Lab #2
Subroutines

Part 3 and 5 of this exercise will count towards your final coursework mark for CS1022. Submit your solutions to Parts 3 and 5 using Blackboard no later than 23:59 on Monday 13th February 2017.

1 String Reversal Using a Stack

Design and write an ARM Assembly Language program to reverse a NULL-terminated ASCII string stored in memory, **without creating a new copy of the string**. (i.e. The reversed string must overwrite the original string in memory). Use the system stack to store a temporary copy of the string.

Use the StackReverse project for this exercise. `R1` contains the address of the start of the NULL-terminated string in memory.

2 Division Subroutine

You do not need to submit your solution to this part of the exercise separately. It should instead be included with your solution to Part 3. (Use the same Val2Dec project that you will be using for Part 3.)

Design, write and test an ARM Assembly Language subroutine that will divide two unsigned, non-negative integers and return the quotient (whole part) and remainder. Begin by designing and documenting a suitable interface for your subroutine. (The *interface* specifies which registers are used to pass each parameter to the subroutine or return each result to the calling program.) Provide a program to test your subroutine by invoking (calling) it.

3 Value to Decimal String

For this part of the lab exercise we are revisiting the conversion of a binary value to an ASCII string representing the binary value in decimal form. Last term we saw that this could be done easily by repeatedly dividing the binary value by 10_{10}. Each division produces a quotient (whole part) and a remainder. The remainder can be converted to an ASCII digit while the quotient is further divided to produce the next digit.

The only problem with this approach is that it produces the digits in right-to-left order whereas we want to obtain the digits in left-to-right order.

Using your division subroutine from Part 2 above and making use of the system stack, design, write and test an ARM Assembly Language program to convert the binary value stored in R4 into an ASCII string representing the value in decimal form. The string should be stored in memory.

Solutions that do not make proper use of a division subroutine or the system stack will receive lower grades. (You may choose to use a more efficient solution to the one described above. Your solution, however, must demonstrate the use of subroutines to at least the same extent.)

4 Swap Subroutine

You do not need to demonstrate or submit your solution to this part of the exercise separately. It should instead be included with your solution to Part 5. (Use the same Sort project that you will be using for Part 5.)

Write an ARM Assembly Language subroutine, **swap(array, i, j)** that will swap two elements in a 1-dimensional array of word-size integers. Your subroutine should accept the address of the array and the indices of the two elements to be swapped as parameters. Begin by designing an appropriate interface for your subroutine. Test your subroutine fully.

5 Sort

Translate the pseudo-code shown below into an ARM Assembly Language subroutine. You must make use of your **swap(array, i, j)** subroutine from Part 4.

```
sort(array, N)
{
    do
    {
        swapped = false;
        for (i = 1; i < N; i++)
        {
            if (array[i-1] > array[i])
            {
                swap(array, i-1, i);
                swapped = true;
            }
        }
    } while (swapped);
}
```