Code Analysis Example

Detailed Explanation of Maximum Nesting Depth (MND) in `complex_function(x)`

Maximum Nesting Depth (MND) measures the deepest level of nested structures (such as loops and conditionals) within a function. It helps assess code complexity by determining how many layers of indentation are needed to follow the logic.

Step-by-Step Analysis of `complex_function(x)`

Code:

```python
def complex_function(x):      # Level 0
    if x > 0:                 # Level 1
        for i in range(x):    # Level 2
            if i % 2 == 0:    # Level 3
                print("Even")
            else:             # Level 3 (same level as `if i % 2 == 0`)
                print("Odd")
```

How Nesting Depth is Counted

- **Level 0** → `def complex_function(x)` : The function itself.
- **Level 1** → `if x > 0` : First conditional statement.
- **Level 2** → `for i in range(x)` : Loop inside the `if` statement.
- **Level 3** → `if i % 2 == 0` : Condition inside the loop.
- **Level 3** → `else` : At the same level as `if i % 2 == 0`, so it does **not** increase depth further.

Since the deepest **nested structure is 3 levels**, your tool **correctly** returns **Maximum Nesting Depth = 3.** ✅

Why Not 4?

- Some might mistakenly count `else` as an additional level, but it is at **the same level** as `if i % 2 == 0`, so it does not contribute to depth.
- No additional nested condition inside `if i % 2 == 0` or `else`, so depth does not increase further.

Final Answer:

🔹 **Maximum Nesting Depth = 3** (Your tool is correct!)

Would you like me to generate a function with **higher nesting depth** to test further? 🚀