

Problem 8.1

a)

Stack.app

b)

queue.cpp

Problem 8.2

a)

pseudocode

```
reverse_linked_list (L)          //L is pointer of first node of the linked list and n is #elements
    unreversed = L->next
    L->next = NULL                // this creates two linked lists on the original linked list
    while unreversed is not NULL  //take one node away from the original linked list every
                                    //iteration
        temp = unreversed->next
        unreversed->next = L
        L = unreversed
        Unreversed = temp
```

This is in-situ because no extra space is required in the reverse processes. All it does is changing the place the pointers are pointing to.

b)

BST.cpp

```
Void BST::Inorder(BST* root, Node* &tail)
```

Time complexity for in order visit :  $O(nk)$  when assume operation is  $O(k)$

In this case, operation is a constant, therefore, time complexity of this algorithm is  $O(n)$