ECM2433: The C family

C Workshop 3

Exercise

On the study resources website on ELE (http://vle.exeter.ac.uk) you will find a zip file containing (1) a text data file called ***monarchs.txt***, (2) three files which make up the template for a C program called ***readMonarchsMain*** (two .c programs and one .h), and (3) a Linux shell script called ***compMonarchs*** for compiling and linking the aforementioned C program.

The monarchs.txt has the following contents:

```
GeorgeIII       1760
ElizabethII     1962
Anne            1701
Victoria        1837
```

These represent the names of British monarchs and the first year of their respective reigns. Your task in this workshop is to write a C program that reads this file, creates a linked list in order of increasing year and then prints out the contents of the linked list. The linked list should end up in the following order:

```
Anne            1701
GeorgeIII       1760
Victoria        1837
ElizabethII     1962
```

The programs that have been provided are a template for a possible C program. You may choose to use them, or you may choose to write your own programs from scratch. If you use the template, then the ***compMonarchs*** shell script will compile and link it for you. To use it you may need to tell Linux that it is executable; you can do this using the following command:

```
chmod u+x compMonarchs
```

(the "u+x" means "add executable access to the owner of the file"). The resulting executable C program is called ***readMonarchsMain***.

The instructions below provide a logical process for developing the program.

1. Reading command line arguments

Your program must accept the name of the data file (i.e. monarchs.txt) as a parameter from the command line, i.e. the program should be run like this:

```
readMonarchsMain monarchs.txt
```

Read in the name of the file and print it out to the terminal.

2. File input/output

Open the file for read access, read each record as name/year pairs and print them out to the terminal. The year should be read as an integer, not a string.

3. Construct a linked list

Each time you read in a record from the data file, create a node (the suggested structure for a node is given in the .h file provided) and add it in the correct place in the linked list.

This should make up the ***processFile*** function in the readMonarchs.c program, with the ***newNode*** function used to create and populate a new node.

4. Free up the memory allocated to the linked list

This should make up the ***deleteList*** function in the readMonarchs.c program.

5. Print the contents of the linked list in year order

This should make up the ***printList*** function in the readMonarchs.c program. It is most succinctly achieved using a recursive function.

6. Print the contents of the linked list in reverse year order

Amend the ***printList*** function in the readMonarchs.c program so that it is able to print the list in increasing or decreasing year order, depending on the value of a new parameter you pass to it. Call the function once to print the list in increasing order and a second time to print it in decreasing order