Министерство образования и науки РФ

ФГБОУ ВПО «Тамбовский государственный технический университет»

Кафедра « »

Лабораторная работа №5

по дисциплине «Теория информации»

«Решётка Кардано»

Выполнил: студент гр. −31 Д. Ю.

Проверил: Н. В.

Тамбов,

***Цели и задачи.***

Разработать приложение, реализующее любой выбранный алгоритм шифрования.

***Описание.***

Шифрование с помощью решеток применяется для защиты информации, представляющую ценность в течение ограниченного времени (несколько часов). Этот шифр также является перестановочным, т.е. криптограммы этого шифра представляют собой анаграммы открытого текста. Данный метод шифрования активно применялся во время второй мировой войны, и до сих пор используется в качестве армейского шифра.

***Решение задачи****.*

Приведу пример шифрования строки «барабулька». В строке 10 букв, параметр размера стороны таблицы k = 2, так как . Заполняем таблицу со сторонами k цифрами в порядке увеличения.

|  |  |
| --- | --- |
| 1 | 2 |
| 3 | 4 |

Далее поворачиваем получившееся по часовой стрелке:

|  |  |
| --- | --- |
| 3 | 1 |
| 4 | 2 |

И приписываем к предыдущей таблице:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 1 |
| 3 | 4 | 4 | 2 |

В итоге получим такую таблицу:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 область | |  | 2\_область |
| 1 | 2 | 3 | 1 |
| 3 | 4 | 4 | 2 |
| 2 | 4 | 4 | 3 |
| 1 | 3 | 2 | 1 |
| 4 область | |  | 3 область |

В этом квадрате закрасим произвольным образом все цифры, причем каждая цифра может быть закрашена только один раз. Это и будет решеткой для шифрования. Из второй области закрашена единица, из третьей – двойка, из первой – тройка, а четыре закрашена из третьей области. Получен код 2313.

Разбиваем строку на части по k\*k = 4 буквы:

|  |  |
| --- | --- |
| Разбит: | бара |
|  | буль |
|  | ка\*\* |
|  | \*\*\*\* |

Так как последняя строка не заполнена, а решетка равносторонняя, то дополняем «шумом», толку от этого мало, но не помешает. Далее, берем из первой строки «бара» первый символ «б» и смотрим на код, первая цифра 2, значит в таблице надо во вторую область вписать букву «б» за место цифры 1. Следующая буква «а» и код 3, это третья область и по порядку цифра 2, вписываем «а» за место 2. После прохода по первой строке, необходимо повернуть таблицу (расположение областей) на 90 градусов по часовой стрелке. Проделываем тот же финт, как и в первый раз, единственное отличие в сменившихся областях. Где была 1 область, теперь там 4 и так со всеми 4-мя.

В итоге получим таблицу

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | а | л | б |
| р | \* | 4 | 2 |
| у | ь | а | \* |
| к | 3 | а | б |

В ней помечены места, на которые не хватило букв из строки, это дает право записать любые знаки вместо цифр.

Стойкость шифра рассчитывается по формуле = для рассматриваемой строки в 10 знаков.

***Модули***

**private** void btn\_Ok\_Click**(object** sender**,** EventArgs e**)**

**{**

textBox\_Решето**.**Text **=** ""**;**

\_nline **=** 0**;**

**if(**textBox\_InputTXT**.**Text**.**Length**<**1**)return;**

var k **=** 1**;**

//определить количество клеток у стороны

**while** **(**4**\*(**k**\***k**)** **<** textBox\_InputTXT**.**Text**.**Length**)**

k**++;**

\_k **=** k**;**

label1**.**Text **=** @" Асимптотическая сложность: " **+**

Math**.**Pow**(**4**,** Math**.**Pow**(**k**,** 2**));**

toolStripStatusLabel1**.**Text **=** @"Длина текста " **+** textBox\_InputTXT**.**Text**.**Length **+**

string**.**Format**(**". Стороны решётки {0} на {0}"**,** **(**k **+** k**));**

//строим квадрат со стороной k, вписываем в него числа [1..k\*k]

var quad **=** **new** int**[**k**,** k**];**

var tmp **=** 1**;**

**for** **(**int i **=** 0**;** i **<** k**;** i**++)**

**{**

**for** **(**int j **=** 0**;** j **<** k**;** j**++)**

**{**

quad**[**i**,** j**]** **=** tmp**;**

tmp**++;**

**}**

**}**

//вывод на форму

AddArray**(**quad**,** k**);**

//поворачиваем квадрат по часовой стрелке

var q2 **=** RotateQuad**(**quad**,** k**);**

//вывод на форму

AddArray**(**q2**,** k**);**

//поворачиваем квадрат по часовой стрелке

var q3 **=** RotateQuad**(**q2**,** k**);**

//поворачиваем квадрат по часовой стрелке

var q4 **=** RotateQuad**(**q3**,** k**);**

//вывод на форму

AddArray**(**q4**,** k**);**

//вывод на форму

AddArray**(**q3**,** k**);**

\_1 **=** quad**;**

\_2 **=** q2**;**

\_3 **=** q3**;**

\_4 **=** q4**;**

//разбить входную строку на подстроки длиной k

var strArr **=** SplitInputText**(**textBox\_InputTXT**.**Text**,** k**);**

//вписать строки в позиции \_arrRnd[]

**if** **(**\_arrRnd**!=null)**

**{**

Cripto**(**\_arrRnd**,** strArr**);**

TextOut**();**

**}**

**}**

/// <summary>

/// Накладываем код на строку

/// </summary>

**private** void Cripto**(**int**[]** arrCod**,** string**[,]** strArr**)**

**{**

//первое наложение (без поворота)

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

**{**

//область изменения

int pos **=** arrCod**[**i**];**

//вписать на место порядковой i-цифры букву из strArr[]

var str **=** strArr**[**0**,** i**];**

**if** **(!**string**.**IsNullOrEmpty**(**str**))**

Rest**(**pos**,** str**,** i **+** 1**);**

**}**

//4->1//1->2//2->3//3->4

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

**{**

//область изменения

int pos **=** arrCod**[**i**];**

**switch** **(**pos**)**

**{**

**case** 4**:** pos **=** pos **==** 4 **?** 1 **:** pos**;** **break;**

**case** 1**:** pos **=** pos **==** 1 **?** 2 **:** pos**;** **break;**

**case** 2**:** pos **=** pos **==** 2 **?** 3 **:** pos**;** **break;**

**case** 3**:** pos **=** pos **==** 3 **?** 4 **:** pos**;** **break;**

**}**

//вписать на место порядковой i-цифры букву из strArr[]

var str **=** strArr**[**1**,** i**];**

**if** **(!**string**.**IsNullOrEmpty**(**str**))**

Rest**(**pos**,** str**,** i **+** 1**);**

**}**

//4->2//1->3//2->4//3->1

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

**{**

//область изменения

int pos **=** arrCod**[**i**];**

**switch** **(**pos**)**

**{**

**case** 4**:** pos **=** pos **==** 4 **?** 2 **:** pos**;** **break;**

**case** 1**:** pos **=** pos **==** 1 **?** 3 **:** pos**;** **break;**

**case** 2**:** pos **=** pos **==** 2 **?** 4 **:** pos**;** **break;**

**case** 3**:** pos **=** pos **==** 3 **?** 1 **:** pos**;** **break;**

**}**

//вписать на место порядковой i-цифры букву из strArr[]

var str **=** strArr**[**2**,** i**];**

**if** **(!**string**.**IsNullOrEmpty**(**str**))**

Rest**(**pos**,** str**,** i **+** 1**);**

**}**

//4->2//1->3//2->4//3->1

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

**{**

//область изменения

int pos **=** arrCod**[**i**];**

**switch** **(**pos**)**

**{**

**case** 4**:** pos **=** pos **==** 4 **?** 3 **:** pos**;** **break;**

**case** 1**:** pos **=** pos **==** 1 **?** 4 **:** pos**;** **break;**

**case** 2**:** pos **=** pos **==** 2 **?** 1 **:** pos**;** **break;**

**case** 3**:** pos **=** pos **==** 3 **?** 2 **:** pos**;** **break;**

**}**

//вписать на место порядковой i-цифры букву из strArr[]

var str **=** strArr**[**3**,** i**];**

**if** **(!**string**.**IsNullOrEmpty**(**str**))**

Rest**(**pos**,** str**,** i **+** 1**);**

**}**

**}**

**private** void Rest**(**int pos**,** string str**,** int index**)**

**{**

**if** **(**pos **==** 1**)**

**{**

**for** **(**int i **=** 0**;** i **<** \_k**;** i**++)**

**{**

**for** **(**int j **=** 0**;** j **<** \_k**;** j**++)**

**{**

**if(**\_1**[**i**,** j**]** **==** index**)**

\_1**[**i**,** j**]** **=** Convert**.**ToChar**(**str**);**

**}**

**}**

**}**

**if** **(**pos **==** 2**)**

**{**

**for** **(**int i **=** 0**;** i **<** \_k**;** i**++)**

**{**

**for** **(**int j **=** 0**;** j **<** \_k**;** j**++)**

**{**

**if(**\_2**[**i**,** j**]** **==** index**)**

\_2**[**i**,** j**]** **=** Convert**.**ToChar**(**str**);**

**}**

**}**

**}**

**if** **(**pos **==** 3**)**

**{**

**for** **(**int i **=** 0**;** i **<** \_k**;** i**++)**

**{**

**for** **(**int j **=** 0**;** j **<** \_k**;** j**++)**

**{**

**if(**\_3**[**i**,** j**]** **==** index**)**

\_3**[**i**,** j**]** **=** Convert**.**ToChar**(**str**);**

**}**

**}**

**}**

**if** **(**pos **==** 4**)**

**{**

**for** **(**int i **=** 0**;** i **<** \_k**;** i**++)**

**{**

**for** **(**int j **=** 0**;** j **<** \_k**;** j**++)**

**{**

**if** **(**\_4**[**i**,** j**]** **==** index**)**

\_4**[**i**,** j**]** **=** Convert**.**ToChar**(**str**);**

**}**

**}**

**}**

**}**

/// <summary>

/// Разбить строку на части длиной k\*k

/// </summary>

/// <param name="str"></param>

/// <param name="k"></param>

**private** string**[,]** SplitInputText**(**string str**,** int k**)**

**{**

//if(\_arrRnd==null) return;

var chrArr **=** **new** string**[**k**+**k**,**k**\***k**];**

var indx **=** 0**;**

int ii**,** jj**;**

**for** **(**int i **=** 0**;** **;** i**++)**

**{**

**for** **(**int j **=** 0**;** j **<** k**\***k**;** j**++)**

**{**

**if** **(**indx **<** str**.**Length**)** chrArr**[**i**,** j**]** **=** str**[**indx**].**ToString**();**

**else**

**{**

ii **=** i**;**

jj **=** j**;**

**if(**jj**!=**0**)**

**for** **(**int ie **=** jj**;** ie **<** k **\*** k**;** ie**++)**

**{**

chrArr**[**ii**,** ie**]** **=** "•"**;**

**}**

**return** chrArr**;**

**}**

indx**++;**

**}**

**}**

**}**

/// <summary>

/// Поворот входного массива по часовой стрелке

/// </summary>

/// <returns></returns>

**private** int**[,]** RotateQuad**(**int **[,]** arr**,** int k**)**

**{**

var extArr **=** **new** int**[**k**,**k**];**

//столбец переписать в строку

**for** **(**int i **=** 0**;** i **<** k**;** i**++)**

**{**//столбец

**for** **(**int j **=** 0**,** r **=** k**-**1**;** j **<** k**;** j**++,** r**--)**

**{**//строки

extArr**[**i**,** r**]** **=** arr**[**j**,** i**];**

**}**

**}**

**return** extArr**;**

**}**

/// <summary>

/// Разбросать в четыре области решетки цифры от 1 до k\*k

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

**private** void btn\_Rnd\_Click**(object** sender**,** EventArgs e**)**

**{**

flagRnd **=** **!**flagRnd**;**

toolStripStatusLabel2**.**Text **=** " Код ≈"**;**

\_arrRnd **=** **new** int**[**\_k**\***\_k**];**

**for** **(**int i **=** 0**;** i **<** \_k**\***\_k**;** i**++)**

**{**

\_arrRnd**[**i**]** **=** \_rnd**.**Next**(**1**,** 5**);**

toolStripStatusLabel2**.**Text **+=** " " **+** \_arrRnd**[**i**];**

**}**

**if** **(**\_k **\*** \_k **>** 50**)**

**{**

MessageBox**.**Show**(**toolStripStatusLabel2**.**Text**);**

toolStripStatusLabel2**.**Text **=** "≈≈"**;**

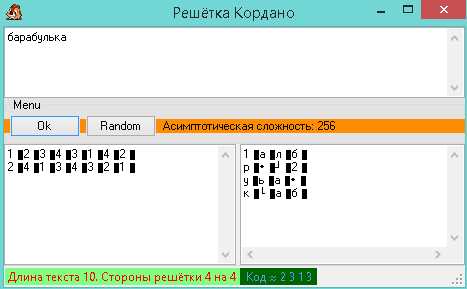
**}**

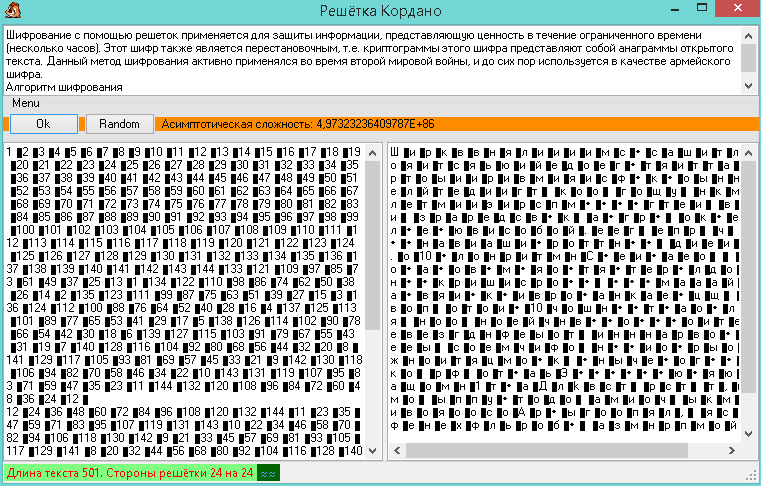
**}**

***Выбор языка программирования***

Язык C#, самый легкий и больше всего изученный мною.

***Тестирование программы***

******

******