Data Structure and Programming Spring 2024
Programming Assignment 1

Shao-Ming, Chen
r11942095@ntu.edu.tw

Chiao-Yun, Chin
b10901154@ntu.edu.tw

Pei-Yuan, Wu
peiyuanwu@ntu.edu.tw

RELEASE DATE: 03/14/2024
DUE DATE: 03/28/2024

1 Problem Statement

This programming assignment asks you to read in a series of commands e.x. (PUSH 10, PUSH 9, POP, ...), and execute it using stack. We have written a code template for you, and *you should only fill in the **TODO** in stack.py.*

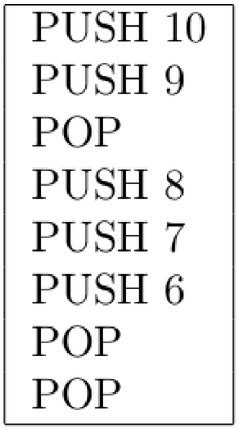
Notice:

- For stack.py, please use python3 to run.
- You should not modify the codes which are not specified by **TODO**.
- You should call the class **node()** when storing your data structure. That is, your data structure is limited to linked lists.
- You should be able to implement **push and pop** operations in stack with **O(1)**.

2 Input/Output Specification

2.1 Input Format

Inputs are a series of commands separated by newlines. The following is an example:

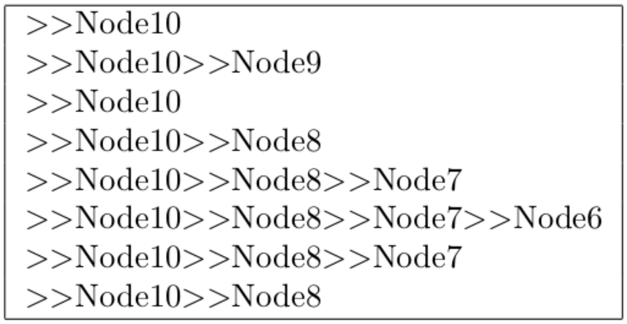


```
PUSH 10
PUSH 9
POP
PUSH 8
PUSH 7
PUSH 6
POP
POP
```

Figure 1: Input format

2.2 Output Format

You should print out your stack of each execution process. Below is an example of executing the above input using stack.



```
>>Node10
>>Node10>>Node9
>>Node10
>>Node10>>Node8
>>Node10>>Node8>>Node7
>>Node10>>Node8>>Node7>>Node6
>>Node10>>Node8>>Node7
>>Node10>>Node8
```

Figure 2: Output format

3 Submission

Please put stack.py into a directory named studentID and compress the directory into studentID.zip. Finally, upload studentID.zip to NTU COOL.The homework is due on 3/28, at 23:59.

4 Evaluation

All of our test cases will not execute **pop()** on an empty stack. You won't need to handle this exception.

Scoring Criteria:

1. **Correctness and Time Complexity, 100%** : We will evaluate your code on five test cases. We provide you three of these test cases: input_1.txt, input_2.txt, and input_3.txt, and their corresponding golden output: golden_1.txt, golden_2.txt, and golden_3.txt. We also provide a script: evaluation.sh so that you can check whether your programs pass these test cases. Furthermore, this script will display the runtime. Passing each test case with time complexity O(1) will get 20% of the total score.

5 Appendix

As a reference, the python runtime of input_3.txt is around 0.0067s – 0.019s. Remember that it is a reference value and we will grade based on your code if your code runtime exceeds the range.