**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ**

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет радіофізики, електроніки та комп’ютерних систем

Кафедра комп’ютерної інженерії

**Лабораторна робота №1**

за спеціальністю 123 Комп’ютерна інженерія

з предмету:

**КОМП’ЮТЕРНІ СИСТЕМИ**

Виконав студент 3-го курсу

Бількевич Борис Борисович

Науковий керівник:

кандидат технічних наук

Слюсар Євген Андрійович

**КИЇВ 2020**

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

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

**План:**

1. Дослідження кількості інформації в тексті
2. Дослідження способів кодування інформації на прикладі Base64

Хід 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)

В процесі виконання лабораторно роботи, було обрано 2 статті та 1 вірш для аналізу:

* ["І все на світі треба пережити" Ліна Костенкко](https://www.livelib.ru/quote/1119775-trista-poezij-lina-kostenko)
* [Фішинг](https://yur-gazeta.com/publications/practice/inshe/kiberbezpeka-vrazlivi-momenti.html)
* [Дихлордифенілтрихлоретан](https://www.epochtimes.com.ua/suspilstvo/movchazna-vesna-istoriya-i-tragichni-naslidky-zastosuvannya-ddt-128594)

1. Переконайтесь, що тексти, які ви використовуєте є унікальними і не повторюються у ваших колег! Використовуйте наявні електронні засоби зв’язку та документообігу, щоб уникнути дублювання! Вдруге аналіз того самого тексту не зараховується!

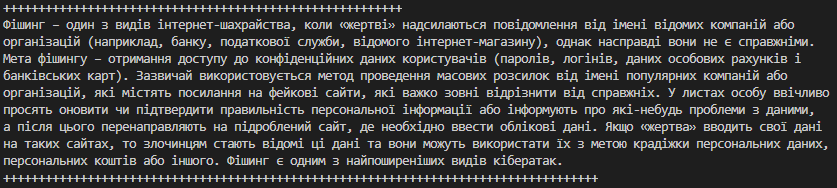
Для уникнення дублювання було використано спільний Excel документ:

* [Комп'ютерні системи](https://docs.google.com/spreadsheets/d/14NT1swSbKaBT9e6BQB3mZcQ5S148B_oJ7cWNkTyq79U/edit#gid=0)

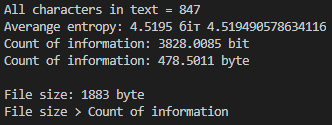
1. Створіть програму (будь-якою зручною для вас мовою), яка в якості вхідних даних приймає текстовий файл, та аналізуючи його вміст:
2. обраховує частоти (імовірності) появи символів в тексті
3. обраховує середню ентропію алфавіту для даного тексту
4. виходячи з ентропії визначає кількість інформації та порівнює її з розмірами файлів
5. виводить на екран значення частот, ентропії та кількості інформації

Дану програму вирішив реалізувати на мові Python:

Стаття про «Фішинг»

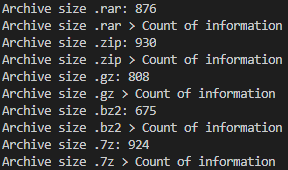


Далі виводжу всю інформацію про даний уривок тексту, який я в процесі дослідив, а саме загальну кількість літер, середню ентропію, а також кількість інформації в бітах та байтах.

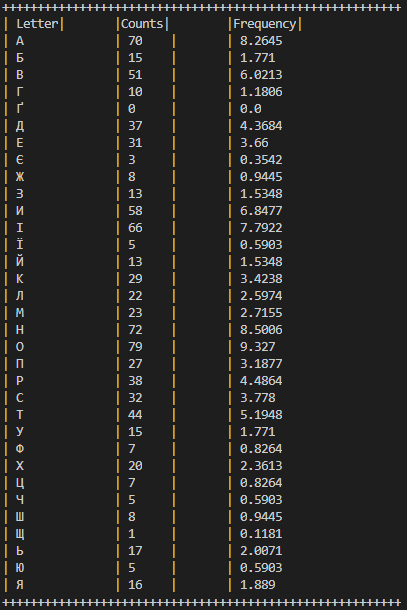


Після чого проводжу перевірку на порівняння розмірів до стиснення та після.

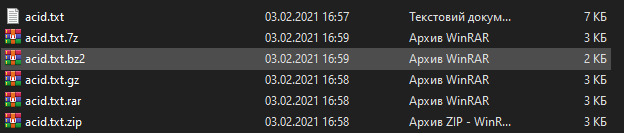
Використав 5 варіантів стиснення (.rar, .zip, .gz, .bz2, .7z), які виконував за допомогою онлайн архіваторів.



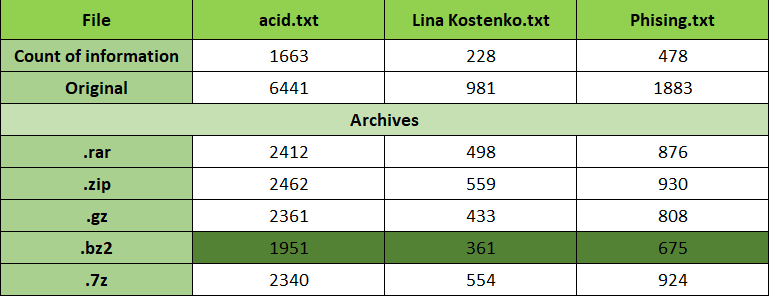
А також виводжу частоту появи кожної з букв:

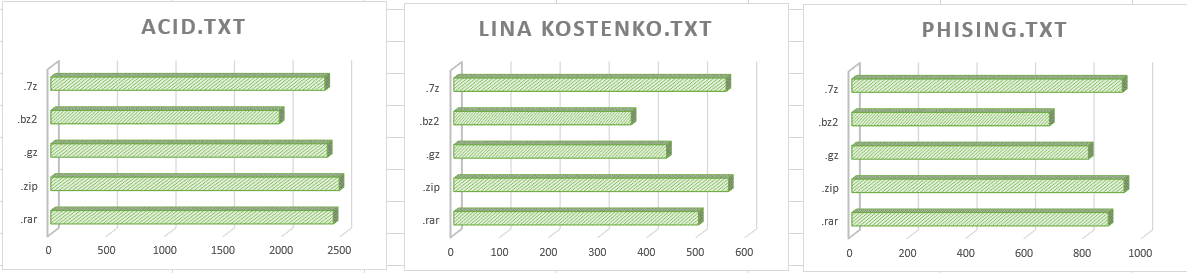


1. Проведіть стиснення кожного вхідного файлу за допомогою 5 різних алгоритмів стиснення (zip, rar, gzip, bzip2, xz, або будь-які інші на ваш вибір, можна використовувати готові програмні засоби для стиснення).



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)





Відповідно до графіків ми можемо побачити, що найкраще справився з стисненням bzip2. А також на основі даних, можна зробити висновок, кількість інформації для всіх стиснених файлів менша, ніж фактичні розміри.

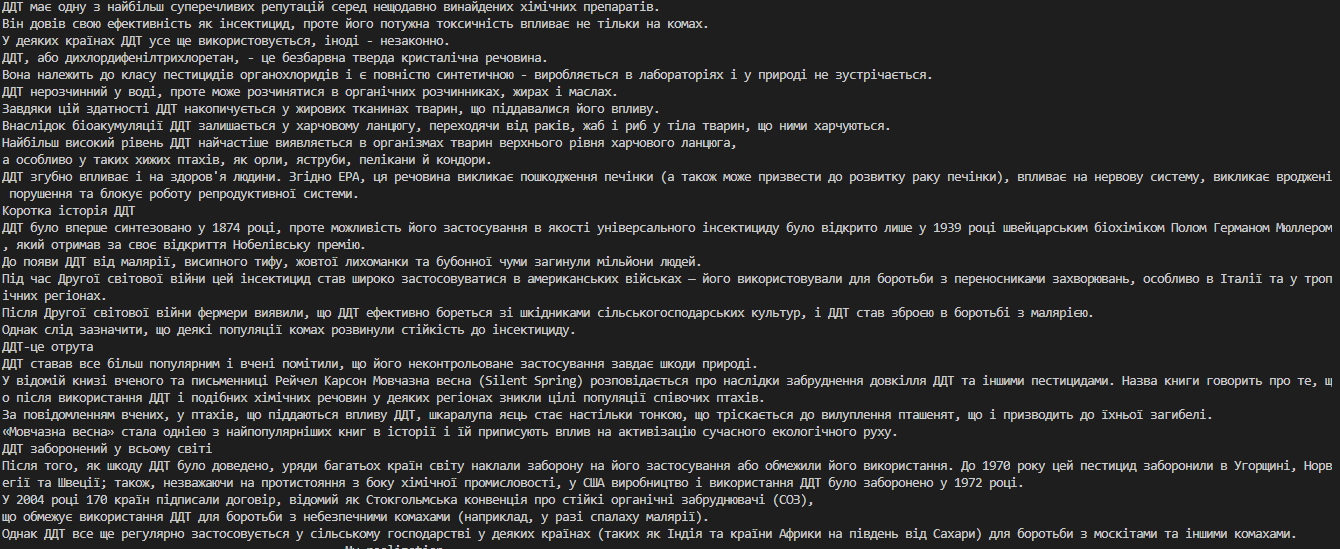
Як висновок, при ідеальному стисненні розмір файлу мав би бути рівним кількості інформації. Але насправді, ми можемо побачити з результатів таблиці у всіх випадках розміри архівованих файлів, дещо більші за кількість інформації. Це відбувається тому, що алгоритми архіваторів налаштовані таким чином, щоб використовувати повторювані частини тексту. Саме тому формула розрахунку кількості інформації, використана для програми, не є досконалою, бо вона не враховує даний уривок тексту.

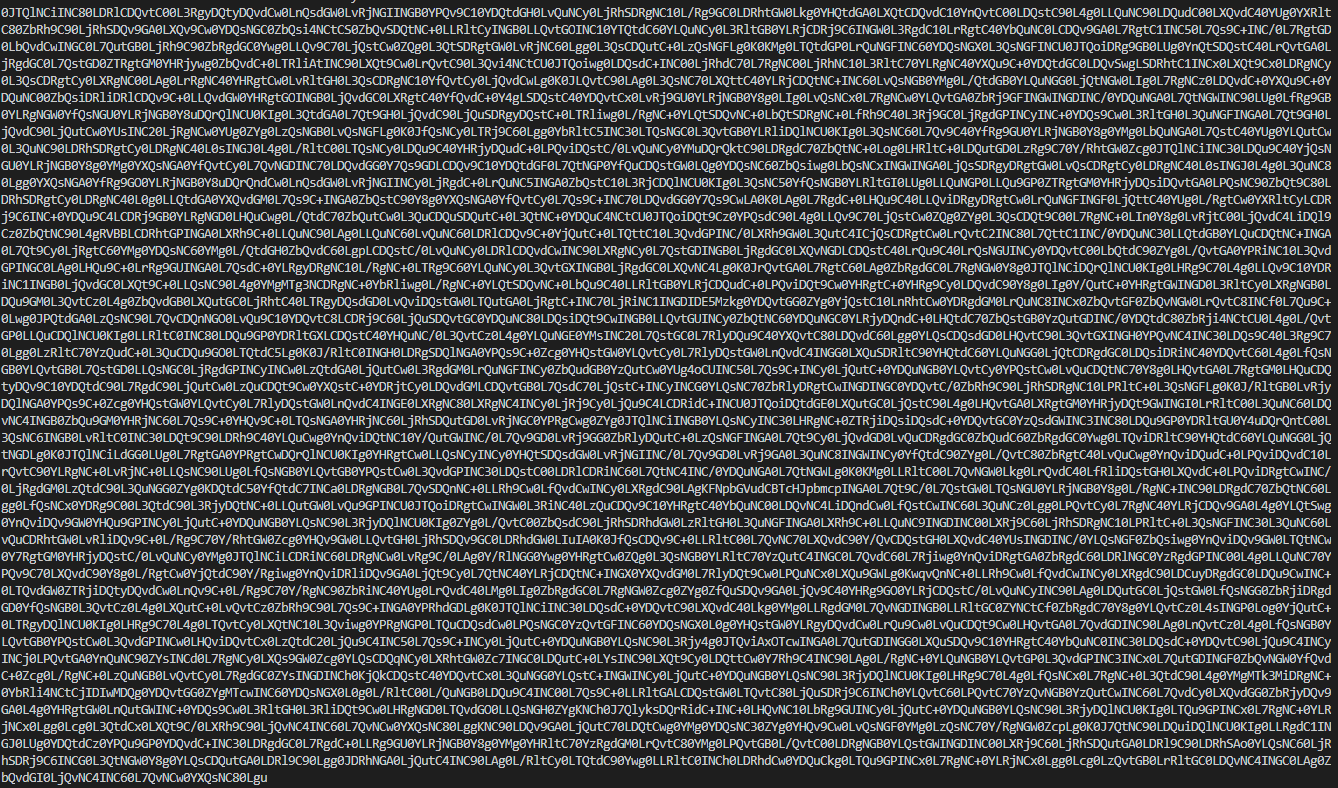
1. Ознайомтесь зі стандартом RFC4648

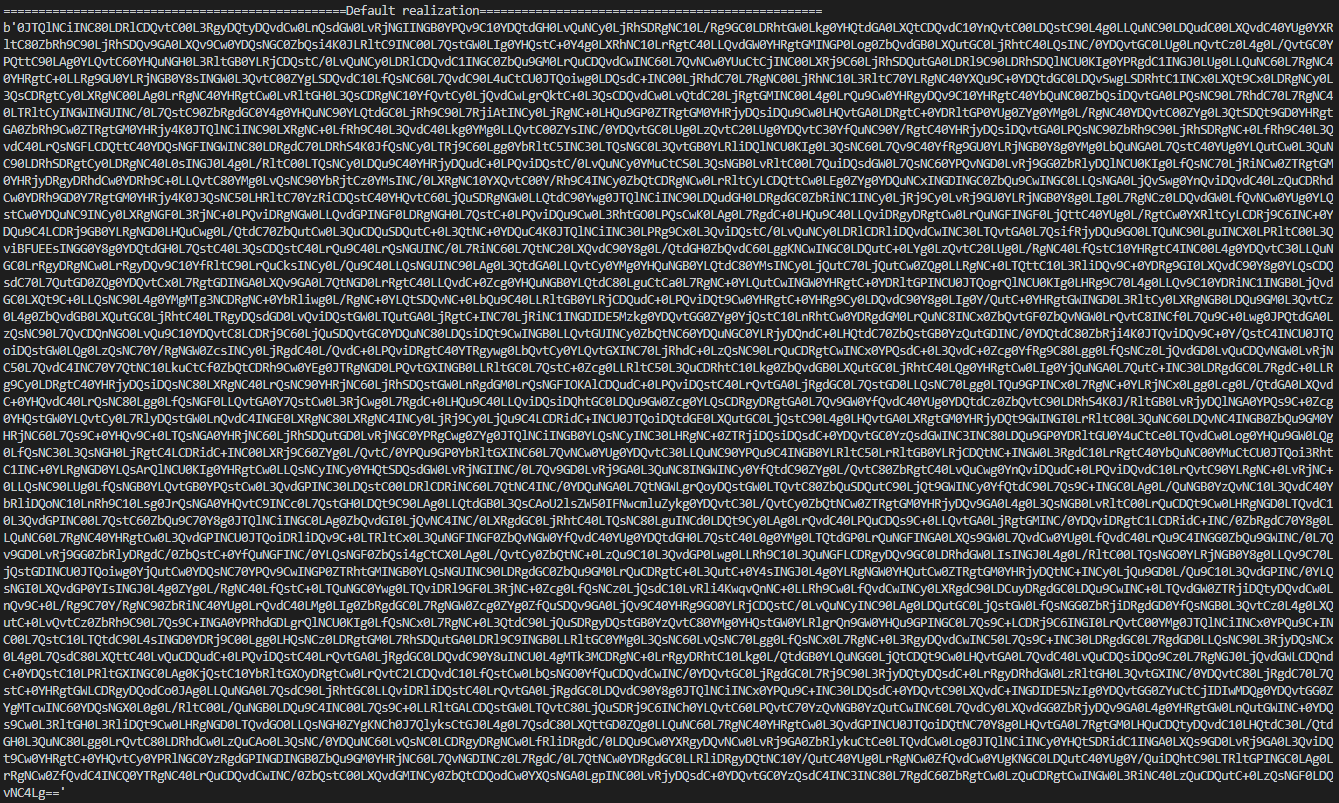
2. Для практичного засвоєння методу кодування, створіть програму, що кодує довільний файл в Base64 (шляхом реалізації алгоритму вручну, а не виклику бібліотечної функції)

1. перевірте коректність роботи програми, порівнявши результат з існуючими програмними засобами (наприклад, openssl enc -base64)

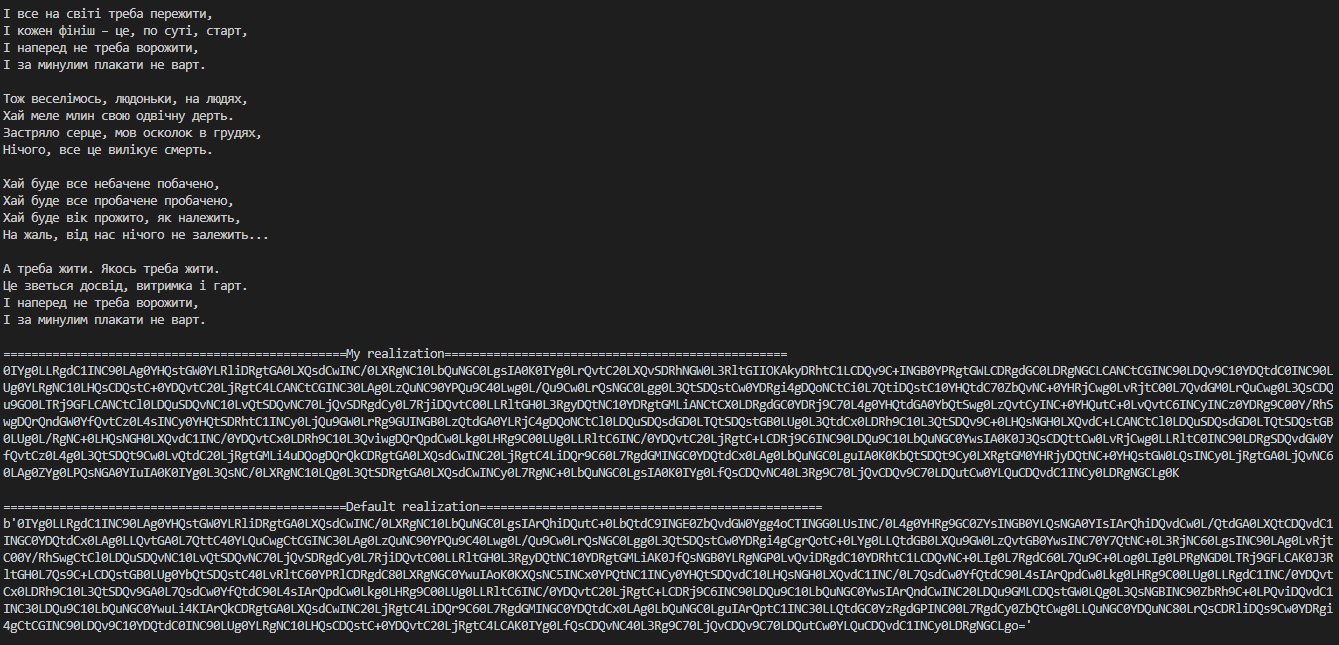
* **Acid.txt**

****

****

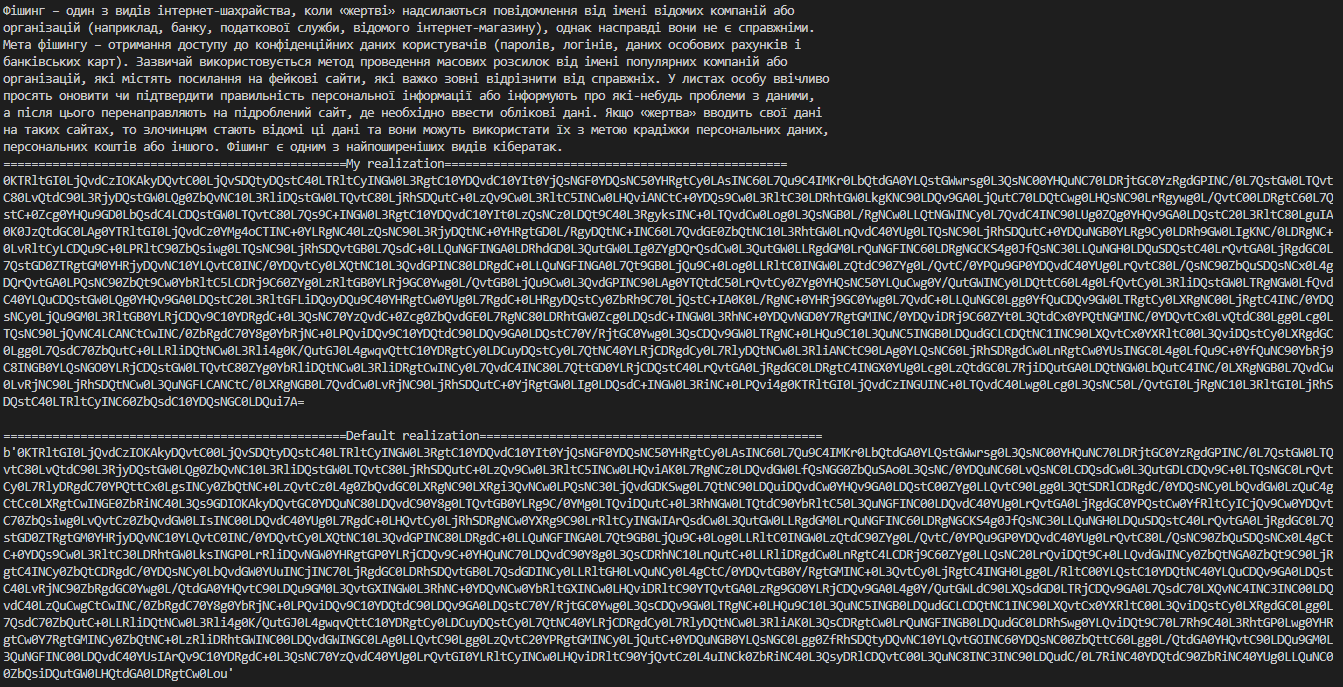
****

* **Lina Kostenko.txt**

****

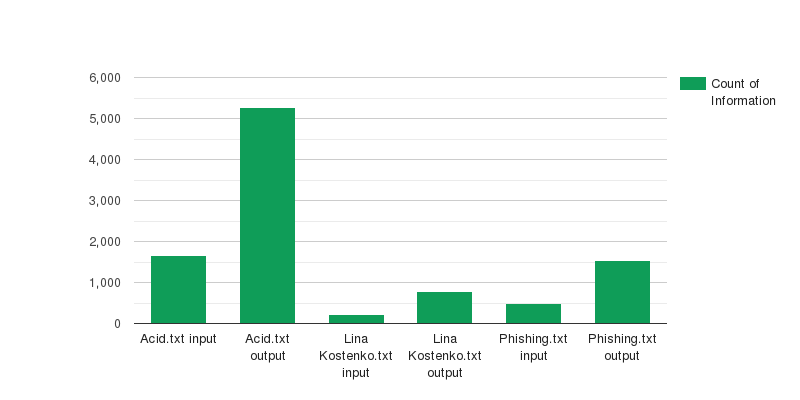
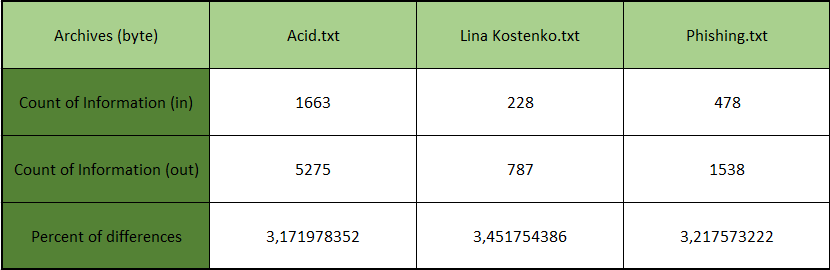
Коректність було також перевірено за допомогою сайту:

* [Online Base64 Encode and Decode](https://www.base64decode.org/)
* **Phishing.txt**

****

3. Закодуйте в Base64 обрані вами текстові файли

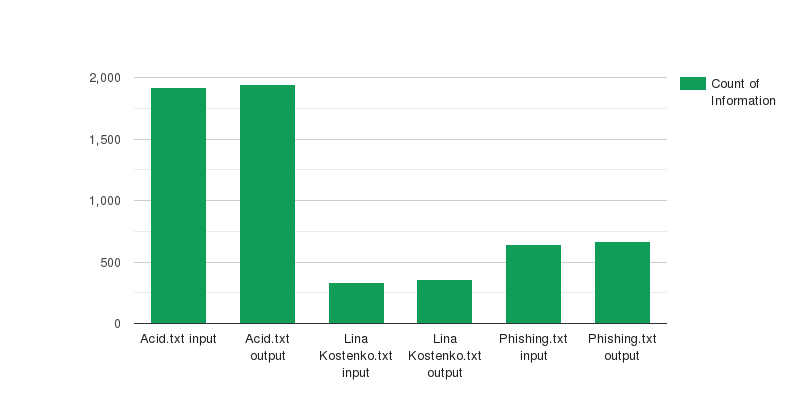
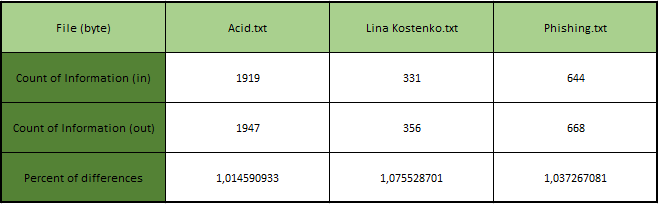
1. Обрахуйте кількість інформації в base64-закодованому варіанті файлу
2. Порівняйте отримане значення з кількістю інформації вихідного файлу
3. Зробіть висновки з отриманого результату



Як висновок, можна сказати, що у закодованому файлі кількості інформації більше у 3.17 – 3.45 разів ніж у вихідному.

4. Закодуйте в Base64 стиснені кращим з алгоритмів текстові файли

1. Обрахуйте кількість інформації в base64-закодованому варіанті стисненого файлу
2. Порівняйте отримане значення з кількістю інформації вихідного файлу та base64-закодованого файлу
3. Зробіть висновки з отриманого результату



Як висновок, можна сказати, що у закодованому файлі кількості інформації більше у 1.015 – 1.075 разів, ніж у вихідному.

**Висновок:**

В процесі виконання лабораторної роботи, було опановано базові знання таких фундаментальних понять, як base64 та процес роботи, а також визначення ентропії інформації. Весь процес намагався побудувати на мові програмування Python, оскільки бачу преспективу даної мови, а також для розширення своїх вмінь та навичок. Протягом виконання роботи, було застосовано 5 варіантів стиснення текстового файлу, а саме ормати: .rar, .zip, .7z, .bz2 та .gz.

Розібрався з алгоритмом кодування base64. Всі додакові матеріали (файли, програми та архіви можна знайти в закріпленому нижче посиланні)

* [Лабораторна робота №1 з Комп'ютерних систем](https://github.com/Boniker/Lab_1_Computer_System)

