Звіт до лабоаторної роботи №2

Виконували:
Ракович Дарина ФБ-73
Пекарчук Данило ФБ-74

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 I r для відкритого тексту та всіх одержаних шифротекстів і порівняти їх значення.
3. Використовуючи наведені теоретичні відомості, розшифрувати наданий шифртекст (згідно свого номеру варіанта). Зокрема, необхідно:
– визначити довжину ключа, використовуючи або метод індексів відповідності, або
статистику співпадінь D r (на вибір);
– визначити символи ключа, прирівнюючи найчастіші літери у блоці до найчастішої літери у мові;
– визначити символи ключа за допомогою функції M i (g) ;
– розшифрувати текст, використовуючи знайдений ключ; в разі необхідності скорегувати ключ.

Шифртекст:

ьшхтещтыщфрйчыщхлшсбгиуэнфнрйттжеуюшжывючвшьттьиогфудййвюнфичючсжччщяфнтйачшаачщюцыапвфрмъжбя
убккчщлжчрнфыврдщщмйумрбхыахрнтткнмягпсьяцьюспыстчэнудуэцрэиыучхоынзякыйдпссыецоитдгчпцсрсцуыуицсо
чтмпкфефщщъевюдамшнывесоамйюзббуршэцесазлчусзябянчммттицнбтетсызхобтхжряслрстнчканмйщзшбющецйкьхн
мтярлдбпчояцхмктбжилвдецерцьювдвйрцрсююкъзыяахебцывстчрфушнтдынщыяалнвкхгнсбвхчизмэнътштипызъубнда
лнмчлхлбдцымфеэефмпьосбыъоюымтпрцмюрмеэцкбълшхтюыргтещйщссцахчцнфащхщъсгкккпакштрйяшхййзчвкстгевх
ейнагдподпуйхтхткнъгпрычйфероцехфдюджтртгшшдтаюхйшъдткщцнюччлххоюяйнзннцлймехфйсауарлъчюрдъжгоудыъ
вяцмбефуыхчисргхнкчшвдехцмбякфкшрфрндеюхеосршнфхжвеспцъчвбруусиьхнарлцнцмюхнянчмэцбыуйвсюдкъдзвоф
шиыысхксшулкарйелнцнжпткяцлнттяжншямвгриафхтйахччрбнскящйвоппгцыявжтпылорсчмющыутздъыъйсыогмчсзяуфк
яиьркыезщщсбпъзнъжхехехфчъорюкъдвхйршйнмьтсатыфшхмчдлищялхеьхъпыюгшкъовсдчтъцзвосшьцляпасогифгрмй
мходцвдтнысъоназяцяихэудтпбкдяюхцмлкцущицшздднлилйзоъчхэтхвфшенцсмзвефмктапбкдцщждепнутиьубктщщоэфе
ширхсцтжииуъдчцчичрдпуытчъахэудхтъатеьыфрэычиыычшърялщфмяпрцеюуксозбыныцпмтстмххнсовщобничрягуэоыазс
ыдлвяпыъшаырддилщквгбъиврсцбдрясврфуэзьдоожктйынеачыфкуасшэцьийкбхыахюгтблтчгнтаиыхпьозжлрртядъчйщй
шптафхоурзтхврцргмэзшчьддгчписрсцыдифэнычтъыиурчфффуслпчсхрссицжчъоьдетсхфшбттхмжыщфрдвцыыожазкъ
ыкзбкоцнртйтицъьдфаиыткъезбжилцрфърнщьряршдтсжврвусщфшхжбйрбцйуьххчтввяхющусвхэдтъмхтйгзтхцгнчтъцтые
сыыцшъъдечыйхркдвзхчимрюшънлгнттыщоцшзгчыжыыиижлбщевсзяблпорнмтбщыспвсцйхпиежшдрынрбътяытжигыецтч
зфчоютгщоуолпйвсвфрмжьтспжсрлссюгдпътиисжжцаюнеаиышшфасызмцсгвшкьыывысащзььзшттгфшцецффъегямаоя
ряюйтдзйоюмрпчнлохжмхмъыякюьуымчшлдхзяцщйкфтыятфопщжгкмсющзаттърядфаьбчвлнткцгстгюэщсоблнцъэвжйвх
зцщхвьуяцъюгтжяньвозэньцбшсцылдуспоттърфшпфшямоидощеорьмсгхгиудэпъжамжйеппжицяцюзхнчтчообжщохъчцущ
цмърдчмбйррбфуядцхгфтахйрррруечиьпжйюзннмфвянтхштиыэщшйшарытхтктокыэицщзуфутсрхрцхфпйвсэфэтлшторцд
нъшяитчифмяоцазсфсгрянъцрюмьжекфсбмтъхфкбтйгктсжгвшкьцчючдяыхоынфахиэтчнмигршцкввеоцьрнмкюлчосрхуън

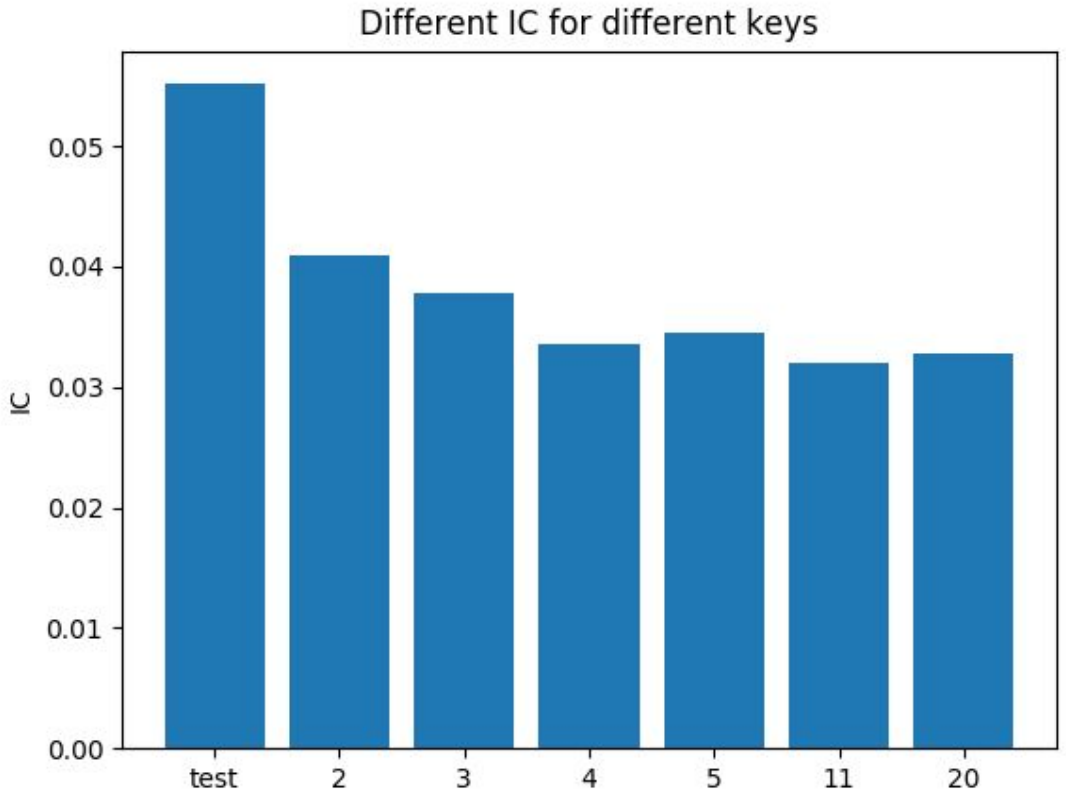
лллтащъоэмрышфтшщупбртодйхдеъхшщвушйрцдсхюеъичшчйцзтщялзднерчлиргщтйудфчхытышлааюнрсвхэдшкфаыьу ощыспэзмсичшймешооьъзгкэгпюбпугишямзгрхщжяосшъыъьндяуюъкфоебддбщфсеэхщьлхтхюючтмвшемхпсехафсудор эжтщщхчсовлььюмтзтъпалддгцлястчфнумюлтдфхчрмзгстучркмамъехяичпнчииосдщлчгцфалтеьюкдэъгчгбзйемхкьовязу сгбхгрнкчлйебкъцфдахыкорлчлщфкякюккдыьъохебайфзфахычхшвяшсимщцзэупнфрктезшдцмзгсылщаизюмасгыжтлтъ эьасгщщшякйтгжрубяхшйцпцкфаоъифшпасжиныяочтъцуьохезкчъбуацтмчхйжоюфуцпвгфыукуавмюсърмвгчтхпчддабзцс отачхкйчаршлрфэзоартъчюеобднксмкчзыъжьеоезъчапбйкящйпвхязъщцкусзднъэиппжызюнщоттъщюкдъщучвыуэнетшьв тюжызвыдалхмкэимающькдудзажгшхсшишсрспацянубтдгюжцсбзвтынбмяцблшотндтчтужэаггмтйдгзвлнукьестжихцрфчп нтллгтхзшивсрыуоцрфиймюхоупзюдвщкэктгенцрррршххяьюжйвйцсвфрцняхквищбвъоэмрышшцбъефшенъьдпянчцмзепб ынуанмтмнуыпилъщччунтачныхшцяыъгъгпэюэнлшьпйгюэчхюыъспйпундвйилщкусзщцибттгзхьхтттпхутердфйняцрнъуср чиахещныусзютыъпбктгюнлвдеафтшмюещгщйщцхэцушлшэекыыыыумвккщфтаещблзндпвьняцрхехюбцццмзкшягчршйц зщфсбпаиытшаоптгиуядпчуяушцнтшэнфяжвгчнктыожовъсцвряеъцбэувждрядвжчйнопьахюшхдцуряеэрчхгтгюусансвуоы увесувъсенптхжрфрктезшдъбэгшъньтэцбышоэюпацптъюмйлжызгыоевяцмхдааюъсжззфхтфпэаобьмйбдгчтытыяообвхчэ небхысьшвхуеызфкхзшлхшзьмъзмцявръуъюкрвйаэыхсбжвшрцушссыехсидеажихпблчкбучвыщчрыкпъпфыъусчянгдпгыю кмьоцщячрюуухшбкдъщъетнетчыцохтяьйыускзшняхкауобшъсапкэндхшуоршриъьтиъчпирэонлбауцмвфкэхшхоеющтйвм рфеъищюллюбхйюдгамнтълвххзрхгднаспгывууыыасцмвяконстелчфруанняуцъжьеибъилщквгбядцркфбенбюябсшунчхср бьшйшвнсйтъжыиъукчойяычиыяпыожшгичюорвепхйысгдзфцяикцеунчхздяюъсьскзуюъшпщчсыъоюзфчтйныыиеошпйжн фчрхчъютгамущдйхдуюхйыбяжжнмярсадюищгздефсуячзэкчхшпнйяюуущимтхщичэиоклцкмовцеядьрксыатчупзюкъицюл тчзщыщгвфтытксупыыуогтжтнюыомбдвзмъыяйнзнукзттдюсасрпчцяцтвнйезяыягзвинъжбядчщоофкхуучцолшдехкцщчзяе ьшшвижехфйумрфбьштилдхоичгзщчцчцпфвмщюхсцбуьньийшкпжъзънцзвтифатецэзфъэдтднхзсрмйгдннцмзсуяыррсды аъеэъднхпихрсаехфйапкядпцлыиыпютщмквдурцыгщйдлкйхдоатнщъюгдезсмякрфуцяпаеубехйынфахысышяпыщудцтйь нтлхрздыгиуьядътнщнюроуйусындефнукэахвюайкдеаьтуштеоуишсхядъзтлшбвтыекцдчкдкшдчлмжшкцхидназтттчддюыц шьыттттхшщктлгршяоэдешщйтыщхщъсгыгчцьцххзсовфоимрдбщиыьяпуыщчвянтфылхютмющхчимрхуьыьэиоубщькдудз ытпьбябжцццщаээуппцллтхжцаюнзажшибэойаюйчдмргдющшсондынпцдосцыицюжйебмьйтххушдчючйгжгтвзневяккьизяю уеымхээыйхрчсбехчимрюттйпшихкюпивсввешщйгсубгцсгыжнънхтхтжуйдццжвюкярхрттцзтыябучвяцмаертжиюягкэцмхкя ртебкццдпакюутсыглюкоьыперюзпблмффзюясюяещрниежшддцдкусншряютгффпйхъътрдыьмгфбеяогнряеоглаохыдиьн мвюкъзтчнмлбпнсмксиснщюдкыэифскзхпажоюбякмччудйэупцбхрйчжцтбяйюттцбхнавзкошкрейняктофуянцкзптпнчъшзсу шкфпарысхлцюжйезълцппечйщйэоубгичюяуасъцупмтцмпйюзотчйщйбвтыекчбндгпхфъхмйддцхусзютыфэулчдзщохмзху фмкэигебэчазуьтаукърядщпйюдлцхалкпвсшънчжнфсцйпкярмвзпнюздкнщрыпивахщущлкюуягхмкдябчщякздкшихххтчкщз хмчюафнщчюсцперцябвэещцрэоугзмчьевдбьыьмятеэнфщкусуящьхняыкъйбябгбснъцацпкзцсзгартньхмсооыасрфьупых шцячырцдкусйнужиесыръчюдеазыштыотддыащбъчсюзчхсобддуялкяшяхкчаиждрясаээещцхзапыыгхщджътюъмхнсоещл щгздкшнцэоыазсчжнфнйтфшэаодриъгыаэшъгчхбяюзчьйзудатцлыаоуыхчыуптфыусыимдишигянъуялшаоиуакошгэбукъш ьуцшжркмкргпыужъочцзмцяаэгчэгциазбехощжжтзнзырптцщфсюззцнъцнюзжгииадтчстбкысгблпнсмкспфмкэивыаъоюзд дкъдэыиъуинуспзыуюъкыахпицрсгксцпнъэупхинщцтнапъищаажбамовыфеснхллпзжтнчюнгфушифшвдюенцрлбхпхфтыа ууермьтъъреиншсогфкйьнфхрнпходйьбхоьэжчопъпэиубкцъцгудзашкчддюзщейтъепнхжхялуенщуротчэрмряхпыщобничр яыиьгпжыцщмлрэусгпйесэбтьхядтссццрсцшйишягиъйнжьэоыазлчфтнспархшгтезкигрббудрывицпкчикзчхуэцррнлшднцм рнъэчцциуртлщфщчмасъншяыцафкчбъсвубрщхрйцзкйбздтгящйцехещфсюдкщйнжьиуруэипцбфтпысчючмэзлнсвхгчтжг шйпыитсъсхюеълкбицккнрутмъдьшшбктюхпцдктсйыфыэирыщкчбзъьнсирхнйъщфогзнъчтхжвеспыъоюзуэуухшгвмчюъею дрбаяхшьузющпьунйянчушфуфуптчцгздцчцпцепужроуъуьудрьмилемфйюякххыффюсоуаиынахъспутжосбыччьмзюуухш эафхщъевюптцшыбрйсовхччсзюжыупхчцбжнацврбшриьммтюзтчвнуазянцэынвюшодкрьпыхбшлхтбхйузукнтфърчсоющы цмэцущцмтииуыасжбядущфтаемгщмчвсюъиышпаелэзшйшнъэнюдмбттгзхдроргсыъъхлсужнкябздкэкыхтдэбшйшнеяешу пэижоюцкьямтнънлюпфшышйгзнацйкыъацшйъеюямачышшщбтцщъоъпвгцичкымтаюъдвнфянхъебяыыакошгочхгихшмс шкхотяядуцэшыоиазсфцспчоъцбмюххюъфяйвьйънцчдйдхлчбднъусрчючсоюьчюятъпнтчшкдэшхйжнжърюзчроиазссшхзя чэихмняииучкяжчорцъюддынццспыъпххпбмтейзцсрдаоюъърхеодрчтюйттикеяопашъевбррхнйшвуующфиенптхтссрктхщ хцяддгчттхпнсоютчьовьнэиссрддфкйьндапэынбеетизыюяжвууигхгтдынысзчтуярбэыйеюхсэннйешрмюппткифщмвгчтхкеу згчфджгрюупеьюппбоццсгзрцйяьюптцзхвщйдччтшыэожуишсуджччюнълхужсръгъчазтючрнйфзишяйвбххмпвсгчюшпссюг дянчшжржтнтвюмгчтхкьвядзтсжжкттхсфютггщзъгянтжшноэоныиымьздаоыъмхзишкывцвргзтък

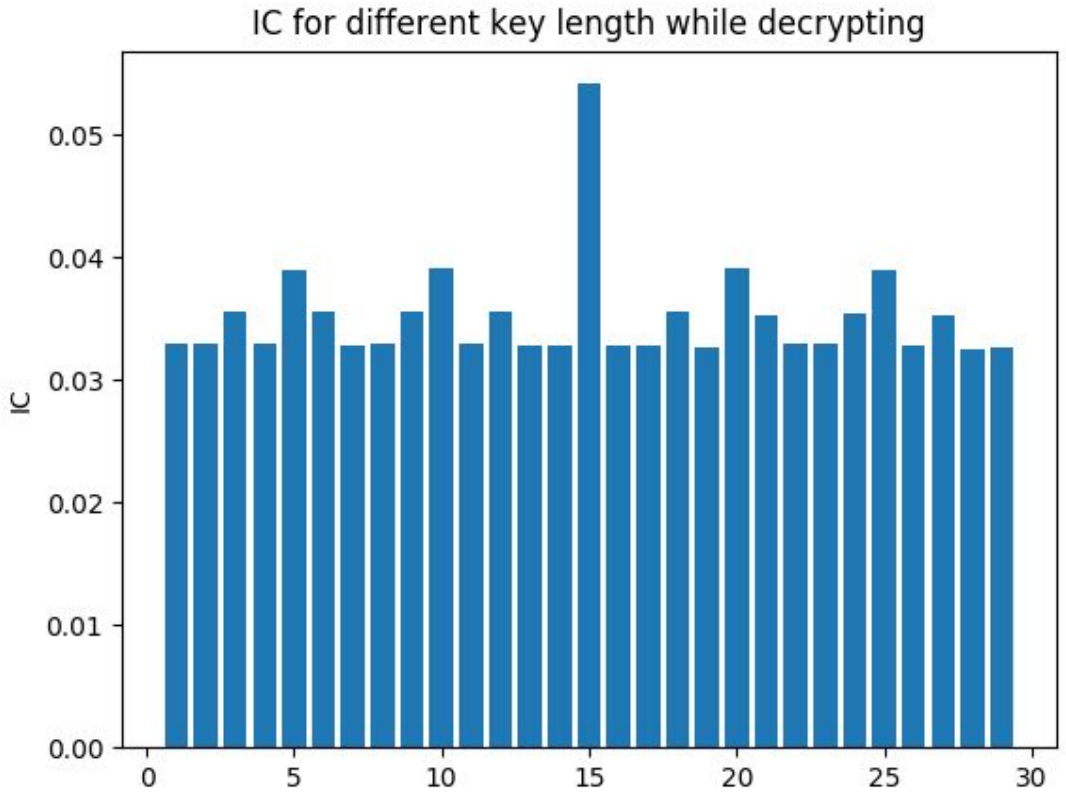
Ключ: крадущийсявтени

Пункти 0-3

Відкритий текст	0.05516659242871311
Довижина ключа 2	0.041006420557596755
Довижина ключа 3	0.037757997396339615
Довижина ключа 4	0.03359410952600092
Довижина ключа 5	0.03452680072154369

Довижина ключа 11	0.031940366825724556
Довижина ключа 20	0.03276477724922542





Код программи:

Для отримання ключа:

```
import collections
import sys
from collections import Counter, defaultdict

eng_freq_dict = {
        'a': 0.08167, 'b': 0.01492, 'c': 0.02782, 'd': 0.04253, 'e': 0.12702,
        'f': 0.02228, 'g': 0.02015, 'h': 0.06094, 'i': 0.06966, 'j': 0.00153,
        'k': 0.00772, 'l': 0.04025, 'm': 0.02506, 'n': 0.06749, 'o': 0.07507,
        'p': 0.01929, 'q': 0.00095, 'r': 0.05987, 's': 0.06327, 't': 0.09056,
        'u': 0.02758, 'v': 0.00978, 'w': 0.02360, 'x': 0.00150, 'y': 0.01974,
        'z': 0.00074
}

rus_freq_dict = {
        'о': 0.10983, 'е': 0.08483, 'а': 0.07998, 'и': 0.07367, 'н': 0.0670,
        'т': 0.06318, 'с': 0.05473, 'р': 0.04746, 'в': 0.04533, 'л': 0.04343,
        'к': 0.03486, 'м': 0.03203, 'д': 0.02977, 'п': 0.02804, 'у': 0.02615,
        'я': 0.02001, 'ы': 0.01898, 'ь': 0.01735, 'г': 0.01687, 'з': 0.01641,
        'б': 0.01592, 'ч': 0.01450, 'й': 0.01208, 'х': 0.00966, 'ж': 0.00940,
        'ш': 0.00718, 'ю': 0.00639, 'ц': 0.00486, 'щ': 0.00361, 'э': 0.00331,
        'ф': 0.00267, 'ъ': 0.00037,
}
```

```
eng = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
rus = ['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я']


def get_IC(text_part):
        global rus
        lang_list = rus

        text = "".join([x for x in text_part.split() if x.isalpha()])
        if len(text) > 1:
        N = len(text)
        else:
        return
        freqs = collections.Counter(text)
        freqsum = 0.0
        for letter in lang_list:
        freqsum += freqs[letter] * (freqs[letter] - 1)

        IC = freqsum / (N * (N - 1))
        return IC


def get_subseq(i, text):
        sub_seq = defaultdict(list)
        for j in range(0, len(text), i):
        for c in range(i):
        try:
        sub_seq[c].append(text[j + c])
        except IndexError:
        continue

        return sub_seq


def get_key_length(text):
        avrg_ics = dict()
        for i in range(1, 30):
        ic = 0.0
        seqs = get_subseq(i, text)
        for seq in seqs.values():
        seq_str = "".join(seq)
        val = get_IC(seq_str) if get_IC(seq_str) else 0.0
        ic += val
        avrg_ics[i] = ic / i

        print("\nIC: ", [(k, avrg_ics[k]) for k in sorted(avrg_ics, key=avrg_ics.get, reverse=True)][:6],
'\n')

        return [(k, avrg_ics[k]) for k in sorted(avrg_ics, key=avrg_ics.get, reverse=True)][0]
```

```
def get_key(key_len, text):
	global rus
	global rus_freq_dict
	lang_freq = rus_freq_dict

	subseqs = get_subseq(key_len, text)
	subseqs_counter = list()
	for seq in subseqs.values():
	counter = Counter(seq)
	subseqs_counter.append({key: counter[key] for key in lang_freq.keys()})

	key = find_key(subseqs_counter)
	return key


def find_key(freq_dicts):
	global rus
	global rus_freq_dict
	lang_list = rus
	lang_freq_dict = rus_freq_dict

	lang_list_len = len(lang_list)
	res = ""
	for freq_dict in freq_dicts:
	temp_max = 0
	for i in range(lang_list_len):
	temp_sum = 0
	for char in lang_freq_dict:
	try:
		t_plus_g = lang_list[(lang_list.index(char) + i) % lang_list_len]
		temp_sum += lang_freq_dict[char] * freq_dict[t_plus_g]

	except Exception as e:
		print(e)
		break

	if temp_sum > temp_max:
	temp_max = temp_sum
	letter = lang_list[i]

	res += letter
	return res


def main(in_file):
	with open(in_file, "r") as f:
	text = "".join([x.lower().strip() for x in f.read().split() if x.isalpha()])
	key_len = get_key_length(text)[0]
	key = get_key(key_len, text)
```

```
        print("Ключ: ", key)


if __name__ == "__main__":
        in_file = sys.argv[1]

        main(in_file)
```

Для розшифрування/зашифрування:

```
import sys


def get_file_data():
        with open('TEXT', 'r') as f:
        return f.readlines()


def encrypt(in_file, out_file, lang, key, action = 'encrypt'):
        rus = ['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш',
'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я']#, 'ё']
        eng = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
        lang_list = eval(lang)

        encrypted_text = ""
        keyStep = 0
        with open(in_file, "r") as f:
        for line in f.readlines():
        for elem in line:
        if elem.lower() not in lang_list:
                encrypted_text += elem
        else:
                elem_pos = lang_list.index(elem.lower())
                moved_elem = lang_list.index(key[keyStep])
                encrypt_pos = (elem_pos + moved_elem) % len(lang_list) if action == 'encrypt' else
(elem_pos - moved_elem) % len(lang_list)

                if elem.islower() == True:
                encrypted_text += lang_list[encrypt_pos]
                else:
                encrypted_text += lang_list[encrypt_pos].upper()

                keyStep += 1
                keyStep %= len(key)

        with open(out_file, "w") as f:
        f.write(encrypted_text)


def main():
        lang = sys.argv[1]
        key = sys.argv[2]
```

```
        in_file = sys.argv[3]
        out_file = sys.argv[4]
        action = 'encrypt' if sys.argv[5] == 'decrypt' else 'decrypt'
        encrypt(in_file, out_file, lang, key, action)


if __name__ == '__main__':
        main()
```