May 27, 2022

# CSE108 – Computer Programming Laboratory

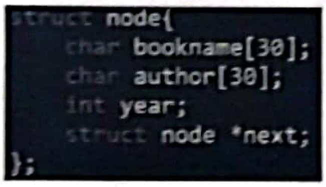
# Spring 2022, May 27

# Lab #13

**Hand-in Policy**: Via Teams. No late submissions will be accepted. File name that you submit should be as following: *StudentNo.c*
**Collaboration Policy**: No collaboration is permitted.

In this lab, you should write a program which reads a .txt file and save each row of the file as a node of a linked list. Your program should have insert, delete and print functions for this linked list. Please read the whole document before starting to code.

The node structure should be as following:



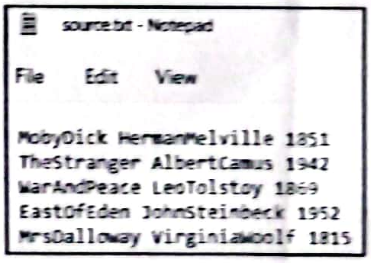
```
struct node{
    char bookname[30];
    char author[30];
    int year;
    struct node *next;
};
```

***The program should have the following functions:***

1) **struct node * read_file()**

   This function should open and read the "source.txt" file. For each row, it should call insert_node function to add that row's variables to the linked list. Finally, it should return root of the linked list to main().

   ***You should create the source.txt file manually as you see on the right. (There should be no space between the words of the book name or author.)***



```
MobyDick HermanMelville 1851
TheStranger AlbertCamus 1942
WarAndPeace LeoTolstoy 1869
EastOfEden JohnSteinbeck 1952
MrsDalloway VirginiaWoolf 1815
```

2) **struct node* insert_node(struct node *root, char * bookname, char * author, int year)**

   This function should take 3 variables and root as input. Then, it should add this new information to the linked list in an <u>ascending order</u> (by year). Finally it returns the root of the linked list to main(). To do this, you may follow the algorithm below:

   - ***Case 1:*** If there is no node in the list (if root is NULL), then the root should be updated with the incoming data.
   - ***Case 2:*** If there is at least one node in the list, create a new node with the incoming data and check if this node should be added to left side of the list. In other terms, if the new node should be the new root.
   - ***Case 3:*** If above two doesn't hold, create a new node with incoming data. Use a loop to find the correct place for new node and insert it.
   - ***Return root.***

3) **void print_nodes(struct node * root)**

   This function should take root and print the whole list as shown in the screenshots below (You can assume that input 'root' is not NULL).

May 27, 2022

4) **struct node* delete_node(struct node *root, char * bookname)**

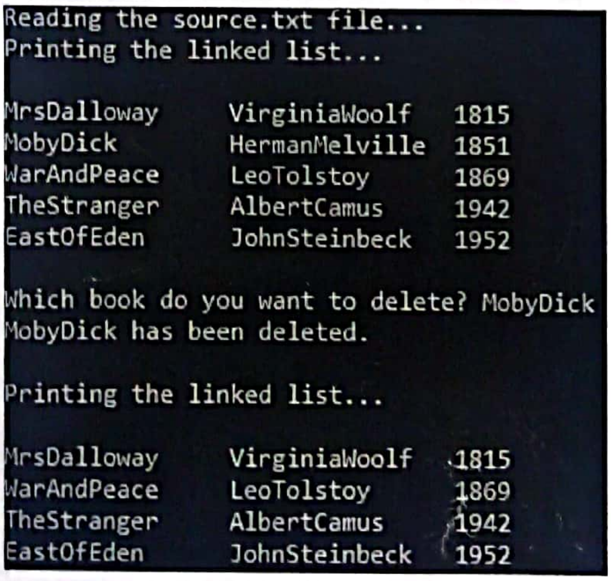
This function deletes the node which has the incoming data (I.e. name of the book) from the list *(You can use strcmp() function from string.h library to compare two strings)*. Then it returns the root of the list to main(). To do this, you may follow the algorithm below:

- ***Case 1:*** If the root itself is the node you are looking for the root should be changed.
- ***Case 2:*** If the root is not what we are looking for, use a loop to find this node in the list. If you find this node, you should delete it. Make sure the list is still connected if you are deleting a node which is not root or the end node.
- Use free() function to free the node.
- ***Return root.***

***Program flow:***

1) The program should read the file, save all rows to the linked list in an ascending order.
2) The program should print the linked list. print_nodes function should be called in main.
3) The program should ask for a book name to delete.
4) According to the input, the program should search for that node.
5) If the node presents, it should be deleted. If the node is not in the list, the program should warn the user.
6) Finally, the program should print the linked list. print_nodes function should be called in main.

***Two different program outputs obtained by using source.txt file shown above:***



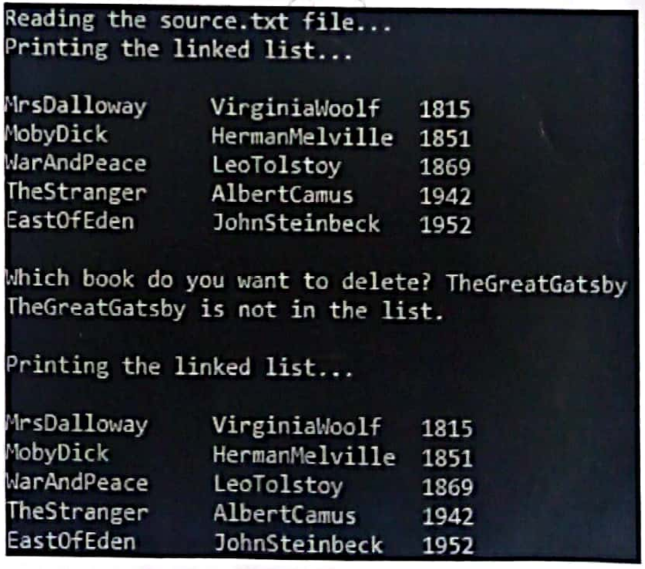
```
Reading the source.txt file...
Printing the linked list...

MrsDalloway     VirginiaWoolf   1815
MobyDick        HermanMelville  1851
WarAndPeace     LeoTolstoy      1869
TheStranger     AlbertCamus     1942
EastOfEden      JohnSteinbeck   1952

Which book do you want to delete? MobyDick
MobyDick has been deleted.

Printing the linked list...

MrsDalloway     VirginiaWoolf   1815
WarAndPeace     LeoTolstoy      1869
TheStranger     AlbertCamus     1942
EastOfEden      JohnSteinbeck   1952
```



```
Reading the source.txt file...
Printing the linked list...

MrsDalloway     VirginiaWoolf   1815
MobyDick        HermanMelville  1851
WarAndPeace     LeoTolstoy      1869
TheStranger     AlbertCamus     1942
EastOfEden      JohnSteinbeck   1952

Which book do you want to delete? TheGreatGatsby
TheGreatGatsby is not in the list.

Printing the linked list...

MrsDalloway     VirginiaWoolf   1815
MobyDick        HermanMelville  1851
WarAndPeace     LeoTolstoy      1869
TheStranger     AlbertCamus     1942
EastOfEden      JohnSteinbeck   1952
```

***Grading:***

- read_file function: 10 points (no partial points)
- insert_node function: 40 points (no partial points)
- print_nodes function: 10 points (no partial points)
- delete_node function: 40 points (deleting the first and last element: 10 + 10 points, deleting any other element: 20 points)
- You should write comments and a makefile, otherwise: -10 points.
- You will not get any credits if you don't use a proper linked list and do the task in other ways.