Chem/Stat3240: Homework 9b
Matlab

October 29, 2015

3 The following functions are exercises in character array manipulation.

Write a the following function that performs as specified:

```
function n = countChar(c,A)
% c is a character
% A is a character array
% n is the number of time c occurs in A
```

The body of this function is just one command. Use the `find` command to complete this function. Hint: The argument of `find` should be a a relational operation.

Now write the following function so that it performs as specified:

```
function vk = findSubstrings(S1,S2)
%Find the occurences of string S1 in string S2.
%If S1 is a substring of S2 then vk is the vector positions
%of the first matching character of the matches of S1 in S2.
%If S1 is not a substring of S2, then vk is zero.
```

Use the `strcmp` command to compare successsive substrings of S2 to S1. When a string match is detected, save the index of the leading matching character to a vector. See the matching example in the slides.

4 This function introduces the use of cell arrays and character strings. Complete the following function so that it performs as specified:

```
function y = romanNumeral(x)
% x is an integer 0 <= x <= 3999
% y is the Roman numeral equivalent to x
```

Within the function, you will create four cell arrays called `Ones`, `Tens`, `Hundreds`, and `Thousands` with the character string entries as shown below (disregard the sentence about all possible concatenations). Using these cell arrays, you will perform the conversion by indexing to the proper entry of each array and concatenating the four entries to form the function output y. Check that $1 \leq x \leq 3999$, and if not print an error and return y as an empty string.

Thousands	
''	0
'M'	1000
'MM'	2000
'MMM'	3000

Hundreds	
''	0
'C'	100
'CC'	200
'CCC'	300
'CD'	400
'D'	500
'DC'	600
'DCC'	700
'DCCC'	800
'CM'	900

Tens	
''	0
'X'	10
'XX'	20
'XXX'	30
'XL'	40
'L'	50
'LX'	60
'LXX'	70
'LXXX'	80
'XC'	90

Ones	
''	0
'I'	1
'II'	2
'III'	3
'IV'	4
'V'	5
'VI'	6
'VII'	7
'VIII'	8
'IX'	9

Thus, 1907 is the concatenation of `'M'`, `'CM'`, `''`, and `'VII'`, i.e., `'MCMVII'`. Working with this idea, we can build a list of Roman numeral strings by computing all possible concatenations of entries taken from the following "place value" lists of strings:

Ones: { '', 'I', 'II', 'III', ..., 'VIII', 'IX' }
Tens: { '', 'X', 'XX', 'XXX', ..., 'LXXX', 'XC' }
Hundreds: { '', 'C', 'CC', 'CCC', ..., 'DCCC', 'CM' }
Thousands: { '', 'M', 'MM', 'MMM' }

The following are examples of how the function should perform.

```
% Problem 3 output
% ------------------------------------
%
% x=7; y = roman_numeral(x); disp(y)
% VII
%
% x=38; y = roman_numeral(x); disp(y)
% XXXVIII
%
% x=469; y = roman_numeral(x); disp(y)
% CDLXIX
%
% x=3847; y = roman_numeral(x); disp(y)
% MMMDCCCXLVII
```