МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2 НА ТЕМУ:**

**ЭЛЕМЕНТЫ ТЕОРИИ ИНФОРМАЦИИ. ИНФОРМАТИВНОСТЬ ДАННЫХ В РАЗЛИЧНЫХ КОДИРОВКАХ**

Выполнила студентка 3 курса 5 группы

Авсюкевич Полина Вадимовна

Минск 2024

Задание 1: создать собственное приложение или воспользоваться Base64-онлайн-кодировщиком, с помощью которого конвертировать произвольный документ (а) на латинице (можно использовать документ из лабораторной работы № 1) в документ (б) формата base64. Код выполненного задания представлен в листенге 1.

|  |
| --- |
| public void ConvertToBase64()  {  string base64Result = "";  using (StreamReader reader = new StreamReader(@"C:\instit\kurs3\_2\IB\lab2\lab3\pol.txt"))  {  string fileContent = reader.ReadToEnd();  Console.WriteLine("text:");  Console.WriteLine(fileContent);  Console.WriteLine();  byte[] asciiBytes = Encoding.ASCII.GetBytes(fileContent);  base64Result = Convert.ToBase64String(asciiBytes);  Console.WriteLine("Converted to base64:");  Console.WriteLine(base64Result);  }  using (StreamWriter writer = new StreamWriter(@"C:\instit\kurs3\_2\IB\lab2\lab3\base64.txt", false))  {  writer.WriteLine(base64Result);  }} |

Листинг 1 – Перевод к Base64

Результат выполнение на рисунке 1.

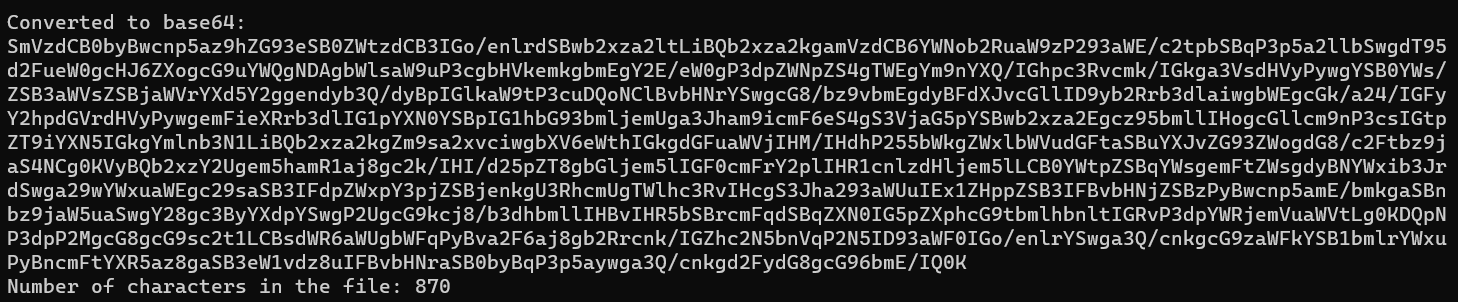


Рисунок 1 – Строка Base64

Задание 2: С помощью приложения, созданного в лабораторной работе № 1, получить распределение частотных свойств алфавитов по документам (а) и (б). Вычислить энтропию Хартли и Шеннона, а также избыточность алфавитов.

Для выполнения данного задания была написана функция, представленная в листинге 2.

|  |
| --- |
| public static void CalculateEntropy(string alphabet)  {  int[] letterCount = new int[alphabet.Length];  int totalLettersInFile = 0;  double[] letterProbabilities = new double[alphabet.Length];  using (StreamReader reader = new StreamReader(@"C:\instit\kurs3\_2\IB\lab2\lab3\pol.txt"))  {  string fileContent = reader.ReadToEnd().ToLower();  totalLettersInFile = fileContent.Length;  Console.WriteLine($"Number of characters in the file: {totalLettersInFile}");  for (int j = 0; j < alphabet.Length; j++)  {  letterCount[j] = fileContent.Count(x => x == alphabet[j]);  if (letterCount[j] != 0)  {  Console.WriteLine($"{alphabet[j]}: {letterCount[j]}");  letterProbabilities[j] = (double)letterCount[j] / totalLettersInFile;  Console.WriteLine($"P({alphabet[j]}): {letterProbabilities[j]}");  Console.WriteLine();  ShannonEntropy += letterProbabilities[j] \* (Math.Log(letterProbabilities[j]) / Math.Log(2)) \* (-1);  }  }  Console.WriteLine($"Entropy of the alphabet by Shannon: {ShannonEntropy}");  } |

Листинг 2 – Расчет энтропии

Результат выполнени представлен на рисунке 2.

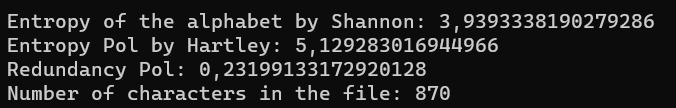


Рисунок 2 – Энтропия польского текста

Задание 3: Написать функцию, которая принимает в качестве аргументов два буфера (*а* и *b*) одинакового размера и возвращает XOR (собственная фамилия (*а*) и имя (*b*); при разной длине меньшую дополнить нулями). Входные аргументы представлять: 1) в кодах ASCII; 2) в кодах base64. Что будет результатом операции *а*XOR*b*XOR*b*?

|  |
| --- |
| byte[] asciiName = Encoding.ASCII.GetBytes(name);  byte[] asciiFamily = Encoding.ASCII.GetBytes(family);  // Ensure both arrays have the same length by padding the shorter one with zeros  int maxLength = Math.Max(asciiName.Length, asciiFamily.Length);  Array.Resize(ref asciiName, maxLength);  Array.Resize(ref asciiFamily, maxLength);  // Perform XOR operation in ASCII  byte[] xorResultAscii = new byte[maxLength];  for (int i = 0; i < maxLength; i++)  {  xorResultAscii[i] = (byte)(asciiName[i] ^ asciiFamily[i] ^ asciiFamily[i]);  }  // Convert byte array to string  string resultAsciiString = Encoding.ASCII.GetString(xorResultAscii);  // Convert strings to byte arrays with Base64 encoding  byte[] base64Name = Convert.FromBase64String(Convert.ToBase64String(Encoding.UTF8.GetBytes(name)));  byte[] base64Family = Convert.FromBase64String(Convert.ToBase64String(Encoding.UTF8.GetBytes(family)));  maxLength = Math.Max(base64Name.Length, base64Family.Length);  Array.Resize(ref base64Name, maxLength);  Array.Resize(ref base64Family, maxLength);  Perform XOR operation in Base64  byte[] xorResultBase64 = new byte[maxLength];  for (int i = 0; i < maxLength; i++)  {  xorResultBase64[i] = (byte)(base64Name[i] ^ base64Family[i] ^ base64Family[i]);  }  // Convert byte array to string  string resultBase64String = Convert.ToBase64String(xorResultBase64);  Console.WriteLine("\n\nResult of XOR in ASCII:");  Console.WriteLine(resultAsciiString);  Console.WriteLine("\nResult of XOR in Base64:");  Console.WriteLine(resultBase64String); |

Листинг 3 – Листинг функции

Результат выполнение на рисунке 3.

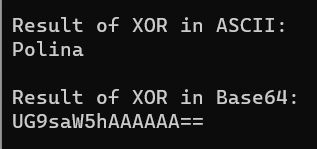


Рисунок 3 – Результат выполнения

Гистограмма для энтропии base64 представлена на рисунке 4.

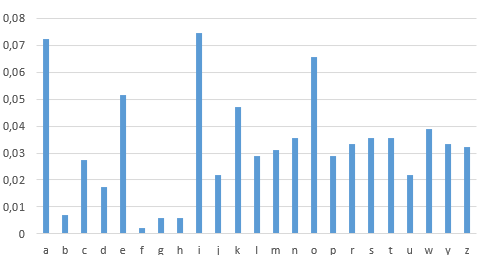


Рисунок 4 – Гистограмма для энтропии base64

По анализу данной гистограммы, можно сделать вывод, что чаще всего повторяющийся символ: «*i*».

Вывод:лабораторная работа демонстрирует важность анализа информации с использованием энтропии и избыточности. Операция XOR применяется как к данным в формате ASCII, так и в формате base64, и результаты, полученные без использования стандартных функций, соответствуют результатам с использованием стандартных функций.