



# Cambridge IGCSE™

---

**COMPUTER SCIENCE**

**0478/22**

Paper 2

**February/March 2022**

**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 February/March 2022 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

---

This document consists of **11** 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)	<p>Many correct answers, the names used must be meaningful. The names given are examples only.</p> <p><b>One mark per mark point, max three</b></p> <p>Constant name      <code>MaxNumberPlayers</code>  Value                  4  Why constant used   This number will not change whilst the program is running</p> <p><b>One mark per mark point, max three</b></p> <p>Variable name      <code>NumberOfPlayers</code>  Data type            <code>Integer/int</code>  Why variable used   A value between 2 and 4 inclusive is input and stored at the start of the round</p>	<b>6</b>
1(b)	<p><b>One mark per mark point, max four</b></p> <p>MP1    using a <code>FOR ... NEXT // REPEAT ... UNTIL // DO ... WHILE</code> loop  MP2    starting at 1/0 and finishing at 18/17 <b>or</b> 9/8 // 18/9 iterations  MP3    setting the elements in the player scores arrays to zero  MP4    ... for all four arrays // number of arrays for players in this round  MP5    setting variables / array for total scores to 0  MP6    use of assignment / append statement(s)</p> <p>Any programming statements included must be explained.</p>	<b>4</b>

Question	Answer	Marks
1(c)	<p><b>One</b> mark per mark point, max <b>six</b></p> <p>MP1 loop through the number of holes played  MP2 for each hole work / loop through the number of players actually playing  MP3 for each player display their name  MP4 ... prompt to enter the number of strokes played for the hole  MP5 ... input the number of strokes twice  MP6 ... validate both inputs are the same  MP7 ... store the number of strokes in the appropriate array  MP8 ... update the total score for that player  MP9 ... prompt and input to ask if the player wants to see the number of strokes played so far  MP10 ... check if required then output number of strokes</p> <p><b>Example Answer</b></p> <pre> FOR Hole ← 1 TO NumberOfHoles   FOR Player ← 1 TO NumberOfPlayers     REPEAT       OUTPUT playerName[Player], "Please enter the number of strokes played for hole ",         Hole       INPUT NumberStrokes       OUTPUT playerName[Player], " please re-enter the number"       INPUT NumberStrokesAgain     UNTIL NumberStrokes = NumberStrokesAgain     IF Player = 1       THEN         Player1[Hole] ← NumberStrokes         Player1Total ← Player1Total + NumberStrokes         OUTPUT "Do you want to see number total of strokes played so far Y/N? "         INPUT SeeTotal         IF SeeTotal = "Y"           THEN             OUTPUT "Total number of strokes so far ", Player1Total           ENDIF         ENDIF       ENDIF     IF Player = 2 </pre>	6

Question	Answer	Marks
1(c)	<pre> THEN   Player2[Hole] ← NumberStrokes   Player2Total ← Player2Total + NumberStrokes   OUTPUT "Do you want to see number total of strokes played so far Y/N? "   INPUT SeeTotal   IF SeeTotal = "Y"     THEN       OUTPUT "Total number of strokes so far ", Player2Total     ENDIF   ENDIF IF Player = 3   THEN     Player3[Hole] ← NumberStrokes     Player3Total ← Player3Total + NumberStrokes     OUTPUT "Do you want to see number total of strokes played so far Y/N? "     INPUT SeeTotal     IF SeeTotal = "Y"       THEN         OUTPUT "Total number of strokes so far ", Player3Total       ENDIF     ENDIF IF Player = 4   THEN     Player4[Hole] ← NumberStrokes     Player4Total ← Player4Total + NumberStrokes     OUTPUT "Do you want to see number total of strokes played so far Y/N? "     INPUT SeeTotal     IF SeeTotal = "Y"       THEN         OUTPUT "Total number of strokes so far ", Player4Total       ENDIF     ENDIF   NEXT Player NEXT Hole </pre>	

Question	Answer	Marks
1(d)	<p>Explanation  <b>One</b> mark per mark point, max <b>four</b></p> <p>MP1    Work/ Loop through all the total scores  MP2    compare each total score using selection / IF statements // use an appropriate function  MP3    select the score with the <b>lowest</b> value  MP4    ... also select the player name for that score  MP5    output the player name and <b>either</b> the difference between par and their score <b>or</b> their score  MP6    compare the score with the lowest value with the value of par using selection / IF statements  MP7    ... identify as “over par” if the value is greater than par or output “under par” if the value is less than par or output “par” if there is no difference</p> <p>Any programming statements included must be explained.</p>	<b>4</b>

Question	Answer	Marks
<b>Section B</b>		
2(a)	<b>One</b> mark per mark point, max <b>four</b> <ul style="list-style-type: none"> <li>• 100</li> <li>• AND Age &lt; 12</li> <li>• Count12to18 + 1</li> <li>• CountOver18</li> </ul>	<b>4</b>
2(b)	<b>One</b> mark suitable IF construct, <b>one</b> mark correct assignment statement, for example IF Age < 7 THEN CountUnder7 ← CountUnder7 + 1 ENDIF <b>One</b> mark suitable message, <b>one</b> mark correct use of countUnder7 variable, for example OUTPUT "There are ", CountUnder7, " students aged under 7."	<b>4</b>



Question	Answer	Marks
3	<p><b>One</b> mark for each correct <b>single</b> line from the validation check, max <b>four</b></p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p><b>Validation check</b></p> <div style="display: flex; flex-direction: column; gap: 10px;"> <div style="border: 1px solid black; padding: 5px; width: 150px;">length check</div> <div style="border: 1px solid black; padding: 5px; width: 150px;">check digit</div> <div style="border: 1px solid black; padding: 5px; width: 150px;">range check</div> <div style="border: 1px solid black; padding: 5px; width: 150px;">type check</div> </div> </div> <div style="text-align: center;"> <p><b>Description</b></p> <div style="display: flex; flex-direction: column; gap: 10px;"> <div style="border: 1px solid black; padding: 5px; width: 200px;">checks that the data input is between two values</div> <div style="border: 1px solid black; padding: 5px; width: 200px;">checks that the data input is an integer</div> <div style="border: 1px solid black; padding: 5px; width: 200px;">checks that the data input has three digits</div> <div style="border: 1px solid black; padding: 5px; width: 200px;">checks that the data has been input</div> <div style="border: 1px solid black; padding: 5px; width: 200px;">checks that the data input has the correct digits</div> </div> </div> </div>	<b>4</b>

Question	Answer	Marks																																																				
4(a)	<p><b>One</b> mark each for columns <b>Number</b> and <b>OUTPUT</b> <b>Two</b> marks for column <b>C</b> first four values (1) last three values (1) <b>Two</b> marks for column <b>D</b> first six values (1) last four values (1)</p> <table><tr><th>Number</th><th>C</th><th>D</th><th>OUTPUT</th></tr><tr><td>7</td><td>0</td><td>3</td><td></td></tr><tr><td></td><td></td><td>2</td><td></td></tr><tr><td></td><td></td><td>1</td><td></td></tr><tr><td>6</td><td>0</td><td>3</td><td></td></tr><tr><td></td><td>1</td><td>2</td><td></td></tr><tr><td></td><td>2</td><td>1</td><td>2</td></tr><tr><td>5</td><td>0</td><td>2</td><td></td></tr><tr><td></td><td></td><td>1</td><td></td></tr><tr><td>4</td><td>0</td><td>2</td><td></td></tr><tr><td></td><td>1</td><td>1</td><td>1</td></tr><tr><td>-1</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	Number	C	D	OUTPUT	7	0	3				2				1		6	0	3			1	2			2	1	2	5	0	2				1		4	0	2			1	1	1	-1								6
Number	C	D	OUTPUT																																																			
7	0	3																																																				
		2																																																				
		1																																																				
6	0	3																																																				
	1	2																																																				
	2	1	2																																																			
5	0	2																																																				
		1																																																				
4	0	2																																																				
	1	1	1																																																			
-1																																																						
4(b)	<p><b>One</b> mark per mark point, max <b>two</b></p> <ul style="list-style-type: none"><li>to <b>count</b> the factors / the numbers that go into (other than 1 or itself) of a number</li><li>to output the <b>number of factors</b></li></ul>	2																																																				

Question	Answer	Marks
4(c)(i)	<b>One</b> mark per mark point, max <b>two</b> <ul style="list-style-type: none"> <li>the value of <u>D</u> becomes zero</li> <li>division by zero error</li> <li>endless loop</li> </ul>	<b>2</b>
4(c)(ii)	<b>One</b> mark per mark point, max <b>two</b> <ul style="list-style-type: none"> <li>after the decision box to test if the number is -1</li> <li>insert another decision box to test if the number is less than 4 / less than or equal to 3</li> <li>return to INPUT Number if true</li> </ul>	<b>2</b>

Question	Answer	Marks																														
5	<p>Explanation</p> <p><b>One</b> mark per mark point, max <b>three</b></p> <ul style="list-style-type: none"><li>field, FlowerID, not required / should not be displayed</li><li>Type field not included and displayed</li><li>Fragrance field should not be displayed</li><li>Fragrance criteria should not be Y / should be N</li></ul> <table><tr><td>Field:</td><td>Type</td><td>Fragrance</td><td>Style</td><td>Colour</td></tr><tr><td>Table:</td><td>FLOWER</td><td>FLOWER</td><td>FLOWER</td><td>FLOWER</td></tr><tr><td>Sort:</td><td></td><td></td><td></td><td></td></tr><tr><td>Show:</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td></tr><tr><td>Criteria:</td><td></td><td>= N</td><td></td><td></td></tr><tr><td>or:</td><td></td><td></td><td></td><td></td></tr></table> <p>query-by-example grid</p> <p><b>One</b> mark per mark point, max <b>three</b></p> <ul style="list-style-type: none"><li>One mark for changing Flower ID to Type</li><li>One mark for changing Criteria in Fragrance to N</li><li>One mark for changing Show in Fragrance to <input type="checkbox"/></li></ul>	Field:	Type	Fragrance	Style	Colour	Table:	FLOWER	FLOWER	FLOWER	FLOWER	Sort:					Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Criteria:		= N			or:					6
Field:	Type	Fragrance	Style	Colour																												
Table:	FLOWER	FLOWER	FLOWER	FLOWER																												
Sort:																																
Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																												
Criteria:		= N																														
or:																																