Python Programming - 2301CS404

Lab - 6

Bhalodiya Charmi

23010101020

448-8th batch

Tuple

01) WAP to find sum of tuple elements.

In [7]:

```python
def sum_of_tuple(tup):
    return sum(tup)
t_tuple= (1,2,3,4,5)
result= sum_of_tuple(t_tuple)
print("sum of tuple elements",result)
```

```
sum of tuple elements 15
```

02) WAP to find Maximum and Minimum K elements in a given tuple.

In [11]:

```python
def find_max_min_k_elements(input_tuple, k):
    if k <= 0 or k > len(input_tuple):
        return "Invalid value of K"
    sorted_elements = sorted(input_tuple)
    min_k_elements = sorted_elements[:k]
    max_k_elements = sorted_elements[-k:]
    return min_k_elements, max_k_elements
input_tuple = (10, 4, 5, 8, 6, 11, 2, 20)
k = 2
min_k, max_k = find_max_min_k_elements(input_tuple, k)
print(f"The {k} smallest elements are: {min_k}")
print(f"The {k} largest elements are: {max_k}")
```

```
The 2 smallest elements are: [2, 4]
The 2 largest elements are: [11, 20]
```

03) WAP to find tuples which have all elements divisible by K from a list of tuples.

In [5]:

```python
def find_divisible_tuples(tuples_list, k):
    return [t for t in tuples_list if all(i % k == 0 for i in t)]
tuples_list = [(1, 2, 3), (4, 8, 12), (5, 10, 15), (7, 14, 21)]
k = 2
result = find_divisible_tuples(tuples_list, k)
print("Tuples with all elements divisible by", k, "are:")
print(result)
```

```
Tuples with all elements divisible by 2 are:
[(4, 8, 12)]
```

04) WAP to create a list of tuples from given list having number and its cube in each tuple.

In [3]:

```python
def create_tuples(numbers):
    return [(n, n**3) for n in numbers]
numbers = [1, 2, 3, 4, 5]
result = create_tuples(numbers)
print("List of tuples with numbers and their cubes:")
print(result)
```

```
List of tuples with numbers and their cubes:
[(1, 1), (2, 8), (3, 27), (4, 64), (5, 125)]
```

05) WAP to find tuples with all positive elements from the given list of tuples.

In [11]:

```
def find_positive_tuples(tuples_list):
    return [t for t in tuples_list if all(isinstance(i, (int, float)) and i > 0 for i in t)]
tuples_list = [(1, 2, 3), (-1, 2, 3), (4, 5, 6), (-7, -8, -9), (10, 20, 30)]
result = find_positive_tuples(tuples_list)
print("Tuples with all positive elements:")
print(result)
```

```
Tuples with all positive elements:
[(1, 2, 3), (4, 5, 6), (10, 20, 30)]
```

06) WAP to add tuple to list and vice – versa.

In [15]:

```
my_list = [1, 2, 3]
print("Original List:", my_list)
my_tuple = (4, 5, 6)
print("Original Tuple:", my_tuple)
my_list.append(my_tuple)
print("List after adding tuple:", my_list)
new_list = list(my_tuple)
print("New list created from tuple:", new_list)
new_tuple = tuple(my_list)
print("New tuple created from list:", new_tuple)
```

```
Original List: [1, 2, 3]
Original Tuple: (4, 5, 6)
List after adding tuple: [1, 2, 3, (4, 5, 6)]
New list created from tuple: [4, 5, 6]
New tuple created from list: (1, 2, 3, (4, 5, 6))
```

07) WAP to remove tuples of length K.

In [17]:

```
def remove_tuples_of_length_k(lst, k):
    return [tup for tup in lst if len(tup) != k]

tuples_list = [(1, 2), (3, 4, 5), (6, 7), (8, 9, 10)]
k = 2
result = remove_tuples_of_length_k(tuples_list, k)
print("After removing tuples of length", k, ":", result)
```

```
After removing tuples of length 2 : [(3, 4, 5), (8, 9, 10)]
```

08) WAP to remove duplicates from tuple.

In [19]:

```
def remove_duplicates_from_tuple(tup):
    return tuple(sorted(set(tup)))

my_tuple = (1, 2, 2, 3, 4, 4, 5)
result = remove_duplicates_from_tuple(my_tuple)
print("Tuple without duplicates:", result)
```

```
Tuple without duplicates: (1, 2, 3, 4, 5)
```

09) WAP to multiply adjacent elements of a tuple and print that resultant tuple.

In [23]:

```
def multiply_adjacent_elements(tup):
    return tuple(tup[i] * tup[i + 1] for i in range(len(tup) - 1))


my_tuple = (1, 2, 3, 4,5)
result = multiply_adjacent_elements(my_tuple)
print("Resultant tuple after multiplying adjacent elements:", result)
```

```
Resultant tuple after multiplying adjacent elements: (2, 6, 12, 20)
```

10) WAP to test if the given tuple is distinct or not.

In [27]:

```
def is_distinct_tuple(tup):
    return len(tup) == len(set(tup))
```

```
my_tuple = (1, 2, 3, 4, 5)
result = is_distinct_tuple(my_tuple)
print("Is the tuple distinct?", result)


my_tuple_with_duplicates = (1, 2, 2, 4)
result = is_distinct_tuple(my_tuple_with_duplicates)
print("Is the tuple distinct?", result)
```

```
Is the tuple distinct? True
Is the tuple distinct? False
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js