Homework 5 Due: Monday 10/20 in Class (turn in two pieces of paper)

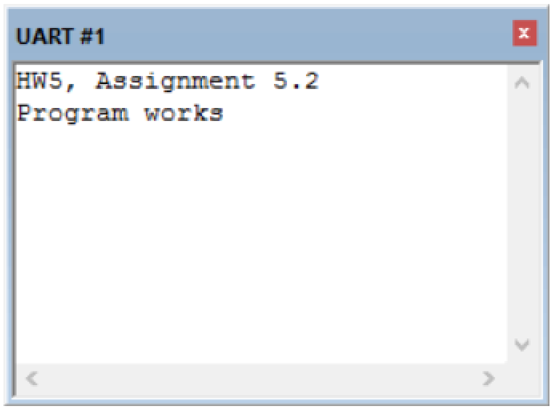
1) Read EdX Chapter 10: sections 10.1, 10.2, and 10.4; watch the videos
http://users.ece.utexas.edu/~valvano/Volume1/E-Book/

2) Log into zybooks and read and perform all activities in Sections 6.1 through, 6.9
The purpose of this homework is to practice functions in C. The exercises are practice programs to write, but not turn in. Assignments are tasks you will complete, printout and turn in.

You are allowed to work in groups of 2 on homework. Each student must turn in their own solution. If you will miss class you are allowed to turn in homework to your professor before class. To get credit for homework you must complete all questions, but the official score will be completion. I.e., we will not check the answers. However, the professors have answers to the homework, so if you are uncertain about your answers go to their office hours to check your answers against the solution key. We will not post the answers.

Assignment 5.1: Do Homework challenge 6.2.1 and 6.9.1

Assignment 5.2: Download and unzip **HW5_Assignment5_2.zip**. You can run this in the simulator and observe the output in a UART#1 window. Write a function that counts the mode of the characters in a string. The strings are null-terminated. Strings are variable length (with a maximum length of 29 characters) and only contain the letters 'a' through 'z'. Complete the implementation of the function **`Mode`**, and test it using the following Keil project in **HW5_Assignment5_2**. You may use pointer or index syntax to access data from the string. Take a screenshot of the Keil debugger showing your code and the output results of running your code. Show this printed page to your TA at the start of class. You are expected to write the function **`Mode`**. It is not necessary that you understand the rest of the code. What you need to know is what the **`Mode`** function is supposed to return the letter that occurs most frequently in the string. If there are two or more letters that occur the same number of time, return the one that is alphabetically first. This main program is an example of functional debugging; it defines a set of input parameters, calls your function, and evaluates the equivalence of your output to the expected output.



```
HW5, Assignment 5.2
Program works
```

Assignment 5.3: Run the interactive tool from chapter 12 posted at http://users.ece.utexas.edu/~valvano/UTx319K/context_switch.html and the value of the IPSR (typo ISPR should have been IPSR) that is set during the third step of the context switch