Assignment 5
CISC 220, Fall 2018

Administrative Notes:
All of the administrative notes from previous assignments apply to this assignment as well. I will just summarize the main points here and you can refer back to the previous assignments instructions for more details:

- Your program should work in CASLab Linux servers.
- No group work allowed.
- Hand in your file to OnQ in the “Assignment 5” Dropbox area.
- Follow instructions carefully or you may lose points.
- There is a 24-hour “grace period” after the official deadline for each assignment; submissions made within this period lose 10% but are still accepted.

Deadline:
The deadline for this assignment is **Monday, November 19 at 8 a.m.** The grace period lasts until Tuesday, November 20, 2018 at 8 a.m.

Required Program:
This is your structures assignment and your task is to write a fairly simple C program. It can be done with the features of C we've discussed in the “Structures” topic. The program requires to create a stack structure using linked lists. The size of the stack is only 3 units. You are going to push 4 elements into the stack, and pop also 4 elements. You may use what we have covered and explained in the class.

The object of your program is to demonstrate the skills you have learned in C programming. Your program will work as follows when it is executed:

1- It will create a stack structure using struct, typdef, and linked list. Your program must use these data structure to build and use the stack.
2- You will push 3 strings into the stack successfully, and get an error when you push the fourth.
3- You will pop the 3 strings from the stack successfully, and get an error when you try to pop the fourth from an empty stack.
4- Pushing and popping functions are run by the program in order, pushing first then popping. Pushing function should use the following set of strings { “Fist Element” , “Second Element” , “Third Element” , “Fourth Element” }. The program should print the status of the stack after each push including the error when the fourth string is pushed.

5- Popping function will pop all elements of the stack plus one more pope when the stack is empty. The program should print the status of the stack after each pop including the error when you try to pop an empty stack.
6- Your program should use “malloc” to reserve memory units to be used by the stack, and the allocated memory should be deallocated by using free() function.
7- Quit

For example: When you run the program, it should produce the following output:

```
Creating a Stack that can take only 3 string elements.
Pushing
The stack status: { “First Element” }
Pushing
The stack status : { “First Element” , “Second Element” }
Pushing
The stack status: { “First Element” , “Second Element” , “ Third Element” }
Pushing
The stack is full and cannot take any more elements.
Popping
The stack status: { “First Element” , “Second Element” }
Popping
The stack status: { “First Element” }
Popping
The stack is empty
Popping
Cannot pop an empty stack
End of program
```

You have to use:

1- Struct, typedef, and linked lists.
2- Push function.
3- Pop function
4- Control memory with malloc and free functions
5- The output should be identical to the given one.

Your assignment worth 20 points.

More Requirements:

- Your program must be in a single file called CISC220-Assn5.XXXX.c
  the XXXX represent your last four digits of your students ID.

Marking Scheme:

- Use of struct : 2
- Use of typedef: 2
- Use of linked list: 2
- Controlling memory with malloc and free functions: 2
- Correctness of the program. Does it have push and pop functions? Does the main part works correctly: 8
- Formatting of the output, does it match the given format: 2
- Clear and organized Coding style: 2

Total: 20

If you do not follow the assignment requirements -- especially if you deviate from them in a way that costs us time while marking -- you will lose administrative points even if your program works correctly. So please read the directions carefully.

Possible Penalties:
10% if you hand in during the late period (24 hours after the deadline).
10-20% if there are administrative issues that cause the graders extra time.

Style:
Use tabs and proper indentation to organize your program. Use common naming conventions we have been using in our examples at classes.