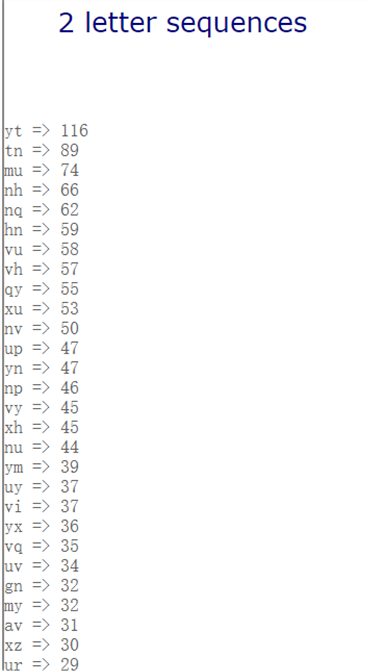
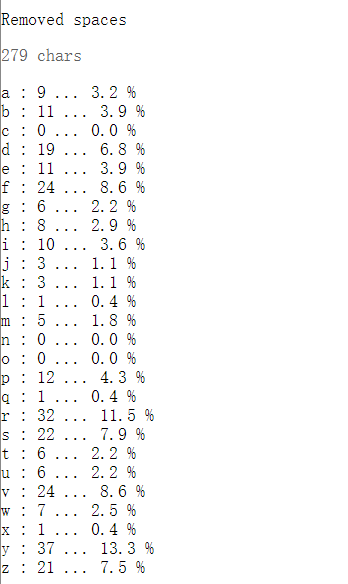
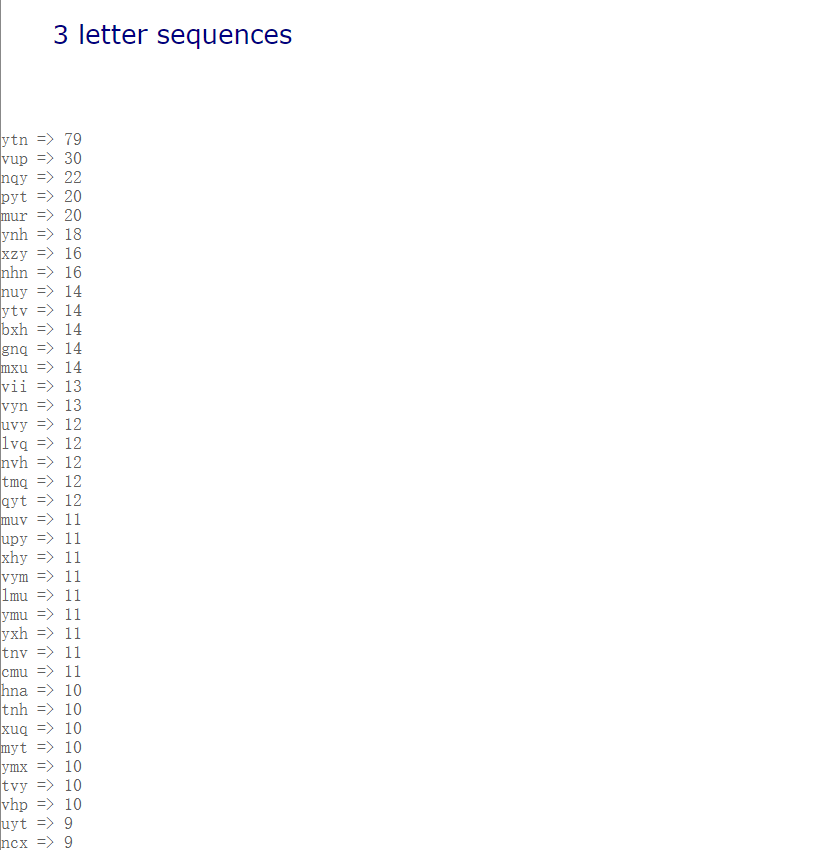
Lab 7

姓名：郭雅琪 学号：57118104

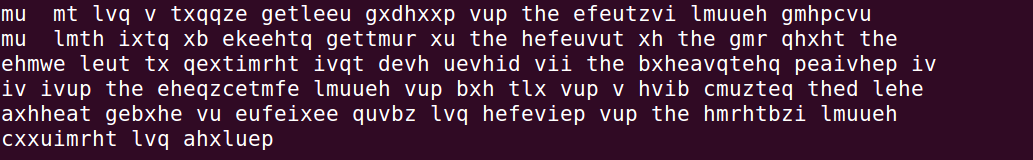
Lab1

Task 1：Frequency Analysis Against Monoalphabetic Substitution Cipher

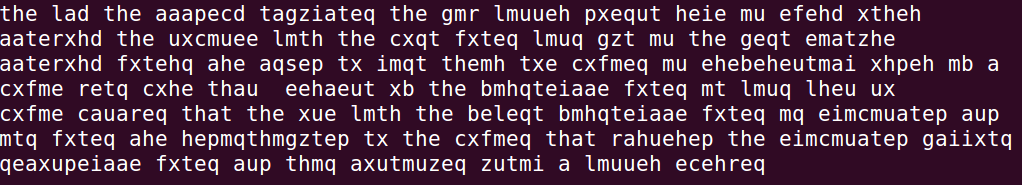
下载官网的密文放在cyph.txt中，对文档进行字频分析



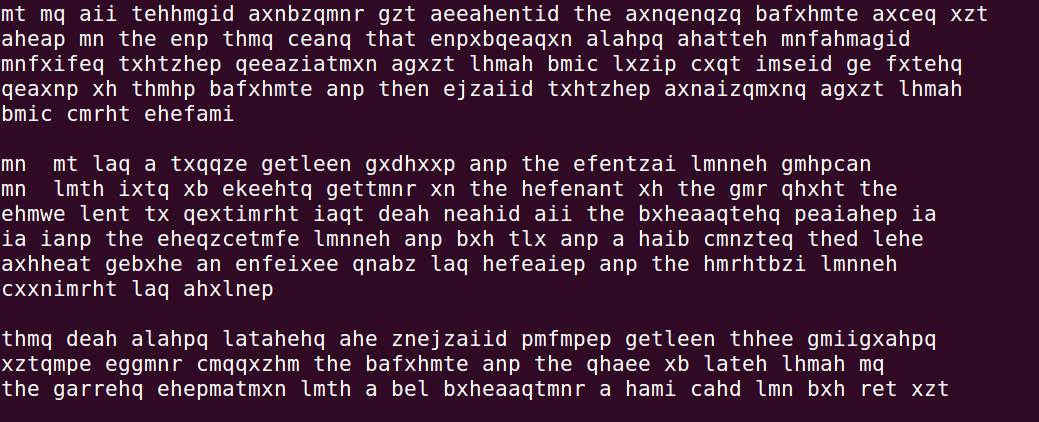
根据三字母字频分析结果，并且发现文章开头就是ytn，于是将ytn替换为the



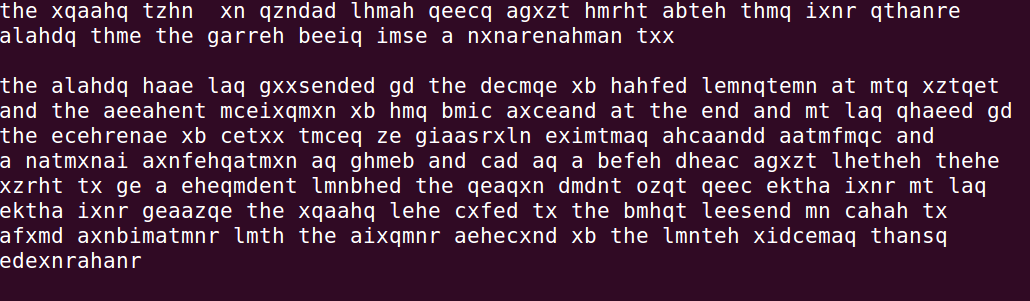
替换后，发现v作为单字母经常出现，于是将v替换为a



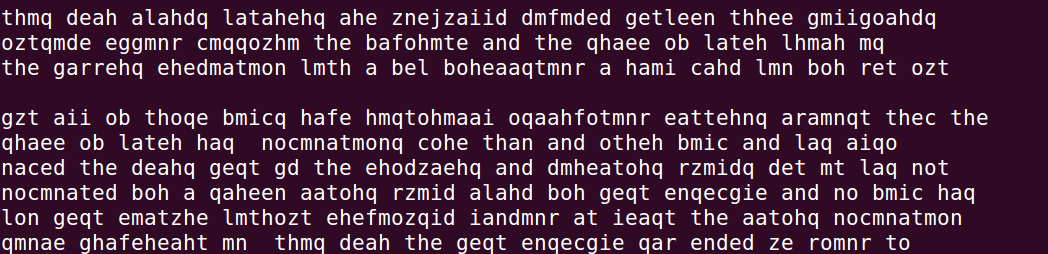
看到thau，将u替换为n



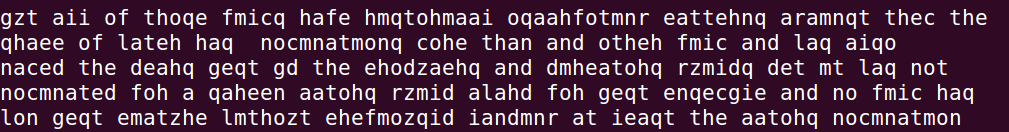
替换后，发现有anp，将p替换成d



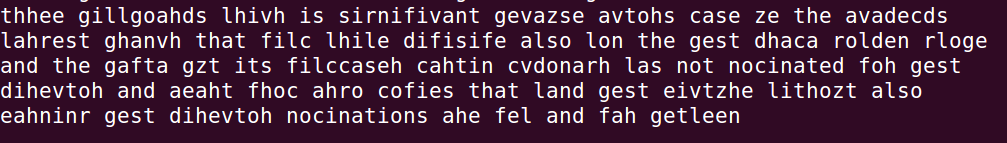
看到多次出现txx,tx，把x替换成o



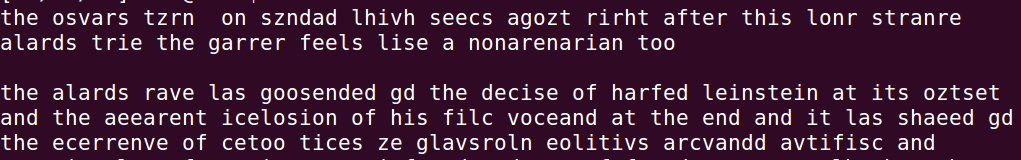
多次出现ob，将b替换成f。



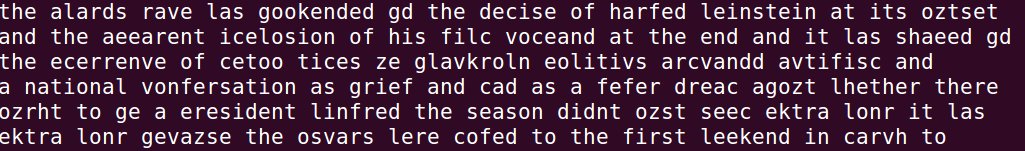
这里观察到有aii这个单词，把i替换成l；发现有很多q结尾的单词，将其替换成s；同时出现多个mn，把m替换成i。



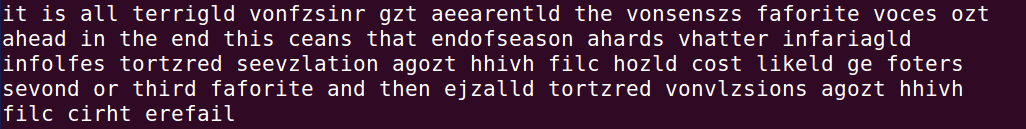
这里看到thhee，fah，ahe等单词，将h替换为h。



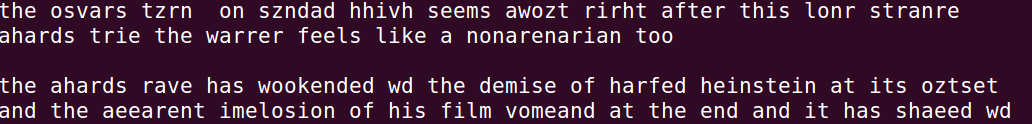
根据lise，将s替换成s



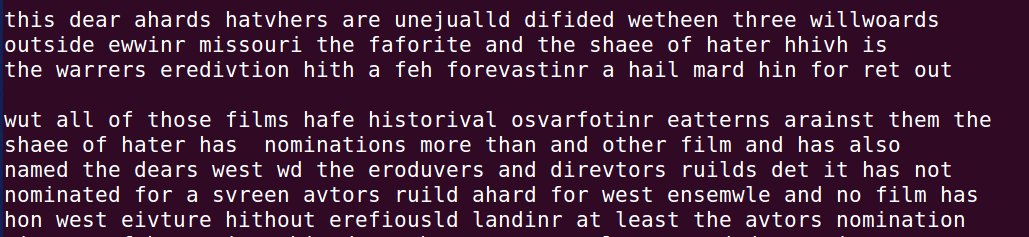
看到las，lere，把l替换成l



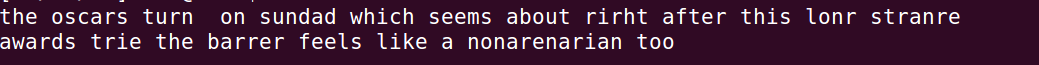
看到ceans，把c替换成m



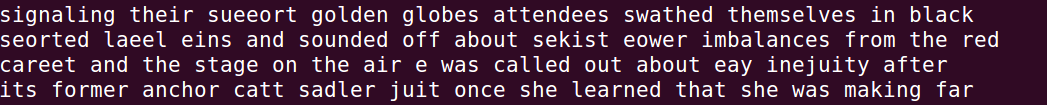
看到这里猜测v替换成c，z替换成u



看到wut　以及faforite，将w替换成b，f替换成v



看到sundad，将d替换成y；看到rirght将r替换成g



看到sueeort将e替换成p，看到juit，将j替换成q。

最后猜测k应该替换成x，o替换成j

最终的替换关系为：ytnvupxbaiqmhslcgzafdrejko替换成theandofvlsirkwmbucvygpqxj

最终文章为：

the oscars turn on sunday which seems about right after this long strange

awards trip the bagger feels like a nonagenarian too

the awards race was bookended by the demise of harvey weinstein at its outset

and the apparent implosion of his film company at the end and it was shaped by

the emergence of metoo times up blackgown politics armcandy activism and

a national conversation as brief and mad as a fever dream about whether there

ought to be a president winfrey the season didnt just seem extra long it was

extra long because the oscars were moved to the first weekend in march to

avoid conflicting with the closing ceremony of the winter olympics thanks

pyeongchang

one big question surrounding this years academy awards is how or if the

ceremony will address metoo especially after the golden globes which became

a jubilant comingout party for times up the movement spearheaded by

powerful hollywood women who helped raise millions of dollars to fight sexual

harassment around the country

signaling their support golden globes attendees swathed themselves in black

sported lapel pins and sounded off about sexist power imbalances from the red

carpet and the stage on the air e was called out about pay inequity after

its former anchor catt sadler quit once she learned that she was making far

less than a male cohost and during the ceremony natalie portman took a blunt

and satisfying dig at the allmale roster of nominated directors how could

that be topped as it turns out at least in terms of the oscars it probably wont be

women involved in times up said that although the globes signified the

initiatives launch they never intended it to be just an awards season

campaign or one that became associated only with redcarpet actions instead

a spokeswoman said the group is working behind closed doors and has since

amassed million for its legal defense fund which after the globes was

flooded with thousands of donations of or less from people in some

countries

no call to wear black gowns went out in advance of the oscars though the

movement will almost certainly be referenced before and during the ceremony

especially since vocal metoo supporters like ashley judd laura dern and

nicole kidman are scheduled presenters

another feature of this season no one really knows who is going to win best

picture arguably this happens a lot of the time inarguably the nailbiter

narrative only serves the awards hype machine but often the people forecasting

the race socalled oscarologists can make only educated guesses

the way the academy tabulates the big winner doesnt help in every other

category the nominee with the most votes wins but in the best picture

category voters are asked to list their top movies in preferential order if a

movie gets more than percent of the firstplace votes it wins when no

movie manages that the one with the fewest firstplace votes is eliminated and

its votes are redistributed to the movies that garnered the eliminated ballots

secondplace votes and this continues until a winner emerges

it is all terribly confusing but apparently the consensus favorite comes out

ahead in the end this means that endofseason awards chatter invariably

involves tortured speculation about which film would most likely be voters

second or third favorite and then equally tortured conclusions about which

film might prevail

in it was a tossup between boyhood and the eventual winner birdman

in with lots of experts betting on the revenant or the big short the

priwe went to spotlight last year nearly all the forecasters declared la

la land the presumptive winner and for two and a half minutes they were

correct before an envelope snafu was revealed and the rightful winner

moonlight was crowned

this year awards watchers are unequally divided between three billboards

outside ebbing missouri the favorite and the shape of water which is

the baggers prediction with a few forecasting a hail mary win for get out

but all of those films have historical oscarvoting patterns against them the

shape of water has nominations more than any other film and was also

named the years best by the producers and directors guilds yet it was not

nominated for a screen actors guild award for best ensemble and no film has

won best picture without previously landing at least the actors nomination

since braveheart in this year the best ensemble sag ended up going to

three billboards which is significant because actors make up the academys

largest branch that film while divisive also won the best drama golden globe

and the bafta but its filmmaker martin mcdonagh was not nominated for best

director and apart from argo movies that land best picture without also

earning best director nominations are few and far between

lab2

破解维吉尼亚密码

具体代码如下：

def findindexkey(subarr):

visiable\_chars=[]

for x in range(32,126):

visiable\_chars.append(chr(x))

#print(vi)

test\_keys=[]

ans\_keys=[]

for x in range(0x00,0xFF):

test\_keys.append(x)

ans\_keys.append(x)

for i in test\_keys:

for s in subarr:

if chr(s^i) not in visiable\_chars:

ans\_keys.remove(i)

break

#print(ans\_keys)

return ans\_keys

strmi='F96DE8C227A259C87EE1DA2AED57C93FE5DA36ED4EC87EF2C63AAE5B9A7EFF\

D673BE4ACF7BE8923CAB1ECE7AF2DA3DA44FCF7AE29235A24C963FF0DF3CA3\

599A70E5DA36BF1ECE77F8DC34BE129A6CF4D126BF5B9A7CFEDF3EB850D37CF\

0C63AA2509A76FF9227A55B9A6FE3D720A850D97AB1DD35ED5FCE6BF0D138A\

84CC931B1F121B44ECE70F6C032BD56C33FF9D320ED5CDF7AFF9226BE5BDE3F\

F7DD21ED56CF71F5C036A94D963FF8D473A351CE3FE5DA3CB84DDB71F5C17F\

ED51DC3FE8D732BF4D963FF3C727ED4AC87EF5DB27A451D47EFD9230BF47CA\

6BFEC12ABE4ADF72E29224A84CDF3FF5D720A459D47AF59232A35A9A7AE7D3\

3FB85FCE7AF5923AA31EDB3FF7D33ABF52C33FF0D673A551D93FFCD33DA35B\

C831B1F43CBF1EDF67F0DF23A15B963FE5DA36ED68D378F4DC36BF5B9A7AFF\

D121B44ECE76FEDC73BE5DD27AFCD773BA5FC93FE5DA3CB859D26BB1C63CE\

D5CDF3FE2D730B84CDF3FF7DD21ED5ADF7CF0D636BE1EDB79E5D721ED57CE\

3FE6D320ED57D469F4DC27A85A963FF3C727ED49DF3FFFDD24ED55D470E69E\

73AC50DE3FE5DA3ABE1EDF67F4C030A44DDF3FF5D73EA250C96BE3D327A84D\

963FE5DA32B91ED36BB1D132A31ED87AB1D021A255DF71B1C436BF479A7AF0\

C13AA14794'

arr=[]

for x in range(0,len(strmi),2):

arr.append(int(strmi[x:2+x],16))

for keylen in range(1,14):

for index in range(0,keylen):

subarr=arr[index::keylen]

ans\_keys=findindexkey(subarr)

print('keylen=',keylen,'index=',index,'keys=',ans\_keys)

if ans\_keys:

ch=[]

for x in ans\_keys:

ch.append(chr(x^subarr[0]))

print(ch)

print('###############')

import string

def findindexkey2(subarr):

test\_chars=string.ascii\_letters+string.digits+','+'.'+' '

#print(test\_chars)

test\_keys=[]

ans\_keys=[]

for x in range(0x00,0xFF):

test\_keys.append(x)

ans\_keys.append(x)

for i in test\_keys:

for s in subarr:

if chr(s^i) not in test\_chars:

ans\_keys.remove(i)

break

return ans\_keys

vigenerekeys=[]

for index in range(0,7):

subarr=arr[index::7]

vigenerekeys.append(findindexkey2(subarr))

print(vigenerekeys)

print("#########")

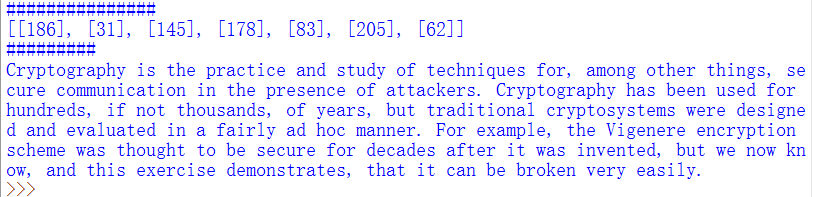
ming=''

for i in range(0,len(arr)):

ming=ming+chr(arr[i]^vigenerekeys[i%7][0])

print(ming)

最终的结果为：



破解一次一密密文

#!/usr/bin/env python3

from typing import List

import binascii

import argparse

SPACE = ord(' ')

def main():

parser = argparse.ArgumentParser(description='Many-time Pad Cracker')

parser.add\_argument(

'--filename',

type=str,

help='Name of the file containing the ciphertexts (default: ciphertexts.txt)',

default='ciphertexts.txt'

)

parser.add\_argument(

'-K', '--getkey',

action='store\_true',

help='Print cracked key instead of cracked cleartexts.'

)

parser.add\_argument(

'-k', '--key',

help='Encrypt messages with provided key.',

default=''

)

args = parser.parse\_args()

try:

with open(args.filename) as file:

ciphertexts = [binascii.unhexlify(line.rstrip()) for line in file]

except Exception as e:

print('Cannot crack {} --- {}'.format(args.filename, e))

raise SystemExit(-1)

cleartexts = [bytearray(b'?' \* len(line)) for line in ciphertexts]

if args.key:

decrypt(ciphertexts, cleartexts, args.key)

else:

crack(ciphertexts, cleartexts, args.getkey)

def decrypt(ciphertexts: List[bytes], cleartexts: List[bytearray], input\_key: str) -> None:

""" Decrypt ciphertexts using provided key and print cleartexts """

key = binascii.unhexlify(input\_key.rstrip())

for row in range(len(ciphertexts)):

for column in range(len(ciphertexts[row])):

cleartexts[row][column] = ciphertexts[row][column] ^ key[column % len(key)]

print(cleartexts[row].decode('ascii'))

def crack(ciphertexts: List[bytes], cleartexts: List[bytearray], getkey: bool) -> None:

""" Try to decrypt ciphertexts and print cleartexts or key """

max\_length = max(len(line) for line in ciphertexts)

key = bytearray(max\_length)

key\_mask = [False] \* max\_length

for column in range(max\_length): # go over characters from the beginning of lines

pending\_ciphers = [line for line in ciphertexts if len(line) > column]

for cipher in pending\_ciphers:

if is\_space(pending\_ciphers, cipher[column], column):

key[column] = cipher[column] ^ SPACE

key\_mask[column] = True

i = 0

for clear\_row in range(len(cleartexts)):

if len(cleartexts[clear\_row]) != 0 and column < len(cleartexts[clear\_row]):

result = cipher[column] ^ pending\_ciphers[i][column]

if result == 0:

cleartexts[clear\_row][column] = SPACE

elif chr(result).isupper(): # XOR with space return letter with swapped case

cleartexts[clear\_row][column] = ord(chr(result).lower())

elif chr(result).islower(): # XOR with space return letter with swapped case

cleartexts[clear\_row][column] = ord(chr(result).upper())

i += 1

break

if getkey:

for pos in range(max\_length):

if key\_mask[pos]:

print('{0:02x}'.format(key[pos]), end='')

else:

print('\_\_', end='')

print()

else:

print('\n'.join(line.decode('ascii') for line in cleartexts))

def is\_space(rows: List[bytes], current: int, column: int) -> bool:

"""

Return whether the current byte is encrypted space

If the current byte is space, XORing with other bytes should return alpha char or zero (when space)

"""

for row in rows:

result = row[column] ^ current

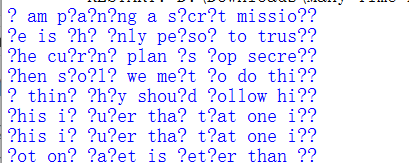
if not (chr(result).isalpha() or result == 0):

return False

return True

if \_\_name\_\_ == '\_\_main\_\_':

main()



最后根据猜测单词，补全明文，结果如下：

I am planning a secret mission.

He is the only person to trust.

The current plan is top secret.

When should we meet to do this?

I think they should follow him.

This is purer than that one is.

Not one cadet is better than I.