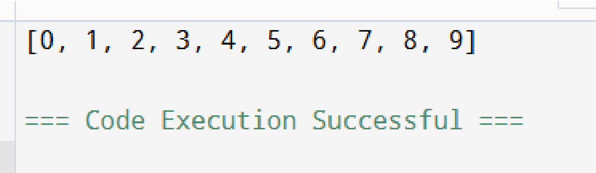
Vaishnavi Ashok Patil
C07
SPP-II

Experiment 02

Title : Implement sorting, merging, and slicing operations on lists and tuples using Python.

1. Create a list of 10 integers and sort it in ascending order.

```
numbers = [5, 2, 9, 1, 7, 6, 3, 8, 4, 0]
numbers.sort()
print(numbers)
```

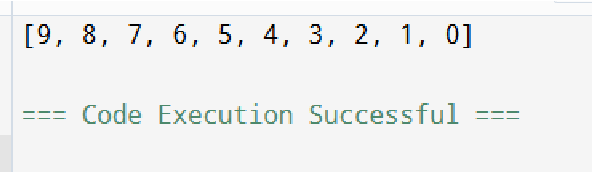


```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

=== Code Execution Successful ===
```

2. Create a list of 10 integers and sort it in descending order.

```
numbers = [5, 2, 9, 1, 7, 6, 3, 8, 4, 0]
numbers.sort(reverse=True)
print(numbers)
```

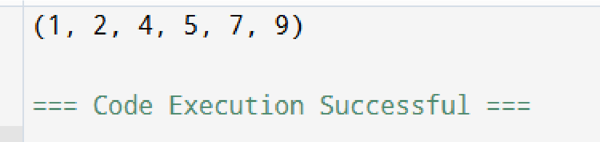


```
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]

=== Code Execution Successful ===
```

3. Create a tuple of 6 numbers and sort it in ascending order.

```
nums = (9, 4, 7, 2, 5, 1)
sorted_tuple = tuple(sorted(nums))
print(sorted_tuple)
```

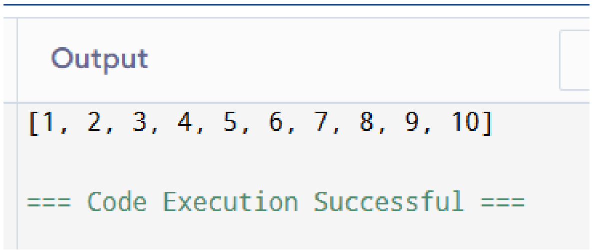


```
(1, 2, 4, 5, 7, 9)

=== Code Execution Successful ===
```

4. Merge two lists of integers and print the merged list.

```
list1 = [1, 2, 3, 4, 5]
list2 = [6, 7, 8, 9, 10]
merged_list = list1 + list2
print(merged_list)
```

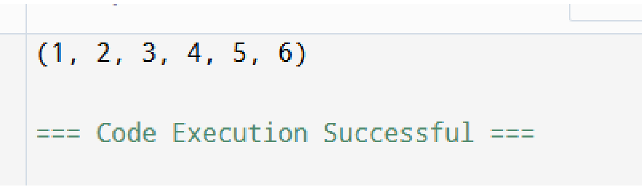


```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

=== Code Execution Successful ===
```

5. Merge two tuples of numbers and print the resulting tuple.

```
tuple1 = (1, 2, 3)
tuple2 = (4, 5, 6)
merged_tuple = tuple1 + tuple2
print(merged_tuple)
```

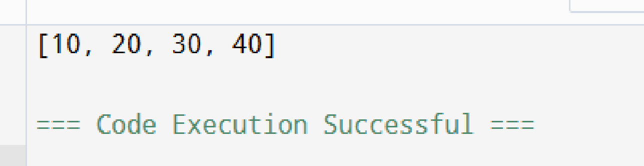


```
(1, 2, 3, 4, 5, 6)

=== Code Execution Successful ===
```

6. Take a list of 8 numbers and print the first 4 elements using slicing.

```
numbers = [10, 20, 30, 40, 50, 60, 70, 80]
print(numbers[:4])
```

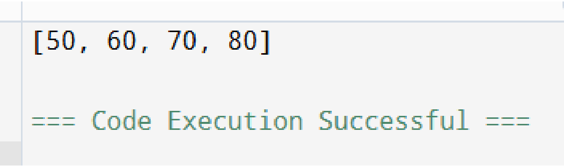


```
[10, 20, 30, 40]

=== Code Execution Successful ===
```

7. Take a list of 8 numbers and print the last 4 elements using slicing.

```
numbers = [10, 20, 30, 40, 50, 60, 70, 80]
print(numbers[-4:])
```



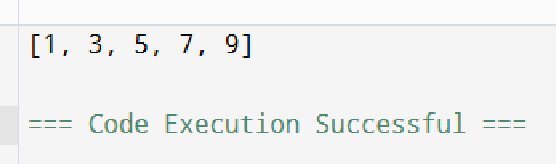
```
[50, 60, 70, 80]

=== Code Execution Successful ===
```

8. From a list of 10 elements, print every second element using slicing.

```
numbers = [1,2,3,4,5,6,7,8,9,10]
print(numbers[::2])
```

Vaishnavi Ashok Patil
C07
SPP-II

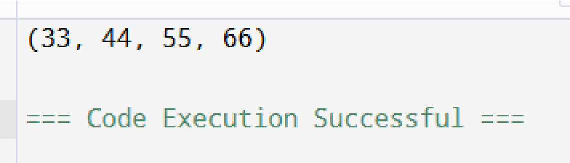


```
[1, 3, 5, 7, 9]

=== Code Execution Successful ===
```

9. From a tuple of 7 elements, print elements from index 2 to 5 using slicing.

```
t = (11, 22, 33, 44, 55, 66, 77)
print(t[2:6])
```

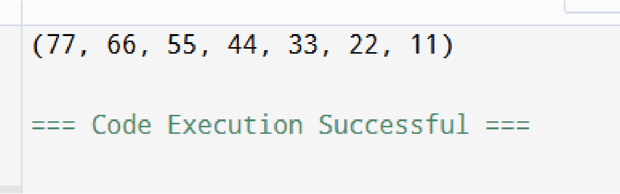


```
(33, 44, 55, 66)

=== Code Execution Successful ===
```

10. From a tuple of 7 elements, print all elements in reverse order using slicing.

```
t = (11, 22, 33, 44, 55, 66, 77)
print(t[::-1])
```

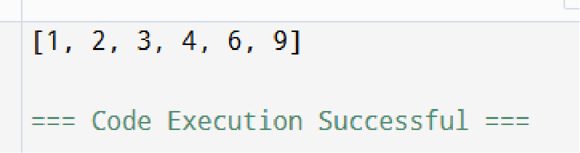


```
(77, 66, 55, 44, 33, 22, 11)

=== Code Execution Successful ===
```

11. Combine two lists, sort the combined list in ascending order, and print the result.

```
list1 = [3, 1, 4]
list2 = [9, 2, 6]
combined = sorted(list1 + list2)
print(combined)
```



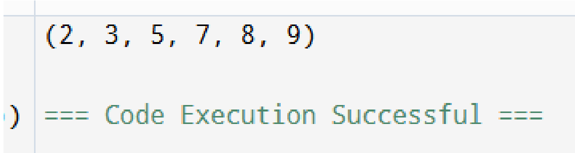
```
[1, 2, 3, 4, 6, 9]

=== Code Execution Successful ===
```

Vaishnavi Ashok Patil
C07
SPP-II

12. Combine two tuples, sort the combined tuple in ascending order, and print the result.

```
tuple1 = (8, 3, 5)
tuple2 = (7, 2, 9)
combined = tuple(sorted(tuple1 + tuple2))
print(combined)
```

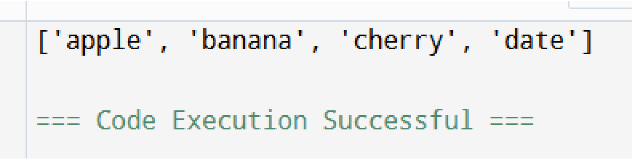


```
(2, 3, 5, 7, 8, 9)

=== Code Execution Successful ===
```

13. Create a list of strings and sort them alphabetically.

```
words = ['banana', 'apple', 'cherry', 'date']
words.sort()
print(words)
```

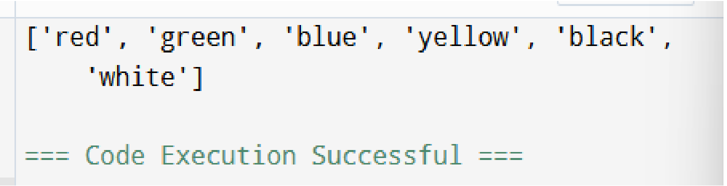


```
['apple', 'banana', 'cherry', 'date']

=== Code Execution Successful ===
```

14. Merge a list of strings with another list of strings and print the merged list.

```
list1 = ['red', 'green', 'blue']
list2 = ['yellow', 'black', 'white']
merged = list1 + list2
print(merged)
```



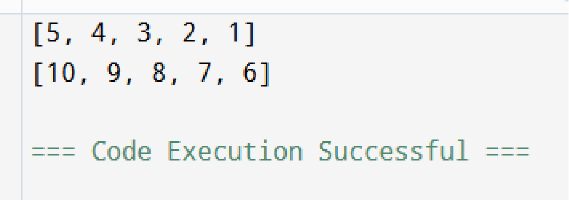
```
['red', 'green', 'blue', 'yellow', 'black',
    'white']

=== Code Execution Successful ===
```

15. Take a list of 10 numbers, split it into two halves, reverse each half separately, and print both halves.

```
numbers = [1,2,3,4,5,6,7,8,9,10]
half1 = numbers[:5][::-1]
half2 = numbers[5:][::-1]
print(half1)
print(half2)
```

Vaishnavi Ashok Patil
C07
SPP-II



```
[5, 4, 3, 2, 1]
[10, 9, 8, 7, 6]

=== Code Execution Successful ===
```