tags: - cocubes - pseudocode - method

Pseudocode Dry-Run Method

Six steps

1. Circle the initial values and inputs.
2. Translate the goal: output, final variable, count, or complexity.
3. Create a small table only for variables that change.
4. Execute one line at a time, top to bottom.
5. For loops, write each iteration value explicitly until the pattern is safe.
6. Check whether the final DISPLAY occurs inside or outside the loop.

Example: update order matters

```
SET a = 4, b = 7
SET a = a + b
SET b = a - b
SET a = a - b
DISPLAY a, b
```

Step	a	b
start	4	7
a = a+b	11	7
b = a-b	11	4
a = a-b	7	4

Output: 7, 4. This is an arithmetic swap.

Example: loop trace

```
SET total = 0
FOR i = 1 TO 5
  IF i MOD 2 = 1 THEN
    SET total = total + i
  END IF
END FOR
DISPLAY total
```

i	odd?	total after iteration
1	yes	1
2	no	1
3	yes	4
4	no	4
5	yes	9

Output: 9.

Conditional method

For nested conditions, label the IF blocks:

```
IF A
  IF B
    action 1
  ELSE
    action 2
  END IF
ELSE
  action 3
END IF
```

The inner ELSE belongs to the nearest unmatched IF. Explicit END IF markers remove ambiguity; if they are absent, apply the nearest-IF rule.

Recursion method

1. Write the base case first.
2. Expand calls until the base case.
3. Resolve values from the deepest call upward.

```
FUNCTION F(n)
  IF n = 0 THEN RETURN 0
  ELSE RETURN n + F(n - 1)
END FUNCTION
```

F(3) = 3 + F(2) = 3 + 2 + F(1) = 3 + 2 + 1 + F(0) = 6.

Complexity method

Count how many times the innermost body executes.

- one loop from 1 to n: O(n);
- two independent loops nested to n: O(n^2);
- variable doubled each iteration: O(log n);
- n outer iterations and log n inner iterations: O(n log n).

[[04_Pseudocode/Pseudocode_Drills|Practise now]]