```
::::::::::::::
Lab8Status.txt
::::::::::::::
CHANGES MADE:
Fixed Search method and cases in P1Driver.
Fixed Search method & cases in P2Driver.
Fixed Search method in AscendinglyOrderedStringList.
Javadoc html file created and working as intended.
Problem 1: compiles, runs correctly on all provided input.
Problem 2: compiles, runs correctly on all provided input.
Problem 3: compiles, runs correctly on all provided input.
ECI: compiles, runs correctly on all provided input.
ECII: compiles, runs correctly on all provided input.::::::::::::::
Lab8Conclusions.txt
::::::::::::::
Lab 8 showed me how useful Sequential Searches and Binary Searches can be and how
they efficiently iterate through collections in a better way than data structures
that come in the Java source code.::::::::::::::
Lab8P1Driver.java
::::::::::::::
/*
 * Purpose: Data Structure and Algorithms Lab 8
 * Status: Complete and thoroughly tested
 * Last update: 03/27/23
 * Submitted:  03/27/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.03.27
 */
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Lab8P1Driver
{
    static BufferedReader stdin = new BufferedReader(new InputStreamReader(System.
in));
    public static void main (String[] args) throws IOException
    {
        ListArrayBasedPlus list_plus = new ListArrayBasedPlus();
        boolean exit = false;

        while (!exit)
        {
            System.out.println("Select from the following menu: \n"
                               + "0. Exit the program \n"
                               + "1. Insert item into ordered list \n"
                               + "2. Remove item from the list \n"
                               + "3. Get item from the list \n"
                               + "4. Search for a specific item in the list \n"
                               + "5. Clear the list \n"
                               + "6. Print size and content of the list \n");

            System.out.print("Make your menu selection now: " );
            int input = Integer.parseInt(stdin.readLine());
            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:
                // NEW CHANGES MADE
                System.out.println("You are now inserting an item into the list. "
);
                add(list_plus);
                break;

            case 2:
                System.out.println("You are now removing an item from the list. ")
;
                remove(list_plus);
                break;

            case 3:
                System.out.print("You are now retrieving an item. ");
                retrieve(list_plus);
                break;

            case 4:
                System.out.print("You are now searching for an item. \n Enter the
item to search for: ");
                search2(list_plus);
                break;

            case 5:
                System.out.println("Clearing list...");
                list_plus.removeAll();
                System.out.println("List cleared.");
                break;

            case 6:
                if(list_plus.isEmpty())
                {
                    System.out.println("List is empty.");
                }

                else
                {
                    System.out.println("List of size " + list_plus.size() + " has
the following items: " + list_plus.toString());
                }
                break;
            }
        }
    }
    public static void add(ListArrayBasedPlus list_plus) throws IOException
    {
        System.out.print("Enter item: ");
        String item = stdin.readLine().trim();
        System.out.println(item);
        System.out.print("Enter the position to enter the item in: ");
        int position = Integer.parseInt(stdin.readLine());
        System.out.println(position);
        int size = list_plus.size();
        if(position <= size)
        {
            list_plus.add(position, item);
            System.out.println("Item " + item + " inserted in position " + positio
n);
        }
```

```
        else
        {
            System.out.println("Position specified is out of range!");
        }
    }

    public static void remove(ListArrayBasedPlus list_plus) throws IOException
    {
        System.out.print("Enter position to remove item from: ");
        int position = Integer.parseInt(stdin.readLine().trim());
        System.out.println(position);
        int size = list_plus.size();
        if(position >= size)
        {
            System.out.println("Position specified is out of range!");
        }

        else
        {
            System.out.println("Item " + list_plus.get(position) + " removed from
position " + position + " in the list.");
            list_plus.remove(position);
        }
    }

    public static void retrieve(ListArrayBasedPlus list_plus) throws IOException
    {
        System.out.print("Enter position to retrieve item from: ");
        int position = Integer.parseInt(stdin.readLine().trim());
        System.out.println(position);
        int size = list_plus.size();
        if(position >= size)
        {
            System.out.println("Position specified is out of range!");
        }

        else
        {
            System.out.println("Item " + list_plus.get(position) + " retrieved fro
m position " + position + " in the list.");
        }
    }

    public static void search2(ListArrayBasedPlus list_plus) throws IOException
    {
        String key2 = stdin.readLine().trim();
        System.out.println(key2);
        System.out.println("Searching for item...");
        int position = search(key2, list_plus);
        if (position <= -1)
        {
            System.out.println(" -1 no item with that name exists, try again.");
        }

        else
        {
            System.out.println(key2 + " found at position " + position);
        }
    }
    /**
     *  CHANGES MADE TO SEARCH METHOD
     * Searches for an item in the list using compareTo
     * @param key     the item to search for
     * @param list    the list to search in
     * @return        the index of the item if found
     *
     * if (curr == key)
     *    stop(succ, pos)
     * else
     *    advance
     * stop(unsucc)
     */
    public static int search(String key, ListArrayBasedPlus list)
    {
        int position = -1;
        for(int index = 0; index < list.size(); index++)
        {
            if(key.equals(list.get(index)))
            {
                position = index;
            }
        }
        return position;
    }
}
```

```
::::::::::::::
Lab8P1Sampleruns.txt
::::::::::::::
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -7
Enter the position to enter the item in: 1
Position specified is out of range!
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -7
Enter the position to enter the item in: 0
Item -7 inserted in position 0
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
```

```
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 8
Enter the position to enter the item in: 1
Item 8 inserted in position 1
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 7
Enter the position to enter the item in: 2
Item 7 inserted in position 2
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 4
Enter the position to enter the item in: 3
Item 4 inserted in position 3
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 0
Enter the position to enter the item in: 4
Item 0 inserted in position 4
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 1
Enter the position to enter the item in: 4
Item 1 inserted in position 4
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -2
Enter the position to enter the item in: 0
Item -2 inserted in position 0
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -5
Enter the position to enter the item in: 1
Item -5 inserted in position 1
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 6
List of size 8 has the following items: -2 -5 -7 8 7 4 1 0
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 4
You are now searching for an item.
 Enter the item to search for: 7
Searching for item...
7 found at position 4
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list
```

```
Make your menu selection now: 4
You are now searching for an item.
 Enter the item to search for: 8
Searching for item...
8 found at position 3
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 4
You are now searching for an item.
 Enter the item to search for: -20
Searching for item...
 -1 no item with that name exists, try again.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of 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 -2 removed from position 0 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of 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 -5 removed from position 0 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 3
You are now retrieving an item. Enter position to retrieve item from: 1
Item 8 retrieved from position 1 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 3
You are now retrieving an item. Enter position to retrieve item from: 2
Item 7 retrieved from position 2 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 6
List of size 6 has the following items: -7 8 7 4 1 0
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 5
Clearing list...
List cleared.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 6
List is empty.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 0
Exiting program... good bye
::::::::::::::
Lab8P2Driver.java
::::::::::::::
/*
 * Purpose: Data Structure and Algorithms Lab 8
 * Status: Complete and thoroughly tested
 * Last update: 03/27/23
 * Submitted:  03/27/23
 * Comment: test suite and sample run attached
```

```
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.03.27
 */
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Lab8P2Driver
{
    static BufferedReader stdin = new BufferedReader(new InputStreamReader(System.
in));
    public static void main (String[] args) throws IOException
    {
        ListArrayBasedPlus list_plus = new ListArrayBasedPlus();
        boolean exit = false;

        while (!exit)
        {
            System.out.println("Select from the following menu: \n"
                               + "0. Exit the program \n"
                               + "1. Insert item into ordered list \n"
                               + "2. Remove item from the list \n"
                               + "3. Get item from the list \n"
                               + "4. Search for a specific item in the list \n"
                               + "5. Clear the list \n"
                               + "6. Print size and content of the list \n");

            System.out.print("Make your menu selection now: " );
            int input = Integer.parseInt(stdin.readLine());
            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:
                // NEW CHANGES MADE
                System.out.println("You are now inserting an item into the list. "
);
                add(list_plus);
                break;

            case 2:
                System.out.println("You are now removing an item from the list. ")
;
                remove(list_plus);
                break;

            case 3:
                System.out.print("You are now retrieving an item. ");
                retrieve(list_plus);
                break;

            case 4:
                System.out.print("You are now searching for an item. \n Enter the
item to search for: ");
                search2(list_plus);
                break;

            case 5:
                System.out.println("Clearing list...");
                list_plus.removeAll();
                System.out.println("List cleared.");
                break;

            case 6:
                if(list_plus.isEmpty())
                {
                    System.out.println("List is empty.");
                }

                else
                {
                    System.out.println("List of size " + list_plus.size() + " has
the following items: " + list_plus.toString());
                }
                break;
            }
        }
    }
    public static void add(ListArrayBasedPlus list_plus) throws IOException
    {
        System.out.print("Enter item: ");
        String key = stdin.readLine().trim();
        System.out.println(key);
        // Modified Sequential Search III
        int position = search(key, list_plus);

        if(position < 0)
        {
            position = (position + 1)* -1;
        }
        list_plus.add(position, key);
        System.out.println("Item " + key + " inserted in position " + position);
    }

    public static void remove(ListArrayBasedPlus list_plus) throws IOException
    {
        System.out.print("Enter position to remove item from: ");
        int position = Integer.parseInt(stdin.readLine().trim());
        System.out.println(position);
        int size = list_plus.size();
        if(position >= size)
        {
            System.out.println("Position specified is out of range!");
        }

        else
        {
            System.out.println("Item " + list_plus.get(position) + " removed from
position " + position + " in the list.");
            list_plus.remove(position);
        }
    }

    public static void retrieve(ListArrayBasedPlus list_plus) throws IOException
    {
        System.out.print("Enter position to retrieve item from: ");
        int position = Integer.parseInt(stdin.readLine().trim());
        System.out.println(position);
        int size = list_plus.size();
```

```
        if(position < 0 || position >= size)
        {
            System.out.println("Position specified is out of range!");
        }

        else
        {
            System.out.println("Item " + list_plus.get(position) + " retrieved fro
m position " + position + " in the list.");
        }
    }

    public static void search2(ListArrayBasedPlus list_plus) throws IOException
    {
        String key2 = stdin.readLine().trim();
        System.out.println(key2);
        System.out.println("Searching for item...");
        int position = search(key2, list_plus);
        if (position <= -1)
        {
            System.out.println("-1 no item with that name exists, try again.");
        }

        else
        {
            System.out.println(key2 + " found at position " + position);
        }
    }
    /**
     * CHANGES MADE TO SEARCH METHOD
     * Searches for an item in the list using compareTo
     * @param key     the item to search for
     * @param list    the list to search in
     * @return        the index of the item if found
     *
     * MODIFIED SEQUENTIAL SEARCH III EAGERLY ADVANCING
     * if (curr == key)
     *    stop(succ, pos)
     * else
     *    advance
     * stop(unsucc)
     */
    public static int search(String key, ListArrayBasedPlus list_plus)
    {
        int low = 0;
        int size = list_plus.size();
        int high = size - 1;

        while (low <= high)
        {
            int mid = (low + high) / 2;
            int compare = key.compareTo((String)list_plus.get(mid));
            if (compare == 0)
            {
                return mid;
            }
            else if (compare < 0)
            {
                high = mid - 1;
            }
            else
            {
                low = mid + 1;
            }
        }
        return -(low + 1);
    }
}
```

```
::::::::::::::
Lab8P2Sampleruns.txt
::::::::::::::
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 8
Item 8 inserted in position 0
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 7
Item 7 inserted in position 0
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -7
Item -7 inserted in position 0
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -2
Item -2 inserted in position 0
Select from the following menu:
```

```
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -5
Item -5 inserted in position 1
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 1
Item 1 inserted in position 3
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 4
Item 4 inserted in position 4
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 0
Item 0 inserted in position 3
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 6
List of size 8 has the following items: -2 -5 -7 0 1 4 7 8
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 3
You are now retrieving an item. Enter position to retrieve item from: 0
Item -2 retrieved from position 0 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 3
You are now retrieving an item. Enter position to retrieve item from: 1
Item -5 retrieved from position 1 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 3
You are now retrieving an item. Enter position to retrieve item from: 6
Item 7 retrieved from position 6 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 4
You are now searching for an item.
 Enter the item to search for: -5
Searching for item...
-5 found at position 1
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 4
You are now searching for an item.
 Enter the item to search for: 100
```

```
Searching for item...
-1 no item with that name exists, try again.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of 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 -2 removed from position 0 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 6
List of size 7 has the following items: -5 -7 0 1 4 7 8
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 5
Clearing list...
List cleared.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 6
List is empty.
Select from the following menu:
0. Exit the program
1. Insert item into ordered list
2. Remove item from the list
3. Get item from the list
4. Search for a specific item in the list
5. Clear the list
6. Print size and content of the list

Make your menu selection now: 0
Exiting program... good bye
::::::::::::::
AscendinglyOrderedStringList.java
::::::::::::::
```

```java
/*
 * Purpose: Data Structure and Algorithms Lab 8
 * Status: Complete and thoroughly tested
 * Last update: 03/27/23
 * Submitted:  03/27/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.03.27
 */
public class AscendinglyOrderedStringList extends ListArrayBasedPlus implements As
cendinglyOrderedStringListInterface
{
    private static final int MAX_LIST = 10; // Maximum capacity of the list
    private String[] items; // Array to store items
    private int numItems; // Number of items in the list

    /**
     * Default constructor for AscendinglyOrderedStringList.
     * Creates a new AscendinglyOrderedStringList object with an empty array of MA
X_LIST size.
     */
    public AscendinglyOrderedStringList()
    {
        super();
        items = new String[MAX_LIST];
        numItems = 0;
    } // end default constructor

    /**
     * Adds an item to the list in ascending order.
     *
     * @param item the item to be added to the list
     * @throws ListIndexOutOfBoundsException if the index is out of range
     */
    public void add(String item) throws ListIndexOutOfBoundsException
    {
        // Binary Search II
        int pos = search(item);
        // Check if item already exists in list
        if (pos >= 0 && pos < numItems && items[pos].compareTo(item) == 0)
        {
            // Item already exists, don't insert duplicate
            System.out.println(item + " already exists in the list. Try again.");
            return;
        }

        // Resize array if it is full
        if (numItems == items.length)
        {
            resize();
        }
        // Calculate index where item should be inserted
        pos = -(pos + 1);

        // Shift items to make room for new item
        for (int index = numItems - 1; index >= pos; index--)
        {
            items[index + 1] = items[index];
        }
        items[pos] = item;
```

```
        numItems++;
        // Call add method of ListArrayBasedPlus to add item to list
        super.add(pos, item);
    } // end add

    /**
     * Returns the item at the specified index in the list.
     *
     * @param index the index of the item to be returned
     * @return the item at the specified index in the list
     * @throws ListIndexOutOfBoundsException if the index is out of range
     */
    public String get(int index) throws ListIndexOutOfBoundsException
    {
        if (index < 0 || index >= numItems)
        {
            throw new ListIndexOutOfBoundsException("Index out of range!");
        }
        return items[index];
    } // end get

    /**
     * Removes the item at the specified index from the list.
     *
     * @param index the index of the item to be removed from the list
     * @throws ListIndexOutOfBoundsException if the index is out of range
     */
    public void remove(int index) throws ListIndexOutOfBoundsException
    {
        if (index < 0 || index >= numItems)
        {
            throw new ListIndexOutOfBoundsException("Index out of range!");
        }
        // Shift items to remove item at specified index
        for (int j = index; j < numItems - 1; j++)
        {
            items[j] = items[j + 1];
        }
        items[numItems -1] = null;
        numItems--;
    } // end remove

    public void display()
    {
        if(numItems != 0)
        {
            for (int i = 0; i < numItems; i++)
            {
                System.out.print(items[i] + " "  + "\n");
            }
            System.out.println();
        }
    }
    /**
     * CHANGES MADE TO SEARCH METHOD
     * Binary Search II
     * Searches for an item in the list using compareTo
     * @param key     the item to search for
     * @return        the index of the item if found OR -1 for duplicates/nonexist
ing items
     * while(low <= high)
     * {
     *    midIndex = (low + high) / 2
     *    if (key > midKey)
     *       low = midIndex + 1
     *    else
     *       high = midIndex
     * }
     * if(key == currKey)
     *    stop(succ, pos)
     * else
     *    stop(unsucc, pos)
     */
    public int search(String key)
    {
        int low = 0;
        int high = numItems - 1;
        while (low <= high)
        {
            int midIndex = (low + high) / 2;
            int compare = key.compareTo(items[midIndex]);
            if (compare == 0)
            {
                // key found
                return midIndex;
            }

            else if (compare < 0)
            {
                // key smaller, search left half
                high = midIndex - 1;
            }

            else
            {
                // key larger, search right half
                low = midIndex + 1;
            }
        }
        // return index where key should be inserted
        return -(low + 1);
    }
}
```

```
:::::::::::::::
Lab8P3Driver.java
:::::::::::::::
/*
 * Purpose: Data Structure and Algorithms Lab 8
 * Status: Complete and thoroughly tested
 * Last update: 03/27/23
 * Submitted:  03/27/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.03.27
 */
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Lab8P3Driver
{
    static BufferedReader stdin = new BufferedReader(new InputStreamReader(System.
in));
    public static void main (String[] args) throws IOException
```

```
    {
        AscendinglyOrderedStringList list = new AscendinglyOrderedStringList();
        boolean exit = false;
        while (!exit)
        {
            System.out.println("Select from the following menu: \n"
                               + "0. Exit the program \n"
                               + "1. Insert specified item into the list \n"
                               + "2. Remove item in specified position in the list
 \n"
                               + "3. Search list for a specific item \n"
                               + "4. Clear the list \n"
                               + "5. Display the content of 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.");
                    add(list);
                }

                catch(ListIndexOutOfBoundsException e)
                {
                    System.out.println("ListIndexOutOfBoundsException on 'add'. Li
st is full!");
                }
                break;

            case 2:
                try
                {
                    System.out.println("You are now removing an item from the list
.");
                    remove(list);
                }

                catch(ListIndexOutOfBoundsException e)
                {
                    System.out.println("ListIndexOutOfBoundsException on 'remove'.
 Position out of bounds!");
                }
                break;

            case 3:
                System.out.println("You are now searching for an item.");
                search(list);
                break;

            case 4:
                list.removeAll();
                System.out.println("List cleared");
                break;

            case 5:
                if (list.isEmpty())
                {
                    System.out.println("List is empty.");
                }
                else
                {
                    System.out.print("List of size " + list.size() + " has the fol
lowing items : \n");
                    list.display();
                }
                break;

            default:
                System.out.println("Invalid choice.");
                break;
            }
        }
    }
    public static void add(AscendinglyOrderedStringList list) throws IOException
    {
        System.out.print("Enter item: ");
        String item = stdin.readLine().trim();
        System.out.println(item);
        try
        {
            if (list.search(item) >= 0)
            {
                System.out.println("Item " + item + " is already in the list. Plea
se enter a different item.");
            }
            else
            {
                list.add(item);
                System.out.println("Item " + item + " has been added to the list."
);
            }
        }
        catch (ListIndexOutOfBoundsException e)
        {
            System.out.println("List is full.");
        }
    }

    public static void remove(AscendinglyOrderedStringList list) throws IOExceptio
n
    {
        System.out.print("Enter position to remove item from: ");
        int pos = Integer.parseInt(stdin.readLine().trim());
        System.out.println(pos);
        try
        {
            String removed = list.get(pos);
            System.out.println("Item " + removed + " is removed from the list.");
        }

        catch (ListIndexOutOfBoundsException e)
        {
            System.out.println("Position specified is out of bounds.");
```

```
        }
    }

    public static void search(AscendinglyOrderedStringList list) throws IOExceptio
n
    {
        System.out.print("Enter the item to search for: ");
        String key = stdin.readLine().trim();
        System.out.println(key);
        int pos = list.search(key);
        if(pos >= 0)
        {
            System.out.println("Item found in position " + pos);
        }

        else
        {
            System.out.println("Item does not exist, should be inserted in index "
 + -(pos + 1) );
        }
    }
}
::::::::::::::
Lab8P3Sampleruns.txt
::::::::::::::
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -7
Item -7 has been added to the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 8
Item 8 has been added to the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -5
Item -5 has been added to the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 7
Item 7 has been added to the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 7
Item 7 is already in the list. Please enter a different item.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -2
Item -2 has been added to the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 4
Item 4 has been added to the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 1
Item 1 has been added to the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
```

```
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 0
Item 0 has been added to the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 5
List of size 8 has the following items :
-2
-5
-7
0
1
4
7
8

Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 3
You are now searching for an item.
Enter the item to search for: 0
Item found in position 3
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 3
Item 0 is removed from the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 4
List cleared
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of 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 -2 is removed from the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 1
Item -5 is removed from the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of 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 -7 is removed from the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 5
List is empty.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 4
List cleared
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
```

```
5. Display the content of the list

Make your menu selection now: 0
Exiting program... good bye
::::::::::::::
AscendinglyOrderedList.java
::::::::::::::
/*
 * Purpose: Data Structure and Algorithms Lab 8 EXTRA CREDIT I
 * Status: Complete and thoroughly tested
 * Last update: 03/27/23
 * Submitted:  03/27/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.03.27
 */
public class AscendinglyOrderedList<T extends KeyedItem<KT>, KT extends Comparable
<? super KT>> implements AscendinglyOrderedListInterface<T, KT> {

    private T[] items;
    private int numItems;
    private static final int MAX_LIST = 10;
    private boolean assocboolean;
    private int assocint;

    @SuppressWarnings("unchecked")
    public AscendinglyOrderedList()
    {
        items = (T[]) new KeyedItem[MAX_LIST];
        numItems = 0;
        assocboolean = false;
        assocint = 0;
    }

    @SuppressWarnings("unchecked")
    private void resize()
    {
        T[] temp = (T[]) new KeyedItem[items.length * 2];
        for (int i = 0; i < numItems; i++)
        {
            temp[i] = items[i];
        }
        items = temp;
    }

    public boolean isEmpty()
    {
        return numItems == 0;
    }

    public int size()
    {
        return numItems;
    }

    public void setAssocboolean(boolean assocboolean)
    {
        this.assocboolean = assocboolean;
    }

    public boolean getAssocboolean()
    {
        return assocboolean;
    }

    public void setAssocint(int assocint)
    {
        this.assocint = assocint;
    }

    public int getAssocint(int assocint)
    {
        return assocint;
    }

    public void add(T item) throws ListIndexOutOfBoundsException
    {
        KT key = item.getKey();
        int pos = search(key);
        if (pos >= 0 && pos < numItems && items[pos].getKey().compareTo(key) == 0)
        {
            // Item already exists, don't insert duplicate
            System.out.println(item + " already exists in the list. Try again.");
            return;
        }
        else
        {
            if (numItems == items.length)
            {
                resize();
            }
            // Shift items to make room for new item
            for (int index = numItems - 1; index >= pos; index--)
            {
                items[index + 1] = items[index];
            }
            items[pos] = item;
            numItems++;
        }
    }

    public T get(int index) throws ListIndexOutOfBoundsException
    {
        if (index < 0 || index >= numItems)
        {
            throw new ListIndexOutOfBoundsException("Index out of range!");
        }
        return items[index];
    }

    public void remove(int index) throws ListIndexOutOfBoundsException
    {
        if (index < 0 || index >= numItems)
        {
            throw new ListIndexOutOfBoundsException("Index out of range!");
        }
        // Shift items to remove item at specified index
        for (int j = index; j < numItems - 1; j++)
        {
            items[j] = items[j + 1];
        }
        numItems--;
        items[numItems] = null;
```

```
    }

    public int search(KT key)
    {
        int low = 0;
        int high = numItems - 1;
        int position = -1;
        boolean success = false;
        while (low <= high)
        {
            int midIndex = (low + high) / 2;
            KT midKey = items[midIndex].getKey();
            if (key.compareTo(midKey) > 0)
            {
                // key > midKey, search upper half of list
                low = midIndex + 1;
            }
            else if (key.compareTo(midKey) < 0)
            {
                // key < midKey, search lower half of list
                high = midIndex - 1;
            }
            else
            {
                // key == midKey, item found
                position = midIndex;
                success = true;
                break;
            }
        }
        if (success)
        {
            return position;
        }
        else
        {
            return -1;
        }
    }

    @SuppressWarnings("unchecked")
    public void clear()
    {
        items = (T[]) new KeyedItem[MAX_LIST];
        numItems = 0;
    }
}
:::::::::::::::
Lab8ECIDriver.java
:::::::::::::::
/*
 * Purpose: Data Structure and Algorithms Lab 8 EXTRA CREDIT I
 * Status: Complete and thoroughly tested
 * Last update: 03/27/23
 * Submitted:  03/27/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.03.27
 */
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Lab8ECIDriver
{
    static BufferedReader stdin = new BufferedReader(new InputStreamReader(System.
in));
    public static void main (String[] args) throws IOException
    {
        AscendinglyOrderedStringList list = new AscendinglyOrderedStringList();
        boolean exit = false;
        int pos = -1;

        while (!exit)
        {
            System.out.println("Select from the following menu: \n"
                               + "0. Exit the program \n"
                               + "1. Insert specified item into the list \n"
                               + "2. Remove item in specified position in the list
 \n"
                               + "3. Search list for a specific item \n"
                               + "4. Clear the list \n"
                               + "5. Display the content of the list \n");

            System.out.print("Make your menu selection now: " );
            int input = Integer.parseInt(stdin.readLine());
            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("Enter item: ");
                    String item = stdin.readLine();
                    int found = list.search(item);
                    System.out.println(item);
                    if(found != -1)
                    {
                        list.add(item);
                        System.out.println(item + " inserted into the list.");
                    }
                }

                catch(ListIndexOutOfBoundsException e)
                {
                    System.out.println("ListIndexOutOfBoundsException on 'add'. Li
st is full!");
                }
                break;

            case 2:
                try
                {
                    System.out.println("You are now removing an item from the list
.");
                    System.out.print("Enter position to remove item from: ");
```

```
                    pos = Integer.parseInt(stdin.readLine());
                    System.out.println(pos);
                    String removed = list.get(pos);
                    list.remove(pos);
                    System.out.println(removed + " removed from the list.");
                }

                catch(ListIndexOutOfBoundsException e)
                {
                    System.out.println("ListIndexOutOfBoundsException on 'remove'.
 Position out of bounds!");
                }
                break;

            case 3:
                System.out.print("You are now searching for an item. \n Enter the
item to search for: ");
                String key2 = stdin.readLine();
                System.out.println(key2);
                int pos3 = list.search(key2);
                System.out.println("Item found in position " + pos3);
                break;

            case 4:
                list.removeAll();
                System.out.println("List cleared");
                break;

            case 5:
                if (list.isEmpty())
                {
                    System.out.println("List is empty.");
                }
                else
                {
                    System.out.print("List of size " + list.size() + " has the fol
lowing items : \n");
                    list.display();
                }
                break;

            default:
                System.out.println("Invalid choice.");
                break;
            }
        }
    }
}
```

::::::::::::::
Lab8ECISampleruns.txt
::::::::::::::

```
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -7
-7 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 8
8 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -5
-5 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 7
7 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -2
-2 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 4
4 inserted into the list.
Select from the following menu:
```

```
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 1
1 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 0
0 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 5
List of size 8 has the following items :
-2
-5
-7
0
1
4
7
8

Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 3
You are now searching for an item.
 Enter the item to search for: 0
Item found in position 3
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 3
0 removed from the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 4
List cleared
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 0
-2 removed from the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 1
-7 removed from the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 2
4 removed from the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 5
List is empty.
Select from the following menu:
0. Exit the program
```

```
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 4
List cleared
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 0
Exiting program... good bye
:::::::::::::::
AscendinglyOrderedStringListD.java
:::::::::::::::
```

```java
/*
 * Purpose: Data Structure and Algorithms Lab 8 EXTRA CREDIT II
 * Status: Complete and thoroughly tested
 * Last update: 03/27/23
 * Submitted:  03/27/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.03.27
 */
public class AscendinglyOrderedStringListD
{

    private String[] items;
    private int numItems;
    private static final int MAX_LIST = 10;

    public AscendinglyOrderedStringListD()
    {
        items = new String[MAX_LIST];
        numItems = 0;
    }

    public void resize()
    {
        String[] temp = new String[items.length * 2];
        for (int i = 0; i < numItems; i++)
        {
            temp[i] = items[i];
        }
        items = temp;
    }

    public boolean isEmpty()
    {
        return numItems == 0;
    }

    public int size()
    {
        return numItems;
    }

}

public void add(String item) throws ListIndexOutOfBoundsException
{
    int pos = search(item);
    if (numItems == items.length)
    {
        resize();
    }
    // Shift items to make room for new item
    for (int index = numItems - 1; index >= pos; index--)
    {
        items[index + 1] = items[index];
    }
    items[pos] = item;
    numItems++;
}

public String get(int index) throws ListIndexOutOfBoundsException
{
    if (index < 0 || index >= numItems) {
        throw new ListIndexOutOfBoundsException("Index out of range!");
    }
    return items[index];
}

public void remove(int index) throws ListIndexOutOfBoundsException
{
    if (index < 0 || index >= numItems)
    {
        throw new ListIndexOutOfBoundsException("Index out of range!");
    }
    // Shift items to remove item at specified index
    for (int j = index; j < numItems - 1; j++)
    {
        items[j] = items[j + 1];
    }
    numItems--;
    items[numItems] = null;
}

public int search(String item)
{
    int low = 0;
    int high = numItems - 1;
    int position = -1;
    while (low <= high)
    {
        int midIndex = (low + high) / 2;
        String midItem = items[midIndex];
        if (item.compareTo(midItem) > 0)
        {
            // item > midItem, search upper half of list
            low = midIndex + 1;
        }
        else
        {
            // item <= midItem, search lower half of list
            high = midIndex - 1;
            if (item.equals(midItem))
            {
                position = midIndex;
```

```
                }
            }
        }
        if (position == -1)
        {
            // Item not found, return position where it should be inserted
            return low;
        }
        else
        {
            // Item found, return position of first occurrence
            while (position > 0 && item.equals(items[position - 1]))
            {
                position--;
            }
            return position;
        }
    }

    public void clear() {
        items = new String[MAX_LIST];
        numItems = 0;
    }
}
:::::::::::::::
Lab8ECIIDriver.java
:::::::::::::::
/*
 * Purpose: Data Structure and Algorithms Lab 8 EXTRA CREDIT I
 * Status: Complete and thoroughly tested
 * Last update: 03/27/23
 * Submitted:  03/27/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.03.27
 */
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Lab8ECIIDriver
{
    static BufferedReader stdin = new BufferedReader(new InputStreamReader(System.
in));
    public static void main (String[] args) throws IOException
    {
        AscendinglyOrderedStringList list = new AscendinglyOrderedStringList();
        boolean exit = false;
        int pos = -1;

        while (!exit)
        {
            System.out.println("Select from the following menu: \n"
                               + "0. Exit the program \n"
                               + "1. Insert specified item into the list \n"
                               + "2. Remove item in specified position in the list
 \n"
                               + "3. Search list for a specific item \n"
                               + "4. Clear the list \n"
                               + "5. Display the content of the list \n");

            System.out.print("Make your menu selection now: " );
            int input = Integer.parseInt(stdin.readLine());
            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("Enter item: ");
                    String item = stdin.readLine();
                    int found = list.search(item);
                    System.out.println(item);
                    list.add(item);
                    System.out.println(item + " inserted into the list.");
                }

                catch(ListIndexOutOfBoundsException e)
                {
                    System.out.println("ListIndexOutOfBoundsException on 'add'. Li
st is full!");
                }
                break;

            case 2:
                try
                {
                    System.out.println("You are now removing an item from the list
.");
                    System.out.print("Enter position to remove item from: ");
                    pos = Integer.parseInt(stdin.readLine());
                    System.out.println(pos);
                    String removed = list.get(pos);
                    list.remove(pos);
                    System.out.println(removed + " removed from the list.");
                }

                catch(ListIndexOutOfBoundsException e)
                {
                    System.out.println("ListIndexOutOfBoundsException on 'remove'.
 Position out of bounds!");
                }
                break;

            case 3:
                System.out.print("You are now searching for an item. \n Enter the
item to search for: ");
                String key2 = stdin.readLine();
                System.out.println(key2);
                int pos3 = list.search(key2);
                System.out.println("Item found in position " + pos3);
                break;

            case 4:
                list.removeAll();
                System.out.println("List cleared");
```

```
                break;

            case 5:
                if (list.isEmpty())
                {
                    System.out.println("List is empty.");
                }
                else
                {
                    System.out.print("List of size " + list.size() + " has the fol
lowing items : \n");
                    list.display();
                }
                break;

            default:
                System.out.println("Invalid choice.");
                break;
            }
        }
    }
}
```

::::::::::::::
Lab8ECIISampleruns.txt
::::::::::::::

```
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -7
-7 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 8
8 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -5
-5 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -5
-5 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 7
7 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -2
-2 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: -2
-2 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 4
4 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
```

```
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 1
1 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: 0
0 inserted into the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 5
List of size 8 has the following items :
-2
-2
-5
-5
-7
0
1
4
7
8

Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 3
You are now searching for an item.
 Enter the item to search for: 0
Item found in position 3
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 3
0 removed from the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 4
List cleared
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 0
-2 removed from the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 1
-7 removed from the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 2
4 removed from the list.
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 5
List is empty.
Select from the following menu:
0. Exit the program
```

```
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 4
List cleared
Select from the following menu:
0. Exit the program
1. Insert specified item into the list
2. Remove item in specified position in the list
3. Search list for a specific item
4. Clear the list
5. Display the content of the list

Make your menu selection now: 0
Exiting program... good bye
```