Week 6 Tutorial 1

The purpose of this tutorial is to demonstrate declaring a function and using subfunctions to help clean up code and make it more readable.

```
% Always clear workspace variables before a new tutorial or program.
clear
clc
```

Edit the code below and update the variable named **`name`** with your name for this tutorial in the code below.

```
name="";
fprintf("Output for Tutorial_06_1 run by %s.\n", name)
```

Review the function

At this point, please open the **inefficientSort.mlx** file and review the function, read the description and all text within the function. Read through the comments to understand what the function is doing.

Run the function

As shown in the example portion of the function file, run the function in this code block below. First, print out the unsorted version, then the sorted (as shown in the Example Output below).

```

```

Adding doc for help

Adding the proper comments to your function will give you documentation in the help. Go ahead and right click on `inefficientSort` and select "Help on inefficientSort" to see the pop up dialog. It's also shown below. Documentation is important, especially if you plan to share your code with someone else, they need to know how to use the function, and what to expect from it.

inefficientSort

inefficientSort

Syntax

```
result = inefficientSort(x)
```

Description

This is an example of a very inefficient sorting algorithm.

This function is only intended to show how you can easily break up your code into smaller, more manageable tasks by using subfunctions. Doing so makes your code a little easier to read and a little easier to manage when making changes. For example, this algorithm would have normally been a nested for-loop, and while it still technically is, it makes the code much easier to follow by putting the nested for loop in its own function so the code is much cleaner.

Example

From the Command Window, run the following lines of code

x = 1:100;

sortedX = inefficientSort(x)

Example Output:

Run this tutorial from the **Command Window** and ensure your output matches the following.

```
Output for Tutorial_06_1 run by Geoff Berl.

Unsorted x
     1
     2
     3
     4
     5
     6
     7
     8
     9
    10
   -20
   -17
   -14
   -11
    -8
    -5
    12
    14
    16
    18
    20
    22
    24
    26
    28
    30
    32
    34
    36

Sorted x
    36
    34
    32
    30
    28
    26
    24
    22
    20
    18
    16
    14
    12
    10
     9
     8
     7
     6
     5
     4
     3
     2
     1
    -5
    -8
```