# Python Homework Problems

## No Parameters / No Return Value

* 1. Write a function `greet()` that prints 'Hello, World!' when called.
* 2. Create a function `show\_date\_time()` that prints the current date and time.
* 3. Write a function `display\_even\_numbers()` that prints even numbers from 1 to 20.

## Parameters / No Return Value

* 4. Write a function `greet\_user(name)` that takes a name as a parameter and prints 'Hello, [name]!'
* 5. Create a function `print\_square(n)` that prints the square of a given number.
* 6. Write a function `multiply\_numbers(a, b)` that takes two numbers and prints their product.

## No Parameters / Return Value

* 7. Write a function `get\_pi()` that returns the value of π (3.14159).
* 8. Create a function `random\_number()` that returns a random number between 1 and 100.
* 9. Write a function `current\_year()` that returns the current year.

## Parameters / Return Value

* 10. Write a function `add\_numbers(a, b)` that returns the sum of two numbers.
* 11. Create a function `is\_even(n)` that returns `True` if the number is even and `False` otherwise.
* 12. Write a function `get\_factorial(n)` that returns the factorial of a given number.

## Recursive Functions

* 13. Write a recursive function `countdown(n)` that prints numbers from `n` to `1`.
* 14. Create a recursive function `sum\_natural(n)` that returns the sum of the first `n` natural numbers.
* 15. Write a recursive function `fibonacci(n)` that returns the `n`th Fibonacci number.

## \*args and \*\*kwargs

* 16. Write a function `sum\_numbers(\*args)` that takes multiple numbers and returns their sum.
* 17. Create a function `print\_info(\*\*kwargs)` that prints key-value pairs passed as arguments.

## Default Arguments

* 18. Write a function `power(base, exponent=2)` that returns `base` raised to the power of `exponent`.

## Function Type Hinting & Documentation

* 19. Create a function `calculate\_area(length: float, width: float) -> float` that returns the area of a rectangle. Add a docstring explaining the function.

## Scope, map(), and filter()

* 20. Given a list of numbers, use `map()` to square each number and use `filter()` to return only the even squares. Write a function `process\_numbers(nums: list) -> list`.

## Additional Problems

* 21. Write a function `reverse\_string(s: str) -> str` that returns the reversed version of a given string.
* 22. Create a function `count\_vowels(s: str) -> int` that returns the number of vowels in a given string.
* 23. Write a recursive function `gcd(a, b)` to find the greatest common divisor of two numbers.
* 24. Implement a function `merge\_lists(\*lists)` that takes multiple lists as arguments and returns a single merged list.
* 25. Write a function `student\_data(name, age=18, \*\*details)` that takes a name and age (default 18) and prints additional details from `\*\*details`.
* 26. Create a function `calculate\_interest(principal: float, rate: float = 5.0, time: int = 1) -> float` that returns the calculated simple interest.
* 27. Use `map()` to convert a list of Fahrenheit temperatures to Celsius.
* 28. Use `filter()` to extract all prime numbers from a given list.
* 29. Write a function `calculate\_discount(price: float, discount: float = 10.0) -> float` that returns the final price after applying a percentage discount.
* 30. Write a function `unique\_words(sentence: str) -> set` that returns a set of unique words from a given sentence.