**REAL-TIME APPLICATIONS USING LISTS, TUPLES ,SETS AND DICTIONARIES**

**EXERCISE NO** :7

**DATE** :04.02.2023

1. **Create a list for Items present in a Library and and do all the operations on it .**

**AIM:**

To create a list for Items present in a Library and and do all the operations on it .

**PROGRAM:**

lib1=[]

lib2=[]

n=int(input("Enter the number of items to be entered in list1:"))

for i in range(1,n+1):

str1=input("Enter the item:")

lib1.append(str1)

print(lib1)

x=int(input("Enter the number of items to be entered in list2:"))

for i in range(1,x+1):

str2=input("Enter the item:")

lib2.append(str2)

print(lib2)

print("Concatenation:",lib1+lib2)

print("Repetition:",lib1\*3)

print("Indexing:",lib2[-3])

print("Slicing:",lib2[0:2])

lib1.append("story book")

print("Append:",lib1)

lib1.extend([lib2])

print("Extend:",lib1)

lib2.insert(2,"story book")

print("Insert:",lib2)

lib1.pop(3)

print("Pop:",lib1)

lib2.remove("python")

print("Remove:",lib2)

del(lib1[0:2])

print("Deleting:",lib1)

**OUTPUT:**

Enter the number of items to be entered in list1:4

Enter the item: maths book

Enter the item: science book

Enter the item: english book

Enter the item: social book

['maths book', 'science book', 'english book', 'social book']

Enter the number of items to be entered in list2:3

Enter the item: python

Enter the item: java

Enter the item: C++

['python', 'java', 'C++']

Concatenation: ['maths book', 'science book', 'english book', 'social book', 'python', 'java', 'C++']

Repetition: ['maths book', 'science book', 'english book', 'social book', 'maths book', 'science book', 'english book', 'social book', 'maths book', 'science book', 'english book', 'social book']

Indexing: python

Slicing: ['python', 'java']

Append: ['maths book', 'science book', 'english book', 'social book', 'story book']

Extend: ['maths book', 'science book', 'english book', 'social book', 'story book', ['python', 'java', 'C++']]

Insert: ['python', 'java', 'story book', 'C++']

Pop: ['maths book', 'science book', 'english book', 'story book', ['python', 'java', 'story book', 'C++']]

Remove: ['java', 'story book', 'C++']

Deleting: ['english book', 'story book', ['java', 'story book', 'C++']]

1. **Create a tuple for components of a Car and show all the operations**.

**AIM:**

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

**PROGRAM:**

tup1=('Engine','Tyre','Seat','Battery')

tup2=('Brake','Steering-wheel','Gear','Headlights')

print(tup1)

print(tup2)

print("Append/Concatenation: ",tup1+tup2)

print("Repetition: ",tup1\*3)

print("Slicing: ")

print(tup2[2])

print(tup1[1:])

print(tup2[:-1])

print("Membership: ")

print('Seat' in tup1)

print('Accelerator' in tup2)

print("Length: ",len(tup1))

print("Sorting: ",sorted(tup2))

**OUTPUT:**

('Engine','Tyre', 'Seat', 'Battery')

('Brake', 'Steering-wheel', 'Gear', 'Headlights')

Append/Concatenation: ('Engine', 'Tyre', 'Seat', 'Battery', 'Brake', 'Steering-wheel', 'Gear', 'Headlights')

Repetition: ('Engine', 'Tyre', 'Seat', 'Battery', 'Engine', 'Tyre', 'Seat', 'Battery', 'Engine', 'Tyre', 'Seat', 'Battery')

Slicing:

Gear

('Tyre', 'Seat', 'Battery')

('Brake', 'Steering-wheel', 'Gear')

Membership:

True

False

Length: 4

Sorting: ['Brake', 'Gear', 'Headlights', 'Steering-wheel']

1. **Create a set to accept more values and print the elements after removing the duplicate contents.**

**AIM:**

To create a set to accept more values and print the elements after removing the duplicate contents.

**PROGRAM:**

l=[]

n=int(input("Enter the number of elements: "))

for i in range(n):

e=input("Enter an element: ")

l.append(e)

set1=set(l)

print("Set without duplicate elements: ")

print(set1)

**OUTPUT:**

Enter the number of elements:10

Enter an element:35

Enter an element:13

Enter an element:58

Enter an element:35

Enter an element:48

Enter an element:13

Enter an element:8

Enter an element:47

Enter an element:48

Enter an element:38

Set without duplicate elements:

{'48', '38', '8', '35', '47', '58', '13'}

1. **Write a program to print the specifications of the laptop using dictionary with its operations.**

**AIM:**

To write a program to print the specifications of the laptop using dictionary with its operations.

**PROGRAM:**

dict1={'Brand': 'Asus', 'Model': 'HN272T', 'Processor': 'Intel core I5', 'RAM': '8GB', 'OS': 'Windows 10'}

print(dict1)

dict1['RAM']= '16GB'

print('Changing element: ',dict1)

dict1['Hard disk']= '512GB '

print('Adding element: ',dict1)

dict1.popitem()

print('Deleting element: ',dict1)

print('Accessing element: ',dict1['OS'])

print('Keys: ',dict1.keys())

print('Values: ',dict1.values())

**OUTPUT:**

{'Brand': 'Asus', 'Model': 'HN272T', 'Processor': 'Intel core I5', 'RAM': '8GB', 'OS'

: 'Windows 10'}

Changing element: {'Brand': 'Asus', 'Model': 'HN272T', 'Processor': 'Intel core I5', 'RAM': '16GB', 'OS': 'Windows 10'}

Adding element: {'Brand': 'Asus', 'Model': 'HN272T', 'Processor': 'Intel core I5', 'RAM': '16GB', 'OS': 'Windows 10', 'Hard disk': '512GB'}

Deleting element: {'Brand': 'Asus', 'Model': 'HN272T', 'Processor': 'Intel core I5', 'RAM': '16GB', 'OS': 'Windows 10'}

Accessing element: Windows 10

Keys: dict\_keys(['Brand', 'Model', 'Processor', 'RAM', 'OS'])

Values: dict\_values(['Asus', 'HN272T', 'Intel core I5', '16GB', 'Windows 10'])

**RESULT:**

Thus, the programs were executed and the outputs were verified successfully.