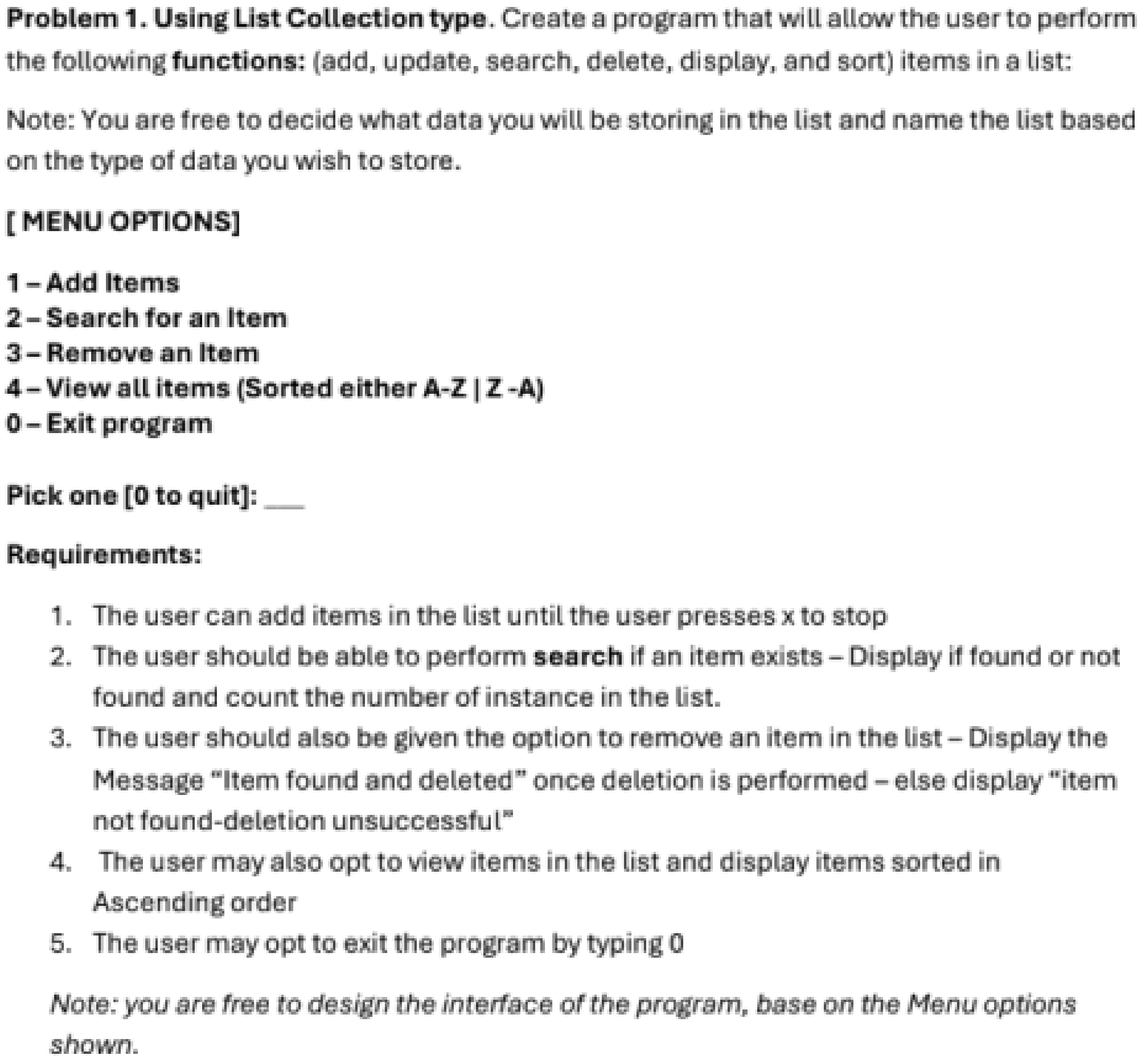
Mechure, Liann S.

BSCS – C204

Midterm Lab Task 3 - Python List Collections

Problem:



Problem 1. Using List Collection type. Create a program that will allow the user to perform the following **functions:** (add, update, search, delete, display, and sort) items in a list:

Note: You are free to decide what data you will be storing in the list and name the list based on the type of data you wish to store.

[MENU OPTIONS]

1 – Add Items
2 – Search for an Item
3 – Remove an Item
4 – View all items (Sorted either A-Z | Z -A)
0 – Exit program

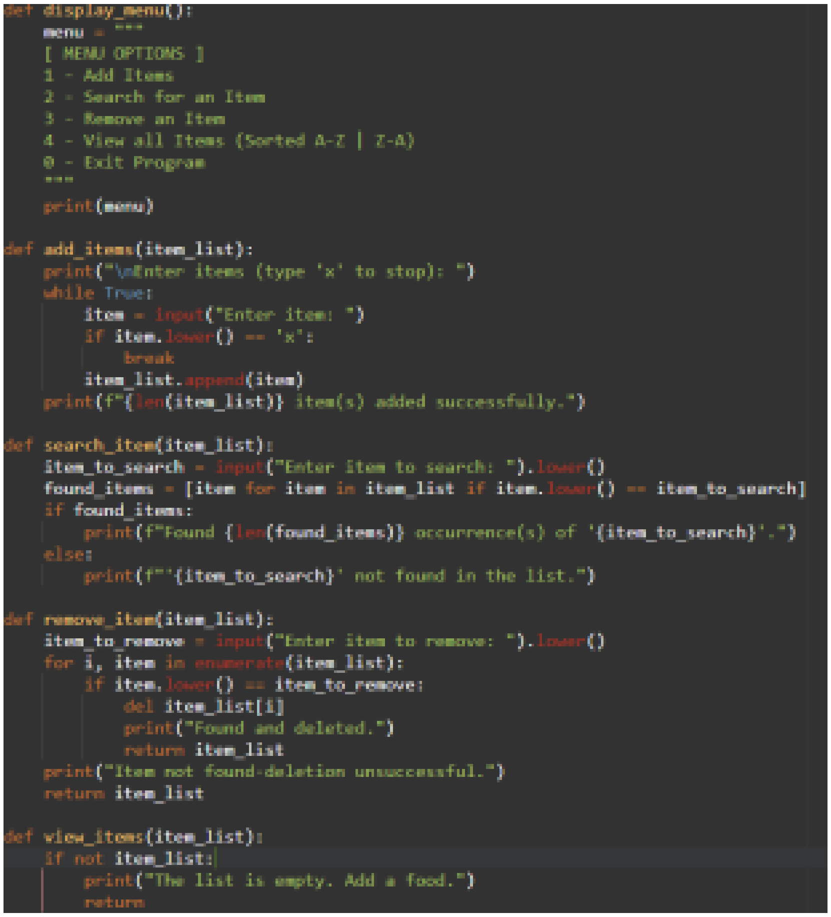
Pick one [0 to quit]: ___

Requirements:

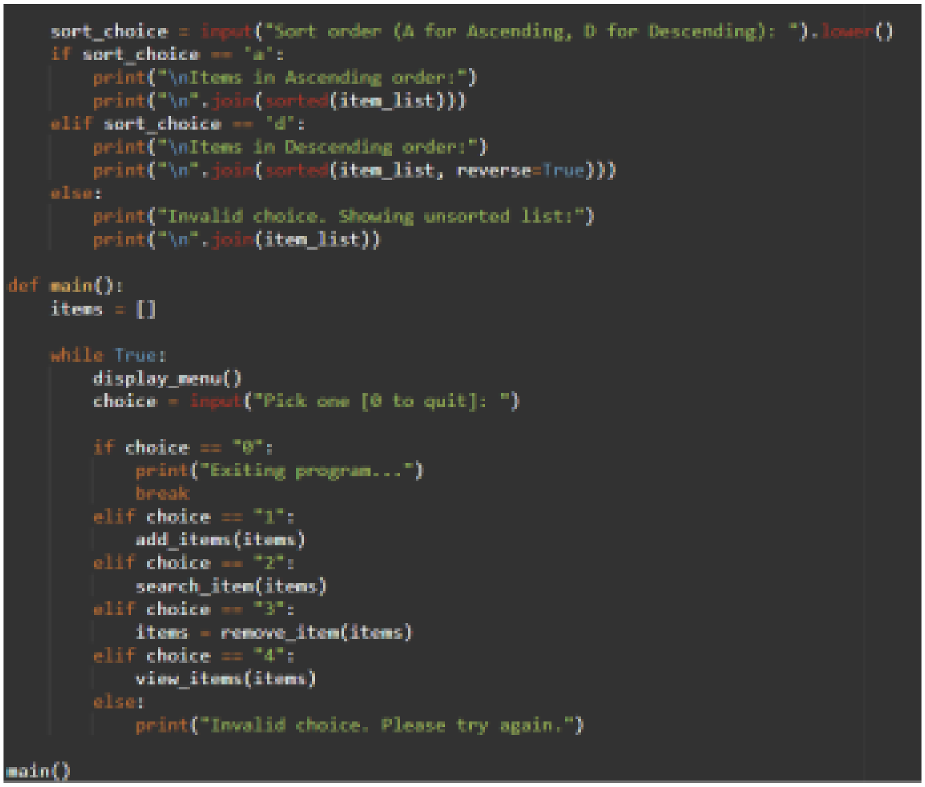
1. The user can add items in the list until the user presses x to stop
2. The user should be able to perform **search** if an item exists – Display if found or not found and count the number of instance in the list.
3. The user should also be given the option to remove an item in the list – Display the Message "Item found and deleted" once deletion is performed – else display "item not found-deletion unsuccessful"
4. The user may also opt to view items in the list and display items sorted in Ascending order
5. The user may opt to exit the program by typing 0

Note: you are free to design the interface of the program, base on the Menu options shown.

Source Code:

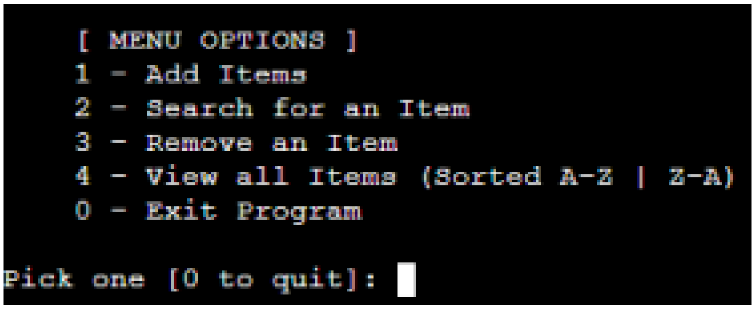


```python
def display_menu():
    menu = """
    [ MENU OPTIONS ]
    1 - Add Items
    2 - Search for an Item
    3 - Remove an Item
    4 - View all Items (Sorted A-Z | Z-A)
    0 - Exit Program
    """
    print(menu)

def add_items(item_list):
    print("\nEnter items (type 'x' to stop): ")
    while True:
        item = input("Enter item: ")
        if item.lower() == 'x':
            break
        item_list.append(item)
    print(f"{len(item_list)} item(s) added successfully.")

def search_item(item_list):
    item_to_search = input("Enter item to search: ").lower()
    found_items = [item for item in item_list if item.lower() == item_to_search]
    if found_items:
        print(f"Found {len(found_items)} occurrence(s) of '{item_to_search}'.")
    else:
        print(f"'{item_to_search}' not found in the list.")

def remove_item(item_list):
    item_to_remove = input("Enter item to remove: ").lower()
    for i, item in enumerate(item_list):
        if item.lower() == item_to_remove:
            del item_list[i]
            print("Found and deleted.")
            return item_list
    print("Item not found-deletion unsuccessful.")
    return item_list

def view_items(item_list):
    if not item_list:
        print("The list is empty. Add a food.")
        return
```

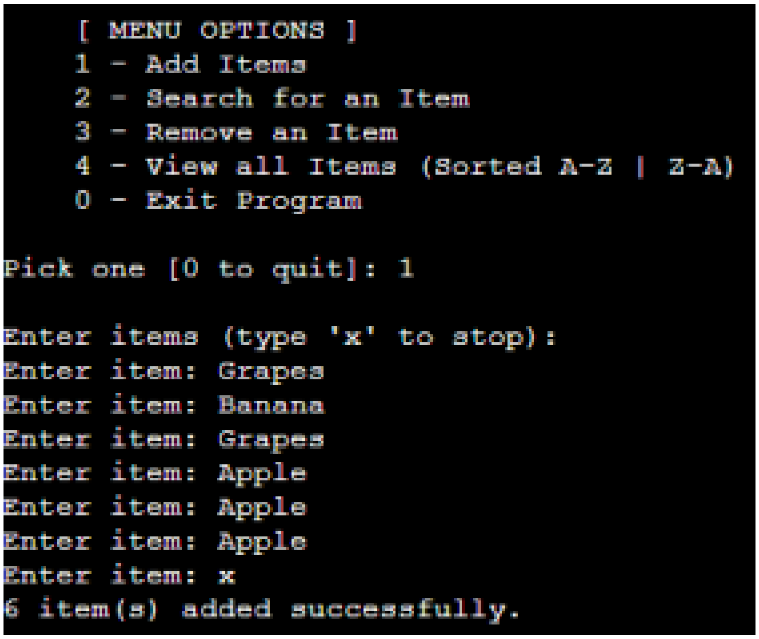


```python
    sort_choice = input("Sort order (A for Ascending, D for Descending): ").lower()
    if sort_choice == 'a':
        print("\nItems in Ascending order:")
        print("\n".join(sorted(item_list)))
    elif sort_choice == 'd':
        print("\nItems in Descending order:")
        print("\n".join(sorted(item_list, reverse=True)))
    else:
        print("Invalid choice. Showing unsorted list:")
        print("\n".join(item_list))

def main():
    items = []

    while True:
        display_menu()
        choice = input("Pick one [0 to quit]: ")

        if choice == "0":
            print("Exiting program...")
            break
        elif choice == "1":
            add_items(items)
        elif choice == "2":
            search_item(items)
        elif choice == "3":
            items = remove_item(items)
        elif choice == "4":
            view_items(items)
        else:
            print("Invalid choice. Please try again.")

main()
```

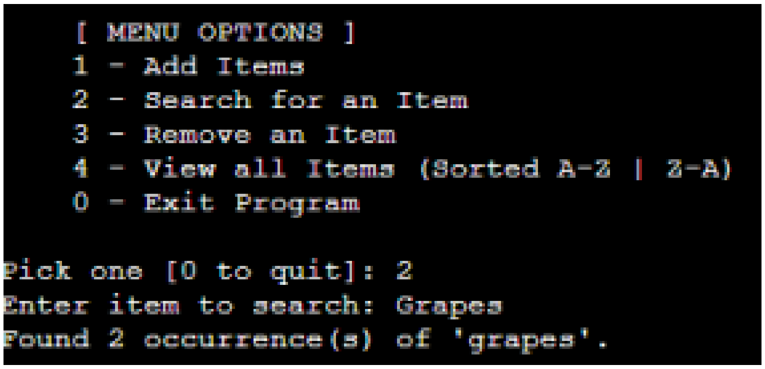
Sample Output:



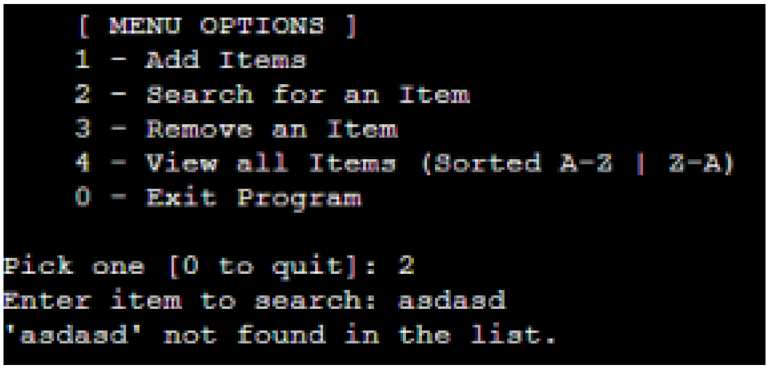
```
    [ MENU OPTIONS ]
    1 - Add Items
    2 - Search for an Item
    3 - Remove an Item
    4 - View all Items (Sorted A-Z | Z-A)
    0 - Exit Program

Pick one [0 to quit]:
```



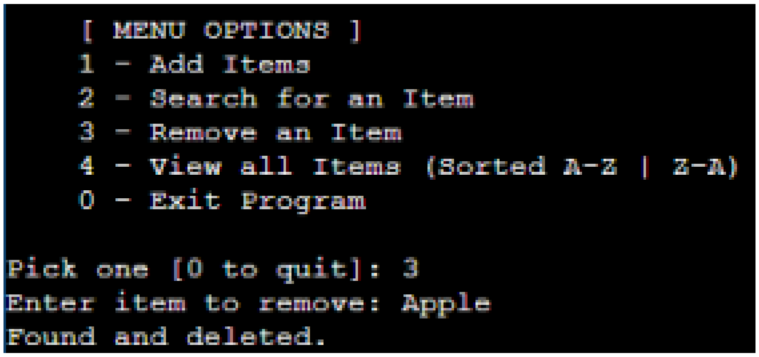
```
    [ MENU OPTIONS ]
    1 - Add Items
    2 - Search for an Item
    3 - Remove an Item
    4 - View all Items (Sorted A-Z | Z-A)
    0 - Exit Program

Pick one [0 to quit]: 1

Enter items (type 'x' to stop):
Enter item: Grapes
Enter item: Banana
Enter item: Grapes
Enter item: Apple
Enter item: Apple
Enter item: Apple
Enter item: x
6 item(s) added successfully.
```



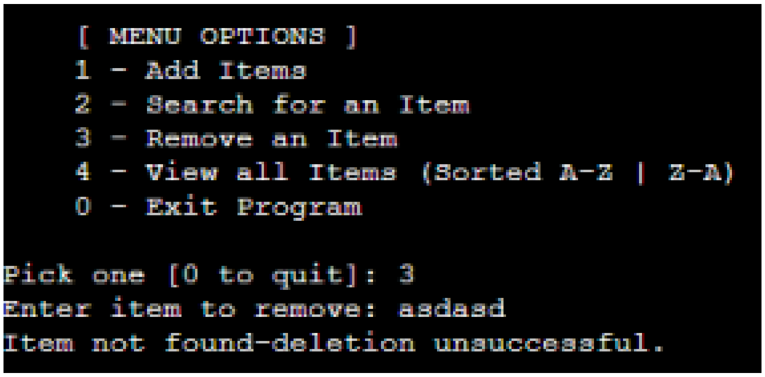
```
    [ MENU OPTIONS ]
    1 - Add Items
    2 - Search for an Item
    3 - Remove an Item
    4 - View all Items (Sorted A-Z | Z-A)
    0 - Exit Program

Pick one [0 to quit]: 2
Enter item to search: Grapes
Found 2 occurrence(s) of 'grapes'.
```



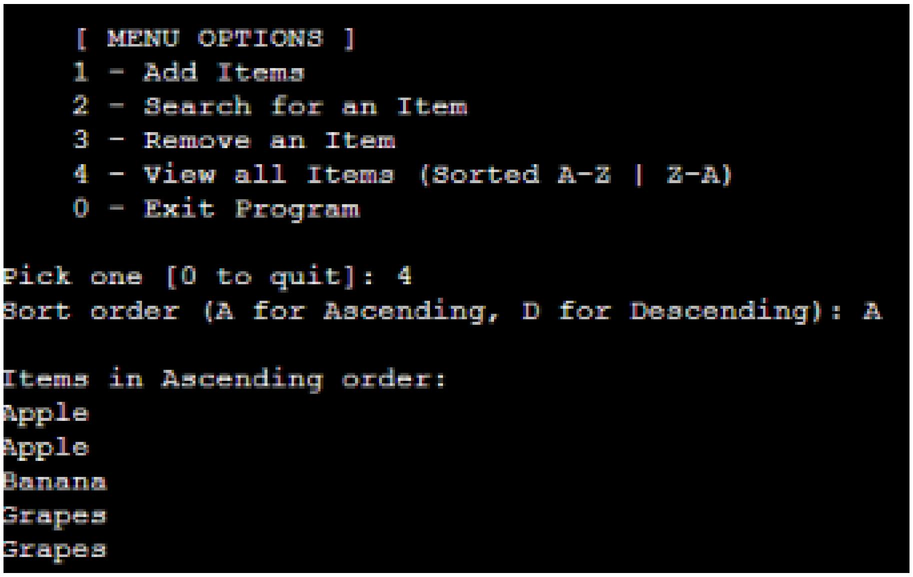
```
    [ MENU OPTIONS ]
    1 - Add Items
    2 - Search for an Item
    3 - Remove an Item
    4 - View all Items (Sorted A-Z | Z-A)
    0 - Exit Program

Pick one [0 to quit]: 2
Enter item to search: asdasd
'asdasd' not found in the list.
```



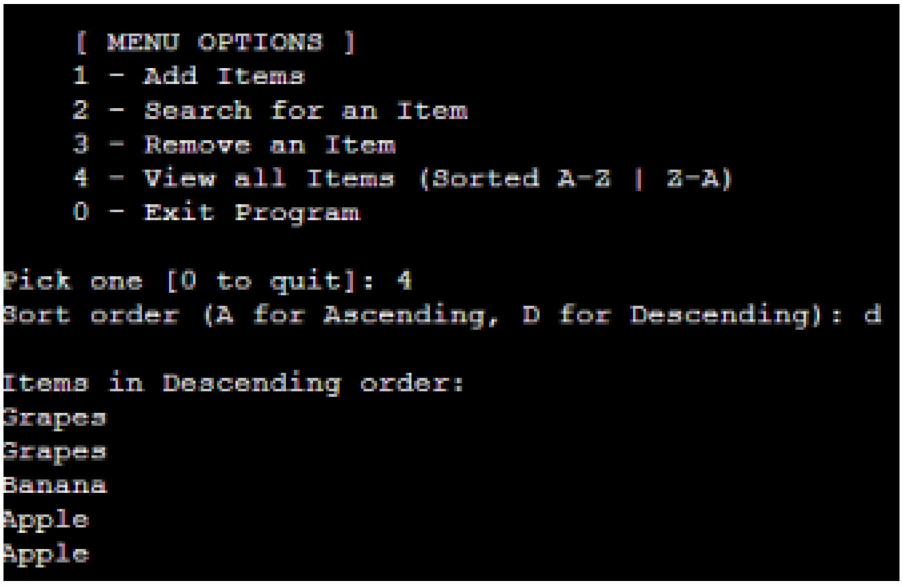
```
    [ MENU OPTIONS ]
    1 - Add Items
    2 - Search for an Item
    3 - Remove an Item
    4 - View all Items (Sorted A-Z | Z-A)
    0 - Exit Program

Pick one [0 to quit]: 3
Enter item to remove: Apple
Found and deleted.
```



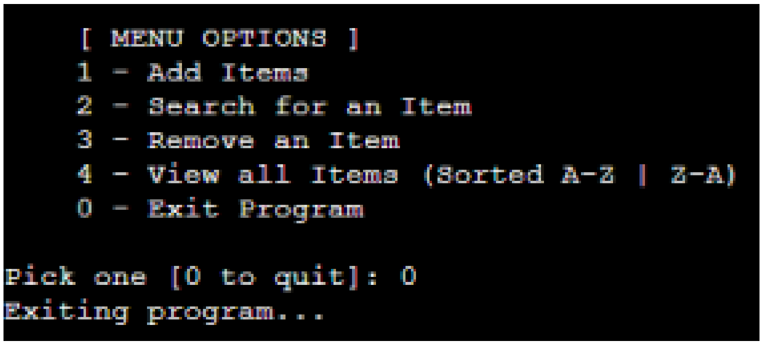
```
    [ MENU OPTIONS ]
    1 - Add Items
    2 - Search for an Item
    3 - Remove an Item
    4 - View all Items (Sorted A-Z | Z-A)
    0 - Exit Program

Pick one [0 to quit]: 3
Enter item to remove: asdasd
Item not found-deletion unsuccessful.
```



```
    [ MENU OPTIONS ]
    1 - Add Items
    2 - Search for an Item
    3 - Remove an Item
    4 - View all Items (Sorted A-Z | Z-A)
    0 - Exit Program

Pick one [0 to quit]: 4
Sort order (A for Ascending, D for Descending): A

Items in Ascending order:
Apple
Apple
Banana
Grapes
Grapes
```



```
    [ MENU OPTIONS ]
    1 - Add Items
    2 - Search for an Item
    3 - Remove an Item
    4 - View all Items (Sorted A-Z | Z-A)
    0 - Exit Program

Pick one [0 to quit]: 4
Sort order (A for Ascending, D for Descending): d

Items in Descending order:
Grapes
Grapes
Banana
Apple
Apple
```



```
    [ MENU OPTIONS ]
    1 - Add Items
    2 - Search for an Item
    3 - Remove an Item
    4 - View all Items (Sorted A-Z | Z-A)
    0 - Exit Program

Pick one [0 to quit]: 0
Exiting program...
```