Министерство образования Республики Беларусь

Учреждение образования

“Белорусский государственный университет информатики и радиоэлектроники”

Кафедра инженерной психологии и эргономики

Практическая работа №2

«Маршрутные и подстановочные шифры»

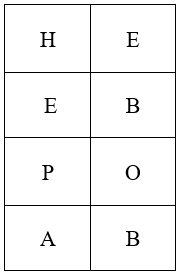
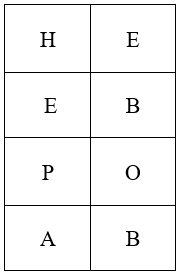
Проверил: Выполнила:

Давыдович К.И. ст. гр.910903 Неверова А.П.

Минск 2022

**Цель работы:** зашифровать свою фамилию шифром маршрутной перестановки, найти ключ шифра Плейфера, написать код шифровки своих фамилии, имени, отчества.

**Задание №1:**

** **

**Задание №2:**

Открытый текст:

Please note that spaces and punctuation characters have been removed before encryption.

Шифротекст:

LMBEUDOPUASIIYUNEDDUDOENSPARTEYOPODGCTEDUASTIDZBCBDPUCNPZBACBKMTDZDPGWZOYOPO

Найти ключ.

Ход работы:

1. Разобьём текста на биграммы (в случае, если биграмма состоит из одинаковых букв, между ними добавляется буква Х):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PL | EA | SE | NO | TE | TH | AT | SP | AC | ES | AN | DP | UN | CT | UA |
| LM | BE | UD | OP | UA | SI | IY | UN | ED | DU | DO | EN | SP | AR | TE |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TI | ON | CH | AR | AC | TE | RS | HA | VE | BE | EN | RE | MO | VE | DB |
| YO | PO | DG | CT | ED | UA | ST | ID | ZB | CB | DP | UC | NP | ZB | AC |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| EF | OR | EX | EN | CR | YP | TI | ON |
| BK | MT | DZ | DP | GW | ZO | YO | PO |

1. Выпишем триграммы, которые располагаются в одном ряду/строке:

PLM, AEB, NOP, RST, EBC.

Делаем вывод, что AEBC в какой-то строке/столбце:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| А | E | B | C |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Замечая пару биграмм AC-ED делаем вывод, что D стоит в одном ряду с AEBC и получаем следующее:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| А | E | B | C | D |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Биграммы TE-UA, AT-IY, TI-YO дают заметить, что буквы A, T, I, Y, O расположены в одном ряду. Исходя из этого, получаем следующее:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Y |  |  |  |  |
| А | E | B | C | D |
| I |  |  |  |  |
| O |  |  |  |  |
| T |  |  |  |  |

Теперь можно заполнить таблицу, зная, что у нас есть триграммы, приведенные выше и получить следующее:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Y |  |  |  |  |
| А | E | B | C | D |
| I |  |  |  |  |
| O | P | L | M | N |
| T |  |  | R | S |

Используя оставшиеся биграммы получаем следующий результат:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Y | Z | V | W | X |
| А | E | B | C | D |
| I | K | F | G | H |
| O | P | L | M | N |
| T | U |  | R | S |

Видим, что не хватает одной буквы, и это буква Q. Заполним таблицу до конца:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Y | Z | V | W | X |
| А | E | B | C | D |
| I | K | F | G | H |
| O | P | L | M | N |
| T | U | Q | R | S |

Чтобы привести ключ к правильному виду, сместим строки вверх на 1 ряд и столбцы влево на 2 ряда, получим окончательный ключ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B | C | D | A | E |
| F | G | H | I | K |
| L | M | N | O | P |
| Q | R | S | T | U |
| V | W | X | Y | Z |

**Задание №3:**

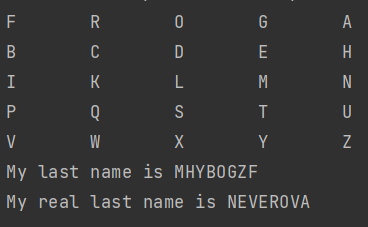
* Придумать ключ.
* Написать код шифровки своих фамилии, имени, отчества.
* Написать код дешифровки своих фамилии, имени, отчества.

Ключ: FROG.

Код программы:

namespace CaesarsSipher;  
  
public static class Playfair  
{  
 private const string **Alphabet** = "ABCDEFGHIGKLMNOPQRSTUVWXYZ";  
 private const string **Key** = "FROG";  
 private const string **LastName** = "LOGVIN";  
  
 private static char[,] GetKeyTable(string key)  
 {  
 var newAlphabet = new List<char>();  
 newAlphabet.AddRange(key);  
 newAlphabet.AddRange(**Alphabet**.Except(**Key**));  
 char[,] keyTable = new Char[5, 5];  
 int index = 0;  
 for (int i = 0; i < 5; i++)  
 {  
 for (int j = 0; j < 5; j++)  
 {  
 keyTable[i, j] = newAlphabet[index];  
 index++;  
 Console.Write("{0}\t", keyTable[i, j]);  
 }  
  
 Console.WriteLine();  
 }  
  
 return keyTable;  
 }  
  
 private static List<string> GetBigrams(string value)  
 {  
 var bigrams = new List<string>();  
  
 for (var i = 0; i < value.Length; i++)  
 {  
 var bigram = value.Substring(i, value.Length - i == 1 ? 1 : 2);  
 if (bigram.Length == 1 || bigram[0] == bigram[1])  
 {  
 bigram = $"{bigram[0]}{'X'}";  
 }  
 else  
 {  
 i++;  
 }  
  
 bigrams.Add(bigram);  
 }  
  
 return bigrams;  
 }  
  
 private static List<char> Sipher(List<string> bigrams, char[,] keyTable, bool isUnsipher = false)  
 {  
 var result = new List<char>();  
 foreach (var bigram in bigrams)  
 {  
 var first = GetPosition(keyTable, bigram[0]);  
 var second = GetPosition(keyTable, bigram[1]);  
 if (first.Item1 == second.Item1)  
 {  
 result.Add(keyTable[first.Item1, (isUnsipher ? first.Item2 + 4 : first.Item2 + 1) % 5]);  
 result.Add(keyTable[second.Item1, (isUnsipher ? second.Item2 + 4 : second.Item2 + 1) % 5]);  
 }  
 else  
 {  
 if (first.Item2 == second.Item2)  
 {  
 result.Add(keyTable[(isUnsipher ? first.Item1 - 1 : first.Item1 + 1) % 5, first.Item2]);  
 result.Add(keyTable[(isUnsipher ? second.Item1 - 1 : second.Item1 + 1) % 5, second.Item2]);  
 }  
 else  
 {  
 result.Add(keyTable[first.Item1, second.Item2]);  
 result.Add(keyTable[second.Item1, first.Item2]);  
 }  
 }  
 }  
  
 return result;  
 }  
  
 private static Tuple<int, int> GetPosition(char[,] keyTable, char symbol)  
 {  
 for (int x = 0; x < 5; ++x)  
 {  
 for (int y = 0; y < 5; ++y)  
 {  
 if (keyTable[x, y].Equals(symbol))  
 return Tuple.Create(x, y);  
 }  
 }  
  
 return Tuple.Create(-1, -1);  
 }  
  
 public static void ShowWork()  
 {  
 var keyTable = GetKeyTable(**Key**);  
 var bigrams = GetBigrams(**LastName**);  
 var sipher = Sipher(bigrams, keyTable);  
 Console.WriteLine($"My last name is {string.Join(String.Empty, sipher)}");  
 var sipherBigrams = GetBigrams(string.Join(String.Empty, sipher));  
 var realLastName = Sipher(sipherBigrams, keyTable, true);  
 Console.WriteLine($"My real last name is {string.Join(String.Empty, realLastName)}");  
 }  
}

Результат работы программы:



**Вывод:** зашифровала свою фамилию шифром маршрутной перестановки, нашла ключ шифра Плейфера с заданным исходным текстом и шифротекстом, написала код шифровки и расшифровки по Плейферу с заданным ключом.