Software Development – List, Tuple

Python programming exercises

1. Write a program that finds the maximum value of the given list, assuming that the list contains at least one element.

   Try your program with the following array

2	4	7	4	23	5	1	4	8	9

```
# Given list
numbers = [2, 4, 7, 4, 23, 5, 1, 4, 8, 9]

# Find the maximum value
max_value = max(numbers)

# Output the result
print("The maximum value in the list is:”, max_value)
```

2. Write a program that calculates the average value of the given list.

   Try your program with the following list

4	7	1	5	11	53	12	46	84	23

```
# Given list
numbers = [4, 7, 1, 5, 11, 53, 12, 46, 84, 23]

# Calculate the average value
average_value = sum(numbers) / len(numbers)

# Output the result
print("The average value of the list is:", average_value)
```

3. Write a program that prints the given list of integers in reverse order.

   Try your program with the following list

2	6	7	45	23	53	14	45	89	5

```
# Given list
numbers = [2, 6, 7, 45, 23, 53, 14, 45, 89, 5]

# Print the list in reverse order
reverse_order = list(reversed(numbers))
print("Original list:", numbers)
print("List in reverse order:", reverse_order)
```

```
Original list: [2, 6, 7, 45, 23, 53, 14, 45, 89, 5]
List in reverse order: [5, 89, 45, 14, 53, 23, 45, 7, 6, 2]
```

4. Write a program that accepts two lists of integers and prints true if each element in the first list is less than the element at the same index in the second list. Your program should print false if the lists are not the same length.

```python
# Input two lists of integers
list1 = [2, 6, 7, 4, 9]
list2 = [5, 8, 10, 6, 12]

# Check if the lists are of the same length
if len(list1) == len(list2):
    # Compare elements at the same index
    result = all(list1[i] < list2[i] for i in range(len(list1)))
    print(result)
else:
    print(False)
```

5. Write a program that accepts a list of integers and two indexes and swaps the elements at those indexes

```python
def swap_elements(lst, index1, index2):
    # Check if the indexes are valid
    if 0 <= index1 < len(lst) and 0 <= index2 < len(lst):
        # Swap elements at the specified indexes
        lst[index1], lst[index2] = lst[index2], lst[index1]
        return lst
    else:
        print("Invalid indexes. Please ensure indexes are within the range of 
the list.")
        return None

# Example usage:
my_list = [2, 6, 7, 4, 9]

# Swap elements at indexes 1 and 3
result = swap_elements(my_list, 1, 3)

# Print the result
print("Original list:", my_list)
```

6. Write a program that accepts two lists of integers and prints a new list containing all elements of the first list followed by all elements of the second.

```python
def concatenate_lists(list1, list2):
    # Concatenate the two lists
    concatenated_list = list1 + list2
    return concatenated_list

# Example usage:
first_list = [2, 6, 7]
second_list = [4, 9, 1]

# Concatenate the lists
result = concatenate_lists(first_list, second_list)
```

```python
# Print the result
print("First list:", first_list)
print("Second list:", second_list)
print("Concatenated list:", result)
```

7. Write a program that accepts a list of integers and an integer value as its parameters and prints the last index at which the value occurs in the list. The program should print –1 if the value is not found. For example, in the list [74, 85, 102, 99, 101, 85, 56], the last index of the value 85 is 5.

```python
def last_index_of_value(lst, value):
    # Check if the value is in the list
    if value in lst:
        # Find the last index of the value
        last_index = len(lst) - 1 - lst[::-1].index(value)
        return last_index
    else:
        return -1

# Example usage:
my_list = [74, 85, 102, 99, 101, 85, 56]
search_value = 85

# Find the last index of the value
result = last_index_of_value(my_list, search_value)
```

8. Write a program that prints the range of values in a list of integers. The range is defined as 1 more than the difference between the maximum and minimum values in the list. For example, if a list contains the values [36, 12, 25, 19, 46, 31, 22], the program should return `35`. You may assume that the list has at least one element.

```python
def calculate_range(lst):
    # Calculate the range
    value_range = max(lst) - min(lst) + 1
    return value_range

# Example usage:
my_list = [36, 12, 25, 19, 46, 31, 22]

# Calculate and print the range
result = calculate_range(my_list)
print("The range of values in the list is:", result)
```

9. Write a program that accepts a list of integers, a minimum value, and a maximum value and prints the count of how many elements from the list fall between the minimum and maximum (inclusive). For example, in the list [14, 1, 22, 17, 36, 7, -43, 5], for minimum value 4 and maximum value 17, there are four elements whose values fall between 4 and 17.

```
def count_elements_in_range(lst, min_value, max_value):
    # Count elements in the specified range
    count = sum(min_value <= element <= max_value for element in lst)
    return count

# Example usage:
my_list = [14, 1, 22, 17, 36, 7, -43, 5]
min_value = 4
max_value = 17

# Count and print the elements in the range
result = count_elements_in_range(my_list, min_value, max_value)

print(f"There are {result} elements in the range [{min_value}, {max_value}],”)
```

10. Write a program that accepts a list of real numbers and prints true if the list is in sorted (nondecreasing) order and false otherwise. For example, if lists named list1 and list2 store [16.1, 12.3, 22.2, 14.4] and [1.5, 4.3, 7.0, 19.5, 25.1, 46.2] respectively, the program should print false for list1 and true for list2 respectively. Assume the list has at least one element. A one-element list is sorted.

```
def is_sorted(lst):
    # Check if the list is sorted in nondecreasing order
    sorted_order = all(lst[i] <= lst[i + 1] for i in range(len(lst) - 1))
    return sorted_order

# Example usage:
list1 = [16.1, 12.3, 22.2, 14.4]
list2 = [1.5, 4.3, 7.0, 19.5, 25.1, 46.2]

# Check and print if the lists are sorted
result1 = is_sorted(list1)
result2 = is_sorted(list2)

print("Is list1 sorted?", result1)
print("Is list2 sorted?", result2)
```