МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность Информационные системы и технологии

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3 НА ТЕМУ:**

**Элементы теории информации. Информативность данных в различных кодировках**

Выполнила студентка 3 курса 1 группы

Пригодич Вера Валерьевна

Минск 2022

**Задание 1:** Конвертировать случайное число (от 999999) в формат base64.

Для выполнения задания был использован следующий код на языке C#:

public static string ConvertToBase64(string inputStr)

{

var base64chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";

var r = "";

var p = "";

var c = inputStr.Length % 3;

// add a right zero pad to make this string a multiple of 3 characters

if (c > 0)

{

for (; c < 3; c++)

{

p += '=';

inputStr += "\0";

}

}

// increment over the length of the string, three characters at a time

for (c = 0; c < inputStr.Length; c += 3)

{

// we add newlines after every 76 output characters, according to the MIME specs

if (c > 0 && (c / 3 \* 4) % 76 == 0)

{

r += "\r\n";

}

// these three 8-bit (ASCII) characters become one 24-bit number

var n = (Convert.ToString(inputStr[c], 2).PadLeft(8, '0') + Convert.ToString(inputStr[c + 1], 2).PadLeft(8, '0') + Convert.ToString(inputStr[c + 2], 2).PadLeft(8, '0'));

// this 24-bit number gets separated into four 6-bit numbers

// those four 6-bit numbers are used as indices into the base64 char\_acter list

var newList = Enumerable.Range(0, n.Count() / 6).Select(x => n.Substring(x \* 6, 6)).ToList();

r+= string.Join("", newList.Select(x => base64chars[Convert.ToByte(x,2)]));

}

// add the actual padding string, after removing the zero pad

return r.Substring(0, r.Length - p.Length) + p;

}

Листинг 1 – задание 1

**Задание 2:** Получить распределение частотных свойств алфавитов по документам (а) и (б), вычислить энтропию Хартли и Шеннона, а также избыточность алфавитов.

Для выполнения этого задания код, разработанный в предыдущем задании был дополнен кодом из предыдущей лабораторной работы, а также следующим кодом:

public static double HartleyEntropy(int count)

{

return Math.Log(count, 2);

}

public static double Redundancy(double EntSh, double EntH)

{

return ((EntH - EntSh) / EntH) \* 100;

}

Листинг 2 – задание 2

**Задание 3:** Написать функцию, которая принимает в качестве аргументов два буфера 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.

Дополним разработанный в предыдущем задании код следующим методом:

public static string XOR(string str1, string str2)

{

string result = null;

if (str1.Length > str2.Length)

{

for (int i = 0; i < (str1.Length - str2.Length); i++)

{

str2 += '0';

}

}

else if (str1.Length < str2.Length)

{

for (int i = 0; i < (str2.Length - str1.Length); i++)

{

str1 += '0';

}

}

for (int i = 0; i < str1.Length; i++)

{

if (str1[i] == str2[i])

{

result += "0";

}

else

{

result += "1";

}

}

return result;

}

Листинг 3 – задание 3

Результат работы итогового кода:

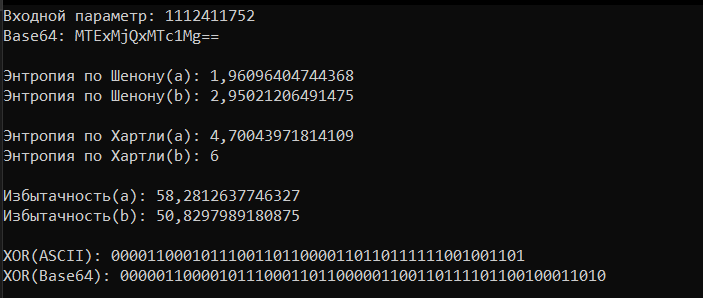


Рисунок 1 ­­­­– Результат работы программы

**Вывод:** в ходе лабораторной работы были приобретены практические навыки трансформации данных и сопоставление энтропийных свойств используемых при этом алфавитов.