EE 354 **August 30, 2019**
Assembly Language Worksheet

1. For each problem below assume that GPIOA has been setup for all output. Briefly explain what each sequence does.

A)
```
    ldr r2, =GPIOA_OD;
    movs r3, #1;
    movs r1, #12;
Lp1  str r3, [r2];
     lsls r3, #1;
     subs r1, #1;
    BNE Lp1;
```

B)
```
    ldr r3, =GPIOA_OD;
    movs r1, #0;
    movs r2, #5;
Lp2 str r1, [r3]
    adds r1, #1;
    cmp r1, r2;
    BNE Lp2;
```

C)
```
    ldr r3, =GPIOA_OD
    movs r0, #1
    ldr r1, =100;
Lp4 ldr r2, =200;
Lp5 str r1, [r3]
      mvn r1, r1
      subs r2, #1;
      BNE Lp5;
    subs r1, #1;
    BNE Lp4;
```

2. The following loop runs 40 times. What would you have to change to get it to go 400 times?

```
    movs r2, #40;
Lp2 ;Loop body goes here
    adds r1, #1;
    cmp r1, r2;
    BNE Lp2;
```

3. The instruction

`ldr r3, =0x40020014`

is not a "real" assembly language instruction. It is a pseudo assembly language instruction. Explain.

4. We often find assembly code that looks something like the following at the beginning of an assembly language program:

```
gpioa_moder   equ   0x50000000
gpioa_pupdr   equ   0x50000004
gpioa_ospeede equ   0x50000008
gpioa_otyper  equ   0x5000000c
gpioa_idr     equ   0x50000010
gpioa_odr     equ   0x50000014
RCC_IOPENR     equ   0x4002102C
```

Why is this used, what does it do, and is it necessary?

5. When we use assembly and c-code together we often see statements in the c-code like this:

```
 extern void DtoASetUp(int);
 extern void DtoA(int DtoANum, int value);
```

What do these do?

6. Explain how the ARM c-compiler passes parameters.

7. The code below has a main program which calls a subroutine which calls a second subroutine. The code in incorrect and will not run. Fix it.

```
Main program        | Sub1                | Sub2
...                 |    ...              |    ...
   bl Sub1          |    bl Sub2          |    pop {pc}
...                 |    pop {pc}         |
                    |                     |
                    |                     |
                    |                     |
                    |                     |
                    |                     |
                    |                     |
```

8. What is the difference between the following instructions?

```
lsrs R2, R3, #5;      shift right R3 5 times and store in R2
asrs R2, R3, #5;      arithmetic shift right R3 5 times and store in R2
```

9. What does the following assembler directive do and how could we use it?

```
MyData  DCD 0x87654321, 0x45362718
```

10. For assembly language functions we make use of three directives: PROC, ENDP, and END. Explain what each of these does.