List Tasks

1. **Count Occurrences**: Given a list and an element, find how many times the element appears in the list.
2. **Sum of Elements**: Given a list of numbers, calculate the total of all the elements.
3. **Max Element**: From a given list, determine the largest element.
4. **Min Element**: From a given list, determine the smallest element.
5. **Check Element**: Given a list and an element, check if the element is present in the list.
6. **First Element**: Access the first element of a list, considering what to return if the list is empty.
7. **Last Element**: Access the last element of a list, considering what to return if the list is empty.
8. **Slice List**: Create a new list that contains only the first three elements of the original list.
9. **Reverse List**: Create a new list that contains the elements of the original list in reverse order.
10. **Sort List**: Create a new list that contains the elements of the original list in sorted order.
11. **Remove Duplicates**: Given a list, create a new list that contains only unique elements from the original list.
12. **Insert Element**: Given a list and an element, insert the element at a specified index.
13. **Index of Element**: Given a list and an element, find the index of the first occurrence of the element.
14. **Check for Empty List**: Determine if a list is empty and return a boolean.
15. **Count Even Numbers**: Given a list of integers, count how many of them are even.
16. **Count Odd Numbers**: Given a list of integers, count how many of them are odd.
17. **Concatenate Lists**: Given two lists, create a new list that combines both lists.
18. **Find Sublist**: Given a list and a sublist, check if the sublist exists within the list.
19. **Replace Element**: Given a list, replace the first occurrence of a specified element with another element.
20. **Find Second Largest**: From a given list, find the second largest element.
21. **Find Second Smallest**: From a given list, find the second smallest element.
22. **Filter Even Numbers**: Given a list of integers, create a new list that contains only the even numbers.
23. **Filter Odd Numbers**: Given a list of integers, create a new list that contains only the odd numbers.
24. **List Length**: Determine the number of elements in the list.
25. **Create a Copy**: Create a new list that is a copy of the original list.
26. **Get Middle Element**: Given a list, find the middle element. If the list has an even number of elements, return the two middle elements.
27. **Find Maximum of Sublist**: Given a list, find the maximum element of a specified sublist.
28. **Find Minimum of Sublist**: Given a list, find the minimum element of a specified sublist.

29. **Remove Element by Index**: Given a list and an index, remove the element at that index if it exists.
30. **Check if List is Sorted**: Determine if the list is sorted in ascending order and return a boolean.
31. **Repeat Elements**: Given a list and a number, create a new list where each element is repeated that number of times.
32. **Merge and Sort**: Given two lists, create a new sorted list that merges both lists.
33. **Find All Indices**: Given a list and an element, find all the indices of that element in the list.
34. **Rotate List**: Given a list, create a new list that is a rotated version of the original list (shift elements to the right).
35. **Create Range List**: Create a list of numbers in a specified range (e.g., from 1 to 10).
36. **Sum of Positive Numbers**: Given a list of numbers, calculate the sum of all positive numbers.
37. **Sum of Negative Numbers**: Given a list of numbers, calculate the sum of all negative numbers.
38. **Check Palindrome**: Given a list, check if the list is a palindrome (reads the same forwards and backwards).
39. **Create Nested List**: Create a new list that contains sublists, where each sublist contains a specified number of elements from the original list.
40. **Get Unique Elements in Order**: Given a list, create a new list that contains unique elements while maintaining the original order.

Tuple Tasks

1. **Count Occurrences**: Given a tuple and an element, find how many times the element appears in the tuple.
2. **Max Element**: From a given tuple, determine the largest element.
3. **Min Element**: From a given tuple, determine the smallest element.
4. **Check Element**: Given a tuple and an element, check if the element is present in the tuple.
5. **First Element**: Access the first element of a tuple, considering what to return if the tuple is empty.
6. **Last Element**: Access the last element of a tuple, considering what to return if the tuple is empty.
7. **Tuple Length**: Determine the number of elements in the tuple.
8. **Slice Tuple**: Create a new tuple that contains only the first three elements of the original tuple.
9. **Concatenate Tuples**: Given two tuples, create a new tuple that combines both.
10. **Check if Tuple is Empty**: Determine if a tuple has any elements.
11. **Get All Indices of Element**: Given a tuple and an element, find all the indices of that element in the tuple.
12. **Find Second Largest**: From a given tuple, find the second largest element.
13. **Find Second Smallest**: From a given tuple, find the second smallest element.
14. **Create a Single Element Tuple**: Create a tuple that contains a single specified element.
15. **Convert List to Tuple**: Given a list, create a tuple containing the same elements.

16. **Check if Tuple is Sorted**: Determine if the tuple is sorted in ascending order and return a boolean.
17. **Find Maximum of Subtuple**: Given a tuple, find the maximum element of a specified subtuple.
18. **Find Minimum of Subtuple**: Given a tuple, find the minimum element of a specified subtuple.
19. **Remove Element by Value**: Given a tuple and an element, create a new tuple that removes the first occurrence of that element.
20. **Create Nested Tuple**: Create a new tuple that contains subtuples, where each subtuple contains specified elements from the original tuple.
21. **Repeat Elements**: Given a tuple and a number, create a new tuple where each element is repeated that number of times.
22. **Create Range Tuple**: Create a tuple of numbers in a specified range (e.g., from 1 to 10).
23. **Reverse Tuple**: Create a new tuple that contains the elements of the original tuple in reverse order.
24. **Check Palindrome**: Given a tuple, check if the tuple is a palindrome (reads the same forwards and backwards).
25. **Get Unique Elements**: Given a tuple, create a new tuple that contains only the unique elements while maintaining the original order.

Set Tasks

1. **Union of Sets**: Given two sets, create a new set that contains all unique elements from both sets.
2. **Intersection of Sets**: Given two sets, create a new set that contains elements common to both sets.
3. **Difference of Sets**: Given two sets, create a new set with elements from the first set that are not in the second.
4. **Check Subset**: Given two sets, check if one set is a subset of the other.
5. **Check Element**: Given a set and an element, check if the element exists in the set.
6. **Set Length**: Determine the number of unique elements in a set.
7. **Convert List to Set**: Given a list, create a new set that contains only the unique elements from that list.
8. **Remove Element**: Given a set and an element, remove the element if it exists.
9. **Clear Set**: Create a new empty set from an existing set.
10. **Check if Set is Empty**: Determine if a set has any elements.
11. **Symmetric Difference**: Given two sets, create a new set that contains elements that are in either set but not in both.
12. **Add Element**: Given a set and an element, add the element to the set if it is not already present.
13. **Pop Element**: Given a set, remove and return an arbitrary element from the set.
14. **Find Maximum**: From a given set of numbers, find the maximum element.

15. **Find Minimum**: From a given set of numbers, find the minimum element.
16. **Filter Even Numbers**: Given a set of integers, create a new set that contains only the even numbers.
17. **Filter Odd Numbers**: Given a set of integers, create a new set that contains only the odd numbers.
18. **Create a Set of a Range**: Create a set of numbers in a specified range (e.g., from 1 to 10).
19. **Merge and Deduplicate**: Given two lists, create a new set that merges both lists and removes duplicates.
20. **Check Disjoint Sets**: Given two sets, check if they have no elements in common.
21. **Remove Duplicates from a List**: Given a list, create a set from it to remove duplicates, then convert back to a list.
22. **Get Unique Elements in Order**: Given a list, create a set that contains unique elements while maintaining their first appearance order.
23. **Create Nested Sets**: Create a set of sets, where each inner set contains a specified number of unique elements.
24. **Count Unique Elements**: Given a list, determine the count of unique elements using a set.
25. **Generate Random Set**: Create a set with a specified number of random integers within a certain range.

Dictionary Tasks

1. **Get Value**: Given a dictionary and a key, retrieve the associated value, considering what to return if the key doesn't exist.
2. **Check Key**: Given a dictionary and a key, check if the key is present in the dictionary.
3. **Count Keys**: Determine the number of keys in the dictionary.
4. **Get All Keys**: Create a list that contains all the keys in the dictionary.
5. **Get All Values**: Create a list that contains all the values in the dictionary.
6. **Merge Dictionaries**: Given two dictionaries, create a new dictionary that combines both.
7. **Remove Key**: Given a dictionary and a key, remove the key if it exists, handling the case if it doesn't.
8. **Clear Dictionary**: Create a new empty dictionary.
9. **Check if Dictionary is Empty**: Determine if a dictionary has any elements.
10. **Get Key-Value Pair**: Given a dictionary and a key, retrieve the key-value pair if the key exists.
11. **Update Value**: Given a dictionary, update the value for a specified key.
12. **Count Value Occurrences**: Given a dictionary, count how many times a specific value appears across the keys.
13. **Invert Dictionary**: Given a dictionary, create a new dictionary that swaps keys and values.

14. **Find Keys with Value**: Given a dictionary and a value, create a list of all keys that have that value.
15. **Create a Dictionary from Lists**: Given two lists (one of keys and one of values), create a dictionary that pairs them.
16. **Check for Nested Dictionaries**: Given a dictionary, check if any values are also dictionaries.
17. **Get Nested Value**: Given a nested dictionary, retrieve a value from within one of the inner dictionaries.
18. **Create Default Dictionary**: Create a dictionary that provides a default value for missing keys.
19. **Count Unique Values**: Given a dictionary, determine the number of unique values it contains.
20. **Sort Dictionary by Key**: Create a new dictionary sorted by keys.
21. **Sort Dictionary by Value**: Create a new dictionary sorted by values.
22. **Filter by Value**: Given a dictionary, create a new dictionary that only includes items with values that meet a certain condition.
23. **Check for Common Keys**: Given two dictionaries, check if they have any keys in common.
24. **Create Dictionary from Tuple**: Given a tuple of key-value pairs, create a dictionary from it.
25. **Get the First Key-Value Pair**: Retrieve the first key-value pair from a dictionary.