Class name : SY CSE(IOT)

Rollno : 2007

Urn no : 1022101007

Batch : S1

EXPERIMENT 3

1. Apply python built in data types: Strings, List, Tuple, Dictionary, Set and theirmethods to solve any given problem.

a) Create a list and perform the following methods 1) insert() 2) remove() 3) append() 4)

len() 5) pop() 6) clear()

CODE:-

a =[10,20,30,40,50,60,70,80,90,100]

print(a)

a.insert(3,300)

print(a)

a.remove(50)

print(a)

a.append(1000)

print(a)

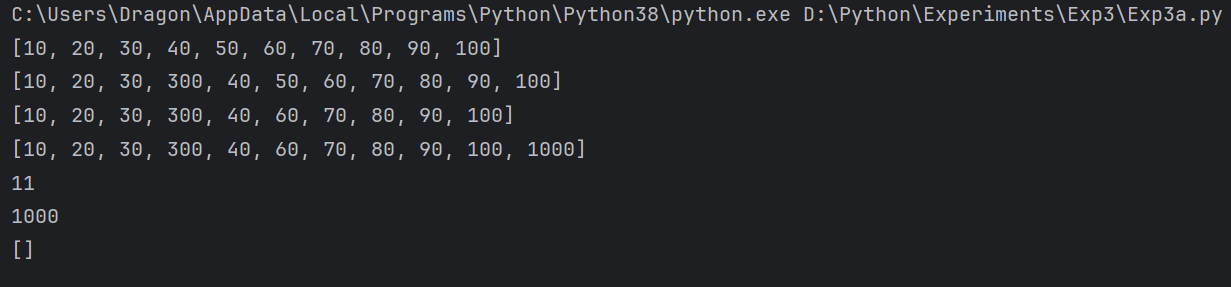
print(len(a))

print(a.pop())

a.clear()

print(a)

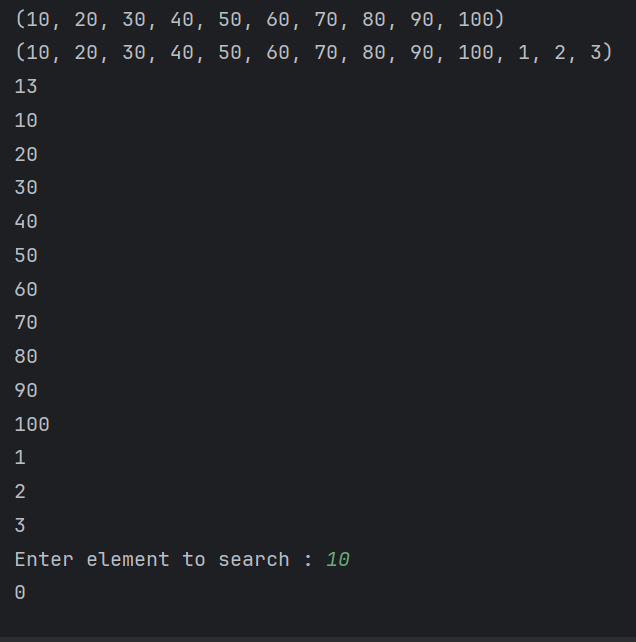
OUTPUT:-



b) Create a tuple and perform the following methods 1) Add items 2) len() 3) check for

item in tuple 4)Access iems

CODE:-



c) Create a dictionary and apply the following methods 1) Print the dictionary items 2)

access items 3) use get() 4)change values 5) use len()

CODE:-

x = {1:'Viraj' ,2:'Riddesh',3 : 'Nikhil',4 : 'Pranita',5:'Gananjay'}

print(x)

print(x[3])

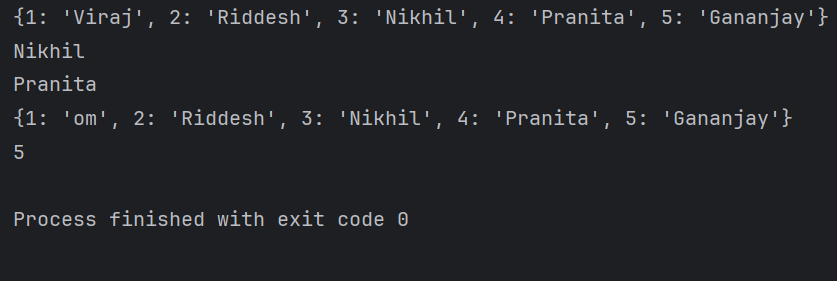
print(x.get(4))

x[1]='om'

print(x)

print(len(x))

OUTPUT:-



d) Demonstrate the following functions/methods which operates on sets in Python with

suitable examples:

i) add( ) ii) update( ) iii) copy( ) iv) pop( )

v) remove( ) vi)discard( ) vii) clear( ) viii) union()

ix) intersection( ) x) difference( )

CODE:-

# Creating sets for demonstration

set1 = {1, 2, 3}

set2 = {3, 4, 5}

set3 = {5, 6, 7}

# i) add() - Adds an element to the set

set1.add(4)

print("After add(4):", set1)

# ii) update() - Updates the set with the union of itself and others

set1.update([5, 6])

print("After update([5, 6]):", set1)

# iii) copy() - Returns a shallow copy of the set

set\_copy = set1.copy()

print("Copy of set1:", set\_copy)

# iv) pop() - Removes and returns an arbitrary set element

popped\_element = set1.pop()

print("Popped element:", popped\_element)

print("After pop():", set1)

# v) remove() - Removes a specified element (raises KeyError if not found)

set1.remove(2)

print("After remove(2):", set1)

# vi) discard() - Removes a specified element (does nothing if not found)

set1.discard(3)

print("After discard(3):", set1)

set1.discard(10) # 10 is not in the set, so nothing happens

print("After discard(10):", set1)

# vii) clear() - Removes all elements from the set

set1.clear()

print("After clear():", set1)

# Resetting sets for further demonstrations

set1 = {1, 2, 3}

set2 = {3, 4, 5}

set3 = {5, 6, 7}

# viii) union() - Returns the union of sets as a new set

union\_set = set1.union(set2, set3)

print("Union of set1, set2, set3:", union\_set)

# ix) intersection() - Returns the intersection of sets as a new set

intersection\_set = set1.intersection(set2, set3)

print("Intersection of set1, set2, set3:", intersection\_set)

# x) difference() - Returns the difference of sets as a new set

difference\_set = set1.difference(set2, set3)

print("Difference of set1 with set2 and set3:", difference\_set)

OUTPUT:-

