

Solutions

Each answer shows the steps, then the **result**.

2.1 any one:

- the procedure can be used without knowing how it works
- its body can be changed without changing the code that calls it
- the code is written once and reused

2.2

```
PROCEDURE isEven(n)
{
    RETURN(n MOD 2 = 0)
}
```

- uses `n MOD 2 = 0`
- **returns** the Boolean result

2.3

- $80 \times (1 - 0.25) = 80 \times 0.75 = \mathbf{60}$

2.4

- **RETURN ends** the procedure at once
- so the count would be returned after the **first** element, not after all of them

3.1 A

- the user of a procedure needs only its name, parameters and result

3.2 B

- when `a > b` it returns `a`; otherwise `b`, which is the larger or equal one

3.3 C

- the procedure still runs the drawing instructions; they are just written once

3.4

(a)

```
PROCEDURE average(aList)
{
    total ← 0
    FOR EACH n IN aList
    {
        total ← total + n
    }
    RETURN(total / LENGTH(aList))
}
```

- sum every element in a loop
- divide by `LENGTH(aList)`
- **return** the result after the loop

(b)

- `DISPLAY(average(steps))`

(c)

- the averaging code is written **once** and can be called for any list, e.g. each week's steps
- without it, the loop would be **repeated** wherever an average is needed, and each copy changed for a fix