

Pseudocode Patterns and Traps

Arithmetic operators

Follow parentheses first, then multiplication/division/modulo, then addition/subtraction. When a question does not define an operator, use the definitions given in its instructions.

- a MOD b: remainder after a/b.
- a DIV b: integer quotient, discarding fractional part.

Example: $17 \text{ DIV } 5 = 3$ and $17 \text{ MOD } 5 = 2$.

Loop comparison

Loop	When condition is checked	Minimum executions
WHILE condition	before body	0
REPEAT body UNTIL condition	after body	1
FOR i = start TO end	defined count	depends on range

The meaning of TO can be inclusive in many pseudocode conventions, but follow the exact test statement. If it writes $i < \text{end}$, the end is excluded.

Digit extraction pattern

```
SET digit = n MOD 10
SET n = n DIV 10
```

This gets and removes the last decimal digit. It is used for reversing a number, digit sum and palindrome checks.

Count pattern

```
SET count = 0
FOR each value x
  IF condition(x) THEN
    SET count = count + 1
  END IF
END FOR
```

Do not confuse count with sum.

Maximum/minimum pattern

```
SET best = first value
FOR each remaining value x
  IF x > best THEN SET best = x
END FOR
```

For minimum, reverse the comparison.

Flag pattern

```
SET found = FALSE
FOR each value x
  IF x = target THEN
    SET found = TRUE
  END IF
END FOR
```

Questions may ask whether a flag starts false/true, or whether a later statement overwrites it.

Array indexing

Do not assume arrays begin at zero or one. A question may say ARRAY[1..n], or show an explicit initial index. Follow that declaration.

CASE pattern

CASE selects one matching alternative. Look for:

- exact cases;
- default/otherwise;
- whether a break/end-case is implied in that pseudocode convention.

Bitwise awareness

Some employer variants include bitwise operations. If shown:

- AND keeps a 1 only where both bits are 1.
- OR keeps a 1 where either bit is 1.
- XOR keeps a 1 where bits differ.
- Left shift by one roughly doubles a non-overflowing positive integer.
- Right shift by one roughly halves a nonnegative integer.

Only use bitwise rules when the prompt explicitly uses bitwise operators.

Dry-run checklist

- ☐ Did I update the table after every assignment?
- ☐ Is the loop condition true before the first WHILE iteration?
- ☐ Did I use integer, not decimal, division?
- ☐ Does the ELSE attach to the correct IF?
- ☐ Did the recursive call reach a base case?
- ☐ Did I track exactly what is displayed?

[[04_Pseudocode/Dry_Run_Method|Dry-run method]] · [[04_Pseudocode/Pseudocode_Drills|Drills]]