CIS 3362 Homework #3: Playfair, Hill
Due: Check WebCourses for the due date.

1) (10 pts) By hand, encrypt the plaintext "WAKEMEUPWHENSEPTMBERENDS" with the keyword "GREENDAY" and the padding character "Q".

Solution

To encrypt with the Playfair cipher, we must set up the Playfair matrix

G	R	E	N	D
A	Y	B	C	F
H	I/J	K	L	M
O	P	Q	S	T
U	V	W	X	Z

*Remember that we remove duplicates and group I and J together!

WA = UB (Box Rule)
KE = QB (Column Rule)
ME = KD (Box Rule)
UP = VO (Box Rule)
WH = UK (Box Rule)
EN = ND (Row Rule)
SE = QN (Box Rule)
PT = QO (Row Rule)
MB = KF (Box Rule)
ER = NE (Row Rule)
EN = ND (Row Rule)
DS = NT (Box Rule)

Our ciphertext = 'UBQBKDVOUKNDQNQOKFNENDNT'

Also, refer to 'Playfair' in Cryptool to check your work 😊

2) (10 pts) For the Hill cipher, for a language with an **alphabet size of 37,** the encryption key is $\begin{pmatrix} 14 & 23 \\ 31 & 19 \end{pmatrix}$, what is the corresponding decryption key?

Solution

First find the determinant of the matrix given in the problem.

$\det(\begin{bmatrix} 14 & 23 \\ 31 & 19 \end{bmatrix})$ = (14*19) - (31*23) = -447 = -447 mod 37 = 34

Next, we must find 34^{-1} mod 37 via EEA

37 = 34(1) + 3
34=3(11) + 1

34 – 3(11) = 1 mod 37

** Note that 37-34(1) =3

34-11(37-34(1)) =1 mod 37

34-37(11) + 34(11) =1 mod 37

34(12)-37(11) = 1 mod 37

34(12) = 1 mod 37

So we find that 34^{-1} mod 37 ≡ 12 mod 37

Now find the inverse matrix with the information we have so far

$$(12)\begin{pmatrix} 19 & -23 \\ -31 & 14 \end{pmatrix} = \begin{pmatrix} 228 & -276 \\ -372 & 168 \end{pmatrix} \equiv \begin{pmatrix} 6 & 20 \\ 35 & 20 \end{pmatrix} mod\ 37$$

Therefore, our answer is $\begin{pmatrix} 6 & 20 \\ 35 & 20 \end{pmatrix}$

3) (30 pts) The following ciphertext was encrypted using the Playfair cipher. For the first few days, I won't give any matching plaintext. But, after a few days I'll reveal some characters or the plaintext, and then I'll reveal some more characters again before the due date. Determine the secret key and decrypt the whole ciphertext.

```
ranxrabuunimlgaelaumtienabovpdudnfaipbmlefsboctfmeztcxnlcix
cvrpzmcudapfpapmadktipabifllipabzanxcflzstlunzalozmbaauilpfb
iplmcqueidzpoztcxcqsabiqsufbiufsidkmtzsumlgsiadzbpnpamcnpaza
trosblarafvapcsosnuicsabimcotubpymcudvromaienabprpamcodsldys
ienlpaprpfimdeoudapepynalstdhfvaldapasidvapbsucapmadklgbutod
uztlglgcuzmbaautkpapamcnospudipfuhomadsbatisnlplgsmupilvolgd
uifqtsientkoqaplgsuhomadsticnprrfapvtykcfsdvroefqqiebcusiarp
ieaesbatapycixccsaqtbmcbatapycixclgdumefsbsblauidsrhoosudtva
lkfqihqlgtlaetspolplguducmsucaphogamckppodktlplmcsqydduidnos
blpapxgdudzponosbltdkaiftlplgsipdpubizuazscouembnmcumcofsmbf
prpcnqehr
```

Solution

Looking at the hint given, we have

Plain	O	N	L	Y	O	N	E	P	R	I	Z	E
Cipher	R	A	N	X	R	A	B	U	U	N	I	M

Doing this will help us build a known plaintext attack. Knowing some of the pairs will help build out the Playfair matrix.

O -> R and N -> A

L-> N and Y->X

		O		R
Y	X	A	L	N

Perhaps ON -> RA is a box rule and LY -> NX is a row rule? Let's add another digraph…

E -> B and P -> U

R -> U and I -> N

		O		R
Y	X	A	L	N
		E		B
		U		P

But this doesn't work since R can't go to U. We must try something else

O -> R and N -> A

L -> N and Y -> X

A		L		N
O				R
		X		Y

Notice we must add this space to make the two sets of diagraph “work”. So, our initial assumption was wrong.

E -> B and P -> U
R -> U and I -> N

A		L	N	I/J
	B			E
O	P		R	U
		X	Y	

Notice we slide these down to accommodate to add ‘E->B’ since the row and column rule didn’t make sense here.

Z -> I and E -> M

A		L	N	I/J
	B			E
				M
O	P		R	U
		X	Y	Z

This worked quite well, so maybe this was a column rule.

A		L	N	I/J
	B			E
				M
O	P		R	U
		X	Y	Z

From here where do we go? Well, considering we have ‘A’ then some letters then ‘B’, and I don’t know any words that have an ‘A’, ‘L’,’N’,’I/J’ AND ‘C’ or ‘D’, it is likely that ‘C’ and ‘D’ aren’t in the keyword. Because of this let’s put them in there and see where we can go.

A		L	N	I/J
	B	C	D	E
				M
O	P		R	U
		X	Y	Z

‘LGAELAUM’ is the cipher text right after the portion of plaintext we were given. ‘ONLYONEPRIZE’ what could come after that? Probably ‘THIS’?

This would mean ‘LG’-> ‘TH’ and ‘AE’ -> ‘IS’

A		L	N	I/J
S	B	C	D	E
F	G	H		M
O	P		R	U
		X	Y	Z

Notice how we added an ‘S’ since we had matching ‘A’,’I/J’, and ‘E’. Still not sure where ‘L’ maps to ‘T’.

Now since ‘S’ fits in to place we can probably start filling in some letters that WEREN’T in the keyword, since I can’t think of any words like ‘A _ L N (I/J) S’ that could contain ‘F’, ‘G’ or even ‘H’

With those additions made, we can decrypt the ‘LG’ -> ‘TH’.

A	T	L	N	I/J
S	B	C	D	E
F	G	H		M
O	P		R	U
		X	Y	Z

Ahhh, now the key word appears (sort of).

ATLNIS = ATLANTIS, we simply removed some repeated A’s. Now that we know the key word let’s fill in the rest of the matrix.

A	T	L	N	I/J
S	B	C	D	E
F	G	H	K	M
O	P	Q	R	U
V	W	X	Y	Z

When all is said and done, we find the keyword to be ‘ATLANTIS’

Running ‘Playfair.java’ will show us that the cipher text is:

onlyoneprizethistimeanditsforbreakingthismesqsageiwilqltelqlyouwheretogotofindanotethatnotewilqlhaveatriviaquestiontogetqtheprizeyouwilqlhavetocometomeandgivemetheanswertothetriviaquestionsotobefairleavethepaperwhereyoufinditsoqotherscanreaditqtoqomakesuretoburyitbacksoitsnoteasytoseqetofindthepaperwiththequestiongotothearboretumofqficestandatqthefrontoftherampleadinguptotheofqficeandloqoktowardshecyoushouldseqeanoutsidestairwelqlbelowthestairwelqlthereisasectionedofqfareawithmulchthatisaboutqthreqefeqetofqfthegroundatqthecornernearestqtowhereyouarestandingatqthearboretumiveburiedthemesqsagegoqodluckq

Cleaned up:

Only one prize this time and it’s for breaking this message I will tell you where to go to find a note that note will have a trivia question to get the prize you will have to come to me and give me the answer to the trivia question so to be fair leave the paper where you find it so others can read it too make sure to bury it back so it’s not easy to see. To find the paper with the question go to the arboretum office stand at the front of the ramp leading up to the office and look towards HEC. You should see an outside stairwell below the stairwell there is a sectioned off area with mulch that is about three feet off the ground at the corner nearest to where you are standing at the arboretum, I’ve buried the message. Good luck!

Note: The attached code Playfair.java was used to help decrypt this once the keyword was determined.

4) (50 pts) Write a program in C, Python or Java that reads in information from standard input about several Playfair key, plaintext message pairs and prints out the corresponding encrypted ciphertext. Please use the padding character 'x', and if there's ever a double x, use the padding character q as the back up padding character. Output should be one message per line. Given below will be both the input format and output format.

Input Format

The first line of input will be a single positive integer, *n*, representing the number of messages to encrypt (test cases).

The test cases follow with information about them taking 2 lines each. The first line of each test case is the secret key to use for Playfair for that test case. This is guaranteed to be a string of lowercase letters. (Remember that duplicate letters get skipped and that i/j are together in the Playfair grid.) The second line of each test case will be the plaintext message to encrypt with the key given on the previous line.

Output Format

For each test case, output a single line with the corresponding cipher text. So that there is ONLY one correct answer, for the purposes of this program, always output 'j' (never 'i') and pad with the character 'x' as necessary. If you have to encrypt "xx", please use the padding character 'q'.

Sample Input

```
3
problems
shewenttothestore
temperature
shewenttothestore
cryptography
codingisquitefunparticularlyinthestateofmississippiletstestxxxal
soandqqq
```

Note: The last line printed above wraps around. Thus, the string "soandqqq" should be appended to the al on the previous line. This is accurate in the separately posted sample file.

Sample Output

```
agmvmkqyqbytmaqbpm
okueuwmvmlmfrnmltm
odekmakqlxkpfjwlyhycdpcugyncfqpbkmybrkadqeqzqkqzqkhphqmdbzrkzbpx
pxwhmlghlfxpxpxp
```

Note: Same issue with the third line of output shown above, it's all in one line. The separately posted file playfair_sample.in is accurate.

See attached code, playfair.py.