IGCSE COMPUTER SCIENCE • EXERCISE SHEET

6.1 Automated systems

Name: _____ Class: _____ Date: _____ **Total: 27 marks**

KEY WORDS sensor 传感器 • microprocessor 微处理器 • automated system 自动化系统 • actuator 执行器 • a rogue value 哨兵值

Objective

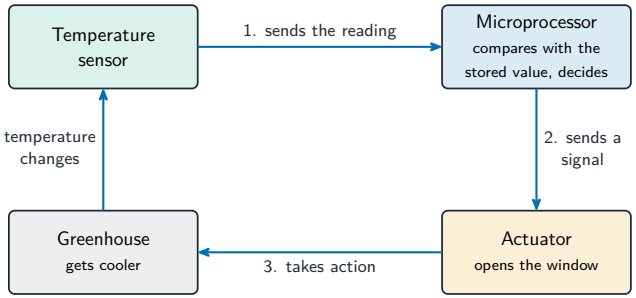
Build the skills to answer exam questions on **automated systems** 自动化系统 — the roles of **sensors** 传感器, **microprocessors** 微处理器 and **actuators** 执行器, and the pros and cons.

You must be able to:

- describe the role of the sensor, microprocessor and actuator
- describe the control loop
- give advantages and disadvantages of automated systems

1 Worked examples

The control loop



Sensor reads data → microprocessor compares with stored values → signals actuator → the loop repeats

- **sensor:** measures a quantity; **microprocessor:** decides; **actuator:** causes action.

Algorithms: what each one is for

Pseudocode or flowchart? Both describe the same algorithm. Pseudocode is quicker to write and translates straight into code; a flowchart shows the flow of control visually and is easier to follow at a glance.

The standard methods the exam expects:

- **totalling** 求和: `Total ← Total + Value` inside the loop, with `Total ← 0` before it.
- **counting** 计数: `Count ← Count + 1` each time a condition is met.
- **finding the largest**: set `Largest` to the first value (or a very small number), then replace it whenever a bigger value appears.
- **finding the smallest**: the same with the comparison reversed.
- **calculating an average**: total divided by the count, after the loop.
- **a rogue value** 哨兵值: a value outside the possible data that stops the loop, such as −1 for a mark out of 100. Test for it before adding, or the rogue value joins the total.

Worked example. Input marks until −1 is entered, then output the total and the average.

```
Total ← 0
Count ← 0
INPUT Mark
WHILE Mark <> -1 DO
    Total ← Total + Mark
    Count ← Count + 1
    INPUT Mark
ENDWHILE
IF Count > 0 THEN
    OUTPUT "Total ", Total
    OUTPUT "Average ", Total / Count
ELSE
    OUTPUT "No marks entered"
ENDIF
```

Two details earn marks: the first `INPUT` **before** the loop so the condition can be tested, and the check that `Count` is not zero before dividing.

Trace tables. Draw one column per variable and one per output, then work through the algorithm one line at a time, writing each new value on a new row. Never rub out an old value: the examiner wants to see the sequence.

2 Practice

2.1 State the role of a sensor in an automated system. [1]

2.2 State the role of an actuator. [1]

2.3 State what a rogue value is and give one example. [2]

2.4 Give the line of pseudocode that adds `Mark` to a running total. [1]

2.5 Explain why the total must be set to zero before the loop begins. [2]

3 Exam-style questions

3.1 In an automated greenhouse, the part that decides whether to open a window is the: [1]

- **A** sensor
- **B** microprocessor
- **C** actuator
- **D** window

3.2 A greenhouse is too hot.

(a) Describe the steps the system takes to cool it. [3]

(b) State one disadvantage of automated systems. [1]

3.3 A program inputs the temperatures recorded each day until the rogue value −99 is entered.

(a) State why −99 is a suitable rogue value here. [1]

(b) Write pseudocode that inputs the temperatures, and outputs the total, the average and the highest temperature. [7]

(c) Explain why the program must check the count before calculating the average. [2]

(d) Complete the trace table for the input data 12, 15, 9, −99. [4]

Mark	Total	Count	Highest	OUTPUT

(e) Give one advantage of writing this algorithm as a flowchart rather than as pseudocode. [1]

Solutions

2.1 it measures a physical quantity and sends the data.

2.2 it receives a signal and causes movement or action.

2.3 A value outside the range of the real data, used to stop a loop; for example −1 for a mark out of 100.

2.4 `Total ← Total + Mark`.

2.5 The variable would otherwise start with whatever value was left in it, so the total would be wrong from the first addition.

3.1 B. The microprocessor compares the data and decides.

3.2 (a) the temperature sensor sends the reading to the microprocessor; the microprocessor compares it with the stored value; it signals an actuator to open a window.
(b) any of: high cost; can break down; may replace jobs; cannot react to unplanned situations.

3.3 (a) A temperature of −99 °C never occurs in the real data, so it cannot be confused with a genuine reading.
(b) `Total ← 0`, `Count ← 0` and `Highest` set to the first value or a very low number; `INPUT Temp` before the loop; `WHILE Temp <> -99 DO … ENDWHILE`; `Total ← Total + Temp` and `Count ← Count + 1` inside; `IF Temp > Highest THEN Highest ← Temp ENDIF` inside; `INPUT Temp` as the last statement inside the loop; outputs of the total, `Total / Count` and `Highest` after the loop.
(c) If no temperatures were entered the count is zero, and dividing by zero would make the program crash.
(d) Row 1: Mark 12, Total 12, Count 1, Highest 12. Row 2: Mark 15, Total 27, Count 2, Highest 15. Row 3: Mark 9, Total 36, Count 3, Highest 15. Row 4: Mark −99, loop ends, OUTPUT 36, 12, 15.
(e) The flow of control, especially the branches and the loop, can be seen at a glance.