ARM Lab #3

Your program will implement a (somewhat) high-level language instruction that moves data directly from memory to memory.

We won' t be parsing this instruction, but you can assume that if we had bothered to write a parser, that the instruction that we would parse would look something like:

COPYDATA source_address, dest_address, length

source_address entered by the user is the location to copy from
dest_address entered by the user is the location you are copying to
length is how much you want to copy

We will assume that the parsing has already taken place, and that the value of source_address is in R0, the value of dest_address is in R1, and the value of length is in R2. To "simulate" this, you will read source_address, dest_address, and length in from the console, and place the values in the appropriate registers.

Your program should display memory at source_address and dest_address before the COPYDATA takes place, then it should prompt the user for source_address, dest_address, and length and stuff those values into the correct registers. Then it should perform the COPYDATA, and finally it should display memory at source_address and dest_address after the COPYDATA has taken place (make sure it is clear which memory area is which).

In your program, you will have to set up your 2 memory areas and store some appropriate data in those areas so you can adequately illustrate that the COPYDATA is really working. Allow the user to copy from memory area 1 to memory area 2 or from memory area 2 to memory area 1.

The only valid addresses a user can enter are the 2 addresses you set up in the your program, so you need to display an error if address1 or address2 entered by the user do not match the actual addresses. You will probably also want to prompt them in a way that lets them know what to enter. Also, make sure length does not exceed the max length for the data areas.