EE 289 – Homework Chapter 6

1 Solve the following introductory problems on strings.

(a) Write a function dayName that consumes a parameter, day, containing the numerical value of a day in the month of September 2008. Your function should return the name of that day as a string. For example:

dayName(8) should return 'Monday'

(b) You are new given a variable named days, a vector that contains the numeric values of days in the month of September 2008. Write a script that will convert each numeric value in the vector days into a string named daysOfWeek with the day names separated by a comma and a space. For example, if days=[8 9 10], daysOfWeek should be 'Monday, Tuesday, Wednesday'

Notice that there is no separator before the first day name or after the last one.

2 Consider the problem the MATLAB system has in parsing the tring:

'v=[1 2 3 4; 5,6, 7; 8; 9 10]'

Your task is to use strtok to parse this line and construct the array it represents. You will write a function arrayParse that consumes a string and returns two variables: a string that is the variable name and an array.

a) Tokenize the string first using '=' as the delimiter to isolate the variable name and the expression to be evaluated. Return the variable name to the user and save the rest of the line as the variable str1 for further processing. You may assume that there are no spaces outside the characters '[…]'.
b) Tokenize str1 with '[' and ']' to remove the concatenation operators and save the first token as str2.
c) Tokenize str1 using ';' as the delimiter. This will produce 0 or more strings that represent the rows of the array. Save each in the variable rowString. You may assume for now that the first row is the longest one.
d) Using nested while loops, tokenize each rowString with ';' and '' as delimiters and use str2num(…) to extract the numerical value of each array entry. Save it as rowEntry.
e) Concatenate the rowEntry elements horizontally to produce each row of the array. If the row is too short, pad it with zeros.
f) Concatenate each row vertically to produce the resulting array and return that array to the caller.
g) Test the function with cases like:
   empty=[]
   row=[1 2 3 4]
   diag=[0 0 0 1; 0 0 1; 0 1; 1]

3 Write a function called DNAcomplement that consumes a set of letters as a character array that forms a DNA sequence such as 'gattaca'. The function will produce the complement of the sequence so that the a's become t's, g's become c's, and vice versa. The string 'gattaca' would therefore befome 'ctaatgt'. You may assume that all the letters in the sequence will be lowercase and that they will all be either a, t, g, or c.

Note: You may be tempted to use iteration for this problem, but you don't need it.

4 The function rot(s,n) is a simple Caesar cipher encryption algorithm that replaces each English letter in places forward or backward along the alphabet in the strings. For example, the result of rot('Baz!,3) is 'Edc!'. An encrypted string can be deciphered by simply forming the inverse rotation on it, that is, rot('Edc!',3), which rotates each English letter in the string three places to the left. Numbers, symbols, and non-letters are not transformed. Implement the following function:

function rotatedText=rot(text,n)

To assist you as you solve this problem, you could write several functions as local functions in the rot.m file:
isUppercaseLetter(letter), getUppercaseLetter(n),
getLowercaseLetter(n), and getPosition(letter). You may also wish to use the built-in functions isletter(…), find(…), and mod(…).