Queen Mary
University of London

School of Electronic Engineering and Computer Science

ECS501U – C Programming (2017/18)
Laboratory Session Week_6

Learning Objectives

- To construct linked lists and stacks.
- To manipulate files as input and output.
- To implement programs with system calls.

Exercises

You should attempt the exercises below by using only the C constructs that you learnt up to teaching week 6, and:

1. Write pseudo code to describe the required algorithm to solve the exercise (or draw up a flowchart), before writing and testing the actual code.
2. Add comments to your code.
3. Make your code neat, by using indentation and parenthesis (where appropriate).
4. Give meaningful names to functions and variables.

Exercise 1
Write code that implements a stack, including the following functions[1]:

- function **push()** creates a new node and places it on top of the stack;
- function **pop()** removes a node from the top of the stack, frees the memory that was allocated to the removed node and returns the value that was in the removed node;
- function **isEmpty()** checks if the stack is empty or not, and thus whether a node can be removed from the stack;
- function **printStack()** displays to all the current nodes in the stack to the standard output stream (the screen), after each call to **push()** and **pop()**.

The functionality above will use a simple node defined as follows:

```
struct stackNode {
  int nodeData;
  struct stackNode *nextPtr;
}
```

Save your code and data structure to a file called **stack.c**.
Now add a **main()** function to **stack.c** that uses the node defined above, to test the code's functionality; your program should allow the user to add and remove a node from the stack, until it decides to terminate the program.

[1] You need to consider which arguments and return types these functions should have.

Exercise 2

Each line in a file contains a distance in miles (an integer) followed by the name of a town which lies at that distance from London. Build a file with 20 towns. Write a program that reads this file and prints in another file the name of the towns that are more than 100 miles from London. Save your program to a file called **`citiesfromlondon.c`**.

Exercise 3

Write a program that reads the input file of exercise 2 via redirection, and writes the result file via redirection too.

Exercise 4

What is the drawback of example 3 of the **File I/O** lecture?
Write a program that has the same input and output as example 3, but using non-formatted data files.
Hint: This program will be similar to example 4 of the **File I/O** lecture.

ECS501U – END of LAB Week_6