Міністерство освіти і науки України
Національний університет “Львівська політехніка”
ІНСТИТУТ ІКНІ Кафедра САПР

Звіт

до лабораторних робіт №1-4 з курсу

“ *Технології захисту інформації* ”

на тему: “**МЕТОДИ ШИФРУВАННЯ ТА ДЕШИФРУВАННЯ ТЕКСТОВИХ ПОВІДОМЛЕНЬ”**

Виконав:
студент гр. КН - 307
Волос І. В.
Перевірив:
Іванців Р. Д.

2020

Лабораторна робота 1: Статистичні властивості відкритого та шифрованого тексту.

Мета роботи: Мета роботи – дослідження статистичних властивостей відкритого тексту (ВТ) та шифрованого тексту (ШТ). Зображення статистичних даних на гістограмах за алфавітом та за спаданням.

Короткі теоретичні відомості:

Криптографічні системи.

Перетворення секретної інформації – це кодування даних, яке використовується для маскування інформації. Ці перетворення змінюють дані, представлені в явній формі, так, що вони стають нерозпізнаваними (з певною мірою надійності) неавторизованими людьми. Шифрування, тобто перетворення секретної інформації, є особливо ефективним засобом проти несанкціонованого доступу до повідомлень, які передаються по лініях зв'язку і неавторизованих доступів до даних, що зберігаються у віддалених файлах. Навіть просте перетворення секретної інформації є ефективним засобом, що дає можливість приховати її значення від більшості користувачів.

У загальній формі в криптографічній системі початкове повідомлення М і ключ К є входами в деякий перетворюючий пристрій (який може бути реалізоване або апаратно, або програмно). Якщо до лінії зв'язку підключиться зловмисник, він зможе викрасти зашифроване повідомлення Е. Процес відновлення повідомлення при невідомому ключі називається дешифруванням.

Початкове повідомлення може бути замасковане за рахунок процедури перетворення. На іншому кінці незахищеної (по допущенню) лінії зв'язку знаходиться апаратний або програмний засіб, який розшифровує повідомлення для отримання початкового тексту.

Шифрувальні пристрої можуть встановлюватися між кінцевими пристроями і лінією зв'язку або можуть бути вбудовані виробниками в периферійні пристрої різними способами. Поступаючі сигнали управління зовнішніми пристроями, стани і ідентифікаційна інформація між компонентами можуть передаватися в незашифрованому вигляді, при цьому ефективність захисту не зменшиться.

Шифрування підстановкою. Прямі підстановки.

У прямих підстановках кожний знак початкового тексту замінюється одним або декількома знаками. Одним з важливих підкласів прямих підстановок є моноалфавітні підстановки, в яких встановлюється взаємооднозначна відповідність між кожним знаком аі алфавіту повідомлень А і відповідним знаком hj зашифрованого тексту.

Шифрування підстановкою. Багатоалфавітні підстановки.

Проста багатоалфавітна підстановка послідовно і циклічно міняє алфавіти, що використовуються. При u-алфавітній підстановці знак т1, з початкового повідомлення замінюється знаком з алфавіту В1, т2 відповідним з алфавіту В2. .., ті - знаком з алфавіту Bi, mi+1 знову з алфавіту В1 і т. д.

Шифр Віженера - поліалфавітний шифр, який у якості ключа використовує слово.

Монофонічні шифри.

Монофонічний шифр являє собою багатоалфавітний шифр підстановки, що зрівнює частоту появи зашифрованих знаків і таким чином захищає шифрований текст від розкриття за допомогою частотного аналізу. Для знаків, що зустрічаються часто, потрібна відносно велика кількість зашифрованих еквівалентів. У той же час для знаків, що використовуються нечасто, може виявитися достатнім один або два зашифрованих знаки.

Шифри перестановки.

Розглянутий вище метод ґрунтувався на заміщенні символів відкритого тексту різними символами шифрованого тексту. Принципово інший клас перетворень будується на використанні перестановок букв відкритого тексту. Шифри, створені за допомогою перестановок, називають перестановочными шифрами або шифри перестановки.

Завдання:

1. Написати програму для статистичного аналізу тексту. Статистичний аналіз повинен дати такі параметри аналізованого тексту:

- визначити використаний алфавіт;

- частоти повторень одного символу для всіх символів тексту, які представити двома способами – в алфавітному порядку і по спаданню частоти повторення у вигляді гістограм;

- частоти повторень для біграм (двох символів, які зустрічаються в тексті в кількості 10-15), які представити у вигляді гістограм;

- частоти повторень для триграм (трьох символів, які зустрічаються в тексті в кількості 10-15), які представити у вигляді гістограм;

- знайти повторення символів в тексті для 2, 3 і 4 символів.

Отримати відкритий текст і виконати статистичний аналіз. Зробити висновки на основі отриманих статистичних характеристик відкритого тексту.

2. Знайти в кожному тексті (ШТ і ВТ) повторення буквосполучень (2-3-4).

Хід роботи:

1) Створив сайт в середовищі Brackets використовуючи html, css та javascript. На даній веб-сторінці можна провести статистичний аналіз тексту, а також розшифровувати і зашифровувати введений текст у першому текстовому полі за шифром Цезара. Результат цих дій у текстовому полі під написом “Результат шифрування”. Для статистичного аналізу потрібно натиснути на кнопку “Проаналізувати”. На рис. 1 можна побачити візуальний інтерфейс даної програми:

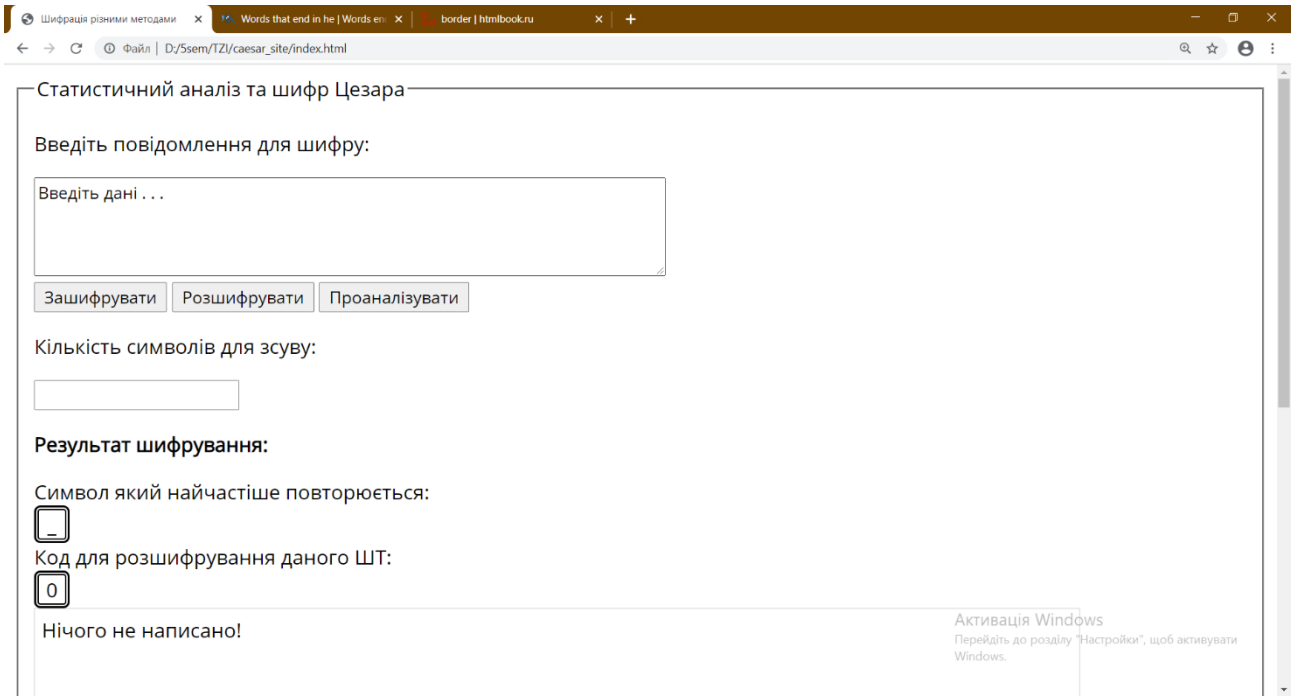

Рис. 1. Веб-сторінка для статистичного аналізу ВТ та ШТ

2) Вписавши наданий викладачем ВТ у відповідне текстове поле, натискаю кнопку “Проаналізувати” та отримую ключ для дешифрування даного ШТ та гістограми повторення символів, біграм, триграм та чотирьох символів. У таблиці 1 виведений використаний алфавіт з кількостями для ВТ.

Таблиця 1 – Алфавіт з кількостями повторень[1]

Символ	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
К-сть	317	82	173	163	455	88	77	154	294	3	33	130	86	277	234	96
Символ	Q	R	S	T	U	V	W	X	Y	Z	_	.	,	;	-	‘
К-сть	3	231	297	317	106	30	60	7	98	7	730	30	57	1	21	11

[1] Таблиця оформлена згідно з стандартами ДСТУ 3008-95.

3) Результат роботи на рис. 2-6:

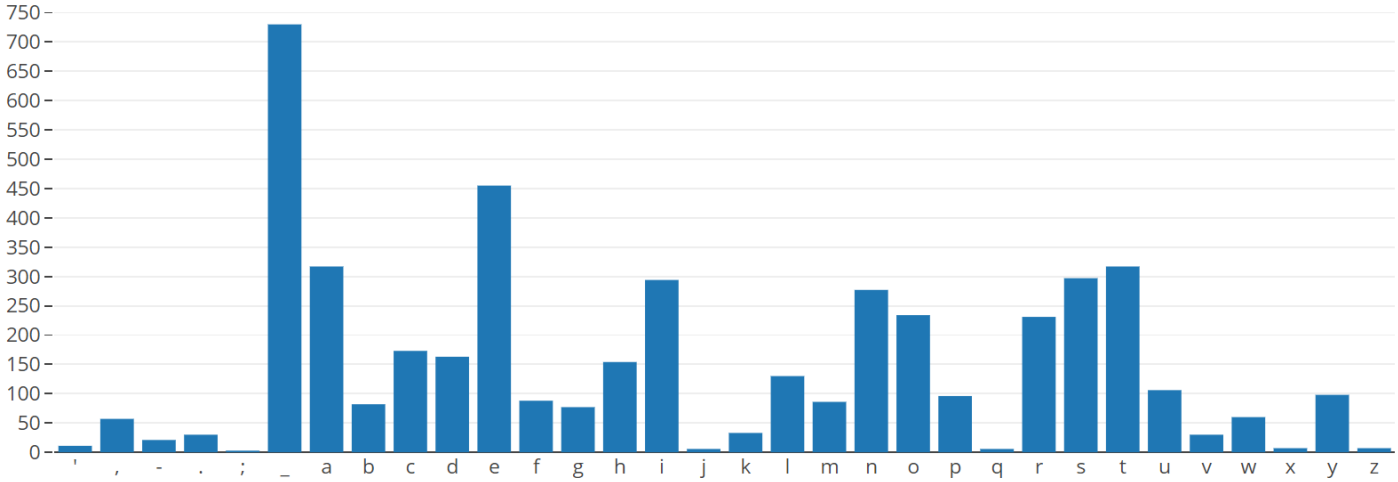

Рис.2 Гістограма для одного символу (за алфавітом)

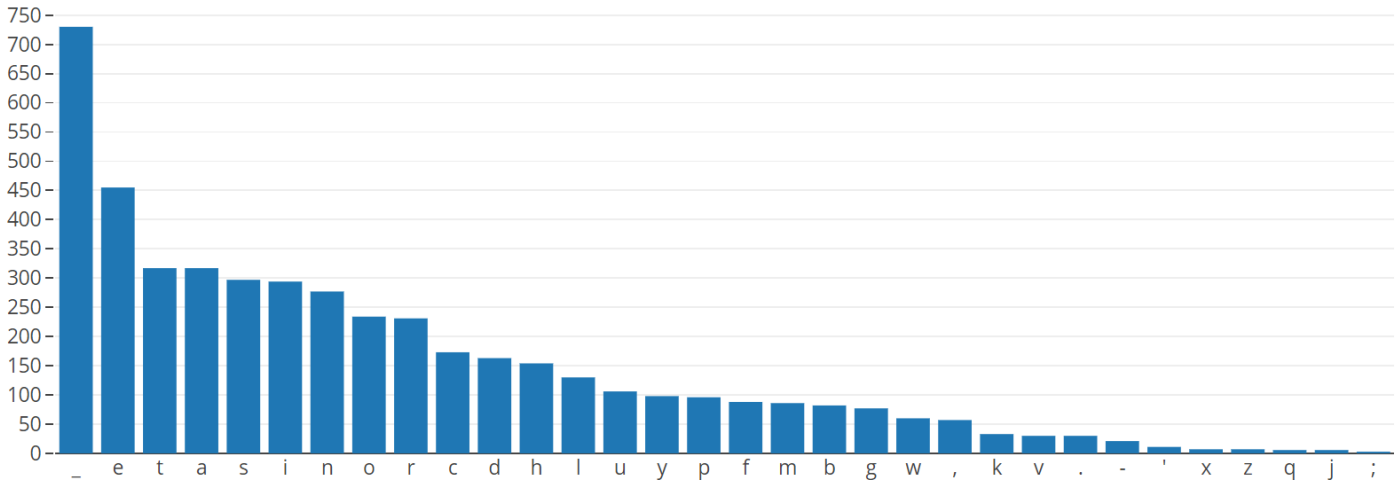

Рис.3 Гістограма для одного символу (за спаданням)

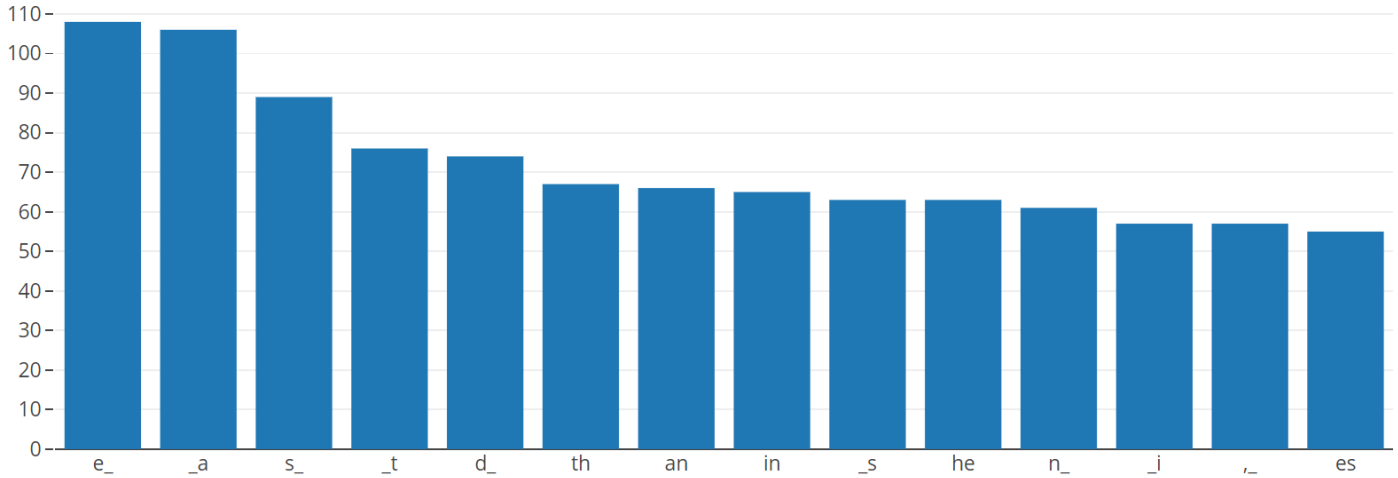

Рис.4 Гістограма для біграм (за спаданням)

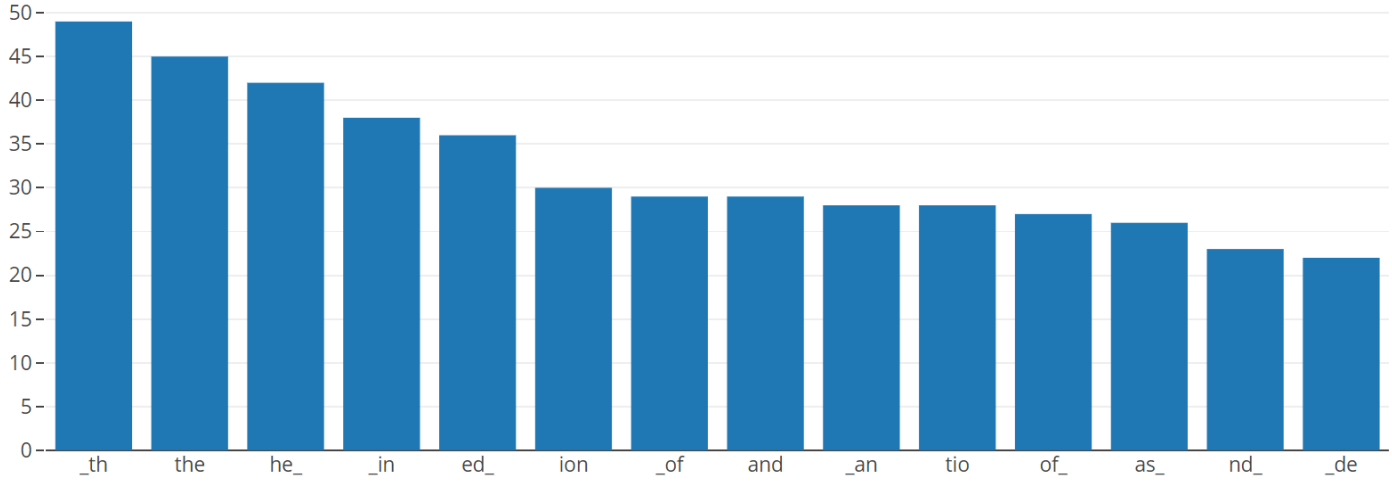

Рис.5 Гістограма для триграм (за спаданням)

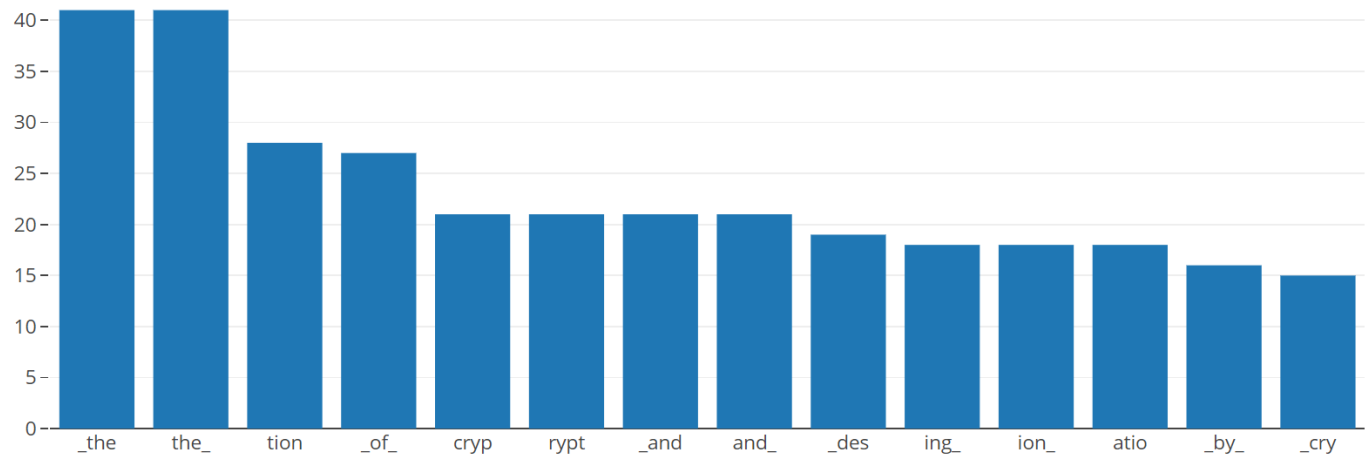

Рис.6 Гістограма для чотириграм (за спаданням)

4) Перефарбував повторювані слова ВТ, ось вивід:

it int roduced a radically new method of distributing cryptographic keys, whichwent far toward sol ving one of the fundamental problems of cryptography, key distributi on, and has become known as diffiehellman key exchange. the article also stimulated the almost immediate public developme nt of a new class of enciphering algorithms, the asymmetric key algorithms. prior to that time, all u seful modern encryption algorithms had been symmetric key algorithms, in which the same crypto graphic key is used with the underlying algorithm by both the sender and the recipient, who must both keep it secret. all of the electromechanical machines used in wwii were of this logical class, as were the caesar and atbash ciphers and essentially all cipher systems throughout history. the 'key' for a code is, of course, the codebook, which must likewise be distributed and kept secret, and so shares most of the same problems in practice. of necessity, the key in every such system had to be exchanged between the communicating parties in some secure way prior to any use of the system the term usually used is 'via a secure channel' such as a trustworthy courier with a briefcase hand cuffed to a wrist, or facetoface contact, or a loyal carrier pigeon. this requirement is never trivial a nd very rapidly becomes unmanageable as the number of participants increases, or when secure c hannels aren't available for key exchange, or when, as is sensible cryptographic practice, keys are f requently changed.the aging des was officially replaced by the advanced encryption standard aes i n when nist announced fips. after an open competition, nist selected rijndael, submitted by two be lgian cryptographers, to be the aes. des, and more secure variants of it such as triple des, are still u sed today, having been incorporated into many national and organizational standards. however, it

s xbit keysize has been shown to be insufficient to guard against brute force attacks one such attac k, undertaken by the cyber civilrights group electronic frontier foundation in, succeeded in- hours. as a result, use of straight des encryption is now withou doubt insecure for use in new cryptosyste m designs, and messages protected by older cryptosystems using des, and indeed all messages sen t since using des, are also at risk. regardless of des' inherent quality, the des key size was thought t o be too small by some even in, perhaps most publicly by whitfield diffie. there was suspicion that government organizations even then had sufficient computing power to break des messages; clear ly others have achieved this capability.the second development, in, was perhaps even more impor tant, for it fundamentally changed the way cryptosystems might work. this was the publication of t he paper new directions in cryptography by whitfield diffie and martin hellman. it introduced a rad ically new method of distributing cryptographic keys, which went far toward solving one of the fun damental problems of cryptography, key distribution, and has become known as diffiehellman key exchange.he mids saw two major public advances. first was the publication of the draft data encry ption standard in the u.s. federal register on march. the proposed des cipher was submitted by a r esearch group at ibm, at the invitation of the national bureau of standards now nist, in an effort to develop secure electronic communication facilities for businesses such as banks and other large fi nancial organizations. after 'advice' and modification by nsa, acting behind the scenes, it was adop ted and published as a federal information processing standard publication in. des was the first pu blicly accessible cipher to be 'blessed' by a national agency such as nsa. the release of its specificati on by nbs stimulated an explosion of public and academic interest in cryptography. the aging des was officially replaced by the advanced encryption standard aes in- when nist announced fips. afte r an open competition, nist selected rijndael, submitted by two belgian cryptographers, to be the a es. des, and more secure variants of it, are still used today, having been incorporated into many na tional and organizational standards. however, its xbit keysize has been shown to be insufficient to guard against brute force attacks one such attack, undertaken by the cyber civilrights group electr onic frontier foundation inz, succeeded in- hours. as a result, use of straight des encryption is now without doubt insecure for use in new cryptosystem designs, and messages protected by older cry ptosystems using des, and indeed all messages sent since using des, are also at risk. regardless of d es' inherent

Висновок:

Протягом виконання даної лабораторної роботи провів дослідження статистичних властивостей шифрованого тексту (ШТ). Вивів гістограми проведеного аналізу для одного символу в алфавітному порядку (рис. 2) та за спаданням (рис. 3). Також, вивів гістограми за спаданням з першими 15 символами, які найчастіше зустрічаються, для біграм (рис. 4), триграм (рис. 5) та чотириграм (рис. 6).

З отриманих результатів на рис. 2 видно, що пробіл – символ, який найчастіше з'являється у тексті. Отже, для достовірного аналізу і дешифрування шифрованого тексту спочатку найпростіше буде замінити символ з найбільшою частотою на пробіл, або символ '_', що розділить текст на окремі слова. Цей крок може дати нам ключ до дешифрування ШТ або його фрагментів, зашифрованих методом моноалфавітної підстановки, а саме шифром Цезара.

Вирішив також зазначити важливість аналізувати символи не тільки по одному, а і символьні рядки (біграми, триграми, чотириграми), оскільки це спрощує дешифрування часток, артиклів та слів з довжиною до 5 символів.

Наприклад, на рис. 5 можна переглянути перші 3 триграми, які аналізують схожі символьні рядки: **"_th"** (к-сть: 49) має пораховану кількість слів, що починаються на th (як приклад: "**th**e", "**th**eater", "**th**ink"), **"the"**(к-сть: 45) – це слова, в яких можна зустріти цю комбінацію символів (як приклад "**the**", "mo**the**r", "wea**the**r") та "**he_**" (42 повторень), що вказує на кількість слів, що закінчуються на **he** (Приклади: "t**he**", "s**he**", "cac**he**").

А от проаналізувавши чотириграми, можна отримати конкретнішу інформацію про артикль **the**. Якщо поглянути на 2 найчастіші чотириграми на рис. 6, можна побачити що слів що починаються на **the** є 41, і так само 41 є таких, що закінчуються на **the**. Значить можна зробити висновок, що к-сть артиклів **the** є не більше 41, а при аналізі п'ятиграм можна би було сказати з впевненістю конкретне число повторень. Також, на рис. 6 можна побачити чотириграму "**_of_**" що дає нам конкретне число повторень (27) даної частки з англійської мови.

Отже, статистичний аналіз тексту значно полегшує дешифрування зашифрованих повідомлень для криптографа, оскільки на таку роботу, яку робить обчислювальна програма, піде в 100 раз більше часу для цілого відділу криптографів.

Лабораторна робота 2: Шифр Цезара.

Мета роботи: дослідження статистичних властивостей відкритого тексту (ВТ) та шифрованого тексту (ШТ). Вивчення простих методів шифрування та дешифрування інформації та їх властивостей (для шифрів заміни та шифрів перестановок).

Завдання:

1. Ознайомитись з методом шифрування.
2. Отримати зашифрований текст з допомогою шифра Цезара. Виконати для цього ШТ статистичний аналіз і на основі отриманих результатів і висновків, які отримані в п.2 знайти ключ, прочитати і надрукувати ВТ і отриманий ключ шифрування. Зробити висновки які властивості відкритого і шифрованого текстів були використані при розшифруванні ШТ.

Індивідуальне завдання. Розшифрувати текст:

C'VDIQ'IODJIVJAVOC'VAM'LP'I;TV.I.GTNDNVO';CIDLP'VAJMV,M'.FDIBVHJ
IJ.GKC.,'OD;VNP,NODOPODJIV;DKC'MNXV,TV.GZFDI-DXV.IV.M.,VH.OC'H.
OD;D.IXVNJH'ODH'V.MJPI-V. -WVDOVR.NVOC'VHJNOVAPI-.H'IO.GV;MTK
O.I.GTOD;V. -Q.I;'VPIODGVRRDDWV.GZFDI-DVRMJO'V.V,JJFVJIV;MTKOJB
M.KCTV'IODOG'-VMDN.G.CVADVDNODFCM.EV.GZHP.HH.VH.IPN;MDKOV
AJMVOC'V-';DKC'MDIBV;MTKOJBM.KCD;VH'NN.B'NXVDIVRCD;CVC'V-'N;
MD,'-VOC'VADMNOV;MTKO.I.GTNDNVO';,.SAFLGSEGFME,FLKS;JGESL',S
GD.SCAF-.GESG;S, -QHLS AJ YSZ
TSL',K,SYJ,SFGLSL'GM-'LSLGSZ,SK,JAGMKSYLL,EHLKSYLSK, J,LS
GEEMFAYLAGFKUS'GO,N,JUSZMLSJYL',JSLGS'YN,SZ,,R.NVKMJ,.,GTVM'G
DBDJPNGTVHJODQ.O'-VO'SOP.GV.I.
GTNDNVJAVOC'VLPM .IVRCD;CVG'-VOJVO

Хід роботи:

1) Виконав статистичний аналіз для даного ШТ. Ввів ШТ у відведене поле на сайті, натиснув кнопку “Проаналізувати”. Виведені гістограми з частотами повторень символів та символьних рядків у проаналізованому тексті на рис. 7-11:

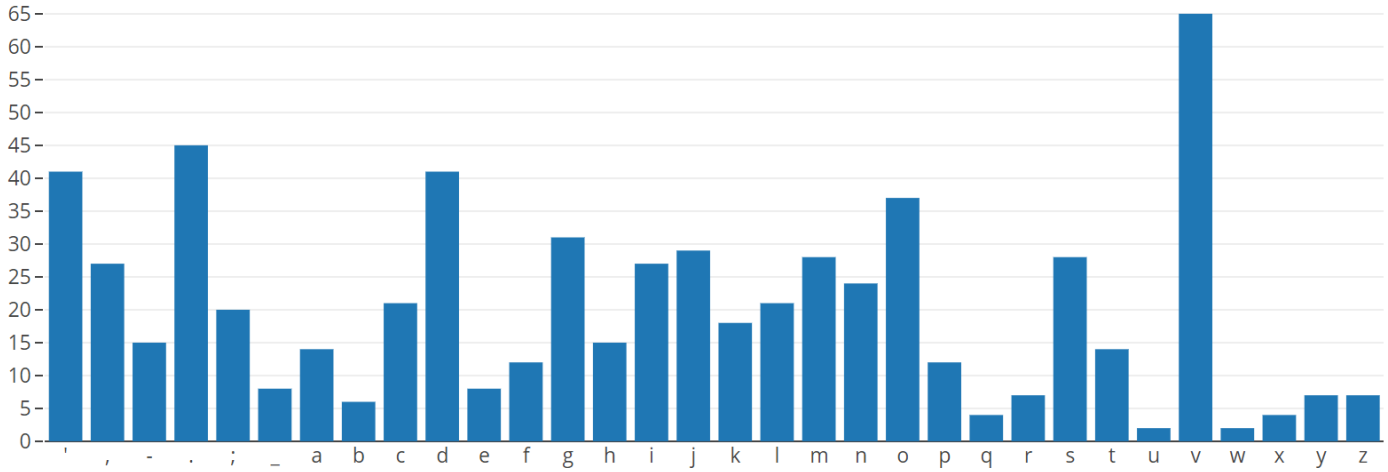

Рис.7 Гістограма для одного символу (за алфавітом)

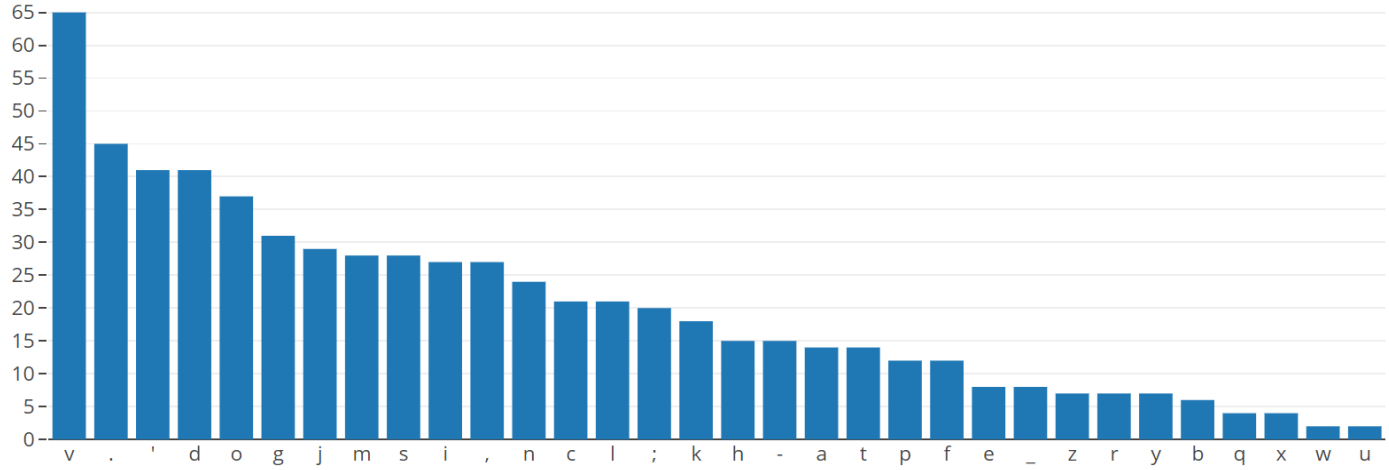

Рис.8 Гістограма для одного символу (за спаданням)

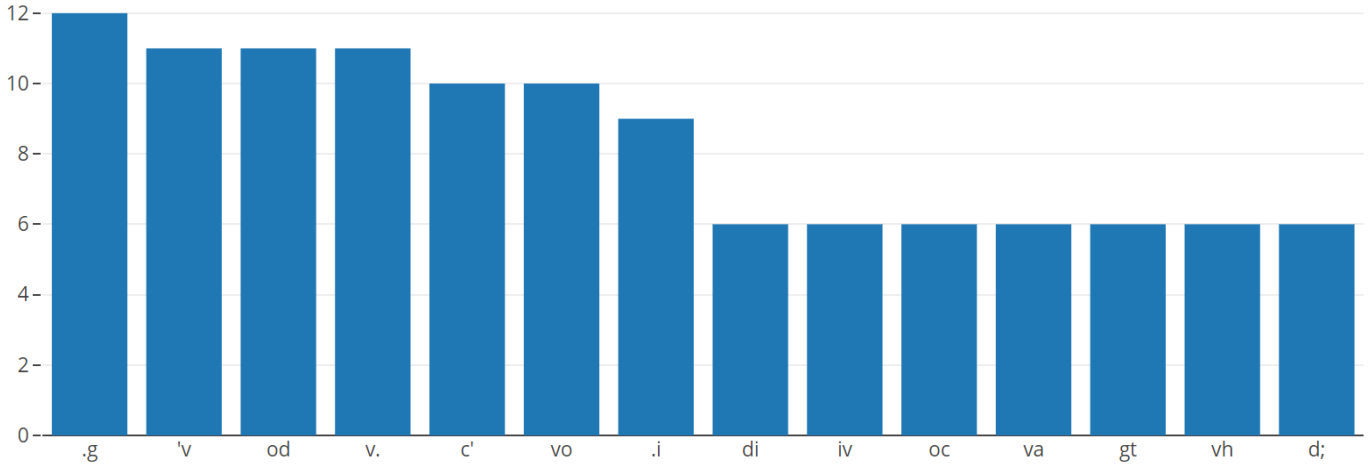

Рис.9 Гістограма для біграм (за спаданням)

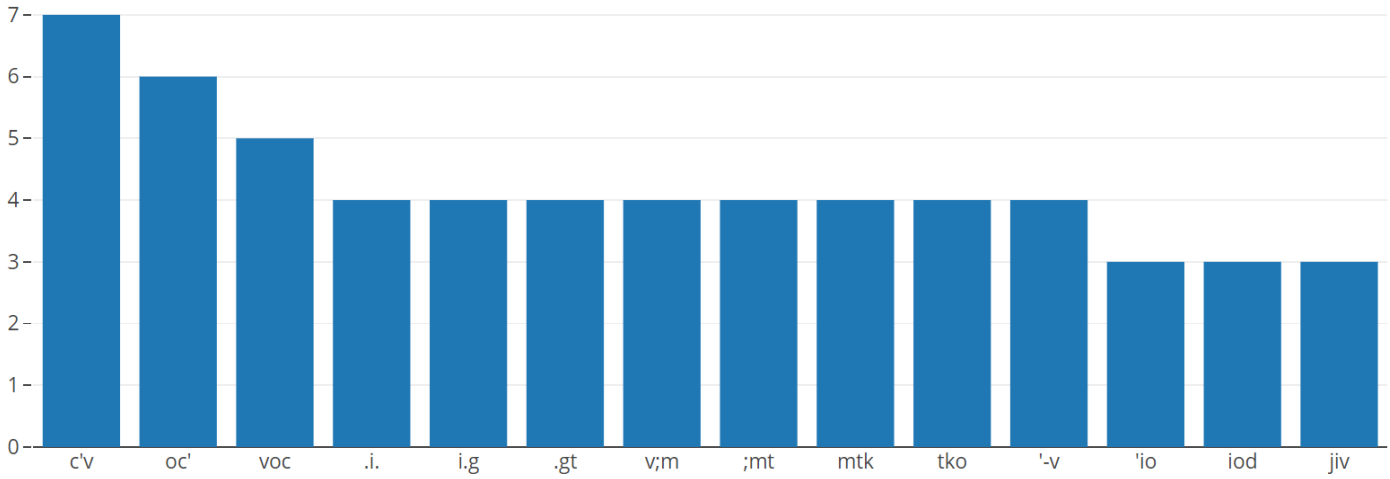

Рис.10 Гістограма для триграм (за спаданням)

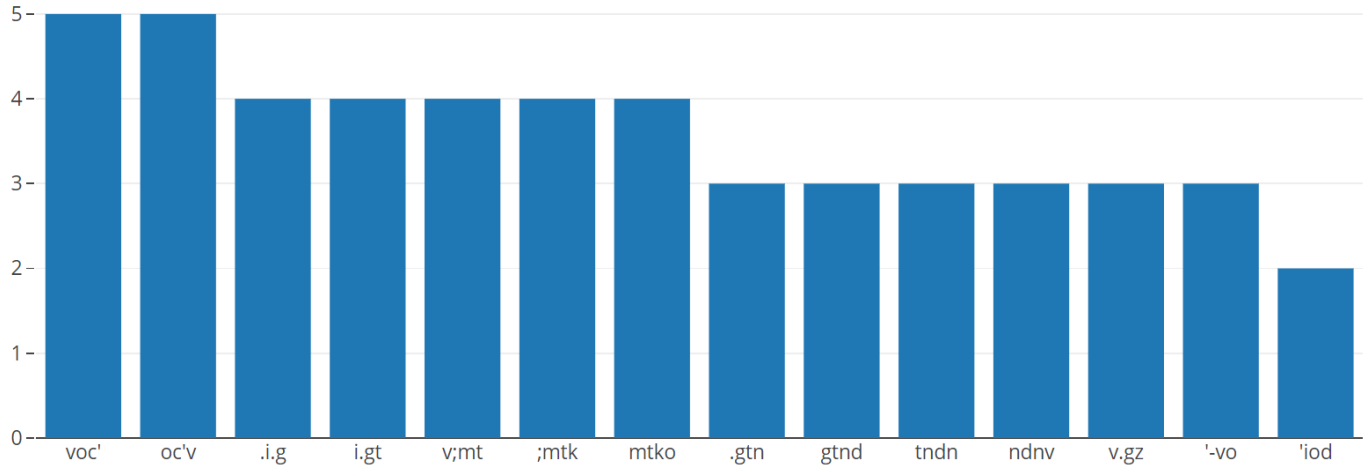

Рис.11 Гістограма для чотириграм (за спаданням)

2) Коли я натиснув на кнопку “Проаналізувати” в пункті 1, я надрукував під результатом символ який найчастіше зустрічається, і ключ, що потрібний для розшифрування даного ШТ. Цей вивід можна побачити на рис. 12:

Рис.12 Вивід програми про найчастіший символ і ключ для шифру Цезара

Результат шифрування:

Символ який найчастіше повторюється:

v

Код для розшифрування даного ШТ:

27

3) Ввів даний ключ у поле для ключа розшифрування, і натиснувши кнопку “Розшифрувати”, отримав вивід на рис. 13:

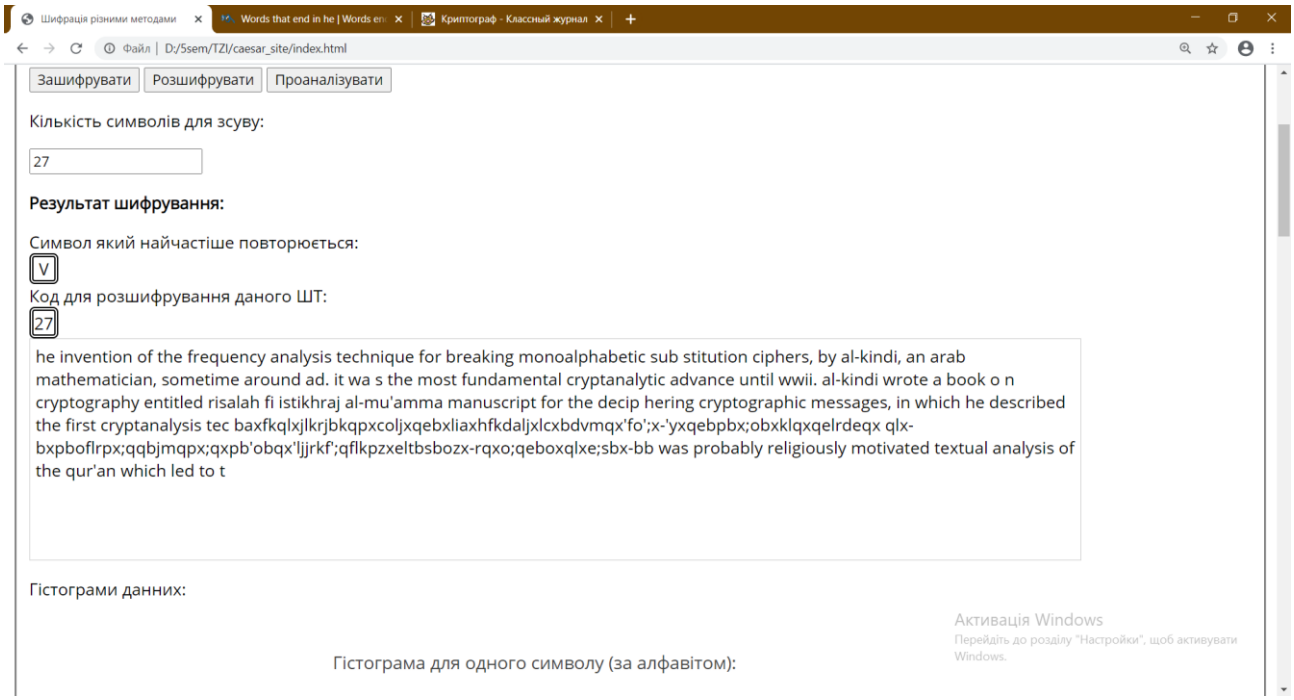

Рис.13 Результат програми для шифру Цезара

Отриманий результат після дешифрування за ключем 27:

he invention of the frequency analysis technique for breaking monoalphabetic sub stitution ciphers, by al-kindi, an arab mathematician, sometime around ad. it wa s the most fundamental cryptanalytic advance until wwii. al-kindi wrote a book o n cryptography entitled risalah fi istikhraj al-mu'amma manuscript for the decip hering cryptographic messages, in which he described the first cryptanalysis tec baxfkqlxjlkrjbkqpxcoljxqebxliaxhfkdaljxlcxbdvmqx'fo';x'yxqebpbx;obxklqxqelrdeqx qlxbxpboflrpx;qqbjmqpx;qxpb'obqx'ljjrkf';qflkpzxeltbsbozx-rqxo;qeboxqlxe;sbx-bb was probably religiously motivated textual analysis of the qur'an which led to t

4) Як видно з минулого пункту, текст не є ще повністю розшифрований, роблю окремо аналіз для частини ШТ, яка не є дешифрована. Аналіз символів по спаданню на рис. 14, а знайдений ключ для дешифрування на рис. 15:

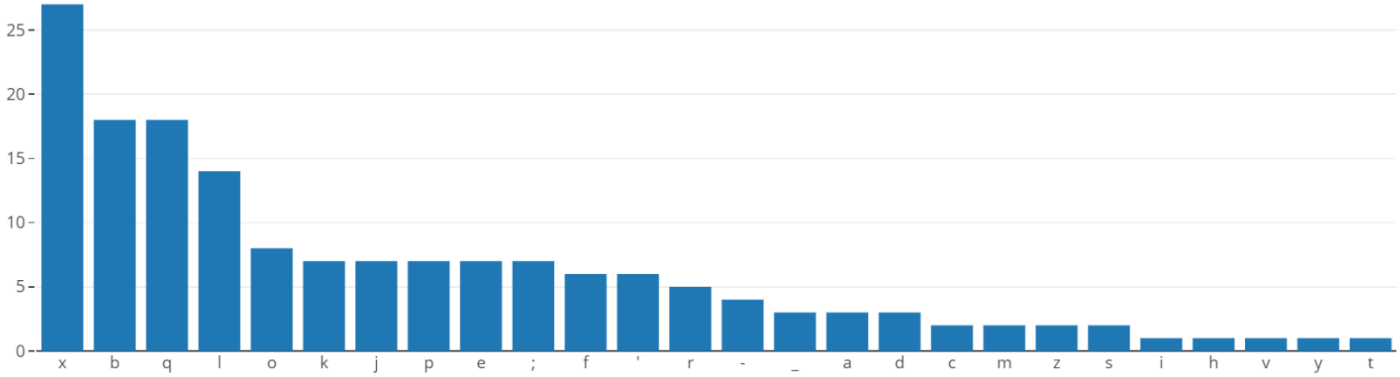

Рис.14 Гістограма для одного символу (за спаданням) для частини ШТ

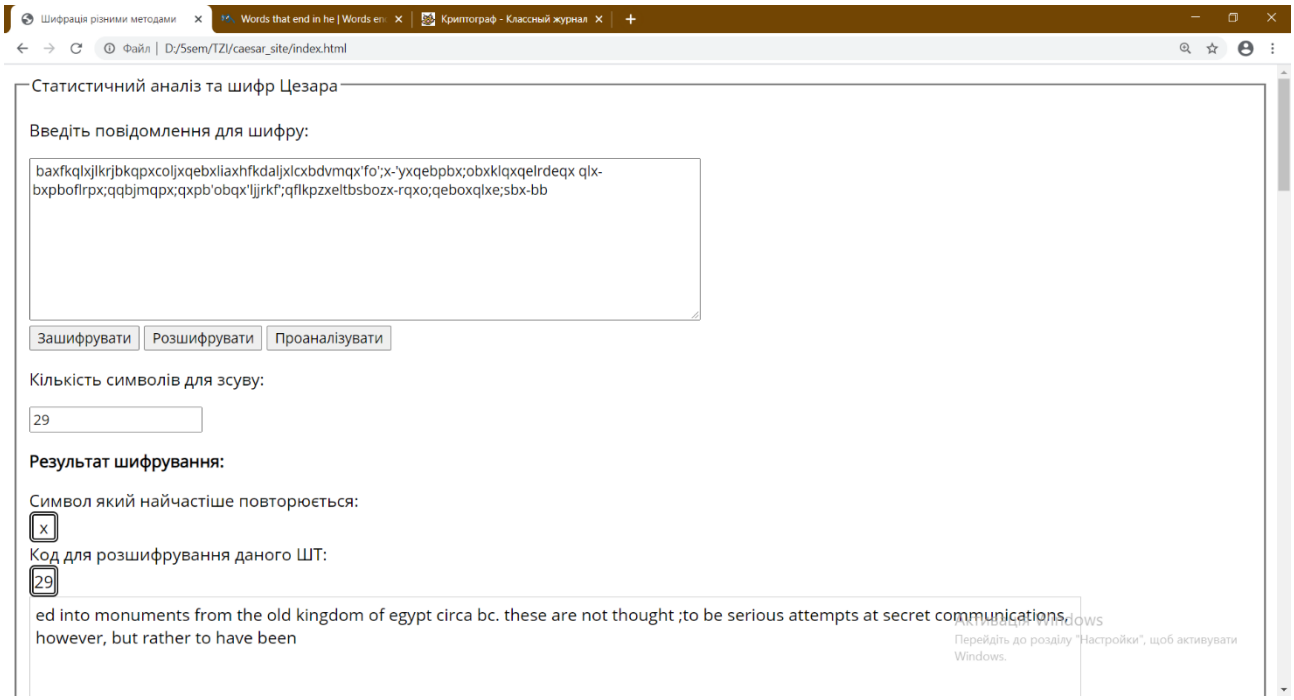

Рис.15 Результат програми для нерозшифрованої частини ШТ

Дешифрована частина за ключем 29:

ed into monuments from the old kingdom of egypt circa bc. these are not thought to be serious attempts at secret communications, however, but rather to have been

Повністю розшифрований текст:

he invention of the frequency analysis technique for breaking monoalphabetic sub stitution ciphers, by al-kindi, an arab mathematician, sometime around ad. it was the most fundamental cryptanalytic advance until wwii. al-kindi wrote a book on cryptography entitled risalah fi istikhraj al-mu'amma manuscript for the deciphering cryptographic messages, in which he described the first cryptanalysis tec ed into monuments from the old kingdom of egypt circa bc. these are not thought to be serious attempts at secret communications, however, but rather to have been was probably religiously motivated textual analysis of the qur'an which led to t

Для розшифрування більшої частини тексту підійшов ключ **27**, для окремого речення підійшов ключ **29** (при умові якщо дешифрування по цьому ключу проводиться опісля дешифрування по ключу 27) або якщо з початкового повідомлення, то ключ для речення дорівнює **24**.

Висновок:

Протягом виконання даної лабораторної роботи провів дослідження шифрованого тексту методом Цезара, розробив сайт для шифрування та дешифрування повідомлень цим методом, та за його допомогою найшов ключ для дешифрації ШТ і розшифрував даний викладачем текст.

З отриманих результатів у лабораторній роботі №1, видно, що пробіл – символ, який найчастіше з'являється у тексті. Отже, щоб віднайти зсув в тексті, потрібно знайти відстань від найчастішого символу до пробілу в масиві алфавіту крокуючи протилежно алфавітному порядку. Цей зсув находиться з такого рівняння: key = (32 + I - 26) % 32. i – це порядковий номер найчастішого символа в масиві алфавіту, а 26 – це порядковий номер пробілу.

Спочатку відформатував текст для подальшого аналізу:

- Видалив всі абзаци у тексті;
- Змінив регістр, зменшивши всі букви;
- Замінив пробіли на символ '_' для подальшого зображення на гістограмах і у виводі.

Потім, проаналізував текст, заповнивши масиви з частотами символів, біграм, триграм, чотириграм, і використовуючи ці дані, побудував через бібліотеку Javascript'а "**Plottly**" гістограми. Дані гістограми дозволили зрозуміти як правильно рахувати відстань від найчастішого символа до пробіла, і дали базові поняття про повторення символів у текстах, на які можна спиратись при подальшій дешифрації моноалфавітних шифрів.

Наприклад, як і у минулому тексті у лабораторній роботі, найчастішим символом у ВТ є пробіл, але вже другий символ був інший. У першій лабораторній роботі це був символ **e**(**455**, коли **t: 317**), а тут **t** (**54**, коли **e: 52**), хоч і не на багато. Це можна просто пояснити, оскільки даний текст у другій лабораторній роботі є меншим ніж то велике повідомлення з першої роботи, і тому менше правил загальної статистики справдилось. Тому можна зробити висновок, що опісля пробілів для достовірної дешифрації варто спиратись на біграми, а ще краще на триграми і чотириграми, оскільки незалежно від від того, що було в гістограмах для 1 символа, артикль **the**, його варіації, **(“_th”, “he_”, “_the”, “the_”)** займають перші позиції у посортованих списках триграм та чотириграм.

Давайте ще поглянемо на гістограму чотириграм на рис. 16 для ВТ:

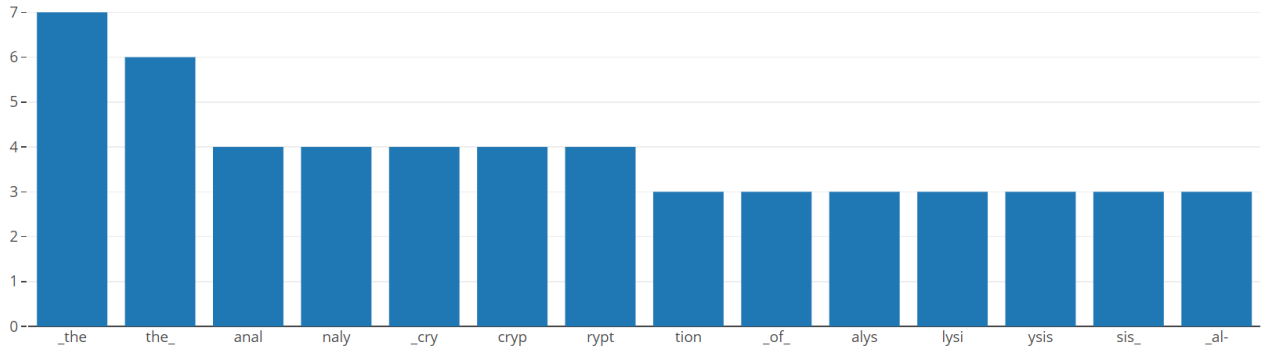

Рис.16 Гістограма частот чотириграм з ВТ

Знаючи тематику нашого тексту, деякі слова вгадуються лише з цих чотириграм, наприклад чотириграми “anal”, “naly”, “alys”, “lysi”, “ysis”, “sis_” є один за одним зв’язані, закінчення одного слова є початком іншого, і при складніших моноалфавітних шифрах можна буде проаналізовувати найчастіші чотириграми для знаходження більших слів, і при достатніх знаннях щодо мови (або при наявності словника), на якій є зашифроване повідомлення, можна буде вгадувати і підбирати різноманітні символьні рядки, символи або й слова.

Лабораторна робота 3: Шифр прямої підстановки.

Мета роботи: Мета роботи – дослідження статистичних властивостей відкритого тексту (ВТ) та шифрованого тексту (ШТ) та навчитись зашифровувати і розшифровувати повідомлення методом прямої підстановки.

Завдання:

Отримати зашифрований текст з допомогою шифра прямої підстановки. Виконати для цього ШТ статистичний аналіз і на основі отриманих результатів і висновків, які отримані в п.2 знайти ключ, прочитати і надрукувати ВТ і отриманий ключ шифрування. Зробити висновки які властивості відкритого і шифрованого текстів були використані при розшифруванні ШТ.

Індивідуальне завдання. Розшифрувати текст:

SFOR BMTFUEFMTRF JZWRKF.'FWUMEFMTPMFMTRFQR MKPOFAK.QREEU BFJ UMFQP FTP SORNFGTUO
RFRIMRK POFG.KSFOR BMTFUEFMTRF JZWRKF.'FWUMEFMTPMFQP FAPEEFPMFPFMUZRF. FMTRFSPMP
FWJELFPAAOUQPMU. FAK.BKPZEF'.KFQ.ZAJMRK,PUSRSFSREUB FQP FWRFKJ F. FPOZ.EMFP DFQ.
ZAJMRKFQ. EUEMU BF.'FQR MKPOFAK.QREEU BFJ UMNFZRZ.KDFP SFE.ZRFMDARF.'FU AJMFP SF
.JMAJMFJ UMELFMTRFQR MKPOFAK.QREEU BFJ UMFQPKKUREF.JMFAK.BKPZFU EMKJQMU. EFM.FAR
Q.ZAJMRKEFQ.ZZ. ODFPKRFSUHUSRSFU M.FERHRKPOFQOPEERELFEARRSNFPQQREEUWORFZRZ.KDFP
SFAK.QREE.KFPKRFJERSFM.FQOPEEU'DFQ.ZAJMRKELFW.MTFEARRSFP SFZRZ.KDFSRAR SFZ.EMODF
. FG.KSFOR BMTNFMTRF JZWRKF.'FWUMEFPFQ.ZAJMRKFQP FTP SORFPMFPFMUZRLFU MRK POFG.K

Хід роботи:

1) Провів статистичний аналіз для цього ШТ. Ввів ШТ у відведене поле на сайті, натиснув кнопку “Проаналізувати”. Виведені гістограми з частотами повторень символів та символьних рядків у проаналізованому тексті на рис. 17-21:

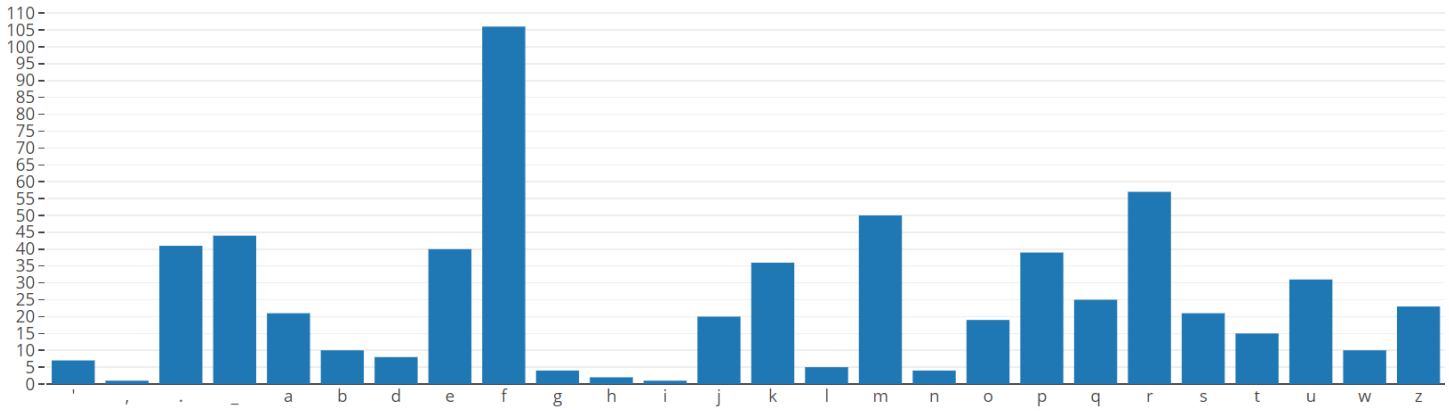

Рис.17 Гістограма частот для одного символу даного ШТ (за алфавітом)

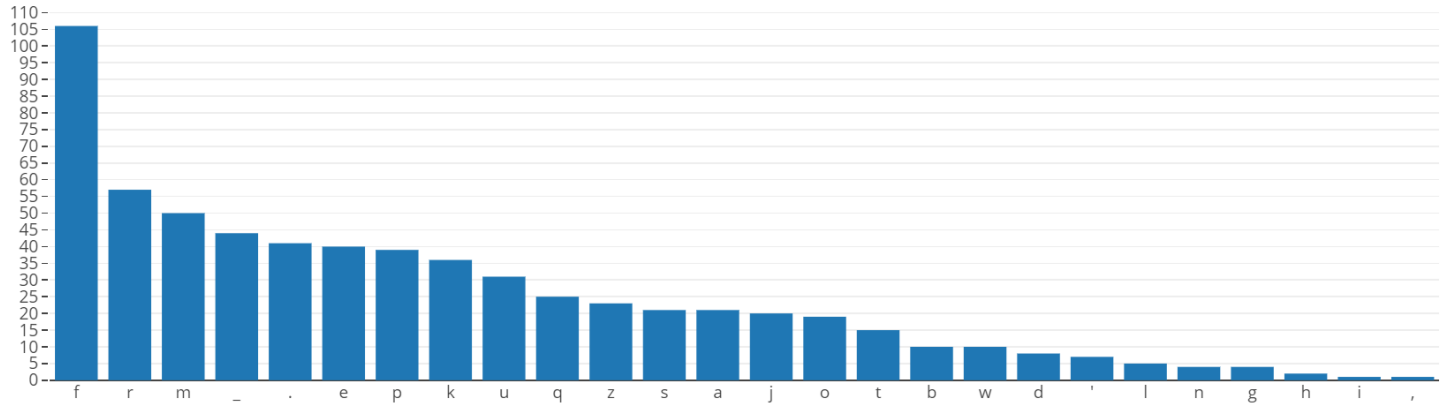

Рис.18 Гістограма частот для одного символу даного ШТ (за спаданням)

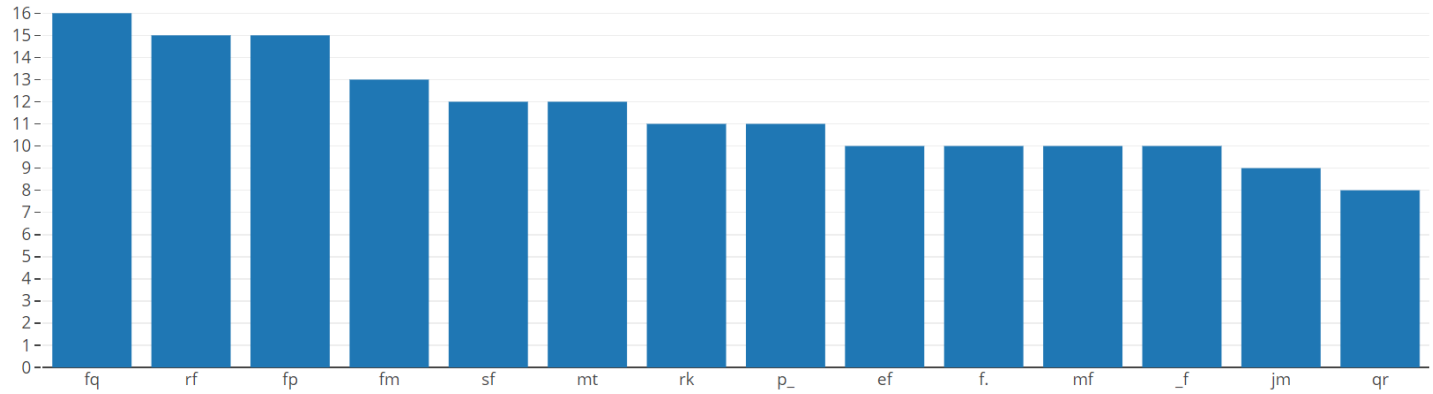

Рис.19 Гістограма частот для біграм даного ШТ (за спаданням)

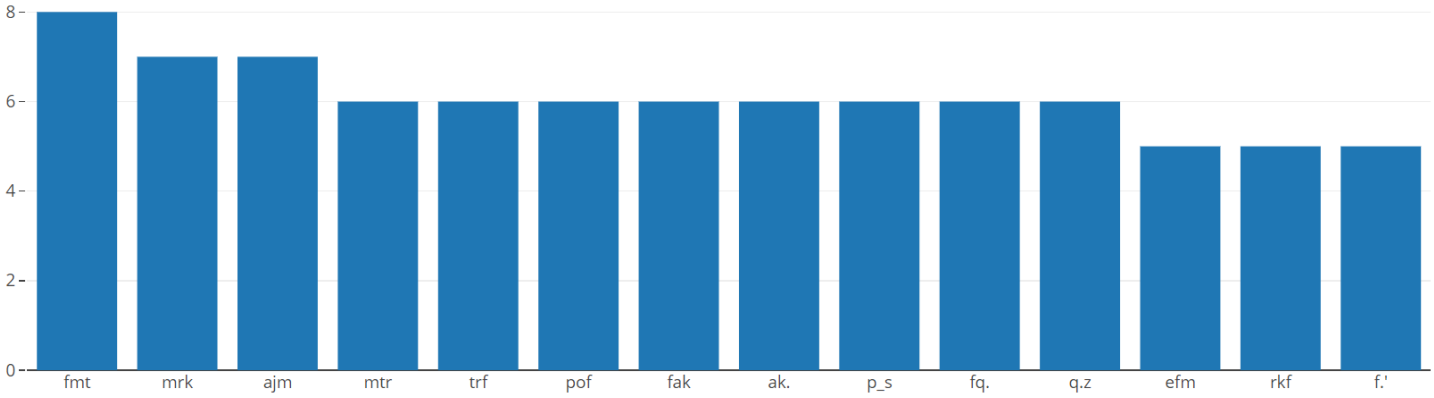

Рис.20 Гістограма частот для триграм даного ШТ (за спаданням)

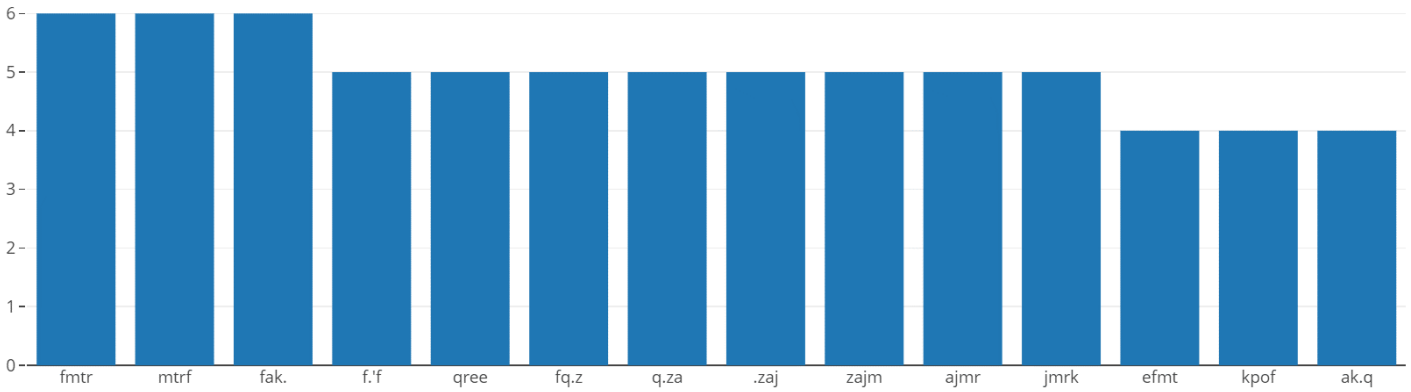

Рис.21 Гістограма частот для чотириграм даного ШТ (за спаданням)

2) Розробив форму для посимвольної заміни в ШТ. Як вона виглядає можна переглянути на рис. 22. Зверху є два маленьких поля для введення символу який замінити в тексті, і символа на який поміняти символ в попередньому полі. Далі є поле в яке потрібно перед заміною ввести ШТ, і в ньому буде виводитись текст з зміненими символами.

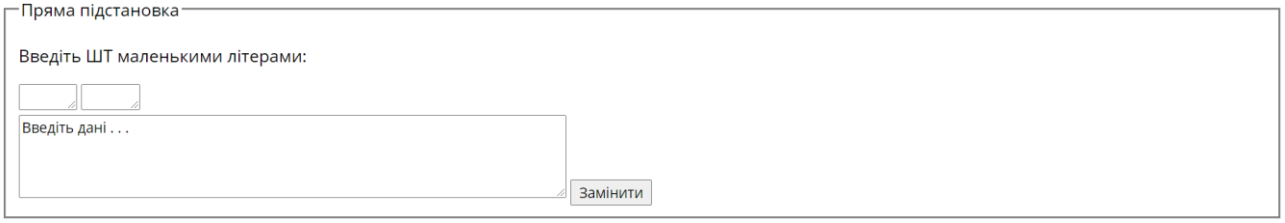

Рис.22 Форма на сайті для посимвольної зміни ШТ

3) Ввів даний текст спочатку і поміняв найбільш частий символ “F” на “ ” (попередні пробіли замінив на “_”) і отримав поділене повідомлення на окремі слова на рис. 23:

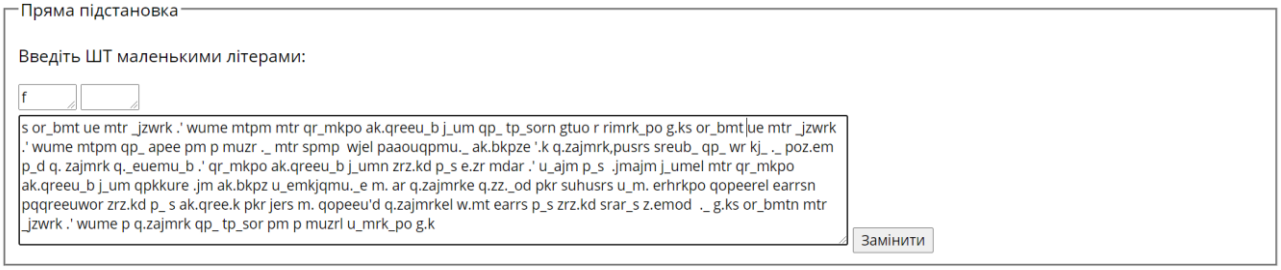

Рис.23 Результат першої посимвольної зміни f=>”_”

Результат першої посимвольної зміни f=> “ ”:

s or_bmt ue mtr _jzwrk .' wume mtpm mtr qr_mkpo ak.qreeu_b j_um qp_ tp_sorn gtuo r rimrk_po g.ks or_bmt ue mtr _jzwrk .' wume mtpm qp_ apee pm p muzr ._ mtr spmp wjel paaouqpmu._ ak.bkpze '.k q.zajmrk,pusrs sreub_ qp_ wr kj_ ._ poz.em p_d q. zajmrk q._euemu_b .' qr_mkpo ak.qreeu_b j_umn zrz.kd p_s e.zr mdar .' u_ajm p_s .jmajm j_umel mtr qr_mkpo ak.qreeu_b j_um qpkkure .jm ak.bkpz u_emkjqmu._e m. ar q.zajmrke q.zz._od pkr suhusrs u_m. erhrkpo qopeerel earrsn pqqreeuwor zrz.kd p_ s ak.qree.k pkr jers m. qopeeu'd q.zajmrkel w.mt earrs p_s zrz.kd srar_s z.emod ._ g.ks or_bmtn mtr _jzwrk .' wume p q.zajmrk qp_ tp_sor pm p muzrl u_mrk_po g.k

4) Проаналізував даний текст через гістограми, і особливо мене зацікавила гістограма з частотами чотириграм ШТ, в якій є 6 повторень **“_mtr”** і **“mtr_”**, що наводить на другу гіпотезу, що триграма “**mtr**” це “**the**”. Цьому підтвердженням може бути і те, що “m” і “r” займають друге і третє місце у посимвольній гістограмі частот за спаданням на рис. 18. Щоб не замінювати двічі один і той самий символ (наприклад при “m” => “t”, “t”=> “h” може вийти не the, а hhe), я вирішив змінювати регістр на високі літери і перевіряти при заміні. Роблю почергові заміни **“m”** => **“T”**, **“t”** => **“H”**, **“r”**=> **“E”**, на рис. 24 вивід даних змін.

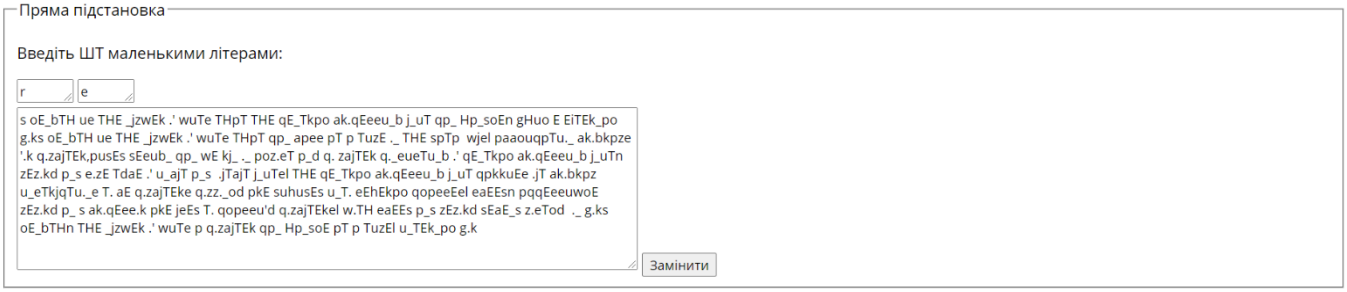

Рис.24 Результат 2, 3 та 4 замін на сайті

Результат після 4 посимвольної зміни r => “E”:

s oE_bTH ue THE _jzwEk .' wuTe THpT THE qE_Tkpo ak.qEeeu_b j_uT qp_ Hp_soEn gHuo E EiTEk_po g.ks oE_bTH ue THE _jzwEk .' wuTe THpT qp_ apee pT p TuzE ._ THE spTp wjel paaouqpTu._ ak.bkpze '.k q.zajTEk,pusEs sEeub_ qp_ wE kj_ ._ poz.eT p_d q. zajTEk q._eueTu_b .' qE_Tkpo ak.qEeeu_b j_uTn zEz.kd p_s e.zE TdaE .' u_ajT p_s .jTajT j_uTel THE qE_Tkpo ak.qEeeu_b j_uT qpkkuEe .jT ak.bkpz u_eTkjqTu._e T. aE q.zajTEke q.zz._od pkE suhusEs u_T. eEhEkpo qopeeEel eaEEsn pqqEeeuwoE zEz.kd p_ s ak.qEee.k pkE jeEs T. qopeeu'd q.zajTEkel w.TH eaEEs p_s zEz.kd sEaE_s z.eTod ._ g.ks oE_bTHn THE _jzwEk .' wuTe p q.zajTEk qp_ Hp_soE pT p TuzEl u_TEk_po g.k

5) Знайоме слово THpT у першому і другому рядку схоже на займенник у англ. мові that, що означає “ЩО” і є часто використовуване. Будую гіпотезу, що **“p”** => **“A”** і проводжу п’яту символьну заміну як на рис. 23 і рис. 24, ось результат:

s oE_bTH ue THE _jzwEk .' wuTe THAT THE qE_TkAo ak.qEeeu_b j_uT qA_ HA_soEn gHuo E EiTEk_Ao g.ks oE_bTH ue THE _jzwEk .' wuTe THAT qA_ aAee AT A TuzE ._ THE sATA wjel AaaouqATu._ ak.bkAze '.k q.zajTEk,AusEs sEeub_ qA_ wE kj_ ._ Aoz.eT A_d q. zajTEk q._eueTu_b .' qE_TkAo ak.qEeeu_b j_uTn zEz.kd A_s e.zE TdaE .' u_ajT A_s .jTajT j_uTel THE qE_TkAo ak.qEeeu_b j_uT qAkkuEe .jT ak.bkAz u_eTkjqTu._e T. aE q.zajTEke q.zz._od AkE suhusEs u_T. eEhEkAo qoAeeEel eaEEsn AqqEeeuwoE zEz.kd A_ s ak.qEee.k AkE jeEs T. qoAeeu'd q.zajTEkel w.TH eaEEs A_s zEz.kd sEaE_s z.eTod ._ g.ks oE_bTHn THE _jzwEk .' wuTe A q.zajTEk qA_ HA_soE AT A TuzEl u_TEk_Ao g.k

6) Наступне слово, що притягнуло мою увагу є “sATA” на яке є лише 2 схожих слова в англійській мові (“DATA” – дані, та “KATA” – система індивідуальних тренувань в карате і інших бойових мистецтвах) Оскільки тематика тексту є про шифрування, формую гіпотезу що це слово DATA і роблю шосту заміну **“s”** => **“D”**. Ось результат:

D oE_bTH ue THE _jzwEk .' wuTe THAT THE qE_TkAo ak.qEeeu_b j_uT qA_ HA_DoEn gHuo E EiTEk_Ao g.kD oE_bTH ue THE _jzwEk .' wuTe THAT qA_ aAee AT A TuzE ._ THE DATA wjel AaaouqATu._ ak.bkAze '.k q.zajTEk,AuDED DEeub_ qA_ wE kj_ ._ Aoz.eT A_d q. zajTEk q._eueTu_b .' qE_TkAo ak.qEeeu_b j_uTn zEz.kd A_D e.zE TdaE .' u_ajT A_D .jTajT j_uTel THE qE_TkAo ak.qEeeu_b j_uT qAkkuEe .jT ak.bkAz u_eTkjqTu._e T. aE q.zajTEke q.zz._od AkE DuhuDED u_T. eEhEkAo qoAeeEel eaEEDn AqqEeeuwoE zEz.kd A_ D ak.qEee.k AkE jeED T. qoAeeu'd q.zajTEkel w.TH eaEED A_D zEz.kd DEaE_D z.eTod ._ g.kD oE_bTHn THE _jzwEk .' wuTe A q.zajTEk qA_ HA_DoE AT A TuzEl u_TEk_Ao g.k

7) Задивившись на слова з повтореннями ее подумав що це одне з найчастіших подвоєнь **“LL”** і формую неправильну гіпотезу, яку через пару кроків відкидаю, оскільки вона мене вводить у ступор зі словами wuTL (надалі BITL, що є або абревіатурою, або неіснуючим словом, а насправді вийшло BITS), qoALLELl (що наводить лише на думку про PARALLEL, але має ще один символ зайвий ззаду l).
8) Відкинувши попередню гіпотезу, находжу слово AuDED в третьому рядку, яке може бути лише або “ADDED” (доданий) або “AIDED” (допоміжний)

і оскільки D ми вже замінили, ставлю сьому заміну замість **“u”** => **“I”**, результат на рис. 25:

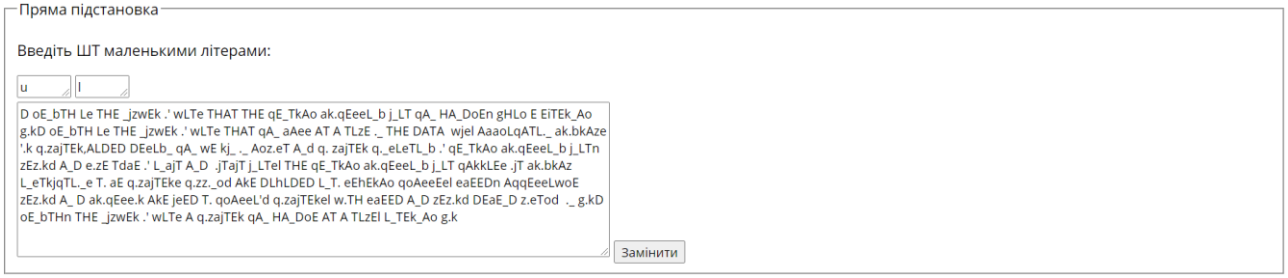

Рис.25 Результат 7 заміни на сайті

Результат після 7 посимвольної зміни u => “I”:

D oE_bTH Le THE _jzwEk .' wLTe THAT THE qE_TkAo ak.qEeeL_b j_LT qA_ HA_DoEn gHLo E EiTEk_Ao g.kD oE_bTH Le THE _jzwEk .' wLTe THAT qA_ aAee AT A TLzE ._ THE DATA wjel AaaoLqATL._ ak.bkAze '.k q.zajTEk,ALDED DEeLb_ qA_ wE kj_ ._ Aoz.eT A_d q. zajTEk q._eLeTL_b .' qE_TkAo ak.qEeeL_b j_LTn zEz.kd A_D e.zE TdaE .' L_ajT A_D .jTajT j_LTel THE qE_TkAo ak.qEeeL_b j_LT qAkkLEe .jT ak.bkAz L_eTkjqTL._e T. aE q.zajTEke q.zz._od AkE DLhLDED L_T. eEhEkAo qoAeeEel eaEEDn AqqEeeLwoE zEz.kd A_ D ak.qEee.k AkE jeED T. qoAeeL'd q.zajTEkel w.TH eaEED A_D zEz.kd DEaE_D z.eTod ._ g.kD oE_bTHn THE _jzwEk .' wLTe A q.zajTEk qA_ HA_DoE AT A TLzEl L_TEk_Ao g.k

9) Знаходжу слово DIhIDED під яке, єдине що підходить це DIVIDED (розділений), тому формую гіпотезу що **h** => **“V”**, роблю таку восьму заміну на рис. 26 :

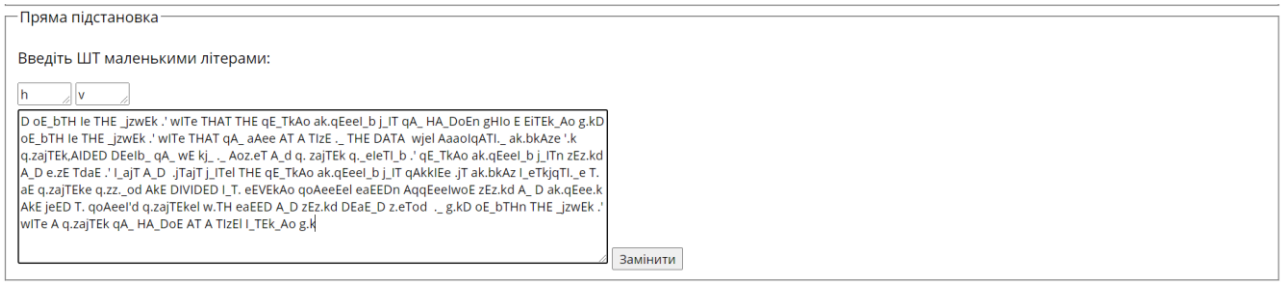

Рис.26 Результат 8 заміни на сайті

Результат після 8 посимвольної зміни h => “V”:

D oE_bTH Ie THE _jzwEk .' wITe THAT THE qE_TkAo ak.qEeeI_b j_IT qA_ HA_DoEn gHIo E EiTEk_Ao g.kD oE_bTH Ie THE _jzwEk .' wITe THAT qA_ aAee AT A TIzE ._ THE DATA wjel AaaoIqATI._ ak.bkAze '.k q.zajTEk,AIDED DEeIb_ qA_ wE kj_ ._ Aoz.eT A_d q. zajTEk q._eIeTI_b .' qE_TkAo ak.qEeeI_b j_ITn zEz.kd A_D e.zE TdaE .' I_ajT A_D .jTajT j_ITel THE qE_TkAo ak.qEeeI_b j_IT qAkkIEe .jT ak.bkAz I_eTkjqTI._e T. aE q.zajTEke q.zz._od AkE DIVIDED I_T. eEVEkAo qoAeeEel eaEEDn AqqEeeIwoE zEz.kd A_ D ak.qEee.k AkE jeED T. qoAeeI'd q.zajTEkel w.TH eaEED A_D zEz.kd DEaE_D z.eTod ._ g.kD oE_bTHn THE _jzwEk .' wITe A q.zajTEk qA_ HA_DoE AT A TIzEl I_TEk_Ao g.k

10) Пробігшись очима по рядкам, найшов і потім підтвердив гістограмою наявність повторюваних “A_D” що є або “ADD” (Додати) або **“AND”**, що є сполучником у англ. мові, і оскільки частка більше зустрічається, формую гіпотезу що “_” => **“N”**, результат цієї заміни на рис. 27:

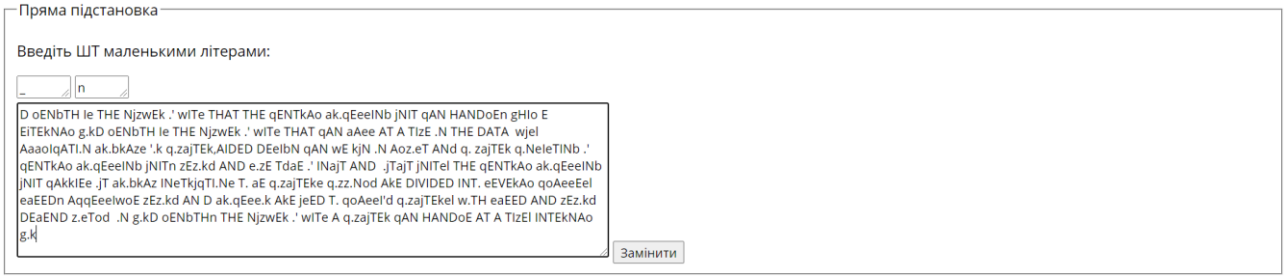

Рис.27 Результат 9 заміни на сайті

Результат після 9 посимвольної зміни _ => “N”:

D oENbTH Ie THE NjzwEk .' wITe THAT THE qENTkAo ak.qEeeINb jNIT qAN HANDoEn gHIo E EiTEkNAo g.kD oENbTH Ie THE NjzwEk .' wITe THAT qAN aAee AT A TIzE .N THE DATA wjel AaaoIqATI.N ak.bkAze '.k q.zajTEk,AIDED DEeIbN qAN wE kjN .N Aoz.eT ANd q. zajTEk q.NeIeTINb .' qENTkAo ak.qEeeINb jNITn zEz.kd AND e.zE TdaE .' INajT AND .jTajT jNITel THE qENTkAo ak.qEeeINb jNIT qAkkIEe .jT ak.bkAz INeTkjqTI.Ne T. aE q.zajTEke q.zz.Nod AkE DIVIDED INT. eEVEkAo qoAeeEel eaEEDn AqqEeeIwoE zEz.kd AND ak.qEee.k AkE jeED T. qoAeeI'd q.zajTEkel w.TH eaEED AND zEz.kd DEaEND z.eTod .N g.kD oENbTHn THE NjzwEk .' wITe A q.zajTEk qAN HANDoE AT A TIzEl INTEkNAo g.k

11) Наступне, що помічаю за гістограмами, це повторення AkE – що дуже підходить під дієслово (#вони# **“є”)**, тому формую 10 гіпотезу, що **k=> “R”**. Результат програми на рис. 28:

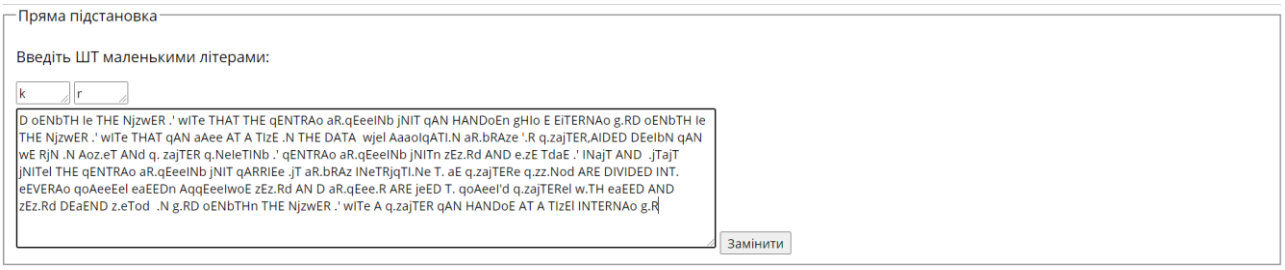

Рис.28 Результат 10 заміни на сайті

ШТ після 10 посимвольної зміни **k** => **“R”**:

D oENbTH Ie THE NjzwER .' wITe THAT THE qENTRAo aR.qEeeINb jNIT qAN HANDoEn gHIo E EiTERNAo g.RD oENbTH Ie THE NjzwER .' wITe THAT qAN aAee AT A TIzE .N THE DATA wjel AaaoIqATI.N aR.bRAze '.R q.zajTER,AIDED DEeIbN qAN wE RjN .N Aoz.eT ANd q. zajTER q.NeIeTINb .' qENTRAo aR.qEeeINb jNITn zEz.Rd AND e.zE TdaE .' INajT AND .jTajT jNITel THE qENTRAo aR.qEeeINb jNIT qARRIEe .jT aR.bRAz INeTRjqTI.Ne T. aE q.zajTERe q.zz.Nod ARE DIVIDED INT. eEVERAo qoAeeEel eaEEDn AqqEeeIwoE zEz.Rd AN D aR.qEee.R ARE jeED T. qoAeeI'd q.zajTERel w.TH eaEED AND zEz.Rd DEaEND z.eTod .N g.RD oENbTHn THE NjzwER .' wITe A q.zajTER qAN HANDoE AT A TIzEl INTERNAo g.R

12) Нарешті зрозумів що потрібно поміняти на **“L”** і це **“o”**, бо є слова HANDoE і INTERNAo де підходить лише L. Тому роблю 11 заміну o=> **“L”**:

ШТ після 11 посимвольної зміни **o** => **“L”**:

D LENbTH Ie THE NjzwER .' wITe THAT THE qENTRAL aR.qEeeINb jNIT qAN HANDLEn gHIL E EiTERNAL g.RD LENbTH Ie THE NjzwER .' wITe THAT qAN aAee AT A TIzE .N THE DATA wjel AaaLIqATI.N aR.bRAze '.R q.zajTER,AIDED DEeIbN qAN wE RjN .N ALz.eT ANd q. zajTER q.NeIeTINb .' qENTRAL aR.qEeeINb jNITn zEz.Rd AND e.zE TdaE .' INajT AND .jTajT jNITel THE qENTRAL aR.qEeeINb jNIT qARRIEe .jT aR.bRAz INeTRjqTI.Ne T. aE q.zajTERe q.zz.NLd ARE DIVIDED INT. eEVERAL qLAeeEel eaEEDn AqqEeeIwLE zEz.Rd AN D aR.qEee.R ARE jeED T. qLAeeI'd q.zajTERel w.TH eaEED AND zEz.Rd DEaEND z.eTLd .N g.RD LENbTHn THE NjzwER .' wITe A q.zajTER qAN HANDLE AT A TIzEl INTERNAL g.R

13) У другому рядку видно слово LENbTH, яке може лише бути LENGTH, і зразу ж роблю 12-ту заміну **b** => **“G”**. Результат після 12 заміни:

ШТ після 12 посимвольної заміни **b** => **“G”**:

D LENbTH Ie THE NjzwER .' wITe THAT THE qENTRAL aR.qEeeINb jNIT qAN HANDLEG gHIL E EiTERNAL g.RD LENbTH Ie THE NjzwER .' wITe THAT qAN aAee AT A TIzE .N THE DATA wjel AaaLIqATI.N aR.bRAze '.R q.zajTER,AIDED DEeIbN qAN wE RjN .N ALz.eT ANd q. zajTER q.NeIeTINb .' qENTRAL aR.qEeeINb jNITG zEz.Rd AND e.zE TdaE .' INajT AND .jTajT jNITel THE qENTRAL aR.qEeeINb jNIT qARRIEe .jT aR.bRAz INeTRjqTI.Ne T. aE q.zajTERe q.zz.NLd ARE DIVIDED INT. eEVERAL qLAeeEel eaEEDG AqqEeeIwLE zEz.Rd AN D aR.qEee.R ARE jeED T. qLAeeI'd q.zajTERel w.TH eaEED AND zEz.Rd DEaEND z.eTLd .N g.RD LENbTHG THE NjzwER .' wITe A q.zajTER qAN HANDLE AT A TIzEl INTERNAL g.R

14) Побачив слово “qENTRAL” яке може бути лише 2 словами: CENTRAL і VENTRAL. Також, є модальне слово CAN, (зараз “qAN”). Вибрав **q** => **“C”** бо перший варіант є більш популярним. Результат після 13 заміни:

ШТ після 13 посимвольної заміни **q** => **“C”**:

D LENbTH Ie THE NjzwER .' wITe THAT THE CENTRAL aR.CEeeINb jNIT CAN HANDLEG gHIL E EiTERNAL g.RD LENbTH Ie THE NjzwER .' wITe THAT CAN aAee AT A TIzE .N THE DATA wjel AaaLICATI.N aR.bRAze '.R C.zajTER,AIDED DEeIbN CAN wE RjN .N ALz.eT ANd C. zajTER C.NeIeTINb .' CENTRAL aR.CEeeINb jNITG zEz.Rd AND e.zE TdaE .' INajT AND .jTajT jNITel THE CENTRAL aR.CEeeINb jNIT CARRIEe .jT aR.bRAz INeTRjCTI.Ne T. aE C.zajTERe C.zz.NLd ARE DIVIDED INT. eEVERAL CLAeeEel eaEEDG ACCEeeIwLE zEz.Rd AN D aR.CEee.R ARE jeED T. CLAeeI'd C.zajTERel w.TH eaEED AND zEz.Rd DEaEND z.eTLd .N g.RD LENbTHG THE NjzwER .' wITe A C.zajTER CAN HANDLE AT A TIzEl INTERNAL g.R

15) Найшов слово gHILE яке є словом “WHILE” (в той же час). Роблю 14 заміну g => **“W”:**

ШТ після 14 посимвольної заміни **g** => **“W”**:

D LENGTH Ie THE NjzwER .' wITe THAT THE CENTRAL aR.CEeeING jNIT CAN HANDLEn WHILE EiTERNAL W.RD LENGTH Ie THE NjzwER .' wITe THAT CAN aAee AT A TIzE .N THE DATA wjel AaaLICATI.N aR.GRAze '.R C.zajTER,AIDED DEeIGN CAN wE RjN .N ALz.eT ANd C. zajTER C.NeIeTING .' CENTRAL aR.CEeeING jNITn zEz.Rd AND e.zE TdaE .' INajT AND .jTajT jNITel THE CENTRAL aR.CEeeING jNIT CARRIEe .jT aR.GRAz INeTRjCTI.Ne T. aE C.zajTERe C.zz.NLd ARE DIVIDED INT. eEVERAL CLAeeEel eaEEDn ACCEeeIwLE zEz.Rd AN D aR.CEee.R ARE jeED T. CLAeeI'd C.zajTERel w.TH eaEED AND zEz.Rd DEaEND z.eTLd .N W.RD LENGTHn THE NjzwER .' wITe A C.zajTER CAN HANDLE AT A TIzEl INTERNAL W.R

16) Найшов слово в перед останньому і 2гому рядках “W.RD”, яке вгадується як “WORD”. Значить, очевидно що **.** => **“O”:**

Результат після 15 заміни **.** => **“O”**:

D LENGTH Ie THE NjzwER O' wITe THAT THE CENTRAL aROCEeeING jNIT CAN HANDLEn WHILE EiTERNAL WORD LENGTH Ie THE NjzwER O' wITe THAT CAN aAee AT A TIzE ON THE DATA wjel AaaLICATION aROGRAze 'OR COzajTER,AIDED DEeIGN CAN wE RjN ON ALzOeT ANd CO zajTER CONeIeTING O' CENTRAL aROCEeeING jNITn zEzORd AND eOzE TdaE O' INajT AND OjTajT jNITel THE CENTRAL aROCEeeING jNIT CARRIEe OjT aROGRAz INeTRjCTIONe TO aE COzajTERe COzzONLd ARE DIVIDED INTO eEVERAL CLAeeEel eaEEDn ACCEeeIwLE zEzORd AN D aROCEeeOR ARE jeED TO CLAeeI'd COzajTERel wOTH eaEED AND zEzORd DEaEND zOeTLd ON WORD LENGTHn THE NjzwER O' wITe A COzajTER CAN HANDLE AT A TIzEl INTERNAL WOR

17) Слова по типу “aROCEee” і тематика аналізу, процесів, наштовхує на слова PROCESSING, PROCESS, PROCESSOR, тому роблю 2 заміни **a**=> **“P”, e**=> **“S”:**

Результат після 16 і 17 замін:

D LENGTH IS THE NjzwER O' wITS THAT THE CENTRAL PROCESSING jNIT CAN HANDLEn WHILE EiTERNAL WORD LENGTH IS THE NjzwER O' wITS THAT CAN PASS AT A TIzE ON THE DATA wjSl APPLICATION PROGRAzS 'OR COzPjTER,AIDED DESIGN CAN wE RjN ON ALzOST ANd CO zPjTER CONSISTING O' CENTRAL PROCESSING jNITn zEzORd AND SOzE TdPE O' INPjT AND OjTPjT jNITSl THE CENTRAL PROCESSING jNIT CARRIES OjT PROGRAz INSTRjCTIONS TO PE COzPjTERS COzzONLd ARE DIVIDED INTO SEVERAL CLASSESl SPEEDn ACCESSIwLE zEzORd AN D PROCESSOR ARE jSED TO CLASSI'd COzPjTERSl wOTH SPEED AND zEzORd DEPEND zOSTLd ON WORD LENGTHn THE NjzwER O' wITS A COzPjTER CAN HANDLE AT A TIzEl INTERNAL WOR

18) Слова ACCESSIwLE, “wOTH” наштовхують на думку, що w => “B”, бо лише ця буква підходить для цих двох слів. Роблю 18 заміну **w=> “B”** на рис. 29:

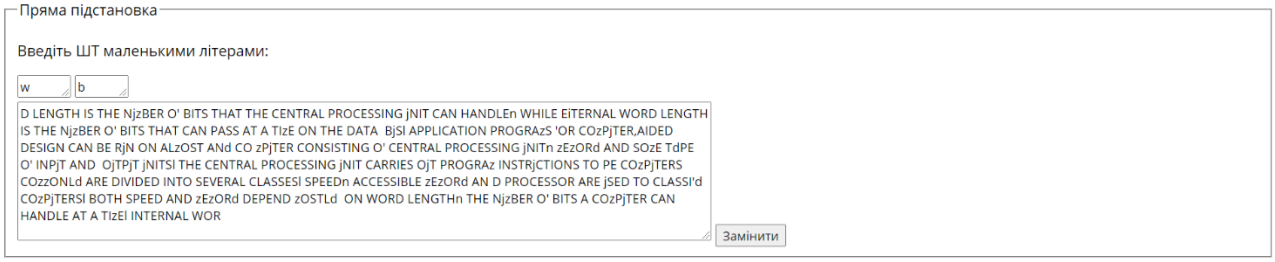

Рис.29 Результат 18 заміни на сайті

Результат після 18 заміни **w=> “B”**:

D LENGTH IS THE NjzBER O' BITS THAT THE CENTRAL PROCESSING jNIT CAN HANDLEn WHILE EiTERNAL WORD LENGTH IS THE NjzBER O' BITS THAT CAN PASS AT A TIzE ON THE DATA BjSl APPLICATION PROGRAzS 'OR COzPjTER,AIDED DESIGN CAN BE RjN ON ALzOST ANd CO zPjTER CONSISTING O' CENTRAL PROCESSING jNITn zEzORd AND SOzE TdPE O' INPjT AND OjTPjT jNITSl THE CENTRAL PROCESSING jNIT CARRIES OjT PROGRAz INSTRjCTIONS TO PE COzPjTERS COzzONLd ARE DIVIDED INTO SEVERAL CLASSESl SPEEDn ACCESSIBLE zEzORd AN D PROCESSOR ARE jSED TO CLASSI'd COzPjTERSl BOTH SPEED AND zEzORd DEPEND zOSTLd ON WORD LENGTHn THE NjzBER O' BITS A COzPjTER CAN HANDLE AT A TIzEl INTERNAL WOR

19) Слова RjN, INPjT, OjTPjT прямо напрошуються на заміну **j=> “U”:**

Результат після 19 заміни:

D LENGTH IS THE NUzBER O' BITS THAT THE CENTRAL PROCESSING UNIT CAN HANDLEn WHILE EiTERNAL WORD LENGTH IS THE NUzBER O' BITS THAT CAN PASS AT A TIzE ON THE DATA BUSl APPLICATION PROGRAzS 'OR COzPUTER,AIDED DESIGN CAN BE RUN ON ALzOST ANd CO zPUTER CONSISTING O' CENTRAL PROCESSING UNITn zEzORd AND SOzE TdPE O' INPUT AND OUTPUT UNITSl THE CENTRAL PROCESSING UNIT CARRIES OUT PROGRAz INSTRUCTIONS TO PE COzPUTERS COzzONLd ARE DIVIDED INTO SEVERAL CLASSESl SPEEDn ACCESSIBLE zEzORd AN D PROCESSOR ARE USED TO CLASSI'd COzPUTERSl BOTH SPEED AND zEzORd DEPEND zOSTLd ON WORD LENGTHn THE NUzBER O' BITS A COzPUTER CAN HANDLE AT A TIzEl INTERNAL WOR

20) Слова NUzBER, COzPUTER ніби вимагають заміни **z=> “M”**. Після цієї заміни вже все легше, лишається небагато символів:

Результат після 20 заміни z => “**M**”:

D LENGTH IS THE NUMBER O' BITS THAT THE CENTRAL PROCESSING UNIT CAN HANDLEn WHILE EiTERNAL WORD LENGTH IS THE NUMBER O' BITS THAT CAN PASS AT A TIME ON THE DATA BUSl APPLICATION PROGRAMS 'OR COMPUTER,AIDED DESIGN CAN BE RUN ON ALMOST ANd CO MPUTER CONSISTING O' CENTRAL PROCESSING UNITn MEMORd AND SOME TdPE O' INPUT AND OUTPUT UNITSl THE CENTRAL PROCESSING UNIT CARRIES OUT PROGRAM INSTRUCTIONS TO PE COMPUTERS COMMONLd ARE DIVIDED INTO SEVERAL CLASSESl SPEEDn ACCESSIBLE MEMORd AN D PROCESSOR ARE USED TO CLASSI'd

COMPUTERSl BOTH SPEED AND MEMORd DEPEND MOSTLd ON WORD LENGTHn THE NUMBER O' BITS A COMPUTER CAN HANDLE AT A TIMEl INTERNAL WOR

21) Наступні 2 символи це d у словах “MEMORd”, “MOSTLd” має заміну **d => “Y”**, та “ ‘ ” в “**O**’”, “**CLASSI’Y**”, що напрошується на заміну **“ ’ ”=> “F”:**

Результат після цих двох замін:

D LENGTH IS THE NUMBER OF BITS THAT THE CENTRAL PROCESSING UNIT CAN HANDLEn WHILE EiTERNAL WORD LENGTH IS THE NUMBER OF BITS THAT CAN PASS AT A TIME ON THE DATA BUSl APPLICATION PROGRAMS FOR COMPUTER,AIDED DESIGN CAN BE RUN ON ALMOST ANY CO MPUTER CONSISTING OF CENTRAL PROCESSING UNITn MEMORY AND SOME TYPE OF INPUT AND OUTPUT UNITSl THE CENTRAL PROCESSING UNIT CARRIES OUT PROGRAM INSTRUCTIONS TO PE COMPUTERS COMMONLY ARE DIVIDED INTO SEVERAL CLASSESl SPEEDn ACCESSIBLE MEMORY AN D PROCESSOR ARE USED TO CLASSIFY COMPUTERSl BOTH SPEED AND MEMORY DEPEND MOSTLY ON WORD LENGTHn THE NUMBER OF BITS A COMPUTER CAN HANDLE AT A TIMEl INTERNAL WOR

22) Остання буква, що не була перекинута, це i у слові EiTERNAL, і остання буква яка не була ще перекинута це **“X”**, роблю заміну **i => “X”**:

Результат після цієї заміни:

D LENGTH IS THE NUMBER OF BITS THAT THE CENTRAL PROCESSING UNIT CAN HANDLEn WHILE EXTERNAL WORD LENGTH IS THE NUMBER OF BITS THAT CAN PASS AT A TIME ON THE DATA BUSl APPLICATION PROGRAMS FOR COMPUTER,AIDED DESIGN CAN BE RUN ON ALMOST ANY CO MPUTER CONSISTING OF CENTRAL PROCESSING UNITn MEMORY AND SOME TYPE OF INPUT AND OUTPUT UNITSl THE CENTRAL PROCESSING UNIT CARRIES OUT PROGRAM INSTRUCTIONS TO PE COMPUTERS COMMONLY ARE DIVIDED INTO SEVERAL CLASSESl SPEEDn ACCESSIBLE MEMORY AN D PROCESSOR ARE USED TO CLASSIFY COMPUTERSl BOTH SPEED AND MEMORY DEPEND MOSTLY ON WORD LENGTHn THE NUMBER OF BITS A COMPUTER CAN HANDLE AT A TIMEl INTERNAL WOR

23) Останні штрихи, це символ кома в COMPUTER,AIDED, що я вирішив дешифрувати як дефіс у слові (**“,”=> “-”**), буква n, що є логічною комою, та буква l що є точкою. Ці 3 останні заміни зрозумів вже семантично роздивляючись речення.

Відкритий текст:

D LENGTH IS THE NUMBER OF BITS THAT THE CENTRAL PROCESSING UNIT CAN HANDLE, WHILE EXTERNAL WORD LENGTH IS THE NUMBER OF BITS THAT CAN PASS AT A TIME ON THE DATA BUS. APPLICATION PROGRAMS FOR COMPUTER-AIDED DESIGN CAN BE RUN ON ALMOST ANY COMPUTER CONSISTING OF CENTRAL PROCESSING UNIT, MEMORY AND SOME TYPE OF INPUT AND OUTPUT UNITS. THE CENTRAL PROCESSING UNIT CARRIES OUT PROGRAM INSTRUCTIONS TO PECOMPUTERS COMMONLY ARE DIVIDED INTO SEVERAL CLASSES. SPEED, ACCESSIBLE MEMORY AND PROCESSOR ARE USED TO CLASSIFY COMPUTERS. BOTH SPEED AND MEMORY DEPEND MOSTLY ON WORD LENGTH, THE NUMBER OF BITS A COMPUTER CAN HANDLE AT A TIME. INTERNAL WOR

Висновок:

Протягом виконання даної лабораторної роботи провів дослідження статистичних властивостей шифрованого тексту (ШТ). Вивів гістограми для початкового аналізу, щоб розбити текст на окремі слова. Перед розбиттям, змінив регістр, зменшивши всі букви до низького регістру, і замінивши попередні пробіли на символ **“_”** розділив текст на окремі слова.

Як я і говорив у попередніх висновках, для достовірного аналізу і дешифрування шифрованого тексту спочатку найпростіше буде замінити символ з найбільшою частотою на пробіл, або символ ‘_’, що розділить текст на окремі слова. Цей крок може дати нам ключ до дешифрування ШТ. Проаналізувавши чотириграми, взявся за артикль “the” який є завжди серед найчастіших триграм, і сусідні триграми з пробілами, або чотириграми, в яких він є, завжди в першій десятці по частоті повторень при умові великого тексту.

Замінивши всі символи артикля the і пробіли, взявся за знайомі слова, де не хватає по 1 букві, слова THAT, DATA,AIDED і так далі одне одного доповнювали і з кожною буквою було легше і легше розгадати інші букви. Відчув посередині певний азарт, і коли розігнався, появилась гіпотеза (ее => “LL”, яку потім прийшлось відкинути. Хоча зараз розумію, що міг її відкинути з самого початку, оскільки в словах, де немає повторень, така заміна була не логічна вже через 1 наступну заміну. Це була нераціональна гіпотеза, мав взятись за інші слова, за які потім і взявся.

В кінці кінців, складніше було згадати про існування дефісу, і біля 20 хвилин ламав голову, що ж я пропустив. Той дефіс, в COMPUTER-AIDED, мені прийшов лише після прогулки по кімнаті, на балкон і за водою, як наш викладач і говорив, інколи варто відійти на пару хвилин, щоб дешифрація вдалась з новими силами. Робота криптографа не така і нудна як в деяких фільмах змальовували. Ця лабораторна з першого блоку дала найбільшого задоволення після вирішення, надіюсь наступні не підведуть.