**HW # 6:  Stacks and Procedures**

*All main questions carry equal weight. Credit awarded to only those answers for which work has been shown*

1. [Procedures] Write a main program which sets the registers BX to either 0, 1 or 2.  Write a  procedure *DisplayTiger which* will display the string "War Eagle" in Blue or Orange depending upon the whether the input is 0 or 1.  Single step through the program, displaying the values of the stack pointer so that you understand how the call and return are implemented.

2. [Arrays] Write a program that:

* 1. Prompts the user for integer input 5 times
  2. Stores these inputs in a stack using the Push instruction
  3. After the storing is complete in Step 2, pop the stored values and display them on the screen using WriteInt (notDumpRegs).

Use the following:

.data  
PromptUser BYTE "Please enter a value:", 0

In your submission, please embed the full program (.asm and .lst file) and one screen shot with at least one positive and one negative input value.

3. [Compares, Procedures] Write a procedure, *Search* which searches the stack for the value that you provide in the register AX and returns its index, assuming the first value is stored in index *0*.   Write a main program that fills the stack with negative values, sets AX and calls *Search* and prints the index at which the value was found.

For example, if the inputs are:  -5, -6, -1, -10, -44, -79

and AX is set in the main program to be -1, then the expected output of your code is:

The target value is -1, and is located at index: 2

In cases where more than one element has the same value, you only have to output one of them. If the value is not found, print 0.

Use the following:

.data  
prompt      BYTE "Please input a value: ", 0  
spacing     BYTE ", ",0;  
String2     BYTE "The target value is,” 0  
String2     BYTE “and is located at index: ",0  
String3 BYTE "Value not found,", 0

In your submission, please embed the full program (.asm and .lst file) and one screen shot showing the values found.  Please test several sets of positive and negative values