

# Solutions

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

## 2.1

- **analysis**
- **design**
- **coding**
- **testing**

## 2.2

- validation checks that data is sensible
- verification checks that data was copied or entered correctly

## 2.3

- at least 8 characters: **length check**
- age from 11 to 18: **range check**
- name not blank: **presence check**
- mark must be a number: **type check**

## 2.4

- one normal value strictly inside 10 to 50, such as **30**
- extreme values **10** and **50**
- one rejected boundary value, **9** or **51**

## 2.5

- it records the value of each variable after each step
- it is used to follow an algorithm and find what it does, or to find an error

## 2.6

- input: the three numbers
- process: keep the largest value seen so far
- output: that largest value

## 3.1 B

- abstraction keeps only the important details
- A is a copy check, C is a validation check, and D is a sort

## 3.2

(a)

- **range check**

(b)

- **double entry**

(c)

- normal: a value such as **40**
- abnormal: a non-number such as **cat**, or a value such as **-5**
- extreme: **0** or **100**
- boundary: an accepted edge together with the rejected neighbour, such as **0 and -1** or **100 and 101**

## 3.3

- after input 3, **Max** is **3**
- after input 9, **Max** is **9**
- after input 4, **Max** stays **9**
- the output is **9**

| I | Number | Max | Output |
|---|--------|-----|--------|
| 1 | 3      | 3   |        |
| 2 | 9      | 9   |        |
| 3 | 4      | 9   | 9      |

## 3.4

(a)

- 0 is greater than every negative input, so the condition **Number > Max** is never true
- **Max** stays 0, which was not one of the inputs

(b)

- store the **first number** in **Max** before comparing the rest
- that value is **List[1]**, or the first input