

**1. Write a Python Program to Calculate the Average of Numbers in a given List.**

```
def calculate_average(numbers):
    if not numbers:
        return 0
    total = sum(numbers)
    count = len(numbers)
    average = total / count
    return average
my_list = [10, 20, 30, 40, 50]
result = calculate_average(my_list)
print("The average of the numbers is:", result)
```

**2. Write a program which accepts 6 integer values and prints “DUPLICATES” if any of the values entered are duplicates otherwise it prints “ALL UNIQUE”.**

**Example:** Let 6 integers are (32, 10, 45, 90, 45, 6) then output “DUPLICATES” to be printed.

```
def check_duplicates(numbers):
    seen = set()
    for num in numbers:
        if num in seen:
            return True
        seen.add(num)
    return False
numbers = []
for i in range(6):
    number = int(input("Enter integer {}: ".format(i+1)))
    numbers.append(number)
if check_duplicates(numbers):
    print("DUPLICATES")
else:
    print("ALL UNIQUE")
```

**3. Write a Python program to add and remove operation on set.**

```
def add_and_remove_set_elements(set_a):
    set_a.add(10)
    set_a.add(20)
    set_a.add(30)
    set_a.remove(20)
    set_a.discard(40)
    return set_a
my_set = {1, 2, 3}
modified_set = add_and_remove_set_elements(my_set)
print("Modified set:", modified_set)
```

**4. Write a Python program to find maximum and the minimum value in a set.**

```
def find_max_min(my_set):
    if not my_set:
        return None, None # Handle empty set
    max_value = max(my_set)
    min_value = min(my_set)
    return max_value, min_value
my_set = {10, 20, 5, 30, 15}
max_val, min_val = find_max_min(my_set)
print("Maximum value:", max_val)
print("Minimum value:", min_val)
```

**5. Write a python program to create an array of 'n' integers and display the array elements. Access individual elements through indexes.**

```
import array as arr
def create_and_display_array(n):
    my_array = arr.array('i', [])
    for i in range(n):
        element = int(input("Enter element {}: ".format(i+1)))
        my_array.append(element)
    print("Array elements:")
    for i in range(n):
        print("Element {}: {}".format(i+1, my_array[i]))
n = int(input("Enter the number of elements: "))
create_and_display_array(n)
```

**6. Write a python program to get the number of occurrences of specified elements in an array.**

```
def count_occurrences(array, element):
    count = 0
    for num in array:
        if num == element:
            count += 1
    return count
my_array = [1, 2, 3, 2, 1, 4, 2]
element_to_count = 2
occurrences = count_occurrences(my_array, element_to_count)
print("Number of occurrences of", element_to_count, ":", occurrences)
```

**7. Write a python program to reverse the order of the items in the array.**

```
def reverse_array(array):
    n = len(array)
    for i in range(n // 2):
        array[i], array[n - i - 1] = array[n - i - 1], array[i]
my_array = [1, 2, 3, 4, 5]
print("Original array:", my_array)
reverse_array(my_array)
print("Reversed array:", my_array)
```

**8. Write a python program to find sum of all the elements in a list.**

```
def sum_of_elements(numbers):
    total = 0
    for num in numbers:
        total += num
    return total
my_list = [10, 20, 30, 40, 50]
result = sum_of_elements(my_list)
print("The sum of the numbers is:", result)
```

**9. Write a python function to calculate the factorial of a number. The function accepts the number as an argument.**

```
def factorial(n):
    if n < 0:
        raise ValueError("Factorial is not defined for negative numbers.")
    elif n == 0:
        return 1
    else:
        return n * factorial(n - 1)
number = 5
result = factorial(number)
print("The factorial of", number, "is", result)
```

**10. Write a program to generate Fibonacci numbers using function.**

```
def fibonacci(n):
    if n <= 0:
        return []
    elif n == 1:
        return [0]
    else:
        fib_sequence = [0, 1]
        for i in range(2, n):
            next_fib = fib_sequence[i-1] + fib_sequence[i-2]
            fib_sequence.append(next_fib)
        return fib_sequence
n = 10
result = fibonacci(n)
print("Fibonacci sequence:", result)
```

**11. Write a Python script to generate and print a dictionary that contains a number (Between 1 and n) in the form (x : x\*x). Sample Dictionary (n = 5)**

Expected Output: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}.

```
def generate_square_dict(n):
    square_dict = {}
    for x in range(1, n + 1):
        square_dict[x] = x * x
    return square_dict
n = int(input("Enter the upper limit: "))
result_dict = generate_square_dict(n)
print(result_dict)
```

**12. Write a Python script to sort (ascending and descending) a dictionary by value.**

```
def sort_dict_by_value(dictionary, ascending=True):
    sorted_items = sorted(dictionary.items(), key=lambda x: x[1], reverse=not ascending)
    return sorted_items

my_dict = {'a': 3, 'b': 1, 'c': 5, 'd': 2}
sorted_asc = sort_dict_by_value(my_dict)
print("Ascending order:", sorted_asc)
sorted_desc = sort_dict_by_value(my_dict, False)
print("Descending order:", sorted_desc)
```

**13. Write a Python program to combine two dictionary adding values for common keys. Sample Dictionary:**

**d1={ 'a':100, 'b':200, 'c':300}**

**d2={ 'a':300, 'b':200, 'd':400}**

**Sample output: Counter ({ 'a': 400, 'b': 400, 'd': 400, 'c': 300})**

```
from collections import Counter
def combine_dictionaries(d1, d2):
    combined_dict = Counter(d1) + Counter(d2)
    return combined_dict

d1 = {'a': 100, 'b': 200, 'c': 300}
d2 = {'a': 300, 'b': 200, 'd': 400}
result = combine_dictionaries(d1, d2)
print(result)
```

**14. Write a Python program to create a list of tuples with the first element as the number and second element as the square of the number, also display original list in reverse.**

```
def create_square_tuples(n):
    square_tuples = [(x, x**2) for x in range(1, n+1)]
    return square_tuples

n = int(input("Enter the upper limit: "))
result_list = create_square_tuples(n)
print("Original list in reverse order:")
print(result_list[::-1])
```

**15. Write a python code to copy element 44 and 55 from the following tuple into a new tuple tuple1 = (11, 22, 33, 44, 55, 66), also display the same tuple in reverse order.**

```
tuple1 = (11, 22, 33, 44, 55, 66)
new_tuple = tuple1[3:5]
print("New tuple:", new_tuple)
reversed_tuple = tuple1[::-1]
print("Reversed tuple:", reversed_tuple)
```

**16. Write a Pyt progr to get the 5th element from front and 5th element from last of a tuple.**

```
tuple1 = (11, 22, 33, 44, 55, 66)
fifth_from_front = tuple1[4]
fifth_from_last = tuple1[-5]
print("5th element from the front:", fifth_from_front)
print("5th element from the last:", fifth_from_last)
```

**17. Write a program to display following pattern.**

1

2 3

4 5 6

7 8 9 10

```
def print_pattern(n):
    num = 1
    for i in range(1, n+1):
        for j in range(1, i+1):
            print(num, end=" ")
            num += 1
        print()
n = int(input("Enter the number of rows: "))
print_pattern(n)
```

**18. Write a Python program to find repeated items in a tuple.**

```
def find_repeated_items(tuple1):
    seen = set()
    repeated = set()
    for item in tuple1:
        if item in seen:
            repeated.add(item)
        else:
            seen.add(item)
    return repeated
my_tuple = (1, 2, 3, 2, 1, 4, 5, 3)
repeated_items = find_repeated_items(my_tuple)
print("Repeated items:", repeated_items)
```