EECS 373 Fall 2015 Homework #3

Due Monday, September 28th at 1:40 PM sharp (i.e. at the start of class).

Name: uniqname:

This is an individual assignment; all work should be your own.
This document is an editable pdf. You should submit your work by typing in the fields of this document. If you have technical difficulties, you may print this document, handwrite answers, and scan, however you **must** submit this document.

1. Using the ARMv7-M Architecture Reference Manual. **[10 points]**
   a. In straightforward English, explain what the LSR (register) instruction does. **[2 points]**

   b. Write the hexadecimal for the machine code you would expect to get for the following instructions. **[4 points, 2 each]**

      i. `ADDS R5, R5, R6`

      ii. `ADD R5, R6`

   c. What does the processor do differently when executing (i) and (ii)? **[1 point]**

   d. Give values for R5 and R6 that could result in different machine state after executing (i) and (ii) **[1 point]**

   e. What is the fewest number of bits that can encode (i)? What if the destination had been R10? **[1 point]**

   f. What is the fewest number of bits that can encode (ii)? What if the destination had been R10? **[1 point]**

2. For each of the following program segments, assume you start with all memory locations in question equal to zero. Indicate the values found in these memory locations when the programs end. Write all answers in hex. **[10 points, 5 for each part]**

a.)

```
BASE_EMC = 0xB7000000;
uint32_t *x = (uint32_t*)BASE_EMC;
*x = 0xabcdfe98;
*(x-1) = 0x76543210;
```

Base Address	00	01	02	03
0xB7000004				
0xB7000000				
0xB6FFFFFC				
0xB6FFFFF8				

b.)

```
mov  r1, #0x100
movw r2, #136         % Be careful with constants
movt r2, #8
strh r2, [r1, #2]!    % The ARMv7-M Architecture Reference Manual
strh r2, [r1], #-7    % may be important to look at for these.
str  r2, [r1, #1]
```

Address	Value
0FC	
0FD	
0FE	
0FF	
100	
101	
102	
103	

3. Write a program in UAL assembly that does the same thing as the following C code. You should assume "print" is an ABI compliant function that takes a single integer argument and does something with it. Have the main return to whatever called it just as any function might. Do not use IT statements or conditional instructions (e.g. ADDNE). Comment each line of code with what it is doing. **[10 points]**
Answers without comments will NOT be graded and will receive 0 points.

```
int main() {
  int i, a=1;
  for(i=0; i<9; i+=2) {
    a = a + i;
    print(a);
  }
  return a;
}
```

4. Write a program in C that does the same thing as the following ARM assembly language code. Your C code must not exceed three lines and should compile without any warnings or errors. **[10 Points]**

```
movw  r0, #0x1008
movt  r0, #0xB700
ldr   r1, [r0]
bic   r1, r1, #5
str   r1, [r0]
```

5. Write an ABI compliant function using the fewest instructions possible named “mean” which takes four integer arguments and returns the average. Do not worry about fractions or overflow. **[10 points]**