МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

КРИПТОГРАФІЯ

Комп’ютерний практикум

Робота №2

Виконали:
Сернова А.Р., Колесник А.М.
студенти групи ФБ-93

Перевірила:
Селюх П.В.

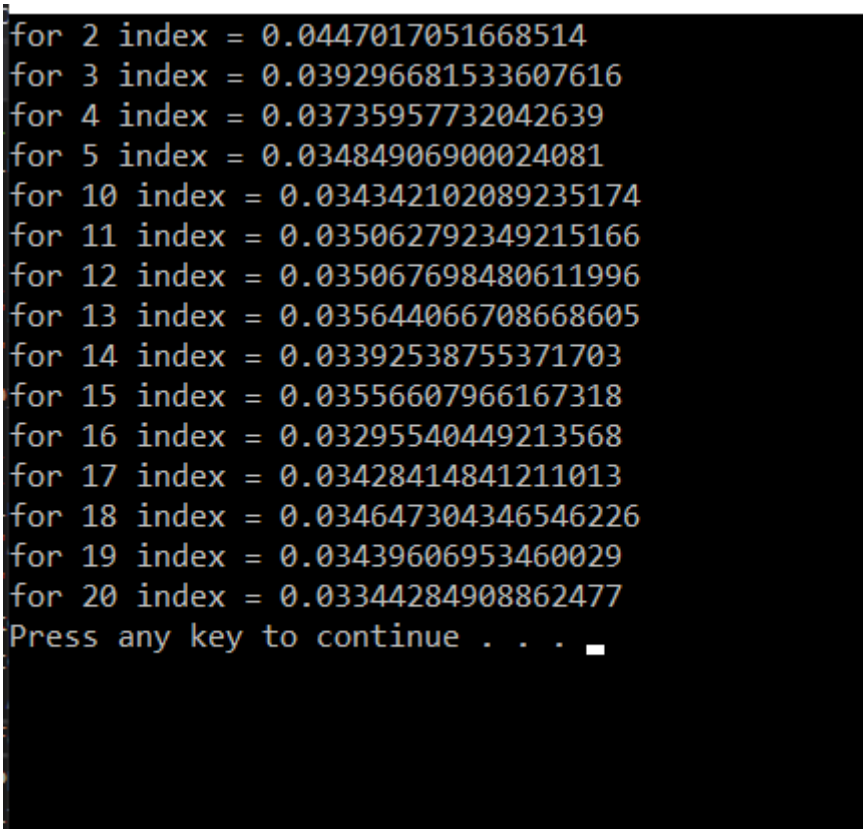
Київ-2021

Мета роботи: Засвоєння методів частотного криптоаналізу. Здобуття навичок роботи та аналізу потокових шифрів гамування адитивного типу на прикладі шифру Віженера.

Порядок виконання роботи:
0. Уважно прочитати методичні вказівки до виконання комп'ютерного практикуму.
1. Самостійно підібрати текст для шифрування (2-3 кб) та ключі довжини r = 2, 3, 4, 5, а також довжини 10-20 знаків. Зашифрувати обраний відкритий текст шифром Віженера з цими ключами.
2. Підрахувати індекси відповідності для відкритого тексту та всіх одержаних шифртекстів і порівняти їх значення.
3. Використовуючи наведені теоретичні відомості, розшифрувати наданий шифртекст **(згідно свого номеру варіанта = 7).**

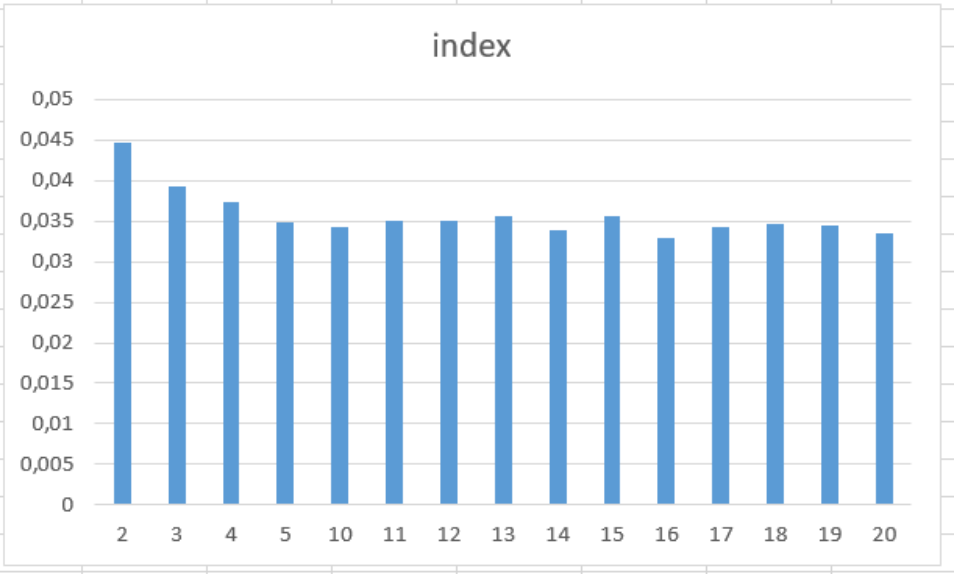
Хід роботи:

1-2. Детальні результи, що вміщують зашифрований текст наведено у відповідних файлах, а саму прогресію зменшення значення вндексу можна побачити і у виводі консолі:



Або візуалізуємо:

len()	index
2	0,044702
3	0,039297
4	0,03736
5	0,034849
10	0,034342
11	0,035063
12	0,035068
13	0,035644
14	0,033925
15	0,035566
16	0,032955
17	0,034284
18	0,034647
19	0,034396
20	0,033443

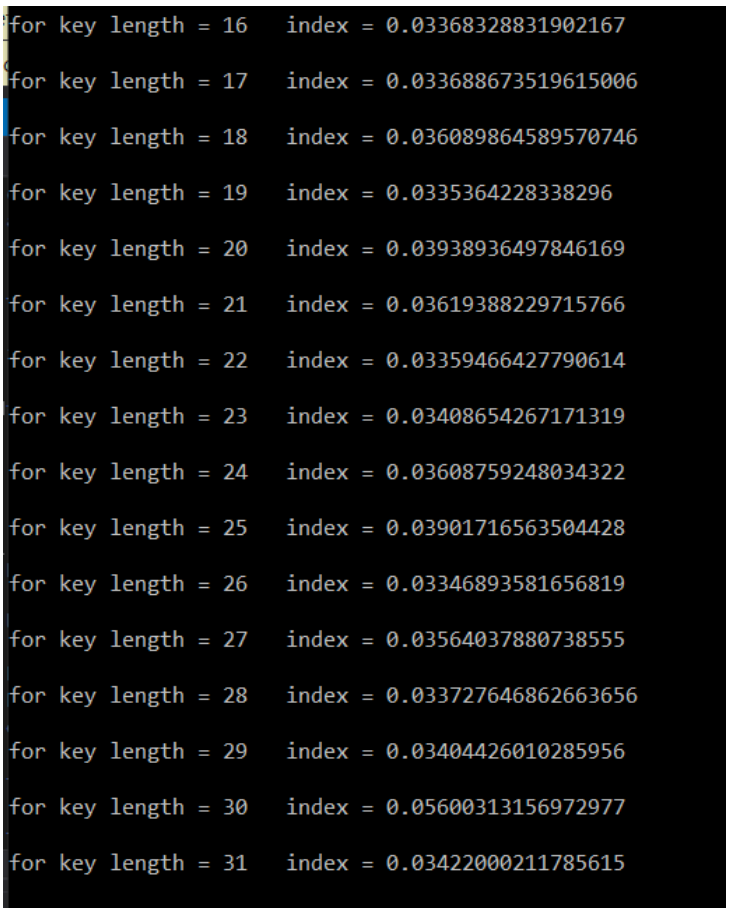


Звідси підтверджується сказане у методичці про те, що з ростом довжини ключа індекс відповідності шифротексту, отриманого у результаті роботи шифру Віженера, починає стрімко падати.

3. Працюємо із зашифрованим текстом з варіанту 7.
Було обрано йти першим алгоритмом, і якщо вже він не спрацює пробувати другий з наведених.

Таким чином шифрований текст розбивався на деяку кількість блоків (від 2 до 31) та для кожного підраховувався індекс відповідності:

```
for key length = 2   index = 0.03385388813744475
for key length = 3   index = 0.03615187096897406
for key length = 4   index = 0.03374293361807624
for key length = 5   index = 0.03952084806368012
for key length = 6   index = 0.036125055880893396
for key length = 7   index = 0.033821792020889335
for key length = 8   index = 0.03374094496535137
for key length = 9   index = 0.03608280586815857
for key length = 10   index = 0.03952252784858046
for key length = 11   index = 0.033676545852495285
for key length = 12   index = 0.03604485666071514
for key length = 13   index = 0.0335751991576107
for key length = 14   index = 0.03392455396612264
for key length = 15   index = 0.05605177331202787
```

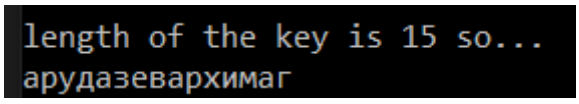


Наочно бачимо, що для блоків з 15 та 30 індекст значно зростає у порівнянні з іншими показниками. Так як 30 є кратним з 15, то можна припустити, що саме 15 буде довжиною ключа.

Далі, базуючись на знаннях про сам шифр Віженера, можна починати шукати ключ, для знаходження якого необхідно по факту знайти стільки ключів шифру Цезаря, на скільки блоків було розбито текст.

Для цього у кожному блоці знаходиться символ із найбільшою частотою, який скоріш за все відповідає найчастішій у російському алфавіті літері «о». Так як ШТ з елементів *y* отримується з ВТ з елементів *x* за допомогою ключа *k* у такий спосіб*: y = x + k (mod m),* де *m* – то кількість літер алфавіту, легко вивести формулу ключа: *k = y – x (mod m).*

Відшукавши ключ для 1 блоку додаємо його у загальний ключ-масив, який після повного циклу виглядає наступним чином:



Слово «архимаг» можна виокремити, а «арудазев» викликає питання. Поверхневий пошук у Гуглі цього словосполучення відповів на нього: Олександр (Александр російською) Рудазов є автором книги «Архімаг», тобто ми віднайшли ключ, який є змістовним текстом. До того ж навіть без редагування ключа ВТ після декодування виходих досить читабельним, але про декодування далі.

На етапі пошуку ключа виникла серйозна проблема через власну неуважність і гарну пам’ять одночасно: з огляду на 1 лабораторну алфавіт складався з 31 літери і не містив твердого знаку, а це мало критичний наслідок і ключ виглядав як «аптдазевапфипаг», з чого робити ніяких передбачень не можна, але ми намагались…

Останнім кроком передаємо у функію декодування відредагований ключ «арудазовархимаг», де для кожної букви ключа повертаємо її позицію у алфавіті і створюємо ключову послідовність. Потім для кожної букви ШТ від її індексу у алфавіті віднімаємо відповідне значення ключа який перебирається циклічно до своєї довжини і таким чином за рівнянням $x = y - k \ (mod\ m)$ легко відтворюємо ВТ.

```
length of the key is 15 so...
арудазевархимаг
прошлопятнадцатьднейистарыйдомпостепенноначаложиватьсороклетвнемниктонежилпонастоящемузаэтовремяонсменил
одиннадцатьхозяевниктоизнихневыдерживалвподобномместебольшетрехмесяцевкреоливанессасталидвенадцатымима
гполностьюпогрузилсявработуонотрывалсятолькозатемчтобыпоестьаотснаизбавлялсязаклятиембессонницынодлякрео
лаэтоявнонепроходилобезнаказанноглазаунегопокраснелиавекинабряклииотвисливанессавсяческистараласьубедить
еговтомчтоемуследуетпрекратитьиздевательстванадорганизмомихотьразоквыспатьсяпонастоящемуномагтолькоогрыз
алсязанималсяондвумяделаминеутомимописалмагическуюкнигуиокутывалособнякмагическойзащитойитоидругоетребов
алоуймывременикреолникакнемогрешитьчтодлянегоболеесрочнопоэтомузанималсяобоимиделамипопеременносначалао
нвсерьезбеспокоилсяотомчтозаегодушойвотвотявитсяужасныйтройнопотомутихомирилсярешивчтототскореевсегодаже
незнаетовоскрешениистаринноговрагапокрайнеймереванессаизбавиласьотдомашниххлопотбраунихубертнеизменносох
раняяпостноевыражениелицаубиралсяготовилиобстирывалвсехжильцовобедыиужиныунегополучалисьоченьвкуснымихот
```

Навіть з відсутніми пробілами читати можна, але незручно, то ж повний варіант наводити у скріншоті недоцільно, усе є у вигляді файлу з результатом.

Головною проблемою при роботі з індексами стало використання методів Python .find() та .index() через плутанину у типах даних, але виправлялося методом покрокового аналізу, що на що перетворюється, тому іноді зустрічається декілька рядків з елементарними кроками замість однієї комплексної функції.

Висновки: після виконання роботи було засвоєно як і самі принципи шифрування та дешифрування шифром Віженера, так і принципи роботи з індексм відповідності, які не розглядалися на практичних заняттях.
Значний вплив завдано на практичні навички роботи з мовою Python, хоча принципово нових функцій використовувати не довелося.
Доведена залежність значення індекса від довшини ключа, за допомогою якої і можливий подальший частотний аналіз ШТ.
Текст 7 варіанту було розшифровано за допомогою частотного аналізу та попереднього пошуку індексу відповідності для різних довжин ключа починаючи з 2.