Pseudocode: Project 4

Function: Tree Application Class
Description: Creates a class, which will provide functions to access a doubly linked list.

Initialize number of nodes to zero

Function: Tree Constructer
Description: Provides a constructer to create an empty tree
Calls: N/A
Called by: Main
Input Parameters: N/A
Returns: N/A

Set root to null

Function: Insert
Description: Method to insert a node into the tree
Calls: get Abbr, Compare To
Called by: Main
Input Parameters: tree node
Returns: N/A

```
Get abbr of tree node
If root is null
   Set root to tree node
Else
   Create current node and set to root of tree
   Create parent node
   While true
      Set parent node to current
       If abbr is less than abbr of current node
         Set current to current.leftChild
         If current is null
            Set left child of parent to tree node
            Return
         End If
       Else
         Set current to current.rightChild
         If current is null
            Set right child of parent to tree node
            Return
         EndIf
       End Else
   End While
End Else
```

Function: Traverse
Description: Traverses the tree based on type of scan required
Calls: LNR Recursive, NLR Recursive, NLR Iterative, RNL Iterative
Called by: Main
Input Parameters: type of traversal
Returns: N/A

Switch (Input traverse type)
    Case 1: Print tree using LNR recursive scan
        Print header consisting of node num, abbr, population and state
        Call LNR Recursive method to traverse tree staring at the root
        Break;
    Case 2: Print tree using NRL recursive scan
        Print header consisting of node num, abbr, population and state
        Call NRL Recursive method to traverse tree staring at the root
        Break;
    Case 3: Print tree using NLR Iterative scan
        Print header consisting of node num, abbr, population and state
        Call NLR Iterative method to traverse tree staring at the root
        Break;
    Case 4: Print tree using RNL Iterative scan
        Print header consisting of node num, abbr, population and state
        Call RNL Iterative method to traverse tree staring at the root
        Break;
End switch

Function: LNR Recursive
Description: Performs a In order traversal on the tree using recursion
Calls: LNR Recursive (itself), display Node
Called by: Traverse
Input Parameters: tree node
Returns: N/A

If the tree node is not null
    Call itself to perform function on left child
    Display the node
    Call itself to perform function on right child
EndIf

Function: NLR Recursive
Description: Performs a In order traversal on the tree using recursion
Calls: NLR Recursive (itself), display Node
Called by: Traverse
Input Parameters: tree node
Returns: N/A

```
If the tree node is not null
    Display the node
    Call itself to perform function on left child
    Call itself to perform function on right child
EndIf
```

Function: RNL Iterative
Description: Performs a RNL Iterative scan on the tree
Calls: is Empty, Stack constructor, push, pop, display Node
Called by: Traverse
Input Parameters: tree node
Returns: N/A

```
Create stack of type tree node

Set node to the root of tree
While true
   While node isn't null
     Push node onto stack
     Set node to right child of node
   End while

  If the stack is not empty
     Pop last item on stack
     Display the node
     Set node to left child of node
  End if
  Else
     Break;
End while
```