Cambridge Assessment International Education
Cambridge International General Certificate of Secondary Education

COMPUTER SCIENCE **0478/23**

Paper 2 **May/June 2018**

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 May/June 2018 series for most Cambridge IGCSE™, Cambridge International A and AS Level and Cambridge Pre-U components, and some Cambridge O Level components.

IGCSE™ is a registered trademark.

This syllabus is approved for use in England, Wales and Northern Ireland as a Cambridge International Level 1/Level 2 Certificate.

© UCLES 2018

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.

© UCLES 2018

Question	Answer	Marks
	Section A	

1(a)(i)	**One** mark for any meaningful array name related to Task 1 × 2 e.g.
	
	`Tickets`
	`EntryTime`
	
	One mark for correct data type **AND** use related to Task 1 × 2 e.g.
	
	… integer to store the ticket numbers
	… real to store the entry times
		4

1(a)(ii)	**One** mark for any meaningful name for a constant **AND** value related to Task 3 × 2 e.g.
	
	`MaxTime`
	`8`
	`MaxFine`
	`100`
	
	One mark for correct use related to Task 3 × 2 e.g.
	
	…to store the maximum number of hours allowed in the car park
	…to store the current value of the fine for staying too long (in $)
		4

© UCLES 2018

Question	Answer	Marks

1(b)	Any **six** from:
	• Initialisation of ticket numbers and arrays
	• Loop for day's transactions
	• Check for entry
	• Output of free spaces before entry
	• Suitable input prompts to select on entry
	• Relevant Inputs (to get ticket and to input time)
	• Recording of entry time and ticket number in arrays
	• Generate next ticket number
	• Update and display number of car park spaces available after entry
	
	`Ticket_Number ← 1 //This number would not reset each day`
	`Ticket_Array[1:100]`
	`Entry_Time[1:100]`
	`Spaces ← 100`
	`Count ← 1 //This number would reset each day`
	
	`WHILE Count > 1 DO`
	`  OUTPUT "Available Spaces" Spaces`
	`  INPUT "Press enter to get ticket", Entry`
	`  OUTPUT "Next Ticket Number", Ticket_Number`
	`  INPUT "Current time", Current_Time`
	`  Ticket_Array[Count] ← Ticket_Number`
	`  Entry_Time[Count] ← Current_Time`
	`  Ticket_Number ← Ticket_Number + 1`
	`  Spaces ← Spaces – 1`
	`  Count ← Count + 1`
	`ENDWHILE`
		6

1(c)	Any **four** from: • Explanation of how the checking of the length of stay was done • Correct comparison to check length of stay against maximum • Explanation of how the extra charge is calculated • Correct calculation to work out the regular parking charge • Explanation of suitable output to show parking charge and extra charge if appropriate	**4**
1(d)	**One** mark for each correct test data item and related reason for **Task 1** (Answers **MUST** relate to pre-release task) e.g. Test data: –102 Reason: To check that negative values for ticket numbers are rejected Test data: 85 Reason: To check that normal ticket number data is accepted	**2**

© UCLES 2018

Question	Answer	Marks
	Section B	
2(a)	**One** mark for description **one** mark for example e.g. To test if the data entered is possible / reasonable … … A range check tests that data entered fits within specified values. Allow any correct validation check as an example	**2**
2(b)	**One** mark for description **one** mark for example e.g. To test if the data input is the same as the data that was intended to be input … … A double entry check expects each item of data to be entered twice and compares both entries to check they are the same. Allow any correct verification check as an example	**2**

Question	Answer	Marks

3	**One** mark for each correct answer
	
	`Counter = 0`
	`FOR Count = 1 TO 30`
	`Total = Total + Number`
	`NEXT Count`
		4

Question	Answer	Marks
4(a)	Any **two** from: • Expects a number to be input • Checks if the number is greater than 100 • Outputs the result of the test • Specific output example	**2**
4(b)(i)	**One** mark for correct answer e.g. Use a (condition controlled) loop	**1**

4(b)(ii)	**One** mark for each point
	• Initialisation of Number variable
	• Correct loop statements
	• Correct INPUT and OUTPUT
	e.g.
	`INPUT Number`
	`WHILE Number > 100 DO`
	`  OUTPUT "The number is too large"`
	`  INPUT Number`
	`ENDWHILE`
	`OUTPUT "The number is acceptable"`
	or
	`INPUT Number`
	`REPEAT`
	`  IF Number > 100`
	`  THEN`
	`  OUTPUT "The number is too large"`
	`  ENDIF`
	`  INPUT Number`
	`UNTIL Number <= 100`
	`OUTPUT "The number is acceptable"`
		3

© UCLES 2018

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(a)</td><td>
<table>
<tr><th>Max</th><th>Counter</th><th>Num</th><th>OUTPUT</th></tr>
<tr><td>–1000.00</td><td>0</td><td>6.30</td><td></td></tr>
<tr><td>6.30</td><td>1</td><td>18.62</td><td></td></tr>
<tr><td>18.62</td><td>2</td><td>50.01</td><td></td></tr>
<tr><td>50.01</td><td>3</td><td>3.13</td><td></td></tr>
<tr><td>50.01</td><td>4</td><td>2.05</td><td></td></tr>
<tr><td>50.01</td><td>5</td><td>50.10</td><td></td></tr>
<tr><td>50.10</td><td>6</td><td>40.35</td><td></td></tr>
<tr><td>50.10</td><td>7</td><td>30.69</td><td></td></tr>
<tr><td>50.10</td><td>8</td><td>0.85</td><td></td></tr>
<tr><td>50.10</td><td>9</td><td>17.30</td><td></td></tr>
<tr><td>50.10</td><td>10</td><td></td><td>50.10</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>
← 1 mark →← 1 mark →← 1 mark →
</td><td>3</td></tr>
<tr><td>5(b)</td><td>**One** mark for each correct change (max **two**)

Box 2 Change the initialization value of the current ‘Max’ variable to a very high number

Box 4 Change the inequality from > to <

Boxes 2, 4, 5, 8 Change the Max variable to something more suitable e.g. Min</td><td>2</td></tr>
</table>

Question	Answer	Marks
6(a)	**One** mark for correct answer 20	1
6(b)(i)	**One** mark for correct answer CatNo	1
6(b)(ii)	**One** mark for correct answer It is a unique identifier	1

© UCLES 2018

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>6(c)</td><td>One mark for every two correct data types

<table>
<tr><th>Field</th><th>Data Type</th></tr>
<tr><td>CatNo</td><td>Text</td></tr>
<tr><td>Title</td><td>Text</td></tr>
<tr><td>Genre 1</td><td>Text</td></tr>
<tr><td>Stream</td><td>Boolean / Text</td></tr>
</table></td><td>2</td></tr>
<tr><td>6(d)</td><td>One mark for each correct row

18m02 Golfwatch Yes No Yes
18m03 Chair 27 Yes Yes No

Accept if drawn in a table, but don’t allow any punctuation</td><td>2</td></tr>
<tr><td>6(e)</td><td>
<table>
<tr><td>Field:</td><td>CatNo</td><td>Title</td><td>Genre 1</td><td>Genre 2</td><td>Stream</td></tr>
<tr><td>Table:</td><td>2018MOV</td><td>2018MOV</td><td>2018MOV</td><td>2018MOV</td><td>2018MOV</td></tr>
<tr><td>Sort:</td><td></td><td>Ascending</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>=”Sci-Fi”</td><td></td><td>Yes</td></tr>
<tr><td>or:</td><td></td><td></td><td></td><td>=”Sci-Fi”</td><td>Yes</td></tr>
</table>
← 1 mark → ← 1 mark → ← 1 mark → ← 1 mark →

One mark per completely correct column / group of columns as shown.</td><td>4</td></tr>
</table>