МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5 НА ТЕМУ:**

**Исследование криптографических шифров на основе перестановки символов**

Выполнила студентка 3 курса 5 группы

Авсюкевич Полина

Минск 2024

**Задание 1:** Разработать авторское приложение в соответствии с целью лабораторной работы. Приложение должно реализовывать следующие операции: выполнять зашифрование/расшифрование текстовых документов (объемом не менее 500 знаков), созданных на основе алфавита Белорусского языка в соответствии с маршрутной перестановкой. Реализация данного функционала представлена в листинге 1.

|  |
| --- |
| public string Encrypt(string msg, int[] key1, int[] key2)  {  string result = string.Empty;  string[] msgInArray = new string[(msg.Length / key1.Length) + 1];  for (int i = 0; i < msg.Length / key1.Length + 1; i++)  {  int length = Math.Min(key1.Length, msg.Length - i \* key1.Length);  msgInArray[i] = msg.Substring(i \* key1.Length, length);  Console.WriteLine($"msgInArray[{i}] = {msgInArray[i]}");  }  char[,] res = new char[key1.Length, key2.Length];  for (int i = 0; i < key1.Length; i++)  {  for (int k = 0; k < key2.Length; k++)  {  if (k < msgInArray.Length && i < msgInArray[k].Length)  {  res[key1[i], key2[k]] = msgInArray[k][i];  }  }  }  result = GetStringFromMatrix(res);  return result;  }  public string Decrypt(string msg, int[] key1, int[] key2)  {  string result = string.Empty;  string[] msgInArray = new string[(msg.Length / key1.Length) + 1];  for (int i = 0; i < msg.Length / key1.Length + 1; i++)  {  int length = Math.Min(key1.Length, msg.Length - i \* key1.Length);  msgInArray[i] = msg.Substring(i \* key1.Length, length);  }  char[,] res = new char[key1.Length, key2.Length];  for (int i = 0; i < key1.Length; i++)  {  for (int k = 0; k < key2.Length; k++)  {  if (k < msgInArray.Length && key1[i] < msgInArray[k].Length)  {  res[i, k] = msgInArray[k][key1[i]];  }  }  }  result = GetStringFromMatrix(res);  return result;  } |

Листинг 1 – Реализация для маршрутной перестановки

Гистограмма частот для зашифрованного и исходного текста идентичны, тк используются перестановки, и представлена на рисунке 1.

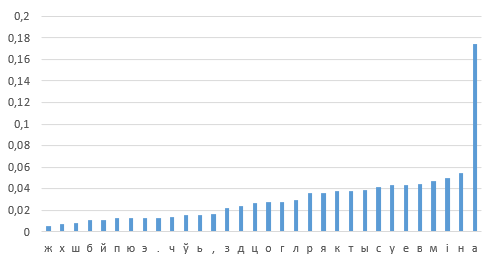


Рисунок 1 – Гистограмма частот для текста

Время для шифрования и расшифрования представлена на рисунках 2 и 3.

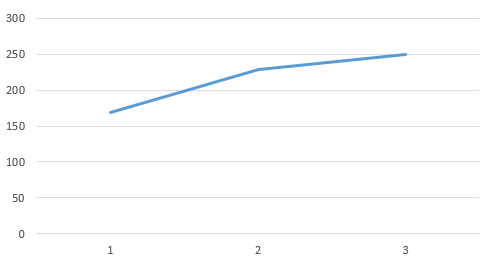


Рисунок 2 – Время выполнения зашифрования

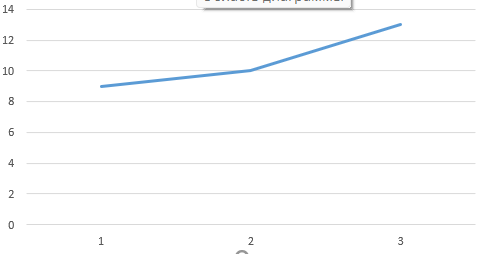


Рисунок 3 – Время выполнения расшифрования

**Задание 2:** Выполнение зашифрования/расшифрования фамилии и имени созданных на основе алфавита Белорусского языка в соответствии с множественной перестановкой представлен в листинге 1.

|  |
| --- |
| public string Encrypt(string msg, int[] key1, int[] key2)  {  string result = string.Empty;  string[] msgInArray = new string[(msg.Length / key1.Length) + 1];  for (int i = 0; i < msg.Length / key1.Length + 1; i++)  {  int length = Math.Min(key1.Length, msg.Length - i \* key1.Length);  msgInArray[i] = msg.Substring(i \* key1.Length, length);  Console.WriteLine($"msgInArray[{i}] = {msgInArray[i]}");  }  char[,] res = new char[key1.Length, key2.Length];  for (int i = 0; i < key1.Length; i++)  {  for (int k = 0; k < key2.Length; k++)  {  if (k < msgInArray.Length && i < msgInArray[k].Length)  {  res[key1[i], key2[k]] = msgInArray[k][i];  }  }  }  result = GetStringFromMatrix(res);  return result;  }  public string Decrypt(string msg, int[] key1, int[] key2)  {  string result = string.Empty;  string[] msgInArray = new string[(msg.Length / key1.Length) + 1];  for (int i = 0; i < msg.Length / key1.Length + 1; i++)  {  int length = Math.Min(key1.Length, msg.Length - i \* key1.Length);  msgInArray[i] = msg.Substring(i \* key1.Length, length);  }  char[,] res = new char[key1.Length, key2.Length];  for (int i = 0; i < key1.Length; i++)  {  for (int k = 0; k < key2.Length; k++)  {  if (k < msgInArray.Length && key1[i] < msgInArray[k].Length)  {  res[i, k] = msgInArray[k][key1[i]];  }  }  }  result = GetStringFromMatrix(res);  return result;  } |

Листинг 2 – Реализация множественной перестановки

Гистограмма частот для зашифрованного и исходного текста идентичны, тк используются перестановки, и представлена на рисунке 4.

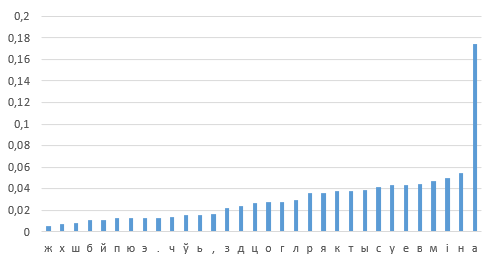


Рисунок 4 – Гистограмма частот для текста

Время для шифрования и расшифрования представлена на рисунках 5 и 6.

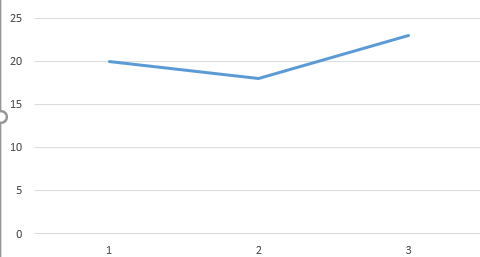


Рисунок 5 – Время выполнения зашифрования

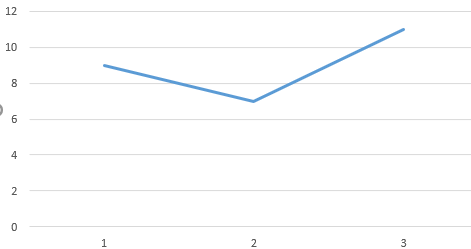


Рисунок 6 – Время выполнения расшифрования

**Вывод:** в ходе лабораторной работы было разработано приложение, позволяющее шифровать и расшифровывать сообщения с помощью шифров множественной перестановки и маршрутной перестановки. Были построены гистограммы частоты встречаемости символов и измерено время шифрования/дешифования. Временные затраты незначительны.