

## Solutions

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

### 2.1

- bounds [1:7] mean seven elements
- 7 values

### 2.2

- a **1D** array is a single list using one index
- a **2D** array is a table using two indexes [row, column]

### 2.3

- bounds [1:20] and type REAL
- **DECLARE Numbers : ARRAY[1:20] OF REAL**

### 2.4

- date as DD/MM/YYYY: **format check**
- age 0 to 120: **range check**
- name not empty: **presence check**

### 2.5

- **validation**: data is sensible/reasonable; the computer does it automatically
- **verification**: data has been copied or entered **accurately** compared with the original

### 3.1 B

- one subject is a single list, so a **1D array**
- A cannot hold 30 values; C is for two dimensions; D cannot change

### 3.2

(a)

- **FOR I ← 1 TO 5; OUTPUT Names[I]; NEXT I**

```
FOR I ← 1 TO 5
    OUTPUT names[I]
NEXT I
```

(b)

- a **2D array**
- each student needs 5 marks, so two indexes [student, subject]

### 3.3

(a)

- bounds [1:30], type INTEGER

- **DECLARE Marks : ARRAY[1:30] OF INTEGER**

(b)

- outer **FOR Count ← 1 TO 30**
- inner **REPEAT ... UNTIL Mark >= 0 AND Mark <= 100**
- **INPUT Mark** inside; error message if rejected; **Marks[Count] ← Mark**

```
FOR Count ← 1 TO 30
    REPEAT
        INPUT Mark
        IF Mark < 0 OR Mark > 100
            THEN
                OUTPUT "Invalid mark"
            ENDIF
        UNTIL Mark >= 0 AND Mark <= 100
        Marks[Count] ← Mark
    NEXT Count
```

(c)

- **normal**: 45, accepted
- **abnormal**: -5 or A, rejected
- **extreme**: 0 or 100 accepted; 101 rejected

(d)

- **OPENFILE "Marks.txt" FOR WRITE**
- **WRITEFILE "Marks.txt", Marks[1]**
- **CLOSEFILE "Marks.txt"**

(e)

- data still in memory may not have been written, so closing **makes sure it is saved**

(f)

- **double entry**, a verification check
- a typing slip is unlikely to be repeated identically the second time