Solutions

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

2.1

- 12, 7, **40**
- **5** comparisons: every item

2.2

- pass 1: `[2, 5, 1, 4, 8]`
- pass 2: `[2, 1, 4, 5, 8]`

2.3

- pass 1: 1 swaps with 5: `[1, 2, 8, 5, 4]`
- pass 2: 2 is already in place: `[1, 2, 8, 5, 4]`

2.4

- (i) **1000**
- (ii) **10**, since $2^{10} = 1024 \geq 1000$

3.1 C

- it discards the half that cannot hold the target

3.2 D

- a loop inside a loop

3.3 A

- every neighbouring pair is already in order

3.4

(a)

`low`	`high`	`mid`	`a[mid]`
0	10	5	166
6	10	8	190

- first probe correct: `mid` 5, 166
- $166 < 190$, so `low` $= 6$
- second probe: `mid` $(6 + 10)//2 = 8$
- 190 found at index **8**

(b)

- **9**

(c)

- the list is no longer sorted
- the first probe finds $166 > 140$ and discards the right half, which holds 140, so it returns not found

(d)

- **binary** search
- it is $O(\log n)$: about 16 comparisons for 50 000 items, against up to 50 000 for linear
- the list must be kept sorted, which is cheap as books are added rarely