COMP10120 Practical Set 9: Linked Lists and Structs

Please read the questions carefully. Name each program based on your student number, the practical set number and question number. For this set (set9), question 1 should be named 1234567s9q1.c where your student number replaces 1234567. All questions that you are submitting can be zipped into a single file called 1234567s9.zip, where 1234567 is your student number and s9 refers to set 9. Please also include a readme.txt file which says which compiler you used to test your implementation. This zipped file must be submitted via Moodle for grading.

1. Write a C code snippet which creates 5 **structs** from the integer array (myCustomers) and writes each struct to a binary file named **cutomers.dat**. You can assume each struct is defined with the members shown below. The integer array (myCustomers) has 2 columns. Column 0 contains the area/zip code and column 1 contains the number of customers living there available as shown below.

```
int myCustomers [5][2] = {{86956, 1}, {36568, 3}, {6565, 0}, {999555,
                              22}, {85446, 88}};

                struct zip_custs{

                        int zip_code;
                        int customer_count;
                    };
```

2. Modify the **C Program** *Linked List* given in the lecture and on Moodle to include a new function which replaces a character in a linked list with another character. The function should take three arguments – the list, the character to be replaced and the new character. Note the list does not need to be in sorted order. The function should print the elements of the old list and then the new list to reflect the changes.

3. A stack is a data structure which can be represented as a linked list. A stack is a constrained version of a linked list because it is a last in first out (LIFO) data structure. Nodes can only be added or removed at the top of the stack. Modify the **C Program** Linked List given in the lecture and on Moodle to create **a stack.** The same functions for adding, removing, printing nodes should be present.