

## COMP2121 - Tutorial 4

1. Write AVR assembly code segments for the following scenarios,

- a) Initialize an array A of size 20 (each element is one byte) with values ranging from 1 to 20.
- b) Initialize an array B of size 20 (each element is two bytes) with values ranging from -1 to -20.
- c) Add the arrays A and B together and store the result into an array C.
- d) Store the string 12345678 into program memory using `.db` and `.dw`.
- e) Load the values stored in the program memory in (d) and store them into data memory in the reverse order.

2. How do you multiply a two byte number by a one byte number? (Explain using a simple example). Do we have to consider the carry bit in the STATUS register for this case?

3. Investigate the different ways of writing AVR assembly code for the following scenarios,

- a) Copying a pair of registers into another pair of register.
- b) Multiply a number by 4.
- c) Divide a number by 4.

4. When are `MUL`, `MULS` and `MULSU` instructions used and how are they are used?

Write AVR assembly code to perform multiplication for the following set of numbers,

- |           |                 |             |                  |
|-----------|-----------------|-------------|------------------|
| a) 10, 12 | (1 byte result) | d) 32, 258  | (2 bytes result) |
| b) -11,11 | (1 byte result) | e) -352, 28 | (2 bytes result) |
| c) -4,-14 | (1 byte result) | f) -27,-375 | (2 byte result)  |

5. 1 Minimally modify the code below to add two numbers (in `r17:r16` and `r19:r18`) when the result is bigger than 255.

```
ldi r16, 1
ldi r17, 0
ldi r18, 255
ldi r19, 0
add r16, r18
add r17, r19
```

5.2 Write AVR code to add two 32 bits values?(Using `R16-R23` to hold all values.)

```
a = 0x00000100
b = 0x002000FF
```