Task4:- Use various data types, list, Tuples and Dictionary in Python programming.

Aim:- To use various data types, list, Tuples and dictionary in the Python programming.

(a) You are working on a python project that requires you to manage and manipulate a list of numbers. Your task is to create a python program that demonstrates the following list operations:

Algorithm:-

1. Start
2. for adding elements to a list first create a list with name "list" and assign the values within [] breackets, in order to add a new value use the function append().
3. for removing a specific element use "pop (index value)" or remove (item name)".
4. for sorting the elements use "sorted (list)" function.
5. for finding minimum value use "min (list)" and for maximum use "max (list)".
6. For sum use function "sum (list)" and for average use the formula "sum (list) / len (list)".
7. print the output
8. End.

Programm:-

```
# Add elements: Add elements to the list.
list = [10, 20]
a=30
list.append(a)
print(list)
# Remove elements: Remove specific elements from the list
```

output:

[10, 20, 30]

[10, 30]

[30]

[5, 8, 9, 15, 30, 89]

The minimum value is : 5

The maximum value is: 89

The sum is : 156

The average is: 26.0.

o/p

```
list.pop(1) # by index value
print (list)
list.remove (10) # by item name
print (list)
# sort elements : sort the list in ascending and descending order.
l= [5,8,9,15,30,89]
Print (sorted (l))
# Find minimum and maximum : find the minimum and maximum
elements in the list
Print ("The maximum value is:", max ())
print ("The minimum value is:", min())
# calculate sum and average
print ("The sum is :", sum (l))
print ("The average is :", ((sum(l) / len (l))))
```

(5) You are tasked with creating a python program that showcases operations on tuples. Tuples are immutable sequences, similar to lists but with the key difference that they cannot be changed after creation. Your program should illustrate the following tuple operations.

Algorithm:-

1. Start
2. To create a tuple use "tuple_name = (values)".
3. To access the elements of a tuple either use the index values (tuple_name, index_value)] or the tuple slicing (tuple_name [start: end])
4. To Concatenate tuples use the operator "+" (tuple1 "+" tuple2)
5. Try to modify the tuple elements by assigning the values directly like; tuple [index] = new_value, will result in an error as it immutable.

output:

(10, 'hello', 3.14, 'world')

10
hello
3.14
world

('hello', 3.14)

(10, 'hello', 3.14)

dp

6. print the output
7. End.

Program:

```
# create a tuple: Define a tuple with elements of different
data types (10, 'hello', 3.14, 'world').
tuple = (10, 'hello', 3.14, 'world')
print (tuple)
# Access elements: Access individual elements and slices of the tuple
for i in tuple:
    print(i)
print (tuple[1:3])
print (tuple[:-1])
# concatenate tuples: combine two tuples to create a new tuple
t2 = (5, 0.5)
t3 = tuple + t2
print (t3)
# Immutable nature: Attempt to modify elements of the tuple
and handle the resulting error.
tuple(3) = "V" # error.
```

(c) You are tasked with creating a python program that showcases operations on dictionaries. Dictionaries in python are unordered collections of items. Each item is a pair consisting of a key and a value.

Algorithm:

1. Start the program.
2. Define a dictionary with key value pairs of different data types

output -

{'name': 'Alice', 'age': 30, 'city': 'New york'}

Alice

so

{'name': 'James', 'age': 30, 'city': 'New york'}

{'name': 'James', 'asc': 30}

key : name

key : age

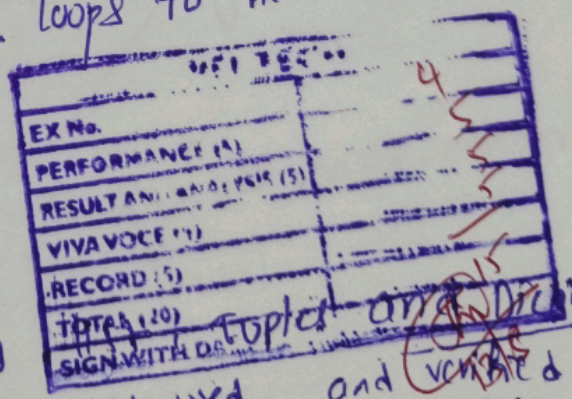
dict.items([('name', 'James'), ('asc', 30)])

o/p

3. Retrieve values from the dictionary using their corresponding keys.
4. modify Dictionary
5. Iterate over dictionary
6. Stop the program.

Program:-

```
# create a dictionary: Define a dictionary with key-value pairs of different data types ({'name': 'Alice', 'age': 30, 'city': 'New York'})
dictionary = {'name': 'Alice', 'age': 30, 'city': 'New York'}
Print (dictionary)
# Access values: Access values using keys.
print (dictionary ['name'])
print (dictionary ['age'])
# Modify Dictionary: update values, add new key-value pairs, and remove existing pairs.
dictionary ['name'] = "James".
print (dictionary.
dictionary.pop ('city')
print (dictionary)
# Iterate over Dictionary: use loops to iterate over keys or the values.
for k in dictionary:
print ("keys:", k)
print (dictionary.items())
```



Result:- Thus, various data types in python programming was used and executed successfully.