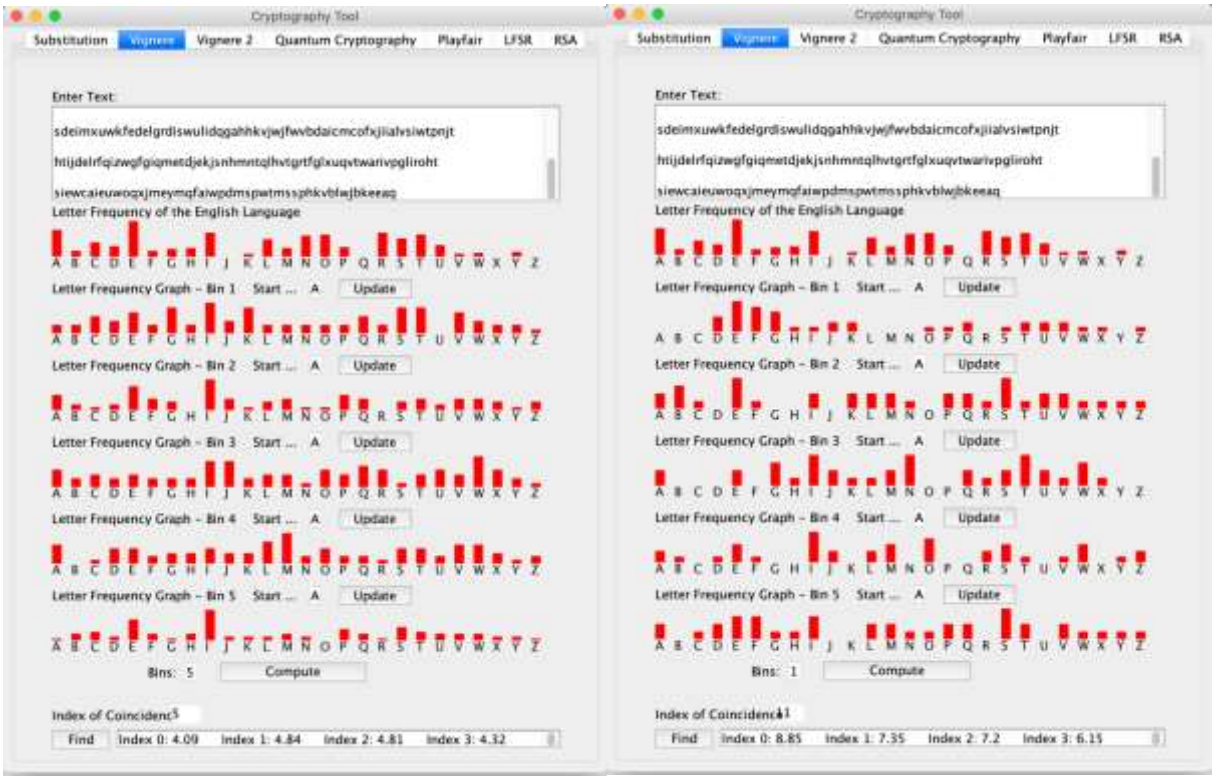
CIS 3362 Homework #3: Vigenere Code Breaking, Playfair, Hill, ADFGVX Solutions

1) Because vigenere is still a substitution cipher, using cryptool we can find the keyword length by increasing the number for Index of coincidence decryption until we get IC's close to the English language. Pictured below are the calculations for 5, not the keyword, and 11, the actual keyword. Once we start seeing IC's between high 5's to 8's we know we have the right keyword length.



The full IOC's for all bins is as follows:

```
Index 0: 8.85      Index 1: 7.35      Index 2: 7.2      Index 3: 6.15
Index 4: 6.3       Index 5: 6.75      Index 6: 6.98     Index 7: 6.5
Index 8: 6.98      Index 9: 7.77      Index 10: 7.46
```

After finding the correct keyword length, we can calculate the mutual index of coincidence to find relative shifts to bin 1. The formula is:

$$MI_C(x,y) = \frac{\sum_{i=0}^{25} f_i \acute{f}_i}{n\acute{n}}$$

In plain English this means the sum of the number of each letter 'a' in the first bin, times the number of the letter 'a' in the second bin, plus the number of each letter 'b' in the first bin, times the number of each letter 'b' in the second bin, up through z, divided by the total number of letters in the first bin, multiplied by the number of letters in the second bin.

I repurposed a program Mr. Guha wrote last year for doing just this, **micanalysisq1.java**, as well as printing out possible key words. The possible relative shifts for each bin are as follows:

```
2: 13 and 25

3: 25 and 21

4: 10 and 21

5: 3 and 14

6: 1 and 23

7: 7 and 3

8: 25 and 14

9: 6 and 2

10: 3 and 25

11: 16 and 12
```

By printing out all possible shifts, starting with a, b, c, …, z, we can find a probably keyword by eye from the results. For example, in the first possible keyword, we assume index 0 is 'a', then index 1 is 13 more than 'a' which is 'm'. Index two is 25 more than 'a', 'm', and so on and so forth until we have 11 letters for our keyword.

```
anzkdbhzgdq
boaleciaher
cpbmfdjbifs
dqcngekcjgt
erdohfldkhu
fsepigmeliv
gtfqjhnfmjw
hugrkiognkx
ivhsljpholy
jwitmkqipmz
kxjunlrjqna
lykvomskrob
mzlwpntlspc
namxqoumtqd
obnyrpvnure
pcozsqwovsf
qdpatrxpwtg
reqbusyqxuh
sfrcvtzryvi
tgsdwuaszwj
uhtexvbtaxk
viufywcubyl
wjvgzxdvczm
xkwhayewdan
ylxibzfxebo
zmyjcagyfcp
```

Eyeing the results, the second option, “boaleciaher”, looks most like English. Decrypting with this as the keyword gives us:

```
“newthatxhaveunoughiimeimillbepbletegiveaerizeqndmayqeifyeulooktdatlqs
tyeagsmesiagesydullrualizeihatiksethelordphizealdtandjhatitxsprejtyuni
fueinjermsouitslutterfgequedcydisiribujionannwayjkstliktthebualecieher
imonttealyoumheretwepripeisinihismussagexnsteqdthejdboftxismeshageiiju
sttdtellooutha itheruwillbtapripeandiisspesificldcatienwillqedessribedx
nmesiagethgeeantthecoctentioftheerizemillbesisclesedinbessawetwo”
```

Looking at this a few substitutions are clear, the second letter is off, and the eighth letter is off in the keyword. Changing these so the first word is “now”, and the x in the middle of that and have is “I” gives us the following math:

```
s-?=o
18-?=14
4 or E
x-?=i
23-?=8
15 or P
```

The final cipher is “BEALECIPHER” which gives us the following plaintext:

“Now that I have enough time I will be able to give a prize and maybe if you looked at last year’s message you’ll realize that I use the word prize a lot and that it is pretty unique in terms of its letter frequency distribution. Anyway just like the Beale cipher I won’t tell you where the prize is in this message, instead the job of this message is just to tell you there will be a prize and its specific location will be described in message three and the contents of the prize will be disclosed in message two”

2) To make this answer a bit more brief, I’ll cut out a lot of the repeated explanations and just show data.

The first step is to find the keyword length through Index of Coincidence. The keyword length is 7 with the following IOC’s:

```
Index 0: 4.7      Index 1: 6.21      Index 2: 9.57      Index 3: 7.39
Index 4: 5.88      Index 5: 7.39      Index 6: 6.41
```

We can run **micanalysisq2.java** to calculate possible Mutual Indexes of Coincidence, and keywords to translate:

```
2: 19 and 16
3: 16 and 9
4: 23 and 20
5: 22 and 0
6: 18 and 11
7: 13 and 10
```

```
atqxwsn
buryxto
cvszyup
dwtazvq
exubawr
fyvcbxs
gzwdcyt
haxedzu
ibyfeav
jczgfbw
kdahgcx
lebihdy
mfcjiez
ngdkjfa
ohelkgb
pifmlhc
qjgnmid
rkhonje
slipokf
tmjqplg
unkrqmh
volsrni
wpmtsoj
xqnutpk
yrovuql
zspwvrm
```

The only two words that resemble English to me are “lebhidy” and “buryxto”. The subsequent possible plaintext conversions are:

```
“eaqitxyicucsqwuyituijjejolboekmrajjxufbipuyiydwybbrucocuseevcqixjuxde
bbqhcafyusupohuqsxqrekfcuwbuhedunaoymybvtubboeeidsbqicwxoydufehwujwsfj
sqhtcadocehoidjxucoadjycugheuluhlruqaicosiqwujrruuvyhctudzeodhuceduiii
kffeceqbbyjcgeetvebiiqdysobkhhyjyfhecgtybq”
```

And

```
“okarehismeldageiscfsttotxwlyouwaltthepktzeisimhillbelzmecooenashtegoo
llarllpieceyzreachzcoupmefmeronewlyiwileeellyontnclaslhhyineoprgetgbqt
cardllnymorxtnthemxlntimepsoeveruceaksmxdsagetaceefirleenjoymsemonerts
uppolpallitlroodfoktsanicxmurrithqromqdhma”
```

It is much less immediately obvious that the second keyword is the correct answer than in the last question, but many English words can be seen such as “you”, “piece” and “card”. Because the keyword is so much shorter, any mistakes will be much more obvious in the text. To isolate the mistake I found a segment of English longer than the keyword where I could isolate the issue, “ehismeldage” looks like “thismessage” to me.

Substituting for mistakes:

```
B-?=T
1-?=19 (19-26=-7)
8 or I
A-?=H
0-?=7 (7-26=-19)
19 or T
```

Making these substitutions gives us a new keyword of "BURRITO" a normal English word, which is a safe bet for our keyword. This gives us the following decryption:

"Okay this message is just to tell you what the prize is. It will be some cool cash, ten dollars a piece for each group member. One day I will tell you in class why I never get gift cards anymore. In the mean time whoever breaks message three first enjoy the money I suppose. All its good for is a nice burrito from Qdoba."

3) The third cipher is a good example of a cipher that is easier to crack by hand. The easiest way to see this is by first calculating the keyword size (you want a smaller number, which will give you more text to analyze for each bin):

```
Index 0: 9.41      Index 1: 6.75      Index 2: 6.96      Index 3: 6.5
Index 4: 5.34      Index 5: 5.57
```

This gives us a keyword of 6, sufficiently short and easy to analyze. Next, the more trigrams with the more repetitions the better. This text has "WYK" repeating 3 times, which is perfect to analyze for 3 letter words. If we guess it correctly we will have half of the keyword figured out. It has a second repeating trigram in "CUD", which takes the remaining indicies that "WYK" does not, which means if we can correctly decode both trigrams we have the full key. Since "WYK" appears more, I will first assume that is "the" the most common word in the English language.

```
W-?=T
22-?=19
3 or D
Y-?=H
24-?=7
17 or R
K-?=E
10-?=4
6 or G
```

Before continuing, I decided to decrypt with these solutions as the first 3 letters of the key, and AAA as the last 3 (DRGAAA).

```
"thellisapfhquewcahmyfuahershhmeoncaitisvfhisfiymerozmiceizfouloirbehi
hkityoodillfcudthejyizewbpchtwinroupjhrtnelzcansjsitonwlyoufcuditpflas
elyameknidsothuaothelzdontavpokihnarouhkdrmabhlanovpssofzpcethibghthua
woulxiefunhfletmyrnowozhnyfuhuystolpesifnoathajwens"
```

This looked close enough to plain text to me, giving words like “arounh” and “you” and “win” which we know we are looking for, for me to assume that “WYK” was indeed the. This only left deciphering “CUD” to find the last 3 letters of the keyword. I also noticed there were often 4 letters that looked correct next to each other, meaning I probably picked A correctly by luck for one of the remaining letters. (The reason I picked A is because it wouldn’t change the rest of the ciphertext, meaning I could look at the CUD’s in context and try to figure out what trigram might go there to get the key). Both CUD’s are being led by an f: “yoodillfcudthejyize” and “onwlyoufcuditpflas”. We can assume the last three letters of the first segment are part of prize, so let’s solve for jy assuming that.

```
J-?=P
9-?=15 (15-26=-11)
20 or U
Y-?=R
24-?=17
7 or H
```

Putting this all together we get “DRGUHA” as the key, which seems as likely as any to me. Decryption gives the following:

“There is a plaque with my father’s name on it. It is by his former office. If you look behind it you will find the prize which two group partners can split. Once you find it please let me know so that others don’t go poking around Dr Mahalanobis office. Though that would be funny. Let me know of any funny stories if that happens.”

4) The first step to decrypting a playfair cipher is to separate the encrypted text into digrams. The encrypted text will always have an even number of characters due to padding. From this we get the following:

```
“tr sr le ci ve id is or bs ae ve le ar eu id sf me ae sf ge ln
rc oa ms ot pt ai ed fp pt nq be lk df ft th tr dn or hu mi sv
he ie qe ae mn rn me rv ms lk ev ec vk hy pt pt ea se om ao gp
ap al ot fx sr ab mh is ml om by tr on sn sb ic ml or ec sr”
```

We know that if we have anagrams “bs” and “sb” that the letters will be the same but swapped decrypted. In this example, “bs” decrypts to “re” so “sb” will decrypt to “er”. Another restriction is that any plaintext character can only encrypt to 5 things, the 4 other letters on its row, and the letter below it.

```
“tr” -> “he”
“sr” -> “re”
“le” -> “is”
“ci” -> “al”
“ve” -> “it”
“id” -> “tl”
“is” -> “es”
“or” -> “ec”
“bs” -> “re”
```

"ae" -> "t?"

Unfortunately, this doesn't give us the full picture. As we go through the plaintext, we may notice that 'e' also maps to 'm', which shouldn't be possible because that would mean 'e' maps to 6 letters instead of 5. This is our first indication that this Playfair cipher doesn't function exactly the same as the ones we saw in class.

If you look at these pairings, something is off. "le" becomes "is" early on in the text, however "is" becomes "es" when deciphered. This should not be possible because Playfair is a symmetric cipher (the same key is used for enciphering and deciphering). Although this is not what was taught in class, let's try separating the double letters (li*tt*le) with a padding letter and see what happens.

"tr" -> "he"
"sr" -> "re"
"le" -> "is"
"ci" -> "al"
"ve" -> "it"
"id" -> "**x**t"
"is" -> "le"
"or" -> "se"
"bs" -> "cr"
"ae" -> "et"

"is" now correctly maps to "le".

With the transcription of "sr" to "re" we get our first line of letters. Because we are decrypting we know that they will match to the left instead of right, and "ERS" will be 3 consecutive letters in our key grid.

Using this principle and the listed plaintext we get the following:

"he re is al it xt le se cr et it is ar eu xt sf me et sf ge ln
rc oa ms ot pt ai ed fp pt nq be lk df ft th he dn se hu mi sv
tr ie qe et mn rn me rv ms lk ti ec vk hy pt pt ea or om ao gp
ap al ot fx re ab mh le ml om by he on sn sb ic ml se ec re"

With this combined with the first few decrypted words, "hereisalittlesecret", we can begin to figure out the substitution matrix.

From the plaintext we know "e" is in the same row, or adjacent to the following letters: "r", "s", "t","o". Because only 5 letters can be on a line, with "e" being one of these letters, we know at least one letter will be the letter above or below e. We also know that "t", "r", "h" and "e" make a box, because "e", "r" and "t" are all theoretically on the same line due to all decrypting to e. Any set of two letters that has one of the original letters in its decrypted text must also not only be in the same row or column, but also next to each other, because when letters are on the same row or column, you take the letter to the right (left when decrypting) or below (above when decrypting).

We know the first two letters “tr” map to “he”, and that “sr” maps to “re” from this we can gather that s, r, and e are in the same row.

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

After filling in the matrix with the letters ‘t’, ‘h’, ‘e’, ‘r’, ‘s’, ‘o’, ‘a’ and ‘I’ in the correct positions, I was able to guess the key of THUNDERSTORM, which led to the following plaintext.

“HE RE IS AL IT QT LE SE CR ET IT IS BE ST QT OC OM ET OC AM PU
SB EF OR EN IN EA MT OF IN DP AR KI NG AN DT HE NU SE TH EQ EX
TR AT IM ET OD OH OM EW OR KI TI SA WI NW IN IN TE RM SO FE FQ
FI CI EN CY RE GA RD LE SQ SO FW HE NY OU RC LA SQ SE SA RE”

5) The attached file, HillCipher.java has the plaintext and encryption matrix hard-coded and produces the following ciphertext:

kjnzfukujinbrkqknuqfzdyxfbrkjftuctcoqatkbbep

6) The work and solution for this problem appears on the following two pages.

Monday, September 30, 2019 4:55 PM

	A	D	F	G	V	X
A	k	3	g	7	v	i
D	u	a	q	d	m	2
F	y	l	r	5	s	o
G	4	0	b	z	8	e
V	p	j	x	n	h	6
X	c	9	+	f	1	w

step 1: change letters

L	E	T	U	S	M	E	E
FD	GX	XF	DA	FV	DV	GX	GX
T	A	T	4	P	M	A	T
XF	DD	XF	GA	VA	DV	DD	XF
9	2	7	0	K	N	I	G
XD	DX	AG	GD	AA	VG	AX	AF
H	T	S	C	I	R	C	L
VV	XF	FV	XA	AX	FF	XA	FD

E	
GX	

2 4 1 6 3 5

C H A P E L

F D G X X F

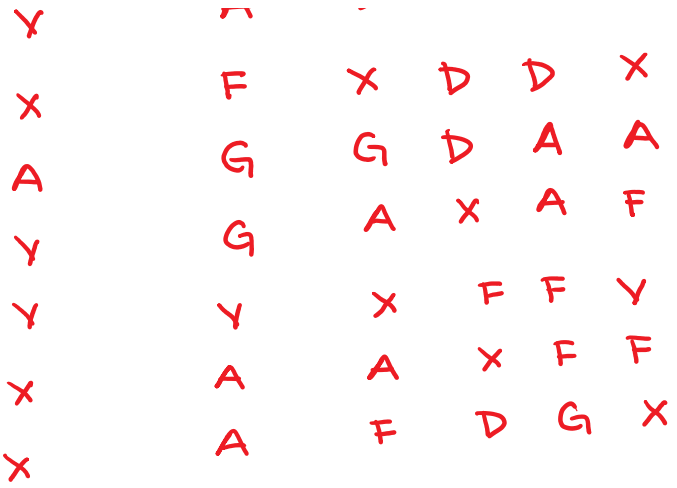
D A F V D V

G X G X X F

D X F G A

Y A D V D D

V F X D D X



GFGXDXGAXAFFDGDVXAVVXXXDXGDDAAFFGDAXDAFGGVAAFVFADXAFVFXXVXFVDDXFXD