quiz1 111350076 楊子賢

a) Please write a program to find out the frequencies of letters in the ciphertext.

```
import math
text = """C UYGHARMZ IUWMPRWIR GAIR YVRMP
MBHMZWMPUM C VMMXWPE YV PYR VCZ
ZMGYQMD VZYG CXCZG YP CPCXKTWPE CPD MBHXYZM
RNM VXYYD YV CDQCPUMD OPYSXMDEM SNWUN MCUN
KMCZ LZWPEI SWRN WR
"""
f = [0]*26
s = 0
for i in text:
    if i!=" "and i!='\n':
        s +=1
        f[ord(i)-ord('A')]+=1

for i in range(26):
    print(chr(ord('A')+i)+':',end = ' ')
    print(f[i])
    print(math.floor(f[i]/s*10000)/100)
```

```
A: 2
1.4
B: 2
1.4
C: 12
8.45
D: 6
4.22
E: 4
2.81
F: 0
0.0
G: 5
3.52
H: 3
2.11
```

```
I: 4
2.81
J: 0
0.0
K: 2
1.4
L: 1
0.7
M: 19
13.38
N: 5
3.52
O: 1
0.7
```

```
P: 12
8.45
Q: 2
1.4
R: 9
6.33
S: 3
2.11
T: 1
0.7
U: 6
4.22
V: 7
4.92
W: 9
6.33
X: 6
4.22
Y: 12
8.45
Z: 9
6.33
```

b) Use the plaintext frequency count information below as a reference to break this encrypted messages.

A COMPUTER SCIENTIST MUST OFTEN EXPERIENCE A FEELING OF NOT FAR REMOVED FROM ALARM ON ANALYZING AND EXPLORE THE FLOOD OF ADVANCED KNOWLEDGE WHICH EACH YEAR BRINGS WITH IT

c) Assume C is ciphertext, and P is plaintext. Can you find a particular relationship between C and P?

Table 3: Ciphertext to plaintext mapping

Ciphertext	A	B	C	D	E	F	G	H	I	J	K	L	M
	0	1	2	3	4	5	6	7	8	9	10	11	12
Plaintext	V	X	A	D	G	*	M	P	S	*	Y	B	E
Ciphertext	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	13	14	15	16	17	18	19	20	21	22	23	24	25
Plaintext	H	J	N	V	T	W	Z	C	F	I	L	O	R

d) Suppose "$f(x) = ax + b \bmod 26$", where x is plaintext, please solve the value of a and b.

A: $0 \to 2$ $\quad b \bmod 26 = 2$ $\quad b = 2$

B: $1 \to 11$ $\quad a + 2 \bmod 26 = 11$ $\quad a = 9$

$\Rightarrow f(x) = 9x + 2 \bmod 26$ #

e) What is the key size of the Mono-Alphabetic Substitution Cipher? Such a size makes exhaustive search becomes difficult?

26 alphabet $\Rightarrow$ $26!$

f) (Bonus) Please try to see if it is possible to decipher this problem with ChatGPT or another tool.

Even if I use GPT4, I still can't solve it, but I ask it alot about the way to use python to count the frequences of each word

Problem 2

Plaintext is encrypted using an affine cipher. A plaintext symbol, x, is drawn from $\mathbb{Z}_{30}$ and, hence, encryption is defined as "$y = ax + b \bmod 30$", where y is the resulting ciphertext and the encryption key is given by $k_{\text{enc}} = (a, b)$.

a) Determine the size of the key space (that is, the total number of keys).

a should be prime with 30, a might be (1, 7, 11, 13, 19, 23, 29)

b is a shifting parameter, 0~29, key space: $30 \times 8 = 240$

b) Determine all values in $\mathbb{Z}_{30}$ that have inverses and, by trail-and-error, determine the inverses.

```
while True:
    x = int(input())
    f = False
    for i in range(1,30):
        if(x*i)%30 == 1:
            print(str(x)+'->'+str(i))
            f = True
            break
```

1 → 1　　13 → 7　　23 → 17

7 → 13　　17 → 23　　29 → 29

11 → 11　　19 → 19

c) An attacker intercepts the following plaintext/ciphertext pairs:

x	y
4	8
10	26
27	7

Determine the encryption key $k_{\text{enc}} = (a, b)$.

```
for a in range(30):
    for b in range(30):
        if(4*a+b)%30==8 and (10*a+b)%30==26 and (27*a+b)%30 == 7:
            print(a,b)
```

a = 13　　b = 16 Ans

d) Determine the decryption key $k_{\text{dec}} = (c, d)$, where "$x = cy + d \bmod 30$".

```
for c in range(30):
    for d in range(30):
        if(8*c+d)%30==4 and (26*c+d)%30==10 and (7*c+d)%30 == 27:
            print(c,d)
```

c = 7　　d = 8 Ans