Exercise walk-through

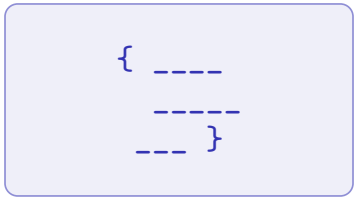
8.1 Programming concepts

IGCSE Computer Science

Programming Fundamentals

IGCSE Computer Science ▪ Topic 8

IGCSE Computer Science



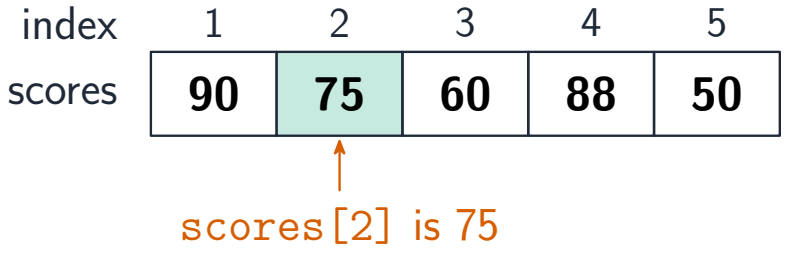
What we will cover

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

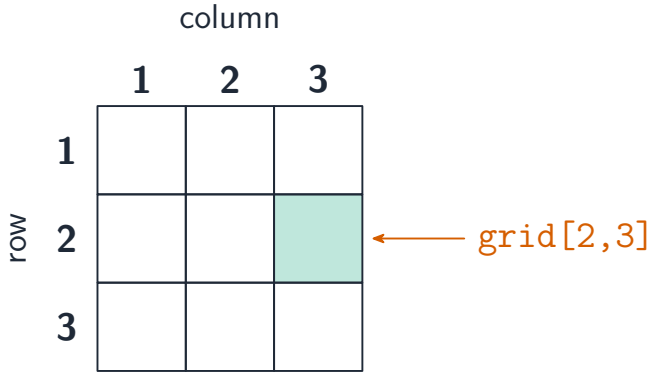


a variable is a named box that holds a value

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.

Exercises 2.1·2.3

2.1 State the difference between a variable and a constant. [2]

2.2 Give the value of `23 MOD 4`. [1]

2.3 Give the pseudocode declaration for a variable that stores a student's name, and a constant for a tax rate of 0.2. [2]

Exercises 2.1·2.3

2.1 State the difference between a variable and a constant. [2]

2.1

- a **variable** can change while the program runs
- a **constant** is fixed

2.2 Give the value of `23 MOD 4`. [1]

2.2

- `MOD` is the remainder after division
- $23 = 5 \times 4 + 3$, so `3`

2.3 Give the pseudocode declaration for a variable that stores a student's name, and a constant for a tax rate of 0.2. [2]

2.3

- name: `DECLARE Name : STRING`
- tax rate: `CONSTANT TaxRate ← 0.2`

Exercises 2.4·2.5

2.4 Identify the data type you would use for each of: a price, whether a box is ticked, a single grade letter. [3]

2.5 Explain the difference between a WHILE loop and a REPEAT loop. [2]

Exercises 2.4·2.5

2.4 Identify the data type you would use for each of: a price, whether a box is ticked, a single grade letter. [3]

2.4

- price: `REAL`
- box ticked: `BOOLEAN`
- grade letter: `CHAR`

2.5 Explain the difference between a WHILE loop and a REPEAT loop. [2]

2.5

- **WHILE** tests before the body, so the body may run **zero times**
- **REPEAT** tests after the body, so it **always runs at least once**

Exercise 3.1

3.1 A loop that always runs **at least once** is a: [1]

- ☐ **A** FOR loop
- **B** WHILE (pre-condition) loop
- **C** REPEAT (post-condition) loop
- **D** CASE statement

Exercise 3.1

3.1 A loop that always runs **at least once** is a: [1]

- ☐ **A** FOR loop
- **B** WHILE (pre-condition) loop
- **C** REPEAT (post-condition) loop
- **D** CASE statement

3.1 C

- a **REPEAT (post-condition)** loop always runs at least once
- A and B can run zero times; D is not a loop

Exercise 3.2 (a)

3.2 A program reads 10 marks and outputs their total.

(a) State which control structure repeats the input 10 times. [1]

Exercise 3.2 (a)

3.2 A program reads 10 marks and outputs their total.

(a) State which control structure repeats the input 10 times. [1]

(a)

- repeating a known number of times is **iteration** (a count-controlled / FOR loop)

Exercise 3.2 (b)

3.2 A program reads 10 marks and outputs their total.

(b) Write pseudocode using a count-controlled loop to total 10 input marks. [4]

Exercise 3.2 (b)

3.2 A program reads 10 marks and outputs their total.

(b) Write pseudocode using a count-controlled loop to total 10 input marks. [4]

(b)

- `Total ← 0`; `FOR I ← 1 TO 10`; input and add; output after

```
Total ← 0
FOR I ← 1 TO 10
    INPUT Mark
    Total ← Total + Mark
NEXT I
OUTPUT Total
```

Exercise 3.3

3.3 State the difference between a procedure and a function. [2]

Exercise 3.3

3.3 State the difference between a procedure and a function. [2]

3.3

- a **function** returns a value to where it was called
- a **procedure** does a task but returns no value

Exercise 3.4 (a)

3.4 A program records the times, in whole seconds, taken by 20 runners.

(a) Complete the pseudocode declarations for the total, one runner's time and the loop counter. [3]

Exercise 3.4 (a)

3.4 A program records the times, in whole seconds, taken by 20 runners.

(a) Complete the pseudocode declarations for the total, one runner's time and the loop counter. [3]

(a)

- `DECLARE Total : INTEGER`
- `DECLARE Time : INTEGER`
- `DECLARE Count : INTEGER`

Exercise 3.4 (b)

3.4 A program records the times, in whole seconds, taken by 20 runners.

(b) Write pseudocode that inputs the 20 times, adds them to a total, and outputs the average. [6]

Exercise 3.4 (b)

3.4 A program records the times, in whole seconds, taken by 20 runners.

(b) Write pseudocode that inputs the 20 times, adds them to a total, and outputs the average. [6]

(b)

- `Total ← 0` before the loop; `FOR Count ← 1 TO 20`; `INPUT Time`; `Total ← Total + Time`; `Average ← Total / 20`; `OUTPUT Average`

```
Total ← 0
FOR Count ← 1 TO 20
    INPUT Time
    Total ← Total + Time
NEXT Count
Average ← Total / 20
OUTPUT Average
```

Exercise 3.4 (c)

3.4 A program records the times, in whole seconds, taken by 20 runners.

(c) The program must also output the fastest time. Describe, in words, the extra steps needed inside the loop. [3]

Exercise 3.4 (c)

3.4 A program records the times, in whole seconds, taken by 20 runners.

(c) The program must also output the fastest time. Describe, in words, the extra steps needed inside the loop. [3]

(c)

- set `Fastest` to the first time (or a very large value) before the loop
- compare each new time with `Fastest`
- if smaller, replace `Fastest`; output it after the loop

Exercise 3.4 (d)

3.4 A program records the times, in whole seconds, taken by 20 runners.

(d) A time of 137 seconds must be shown as minutes and seconds. Give the two expressions, using MOD and DIV, and state the values they produce. [3]

Exercise 3.4 (d)

3.4 A program records the times, in whole seconds, taken by 20 runners.

(d) A time of 137 seconds must be shown as minutes and seconds. Give the two expressions, using MOD and DIV, and state the values they produce. [3]

(d)

- `137 DIV 60` = **2** minutes
- `137 MOD 60` = **17** seconds
- displayed as **2 minutes 17 seconds**

Exercise 3.4 (e)

3.4 A program records the times, in whole seconds, taken by 20 runners.

(e) The conversion in part (d) is needed in three places in the program. Explain why writing it as a function is better than repeating the code, and state what a function does that a procedure does not. [3]

Exercise 3.4 (e)

3.4 A program records the times, in whole seconds, taken by 20 runners.

(e) The conversion in part (d) is needed in three places in the program. Explain why writing it as a function is better than repeating the code, and state what a function does that a procedure does not. [3]

(e)

- written once and called three times, so a correction is made in **one place**
- easier to read and test
- a **function returns a value**; a procedure does not