



# Cambridge IGCSE™

---

COMPUTER SCIENCE

0478/22

Paper 2

March 2021

MARK SCHEME

Maximum Mark: 50

---

**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 March 2021 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

---

This document consists of **8** printed pages.

**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.

Question	Answer	Marks
<b>Section A</b>		
1(a)	Variable <code>answer</code> (example only) Data type <code>integer/int</code> Validation            two distinct different points <b>OR</b> one point and an expansion <b>One</b> mark per mark point, max <b>two</b> <ul style="list-style-type: none"> <li>– restricted input to positive numbers</li> <li>– checked if input was numeric</li> <li>– ensured that input was a whole number / integer</li> <li>– checked if a value was input</li> </ul> <b>Or</b> one point and an expansion <b>two</b> marks, for example <ul style="list-style-type: none"> <li>– restricted input to positive numbers (1) by using a range/limit check / REPEAT... UNTIL looping until value greater than zero (1)</li> <li>– checked if input was numeric (1) by using a type check (1)</li> <li>– ensured the value input was a whole number (1) any suitable method e.g. rounding to whole number</li> <li>– checked if a value was input (1) by using a presence check (1)</li> </ul>	<b>4</b>
1(b)	Variable <code>noAttempts</code> (example only) Data type <code>integer/int</code> Use                     storing the number of attempts a pupil has made on a question	<b>3</b>

Question	Answer	Marks
1(c)	<p><b>One</b> mark per mark point, max <b>six</b></p> <p>MP1      initialisation (must include number of correct answers)  MP2      input name and table number with prompts  MP3      validate table number <b>and reinput</b>  MP4      ensure each question uses a different multiplier  MP5      display question  MP6      input <b>and validate</b> answer  MP7      check if answer correct  MP8      update score appropriately  MP9      repeat for 5 questions  MP10     display score <b>and name</b> at end ...  MP11     ...with an attempt to display an appropriate message dependent on score</p> <p>Example</p> <pre>Score = 0 Multiplier[1] ← 2 Multiplier[2] ← 5 Multiplier[3] ← 7 Multiplier[4] ← 8 Multiplier[5] ← 11 OUTPUT "Multiplication Table Test" OUTPUT "Please enter your name" INPUT Name REPEAT     OUTPUT "Please enter the table (2 to 12) you would like to be tested"     INPUT Table UNTIL (Table &gt;=2) AND (Table &lt;=12) FOR Count = 1 TO 5     OUTPUT ("Question ", Count)     OUTPUT (Multiplier[Count], " X ", Table, " = ")     REPEAT         INPUT Answer         UNTIL Answer &gt; 0 AND Answer = INT(Answer)         IF Answer = Multiplier[Count] * Table             THEN                 Score ← Score + 1             ENDIF     NEXT OUTPUT (Name, " your score is ", Score) IF Score = 5     THEN         OUTPUT ("Well done full marks")     ELSE         OUTPUT ("Have another practice") ENDIF</pre>	6

Question	Answer	Marks
1(d)	<p>Explanation</p> <p><b>One</b> mark per mark point illustrated with a suitable programming statement, max <b>four</b></p> <p>MP1 extra prompt and input to choose the number of questions and/or mixed set</p> <p>MP2 method of choosing number of questions and/or mixed set</p> <p>MP3 extra prompt and input for number of questions</p> <p>MP4 explanation of validation for either input</p> <p>MP5 how the end value for the question loop was changed if required</p> <p>MP6 how the program was changed to allow for mixed tables</p> <p>MP7 ... how the program ensured that the test used more than one multiplication table</p>	<b>4</b>
1(e)	<p>Explanation</p> <p><b>One</b> mark per mark point, max <b>three</b></p> <p>MP1 how to provide 3 answers, including the correct one</p> <p>MP2 how to ensure that all 3 answers are different</p> <p>MP3 how 3 answers are displayed e.g. output all 3 answers as a numbered list</p> <p>MP4 how to select the correct answer e.g. input the number of the correct answer</p> <p>MP5 how to ensure that the correct answer is not always in the same position</p>	<b>3</b>

Question	Answer	Marks
<b>Section B</b>		
2(a)	<ul style="list-style-type: none"> <li>– 12</li> <li>– 13</li> <li>– 07</li> <li>– 15</li> </ul>	<b>4</b>
2(b)	<ul style="list-style-type: none"> <li>– 27 (example many correct answers any whole number 1-99 inclusive)</li> <li>– 106 (example many correct answers)</li> <li>– 99/1</li> </ul>	<b>3</b>
2(c)	<p><b>One</b> mark per mark point, max <b>four</b></p> <ul style="list-style-type: none"> <li>– <b>new</b> variable for minimum...</li> <li>– ... set to first value/high value</li> <li>– ... at start of program / before line 4</li> <li>– test input / D for less than minimum</li> <li>– ... replace value minimum if input less than</li> <li>– ... after line 7 and before line 14</li> <li>– new output for minimum (with appropriate message)</li> <li>– ... at end of program // after line 14</li> </ul>	<b>4</b>

Question	Answer	Marks
3	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p><b>Pseudocode statement</b></p> <div style="border: 1px solid black; padding: 5px; width: 150px; margin-bottom: 5px;">IF X &gt; 12</div> <div style="border: 1px solid black; padding: 5px; width: 150px; margin-bottom: 5px;">INPUT X</div> <div style="border: 1px solid black; padding: 5px; width: 150px; margin-bottom: 5px;">X ← Y + Z</div> <div style="border: 1px solid black; padding: 5px; width: 150px;">OUTPUT X</div> </div> <div style="text-align: center;"> <p><b>Flowchart symbol</b></p> <div style="border: 2px solid black; width: 100px; height: 30px; margin-bottom: 20px;"></div> <div style="border: 2px solid black; width: 100px; height: 40px; margin-bottom: 20px; transform: rotate(45deg);"></div> <div style="border: 2px solid black; width: 100px; height: 30px; margin-bottom: 20px; transform: rotate(30deg);"></div> </div> </div> <p><b>One</b> mark for a single line to decision box from IF X &gt; 12  <b>One</b> mark for a single line to process box from X ← Y + Z</p> <p><b>Two</b> marks for two single lines from INPUT X and OUTPUT X to output box  <b>OR</b>  <b>One</b> mark for a single line from INPUT X or OUTPUT X to output box // two single lines from INPUT X and OUTPUT X to process box</p>	<b>4</b>

Question	Answer						Marks	
4(a)	<b>One mark for each correct column</b>						<b>7</b>	
	<b>Bag Weight</b>	<b>Accept</b>	<b>Over</b>	<b>Under</b>	<b>Error</b>	<b>Total</b>		<b>OUTPUT</b>
		0	0	0				(Enter weight of first cookie bag)
	1.05	1			0			(Weight of next bag?)
	0.99	2			0			(Weight of next bag?)
	1.2		1		1			(Weight of next bag?)
	0.85			1	2			(Weight of next bag?)
	1.1	3			0			(Weight of next bag?)
	0.9	4			0			(Weight of next bag?)
	1.5		2		1			(Weight of next bag?)
	0.95	5			0			(Weight of next bag?)
	1.05	6			0			(Weight of next bag?)
	1.00	7			0			(Weight of next bag?)
	1.07	8			0			(Weight of next bag?)
	0.89			2	2			(Weight of next bag?)
	-10					4		
								Number of bags weighed 4
							Number overweight 2	
							Number underweight 2	
4(b)	<b>Error Total</b> ← Accept – Over – Under // line 26 <b>Correction Total</b> ← Accept + Over + Under						<b>2</b>	

Question	Answer	Marks																														
5(a)	<p><b>One</b> mark for data type and reason            SIZE text, expressed as a single word // Boolean, only two choices            PRICE currency, needs to be expressed as Rupees/money            NUMBERSOLD number, integer values/could be used in calculations</p>	<b>3</b>																														
5(b)	<table border="1" data-bbox="424 416 1283 801"> <tr> <td data-bbox="424 416 619 479">Field:</td> <td data-bbox="619 416 826 479">SIZE</td> <td data-bbox="826 416 1018 479">FILLING</td> <td data-bbox="1018 416 1182 479">PRICE</td> <td data-bbox="1182 416 1283 479">NUMBERSOLD</td> </tr> <tr> <td data-bbox="424 479 619 542">Table:</td> <td data-bbox="619 479 826 542">CHOCBAR</td> <td data-bbox="826 479 1018 542">CHOCBAR</td> <td data-bbox="1018 479 1182 542">CHOCBAR</td> <td data-bbox="1182 479 1283 542">CHOCBAR</td> </tr> <tr> <td data-bbox="424 542 619 604">Sort:</td> <td data-bbox="619 542 826 604"></td> <td data-bbox="826 542 1018 604"></td> <td data-bbox="1018 542 1182 604"></td> <td data-bbox="1182 542 1283 604"></td> </tr> <tr> <td data-bbox="424 604 619 674">Show:</td> <td data-bbox="619 604 826 674"><input type="checkbox"/></td> <td data-bbox="826 604 1018 674"><input checked="" type="checkbox"/></td> <td data-bbox="1018 604 1182 674"><input checked="" type="checkbox"/></td> <td data-bbox="1182 604 1283 674"><input checked="" type="checkbox"/></td> </tr> <tr> <td data-bbox="424 674 619 739">Criteria:</td> <td data-bbox="619 674 826 739">= "small"</td> <td data-bbox="826 674 1018 739"></td> <td data-bbox="1018 674 1182 739"></td> <td data-bbox="1182 674 1283 739">&lt;10</td> </tr> <tr> <td data-bbox="424 739 619 801">or:</td> <td data-bbox="619 739 826 801"></td> <td data-bbox="826 739 1018 801"></td> <td data-bbox="1018 739 1182 801"></td> <td data-bbox="1182 739 1283 801"></td> </tr> </table> <p data-bbox="277 842 906 943"> <b>One</b> mark for correct rows Field, Table and Sort  <b>One</b> mark for correct Show row  <b>One</b> mark for correct Criteria row         </p>	Field:	SIZE	FILLING	PRICE	NUMBERSOLD	Table:	CHOCBAR	CHOCBAR	CHOCBAR	CHOCBAR	Sort:					Show:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Criteria:	= "small"			<10	or:					<b>3</b>
Field:	SIZE	FILLING	PRICE	NUMBERSOLD																												
Table:	CHOCBAR	CHOCBAR	CHOCBAR	CHOCBAR																												
Sort:																																
Show:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																												
Criteria:	= "small"			<10																												
or:																																