REALTIME APPLICATIONS USING LIST, TUPLE, SET, DICTIONARY

EX.NO.7A 1. ITEMS PRESENT IN LIBRARY(LIST)

DATE:5.2.23

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 Grammar, Physics, Comics, Electronics, Economics as the elements .

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

STEP 4: Append "History" to the list.

STEP 5: Display liblist.

STEP 6: Insert "Zoology " 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 "Grammar"

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 Comics using the index() function

STEP 15: Display the count of Comics 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

PROGRAM:

```
#Items present in library
liblist=["Grammar","Physics","Comics","Electronics","Economics"]
print("Length of list: ", len(liblist))
liblist.append("History")
print("After append: ", liblist)
liblist.insert(0, "Zoology")
print("After insert: ", liblist)
more_liblist = ["Novel", "Science", "Projects"]
conc= liblist + more_liblist
print("List concatenation: ",conc)
liblist.remove("Grammar")
print("After remove: ", liblist)
popped = liblist.pop()
print("Popped value: ", popped)
print("After pop: ", liblist)
print("Index of Novel: ", liblist.index("Comics"))
print("Count of Novel: ", liblist.count("Comics"))
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:  ['Grammar', 'Physics', 'Comics', 'Electronics', 'Economics', 'History']
After insert:  ['Zoology', 'Grammar', 'Physics', 'Comics', 'Electronics', 'Economics', 'History']
List concatenation:  ['Zoology', 'Grammar', 'Physics', 'Comics', 'Electronics', 'Economics',
'History', 'Novel', 'Science', 'Projects']
After remove:  ['Zoology', 'Physics', 'Comics', 'Electronics', 'Economics', 'History']
Popped value:  History
After pop:  ['Zoology', 'Physics', 'Comics', 'Electronics', 'Economics']
Index of Novel:  2
Count of Novel:  1
After sort:  ['Comics', 'Economics', 'Electronics', 'Physics', 'Zoology']
After reverse:  ['Zoology', 'Physics', 'Electronics', 'Economics', 'Comics']
Minimum value:  Comics
Maximum value:  Zoology
List repetition:  ['Zoology', 'Physics', 'Electronics', 'Economics', 'Comics', 'Zoology', 'Physics',
'Electronics', 'Economics', 'Comics', 'Zoology', 'Physics', 'Electronics', 'Economics', 'Comics']
```

EXNO.7B 2. COMPONENTS OF CAR(TUPLE)

DATE:5.2.23

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 Engine, Turbo, Doors, Steering , Tyres 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

PROGRAM:

```
#Components of car
car_components = ("Engine","Turbo","Doors","Steering","Tyre")
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:  1
Minimum value:  Doors
Maximum value:  Tyre
Tuple repetition:  ('Engine', 'Turbo', 'Doors', 'Steering', 'Tyre', 'Engine', 'Turbo', 'Doors',
'Steering', 'Tyre', 'Engine', 'Turbo', 'Doors', 'Steering', 'Tyre')
Tuple concatenation:  ('Engine', 'Turbo', 'Doors', 'Steering', 'Tyre', 'Spoilers', 'Rims', 'Alloy
Wheels')
```

EX.NO.7C 3. REMOVING DUPLICATES (SET)

DATE:5.2.23

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 8 if true goto step 3.3 else goto step 4

3.1: Append the value of i

3.2: Append the value 4

3.3: Append the value 5

STEP 4: print list

STEP 5: Assign the list value to x as set

STEP 6: Print x

STEP 7: Stop

PROGRAM:

```
#Removing duplicates
list=[]
for i in range(0,8):
    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, 5, 4, 5, 6, 4, 5, 7, 4, 5]
After creating set removes duplicate elements :  {0, 1, 2, 3, 4, 5, 6, 7}
```

EX.NO.7D 4. SPECIFICATIONS OF LAPTOP(DICTIONARY)

DATE:5.2.23

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 My_laptop.

STEP 3: Assign the key values for Brand:’Lenovo ’ , Model:’IdeaPad slim3i ’ , Processor:’intel core i3’ , RAM: 8 , Storage:’512GB’ , Graphics:’ NVIDIA MX330 ’ , 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

PROGRAM:

```
#Specifications of laptop
My_laptop={ "Brand":"Lenovo","Model":"Ideapad slim3i","Processor":"Intel Core i3",
      "RAM":8,"Storage":"512GB","Graphics":"NVIDIA MX330",
        "Screen_size":15.6 }
print("Laptop Specification : ")
print("Brand : ",My_laptop["Brand"])
print("Model : ",My_laptop["Model"])
print("Processor : ",My_laptop["Processor"])
print("RAM : ",My_laptop["RAM"])
print("Storage : ",My_laptop["Storage"])
print("Graphics : ",My_laptop["Graphics"])
print("Screen Size : ",My_laptop["Screen_size"])
```

OUTPUT:

```
Laptop Specification :
Brand :  Lenovo
Model :  Ideapad slim3i
Processor :  Intel Core i3
RAM :  8
Storage :  512GB
Graphics :  NVIDIA MX330
Screen Size :  15.6
```

RESULT:

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