Exercise walk-through

Automated systems ▪ 6.1

eXercise

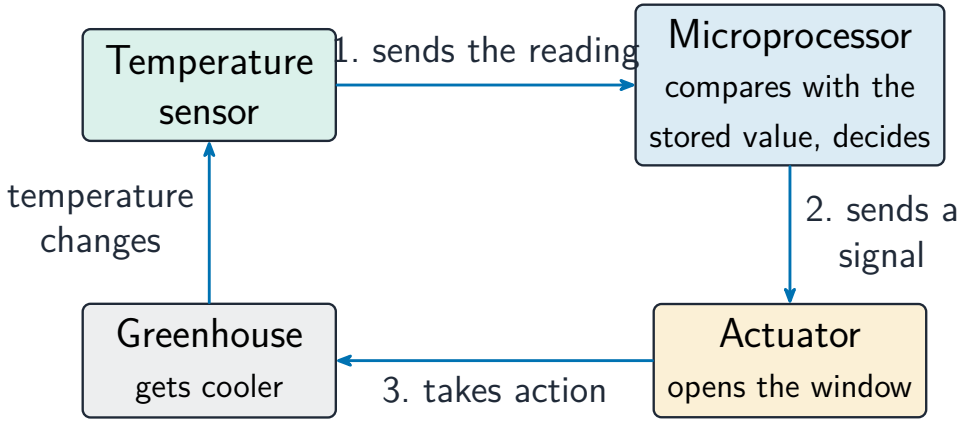
You must be able to

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

Method — The control loop

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

Method — The control loop



A greenhouse loop: sensor → microprocessor → actuator opens a window

Method — 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.

Method — Algorithms: what each one is for

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
```

Method — Algorithms: what each one is for

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.

Practice 2.1 ▪ 1 mark

State the role of a sensor in an automated system.

Solution 2.1

Worked steps

it measures a physical quantity and sends the data B1.

Practice 2.2 ▪ 1 mark

State the role of an actuator.

Solution 2.2

Worked steps

it receives a signal and causes movement or action B1.

Practice 2.3 ▪ 2 marks

State what a rogue value is and give one example.

Solution 2.3

Method: Algorithms: what each one is for

Worked steps

A value outside the range of the real data, used to stop a loop **B1**; for example -1 for a mark out of 100 **B1**.

Practice 2.4 ▪ 1 mark

Give the line of pseudocode that adds `Mark` to a running total.

Solution 2.4

Method: Algorithms: what each one is for

Worked steps

`Total ← Total + Mark` B1.

Practice 2.5 ▪ 2 marks

Explain why the total must be set to zero before the loop begins.

Solution 2.5

Method: Algorithms: what each one is for

Worked steps

The variable would otherwise start with whatever value was left in it B1, so the total would be wrong from the first addition B1.

Exam-style 3.1 ▪ 1 mark

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

A sensor

B microprocessor

C actuator

D window

Solution 3.1

Method: The control loop

A sensor

B microprocessor ✓

C actuator

D window

Worked steps

B. The microprocessor compares the data and decides.

Exam-style 3.2 ▪ 3 marks

A greenhouse is too hot.
(a) Describe the steps the system takes to cool it.
(b) State one disadvantage of automated systems.

Solution 3.2

Worked steps

(a) the temperature sensor sends the reading to the microprocessor M1; the microprocessor compares it with the stored value M1; it signals an actuator to open a window A1.

(b) any of: high cost; can break down; may replace jobs; cannot react to unplanned situations B1.

Exam-style 3.3 ▪ 1 mark

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.
(b) Write pseudocode that inputs the temperatures, and outputs the total, the average and the highest temperature.
(c) Explain why the program must check the count before calculating the average.
(d) Complete the trace table for the input data 12, 15, 9, -99.

Mark	Total	Count	Highest	OUTPUT

Exam-style 3.3 ▪ 1 mark

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

Solution 3.3

Method: Algorithms: what each one is for

Worked steps

(a) A temperature of -99 °C never occurs in the real data, so it cannot be confused with a genuine reading B1.
(b) `Total` ← `0`, `Count` ← `0` and `Highest` set to the first value or a very low number B1; `INPUT Temp` before the loop B1; `WHILE Temp <> -99 DO … ENDWHILE` B1; `Total ← Total + Temp` and `Count ← Count + 1` inside B1; `IF Temp > Highest THEN Highest ← Temp ENDIF` inside B1; `INPUT Temp` as the last statement inside the loop B1; outputs of the total, `Total / Count` and `Highest` after the loop B1.
(c) If no temperatures were entered the count is zero B1, and dividing by zero would

Solution 3.3

make the program crash B1.
(d) Row 1: Mark 12, Total 12, Count 1, Highest 12 B1. Row 2: Mark 15, Total 27, Count 2, Highest 15 B1. Row 3: Mark 9, Total 36, Count 3, Highest 15 B1. Row 4: Mark -99, loop ends, OUTPUT 36, 12, 15 B1.
(e) The flow of control, especially the branches and the loop, can be seen at a glance B1.