**CIS 21JA Assignment 7**

**Overview**  
Modify the code of assignment 5 so that the program is made of procedures. The resulting code should still do the same work as assignment 5: add 2 unsigned, 16-bit integers and print the sum as a numeric text string.  
  
**Details**  
Take the code of assignment 5 and divide it into the following 4 procedures.

1. A readInput procedure that uses the stack to pass data.

The procedure prompts the user for a 16-bit integer and keeps prompting until there is a valid input. The procedure:

1. has 2 input arguments: the address of the prompt string and the address of the error string (for "input out of range" error)
2. returns a valid user input

2. An adding procedure that uses the stack to pass data.

The procedure adds the 2 numbers, passes back the sum through its address, and returns a boolean to show whether the sum is valid. The procedure:

1. has 3 input arguments: the 2 input numbers and the address of the sum. The sum is passed back through the address
2. returns a boolean to show valid or not valid sum

3. A printOutput procedure that uses registers to pass data.

The procedure converts the sum to a numeric text string and uses writeString to print the text string. The procedure:

1. accepts 3 input arguments: the sum, and the address of the text explanation string and the address of the numeric text string
2. has no return value

4. A main procedure that coordinates the calling of the 3 previous procedures. The main procedure:

1. calls the readInput procedure twice, once for each user input
2. calls the adding procedure and checks the boolean return value
3. calls the printOutput procedure if the sum is valid
4. loops to ask the user to continue or end

**Additional requirements**

* Document your program to get full credit.
* Create 3 additional variables in the .data section for the 2 user input and the sum. These should all be 16-bit variables.
* Use 16-bit registers for calculation (the adding procedure)
* Make sure the procedures pass data either through the stack or through registers as described above. 2pts per procedure are for using the right way to pass data.
* For the 2 procedures that pass data through the stack, the code should clear out the stack frame completely after the procedure call is done. 1 pt per procedure is for clearing out the stack correctly.
* Except for the main procedure, no other procedure should access variable names in .data directly. Only use data that are passed through the stack or register.

**Testing**

Test your result adequately: with valid and invalid input, with valid and invalid sum. Your program output should be the same as with assignment 5.