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**

Выполнил студент 3 курса 1 группы

Халалеенко Андрей Николаевич

Минск 2024

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

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

**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.

Для выполнения заданий был использован следующий код на языке Typescript:

import { createBase64Doc, createReport, XOR } from "./third";

import { entropyOfAlphabet, Alphabets } from "./second";

// Функция для преобразования байтов в строку бинарного представления

function bytesToBinaryString(bytes: Uint8Array): string {

    return Array.from(bytes, byte => byte.toString(2).padStart(8, '0')).join('');

}

function main(): void {

    createBase64Doc();

    console.log(createReport());

    console.log("ASCII XOR: ");

    const surname = "Halaleenko";

    const name = "Andrew";

    let surnameASCII = "";

    let nameASCII = "";

    for (const ch of surname) {

        surnameASCII += ch.charCodeAt(0);

    }

    for (const ch of name) {

        nameASCII += ch.charCodeAt(0);

    }

    while (surnameASCII.length !== nameASCII.length) {

        nameASCII += '0';

    }

    // XOR для ASCII

    console.log(bytesToBinaryString(XOR(Buffer.from(surnameASCII, 'utf-16le'), Buffer.from(nameASCII, 'utf-16le'))));

    console.log("Base64 XOR: ");

    const name64 = Buffer.from(name).toString('base64');

    const surname64 = Buffer.from(surname).toString('base64');

    let paddedName64 = name64;

    while (paddedName64.length !== surname64.length) {

        paddedName64 += '0';

    }

    // XOR для Base64

    console.log(bytesToBinaryString(XOR(Buffer.from(surname64, 'utf-16le'), Buffer.from(paddedName64, 'utf-16le'))));

    console.log("aXORbXORb: \t");

    // XOR для aXORbXORb

    const aXORbXORb = XOR(Buffer.from(surname64, 'utf-16le'), XOR(Buffer.from(name64, 'utf-16le'), Buffer.from(surname64, 'utf-16le')));

    console.log("aXORbXORb: \t");

    console.log(Buffer.from(aXORbXORb).toString('hex')); // Используем 'hex' для вывода

}

main();

Листинг 1 – задания 1, 2, 3

Результат работы итогового кода:

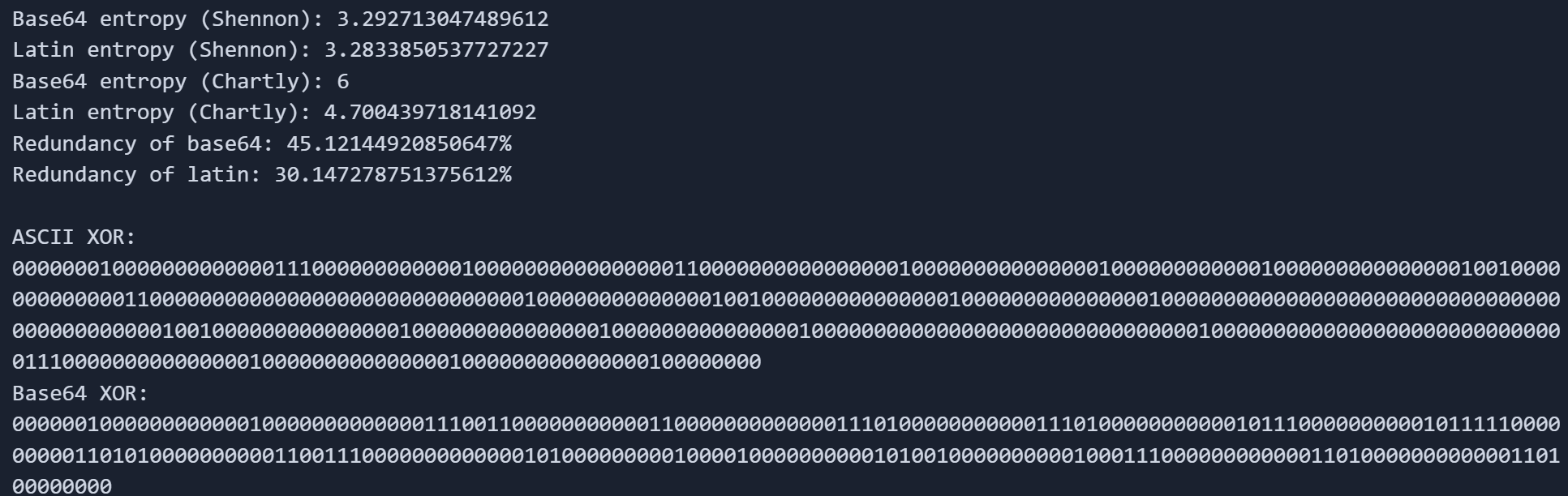


Рисунок 1 ­­­­– Результат работы программы

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