

# Cambridge IGCSE™

---

**COMPUTER SCIENCE****0478/22**

Paper 2 Algorithms, Programming and Logic

**May/June 2024**

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 2024 series for most Cambridge IGCSE, Cambridge International A and AS Level and Cambridge Pre-U components, and some Cambridge O Level components.

---

This document consists of **16** printed pages.

**PUBLISHED****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.

**Mark scheme abbreviations**

/ separates alternative words / phrases within a marking point

// separates alternative answers within a marking point



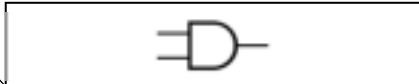

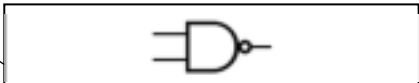


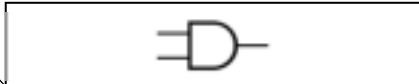

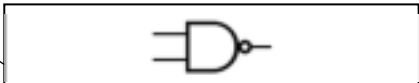


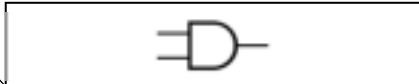

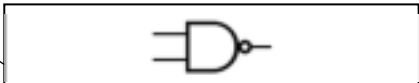
underline actual word given must be used by candidate (grammatical variants accepted)

**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	C	1

Question	Answer	Marks												
2	<p><b>One mark for each correct line</b></p> <table border="0" style="width: 100%;"> <thead> <tr> <th style="text-align: center; width: 30%;">Logic function</th> <th style="text-align: center; width: 40%;">Standard symbol</th> </tr> </thead> <tbody> <tr> <td style="text-align: center; border: 1px solid black; padding: 5px;">AND</td> <td style="text-align: center; border: 1px solid black; padding: 5px;"></td> </tr> <tr> <td style="text-align: center; border: 1px solid black; padding: 5px;">XOR</td> <td style="text-align: center; border: 1px solid black; padding: 5px;"></td> </tr> <tr> <td style="text-align: center; border: 1px solid black; padding: 5px;">NAND</td> <td style="text-align: center; border: 1px solid black; padding: 5px;"></td> </tr> <tr> <td style="text-align: center; border: 1px solid black; padding: 5px;">OR</td> <td style="text-align: center; border: 1px solid black; padding: 5px;"></td> </tr> <tr> <td style="text-align: center; border: 1px solid black; padding: 5px;"></td> <td style="text-align: center; border: 1px solid black; padding: 5px;"></td> </tr> </tbody> </table>	Logic function	Standard symbol	AND		XOR		NAND		OR				<b>4</b>
Logic function	Standard symbol													
AND														
XOR														
NAND														
OR														
														

Question	Answer	Marks
3	<p><b>One</b> mark for each correct answer <b>max three</b></p> <p>MP1    abstraction  MP2    decomposition  MP3    identification of problem  MP4    identification of requirements // outline of success criteria</p>	<b>3</b>

Question	Answer	Marks
4(a)	<p><b>One</b> mark per mark point</p> <p>MP1    length check ...  MP2    ... to ensure the product code entered is 6 characters in length  MP3    format check ...  MP4    ... to ensure the first two characters of the product code entered are "PD"  MP5    range check ...  MP6    ... to ensure that the value of the last four figures of the product code entered is between 1000 and 9999</p>	<b>6</b>
4(b)(i)	<p><b>One</b> mark for correct use of LENGTH operation, <b>one</b> mark for appropriate test</p> <p>Example:</p> <pre>REPEAT   INPUT Product UNTIL LENGTH(Product) = 6</pre>	<b>2</b>
4(b)(ii)	<p><b>One</b> mark for correct use of SUBSTRING operation, <b>one</b> mark for appropriate test</p> <p>Example:</p> <pre>REPEAT   INPUT Product UNTIL SUBSTRING(Product, 1, 2) = "PD"</pre>	<b>2</b>

Question	Answer	Marks
5	<p><b>One</b> mark for each description, <b>one</b> mark for each example</p> <ul style="list-style-type: none"> <li>• arithmetic – used in calculations (1)    <math>A \leftarrow B + C</math> (1)</li> <li>• Boolean – used for operations with true or false values (1)    <code>IF B AND C</code> (1)</li> <li>• logical – used in comparisons/conditional statements/selection statements (1)    <code>IF B &gt; C</code> (1)</li> </ul>	6

Question	Answer	Marks
6(a)	<p><b>One</b> mark for: MP1    adding current value to total</p> <p><b>One</b> mark for each point <b>max three</b>.</p> <p>MP2    input more than one number MP3    setting total to zero before loop MP4    correct use of loop including terminal condition MP5    output total <b>after</b> loop</p> <p>Example:</p> <pre>Total ← 0 INPUT Value WHILE Value &lt;&gt; 9999.9     Total ← Total + Value     INPUT Value ENDWHILE OUTPUT Total  Value ← 0 Total ← 0 REPEAT     Total ← Total + Value     INPUT Value UNTIL Value = 9999.9 OUTPUT Total</pre>	4

Question	Answer	Marks
6(b)	<p><b>One</b> mark for each point</p> <p>MP1 adding one to counter  MP2 <b>correct</b> use of selection, if current value &gt; 100 THEN ... ENDIF</p> <p><b>One</b> mark for each point, <b>max two</b></p> <p>MP3 input more than one number  MP4 setting counter to zero before loop  MP5 correct use of loop including terminal condition  MP6 output value of counter <b>after loop</b></p> <p>Example:</p> <pre>Counter ← 0 INPUT Value WHILE Value &lt;&gt; 9999.9   IF Value &gt; 100     THEN       Counter ← Counter + 1     ENDIF   INPUT Value ENDWHILE OUTPUT Counter</pre>	4

Question	Answer	Marks
7(a)	<p>01//02//06//10  04(07) and/or 08  03(12)</p>	3

Question	Answer	Marks
7(b)	<p><b>One mark for each error identified and corrected</b></p> <p>Line 04 &lt; should be &gt;</p> <p>Line 08 Count should be Counter</p> <p>Line 11 ENDWHILE should be ENDIF</p> <pre> 01 Max ← List[1] 02 Min ← List[1] 03 FOR Counter ← 2 TO 1000 04     IF List[Counter] &gt; Max 05         THEN 06             Max ← List[Counter] 07     ENDIF 08     IF List[Counter] &lt; Min 09         THEN 10         Min ← List[Counter] 11     <b>ENDIF</b> 12 NEXT Counter 13 OUTPUT "Maximum value is ", Max 14 OUTPUT "Minimum value is ", Min </pre>	3

Question	Answer	Marks
8(a)	<p>X =                                    <b>1 mark</b></p> <p>(A AND B) // A AND B            <b>1 mark</b></p> <p>AND NOT C                            <b>1 mark</b></p> <p>X = (A AND B) AND NOT C</p>	3



Question	Answer	Marks																																				
8(b)	<p> <b>Four</b> marks for 8 correct outputs  <b>Three</b> marks for 6/7 correct outputs  <b>Two</b> marks for 4/5 correct outputs  <b>One</b> mark for 2/3 correct outputs                 </p> <table border="1" data-bbox="338 384 1072 971"> <thead> <tr> <th>A</th> <th>B</th> <th>C</th> <th>X</th> </tr> </thead> <tbody> <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>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</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> </tbody> </table>	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	0	4
A	B	C	X																																			
0	0	0	0																																			
0	0	1	0																																			
0	1	0	0																																			
0	1	1	0																																			
1	0	0	0																																			
1	0	1	0																																			
1	1	0	1																																			
1	1	1	0																																			

Question	Answer							Marks																																																																																																																																								
9(a)	<b>One</b> mark for each of columns <b>A</b> , <b>B</b> and <b>T</b> <b>Two</b> marks for columns <b>List[1]</b> to <b>List[5]</b> all entries correct or <b>One</b> mark for columns <b>List[1]</b> to <b>List[5]</b> with one error							<b>5</b>																																																																																																																																								
	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 5%;">A</th> <th style="width: 5%;">B</th> <th style="width: 10%;">List[1]</th> <th style="width: 10%;">List[2]</th> <th style="width: 10%;">List[3]</th> <th style="width: 10%;">List[4]</th> <th style="width: 10%;">List[5]</th> <th style="width: 5%;">T</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td>15</td> <td>17</td> <td>20</td> <td>5</td> <td>9</td> <td></td> </tr> <tr> <td>FALSE</td> <td>1</td> <td>17</td> <td>15</td> <td></td> <td></td> <td></td> <td>15</td> </tr> <tr> <td>TRUE</td> <td>2</td> <td></td> <td>20</td> <td>15</td> <td></td> <td></td> <td>15</td> </tr> <tr> <td>TRUE</td> <td>3</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>TRUE</td> <td>4</td> <td></td> <td></td> <td></td> <td>9</td> <td>5</td> <td>5</td> </tr> <tr> <td></td> <td>5</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>FALSE</td> <td>1</td> <td>20</td> <td>17</td> <td></td> <td></td> <td></td> <td>17</td> </tr> <tr> <td>TRUE</td> <td>2</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>3</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>4</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>5</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>FALSE</td> <td>1</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>2</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>3</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>4</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>5</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>								A	B	List[1]	List[2]	List[3]	List[4]	List[5]	T			15	17	20	5	9		FALSE	1	17	15				15	TRUE	2		20	15			15	TRUE	3							TRUE	4				9	5	5		5							FALSE	1	20	17				17	TRUE	2								3								4								5							FALSE	1								2								3								4								5						
	A	B	List[1]	List[2]	List[3]	List[4]	List[5]		T																																																																																																																																							
			15	17	20	5	9																																																																																																																																									
	FALSE	1	17	15					15																																																																																																																																							
	TRUE	2		20	15				15																																																																																																																																							
	TRUE	3																																																																																																																																														
	TRUE	4				9	5		5																																																																																																																																							
		5																																																																																																																																														
	FALSE	1	20	17					17																																																																																																																																							
	TRUE	2																																																																																																																																														
		3																																																																																																																																														
		4																																																																																																																																														
		5																																																																																																																																														
	FALSE	1																																																																																																																																														
		2																																																																																																																																														
		3																																																																																																																																														
		4																																																																																																																																														
	5																																																																																																																																															

Question	Answer	Marks
9(b)	<p><b>One</b> mark for each point</p> <p>MP1 (bubble) sort data in array</p> <p>MP2 in descending order</p>	<b>2</b>

Question	Answer	Marks										
10(a)	ContractNumber	<b>1</b>										
10(b)	<p><b>One</b> mark for every <b>two</b> correct data types</p> <table border="1" data-bbox="338 608 1014 938"> <thead> <tr> <th>Field</th> <th>Data type</th> </tr> </thead> <tbody> <tr> <td>ContractNumber</td> <td>text/alphanumeric</td> </tr> <tr> <td>Months</td> <td>integer</td> </tr> <tr> <td>EndDate</td> <td>date/time</td> </tr> <tr> <td>Sport</td> <td>Boolean</td> </tr> </tbody> </table>	Field	Data type	ContractNumber	text/alphanumeric	Months	integer	EndDate	date/time	Sport	Boolean	<b>2</b>
Field	Data type											
ContractNumber	text/alphanumeric											
Months	integer											
EndDate	date/time											
Sport	Boolean											
10(c)	<p><b>One</b> mark for each point</p> <p>MP1 to find the total number of months for <b>all</b> contracts</p> <p>MP2 to find the number of contracts</p> <p>MP3 ... that are subscribed to News</p>	<b>3</b>										
10(d)	<p>ContractNumber</p> <p>News AND Sport // Sport AND News</p> <p>Example answer:</p> <pre>SELECT ContractNumber FROM Contract WHERE News // News = TRUE AND Sport // Sport = TRUE ;</pre>	<b>2</b>										

Question	Answer	Marks
11	<p><b>Data Structures required</b> with names as given in the scenario: Arrays or lists <u>Grid</u></p> <p><b>Requirements (techniques)</b></p> <p><b>R1</b> Set up game – generate random cell, clear all other cells in array, set player start position and start player moves counter (iteration, use of arrays and library routines (round and random))</p> <p><b>R2</b> Input and check move – is it valid? (input, output, iteration and selection)</p> <p><b>R3</b> Decide outcome – has move found the X? If so, give appropriate output. If not increment counter and continue. If 10 moves exceeded, give appropriate output (use of arrays, iteration, selection and output).</p> <p><b>Example 15-mark answer in pseudocode</b></p> <pre>// Set up game FOR Row ← 1 TO 5   FOR Column ← 1 TO 5     Grid[Row, Column] ← '' // set grid cells to be empty   NEXT Column NEXT Row</pre>	15

Question	Answer	Marks
11	<pre> REPEAT // not in cell 1,1   XRow ← ROUND ((RANDOM() * 4) + 1, 0) // Random row position between 1 and 5 in GRID   XColumn ← ROUND ((RANDOM() * 4) + 1, 0) // Random column position between 1 and 5 in GRID UNTIL XRow &lt;&gt; 1 and XColumn &lt;&gt; 1 // not in cell 1,1  Grid [XRow, XColumn] ← 'X' MaxMove ← 10 NumberMoves ← 0 PlayerRow ← 1 PlayerColumn ← 1 Win ← FALSE  // during game WHILE NumberMoves &lt; MaxMove AND NOT Win   MoveError ← FALSE   OUTPUT "Please enter your move, L - Left, R - Right, U - Up or D - Down"   INPUT UPPER(PlayerMove)   REPEAT     CASE OF PlayerMove       'L' : TempColumn ← PlayerColumn - 1       'R' : TempColumn ← PlayerColumn + 1       'U' : TempRow ← PlayerRow - 1       'D' : TempRow ← PlayerRow + 1       OTHERWISE MoveError ← TRUE     ENDCASE    // check for out-of-range moves   IF TempColumn &lt; 1 or TempColumn &gt; 5     THEN       MoveError ← TRUE     ELSE       PlayerColumn ← TempColumn   ENDIF </pre>	

Question	Answer	Marks
11	<pre> IF TempRow &lt; 1 or TempRow &gt; 5   THEN     MoveError ← TRUE   ELSE     PlayerRow ← TempRow   ENDIF  // check win if X Found IF Grid [PlayerRow, PlayerColumn] = 'X'   THEN     OUTPUT "You Win"     Win ← TRUE   ELSE     IF NOT MoveError       THEN         NumberMoves ← NumberMoves + 1       ENDIF     ENDIF   UNTIL NOT MoveError ENDWHILE  IF NOT Win   THEN     OUTPUT "You Lose"   ENDIF </pre>	

<b>Marking Instructions in italics</b>			
<b>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</b>			
<b>0</b>	<b>1–3</b>	<b>4–6</b>	<b>7–9</b>
No creditable response.	At least one programming technique has been used.  <i>Any use of selection, iteration, counting, totalling, input and output.</i>	Some programming techniques used are appropriate to the problem.  <i>More than one technique seen applied to the scenario, check the list of techniques needed.</i>	The range of programming techniques used is appropriate to the problem.  <i>All criteria stated for the scenario have been covered by the use of appropriate programming techniques, check the list of techniques needed.</i>
	Some data has been stored but not appropriately.  <i>Any <b>use</b> of variables or arrays or other language dependent data structures e.g. Python lists.</i>	Some of the data structures chosen are appropriate and store some of the data required.  <i>More than one data structure <b>used</b> to store data required by the scenario.</i>	The data structures chosen are appropriate and store all the data required.  <i>The data structures <b>used</b> store all the data required by the scenario.</i>

<b>Marking Instructions in italics</b>			
<b>AO3: Provide solutions to problems by:</b>			
<ul style="list-style-type: none"> <li><b>evaluating computer systems</b></li> <li><b>making reasoned judgements</b></li> <li><b>presenting conclusions</b></li> </ul>			
<b>0</b>	<b>1–2</b>	<b>3–4</b>	<b>5–6</b>
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.  <i>Some of the data structures used have meaningful names.</i>	The majority of identifiers used are appropriately named.  <i>Most of the data structures used have meaningful names.</i>	Suitable identifiers with names meaningful to their purpose have been used throughout.  <i>All of the data structures used have meaningful names.</i>
	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.  <i>Solution contains few lines of code with errors that attempt to perform a task given in the scenario</i>	The solution contains parts that are inaccurate.  <i>Solution contains lines of code with some errors that logically perform tasks given in the scenario. Ignore minor syntax errors.</i>	The solution is accurate.  <i>Solution logically performs all the tasks given in the scenario. Ignore minor syntax errors.</i>
	The solution attempts at least one of the requirements.  <i>Solution contains lines of code that attempt at least one task given in the scenario.</i>	The solution attempts to meet most of the requirements.  <i>Solution contains lines of code that attempt most tasks given in the scenario.</i>	The solution meets all the requirements given in the question.  <i>Solution performs all the tasks given in the scenario.</i>