SECTION A: Fundamentals and Syntax (10 Questions)

1. What is the **primary purpose** of the return statement within a Python function?
2. What happens to the function's execution immediately after the return statement is executed?
3. If a function reaches the end of its defined body without encountering any return statement, what value does the function implicitly return?
4. Write a simple function, add_one(x), that takes one argument and uses return to send back the result of adding one to that argument.
5. What is the specific type of the value returned by a function that implicitly returns a value (as described in Q3)?
6. Write a function called print_and_return(message) that both prints a message to the console *and* returns the boolean value True.
7. If a function has multiple return statements, how many of them can be executed during a single call to that function?
8. Can you place a return statement outside of a function definition (i.e., directly in the main script body)? Explain briefly.
9. Explain the difference between print(x) and return x within the context of a function.
10. Can a Python function return multiple values? If so, what is the data type of the object that contains these multiple values?

SECTION B: Control Flow and Scope (10 Questions)

11. Analyze the following function. What is the output printed to the console, and what is the final return value?

    Python

```
def analyze_flow(n):
    if n > 5:
        return "Too High"
    print("Checking...")
    return "Just Right"
```

```
result = analyze_flow(7)
# Output: ?
# Return Value: ?
```

12. Modify the function from Q11 so that if n > 5, it prints "Exiting Early" *before* returning "Too High".
13. Describe how the return statement affects the values of local variables defined within the function after the function call has completed.
14. What happens if you try to put a statement *immediately after* a non-conditional return statement within a function body? (e.g., return result; print("Done")).
15. Explain how the return statement is used to terminate a recursive function, and what this terminating condition is called.
16. In the context of a for loop inside a function, how does return differ from the break statement?
17. Write a function check_list(data_list) that iterates through a list and immediately returns True the first time it finds the string "target". If the loop finishes without finding the target, it should return False.
18. Can a return statement be used inside a try...finally block? If so, what is the crucial consideration regarding the code in the finally block?
19. What specific effect does return have on an else block that immediately follows an if block within the same function?
20. Explain why using return inside a nested loop is often the cleanest way to exit both loops simultaneously and report a result.

SECTION C: Advanced Concepts and Applications (10 Questions)

21. What is a **Generator function** in Python, and what specific keyword is used instead of return to yield values?
22. If a generator function uses a standard return statement, what is the effect on the iteration process, and what value can be optionally returned?

23. In the context of **Class Methods** or **Static Methods**, what does the return statement typically provide back to the caller (e.g., an instance of the class, or a simple value)?
24. Explain the concept of **Function Composition** and how the return statement is essential for chaining functions together (e.g., result = func_B(func_A(data))).
25. Write a function called get_config() that returns a **dictionary** containing at least two key-value pairs (e.g., configuration settings).
26. What is the value of x after the following execution?

Python

```
def set_x(val):
    x = val
    return 1

x = 0
set_x(100)
# x = ?
```

27. Consider a scenario where a function calculates two different metrics, M_1 and M_2. Write the most Pythonic way to return these two metrics in a single line.
28. When writing unit tests for a function, why is the return value often more important to test than the function's side effects (like printing)?
29. Describe a situation where explicitly returning None (e.g., return None) is better practice than relying on the implicit return.
30. If a function is expected to return a list of items, what should the function return when it finds no items to satisfy the request (e.g., an empty list, None, or raise an exception)? Justify your choice.