



**Cambridge International Examinations**

Cambridge International General Certificate of Secondary Education

---

**COMPUTER SCIENCE**

**0478/22**

Paper 2 Algorithms, Programming and Logic

**February/March 2024**

MARK SCHEME

Maximum Mark : 75

---



**Cambridge Assessment  
International Education**

[Turn over



**Cambridge Assessment  
International Education**

INSERT STANDARD FRONT PAGE

## 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 descriptors 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.

Please note the following further points:

Please also read the additional guidance in the mark scheme, it provides further information about how to mark the question.

Please annotate your scripts. The number of ticks given must match the number of marks given.

If you award a benefit of doubt (BOD) mark, this must also have a tick.

The sections in brackets in the mark scheme are not necessary in the candidates answer.

The words in bold in the mark scheme are important text that needs to be present, or some notion of it needs to be present. It doesn't have to be the exact word, but something close to the meaning.

If a word is underlined, this exact word must be present.

A single forward slash means this is an alternative word. A double forward slash means that this is an alternative mark point.

Red text in the mark scheme is a response that we think it just about okay, but that will not be published as a response.

Ellipsis (...) on the end of one mark point and the start of the next means that the candidate cannot get the second mark point without being awarded the first one.

Please inform your team leader when you have submitted your standardisation scripts. They may have many examiners that they are monitoring, so may not see your submission immediately.

The mark scheme has been agreed at standardisation, so it is the AE's responsibility to apply it. You may have differing opinions on how the paper should be marked, but the mark scheme is what has been agreed by a panel and AE's are required to apply it.

Please mark your allocation at a steady rate. If you are not able to mark your allocation for a number of days, inform your team leader, do not wait for them to have to chase you.

Please ensure that you meet the 40% deadline. If you are not close to or meeting this at the 40% deadline, some of your allocation may be reallocated. If you are going to struggle with this deadline, you must inform your team leader ASAP.

There are blank pages at the start of the exam paper. These needs to be annotated with a SEEN annotation, to indicate it has been checked for any further responses.

If a candidate writes outside the zoned area for the question, this must be linked to the response, even if it is not awarded a mark. This demonstrates at EAR that you did read this part of the response.

If a candidate has crossed out the final section of a response to a question and has not written anything after the crossed-out work, the crossed-out work must be marked.

## Mechanics of Marking:

Every mark given should have a corresponding tick on the script. The number of ticks on each (part) question should match the number of marks awarded for that (part) question. If giving Benefit of Doubt, **the BOD must be accompanied by a tick.**

If a candidate has not given a response or the response is in no way related to the question, such as 'don't know', NR (the Hash key) should be awarded rather than zero.

**Every part question must be annotated to show that it has been read even if awarding NR. Please ensure that all part questions that are marked as NR are also annotated with the SEEN icon. This is a requirement of RM3.**

There are **two** blank pages at the start of each script that must be annotated with the SEEN icon. 😞

**NEW: Words or phrases that are underlined, must be present in the candidate's answer. Words or phrases that are emboldened indicate that the idea represented by the bold text must be included.**

Even though the comments box is visible at the bottom of the screen, please do not put comments or question marks on the scripts. When scripts are returned to centres all the annotations including comments, are visible.

If work has been crossed out and something written in its place, the replacement work is marked even if the crossed-out work is correct. If the crossed-out work has not been replaced, mark the crossed-out answer. Please also annotate the unmarked work as SEEN, especially if the replacement answer is on a separate sheet.

For single mark answers, mark the first answer on the line, unless there is a note to the contrary on the mark scheme.

If a candidate writes something that is not enough (NE) for a mark, but is not actually incorrect, continue reading, even if the mark scheme says, for example, mark first two answers.

Question	Answer	Marks	Guidance												
1	<b>B</b>	1													
2(a)	<p><b>One mark for each correct line(s) from the test data type</b></p> <table border="0" style="width: 100%;"> <thead> <tr> <th style="text-align: left; width: 30%;">Test data type</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td style="border: 1px solid black; padding: 5px;">abnormal</td> <td style="border: 1px solid black; padding: 5px;">a value that is accepted</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">boundary</td> <td style="border: 1px solid black; padding: 5px;">a value that is the highest or lowest value to be accepted and the corresponding lowest or highest value to be rejected</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">extreme</td> <td style="border: 1px solid black; padding: 5px;">a value that is the highest or lowest value to be rejected</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">normal</td> <td style="border: 1px solid black; padding: 5px;">a value that is rejected</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;"></td> <td style="border: 1px solid black; padding: 5px;">a value that is the highest or lowest value to be accepted</td> </tr> </tbody> </table>	Test data type	Description	abnormal	a value that is accepted	boundary	a value that is the highest or lowest value to be accepted and the corresponding lowest or highest value to be rejected	extreme	a value that is the highest or lowest value to be rejected	normal	a value that is rejected		a value that is the highest or lowest value to be accepted	4	
Test data type	Description														
abnormal	a value that is accepted														
boundary	a value that is the highest or lowest value to be accepted and the corresponding lowest or highest value to be rejected														
extreme	a value that is the highest or lowest value to be rejected														
normal	a value that is rejected														
	a value that is the highest or lowest value to be accepted														
2(b)	<p><b>One mark for each point:</b></p> <ul style="list-style-type: none"> <li>• abnormal for example 31</li> <li>• boundary 4 <b>and</b> 5 // 10 <b>and</b> 11</li> <li>• extreme 5 // 10</li> <li>• normal for example 6</li> </ul>	4	Discuss FT at STM												

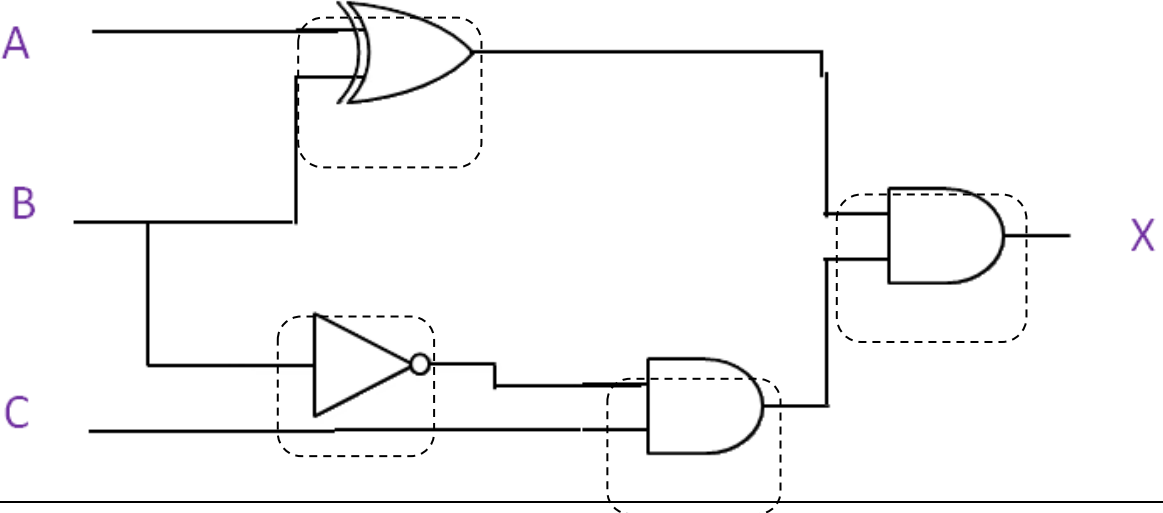
Question	Answer	Marks	Guidance
3(a)	<p><b>One mark for each point max four.</b></p> <ul style="list-style-type: none"> <li>• input value, outside loop</li> <li>• correct use of loop</li> <li>• checking value input against contents of array ...</li> <li>• .. appropriate action</li> <li>• correct outputs</li> </ul> <p><b>Example</b></p> <pre> INPUT MyNumber Found ← 0 FOR Index ← 1 TO 50   IF Values[Index] = MyNumber     THEN       Found ← Index     ENDIF NEXT IF Found = 0   THEN     OUTPUT "Not found"   ELSE     OUTPUT Found ENDIF </pre>	<b>4</b>	Discuss other solutions at STM

Question	Answer	Marks	Guidance
3(b)	<p><b>One mark for each point max four.</b></p> <ul style="list-style-type: none"> <li>• use of inner and outer loop</li> <li>• correct use of loops</li> <li>• checking value adjacent values in array ...</li> <li>• ... swap if required</li> <li>• correct stopping condition</li> </ul> <pre> Last ← 50 Repeat   Swap ← FALSE   FOR Index ← 1 TO Last - 1     IF Values[Index] &gt; Values[Index + 1]       THEN         Temp ← Values[Index]         Values[Index] ← Values[Index + 1]         Values[Index + 1] ← Temp       Swap ← TRUE     ENDIF   NEXT   Last ← Last - 1 UNTIL NOT Swap or Last = 1 </pre>	4	Discuss solutions at STM
4	<p><b>One mark for each point max three.</b></p> <ul style="list-style-type: none"> <li>• integer</li> <li>• real</li> <li>• char</li> <li>• string</li> <li>• Boolean</li> </ul>	3	Discuss other data types at STM. Especially those used in Python, Java and VB (recommend languages).



Question	Answer	Marks	Guidance
5(a)	<p><b>One</b> mark for each error identified and correction</p> <ul style="list-style-type: none"> <li>Line 05 OUTPUT should be INPUT</li> <li>Line 06 OR should be AND</li> <li>Line 10 NEXT should be ENDIF</li> </ul>	3	
5(b)(i)	<p><b>One</b> mark for checking for <math>&lt; 0</math> or <math>\geq 0</math></p> <p><b>One</b> mark for repeating input e.g. use of REPEAT WHILE or using existing loop</p> <p><b>One</b> mark for checking both inputs</p> <p>For example</p> <pre> REPEAT   OUTPUT "Enter cost price "   INPUT Cost UNTIL Cost &gt;= 0 REPEAT   OUTPUT "Enter selling price "   INPUT Sell UNTIL Sell &gt;= 0 </pre> <p><b>or</b></p> <pre> OUTPUT "Enter cost price " INPUT Cost WHILE Cost &lt; 0   OUTPUT "Enter cost price "   INPUT Cost ENDWHILE OUTPUT "Enter selling price " INPUT Sell WHILE Sell &lt; 0   OUTPUT "Enter selling price "   INPUT Sell ENDWHILE </pre>	3	Solutions are examples only
5(b)(ii)	<p><b>One</b> mark for identifying validation check and <b>one</b> mark for accompanying description <b>max four</b></p> <ul style="list-style-type: none"> <li>presence check (1) to check that values have been input (1)</li> <li>type check (1) to check for numerical values (1)</li> </ul>	4	Not range check as it is given in part g(i).

Question	Answer	Marks	Guidance
6	<b>One</b> mark for identifying a type of iteration, <b>one</b> mark for accompanying description <b>max four</b> <ul style="list-style-type: none"><li>• count controlled (1) number of iterations is pre-determined (1)</li><li>• pre-condition (1) checks condition at start of loop // loop may not iterate (1)</li><li>• post-condition (1) checks condition at end of loop // loop always iterates at least once(1)</li></ul>	4	Allow FT for description if code for loop is given   e.g. FOR . . NEXT Discuss omission of pre/post at STM

Question	Answer	Marks	Guidance																																				
<p>7(a)</p>	<p>One mark for each correct gate, with the correct input(s) as shown.</p> 	<p>4</p>	<p>Allow 1 mark if two AND gates reversed</p>																																				
<p>7(b)</p>	<table border="1" data-bbox="264 817 1003 1136"> <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>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </tbody> </table> <p>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</p>	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	1	1	1	0	0	1	1	1	0	<p>4</p>	
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	1																																				
1	1	0	0																																				
1	1	1	0																																				

Question	Answer					Marks	Guidance
8	<b>NumberGroups</b>	<b>Total</b>	<b>GroupSize</b>	<b>Average</b>	<b>OUTPUT</b>	4	
	0	0					
	1	7	7				
	2	17	10				
	3	19	2				
	4	27	8				
	5	30	3				
	6	39	9				
			0	6	Average group size 6		
	<p><b>One</b> mark for each column <b>Total</b>, <b>Average</b> and <b>OUTPUT</b>  <b>One</b> mark for columns <b>NumberGroups</b> and <b>GroupSize</b></p>						

Question	Answer	Marks	Guidance										
9(a)(i)	StorageID	1											
9(a)(ii)	It is a unique identifier	1											
9(b)	<p><b>One</b> mark for every <b>two</b> correct data types</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Data type</th> </tr> </thead> <tbody> <tr> <td>SizeMetres</td> <td>Real</td> </tr> <tr> <td>Position</td> <td>Char/Integer/Text</td> </tr> <tr> <td>Hoist</td> <td>Boolean/Text</td> </tr> <tr> <td>StorageID</td> <td>Text/Alphanumeric</td> </tr> </tbody> </table>	Field	Data type	SizeMetres	Real	Position	Char/Integer/Text	Hoist	Boolean/Text	StorageID	Text/Alphanumeric	2	
Field	Data type												
SizeMetres	Real												
Position	Char/Integer/Text												
Hoist	Boolean/Text												
StorageID	Text/Alphanumeric												
9©	<p><b>One</b> mark if two correct <b>or two</b> marks if completely correct</p> <pre>SELECT StorageID, PriceMonth, SizeMetres</pre> <p><b>One</b> mark each point max <b>two</b></p> <pre>FROM StorageUnits</pre> <pre>WHERE Hoist = TRUE;</pre>	4	Fields can be in any order in SELECT line.										
10	<p><b>One</b> mark for each point <b>max six</b></p> <ul style="list-style-type: none"> <li>• use comments ...</li> <li>• ... to explain the purpose of each section of code</li> <li>• ...for example, logic / syntax</li> <li>• use meaningful identifier names to ...</li> <li>• ... clearly identify the purpose ...</li> <li>• ... of variables, constants, arrays, procedures etc</li> <li>• use procedures and functions ...</li> <li>• ... to avoid repeated code</li> <li>• ... simplify logic</li> <li>• use indentation and white space ...</li> <li>• ... to make the program readable</li> </ul>	6											

Question	Answer	Marks
11	<p>Read the whole answer:</p> <p>Check if each requirement listed below has been met. Requirements may be met using a suitable built-in function from the programming language used (Python, VB or Java)</p> <p>Mark SEEN on script if requirement met, cross if no attempt seen, NE if partially met (see marked scripts). Use the tables for AO2 and AO3 below to award a mark in a suitable band using a best fit approach</p> <p>Then add up the total.</p> <p>Marks are available for:</p> <ul style="list-style-type: none"> <li>• AO2 (maximum 9 marks)</li> <li>• AO3 (maximum 6 marks)</li> </ul> <p><b>Data Structures required</b> the names shown underlined must match those given in the scenario  Arrays or lists <u>StudentName</u>, <u>ScreenTime</u>  Constant <u>ClassSize</u> could be variable</p> <p><b>Requirements (techniques)</b></p> <p><b>R1</b> allows a student to enter their weekly screen time and calculates the total number of minutes of screen time for each student in the week (input, iteration and totalling)</p> <p><b>R2</b> counts the number of days with more than 300 minutes screen time each day and calculates the average week's screen time for the whole class (selection, counting, iteration, calculating average)</p> <p><b>R3</b> finds the student with the lowest weekly minutes. Outputs for each student: name, total week's screen time in hours and minutes, number of days with more than 300 minutes screen time, outputs the average weeks screen time for the whole class and the name of the student with the lowest number of minutes (finding minimum value, output)</p>	15

Question	Answer	Marks
	<p><b>Example 15 mark answer in pseudocode</b></p> <pre> WeekLength ← 5 LowestMinutes ← 1000 ClassTotal ← 0 FOR StudentCounter ← 1 to ClassSize // loop for each student     Total ← 0     DaysOver300 ← 0     FOR DayCounter ← 1 to WeekLength // loop for each day         REPEAT             OUTPUT "Please enter number of minutes for day ", DayCounter             INPUT Minutes             UNTIL Minutes &gt;= 0             ScreenTime[StudentCounter, DayCounter] ← Minutes             Total ← Total + Minutes             IF Minutes &gt; 300                 THEN                     DaysOver300 ← DaysOver300 + 1             ENDIF             IF Minutes &lt; LowestMinutes                 THEN                     LowestMinutes ← Minutes                     LowestIndex ← StudentCounter             ENDIF         NEXT DayCounter         OUTPUT StudentName[StudentCounter]         OUTPUT "Screen time ", DIV(Total, 60), " hours ", MOD(Total, 60), " minutes "         OUTPUT "Days with more than 300 minutes screen time ", DaysOver300         ClassTotal ← ClassTotal + Total     NEXT StudentCounter  OUTPUT "Average weekly screen time for class ", ClassTotal / ClassSize, " minutes " OUTPUT "Lowest weekly time ", StudentNames[LowestIndex]</pre>	

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. <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>			
	<b>evaluating computer systems</b>	<b>making reasoned judgements</b>	<b>presenting conclusions</b>
<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 meets most of the requirements. <i>Solution contains lines of code that perform 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>