String creation and manipulation

\[
\text{str} = 'Age 19' \quad \% \text{a 1-d array of characters}
\]
\[
\text{code} = \text{double}(\text{str}) \quad \% \text{convert chars to ASCII values}
\]
\[
\text{str1} = \text{char}(	ext{code}) \quad \% \text{convert ASCII values to chars}
\]

\[
\% \text{2-d array of characters}
\]
\[
\text{block} = \begin{bmatrix}
\text{'one row'}; \\
\text{'two rows'}
\end{bmatrix} \quad \% \text{Error! Rows must have same length}
\]
\[
\text{block} = \begin{bmatrix}
\text{'one row '}; \\
\text{'two rows'}
\end{bmatrix}
\]
\[
\text{blk} = \text{char}('\text{'one row'}, 'two rows')
\]
\[
\text{line1} = \text{blk}(1,:); \quad \% \text{length 8}
\]
\[
\text{line1trim} = \text{deblank}(	ext{blk}(1,:)) \quad \% \text{length 7, trailing blank removed}
\]

\[
\% \text{string functions}
\]
\[
\text{str} = 'Age 19'
\]
\[
\text{ischar(str)} \quad \% \text{is the variable a char array? Returns ONE value}
\]
\[
\text{isletter(str)}
\]
\[
\text{isspace(str)}
\]
\[
\text{caps} = \text{upper(str)}
\]
\[
\text{small} = \text{lower(str)}
\]

\[
\% \text{char arithmetic, relation}
\]
\[
\text{base} = 'a'
\]
\[
\text{nextcode} = \text{base} + 1
\]
\[
\text{nextletter} = \text{char}($\text{nextcode}$)
\]
\[
\text{letter18} = \text{char}($\text{base}+18-1$)
\]
\[
\text{ans1} = 'a' \gt 'b'
\]
\[
\text{ans2} = \text{base} == 'a'
\]
\[
\text{ans3} = \text{base} == \text{letter18}
\]
\[
\text{blk} = \text{char}('\text{'one row'}, 'two rows')
\]
\[
\text{ans4} = \text{blk} == 'o' \quad \% \text{character-by-character comparison}
\]
\[
\text{ans5} = \text{blk}(1,:) == \text{blk}(2,:) \quad \% \text{character-by-character comparison}
\]

Example 1

Write a function \texttt{caps} that capitalizes the first letter in each word of a string. Function \texttt{caps} accepts as input argument one string and returns two variables: the partially capitalized string and the number of capitalized letters in that string. Assume the string contains lower case letters and spaces only.
Example 2

Write a program to calculate the “day of year” from a user entered date in the string format \texttt{mm/dd/yyyy}. Note the user may omit any leading zero. The only MATLAB predefined functions you can use are \texttt{length} and \texttt{find}. Assume you have access to two user defined functions with the following headers:

\begin{verbatim}
function days = daysInMonth(m)
    % daysInMonth returns the number of days in month m, m is numeric, 1<=m<=12
    % Assume non-leap year

    function out = isLeapYear(y)
    % isLeapYear returns 1 if year y is a leap year and 0 otherwise
\end{verbatim}

\begin{verbatim}
date= input('Enter date as mm/dd/yyyy: ', 's');
% separate into substrings for month, day, year
\end{verbatim}

Example 3

Given a string \texttt{str}, write a program fragment to count how many times the alphabet \texttt{o} appears in \texttt{str}. 