EE 371 Second Semester Test - Thursday October 26, 2000
35 points, 16.667% of Final Grade

Please put your name on the outside of the paper also Name _____KEY_____

1. The data in memory location $6000 - $600F, as shown by a MD 6000 command in the Dbug-12 monitor is:

6000 60 01 94 00 - 65 60 00 02 - 11 22 48 65 - 6C 70 4D 65 `...e`.."HelpMe.

Give the results (in hex) of the following instructions which are executed in sequence. (10 points)

```
ldx    #$6000      X = 6000
ldaa   4,x         A = 65
ldx    $6000       X = 6001
ldd    4,x         D = 6000
ldaa   [4,x]       A = 60
```

2. The following instruction sequence is executed:

```
ldaa   #$91
cmpa   $6007
```

State whether or not the following conditional branch instructions are taken: (8 points)

BNE? Yes BLE? Yes BHI? Yes BLS? No

3. Write an HC12 program that will do the following. Put your code in the blank lines following the comments. (10 points)

```
        ORG   $4000
    ; Initialize the stack pointer to $7000
4000 CF7000        3     lds     #$7000
    ; Initialize the Y register to point to $5000
4003 CD5000        5     ldy     #$5000
    ; Initialize memory locations $5000 - $500F to the alternating pattern
    ; 00,FF,00,FF,...
4006 8608          8     ldaa    #8      ; Initialize counter
                   9  loop:
4008 18007100     10     movw    #$00FF,2,y+
     FF
400D 0430F8       11     dbne    a,loop
    ; Return to the Dbug-12 Monitor
4010 3F           13     swi
```

4. The memory map of an HC12 microcontroller embedded system is shown below:

Address	Function	Show the EQUs appropriate to locate your program, stack and variable data: (3 points) CODE EQU $8000____ STACK EQU $1400____ DATA EQU $1000____
$0000 - $006F	I/O Registers	
$0070- $0FFF	Nothing	
$1000- $13FF	RAM	
$1400- $7FFF	Nothing	
$8000- $FFFF	ROM	

5. The following top down design has been implemented as a program to run on the HC12 EVBs.

```
;       1. Initialize all registers as needed.
;       2. Print the message "Type a character on the keyboard."
;       3. Get a character from the keyboard.
;       4. Print the message "The ASCII code in binary is %"
;       5. Print out the binary value in 1's and 0's
;       6. Return to the monitor
```

Assume the program is written, assembled and downloaded to the EVB. Assume DBUG-12 monitor routines are being used to print messages and get the character. Assume step 5 is coded in a subroutine called pr_binary. When the program is executed, you observe the following:
The message "Type a character on the keyboard." is printed, and after you do that the message "The ASCII code in binary is %" is printed and then the program hangs up. The binary value isn't printed and you have to hit reset to get back to the monitor.
Propose a debugging scheme that will allow you to find out where the problem is. (4 points)

The program appears to be working correctly up to step 4. To find out where the problem is, place break points at the instruction immediately following the jsr for step 4, at the first and last instructions in the binary printing subroutine, and at the instruction immediately following the jsr for step 5. Then run the program inspecting registers each time a break point is hit to look for obvious data problems. Ultimately, the program should hang up between two break points allowing me to narrow the search for the problem. If no Debug-12 routines are involved, I could use the trace feature to follow the program flow in the routine.