UNIVERSITY OF UTAH
ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT

ECE1020 **COMPUTING ASSIGNMENT 5**
N. E. COTTER **MATLAB® CHARACTER STRINGS: ASCII MESSAGES**

READING
Matlab® Student Version: learning Matlab 6, Ch 6-11, 6-22, and A-16
Mastering Matlab® 6, Ch 8.1-8.4

TOPICS
String manipulations

OVERVIEW
In our design of a communication link for an imagined rover on a distant world, we will want to specify what distance and in which direction we want the rover to move. It is convenient to describe these moves in English, such as "5 left" or "3 right" where the units are understood to be meters.

To use these commands in our communication system, we enter them as character strings and then map them to numbers that represent the commands. We map these numbers, in turn, to codewords that we ultimately transmit.

PROCEDURE
In this assignment, you will use string commands to create a list of movement commands and corresponding codewords for our imagined rover.

+5 pts Script File for calculations
Using a text editor program on your PC, create a script file called **trip.m** containing matlab commands to perform the calculations in this assignment.

+5 pts Do **not** use semicolons at the ends of commands in your script files.

+5 pts Create a horizontal numerical array called `dist` containing the following four distance values: 3, 1, 4, and 2.

+10 pts The sprintf function is very much like the sprintf command in C. One difference is that Matlab allows the use of an array in place of the list of variables to be printed. Use this feature of `sprintf` to write a one-line command that prints the values in dist in a string called `dist_str`. Print the values as one-digit integers with no spaces in between.

+5 pts Use the transpose operator ' to create a variable called vert_dist_str from dist_str. The variable vert_dist_str consists of rows, each of which is one *character* representing a distance for the rover to move. The first row of vert_dist_str, for example, will be the character '3'.

+5 pts Because of issues relating to how Matlab handles spaces when concatenating strings, we will make use later on of a vertical array containing 4 rows with one period on each row. Use the strvcat function to create a vertical array called vert_periods whose four rows are each equal to the following string: '.'

+5 pts Now create a string variable called `direct_str` containing the following four direction command words separated by spaces: "left", "right", "right", and "left".

+10 pts The strtok function, like the strtok function in C, finds tokens in a character string. A token may be thought of as a word. You may choose the delimiter separating the words to be something other than the default white space, however. You might wish to use tabs, or even commas, for example.

In its simplest form, strtok returns the first word, (i.e., group of letters surrounded by white space), in a string. If we wish to extract the second word and so on, we use the following form of strtok:

[R,T] = strtok(S)

where S is the original string, R is the first token or word found in S, and T is what remains of S after the first token is removed.

Use strtok four times to extract the four direction words from direct_str. Save the four words in variables called d1, ..., d4.

+5 pts Use strvcat to create a vertical array called vert_direct_str whose rows are d1 through d4.

+5 pts We now wish to put the distance and direction information side by side in strings such as '3 left'. We can use the strcat function to place vert_dist_str and vert_direct_str side-by-side. Unfortunately, the strcat function eliminates spaces. To work around this problem we first put periods between the distance values and the direction words. Our first row will be '3.left', for example.

Use strcat to put vert_dist_str, vert_periods, and vert_direct_str side-by-side. The result will be an array, (call it vert_commands), with four rows. These rows are our commands, except for the unwanted periods in them.

+5 pts Use the strrep function on each row of vert_commands to replace the periods with spaces. Leave the result in vert_commands. In other words, overwrite the existing vert_commands array as you go.

+5 pts Now that we have created a series of commands for our rover, we may translate the commands into codewords for reliable communication with the rover. We will simulate this process in a simplified form by translating only the distance values into codewords.

Our first step is to create a codebook. We could use the methods developed in the previous lab, but we'll simplify further by just entering the following matrix of codewords:

codewords = [1 1 0 1 0 0; 0 1 1 0 1 1; 1 0 0 1 0 1; 0 0 1 0 1 1]

The rows of this array are the bits we transmit to indicate a 1, 2, 3, or 4, respectively. For practice, use the eval function to enter the above command. In other words, turn the command into a string and let eval interpret it as a Matlab command.

+10 pts Now we want to find a method of indexing that allows us to extract the appropriate rows of the codewords array as we scan through our list of commands for the rover. One approach is to use the strmatch function to find the indices of the rows in vert_commands where each of the digits 1 through 4 appear.

Use the strmatch command on vert_commands to find the indices of the rows where each distance, 1, 2, 3, or 4 appears. Save the indices in arrays called ind(1), ..., ind(4). Note that you are actually setting the column values one at a time of an array called ind.

+10 pts Using comment lines in your script file, explain in detail what the following command line does. Include this command in your file.
>> transmit_data(ind,:) = codewords(1:4,:)

+5 pts Run Script File
Run your script file by typing the name of the file without the .m
>> trip
If you make any changes in your **trip.m** file, be sure to run the following Matlab command to insure that Matlab reads your file again the next time you run it:
>> clear all

+5 pts End of Diary
>> diary off % Close the diary file. Look for the diary in e.g., c:\matlab\work directory.

E-mail your script file (trip.m) and your diary file to your TA, (as two separate e-mails). In the Subject line of your e-mail, be sure to put Your Name, "ECE1020 Comp5," and the file name, (e.g. trip.m). Also, print out the files and hand them in to the TA or to the ECE1020 locker.