Author: Kyle Den Hartog

For this application I chose to use a textual menu to select each data structure. A user will enter an option of 1, 2, or 3. If none of these are entered in number form, the application will close.

For data structure 1 (BST):

A user has chosen a BST. The user will now use numbers to navigate through using the BST. If they wish to select a new data structure, the user must close the program choosing "3" then restart it. I had issues displaying the BST level by level, so I opted to print them in order in a single line. (Using in-order traversal) (This fits the constraints of the problem and was easier to code.)

If user chooses "1"

This is how a user will add a new word to the BST. After the word has been added the program will automatically print out the entire tree in-order traversal. The user will then be prompted to reselect, creating a loop until the user wishes to close the program using option 3.

If user choose "2"

This is how the user will delete a word from the BST. The program will search through the tree using compareTo function and if a match is found it will remove that word. Otherwise the user will be informed that the word is not in the BST.

If user choose "3'

Program closed. (This is to prevent an infinite loop through user commands)

For Data Structure 2 (LinkedList as a list):

The user will choose between 1, 2, 3, or 4. A user will be given a prompt ahead of time to identify each choice. If a user does not enter one of these options the application is closed.

If user chooses "1"

This is how a user will add a word to the end of the list. The program will print the entire list then the user will be prompted to choose another option until the program is closed.

If user chooses "2"

This is how a user will add a word at a specific index. First the user will enter a word, then they will enter an index, finally the list will be printed and the user will choose another option until the program is closed.

If user chooses "3"

This is how a user removes an item. The user will be prompted to enter an index, which will be removed, then the user will choose another option until the program is closed.

If user chooses "4"

Program closed. (This is to prevent an infinite loop through user commands)

For Data Structure 3 (Queue as a circular array):

A user has the option to enter 1, 2, or 3 this will create Queue data structure. If one of these options is not entered the application will be closed.

If user choose “1”

This is how a user will enqueue an item into the data structure. The user will be prompted to enter a word, then the program will print out the entire Queue, if it is full, it will automatically double the size of the Queue.

If user chooses “2”

This is how a user will dequeue a word from the data structure. After this is complete the queue will be displayed and the user will be notified what word will be removed.

If user choose “3”

Program closed. (This is to prevent an infinite loop through user commands)