### Laboratory 3 - Python List Operations and Data Structures

#### Laboratory Objectives

* Write a Python program that extensively utilizes Python list operations, including advanced manipulations.
* Employ loops, conditional statements, input/output, and functions to demonstrate various data structure manipulations including using lists as stacks and queues.

#### Program Instructions

1. **Advanced List Operations**
   * Create a list of items and implement the following list operations:
     + append\_item (list, item): Adds an item to the end of the list.
     + insert\_item (list, index, item): Inserts an item at a specified index.
     + remove\_item (list, item): Removes the first occurrence of the specified item.
     + pop\_item (list, index=-1): Pops an item from the list at the given index.
     + clear\_list (list): Clears all items from the list.
     + sort\_list (list): Sorts the list in ascending order.
     + reverse\_list (list): Reverses the order of items in the list.
     + index\_of\_item (list, item): Returns the index of the first occurrence of the item.
     + count\_item (list, item): Counts how many times the item appears in the list.
     + slice\_list (list, start, end): Returns a slice of the list from start to end index.
2. **Stack and Queue Implementations**
   * **Stack Operations**:
     + Use append() to push an item onto the stack.
     + Use pop() to remove the top item from the stack.
   * **Queue Operations**:
     + Use append() to enqueue an item at the end of the queue.
     + Implement dequeue(queue) which uses the first item of the list and then removes it, effectively dequeuing it.
3. **Interactive Menu for Data Structure Manipulation**
   * Implement a menu-driven interface in main.py to select and perform operations on the list, stack, or queue.
   * Menu Options:
     + List Operations (Add, Insert, Remove, Pop, Clear, Sort, Reverse, Index, Count, Slice)
     + Stack Operations (Push, Pop)
     + Queue Operations (Enqueue, Dequeue)
     + Exit

### 4. Run the program

Execute the following command:

python3 -m main

### 5. Example Output

## \*\*\* MAIN MENU \*\*\*

## 1. List Operations

## 2. Stack Operations

## 3. Queue Operations

## 4. Exit

## Enter your choice: 1

## \*\*\* LIST OPERATIONS \*\*\*

## 1. Append item

## 2. Insert item

## 3. Remove item

## 4. Pop item

## 5. Clear list

## 6. Sort list

## 7. Reverse list

## 8. Index of item

## 9. Count of item

## 10. Slice list

## 11. Return to Main Menu

## Enter your choice: 1

## Enter item to append: apple

## Item 'apple' appended successfully!

## \*\*\* CURRENT LIST: ['apple'] \*\*\*

## Enter your choice: 11

## \*\*\* MAIN MENU \*\*\*

## 2. Stack Operations

## Enter your choice: 2

## \*\*\* STACK OPERATIONS \*\*\*

## 1. Push item

## 2. Pop item

## 3. Return to Main Menu

## Enter your choice: 1

## Enter item to push: cherry

## Item 'cherry' pushed to stack!

## \*\*\* CURRENT STACK: ['cherry'] \*\*\*

## Enter your choice: 3

## \*\*\* MAIN MENU \*\*\*

## 3. Queue Operations

## Enter your choice: 3

## \*\*\* QUEUE OPERATIONS \*\*\*

## 1. Enqueue item

## 2. Dequeue item

## 3. Return to Main Menu

## Enter your choice: 1

## Enter item to enqueue: pear

## Item 'pear' enqueued!

## \*\*\* CURRENT QUEUE: ['pear'] \*\*\*

## Enter your choice: 3

## \*\*\* MAIN MENU \*\*\*

## 4. Exit

## Enter your choice: 4

## Exiting program...

**Grading Criteria**

• **60%** for the coding part (.py files).

• **20%** for executing at least 4 different operations (2 from list, 1 from stack and 1 from Queue) outputs from VsCode.

• **20%** for executing 4 different operations (2 from list, 1 from stack and 1 from Queue but different from the operations executed in VsCode) in the terminal.

• Total: **100%**.

## Submission Instructions:

Save your folder named as <CWID>\_<LastName>\_lab.zip.  
It should contain list\_operations.py, main.py and add screenshot of output.  
Submit the .zip file by the deadline.