**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-го курсу

КІ-МА

Пословський Максим Васильович

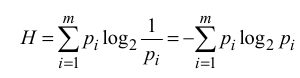
Київ 2020

**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

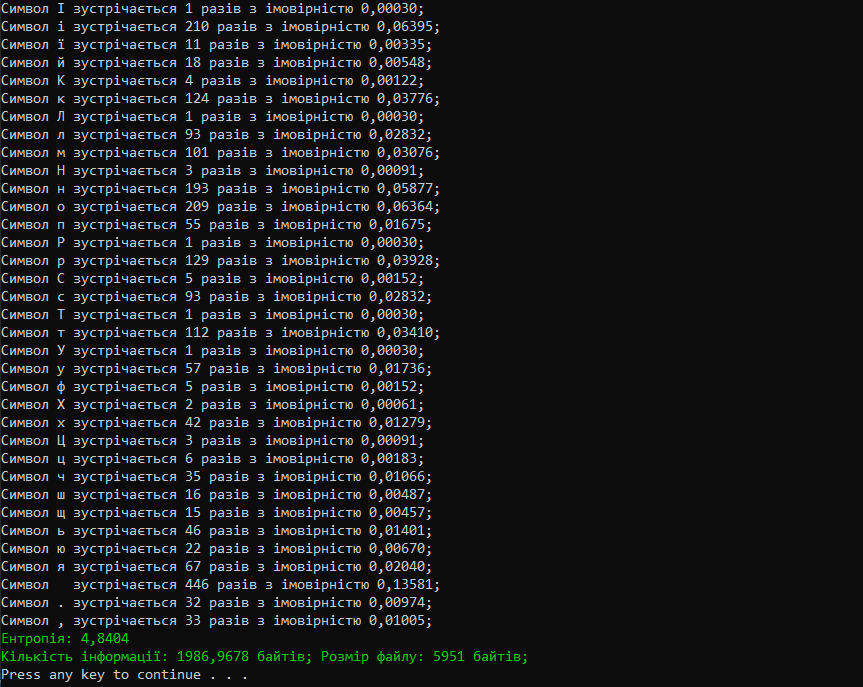
Ентропія вимірюється в **БІТАХ** (як представлення кількості можливих варіантів).

***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)

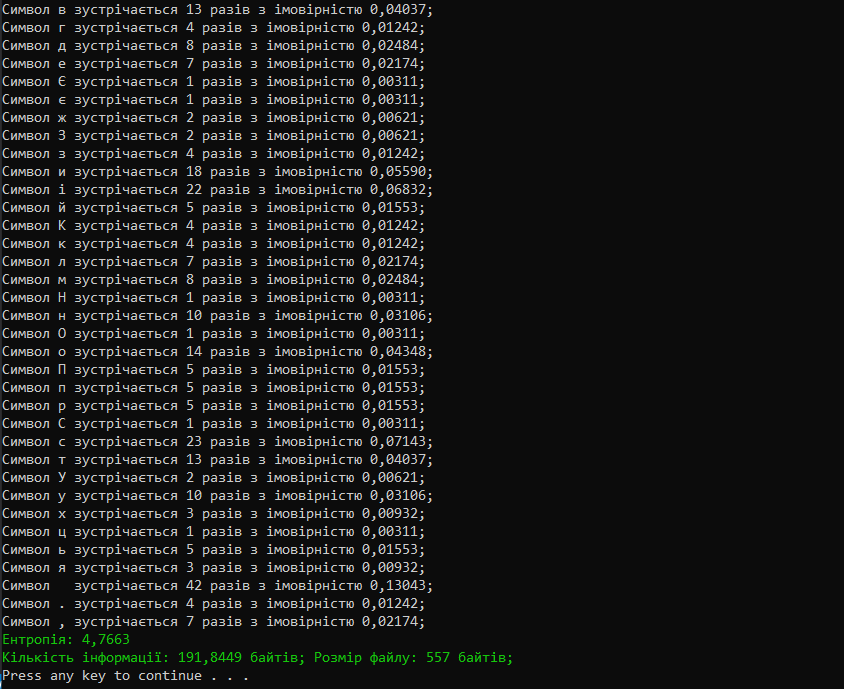
**Хід виконання роботи:**

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* – Стаття про квазари;
   * *Text2.txt* – Пісня про КотаПса;
   * *Text3.txt* – Критика Теорії змови;
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

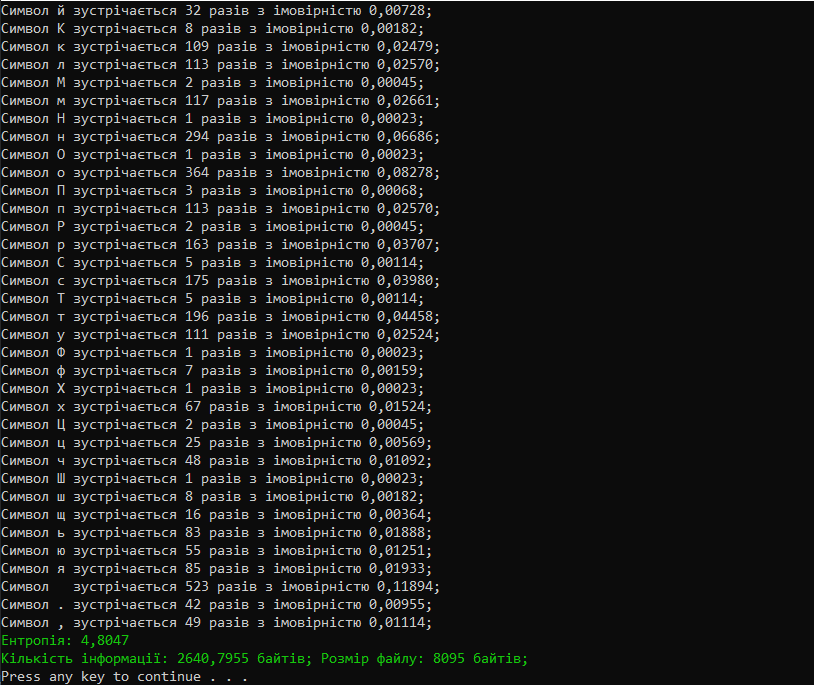
*Аналіз тексту №1*



*Аналіз тексту №2*



*Аналіз тексту №3*



1. 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).

Розміри файлів:

*Text1.txt* – **5951** байтів

*Text2.txt* – **557** байтів

*Text3.txt* – **8095** байтів

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Назва файлу | Zip | RAR | Lzma2 | BZip2 | PPMd | Кількість  Інформації | Ентропія |
| Text1.txt | 2172 | 2112 | 2 048 | 1 837 | 1679 | 1986 | 4.84 |
| Text2.txt | 417 | 363 | 412 | 384 | 344 | 191 | 4.76 |
| Text3.txt | 2850 | 2796 | 2709 | 2332 | 2180 | 2640 | 4.80 |
|  |  |  |  |  |  |  |  |

1. Порівняйте результуючі обсяги архівів з обчисленою кількістю інформації та **наведіть у звіті висновки** щодо кореляції цих величин для обраних вами файлів (яка відмінність, що вийшло більше і чому)

# 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

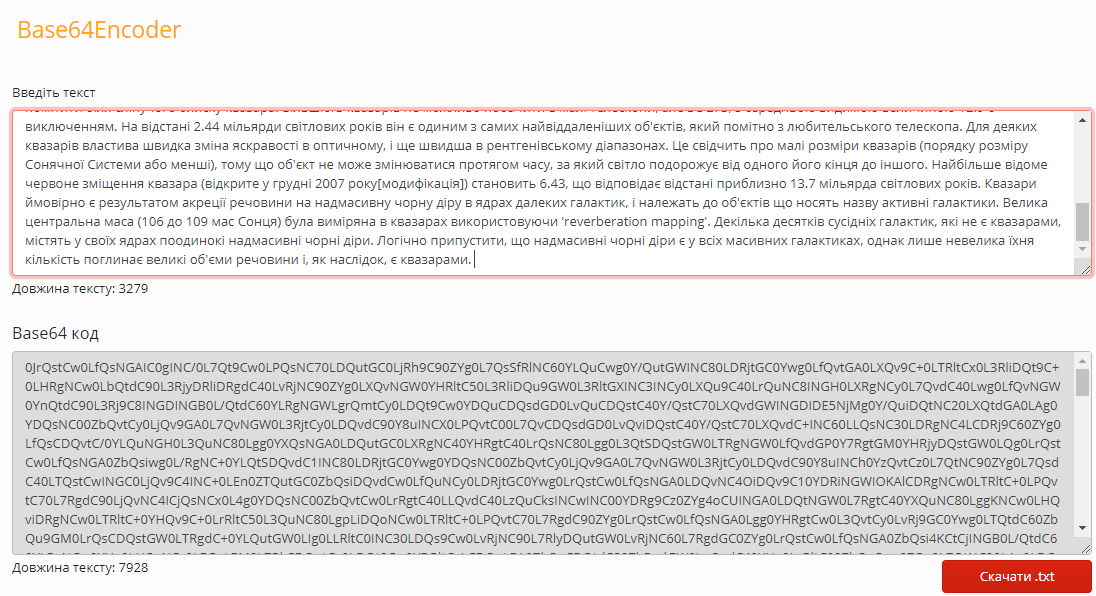
**Обрахуйте кількість інформації в base64-закодованому варіанті стисненого файлу**

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

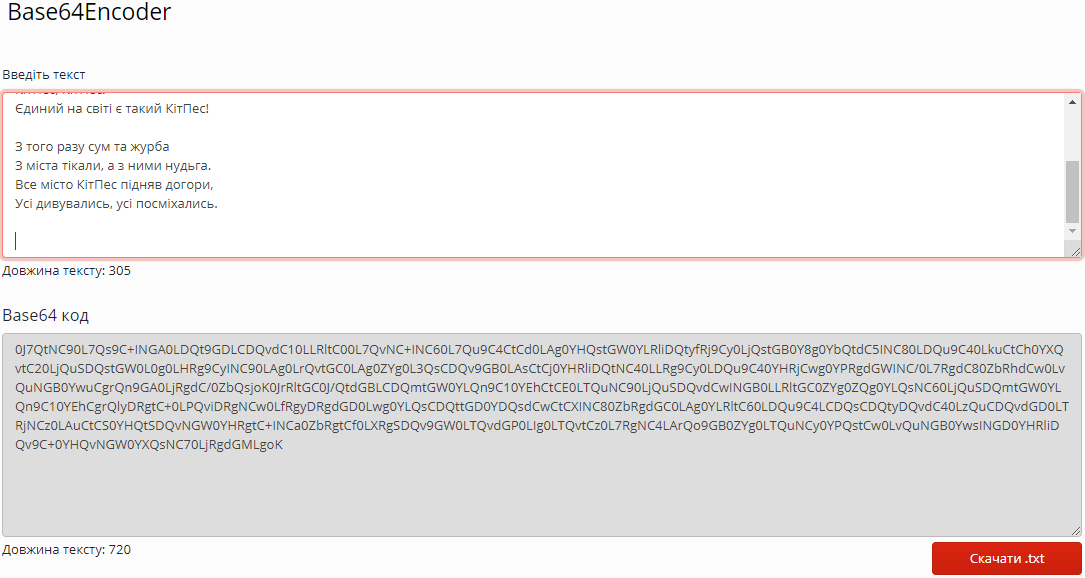
**Зробіть висновки з отриманого результату**

# Хід виконання роботи:

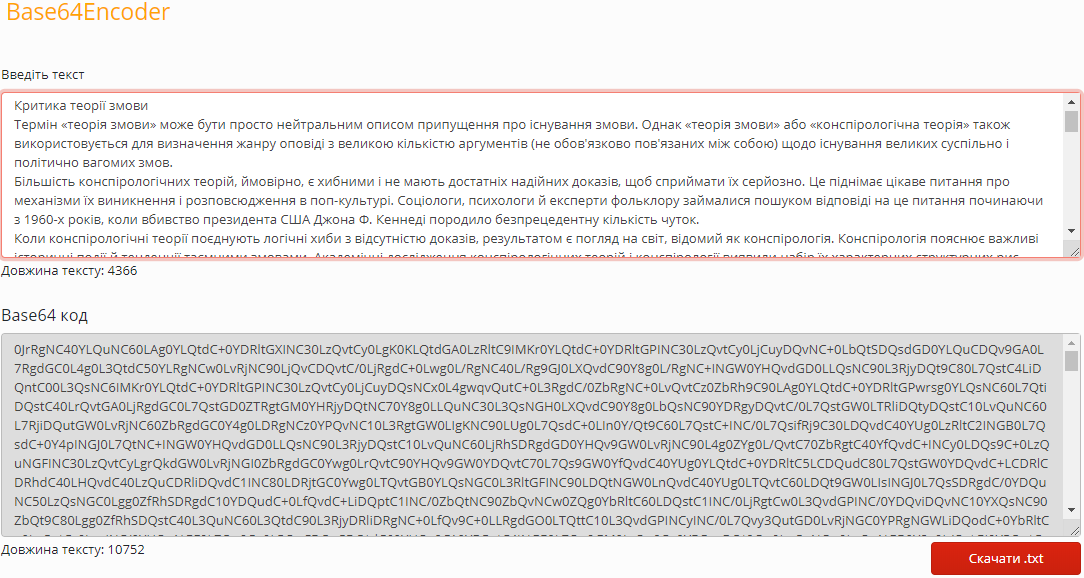
*BASE64-кодування Text1.txt*



*BASE64-кодування Text2.txt*



*BASE64-кодування Text3.txt*



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Текст | Довжина тексту | Довжина закодованого тексту | Розмір файлу | Розмір закодованого файлу |
| Text1.txt | 3279 | 7928 | 5951 | 7928 |
| Text2.txt | 305 | 720 | 557 | 720 |
| Text3.txt | 4366 | 10752 | 8095 | 10752 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Назва закодованого файлу | Lzma2 | BZip2 | PPMd | Кількість  Інформації | Ентропія |
| Text1\_encrypted.txt | 2 732 | 2 452 | 2240 | 1668 | 5.95 |
| Text2\_encrypted.txt | 552 | 512 | 460 | 319 | 5.52 |
| Text3\_encrypted.txt | 3612 | 3 112 | 2 908 | 2167 | 5.96 |
|  |  |  |  |  |  |

**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 GitHub link: ***https://github.com/zamoox***