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
Гуліцький Олександр Сергійович

Київ-2021

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

Хід виконання роботи

Посилання на репозиторій: github.

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).

    - REST
    - Франко - Привид
    - Terraria – ігровий процес

2. 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!

    Виконано.

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

1) Франко – Привид

```
FileName: franko
Entropy in bits: 4.62450
Amount of information in bits 9027.02214
Amount of information in bytes 1128.37777
File size: 4731 bytes
File size is more than amount of information
Archive .7z size: 1964
Archive .7z size is more than amount of information
Archive .bz2 size: 1565
Archive .bz2 size is more than amount of information
Archive .gz size: 1889
Archive .gz size is more than amount of information
Archive .xz size: 1888
Archive .xz size is more than amount of information
Archive .zip size: 2014
Archive .zip size is more than amount of information
Number and frequency of letter appearance
```

(index)	repeats	frequency
х	33	0.01691
о	187	0.0958
л	92	0.04713
д	54	0.02766
н	115	0.05891
а	149	0.07633
і	110	0.05635
ч	28	0.01434
с	95	0.04867
п	46	0.02357
к	53	0.02715
й	37	0.01895
в	116	0.05943
ж	13	0.00666
з	42	0.02152
ь	40	0.02049
м	66	0.03381
т	106	0.0543
г	39	0.01998
и	108	0.05533
е	88	0.04508
у	68	0.03484
ї	14	0.00717
р	84	0.04303
б	45	0.02305
я	40	0.02049
ю	22	0.01127
є	12	0.00615
ц	13	0.00666
ф	1	0.00051
щ	16	0.0082
ш	20	0.01025

2) REST

```
FileName: rest
Entropy in bits: 4.51434
Amount of information in bits 4383.41960
Amount of information in bytes 547.92745
File size: 2288 bytes
File size is more than amount of information
Archive .7z size: 1080
Archive .7z size is more than amount of information
Archive .bz2 size: 887
Archive .bz2 size is more than amount of information
Archive .gz size: 990
Archive .gz size is more than amount of information
Archive .xz size: 1012
Archive .xz size is more than amount of information
Archive .zip size: 1113
Archive .zip size is more than amount of information
Number and frequency of letter appearance
```

(index)	repeats	frequency
с	40	0.04119
к	40	0.04119
о	95	0.09784
р	55	0.05664
а	55	0.05664
н	57	0.0587
г	10	0.0103
л	37	0.03811
п	42	0.04325
е	58	0.05973
д	45	0.04634
ч	4	0.00412
з	20	0.0206
т	55	0.05664
и	65	0.06694
в	56	0.05767
у	32	0.03296
і	67	0.069
х	15	0.01545
м	30	0.0309
ж	8	0.00824
я	11	0.01133
ю	6	0.00618
ь	20	0.0206
ф	6	0.00618
ц	8	0.00824
й	8	0.00824
б	9	0.00927
є	7	0.00721
ї	4	0.00412
ш	3	0.00309
щ	3	0.00309

3) Terraria – ігровий процес

```
FileName: terraria
Entropy in bits: 4.54181
Amount of information in bits 8847.44428
Amount of information in bytes 1105.93053
File size: 4491 bytes
File size is more than amount of information
Archive .7z size: 1797
Archive .7z size is more than amount of information
Archive .bz2 size: 1456
Archive .bz2 size is more than amount of information
Archive .gz size: 1731
Archive .gz size is more than amount of information
Archive .xz size: 1720
Archive .xz size is more than amount of information
Archive .zip size: 1858
Archive .zip size is more than amount of information
Number and frequency of letter appearance
```

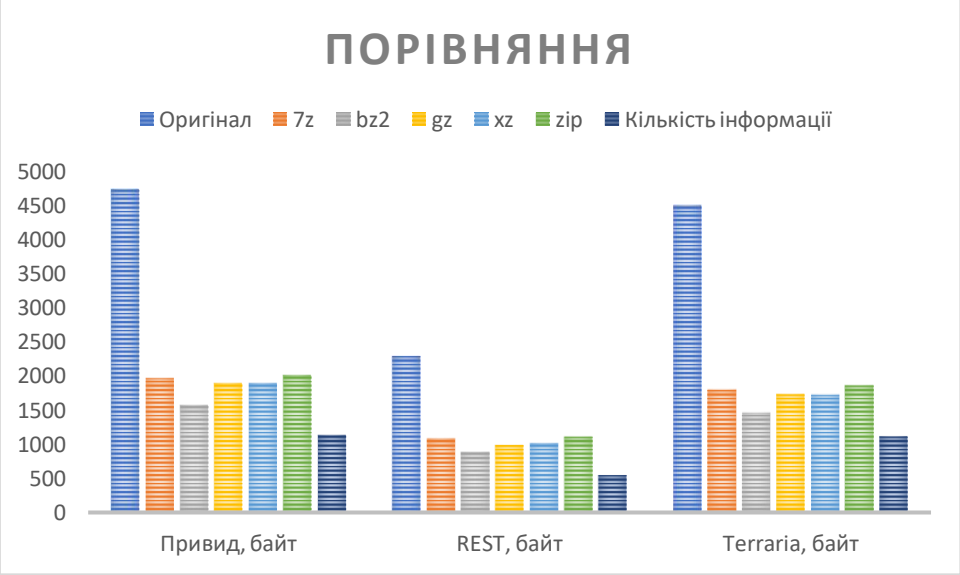
(index)	repeats	frequency
п	52	0.02669
е	89	0.04569
р	92	0.04723
д	77	0.03953
г	24	0.01232
о	183	0.09394
ю	22	0.01129
а	148	0.07598
в	114	0.05852
ц	14	0.00719
і	127	0.0652
т	118	0.06057
б	49	0.02515
н	133	0.06828
с	84	0.04312
и	127	0.0652
ж	32	0.01643
й	19	0.00975
з	42	0.02156
у	51	0.02618
х	20	0.01027
м	66	0.03388
ь	29	0.01489
ч	31	0.01591
к	64	0.03285
я	55	0.02823
л	58	0.02977
ш	10	0.00513
є	12	0.00616
щ	3	0.00154
ф	3	0.00154

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).

   За допомогою програми 7zip файли було стиснено у 7z, bz2, gz, xz та zip.

5. 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).

Файл	Привид, байт	REST, байт	Terraria, байт
Оригінал	4731	2288	4491
7z	1964	1080	1797
bz2	1565	887	1456
gz	1889	990	1731
xz	1888	1012	1720
zip	2014	1113	1858
Кількість інформації	1128	547	1105



Для всіх файлів найкраще стискав алгоритм bzip2, але кількість інформації все одно більше чим обсяг файлу/архіву тому, що алгоритми стискання побудовані таким чином що вони використовують повторюванні частини тексту.

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 (шляхом реалізації алгоритму вручну, а не виклику бібліотечної функції).

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

   За допомогою існуючої функції в node.js декодив результат і отримав аналогічний текст, який використовувався для кодування.


```
0Lkg0L3QsCDQstC40LrQu9C40LrRgyDQstGW0LTQtNCw0LvQtdC90LjRhSDQv9GA0L7RhtC10LTRg9GAIChSZW1vdGUgUHJvY2Vk
dXJlIENhbGwsIFJQQykuINCf0ZbQtNGF0ZbQtCBSUEMg0LTQvtC30LLQvtC70Y/RlCDQstC40LrQvtGA0LjRgdGC0L7QstGD0LLQ
sNGC0Lgg0L3QtdCy0LXQu9C40LrRgyDQutGW0LvRjNC60ZbRgdGC0Ywg0LzQtdGA0LXQttC10LLQuNGFINGA0LXRgdGD0YDRgdGW
0LIg0Lcg0LLQtdC70LjQutC+0Y4g0LrRltC70YzQutGW0YHRgtGOINC80LXRgtC+0LTRltCyINGWINGB0LrQu9Cw0LTQvdC40Lwg
0L/RgNC+0YLQvtC60L7Qu9C+0LwuINCf0YDQuCDQv9GW0LTRhdC+0LTRliBSRVNUINC60ZbQu9GM0LrRltGB0YLRjCDQvNC10YLQ
vtC00ZbQsiDRliDRgdC60LvQsNC00L3RltGB0YLRjCDQv9GA0L7RgtC+0LrQvtC70YMg0YHRg9Cy0L7RgNC+INC+0LHQvNC10LbQ
tdC90ZYsINGJ0L4g0L/RgNC40LfQstC+0LTQuNGC0Ywg0LTQviDRgtC+0LPQviwg0YnQviDQutGW0LvRjNC60ZbRgdGC0Ywg0L7Q
utGA0LXQvNC40YUg0YDQtdGB0YPRgNGB0ZbQsiDQvNCw0ZQg0LHRg9GC0Lgg0LLQtdC70LjQutC+0Y4uDQpSRVNUIOKAlCDRhtC1
INCw0YDRhdGW0YLQtdC60YLRg9GA0L3QuNC5INGB0YLQuNC70Ywg0LTQu9GPINGA0L7Qt9C/0L7QtNGW0LvQtdC90LjRhSDQs9GW
0L/QtdGA0YLQtdC60YHRgtC+0LLQuNGFINGB0LjRgdGC0LXQvC4A=
```


```
Decoded Text:
REST (скор. англ. Representational State Transfer, «передача репрезентативного стану») — підхід до а
рхітектури мережевих протоколів, які надають доступ до інформаційних ресурсів. Був описаний і популя
ризований 2000 року Роєм Філдінгом, одним із творців протоколу HTTP. В основі REST закладено принцип
и функціонування Всесвітньої павутини і, зокрема, можливості HTTP. Філдінг розробив REST паралельно
з HTTP 1.1 базуючись на попередньому протоколі HTTP 1.0.
Дані повинні передаватися у вигляді невеликої кількості стандартних форматів (наприклад, HTML, XML,
JSON). Будь-який REST протокол (HTTP в тому числі) повинен підтримувати кешування, не повинен залежа
ти від мережевого прошарку, не повинен зберігати інформації про стан між парами «запит-відповідь». С
тверджується, що такий підхід забезпечує масштабовність системи і дозволяє їй еволюціонувати з новим
и вимогами.
```

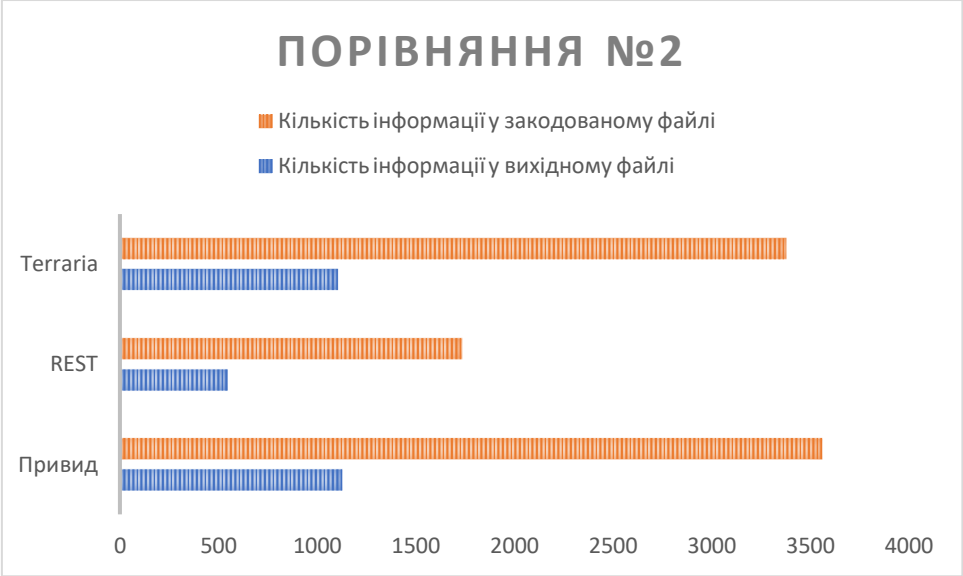
3. Закодуйте в Base64 обрані вами текстові файли

   a. Обрахуйте кількість інформації в base64-закодованому варіанті файлу

   b. Порівняйте отримане значення з кількістю інформації вихідного файлу

   c. Зробіть висновки з отриманого результату

Файл	Кількість інформації у вихідному файлі	Кількість інформації у закодованому файлі
Привид	1128	3560
REST	547	1734
Terraria	1105	3378



Кількість інформації у закодованому файлі в середньому в 3.1 раза більше чим в вихідному.

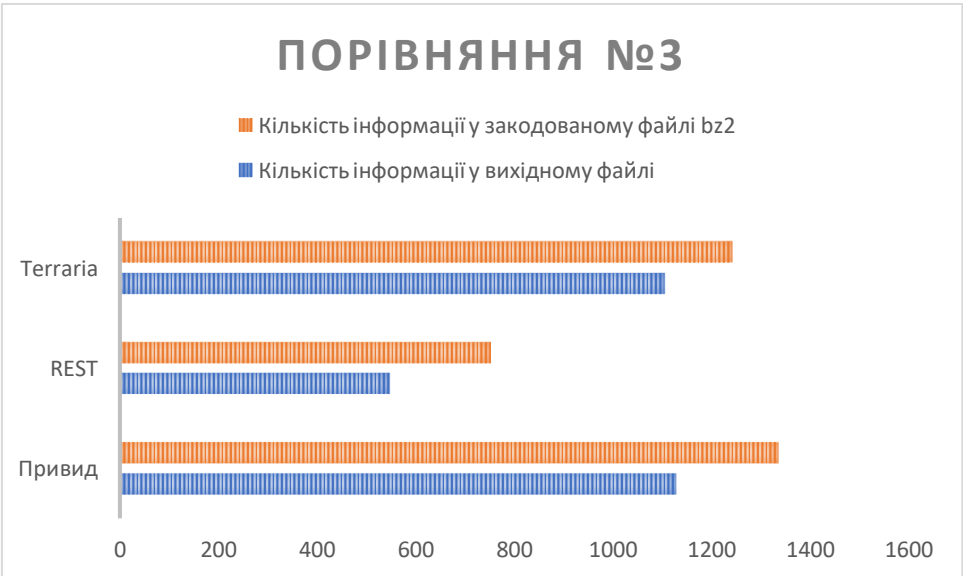
4. Закодуйте в Base64 стиснені кращим з алгоритмів текстові файли

   a. Обрахуйте кількість інформації в base64-закодованому варіанті стисненого файлу

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

   c. Зробіть висновки з отриманого результату

Файл	Кількість інформації у вихідному файлі	Кількість інформації у закодованому файлі bz2
Привид	1128	1336
REST	547	752
Terraria	1105	1242



Кількість інформації у закодованому bz2 файлі в середньому в 1.2 рази більше чим у вихідному файлі.

Висновок: під час виконання роботи було досліджено кількість інформації у тексті, розглянуто алгоритм кодування інформації Base64, і на основі алгоритму було створено програму кодування інформації в Base64. Правильність кодування власноруч написаної програми було перевірено за допомогою вже існуючих засобів декодування інформації з Base64.