Read Me

Part 1:

The goal of this part was to create an ArrayList called CS401ArrayImpl that grows in size if full when employees objects are being added to the list. This was done by the following steps:

1. Created a Collection Interface file that describes the methods that will be needed for our ArrayList, since ArrayLists are a type of collections ADT.
2. Created CS401ArrayImpl class that implements CollectionInterface. There was a protected method called enlarge() that was created here to use in the add(E element) method. This protected method can only be accessed within the CS401ArrayImpl class and is used in the precondition if statement that checks if list if full.
3. An additional method was added to CS401ArrayImpl class to print all elements in the list.
4. The CS401ArrayImplClient class was made to test our implementation where at first the file is read into an array and then added to our list initiated list with size of 10. The program prints a message that list is full to indicate resizing and then prints all elements.

Part 2:

The goal of this part was to create a sorted Linked List called SortedLinkedList that adds and sorts employees from file based on ID. The steps to create this were:

1. Create a LLNode class that defines the logic of a node
2. Create a LinkedList class that defines the relationship between nodes
3. Create a SortedLinkedList class that extends the LinkedList class and modifies the add method to add Employees objects in a sorted way based on ID
4. Lastly, create the client class that adds employees form file, sorts them and print them to console.

Part 3:

The goal of this part was to create a double Linked List called DLL. Steps were:

1. Create a DLLNode class to define nodes logic
2. Create a DLLInterface
3. Create a DLL class that defines the Double Linked List which implements DLL interface and uses DLLNode as an attribute
4. Create a DLLClient class to test list's method add, remove, and contains.