CAMBRIDGE
International Education

Cambridge IGCSE™

COMPUTER SCIENCE **0478/22**

Paper 2 Algorithms, Programming and Logic **May/June 2025**

MARK SCHEME

Maximum Mark: 75

Published

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the May/June 2025 series for most Cambridge IGCSE, Cambridge International A and AS Level components, and some Cambridge O Level components.

Generic Marking Principles

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptions for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

GENERIC MARKING PRINCIPLE 2:

Marks awarded are always **whole marks** (not half marks, or other fractions).

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

GENERIC MARKING PRINCIPLE 4:

Rules must be applied consistently, e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

GENERIC MARKING PRINCIPLE 5:

Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).

GENERIC MARKING PRINCIPLE 6:

Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

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
	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

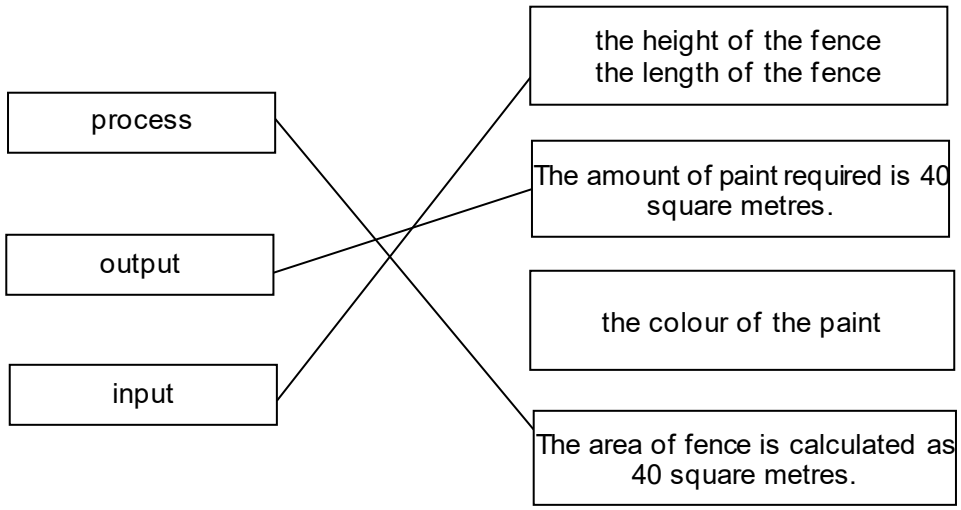
Annotation	Meaning
R2	Required item two
R3	Required item three
✓ 1	Correct awarding one mark
✓ 2	Correct awarding two marks
✓ 3	Correct awarding three marks
✓ 4	Correct awarding four marks
✓ 5	Correct awarding five marks
✓ 6	Correct awarding six marks
✓ 7	Correct awarding seven marks
✓ 8	Correct awarding eight marks
✓ 9	Correct awarding nine marks

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.

Question	Answer	Marks
1	D	**1**

Question	Answer	Marks
2	**One** mark for each correct line, **max three** **Component part** **Example statements**  	**3**

Question	Answer	Marks
3	**One** mark per correct piece of test data, **Normal test data:** −50 (any integer value between 0 and −100, inclusive) **Abnormal test data:** 50 (any value outside the range 0 and −100, inclusive, or any non-numeric input and non-integers **Extreme test data**: 0 // −100 **Boundary test data:** 1 // 0 // −100 // −101	**4**

Question	Answer	Marks
4(a)	To check that the value entered has not changed on input.	**1**

Question	Answer	Marks

4(b)	**One** mark per mark point, **max six**
	
	MP1 At least one appropriate working loop
	MP2 First inputs of numbers all stored in array `Numbers`
	MP3 Correct second input made for each number
	MP4 Working selection statement to compare first and second inputs one at a time
	MP5 If numbers don't match, appropriate output, including both values input and a message requiring re-input
	MP6 New input stored in current array location (replacing original input)
	MP7 Final output confirming completion, **outside loop**.
	
	For example,
	
	`// variable and array declarations or initialisations`
	`// are not required for this question`
	`FOR Index ← 1 TO 10`
	`  INPUT Numbers[Index]`
	`NEXT Index`
	`FOR Index ← 1 TO 10`
	`  INPUT CheckNumber`
	`  IF CheckNumber <> Numbers[Index]`
	`  THEN`
	`  OUTPUT CheckNumber, " and ", Numbers[Index], " do not match. Please re-enter the number. "`
	`  INPUT Numbers[Index]`
	`  ENDIF`
	`NEXT Index`
	`OUTPUT "The check has been completed."`
		6

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(a)</td><td>One mark per mark point

• Line 01 / <code>DECLARE Names : ARRAY[1:500] OF REAL</code>
should be <code>DECLARE Names : ARRAY[1:500] OF STRING</code>

• Line 09 / Total ← <code>100</code>
should be Total ← <code>0</code>

• Line 11 / <code>INPUT Names[Index]</code>
should be <code>INPUT Names[Counter]</code>

• Line 13 / <code>Total ← Counter + Heights[Counter]</code>
should be <code>Total ← Total + Heights[Counter]</code>

• Line 20 / <code>OUTPUT "The shortest height is ", Heights</code>
should be <code>OUTPUT "The shortest height is ", Shortest</code>
or <code>OUTPUT "The shortest height is ", Heights[Index]</code></td><td>5</td></tr>
<tr><td></td><td><pre>
01 DECLARE Names : ARRAY[1:500] OF <u>STRING</u>
02 DECLARE Heights : ARRAY[1:500] OF REAL
03 DECLARE Shortest : REAL
04 DECLARE Total : REAL
05 DECLARE Counter : INTEGER
06 DECLARE Index : INTEGER
07 Shortest ← 500
08 Index ← 0
09 Total ← <u>0</u>
10 FOR Counter ← 1 TO 500
11   INPUT Names[<u>Counter</u>]
12   INPUT Heights[Counter]
13   Total ← <u>Total</u> + Heights[Counter]
14   IF Heights[Counter] < Shortest
15   THEN
16   Shortest ← Heights[Counter]
17   Index ← Counter
18   ENDIF
19 NEXT Counter
20 OUTPUT "The shortest height is ", <u>Shortest</u>
21 OUTPUT "The shortest person is ", Names[Index]
22 OUTPUT "The average height is ", Total / 500
</pre></td><td></td></tr>
<tr><td>5(b)</td><td>One mark per mark point

MP1 Output statement with correct calculation of average, or use of a variable average.
MP2 Correct use of <code>ROUND</code> function set to 1 decimal place.

For example,

<code>OUTPUT ROUND(Total / 500, 1)</code></td><td>2</td></tr>
</table>

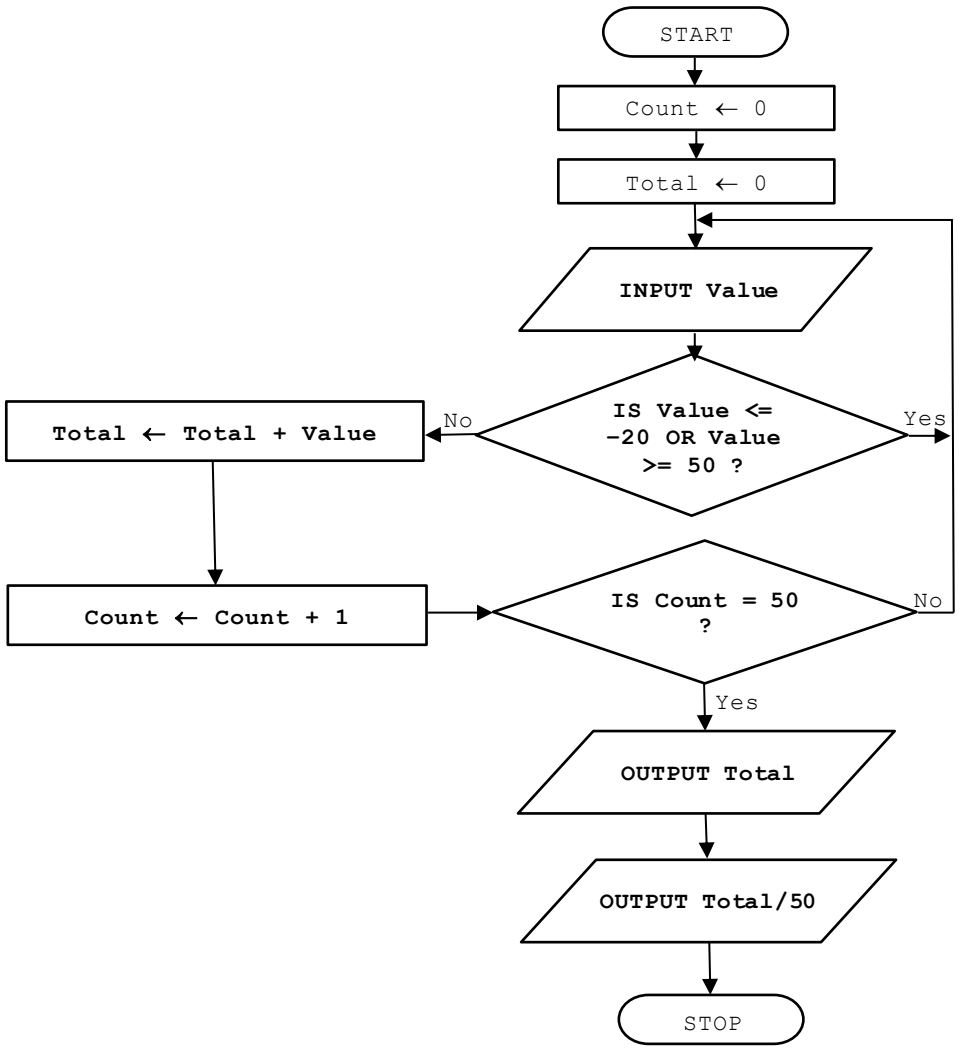
Question	Answer	Marks
5(c)	**One** mark per statement, **max four** MP1 Declaration of new variable for tallest person / largest value at start of algorithm MP2 Initialisation of largest variable to a low number e.g. 0 MP3 New selection statement after input of height to compare it with current largest value MP4 If input of height is larger than current largest variable, it should become the new largest value MP5 **Outside the loop**, output the current value of the largest variable.	**4**

Question	Answer	Marks
6(a)	**One** mark per mark point MP1 Correct `Number` and `Index` columns MP2 Correct `Word` column MP3 Correct `Continue` column MP4 Correct `V1` and `V2` column MP5 Correct `L1` and `L2` column MP6 Correct OUTPUT column	**6**

Number	Index	Word	Continue	V1	V2	L1	L2	OUTPUT
2	1	RACECAR	TRUE	1	7	R	R	
				2	6	A	A	
				3	5	C	C	
				4	4			
								Successful
	2	TREAT	TRUE	1	5	T	T	
				2	4	R	A	
			FALSE					NOT successful

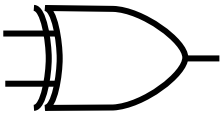
6(b)	Finds words that read the same forwards and backwards.	**1**

Question	Answer	Marks
6(c)	**One** mark per mark point, **max three** MP1 Uses an index to mark the start and end of each word // uses `V1` to mark the start of the word and `V2` to mark the end MP2 After each check, these are moved to the next character in the word at each end of the word // `V1` is incremented by 1 and `V2` is decremented by 1 after each check. MP3 When the two indexes match or cross, the whole word has been checked // When `V1` = `V2` or `V1 > V2`, the whole word has been checked MP4 The algorithm will exit if it is found that two compared letters are not the same.	**3**

Question	Answer	Marks
7(a)	**One** mark per mark point, max **six** MP1 input `Value` MP2 correct check that Value is between −20 and 50, exclusive MP3 totalling values MP4 counting number of values MP5 checking `Count = 50` / `>= 50` MP6 output of total and average 	**6**
7(b)	**One** mark per statement MP1 Change INPUT Value so that it saves values into an array MP2 … using the counter as the index MP3 **Outside the (first) loop**, set up a loop that will iterate 50 times MP4 Use an output statement inside the (second) loop to output the contents of the array, using the counter as the index.	**4**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>8(a)</td><td>One mark for the logic gate and one mark for the truth table

XOR logic gate symbol:

Truth table:

<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td><td>2</td></tr>
<tr><td>8(b)(i)</td><td>One mark per mark point

MP1 A AND
MP2 (NOT B OR NOT C) // NOT(B AND C) // (B NAND C)

Or

One mark per mark point

MP3 A AND NOT B
MP4 OR A AND NOT C

Example answers

(X =) A AND (NOT B OR NOT C) // A AND NOT (B AND C) // A AND (B NAND C)
(X =) A AND NOT B OR A AND NOT C</td><td>2</td></tr>
</table>

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>8(b)(ii)</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>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>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</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></tr>
</table>

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>9(a)</td><td>Two marks for all four fields correctly identified
One mark for two or three fields correctly identified

<table>
<tr><th>Data type</th><th>Field</th></tr>
<tr><td>Boolean</td><td>Subway</td></tr>
<tr><td>integer</td><td>Population // Founded // Identification</td></tr>
<tr><td>real</td><td>AverageIncome</td></tr>
<tr><td>text</td><td>Name // Country // Continent // Identification</td></tr>
</table></td><td>2</td></tr>
<tr><td>9(b)</td><td>One mark per mark point

MP1 All correct fields in <code>SELECT</code>
MP2 Correct use of key word <code>FROM</code>
MP3 Correct field in <code>WHERE</code>: <code>Continent</code>
MP4 Correct criterion in <code>WHERE</code>: <code>"Asia"</code> // <code>'Asia'</code>

Correct code:

<pre>SELECT Name, Country, Population
FROM CapitalCity
WHERE Continent = "Asia"; // 'Asia';</pre></td><td>4</td></tr>
</table>

Question	Answer	Marks
10	Due to an issue with this question, the question has been removed from the question paper.	**15**

