| | | |
|---|---|---|
| **Name of student: Abhay Omprakash Prajapati** | | |
| **Roll no: 41** | **Tutorial No: 3** | |
| **Title of LAB Assignment: To implement Python programs using List, String, Set and Dictionary** | | |
| **DOP: 25-09-2023** | **DOS:02-10-2023** | |
| **CO Mapped:** Co1,Co2 | **PO Mapped:** PO3 ,PO6 | **Signature:** |

**1. Merge two lists and find the second largest element using bubble sort**

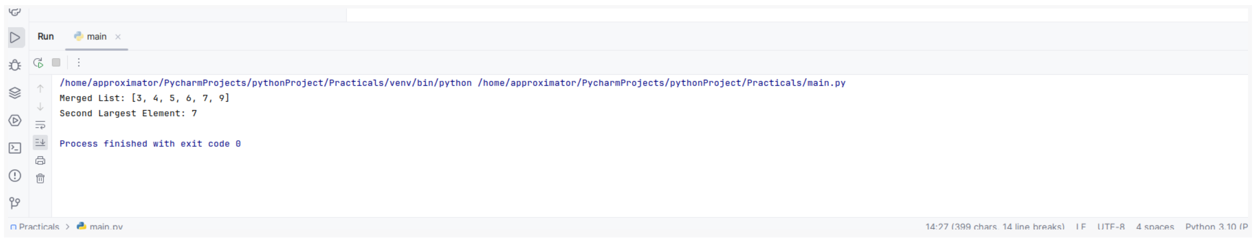
**Aim:** To merge two lists and find the second largest element using bubble sort.
**Theory:** In this task, we'll merge two lists into one and then use the bubble sort algorithm to find the second largest element in the merged list.
**Code:**

```
def bubble_sort(arr):
   n = len(arr)
   for i in range(n):
       for j in range(0, n - i - 1):
           if arr[j] > arr[j + 1]:
               arr[j], arr[j + 1] = arr[j + 1], arr[j]
```

```
list1 = [3, 9, 7]
list2 = [6, 5, 4]
merged_list = list1 + list2
bubble_sort(merged_list)
second_largest = merged_list[-2]
print("Merged List:", merged_list)
print("Second Largest Element:", second_largest)
```

**Conclusion:** We successfully merged two lists and found the second largest element using bubble sort.

**Output:**



```
/home/approximator/PycharmProjects/pythonProject/Practicals/venv/bin/python /home/approximator/PycharmProjects/pythonProject/Practicals/main.py
Merged List: [3, 4, 5, 6, 7, 9]
Second Largest Element: 7

Process finished with exit code 0
```

Run
main
Practicals > main.py
14:27 (399 chars, 14 line breaks)  LF  UTF-8  4 spaces  Python 3.10 (P

## 2. Calculate the number of uppercase, lowercase letters, and digits in a string

**Aim**: To calculate the number of uppercase, lowercase letters, and digits in a given string.

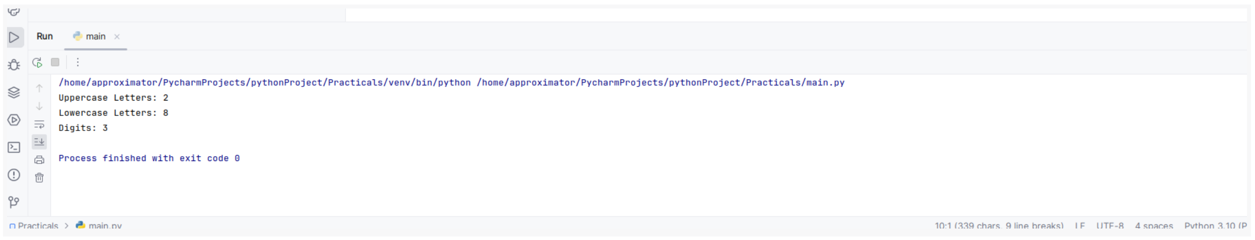
**Theory**: In this task, we'll iterate through the characters in a string and count the number of uppercase letters, lowercase letters, and digits.

**Code:**

```
input_string = "Hello World 123"

upper_count = sum(1 for char in input_string if char.isupper())
lower_count = sum(1 for char in input_string if char.islower())
digit_count = sum(1 for char in input_string if char.isdigit())

print("Uppercase Letters:", upper_count)
print("Lowercase Letters:", lower_count)
print("Digits:", digit_count)
```

**Conclusion:** We successfully counted the number of uppercase, lowercase letters, and digits in the given string.

**Output:**



```
/home/approximator/PycharmProjects/pythonProject/Practicals/venv/bin/python /home/approximator/PycharmProjects/pythonProject/Practicals/main.py
Uppercase Letters: 2
Lowercase Letters: 8
Digits: 3

Process finished with exit code 0
```

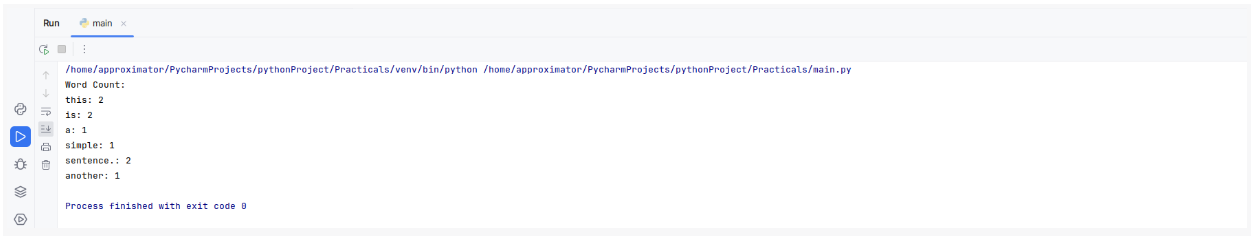
## 3. Count the occurrences of each word in a given string sentence

**Aim**: To count the occurrences of each word in a given string sentence.
**Theory**: In this task, we'll tokenize the string into words, count the occurrences of each word, and store the results in a dictionary.
**Code**:

```
input_sentence = "This is a simple sentence. This is another sentence."

words = input_sentence.split()
word_count = {}

for word in words:
   word = word.lower()
   word_count[word] = word_count.get(word, 0) + 1

print("Word Count:")
for word, count in word_count.items():
   print(f"{word}: {count}")
```

**Conclusion:** We successfully counted the occurrences of each word in the given string sentence.

**Output:**



```
/home/approximator/PycharmProjects/pythonProject/Practicals/venv/bin/python /home/approximator/PycharmProjects/pythonProject/Practicals/main.py
Word Count:
this: 2
is: 2
a: 1
simple: 1
sentence.: 2
another: 1

Process finished with exit code 0
```

## 4. Add a key-value pair to a dictionary, search for a given key, and then delete the key

**Aim:** To add a key-value pair to a dictionary, search for a given key, and then delete the key.

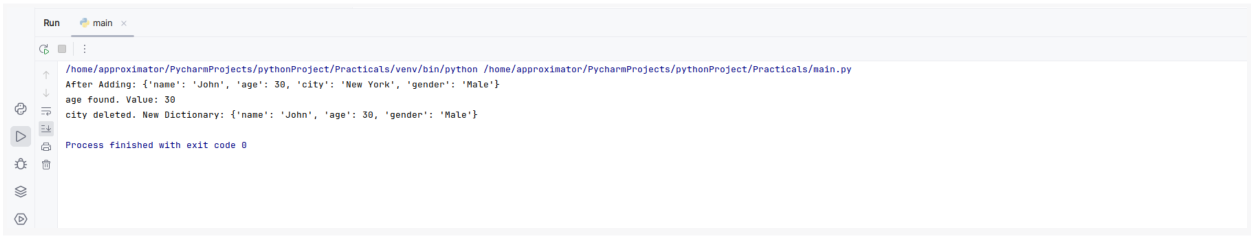
**Theory**: In this task, we'll demonstrate how to add a key-value pair to a dictionary, search for a specific key, and delete that key if found.

**Code:**

```python
sample_dict = {"name": "John", "age": 30, "city": "New York"}

sample_dict["gender"] = "Male"
print("After Adding:", sample_dict)

search_key = "age"
if search_key in sample_dict:
   print(f"{search_key} found. Value: {sample_dict[search_key]}")
else:
   print(f"{search_key} not found.")

delete_key = "city"
if delete_key in sample_dict:
   del sample_dict[delete_key]
   print(f"{delete_key} deleted. New Dictionary: {sample_dict}")
else:
   print(f"{delete_key} not found, no deletion.")
```

**Conclusion**: We successfully added a key-value pair, searched for a key, and deleted a key in the dictionary.

**Output:**



```
/home/approximator/PycharmProjects/pythonProject/Practicals/venv/bin/python /home/approximator/PycharmProjects/pythonProject/Practicals/main.py
After Adding: {'name': 'John', 'age': 30, 'city': 'New York', 'gender': 'Male'}
age found. Value: 30
city deleted. New Dictionary: {'name': 'John', 'age': 30, 'gender': 'Male'}

Process finished with exit code 0
```

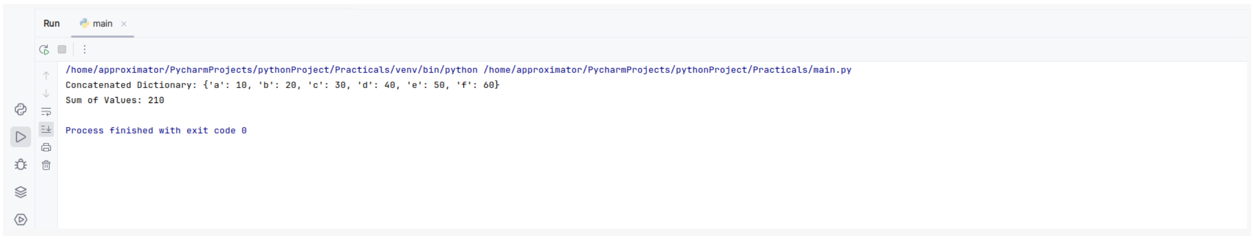
## 5. Concatenate two dictionaries and find the sum of all values in the resulting dictionary

**Aim**: To concatenate two dictionaries and find the sum of all values in the resulting dictionary.

**Theory:** In this task, we'll merge two dictionaries into one, and then calculate the sum of all values in the merged dictionary.

**Code:**
```
dict1 = {'a': 10, 'b': 20, 'c': 30}
dict2 = {'d': 40, 'e': 50, 'f': 60}

concatenated_dict = {**dict1, **dict2}
total_sum = sum(concatenated_dict.values())

print("Concatenated Dictionary:", concatenated_dict)
print("Sum of Values:", total_sum)
```

**Conclusion:** We successfully concatenated two dictionaries and found the sum of all values in the merged dictionary.
**Output:**



```
/home/approximator/PycharmProjects/pythonProject/Practicals/venv/bin/python /home/approximator/PycharmProjects/pythonProject/Practicals/main.py
Concatenated Dictionary: {'a': 10, 'b': 20, 'c': 30, 'd': 40, 'e': 50, 'f': 60}
Sum of Values: 210

Process finished with exit code 0
```

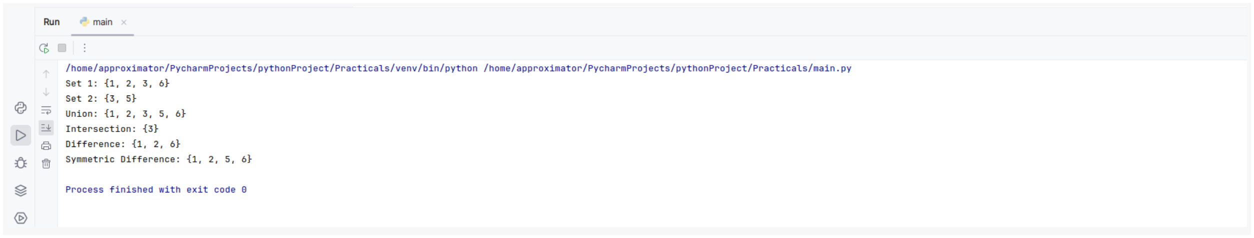
**6. Add and remove elements from a set and perform all set operations, such as union, intersection, difference, and symmetric difference**

**Aim:** To perform various operations on sets, including adding and removing elements, and set operations.
**Theory:** In this task, we'll create two sets, add and remove elements, and perform set operations such as union, intersection, difference, and symmetric difference.
**Code:**

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}

set1.add(6)

set2.remove(4)

union_set = set1 | set2
intersection_set = set1 & set2
difference_set = set1 - set2
symmetric_difference_set = set1 ^ set2

print("Set 1:", set1)
```

```python
print("Set 2:", set2)
print("Union:", union_set)
print("Intersection:", intersection_set)
print("Difference:", difference_set)
print("Symmetric Difference:", symmetric_difference_set)
```

**Conclusion:** We successfully performed various operations on sets and set operations.
**Output:**



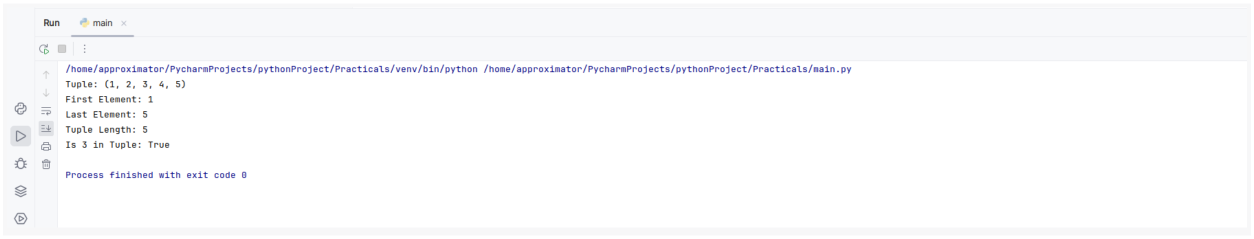
```
/home/approximator/PycharmProjects/pythonProject/Practicals/venv/bin/python /home/approximator/PycharmProjects/pythonProject/Practicals/main.py
Set 1: {1, 2, 3, 6}
Set 2: {3, 5}
Union: {1, 2, 3, 5, 6}
Intersection: {3}
Difference: {1, 2, 6}
Symmetric Difference: {1, 2, 5, 6}

Process finished with exit code 0
```

## 7. Perform different operations on tuples

**Aim:** To perform different operations on tuples.
**Theory:** In this task, we'll demonstrate various operations on tuples, including creating a tuple, accessing elements, finding the length, and checking for the presence of an element.
**Code:**

```python
my_tuple = (1, 2, 3, 4, 5)

first_element = my_tuple[0]
last_element = my_tuple[-1]

tuple_length = len(my_tuple)

is_present = 3 in my_tuple

print("Tuple:", my_tuple)
print("First Element:", first_element)
print("Last Element:", last_element)
print("Tuple Length:", tuple_length)
print("Is 3 in Tuple:", is_present)
```

**Conclusion:** We performed various operations on tuples, including creation, element access, length calculation, and element presence check.
**Output:**



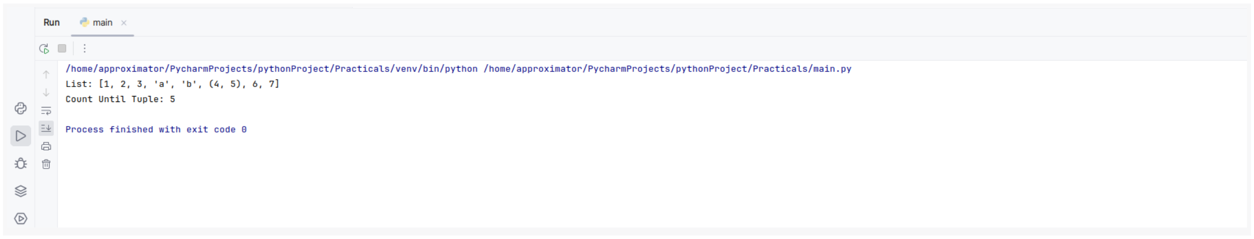
```
/home/approximator/PycharmProjects/pythonProject/Practicals/venv/bin/python /home/approximator/PycharmProjects/pythonProject/Practicals/main.py
Tuple: (1, 2, 3, 4, 5)
First Element: 1
Last Element: 5
Tuple Length: 5
Is 3 in Tuple: True

Process finished with exit code 0
```

## 8. Count the elements in a list until an element is a tuple

**Aim**: To count the elements in a list until an element is a tuple.
**Theory**: In this task, we'll iterate through a list and count the elements until we encounter a tuple. We'll then stop counting.
**Code**:

```
my_list = [1, 2, 3, 'a', 'b', (4, 5), 6, 7]

count_until_tuple = 0
for element in my_list:
    if isinstance(element, tuple):
        break
    count_until_tuple += 1

print("List:", my_list)
print("Count Until Tuple:", count_until_tuple)
```

**Conclusion**: We counted the elements in the list until an element of the tuple type was encountered.
**Output:**



```
/home/approximator/PycharmProjects/pythonProject/Practicals/venv/bin/python /home/approximator/PycharmProjects/pythonProject/Practicals/main.py
List: [1, 2, 3, 'a', 'b', (4, 5), 6, 7]
Count Until Tuple: 5

Process finished with exit code 0
```