

# Systems Programming 1 – COMP2019

## *Tutorial and Lab Practice Three*

This work **will** be marked and is due by Wednesday 12th of April, and is worth **4%** of your assessment. Submit your work online via vUWS. You should explain your answers and demonstrate your programs to your Tutor during your scheduled lab session by the week that the work is due. **Email submission is not accepted.**

## Tutorial

1. From your chosen C resource, read the chapters about C data structures, recursive function, function call, function pointers and command line argument.
2. Review the terminology introduced and concepts taught in Lecture 3.
3. List all of the possible operations on a stack. Write pseudo code for each of these operations.

## Lab Practice

1. Write a recursive program to calculate and output the first  $n$  numbers of Fibonacci sequence, where  $n$  is input by a user. The sequence starts with 0, and is defined as  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$ . That is, the numbers are in the following integer sequence (**2%**):

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

2. Modify the above recursive program where argument  $n$  is given by a command line argument.
3. Write a program implementing basic stack operations. (**2%**)

You should use the following definitions:

```
typedef char stackitem;

struct stack
{
    stackitem d;
    struct stack *next;
};

typedef struct stack ELEMENT;
typedef ELEMENT *POINTER;
```

Your program should be able to perform the following operations:

- (a) Push one character onto the stack. Your function should have the prototype:  
`void push(POINTER *Top, stackitem a);`
- (b) Pop the top character from the stack. Your function should have the prototype:  
`stackitem pop(POINTER *Top);`
- (c) Push a string of characters into a stack. Your function should have the prototype:  
`void push_string(POINTER *Top, char *string);`  
Your implementation should make use of the `push()` function.
- (d) Delete the stack, i.e. remove all items from the stack. Your function should have the prototype:  
`void delete_stack(POINTER *Top);`  
Your implementation should make use of the `pop()` function.
- (e) Print the contents the stack. Your function should have the prototype:  
`void print_stack(POINTER Top);`  
It should not modify the stack in any way, only display its contents.

A sample code is provided, which include functions `push()`, `pop()` and a simple `main()`. You need to complete the missing code of the other functions.

## Optional Work (For those students who wish to practise more.)

1. Add the following functions to your stack program:
  - (a) a function to exchange the top two elements of the stack;
  - (b) a function to reverse the stack and print the reversed;
2. Change the `main()` function so that stack operations can be chosen from a simple menu.