EECS 373 Fall 2017 Homework #2

Due September 18th before the start of class. Late homework is not accepted.

Name: _____ unique name: _____

You are to turn in this assignment filling in the blanks as needed. ***Assignments that are unstapled, are not done on this worksheet, or are difficult to read will lose at least 50% of the possible points and we may not grade them at all.*** This is an individual assignment; all work should be your own. 50 points.

---

1. Using the ARMv7-M Architecture Reference Manual.
   a. In straightforward English, explain the difference between the 'B', 'BX' and 'BL' instructions. **[4 points]**

   b. Write the hexadecimal for the machine code you would expect to get for the following instructions in Thumb mode.
   [6 points, 2 each)
      a. `LDR R1, [R3, #3]`

      b. `MOVW R4, #2000`

      c. `BX R14`

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* (*note that they aren't always sequential*!) memory locations when the programs end. Write all answers in hex. **[10 points, 5 for each part]**

a.
```
BASE_EMC = 0x64000000;
uint32_t *a = (uint32_t*)BASE_EMC;
*(a-1) = 0xabcdef89;
*a = 0x01234567;
```

Address	value
0x63ffffe	
0x63fffff	
0x64000000	
0x64000001	
0x64000002	
0x64000003	
0x64000004	

b.
```
mov r2, #100
movw r1, #2000
movt r1, #100
strh r1, [r2], 2
str r1, [r2,1]!
strb r1,[r2,-1]
```

Address	Value
100	
101	
102	
103	
104	
105	
106	

Hint: Page A6-15 of the ARMv7-M Architecture Reference Manual may be useful here.

3. Write a program in UAL assembly which does the same thing as the following C code. Have the function return to whatever called it just as any function might. **[12 points]**

```
uint32_t avg_5(uint32_t* a){
    uint32_t sum = 0;
    int i = 0;
    for(i = 0; i < 5; i++){
        sum += a[i];
    }
    return sum / 5;
}
```

4. Write a program in C that does the same thing as the following ARM assembly language code. Your C code must not exceed four lines and should compile without any warnings or errors. **[8 points]**

```
movw r1, #0x1248
movt r1, #0xB650
movw r2, #0xB700
movt r2, #0x1008
eor r0, r0, r2
str r0, [r1]
```

5. Write an ABI compliant UAL function named “mulAcc” which takes 3 integer arguments a, b, and c; multiplies b and c together; and returns that value added to a. **[10 points]**