

## Assessment-4

**1. Write a Python code that takes a list of numbers as input and removes all occurrences of a specified number using the remove() method. Return the modified list.**

```
def remove_all_occurrences(numbers, target):
    while target in numbers:
        numbers.remove(target)
    return numbers

numbers = list(map(int, input("Enter the list of numbers separated by commas: ").split(",")))
target = int(input("Enter the number to remove: "))
modified_list = remove_all_occurrences(numbers, target)
print("Modified list:", modified_list)
```

**2. Write a Python code that takes a dictionary representing student names and their scores and a new student-score pair. If the student already exists, update the score; if not, add the new student. Use the update() method.**

**Example Input: {'John': 85, 'Emma': 92}, {'Lucas': 78}**

```
stud1 = {
    "Aman": 90,
    "Rahul": 45,
    "Ritesh": 70,
    "Arun": 80,
}

stud2 = {
    "Saiyam": 60,
    "Aman": 50,
}

stud1.update((stud2))
print(stud1)
```

**3. Write a Python code that takes a dictionary and returns the sum of all its values.**

```
dict1 = {
```

```
"A":10,  
"B":20,  
"C":30,  
}
```

```
a = dict1.values()  
print(a)  
sum =sum(a)  
print(sum)
```

**4. Write a Python code that converts a tuple of numbers to a list, appends a given number to the list, and then converts it back to a tuple.**

```
t1 = (10,20,30,40,50)
```

```
t2 = list(t1)  
t2.append(60)  
t3 = tuple(t2)  
print(t3)
```

**5. Write a Python code that removes the minimum and maximum values from a set.**

```
s1 = {10,20,30,56,40}
```

```
min_value = float('inf')  
max_value = float('-inf')
```

```
for i in s1:  
    if i<min_value:  
        min_value = i  
    if i > max_value:  
        max_value = i
```

```
print(min_value,max_value)
```

```
s1.remove(min_value)  
s1.remove(max_value)  
print(s1)
```

## 6. Return the symmetric difference of two sets.

```
set1 = set(map(int,input("Enter first elements of sets:").split(",")))
set2 = set(map(int,input("Enter second elements of sets:").split(",")))
print(set1 ^ set2)          #first method
print(set1.symmetric_difference(set2))    #second method
```

## 7. Write a Python function that finds and returns the index of all occurrences of a specified element in a list using the index() method. If the element is not found, return None.

```
my_list = list(map(int, input("Enter the elements of the list separated by commas: ").split(",")))
target = int(input("Enter the element to find: "))
indices = []
```

```
for i in range(len(my_list)):
    if my_list[i] == target:
        indices.append(i)

if indices:
    print(f"Element found at indices: {indices}")
else:
    print("Element not found.")
```

## 8. Write a Python function that counts the occurrences of a specific element in a tuple

```
def count_occurrences(n):
    d = {}
    for i in n:
        if i in d:
            d[i] += 1
        else:
            d[i] = 1
    return d
```

```
n = tuple(map(int,input("Enter the elements of tuples: ").split(",")))
result = count_occurrences(n)
```

```
print(result)
```

**9. Write a Python function that takes two sets and returns the difference between the two sets**

```
my_set1 = set(map(int,input("Enter the elements of set1: ").split(",")))
my_set2 = set(map(int,input("Enter the elements of set2: ").split(",")))

print(my_set1 - my_set2)
print(my_set1.difference(my_set2))
```

**10. Write a Python function that checks whether a given key exists in a dictionary. If the key is present, return its value; otherwise, return 'Key not found'.**

```
def check_key_exists(dictionary, key):
    if key in dictionary:
        return dictionary[key]
    else:
        return 'Key not found'

dict1 = {
    "a": 10,
    "b": 20,
    "c": 10,
    "d": 10,
}

key_to_check = input("Enter the key to check: ")
result = check_key_exists(dict1, key_to_check)
print(result)
```

**11. Write a Python function that sorts a list of strings based on the length of the strings**

```
def sort_by_length(strings):
    return sorted(strings, key=len)

strings = input("Enter a list of strings separated by commas: ").split(",")
```

```
sorted_strings = sort_by_length(strings)
print("Strings sorted by length:", sorted_strings)
```

**12. Write a Python function that finds all keys in a dictionary that have a specific value**

```
def find_keys_by_value(dictionary, target_value):
    result = []
    for key in dictionary:
        if dictionary[key] == target_value:
            result.append(key)
    return result

dict1 = {
    "a": 10,
    "b": 20,
    "c": 10,
    "d": 30,
}

value_to_find = int(input("Enter the value to find keys for: "))
keys = find_keys_by_value(dict1, value_to_find)

if keys:
    print(f"Keys with value {value_to_find}: {keys}")
else:
    print(f"No keys found with value {value_to_find}.")
```