int i = 0; i < key; i++)
            {
               EncryptABC[ABC.Length - key+i] = ABC[i];
            }
         }

         //Расшифровка
```

```
            for (int i = 0; i < textIn.Length-1000; i++)
            {
                for (int j = 0; j < ABC.Length ; j++)
                {
                    if (textIn[i] == ABC[j])
                    {
                        textIn[i] = EncryptABC[j];
                        break;
                    }
                    if (textIn[i].ToString() == ABC[j].ToString().ToLower())
                    {
                        textIn[i] = Convert.ToChar(EncryptABC[j].ToString().ToLower());
                        break;
                    }
                }
            }
            Console.WriteLine("Готово!");
            for (int i = 0; i < textIn.Length - 1000; i++)
            {
                Console.Write(textIn[i]);
            }
            Console.WriteLine();
            //Запись ответа
            FileStream fileOut = new FileStream("OUT.txt", FileMode.Open);
            StreamWriter writer = new StreamWriter(fileOut);
            for (int i = 0; i < textIn.Length - 1000; i++)
            {
                writer.Write(textIn[i]);
            }
            writer.Close();
        }

        static void Decryption(char[] textIn, int key)//Расшифровка
        {
            char[] ABC = new char[] { 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У',
'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я' };
            char[] EncryptABC = new char[] { 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С',
'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я' };
            //Преобразование алфавита
            if (key < 0)
            {
                for (int i = Math.Abs(key); i < ABC.Length; i++)
                {
                    EncryptABC[i] = ABC[i + key];
                }
                for (int i = 0; i < Math.Abs(key); i++)
                {
                    EncryptABC[i] = ABC[ABC.Length + key + i];
                }
            }
            else
            {
                for (int i = 0; i < ABC.Length - key; i++)
                {
                    EncryptABC[i] = ABC[i + key];
                }
                for (int i = 0; i < key; i++)
                {
                    EncryptABC[ABC.Length - key + i] = ABC[i];
                }
            }
```

```
    //Расшифровка
    for (int i = 0; i < textIn.Length - 1000; i++)
    {
        for (int j = 0; j < ABC.Length; j++)
        {
            if (textIn[i] == EncryptABC[j])
            {
                textIn[i] = ABC[j];
                break;
            }
            if (textIn[i].ToString() == EncryptABC[j].ToString().ToLower())
            {
                textIn[i] = Convert.ToChar(ABC[j].ToString().ToLower());
                break;
            }
        }
    }
    Console.WriteLine("Готово!");
    for (int i = 0; i < textIn.Length - 1000; i++)
    {
        Console.Write(textIn[i]);
    }
    Console.WriteLine();

    //Запись ответа
    FileStream fileOut = new FileStream("OUT.txt", FileMode.Open);
    StreamWriter writer = new StreamWriter(fileOut);
    for (int i = 0; i < textIn.Length - 1000; i++)
    {
        writer.Write(textIn[i]);
    }
    writer.Close();
}

static void Main(string[] args)
{
    int namber=0;
    while (namber != 3)
    {
        Console.WriteLine("Выберите действие:");
        Console.WriteLine("1) Зашифровать");
        Console.WriteLine("2) Расшифровать");
        Console.WriteLine("3) Выход");
        namber = Convert.ToInt32(Console.ReadLine());
        switch (namber)
        {
            case 1:
                Console.WriteLine("Зашифровка");
                FileStream fileIn1 = new FileStream("IN.txt", FileMode.Open);
                StreamReader reader1 = new StreamReader(fileIn1);
                char[] textIn = new char[reader1.Peek()];
                Console.Write("Исходный текст: ");
                for (int i=0; reader1.Peek() >= 0; i++)
                {
                    textIn[i] = Convert.ToChar(reader1.Read());
                    Console.Write(textIn[i]);
                }
                Console.WriteLine();
                reader1.Close();

                int key;
                Console.Write("Введите ключ шифрования: ");
```

```
                key = Convert.ToInt32(Console.ReadLine());
                Encryption(textIn, key);
                break;

            case 2:
                Console.WriteLine("Расшифровка");
                FileStream fileIn2 = new FileStream("IN.txt", FileMode.Open);
                StreamReader reader2 = new StreamReader(fileIn2);
                char[] textIn2 = new char[reader2.Peek()];
                Console.Write("Исходный текст: ");
                for (int i = 0; reader2.Peek() >= 0; i++)
                {
                    textIn2[i] = Convert.ToChar(reader2.Read());
                    Console.Write(textIn2[i]);
                }
                Console.WriteLine();
                reader2.Close();

                Console.Write("Введите ключ шифрования: ");
                key = Convert.ToInt32(Console.ReadLine());
                Decryption(textIn2, key);
                break;
        }
    }
  }
}
```

5 Примеры работы программы

Первый пример. В первом примере мы зашифруем текст из входного текстового файла IN.txt (рисунок 5.1).

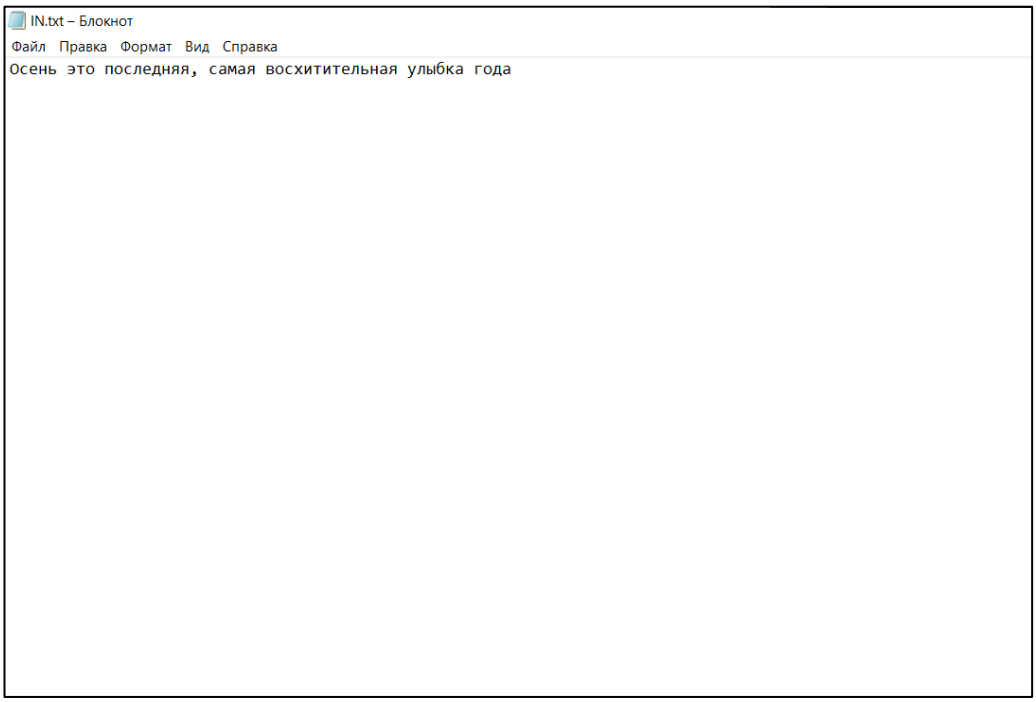

Рисунок 5.1 – Входные данные файла IN.txt

При запуске программы нас встречает меню выбора действий (рисунок 5.2).

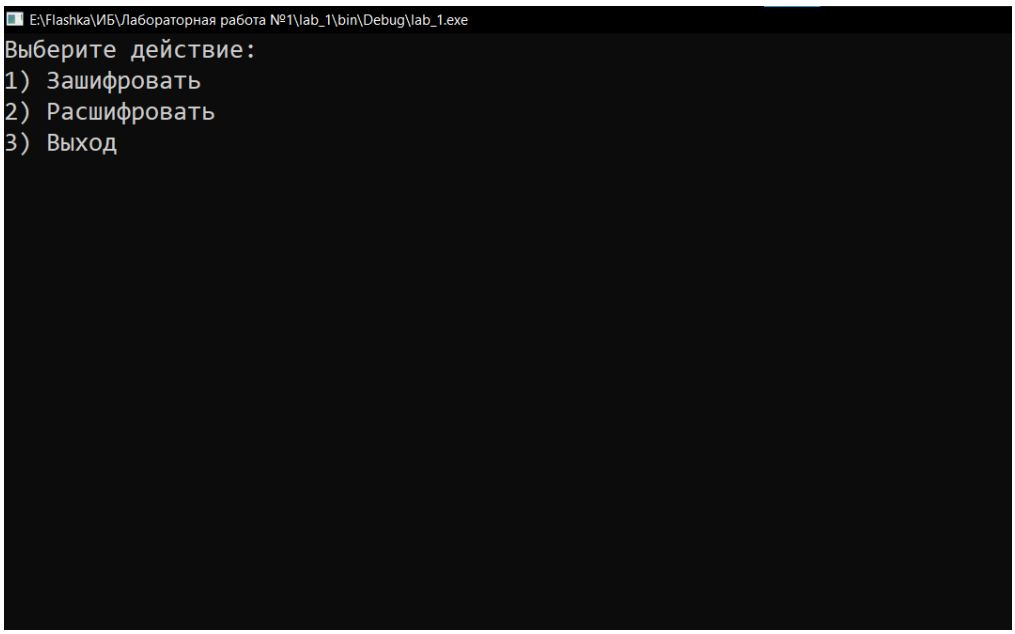

Рисунок 5.2 – Меню выбора действия

После выбора первого варианта программа выводит на экран исходный текст полученный из файла и просит нас вести ключ шифрования (рисунок 5.3).

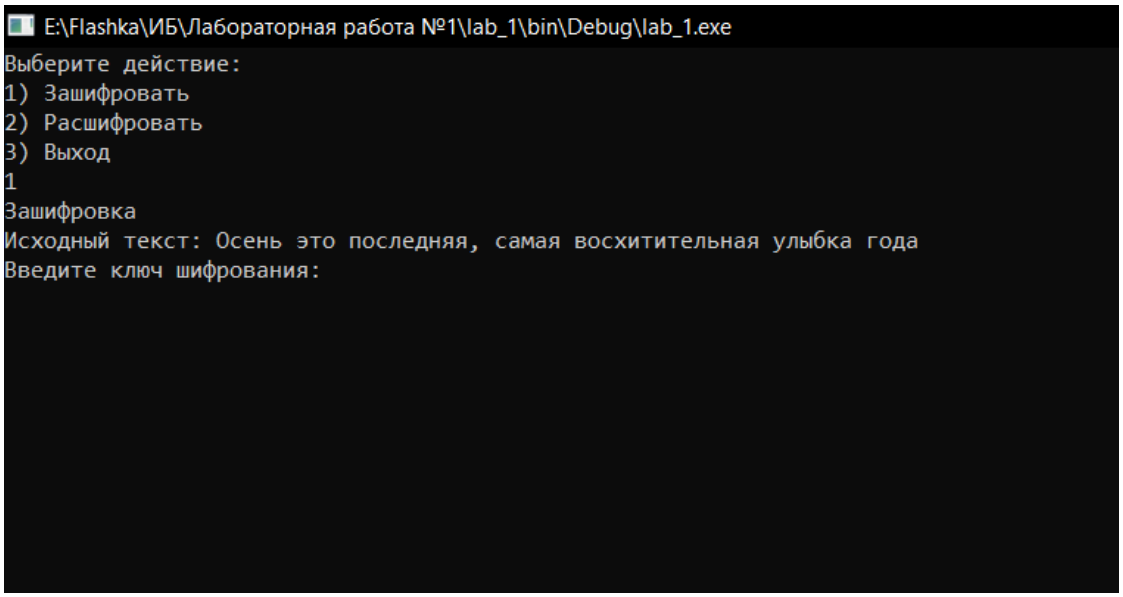

Рисунок 5.3 – Вывод исходного текста и запрос ключа шифрования

Вводим ключ шифрования после чего программа выводит зашифрованный текст и записывает его в файл OUT.txt (рисунок 5.4-5.5).

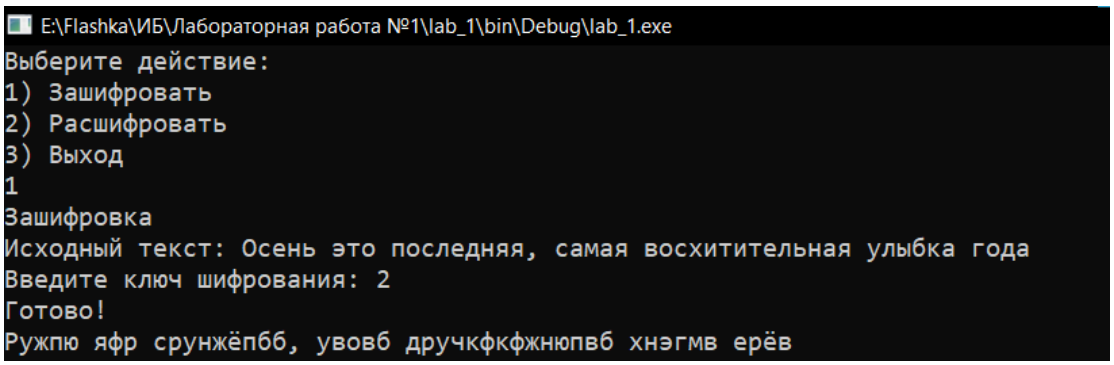

Рисунок 5.4 – Вывод зашифрованного текста

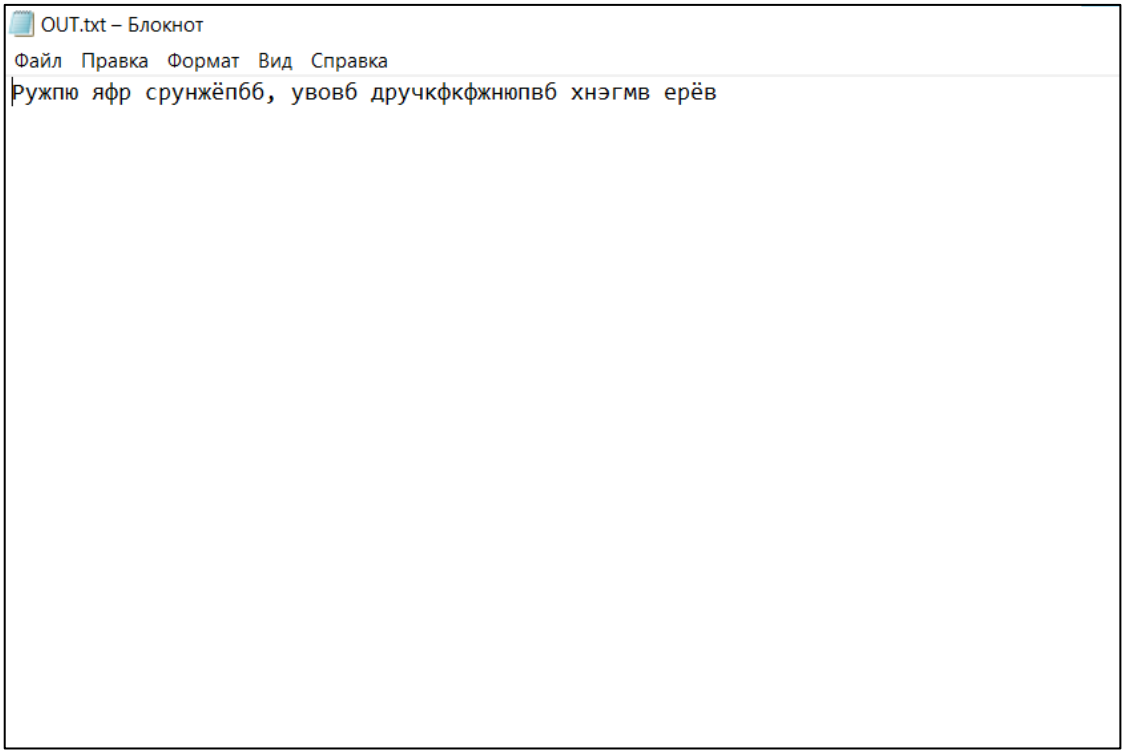

Рисунок 5.5 – Файл OUT.txt в котором программа записала зашифрованный текст

Второй пример. Во втором примере мы расшифруем сообщение из входного файла IN.txt (рисунок 5.6).

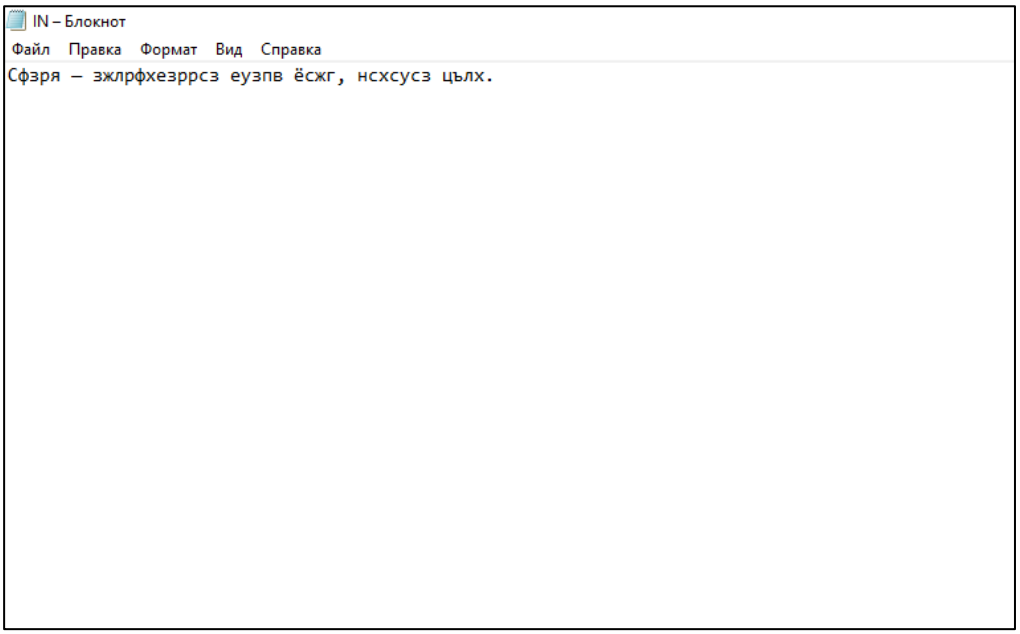

Рисунок 5.6 – Файл IN.txt с зашифрованным сообщением

При запуске программы нас встречает меню выбора действий (рисунок 5.7).

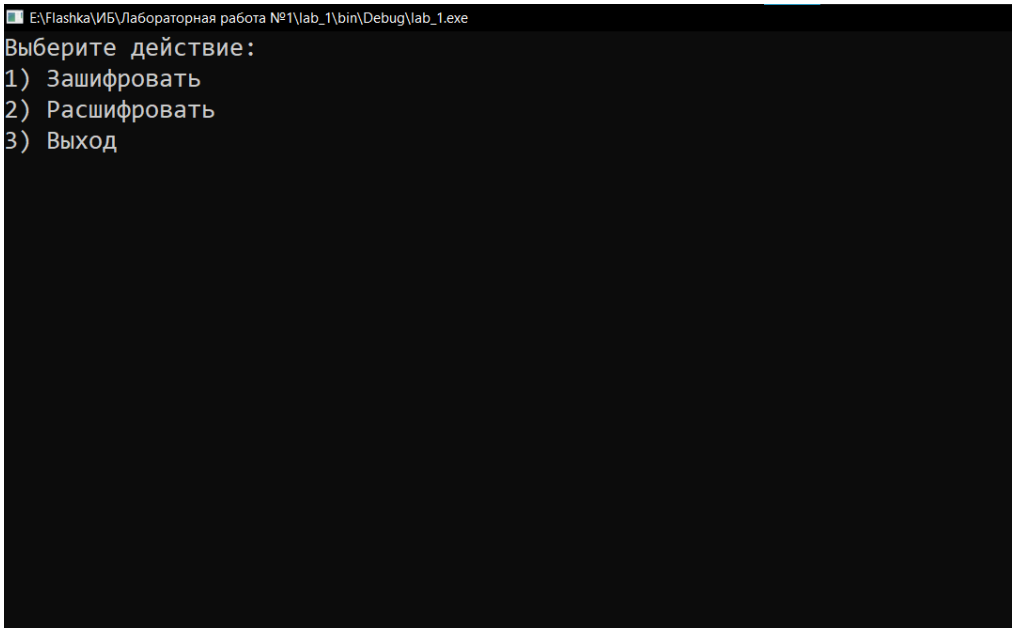

Рисунок 5.7 – Меню выбора действия

После выбора второго варианта программа выводит на экран исходный текст, полученный из файла и просит нас вести ключ шифрования (рисунок 5.8).

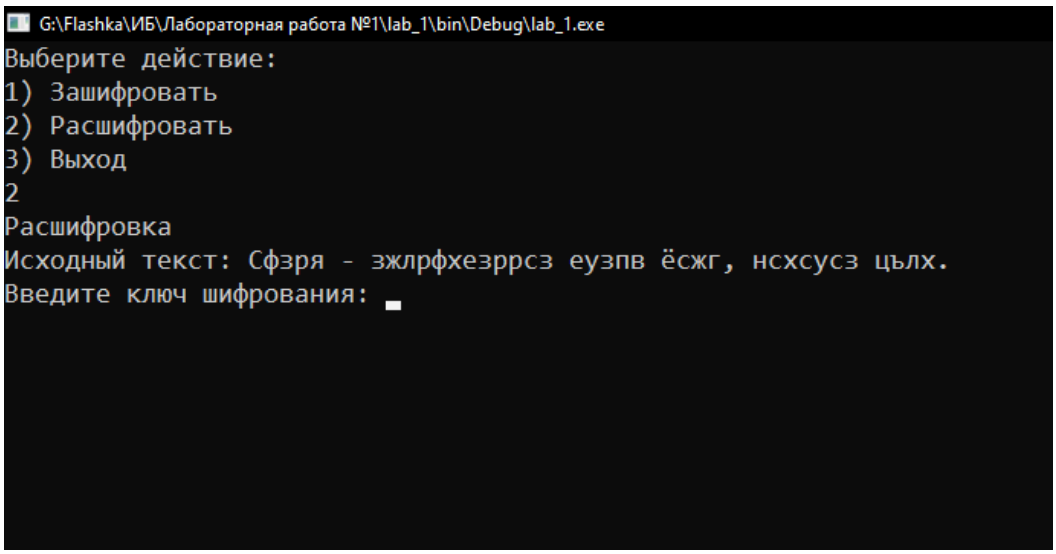

Рисунок 5.8 – Вывод исходного текста и запрос ключа шифрования

Вводим ключ шифрования после чего программа выводит расшифрованный текст и записывает его в файл OUT.txt (рисунок 5.9-5.10).

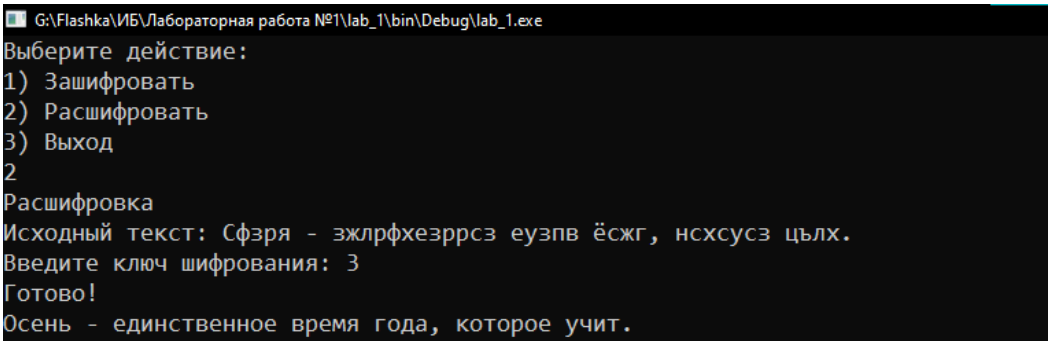

Рисунок 5.9 – Вывод расшифрованного текста

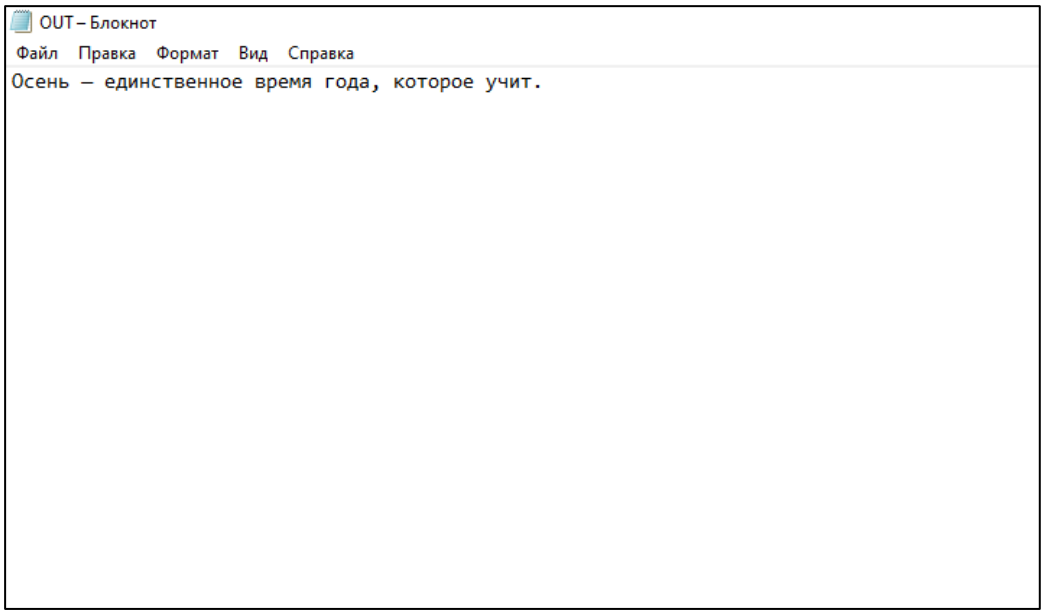

Рисунок 5.10 – Файл OUT.txt в котором программа записала расшифрованный текст

6 Вывод

В ходе выполнения лабораторной работы я изучил метод шифрования Цезаря и его реализация на языке C#. У Шифра Цезаря, как у алгоритма шифрования, я могу выделить две основные особенности.