Solutions

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

2.1

- `sum` 2, 7, 15 and `n` 5, 8, 11
- prints **15 11**; the body runs **3** times

2.2

- `5 < 5` is false at the first test: **0** times

2.3

- `i` never changes, so `i < 10` stays true
- add `i++;` inside the body

2.4

- digits 0, 0, 5, 0, 1: **3**

3.1 B

- 20 → 10, 5, 2, 1; then `1 > 1` is false

3.2 B

- `p` 3, 9, 27, 81, 243: five passes

3.3 B

- an odd `n` goes ..., 1, -1, ... and skips 0

3.4

(a)

```
int years = 0;
while (balance < 2000)
{
    balance = balance * 1.05;
    years++;
}
```

- `years` starts at 0 and the condition is `balance < 2000`
- multiplies by 1.05 (or adds 5%) each pass
- adds 1 to `years` each pass

(b)

- **0**
- `2500 < 2000` is false at the first test, so the body never runs

(c)

- the balance becomes 5% of itself (50), getting **smaller** every year
- it never reaches 2000, so the loop never stops