

## Midterm Lab Task 3 - Python List Collections

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.

```
def add_items(items): #ADD ITEM/S 1 usage
    while True:
        item = input("Enter item (x to stop): ")
        if item.lower() == "x":
            break
        items.append(item)
```

```
*****[ MENU ]*****
1 - Add Items
2 - Search for an Item
3 - Remove an Item
4 - Update an Item
5 - View all Items
0 - Exit
Pick one: 1
Enter item (x to stop): Hotdog
Enter item (x to stop): Itlog
Enter item (x to stop): Manok
Enter item (x to stop): Kanin
Enter item (x to stop): A
Enter item (x to stop): B
Enter item (x to stop): C
Enter item (x to stop): x
*****[ MENU ]*****
1 - Add Items
2 - Search for an Item
3 - Remove an Item
4 - Update an Item
5 - View all Items
0 - Exit
Pick one:
```

```
def search_item(items): #SEARCH ITEM 1 usage
    item = input("Enter item to search: ")
    count = items.count(item)
    if count > 0:
        print(f"{item} found {count} time(s).")
    else:
        print(f"{item} not found.")
```

```
*****[ MENU ]*****
1 - Add Items
2 - Search for an Item
3 - Remove an Item
4 - Update an Item
5 - View all Items
0 - Exit
Pick one: 2
Enter item to search: Hotdog
Hotdog found 1 time(s).
```

```
*****[ MENU ]*****
1 - Add Items
2 - Search for an Item
3 - Remove an Item
4 - Update an Item
5 - View all Items
0 - Exit
Pick one: 2
Enter item to search: Samgyupsal
Samgyupsal not found.
```

```
def remove_item(items): #REMOVE ITEM 1 usage
    item = input("Enter item to remove: ")
    if item in items:
        items.remove(item)
        print("Item found and deleted.")
    else:
        print("Item not found.")
```

```
*****[ MENU ]*****
1 - Add Items
2 - Search for an Item
3 - Remove an Item
4 - Update an Item
5 - View all Items
0 - Exit
Pick one: 3
Enter item to remove: Hotdog
Item found and deleted.
```

```
*****[ MENU ]*****
1 - Add Items
2 - Search for an Item
3 - Remove an Item
4 - Update an Item
5 - View all Items
0 - Exit
Pick one: 3
Enter item to remove: Bulgogi
Item not found.
```

```
def update_item(items): #UPDATE ITEM 1 usage
    old_item = input("Enter item to update: ")
    if old_item in items:
        new_item = input("Enter new item: ")
        index = items.index(old_item)
        items[index] = new_item
        print(f"{old_item} updated to {new_item}.")
    else:
        print("Item not found - update unsuccessful.")
```

```
*****[ MENU ]*****
1 - Add Items
2 - Search for an Item
3 - Remove an Item
4 - Update an Item
5 - View all Items
0 - Exit
Pick one: 4
Enter item to update: Itlog
Enter new item: Kimchi
Itlog updated to Kimchi.
```

```
*****[ MENU ]*****
1 - Add Items
2 - Search for an Item
3 - Remove an Item
4 - Update an Item
5 - View all Items
0 - Exit
Pick one: 4
Enter item to update: Fish Cake
Item not found - update unsuccessful.
```

```
def view_items(items): #VIEW ITEMS 1 usage
    if not items:
        print("List is empty.")
        return
    order = input("Sort order (A=Ascending, D=Descending): ").lower()
    if order == "a":
        print(sorted(items))
    elif order == "d":
        print(sorted(items, reverse=True))
    else:
        print(items)
```

```
*****[ MENU ]*****
```

```
1 - Add Items
2 - Search for an Item
3 - Remove an Item
4 - Update an Item
5 - View all Items
0 - Exit
```

Pick one: 5

Sort order (A=Ascending, D=Descending): a

```
['A', 'B', 'C', 'Kanin', 'Kimchi', 'Manok']
```

```
*****[ MENU ]*****
```

```
1 - Add Items
2 - Search for an Item
3 - Remove an Item
4 - Update an Item
5 - View all Items
0 - Exit
```

Pick one: 5

Sort order (A=Ascending, D=Descending): d

```
['Manok', 'Kimchi', 'Kanin', 'C', 'B', 'A']
```

```
def main(): 1 usage
    items = [] #FOR LISTING ITEMS

    while True:
        print("*****[ MENU ]*****")
        print("1 - Add Items")
        print("2 - Search for an Item")
        print("3 - Remove an Item")
        print("4 - Update an Item")
        print("5 - View all Items")
        print("0 - Exit")
        choice = input("Pick one: ")

        if choice == "1":
            add_items(items)
        elif choice == "2":
            search_item(items)
        elif choice == "3":
            remove_item(items)
        elif choice == "4":
            update_item(items)
        elif choice == "5":
            view_items(items)
        elif choice == "0":
            print("Salamat! Thanks!!")
            break
        else:
            print("Invalid choice!")

#RUN PROGRAM
main()
```