Functions

1. Write a function add_numbers(a, b) that returns the sum of two numbers.
2. Write a function is_even(n) that returns True if a number is even, and False otherwise.
3. Write a function factorial(n) that returns the factorial of a given number n.
4. Write a function find_max(a, b, c) that returns the largest of three numbers.
5. Write a function reverse_string(s) that returns the reverse of the input string s.
6. Write a function is_palindrome(s) that checks if the input string s is a palindrome.
7. Write a function fibonacci(n) that returns the first n numbers in the Fibonacci sequence.
8. Write a function count_vowels(s) that returns the number of vowels in the string s.
9. Write a function is_prime(n) that checks if a number n is prime.
10. Write a function sum_list(lst) that takes a list of numbers and returns their sum.
11. Write a function capitalize_words(s) that capitalizes the first letter of each word in the string s.
12. Write a function remove_duplicates(lst) that removes duplicate elements from a list.
13. Write a function merge_two_lists(lst1, lst2) that merges two lists into one.
14. Write a function sort_list(lst) that sorts a list of numbers in ascending order.
15. Write a function find_gcd(a, b) that returns the greatest common divisor of two numbers.
16. Write a function sum_of_squares(n) that returns the sum of squares of the first n natural numbers.

17. Write a function longest_word(words) that returns the longest word from a list of words.
18. Write a function calculate_area(shape, dimension) that calculates the area of a shape (circle, square, or rectangle) based on the given dimensions.
19. Write a function temperature_conversion(celsius) that converts Celsius to Fahrenheit.
20. Write a function is_anagram(s1, s2) that checks if two strings s1 and s2 are anagrams of each other.
21. Write a recursive function factorial_recursive(n) to find the factorial of a number.
22. Write a recursive function fibonacci_recursive(n) to return the n-th Fibonacci number.
23. Write a recursive function sum_digits(n) that returns the sum of the digits of a number.
24. Write a recursive function reverse_string_recursive(s) to reverse a string.
25. Write a recursive function gcd_recursive(a, b) to find the greatest common divisor of two numbers.
26. Write a recursive function power(x, n) that returns x raised to the power of n.
27. Write a recursive function flatten(lst) that flattens a nested list structure.
28. Write a lambda function to add two numbers.
29. Write a lambda function that takes a list of numbers and returns a list of their squares.
30. Write a lambda function to sort a list of tuples based on the second element.
31. Write a lambda function to filter even numbers from a list.
32. Write a lambda function that finds the maximum of two numbers.
33. Write a lambda function to check if a given string is a palindrome.
34. Use a lambda function with map() to convert a list of temperatures from Celsius to Fahrenheit.

35. Write a function sum_all(*args) that takes any number of arguments and returns their sum.
36. Write a function multiply_all(*args) that multiplies all the numbers passed as arguments.
37. Write a function print_details(**kwargs) that prints out all the key-value pairs passed as keyword arguments.
38. Write a function concatenate_strings(*args) that concatenates an arbitrary number of strings.
39. Write a function describe_person(name, **kwargs) where name is a required argument, and other details like age, job, and city are optional keyword arguments.
40. Write a function combine_args_kwargs(*args, **kwargs) that prints all positional arguments and keyword arguments passed to it.
41. Write a function find_max_in_args(*args) that returns the maximum value from the arguments passed.
42. Write a recursive function sum_all_recursive(*args) that sums all numbers passed as arguments (including nested tuples or lists).
43. Write a recursive function flatten_recursive(*args) that flattens any number of nested lists passed as arguments.