**PYTHON FUNCTIONS EXERCISE**

1. Write a function that calculates the area of a rectangle with default values for width and height.

2. Create a function that takes a string and a number (default value 1) and repeats the string the given number of times.

3. Implement a function that computes the compound interest given principal, rate, and time with default rate as 5% and time as 10 years.

4. Write a function to find the nth Fibonacci number with a default value for n as 10.

5. Create a function that converts a temperature from Celsius to Fahrenheit with a default value for the temperature.

6. Implement a function to calculate the volume of a cylinder with default values for radius and height.

7. Write a function to check if a number is even, with a default argument for the number as 2.

8. Create a function that prints the current date and time with an optional format string defaulting to a standard format.

9. Implement a function that returns a list of squares of numbers from 1 to n with a default value of n as 10.

10. Write a function that generates a list of random numbers with default values for the list size and range.

11. Write a function that calculates the total cost of an order with keyword arguments for price, quantity, and discount.

12. Create a function to display information about a student with keyword arguments for name, age, and grade.

13. Implement a function to calculate the perimeter of a rectangle using keyword arguments for length and width.

14. Write a function that formats a string with keyword arguments for prefix and suffix.

15. Create a function to find the maximum of three numbers with keyword arguments.

16. Implement a function that generates a formatted address string using keyword arguments for street, city, and zip code.

17. Write a function that calculates the BMI of a person using keyword arguments for weight and height.

18. Create a function to find the distance between two points using keyword arguments for coordinates.

19. Implement a function to create a user profile with keyword arguments for username, email, and password.

20. Write a function that generates a greeting message using keyword arguments for first name and last name.

21. Write a function to find the GCD of two numbers using positional arguments.

22. Create a function that accepts a list and an integer and returns the element at the given position.

23. Implement a function that computes the average of a list of numbers using positional arguments.

24. Write a function that finds the dot product of two vectors using positional arguments.

25. Create a function to concatenate three strings using positional arguments.

26. Implement a function that swaps the values of two variables using positional arguments.

27. Write a function to calculate the distance traveled given speed and time using positional arguments.

28. Create a function that finds the median of three numbers using positional arguments.

29. Implement a function to find the intersection of two sets using positional arguments.

30. Write a function that multiplies all the elements of a list by a given number using positional arguments.

31. Write a function that accepts any number of positional arguments and returns their sum.

32. Create a function that accepts any number of keyword arguments and prints them in a formatted string.

33. Implement a function that finds the maximum value in a list of numbers using \*args.

34. Write a function that accepts any number of strings and concatenates them using \*args.

35. Create a function that accepts any number of dictionaries and merges them using \*\*kwargs.

36. Implement a function to print all positional and keyword arguments in a formatted string using \*args and \*\*kwargs.

37. Write a function that accepts any number of positional arguments and returns their product.

38. Create a function that accepts any number of keyword arguments and calculates the sum of their values.

39. Implement a function that finds the longest string among the given positional arguments using \*args.

40. Write a function that accepts any number of lists and returns their concatenation using \*args.

41. Create a function that accepts any number of positional arguments and returns a tuple of their squares.

42. Implement a function to filter even numbers from a list of numbers using \*args.

43. Write a function that accepts any number of keyword arguments and returns a dictionary of their uppercase keys.

44. Create a function that accepts any number of positional arguments and finds their average using \*args.

45. Implement a function that accepts any number of keyword arguments and returns a list of their keys and values.

46. Write a function that accepts any number of lists and returns a flattened list using \*args.

47. Create a function that accepts any number of keyword arguments and returns a string of their concatenated values.

48. Implement a function to find the union of multiple sets using \*args.

49. Write a function that accepts any number of positional and keyword arguments and prints them in reverse order.

50. Create a function that accepts any number of keyword arguments and returns a dictionary with keys sorted alphabetically.