CIS 3362 Homework #3: Playfair, Hill Solutions

1) (10 pts) By hand, using the Playfair cipher, encrypt the plaintext "BENOTAFRAIDOFGREATNESS" with the keyword "SHAKESPEARE" and the padding character "X".

Solution

First fill in the Playfair square:

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

Now, encrypt each pair:

BE --> DA	AI --> BQ	AT --> KQ
NO --> OQ	DO --> RU	NE --> US
TA --> QK	FG --> GI (or GJ)	SX --> AV
FR --> GP	RE --> DH	SX --> AV

Ciphertext: DAOQQKGPBQRUGIDHKQUSAVAV

2) (10 pts) For the Hill cipher, for a language with an **alphabet size of 43,** the encryption key is $\begin{pmatrix} 23 & 16 \\ 19 & 37 \end{pmatrix}$, what is the corresponding decryption key?

Solution

First, let's determine the determinant of the given matrix mod 43: 23 x 37 – 19 x 16 = 547 ≡ 31 (mod 43). Using the formula given in class, it follows that the desired inverse matrix (and decryption key) is:

$$(31^{-1} mod 43)\begin{pmatrix} 37 & -16 \\ -19 & 23 \end{pmatrix}$$

First, let's determine 31^{-1} mod 43 via the Extended Euclidean Algorithm:

43 = 1 x 31 + 12
31 = 2 x 12 + 7
12 = 1 x 7 + 5
7 = 1 x 5 + 2
5 = 2 x 2 + 1

5 – 2 x 2 = 1
5 – 2(7 – 5) = 1
5 – 2 x 7 + 2 x5 = 1
3 x 5 – 2 x 7 = 1
3(12 – 7) – 2 x 7 = 1
3 x 12 – 3 x 7 – 2 x 7 = 1
3 x 12 – 5 x 7 = 1
3 x 12 – 5(31 – 2 x 12) = 1
3 x 12 – 5 x 31 + 10 x 12 = 1
13 x 12 – 5 x 31 = 1
13(43 – 31) – 5 x 31 = 1
13 x 43 – 13 x 31 – 5 x 31 = 1
13 x 43 – 18 x 31 = 1

Take this equation mod 43 to find that $-18 \times 31 \equiv 1\ (mod\ 43)$. It follows that

$$31^{-1} \equiv -18 \equiv 25\ (mod\ 43)$$

Plugging in, we get:

$$(31^{-1} mod 43)\begin{pmatrix} 37 & -16 \\ -19 & 23 \end{pmatrix} = 25\begin{pmatrix} 37 & -16 \\ -19 & 23 \end{pmatrix} \equiv \begin{pmatrix} 925 & -400 \\ -475 & 575 \end{pmatrix} \equiv \begin{pmatrix} \mathbf{22} & \mathbf{30} \\ \mathbf{41} & \mathbf{16} \end{pmatrix}$$

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.

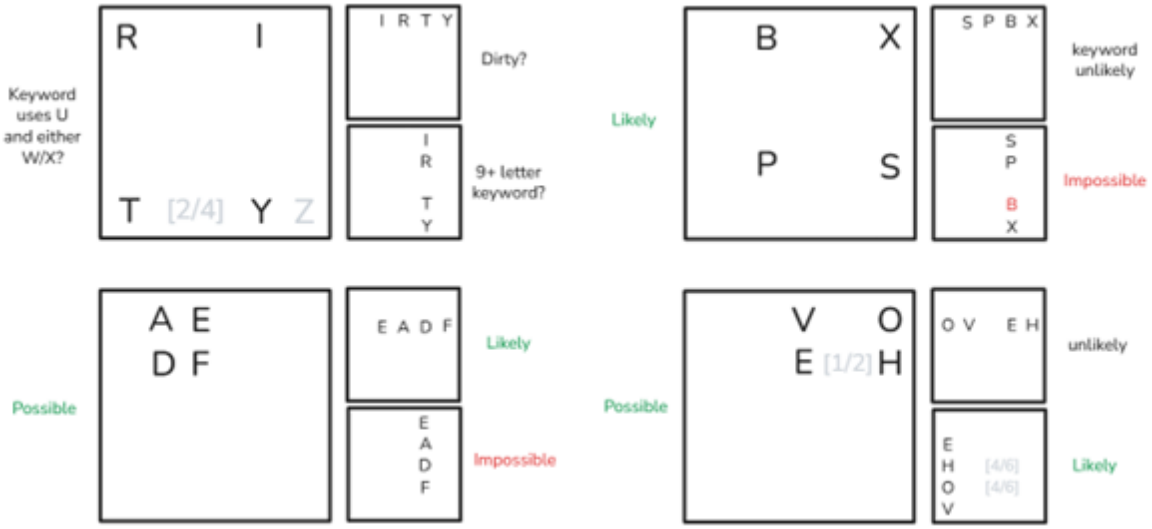
```
yrpxafhvqgzlnmvniuahexicxqfpntfwrecpdrgxfonuyrtrnvrznqdhrqhq
mhrnshvargvnkfharykarygufainucdfrydxxnynvnhcynvgdhpsnvvkrdge
cpuaesroeqtrohvgxufchqspunemaeiwvgaudehahdwevnuxdzvehqsppkgc
pnsnryvqbvdqyqofodkaihyqcvrqiueqtrkewrpeicvgcupfcdergyilxuae
vnitqbosoccykphvdsczohvnqsqbpcqcualuqcvgaokwcuvqihyqnqwdopvg
cuwkoahdotdhpqavpsvopsnvdfdbznvqbupxehopnsoxcuvcodnziebdtrfo
```

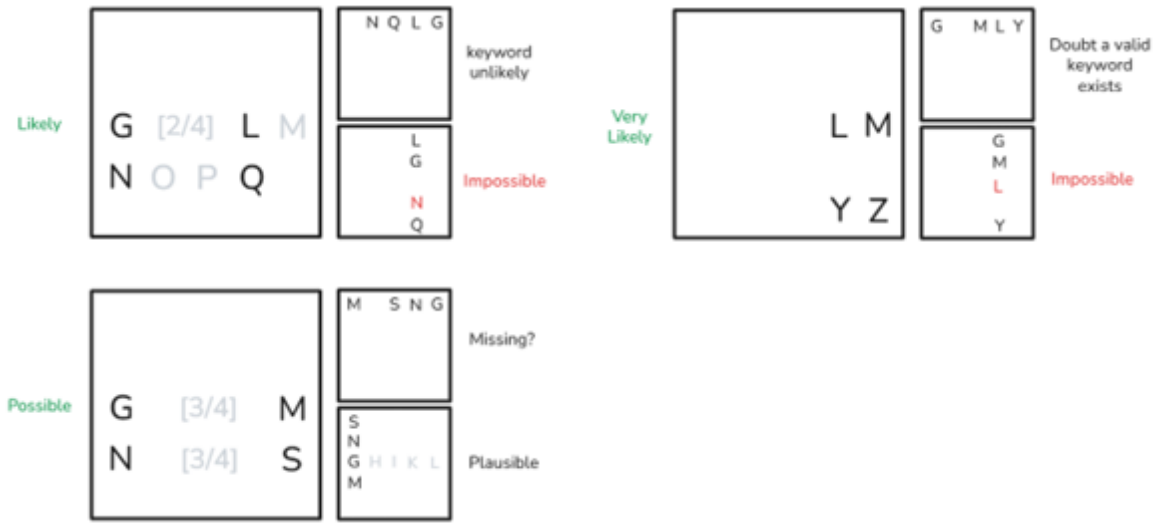
Answer

```
This solution uses the matching plaintext to the first 14
letters of ciphertext given.

Ciphertext:    YR PX AF HV QG ZL NM
Plaintext:     TI SB ED EO NL YM SG

Trying to apply the different rules to see if any seem likely:
```





I start filling in the ones that I feel most confident in, and hopefully by the time I get to the ones I'm unsure on, the grid will already be partially filled in to help guide my choices.

```
_____     _____     _____     _B__X
_____     _____     _____     _____
___LM  >  G__LM  >  G__LM  >  G__LM
_____     N___S     N__QS     NP_QS
___YZ     ___YZ     ___YZ     ___YZ
```

Now let's check how many of the letters among the uncertain pairings were used up:

```
     RITY: 1
AEDF: 0
VOEH: 0
```

We can work with the Y in RITY:

```
_B__X     RB_IX
_____     _____
G__LM  >  G__LM
NP_QS     NP_QS
___YZ     T__YZ T fits after the S
```

Let's see if there are any gaps we can fill in based on our 5x5 board status.

```
RB_IX     R_BIX
_____     _____
G__LM  >  G__LM
NP_QS     NOPQS P fits next to Q, O fits between N and P
T__YZ     T__YZ
```

We can work with the O in VOEH:

R_BIX
_E___
GH_LM
NOPQS
TV_YZ

Let's see if there are any gaps we can fill in based on our 5x5 board status.

R_BIX R_BIX
_E___ _E___
GH_LM > GHKLM K fits between H and L because I is in the keyword
NOPQS NOPQS
TV_YZ TVWYZ W fits between V and Y

We can work with the E in AEDF:

R_BIX
_EADF
GHKLM
NOPQS
TVWYZ

We know that the two remaining letters in our keyword are C and U, which it should be clear now that the keyword is RUBIX CUBE

RUBIX
CEADF
GHKLM
NOPQS
TVWYZ

Playfair Matrix:
RUBIX
CEADF
GHKLM
NOPQS
TVWYZ

Secret Key:
RUBIX CUBE

Ciphertext:
yr px af hv qg zl nm vn iu ah ex ic xq fp nt fw re cp dr gx fo
nu yr tr nv rz nq dh rq hq mh rn sh va rg vn kf ha ry ka ry gu
fa in uc df ry dx xn yn vn hc yn vg dh ps nv vk rd ge cp ua es
ro eq tr oh vg xu fc hq sp un em ae iw vg au de ha hd we vn ux

dz ve hq sp pk gc pn sn ry vq bv dq yq of od ka ih yq cv rq iu
eq tr ke wr pe ic vg cu pf cd er gy il xu ae vn it qb os oc cy
kp hv ds cz oh vn qs qb pc qc ua lu qc vg ao kw cu vq ih yq nq
wd op vg cu wk oa hd ot dh pq av ps vo ps nv df db zn vq bu px
eh op ns ox cu vc od nz ie bd tr fo

Plaintext:
TI SB ED EO NL YM SG TO BR EK FU RD IS AS GN AZ UC AN CI MR
ES OR TI NT OT XT SP EL IN LO LG TG OM WE TC TO MA KE IT AB
IT HR DE RQ RE AD IT FI RS TQ TO GE TQ TH EL OQ OT WH IC HC
AN BE FO UN DO NT HE TH IR DF LO QO RO FH EC BY TH EB AC KE
LE VA TO RI FY OU LO QO KA CR OS QS IT YO UW IL QL SE QE AB
UL QL ET IN BR DO NT HA TB OA RD TH ER SA FA CU LT YD IR EC
TO RY PI NQ NE DT AK EO FQ FT HE TO PQ PI NA ND BE HI ND TH
EP AP ER YO UL QL SP YA NO TH ER PA PE LE NV EL OP EW OQ OH
OQ OT AD AI TS YO UR SB UE NO SQ SU ER TE QE ST UD IA NT ES

Plaintext (string format):
TIS BE DE ONLY MSG TO BREK FUR DIS ASGN AZ U CAN C IM RESORTIN
TO TXT SPEL IN LOL GTG OMW ETC TO MAKE IT A BIT HRDERQ. READ IT
FIRSTQ TO GETQ THE LOQOT WHICH CAN BE FOUND ON THE THIRD FLOQOR
OF HEC BY THE BACK ELEVATOR. IF YOU LOQOK ACROSQS IT, YOU WILQL
SEQE A BULQLETIN BRD. ON THAT BOARD THERS A FACULTY DIRECTORY
PINQNED. TAKE OFQF THE TOPQ PIN AND BEHIND THE PAPER YOU'LQL SPY
ANOTHER PAPEL ENVELOPE. WOQOHOQO TADA ITS YOURS. BUENOSQ SUERTEQ
ESTUDIANTES.

Plaintext (cleaned up):

(sms/pirate language)
TIS BE DE ONLY MSG TO BREK FUR DIS ASGN AZ U CAN C IM RESORTIN
TO TXT SPEL IN LOL GTG OMW ETC TO MAKE IT A BIT HRDER.
(english)
READ IT FIRST TO GET THE LOOT WHICH CAN BE FOUND ON THE THIRD
FLOOR OF HEC BY THE BACK ELEVATOR. IF YOU LOOK ACROSS IT, YOU
WILL SEE A BULLETIN BOARD. ON THAT BOARD THERE'S A FACULTY
DIRECTORY PINNED. TAKE OFF THE TOP PIN AND BEHIND THE PAPER
YOU'LL SPY ANOTHER PAPER ENVELOPE. WOOHOO! TADA! ITS YOURS.
(spanish)
BUENOS SUERTE ESTUDIANTES.

4) (50 pts) Write a program in C, Python or Java that transforms an input string of letters, digits and underscores via the Railfence Cipher. For this version of the cipher, we will pick two parameters, **n**, the number of rows in the cipher grid, and **r**, the starting row number. To encrypt the input message, **s**, create a grid with **n** rows and |**s**| columns, and fill in the grid from the top left, always moving forward and diagonal, first going down, then bouncing back up and alternating. For example, if **n** = 5 and the input message is "WAKEUPATTHECRACKOFDAWN", then the grid would look as follows after being filled in:

```
W.......T.......O.....
.A.....T.H.....K.F....
..K...A...E...C...D...
...E.P.....C.A.....A.N
....U.......R.......W.
```

The way in which the parameter **r** will be used is as follows:

We will label row **r** with the number 1, and continue labeling all the rows moving down in sequential order. When we reach the bottom, we'll loop back to the top. For example, if **r** = 4, here is how the rows would be labeled:

```
3  W.......T.......O.....
4  .A.....T.H.....K.F....
5  ..K...A...E...C...D...
1  ...E.P.....C.A.....A.N
2  ....U.......R.......W.
```

Finally, to encrypt, read all the valid characters on row 1, then row 2, then row 3, etc. For this example, the encrypted text would be:

```
EPCAANURWWTOATHKFKAECD
```

Input Format

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

The test cases follow with information about each on 2 lines.

The first line of each test case will contain two space-separated positive integers: ***n*** ($1 \leq n \leq 100$) and ***r*** ($1 \leq r \leq n$), representing the number of rows for the cipher and the starting row, respectively.

The second line of each test case will contain a string of letters, digits and underscores only, representing the message to encrypt. The underscores are part of the message as well and should not be ignored.

Output Format

For each test case, output a single line with the corresponding cipher text.

Sample Input

```
2
5 4
WAKEUPATTHECRACKOFDAWN
3 3
THIS_IS_THE_SECOND_SAMPLE_CASE
```

Sample Output

```
EPCAANURWWTOATHKFKAECD
ISEC_PCT_TSNAESHSI_H_EODSML_AE
```

Please submit your program as an attachment called railfence.c, railfence.py or railfence.java.

Posted solution is the attached file railfence.java.