CIS 21JA – Lab 7

Change your lab5.asm file so the program is made up of functions or procedures.

Lab requirement:

1. Copy lab5.asm to lab7.asm. If there was any point deduction or change suggestion in lab5.asm, fix the code so you don’t get point deduction again in Lab 5.  
   Note that you should still use the smallest data size, and only move up to the next data size when needed.
2. Create an array of 2 bytes to store the user input hour and minute, where arr[0] is the hour and arr[1] is the minute.
3. Write a procedure called readTime that is used to get the input hour or the input minute or snooze time.

* Use the stack for input arguments and return value.
* Accept 3 input arguments: the address of the prompt string, the address of the error string, and the upper limit of the valid range for the input.
* Use the prompt string to ask the user for the hour or the minute, read in and check that the input is within the range of 0 and the upper limit.
* Print the error string (with the valid range) and keep prompting until you get a valid input.
* Return the valid hour or input.
* An example call to this procedure in pseudocode is: inputHour = readTime( hrPrompt, errStr, 23 )  
   or: inputMin = readTime( minPrompt, errStr, 59 )

1. Write a procedure called calcTime that calculates the alarm time.

* Use registers for input arguments and return value.
* Accept 2 input arguments: the array of user input time, the snooze time.
* Do the same calculation as in lab5.asm to find the wake up time.
* If the wake up time is valid, return the wake up hour and minute.
* If the wake up time is not valid, return -1 in place of the wake up hour

1. Write a macro called zeroFill that accepts a number as input.

* The number has a size that’s smaller than 32 bits (due to the requirement that you use the smallest data size)
* The macro prints the number as 2 digits, zero fill if needed.

1. Write a main procedure that implements the following pseudocode:

keep looping

arr[0] = readTime( hrPrompt, errStr, 23 )

arr[1] = readTime( minPrompt, errStr, 59 )

snoozeTime = readTime( snoozePrompt, errStr, 59 )

if calcTime( arr, snoozeTime )

print “Alarm set for”, zeroFill arr[0], “:”, zeroFill arr[1]

else

print “Invalid result”

Then put the main block of code in a loop to keep testing the code.

**Additional requirements**

* As with lab 5, use the smallest size registers and increase size only when needed.
* Make sure the 2 procedures pass data either through the stack or through registers as described above.

*1pt per procedure* is for using the required way to pass data.

* The readTime procedure should clear out the stack frame completely when the procedure call is done.   
  *1 pt* is for clearing out the stack correctly.
* Make sure readTime and zeroFill save and restore registers that they use.
* 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.  
  *1pt per procedure* is for not accessing variable names in .data directly

The program output should be the same as with Lab 5.