1. Program to create a list and print its index and negative index: my_list = ['apple', 'banana', 'cherry', 'date', 'elderberry']

for index, item in enumerate(my_list): print(f"Index: {index}, Item: {item}, Negative Index: {-len(my_list) + index}")

2. Program to demonstrate various functions in a list: my_list = [10, 20, 30, 40, 50, 20, 30]

print(f"Range of list item: {range(len(my_list))}") print(f"Reverse of list item: {my_list[::-1]}") print(f"Count of 20 in list: {my_list.count(20)}")

3. Program to demonstrate more functions in a list: my_list = [1, 2, 3, 4, 5]

print(f"Pop element at index 2: {my_list.pop(2)}") print(f"Replicated list: {my_list * 2}") my_list.sort() print(f"Sorted list: {my_list}")

4. Program to demonstrate membership test using 'in' and 'not in' keywords: my_list = [10, 20, 30, 40, 50]

print(f"Is 30 in list? {'Yes' if 30 in my_list else 'No'}") print(f"Is 60 not in list? {'Yes' if 60 not in my_list else 'No'}")

5. Program to create a tuple with 5 items: my_tuple = (1, 2, 3, 4, 5) print(my_tuple)

6. Compare two tuples, Length of the tuple, Maximum and minimum tuple1 = (1, 2, 3, 4, 5) tuple2 = (5, 6, 7, 8, 9) print(f"Comparison of tuples: {tuple1 == tuple2}") print(f"Length of tuple1: {len(tuple1)}") print(f"Maximum of tuple1: {max(tuple1)}, Minimum of tuple1: {min(tuple1)}")

7. Replicating Tuple, Slicing Tuple, Search an item with its index tuple3 = tuple1 * 2 print(f"Replicated tuple1: {tuple3}") print(f"Sliced tuple1: {tuple1[1:3]}") print(f"Index of item 3 in tuple1: {tuple1.index(3)}")

8. Max and min value from list, Length of the list, Sorting a list, Sum of list item my_list = [10, 5, 20, 15, 25] print(f"Maximum value in list: {max(my_list)}, Minimum value in list: {min(my_list)}") print(f"Length of list: {len(my_list)}") my_list.sort() print(f"Sorted list: {my_list}") print(f"Sum of list items: {sum(my_list)}")

9. Basic tuple operations: Repetition, Membership, Iteration tuple4 = (1, 2, 3) print(f"Repetition of tuple4: {tuple4 * 3}") print(f"Membership test: {'Yes' if 2 in tuple4 else 'No'}") for item in tuple4: print(item)

10. All, Any, Enumerate my_bools = [True, False, True] print(f"All true? {all(my_bools)}, Any true? {any(my_bools)}") for index, item in enumerate(my_list): print(f"Index: {index}, Item: {item}")

11. Create a dictionary with 5 items and print items with index my_dict = {'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5} for index, (key, value) in enumerate(my_dict.items()): print(f"Index: {index}, Key: {key}, Value: {value}")

12. Return all keys, values, items of a dictionary print(f"All keys: {list(my_dict.keys())}") print(f"All values: {list(my_dict.values())}") print(f"All items: {list(my_dict.items())}")

13. Delete an item, Compare two dictionaries, Remove all items del my_dict['a'] dict2 = {'b': 2, 'c': 3} print(f"Comparison of dictionaries: {my_dict == dict2}") my_dict.clear()

14. Copy dictionary, Update an item, Return the value of given key my_dict = {'a': 1, 'b': 2} my_dict_copy = my_dict.copy() my_dict.update({'c': 3}) print(f"Updated dictionary: {my_dict}") print(f"Value of key 'a': {my_dict.get('a')}")

15. Define a class Teacher class Teacher: def ___init___(self): self.s_name = "" self.subject = "" self.basic = 0 self.da = 0 self.hra = 0 self.salary = 0

def get_data(self): self.s_name = input("Enter Student Name: ") self.subject = input("Enter Subject: ") self.basic = float(input("Enter Basic Salary: ")) self.da = float(input("Enter DA: ")) self.hra = float(input("Enter HRA: ")) self.calculate_salary()

def calculate_salary(self): self.salary = self.___Basic + self.___DA + self.___HRA

def put_data(self): print(f"Student Name: {self.___Stu_Name}") print(f"Subject: {self.___Subject}") print(f"Basic Salary: {self.___Basic}") print(f"DA: {self.___DA}") print(f"HRA: {self.___HRA}") print(f"Total Salary: {self.___Salary}")

16. Define a class Shape class Shape: def ___init___(self, x, y): self.x = x self.y = y

def area(self): return self.x * self.y

def perimeter(self): return 2 * (self.x + self.y)

def print_doc_string(self): print(self.___doc___)

17. Create a class Employee class Employee: def ___init___(self): self.Ename = "" self.Esalary = 0 self.Eage = 0

def get_data(self): self.Ename = input("Enter Employee Name: ") self.Esalary = float(input("Enter Employee Salary: ")) self.Eage = int(input("Enter Employee Age: "))

def put_data(self): print(f"Employee Name: {self.___Ename}") print(f"Employee Salary: {self.___Esalary}") print(f"Employee Age: {self.___Eage}")

Display details of 3 manager employees for _ in range(3): emp = Employee() emp.get_data() emp.put_data() print()