ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

Московский технический университет связи и информатики

(МТУСИ)

Кафедра «Информационная безопасность»

По дисциплине: «Криптографические методы и средства обеспечения информационной безопасности инфокоммуникаций»

Лабораторная работа на тему:

«Простые симметричные шифры».

Выполнили студенты

группы БПЗ-1701:

Болябкин М.В.

Багажков Д.И

Исобоев Ш.И.

Проверила:

Магомедова Д.И.

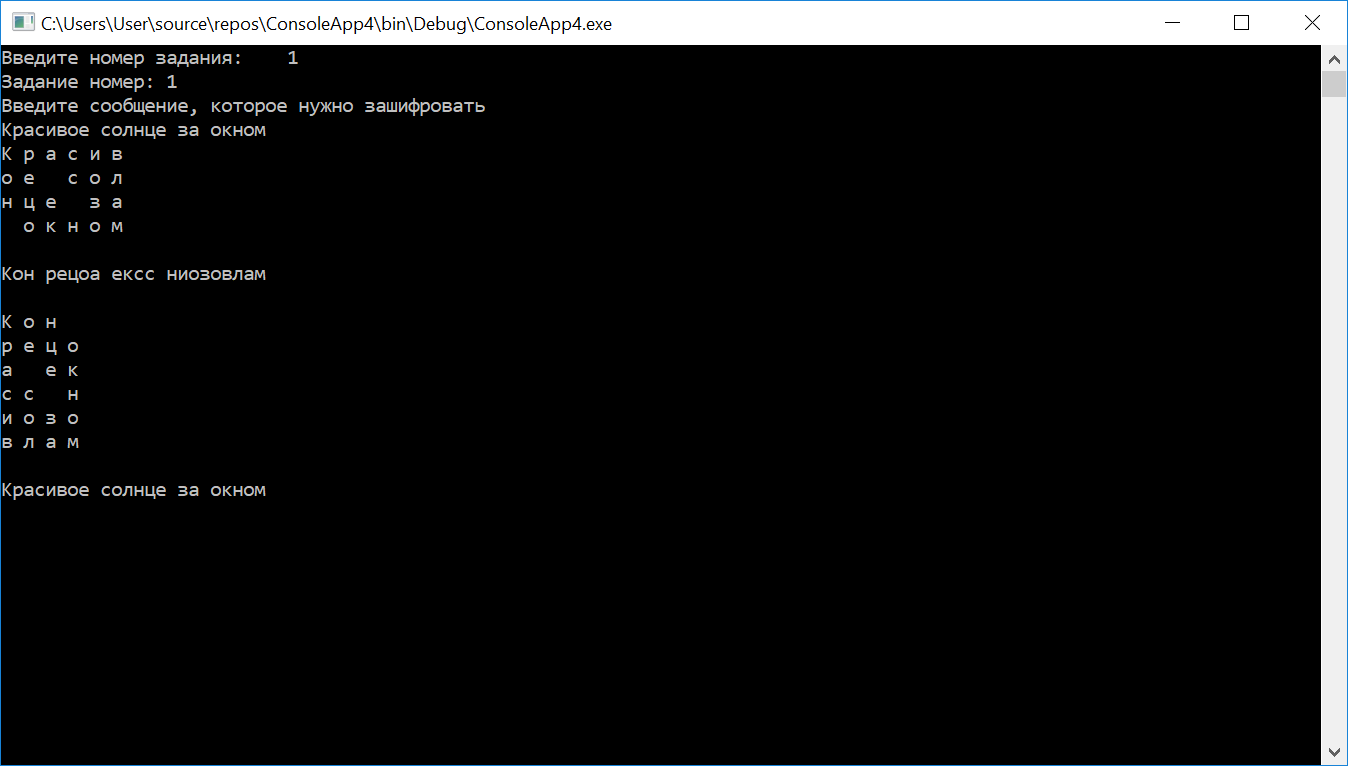
Москва 2020г.

Цель работы: ознакомление с простыми симметричными криптографическими шифрами.

Ход работы:

Задание 1.

Придумайте свой перестановочный шифр, реализуйте его на любом языке программирования, зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его. Опишите достоинства и недостатки шифра.

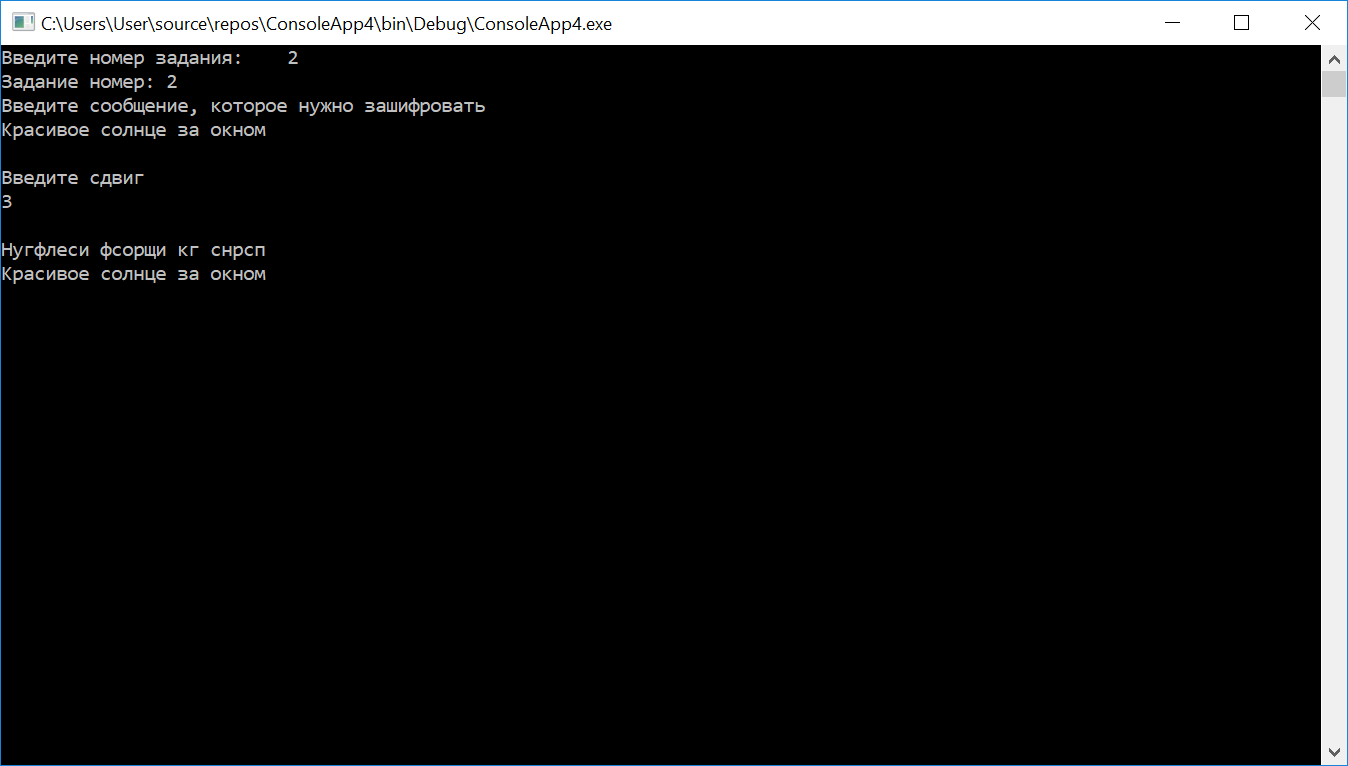


Достоинства: так как при шифровании меняется не открытый текст, а порядок символов, то при помощи таблиц частоты использования букв в языках, данный шифр не взломать.

Недостатки: нужно чтобы все ячейки таблицы были заполнены. При необходимости строка текста дополняется нужным количеством пробелов.

Задание 2.

Придумайте свой простой подстановочный шифр (моноалфавитный шифр), реализуйте его на любом языке программирования зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его.

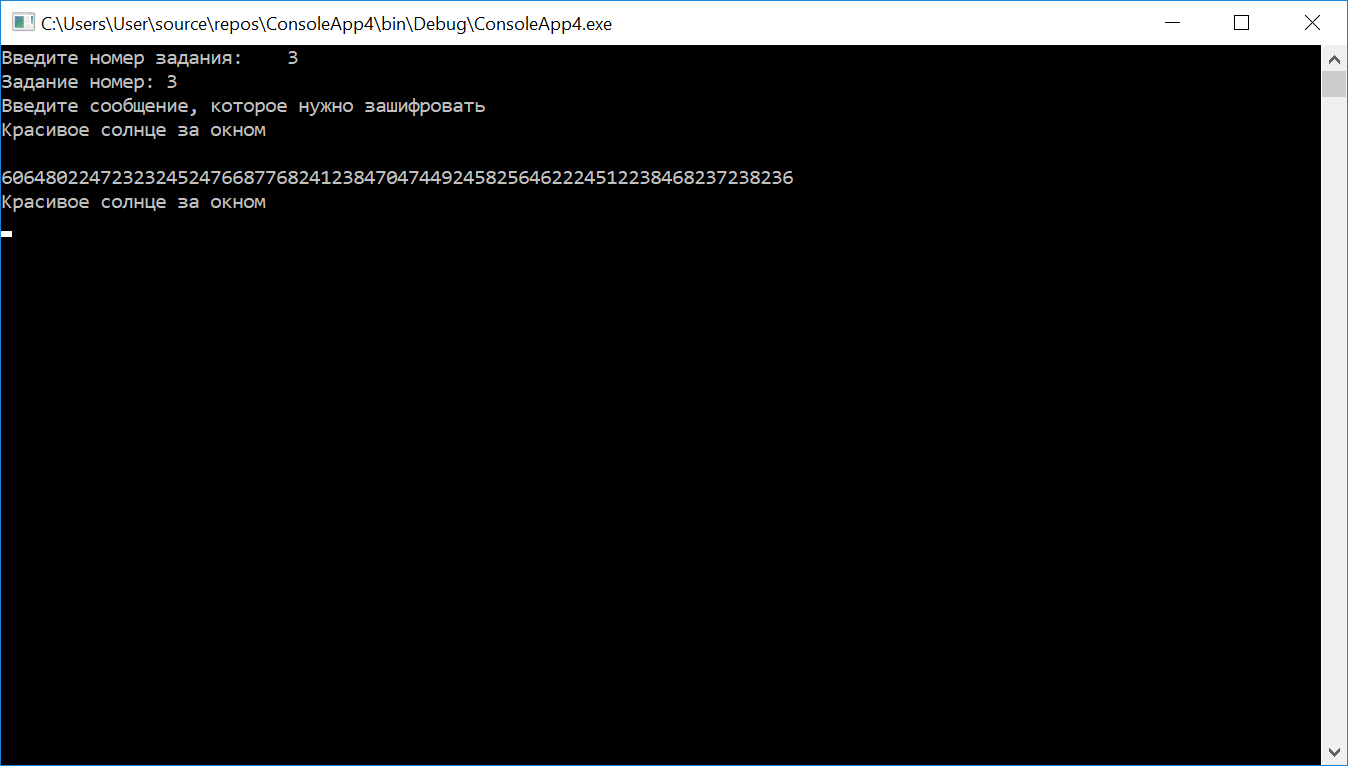


Достоинства: метод использует случайный ключ для каждой отдельной буквы для шифрования, который делает моноалфавитный шифр защищенным от атаки грубой силы.

Недостатки: сложно запомнить порядок букв в ключе, и поэтому требуется много времени и усилий для шифрования или расшифровки текста вручную. Моноалфавитная замена уязвима для частотного анализа.

Задание 3.

Придумайте свой однозвучный подстановочный шифр, реализуйте его на любом языке программирования, зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его. Опишите достоинства и недостатки шифра.

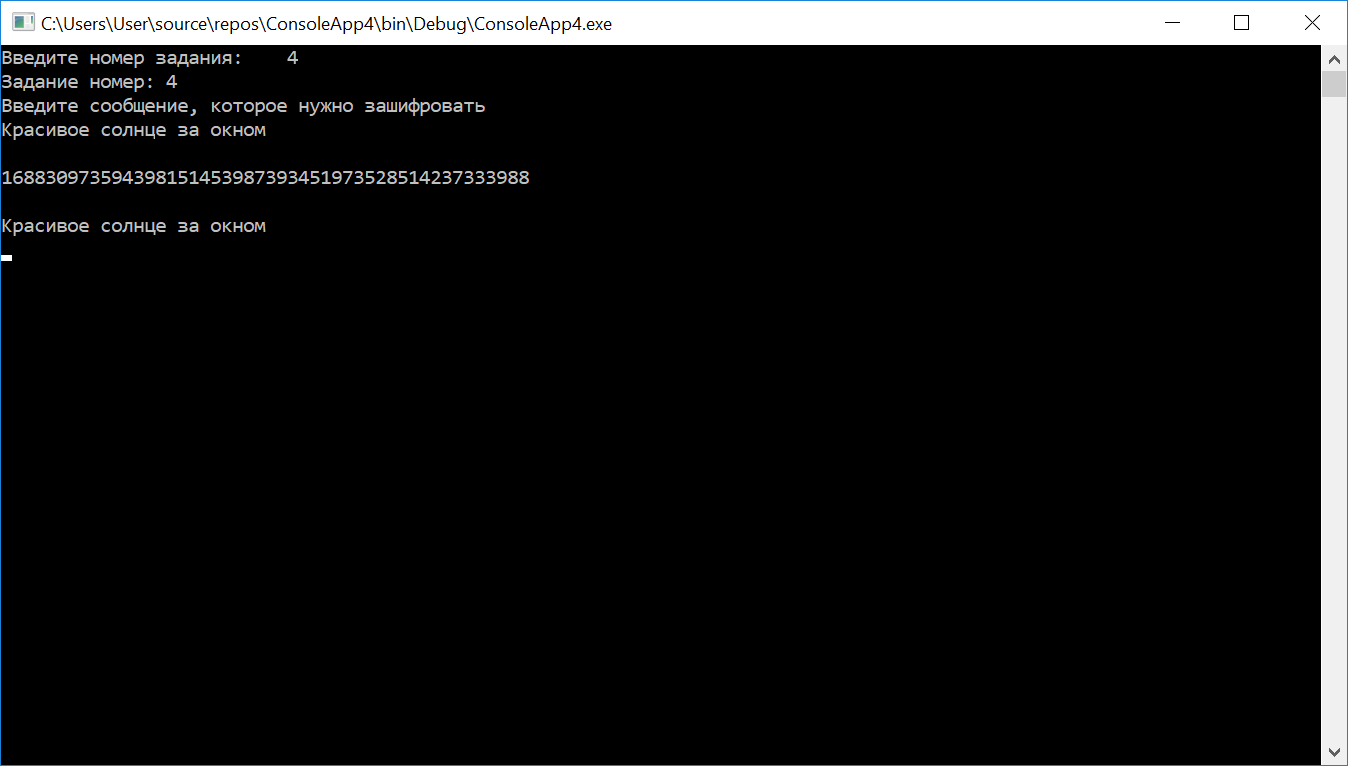


Достоинства: так как при шифровании для каждого символа используется по три различных ключа, то при помощи таблиц частоты использования букв в языках, вероятность взлома данного шифра снижается.

Недостатки: вероятность взлома данного шифра возрастает при увеличении длины текста.

Задание 4.

Придумайте полиграммный подстановочный шифр, реализуйте его на любом языке программирования, зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его. Опишите достоинства и недостатки шифра.

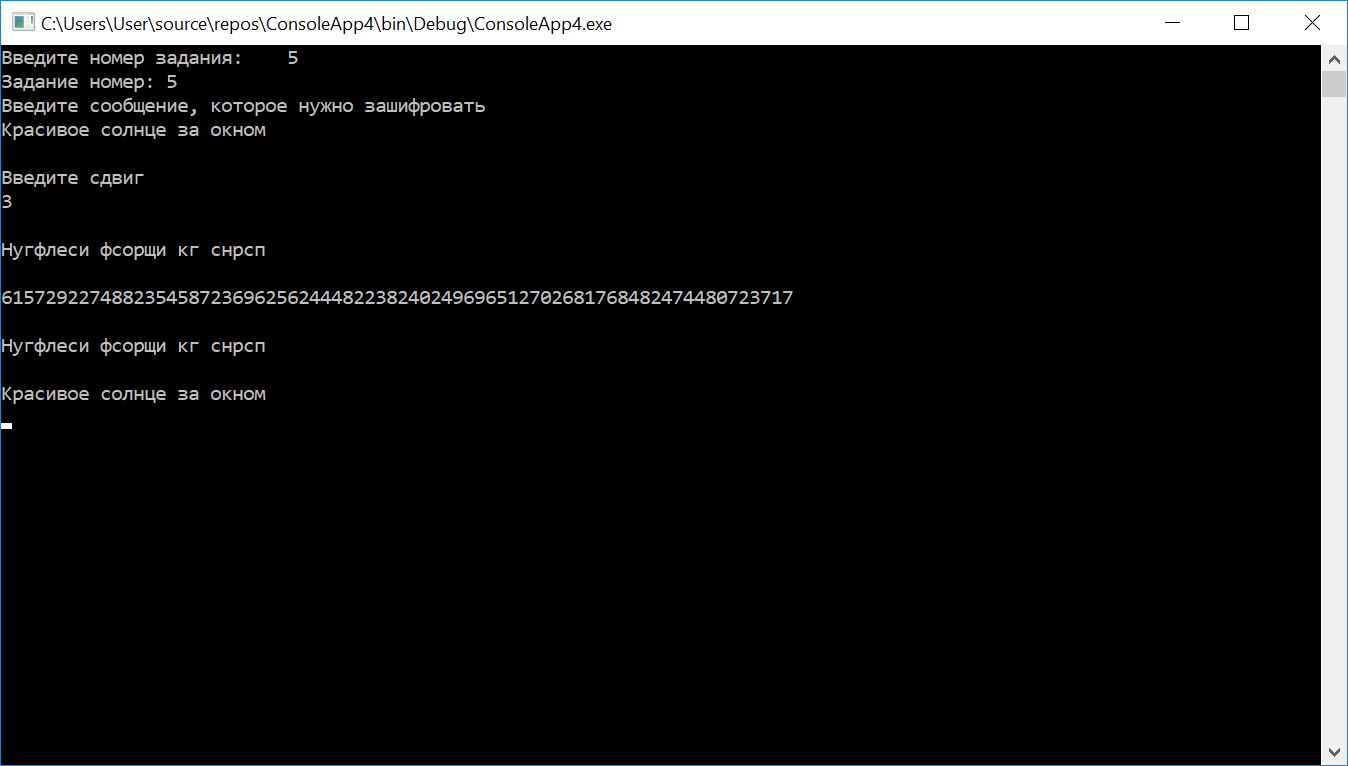


Достоинства: так как символы шифруются парами, из-за этого возникает множество различных ключей. Первое преимущество такого способа заключается в том, что распределение частот групп букв значительно более равномерное, чем отдельных символов. Во-вторых, для продуктивного частотного анализа требуется больший размер зашифрованного текста, так число различных групп букв значительно больше, чем просто алфавит.

Недостатки: имеет только открытый ключ.

Задание 5.

Придумайте полиалфавитный подстановочный шифр, реализуйте его на любом языке программирования, зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его. Опишите достоинства и недостатки шифра.

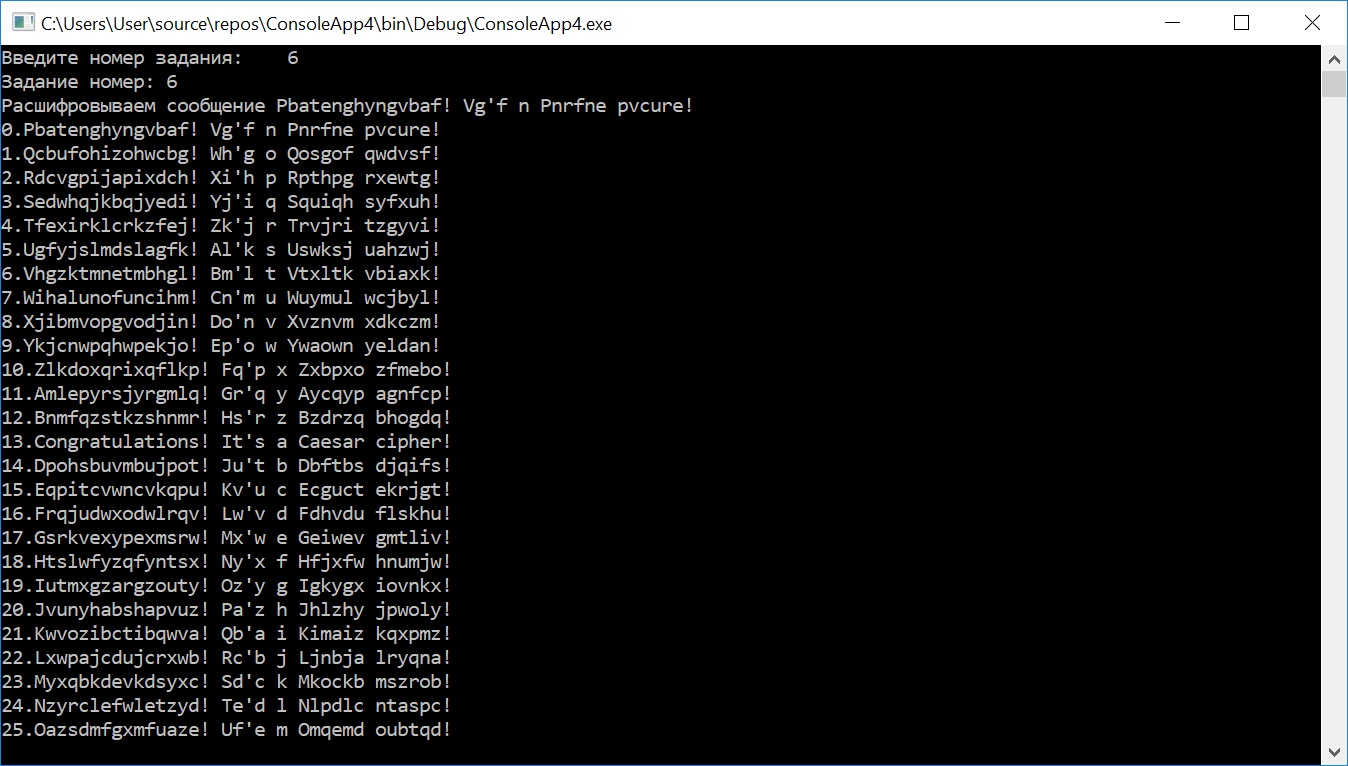


Достоинства: так как используется два различных вида шифрования, то в связи с этим вероятность взлома полученного шифра стремится к нулю.

Недостатки: имеет только открытый ключ.

Задание 6.

Известно, что сообщение " Pbatenghyngvbaf! Vg'f n Pnrfne pvcure!" было зашифровано шифром Цезаря, но ключ k неизвестен. Напишите вспомогательную программу, которая поможет вам в расшифровке этого сообщения.



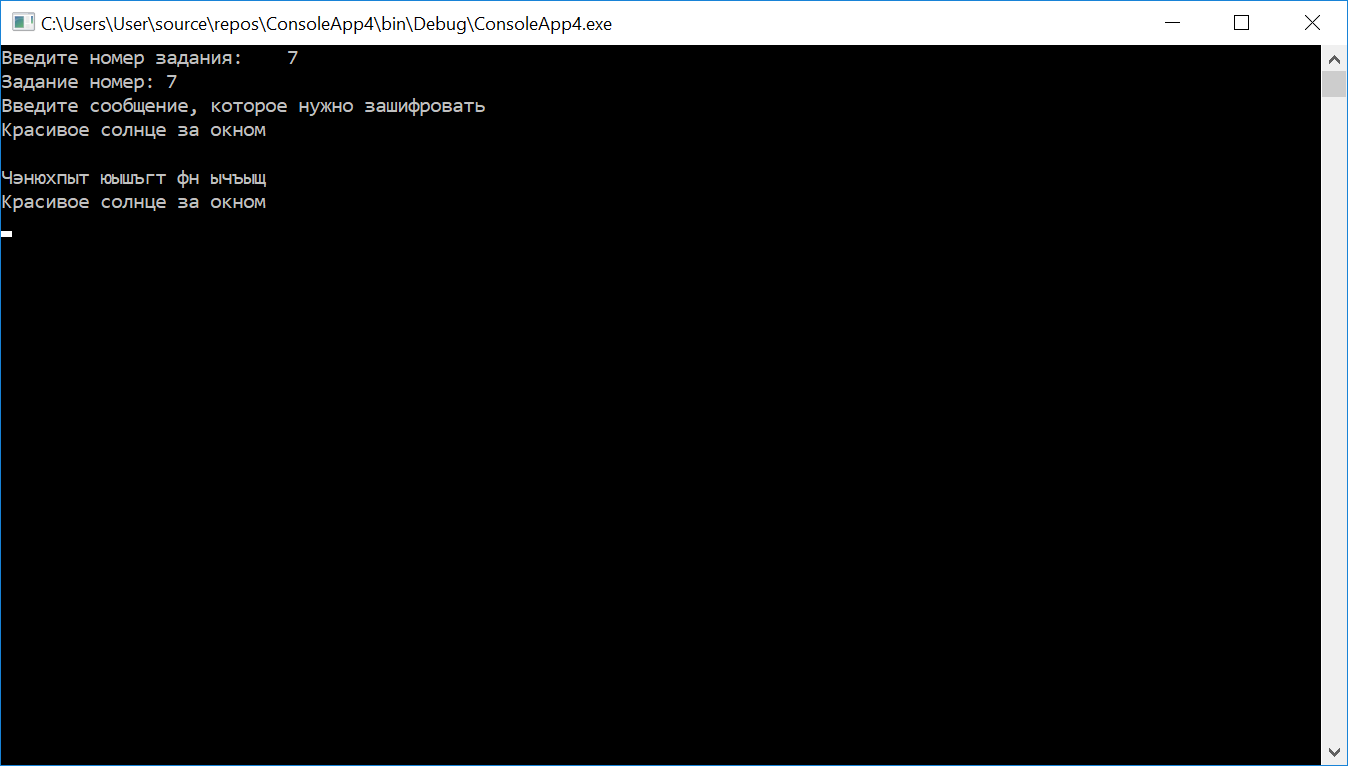
Достоинства: простота шифрования и расшифровывания.

Недостатки: подстановки, выполняемые в соответствии с системой Цезаря, не маскируют частот появления различных букв исходного открытого текста;

число возможных ключей К мало; шифр Цезаря легко вскрывается на основе анализа частот появления букв в шифр тексте.

Задание 7.

Реализуйте собственный аналог функции ROT13 его на любом языке программирования, зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его. Опишите достоинства и недостатки шифра.



Достоинства: прост в реализации.

Недостатки: при помощи таблиц частоты использования букв в языках, вероятность взлома данного шифра значительно возрастает.

Код программы:

using System;

using System.Collections.Generic;

using System.Text;

namespace Project

{

class Project

{

static void Main(string[] args)

{

Console.Write("Введите номер задания: ");

int n = Convert.ToInt32(Console.ReadLine());

Console.Write("Задание номер: " + n + "\n");

if (n == 1)

{

zadanie1();

}

else if (n == 2)

{

zadanie2();

}

else if (n == 3)

{

zadanie3();

}

else if (n == 4)

{

zadanie4();

}

else if (n == 5)

{

zadanie5();

}

else if (n == 6)

{

zadanie6();

}

else if (n == 7)

{

zadanie7();

}

else

{

return;

}

}

// Задание 1

public static void zadanie1()

{

//Шифрование

int rowsIndex1 = 0;

int q1 = 0;

int columnsMax = 6;

Console.WriteLine("Введите сообщение, которое нужно зашифровать");

string IshodText = Console.ReadLine();

string[,] Tabl = new string[IshodText.Length / columnsMax + 1, columnsMax];

while (IshodText.Length % columnsMax != 0)

{

IshodText += " ";

}

for (int i = 0; i < IshodText.Length; i++)

{

Tabl[rowsIndex1, i - q1] = IshodText.Substring(i, 1);

Console.Write(Tabl[rowsIndex1, i - q1] + " ");

if ((i + 1) % columnsMax == 0)

{

rowsIndex1++;

q1 += columnsMax;

Console.Write("\n");

}

}

string Result = "";

for (int i = 0; i < columnsMax; i++)

{

for (int k = 0; k < IshodText.Length / columnsMax; k++)

{

Result += Tabl[k, i];

}

}

Console.WriteLine("\n" + Result + "\n");

//Дешифрование

int rowsIndex2 = 0;

int q2 = 0;

int columnsMax2 = rowsIndex1;

string[,] Tabl2 = new string[Result.Length / columnsMax2 + 1, columnsMax2];

while (Result.Length % columnsMax2 != 0)

{

Result += " ";

}

for (int i = 0; i < Result.Length; i++)

{

Tabl2[rowsIndex2, i - q2] = Result.Substring(i, 1);

Console.Write(Tabl2[rowsIndex2, i - q2] + " ");

if ((i + 1) % columnsMax2 == 0)

{

rowsIndex2++;

q2 += columnsMax2;

Console.Write("\n");

}

}

string Deshifr = "";

for (int i = 0; i < columnsMax2; i++)

for (int k = 0; k < Result.Length / columnsMax2; k++)

{

Deshifr += Tabl2[k, i];

}

Console.WriteLine("\n" + Deshifr);

Console.ReadKey();

}

//Задание 2

public static string ROT(int N, string text)

{

String output = "";

int cod = 0;

for (int i = 0; i < text.Length; i++)

{

if (Char.IsLetter(text[i]))

{

byte[] bite = Encoding.GetEncoding(1251).GetBytes(text.Substring(i, 1));

cod = Convert.ToInt32(bite[0]);

//Русский

if (cod > 191 && cod < 224)

{

if (cod + N > 223)

{

output += (char)(text[i] + N - 32);

}

else

{

if (cod + N < 192)

{

output += (char)(text[i] + N + 32);

}

else

{

output += (char)(text[i] + N);

}

}

}

if (cod > 223 && cod < 256)

{

if (cod + N > 255)

{

output += (char)(text[i] + N - 32);

}

else

{

if (cod + N < 224)

{

output += (char)(text[i] + N + 32);

}

else

{

output += (char)(text[i] + N);

}

}

}

//Английский

if (cod > 64 && cod < 91)

{

if (cod + N > 90)

{

output += (char)(text[i] + N - 26);

}

else

{

if (cod + N < 65)

{

output += (char)(text[i] + N + 32);

}

else

{

output += (char)(text[i] + N);

}

}

}

if (cod > 96 && cod < 123)

{

if (cod + N > 122)

{

output += (char)(text[i] + N - 26);

}

else

{

if (cod + N < 97)

{

output += (char)(text[i] + N + 32);

}

else

{

output += (char)(text[i] + N);

}

}

}

}

else

{

output += text[i];

}

}

return output;

}

public static void zadanie2()

{

//Первая часть

Console.WriteLine("Введите сообщение, которое нужно зашифровать");

string IshodText = Console.ReadLine();

Console.WriteLine("\nВведите сдвиг");

int Sdvig = Convert.ToInt32(Console.ReadLine());

string resultat = ROT(Sdvig, IshodText);

Console.WriteLine("\n" + resultat);

//Вторая часть

Console.WriteLine(ROT(-Sdvig, resultat));

Console.ReadKey();

}

//Задание 3

//Первая часть

static string Shifrovanie3(string text)

{

string output = "";

Random k = new Random();

for (int i = 0; i < text.Length; i++)

{

if (text[i] == ' ')

{

output += 256 \* k.Next(1, 4);

}

else

{

byte[] bite = Encoding.GetEncoding(1251).GetBytes(text.Substring(i, 1));

output += Convert.ToInt32(bite[0]) \* k.Next(1, 4);

}

}

return output;

}

static string DeShifrovanie3(string text)

{

string output = "";

int cod = 0;

byte[] bite = new byte[1];

while (text.IndexOf(" ") > (-1))

{

text = text.Replace(" ", "");

}

for (int i = 0; i < text.Length / 3; i++)

{

cod = Convert.ToInt32(text.Substring(i \* 3, 3));

if (cod % 256 == 0)

{

output += " ";

}

else

{

if (cod > 510)

{

bite[0] = Convert.ToByte(cod / 3);

}

else

{

if (cod > 255)

{

bite[0] = Convert.ToByte(cod / 2);

}

else

{

bite[0] = Convert.ToByte(cod);

}

}

output += Encoding.GetEncoding(1251).GetString(bite);

}

}

return output;

}

public static void zadanie3()

{

Console.WriteLine("Введите сообщение, которое нужно зашифровать");

string IshodText = Console.ReadLine();

string resultat = Shifrovanie3(IshodText);

Console.WriteLine("\n" + resultat);

//Вторая часть

Console.WriteLine(DeShifrovanie3(resultat));

Console.ReadKey();

}

//Задание 4

static string[] key4 = new string[4224];

static void SozdanieShifra(int koef)

{

int j = 0;

for (char i = 'А'; i <= 'я'; i++)

{

for (char k = 'А'; k <= 'я'; k++)

{

key4[j] = Convert.ToString(i) + Convert.ToString(k);

j++;

}

}

for (char i = 'А'; i <= 'я'; i++)

{

key4[j] = " " + Convert.ToString(i);

j++;

}

for (char i = 'А'; i <= 'я'; i++)

{

key4[j] = Convert.ToString(i) + " ";

j++;

}

}

public static void zadanie4()

{

//Первая часть

Console.WriteLine("Введите сообщение, которое нужно зашифровать");

string IshodText = Console.ReadLine();

int koef = 1000;

string resultat = "";

SozdanieShifra(koef);

for (int i = 0; i < IshodText.Length / 2; i++)

{

for (int k = 0; k < 4224; k++)

{

if (IshodText.Substring(i \* 2, 2) == key4[k])

{

resultat += Convert.ToString(k + koef);

k = 4224;

}

}

}

if (IshodText.Length % 2 == 1)

{

byte[] bite = Encoding.GetEncoding(1251).GetBytes(IshodText.Substring(IshodText.Length - 1, 1));

resultat += Convert.ToString(bite[0]);

}

Console.WriteLine("\n" + resultat);

//Вторая часть

string resultat2 = "";

for (int i = 0; i < resultat.Length / 4; i++)

{

resultat2 += key4[Convert.ToInt32(resultat.Substring(i \* 4, 4)) - koef];

}

if (resultat.Length % 4 > 0)

{

int index = (resultat.Length / 4) \* 4;

int lengthSimvol = resultat.Length - index;

byte[] bite = new byte[1];

bite[0] = Convert.ToByte(resultat.Substring(index, lengthSimvol));

resultat2 += Encoding.GetEncoding(1251).GetString(bite);

}

Console.WriteLine("\n" + resultat2);

Console.ReadKey();

}

// Задание 5

public static void zadanie5()

{

//Первая часть

Console.WriteLine("Введите сообщение, которое нужно зашифровать");

string IshodText = Console.ReadLine();

Console.WriteLine("\nВведите сдвиг");

int Sdvig = Convert.ToInt32(Console.ReadLine());

string resultat = ROT(Sdvig, IshodText);

Console.WriteLine("\n" + resultat);

resultat = Shifrovanie3(resultat);

Console.WriteLine("\n" + resultat);

//Вторая часть

resultat = DeShifrovanie3(resultat);

Console.WriteLine("\n" + resultat);

resultat = ROT(-Sdvig, resultat);

Console.WriteLine("\n" + resultat);

Console.ReadKey();

}

// Задание 6

public static void zadanie6()

{

string IshodText = " Pbatenghyngvbaf! Vg'f n Pnrfne pvcure!";

Console.WriteLine("Расшифровываем сообщение " + IshodText);

for (int i = 0; i < 26; i++)

{

Console.WriteLine(i + "." + ROT(i, IshodText));

}

Console.ReadKey();

}

// Задание 7

public static void zadanie7()

{

//Первая часть

Console.WriteLine("Введите сообщение, которое нужно зашифровать");

string IshodText = Console.ReadLine();

string resultat = ROT(13, IshodText);

Console.WriteLine("\n" + resultat);

//Вторая часть

Console.WriteLine(ROT(-13, resultat));

Console.ReadKey();

}

}

}

Контрольные вопросы:

1. Какое шифрование называется симметричным?

Симметричные алгоритмы, представляют собой алгоритмы, в которых ключ шифрования может быть рассчитан по ключу дешифрирования и наоборот.

Эти алгоритмы, также называемые алгоритмами с секретным ключом или алгоритмами с одним ключом, требуют, чтобы отправитель и получатель согласовали используемый ключ перед началом безопасной передачи сообщений.

1. Сколько ключей используется в алгоритмах симметричного шифрования?

Используется один ключ.

1. В чем опасность шифрования симметричными алгоритмами?

Безопасность симметричного алгоритма определяется ключом, раскрытие ключа означает, что кто угодно сможет шифровать и дешифрировать сообщения. Пока передаваемые сообщения должны быть тайными, ключ должен храниться в секрете.

1. Опишите основную идею подстановочных шифров.

При подстановочном шифре, каждый символ открытого текста в шифротексте заменяет другим символом. Получатель инвертирует подстановку шифротекста, восстанавливая открытый текст.

1. Перечислите все известные Вам типы перестановочных шифров.

* Простой подстановочный шифр или моноалфавитный шифр;
* Однозвучный подстановочный шифр;
* Полиграммный подстановочный шифр;
* Полиалфавитный подстановочный шиф.

1. Опишите основную идею перестановочных шифров.

В перестановочном шифре меняется не открытый текст, а порядок символов.

1. Назовите достоинства и недостатки перестановочных и подстановочных шифров.

Достоинства:

* Высокая скорость получения шифр-текста;

Недостатки:

* Сохранение частотных характеристик открытого текста;
* Малое число возможных ключей шифрования.