# Laboratory Report

## LABORATORY 8/9 – REPORT

**Student name:** Michael Lenehan

**Student ID:** 15410402

**Programme:** CE

I hereby declare that the attached submission is all my own work, that it has not previously been submitted for assessment, and that I have not knowingly allowed it to be used by another student. I understand that deceiving or attempting to deceive examiners by passing off the work of another as one's own is not permitted. I also understand that using another's student’s work or knowingly allowing another student to use my work is against the University regulations and that doing so will result in loss of marks and possible disciplinary proceedings.

**Signed:** Michael Lenehan **Date:**  07 April 2016

Note: Coursework examiners are entitled to reject any coursework which does not have a signed copy of this form attached or are submitted late.

Problem

The aim of this exercise was to create an array of structures to store a set of car information read in from a file – “indata.txt”. The program must then compute the tax of the cars based on the information which it reads in. After this process is deemed as completely working, the structure must then be updated to store the tax value and write it to the file, as well as printing it to the screen. A user menu must also be implemented.

Plan

Include standard C libraries.

Declare the list of functions to be used: read\_records, compute\_tax, write\_record, print\_results.

Define the structure “car\_type”.

Include the main function, which prints the welcome message, and then calls all other functions.

Write the code for read function, which will read the information contained in the file.

Write the code for the tax computation function, which will give a value for tac for each car, given its age and engine size.

Write the code to write the values to the file.

Write the code to print all of the results to the screen.

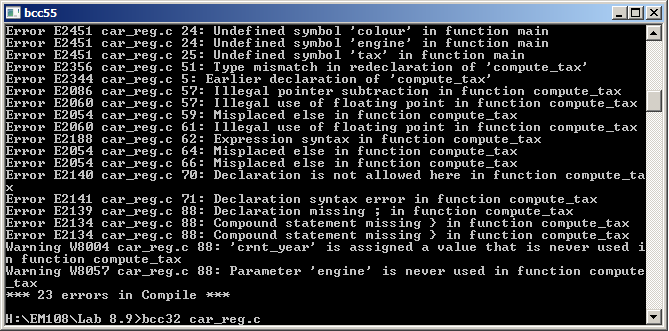
Development

Development of the code began with the inclusion of the standard C libraries: stdio.h, stdlib.h and string.h. All of the necessary functions were then defined: read\_records, compute\_tax, write\_record, print\_results. The main function was then coded, with variables norecs, i and tax declared. The files “indata.txt”, and “results.txt” are then opened. A welcome message is then printed. The read function reads the “indata.txt” file. A for loop is added which loops through the compute tax, write records, and print results function. A closing message then prints and the program ends. The compute tax function runs using a series of if statements to determine how much tax should be charged as determined by the car age and engine size. The write function then writes all values recorded in the read function and the tax values to the file “results.txt”. The print function then prints all of these values to the screen.

Once this section of the code was deemed as correctly working, the tax value is added to the structure. A user\_menu function was then declared, and the user menu function defined using switch statements. Case 1 will run the program, and case two will exit the program.

Testing

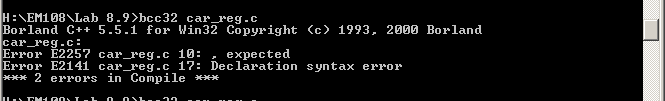
Multiple Errors were encountered. Many errors present were syntax errors in the functions and the “if” statement in the compute\_tax function.



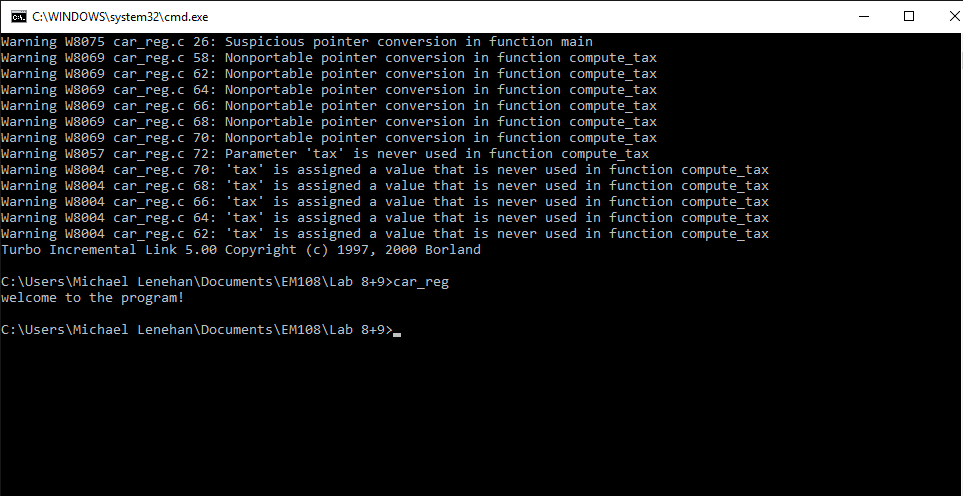
Total number of errors were reduced by fixing some simple syntax errors such as missing semi colons.



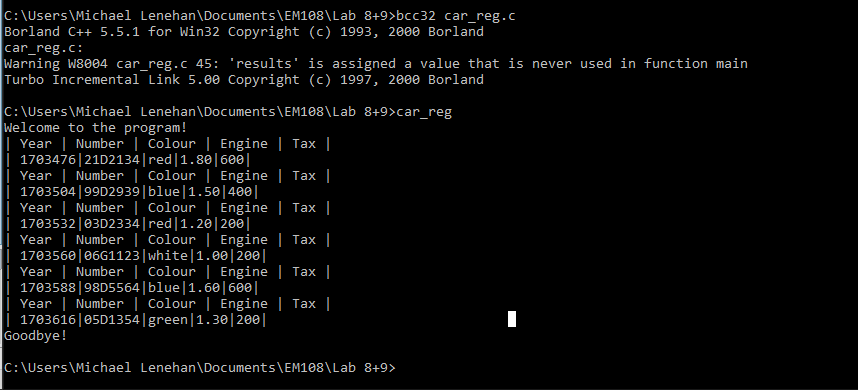
Number of errors reduced to two.



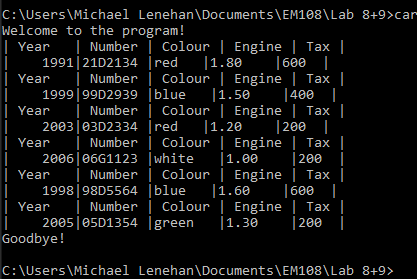
Multiple Warnings were present and the program will not run correctly. This was due to incorrect addition of parameters to the functions.



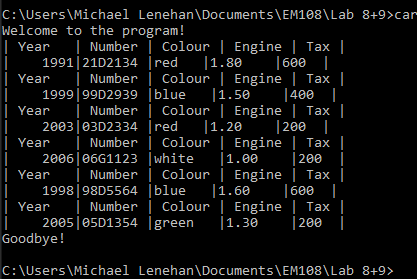
Initial Code deemed to be correctly working, but with incorrect value of year being printed.



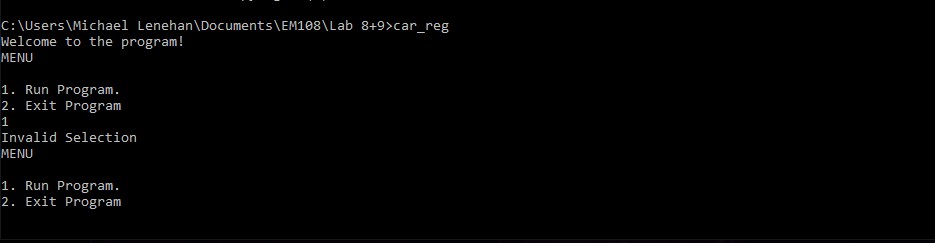
Formatting of the table was improved, incorrect value of year still present.



Printed value of year was corrected with the removal of an ampersand (&) in the print function.

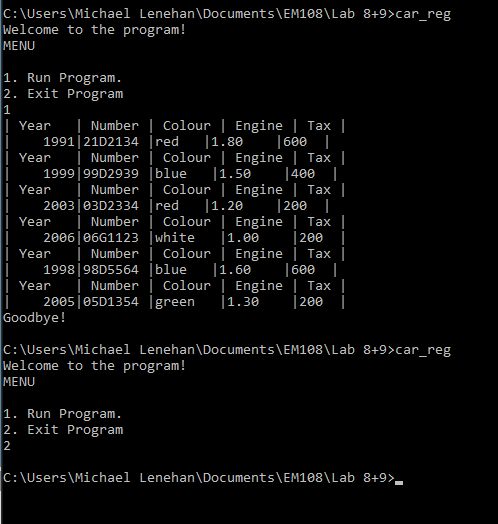


Menu added but working incorrectly.



Scan for selection was incorrectly scanning for a string while the variable was defined as an integer.

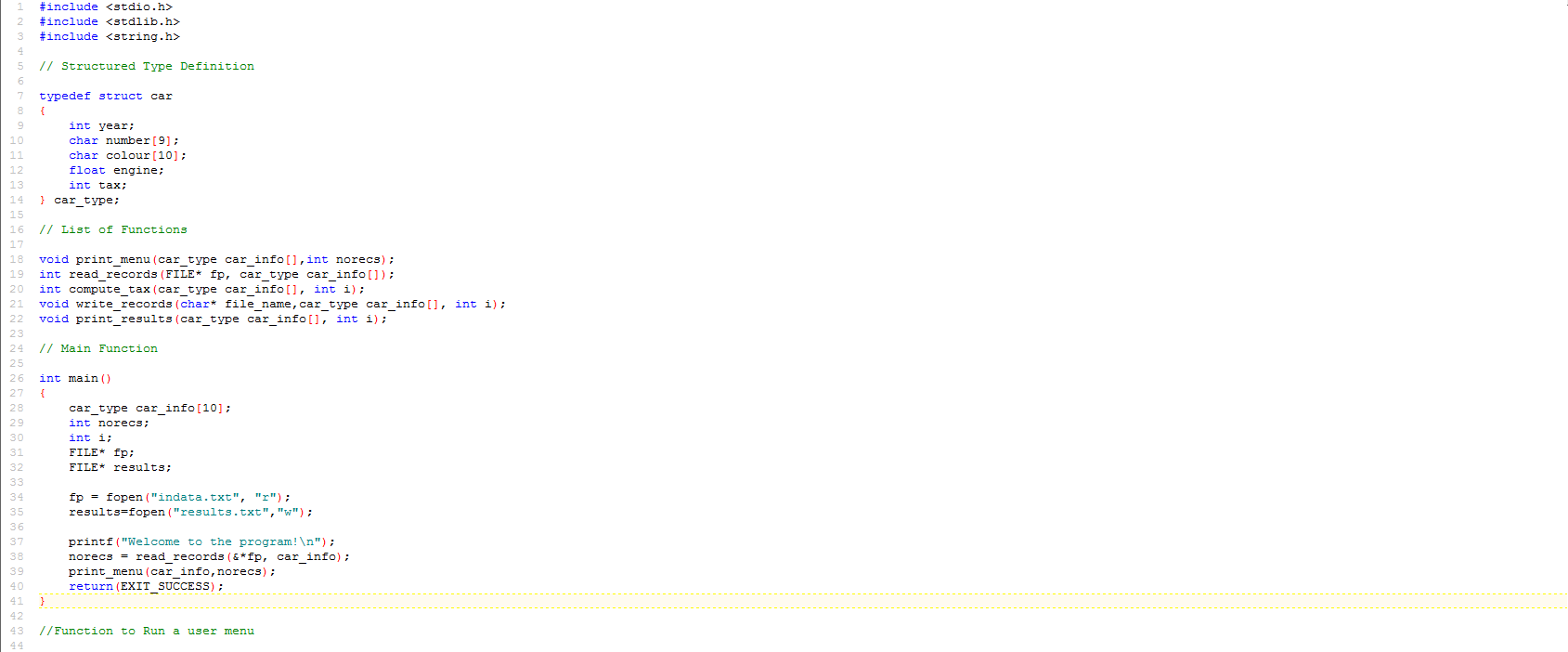
This error was corrected and the program deemed to be correctly working.

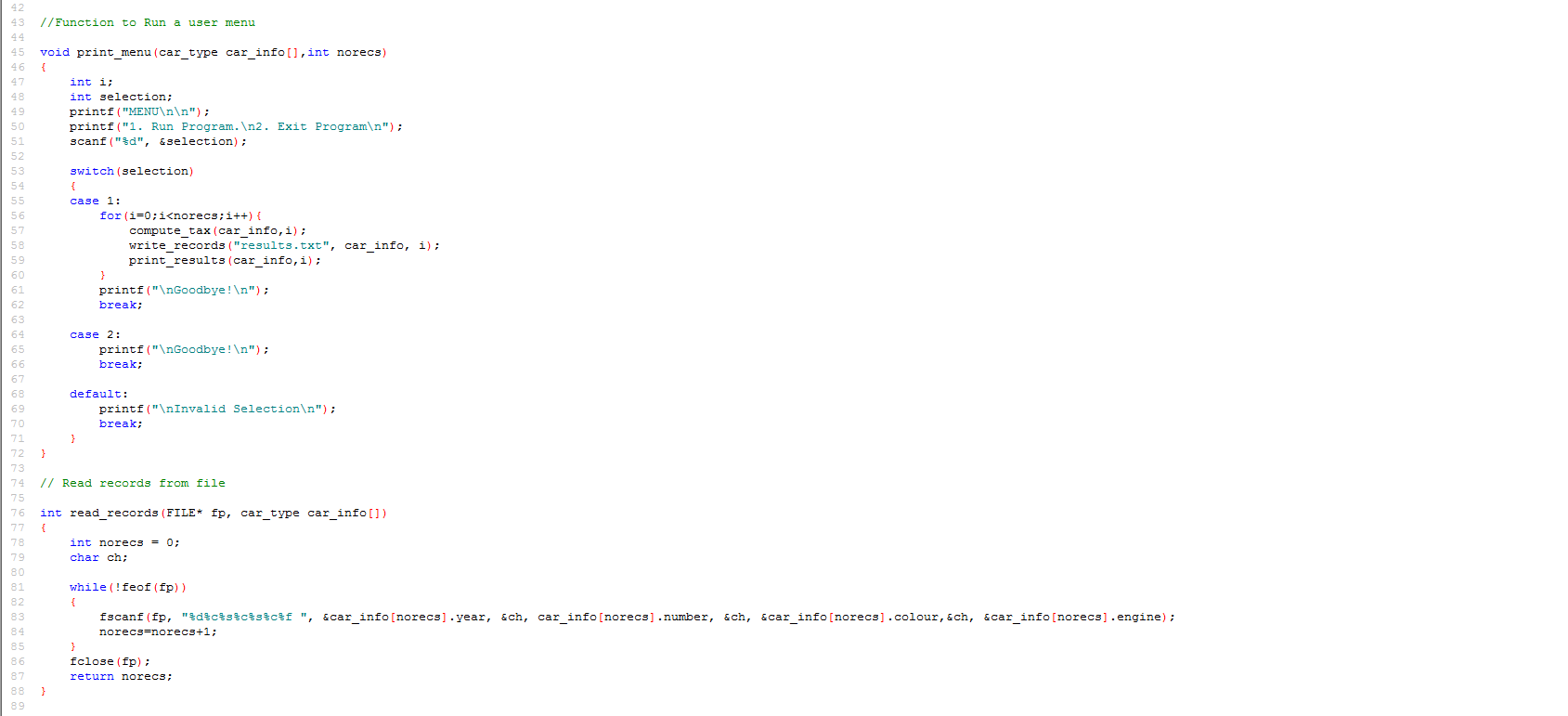


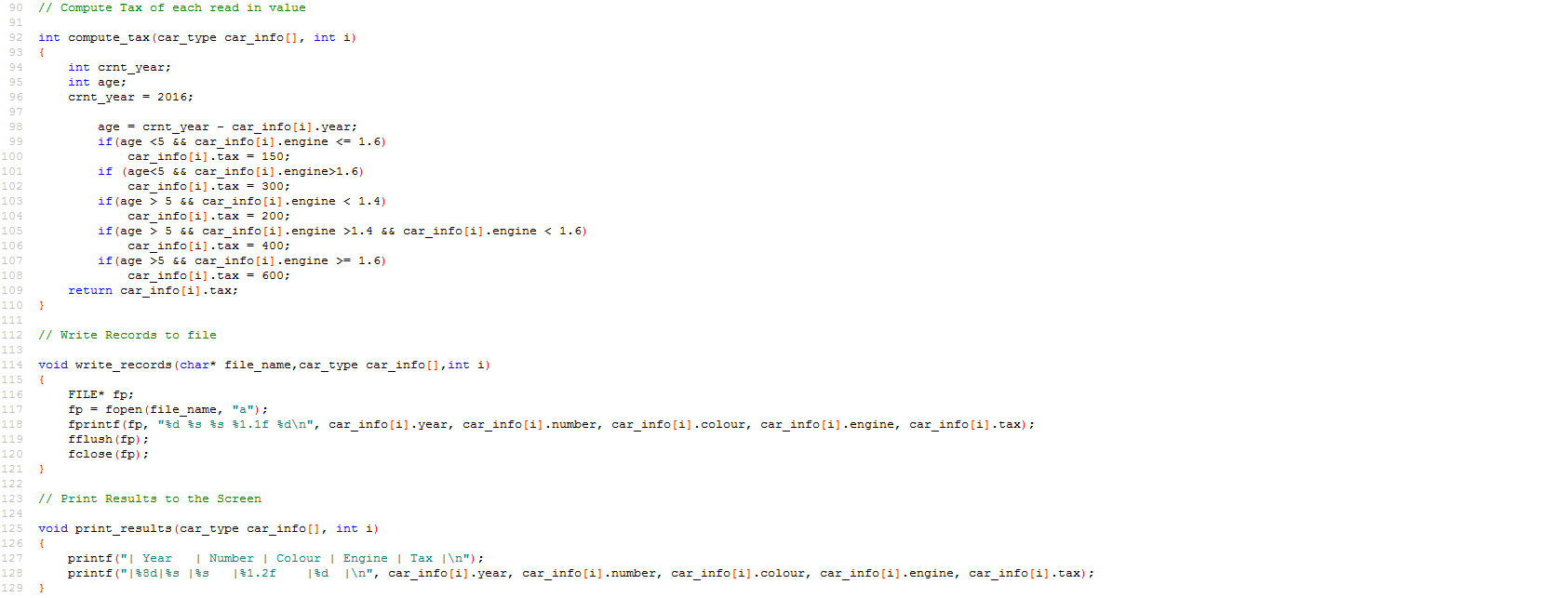
Conclusion

This exercise tested my knowledge of structures, functions, files, and switch statements. For this exercise it was necessary to improve my understanding of structures, and how to define and implement them in the code. It was also necessary to use a switch statement to create a user menu, through which the function will run. The use of files, and reading and writing to the files was also necessary in order to read values to the functions from a file, and to then write the results to a file. In conclusion this exercise has both tested and improved my knowledge in these areas.

Code

****

****

****