```
::::::::::::::
Lab4Status.txt
::::::::::::::
Problem 1: compiles, runs correctly on all provided input
Problem 2: Completed::::::::::::::
Lab4Conclusions.txt
::::::::::::::
This lab was easily the most helpful lab as far as learning is concerned. I found
the implementation of a Circular Doubly Linked Structure to be insanely efficient
and is easily the most advanced material I have dabbled with thus far as a program
mer. While I won't say I'm an expert in calculating space/time complexity for algo
rithms, I still found the investigation to be informative and hopefully in the fut
ure I can utilize it to figure out what methods to implement for programs in the f
uture. ::::::::::::::
DNode.java
::::::::::::::
/**
 * Purpose: Data Structure and Algorithms Lab 4
 * Status: Complete and thoroughly tested
 * Last update: 02/13/23
 * Submitted:  02/13/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.02.13
 */

public class DNode
{
    private Object item;
    private DNode next; // next item
    private DNode back; // previous item

    public DNode(Object newItem)
    {
        item = newItem;
        next = this;
        back = this;
    } // end constructor

    public DNode(Object newItem, DNode nextNode, DNode lastNode)
    {
        item = newItem;
        next = nextNode;
        back = lastNode;
    } // end constructor

    public void setItem(Object newItem)
    {
        item = newItem;
    } // end setItem

    public Object getItem()
    {
        return item;
    } // end getItem

    public void setNext(DNode nextNode)
    {
        next = nextNode;
    } // end setNext

    public DNode getNext()
    {
        return next;
    } // end getNext

    public void setBack(DNode lastNode)
    {
        back = lastNode;
        // end setLast
    }

    public DNode getBack()
    {
        return back;
        // end getLast
    }
} // end class DNode
::::::::::::::
ListCDLSBased.java
::::::::::::::
/**
 * Purpose: Data Structure and Algorithms Lab 4
 * Status: Complete and thoroughly tested
 * Last update: 02/13/23
 * Submitted: 02/13/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 *
 * @author: Antonio Rosado
 * @version: 2023.02.13
 */

public class ListCDLSBased implements ListInterface
{
    private DNode head; // 'beginning' of list
    private int numItems; // number of items in list

    public ListCDLSBased()
    {
        head = null;
        numItems = 0;
    }

    /**
     * Check if DNode is empty
     *
     * @return head == null
     */
    public boolean isEmpty()
    {
        return head == null;
    }

    /**
     * Return size of DNode
     *
     * @return size of DNode
     */
    public int size()
    {
        return numItems;
    } // end size
```

```
    /**
     * Get item from DNode
     *
     * @param int index index to find
     * @return DNode curr item found
     */
    private DNode find(int index) throws ListIndexOutOfBoundsException
    {
        DNode curr = head;
        if(index >= 0 && index <= numItems)
        {
            if (index <= (numItems / 2))
            {
                for(int i = 0; i < index; i++)
                {
                    curr = curr.getNext();
                }
            }

            else
            {
                for(int i = numItems; i > index; i--)
                {
                    curr = curr.getBack();
                }
            }
        }
        return curr;
    }

    /**
     * Get item from DNode
     * @param int index index of item
     * @return Object nodeTtem item from specified index
     */
    public Object get(int index) throws ListIndexOutOfBoundsException
    {
        if (index >= 0 && index < numItems)
        {
            // get reference to node, then data in node
            DNode curr = find(index);
            Object nodeItem = curr.getItem();
            return nodeItem;
        }

        else
        {
            throw new ListIndexOutOfBoundsException("List index out of bounds exce
ption on get");
        } // end if
        // end get
    }

    /**
     * Add item to DNode
     * @param int    index index of item
     * @param Object item item Object
     */
    public void add(int index, Object item) throws ListIndexOutOfBoundsException
    {
        if(index >= 0 && index <= numItems)
        {
            DNode temp = new DNode(item);
            // if new structure, no items inserted
            if (index == 0)
            {
                if (numItems == 0)
                {
                    temp.setNext(temp); // new DNode is next
                    temp.setBack(temp);
                    head = temp; // first = new item
                }

                else
                {
                    // insertion into non-empty list
                    // store previous DNode which will then have its following ind
ex reference new
                    temp.setNext(head);
                    temp.setBack(head.getBack());
                    head.getBack().setNext(temp);
                    head.setBack(temp);
                    head = temp;
                }
            }

            else
            {
                // insertion into non-empty list
                // store previous DNode which will then have its following index r
eference
                DNode prev = find(index - 1);
                DNode curr = prev.getNext(); // curr = next thing after prev
                temp.setNext(curr); // curr = next
                temp.setBack(prev); // prev = prev
                prev.setNext(temp); // new insertion is next
                curr.setBack(temp);
            }
            numItems++;
        }

        else
        {
            throw new ListIndexOutOfBoundsException("List index out of bounds on a
dd.");
        }

    }

    /**
     * Remove item from DNode
     * @param int index index of item
     */
    public Object remove(int index) throws ListIndexOutOfBoundsException
    {
        Object result;
        // if list is empty, no items inserted yet
        if (index >= 0 && index < numItems)
        {
            if(numItems == 1) // list only contains single item
            {
                head = null;
```

```
                result = head;
            }

            else
            {   // if numItems > 1
                // delete the first DNode from the list
                // reassign positions
                DNode curr = find(index);
                result = curr.getItem();
                curr.getBack().setNext(curr.getNext());
                curr.getNext().setBack(curr.getBack());
                if (index == 0) // if item is first or 0
                {
                    head = curr.getNext(); // only item is first item, i.e. 'head'
                }
            }
            numItems--;
            return result;
        }
        else
        {
            throw new ListIndexOutOfBoundsException("List index out of bounds exce
ption on remove");
        } // end if
        // end remove
    }


    /**
     * Remove all items from DNode
     */
    public void removeAll()
    {
        numItems = 0; // if empty, no items
        head = null; // no head if empty
    }

    /**
     * Returns a string value of DNode items
     */
    public String toString()
    {
        String list = "";
        DNode curr = head;
        for (int index = 0; index < numItems; index++)
        {
            list += curr.getItem() + " "; // retrieve and collect
            curr = curr.getNext(); // retrieve next item
        }
        return list.toString(); // collection becomes a string
    }


    /**
     * Returns a string value of DNode items reversed
     */
    public String toStringR()
    {
        String reversed = "";
        DNode curr = head.getBack();
        for (int index = numItems - 1; index >= 0; index--)
        {
            reversed = reversed + curr.getItem() + " "; // retrieve and collect it
em
            curr = curr.getBack(); // retrieve last item
        }
        return reversed.toString(); // collection becomes a string
    }
}
```

```
::::::::::::::
Lab4P1Driver.java
::::::::::::::
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Lab4P1Driver
{
    static BufferedReader stdin = new BufferedReader(new InputStreamReader(System.
in));
    public static void main (String[] args) throws IOException
    {
        ListCDLSBased myList = new ListCDLSBased();
        boolean exit = false;
        int pos = -1;
        while (!exit) {
            System.out.println("Select from the following menu: \n"
                               + "\t 0. Exit the program \n"
                               + "\t 1. Insert item into the list \n"
                               + "\t 2. Remove item from the list \n"
                               + "\t 3. Get item from the list \n"
                               + "\t 4. Clear the list \n"
                               + "\t 5. Display size and content of the list in or
der and in reversed order \n"
                               + "\t 6. Delete the smallest and largest item in th
e list \n"
                               + "\t 7. Reverse the list \n");

            System.out.print("Make your menu selection now: ");
            int input = Integer.parseInt(stdin.readLine().trim());
            System.out.println(input);
            // possible cases for initial input
            switch (input) {
            case 0:
                System.out.println("Exiting program... good bye");
                exit = true;
                break;

            case 1:
                try
                {
                    System.out.println("You are now inserting an item into the lis
t.");
                    System.out.print("\t Enter item: ");
                    String item = stdin.readLine().trim();
                    System.out.println(item);

                    System.out.print("\t Enter the position to insert item in: ");
                    pos = Integer.parseInt(stdin.readLine().trim());
                    System.out.println(pos);
                    if (pos <= myList.size())
                    {
                        myList.add(pos, item);
                        System.out.println("Item " + item + " inserted in position
```

```
 " + pos + " in the list.");
                    }
                }

                catch(ListIndexOutOfBoundsException e)
                {
                    System.out.println("Position specified is out of range!");
                }
                break;

            case 2:
                try
                {
                    System.out.println("You are now removing an item from the list
.");
                    System.out.print("\t Enter position to remove item from: ");
                    pos = Integer.parseInt(stdin.readLine().trim());
                    System.out.println(pos);
                    System.out.println("Item " + myList.get(pos) + " removed from
position " + pos + " in the list.");
                    myList.remove(pos);
                }
                catch(ListIndexOutOfBoundsException e)
                {
                    System.out.println("Position specified is out of range!");
                }
                break;

            case 3:
                try
                {
                    System.out.print("\t Enter position to retrieve item from: ");
                    pos = Integer.parseInt(stdin.readLine().trim());
                    System.out.println(pos);
                    System.out.println("Item " + myList.get(pos) + " retrieved fro
m position " + pos + " in the list.");
                }

                catch(ListIndexOutOfBoundsException e)
                {
                    System.out.println("Position specified is out of range!");
                }
                break;

            case 4:
                if(myList.isEmpty())
                {
                    System.out.println("List is empty, nothing to clear!");
                }

                else
                {
                    System.out.println("Clearing list...");
                    myList.removeAll();
                    System.out.println("List cleared.");
                }
                break;

            case 5:
                if(!(myList.size() == 0))
                {
                    System.out.println("In order: List of size " + myList.size() +
 " has the following items: " + myList.toString());
                    System.out.println("In reverse order: List of size " + myList.
size() + " has the following items: " + myList.toStringR());
                }

                else
                {
                    System.out.println("List is empty.");
                }
                break;

            case 6:
                displayAndDeleteLargeAndSmall(myList);
                break;

            case 7:
                if(!(myList.isEmpty()))
                {
                    System.out.println("List has been reversed.");
                    myList = reverse(myList);
                    System.out.println("\t Here is the content: " + myList.toStrin
g());
                }

                else
                {
                    System.out.println("List is empty... nothing to reverse!");
                }
                break;
            }
        }
    }

    /**
     * Find largest and smallest items from DNode collection lexicographically
     * @param ListCDLSBased myList              list to be iterated
     * @param int[] numbers                 Array of largest and smallest values
     */
    static void findIndexLargeAndSmall(ListCDLSBased myList, int[] numbers)
    {
        int size = myList.size();
        int smallIndex = 0;
        int largeIndex = 0;
        String curr = "";
        String smallestValue = myList.get(0).toString();
        String largestValue = myList.get(0).toString();
        for (int index = 0; index < size; index++)
        {
            curr = myList.get(index).toString();
            if(curr.compareTo(smallestValue) <= 0)
            {
                smallIndex = index;
                smallestValue = myList.get(smallIndex).toString();
            }

            else if(curr.compareTo(largestValue) >= 0)
            {
                largeIndex = index;
                largestValue = myList.get(largeIndex).toString();
            }
        }
        numbers[0] = smallIndex;
```

```
        numbers[1] = largeIndex;
    }

    /**
    * Display and delete largest and smallest items from DNode collection lexicogr
aphically
    * @param ListCDLSBased myList        list to be iterated
    */
    static ListCDLSBased displayAndDeleteLargeAndSmall(ListCDLSBased myList) throw
s ListIndexOutOfBoundsException
    {
        ListCDLSBased temp = new ListCDLSBased();
        if(myList.size() == 0)
        {
            System.out.println("List is empty, nothing to delete!");
        }
        else if(myList.size() == 1)
        {
            System.out.println(myList.toString() + " is deleted.");
            myList.remove(0);
        }

        else
        {
            int numbers[] = new int[2];
            findIndexLargeAndSmall(myList, numbers);
            System.out.println("Smallest item " + myList.get(numbers[0]) + " delet
ed.");
            System.out.println("Largest item " + myList.get(numbers[1]) + " delete
d.");
            if(numbers[0] > numbers[1])
            {
                myList.remove(numbers[0]);
                myList.remove(numbers[1]);
            }

            else
            {
                myList.remove(numbers[0]);
                myList.remove(numbers[1] - 1);
            }
        }
        myList = temp;
        return temp;
    }

    /**
    * Reverse collection and return it reversed
    * @param ListCDLSBased myList        list to be iterated
    * @return temp                       reversed list
    */
    static ListCDLSBased reverse(ListCDLSBased myList)
    {
        ListCDLSBased temp = new ListCDLSBased();
        int size = myList.size();
        for (int index = 0; index < size; index++)
        {
            temp.add(index, myList.get(size - 1 - index));
        }
        myList = temp;
        return myList;
    }
}
```

```
::::::::::::::
Lab4P1Sampleruns.txt
::::::::::::::
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
List is empty.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 6
List is empty, nothing to delete!
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List is empty... nothing to reverse!
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Data
        Enter the position to insert item in: 0
Item Data inserted in position 0 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
```

```
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 1 has the following items: Data
In reverse order: List of size 1 has the following items: Data
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List has been reversed.
        Here is the content: Data
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Beverly
        Enter the position to insert item in: 0
Item Beverly inserted in position 0 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 2 has the following items: Beverly Data
In reverse order: List of size 2 has the following items: Data Beverly
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Jean-Luc
        Enter the position to insert item in: 5
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 2 has the following items: Beverly Data
In reverse order: List of size 2 has the following items: Data Beverly
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Jean-Luc
        Enter the position to insert item in: 2
Item Jean-Luc inserted in position 2 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Geordi
        Enter the position to insert item in: 2
Item Geordi inserted in position 2 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Worf
        Enter the position to insert item in: 3
Item Worf inserted in position 3 in the list.
Select from the following menu:
```

```
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 5 has the following items: Beverly Data Geordi Worf Jean-Lu
c
In reverse order: List of size 5 has the following items: Jean-Luc Worf Geordi Dat
a Beverly
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List has been reversed.
        Here is the content: Jean-Luc Worf Geordi Data Beverly
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List has been reversed.
        Here is the content: Beverly Data Geordi Worf Jean-Luc
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 6
Smallest item Beverly deleted.
Largest item Worf deleted.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 3 has the following items: Data Geordi Jean-Luc
In reverse order: List of size 3 has the following items: Jean-Luc Geordi Data
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List has been reversed.
        Here is the content: Jean-Luc Geordi Data
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List has been reversed.
        Here is the content: Data Geordi Jean-Luc
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 2
You are now removing an item from the list.
        Enter position to remove item from: 9
Position specified is out of range!
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 2
You are now removing an item from the list.
        Enter position to remove item from: 2
Item Jean-Luc removed from position 2 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
```

```
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 2 has the following items: Data Geordi
In reverse order: List of size 2 has the following items: Geordi Data
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 2
You are now removing an item from the list.
        Enter position to remove item from: 0
Item Data removed from position 0 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Will
        Enter the position to insert item in: 0
Item Will inserted in position 0 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 2 has the following items: Will Geordi
In reverse order: List of size 2 has the following items: Geordi Will
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 3
        Enter position to retrieve item from: 1
Item Geordi retrieved from position 1 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 3
        Enter position to retrieve item from: 0
Item Will retrieved from position 0 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 3
        Enter position to retrieve item from: 8
Position specified is out of range!
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 2 has the following items: Will Geordi
In reverse order: List of size 2 has the following items: Geordi Will
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 6
Smallest item Geordi deleted.
Largest item Will deleted.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
```

```
        7. Reverse the list

Make your menu selection now: 5
List is empty.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List is empty... nothing to reverse!
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Velcro
        Enter the position to insert item in: 0
Item Velcro inserted in position 0 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 1 has the following items: Velcro
In reverse order: List of size 1 has the following items: Velcro
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List has been reversed.
        Here is the content: Velcro
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 6
Velcro  is deleted.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
List is empty.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 0
Exiting program... good bye
::::::::::::::
Lab4P2.txt
::::::::::::::
a) Space Complexity analysis:
The total space complexity of the CDLS implementation is 12 bytes.

The CDLS implementation uses class DNode that itself contains 3 instance variables
.
Those being because:
head = ref1 (next) + ref2 (back) (DNode)
numItems = int: ref + int + n * (ref1 + ref2 + ref3) = (2n + 3) * 4 bytes.


b) Time Complexity analysis:
Operations used: get(int index), add(int index, Object item), remove(int index), f
ind(int index) and shifting (items[i] = items[i + or -] different size)

Traversal: curr = curr.getNext() or curr = curr.getBack();

get(int index):
Depends
Best case: index 0 costs 0
Worst case: index n/2 costs n/2
Average: index n/4 costs n/4


add(int index, Object item):
Depends
Best: index 0 costs 1
Worst: index n-1/2 costs n/4
Average: index n-1/4 costs n-1/8
```

```
find(int index):
Depends
Best: index 0 costs 0
Worst: index n/2 costs n/2
Average: index n/4 costs n/4

Time used:
Best case - 0
Worst case - n/2
Average case - n/4
```