KOÇ UNIVERSITY

COMP 201 - Fall 2021
Lab 5 - Linked List & Makefile
(with Const, Struct & Multi-file C Programs)
Due: 15 November 2021 23:59

1 Introduction

Linked List is an important data structure that is used widely. Const, Struct, Multi-file C programs & Makefile are also useful concepts for C programming. Here, we provide you a code base where Linked List is implemented using Const, Struct and Multi-file C programming concepts. In this exercise, you will **implement a missing function of Linked List** and **write a makefile** for the code to compile and run.

2 Exercise

2.1 Download exercise.zip from Blackboard

2.2 Unzip it

2.3 Copy exercise folder to linuxpool.ku.edu.tr servers

```
scp -r exercise USERNAME@linuxpool.ku.edu.tr:/Users/USERNAME/
```

2.4 Write a makefile that includes following target rules (rule names are given in bold

- **all** cleans (depend on ”clean”), compiles (depend on ”install”) and runs the program
- **install** compiles everything by depending on ”program” target rule (below)
- **clean** removes the object files (.o) and program executable
- **program** links the object files generated by main.o, and linkedlist.o target rules (below) to generate an executable called ”program”
- **main.o** compiles main.c to generate object file main.o
- **linkedlist.o** compiles linkedlist.c to generate object file linkedlist.o

Note: You are expected to write dependencies between target rules correctly so that make program can run the target rules in the correct order.

2.5 Implement the missing function

You should **only change the linkedlist.c file by implementing the following function:**

```
int LinkedList_remove_value(LinkedList* l, int value)
```

It finds the first occurrence of *value* (int) in the LinkedList *l* and removes it from the LinkedList. Also, it returns the index of the the first occurance. If not found, returns -1. Note that indexing starts from 0 (as with arrays). Please do not forget to free the unused (removed) space since it is dynamically allocated.

Example Let *l* contains [3,2,5,1] and *value*=2 then *l* should be changed to [3,5,1] and return 1.

Note: you may **inspect the test cases in the main.c file** to further understand the expected behaviour. Also, you may see the other functions implemented in linkedlist.c file in order to understand the implementation of the Linked List. However, do not change anything in those functions.

2.6 Test your work

If you wrote everything correctly, you should see this kind of output when you run `make`.

```
(base) simitii@pop-os:~/Documents/KOC-PHD/TA/Fall2021/COMP201/Labs/Lab5/exercise/solution$ make
rm -rf *.o program
gcc -c linkedlist.c
gcc -c main.c
gcc -o program linkedlist.o main.o
./program
Test Case 1 - Middle Remove: PASS!
Test Case 2 - Beginning Remove: PASS!
Test Case 3 - End Remove: PASS!
Test Case 4 - Not Found: PASS!
```

Figure 1: Testing

2.7 Copy exercise folder from linuxpool.ku.edu.tr servers back to your local machine

```
scp -r USERNAME@linuxpool.ku.edu.tr:/Users/USERNAME/exercise .
```

2.8 Zip the updated ”exercise” folder

2.9 Submit your exercise.zip file to Blackboard.