CAMBRIDGE IGCSE (0478)

Module M: Programming

Workbooklet · 21 parts · 447 marks

M27 is GCSE (OCR) only and not on 0478.

Contents

Part	Topic	Marks
M1	Variables, constants, input and output	22
M3	Data types	20
M5	Arithmetic operators, including MOD, DIV and exponentiation	21
M6	Relational (comparison) operators	21
M7	Boolean operators AND, OR and NOT in code	22
M8	Selection: IF, ELSE IF and ELSE	20
M9	Selection: CASE statements	21
M10	Count-controlled loops (FOR)	22
M11	Pre-condition loops (WHILE)	21
M12	Post-condition loops, and choosing the right loop	22
M14	Nested selection and nested loops	22
M16	String handling: length, substring, upper and lower case	22
M18	One-dimensional arrays	22
M19	Two-dimensional arrays and looping through them	21
M21	File handling: open, read, write and close	20
M22	Procedures	22
M23	Functions and return values	22
M24	Parameters and arguments	20
M25	Local and global variables	22
M26	Library routines and random numbers	22
M28	Writing maintainable code: identifiers, comments and sub programs	20
	Total	**447**

MODULE M PROGRAMMING · M1

Variables, constants, input and output

Original practice questions · 22 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.1, 8.1.3, 8.1.4(a)

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **22 marks**. Pseudocode on this sheet follows the Cambridge 0478 pseudocode guide.

1

Total: 3 marks

Complete the table by writing the correct term for each description. Choose each term from this list. Not all the terms are used. **[3]**

Constant | Input | Output | Sequence | Variable

Description	Term
A named value that is set once and cannot change while the program is running	
Statements are carried out one after another, in the order they are written	
A named memory location that stores a value that can change while the program is running	

2 **Total: 5 marks**

A river boat company uses this algorithm to work out the cost of adult tickets for a trip.

```
01 CONSTANT TicketPrice ← 7.50
02 DECLARE Adults : INTEGER
03 DECLARE TotalCost : REAL
04 OUTPUT "How many adult tickets?"
05 INPUT Adults
06 TotalCost ← Adults * TicketPrice
07 OUTPUT "Total to pay: ", TotalCost
```

(a) State the identifier of the constant in the algorithm. **[1]**

(b) Give the identifier of **one** variable in the algorithm. **[1]**

(c) The input is 4. Write down the output from line 07. **[1]**

(d) Give **two** reasons why the programmer used a constant for the ticket price instead of writing 7.50 in line 06. **[2]**

1

2

3 **Total: 5 marks**

Look at this algorithm.

```
01 DECLARE Points : INTEGER
02 DECLARE Bonus : INTEGER
03 Points ← 3
04 INPUT Bonus
05 Points ← Points + Bonus
06 Bonus ← Points * 2
07 Points ← 1
08 OUTPUT Points, " and ", Bonus
```

(a) Complete the trace table for the algorithm using the input data: `6` **[3]**

Points	Bonus	OUTPUT

(b) `Points` stores 9 after line 05. Explain why the output does **not** show the value 9 for `Points`. **[2]**

4 **Total: 4 marks**

A storage shelf holds up to 60 boxes. This algorithm should ask a worker for the number of boxes on the shelf, store the number of free spaces in `FreeSpace`, and then output it with a message.

```
01 CONSTANT ShelfLimit ← 60
02 DECLARE Stored : INTEGER
03 DECLARE FreeSpace : INTEGER
04 OUTPUT "Enter the number of boxes on the shelf"
05 INPUT "Enter the number of boxes on the shelf", Stored
06 ShelfLimit ← ShelfLimit - Stored
07 OUTPUT "Free spaces: ", FreeSpace
```

Identify the line numbers of the **two** errors in the algorithm and suggest a correction for each error. **[4]**

Line number	Correction

5 **Total: 5 marks**

A garden centre sells compost at a fixed price of 4.25 for each bag.

Write an algorithm using pseudocode that: **[5]**

- uses a constant for the price of one bag, 4.25
- declares the variables that are needed
- asks the user to input the number of bags wanted and stores it
- calculates the total cost
- outputs the total cost with a suitable message.

Total for this worksheet: 22 marks

MODULE M PROGRAMMING · M3

Data types

Original practice questions · 20 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.2 (with 8.1.1)

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **20 marks**. Pseudocode on this sheet follows the Cambridge 0478 pseudocode rules. Write any pseudocode answers in the same style.

1 **Total: 5 marks**

Tick (✓) **one** box in each row to show the data type of each literal value, written as it would appear in pseudocode. **[5]**

Literal value	INTEGER	REAL	CHAR	STRING	BOOLEAN
`'Q'`					
`"Q12"`					
`-12`					
`12.0`					
`FALSE`					

2 **Total: 5 marks**

A museum's ticket machine uses a program. The table shows five of the variables it uses.

Identifier	What it stores
`AdultTickets`	the number of adult tickets bought, for example 3
`TotalPrice`	the total price to pay, for example 37.50
`IsMember`	whether the buyer is a museum member or not
`PriceBand`	a single letter for the price band: A, B or C
`VisitorName`	the name to print on the ticket

Write pseudocode statements to declare each of these variables with a suitable data type. **[5]**

3 **Total: 3 marks**

Another part of the museum program declares five variables and then gives each one a value.

```
01 DECLARE Visitors : INTEGER
02 DECLARE Fee : REAL
03 DECLARE Band : CHAR
04 DECLARE IsOpen : BOOLEAN
05 DECLARE Venue : STRING
06 Visitors ← 0
07 IsOpen ← "TRUE"
08 Venue ← "East Hall"
09 Fee ← 12
10 Band ← "C"
11 OUTPUT Venue, " entry fee: ", Fee
```

Three of the lines from 06 to 10 do not write the literal value in the way the pseudocode rules require for the data type of the variable. **[3]**
Identify the line number of each of these three lines and write the corrected line.

Line number ..... Correction

Line number ..... Correction

Line number ..... Correction

4 **Total: 4 marks**

A delivery robot's program stores data about each delivery.

(a) The robot's battery charge is stored as a percentage, for example 87.5 **[2]**
State the most suitable data type for the battery charge and give a reason for your choice.

Data type

Reason

(b) Each delivery has a door entry code made of four digits, for example 0472 **[2]**
State the most suitable data type for the door entry code and give a reason for your choice.

Data type

Reason

5 **Total: 3 marks**

The robot's program contains these declarations:

```
DECLARE Parcels : INTEGER
DECLARE Street : STRING
```

State the data type of the value produced by each expression. **[3]**

Expression	Data type
`Parcels > 20`	
`LENGTH(Street)`	
`SUBSTRING(Street, 1, 4)`	

Total for this worksheet: 20 marks

MODULE M PROGRAMMING · M5

Arithmetic operators, including MOD, DIV and exponentiation

Original practice questions · 21 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.4(f), 8.1.7

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **21 marks**. All algorithms on this sheet are written in pseudocode. Write your own algorithms in pseudocode. No calculator.

1

Total: 4 marks

(a) Complete the table to show the value assigned to `Result` by each pseudocode statement. **[4]**

Statement	Value of Result
`Result ← MOD(23, 4)`	
`Result ← DIV(23, 4)`	
`Result ← 2 ^ 5`	
`Result ← 15 / 4`	

2

Total: 4 marks

A car ferry parks cars in lanes. Each lane holds 9 cars. The lanes are filled one at a time. The integer variable `Cars` stores the number of cars waiting to board.

(a) Write a pseudocode statement that stores in `FullLanes` the number of lanes that will be completely full. **[1]**

(b) Write a pseudocode statement that stores in `LastLane` the number of cars in the lane that is only partly full. **[1]**

(c) State the values stored in `FullLanes` and `LastLane` when `Cars` is 58. **[2]**

FullLanes .....

LastLane .....

3 **Total: 5 marks**

A ticket office uses this algorithm to carry out a quick check on a ticket number before it is accepted. `Number`, `Digit` and `Total` are integers.

```
01 INPUT Number
02 Total ← 0
03 WHILE Number > 0 DO
04     Digit ← MOD(Number, 10)
05     Total ← Total + Digit
06     Number ← DIV(Number, 10)
07 ENDWHILE
08 IF MOD(Total, 3) = 0
09   THEN
10     OUTPUT "Accepted"
11   ELSE
12     OUTPUT "Rejected"
13 ENDIF
```

(a) Complete the trace table for the algorithm using this input data: **[4]**

4071

Number	Digit	Total	OUTPUT

(b) State the output of the algorithm when the input is **76**. **[1]**

4 **Total: 3 marks**

A dance competition has three judges. A programmer writes this statement to find the mean of their three scores.

```
Mean ← Score1 + Score2 + Score3 / 3
```

(a) The scores are `Score1` = 6, `Score2` = 9 and `Score3` = 12. State the value that the statement assigns to `Mean`. **[1]**

(b) Explain why the statement does not calculate the mean correctly. **[1]**

(c) Rewrite the statement so that it calculates the mean correctly. **[1]**

5 **Total: 5 marks**

A warehouse sends each parcel to one of two loading bays. Parcels with an even parcel number go to bay A.

(a) Write an algorithm using pseudocode to: **[5]**

- input 10 parcel numbers, one at a time (each parcel number is a whole number)
- use `MOD` to decide whether each parcel number is even
- count how many of the parcel numbers are even
- output the count, with a suitable message, after all 10 parcel numbers have been input.

You do not need to declare any variables.

Total for this worksheet: 21 marks

MODULE M PROGRAMMING · M6

Relational (comparison) operators

Original practice questions · 21 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.4(f), 8.1.4(b), 8.1.4(c)

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. All code on this sheet is written in pseudocode. Write all code answers in pseudocode. The marks for each part are shown in brackets [ ]. The total for this worksheet is **21 marks**.

1

Total: 4 marks

(a) Complete the table. Write the meaning of each relational operator given, or the relational operator for the meaning given. **[3]**

Relational operator	Meaning
<>	
	Less than or equal to
=	

(b) State the data type of the result of a relational operation, such as `Price < 10`. **[1]**

2

Total: 4 marks

A booking algorithm for a coach company has assigned these values.

```
Seats ← 40
Booked ← 40
Row ← 'C'
Groups ← 6
```

(a) Complete the table to show whether each expression is **TRUE** or **FALSE**. **[4]**

Expression	TRUE or FALSE
`Booked < Seats`	
`Booked <= Seats`	
`Row <> 'D'`	
`Groups * 6 > Seats`	

3 **Total: 3 marks**

Write a condition in pseudocode for each rule. Use the identifier given in bold.

(a) A bus is full when the number of passengers, **Passengers**, is 72 or more. **[1]**

(b) A tablet shows a battery warning when the charge, **Charge**, is less than 15. **[1]**

(c) A login screen shows an error when the password entered, stored as a string in **Password**, is not Kestrel9. **[1]**

4 **Total: 3 marks**

An algorithm controls a fridge in a sandwich shop. It should:

- output **Too warm** if the temperature is 8 °C **or higher**, otherwise output **Temperature OK**
- run the cooler **while** the temperature is **above** 4 °C. Each time the cooler runs, the temperature goes down by 1
- output **Close the door** if the door status is **not** 'C' (closed).

```
01 DECLARE Temp : INTEGER
02 DECLARE Door : CHAR
03 INPUT Temp
04 IF Temp > 8
05   THEN
06     OUTPUT "Too warm"
07   ELSE
08     OUTPUT "Temperature OK"
09 ENDIF
10 WHILE Temp < 4 DO
11     OUTPUT "Cooler on"
12     Temp ← Temp - 1
13 ENDWHILE
14 INPUT Door
15 IF Door != 'C'
16   THEN
17     OUTPUT "Close the door"
18 ENDIF
```

(a) There are **three** errors in the algorithm. Identify the line number of each error and suggest a correction. **[3]**

Line number	Correction

5 **Total: 3 marks**

A wind turbine checks the wind speed, in km/h.

```
01 INPUT Speed
02 IF Speed > 60
03   THEN
04     OUTPUT "Stop blades"
05   ELSE
06     IF Speed < 25
07       THEN
08         OUTPUT "Too little wind"
09       ELSE
10         OUTPUT "Generating"
11     ENDIF
12 ENDIF
```

(a) Complete the table to show the output of the algorithm for each input. **[3]**

Input	OUTPUT
60	
25	
24	

6 **Total: 4 marks**

A farm uses an algorithm to check the water level, in cm, in a cattle trough.

(a) Write an algorithm using pseudocode to: **[4]**

- input the water level
- output **Overflow warning** if the level is more than 180
- output **Refill needed** if the level is 20 or less
- output **Level normal** for any other level.

Only one message should be output. You do not need to declare any variables.

Total for this worksheet: 21 marks

MODULE M PROGRAMMING · M7

Boolean operators AND, OR and NOT in code

Original practice questions · 22 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.4(f), 7.8, 7.9

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **22 marks**. Write all algorithms and conditions in pseudocode. All data on this sheet is made up.

1

Total: 3 marks

(a) Complete the table by writing the logical operator (`AND`, `OR` or `NOT`) that matches each description. **[3]**

Description	Logical operator
The result is TRUE when either one or both of the two conditions are TRUE.	
The result is TRUE only when both conditions are TRUE.	
The result is the opposite of the single Boolean value it is applied to.	

2

Total: 3 marks

A greenhouse controller has just carried out these statements:

```
Humidity ← 64
FanOn ← TRUE
Level ← 3
```

(a) Complete the table to show whether each expression evaluates to TRUE or FALSE. **[3]**

Expression	TRUE or FALSE
`(Humidity > 60) AND (Level = 3)`	
`(NOT FanOn) OR (Level > 5)`	
`NOT ((Humidity >= 64) AND FanOn)`	

3 **Total: 4 marks**

The temperature setting of a cold-store unit is stored as an integer in the variable `Setting`. A setting is accepted only if it is from −25 to −15 inclusive.

(a) Write a condition, in pseudocode, that is TRUE when the setting is accepted. **[2]**

(b) Write a condition, in pseudocode, using the **OR** operator, that is TRUE when the setting is **not** accepted. **[2]**

4 **Total: 6 marks**

A bus station display uses this algorithm to choose which timetable to show. It should show:

- the weekend timetable when `Day` is `"Sat"` or `"Sun"`
- on other days, the peak timetable when `Hour` is from 6 to 9 inclusive, and the off-peak timetable at all other hours.

The algorithm contains **two** errors.

```
01 INPUT Day
02 INPUT Hour
03 IF Day = "Sat" AND Day = "Sun"
04   THEN
05     OUTPUT "Weekend timetable"
06   ELSE
07     IF Hour >= 6 AND <= 9
08       THEN
09         OUTPUT "Peak timetable"
10       ELSE
11         OUTPUT "Off-peak timetable"
12     ENDIF
13 ENDIF
```

(a) Identify the line number of each error and suggest a correction. **[4]**

Error 1 Line number ..... Correction

Error 2 Line number ..... Correction

(b) Explain why, before it is corrected, the algorithm never outputs `Weekend timetable`. **[2]**

5 **Total: 6 marks**

A survey drone checks whether it is safe to take off. The BOOLEAN variable `Raining` has already been set to TRUE or FALSE by a sensor.

(a) Write an algorithm using pseudocode to: **[6]**

- input the battery charge as a whole number, and the wind speed
- output `Take-off allowed` if the battery charge is 40 or more, the wind speed is less than 30 **and** it is **not** raining
- otherwise output `Take-off blocked`.

You do not need to declare any variables.

Total for this worksheet: 22 marks

MODULE M PROGRAMMING · M8

Selection: IF, ELSE IF and ELSE

Original practice questions · 20 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.4(b), 8.1.5, 7.7, 7.8, 7.9

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **20 marks**. All algorithms on this sheet are written in pseudocode. Write your own algorithms in pseudocode.

1

Total: 2 marks

(a) Tick (✓) **one** box to show which statement describes **selection**. **[1]**

	Statement	Tick (✓)
A	The statements are run one after another, in the order they are written.	
B	A condition is tested and the result decides which statements are run next.	
C	A group of statements is repeated a set number of times.	
D	A value is checked to see if it is reasonable before it is accepted.	

(b) An IF statement has a THEN clause and an ELSE clause. **[1]**
State when the statements in the ELSE clause are run.

2

Total: 2 marks

A delivery robot stores its battery charge, as a percentage, in the variable `Charge`.

(a) Write a pseudocode IF statement that outputs `Return to base` when `Charge` is less than 15. **[2]**
Nothing is output when `Charge` is 15 or more.

3 **Total: 5 marks**

A pump fills a water tank. This algorithm sets the pump speed from the water level in the tank, as a percentage.

```
01 INPUT Level
02 IF Level > 90
03   THEN
04     OUTPUT "Pump off"
05   ELSE
06     IF Level >= 40
07       THEN
08         OUTPUT "Pump low"
09       ELSE
10         OUTPUT "Pump high"
11     ENDIF
12 ENDIF
```

(a) Complete the table to show the output for each input value. **[4]**

Level	OUTPUT
90	
95	
40	
12	

(b) Explain why the condition on line 06 is **not** tested when the input is 95. **[1]**

4 **Total: 5 marks**

A shop gives each customer a loyalty level from the points they have collected:

- 250 points or more: `Gold`
- 100 to 249 points: `Silver`
- fewer than 100 points: `Standard`

This algorithm should output the correct level. It contains an error.

```
01 INPUT Points
02 IF Points >= 100
03   THEN
04     Level ← "Silver"
05   ELSE
06     IF Points >= 250
07       THEN
08         Level ← "Gold"
09       ELSE
10         Level ← "Standard"
11     ENDIF
12 ENDIF
13 OUTPUT Level
```

(a) State the output when the input is 300. **[1]**

(b) Explain why the algorithm never outputs `Gold`. **[1]**

(c) Rewrite lines 02 to 12 using pseudocode so that every customer is given the correct level. **[3]**

5 **Total: 6 marks**

At a ferry port, vehicles queue in different lanes depending on their height, in metres.

(a) Write an algorithm using pseudocode to: **[6]**

- input the height of a vehicle
- output `Lane A` if the height is up to and including 2.0
- output `Lane B` if the height is more than 2.0 and up to and including 3.5
- output `Lane C` if the height is more than 3.5 and up to and including 4.5
- output `Too tall` if the height is more than 4.5.

Only **one** message must be output for each vehicle. Use IF statements. You do not need to declare any variables.

Total for this worksheet: 20 marks

MODULE M PROGRAMMING · M9

Selection: CASE statements

Original practice questions · 21 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.4(b) (with 7.8, 7.9)

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **21 marks**. Pseudocode on this sheet follows the Cambridge 0478 pseudocode rules. Write any pseudocode answers in the same style. All data on this sheet is made up.

1 **Total: 3 marks**

Tick (✓) **one** box in each row to show whether each statement about a pseudocode CASE statement is true or false. **[3]**

Statement	True	False
When a case matches, its statement is executed and the remaining cases are not tested.		
Every CASE statement must include an OTHERWISE clause.		
If an OTHERWISE clause is present, its statement is executed only when none of the cases before it apply.		

2 **Total: 5 marks**

This algorithm sets the brightness of the lights in a meeting room from a scene letter.

```
01 DECLARE Scene : CHAR
02 DECLARE Brightness : INTEGER
03 Brightness ← 50
04 OUTPUT "Enter scene letter"
05 INPUT Scene
06 CASE OF Scene
07   'R' : Brightness ← 30
08   'W' : Brightness ← 100
09   'N' : Brightness ← Brightness - 40
10   OTHERWISE OUTPUT "Scene not found"
11 ENDCASE
12 OUTPUT "Brightness is ", Brightness
```

(a) Complete the table to show the output(s) after the prompt for each input. **[3]**

Input	Output(s)
W	
N	
r	

(b) The user enters R. Describe what happens in the algorithm from line 06 onwards. **[2]**

3 **Total: 4 marks**

A bicycle repair shop uses this algorithm to set the cost of a service from a service code. `ServiceCode` is of type CHAR and `Cost` is of type INTEGER.

```
IF ServiceCode = 'B'
  THEN
    Cost ← 25
  ELSE
    IF ServiceCode = 'F'
      THEN
        Cost ← 60
      ELSE
        IF ServiceCode = 'W'
          THEN
            Cost ← 15
          ELSE
            OUTPUT "Unknown service"
        ENDIF
    ENDIF
ENDIF
```

Rewrite this algorithm, using pseudocode, with a CASE statement instead of the IF statements. Your algorithm must give the same result for every value of `ServiceCode`. **[4]**

4 **Total: 4 marks**

A lift in an office building shows a message when a floor button is pressed:

- button 0 shows `Ground floor: reception`
- button 1 shows `First floor: library`
- button 2 shows `Second floor: offices`
- any other number shows `Button not in use`.

This pseudocode algorithm has **four** errors.

```
01 DECLARE Button : INTEGER
02 DECLARE Message : INTEGER
03 OUTPUT "Press a button from 0 to 2"
04 INPUT Button
05 CASE Button
06   0 : Message ← "Ground floor: reception"
07   1 : Message ← "First floor: library"
08   OTHERWISE Message ← "Button not in use"
09   2 : Message ← "Second floor: offices"
10 ENDIF
11 OUTPUT Message
```

Identify the line number of each error and suggest a correction for each error. **[4]**

Error 1 Line number ..... Correction

Error 2 Line number ..... Correction

Error 3 Line number ..... Correction

Error 4 Line number ..... Correction

5 **Total: 5 marks**

A coffee machine sells three cup sizes. Each size is entered as a single character.

Write an algorithm using pseudocode to: **[5]**

- input a cup size, with a suitable prompt
- use a CASE statement to set `Price` to 1.80 for size `'S'`, 2.40 for size `'M'` and 2.90 for size `'L'`
- set `Price` to 0.0 for any other input
- output `Price` with a suitable message.

You do not need to declare any variables.

Total for this worksheet: 21 marks

MODULE M PROGRAMMING · M10

Count-controlled loops (FOR)

Original practice questions · 22 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.4(c) (with 8.1.4(d))

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **22 marks**. Pseudocode on this sheet follows the Cambridge 0478 pseudocode rules. Write any pseudocode answers in the same style.

1

Total: 5 marks

Each loop below contains one `OUTPUT` statement inside it.

Complete the table to show how many times the statement inside each loop is executed. **[5]**

First line of the loop	Number of times executed
`FOR Count ← 1 TO 8`	
`FOR Count ← 0 TO 15 STEP 5`	
`FOR Count ← 20 TO 2 STEP -3`	
`FOR Count ← 6 TO 6`	
`FOR Count ← 7 TO 3`	

2

Total: 5 marks

A bakery records how many loaves it sells each day for four days.

```
01 Total ← 0
02 Best ← 0
03 FOR Day ← 1 TO 4
04     INPUT Sold
05     Total ← Total + Sold
06     IF Sold > Best
07       THEN
08         Best ← Sold
09     ENDIF
10 NEXT Day
11 OUTPUT "Total ", Total, " Best ", Best
```

Complete the trace table for the algorithm using the input data: 35, 52, 41, 60 **[5]**

Day	Sold	Total	Best	OUTPUT

3

Total: 3 marks

The display on a water tank is being tested. It must show the values 100, 75, 50, 25 and 0, in that order, one value on each line.

Write pseudocode that uses a count-controlled loop to output these five values. **[3]**

4 **Total: 4 marks**

A park has a walking trail with a marker post every 5 metres. This algorithm should store the distances 5, 10, 15, … 60 in the array `Marker`, so that `Marker[1]` stores 5 and `Marker[12]` stores 60. It then outputs the distance of the last marker.

```
01 DECLARE Post : REAL
02 DECLARE Marker : ARRAY[1:12] OF INTEGER
03 FOR Post ← 1 TO 11
04     Marker[Post] ← Post + 5
05 NEXT Marker
06 OUTPUT "Last marker is at ", Marker[12], " metres"
```

There are four errors in this algorithm. **[4]**
Identify the line number of each error and suggest a correction.

Line number	Correction

5 **Total: 5 marks**

A minibus company makes 12 journeys each day.

Write an algorithm using pseudocode to: **[5]**

- use a count-controlled loop to input the number of passengers on each of the 12 journeys
- total the number of passengers
- count how many journeys had more than 10 passengers
- output the total and the count, with suitable messages, after all 12 journeys have been input.

You do not need to declare any variables.

Total for this worksheet: 22 marks

MODULE M PROGRAMMING · M11

Pre-condition loops (WHILE)

Original practice questions · 21 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.4(c) (with 7.7, 7.8, 7.9)

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **21 marks**. Pseudocode on this sheet follows the Cambridge 0478 pseudocode rules. Write any pseudocode answers in the same style.

1

Total: 3 marks

Tick (✓) **one** box in each row to show whether each statement about a pre-condition (`WHILE`) loop is true or false. **[3]**

Statement	True	False
The condition is tested before the statements inside the loop are executed.		
The statements inside the loop are always executed at least once.		
The loop terminates when the condition evaluates to FALSE.		

2

Total: 3 marks

A garden centre grows hedge plants. A new plant is 12 cm tall and grows 5 cm each week. It is ready to sell when it is at least 40 cm tall. This algorithm uses a pre-condition loop to work out how many weeks it takes before the plant is ready to sell.

```
01 Height ← 12
02 Weeks ← 0
03 WHILE ............................... DO
04     Height ← ..............................
05     Weeks ← Weeks + 1
06 ..............................
07 OUTPUT "Ready to sell after ", Weeks, " weeks"
```

Complete the pseudocode by writing the missing parts of lines 03, 04 and 06. **[3]**

Line 03 _____

Line 04 _____

Line 06 _____

3 **Total: 7 marks**

A factory has a noise sensor. A worker enters each noise reading, in decibels, into this algorithm. A reading above 50 is loud. The value −1 is entered to stop.

```
01 LoudCount ← 0
02 ReadingCount ← 0
03 INPUT Reading
04 WHILE Reading <> -1 DO
05     IF Reading > 50
06       THEN
07         LoudCount ← LoudCount + 1
08     ENDIF
09     ReadingCount ← ReadingCount + 1
10     INPUT Reading
11 ENDWHILE
12 OUTPUT LoudCount, " out of ", ReadingCount, " readings were loud"
```

(a) Complete the trace table for the algorithm using this input data: **[4]**
42, 65, 51, 50, −1

Reading	LoudCount	ReadingCount	OUTPUT

(b) The algorithm is run again. The first value input is −1. **[1]**
State the output.

(c) Explain why the algorithm needs the `INPUT` statement on line 03 as well as the one on line 10. **[2]**

4 **Total: 3 marks**

A farm shop packs eggs into boxes that each hold 12 eggs. This algorithm should input the number of eggs collected, then output how many boxes can be completely filled and how many eggs are left over. It uses a pre-condition loop.

```
01 DECLARE Eggs : INTEGER
02 DECLARE Boxes : INTEGER
03 Boxes ← 1
04 OUTPUT "Enter the number of eggs"
05 INPUT Eggs
06 WHILE Eggs > 12 DO
07     Eggs ← Eggs + 12
08     Boxes ← Boxes + 1
09 ENDWHILE
10 OUTPUT Boxes, " full boxes and ", Eggs, " eggs left over"
```

There are **three** errors in this algorithm. **[3]**
Identify the line number of each error and write the corrected pseudocode for that line.

Error 1: line number ..... Correction .....

Error 2: line number ..... Correction .....

Error 3: line number ..... Correction .....

5 **Total: 5 marks**

A ferry company takes bookings for small groups. Each booking must be for 1 to 9 passengers inclusive. The fare is $12.50 for each passenger.

Write an algorithm using pseudocode to: **[5]**

- input the number of passengers in a booking
- use a pre-condition loop to output an error message and input the number again, until a number from 1 to 9 is entered
- calculate the total fare for the booking
- output the total fare with a suitable message.

You do not need to declare any variables.

Total for this worksheet: 21 marks

MODULE M PROGRAMMING · M12

Post-condition loops, and choosing the right loop

Original practice questions · 22 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.4(c), 7.7, 7.9

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **22 marks**. Write all algorithms in pseudocode. All data on this sheet is made up.

1

Total: 3 marks

(a) Complete the table by writing the type of loop that matches each description: **count-controlled**, **pre-condition** or **post-condition**. **[3]**

Description	Type of loop
The statements in the loop are always executed at least once, and the condition is tested after them.	
The statements in the loop are executed a set number of times.	
The condition is tested before the statements, so the statements may not be executed at all.	

2 **Total: 6 marks**

A packing robot places items in a crate. The crate is full when the total weight of its items is more than 50 kg. This algorithm counts the items placed in one crate.

```
01 Total ← 0
02 Count ← 0
03 REPEAT
04     INPUT Weight
05     Total ← Total + Weight
06     Count ← Count + 1
07 UNTIL Total > 50
08 OUTPUT "Items in crate: ", Count
```

(a) Complete the trace table for the algorithm using the input data: 12, 18, 9, 15, 20 **[4]**

The first row has been done for you. You may not need to use all the rows.

Weight	Total	Count	OUTPUT
	0	0	

(b) The algorithm is run again with the input data: 60, 5

(i) State the output. **[1]**

(ii) Explain why only one value is input this time. **[1]**

3 **Total: 5 marks**

(a) Complete the table by writing the most suitable type of loop for each task: **count-controlled**, **pre-condition** or **post-condition**. **[3]**

Task	Most suitable type of loop
A test program for a set of traffic lights runs the red, amber, green sequence exactly 20 times.	
A staff entrance asks for a staff code. It must always ask at least once, and keeps asking until a valid code is entered.	
A car park barrier stays raised for as long as a sensor detects a car underneath it. Usually no car is underneath when the sensor is first checked.	

(b) Explain why a post-condition loop is suitable for the staff entrance task. **[2]**

4 **Total: 3 marks**

A lift display shows the floors a lift passes on its way down.

```
01 Floor ← 10
02 WHILE Floor > 0 DO
03     OUTPUT "Floor ", Floor
04     Floor ← Floor - 3
05 ENDWHILE
```

(a) Rewrite this algorithm in pseudocode using a `REPEAT ... UNTIL` loop. Your algorithm must give exactly the same output. **[3]**

5 **Total: 5 marks**

A garden centre takes online orders for plant pots. The number of pots in one order must be from 1 to 40 inclusive.

(a) Write an algorithm using pseudocode and a **post-condition** loop to: **[5]**

- ask the user to enter the number of pots, and input it
- output an error message each time the number is not from 1 to 40 inclusive
- keep asking until a number from 1 to 40 inclusive is entered
- then output a message that includes the number of pots ordered.

You may assume the user always enters a whole number. You do not need to declare any variables.

Total for this worksheet: 22 marks

MODULE M PROGRAMMING · M14

Nested selection and nested loops

Original practice questions · 22 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.5, 8.2, 7.7, 7.8, 7.9

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **22 marks**. Write all algorithms in pseudocode. All data on this sheet is made up.

1

Total: 3 marks

Here are three pseudocode algorithms.

Algorithm P

```
IF Stock < 5
  THEN
    OUTPUT "Reorder"
ENDIF
IF Stock = 0
  THEN
    OUTPUT "Sold out"
ENDIF
```

Algorithm Q

```
IF Speed > 50
  THEN
    IF Lane = 1
      THEN
        OUTPUT "Move over"
    ENDIF
ENDIF
```

Algorithm R

```
FOR Week ← 1 TO 4
    FOR Day ← 1 TO 7
        OUTPUT Week * Day
    NEXT Day
NEXT Week
```

(a) Tick (✓) **one** box in each row to show what each algorithm contains. **[3]**

Algorithm	Nested selection	Nested iteration	No nesting
P			
Q			
R			

2

Total: 4 marks

A delivery firm stores the number of parcels each of its 2 vans delivered on each of 3 days in the 2D array `Parcels[1:2, 1:3]`. `Parcels[Row, Col]` is the number for van `Row` on day `Col`.

	Col 1	**Col 2**	**Col 3**
Row 1	12	4	15
Row 2	9	11	8

```
01 FOR Row ← 1 TO 2
02     Count ← 0
03     FOR Col ← 1 TO 3
04         IF Parcels[Row, Col] > 10
05           THEN
06             Count ← Count + 1
07         ENDIF
08     NEXT Col
09     OUTPUT Count
10 NEXT Row
```

(a) Complete the trace table for the algorithm using the data in the array `Parcels`. You may not need to use all the rows. **[4]**

Row	**Col**	**Count**	**OUTPUT**

3

Total: 4 marks

A meeting-room heater uses this algorithm. `Temp` is the room temperature in °C and `People` is the number of people in the room.

```
01 INPUT Temp
02 INPUT People
03 IF Temp < 18
04   THEN
05     IF People > 0
06       THEN
07         OUTPUT "Heating on"
08       ELSE
09         OUTPUT "Low heat"
10     ENDIF
11   ELSE
12     OUTPUT "Heating off"
13 ENDIF
```

(a) Complete the table to show the output for each set of input data. **[3]**

Temp	People	OUTPUT
15	3	
15	0	
18	2	

(b) State why line 05 is **not** executed when `Temp` is 21. **[1]**

4 **Total: 5 marks**

A hotel has 5 floors, numbered 1 to 5, with 12 rooms on each floor, numbered 1 to 12. The 2D array `Cleaned[1:5, 1:12]` of type BOOLEAN stores TRUE if a room has been cleaned and FALSE if it has not.

This algorithm should output, for each floor, the number of rooms that have **not** been cleaned. The algorithm contains **two** errors.

```
01 FOR Floor ← 1 TO 5
02     NotDone ← 0
03     FOR Room ← 1 TO 12
04         IF Cleaned[Floor, Room] = FALSE
05           THEN
06             NotDone ← NotDone - 1
07         ENDIF
08     NEXT Floor
09     OUTPUT "Floor ", Floor, " rooms left: ", NotDone
10 NEXT Floor
```

(a) Identify the line numbers of the **two** errors and suggest a correction for each error. **[4]**

Error 1 Line number ..... Correction

Error 2 Line number ..... Correction

(b) The errors have been corrected. State how many times line 04 is executed when the algorithm is run once. **[1]**

5 **Total: 6 marks**

A weather station records the rainfall for each day of one week, as a whole number of millimetres (mm).

(a) Write an algorithm using pseudocode to: **[6]**

- input a rainfall reading for each of the 7 days, using a loop
- inside that loop, validate each reading so that only readings from 0 to 200 are accepted, and input the reading again until it is valid
- count how many days had more than 25 mm of rainfall
- output this count, with a suitable message, after all 7 readings have been input.

You do not need to declare any variables.

Total for this worksheet: 22 marks

MODULE M PROGRAMMING · M16

String handling: length, substring, upper and lower case

Original practice questions · 22 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.4(e), 7.7, 7.8, 7.9

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **22 marks**. Write all algorithms and statements in pseudocode. On this sheet, the first character of a string is in position 1. `SUBSTRING(S, Start, Length)` returns `Length` characters of S, beginning at position `Start`. All data on this sheet is made up.

1

Total: 4 marks

A tourist map program has carried out this statement:

```
Place ← "Copper Valley"
```

(a) Complete the table to show the value returned by each expression. **[4]**

Expression	Value returned
`LENGTH(Place)`	
`SUBSTRING(Place, 8, 3)`	
`UCASE(SUBSTRING(Place, 1, 3))`	
`LCASE(Place)`	

2

Total: 3 marks

The STRING variable `Product` has already been given a value by a stock control program.

(a) Write a pseudocode statement to store the number of characters in `Product` in the variable `Size`. **[1]**

(b) Write a pseudocode statement to store the **last** character of `Product` in the variable `LastChar`. **[2]**
Your statement must work for a string of any length.

3 **Total: 6 marks**

A word-puzzle program counts how many times the letter t appears in a word.

```
01 DECLARE Word : STRING
02 DECLARE Letter : STRING
03 DECLARE Count : INTEGER
04 DECLARE Index : INTEGER
05 Word ← "Trust"
06 Count ← 0
07 FOR Index ← 1 TO LENGTH(Word)
08     Letter ← LCASE(SUBSTRING(Word, Index, 1))
09     IF Letter = "t"
10       THEN
11         Count ← Count + 1
12     ENDIF
13 NEXT Index
14 OUTPUT "Number found: ", Count
```

(a) Complete the trace table for the algorithm. **[4]**

Index	Letter	Count	OUTPUT

(b) Explain why `LCASE` is used on line 08. **[2]**

4 **Total: 3 marks**

A holiday park checks booking references. A reference is accepted only if it is exactly 8 characters long **and** its first two characters are BK, typed in upper case or lower case.

The algorithm contains **three** errors.

```
01 DECLARE Ref : STRING
02 DECLARE Prefix : CHAR
03 OUTPUT "Enter the booking reference"
04 INPUT Ref
05 Prefix ← LCASE(SUBSTRING(Ref, 1, 2))
06 IF LENGTH(Ref) <> 8 AND Prefix = "BK"
07   THEN
08     OUTPUT "Reference accepted"
09   ELSE
10     OUTPUT "Reference rejected"
11 ENDIF
```

(a) Identify the line number of each error and suggest a correction. **[3]**

Error 1 Line number ..... Correction

Error 2 Line number ..... Correction

Error 3 Line number ..... Correction

5 **Total: 6 marks**

A garden centre prints a short code on each plant label. For example, for a plant called Lavender the output could be: `Label code: LAV 8`

(a) Write an algorithm using pseudocode to: **[6]**

- input the name of a plant, repeating the input until the name has at least 3 characters
- convert the plant name to upper case
- output, with a suitable message, the first three characters of the upper case name and the number of characters in the name.

You do not need to declare any variables.

Total for this worksheet: 22 marks

MODULE M PROGRAMMING · M18

One-dimensional arrays

Original practice questions · 22 marks · ~25 minutes · Cambridge 0478 syllabus 8.2.1, 8.2.2, 8.2.3 (with 7.7, 7.9)

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **22 marks**. Write all algorithms and statements in pseudocode. All data on this sheet is made up.

1

Total: 4 marks

(a) Tick (✓) the **two** statements that are true for a one-dimensional (1D) array. **[2]**

Statement	Tick (✓)
All the elements of the array have the same data type.	
The elements of the array can be a mix of STRING and INTEGER values.	
The number of elements can be increased while the program is running.	
The lower bound of an array must always be 1.	
Each element is accessed using its index.	

An air-quality monitor stores one reading for each hour of the day. The array is declared as:

```
DECLARE Hourly : ARRAY[0:23] OF REAL
```

(b) (i) State the number of elements in the array `Hourly`. **[1]**

(ii) State the data type of `Hourly[5]`. **[1]**

2 **Total: 2 marks**

A parcel depot stores the weights, in kg, of five parcels in the array `Weight`:

```
DECLARE Weight : ARRAY[1:5] OF REAL
```

Index	1	2	3	4	5
Weight	2.5	7.0	1.2	4.8	3.3

(a) State the value of `Weight[4]`. **[1]**

(b) The parcel that weighs 1.2 kg is re-weighed and now weighs 1.5 kg. Write **one** pseudocode statement to store this new weight in the array. **[1]**

3 **Total: 2 marks**

(a) The depot will soon weigh 400 parcels a day. Explain why the weights should be stored in an array rather than in a separate variable for each parcel. **[2]**

4

Total: 5 marks

A bakery has four shelves. The array `Loaves[1:4]` stores the number of loaves left on each shelf. At the end of the day, this algorithm refills any shelf with fewer than 5 loaves so that it holds 12.

```
01 Count ← 0
02 FOR Shelf ← 1 TO 4
03     IF Loaves[Shelf] < 5
04       THEN
05         Loaves[Shelf] ← 12
06         Count ← Count + 1
07     ENDIF
08 NEXT Shelf
09 OUTPUT Count, " shelves refilled"
```

(a) Complete the trace table for the algorithm. The starting contents of the array are shown. **[4]**

Count	Shelf	Loaves[1]	Loaves[2]	Loaves[3]	Loaves[4]	OUTPUT
		8	3	0	5	

(b) Explain why the value in `Loaves[4]` is **not** changed by the algorithm. **[1]**

5

Total: 3 marks

A walking club has 20 members. Their names are to be stored in an array called `Members`.

(a) Write pseudocode to: **[3]**

- declare the array `Members`, with a lower bound of 1
- use a loop to input the name of each member and store it in the next element of the array.

You do not need to declare any other variables.

6 **Total: 6 marks**

A tool-hire shop has eight types of tool, numbered 1 to 8. The array `Stock[1:8]` stores how many of each tool are in the shop. The array has already been declared and filled with values.

(a) Write an algorithm using pseudocode to: **[6]**

- input a tool number from 1 to 8 and the number of that tool being returned to the shop
- add the number being returned to the element of `Stock` for that tool
- then use a loop to output the tool number of every tool that has a stock of 0.

You do not need to declare any variables or validate the inputs.

Total for this worksheet: 22 marks

MODULE M PROGRAMMING · M19

Two-dimensional arrays and looping through them

Original practice questions · 21 marks · ~25 minutes · Cambridge 0478 syllabus 8.2.1, 8.2.2, 8.2.3, 7.7, 7.8, 7.9

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **21 marks**. Write all algorithms and statements in pseudocode. All data on this sheet is made up.

1

Total: 4 marks

A water company records the total rainfall, in millimetres to one decimal place, for each of the 12 months of a year at each of its 5 reservoirs. The data is to be stored in a 2D array called `Rainfall`. The first index is the month (1 to 12) and the second index is the reservoir (1 to 5).

(a) Write a pseudocode statement to declare the array `Rainfall`. Use a lower bound of 1 for both indexes. **[2]**

(b) State **two** benefits of storing this data in one 2D array instead of in 60 separate variables. **[2]**

1

2

2

Total: 3 marks

A bank of parcel lockers has 4 rows of 5 lockers. The 2D array `Lockers` is declared as:

```
DECLARE Lockers : ARRAY[1:4, 1:5] OF CHAR
```

The first index is the row and the second index is the column. Each element stores `'E'` if the locker is empty or `'F'` if it is full. The array currently holds:

	Column 1	Column 2	Column 3	Column 4	Column 5
Row 1	'E'	'F'	'F'	'E'	'F'
Row 2	'F'	'F'	'F'	'F'	'F'
Row 3	'E'	'E'	'F'	'E'	'E'
Row 4	'F'	'E'	'F'	'F'	'F'

(a) State the value stored in `Lockers[4, 2]` **[1]**

(b) A parcel is placed in the locker in row 1, column 4. Write a pseudocode statement to record this in the array. **[1]**

(c) Explain why the statement `OUTPUT Lockers[5, 2]` would cause an error. **[1]**

3 **Total: 5 marks**

This algorithm uses the array `Lockers` from **Question 2**, holding the values shown in the table.

```
01 FOR Row ← 1 TO 4
02     Count ← 0
03     FOR Column ← 1 TO 5
04         IF Lockers[Row, Column] = 'E'
05           THEN
06             Count ← Count + 1
07         ENDIF
08     NEXT Column
09     OUTPUT "Row ", Row, ": ", Count, " empty"
10 NEXT Row
```

(a) Write down the output from the algorithm. **[4]**

(b) Explain why line 02 is placed **inside** the outer loop and not before line 01. **[1]**

4 **Total: 3 marks**

A company has 4 coffee kiosks. The number of drinks sold by each kiosk on each of 7 days is stored in the 2D array `Sales[1:4, 1:7]`. The first index is the kiosk (1 to 4) and the second index is the day (1 to 7).

This algorithm should output the total number of drinks sold by all 4 kiosks on each day. It contains **three** errors.

```
01 FOR Day ← 1 TO 4
02     DayTotal ← 0
03     FOR Kiosk ← 1 TO 4
04         DayTotal ← DayTotal + Sales[Day, Kiosk]
05     NEXT Day
06     OUTPUT "Day ", Day, " total: ", DayTotal
07 NEXT Day
```

(a) Identify the line number of each error and write the corrected pseudocode for that line. **[3]**

Error 1 Line number ..... Correction

Error 2 Line number ..... Correction

Error 3 Line number ..... Correction

5 **Total: 6 marks**

A fish farm has 6 tanks. The pH of the water in each tank is tested 8 times a day. One day's results have already been stored in the 2D array `TankPH[1:6, 1:8]` of type REAL. The first index is the tank number (1 to 6) and the second index is the test number (1 to 8).

(a) Write an algorithm, using pseudocode, to: **[6]**

- input a tank number, and keep asking for it until a number from 1 to 6 inclusive is entered
- calculate the average pH of that tank over its 8 tests
- output the average, rounded to one decimal place, with a suitable message
- output `Check water` once, after the average, if **any** of that tank's results is below 6.5

You do not need to declare any variables.

Total for this worksheet: 21 marks

MODULE M PROGRAMMING · M21

File handling: open, read, write and close

Original practice questions · 20 marks · ~25 minutes · Cambridge 0478 syllabus 8.3.1, 8.3.2

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **20 marks**. Pseudocode on this sheet follows the Cambridge 0478 pseudocode guide.

1

Total: 2 marks

A ferry company's ticket machine counts the tickets it sells. While the program is running, the count is held in the variable `TicketsSold`. When the machine is switched off at night, the program also writes the count to a text file.

Explain why the program stores the count in a text file as well as in the variable. **[2]**

2

Total: 4 marks

Tick (✓) **one** box in each row to show whether each statement about Cambridge pseudocode file handling is true or false. **[4]**

Statement	True	False
`OPENFILE "Stock.txt" FOR WRITE` adds new data after the data already stored in `Stock.txt`		
A file should be opened in only one mode at a time		
`READFILE "Stock.txt", Item` reads a data item from `Stock.txt` and assigns it to the variable `Item`		
`CLOSEFILE "Stock.txt"` deletes the file `Stock.txt`		

3

Total: 4 marks

A rain gauge saves three rainfall readings, in millimetres, in the text file `Rainfall.txt`. The file holds one reading on each line: `12`, then `0`, then `7`.

```
01 DECLARE Reading : INTEGER
02 DECLARE Total : INTEGER
03 DECLARE Count : INTEGER
04 Total ← 0
05 OPENFILE "Rainfall.txt" FOR READ
06 FOR Count ← 1 TO 3
07     READFILE "Rainfall.txt", Reading
08     Total ← Total + Reading
09 NEXT Count
10 CLOSEFILE "Rainfall.txt"
11 OUTPUT "Total rainfall: ", Total
```

(a) Write down the output from line 11. **[1]**

(b) State the file mode used in line 05. **[1]**

(c) These lines are added to the end of the algorithm.

```
12 OPENFILE "Rainfall.txt" FOR WRITE
13 WRITEFILE "Rainfall.txt", Total
14 CLOSEFILE "Rainfall.txt"
```

Describe what `Rainfall.txt` contains after line 14 has been run. Explain why. **[2]**

4 **Total: 4 marks**

A walking club keeps the name of its newest member in the text file `Members.txt`. Only the newest name needs to be kept. This algorithm should input the name and store it in the file, then close the file.

```
01 DECLARE NewName : STRING
02 OUTPUT "Enter the name of the new member"
03 INPUT NewName
04 OPENFILE "Members.txt" FOR READ
05 WRITEFILE NewName, "Members.txt"
06 CLOSEFILE "Members.txt"
```

Identify the line numbers of the **two** errors in the algorithm and suggest a correction for each error. **[4]**

Line number	Correction

5 **Total: 6 marks**

A workshop keeps a list of its tools in the text file `Tools.txt`. The file holds exactly five lines of text, one tool on each line. The workshop wants a copy of the list in a second text file, `Backup.txt`.

Write an algorithm using pseudocode to: **[6]**

- declare any variables you use
- read the five lines of text from `Tools.txt`, one line at a time
- write each line of text to `Backup.txt`
- close both files when they are no longer needed
- output a message to say that the copy is complete.

Total for this worksheet: 20 marks

MODULE M PROGRAMMING · M22

Procedures

Original practice questions · 22 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.6(a), 8.1.6(b), 8.1.8, 7.7, 7.8, 7.9

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **22 marks**. Write all algorithms and statements in pseudocode. All data on this sheet is made up.

1

Total: 3 marks

(a) Tick (✓) **one** box in each row to show whether each statement about procedures is TRUE or FALSE. **[3]**

Statement	TRUE	FALSE
A procedure returns a single value to the point where it was called.		
The same procedure can be called many times in one program.		
A procedure with no parameters is run using `CALL` followed by its identifier.		

2 **Total: 6 marks**

A label printer in a warehouse uses this algorithm.

```
01 PROCEDURE DrawLine
02     OUTPUT "----------"
03 ENDPROCEDURE
04 PROCEDURE ShowTitle
05     OUTPUT "Parcel labels"
06     CALL DrawLine
07 ENDPROCEDURE
08 DECLARE Parcel : INTEGER
09 CALL ShowTitle
10 FOR Parcel ← 1 TO 3
11     OUTPUT "Parcel ", Parcel
12     IF Parcel = 2
13       THEN
14         CALL DrawLine
15     ENDIF
16 NEXT Parcel
17 CALL DrawLine
```

(a) Write the output from the algorithm. Write one line of output on each answer line. You may not need to use all the lines. **[4]**

(b) Describe what happens when line 09 is executed. **[2]**

3 **Total: 4 marks**

A ticket machine at a ferry terminal shows the same safety notice several times. The notice is these two lines:

```
Keep hold of your ticket
Stand behind the yellow line
```

(a) Write pseudocode to define a procedure called `SafetyNotice`, with no parameters, that outputs the two lines of the notice in order. **[3]**

(b) Write the pseudocode statement that runs the procedure `SafetyNotice`. **[1]**

4 **Total: 3 marks**

This part of an algorithm for a drinks vending machine should keep asking for a choice. When `Q` is entered, it should run the procedure `Goodbye` once and then stop asking.

The algorithm contains **three** errors.

```
01 PROCEDURE Goodbye RETURNS STRING
02     OUTPUT "Thank you for visiting"
03     OUTPUT "Please take your receipt"
04 ENDFUNCTION
05 DECLARE Choice : CHAR
06 REPEAT
07     OUTPUT "Enter B to buy or Q to quit"
08     INPUT Choice
09     IF Choice = 'Q'
10       THEN
11         Goodbye
12     ENDIF
13 UNTIL Choice = 'Q'
```

(a) Identify the line number of each error and suggest a correction. **[3]**

Error 1 Line number ..... Correction

Error 2 Line number ..... Correction

Error 3 Line number ..... Correction

5 **Total: 6 marks**

A tool hire shop checks each tool when it is returned. A member of staff enters a condition code: `'G'` for good or `'D'` for damaged. Two statements in the algorithm are repeated.

```
01 DECLARE Condition : CHAR
02 INPUT Condition
03 IF Condition = 'G'
04   THEN
05     OUTPUT "Good condition"
06     OUTPUT "Clean the tool"
07     OUTPUT "Charge the battery"
08   ELSE
09     IF Condition = 'D'
10       THEN
11         OUTPUT "Damage recorded"
12         OUTPUT "Clean the tool"
13         OUTPUT "Charge the battery"
14       ELSE
15         OUTPUT "Code not recognised"
16     ENDIF
17 ENDIF
```

(a) Rewrite the algorithm, using pseudocode, so that the two repeated statements are in a procedure called `FinishReturn`. The procedure must be defined at the start of the algorithm and called wherever the two statements are needed. The algorithm must still give the same output for every input. **[4]**

(b) Explain **one** benefit of using the procedure `FinishReturn` in this algorithm. **[2]**

Total for this worksheet: 22 marks

MODULE M PROGRAMMING · M23

Functions and return values

Original practice questions · 22 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.6(a), 8.1.6(b)

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **22 marks**. Write all algorithms and statements in pseudocode. All data on this sheet is made up.

1

Total: 3 marks

(a) Tick (✓) **one** box in each row to show whether each statement applies to a function, a procedure, or both. **[3]**

Statement	Function	Procedure	Both
It is run using the keyword CALL, as a complete statement.			
Its header states the data type of the value that is sent back.			
It is defined once and can be used many times in a program.			

2 **Total: 4 marks**

A print shop charges 5 cents for every page it prints. Its ordering program uses this function to work out the cost, in cents, of a print job.

```
01 FUNCTION PrintCost(Pages : INTEGER, Copies : INTEGER) RETURNS INTEGER
02     DECLARE TotalPages : INTEGER
03     TotalPages ← Pages * Copies
04     RETURN TotalPages * 5
05 ENDFUNCTION
```

(a) State the data type of the value returned by the function. **[1]**

(b) State the value returned by `PrintCost(3, 4)` **[1]**

(c) A customer orders two print jobs: **[2]**

- a 2-page leaflet, 5 copies
- a 4-page booklet, 3 copies.

Write **one** pseudocode statement that uses the function `PrintCost` to store the total cost of both jobs in the variable `OrderTotal`

3

Total: 5 marks

A warehouse packs items into boxes. Each box holds up to 8 items. This function works out how many boxes an order needs.

```
01 FUNCTION BoxesNeeded(Items : INTEGER) RETURNS INTEGER
02     DECLARE Boxes : INTEGER
03     Boxes ← DIV(Items, 8)
04     IF MOD(Items, 8) <> 0
05       THEN
06         Boxes ← Boxes + 1
07     ENDIF
08     RETURN Boxes
09 ENDFUNCTION
```

(a) Complete the table to show the value returned by each function call. **[3]**

Function call	Value returned
BoxesNeeded(16)	
BoxesNeeded(17)	
BoxesNeeded(5)	

The main program uses the function like this:

```
10 DECLARE Orders : INTEGER
11 INPUT Orders
12 IF BoxesNeeded(Orders) > 2
13   THEN
14     OUTPUT "Use a trolley"
15   ELSE
16     OUTPUT "Carry by hand"
17 ENDIF
```

(b) State what the algorithm outputs when the input is: **[2]**

20 .....

16 .....

4 **Total: 4 marks**

A water-quality sensor takes three whole-number readings. The algorithm below should:

- use the function `MeanReading` to **return** the mean of three readings; the mean may not be a whole number
- store the mean of the readings 4, 9 and 8 in the variable `Result`, then output it.

The algorithm contains **three** errors.

```
01 FUNCTION MeanReading(First : INTEGER, Second : INTEGER, Third : INTEGER) RETURNS INTEGER
02     DECLARE Mean : REAL
03     Mean ← (First + Second + Third) / 3
04     OUTPUT Mean
05 ENDFUNCTION
06 DECLARE Result : REAL
07 CALL MeanReading(4, 9, 8)
08 OUTPUT "Mean reading is ", Result
```

(a) Identify the line number of each error and suggest a correction. **[3]**

Error 1 Line number ..... Correction

Error 2 Line number ..... Correction

Error 3 Line number ..... Correction

(b) The errors are corrected. State the output from the algorithm. **[1]**

5 **Total: 6 marks**

A hotel heating system checks whether a room is at a comfortable temperature. A temperature, in °C, is comfortable if it is from 18.0 to 24.0 inclusive. Temperatures are real numbers, such as 21.5

(a) Write an algorithm using pseudocode to: **[6]**

- define a function `IsComfortable` that has a temperature as its parameter and returns `TRUE` if the temperature is comfortable, or `FALSE` if it is not
- input a room temperature
- use the function `IsComfortable` to output `"Comfortable"` or `"Adjust heating"` for the temperature input.

You do not need to declare any variables in the main algorithm.

Total for this worksheet: 22 marks

MODULE M PROGRAMMING · M24

Parameters and arguments

Original practice questions · 20 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.6(a), 8.1.6(b)

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **20 marks**. Write all code in pseudocode. All data on this sheet is made up.

1

Total: 1 mark

(a) Tick (✓) **one** box to show which statement describes a parameter. **[1]**

Statement	Tick (✓)
A value that is sent back by a function to the point where it was called.	
A variable named in the header of a procedure or function that receives a value when it is called.	
A variable declared in the main program that can be used anywhere in the program.	
A routine, such as ROUND, that is already provided by the programming language.	

2

Total: 4 marks

A guest house uses this pseudocode to work out the deposit for a booking.

```
01 FUNCTION Deposit(NightlyRate : REAL, Nights : INTEGER) RETURNS REAL
02     DECLARE Amount : REAL
03     Amount ← NightlyRate * Nights / 4
04     RETURN Amount
05 ENDFUNCTION
06
07 DECLARE ToPay : REAL
08 ToPay ← Deposit(80.0, 3)
09 OUTPUT "Deposit due: ", ToPay
```

(a) Identify the **two** parameters of the function `Deposit`. **[1]**

(b) State the value that is given to the parameter `Nights` when line 08 is executed. **[1]**

(c) State the data type of the value that must be passed to the parameter `NightlyRate`. **[1]**

(d) State the output from line 09. **[1]**

3 **Total: 5 marks**

A car park uses this procedure to output the charge for a stay.

```
01 PROCEDURE ParkingCharge(Hours : INTEGER, Rate : REAL, Member : BOOLEAN)
02     DECLARE Charge : REAL
03     Charge ← Hours * Rate
04     IF Member = TRUE
05       THEN
06         Charge ← Charge - 2.0
07     ENDIF
08     OUTPUT "Charge: ", Charge
09 ENDPROCEDURE
10
11 CALL ParkingCharge(3, 1.5, FALSE)
12 CALL ParkingCharge(5, 2.0, TRUE)
13 CALL ParkingCharge(2, 1.5, TRUE)
```

(a) Complete the table to show the output from each of the procedure calls on lines 11, 12 and 13. **[3]**

Call on line	Output
11	
12	
13	

(b) A programmer adds this statement:

```
14 CALL ParkingCharge(TRUE, 4, 2.5)
```

Explain why this statement is **not** a correct call of the procedure. **[2]**

4 **Total: 4 marks**

An office reception desk prints a badge for each visitor. The badge shows the visitor's name and the number of the desk they are visiting.

(a) Write the pseudocode for the **first line** (header) of a procedure called `ShowBadge` that has two parameters: the visitor's name and the desk number, which is a whole number. **[2]**

(b) Write a pseudocode statement that uses `ShowBadge` to print a badge for a visitor called `"Dana Reyes"` who is visiting desk `14`. **[2]**

5 **Total: 6 marks**

A building's heating system checks whether readings are within an acceptable range.

(a) Write pseudocode to define a function called `InRange` that: **[4]**

- has three parameters, all whole numbers: the reading, the lowest acceptable value and the highest acceptable value
- returns TRUE if the reading is from the lowest to the highest acceptable value inclusive
- returns FALSE otherwise.

(b) The variable `Temperature` stores a reading. Write a pseudocode statement that uses your function `InRange` to check whether `Temperature` is from 18 to 24 inclusive, and stores the result in the variable `Comfortable`. **[2]**

Total for this worksheet: 20 marks

MODULE M PROGRAMMING · M25

Local and global variables

Original practice questions · 22 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.6(c), 8.1.6(b)

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **22 marks**. Write all algorithms in pseudocode. All data on this sheet is made up.

1

Total: 3 marks

(a) Tick (✓) **one** box in each row to show whether each statement describes a local variable or a global variable. **[3]**

Statement	Local variable	Global variable
It is declared inside a procedure or function and can only be used there.		
It is declared in the main program and can be used anywhere in the program.		
The memory it uses is freed when the procedure it belongs to ends.		

2

Total: 9 marks

An electric scooter docking station runs this algorithm at the end of each trip. The procedure is defined at the start of the code; the main program starts at line 08.

```
01 PROCEDURE EndTrip(Minutes : INTEGER)
02     DECLARE Credits : INTEGER
03     Credits ← Minutes * 2
04     TotalMinutes ← TotalMinutes + Minutes
05     OUTPUT "Credits used: ", Credits
06 ENDPROCEDURE
07
08 // main program
09 DECLARE TotalMinutes : INTEGER
10 TotalMinutes ← 0
11 CALL EndTrip(15)
12 CALL EndTrip(8)
13 OUTPUT "Total minutes: ", TotalMinutes
```

(a) Identify the global variable in the algorithm. **[1]**

(b) Give the line number of the statement that declares a local variable. **[1]**

(c) State the **three** outputs from the algorithm. **[3]**

1

2

3

(d) A programmer adds this statement after line 13:

```
14 OUTPUT "Last trip credits: ", Credits
```

Explain why this statement will **not** work. **[2]**

(e) Explain why `TotalMinutes` is declared as a global variable rather than inside the procedure `EndTrip`. **[2]**

3 **Total: 4 marks**

A team of programmers is writing a large stock control program for a warehouse. Each programmer writes their own procedures and functions.

(a) Explain **two** reasons why the team should use local variables, rather than global variables, inside their procedures and functions. **[4]**

1

2

4 **Total: 6 marks**

A museum gives each group of visitors one free guidebook for every 6 people in the group. The main program has already declared a global variable `TotalVisitors` of type INTEGER.

(a) Write pseudocode for a procedure `RecordGroup` that: **[5]**

- has one parameter, `GroupSize`, of type INTEGER
- declares a **local** variable and uses it to store the number of free guidebooks for the group
- adds `GroupSize` to the global variable `TotalVisitors`
- outputs the number of free guidebooks with a suitable message.

(b) `TotalVisitors` starts at 0. These two statements are then run:

```
CALL RecordGroup(20)
CALL RecordGroup(13)
```

State the value of `TotalVisitors` after both statements have run. **[1]**

Total for this worksheet: 22 marks

MODULE M PROGRAMMING · M26

Library routines and random numbers

Original practice questions · 22 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.7

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **22 marks**. All algorithms on this sheet are written in pseudocode. Write your own algorithms in pseudocode. `RANDOM()` returns a random number between 0 and 1 inclusive. No calculator.

1

Total: 2 marks

(a) Describe what is meant by a **library routine**. **[2]**

2

Total: 4 marks

(a) Complete the table to show the value assigned to `Answer` by each pseudocode statement. **[4]**

Statement	Value of Answer
`Answer ← ROUND(3.14159, 2)`	
`Answer ← ROUND(18.96, 0)`	
`Answer ← DIV(47, 6)`	
`Answer ← MOD(47, 6)`	

3 **Total: 5 marks**

(a) A board-game program uses this statement.

```
Move ← ROUND(RANDOM() * 4, 0) + 3
```

State the smallest and the largest value that could be stored in `Move`. **[1]**

Smallest .....

Largest .....

(b) A raffle program needs a random whole number from 1 to 50 inclusive. **[2]**
Write a pseudocode statement that stores this number in `Ticket`.

(c) A science simulation uses this statement.

```
Reading ← ROUND(RANDOM() * 10, 1)
```

Describe the values that could be stored in `Reading`. **[2]**

4 **Total: 6 marks**

A delivery company tests its warehouse plan with a simulation. The algorithm models the number of parcels that arrive in each of four hours. Parcels are packed into crates that each hold 12 parcels.

```
01 Total ← 0
02 FOR Hour ← 1 TO 4
03     Parcels ← ROUND(RANDOM() * 20, 0) + 10
04     Total ← Total + Parcels
05     IF Parcels > 25
06       THEN
07         OUTPUT "Busy hour ", Hour
08     ENDIF
09 NEXT Hour
10 OUTPUT DIV(Total, 12), " full crates and ", MOD(Total, 12), " parcels left over"
```

(a) Complete the trace table for the algorithm. When the algorithm is run, `RANDOM()` returns these values in order: **[4]**

0.42, 0.87, 0.06, 0.73

Hour	Parcels	Total	OUTPUT
		0	

(b) State the smallest and the largest value that line 03 could store in `Parcels`. **[1]**

(c) Explain why `ROUND` is used in line 03. **[1]**

5 **Total: 5 marks**

A garden centre runs a prize draw. Every ticket has a whole number from 1 to 500.

(a) Write an algorithm using pseudocode to: **[5]**

- generate a random whole number from 1 to 500 inclusive and store it in `WinningNumber`
- input a customer's ticket number
- repeat the input until the ticket number is between 1 and 500 inclusive
- output "Winner" if the ticket number matches `WinningNumber`, otherwise output a suitable message that includes the winning number.

You do not need to declare any variables.

Total for this worksheet: 22 marks

MODULE M PROGRAMMING · M28

Writing maintainable code: identifiers, comments and sub programs

Original practice questions · 20 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.8, 8.1.6, 8.1.1

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **20 marks**. All algorithms on this sheet are written in pseudocode. Write your own algorithms and statements in pseudocode.

1

Total: 2 marks

(a) Tick (✓) **two** boxes to show which techniques are used to create a maintainable program. **[2]**

	Technique	Tick (✓)
A	Using comments to explain what parts of the program do	
B	Using a range check on each input	
C	Using meaningful identifiers	
D	Using double entry to verify data	
E	Using a trace table to dry-run the algorithm	

2 **Total: 8 marks**

A library charges $0.25 for each day a book is returned late, up to a maximum fee of $5.00. This algorithm works out the fee.

```
01 CONSTANT R ← 0.25
02 CONSTANT M ← 5.0
03 DECLARE D : INTEGER
04 DECLARE F : REAL
05 OUTPUT "Charge per day late is $", R
06 INPUT D
07 F ← D * R
08 IF F > M
09   THEN
10     F ← M
11 ENDIF
12 OUTPUT "Fee to pay is $", F
```

(a) Rewrite lines 06 to 11 using meaningful identifiers in place of `R`, `M`, `D` and `F`. **[4]**

(b) Explain why using meaningful identifiers makes this algorithm easier to maintain. **[2]**

(c) The library changes the charge for each day from $0.25 to $0.30. **[2]**
Explain why storing the charge as a constant with a meaningful identifier makes this change easier to make.

3 **Total: 7 marks**

A garden centre delivers parcels. The delivery charge is $1.50 for each kilogram, with a minimum charge of $4.00. This algorithm works out the total delivery charge for two parcels. The variables are all of data type REAL; their declarations are not shown.

```
01 INPUT Weight1
02 Charge1 ← Weight1 * 1.5
03 IF Charge1 < 4.0
04   THEN
05     Charge1 ← 4.0
06 ENDIF
07 INPUT Weight2
08 Charge2 ← Weight2 * 1.5
09 IF Charge2 < 4.0
10   THEN
11     Charge2 ← 4.0
12 ENDIF
13 OUTPUT "Total delivery charge is $", Charge1 + Charge2
```

A programmer decides to replace the repeated code with a function called `DeliveryCharge`.

(a) Write pseudocode to define the function `DeliveryCharge`. The function: **[4]**

- takes the weight of one parcel, in kilograms, as a parameter
- calculates the charge at $1.50 for each kilogram
- uses a charge of $4.00 if the calculated charge is less than $4.00
- returns the charge.

(b) Write a pseudocode statement to call the function using `Weight1` as the parameter and store the return value in `Charge1`. This statement will replace lines 02 to 06. **[1]**

(c) Explain **one** way that using the function makes this program easier to maintain. **[2]**

4 **Total: 3 marks**

A weather station stores 50 temperature readings, in °C, in the 1D array `Temperature`. This algorithm counts how many of the readings are below zero.

```
01 Count ← 0
02 FOR Index ← 1 TO 50
03     // add 1 to Count
04     IF Temperature[Index] < 0
05       THEN
06         Count ← Count + 1
07     ENDIF
08 NEXT Index
09 OUTPUT "Readings below zero: ", Count
```

(a) Explain why the comment on line 03 is **not** a useful comment. **[2]**

(b) Write a more useful comment, in pseudocode, that could replace line 03. **[1]**

Total for this worksheet: 20 marks