**Київський національний університет імені Тараса Шевченка**

**факультет радіофізики, електроніки та комп’ютерних систем**

Лабораторна робота № 1

**Тема:** « Дослідження кількості інформації при різних варіантах кодування »

Роботу виконав

студент 3 курсу

КІ - СА

Гулівець Владислав Андрійович

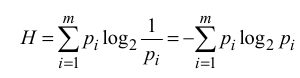
**Київ 2020**

***Мета:*** *Дослідити імовірнісні параметри української мови для оцінки кількості інформації текстів. Дослідити вплив різних методів кодування інформації на її кількість.*

***Теоретичні відомості***

***Відносна частота появи символу*** *- імовірність появи певного символу в певному місці тексту - відношення числа появи символу в тексті до загальної кількості символів.*

***Середня ентропія нерівноймовірного алфавіту****:*

**

*де m - кількість символів алфавіту, p - імовірність появи символу*

*Ентропія вимірюється в* ***БІТАХ*** *(як представлення кількості можливих варіантів).*

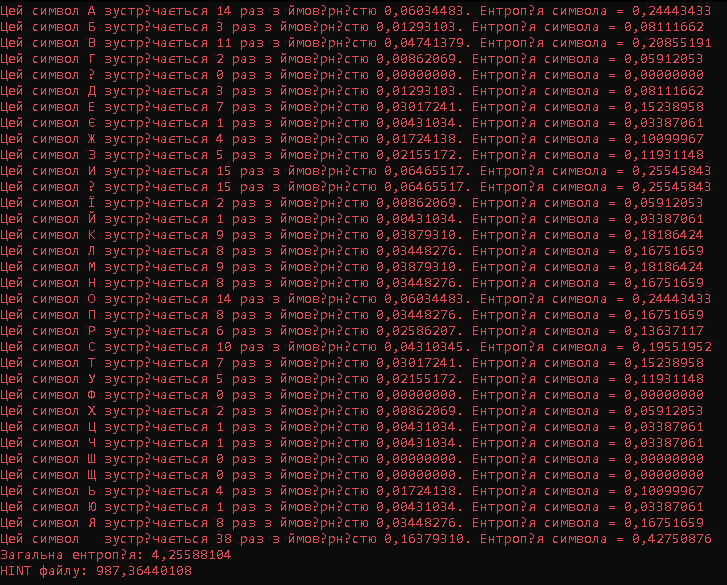
***Кількість інформації в тексті*** *- середня ентропія вихідного алфавіту помножена на кількість символів тексту. (****HINT****: результат обрахунку для порівняння значення з розміром файлів треба перевести з бітів в байти)*

**Хід виконання роботи:**

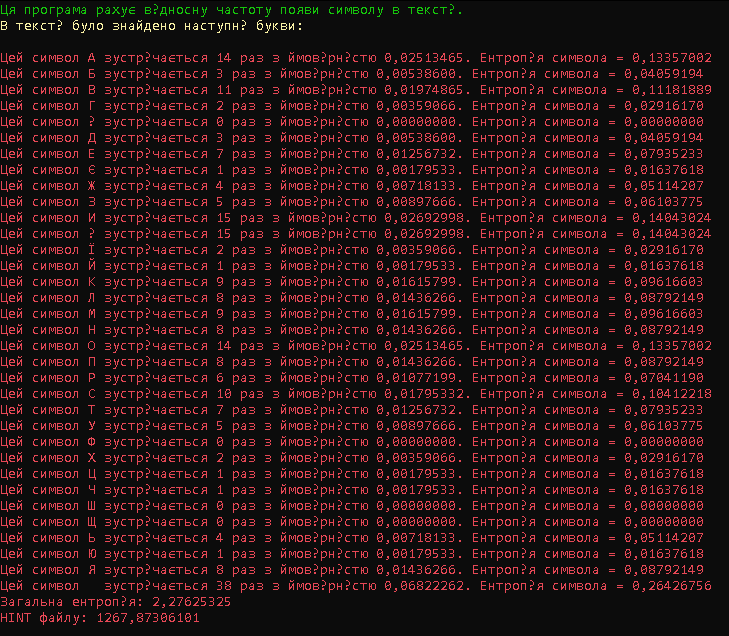
Дослідження кількості інформації в тексті

1. Оберіть 3 текстових файла різного тематичного та лінгвістичного спрямування (наприклад, вірш Тараса Шевченка “Мені тринадцятий минало”, “Казка про рєпку” Леся Подерв'янського та специфікацію інерфейсу PCI)
   * + text1.txt – куплет пісні Скрябіна.
     + text2.txt – історія України за 1917 рік.
     + text3.txt – рецензія на фільм «Захар Беркут».
2. Створіть програму (будь-якою зручною для вас мовою), яка в якості вхідних даних приймає текстовий файл, та аналізуючи його вміст:
   * + **обраховує частоти (імовірності) появи символів в тексті**
     + **обраховує середню ентропію алфавіту для даного тексту**
     + **виходячи з ентропії визначає кількість інформації та порівнює її з розмірами файлів**
     + **виводить на екран значення частот, ентропії та кількості інформації 4. Проведіть стиснення кожного вхідного файлу за допомогою 5 різних алгоритмів стиснення (zip, rar, gzip, bzip2, xz, або будь-які інші на ваш вибір, можна використовувати готові програмні засоби для стиснення).**

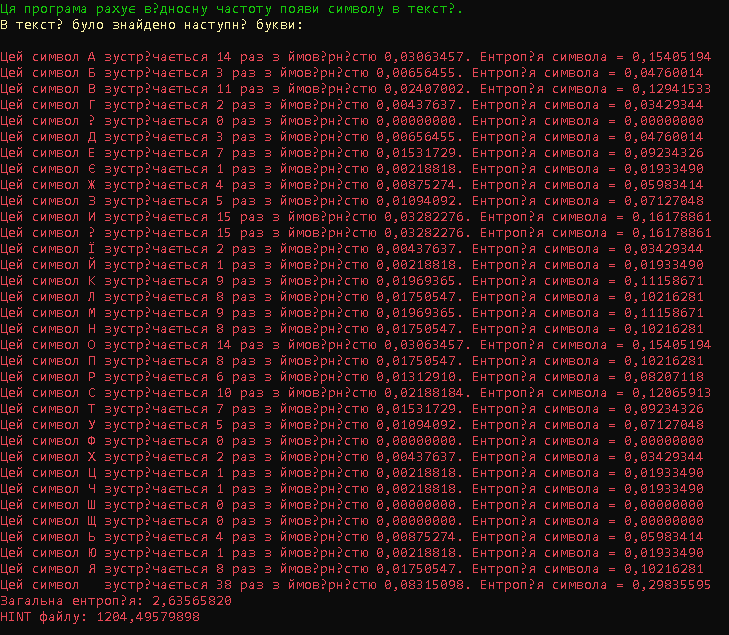
Analyze of ***text1.txt***



Analyze of ***text2.txt***



Analyze of ***text3.txt***



Entropy:

text1.txt – 4.25588104

text2.txt – 2.27625325

text3.txt – 2.63565820

Size of file (bytes):

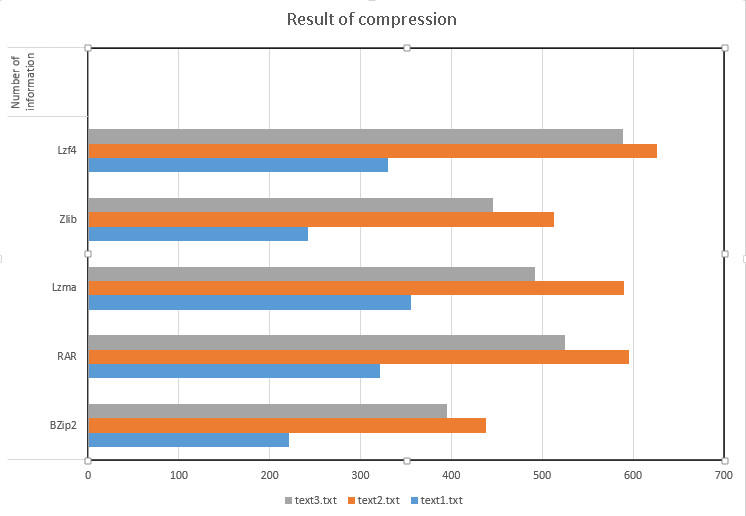
text1.txt – 987bit/8 = 123.375bytes

text2.txt – 1277bit/8 = 159.625bytes

text3.txt – 1204bit/8 = 150.5bytes

Result of compression

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Назва файлу | BZip2 | RAR | Lzma | Zlib | Lzf4 | Number of information |
| text1.txt | 221 | 321 | 356 | 242 | 330 | 123.375 |
| text2.txt | 438 | 595 | 590 | 513 | 626 | 159.625 |
| text3.txt | 395 | 525 | 492 | 446 | 589 | 150.5 |



# Дослідження способів кодування інформації на прикладі Base64

1. Ознайомтесь зі стандартом [RFC4648](https://tools.ietf.org/html/rfc4648)
2. Для практичного засвоєння методу кодування, створіть програму, що кодує довільний файл в Base64 (шляхом реалізації алгоритму вручну, а не виклику бібліотечної функції).   
   **Перевірте коректність роботи програми, порівнявши результат з існуючими програмними засобами (наприклад, openssl enc -base64)**
3. Закодуйте в Base64 обрані вами текстові файли

**Обрахуйте кількість інформації в base64-закодованому варіанті файлу**

**Порівняйте отримане значення з кількістю інформації вихідного файлу**

**Зробіть висновки з отриманого результату**

1. Закодуйте в Base64 стиснені кращим з алгоритмів текстові файли

**Обрахуйте кількість інформації в base64-закодованому варіанті стисненого файлу**

**Порівняйте отримане значення з кількістю інформації вихідного файлу та base64-закодованого файлу**

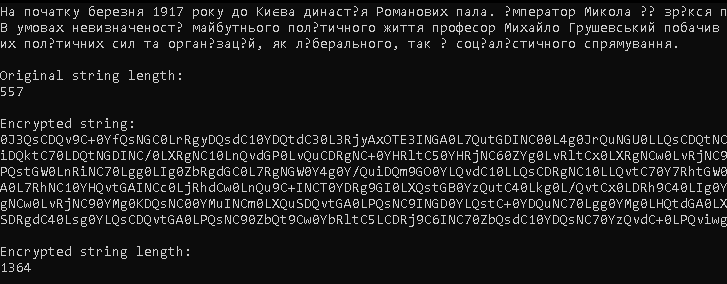
**Зробіть висновки з отриманого результату**

# Хід виконання роботи:

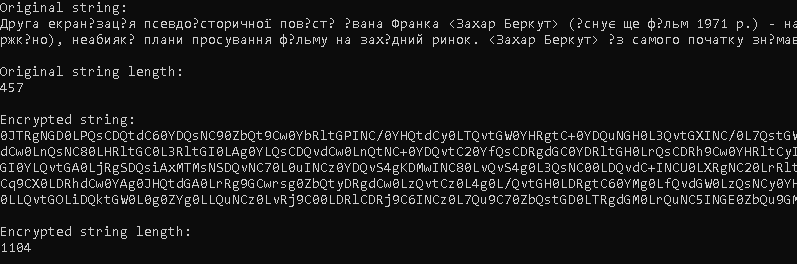
Base64 compression of ***text1.txt***



Base64 compression of ***text2.txt***



Base64 compression of ***text3.txt***



Size of file (bytes):  
text1encrypted.txt = 1267bit/8 = 158.375 bytes

text2encrypted.txt = 1554bit/8 = 194.25 bytes

text3encrypted.txt = 1486bit/8 = 185.75 bytes

Entropy:

text1.txt – 4.25588104

text2.txt – 2.27625325

text3.txt – 2.63565820

Size of file (bytes):

text1.txt – 987bit/8 = 123.375bytes

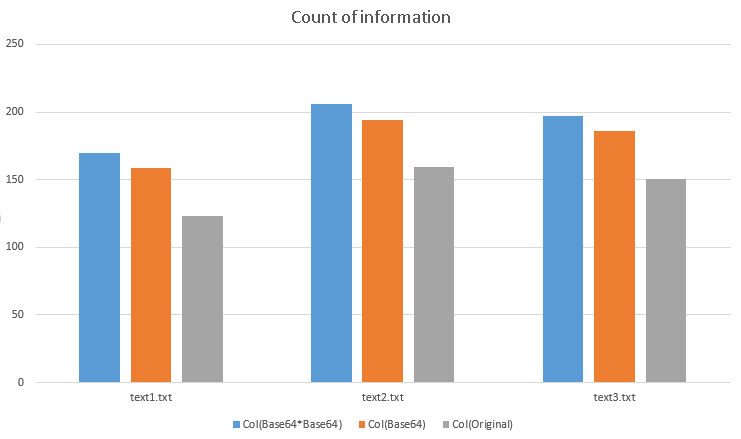
text2.txt – 1277bit/8 = 159.625bytes

text3.txt – 1204bit/8 = 150.5bytes

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Назва файлу | | | Entropy | | Length | | Count of information | | |
| text1encrypted.txt | | | 2,2793 | | 556 | | 158.375 | | |
| text2encrypted.txt | | | 1,1398 | | 1364 | | 194.25 | | |
| text3encrypted.txt | | | 1,3469 | | 1104 | | 185.75 | | |
| Назва файлу | BZip2 | RAR | | Lzma | | Zlib | | Lzf4 |
| text1.txt | 221 | 321 | | 356 | | 242 | | 330 |
| text2.txt | 438 | 595 | | 590 | | 513 | | 603 |
| text3.txt | 395 | 525 | | 492 | | 446 | | 541 |

|  |  |  |  |
| --- | --- | --- | --- |
| Назва файлу | BZip(BASE64) | Count of information | Number of information |
| text1encrypted.txt | 335 | 158.375 | 123.375 |
| text2encrypted.txt | 630 | 194.25 | 159.625 |
| text3encrypted.txt | 577 | 185.75 | 150.5 |

|  |  |  |
| --- | --- | --- |
| Назва файлу | CoI(BASE64)2 | Number of information |
| text1encrypted.txt | 170 | 123.375 |
| text2encrypted.txt | 205.875 | 159.625 |
| text3encrypted.txt | 197.375 | 150.5 |



**Висновок:** В цій лабораторній роботі мною була зроблена програма що аналізує текст та обчислює ймовірності появи букви в тексті, ентропію тексту, загальний обсяг інформації. Мною було проаналізований обсяг зайнятого простору на диску файлів з різним типом стиснення. Як результат аналізу можна сказати, що обсяг інформації є значно меншим аніж обсяг який займає стиснутий цей же файл на диску. Було встановлено що найкращим з перевірених алгоритмом стиснення є BZip2. Об'єм його файлів є найближчим до кількості інформації. Також були опрацьовані навички в кодуванні ***Base64***. Як результат можна сказати, що с кожним повторним кодуванням файлу обсяг інформації збільшується. Це зумовлено тим, що сама кількість символів в файлі збільшується, а отже й сам обсяг збільшується також.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

My personal GitHub link: ***https://github.com/gulivec84***