Solutions

Marking: **M1** a correct method, **A1** the correct answer, **B1** an independent mark.

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

2.1

`1, 4, 7` and `10, 6, 2`

nothing: an empty range

$0, 7, \ldots, 98$: **15** values

2.2

n: 3, 10, 5, 16

8, 4, 2, 1

`steps` = **8**

2.3

first: `i` never changes, so `i < 10` stays `True`

second: `i` goes 10, 7, 4, 1, −2, …, so it skips 0 and `!= 0` stays `True`

2.4

`total = 0` then `for n in range(2, 101, 2): total += n`

2550

3.1

(a) `abc`, 1, `False`; `pass1`, 2, `False`; `secret`, 3, `True`; output `Welcome`

(b) `Locked`: the loop stops after three tries, so `secret` is never read

(c)

```
ok = False
for _ in range(3):
    if input("Password? ") == "secret":
        ok = True
        break
print("Welcome" if ok else "Locked")
```

`for _ in range(3)`; sets `ok` on a match; `break` stops asking

3.2

(a) **45**: `range(1, n)` stops at $n - 1 = 9$, so 10 is never added

(b) `range(1, n + 1)`

(c) $\dfrac{n(n+1)}{2} = \dfrac{10 \times 11}{2} = 55$

(d) $n = 1$: expected 1, the bug gives 0

3.3

(a)

```
balance = 1000
years = 0
while balance < 2000:
    balance *= 1.03
    years += 1
```

start values; condition `< 2000`; grows the balance and counts inside the loop

(b) **24** years: after 23 years £1,973.59; after 24 years £2,032.79

(c) $72/3 = 24$ years

(d) an infinite loop: the balance jumps from below 2000 to above it and is never exactly 2000, so `!=` stays `True`

exercises.lol exercises.com.cn