SSN College of Engineering

Department of Information Technology

UIT2201 — Programming and Data Structures

2022 – 2023

Exercise — 09

May 31, 2023

- This homework is due by 10PM on June 7, 2023
- Grace period may be given up to midnight of June 7, 2023
- You can upload only one ZIP file
- The naming convention is "<Your first name (first letter capital and all the other letters small)>-UIT2201-ex-09.zip"
- Judicious use of Python features and standard modules, version control using 'git', adhering to Python coding standards are expected
- You are expected to use PSP0.1 process for all the code that you write!

The purpose of this exercise is to understand the design, implementation, and applications of Stack,and Queue ADTs.

Part A

1. Design and implement stack and queue data structure as wrapper around the Python List. Using stack and queue check whether a given number is a palindrome.
2. Design and implement data structure to maintain two stacks in a single array. All the basic stack operations should include an argument to select one of the stacks. For example, 'push(0, item)' should push item into the first stack, while 'push(1, item)' should push item into the second stack. Your stack methods should not raise stack full exception, unless every array cell is used.

Part B

3. A food delivery system accepts a maximum of *M* orders. Orders are served in first come first basis. Orders once placed cannot be cancelled. Write a Python code to simulate the system using circular queues.
4. Design and implement data structure for a queue like abstraction, referred to as PQueue, that internally maintains two arrays — one for a high priority queue and the other for the low priority queue. Enqueue operation will mention the priority of the 'item' to be added to the PQueue — for example, 'enqueue(0, item)' will add the 'item' to the high priority queue, while 'enqueue(1, item)' will add it to the low priority queue. 'dequeue()' operation will dequeue from the high priority queue. If the high priority queue is empty, then first item from the low priority queue will be dequeued.
5. Write a python code to evaluate the given arithmetic expression. To handle the operation precedence in given expression, convert the infix expression to postfix and then evaluate the postfix expression for the required answer. Use stack data structure for this purpose.