Name: Xu Yong Lin

Admin No: 212033J

IT2153-01

**Topic 4: Sort**

* Practical
  + For insertion sort, I implemented the sorting to be in both ascending and descending.

This is useful as it gives a wider range of inputs.

Difficulty: Easy

**Topic 5: Recursion**

* Tutorial
  + For Question 3, another method for checking if a number is palindrome.
  + \*only works for numbers as it involves dividing the number by 10 to find the place value.

By checking numbers with their respective values, it provides a understanding of checking if a number is palindrome.

Difficulty level: Hard

**Topic 6: Advance Sort**

* Quick sort
  + Implemented the median to be the pivot

Median is calculated by comparing each value in the array list. By implementing the median to be the pivot, it increases the efficiency of the sorting process.

Difficulty level: Hard

**Topic 7: Array-based Sequences**

* Caesar
  + Accept small caps
    - Difficulty level: Medium
  + Accept Mix caps
    - Difficulty level: Medium-Hard

Accepting small caps and mix caps allows a larger range of input, and also increases the ‘security’ of the Caesar encryption.

**Topic 8: Stack**

* Stack
  + Accept modular
    - Difficulty level: Medium
  + Accept 0
    - Difficulty level: Medium
  + Implement it in prefix way
    - Difficulty level: Hard

By accepting modular, it accepts another form of implementation of the equation.

When 0 is inputted for division or modular equations, it returns an prompt saying that ‘Numbers cannot be divided/modular by 0’, as it will return an DivisionByZero error.

Another enhancement I implemented would be to accept an equation in its prefix form (found in Topic8/Practical/qn2\_prefix.py). This is implemented by reversing the stack when it is first passed in, and then evaluating the equation afterwards.

**Topic 9: Queue**

* Expand the circular array queue
  + Multiplying the size of array by 2

I chose to expand my circular array queue by multiplying by 2 as it is more efficient than adding a fixed number to its maxSize of the array.

Difficulty level: Hard

**Topic 10: Tree**

* Practical
  + Printing out the binary search tree

Printing binary search tree in vertical format. This is useful as it provides a clear visualization of the binary tree.

Difficulty level: Hard

Number of Implementation: **10**