Solutions

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

2.1

- it adds `price` to the current value of `total` and stores the result back in `total`

2.2

- `17` number; `"17"` string
- `true` Boolean; `[3, 8]` list

2.3

- `p = 2`, `q = 3`, then `p = 9`
- `q = 9 - 3 = 6`: **p = 9, q = 6**

2.4

- a meaningful name such as `studentsArrived` or `arrivedCount`

3.1 B

- `y` copies 3; `x` becomes 7; changing `x` does not change `y`

3.2 C

- the temporary variable keeps a's value before it is overwritten
- A and B overwrite one value first, so both end up equal

3.3 B

- `a ← b` gives 20; `b ← c` gives 30; `c ← a` copies a's **current** value, 20

3.4 C

- assignment replaces the old value; a variable holds only its most recent value

3.5

(a)

```
IF(score > best)
{
    best ← score
}
```

- condition `score > best`
- assignment `best ← score` inside the IF

(b)

- `score` becomes **55**
- assignment **replaces** the old value; a variable holds only one, the most recent, value

(c)

- `best` says what the variable holds, so the code is easier to read and maintain