Name: _____ Student ID: _____ Date: _____
Name: _____ Student ID: _____

Worksheet #3
Introduction to ARM Processor and Basic Instructions

Objectives

- To understand the architecture and basic operation of ARM processor
- To access (read/write) ARM's registers using assembly language instruction
- To understand the usage and operation of ARM's basic arithmetic instructions

1. Transform the following English sentences to ARM assembly program.

	AREA transform, CODE, READONLY ENTRY Start
1. Put a decimal number "200" to register R0	……
2. Put a hexadecimal number "4F" to register R1	……
3. Copy data in register R0 to register R2	……
4. Copy data in register R1 to register R3	……
5. Do R0 + R1 and store result in R4	……
6. Do R3 – R1 and store result in R5	……
7. Do R2 + R5 and store result in R2	……
8. Do R0 – R1 and store result in R0	……
	Loop B Loop END

What values do the following registers store after executing the above program ?(in hexadecimal)

R0 = ……..

R1 = ……..

R2 = ……..

R3 = ……..

R4 = ……..

R5 = ……..

2. Given the following code, write down the data stored in register R0, R1 and R2 (in hexadecimal) **after** each instruction is executed.

Code	R0	R1	R2

AREA program1, CODE, READONLY
ENTRY
start MOV r0, #20
MOV r1, #-16
MOV r2, #0x52
MOV r0, #-0x52
 	
loop B loop
END
	……
	……
	……
	……
		……
		……
		……
		……
			……
			……
			……
			……

Explain for each instruction, how do you get the answers?

3. Given the following code, write down the data stored in register R0, R1 and R2 (in hexadecimal) **after** each instruction is executed.

Code	R0	R1	R2
AREA program1, CODE, READONLY ENTRY			
start MOV r0, #20	……..	……	……
MOV r1, #10	……..	……	……
MOV r2, #0x60	……..	……	……
ADD r2, r2, r0	……..	……	……
ADD r0, r1, r2	……..	……	……
loop B loop			
END			

Explain for each instruction, how do you have the answers

4. Given the following code, write down the data stored in register R0, R1 and R2 (in hexadecimal) **after** each instruction is executed.

Code	R0	R1	R2
AREA program2, CODE, READONLY ENTRY			
start MOV r0, #100	……..	……..	…….
MOV r1, #0x14	……..	……..	…….
MOV r2, r1	……..	……..	…….
SUB r1, r0, r2	……..	……..	…….
SUB r1, r1, r0	……..	……..	…….
loop B loop END			

Explain for each instruction how do you have the answers

5. Write ARM Assembly program to calculate this expression by using only register R0 and R1 (output of the computation will be stored in register R0)

$$R0 = 5A_{16} + 18_{16} - 3E_{16}$$

```
        AREA    calculate_1,  CODE,   READONLY
        ENTRY
```

Data in R0 after the execution (in hexadecimal) :

……

6. Write ARM Assembly program to calculate this expression by using only register R0 and R1 (output of the computation will be stored in register R0)

$$R0 = (125_{10} - 2B_{16}) + 4A_{16} - 250_{10}$$

```
AREA    calculate_2,  CODE,   READONLY
ENTRY
```

Data in R0 after the execution (in hexadecimal) :

……

7. Write ARM Assembly program to calculate this expression by using only register R0 and R1 (output of the computation will be stored in register R1)

$$R1 = (4F_{16} + 6E_{16}) - 150_{10} - 1F_{16}$$

```
AREA    calculate_3,  CODE,   READONLY
ENTRY
```

Data in R1 after the execution (in hexadecimal) :

…….

8. Complete the following program in ARM Assembly. This program will **swap** between the data stored in register R0 and R1. So in the end, R0 will store 0xBB and R1 will store 0xAA
    - Move immediate number to register directly **is not allow**
    - You can use only register R0, R1 and R2

```
                AREA    swapdata1,  CODE,   READONLY
                ENTRY
start           MOV     R0,     #0xAA
                MOV     R1,     #0xBB









loop            B       loop
                END
```

9. Complete the following program in ARM Assembly. This program will **swap** between the data stored in register R0, R1, and R2. It means that after program is done executing, R0 will store data of R1, R1 will store data of R2 and R2 will store data of R0 (In the exercise, after program is executed : R0 = 0xBB, R1 = 0xCC, and R2 = 0xAA)
    - Move immediate number to register directly **is not allow**
    - You can use only register R0, R1, R2 and R3

```
                AREA    swapdata2,  CODE,   READONLY
                ENTRY
start           MOV     R0,     #0xAA
                MOV     R1,     #0xBB
                MOV     R2,     #0xCC









loop            B       loop
                END
```

10. Given that, before running the program, the values in R0-R4 are byte-size ASCII code of upper case English letters. Write an assembly language program that converts a sequence of five ASCII codes in R0-R4 to five ASCII codes of lower case English letters and stores the converted codes in R5-R9. E.g. the result, before and after running the program, is shown below

    (Before) R0 = 0x48 (H), R1 = 0x41 (A), R2 = 0x50 (P), R3 = 0x50 (P), R4 = 0x59 (Y)

    (After) R5 = 0x68 (h), R6 = 0x61 (a), R7 = 0x70 (p), R8 = 0x70 (p), R9 = 0x79 (y)

11. Write an assembly language program to calculate the following expression

$$3x + x - 2.$$

Suppose that x is stored in R0 and the result in R7.