

Solutions

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

2.1

- a **function returns a value** (used in an expression); a **procedure does not** return a value (it performs an action)

2.2

- PROCEDURE ... ENDPROCEDURE with a parameter

```
PROCEDURE Greet(Name : STRING)
    OUTPUT "Hello, ", Name
ENDPROCEDURE
```

2.3

- FUNCTION ... RETURNS ... ENDFUNCTION with RETURN

```
FUNCTION Cube(x : INTEGER) RETURNS INTEGER
    RETURN x * x * x
ENDFUNCTION
```

2.4

- an **argument** is the value the caller passes in
- a **parameter** is the variable inside the subroutine that **receives** it

3.1

- FUNCTION ... RETURNS INTEGER; compare; RETURN the larger

```
FUNCTION MaxOf(a : INTEGER, b : INTEGER) RETURNS INTEGER
    IF a > b THEN
        RETURN a
    ELSE
        RETURN b
    ENDIF
ENDFUNCTION
```

3.2

- **by value** — the routine gets a **copy**, so changes inside it do **not** affect the caller
- **by reference** — the routine shares the caller's actual variable, so changes **do** affect it
- Swap must use **pass by reference (BYREF)**
- so the swapped values remain in the caller's variables after the call

3.3

- any two: avoids repeating code; easier to read/maintain; can be tested on its own; supports decomposition/teamwork

3.4

- a variable declared **inside** a subroutine that exists **only while that subroutine runs** (and is not visible outside it)