CS1022 Tutorial #2
Stacks and 2-D Arrays

1 Stack Operations without LDM and STM

(a) Provide an ARM Assembly Language program that will push the contents of R4, R5 and R6 on to the system stack. You may not use the LDM or STM instructions.

(b) Provide an ARM Assembly Language program that will pop the three topmost words off the top of the stack, restoring their values into the registers from which they were originally stored in part (a) above. You may not use the LDM or STM instructions.

(c) Provide a diagram to illustrate the state of the stack (stack pointer and contents) after executing the following instruction:

```
1        STR R0, [SP, #-4]!
```

2 Stack Operations with LDM and STM

(a) Provide STM and LDM ARM assembly language instructions to perform the following stack operations on the full descending stack pointed to by R13 (sp).

(i) Push the contents of R2 on to the stack

(ii) Pop the contents of R2 off the stack

(iii) Push the contents of R2, R3, R4, R7, R8 and R9 on the stack

(iv) Pop the contents of R2, R3, R4, R7, R8 and R9 off the stack

(b) Provide diagrams to illustrate the state of the stack (stack pointer and contents) after executing each of the following instructions:

```
1        STMFD   SP!, {R4, R6, R8-R11}
2        LDMFD   SP!, {R10, R11}          ; Note - no !
```

3 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. Assume that `R1` contains the address of the start of the NULL-terminated string in memory.

4 Magic Squares

A "magic square" is an $N \times N$ array in which

- every element is unique and
- the sum of the elements in each row, each column and each of the two diagonals is

$$\frac{N(N^2+1)}{2}$$

The following is an example of a magic square:

8	1	6
3	5	7
4	9	2

Design and write an ARM assembly language program that will determine whether an $N \times N$ array stored in memory is a magic square. **You may assume that the elements are unique** and need only check the sum of the rows, columns and diagonals.

Assume that R1 contains the address of the start of the 2-D array in memory, R2 contains the array dimension, N, the array is stored in row-major order and each element is one word (32 bits).