```java
package ch.publicept;

import java.util.ArrayList;

/**
 * GroceryList
 *
 * public:
 * addGroceryItem(String item)
 * printGroceryList()
 * modifyGroceryItem(String newItem)
 * removeGroceryItem(String itemToRemove)
 * isItemInList()
 *
 * getters:
 * getGroceryList()
 *
 * private:
 * modifyGroceryItem(int i, String newItem)
 * removeGroceryItem(int i)
 * findItem(String searchItem)
 *
 * @author created by Urs Albisser, on 2020-01-28
 * @version 0.1
 */
public class GroceryList {



    // == fields ==
    // ArrayList<String> groceryList = new ArrayList<String>();
    // explicit type argument ArrayList<String>() can be replaced with ArrayList<>()
    // ArrayList<>() -> () brackets = call empty constructor of the ArrayLIst class.
    private ArrayList<String> groceryList = new ArrayList<>();



    // == public methods ==
    /**
     * addGroceryItem()
     * Add a new item to the ArrayList.
     * @param item New item String to add
     */
    public void addGroceryItem(String item) {

        /*
         * ArrayList.add(item)
         */
        groceryList.add(item);
    }


    /**
     * printGroceryList()
     * Print the whole ArrayList to the console.
     */
    public void printGroceryList() {
        /*
         * ArrayList.size()
         */
        System.out.println("You have " + groceryList.size() + " items in your grocery list.");

        for(int i=0; i<groceryList.size(); i++)
            /*
             * ArrayList.get(index)
             */
            System.out.println((i+1) + ". " + groceryList.get(i));
    }


    /**
     * modifyGroceryItem()
     * @param newItem
     */
    public void modifyGroceryItem(String currentItem, String newItem) {
        int i = findItem(currentItem);
        if (i>=0) {
            modifyGroceryItem(i, newItem);
            System.out.println("Grocery item " + currentItem + " has been modified to " + newItem);
        } else if (i<0) {
            System.out.println("Item " + newItem + " not found.");
        }
    }


    public void removeGroceryItem(String itemToRemove) {
        int i = findItem(itemToRemove);
        if (i>=0) {
            removeGroceryItem(i);
            System.out.println("Grocery item " + itemToRemove + " has been removed.");
        } else if (i<0) {
            System.out.println("Item " + itemToRemove + " not found.");
        }
    }


    /**
```

```java
     * isItemInList()
     * Find item in the ArrayList
     * @param searchItem Item String to search for
     * @return true if found, otherwise false
     */
    public boolean isItemInList(String searchItem) {
        // complicated
//      int i = findItem(searchItem);
//      if(i>=0) {
//          return true;
//      }
//      return false;

        /*
         * ArrayList.contains(item);
         */
        // best practice
        return groceryList.contains(searchItem);
    }



    // == getters ==
    /**
     * getGroceryList()
     * @return groceryList ArrayList<String> Object
     */
    public ArrayList<String> getGroceryList() {
        return groceryList;
    }



    // == private methods ==
    /**
     * modifyGroceryItem()
     * Modifies the item at index i.
     * @param i index
     * @param newItem New String to replace the former item
     */
    private void modifyGroceryItem(int i, String newItem) {
        /*
         * ArrayList.set(index, item)
         */
        groceryList.set(i, newItem); // i = index
        System.out.println("Grocery item " + (i +1) + " modified to " + newItem);
    }


    /**
     * removeGroceryItem()
     * Removes the item with index i form the ArrayList.
     * @param i index
     */
    private void removeGroceryItem(int i) {
        /*
         * ArrayList.remove(index)
         */
        groceryList.remove(i);
    }


    /**
     * findItem()
     * Find item in the ArrayList
     * @param searchItem Item String to search for
     * @return returns the item index if found, otherwise -1 is returned
     */
    private int findItem(String searchItem) {
        /*
         * ArrayList.indexOf(item)
         */
        // Returns the index of the first occurrence of the specified element in this list, or
        // returns -1 if this list does not contain the element.
        return groceryList.indexOf(searchItem);
    }
}
```