

Ex. No: 7-A  
Date 04/02/2023

## List items in Library

**AIM:**

To create a list of items, present in a library and do all the operations on it.

**ALGORITHM:**

STEP 1: Start

STEP 2: Create a list named liblist that contains the elements Fiction, Academics, Novel, Science  
Fiction, Horror as the elements .

STEP 3: Display length of list by using len() function.

STEP 4: Append "Fictional Novel" to the list.

STEP 5: Display liblist.

STEP 6: Insert "Story book" to the index value 0.

STEP 7: Display liblist.

STEP 8: Assign a list named more\_liblist and the values in list are Novel, Science, Project

STEP 9 : Assign a value conc and Concatenate the Lists liblist and more\_liblist

STEP 10: Remove "Horror"

STEP 11: Display liblist.

STEP 12: Pop the value from the list using pop() function

STEP 13: Display liblist.

STEP 14: Display the index value of novel using the index() function

STEP 15: Display the count of novel using the count() function

STEP 16: Sort the liblist using sort() function

STEP 17: Display liblist.

STEP 18: Reverse the liblist using Reverse() function

STEP 19: Display liblist.

STEP 20: Display the minimum value of liblist using the min() function

STEP 21: Display the maximum value of liblist using the max() function

STEP 22: Display the count of novel using the count() function

STEP 23: Calculate liblist\*3 and display

STEP 24: Stop

NAVEEN KUMAR K  
22CSEB47

**SOURCE CODE:**

```
liblist = ["Fiction", "Academics", "Novel", "Science Fiction", "Horror"]  
  
print("Length of list: ", len(liblist))  
  
liblist.append("Fictional Novel")  
  
print("After append: ", liblist)  
  
liblist.insert(0, "Story Books")  
  
print("After insert: ", liblist)  
  
more_liblist = ["Novel", "Science", "Projects"]  
  
conc= liblist + more_liblist  
  
print("List concatenation: ",conc)  
  
liblist.remove("Horror")  
  
print("After remove: ", liblist)  
  
popped = liblist.pop()  
  
print("Popped value: ", popped)  
  
print("After pop: ", liblist)  
  
print("Index of Novel: ", liblist.index("Novel"))  
  
print("Count of Novel: ", liblist.count("Novel"))  
  
liblist.sort()  
  
print("After sort: ", liblist)  
  
liblist.reverse()  
  
print("After reverse: ", liblist)  
  
print("Minimum value: ", min(liblist))  
  
print("Maximum value: ", max(liblist))  
  
print("List repetition: ", liblist * 3)
```

**OUTPUT:**

Length of list: 5

After append: ['Fiction', 'Academics', 'Novel', 'Science Fiction', 'Horror', 'Fictional Novel']

After insert: ['Story Books', 'Fiction', 'Academics', 'Novel', 'Science Fiction', 'Horror', 'Fictional Novel']

List concatenation: ['Story Books', 'Fiction', 'Academics', 'Novel', 'Science Fiction', 'Horror', 'Fictional Novel', 'Novel', 'Novel', 'Science', 'Projects']

After remove: ['Story Books', 'Fiction', 'Academics', 'Novel', 'Science Fiction', 'Fictional Novel']

Popped value: Fictional Novel

After pop: ['Story Books', 'Fiction', 'Academics', 'Novel', 'Science Fiction']

Index of Novel: 3

Count of Novel: 1

After sort: ['Academics', 'Fiction', 'Novel', 'Science Fiction', 'Story Books']

After reverse: ['Story Books', 'Science Fiction', 'Novel', 'Fiction', 'Academics']

Minimum value: Academics

Maximum value: Story Books

List repetition: ['Story Books', 'Science Fiction', 'Novel', 'Fiction', 'Academics', 'Story Books', 'Science Fiction', 'Novel', 'Fiction', 'Academics', 'Story Books', 'Science Fiction', 'Novel', 'Fiction', 'Academics']

**AIM:**

To create a tuple for components of a car and show all the operations.

**ALGORITHM:**

STEP 1: Start

STEP 2: Create a tuple named car\_components that contains the elements Hood, Turo, Doors, Steering , Doors as the elements .

STEP 3: Display length of tuple by using **len()** function.

STEP 4: Display the index value of Doors using the index() function

STEP 5: Display the count of Doors using the count() function

STEP 6: Display the minimum value of car\_components using the min() function

STEP 7: Display the maximum value of car\_components using the max() function

STEP 8: Calculate car\_components\*3 and display

STEP 9: Assign a tuple named more\_car\_components and the values in list are Spoilers, Rims, Alloy Wheels

STEP 10: Concatenate the Tuple Car\_components and more\_car\_components and display

STEP 11: Stop

**SOURCE CODE:**

```
car_components = ("Hood","Turbo","Doors","Steering","Doors")
print("Length of tuple: ", len(car_components))
print("Index of Doors: ", car_components.index("Doors"))
print("Count of 2: ", car_components.count("Doors"))
print("Minimum value: ", min(car_components))
print("Maximum value: ", max(car_components))
print("Tuple repetition: ", car_components * 3)
more_car_components = ("Spoilers","Rims","Alloy Wheels")
print("Tuple concatenation: ", car_components + more_car_components)
```

**OUTPUT:**

Length of tuple: 5

Index of Doors: 2

Count of 2: 2

Minimum value: Doors

Maximum value: Turbo

Tuple repetition: ('Hood', 'Turbo', 'Doors', 'Steering', 'Doors', 'Hood', 'Turbo', 'Doors', 'Steering',  
'Doors', 'Hood', 'Turbo', 'Doors', 'Steering', 'Doors')

Tuple concatenation: ('Hood', 'Turbo', 'Doors', 'Steering', 'Doors', 'Spoilers', 'Rims', 'Alloy Wheels')

**AIM:**

To Create a set to accept more values and print the elements after removing the duplicate content.

**ALGORITHM:**

STEP 1: Start

STEP 2: Create a empty list

STEP 3: Check for i in range 0 to 5 if true goto step 3.3 else goto step 4

STEP 3.1: Append the value of i

STEP 3.2: Append the value 4

STEP 3.3: Append the value 5

STEP 4: print list

STEP 5: Assign the list value to x as set

STEP 6: Print x

STRP 7: Stop

**SOURCE CODE:**

```
list=[]  
for i in range(0,5):  
    list.append(i)  
    list.append(4)  
    list.append(5)  
print("Created list which contains duplicate elements :",list)  
x=set(list)  
print("After creating set removes duplicate elements :",x)
```

**OUTPUT:**

Created list which contains duplicate elements : [0, 4, 5, 1, 4, 5, 2, 4, 5, 3, 4, 5, 4, 4, 5]

After creating set removes duplicate elements : {0, 1, 2, 3, 4, 5}

Ex. No: 7-D  
Date 04/02/2023

## Laptop specification using Dictionary

### AIM:

To write a program to print the specification of the laptop using dictionary with its operation.

### ALGORITHM:

STEP 1: Start

STEP 2: Create a Dictionary and store in a value laptop.

STEP 3: Assign the key values for Brand:'Asus' , Model:'Vivobook 15' , Processor:'intel core i5' ,  
RAM: 8 , Storage:'1TB' , Graphics:' NVIDIA GeForce RTX 3080' , Screen\_size:15.6

STEP 4: Display laptop specification

STEP 5: Display the value in the key Brand

STEP 6: Display the value in the key Model

STEP 7: Display the value in the key Processor

STEP 8: Display the value in the key RAM

STEP 9: Display the value in the key Storage

STEP 10: Display the value in the key Graphics

STEP 11: Display the value in the key Screen\_size

STEP 12 : Stop

**SOURCE CODE:**

```
laptop={ "Brand":"Asus",  
         "Model":"Vivobook 15",  
         "Processor":"Intel Core i5",  
         "RAM":8,  
         "Storage":"1TB",  
         "Graphics":"NVIDIA GeForce RTX 3080",  
         "Screen_size":15.6 }  
  
print("Laptop Specification : ")  
print("Brand : ",laptop["Brand"])  
print("Model : ",laptop["Model"])  
print("Processor : ",laptop["Processor"])  
print("RAM : ",laptop["RAM"])  
print("Storage : ",laptop["Storage"])  
print("Graphics : ",laptop["Graphics"])  
print("Screen Size : ",laptop["Screen_size"])
```

**OUTPUT:**

```
Laptop Specification :  
  
Brand : Asus  
  
Model : Vivobook 15  
  
Processor : Intel Core i5  
  
RAM : 8  
  
Storage : 1TB  
  
Graphics : NVIDIA GeForce RTX 3080  
  
Screen Size : 15.6
```

**RESULT:**

Thus, the programs for the list, tuple, set and dictionaries are given with the result successfully