Exercise walk-through

8.2 Arrays

IGCSE Computer Science

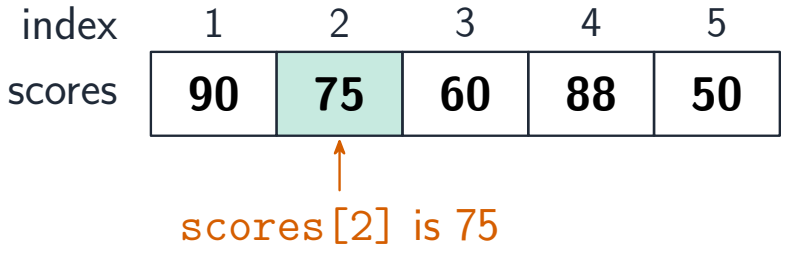
What we will cover

1. Variables, constants, data types
2. The three structures
3. Operators & strings
4. Procedures & functions
5. Arrays
6. File handling

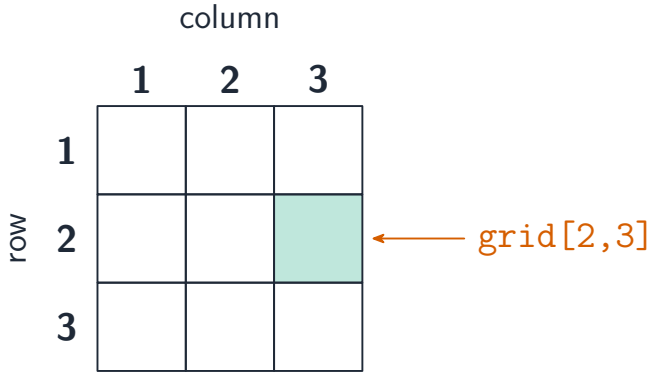


a variable is a named box that holds a value

Arrays



1-D: a list, one index



2-D: a table, [row, column]

An **array** holds many values of the same type under one name, picked out by an **index** : one for a list (`scores[2]`), two for a table (`grid[2,3]`) – perfect for a `FOR` loop.

Topic 8 – key takeaways

1 Structures

sequence, selection, iteration

2 Loops

FOR known; WHILE before; REPEAT after

3 Subroutines

procedure acts; function returns a value

4 Arrays & files

array = same type by index; file survives runs

Exercises 2.1·2.3

2.1 An array is declared `DECLARE Temps : ARRAY[1:7] OF REAL`. State how many values it can hold. [1]

2.2 State one difference between a 1D and a 2D array. [2]

2.3 Give the pseudocode declaration for an array holding 20 real numbers. [2]

Exercises 2.1·2.3

2.1 An array is declared `DECLARE Temps : ARRAY[1:7] OF REAL`. State how many values it can hold. [1]

2.1

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

2.2 State one difference between a 1D and a 2D array. [2]

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 Give the pseudocode declaration for an array holding 20 real numbers. [2]

2.3

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

Exercises 2.4·2.5

2.4 Identify the type of validation check for each: a date entered as DD/MM/YYYY; an age between 0 and 120; a name box that must not be left empty. [3]

2.5 Explain the difference between validation and verification. [2]

Exercises 2.4·2.5

2.4 Identify the type of validation check for each: a date entered as DD/MM/YYYY; an age between 0 and 120; a name box that must not be left empty. [3]

2.4

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

2.5 Explain the difference between validation and verification. [2]

2.5

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

Exercise 3.1

3.1 Which data structure best stores the marks of 30 students in **one** subject? [1]

- ☐ **A** a single variable
- **B** a 1D array
- **C** a 2D array
- **D** a constant

Exercise 3.1

3.1 Which data structure best stores the marks of 30 students in **one** subject? [1]

- ☐ **A** a single variable
- **B** a 1D array
- **C** a 2D array
- **D** a constant

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

Exercise 3.2 (a)

3.2 An array `names` holds 5 names in elements 1 to 5.

(a) Write pseudocode to output every name using a loop. [3]

Exercise 3.2 (a)

3.2 An array `names` holds 5 names in elements 1 to 5.

(a) Write pseudocode to output every name using a loop. [3]

(a)

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

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

Exercise 3.2 (b)

3.2 An array `names` holds 5 names in elements 1 to 5.

(b) State which type of array (1D or 2D) you would use to store the marks of 30 students in **5** subjects, and why. [2]

Exercise 3.2 (b)

3.2 An array `names` holds 5 names in elements 1 to 5.

(b) State which type of array (1D or 2D) you would use to store the marks of 30 students in **5** subjects, and why. [2]

(b)

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

Exercise 3.3 (a)

3.3 A program stores the marks of 30 students in a one-dimensional array called `Marks`, out of a maximum of 100.

(a) Write the pseudocode declaration for the array. [2]

Exercise 3.3 (a)

3.3 A program stores the marks of 30 students in a one-dimensional array called `Marks`, out of a maximum of 100.

(a) Write the pseudocode declaration for the array. [2]

(a)

- bounds `[1:30]`, type `INTEGER`
- `DECLARE Marks : ARRAY[1:30] OF INTEGER`

Exercise 3.3 (b)

3.3 A program stores the marks of 30 students in a one-dimensional array called `Marks`, out of a maximum of 100.

(b) Write pseudocode that inputs 30 marks into the array, rejecting any mark that is not between 0 and 100 and asking again. [6]

Exercise 3.3 (b)

3.3 A program stores the marks of 30 students in a one-dimensional array called `Marks`, out of a maximum of 100.

(b) Write pseudocode that inputs 30 marks into the array, rejecting any mark that is not between 0 and 100 and asking again. [6]

(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
```

Exercise 3.3 (c)

3.3 A program stores the marks of 30 students in a one-dimensional array called `Marks`, out of a maximum of 100.

(c) Complete the test plan by giving one value of each type and the expected result. [3]

Type of test data	Value	Expected result
normal		
abnormal		
extreme		

Exercise 3.3 (c)

3.3 A program stores the marks of 30 students in a one-dimensional array called `Marks`, out of a maximum of 100.

(c) Complete the test plan by giving one value of each type and the expected result. [3]

Type of test data	Value	Expected result
normal		
abnormal		
extreme		

(c)

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

Exercise 3.3 (d)

3.3 A program stores the marks of 30 students in a one-dimensional array called `Marks`, out of a maximum of 100.

(d) The marks are then written to a file. Write the three lines of pseudocode that open the file for writing, write `Marks[1]` and close it. [3]

Exercise 3.3 (d)

3.3 A program stores the marks of 30 students in a one-dimensional array called `Marks`, out of a maximum of 100.

(d) The marks are then written to a file. Write the three lines of pseudocode that open the file for writing, write `Marks[1]` and close it. [3]

(d)

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

Exercise 3.3 (e)

3.3 A program stores the marks of 30 students in a one-dimensional array called `Marks`, out of a maximum of 100.

(e) Explain why the file must be closed. [1]

Exercise 3.3 (e)

3.3 A program stores the marks of 30 students in a one-dimensional array called `Marks`, out of a maximum of 100.

(e) Explain why the file must be closed. [1]

(e)

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

Exercise 3.3 (f)

3.3 A program stores the marks of 30 students in a one-dimensional array called `Marks`, out of a maximum of 100.

(f) The teacher's name is typed twice when the program starts. Identify this check and explain its purpose. [2]

Exercise 3.3 (f)

3.3 A program stores the marks of 30 students in a one-dimensional array called `Marks`, out of a maximum of 100.

(f) The teacher's name is typed twice when the program starts. Identify this check and explain its purpose. [2]

(f)

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