Python Programming and Data Structure Programs (10 Marks programs)

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

```
def calculate_average(numbers):
    return sum(numbers) / len(numbers)

numbers = [10, 20, 30, 40, 50]
average = calculate_average(numbers)
print(f"Average: {average}")
```

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(values):
    if len(values) != len(set(values)):
        print("DUPLICATES")
    else:
        print("ALL UNIQUE")

numbers = [32, 10, 45, 90, 45, 6]
check_duplicates(numbers)
```

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

```
my_set = {1, 2, 3}
# Add an element
my_set.add(4)
print("Set after adding 4:", my_set)
# Remove an element
my_set.remove(2)
print("Set after removing 2:", my_set)
```

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

```
my_set = {1, 2, 3, 4, 5}
max_value = max(my_set)
min_value = min(my_set)
print(f"Maximum value: {max_value}, Minimum value: {min_value}")
```

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

```
n = 5
arr = [i for i in range(n)]
print("Array:", arr)
for i in range(n):
    print(f"Element at index {i}: {arr[i]}")
```

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

```
arr = [1, 2, 2, 3, 4, 2]

element = 2

count = arr.count(element)

print(f"Number of occurrences of {element}: {count}")
```

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

```
arr = [1, 2, 3, 4, 5]
arr.reverse()
print("Reversed Array:", arr)
```

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

```
numbers = [1, 2, 3, 4, 5]
sum_numbers = sum(numbers)
print(f"Sum of all elements: {sum_numbers}")
```

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:
        return 1
    else:
```

```
    return n * factorial(n - 1)

number = 5
print(f"Factorial of {number}: {factorial(number)}")
```

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

```
def fibonacci(n):
    fib_sequence = [0, 1]
    while len(fib_sequence) < n:
        fib_sequence.append(fib_sequence[-1] + fib_sequence[-2])
    return fib_sequence

n = 10
print(f"First {n} Fibonacci numbers: {fibonacci(n)}")
```

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):
    return {x: x**2 for x in range(1, n+1)}

n = 5
print(generate_square_dict(n))
```

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

```
my_dict = {'a': 100, 'b': 200, 'c': 300}

# Sort Ascending
sorted_dict_asc = dict(sorted(my_dict.items(), key=lambda item: item[1]))
print(f"Sorted in Ascending Order: {sorted_dict_asc}")

# Sort Descending
sorted_dict_desc = dict(sorted(my_dict.items(), key=lambda item: item[1], reverse=True))
print(f"Sorted in Descending Order: {sorted_dict_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})

```
d1 = {'a': 100, 'b': 200, 'c': 300}
d2 = {'a': 300, 'b': 200, 'd': 400}
from collections import Counter

combined = Counter(d1) + Counter(d2)
print(f"Combined Dictionary: {combined}")
```

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.

```
numbers = [1, 2, 3, 4, 5]
squared_list = [(x, x**2) for x in numbers]
print("Original List:", squared_list)
squared_list.reverse()
print("Reversed List:", squared_list)
```

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)
print("Reversed New Tuple:", new_tuple[::-1])
```

16. Write a Python program to get the 5th element from front and 5th element from last of a tuple.

```
tuple1 = (11, 22, 33, 44, 55, 66, 77, 88, 99, 100)

fifth_from_front = tuple1[4]

fifth_from_last = tuple1[-5]
```

```
print(f"5th element from front: {fifth_from_front}, 5th element from last:
{fifth_from_last}")
```

17. Write a program to display following pattern.

```
1
2 3
4 5 6
7 8 9 10
```

```
n = 4
num = 1
for i in range(1, n + 1):
    for j in range(i):
        print(num, end=" ")
        num += 1
    print()
```

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

```
tuple1 = (1, 2, 3, 2, 4, 5, 3, 6)
repeated = [item for item in set(tuple1) if tuple1.count(item) > 1]
print("Repeated Items:", repeated)
```