11 Programming – Part 1

Name: _____ Class: _____ Date: _____

Vocabulary

Write each word three times.

English	**Write it 3 times** (English)		
variable			
constant			
assignment			
selection			
iteration			
array			
subroutines			
procedure			
function			
parameters			
local variable			
scope			

Recall

At IGCSE you wrote simple programs with input, output and `IF`. Bring this with you:

- A program follows your instructions in order.

- You used `IF` to make choices and loops to repeat.

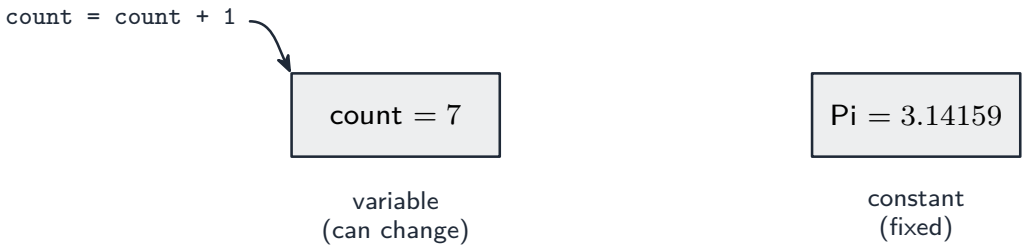
At AS we use variables carefully and break programs into reusable subroutines.

New ideas

Read these first. Each idea has a short worked example. Then do the Practice.

1 Variables, constants and assignment

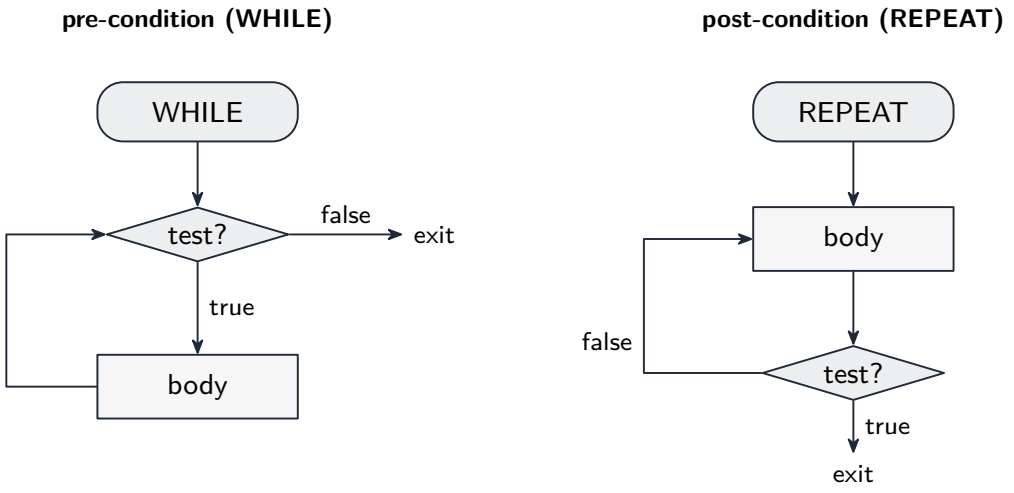
- a **variable** is a named store whose value can change; a **constant** keeps the same value all through the program.
- **assignment** puts a value into a variable, written with an arrow: `count ← count + 1`.



Worked example. `VAT ← 0.20` suits a constant (it rarely changes); `count ← count + 1` is assignment – it reads the old `count`, adds one, and stores the result back.

2 The three constructs

- **selection** (`IF` / `CASE`) chooses between paths.
- **iteration** repeats: a count-controlled loop (`FOR`) repeats a fixed number of times; a condition-controlled loop (`WHILE`) repeats until a condition changes.



An **array** is often processed with a `FOR` loop, one element at a time.

exercises.lol exercises.com.cn

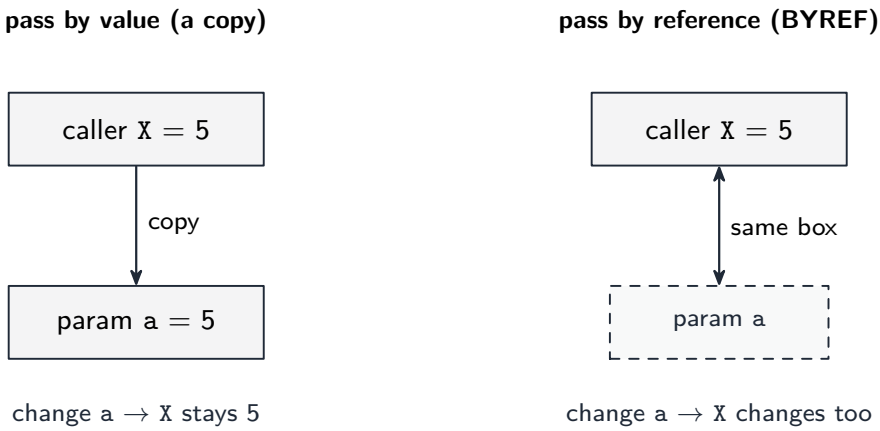
Worked example. To total 10 marks, use `FOR i ← 1 TO 10` (you know it is 10); to keep asking for a password, use `WHILE wrong` (you do not know how many tries it takes).

■ 3 Subroutines

Break a large program into named **subroutines** :

- a **procedure** carries out a job;
- a **function** carries out a job **and returns a value**.

Values passed into a subroutine are **parameters** .



A **local variable** exists only inside its own subroutine – that area is its **scope** .

Worked example. `area(w, h)` is a function: it takes parameters `w` and `h` and *returns* `w * h`; `printReport()` is a procedure: it does a job but returns nothing.

Watch out: a function *returns* a value you can use (`x ← area(3, 4)`); a procedure returns nothing, so `x ← printReport()` makes no sense.

Practice

11.1 State one difference between a variable and a constant. [2]

11.2 State what the assignment `total ← total + 5` does. [1]

11.3 State when you would use a `FOR` loop rather than a `WHILE` loop. [2]

11.4 State one difference between a procedure and a function. [2]

11.5 Explain what is meant by the scope of a local variable. [1]

11.6 Explain one advantage of breaking a large program into subroutines. [2]

11.7 Challenge. Write pseudocode for a function `bigger(a, b)` that returns the larger of the two numbers. [3]

exercises.lol exercises.com.cn