GradeMax

CAMBRIDGE
International Education

Cambridge IGCSE™

COMPUTER SCIENCE **0478/22**

Paper 2 Algorithms, Programming and Logic **May/June 2026**

MARK SCHEME

Maximum Mark: 75

Published

This mark scheme contains the instructions and marking guidance that our examiners used to mark candidate responses for this component.

Before they start marking, examiners complete a standardisation process. They review a range of candidate responses and discuss the mark scheme in detail to agree which answers can and cannot be accepted. Examiners award marks where it is clear that candidate responses have met the requirements of the mark scheme. These requirements may vary between different components or different versions of the same component, depending on the exact requirements of the question and the candidate responses seen during the standardisation process.

Sometimes, our mark schemes may allow marks to be awarded to responses which contain elements that are factually incorrect or for content not covered by the syllabus. We do this in response to the demands of a specific question to support candidates and make sure that our assessments are fair. However, these allowances should not be used as a guide for teaching and learning.

We cannot discuss the mark scheme with schools, and we recommend that schools read the mark scheme together with the assessment material and the Principal Examiner Report for Teachers.

This document consists of **24** printed pages.

© Cambridge University Press & Assessment 2026

[Turn over

GradeMax

General marking principles

All examiners must apply these marking principles when marking candidate responses.

1	Examiners must award marks for each response in line with: • the specific **content** defined in the mark scheme • the specific **skills** defined in the mark scheme or in the level descriptions • the **standard of response** required as exemplified by the standardisation scripts.
2	Examiners must award **whole marks** (not half marks, or other fractions).
3	Examiners must award marks **positively**. This means that: • marks are not deducted for errors or omissions • marks are awarded for correct/valid answers as defined in the mark scheme and also for **other valid answers** which go beyond the scope of the syllabus and mark scheme referring to your team leader as appropriate • the quality of spelling, punctuation and grammar are judged **only** when the mark scheme indicates that these features are **specifically assessed** in the question. However, the meaning of all responses must be unambiguous.
4	Examiners must **consistently** apply the requirements of the mark scheme and any rules for what to do when candidates have not followed instructions correctly.
5	Examiners must use the **full range of marks** defined in the mark scheme, unless limited by the quality of the candidate responses seen.
6	Examiners must award marks according to the mark scheme and **must not** award marks with grade thresholds or grade descriptions in mind.

© Cambridge University Press & Assessment 2026

GradeMax

Annotations guidance for centres

Examiners use a system of annotations as a shorthand for communicating their marking decisions to one another. Examiners are trained during the standardisation process on how and when to use annotations. The purpose of annotations is to inform the standardisation and monitoring processes and guide the supervising examiners when they are checking the work of examiners within their team. The meaning of annotations and how they are used is specific to each component and is understood by all examiners who mark the component.

We publish annotations in our mark schemes to help centres understand the annotations they may see on copies of scripts. Note that there may not be a direct correlation between the number of annotations on a script and the mark awarded. Similarly, the use of an annotation may not be an indication of the quality of the response.

The annotations listed below were available to examiners marking this component in this series.

Annotations

Annotation	Meaning
	Correct point
	Incorrect point
FT	Follow through
REP	Repetition
I	Ignore
BOD	Benefit of doubt given
TV	Content of response too vague
NAQ	Not answered question
	Omission
	Section not relevant

© Cambridge University Press & Assessment 2026

0478/22 Mark Scheme June 2026

GradeMax

Annotation	Meaning
~~	Section incorrect
Highlighter	Highlights part of the answer or shows structure of complex answers
SEEN	Page or response seen by examiner
A2	AO2 mark
A3	AO3 mark
NE	Not enough
R1	Required item one
R2	Required item two
R3	Required item three
✓ 1	Correct awarding one mark
✓ 2	Correct awarding two marks
✓ 3	Correct awarding three marks
✓ 4	Correct awarding four marks
✓ 5	Correct awarding five marks
✓ 6	Correct awarding six marks
✓ 7	Correct awarding seven marks
✓ 8	Correct awarding eight marks
✓ 9	Correct awarding nine marks

© Cambridge University Press & Assessment 2026

GradeMax

Mark scheme abbreviations

/ separates alternative words / phrases within a marking point
// separates alternative answers within a marking point
underline actual word given must be used by candidate (grammatical variants accepted)
max indicates the maximum number of marks that can be awarded
( ) the word / phrase in brackets is not required, but sets the context

Note: No marks are awarded for using brand names of software packages or hardware

© Cambridge University Press & Assessment 2026

Question	Answer	Marks	Guidance
1	C	1	

Question	Answer	Marks	Guidance
2	B	1	

Question	Answer	Marks	Guidance
3(a)(i)	Presence (check)	1	
3(a)(ii)	Check digit	1	
3(b)	**One** mark each • To check that the input matches a specified / required style / pattern / structure • … by containing set numbers of letters and numbers // … by containing specified groups of digits // … by containing specific characters within the text • An example of a format e.g. date format dd/mm/yyyy	3	BOD for arrangement Allow an @ within an email address as an example

PapaCambridge papacambridge.com

Question	Answer	Marks	Guidance

3(c)	**One** mark each
	
	• Appropriate loop to iterate for 15 inputs
	• Input of a number
	• Test if the input is an integer e.g. use of MOD, DIV or ROUND functions
	• Repeated re-input if number not an integer, until it is an integer
	• **Store integer** in the **correct** array location
	• At least **two** outputs seen, from input prompt / error message / completion message.
	
	For example:
	
	`FOR Counter ← 1 TO 15`
	`  REPEAT`
	`  OUTPUT "Input a whole number"`
	`  INPUT Value`
	`  IF MOD(Value,1) <> 0`
	`  // IF ROUND(Value,0) <> Value`
	`  // IF DIV(Value,1) <> Value`
	`  THEN`
	`  OUTPUT "Your number was not an integer, please re-input"`
	`  ELSE`
	`  Accepted[Counter] ← Value`
	`  ENDIF`
	`  UNTIL MOD(Value, 1) = 0`
	`NEXT Counter`
	`OUTPUT "The check has been completed"`
		6	No array or variable declarations required for this algorithm
			
			Accept other methods for testing if a number is an integer
			
			Ignore case

GradeMax

Question	Answer	Marks	Guidance
4	**One** mark per mark point for coding, **max three** • Writing the program code in a **programming/coding language** • … using pseudocode/flowcharts/structure diagrams from the design/earlier stage to create your code • Documenting your code with comments // making sure your code is maintainable • Writing procedures/functions // Use library routines/functions • Using a compiler/interpreter/IDE when writing the code • Testing the code in small parts as you write it allow iterative testing • … to ensure that it works **correctly** / **as expected** // … to refine your code **One** mark per mark point for testing, **max three** • The final stage of testing the **whole program for errors** • Use of **test data** with known outcomes • … to ensure the whole program produces the expected outcomes (from the test data) // … to make sure the program accepts valid data and rejects invalid data • Correct errors // debug the code • Allow **one** mark max for identifying at least two types of test data e.g. normal, abnormal, boundary, extreme	**6**	Allow use of test data as BOD for MP9

© Cambridge University Press & Assessment 2026

GradeMax

Question	Answer	Marks	Guidance

5(a)	**One** mark per mark point, **max four**
	
	• Line 03 / `DECLARE AccountID : INTEGER`
	should be `DECLARE AccountID : STRING`
	
	• Line 05 / `OUTPUT AccountID`
	should be `INPUT AccountID`
	
	• Line 07 / `CASE OF AccountID = Accounts[Row, 1]`
	should be `IF AccountID = Accounts[Row, 1]`
	
	• Line 09 / `OUTPUT "The customer's name is ", Accounts[1, 2]`
	should be `OUTPUT "The customer's name is ", Accounts[Row, 2]`
	
	`01 DECLARE Accounts : ARRAY[1:1000, 1:2] OF STRING`
	`02 DECLARE Row : INTEGER`
	`03 DECLARE AccountID : STRING`
	`04 OUTPUT "Please enter an account ID "`
	`05 INPUT AccountID`
	`06 FOR Row ← 1 TO 1000`
	`07  IF AccountID = Accounts[Row, 1]`
	`08  THEN`
	`09  OUTPUT "The customer's name is ", Accounts[Row, 2]`
	`10  ENDIF`
	`11 NEXT Row`
		4	Ignore case

5(b)	**Linear search**	**1**	Accept **sequential search**

© Cambridge University Press & Assessment 2026

0478/22 Mark Scheme June 2026

GradeMax

Question	Answer	Marks	Guidance
5(c)	**Adding to the existing corrected algorithm** Any **five** from: • Use/initialise a flag at the start / before line 06 • Flag ← FALSE // Flag ← 0 • Between lines 08 and 10 (where the item has been found) • Flag ← TRUE // Flag ← 1 • Outside the loop / after line 11 add another selection statement and output • IF Flag = False … • … THEN OUTPUT "Account number not found in the array" ENDIF **For example:** 01 DECLARE Accounts : ARRAY[1:1000, 1:2] OF STRING 02 DECLARE Row : INTEGER 03 DECLARE AccountID : STRING DECLARE Flag : BOOLEAN **Flag ← FALSE** 04 OUTPUT "Please enter an account ID " 05 INPUT AccountID 06 FOR Row ← 1 TO 1000 07 IF AccountID = Accounts[Row, 1] 08 THEN 09 OUTPUT "The customer's name is ", Accounts[Row, 2] **Flag ← TRUE** 10 ENDIF 11 NEXT Row **IF Flag = FALSE** **THEN** **OUTPUT "Account number not found in the array"** **ENDIF**	**5**	If no location is given, Max 1 for each of MP1/2 or MP3/4 or Max 2 for MP5/6/7. The locations stated in the mark points are for the use of examiners. Any line numbers quoted within these ranges, inclusive, are accepted. If a correct location is clearly identified descriptively without using line numbers, award the mark. Ignore case

© Cambridge University Press & Assessment 2026

Question	Answer	Marks	Guidance
5(c)	**OR replacing FOR loop with a WHILE loop** Any **five** from: • Use/initialise a flag at the start / before line 06 • Flag ← FALSE // Flag ← 0 • Between lines 08 and 10 (where the item has been found) • Flag ← TRUE // Flag ← 1 • Outside the loop / after line 11 add another selection statement and output • IF Flag = False … • … THEN OUTPUT "Account number not found in the array " ENDIF • Replace lines 06 and 11 and increment counter before line 11 • WHILE NOT Flag AND Row <= 1000 (DO) • Row ← Row + 1 **and** ENDWHILE **For example:** 01 DECLARE Accounts : ARRAY[1:1000, 1:2] OF STRING 02 DECLARE Row : INTEGER 03 DECLARE AccountID : STRING DECLARE Flag : BOOLEAN **Row ← 1** **Flag ← FALSE** 04 OUTPUT "Please enter an account ID " 05 INPUT AccountID **06 WHILE NOT Flag AND Row <= 1000 (DO)** 07 IF AccountID = Accounts[Row, 1] 08 THEN 09 OUTPUT "The customer's name is ", Accounts[Row, 2] **Flag ← TRUE** 10 ENDIF **Row ← Row + 1**		If no location given, Max 1 for each of MP1/2 or MP3/4 or Max 2 for MP5/6/7 Max 2 or MP8/9/10. The locations stated in the mark points are for the use of examiners. Any line numbers quoted within these ranges, inclusive, are accepted. If a correct location is clearly identified descriptively without using line numbers, award the mark. Ignore case

GradeMax

Question	Answer	Marks	Guidance
5(c)	11 ENDWHILE IF Flag = FALSE THEN OUTPUT "Account number not found in the array" ENDIF		

© Cambridge University Press & Assessment 2026

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>6(a)</td><td>One mark for each correct column
<table>
<tr><th>Word</th><th>WordLength</th><th>Count</th><th>NumLetter</th><th>Choice</th><th>OUTPUT</th></tr>
<tr><td>post</td><td>4</td><td>0</td><td>1</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>2</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>3</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>4</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>Y</td><td>0</td></tr>
<tr><td>committee</td><td>9</td><td>0</td><td>1</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>2</td><td></td><td></td></tr>
<tr><td></td><td></td><td>1</td><td>3</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>4</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>5</td><td></td><td></td></tr>
<tr><td></td><td></td><td>2</td><td>6</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>7</td><td></td><td></td></tr>
<tr><td></td><td></td><td>3</td><td>8</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>9</td><td></td><td>3</td></tr>
<tr><td></td><td></td><td></td><td></td><td>N</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td><td>6</td><td>If no other marks awarded, allow 1 mark for correct first initialisation of <code>Count</code> and <code>NumLetter</code> to 0 and 1 respectively

FT for the OUTPUT column for candidate's <code>Count</code> final values for each word</td></tr>
</table>

© Cambridge University Press & Assessment 2026

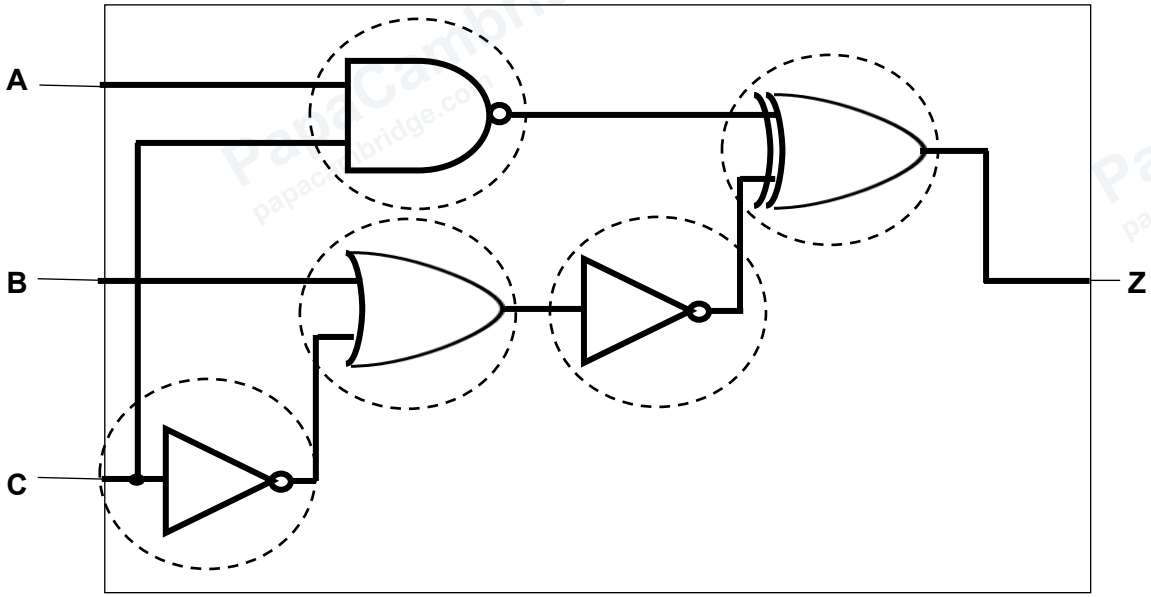
GradeMax

Question	Answer	Marks	Guidance
6(b)	Counting/outputting the number of double/repeated/pairs of letters in a word	1	

Question	Answer	Marks	Guidance
7(a)	**One** from: • So the data is stored permanently // provides a non-volatile copy • So the data is not lost if the machine/program is switched off/crashes • So the data can be used again later // to provide a backup • So the data can be transported to a different (offline) computer // so the data can be shared	1	

7(b)	**One** mark each
	
	• Declaration of an appropriately named variable
	• Correct statement to open `Names.txt` for reading
	• Correct statement to read data from the file into a variable
	• Correct statement to close `Names.txt`
	
	For example:
	
	`DECLARE Name : STRING`
	`OPENFILE "Names.txt" FOR READ`
	`READFILE "Names.txt", Name`
	`CLOSEFILE "Names.txt"`
		4	Ignore case
			
			Filename and variable name must be consistent throughout
			
			If `Names.txt` not used, or quotes are missing, penalise once and FT
			
			Accept single or double quotes for the filename

© Cambridge University Press & Assessment 2026

Question	Answer	Marks	Guidance
8(a)	**One** mark for each correct gate, with the correct inputs as shown. 	**5**	

© Cambridge University Press & Assessment 2026

GradeMax

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>8(b)</td><td>
Four marks for eight correct outputs
Three marks for six or seven correct outputs
Two marks for four or five correct outputs
One mark for two or three correct outputs

<table>
<tr><th>A</th><th>B</th><th>C</th><th>Z</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td></tr>
</table>
</td><td>4</td><td>emboldened parts given in question</td></tr>
</table>

© Cambridge University Press & Assessment 2026

GradeMax

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>9(a)</td><td>One mark for every two correct data types

<table>
<tr><th>field</th><th>data type</th><th>description</th></tr>
<tr><td><code>PartID</code></td><td>text</td><td>uniquely identifies the car part with letters and numbers</td></tr>
<tr><td><code>PartName</code></td><td>text</td><td>name of the car part, for example windscreen wiper</td></tr>
<tr><td><code>CarType</code></td><td>integer/text</td><td>four-digit number code to represent the car type</td></tr>
<tr><td><code>Price</code></td><td>real</td><td>selling price to 2 decimal places</td></tr>
<tr><td><code>Size</code></td><td>text</td><td>specifies the size if relevant, otherwise would say ‘one size’</td></tr>
<tr><td><code>Colour</code></td><td>character</td><td>single letter code for the colour of the item</td></tr>
<tr><td><code>NumberInStock</code></td><td>Integer</td><td>the number of car parts in stock</td></tr>
<tr><td><code>InStock</code></td><td>Boolean/text/character/ integer</td><td>whether or not the part is in stock</td></tr>
</table></td><td>4</td><td>Emboldened parts given in question

Ignore case</td></tr>
</table>

© Cambridge University Press & Assessment 2026

GradeMax

Question	Answer	Marks	Guidance
9(b)	**One** mark each • At least two correct fields in `SELECT` row • Remaining correct field in `SELECT` row • Correct **`FROM`** `CAR_PARTS` • Correct **`WHERE`** `PartName = "brake pads";` **Correct code:** **`SELECT`** `PartID, CarType, NumberInStock` **`FROM`** `CAR_PARTS` **`WHERE`** `PartName = "brake pads";` // `PartName = 'brake pads';`	**4**	Ignore case on field names, but spelling must match names in table. Field names in `SELECT` line can be in any order. If space in field/table name, penalise once and FT. Allow brake_pads, BrakePads, BrakePad for MP4. Ignore minor spelling errors e.g. break_pads
9(c)	**One** mark each • There is an alternative field `NumberInStock` • … a value of 0 in this field indicates the item is out of stock // … a positive value in this field indicates the item is in stock // … this field indicates the quantity in stock	**2**	

© Cambridge University Press & Assessment 2026

GradeMax

Question	Answer	Marks	Guidance
10	Use the tables for AO2 and AO3 below to award a mark in a suitable band using a best fit approach, then add up the total. Marks are available for: • AO2 (maximum 9 marks) • AO3 (maximum 6 marks) **Data structures required** with names as given in the scenario: Arrays or lists `Rainfall[]` **Requirements (techniques)** **R1** Declares of variables and array, initialises array, inputs daily rainfall in mm for a year and stores it in the array (declarations, initialisation, iteration, input, output). **R2** Calculates the total rainfall for the year in cm rounded to 2 d.p., the mean average rainfall in mm rounded to 4 d.p., the number of days with no rainfall and the longest period of days with no rainfall (totalling, counting, iteration, rounding, selection, finding max). **R3** Outputs the total rainfall for the year in cm, the average rainfall, the number of days with no rainfall and the longest period of days with no rainfall. Determines if a drought occurred and outputs the result (output, selection).	**15**	

© Cambridge University Press & Assessment 2026

0478/22 Mark Scheme June 2026

GradeMax

Question	Answer	Marks	Guidance

10	**Example 15-mark answer in pseudocode:**
	
	`// array and variable declaration`
	`DECLARE Rainfall : ARRAY[1:365] OF REAL`
	`DECLARE Index : INTEGER`
	`DECLARE TotalRain : REAL`
	`DECLARE AverageRain : REAL`
	`DECLARE NoRain : INTEGER`
	`DECLARE DryDays : INTEGER`
	`DECLARE MaxDryDays : INTEGER`
	
	`// initialising array`
	`FOR Index ← 1 TO 365`
	`  Rainfall[Index] ← 0`
	`NEXT Index`
	
	`// daily input of rainfall`
	`FOR Index ← 1 TO 365`
	`  OUTPUT "Input the rainfall measurement for day ", Index`
	`  INPUT Rainfall[Index]`
	`NEXT Index`
	
	`// finding the total and average rainfall`
	`TotalRain ← 0`
	`FOR Index ← 1 TO 365`
	`  TotalRain ← TotalRain + Rainfall[Index]`
	`NEXT Index`
	
	`// average daily rainfall to 4dp`
	`AverageRain ← ROUND(TotalRain/365, 4)`
	`// total rainfall in cm to 2dp`
	`TotalRain ← ROUND(TotalRain/10, 2)`		

© Cambridge University Press & Assessment 2026

Question	Answer	Marks	Guidance

10	`// counting the number of days with no rain`
	`NoRain ← 0`
	`FOR Index ← 1 TO 365`
	`  IF Rainfall[Index] = 0`
	`  THEN`
	`  NoRain ← NoRain + 1`
	`  ENDIF`
	`NEXT Index`
	
	`// finding the longest number of consecutive rain free days`
	`IF NoRain > 0`
	`  THEN`
	`  MaxDryDays ← 1`
	`  DryDays ← 1`
	`  FOR Index ← 1 TO 365-1`
	`  IF Rainfall[Index] = 0 AND Rainfall[Index + 1] = 0`
	`  THEN`
	`  DryDays ← DryDays + 1`
	`  IF DryDays > MaxDryDays`
	`  THEN`
	`  MaxDryDays ← DryDays`
	`  ENDIF`
	`  ELSE`
	`  DryDays ← 1`
	`  ENDIF`
	`  NEXT Index`
	`  ELSE`
	`  MaxDryDays ← 0`
	`  DryDays ← 0`
	`ENDIF`		

© Cambridge University Press & Assessment 2026

Question	Answer	Marks	Guidance

10	`// outputting the statistics with messages`
	`OUTPUT "The total rainfall for the year was ", TotalRain, " cm"`
	`OUTPUT "The average daily rainfall was ", AverageRain, " mm"`
	`OUTPUT "The total number of days with no rainfall was ", NoRain, " day(s)"`
	`OUTPUT "The longest period with no rainfall was ", MaxDryDays, " day(s)"`
	
	`// checking if there was a drought`
	`IF MaxDryDays >= 15`
	`  THEN`
	`    OUTPUT ""`
	`    OUTPUT "There was a drought this year because the longest period with no rainfall was 15 days or longer"`
	`  ELSE`
	`    OUTPUT ""`
	`    OUTPUT "There was no drought this year, because the longest period with no rainfall was shorter than 15 days"`
	`ENDIF`		

© Cambridge University Press & Assessment 2026

GradeMax

Marking Instructions in italics			
AO2: Apply knowledge and understanding of the principles and concepts of computer science to a given context, including the analysis and design of computational or programming problems			
0	**1–3**	**4–6**	**7–9**
No creditable response.	At least one programming technique has been used. *Any use of selection, iteration, counting, totalling, input and output.*	Some programming techniques used are appropriate to the problem. *More than one technique seen applied to the scenario, check the list of techniques needed.*	The range of programming techniques used is appropriate to the problem. *All criteria stated for the scenario have been covered by the use of appropriate programming techniques, check the list of techniques needed.*
	Some data has been stored but not appropriately. *Any **use** of variables or arrays or other language dependent data structures e.g. Python lists.*	Some of the data structures chosen are appropriate and store some of the data required. *More than one data structure **used** to store data required by the scenario.*	The data structures chosen are appropriate and store all the data required. *The data structures **used** store all the data required by the scenario.*

© Cambridge University Press & Assessment 2026

0478/22 Mark Scheme June 2026

Marking Instructions in italics			
AO3: Provide solutions to problems by: evaluating computer systems making reasoned judgements presenting conclusions			
0	**1–2**	**3–4**	**5–6**
No creditable response.	Program seen without relevant comments.	Program seen with some relevant comment(s).	The program has been fully commented
	Some identifier names used are appropriate. *Some of the data structures used have meaningful names.*	The majority of identifiers used are appropriately named. *Most of the data structures used have meaningful names.*	Suitable identifiers with names meaningful to their purpose have been used throughout. *All the data structures used have meaningful names.*
	The solution is illogical.	The solution contains parts that may be illogical.	The program is in a logical order.
	The solution is inaccurate in many places. *Solution contains few lines of code with errors that attempt to perform a task given in the scenario.*	The solution contains parts that are inaccurate. *Solution contains lines of code with some errors that logically perform tasks given in the scenario. Ignore minor syntax errors.*	The solution is accurate. *Solution logically performs all the tasks given in the scenario. Ignore minor syntax errors.*
	The solution attempts at least one of the requirements. *Solution contains lines of code that attempt at least one task given in the scenario.*	The solution attempts to meet most of the requirements. *Solution contains lines of code that attempt most tasks given in the scenario.*	The solution meets all the requirements given in the question. *Solution performs all the tasks given in the scenario.*

© Cambridge University Press & Assessment 2026