Task 4: Use various data types, List, Tuples and Dictionary in Python Programming.

Aim:

To use various data types, List, Tuples and Dicitonary in 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.

1. Add Elements: Add elements to the list.
2. Remove Elements: Remove specific elements from the list.
3. Sort Elements: Sort the list in ascending and descending order
4. Find minimum and maximum: Find the minimum and maximum elements in the list.
5. Calculate Sum and Average: Calculate the Sum and average of the elements in the list.

Algorithm:

1. Start
2. For adding elements to a list first create a list with name "list" and assign the value within [] brackets, inorder to add a new value use the function. append ().
3. For removing a specific elements use" pop (index value)" or "remove (itemname)".
4. For sorting the elements use " sort (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.

Out put:

```
[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
```

Program

```
list = [10,20]
a = 30
list.append(a)
Print (list)
# Remove Elements: Remove specific elements from the list.
list.pop (1) # by index value
Print (list)
list.remove (10) # by itemname
print (list)
# Sort Elements: Sort the list in asending and descending order.
l = [5,8,9,15, 30, 89]
Print (sorted(l))
# find minimum and maximum: Find the minimum and max: elements in the list
print ("The minimum value is:", min(l))
print ("The maximum value is:", max(l))
# Calculate sum and average
Print ("The sum is:", sum(l))
Print ("The average is:", ((sum(l)/len(l))))
```

b. 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:

1. Create a Tuple: Define a tuple with elements of different data types (10, 'hello', 3.14, 'world')
2. Access Elements: Access individual elements and slices of the tuple.
3. Concatenate Tuples: combine two tuples to create a new tuple.

Output:

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

4. Immutable Nature: Attempt to modify elements of the tuple and handle the resulting error.

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 "+" (tuple 1 "+" 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 is immutable.
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) = "pI" # Error
```

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 value. Your program should illustrate the following dictionary operations:

1. Create a Dictionary: Define a dictionary with key-value pairs of different data types. ({'name': 'Alice', 'age': 30, 'city': 'New York'}
2. Access values: Access values using keys.
3. Modify Dictionary: Update values, add new key-value pairs, and remove existing pairs.

Algorithm:

1. Start the program
2. Define a dictionary with key-value pairs of different data types.
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': 'Newyork'}
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')
```

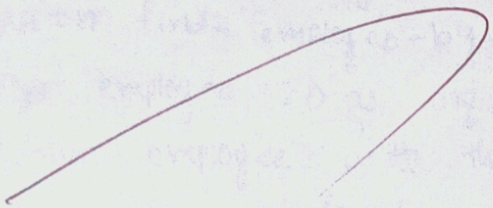
Output:

```
{'name': 'Alice', 'age': 30, 'city': 'New York'}
Alice
30
{'name': 'James', 'age': 30, 'city': 'New York'}
{'name': 'James', 'age': 30}
KEY : name
KEY : age
dict_items([('name', 'James'), ('age', 30)])
```

```
print (dictionary)
```

Iterate Over Dictionary: Use loops to iterate over keys or values.

```
for k in dictionary:
print ("KEY:", k)
print (dictionary.items())
```



VEL TECH	
	A
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	—
TOTAL (20)	15
SIGN WITH DATE	13/8

Result:

Thus, various data types, List, Tuples and Dictionary in Python programming was used and verified successfully.