Київський національний університет імені Тараса Шевченка
Факультет радіофізики, електроніки і комп’ютерних систем

# Звіт
З лабораторної роботи №1
Предмету «Комп’ютерні системи»

Студента 3-го курсу
Спеціальності КІ МА
Горелова Артема

Київ 2021

Хід роботи

I. Дослідження кількості інформації в тексті

1. Оберіть 3 текстових файла різного тематичного та лінгвістичного спрямування.

Обрані тексти:

- Фрагмент листа Нестора ІвановичаМахно з власних мемуарів (mahno.txt)
- Уривок з “Симулякри і симуляція” Жана Бодрійяра (bodriyar.txt)
- Фрагмент інтерв'ю Леоніда Макаровича Кравчука (kravchuk.txt)

2. Переконайтесь, що тексти, які ви використовуєте є унікальними і не повторюються у ваших колег.

3. Створіть програму (будь-якою зручною для вас мовою), яка в якості вхідних даних приймає текстовий файл, та аналізує його вміст

Лістинг програми:

```
using System;
using System.Collections.Generic;
using System.IO;

namespace lab1._1
{
    class Program
    {
        static void Main(string[] args)
        {
            bool err = true;
            int symbolsQ;
            double entropy;
            Dictionary<char, double> symbolsDict = new Dictionary<char, double>();
            try
            {
                do
                {
                    string path = Input();
                    if (File.Exists(path))
                    {
                        FileInfo file = new FileInfo(path);
                        long sizeF = file.Length;
                        symbolsQ = FillDict(path, symbolsDict);
                        Probability(symbolsDict, symbolsQ);
                        entropy = Entropy(symbolsDict);
                        Output(symbolsDict, symbolsQ, entropy, sizeF);
                        err = false;
                    }
                    else
                    {
                        Console.ForegroundColor = ConsoleColor.DarkYellow;
                        Console.WriteLine("Something went wrong, try again");
                        Console.ForegroundColor = ConsoleColor.Green;
                        Console.WriteLine("-----------------------------------------------
--------");
```

```
                }
            } while (err == true);
        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }
        Console.ReadKey();
    }

    public static string Input()
    {
        string nameF;
        string path = @"C:\Users\artem\Desktop\CompSys\lab1\";
        Console.ForegroundColor = ConsoleColor.DarkYellow;
        Console.Write("Введите название файла: ");
        nameF = Console.ReadLine();
        Console.ForegroundColor = ConsoleColor.Green;
        Console.WriteLine("-----------------------------------------------------");
        path += nameF + ".txt";

        return path;
    }
    public static int FillDict(string path, Dictionary<char, double> symbols)
    {
        string text = File.ReadAllText(path);
        int sizeF = text.Length;
        for (int i = 0; i < sizeF; i++)
        {
            if (symbols.ContainsKey(text[i])) symbols[text[i]]++;
            else symbols.Add(text[i], 1);
        }
        return sizeF;
    }

    public static double Entropy(Dictionary<char, double> symbols)
    {
        int elementQ = symbols.Keys.Count;
        char[] keysDict = new char[elementQ];
        symbols.Keys.CopyTo(keysDict, 0);
        double entr = 0;
        for (int i = 0; i < elementQ; i++) entr -= symbols[keysDict[i]] *
Math.Log(symbols[keysDict[i]], 2);
        return entr;
    }

    public static void Probability(Dictionary<char, double> symbols, int symbolsQ)
    {
        int elementQ = symbols.Keys.Count;
        char[] keysDict = new char[elementQ];
        symbols.Keys.CopyTo(keysDict, 0);
        for (int iter = 0; iter < elementQ; iter++) symbols[keysDict[iter]] /=
symbolsQ;
    }

    public static void Output(Dictionary<char, double> symbols, int symbolsQ, double
entr, long fileSize)
    {
        char[] alphabet = new char[32] {
            'а', 'б', 'в', 'г', 'ґ', 'д', 'е', 'є', 'ж', 'з', 'и',
            'ї', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с',
            'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ь', 'ю', 'я'
        };
        Console.ForegroundColor = ConsoleColor.DarkYellow;
        Console.WriteLine("File size = {0} bytes", fileSize);
        Console.WriteLine("The amount of information = {0:F4} bytes", symbolsQ * entr
/ 8);
        Console.WriteLine("Common entropy = {0:F5}", entr);
        Console.ForegroundColor = ConsoleColor.Green;
```

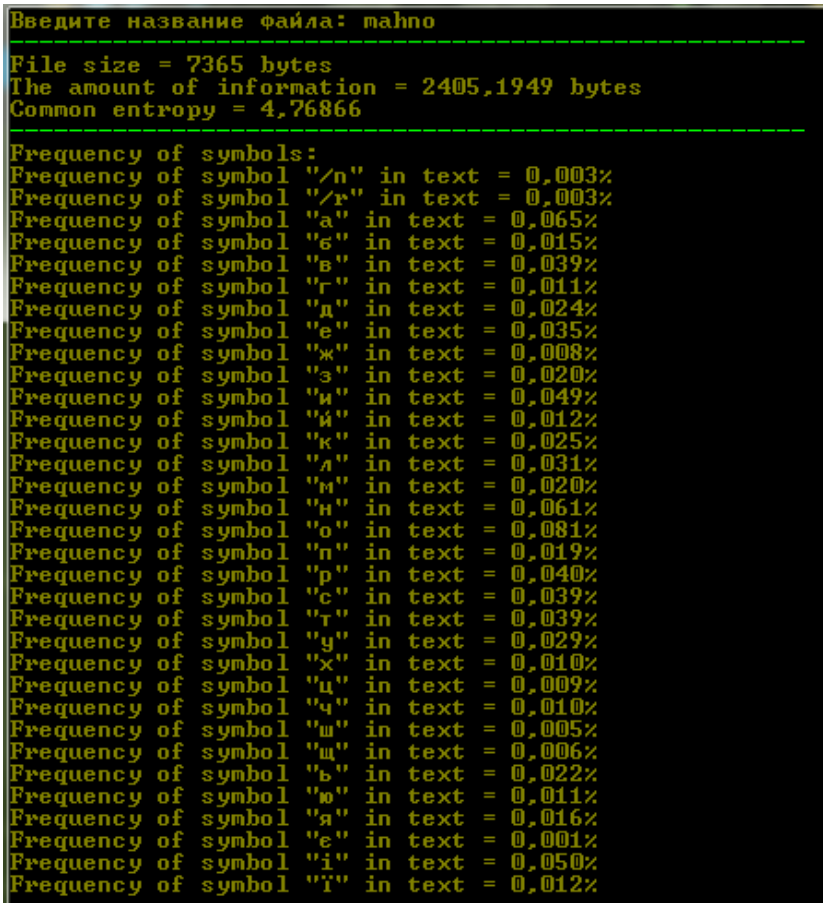
```
            Console.WriteLine("------------------------------------------------------");
            SortedDictionary<char, double> sorted = new SortedDictionary<char,
double>(symbols);
            Console.ForegroundColor = ConsoleColor.DarkYellow;
            Console.WriteLine("Frequency of symbols:");
            foreach (KeyValuePair<char, double> n in sorted)
            {
                switch (n.Key)
                {
                    case '\r':
                        Console.WriteLine("Frequency of symbol \"/r\" in text = {0:f3}%",
n.Value);
                        break;
                    case '\n':
                        Console.WriteLine("Frequency of symbol \"/n\" in text = {0:f3}%",
n.Value);
                        break;
                    case 'i':
                        Console.WriteLine("Frequency of symbol \"i\" in text = {0:f3}%",
n.Value);
                        break;
                    default:
                        for (int i = 0; i < alphabet.Length; i++)
                        {
                            if (alphabet[i] == n.Key) Console.WriteLine("Frequency of
symbol \"{0}\" in text = {1:f3}%", n.Key, n.Value);
                        }
                        break;
                }
            }
        }
    }
}
```

4. Проведіть стиснення кожного вхідного файлу за допомогою 5 різних алгоритмів стиснення

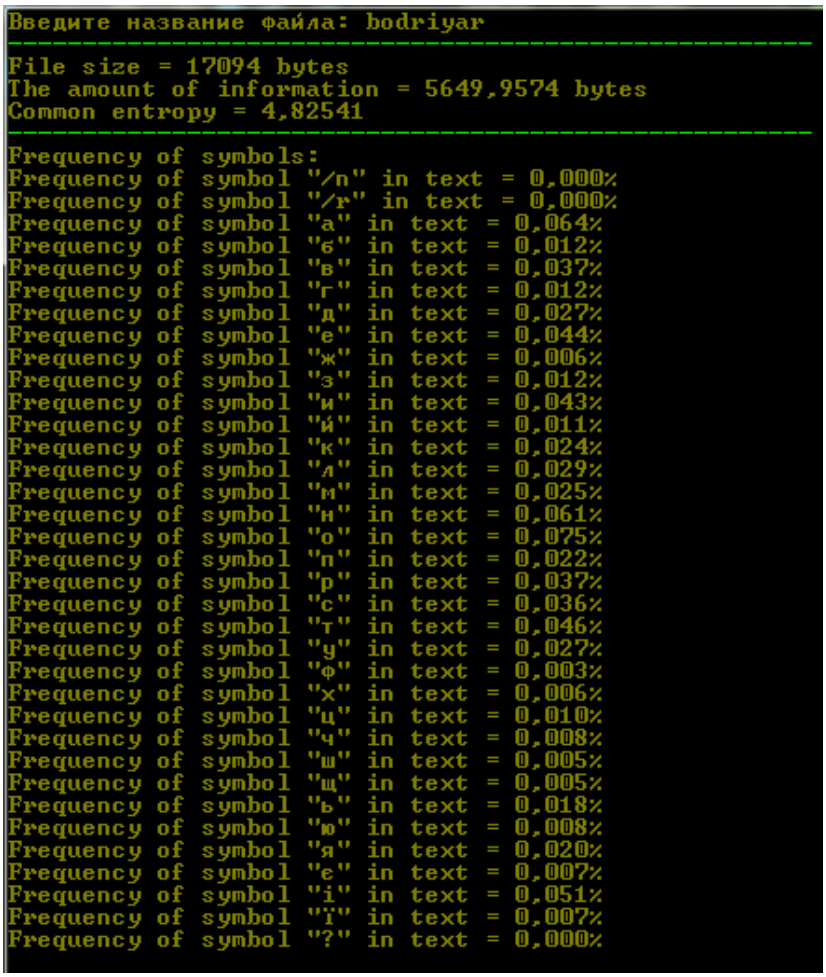
Дані у вигляді таблиці:

| Назва файлу | Розмір файлу | Кількість інформації | 7z | gzip | rar | bzip2 | zip |
|---|---|---|---|---|---|---|---|
| mahno.txt | 7365 | 2405 | 2533 | 2492 | 2614 | 2041 | 2616 |
| bodriyar.txt | 17094 | 5650 | 5335 | 5413 | 5694 | 4435 | 5540 |
| kravchuk.txt | 20634 | 7046 | 6091 | 6204 | 6501 | 5112 | 6331 |

Алфавіт першого файлу:




Алфавіт другого файлу:

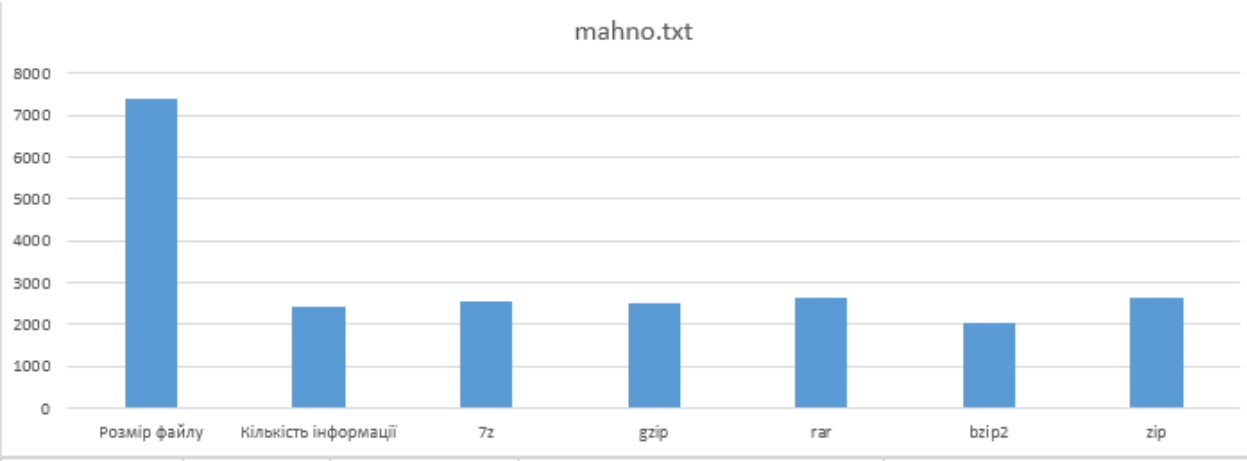



Алфавіт третього файлу:

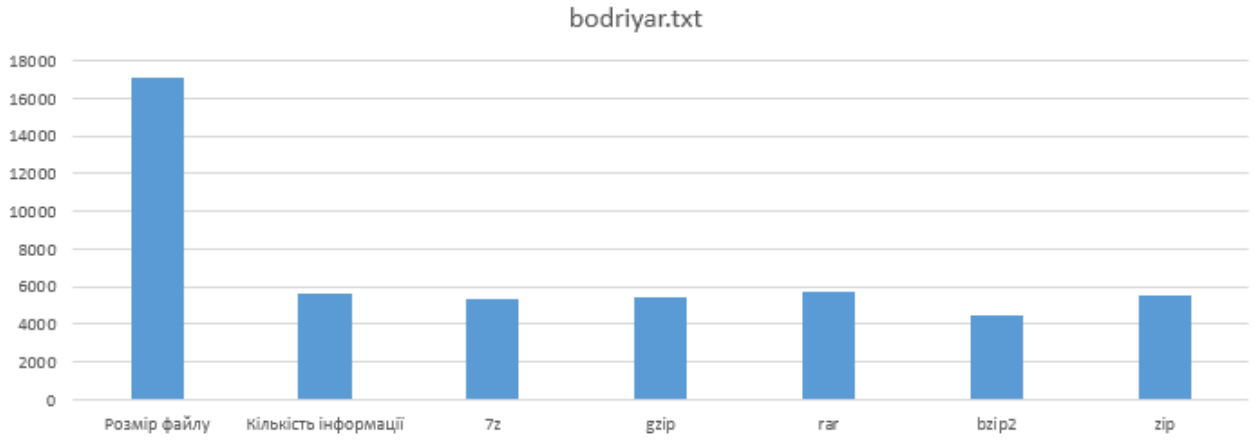
```
Введите название файла: kravchuk
------------------------------------------------------
File size = 20634 bytes
The amount of information = 7046,0892 bytes
Common entropy = 4,88506
------------------------------------------------------
Frequency of symbols:
Frequency of symbol "/n" in text = 0,007%
Frequency of symbol "/r" in text = 0,007%
Frequency of symbol "а" in text = 0,059%
Frequency of symbol "б" in text = 0,010%
Frequency of symbol "в" in text = 0,037%
Frequency of symbol "г" in text = 0,007%
Frequency of symbol "д" in text = 0,024%
Frequency of symbol "е" in text = 0,042%
Frequency of symbol "ж" in text = 0,006%
Frequency of symbol "з" in text = 0,017%
Frequency of symbol "и" in text = 0,052%
Frequency of symbol "й" in text = 0,007%
Frequency of symbol "к" in text = 0,029%
Frequency of symbol "л" in text = 0,022%
Frequency of symbol "м" in text = 0,029%
Frequency of symbol "н" in text = 0,053%
Frequency of symbol "о" in text = 0,074%
Frequency of symbol "п" in text = 0,023%
Frequency of symbol "р" in text = 0,035%
Frequency of symbol "с" in text = 0,033%
Frequency of symbol "т" in text = 0,038%
Frequency of symbol "у" in text = 0,023%
Frequency of symbol "ф" in text = 0,001%
Frequency of symbol "х" in text = 0,006%
Frequency of symbol "ц" in text = 0,007%
Frequency of symbol "ч" in text = 0,008%
Frequency of symbol "ш" in text = 0,006%
Frequency of symbol "щ" in text = 0,008%
Frequency of symbol "ь" in text = 0,014%
Frequency of symbol "ю" in text = 0,006%
Frequency of symbol "я" in text = 0,017%
Frequency of symbol "e" in text = 0,005%
Frequency of symbol "i" in text = 0,046%
Frequency of symbol "ї" in text = 0,008%
Frequency of symbol "?" in text = 0,000%
```

Графіки:

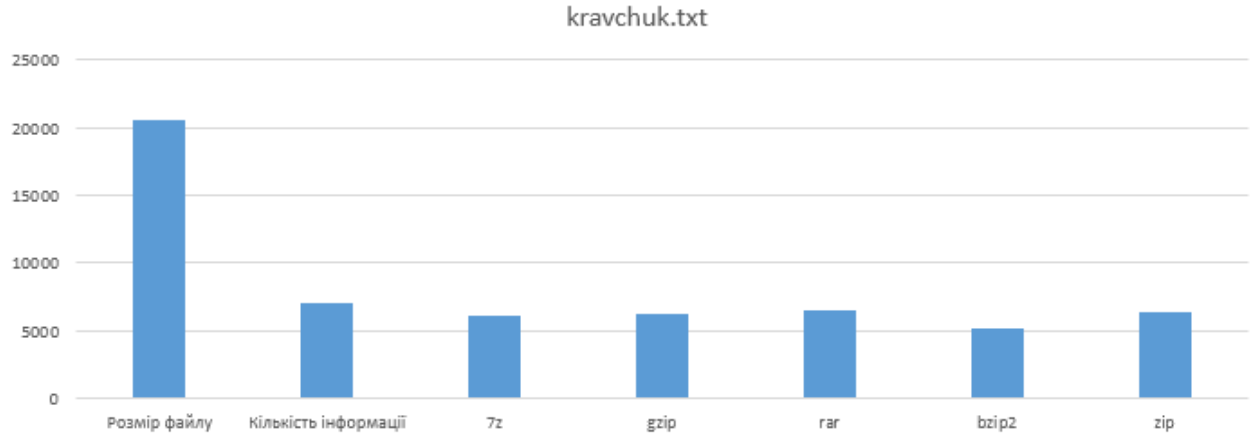
Перший файл:




Другий файл:




Третій файл:




Висновок

У всіх випадках найкращий результат показав алгоритм *bzip2*. На другому місці алгоритм *gzip*. Алгоритми *7z*, *rar* та *zip* продемонстрували найгірший результат; об’єм файлів, закодованих ними перевищував кількість інформації порівняно з *bz2* і *gz*.

II. Дослідження способів кодування інформації на прикладі Base64

1. Ознайомтесь зі стандартом RFC4648

2. Для практичного засвоєння методу кодування, створіть програму, що кодує довільний файл в Base64

Лістинг програми:

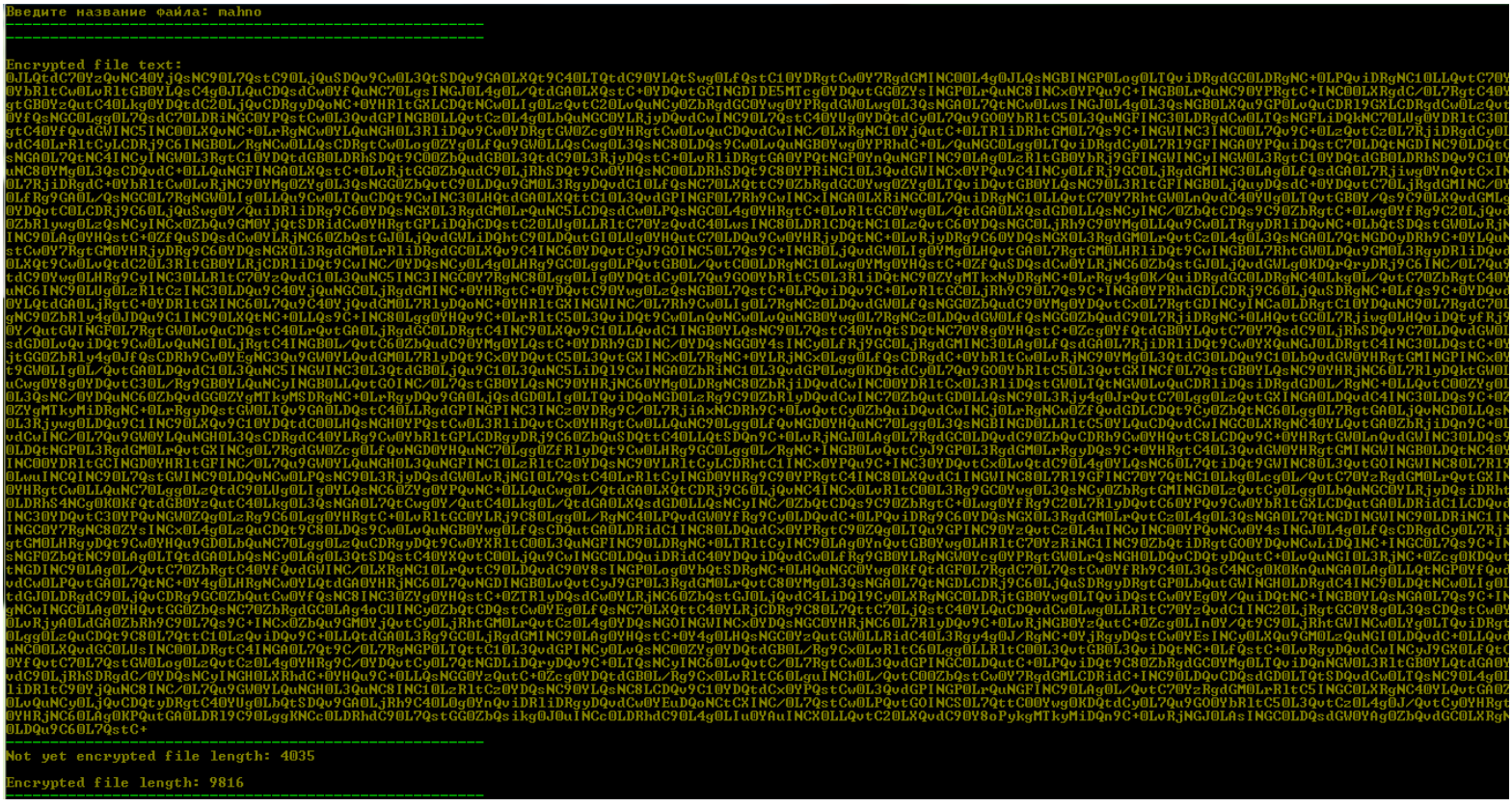
```
using System;
using System.IO;
using System.Security.Cryptography;
using System.Text;
using System.Collections.Generic;
using System.Linq;

namespace lab1._2

{
    class Program
    {
```

```
        static void Main(string[] args)
        {

            bool err = true;
            var str = "";
            try
            {
                do
                {
                    string path = Input();
                    if (File.Exists(path))
                    {
                        str = File.ReadAllText(path);
                        byte[] data = Encoding.Default.GetBytes(str);
                        char[] value = Base64Encoding(data);
                        Output(str, value);
                        err = false;
                    }
                    else
                    {
                        Console.ForegroundColor = ConsoleColor.DarkYellow;
                        Console.WriteLine("Something went wrong, try again");
                        Console.ForegroundColor = ConsoleColor.Green;
                        Console.WriteLine("-----------------------------------------------
--------");
                    }
                } while (err == true);
            }

            catch (Exception e)
            {
                Console.WriteLine(e.Message);
            }
            Console.ReadKey();
        }

        public static void Output(string str, char[] value)
        {
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine("-----------------------------------------------------");
            Console.ForegroundColor = ConsoleColor.DarkYellow;
            Console.WriteLine("\nEncrypted file text: ");
            Console.WriteLine(value);
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine("-----------------------------------------------------");
            Console.ForegroundColor = ConsoleColor.DarkYellow;
            Console.Write("Not yet encrypted file length: ");
            Console.WriteLine(str.Length);
            Console.Write("\nEncrypted file length: ");
            Console.WriteLine(value.Length);
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine("-----------------------------------------------------");

        }

        public static string Input()
        {
            string nameF;
            string path = @"C:\Users\artem\Desktop\CompSys\lab1\";
            Console.ForegroundColor = ConsoleColor.DarkYellow;
            Console.Write("Введите название файла: ");
            nameF = Console.ReadLine();
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine("-----------------------------------------------------");
            path += nameF + ".txt";

            return path;
        }
```

```
private static char SixBitToChar(byte b)
{
    char[] lookupTable = new char[64] {
    'A','B','C','D','E','F','G','H','I','J','K','L','M',
    'N','O','P','Q','R','S','T','U','V','W','X','Y','Z',
    'a','b','c','d','e','f','g','h','i','j','k','l','m',
    'n','o','p','q','r','s','t','u','v','w','x','y','z',
    '0','1','2','3','4','5','6','7','8','9','+','/'
    };
    if ((b >= 0) && (b <= 63)) return lookupTable[(int)b];
    else return ' ';
}

public static char[] Base64Encoding(byte[] data)
{
    int length = data.Length;
    int length2, blockCount, paddingCount;

    if ((length % 3) == 0)
    {
        paddingCount = 0;
        blockCount = length / 3;
    }
    else
    {
        paddingCount = 3 - (length % 3);
        blockCount = (length + paddingCount) / 3;
    }

    length2 = length + paddingCount;
    byte[] source2 = new byte[length2];

    for (int x = 0; x < length2; x++)
    {
        if (x < length) source2[x] = data[x];
        else source2[x] = 0;
    }

    byte b1, b2, b3;
    byte t, t1, t2, t3, t4;
    byte[] temp = new byte[blockCount * 4];
    char[] result = new char[blockCount * 4];

    for (int x = 0; x < blockCount; x++)
    {
        b1 = source2[x * 3];
        b2 = source2[x * 3 + 1];
        b3 = source2[x * 3 + 2];

        t = (byte)((b1 & 3) << 4);
        t1 = (byte)((b1 & 252) >> 2);
        t2 = (byte)((b2 & 240) >> 4);
        t2 += t;

        t = (byte)((b2 & 15) << 2);
        t3 = (byte)((b3 & 192) >> 6);
        t3 += t;
        t4 = (byte)(b3 & 63);

        temp[x * 4] = t1;
        temp[x * 4 + 1] = t2;
        temp[x * 4 + 2] = t3;
        temp[x * 4 + 3] = t4;
    }

    for (int x = 0; x < blockCount * 4; x++)
    {
        result[x] = SixBitToChar(temp[x]);
    }
```

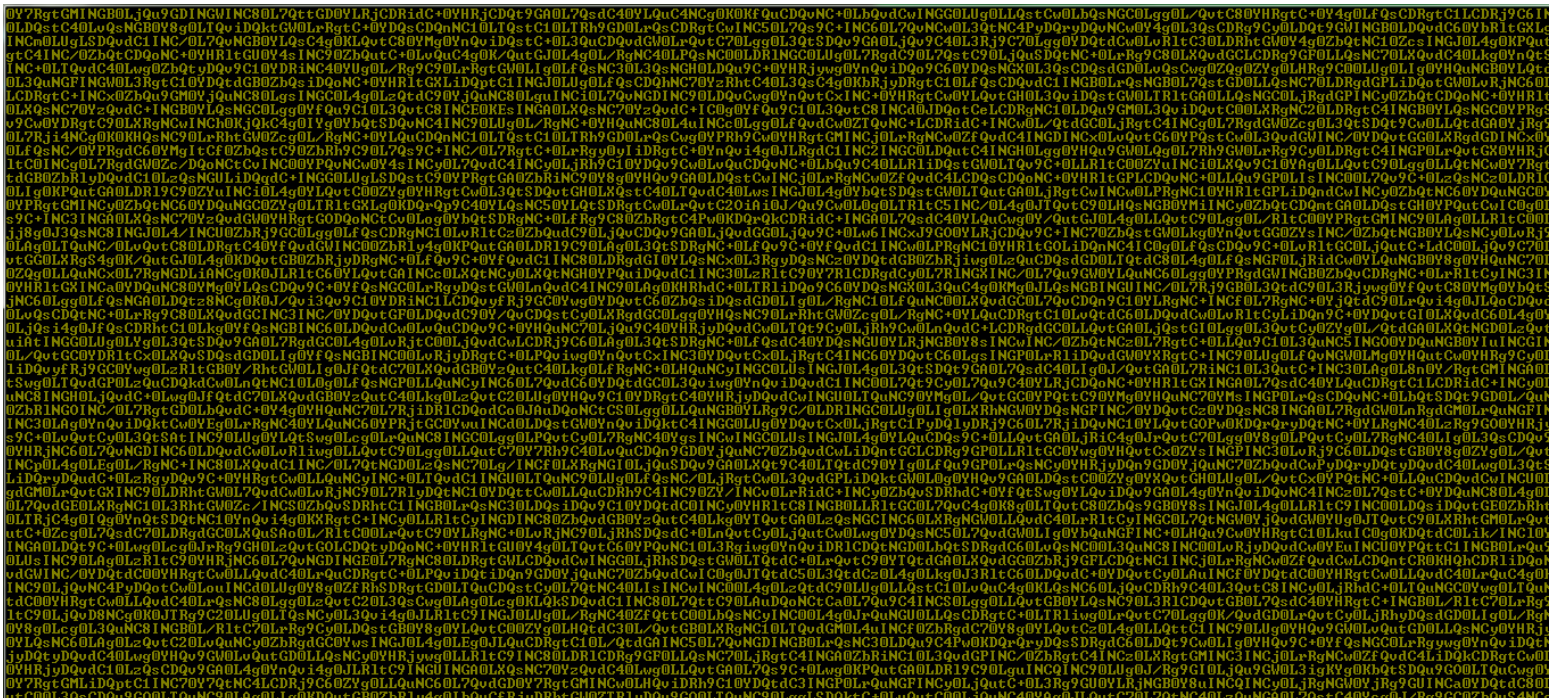
```
            switch (paddingCount)
            {
                case 0:
                    break;
                case 1:
                    result[blockCount * 4 - 1] = '=';
                    break;
                case 2:
                    result[blockCount * 4 - 1] = '=';
                    result[blockCount * 4 - 2] = '=';
                    break;
            }
            return result;
        }

    }
}
```

Перший файл:




Другий файл:

Третій файл:



3. Закодуйте в Base64 обрані вами текстові файли

Кодування виконано у минулому пункті.

| | Кількість інформації(base64)/Кількість інформації(file) |
|---|---|
| mahno.txt | 1,333469 |
| bodriyar.txt | 1,333391 |
| kravchuk.txt | 1,(3) |

Згідно таблиці в середньому кількість інформації у файлі, що був перекодований у base64 більша у 1.33 рази ніж кількість інформації в файлі.

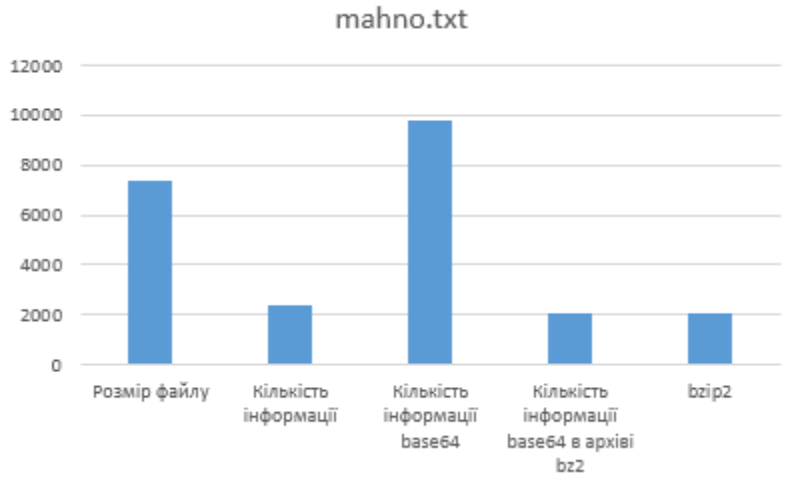
4. Закодуйте в Base64 стиснені кращим з алгоритмів текстові файли

Я обрав розширення .bz2, так як такі архіви мали найменший розмір серед усіх інших у моєму випадку.

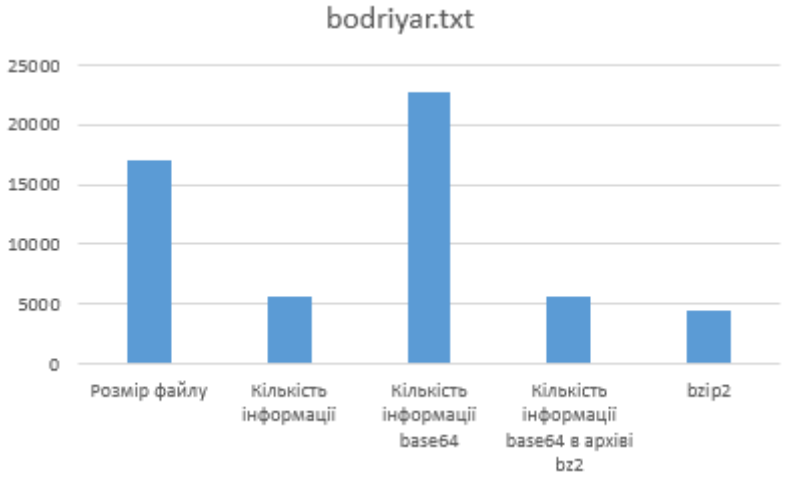
| | Розмір файлу | Кількість інформації | Кількість інформації base64 | Кількість інформації base64 в архіві bz2 | bzip2 |
|---|---|---|---|---|---|
| mahno.txt | 5280 | 2405 | 9803 | 2041 | 2041 |
| bodriyar.txt | 2440 | 5650 | 22787 | 5681 | 4435 |
| kravchuk.txt | 3218 | 7046 | 27399 | 6611 | 5112 |

Графіки:

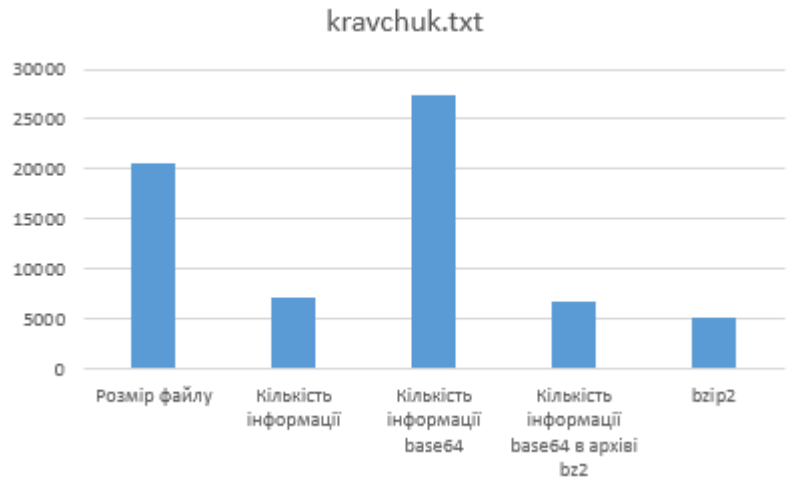
Перший файл:




Другий файл:




Третій файл:




**Висновок:** підчас виконання лабораторної роботи було досліджено частоти появи символу, ентропію нерівномірного алфавіту, як ці параметри впливають на кількість інформації в тексті. На практиці було виявлено, що кращим алгоритмом стискання в рамках лабораторної роботи був алгоритм *.bz2*. Також було реалізовано алгоритм кодування Base64, перевірена правильність його роботи.