



Cambridge IGCSE™

COMPUTER SCIENCE

0478/23

Paper 2

October/November 2020

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 October/November 2020 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 **7** 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
Section A		
1(a)	Constant name <code>MaxSalad</code> Value <code>3</code> Use Storing the salad items a baguette can have Variable name <code>OrderNumber</code> Data type <code>integer</code> Use storing the next order number available Many correct answers, this is an example only.	6

Question	Answer	Marks
1(b)(i)	<p>Any five from:</p> <p>MP1 Display fillings available MP2 Prompt and entry of filling choice MP3 Only accept one correct filling MP4 Prompt and entry of number of salads MP5 Only accept 0 to 3 inclusive MP6 Display salads available MP7 Prompt and entry of correct number of salad choices MP8 Only accept correct salad choices MP9 At least one error message for incorrect data entry</p> <p>Example answer:</p> <pre> REPEAT PRINT "Beef 1" PRINT "Chicken 2" PRINT "Cheese 3" PRINT "Egg 4" PRINT "Tuna 5" PRINT "Turkey 6" OUTPUT "Please enter your choice of filling" INPUT Filling UNTIL Filling >=1 AND Filling <=6 PRINT "Lettuce 1" PRINT "Tomato 2" PRINT "Sweetcorn 3" PRINT "Cucumber 4" PRINT "Peppers 5" REPEAT PRINT "How many salads do you want" INPUT NumSalad UNTIL NumSalad >= 0 and NumSalad <=3 WHILE NumSalad >0 DO PRINT "Enter Salad ", NumSalad OUTPUT "Please enter your choice of salad" INPUT SaladType CASE SaladType OF 1: NumSalad ← NumSalad - 1 2: NumSalad ← NumSalad - 1 3: NumSalad ← NumSalad - 1 4: NumSalad ← NumSalad - 1 5: NumSalad ← NumSalad - 1 OTHERWISE: PRINT "Error" ENDCASE Salad[NumSalad] ← SaladType ENDWHILE </pre>	5
1(b)(ii)	<p>Any three from:</p> <p>MP1 Provide a method of inputting the filling selection MP2 Check input is a correct filling e.g. using a CASE statement MP3 If not output a suitable error message MP4 If not provide a suitable method to re-input e.g. use of REPEAT...UNTIL</p>	3

Question	Answer	Marks
1(c)	Any two from: MP1 Store three sizes of baguette e.g. add third baguette size to array MP2 Change the prompt to output three sizes of baguettes MP3 Change the selection statement to allow for a third size of baguette e.g. IF/REPEAT/WHILE...	2
1(d)	Explanation Any four from: MP1 check the total for each of the baguette fillings e.g. use of FOR loop MP2 use of two variables one for most popular filling and one for least popular filling MP3 method used to select the largest value as the most popular e.g. use of IF statement MP4 method used to select the smallest value as the least popular e.g. use of IF statement MP5 Use these values to calculate percentages MP6 ... with the total number of baguettes sold MP7 Display results including suitable messages e.g. use of PRINT statement ... Programming statements can be used but must be explained.	4

Question	Answer	Marks
Section B		
2(a)	Line 1/2/8/12 Line 3 and/or 14 Line 8/12 Line 6/10/15/19	4
2(b)	One mark for error and correction Line 02 TooCold ← 0 Line 08 TooCold ← TooCold + 1 Line 15 IF TooHot > 5 Line 17 OUTPUT "Alarm!!"	4
2(c)	Any four from: Add a new variable inRange set to zero at start of algorithm Add extra IF statement IF temperature >= -25 AND temperature <= -18 Update inRange by 1 if true	4

Question	Answer	Marks												
3	<p>One mark for each correct line</p> <table border="0" style="width: 100%;"> <thead> <tr> <th style="text-align: left;">Programming concept</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td style="border: 1px solid black; padding: 5px; width: 20%;">Validation</td> <td style="border: 1px solid black; padding: 5px; width: 80%;">A subroutine that does not have to return a value</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">Verification</td> <td style="border: 1px solid black; padding: 5px;">An automatic check to ensure that data input is reasonable and sensible</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">Procedure</td> <td style="border: 1px solid black; padding: 5px;">A subroutine that always returns a value</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">Function</td> <td style="border: 1px solid black; padding: 5px;">An overview of a program or subroutine</td> </tr> <tr> <td></td> <td style="border: 1px solid black; padding: 5px;">A check to ensure that data input matches the original</td> </tr> </tbody> </table>	Programming concept	Description	Validation	A subroutine that does not have to return a value	Verification	An automatic check to ensure that data input is reasonable and sensible	Procedure	A subroutine that always returns a value	Function	An overview of a program or subroutine		A check to ensure that data input matches the original	4
Programming concept	Description													
Validation	A subroutine that does not have to return a value													
Verification	An automatic check to ensure that data input is reasonable and sensible													
Procedure	A subroutine that always returns a value													
Function	An overview of a program or subroutine													
	A check to ensure that data input matches the original													

Question	Answer	Marks																												
4(a)	<p>One mark for each correct column</p> <table border="1" style="border-collapse: collapse; text-align: center; width: 100%;"> <thead> <tr> <th>Y</th> <th>Z</th> <th>A</th> <th>OUTPUT</th> </tr> </thead> <tbody> <tr> <td>11</td> <td>4</td> <td>3</td> <td>Invalid</td> </tr> <tr> <td>6</td> <td>2</td> <td>0</td> <td>Valid</td> </tr> <tr> <td>3</td> <td>9</td> <td>0</td> <td>Valid</td> </tr> <tr> <td>3</td> <td>2</td> <td>1</td> <td>Invalid</td> </tr> <tr> <td>2</td> <td>6</td> <td>0</td> <td>Valid</td> </tr> <tr> <td>0</td> <td>0</td> <td></td> <td></td> </tr> </tbody> </table>	Y	Z	A	OUTPUT	11	4	3	Invalid	6	2	0	Valid	3	9	0	Valid	3	2	1	Invalid	2	6	0	Valid	0	0			4
Y	Z	A	OUTPUT																											
11	4	3	Invalid																											
6	2	0	Valid																											
3	9	0	Valid																											
3	2	1	Invalid																											
2	6	0	Valid																											
0	0																													
4(b)	<p>Any two from: Checking if the remainder, when the larger number is divided by the smaller number, is zero To see if the larger number is a multiple of the smaller number To see if the smaller number is a factor of the larger number</p>	2																												

Question	Answer	Marks																														
5(a)	5 fields 9 records	2																														
5(b)	One mark for each correct column max two , Or two correct rows one mark Short-tailed Albatross Y Emperor Penguin Y Yellow-eyed Penguin Y	2																														
5(c)	<table border="1" data-bbox="443 544 1326 969"> <tr> <td data-bbox="320 544 443 645">Field:</td> <td data-bbox="443 544 663 645">Creature</td> <td data-bbox="663 544 884 645">Quantity</td> <td data-bbox="884 544 1104 645">Offspring</td> <td data-bbox="1104 544 1326 645">Ready for release</td> </tr> <tr> <td data-bbox="320 645 443 710">Table:</td> <td data-bbox="443 645 663 710">MARINE</td> <td data-bbox="663 645 884 710">MARINE</td> <td data-bbox="884 645 1104 710">MARINE</td> <td data-bbox="1104 645 1326 710">MARINE</td> </tr> <tr> <td data-bbox="320 710 443 775">Sort:</td> <td colspan="2" data-bbox="443 710 884 775">Ascending</td> <td data-bbox="884 710 1104 775"></td> <td data-bbox="1104 710 1326 775"></td> </tr> <tr> <td data-bbox="320 775 443 840">Show:</td> <td data-bbox="443 775 663 840"><input checked="" type="checkbox"/></td> <td data-bbox="663 775 884 840"><input type="checkbox"/></td> <td data-bbox="884 775 1104 840"><input type="checkbox"/></td> <td data-bbox="1104 775 1326 840"><input type="checkbox"/></td> </tr> <tr> <td data-bbox="320 840 443 904">Criteria:</td> <td colspan="2" data-bbox="443 840 884 904"></td> <td data-bbox="884 840 1104 904">=N</td> <td data-bbox="1104 840 1326 904">=Y</td> </tr> <tr> <td data-bbox="320 904 443 969">or:</td> <td colspan="2" data-bbox="443 904 884 969"></td> <td data-bbox="884 904 1104 969"></td> <td data-bbox="1104 904 1326 969"></td> </tr> </table> <p data-bbox="304 1003 914 1137">One mark for each correct field and table rows One mark for sort row One mark for show row One mark for correct criteria</p>	Field:	Creature	Quantity	Offspring	Ready for release	Table:	MARINE	MARINE	MARINE	MARINE	Sort:	Ascending				Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Criteria:			=N	=Y	or:					4
Field:	Creature	Quantity	Offspring	Ready for release																												
Table:	MARINE	MARINE	MARINE	MARINE																												
Sort:	Ascending																															
Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																												
Criteria:			=N	=Y																												
or:																																