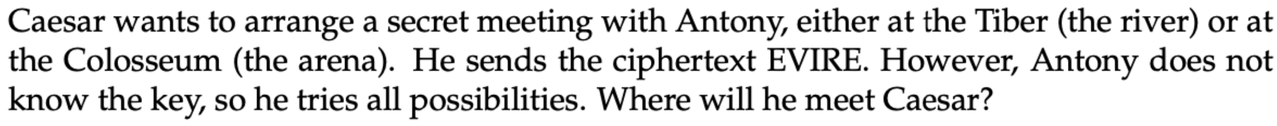
Aryan Nath September 22, 2024

CS-1306 / MAT-1201 Problem Set 1

Collaborators: *none*

Problem 1 [5 points]



Caesar wants to arrange a secret meeting with Antony, either at the Tiber (the river) or at the Colosseum (the arena). He sends the ciphertext EVIRE. However, Antony does not know the key, so he tries all possibilities. Where will he meet Caesar?

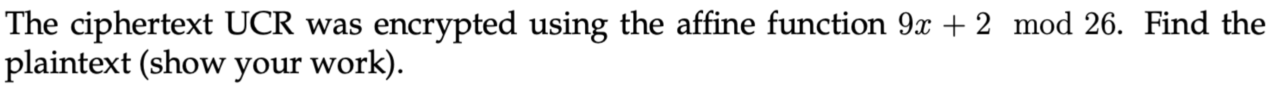
On using all possible keys, Antony gets both RIVER (K = 13) and ARENA (K = 22) as possible plaintexts. So he won't be able to make a decision on where to meet Caesar.

The set of possible decryptions are as follows:

Key	**Decryption**
0	EVIRE
1	FWJSF
2	GXKTG
3	HYLUH
4	IZMVI
5	JANWJ
6	KBOXK
7	LCPYL
8	MDQZM
9	NERAN
10	OFSBO
11	PGTCP
12	QHUDQ
13	**RIVER**
14	SJWFS
15	TKXGT
16	ULYHU
17	VMZIV
18	WNAJW

Key	**Decryption**
19	XOBKX
20	YPCLY
21	ZQDMZ
22	**ARENA**
23	BSFOB
24	CTGPC
25	DUHQD

Problem 2 [10 points]



The ciphertext UCR was encrypted using the affine function $9x + 2 \mod 26$. **Find the plaintext (show your work).**

Use the ciphertext UCR, and assume the encryption is in the form of a stream cipher.

So first convert the ciphertext to a vector of integers $= < 20, 2, 17 >$.

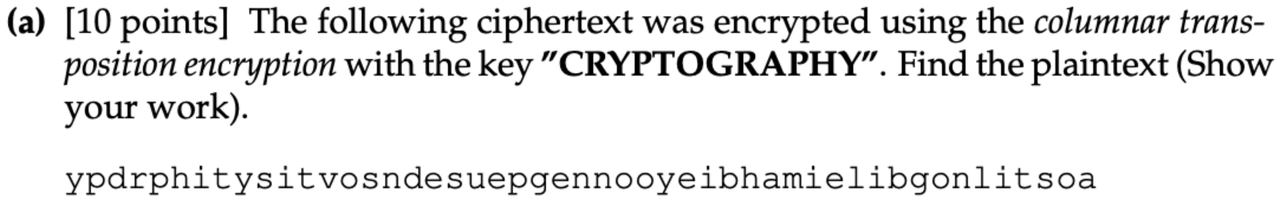
Now feed each of these to the affine function:

1. $20 \rightarrow 9 \times x + 2 \equiv 20 \pmod{26} \Rightarrow 9 \times x \equiv 18 \pmod{26}$.
   The inverse of $9 \pmod{26}$ is 3 [since $9 \times 3 = 27 \equiv 1 \pmod{26}$].
   $\Rightarrow 3 \times 9 \times x \equiv 54 \pmod{26} \Rightarrow 1 \times x \equiv 2 \pmod{26}$
   $\Rightarrow x = 2 \pmod{26}$

2. $2 \rightarrow 9 \times x + 2 \equiv 2 \pmod{26} \Rightarrow 9 \times x \equiv 0 \pmod{26}$
   $\Rightarrow 3 \times 9 \times x \equiv 0 \pmod{26}$
   $\Rightarrow x = 0 \pmod{26}$

3. $17 \rightarrow 9 \times x + 2 \equiv 17 \pmod{26} \Rightarrow 9 \times x \equiv 15 \pmod{26}$
   $\Rightarrow 3 \times 9 \times x \equiv 45 \pmod{26} \Rightarrow 1 \times x \equiv 19 \pmod{26}$
   $\Rightarrow x \equiv 19 \pmod{26}$.

Hence, the obtained vector for the plaintext is $< 2, 0, 19 >$.
Converting these to ASCII characters using an offset of 65 on each integer gives the plaintext as:
$< 2, 0, 19 > \rightarrow$ **CAT**.



(a) [10 points] The following ciphertext was encrypted using the *columnar transposition encryption* with the key **"CRYPTOGRAPHY"**. Find the plaintext (Show your work).

```
ypdrphitysitvosndesuepgennooyeibhamielibgonlitsoa
```

Problem 3 [30 points]

Keeping the order of the key characters in "CRYPTOGRAPHY" in the transposition matrix, with the last row having the remainder of the ciphertext length by the key length, the ciphertext characters are added in the sorted order of the key characters to get back the plaintext:

1	7	10	5	9	4	2	8	0	6	3	11
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
								y			
								p			
								d			
								r			
	-	-	-	-	-	-	-	-	-	-	-

1	7	10	5	9	4	2	8	0	6	3	11
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
p								y			
h								p			
i								d			
t								r			
y	-	-	-	-	-	-	-	-	-	-	-

1	7	10	5	9	4	2	8	0	6	3	11
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
p						s		y			
h						i		p			
i						t		d			
t						v		r			
y	-	-	-	-	-	-	-	-	-	-	-

1	7	10	5	9	4	2	8	0	6	3	11
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
p						s		y		o	
h						i		p		s	
i						t		d		n	
t						v		r		d	
y	-	-	-	-	-	-	-	-	-	-	-

1	7	10	5	9	4	2	8	0	6	3	11
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
p					e	s		y		o	
h					s	i		p		s	
i					u	t		d		n	
t					e	v		r		d	
y	-	-	-	-	-	-	-	-	-	-	-

1	7	10	5	9	4	2	8	0	6	3	11
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
p			p		e	s		y		o	
h			g		s	i		p		s	
i			e		u	t		d		n	
t			n		e	v		r		d	
y	-	-	-	-	-	-	-	-	-	-	-

1	7	10	5	9	4	2	8	0	6	3	11
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
p			p		e	s		y	n	o	
h			g		s	i		p	o	s	
i			e		u	t		d	o	n	
t			n		e	v		r	y	d	
y	-	-	-	-	-	-	-	-	-	-	-

1	7	10	5	9	4	2	8	0	6	3	11
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
p	e		p		e	s		y	n	o	
h	i		g		s	i		p	o	s	
i	b		e		u	t		d	o	n	
t	h		n		e	v		r	y	d	
y	-	-	-	-	-	-	-	-	-	-	-

1	7	10	5	9	4	2	8	0	6	3	11
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
p	e		p		e	s	a	y	n	o	
h	i		g		s	i	m	p	o	s	
i	b		e		u	t	i	d	o	n	
t	h		n		e	v	e	r	y	d	
y	-	-	-	-	-	-	-	-	-	-	-

1	7	10	5	9	4	2	8	0	6	3	11
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
p	e		p	l	e	s	a	y	n	o	
h	i		g	i	s	i	m	p	o	s	
i	b		e	b	u	t	i	d	o	n	
t	h		n	g	e	v	e	r	y	d	
y	-	-	-	-	-	-	-	-	-	-	-

1	7	10	5	9	4	2	8	0	6	3	11
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
p	e	o	p	l	e	s	a	y	n	o	
h	i	n	g	i	s	i	m	p	o	s	
i	b	l	e	b	u	t	i	d	o	n	
t	h	i	n	g	e	v	e	r	y	d	
y	-	-	-	-	-	-	-	-	-	-	-

1	7	10	5	9	4	2	8	0	6	3	11
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
p	e	o	p	l	e	s	a	y	n	o	t
h	i	n	g	i	s	i	m	p	o	s	s
i	b	l	e	b	u	t	i	d	o	n	o
t	h	i	n	g	e	v	e	r	y	d	a
y	-	-	-	-	-	-	-	-	-	-	-

Reading row-wise, the obtained plaintext is:

"peoplesaynothingisimpossiblebutidonothingeveryday".

(b) **[20 points] Write C code for columnar transposition encryption and decryption (key and key-size can be choose randomly or by user input).**

Link to the file:

Question 3b

Output:

```
Enter the key: CRYPTOGRAPHY
Enter the text: peoplesaynothingisimpossiblebutidonothingeveryday
Original text: peoplesaynothingisimpossiblebutidonothingeveryday
Encrypted text: ypdrphitysitvosndesuepgennooyamieeibhlibgonlitsoa
Decrypted text: peoplesaynothingisimpossiblebutidonothingeveryday
```

(Also submitted on google classroom)

Problem 4 [20 points]

(a) **[15 points] Eve captures Bob's Hill cipher machine, which uses a 2-by-2 matrix $M \mod 26$. She tries a chosen plaintext attack and finds that the plaintext BA encrypts to HC and the plaintext ZZ encrypts to GT. What is the matrix M? (Assuming $A = 0, B = 1$, etc.)**

Let the matrix $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

$$\Rightarrow \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 7 \\ 2 \end{bmatrix} \pmod{26}$$

and $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 25 \\ 25 \end{bmatrix} = \begin{bmatrix} 6 \\ 19 \end{bmatrix} \pmod{26}$

This gives us the following system of linear equations:

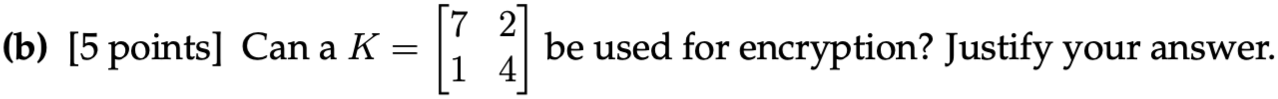
$$a = 7 \pmod{26} \tag{1}$$

$$c = 2 \pmod{26} \tag{2}$$

$$25a + 25b = 6 \pmod{26} \tag{3}$$
$$25c + 25d = 19 \pmod{26} \tag{4}$$

$\Rightarrow b = (6-175)*25^{-1} \pmod{26} = (-169)*25 \pmod{26} = 169 \pmod{2}6 = 13 \pmod{26}$
and $d = (19-50)*25^{-1} \pmod{26} = (-31)*25 \pmod{26} = 31 \pmod{26} = 5 \pmod{26}$

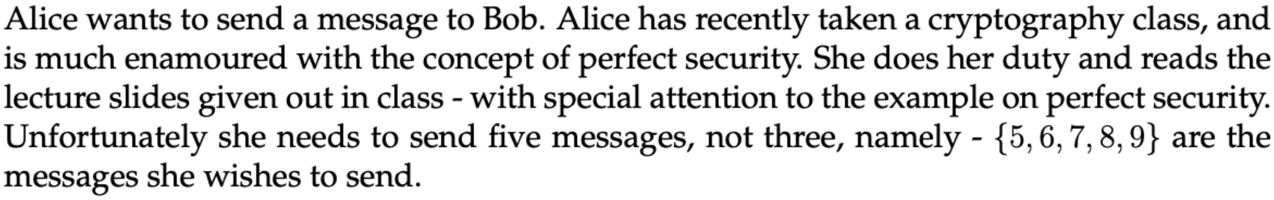
$\Rightarrow$ the matrix $M = \begin{bmatrix} 7 & 13 \\ 2 & 5 \end{bmatrix}$



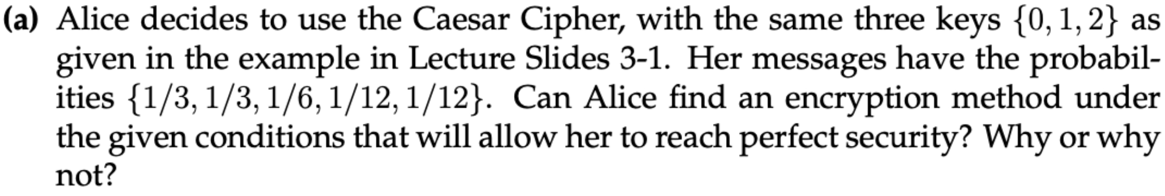
(b) [5 points] Can a $K = \begin{bmatrix} 7 & 2 \\ 1 & 4 \end{bmatrix}$ be used for encryption? Justify your answer.

No, this matrix cannot be used as the key. The determinant of K is 26, which is $0 \pmod{26}$. Hence, the inverse of the determinant does not exist in $\pmod{26}$. For decryption, the key matrix has to be inverted in $\pmod{26}$, since the determinant cannot be inverted, the matrix inverse cannot be calculated and decryption won't be possible with this matrix as the key.

Problem 5 [30 points]



Alice wants to send a message to Bob. Alice has recently taken a cryptography class, and is much enamoured with the concept of perfect security. She does her duty and reads the lecture slides given out in class - with special attention to the example on perfect security. Unfortunately she needs to send five messages, not three, namely - $\{5, 6, 7, 8, 9\}$ are the messages she wishes to send.



(a) Alice decides to use the Caesar Cipher, with the same three keys $\{0, 1, 2\}$ as given in the example in Lecture Slides 3-1. Her messages have the probabilities $\{1/3, 1/3, 1/6, 1/12, 1/12\}$. Can Alice find an encryption method under the given conditions that will allow her to reach perfect security? Why or why not?

We have been given the priori probabilities for message space. We can use the caesar cipher in modulo 5 to each message uniquely:

$5 \to 0$
$6 \to 1$
$7 \to 2$

$8 \rightarrow 3$
$9 \rightarrow 4$

Assuming the key space is uniformly distributed, the probability distribution table can be constructed as follows,

	0	1	2
0	1/9	1/9	1/9
1	1/9	1/9	1/9
2	1/18	1/18	1/18
3	1/36	1/36	1/36
4	1/36	1/36	1/36

The (message, key, ciphertext) triplets can be constructed as follows:

(0,0,0)	(0,1,1)	(0,2,2)
(1,0,1)	(1,1,2)	(1,2,3)
(2,0,2)	(2,1,3)	(2,2,4)
(3,0,3)	(3,1,4)	(3,2,0)
(4,0,4)	(4,1,0)	(4,2,1)

Now suppose the adversary receives the ciphertext 1. The prior probability of the plaintext 4(or 9) is 1/12, i.e $P[m = 9] = 1/12$

Now, given the ciphertext 1, the probability of getting 4(or 9) as the plaintext can be calculated as follows:

$$P[m = 4|c = 1] = \frac{P[m=4 \wedge c=1]}{P[c=1]} = \frac{1/15}{1/9+1/9+1/36} = \frac{1/15}{9/36} = 4/15 \approx 0.266$$

Comparing with the probability of randomly guessing the plaintext,

$$P[m = 4|c = 1] \approx 0.266 > P[m = 4] = 1/12 \approx 0.083$$

So, when the adversary has the ciphertext 1, they have a much higher probability (0.266) of guessing the plaintext 1 than when randomly guessing it with probability 1/12.

Due to the smaller key space, the definition of perfect secrecy is not satisfied.

$$P[m = m_0|c = c_0] = P[m = m_0]$$

is not met by Alice's encryption system.

(b) Alice learns a new language. In this language, all her messages are the same as before, but their probabilities are equal. Is it possible to either modify (if necessary) the previous system (if you found one), or to create one (if you couldn't find one), in such a way that the newer cryptosystem is perfectly secure?

The previous system can be modified to make it perfectly secure by increasing the key space to be equal to the size of the message space.

So, if the messages are $0, 1, 2, 3, 4 \pmod 5$, the key space will be $0, 1, 2, 3, 4 \pmod 5$.

Assuming the key space is uniformly distributed, the probability distribution table can be constructed as follows,

	0	1	2	3	4
0	1/25	1/25	1/25	1/25	1/25
1	1/25	1/25	1/25	1/25	1/25
2	1/25	1/25	1/25	1/25	1/25
3	1/25	1/25	1/25	1/25	1/25
4	1/25	1/25	1/25	1/25	1/25

The (message, key, ciphertext) triplet matrix will become:

$$\begin{array}{ccccc}
(0,0,0) & (0,1,1) & (0,2,2) & (0,3,3) & (0,4,4) \\
(1,0,1) & (1,1,2) & (1,2,3) & (1,3,4) & (1,4,0) \\
(2,0,2) & (2,1,3) & (2,2,4) & (2,3,0) & (2,4,1) \\
(3,0,3) & (3,1,4) & (3,2,0) & (3,3,1) & (3,4,2) \\
(4,0,4) & (4,1,0) & (4,2,1) & (4,3,2) & (4,4,3)
\end{array}$$

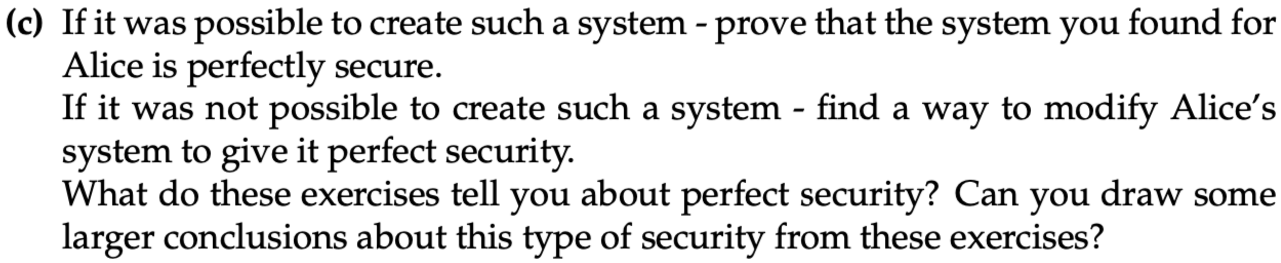
Now each row (each plaintext) has each of the 5 ciphertexts as its possible encryption. So knowing the ciphertext will not give the adversary any additional information about the plaintext. This can be expressed more strongly as,

$P[m = m_0 | c = c_0] = \frac{P[m=m_0 \wedge c=c_0]}{P[c=c_0]} = \frac{1/25}{5/25} = 1/5 = P[m = m_0]$

for any m_0 and c_0 in $\pmod 5$.

Hence, this modified cryptosystem for Caesar cipher encryption is perfectly secure by definition of perfect secrecy.

Given that each message has probability $1/5$ in $\pmod 5$ and each key selection has probability $1/5$, the join probability distribution of the message space and the key space,



(c) **If it was possible to create such a system - prove that the system you found for Alice is perfectly secure.**
If it was not possible to create such a system - find a way to modify Alice's system to give it perfect security.
What do these exercises tell you about perfect security? Can you draw some larger conclusions about this type of security from these exercises?

	0	1	2	3	4
0	1/25	1/25	1/25	1/25	1/25
1	1/25	1/25	1/25	1/25	1/25
2	1/25	1/25	1/25	1/25	1/25
3	1/25	1/25	1/25	1/25	1/25
4	1/25	1/25	1/25	1/25	1/25

(0,0,0)	(0,1,1)	(0,2,2)	(0,3,3)	(0,4,4)
(1,0,1)	(1,1,2)	(1,2,3)	(1,3,4)	(1,4,0)
(2,0,2)	(2,1,3)	(2,2,4)	(2,3,0)	(2,4,1)
(3,0,3)	(3,1,4)	(3,2,0)	(3,3,1)	(3,4,2)
(4,0,4)	(4,1,0)	(4,2,1)	(4,3,2)	(4,4,3)

The (message, key, ciphertext) triplets in this cryptosystem become:

So for any plaintext m_0 and any ciphertext c_0, the conditional probability of guessing m_0 given c_0 is as follows,

$P[m = m_0|c = c_0] = \frac{P[m=m_0 \wedge c=c_0]}{P[c=c_0]} = \frac{1/25}{5/25} = 1/5 = P[m = m_0]$

The probability $P[m = m_0|c = c_0]$ is equal to 1/12 as each unique message ciphertext form a unique element in the triplet matrix, that is no two different encrypt m_0 to c_0.

Since the probability of guessing the plaintext when ciphertext is known is exactly the same as the probability of randomly guessing the ciphertext, the modifed cryptosystem for Alice's Caesar cipher encryption is perfectly secure.

These exercises tell me that perfect secrecy is affected by the size of the message space and the key space, even when the same encryption function is being used. Some larger conclusions that can be drawn are that, in order to have perfect secrecy the key space needs to be at least as large as the message space and that complete distribution of the ciphertext space (message space) over the set of possible encryptions for each message will ensure perfect secrecy.

Problem 6 [15 points]

Beff Jezos, the founder of Ganga, is sending a message to Melon Usk using one of the following cryptosystems. In fact, Beff is bored and his plaintext consists of the letter a repeated a few hundred times.

Zark Muckerberg (the owner of BookFace), who is spying on them, knows what system is being used, but not the key, and intercepts the ciphertext.

For systems (a), (b), and (c), state how Zark will recognize that the plaintext is one repeated letter and decide whether or not Zark can deduce the letter and the key. (Note: For system (c), the solution very much depends on the fact that the repeated letter is a, rather than $b, c, \ldots$)

(a) Shift

(b) Affine

(c) Hill (2×2)

(a) Shift Cipher: Shifting of letter in this cipher is a one to one operation. So all repeated a's will be mapped to the same letter. So for this case it will be simple for Zark to recognize that the plaintext is one repeated letter as all letters of the ciphertext will be the same. However, since the plaintext has no meaning, Zark cannot perform a letter frequency based attack or brute force attack to get the repeated letters of the plaintext as the plaintext is not representative of the letter distribution expected in an english paragraph. So the key can be any of $0, 1, 2, 3, \ldots, 25$ and the repeated letter can be any of $a, b, c, \ldots, z$ and Zark will not be able to deduce it.

(b) Since the affine cipher is also a one to one function, Zark will recognize that the plaintext is one repeated character as all letters of the ciphertext will be the same. However, Zark will not be able to deduce the plaintext and the key through a brute force or frequency analysis attack, this is explained by the following example,

$3 \times x + 5 \equiv 0 \pmod{26} \Rightarrow x \equiv 7 \pmod{26}$

But the encryption function can also be,

$3 \times x + 4 \equiv 0 \pmod{26} \Rightarrow x \equiv 16 \pmod{26}$

So the plaintext can be $hhh \ldots h$ or $qqq \ldots q$, Zark won't be able to figure out which plaintext and key (a, b) is correct as neither of these plaintexts has meaning comparable to an english paragraph.

(c) Suppose the plaintext is $aa \ldots a$. For encryption with a 2×2 matrix in a Hill cipher system, the plaintext is broken into pairs of two and multiplied with the key

$K = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ for $(a, b, c, d \in Z_{26})$ to get the ciphertext for each pair as,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 0 \end{pmatrix} \pmod{26}$$

The ciphertex for $aa \ldots a$ will be $00 \ldots 0$. Let α represent the integer modulo equivalent of any letter chosen other than a. Suppose the received ciphertext is still $00 \ldots 0$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha \\ \alpha \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 0 \end{pmatrix} \pmod{26}$$

This means that, $(a + b)\alpha \equiv 0 \pmod{26}$ and $(c + d)\alpha \equiv 0 \pmod{26}$. Since $\alpha \not\equiv 0 \pmod{26}$, we get

$$(a + b) \equiv 0 \pmod{26} \quad (5)$$
$$(c + d) \equiv 0 \pmod{26} \quad (6)$$

This mean that such a key will map every plaintext to the ciphertext $00 \ldots 0$.

Now, from the above modular equivalence we get,

$$a \equiv (-b) \pmod{26} \text{ and } c \equiv (-d) \pmod{}$$

Through compatibility with scaling, we get,

$$a \cdot c \equiv (-b) \cdot c \pmod{26} \equiv (-b)(-d) \pmod{2}6 \equiv b \cdot d \pmod{26}$$
$$\Rightarrow a \cdot c - b \cdot d \equiv 0 \pmod{26}$$
$$\Rightarrow det(K) \equiv 0 \pmod{26}$$

So such a key will never be used for encryption in the hill cipher as decryption wont be possible with it.

An even simpler proof would be,

Consider distinct $\alpha, \beta \in Z_{26}$,

$$\begin{pmatrix} \alpha \\ \alpha \end{pmatrix} \not\equiv \begin{pmatrix} \beta \\ \beta \end{pmatrix} \pmod{26}$$

But in the encryption we have,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha \\ \alpha \end{pmatrix} \equiv \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha \\ \alpha \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 0 \end{pmatrix} \pmod{26}$$

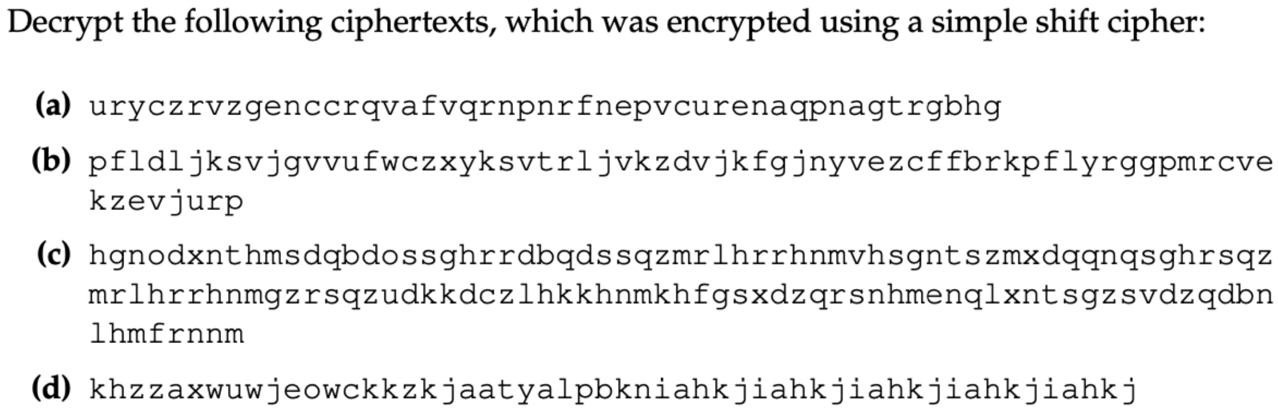
If K is invertible, which is required for decryption, we get,

$$\begin{pmatrix} \alpha \\ \alpha \end{pmatrix} \equiv \begin{pmatrix} \beta \\ \beta \end{pmatrix} \pmod{26}$$

This is a contraction.

Hence, the only possible plaintext for the given ciphertext $00\dots0$ can be $aa\dots a$ and Zark will be able to deduce that the plaintext consists of repeated characters which can only be a. However they won't be able to deduce the key as this result will hold for any key K.

Problem 7 [20 points]



Decrypt the following ciphertexts, which was encrypted using a simple shift cipher:

(a) `uryczrvzgenccrqvafvqrnpnrfnepvcurenaqpnagtrgbhg`

(b) `pfldljksvjgvvufwczxyksvtrljvkzdvjkfgjnyvezcffbrkpflyrggpmrcve kzevjurp`

(c) `hgnodxnthmsdqbdossghrrdbqdssqzmrlhrrhnmvhsgntszmxdqqnqsghrsqz mrlhrrhnmgzrsqzudkkdczlhkkhnmkhfgsxdzqrsnhmenqlxntsgzsvdzqdbn lhmfrnnm`

(d) `khzzaxwuwjeowckkzkjaatyalpbkniahkjiahkjiahkjiahkjiahkj`

The ciphertexts can be decrypted using a brute force attack or using frequency analysis. I have used frequency analysis for the question.

To create the reference frequency table, I used text from the wikipedia pages of two of my favourite movies. The obtained frequency distribution is:

0	:	0.09281588813456278,
1	:	0.015604417874151384,
2	:	0.03485662174485764,
3	:	0.03992299118451718,
4	:	0.11713446144492856,
5	:	0.02289998986726112,
6	:	0.018846894315533488,
7	:	0.04579997973452224,
8	:	0.07802208937075691,
9	:	0.0017225656094842436,
10	:	0.011247340156044179,
11	:	0.04529334279055629,
12	:	0.024622555476745363,

13	:	0.0738676664302361,
14	:	0.06576147532678082,
15	:	0.017326983483635625,
16	:	0.0012159286655182896,
17	:	0.07325970209747695,
18	:	0.07670483331644544,
19	:	0.07781943459317053,
20	:	0.020772114702604115,
21	:	0.009524774546559936,
22	:	0.012767250987942041,
23	:	0.004458405106900395,
24	:	0.016111054818117337,
25	:	0.0016212382206910528

Here $0 \ldots 25$ represent the alphabets $a \ldots z$. The frequency calculation was done by dividing the count of each alphabet by the total length of the reference text.

(a) Given the ciphertext: uryczrvzgenccrqvafvqrnpnrfnepvcurenaqpnagtrgbhg, I start of by constructing the frequency table for the ciphertext. The table I obtained is as follows:

0	:	0.06382978723404255,
1	:	0.02127659574468085,
2	:	0.0851063829787234,
3	:	0,
4	:	0.06382978723404255,
5	:	0.0425531914893617,
6	:	0.0851063829787234,
7	:	0.02127659574468085,
8	:	0,
9	:	0,
10	:	0,
11	:	0,
12	:	0,
13	:	0.1276595744680851,
14	:	0,
15	:	0.06382978723404255,
16	:	0.06382978723404255,

17	:	0.14893617021276595,
18	:	0,
19	:	0.02127659574468085,
20	:	0.0425531914893617,
21	:	0.0851063829787234,
22	:	0,
23	:	0,
24	:	0.02127659574468085,
25	:	0.0425531914893617

Now using shifts going from $0 \dots 25$, I compared the frequencies along the corresponding values of the reference table and the ciphertext table.

I got the minimum error for the key shift 13. So the decryption key is $13 \pmod{26}$ and the obtained ciphertext is:

helpmeimtrappedinsideacaesarcipherandcantgetout

(b) Given the ciphertext: pfldljksvjgvvufwczxyksvtrljvkzdvjkfgjnyvezcffbrkpflyrggpmrcvekzevjurp, I start of by constructing the frequency table for the ciphertext. The table I obtained is as follows:

0	:	0,
1	:	0.014492753623188406,
2	:	0.043478260869565216,
3	:	0.028985507246376812,
4	:	0.043478260869565216,
5	:	0.08695652173913043,
6	:	0.057971014492753624,
7	:	0,
8	:	0,
9	:	0.08695652173913043,
10	:	0.08695652173913043,
11	:	0.057971014492753624,
12	:	0.014492753623188406,
13	:	0.014492753623188406,
14	:	0,
15	:	0.057971014492753624,
16	:	0,
17	:	0.07246376811594203,
18	:	0.028985507246376812,

19	:	0.014492753623188406,
20	:	0.028985507246376812,
21	:	0.13043478260869565,
22	:	0.014492753623188406,
23	:	0.014492753623188406,
24	:	0.043478260869565216,
25	:	0.057971014492753624

Now using shifts going from $0 \ldots 25$, I compared the frequencies along the corresponding values of the reference table and the ciphertext table.

I got the minimum error for the key shift 17. So the decryption key is $17 \pmod{26}$ and the obtained ciphertext is:

youmustbespeedoflightbecausetimestopswhenilookatyouhappyvalentinesday

(c) Given the ciphertext: hgnodxnthmsdqbdossghrrdbqdssqzmrlhrrhnmvhsgntszmxdqqnqsghrsqzmrlhrrhnmgzrsqzudkkdczlhkkhnmkhfgsxdzqrsnhmenqlxntsgzsvdzqdbnlhmfrnnm, I start of by constructing the frequency table for the ciphertext. The table I obtained is as follows:

0	:	0,
1	:	0.023076923076923078,
2	:	0.007692307692307693,
3	:	0.08461538461538462,
4	:	0.007692307692307693,
5	:	0.015384615384615385,
6	:	0.05384615384615385,
7	:	0.1076923076923077,
8	:	0,
9	:	0,
10	:	0.038461538461538464,
11	:	0.038461538461538464,
12	:	0.07692307692307693,
13	:	0.1,
14	:	0.015384615384615385,
15	:	0,
16	:	0.08461538461538462,
17	:	0.09230769230769231,
18	:	0.1076923076923077,

19	:	0.023076923076923078,
20	:	0.007692307692307693,
21	:	0.015384615384615385,
22	:	0,
23	:	0.03076923076923077,
24	:	0,
25	:	0.06923076923076923

Now using shifts going from $0 \dots 25$, I compared the frequencies along the corresponding values of the reference table and the ciphertext table.

I got the minimum error for the key shift 25. So the decryption key is $25 \pmod{26}$ and the obtained ciphertext is:

ihopeyouinterceptthissecrettransmissionwithoutanyerrorthistransmissionhas travelledamillionlightyearstoinformyouthatwearecomingsoon

(d) Given the ciphertext: khzzaxwuwjeowckkzkjaatyalpbkniahkjiahkjiahkjiahkjiahkj, I start of by constructing the frequency table for the ciphertext. The table I obtained is as follows:

0	:	0.16666666666666666,
1	:	0.018518518518518517,
2	:	0.018518518518518517,
3	:	0,
4	:	0.018518518518518517,
5	:	0,
6	:	0,
7	:	0.1111111111111111,
8	:	0.09259259259259259,
9	:	0.12962962962962962,
10	:	0.18518518518518517,
11	:	0.018518518518518517,
12	:	0,
13	:	0.018518518518518517,
14	:	0.018518518518518517,
15	:	0.018518518518518517,
16	:	0,
17	:	0,
18	:	0,

19	:	0.018518518518518517,
20	:	0.018518518518518517,
21	:	0,
22	:	0.05555555555555555,
23	:	0.018518518518518517,
24	:	0.018518518518518517,
25	:	0.05555555555555555

Now using shifts going from $0 \ldots 25$, I compared the frequencies along the corresponding values of the reference table and the ciphertext table.

I got the minimum error for the key shift 22. So the decryption key is $22 \pmod{26}$ and the obtained ciphertext is:

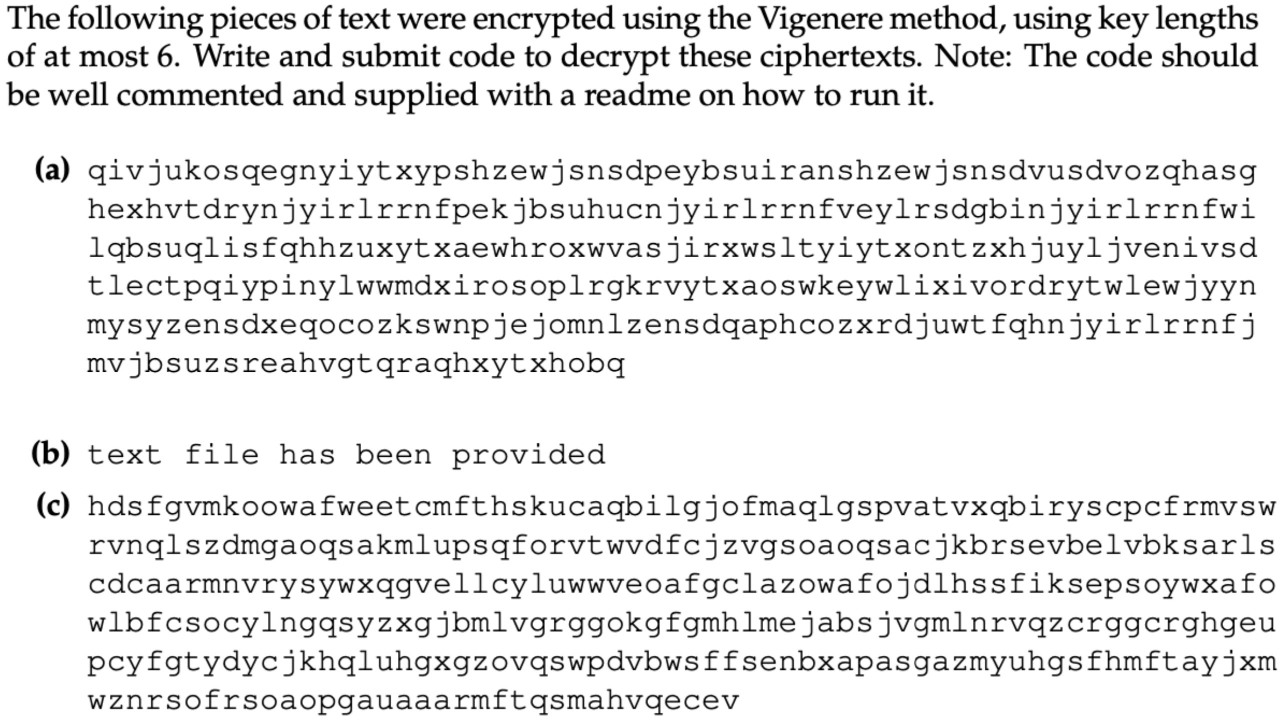
olddebayanisagoodoneexceptformelonmelonmelonmelonmelon.

I have used the following code for this problem:

Question 7

(Also submitted on google classroom)

Problem 8 [30 points]



The following pieces of text were encrypted using the Vigenere method, using key lengths of at most 6. Write and submit code to decrypt these ciphertexts. Note: The code should be well commented and supplied with a readme on how to run it.

(a) qivjukosqegnyiytxypshzewjsnsdpeybsuiranshzewjsnsdvusdvozqhasg
hexhvtdrynjyirlrrnfpekjbsuhucnjyirlrrnfveylrsdgbinjyirlrrnfwi
lqbsuqlisfqhhzuxytxaewhroxwvasjirxwsltyiytxontzxhjuyljvenivsd
tlectpqiypinylwwmdxirosoplrgkrvytxaoswkeywlixivordrytwlewjyyn
mysyzensdxeqocozkswnpjejomnlzensdqaphcozxrdjuwtfqhnjyirlrrnfj
mvjbsuzsreahvgtqraqhxytxhobq

(b) text file has been provided

(c) hdsfgvmkoowafweetcmfthskucaqbilgjofmaqlgspvatvxqbiryscpcfrmvsw
rvnqlszdmgaoqsakmlupsqforvtwvdfcjzvgsoaoqsacjkbrsevbelvbksarls
cdcaarmnvrysywxqgvellcyluwwveoafgclazowafojdlhssfiksepsoywxafo
wlbfcsocylngqsyzxgjbmlvgrggokgfgmhlmejabsjvgmlnrvqzcrggcrghgeu
pcyfgtydycjkhqluhgxgzovqswpdvbwsffsenbxapasgazmyuhgsfhmftayjxm
wznrsofrsoaopgauaaarmftqsmahvqecev

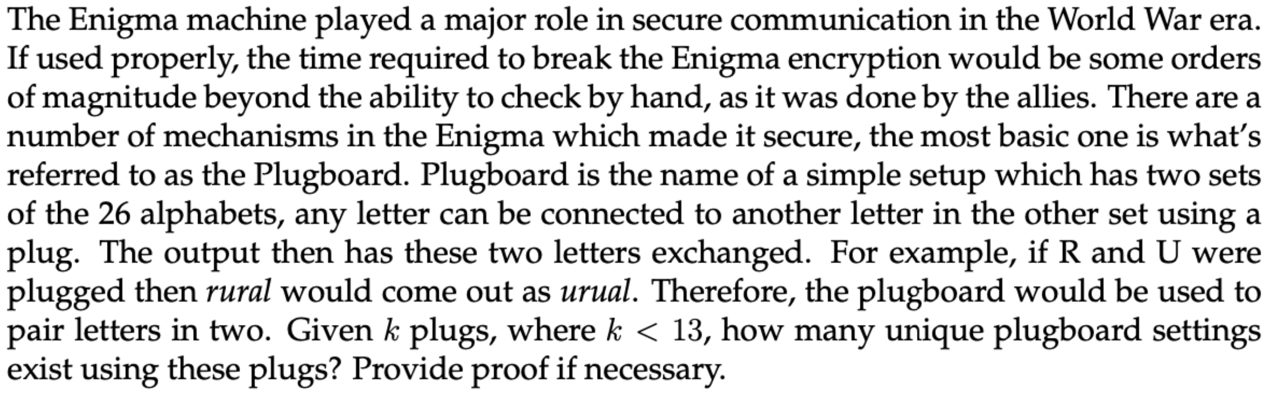
My code for decryption of the ciphertexts by using letter frequency analysis has been attached in:

Question 8

(Also submitted on google classroom)

8.py covers the main code for the reference frequency table generation and decryption of the ciphertexts. The readme 8.md explains how to run my code. 8.ipynb shows my working for how I got my complete code. The building of the frequency table is covered more in 7/7.ipynb.

Problem 9 [10 points]



The Enigma machine played a major role in secure communication in the World War era. If used properly, the time required to break the Enigma encryption would be some orders of magnitude beyond the ability to check by hand, as it was done by the allies. There are a number of mechanisms in the Enigma which made it secure, the most basic one is what's referred to as the Plugboard. Plugboard is the name of a simple setup which has two sets of the 26 alphabets, any letter can be connected to another letter in the other set using a plug. The output then has these two letters exchanged. For example, if R and U were plugged then *rural* would come out as *urual*. Therefore, the plugboard would be used to pair letters in two. Given k plugs, where $k < 13$, how many unique plugboard settings exist using these plugs? Provide proof if necessary.

If there are k plugs in plugboard, that is, there are groups of 13 alphabets being considered in the two sets of the plugboard then the total number of alphabets selected from 26 possible alphabets is $2k$.

The number of ways to select these $2k$ alphabets is:

$\binom{26}{2k} = \frac{26!}{26! \cdot (2k)!}$

Now these $2k$ alphabets need to be divided into pairs. The first plug has $2k$ options, the second has $2k - 1$ options, the third has $2k - 2$ and so on until the last plugboard has 1 option.

Using the multiplication rule of counting,

The total possible pairs is $(2k)!$.

This contains duplicate pairs as the order within a pair does not matter. So we divide this by 2^k, as for each of the k pairs there are 2 orderings.

So $(2k)!/2^k$ gives us the total pair permutations possible without pair duplication.

We further divide this with $k!$ as the permutations of the k pairs should not be counted,

$\frac{(2k)!}{2^k \cdot k!}$

Since, this is the total possible plugboard settings for each of the $\binom{26}{2k}$ selected alphabets, the total number of possible plugboard settings for each of $2k$ alphabet selections is,

$\binom{26}{2k} \cdot \frac{(2k)!}{2^k \cdot k!}$