Cambridge Assessment International Education

Cambridge IGCSE™ (9–1)

COMPUTER SCIENCE **0984/22**

Paper 2 **October/November 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 October/November 2021 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

[Turn over

PUBLISHED

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)(i)	**One** mark per point • Variable PassengerID// StartStage • Use Storing the unique ID number of the passenger// Storing/inputting the start stage of the journey	**2**
1(a)(ii)	**One** mark per point MP1 Name of array MP2 Data type of array MP3 Sample data for array MP4 Use of array MP5 At least two complete arrays with all of the above **Array name** **Data type** **Sample data** **Use** `JourneyStage1` string `C1` to store the code for the home to start station `PriceStage1` real 1.50 to store the price of first stage of the journey	**5**
1(b)	**One** mark per bullet point MP1 Use of validation check, e.g. range check, type check, presence check, length check, format check MP2 Use of conditional statement to check if the validation fails ... MP3 … a re-entry is requested MP4 Use of loop to repeat the process until an acceptable answer is input MP5 More than one appropriate validation check used / described.	**3**
1(c)	Any **six** from: MP1 Conditional statement to check departure time against 10:00 MP2 … calculate 40% discount // calculate 60% of the original price MP3 … calculate discounted total price MP4 Output the discounted total price MP5 Output the **booking details** with suitable messages MP6 Input with prompt for passenger confirmation … MP7 … attempt at action following the confirmation input MP8 Repeating **booking data entry** if incorrect MP9 Re-checking journey details for correctness	**6**

Question	Answer	Marks

1(c)	**Example answer**
	`// Tasks 1 and 2 completed`
	`IF CollectedTime[Index] > 10:00`
	`  THEN`
	`    JourneyCost[Index] ← JouneyCost[Index] * 0.6`
	`ENDIF`
	`PRINT "Your journey cost is: ", JourneyCost[Index]`
	`PRINT "Your journey details are: ", PassengerID[Index], JourneyTime[Index],`
	`       JourneyCodes[Index], JourneyID[Index]`
	`PRINT "Are these details correct? (Y or N)"`
	`INPUT Correct`
	`IF Correct = "N"`
	`  THEN`
	`    WHILE Correct = "N"`
	`      PRINT "Re-enter your journey details"`
	`      PRINT "Correct passenger ID "`
	`      INPUT PassengerID[Index]`
	`      PRINT "Correct journey time "`
	`      INPUT JourneyTime[Index]`
	`      PRINT "Correct journey codes "`
	`      INPUT JourneyCodes[Index]`
	`      PRINT "Your revised journey details are: ", PassengerID[Index],`
	`             JourneyTime[Index], JourneyCodes[Index]`
	`      PRINT "Are these details correct? (Y or N)"`
	`      INPUT Correct`
	`    ENDWHILE`
	`ENDIF`
	`//Program continues`	

1(d)	**Explanation of how** each of the following could be done Any **four** from: MP1 Declaring/using a counter to store the number of bookings for each passenger MP2 Updating the counter for the number of bookings made by each passenger MP3 Attempt to check the number of bookings … MP4 … for the correct condition e.g. if the number of bookings is more than 10 / equal to 10 MP5 Apply the extra discount to the total price of future journeys	**4**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td></td><td>Section B</td><td></td></tr>
<tr><td>2</td><td>One mark for two correct rows
Two marks for three correct rows
Three marks for four correct rows.

<table>
<tr><th>Statement</th><th>Validation (✓)</th><th>Verification (✓)</th><th>Neither (✓)</th></tr>
<tr><td>a check where data is re-entered to make sure no errors have been introduced during data entry</td><td></td><td>✓</td><td></td></tr>
<tr><td>an automatic check to make sure the data entered has the correct number of characters</td><td>✓</td><td></td><td></td></tr>
<tr><td>a check to make sure the data entered is sensible</td><td>✓</td><td></td><td></td></tr>
<tr><td>a check to make sure the data entered is correct</td><td></td><td></td><td>✓</td></tr>
</table></td><td>3</td></tr>
</table>

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>3</td><td>One mark per bullet point

Normal test data
• Test data e.g. 50 (allow any number between 1 and 100 inclusive)
• Reason Data that is within range and should be accepted

Extreme test data
• Test data 100 / 1
• Reason Data at the maximum / minimum end of the range and should be accepted

Erroneous test data
• Test data e.g. 300 (allow anything that isn’t between 1 and 100 inclusive, including other data types)
• Reason Data outside the range that should be rejected</td><td>6</td></tr>
</table>

Question	Answer	Marks
4(a)	**One** mark for error identified and suggested correction (**Max three**) Line 8 `OUTPUT Value2` – should be `INPUT Value2` Line 9 `IF Operator` – should be `CASE OF Operator` Line 15 `OUTPUT "The answer is ", Value1` – should be `Answer` The loop may be corrected using a number of alternative methods: **One** mark for error identified and suggested correction (**Max two**) **Method 1** Line 1 `Continue ← 1` should be `Continue ← 0` Line 22 `UNTIL Continue = 0` should be `ENDWHILE //` Line 2 `WHILE Continue = 0` should be `REPEAT` **and** Line 22 `UNTIL Continue = 0` should be `Until Continue = 1` **OR** **Method 2** Line 2 `WHILE Continue = 0` should be `REPEAT` Line 20 `Continue ← 1` should be `Continue ← 0 //` Line 1 `Continue ← 1` should be `Continue ← 0` **and** Line 22 `UNTIL Continue = 0` should be `Until Continue = 1` **OR** **Method 3** Line 2 `WHILE Continue = 0` should be `WHILE Continue = 1` Line 20 `Continue ← 1` should be `Continue ← 0` **and** Line 22 `UNTIL Continue = 0` should be `ENDWHILE`	**5**

Question	Answer	Marks
4(a)	**Corrected algorithm example 1** 1 Continue ← 0 2 WHILE Continue = 0 (DO) 3 OUTPUT "Enter 1 for +, 2 for -, 3 for * or 4 for /" 4 INPUT Operator 5 OUTPUT "Enter the first value" 6 INPUT Value1 7 OUTPUT "Enter the second value" 8 INPUT Value2 9 CASE OF Operator 10 1: Answer ← Value1 + Value2 11 2: Answer ← Value1 - Value2 12 3: Answer ← Value1 * Value2 13 4: Answer ← Value1 / Value2 14 ENDCASE 15 OUTPUT "The answer is ", Answer 16 OUTPUT "Do you wish to enter more values (Yes or No)?" 17 INPUT MoreValues 18 IF MoreValues = "No" 19 THEN 20 Continue ← 1 21 ENDIF 22 ENDWHILE	

Question	Answer	Marks
4(a)	**Corrected algorithm example 2** 1 Continue ← 1 2 REPEAT 3 OUTPUT "Enter 1 for +, 2 for -, 3 for * or 4 for /" 4 INPUT Operator 5 OUTPUT "Enter the first value" 6 INPUT Value1 7 OUTPUT "Enter the second value" 8 INPUT Value2 9 CASE OF Operator 10 1: Answer ← Value1 + Value2 11 2: Answer ← Value1 - Value2 12 3: Answer ← Value1 * Value2 13 4: Answer ← Value1 / Value2 14 ENDCASE 15 OUTPUT "The answer is ", Answer 16 OUTPUT "Do you wish to enter more values (Yes or No)?" 17 INPUT MoreValues 18 IF MoreValues = "No" 19 THEN 20 Continue ← 0 21 ENDIF 22 UNTIL Continue = 0	

Question	Answer	Marks

4(b)	**One** mark per bullet
	MP1 Appropriate loop (begin and end) / otherwise selection
	MP2 Testing both ends of condition
	MP3 Suitable message
	MP4 Input/re-input
	
	`WHILE Operator < 1 OR Operator > 4 (DO)`
	`  OUTPUT "Enter 1, 2, 3 or 4"`
	`  INPUT Operator`
	`ENDWHILE`
	
	Alternative answer
	`REPEAT`
	`  IF Operator < 1 OR Operator > 4`
	`  THEN`
	`  OUTPUT "Enter 1, 2, 3 or 4"`
	`  INPUT Operator`
	`  ENDIF`
	`UNTIL Operator >= 1 AND Operator <= 4`
	
	One mark
	
	After line 4 / between lines 2 and 5
		5

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5</td><td>One mark for each correct column

<table>
<tr><th>List</th><th>Value</th><th>List1</th><th>List2</th><th>OUTPUT</th></tr>
<tr><td></td><td></td><td>0</td><td>0</td><td></td></tr>
<tr><td>2</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>77</td><td></td><td>77</td><td></td></tr>
<tr><td>2</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>16</td><td></td><td>93</td><td></td></tr>
<tr><td>1</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>35</td><td>35</td><td></td><td></td></tr>
<tr><td>2</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>-7</td><td></td><td>86</td><td></td></tr>
<tr><td>5</td><td></td><td></td><td></td><td></td></tr>
</table></td><td>5</td></tr>
</table>

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5</td><td>
<table>
<tr><th>List</th><th>Value</th><th>List1</th><th>List2</th><th>OUTPUT</th></tr>
<tr><td></td><td>18</td><td></td><td></td><td>Input Error</td></tr>
<tr><td>1</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>11</td><td>46</td><td></td><td></td></tr>
<tr><td>1</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>12</td><td>58</td><td></td><td></td></tr>
<tr><td>2</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>20</td><td></td><td>106</td><td></td></tr>
<tr><td>-1</td><td></td><td></td><td></td><td>List 1 = 58</td></tr>
<tr><td></td><td></td><td></td><td></td><td>List 2 = 106</td></tr>
<tr><td></td><td></td><td></td><td></td><td>List 2 is greatest</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td><td></td></tr>
</table>

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>6(a)(i)</td><td>InStock</td><td>1</td></tr>
<tr><td>6(a)(ii)</td><td>ProductID</td><td>1</td></tr>
<tr><td>6(b)</td><td>One mark for correct fieldnames
One mark for correct table names and show fields
One mark for correct sort
One mark for correct search criteria in all columns

<table>
<tr><td>Field:</td><td>ProductID</td><td>ProductName</td><td>Animal</td><td>InStock</td><td></td></tr>
<tr><td>Table:</td><td>STOCK</td><td>STOCK</td><td>STOCK</td><td>STOCK</td><td></td></tr>
<tr><td>Sort:</td><td>Ascending</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Show:</td><td>☑</td><td>☑</td><td>☐</td><td>☐</td><td>☐</td></tr>
<tr><td>Criteria:</td><td></td><td></td><td>=“cat”</td><td>=Yes</td><td></td></tr>
<tr><td>or:</td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td><td>4</td></tr>
</table>