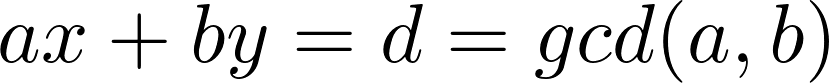
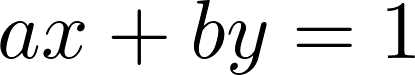
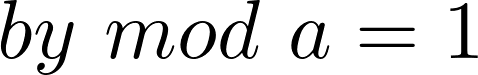
Q1:

Find the gcd(1137, 29), then according to extended Euclidean algorithm:



If gcd(1137, 29) = 1, so , then:



So, wpsoffice, y is the multiplicative inverse of b modulo a.

Now, set a = 1137, b = 29.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i | ri | qi | xi | yi |
| -1 | 1137 |  | 1 | 0 |
| 0 | 29 |  | 0 | 1 |
| 1 | 6 | 39 | 1 | -39 |
| 2 | 5 | 4 | -4 | 157 |
| 3 | 1 | 1 | 5 | -196 |
| 4 | 0 | 5 |  |  |

When i = 3, r3 = gcd(1137, 29) = 1, y = -196, so the multiplicative inverse of 29 modulo 1137 is -196 + 1137 = 941(ensure the answer is a positive number).

Q2:

Doing relative frequency of cipher-text letters, suppose the frequency result is *cipher\_frequency.*

If *cipher\_frequency* equals to the standard frequency distribution for English

The encryption is a transposition cipher

Else

The encryption is a substitution cipher

Q3:

Chose No.8 article. The frequency of single letters, and trigraphs in the cipher-text is as below (number means the percentage of single letter / digraphs / trigraphs in the whole article):

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K | N | I | E | D | G | Z | P | A | L |
| 9.65 | 9.65 | 8.79 | 7.93 | 7.54 | 7.06 | 6.99 | 6.51 | 5.18 | 3.92 |

The standard frequency distribution in English:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| e | t | a | o | i | n | s | h | r | d |
| 12.7 | 9.1 | 8.2 | 7.5 | 7.0 | 6.8 | 6.3 | 6.1 | 6.0 | 4.3 |

The frequency of digraphs:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KA | DE | AN | IE | PI |
| 4.00 | 2.76 | 2.31 | 2.31 | 1.78 |

Common digraphs in English: th, he, in, er, an

The frequency of trigraphs:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KAN | DEF | FPI | ITK | TKA |
| 2.15 | 1.23 | 1.12 | 1.12 | 1.12 |

Common trigraphs in English: the, and, tha, ent, ing

According to the tables, KAN seems equal to ‘the’ in plain letters, DEF seems equal to ‘ing’ in plain letters. Also, there is a lot sequence KAZK, we can translate it into ‘th\_t’, so easily Z is ‘a’. The sequence ZEW can be ‘an\_’, so W is ‘d’. The sequence WIE’K can be “d\_n’t”, so I is ‘o’. After some try, we can break the cipher as below table(the upper case letter denotes cipher text, the lower case letter denotes plain text).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| h | p | u | i | n | g | s | m | o | x | t | c | v | e | k | r | b | q | y | w | z | l | d | f | j | a |

The origin article is:

grow this widely thought to be the panacea for all the major economic ills of the modern world. poverty? just grow the economy (that is, increase the production of goods and services and spur consumer spending) and watch wealth trickle down. don't try to redistribute wealth from rich to poor, because that slows growth. unemployment? increase demand for goods and services by lowering interest rates on loans and stimulating investment, which leads to more jobs as well as growth. overpopulation? just push economic growth and rely on the resulting demographic transition to reduce birthrates, as it did in the industrial nations during the 20th century. environmental degradation? trust in the environmental kuznets curve, an empirical relation purporting to show that with on going growth in gross domestic product (gdp), pollution at first increases but then reaches a maximum and declines. relying on growth in this way might be fine if the global economy existed in avoid, but it does not. rather the economy is a subsystem of the finite biosphere that supports it. when the economy's expansion encroaches too much on its surrounding ecosystem, we will begin to sacrifice natural capital (such as fish, minerals and fossil fuels) that is worth more than the man-made capital (such as roads, factories and appliances) added by the growth. we will then have what i call uneconomic growth, producing "bads" faster than goods--making us poorer, not richer. once we pass the optimal scale, growth becomes stupid in the short run and impossible to maintain in the long run.

The program for computing the frequencies of single letters, digraphs and trigraphs in any cipher text:

# python version: 3.7

import os

import operator

import collections

THIS\_FOLDER = os.path.dirname(os.path.abspath(\_\_file\_\_))

file = os.path.join(THIS\_FOLDER, 'file.txt')

def file\_to\_string():

f = open(file, "r")

if f.mode == "r":

return f.read()

def generateMap(text, step):

count\_map = {}

total = 0

length = len(text)

for i in range(length):

letters = text[i:i+step]

if (len(letters) == step and isLetter(letters)):

total += 1

if letters in count\_map:

count = count\_map[letters] + 1

count\_map[letters] = count

else:

count\_map[letters] = 1

printMap(sortMap(count\_map), total)

def isLetter(string):

for ch in string:

if (ch < "A" or ch > 'Z'):

return False

return True

def printMap(count\_map, total):

for key, value in count\_map.items():

percent = str(format((value / total) \* 100, '0.2f'))

print(key + ":\t" + str(value) + "\t" + percent)

def sortMap(count\_map):

sort\_list = sorted(count\_map.items(),

key=operator.itemgetter(1), reverse=True)

return collections.OrderedDict(sort\_list)

def countSingleLetter(text):

generateMap(text, 1)

def countDigraphs(text):

generateMap(text, 2)

def countTrigraphs(text):

generateMap(text, 3)

if \_\_name\_\_ == "\_\_main\_\_":

TEXT = file\_to\_string()

countSingleLetter(TEXT)

print('--------------------------')

print('--------------------------')

countDigraphs(TEXT)

print('--------------------------')

print('--------------------------')

countTrigraphs(TEXT)

Q4:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| p | c | r | y | p | t | o | g | r | a | p | h | y |
| x | 2 | 17 | 24 | 15 | 19 | 14 | 6 | 17 | 0 | 15 | 7 | 24 |
| f(x) | 7 | 0 | 21 | 20 | 6 | 17 | 19 | 0 | 1 | 20 | 22 | 21 |
| c | h | a | v | u | g | r | t | a | b | u | w | v |

The cipher text of ‘cryptography’ is ‘havugrtabuwv’

Q5:

Suppose that the cipher text has been divided to N units, Ci, ..., CN, and Ck has a bit error in the transmission. Pk is determined by Ck, so Pk is wrong. Then Ck enters into the 8-left shift register, after 64/8 times, Ck leaves the register, which effects 8 times decryption progress. So Ck, ..., Ck+8 are all incorrect, there 9 units in total are wrong.