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

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

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?

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
```

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.