Lab #1
Basic Assembly Language and ASCII

Sections 1 and 2 of this exercise will count towards your final coursework mark for CS1021. Submit your solutions using Blackboard no later than 23:59pm on Monday 16th October 2017.

When submitting your solutions on Blackboard, please submit only `.s` Assembly Language source files. Do not place the `.s` files in a `.zip` or other archive file. Do not rename the .s file.

Download and extract the template projects for this lab from the CS1021 Blackboard site. This will be the first step for all CS1021 Lab exercises and you should begin doing this automatically from now on.

1 Programs to Evaluate Expressions

Write and test an ARM Assembly Language program to evaluate the following expressions:

(a) $3x^2 + 5x$

(b) $2x^2 + 6xy + 3y^2$

(c) $x^3 - 4x^2 + 3x + 8$

Assume that x is stored in `R1`, y is stored `R2` and the result is to be stored in `R0`. Use the **Expressions** project to develop your solutions. Put your solutions to all three problems in the same file (`Expressions.s`).

When answering the questions above, you may find the following ARM Assembly Language instructions useful:

MOV Rd, Rm Move (copy) the value from Rm into Rd

LDR Rd, =x Load the value x into Rd

ADD Rd, Rn, Rm Add Rm to Rn, storing the result in Rd

SUB Rd, Rn, Rm Subtract Rm from Rn, storing the result in Rd

MUL Rd, Rn, Rm Multiply Rn by Rm, storing the result in Rd (Note: Rd and Rn cannot be the same)

Test your programs by using the `LDR` instruction to load test values for x and y into `R1` and `R2`. e.g.:

```
LDR     R1, =5   ; x = 5
LDR     R2, =6   ; y = 6
```

2 ASCII

(a) Design, write and test an ARM Assembly Language program that will add two single-digit decimal values, **represented by their corresponding ASCII character codes**. The byte-size ASCII codes representing the two digits are stored in `R1` and `R2`. The result, which should also be represented by its corresponding ASCII character code, should be stored in `R0`.

You may assume that both of the values to be added are in the range ['0' ... '4']. The suggested approach is to convert the ASCII character codes to the values that they represent, add these values and convert the result back to its equivalent ASCII character code.

For example, if register `R1` contained the value `0x32` representing the character `'2'` and `R2` contained the value `0x34` representing the character `'4'`, then your program should store the value `0x36` in `R0`, representing the character `'6'` (because $2 + 4 = 6$).

Use the **AsciiAdd** template project for this exercise.

(b) A sequence of four ASCII characters, each in the range '0' ... '9', can represent an unsigned decimal value in the range 0 ... 9999 in text form. Design and write an ARM assembly language program that will convert such a sequence to the value represented by the digit characters. Assume that the four byte-size ASCII characters will be stored in `R1` to `R4`, with the most significant digit stored in `R4`. Store the result of the conversion in `R0`.

For example, given the following sequence of ASCII characters ...

'2', '0', '3', '4'

... your program should store the value 2034_{10} (`0x00007F2`) in `R0`.

Use the **AsciiValue** project to develop your program.