GradeMax

CAMBRIDGE
International Education

Cambridge IGCSE™

COMPUTER SCIENCE **0478/21**

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 **20** 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
	Follow through
	Repetition
	Ignore
	Benefit of doubt given
	Content of response too vague
	Not answered question
	Omission
	Section not relevant

© Cambridge University Press & Assessment 2026

0478/21 Mark Scheme June 2026

GradeMax

Annotation	Meaning
	Section incorrect
Highlighter	Highlights part of the answer or shows structure of complex answers
	Page or response seen by examiner
	AO2 mark
	AO3 mark
	Not enough
	Required item one
	Required item two
	Required item three
	Correct awarding one mark
	Correct awarding two marks
	Correct awarding three marks
	Correct awarding four marks
	Correct awarding five marks
	Correct awarding six marks
	Correct awarding seven marks
	Correct awarding eight marks
	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
MP	mark point

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

© Cambridge University Press & Assessment 2026

GradeMax

Question	Answer	Marks	Guidance
1	**D**	**1**	

Question	Answer	Marks	Guidance
2	**One** mark for each correct operator 1 `>` 2 `<=` 3 `AND` 4 `NOT`	**4**	Order not important

Question	Answer	Marks	Guidance
3	**One** mark for each data type:	**3**	

variable	description	example data	data type
`Age`	The age of the student.	`16`	**integer**
`TelephoneNumber`	The telephone number of the student.	`+44 7846398573`	**string**
`Teenager`	Whether the student is a teenager or not	`TRUE`	**Boolean**

© Cambridge University Press & Assessment 2026

Question	Answer	Marks	Guidance

4	**One** mark for each correct completion
	
	`Last ← __50____`
	`REPEAT`
	`  Swap ← __FALSE___`
	`  FOR __Index___ ← 1 TO Last - 1`
	`  IF Names[Index] > Names[Index + 1]`
	`  THEN`
	`  Temp ← Names[Index]`
	`  Names[Index] ← Names[Index + 1]`
	`  Names[Index + 1] ← Temp`
	`  Swap ← TRUE`
	`  ENDIF`
	`  NEXT Index`
	`  Last ← Last - 1`
	`UNTIL __NOT // False =__ Swap OR Last = 1`
		4	

Question	Answer	Marks	Guidance
5(a)	**One** mark for correct use of `RANDOM() * 999 + 1` **One** mark for assignment to `GeneratedNumber` in attempt to randomise number `GeneratedNumber ← RANDOM() * 999 + 1`	**2**	Ignore use of ROUND, unless used to produce the correct answer

© Cambridge University Press & Assessment 2026

GradeMax

Question	Answer	Marks	Guidance

5(b)	**One** mark for correct use of UCASE
	One mark for correct use of SUBSTRING
	One mark for assignment to `Temp` of attempt to extract from `LastName` and/or convert to upper case
	
	`Temp ← UCASE(SUBSTRING(LastName, 1, 2))`
	
	`OR`
	
	`Temp ← SUBSTRING(LastName, 1, 2)`
	`Temp ← UCASE(Temp)`
		3	Allow first character of the string to start at 0

5(c)	**One** mark for each mark point
	
	• Correct opening of file
	• Correct writing to file
	• Correct closing of file
	
	`OPENFILE "Username.txt" FOR WRITE`
	`WRITEFILE "Username.txt", NewUsername`
	`CLOSEFILE "Username.txt"`
		3	If missing quotation marks from file or `Username.txt` not used, penalise once and FT

5(d)	**One** mark for each correct identification and **one** for each correct description • Double entry (1) enter data twice and only accept identical value/automatic check by computer (1) • Visual check (1) looks at the data that has been entered and confirm it matches (1)	**4**	

© Cambridge University Press & Assessment 2026

GradeMax

Question	Answer	Marks	Guidance
6	**One** mark for each correct identification and **one** for each correct description • FOR (Loop) (1) iterates a pre-determined number of times (1) • REPEAT (UNTIL) (1) checks the condition at the end of the loop//will always iterate once (1)	**4**	

Question	Answer	Marks	Guidance

7	**One** mark for each line identified and correction given.
	
	Line 3 – should be `DECLARE Average : REAL`
	Line 9 – should be `WHILE Number <> -1 AND Counter <= 100 DO`
	Line 11 – should be `Total ← Total + Number // Total ← Total + Values[Counter]`
	Line 16 – should be `Average ← Total/Counter`
	
	`01 DECLARE Counter : INTEGER`
	`02 DECLARE Total : INTEGER`
	`03 DECLARE Average : REAL`
	`04 DECLARE Values : ARRAY[1:100] OF INTEGER`
	`05 Counter ← 1`
	`06 Total ← 0`
	`07 OUTPUT "Please enter a positive whole number or -1 "`
	`08 INPUT Number`
	`09 WHILE Number <> -1 AND Counter <= 100 DO`
	`10 Values[Counter] ← Number`
	`11 Total ← Total + Number // Values[Counter]`
	`12 Counter ← Counter + 1`
	`13 INPUT Number`
	`14 ENDWHILE`
	`15 Counter ← Counter - 1`
	`16 Average ← Total / Counter`
	`17 OUTPUT "The average is ", Round(Average, 2)`
		4	

© Cambridge University Press & Assessment 2026

GradeMax

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>8(a)</td><td>One mark for each mark point:

• A XOR B
• B AND NOT C
• correct expression: X = (A XOR B) OR (B AND NOT C) // X = A XOR B OR B AND NOT C</td><td>3</td><td>Allow FT if either MP1 or MP2 incorrect for MP3</td></tr>
<tr><td>8(b)</td><td>
<table>
<tr><th>A</th><th>B</th><th>C</th><th>X</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</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>
4 marks for 8 correct outputs
3 marks for 6/7 correct outputs
2 marks for 4/5 correct outputs
1 mark for 2/3 correct outputs</td><td>4</td><td></td></tr>
</table>

© Cambridge University Press & Assessment 2026

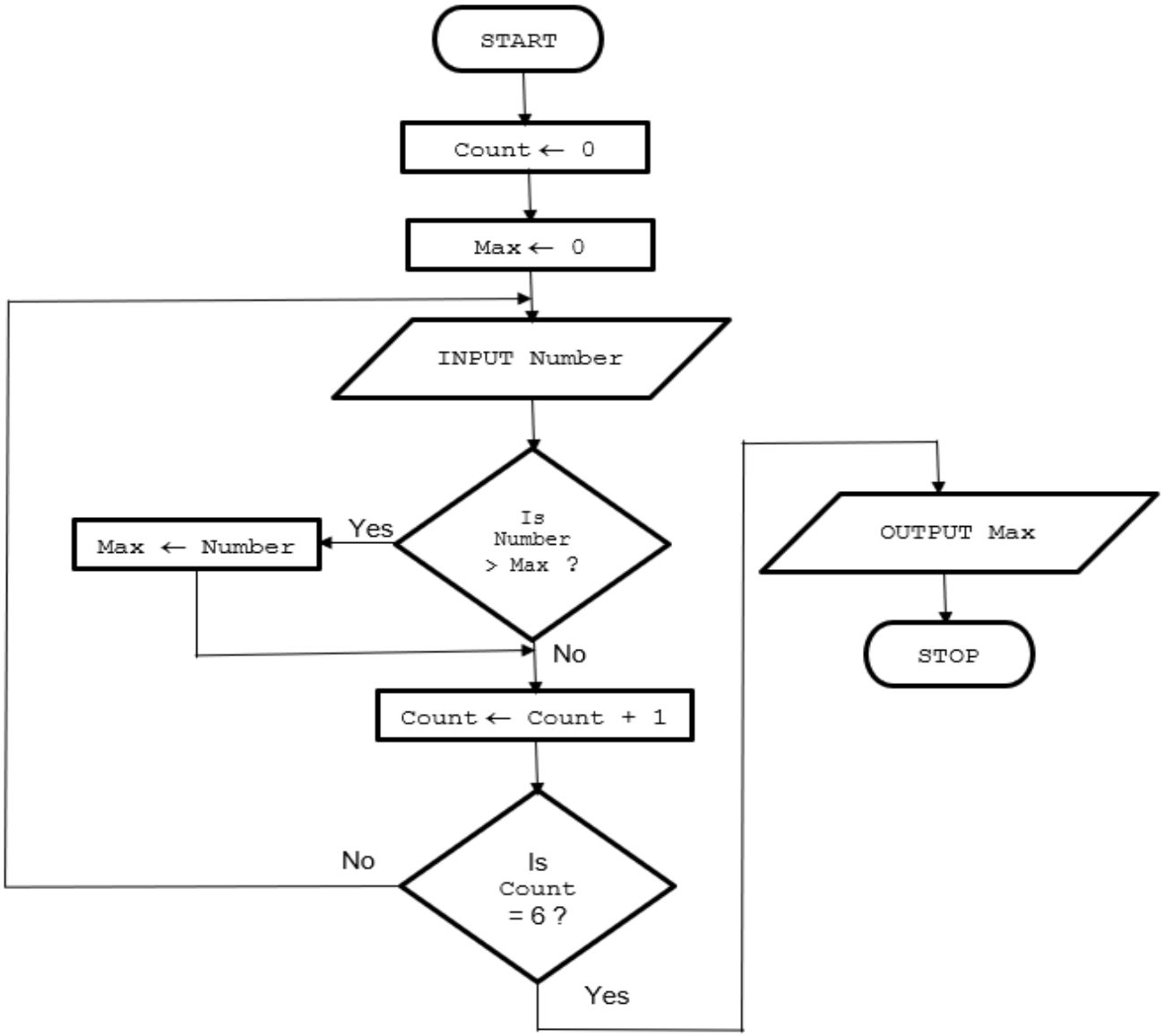
0478/21 Mark Scheme June 2026

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>9(a)(i)</td><td>As <code>MemberName</code> may be repeated/not unique</td><td>1</td><td></td></tr>
<tr><td>9(a)(ii)</td><td><code>MemberID</code></td><td>1</td><td></td></tr>
<tr><td>9(b)</td><td>One mark for two correct data types, two marks for three correct data types

<table><tr><th>field</th><th>data type</th></tr><tr><td><code>Distance</code></td><td>Integer //Real//text//alphanumeric</td></tr><tr><td><code>Date</code></td><td>Date/Time //text//alphanumeric</td></tr><tr><td><code>MemberID</code></td><td>Text //alphanumeric</td></tr></table></td><td>2</td><td>Ignore minor spelling errors</td></tr>
<tr><td>9(c)</td><td>One mark for each correct line

<pre>SELECT MemberName
FROM SWIMMERS
WHERE Distance >= 1500;</pre></td><td>3</td><td></td></tr>
</table>

© Cambridge University Press & Assessment 2026

Question	Answer	Marks	Guidance
10(a)	Example:  	6	Allow structured English if it makes sense. Must use correct flow chart symbols Alternative method is one pass of bubble sort – comparing adjacent and moving largest to end Allow input into an array

GradeMax

Question	Answer	Marks	Guidance
10(a)	**One** mark for each mark point **Max five** marks from: • initialising `Max/highest` and/or `Count` • input of number • checking if number greater than `Max` / adjacent array elements / numbers with correct labels • updating `Max` or `Count` • checking number of entries is equal to 6 // entry of six numbers • outputting correct `Max` value **One** mark for complete flow lines using arrows for attempt at finding highest number; flow chart must terminate at stop/end		
10(b)	**One** mark for each mark point • adding a new decision box with `IS Number > 0 ? …` • … after `INPUT Number` • looping back to before `INPUT Number`	**3**	Allow diagram as answer, accept alternative corrections

© Cambridge University Press & Assessment 2026

Question	Answer	Marks	Guidance

11	**One** mark for each mark point
	
	• defining FUNCTION `Power` with ENDFUNCTION and RETURNS INTEGER/REAL
	• defining the parameters correctly
	• correct calculation
	• return the calculated value
	• correct function call
	
	Example:
	
	`FUNCTION Power(FirstNumber : INTEGER, SecondNumber : INTEGER) RETURNS INTEGER`
	`  RETURN FirstNumber ^ SecondNumber`
	`ENDFUNCTION`
	
	Function call:
	`Value ← Power(A, B)`
	
	Alternative calculations:
	
	`Num ← 1`
	`FOR I ← 1 TO SecondNumber`
	`  Num ← Num * FirstNumber`
	`NEXT I`
	
	`Num ← FirstNumber`
	`FOR I ← 1 TO SecondNumber - 1`
	`  Num ← Num * FirstNumber`
	`NEXT I`
		5	

© Cambridge University Press & Assessment 2026

GradeMax

Question	Answer	Marks	Guidance
12	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** names shown underlined must be used as given in the scenario arrays or lists `Sunshine`, `MonthDays` (Year and Month may be seen but no requirement to store) **Requirements (techniques)** R1 input year, calculates if leap year, updates `MonthDays[]` if appropriate, input month name (selection and input) R2 inputs and validates the daily minutes for the month, calculates the total minutes and finds the highest daily sunshine (validation, iteration, selection, totalling)	**15**	

© Cambridge University Press & Assessment 2026

GradeMax

Question	Answer	Marks	Guidance

12	R3 outputs the year, month, highest daily sunshine for the month, calculate and output the total sunshine in hours and minutes (output)
	
	Example 15-mark answer in pseudocode.
	
	// programming techniques of iteration, selection, totalling, and output are used
	
	`Hours ← 0`
	`Minutes ← 0`
	`MaxHours ← 0`
	`MaxMinutes ← 0`
	`Max ← 0`
	`Days ← 0`
	`Total ← 0`
	
	`// R1`
	`////////////////////////////////////////////////////////////////////////`
	
	`OUTPUT "Please enter a year"`
	`INPUT Year`
	
	`IF MOD(Year, 4) = 0   // updating array if leap year`
	`  THEN`
	`    MonthDays[2] = 29`
	`  ELSE`
	`    MonthDays[2] = 28`
	`ENDIF`
	
	`// entering a month and assigning the`
	`// number of days to variable`
	`OUTPUT "Please enter a month"`
	`INPUT Month`
	
	`Month1 ← LCASE(Month)   // converting to lowercase for use in case`
	`// statement – keeping original for output`		

© Cambridge University Press & Assessment 2026

Question	Answer	Marks	Guidance

12	`// R2`
	`////////////////////////////////////////////////////////////////////`
	
	`CASE OF Month1`
	`  "january" : Days ← MonthDays[1]`
	`  "february" : Days ← MonthDays[2]`
	`  "march" : Days ← MonthDays[3]`
	`ENDCASE`
	
	`FOR Count ← 1 to Days  // inputting sunshine for each day and validating`
	`  REPEAT  // in a valid range`
	`  OUTPUT "Please enter sunshine for day in minutes ", Count`
	`  INPUT Sun`
	`  UNTIL Sun >= 0 AND Sun <= 1440`
	`  Sunshine[Count] ← Sun  // storing in array`
	`  Total ← Total + Sun  // totalling sunshine`
	`  IF Sun > Max`
	`  THEN`
	`  Max ← Sun  // updating highest daily`
	`  ENDIF`
	`NEXT Count`
	
	`// R3`
	`////////////////////////////////////////////////////////////////////`
	
	`Hours ← DIV(Total, 60)  // converting to hours and minutes for total and`
	`Minutes ← MOD(Total, 60)  // daily`
	`// MaxHours ← DIV(Max, 60)  // if learner converts highest daily`
	`// MaxMinutes ← MOD(Max, 60)`		

© Cambridge University Press & Assessment 2026

GradeMax

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>12</td><td>

```
// outputting results
OUTPUT "The sunshine for ", Month, " in the year ", Year
OUTPUT "Total sunshine was ", Hours, " hours and ", Minutes, " minutes"
OUTPUT "The highest daily sunshine was ", MaxHours, " hours and ",
MaxMinutes, " minutes"
```

<table>
<tr><th colspan="4">Marking Instructions in italics</th></tr>
<tr><td colspan="4">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</td></tr>
<tr><td>0</td><td>1–3</td><td>4–6</td><td>7–9</td></tr>
<tr><td rowspan="2">No creditable response.</td><td>At least one programming technique has been used.
Any use of selection, iteration, counting, totalling, input and output.</td><td>Some programming techniques used are appropriate to the problem.
More than one technique seen applied to the scenario, check the list of techniques needed.</td><td>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.</td></tr>
<tr><td>Some data has been stored but not appropriately.
Any use of variables or arrays or other language dependent data structures e.g. Python lists.</td><td>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.</td><td>The data structures chosen are appropriate and store all the data required.
The data structures used store all the data required by the scenario.</td></tr>
</table>

</td><td></td><td></td></tr>
</table>

© Cambridge University Press & Assessment 2026

0478/21 Mark Scheme June 2026

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>12</td><td>
<table>
<tr><td colspan="4">Marking Instructions in italics</td></tr>
<tr><td colspan="4">AO3: Provide solutions to problems by:
• evaluating computer systems
• making reasoned judgements
• presenting conclusions</td></tr>
<tr><td>0</td><td>1–2</td><td>3–4</td><td>5–6</td></tr>
<tr><td rowspan="4">No creditable response.</td><td>Program seen without relevant comments.</td><td>Program seen with some relevant comment(s).</td><td>The program has been fully commented.</td></tr>
<tr><td>Some identifier names used are appropriate.
Some of the data structures used have meaningful names.</td><td>The majority of identifiers used are appropriately named.
Most of the data structures used have meaningful names.</td><td>Suitable identifiers with names meaningful to their purpose have been used throughout.
All of the data structures used have meaningful names.</td></tr>
<tr><td>The solution is illogical.</td><td>The solution contains parts that may be illogical.</td><td>The program is in a logical order.</td></tr>
<tr><td>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.</td><td>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.</td><td>The solution is accurate.
Solution logically performs all the tasks given in the scenario. Ignore minor syntax errors.</td></tr>
</table>
</td><td></td><td></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>12</td><td>
<table>
<tr><th>0</th><th>1-2</th><th>3-4</th><th>5-6</th></tr>
<tr><td></td><td>The solution attempts at least one of the requirements.
Solution contains lines of code that attempt at least one task given in the scenario.</td><td>The solution meets most of the requirements.
Solution contains lines of code that perform most tasks given in the scenario.</td><td>The solution meets all the requirements given in the question.
Solution performs all the tasks given in the scenario.</td></tr>
</table>
</td><td></td><td></td></tr>
</table>

PapaCambridge papacambridge.com

© Cambridge University Press & Assessment 2026