Sample Problem Set #1

Notes:
These problems are typical exam problems; most are drawn from previous homeworks and exams.
This exam is open book, open notes. It may help to have a calculator.
For partial credit, please show all work, reasoning, and steps leading to solutions.

1. The meaning of what is stored in memory depends on your interpretation. Assume that memory locations $800$ and $801$ contain the machine codes for the instructions “COMA” and “INCA”. Give the meaning of these values in these locations if you interpret them as:

(a) ASCII characters

(b) Unsigned 8-bit integers (i.e., give the decimal values)

2. Give the machine code corresponding to the following HCS12 assembly language program. Indicate the contents of memory at each address after the program is loaded into memory.

```
ORG $0D00
MAIN LDX #MAIN
STX $3000
CPX $10
BHS HERE
INX
HERE NOP
```

3. Fill in the blanks below, indicating the address and the contents of memory for the corresponding assembly
language source code.

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents</th>
<th>Assembly language source code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0800</td>
<td></td>
<td>ORG $0800</td>
</tr>
<tr>
<td>_____</td>
<td>_____</td>
<td>M DS.B 1 ; reserve one byte</td>
</tr>
<tr>
<td>_____</td>
<td>_____</td>
<td>N DC.B 1 ; reserve one byte &amp; initialize it</td>
</tr>
<tr>
<td>0D00</td>
<td></td>
<td>ORG $0D00</td>
</tr>
<tr>
<td>0D00</td>
<td>CC_____</td>
<td>LDD #10</td>
</tr>
<tr>
<td>_____</td>
<td>_____</td>
<td>STD $0</td>
</tr>
<tr>
<td>_____</td>
<td>B60100</td>
<td>LDAA $100</td>
</tr>
<tr>
<td>_____</td>
<td>_____</td>
<td>STAA M</td>
</tr>
<tr>
<td>_____</td>
<td>HERE</td>
<td>HERE ABA</td>
</tr>
<tr>
<td>_____</td>
<td>26_____</td>
<td>BNE HERE</td>
</tr>
</tbody>
</table>
4. The interface shown can be used for low current LEDs. Assume the LED voltage drop is 2 V. The resistor is 1000 Ω. When the software outputs a high, the voltage on PP0 becomes 4.9 V. When the software outputs a low, the voltage on PP0 becomes 0.5 V. What is the LED current when the LED is on?

5. A simple security system has two TR257-1 motion sensors connected to a HCS12 microcontroller. The motion sensors have an open collector output - when a motion sensor senses motion, it outputs a logic low. If either sensor detects motion, the system illuminates a light emitting diode (LED). Draw a schematic diagram for this system, showing the connections you would make from these components to the MCU. Don’t forget the resistors. Note – use a single input port pin for both sensors. It is not necessary to write any code for this problem.

6. Write HCS12 C code that sequentially illuminates a single LED segment in a seven-segment display, and traces a figure “8”. Specifically, it illuminates the top segment (“a”) for a short time, then illuminates “b”, “g”, “e”, “d”, “c”, “g”, and “f” in turn.

7. What does the following HCS12 assembly language program do? Describe the result of executing the program; don’t just say what each instruction is doing.

Instructions:
```assembly
ldx #$0800
ldaa #5
ldab #0
loop stab 0,x
incb
inx
dbne a,loop
```
8. Predict the result of executing the code below. What do registers a,b,x,y contain?

```assembly
lda a #$aa
ldab #bb
ldx #$1234
ldy #$5678
psha
pshb
ps hx
pshy
pula
pulb
pulx
puly
```

9. Estimate the running time of the following code fragment (assume a 24 MHz clock).

```assembly
ldab #$10
LOOP
decb
bne LOOP
```

10. Give the contents of the indicated registers or memory locations after the execution of each of the following program modules. Assume that prior to the execution of each of the following parts:

- Memory contains
  ($0080) = $01
  ($0081) = $02
  ($0082) = $03
  ($0083) = $04
- The M68HC12 registers contain: A = $7F, X = $0080
- The NZVC bits in the CCR are 0001

(Note: Do not treat the program modules as executing sequentially, one following another.)

<table>
<thead>
<tr>
<th>Program module</th>
<th>After execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) ADCA $80</td>
<td>A =</td>
</tr>
<tr>
<td></td>
<td>NZVC =</td>
</tr>
<tr>
<td>(b) ADCA #$80</td>
<td>A =</td>
</tr>
<tr>
<td></td>
<td>NZVC =</td>
</tr>
<tr>
<td>(c) BHI $E000</td>
<td>PC =</td>
</tr>
<tr>
<td>BRA $E010</td>
<td>NZVC =</td>
</tr>
<tr>
<td>(d) LDD 0,X</td>
<td>A =</td>
</tr>
<tr>
<td>ABX</td>
<td>X =</td>
</tr>
<tr>
<td>(e) ORG $D00</td>
<td>PC =</td>
</tr>
<tr>
<td>LDS #$82</td>
<td>($0081) =</td>
</tr>
<tr>
<td>JSR $0C10</td>
<td></td>
</tr>
</tbody>
</table>
11. Assume that the stack pointer has the value $0a00$. A HCS12 program calls a subroutine, and the subroutine pushes registers A, X, and Y onto the stack. What does the stack pointer contain now?

12. The C function below is called with the following input arguments: an array M containing N 8-bit numbers, and the size N. Describe what the function does.

```c
int func(int M[], int N)
{
    int i;
    int x = M[0];
    for (i=1; i<N; i++)
        if (M[i] < x)
            x = M[i];
    return x;
}
```

13. Write C code (using a loop) to compute the sum of the squares of the first 100 odd integers.

14. Write a C function that converts all uppercase ASCII letters in a string, to lowercase. The string is passed into the function as an input argument.

15. Write C code that takes an array of 10 integers called “buff”, computes the difference between the maximum and minimum values in the array, and stores it into an integer variable called “diff”. When calculating the difference, don’t worry about possible overflow:
16. The following C program performs the “factorial” operation using “recursive” function calls. Assume that the stack pointer originally has the value 0x900 when the program starts. Assume that you put a breakpoint at the instruction marked with $\$$, and look at the stack pointer (SP) register when the program breaks there.
Choose the most likely correct answer below:
(a) SP still contains 0x900.
(b) SP contains 0x8FD (i.e., slightly smaller than the original value).
(c) SP contains 0x903 (i.e., slightly larger than the original value).
(d) SP contains 0x8EF (i.e., substantially smaller than the original value).
(e) SP contains 0x914 (i.e., substantially larger than the original value).

```c
#include <hidef.h>      /* common defines and macros */
#include "derivative.h"      /* derivative-specific definitions */

unsigned char fact(unsigned char n)
{
    if (n==0)
        return 1;   // $\$$
    else
        return n*fact(n-1);
}

void main(void)
{
    unsigned char  n, result;

    EnableInterrupts;

    n = 5;
    result = fact(n);

    for(;;) ; /* loop forever */
}
```