

# CmpSci 201

## Homework 4

1. Given the following code:

```
function1:
    STMDB    R13!, {R8, R9, R10, R14}
    <some operations>
    BL      function2
    <some other operations>
    LDMIA   R13!, {R8, R9, R10, R14}
    MOV     PC, R14

function2:
    STMDB   R13!, {R8, R14}
    MUL     R8, R4, R4
    MUL     R8, R8, R8
    LDMIA   R13!, {R8, R14}
    MOV     PC, R14
```

What will the stack look like just before `MUL R8, R4, R4` is executed?

2. The following code does not correctly implement the calling convention. Insert the missing code necessary to fix it:

```
foo:
    STMDB   R13!, {R8, R14}
    MOV     R8, #4
    ADD     R8, R8, R4
    BL     bar

bar:
    MOV     R8, R4
ADD     R8, R8, #1
MOV     R4, R8
    LDMIA   R13!, {R8, R14}
    MOV     PC, R14
```

3. You need to calculate Fibonacci numbers, and so come up with the following pseudo-code:

```
int fib(int n)
{
    if(n == 0)
    {
        return 1;
    }else if(n == 1)
    {
        return 1;
    }else
    {
        return fib(n - 1) + fib(n - 2)
    }
}
```

This code computes the  $n^{th}$  Fibonacci number (assuming we're only handed positive values for  $n$ ). Convert the code to ARM assembly, you must implement it as a real function obeying the calling convention described in the lectures. Note that there are more efficient ways of computing Fibonacci numbers (but you don't have to use them, just implement the code above).