Project 0 - Grocery List

In this project, you will create a program that will manage a Grocery List. A user will need to be able to add and remove items from the list, “check off” an item from a list when they obtain that item, and print out the list. You may choose to work with a partner for this project; if you choose to do so, then make sure to put the name of your partner as a comment at the top of your code and also put the name in the comments of your submission when you upload it to Canvas.

Use an Array of Strings in order to maintain the list of items. You also need a second Array to note, for each item in the Array of Strings, if they have been “checked off” or not. This can be done using booleans, characters, Strings, or anything else that you’d like that can be used as a flag to indicate if it is “checked off” or “not checked off” (i.e., the item has been obtained or the item has not been obtained). The second Array’s indexes should be directly related to the first Array’s indexes (i.e., the item in index 0 of the first Array should be related to the “checked off” value in index 0 of the second Array; the same with index 1 of the first and second Arrays and so on). (Note: for those students who know, or are comfortable with, multidimensional Arrays, you may use a 2-dimensional Array instead of two 1-dimensional Arrays).

For the different actions on the list of items please create a menu that will display all the actions available and are numbered 1 through 5 (which action is assigned to which number is up to you) along with a message to prompt the user to input a number to select the action from the menu. The behavior of each action is described below:

- **Add Item** - when the menu number of this action is entered, a prompt will ask the user to input the name of the item they wish to add to the list. The program will check the user input against all existing items in the Array and determine if the item already exists. If the item already exists in the Array, then a message that the item already exists should be displayed, the item should NOT be added to the Array again, and then the program should return to display the menu and prompt the user for a new menu selection/input. If the item does NOT exist in the Array, then it should be added to the end of the list of items and should be marked as “not checked off” in the second Array, then the program should return to display the menu and prompt the user for a new menu selection/input.
- **Remove Item** - when the menu number of this action is entered, a prompt will ask the user to input the name of the item or the number of the item (which is related to the index position of the item in the list, see “Print List” below) they wish to remove from the list. If the item exists in the list (or a valid number/index is given) then remove the item from the list (make sure to move all remaining items in the list over to the left so there are no “gaps” in the array), then the program should return to display the menu and prompt the user for a new menu selection/input. If the item doesn’t exist (or the number/index given is invalid) then a message noting that the item/number doesn’t exist should be displayed and then the program should return to display the menu and prompt the user for a new menu selection/input.
- **“Check Off” Item** - when the menu number of this action is entered, a prompt will ask the user to input the name of the item or the number of the item (which is related to the index position of the item in the list, see “Print List” below) they wish to “check off”. If the

item exists in the list (or a valid number/index is given) then you should note the index of the item in its Array and then mark the same index in the “checked off” Array as “checked off” (note, if the item is already “checked off”, then it should remain “checked off”). If the item doesn’t exist (or the number/index given is invalid) then a message noting that the item/number doesn’t exist should be displayed and then the program should return to display the menu and prompt the user for a new menu selection/input.

- **Print List** - when the menu number of this action is entered, the list of items should be printed/displayed. Each item should be displayed on a separate line, each item should numbered based on its position in the Array (this is the number of the item indicated in the Remove Item and “Check Off” Item actions above), and each item should be notated with a “-” or a “x”, where a “-” means that the item has NOT been checked off and the “x” means that the item has been checked off. After displaying all items, the program should return to display the menu and prompt the user for a new menu selection/input.
- **Exit** - when the menu number of this action is entered, the final list of items should be printed (exactly the same as in “Print List” above), a “goodbye” message should be displayed, and the program should exit/end/terminate.

Submission:
Please export your project from Eclipse (or whatever IDE you are using) as a zip file and upload to the submission on Canvas.

Some additional clarifications, thoughts, and hints:

- How you solve all the problems, display your menu, prompt the user, and modularize your code is all up to you. Please think from the perspective of a potential user (who has NO knowledge of how your program works or all of the different features your program has) interacting with your program and do your best to make it easy to read and understand the prompts and purpose of your program.
- Please modularize (i.e., create methods for code organization and reuse) your code. You should AT LEAST have 5 methods (your “main” method and a method for each of the major actions listed above), but you may have as many methods as you would like. For example, you may want a method that will identify if an item already exists in the list (and return the index of the item) since you will need to do that often (at least for 3 of the 4 actions listed above).
- Take some time to work the problem out on paper before jumping directly to coding. Break down the problem into smaller manageable pieces (try not to get overwhelmed, it seems like a lot of parts, but each part is much smaller than you probably initially think it is), draw some diagrams, write out some pseudocode, figure out any major control structures (i.e., any big branches and loops) you may need, identify common sets of instructions (i.e., sets of instructions you will need to run at many different points in the code to solve different parts of the problem) for modularization, and generally have a plan/solution figured out BEFORE you start coding.
- You need not make the Arrays be the exact same size as the number of items in your list. It is acceptable (and easier) to make the Arrays large (e.g., size/length of 50) and then have a “numberOfItemsInList” variable (note, you can give it a different name) that

will let you know where the “end” of the list is in the Arrays. In this way, you are only using a part of the Arrays at any given moment and you do not have to redefine the Arrays every time you want to add/remove to/from the list. Make sure to increment/decrement the value of “numberOfItemsInList” when you add/remove from the list. You may assume no tests performed for grading purposes will have more than 20 items added to the list.

- When trying to identify if an item already exists in the list, you may simply check for exact match in spelling (e.g., “onion” and “onions” are two different items) but not for case sensitivity (e.g., “onion”, “Onion”, “ONION”, and “oNIOn” are all the same item). (Note, String has a method for this).
- For user input, you may simply use Scanner. Remember to use Scanner’s “has” methods (i.e., “hasNext()”, “hasNextInt()”, “hasNextDouble()”, etc.) to check/determine the type of input the user entered, if needed.
- Please ask questions as needed and seek help in office hours or with our tutors (in the CS main lab or on Slack)!

Examples:

- **Menu Display Example**

  Welcome to Grocery List Management!
  1. Add Item to your Grocery List
  2. Remove Item from your Grocery List
  3. “Check Off” an Item from your Grocery List
  4. Display your Grocery List
  5. Exit

  Please enter the number of an option above:

- **Arrays and Print List Example**

  numberOfItemsInList: 3
  ItemArray:

Chicken	Pasta	Onion	…	...

  //in this example, “not checked off” is “false” and “checked off” is “true”
  //you may do this differently if you wish
  CheckOffArray:

false	false	false	…	…

  Print List:

  1. - Chicken
  2. - Pasta

3. - Onion

After adding the item “Peppers” and checking off “Pasta” and “Onion”:

numberOfItemsInList: 4
ItemArray:

Chicken	Pasta	Onion	Peppers	...

CheckOffArray:

false	true	true	false	…

Print List:

1. - Chicken
2. x Pasta
3. x Onion
4. - Peppers

After removing the item “Chicken”:

numberOfItemsInList: 4
ItemArray:

Pasta	Onion	Peppers	…	...

CheckOffArray:

true	true	false	…	…

Print List:

1. x Pasta
2. x Onion
3. - Peppers

After removing the item number 2:

numberOfItemsInList: 4
ItemArray:

Pasta	Peppers	…	…	...

CheckOffArray:

true	false	…	…	…

Print List:

1. x Pasta
2. - Peppers