# Assignment – 1

## Due: Fri. 16 August (at 11:00 AM)

**Submit a printed copy of your answers, with a cover sheet which indicates your name, ID and the course number.**

**NOTE: Your assembly code must have comments if necessary.**
**Do not use pseudo instructions (except *la* if necessary.)**

1. An array of integers is stored in memory in **big-endian** form. The base address of the array is given in register $t0, and it has 2048 elements. Write *the MIPS assembly instructions* for a **little-endian** machine, which divides each element of the array by 16 and stores the result in memory starting from the address given in register $t1.

2. (a) Write a **function**, which takes a null-terminated string (an array of characters) and finds how many digits (i.e. 0, 1, 2, …, 9) is in the string. The input argument of the function is the address of string (i.e. the base address of the array of characters). The function returns a number, which indicates how many digits is in the given string.

   (b) Write the main program, which calls this function using the example string: “Numbers are 0, 25 and 123 for this.” to test it.
   ***Test your code using SPIM***.

3. **A** is a two-dimensional array (matrix) of double-precision floating-point numbers and **B** is an array of 32-bit integers. Write *the MIPS assembly instructions* to perform the following arithmetic operation:

$$C = \sum_{i=0}^{9} \sum_{j=0}^{4} (A[i][j] + B[j])$$

   **C** is a double-precision floating-point number, which should be stored in register $s2.
   Assume that the processor is ***big-endian***. The base addresses for arrays **A** and **B** are given in registers $s0 and $s1 respectively.

4. Using function ***fibonacci(n)*** the sequence of numbers 1, 1, 2, 3, 5, 8, 13, 21, ... for values of ***n*** equal to 1, 2, 3, 4, 5, 6, 7, 8, … can be generated.

The result for each ***n*** is calculated as follows:

fibonacci(1) = 1, fibonacci(2) = 1,
fibonacci(n) = fibonacci(n-1) + fibonacci(n-2) for n > 2

(*Note that the function is called recursively).*

```
    int fibonacci(int n)
    {
       if (n < 1)       return 0;

        else if (n == 1 || n == 2)
            return 1;
        else
            return (fibonacci(n - 1) + fibonacci(n - 2));
   }
```

(a) Write the MIPS assembly code for function ***fibonacci***.

(b) Write the main program, which prints the Fibonacci numbers up to the calculated value for ***n*** on SPIM Console. (For example, if **n** is 10, then the following sequence will be printed).

**1, 1, 2, 3, 5, 8, 13, 21, 34,**

***Test your code using SPIM***.