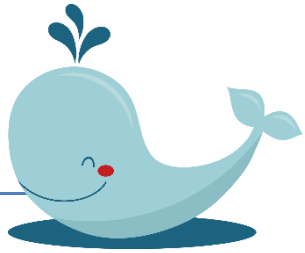


Standard Function Exercises

File: Standard Function Exercises (May 15,2018)

Course Learning Objectives:

- Apply syntax rules to design standard, recursive, anonymous, and nested functions
    - Demonstrate how to design, save, call, run, and debug a function, including
        - Identify input and output arguments to construct a function
        - Develop the algorithm within the function
        - Employ debugging strategies to debug a function

Action verbs for student learning outcomes:
https://www.mnstate.edu/assess/poa/actionverbs.aspx

Output Arguments: Syntax requirements

1. If your function returns one output, you can specify the output name after the function keyword in two ways:
   function myoutput = myFunction(X,Y)
   function [myoutput]=myFunction(X,Y)
2. If your function returns more than one output, enclose the output names in square brackets:
   function [A,B,C] = myFunction(Z)
3. If there is no output, you can omit the brackets or use empty square brackets:
   function myFunction(X,Y)
   function [ ] = myFunction(X,Y)

Input Arguments: Syntax requirements

4. If your function accepts any number of inputs, enclose their names in parentheses after the function name. Separate input names with commas:
   function [A,B] = myFunction(X,Y,Z)
5. If there are not inputs, you can omit the parentheses
   function [C] = myFunction

15-Exercises Standard Functions

BASIC Exercises

1. Write a function to compute the complementary sine, which is defined as:

$$sinc(x) = \frac{\sin(x)}{x}$$

Then write a Driver program to calculate the complementary sine function (i.e., sinc(x)) for values of x from 1.0 up to 10.0 in increments of 0.1

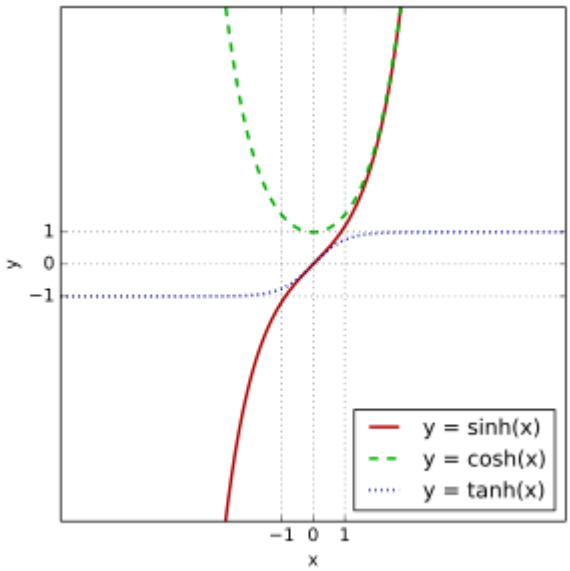
2. Write a function to convert km to miles, then upgrade the functions to handle arrays as Input and Output.
3. Write a MATLAB user-defined function (call it **worldSalute**) aiming to salute the world as "Hello there world," exactly 5 times. The function uses no arguments and returns no values to the calling program.

   Construct three user-defined functions to calculate the hyperbolic sine, cosine, and tangent functions. Choose a name to avoid syntax conflicts, e.g.: iSinh, iCosh, and iTanh, respectively. Use your functions to plot the shapes of the hyperbolic sine, cosine, and tangent functions by writing a program that computes them for common ranges.
4.

$$\sinh(x) = \frac{(e^x - e^{-x})}{2}$$

$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



MEDIUM COOK Exercises

5. Create the MATLAB **iSum** library **function** that adds up the elements of any 1D array, for instance the **t** array with N elements and returns the sum. The sum is defined by:

$$s_n = t_1 + t_2 + \cdots + t_n$$

Assume the **iSum** function will exist in MATLAB for the first time. You can use any other library function within your function except by **sum**. Upgrade the above function to work also with 2D array inputs.

6. Create the **iMax** library function that finds the maximum element of an 1D array, for instance the maximum of the **x** array and returns the result. Use the **numel** function to determine the elements in **x**. Can't use the **max** library function. Does your function could handle 2D array as input, if not upgrade iMax.
7. Write the function **iOnes(x).** A typical call with x=3 as the input argument, gives a 3 x 3 array of ones.
8. Write the function **myTriangulos(x). Where x is an integer variable.** A typical call with x=5 as the input argument, gives a triangle like:

55555
5555
555
55
5

9. Create the MATLAB library **iPi** parameter capable to handle up to 15 decimal digits of accuracy, such that whenever **iPi** is writen MATLAB will insert the mathematical pi value. This parameter can be created as a matlab function. In order to look as a parameter, write the function such as to call it, it won't need of input arguments. Note: The first 100 decimal digits of π are **3.14159 26535 89793 23846 26433 83279 50288 41971 69399 37510 58209 74944 59230 78164 06286 20899 86280 34825 34211 70679…** Show how you can use iPi in a Driver program, e.g., to compute the area of circles for values of radius, r=[1:10]
10. Write the function **iProd** which computes the cumulative product of the elements in a vector. You can't use the prod function. The cumulative product of the vector x, is defined by

$$P = \prod_{i=1}^{n} x_i = x_1 x_2 x_3 x_4 \dots x_n$$

11. Create the MATLAB **myLinspace(a,b,nn)** library **function** that returns a vector with nn equally spaced values from a up to b. Can't use the library function linspace.

WELL DONE Exercises

12. Any program in the Computer Project set (Einstein biker) can be solved using functions.
13. Write the function **iPrime** to find if a number is prime. Your function uses the *Trial Division* algorithm. Can't use isprime library function