

Exercise walk-through

11.2.2

Count-controlled loops: FOR ... NEXT, totals and counts

A-Level Computer Science

You must be able to

- write a FOR ... NEXT loop, with and without STEP
- state how many times a loop runs, and what it outputs
- use a loop to find a **running total** 累计总和, a **count**, a mean, and the largest or smallest value
- put an IF statement inside a loop
- write and follow **nested loops** 嵌套循环

1

The method

Method — The FOR ... NEXT loop

Iteration 迭代 means repeating statements. When the number of repetitions is **known before the loop starts**, use a count-controlled loop.

- The **counter** (here Count) is an INTEGER variable. The loop gives it each value from the start value to the end value, one after the other.
- FOR Count ← 1 TO 5 runs **5** times, and FOR Count ← 0 TO 5 runs **6** times. In general the number of passes is $end - start + 1$.
- Write the counter after NEXT. It shows which loop ends there.

STEP changes the counter by a different amount:

Method — The FOR ... NEXT loop

```
FOR Number ← 10 TO 0 STEP -2  
    OUTPUT Number  
NEXT Number
```

This outputs 10, 8, 6, 4, 2, 0. Without STEP, the counter goes up by 1.

Method — The FOR ... NEXT loop

Iteration 迭代 means repeating statements.

```
FOR Count ← 1 TO 5
```

```
    OUTPUT Count
```

```
NEXT Count
```

the **counter** starts at 1 and ends at 5,
so the loop runs **5 times**

the **body**: it runs once for each value of the counter

adds 1 to the counter and goes back to FOR;
after 5 the loop ends

Method — Loop patterns: a total and a count

```
DECLARE Day, BelowZero : INTEGER
DECLARE Temp, Total : REAL
Total ← 0
BelowZero ← 0
FOR Day ← 1 TO 30
    INPUT Temp
    Total ← Total + Temp
    IF Temp < 0 THEN
        BelowZero ← BelowZero + 1
    ENDIF
NEXT Day
OUTPUT BelowZero
OUTPUT Total / 30
```

Method — Loop patterns: a total and a count

- **Running total:** set it to 0 **before** the loop, and add to it **inside** the loop.
- **Count:** set it to 0 before the loop, and add 1 inside an IF.
- **Output after the loop.** An OUTPUT inside the loop would run 30 times.
- If `Total ← 0` were inside the loop, the total would go back to 0 on every pass. This is a very common error.

Method — Loop pattern: the largest value

```
INPUT Highest
FOR Day ← 2 TO 30
    INPUT Temp
    IF Temp > Highest THEN
        Highest ← Temp
    ENDIF
NEXT Day
OUTPUT Highest
```

Method — Loop pattern: the largest value

- Keep the **largest value so far** in a variable. Each new value replaces it only if it is greater.
- Start with the **first value**, not with 0: every temperature could be below zero. The loop then runs from **2**.
- For the smallest value, change $>$ to $<$.

Method — Nested loops

```
INPUT Rows
FOR Row ← 1 TO Rows
  Line ← ""
  FOR Star ← 1 TO Row
    Line ← Line & "*"
  NEXT Star
  OUTPUT Line
NEXT Row
```

Method — Nested loops

- The **inner** loop runs completely for **each** pass of the **outer** loop.
- The two loops need **different** counters, and the inner loop must end (NEXT Star) before the outer loop ends (NEXT Row).
- When Rows is 3, the statement inside the inner loop runs $1 + 2 + 3 = 6$ times.

2

Practice

Practice 2.1 ■ 3 marks

State how many times each loop runs.

Practice 2.1 (a)

```
FOR Index  $\leftarrow$  1 TO 20
```

Solution 2.1 (a)

Method: The FOR ... NEXT loop — p.4

Worked steps

$20 - 1 + 1 = \mathbf{20}$ times **B1**

Practice 2.1 (b)

```
FOR Index  $\leftarrow$  0 TO 9
```

Solution 2.1 (b)

Worked steps

$$9 - 0 + 1 = \mathbf{10} \text{ times } \text{B1}$$

Practice 2.1 (c)

```
FOR Index  $\leftarrow$  5 TO 25 STEP 5
```

Solution 2.1 (c)

Worked steps

the counter takes the values 5, 10, 15, 20, 25: **5** times **B1**

Practice 2.2 ■ 2 marks

State what each loop outputs.

Practice 2.2 (a)

```
FOR Count ← 3 TO 6  
    OUTPUT Count * 2  
NEXT Count
```

Solution 2.2 (a)

Method: The FOR ... NEXT loop — p.4

Worked steps

the counter is 3, 4, 5, 6, and each is doubled: **6, 8, 10, 12** **B1**

Practice 2.2 (b)

```
FOR Count ← 9 TO 1 STEP -3  
    OUTPUT Count  
NEXT Count
```

Solution 2.2 (b)

Worked steps

the counter goes down by 3 each time: **9, 6, 3** B1

Practice 2.3 ■ 2 marks

Write a FOR loop that outputs the **even** numbers from 2 to 20.

Solution 2.3

Method: The FOR ... NEXT loop — p.4

Worked steps

■ FOR ... STEP 2 from 2 to 20 M1 A1

```
FOR i ← 2 TO 20 STEP 2  
  OUTPUT i  
NEXT i
```

Solution 2.3

- *(outputting i only when $i \bmod 2 = 0$ also scores)*

Practice 2.4 ■ 4 marks

Complete the pseudocode so that it inputs 10 values and outputs their mean. Write what goes in each blank.

```
Total ← .....(1)
FOR Index ← 1 TO 10
    INPUT Value
    Total ← .....(2)
NEXT .....(3)
OUTPUT .....(4)
```

Solution 2.4

Worked steps

- (1) the total starts at 0 **B1**
- (2) each value is added: `Total + Value` **B1**
- (3) the counter of this loop: `Index` **B1**
- (4) the mean of 10 values: `Total / 10` **B1**

Practice 2.5 ■ 2 marks

A student wants to add up five prices, but the output is always the last price that was input.

```
FOR Item ← 1 TO 5
  Total ← 0
  INPUT Price
  Total ← Total + Price
NEXT Item
OUTPUT Total
```

Practice 2.5 ■ 2 marks

Explain the error, and state how to correct it.

Solution 2.5

Method: Loop patterns: a total and a count — p.7

Worked steps

- `Total ← 0` is **inside** the loop, so the total goes back to 0 on every pass, and only the last price is left [B1](#)
- move `Total ← 0` to the line **before** `FOR` [B1](#)

Practice 2.6 ■ 4 marks

Write pseudocode that inputs the heights of 25 students, one at a time, and outputs the smallest height.

Solution 2.6

Method: The FOR ... NEXT loop — p.4

Worked steps

- input the first height and store it as the smallest **B1**
- a loop that runs for the other 24 heights **B1**
- compare each height with the smallest, and replace it if smaller **B1**
- output the smallest after the loop **B1**

Solution 2.6

```
DECLARE Count : INTEGER
DECLARE Height, Smallest : REAL
INPUT Height
Smallest ← Height
FOR Count ← 2 TO 25
    INPUT Height
    IF Height < Smallest THEN
        Smallest ← Height
    ENDIF
NEXT Count
OUTPUT Smallest
```

3

Exam-style

Exam-style 3.1 ■ 5 marks

Write pseudocode that inputs an integer `Table`, and then outputs the first 12 lines of its multiplication table. For the input 7, the first two lines are:

```
1 x 7 = 7  
2 x 7 = 14
```

Declare your variables.

Solution 3.1

Method: Loop patterns: a total and a count — p.7

Worked steps

- the variables declared, and INPUT Table **B1**
- a FOR loop from 1 to 12, closed with NEXT **B1**
- the product Line * Table worked out inside the loop **B1**
- the OUTPUT has the values and the text in the correct order **B1**
- the text " x " and " = " is in quotes, with commas between the items **B1**

Solution 3.1

```
DECLARE Table, Line : INTEGER
INPUT Table
FOR Line ← 1 TO 12
    OUTPUT Line, " x ", Table, " = ", Line * Table
NEXT Line
```

Exam-style 3.2 ■ 6 marks

A swimming club times 40 swimmers over 100 metres. An algorithm will:

- input the time of each swimmer, in seconds, one value at a time
- count the swimmers with a time of less than 60 seconds
- output this count, and the mean time of all the swimmers.

Write pseudocode for the algorithm. Declare your variables.

Solution 3.2

Method: Loop patterns: a total and a count — p.7

Worked steps

- the variables declared with suitable types **B1**
- the total and the count set to 0 before the loop **B1**
- a FOR loop that runs 40 times, with the input inside it **B1**
- each time added to the total **B1**
- an IF that adds 1 to the count when the time is less than 60 **B1**
- the count and Total / 40 output **after** the loop **B1**

Solution 3.2

```
DECLARE Swimmer, Fast : INTEGER
DECLARE Time, Total : REAL
Total ← 0
Fast ← 0
FOR Swimmer ← 1 TO 40
    INPUT Time
    Total ← Total + Time
    IF Time < 60 THEN
        Fast ← Fast + 1
    ENDIF
NEXT Swimmer
OUTPUT Fast
OUTPUT Total / 40
```

Exam-style 3.3 ■ 7 marks

A weather station records the rainfall, in mm, for each of the 30 days of a month. An algorithm will:

- input the rainfall for each day, one value at a time
- count the days with no rain
- find the total rainfall for the month
- find the day with the most rain; if two days have the same rainfall, keep the earlier day
- output the count, the total and the day number.

Write pseudocode for the algorithm. Declare any variables you use.

Solution 3.3

Method: Loop patterns: a total and a count — p.7

Worked steps

- declarations (B1); a count-controlled loop from 1 to 30 (B1); input inside the loop (B1)
- count the days with rainfall 0 (B1); running total (B1)
- keep the largest rainfall and its day number, replacing only when greater (B1)
- output all three after the loop (B1)

Solution 3.3

```
DECLARE Day, DryDays, WettestDay : INTEGER
DECLARE Rain, Total, MostRain : REAL
DryDays ← 0
Total ← 0
MostRain ← -1
WettestDay ← 0
```

Solution 3.3

```
FOR Day ← 1 TO 30
  INPUT Rain
  IF Rain = 0 THEN
    DryDays ← DryDays + 1
  ENDIF
  Total ← Total + Rain
  IF Rain > MostRain THEN
    MostRain ← Rain
    WettestDay ← Day
  ENDIF
NEXT Day
OUTPUT DryDays
OUTPUT Total
OUTPUT WettestDay
```

- MostRain starts at -1, below any real rainfall, so day 1 is always stored first; > rather than >= keeps the earlier day

Exam-style 3.4 ■ 7 marks

Study this pseudocode.

```
FOR Outer ← 1 TO 3
  FOR Inner ← 1 TO 4
    OUTPUT Outer * Inner
  NEXT Inner
NEXT Outer
```

Exam-style 3.4 (a) ■ 1 mark

State how many values are output.

Solution 3.4 (a)

Method: Nested loops — p.11

Worked steps

the inner loop runs 4 times for each of the 3 passes of the outer loop: $3 \times 4 = 12$
values **B1**

Exam-style 3.4 (b) ■ 2 marks

State the first **five** values that are output.

Solution 3.4 (b)

Worked steps

the first pass of the outer loop gives 1×1 to 1×4 , and the second pass starts with 2×1 **M1**

■ 1, 2, 3, 4, 2 **A1**

Exam-style 3.4 (c) ■ 4 marks

The programmer wants each pass of the **outer** loop to produce one line of output, for example 1 2 3 4. Rewrite the pseudocode. Use a string variable `Line`, and the function `NUM_TO_STR()` to change a number into a string.

Solution 3.4 (c)

Worked steps

Line is set to an empty string at the start of **each** pass of the outer loop B1

- inside the inner loop, the product is changed to a string and joined to Line with &
B1
- a space is joined between the values B1
- OUTPUT Line comes after NEXT Inner and before NEXT Outer B1

Solution 3.4 (c)

```
FOR Outer ← 1 TO 3
  Line ← ""
  FOR Inner ← 1 TO 4
    Line ← Line & NUM_TO_STR(Outer * Inner) & " "
  NEXT Inner
  OUTPUT Line
NEXT Outer
```