



---

**COMPUTER SCIENCE**

**0478/22**

Paper 2

**March 2019**

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 2019 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 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, they must be meaningful. The names are examples only.</p> <p>Constant name      <code>Max_Extra_Top</code>  Value                    <code>3</code>  Use                        Storing the number of extra toppings a pizza can have</p> <p>Variable name        <code>NextOrderNo</code>  Data type                <code>integer</code>  Use                        storing the next order number available</p>	<b>6</b>
1(b)	<p>Any <b>two</b> from  Store an extra type of base  Display another option  Change the if statement/validation check to include the third option//extra crispy</p>	<b>2</b>

Question	Answer	Marks
1(c)(i)	<p>Any <b>five</b> from:</p> <ul style="list-style-type: none"> <li>Enter Number of toppings</li> <li>Check number of toppings chosen</li> <li>Display toppings available</li> <li>Provide method of selection</li> <li>Only accept correct selections</li> <li>Error message if topping not found</li> <li>Finish selection</li> </ul> <p>There are many possible correct answers, this is an example only.</p> <p>Sample answer</p> <pre> PRINT "Pepperoni      1" PRINT "Chicken       2" PRINT "Extra cheese   3" PRINT "Mushrooms     4" PRINT "Spinach       5" PRINT "Olives        6" REPEAT   PRINT "How many extra toppings do you want"   INPUT NoTopping UNTIL NoTopping &gt;= 0 and NoTopping &lt;=3 WHILE NoTopping &gt;0 DO   PRINT "Enter Topping ", NoTopping   INPUT ToppingType   CASE OF ToppingType     1: Pepperoni ← Pepperoni + 1     2: Chicken ← Chicken + 1     3: ExtraCheese ← ExtraCheese + 1     4: Mushrooms ← Mushrooms + 1     5: Spinach ← Spinach + 1     6: Olives ← Olives + 1   OTHERWISE: PRINT "Error" ENDCASE </pre>	<b>5</b>

Question	Answer	Marks
1(c)(i)	<pre> IF ToppingType &gt;=1 AND ToppingType &lt;=6   THEN     PizzaTop[NoTopping] ← ToppingType     NoTopping ← NoTopping - 1   ENDIF ENDWHILE </pre>	
1(c)(ii)	<p>Answers must relate to the algorithm provided for (c)(i) Any <b>three</b> from:</p> <ul style="list-style-type: none"> <li>Display the valid toppings e.g. choose from a menu</li> <li>Check input for each topping is valid</li> <li>... method e.g. using a CASE statement / range check</li> <li>Provide a suitable error message for invalid toppings</li> <li>Provide a method to re-input a topping e.g. use of REPEAT...UNTIL</li> </ul>	<b>3</b>
1(d)	<p>Explanation Any <b>four</b> from:</p> <ul style="list-style-type: none"> <li>How the total of each additional pizza toppings was checked</li> <li>How the largest value was selected</li> <li>How the smallest value was selected</li> <li>How the topping descriptions were recorded for largest/smallest</li> <li>Method used to calculate percentages ...</li> <li>... calculation relates to the total number of additional toppings ordered</li> <li>Display results including suitable messages</li> </ul>	<b>4</b>

Question	Answer	Marks
<b>Section B</b>		
2(a)	<p>Total ← Count <b>should be</b> Total ← 0</p> <p>Number &lt;&gt; 0 <b>should be</b> Number &gt; 0</p> <p>Total ← Total + Count <b>should be</b> Total ← Total + Number</p> <p>UNTIL Count &lt; 50 <b>should be</b> UNTIL Count &gt; 50, UNTIL Count &gt;= 51, UNTIL Count = 51</p> <p>1 mark for each error identified + suggested correction</p>	<b>4</b>
2(b)	<p>The test should be IF Number &gt; 0 AND Number &lt;20</p> <p>One mark for both ends of the range and one mark for the <b>AND</b>.</p>	<b>2</b>

Question	Answer					Marks
3	<b>Senior</b>	<b>Adult</b>	<b>Child</b>	<b>Type</b>	<b>OUTPUT</b>	5
	0	0	0			
				S		
	1			S		
	2			S		
	3			A		
		1		C		
			1	C		
			2	C		
			3	A		
		2		A		
		3		A		
		4		A		
		5		W		
				S		
	4			S		
	5			D		
				C		
			4	Z	Seniors 5	
					Adults 5	
				Children 4		

One mark for each correct column.

Question	Answer			Marks															
4		<table border="1"> <thead> <tr> <th data-bbox="743 217 1171 284">Statements</th> <th data-bbox="1171 217 1346 284">Selection</th> <th data-bbox="1346 217 1534 284">Repetition</th> </tr> </thead> <tbody> <tr> <td data-bbox="743 284 1171 421"> <pre>FOR X ← 1 TO 10   SUM ← SUM + 1 NEXT X</pre> </td> <td data-bbox="1171 284 1346 421"></td> <td data-bbox="1346 284 1534 421">✓</td> </tr> <tr> <td data-bbox="743 421 1171 572"> <pre>WHILE X &gt; 10 DO   SUM ← SUM + 1   X ← X - 1 ENDWHILE</pre> </td> <td data-bbox="1171 421 1346 572"></td> <td data-bbox="1346 421 1534 572">✓</td> </tr> <tr> <td data-bbox="743 572 1171 778"> <pre>IF X &gt; 10 THEN   SUM ← SUM + 1   X ← X - 1 ENDIF</pre> </td> <td data-bbox="1171 572 1346 778">✓</td> <td data-bbox="1346 572 1534 778"></td> </tr> <tr> <td data-bbox="743 778 1171 951"> <pre>REPEAT   SUM ← SUM + 1   X ← X - 1 UNTIL X &gt; 10</pre> </td> <td data-bbox="1171 778 1346 951"></td> <td data-bbox="1346 778 1534 951">✓</td> </tr> </tbody> </table>	Statements	Selection	Repetition	<pre>FOR X ← 1 TO 10   SUM ← SUM + 1 NEXT X</pre>		✓	<pre>WHILE X &gt; 10 DO   SUM ← SUM + 1   X ← X - 1 ENDWHILE</pre>		✓	<pre>IF X &gt; 10 THEN   SUM ← SUM + 1   X ← X - 1 ENDIF</pre>	✓		<pre>REPEAT   SUM ← SUM + 1   X ← X - 1 UNTIL X &gt; 10</pre>		✓		4
Statements	Selection	Repetition																	
<pre>FOR X ← 1 TO 10   SUM ← SUM + 1 NEXT X</pre>		✓																	
<pre>WHILE X &gt; 10 DO   SUM ← SUM + 1   X ← X - 1 ENDWHILE</pre>		✓																	
<pre>IF X &gt; 10 THEN   SUM ← SUM + 1   X ← X - 1 ENDIF</pre>	✓																		
<pre>REPEAT   SUM ← SUM + 1   X ← X - 1 UNTIL X &gt; 10</pre>		✓																	

Question	Answer		Marks
5(a)	Validation Range check		2



Question	Answer	Marks
5(b)	<p>For each of <b>three</b> different data types Data type - 1 mark, Example - 1 mark, Reason - 1 mark</p> <p>There are many possible correct answers, this is an example only.</p> <p>Normal data (1 mark) 65 (1 mark) to show that the program accepts this value (1 mark) Erroneous data (1 mark) seventy (1 mark) to show that the program rejects this value (1 mark) Extreme data (1 mark) 89 (1 mark) to show that the program accepts this value (1 mark)</p>	<b>9</b>

Question	Answer				Marks																											
6	<table border="1" data-bbox="533 635 1955 1046"> <tr> <td data-bbox="443 643 521 699">Field:</td> <td data-bbox="533 643 824 699">Tyre Code</td> <td data-bbox="835 643 1193 699">Stock Level</td> <td data-bbox="1205 643 1585 699">Width</td> <td data-bbox="1597 643 1955 699">Terrain</td> </tr> <tr> <td data-bbox="443 707 521 762">Table:</td> <td data-bbox="533 707 824 762">BIKETYRES</td> <td data-bbox="835 707 1193 762">BIKETYRES</td> <td data-bbox="1205 707 1585 762">BIKETYRES</td> <td data-bbox="1597 707 1955 762">BIKETYRES</td> </tr> <tr> <td data-bbox="443 770 521 826">Sort:</td> <td colspan="2" data-bbox="533 770 1193 826">Ascending</td> <td></td> <td></td> </tr> <tr> <td data-bbox="443 834 521 890">Show:</td> <td data-bbox="533 834 824 890"><input checked="" type="checkbox"/></td> <td data-bbox="835 834 1193 890"><input checked="" type="checkbox"/></td> <td data-bbox="1205 834 1585 890"><input type="checkbox"/></td> <td data-bbox="1597 834 1955 890"><input type="checkbox"/></td> </tr> <tr> <td data-bbox="443 898 521 954">Criteria:</td> <td></td> <td></td> <td data-bbox="1205 898 1585 954">= 24</td> <td data-bbox="1597 898 1955 954">= 'Asphalt'</td> </tr> <tr> <td data-bbox="443 962 521 1018">or:</td> <td></td> <td></td> <td></td> <td></td> </tr> </table> <p data-bbox="320 1082 768 1114">One mark for each correct column</p>	Field:	Tyre Code	Stock Level	Width	Terrain	Table:	BIKETYRES	BIKETYRES	BIKETYRES	BIKETYRES	Sort:	Ascending				Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Criteria:			= 24	= 'Asphalt'	or:					<b>4</b>
Field:	Tyre Code	Stock Level	Width	Terrain																												
Table:	BIKETYRES	BIKETYRES	BIKETYRES	BIKETYRES																												
Sort:	Ascending																															
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																												
Criteria:			= 24	= 'Asphalt'																												
or:																																