Cambridge Assessment International Education
Cambridge International General Certificate of Secondary Education

COMPUTER SCIENCE **0478/23**

Paper 2 **October/November 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 October/November 2018 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

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

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)	1 mark any meaningful name in the format of an array related to Task 1 × 3 e.g.
	
	1 mark correct data type **AND** purpose related to Task 1 × 3 e.g.
	
	`CabinName`
	…string
	…to store the cabin names
	
	`CabinPricePeak`
	…real
	…to store the cabin peak price
	
	`CabinCapacity`
	…integer
	…to store the number of occupants
	
	`WeekNumber23`
	… integer/string/Boolean
	… to store whether a cabin has been booked for that week
		6

1(a)(ii)	1 mark for any meaningful name for a variable related to Task 2 e.g.
	
	`Cabin / CabinID / CabinName`
	`NumWeeks`
	`StartWeek`
	`BookingCode`
	`Capacity`
	`TotalCost`
	
	1 mark for correct data type **AND** purpose related to Task 2 e.g.
	
	… string to enter the ID/Name of the required cabin or chalet
	… integer to record the number of weeks for the holiday
	… string (allow integer) to enter the start week of the holiday.
		2

Question	Answer	Marks
1(b)	Any **four** from: • Loop to search all log cabins // loop to search all weeks • Method to search all weeks // method to search all log cabins • Check that value in array is blank/zero/false for each log cabin… • … every week • Output the name and capacity of each free log cabin … • … show the week number for that cabin Example	**4**

```
FOR Cabin ← 0 TO 9
   IF Booking23[Cabin] = ""
      THEN
         OUTPUT "Week 23 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin]
   ENDIF
   IF Booking24[Cabin] = ""
      THEN
         OUTPUT "Week 24 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin]
   ENDIF
   IF Booking25[Cabin] = ""
      THEN
         OUTPUT "Week 25 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin]
   ENDIF
   IF Booking26[Cabin] = ""
      THEN
         OUTPUT "Week 26 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin]
   ENDIF
   IF Booking27[Cabin] = ""
      THEN
         OUTPUT "Week 27 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin]
   ENDIF
   IF Booking28[Cabin] = ""
      THEN
         OUTPUT "Week 28 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin]
   ENDIF
```

Question	Answer	Marks
1(b)	IF Booking29[Cabin] = "" THEN OUTPUT "Week 29 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin] ENDIF IF Booking30[Cabin] = "" THEN OUTPUT "Week 30 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin] ENDIF IF Booking31[Cabin] = "" THEN OUTPUT "Week 31 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin] ENDIF IF Booking32[Cabin] = "" THEN OUTPUT "Week 32 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin] ENDIF IF Booking33[Cabin] = "" THEN OUTPUT "Week 33 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin] ENDIF IF Booking34[Cabin] = "" THEN OUTPUT "Week 34 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin] ENDIF IF Booking35[Cabin] = "" THEN OUTPUT "Week 35 ", " Cabin ", CabinName[Cabin] " Capacity ", Capacity[Cabin] ENDIF	

Question	Answer	Marks

1(b)	`IF Booking36[Cabin] = ""`
	`  THEN`
	`  OUTPUT "Week 36 ", " Cabin ", CabinName[Cabin] " Capacity ",`
	`  Capacity[Cabin]`
	`ENDIF`
	`IF Booking34[Cabin] = ""`
	`  THEN`
	`  OUTPUT "Week 37 ", " Cabin ", CabinName[Cabin] " Capacity ",`
	`  Capacity[Cabin]`
	`ENDIF`
	`IF Booking37[Cabin] = ""`
	`  THEN`
	`  OUTPUT "Week 38 ", " Cabin ", CabinName[Cabin] " Capacity ",`
	`  Capacity[Cabin]`
	`ENDIF`
	`IF Booking39[Cabin] = ""`
	`  THEN`
	`  OUTPUT "Week 39 ", " Cabin ", CabinName[Cabin] " Capacity ",`
	`  Capacity[Cabin]`
	`ENDIF`
	`NEXT Cabin`	

1(c)	Max 2 marks for suitable description of validation related to **Task 2** Name or description of check (1 mark) further expansion (1 mark) E.g. Presence check (1 mark) to check value has been input (1 mark) Search list of stored cabin names (1 mark) if found input accepted // if not found input rejected (1 mark) 1 mark for each test data item related to **Task 2** (Answers **MUST** relate to pre-release task and match check described) e.g. Valid test data: Hetty Invalid test data: Henry	**4**

Question	Answer	Marks
1(d)	Any **four** from: • Use the booking code // use a length of stay variable • Explanation of finding the length of stay e.g. counting the number of times the booking code is found • …explanation of how your program checks the length of stay > 2 weeks • Explanation of how your program calculated the discounted cost • Explanation of how your program output the original holiday cost and the discounted cost	**4**

Question	Answer	Marks
	Section B	
2	Integer – 1 mark for description 1 mark for example e.g. Any whole number … … for example a week number / 26 String – 1 mark for description 1 mark for example e.g. Any data item that contains letters and/or numbers and/or special characters … for example someone's name / def7773@.	**4**

Question	Answer	Marks

3	Condition controlled loop – 1 mark for each correct answer e.g.
	
	`WHILE Number > 0 DO … ENDWHILE // REPEAT … UNTIL Number > 0`
	
	Conditional statement - 1 mark for each correct answer e.g.
	
	`IF Number = 0 THEN (… ELSE) Number ← 1 ENDIF //`
	`CASE Number OF`
	`0: Number ← 1`
	`(… OTHERWISE) … (ENDCASE)`
	
	Totalling - 1 mark for each correct answer e.g.
	
	`Total ← Total + Number`
		3

Question	Answer	Marks
4(a)	1 mark for each error identified plus suggested correction Line 1 or `Total = 100.00`: **correction** Total = `0(.00)` Line 8 or `Count = Count + 1`: **correction** This line should be removed (not required in a FOR loop) // use of `REPEAT…UNTIL` or `WHILE…DO…ENDWHILE` Line 7 or `PRINT Total /30`: **correction** This line should be outside the loop (or it will print each iteration)	**3**
4(b)	1 mark for correct purpose: Find/output average height	**1**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(a)</td><td>
<table>
<tr><th>Fib</th><th>Prev2</th><th>Prev1</th><th>Number</th><th>OUTPUT</th></tr>
<tr><td>1</td><td>0</td><td>1</td><td>7</td><td></td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>6</td><td></td></tr>
<tr><td>2</td><td>1</td><td>2</td><td>5</td><td></td></tr>
<tr><td>3</td><td>2</td><td>3</td><td>4</td><td></td></tr>
<tr><td>5</td><td>3</td><td>5</td><td>3</td><td></td></tr>
<tr><td>8</td><td>5</td><td>8</td><td>2</td><td>8</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>< 1 Mark ></td><td colspan="2"><-------- 1 Mark --------></td><td><1 Mark></td><td><1 Mark></td></tr>
</table>
</td><td>4</td></tr>
</table>

Question	Answer	Marks
5(b)		**2**

Fib	Prev2	Prev1	Number	OUTPUT
1	0	1	2	1
<---------- 1 Mark ---------->			<-------- 1 Mark -------->	

Question	Answer	Marks
6(a)	1 mark for correct answer: No 1 mark for correct explanation: No field in this table contains unique identifier	**2**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>6(b)</td><td>1 mark for each correct answer

<table><tr><th>Field</th><th>Data type</th></tr><tr><td>Tree Type</td><td>Text</td></tr><tr><td>Size3</td><td>Number</td></tr><tr><td>Size2 In</td><td>Boolean/Text</td></tr></table></td><td>3</td></tr>
<tr><td>6(c)</td><td>1 mark for each correct row (max 3) and
1 mark for the correct order

Peach 9.25 No
Plum 8.95 Yes
Nectarine 8.50 Yes</td><td>4</td></tr>
<tr><td>6(d)</td><td>1 mark correct Fields included
1 mark correct Table and Show on all the four fields required
1 mark for correct Sort, must be ascending
1 mark for correct Criteria for the four fields

<table><tr><td>Field:</td><td>Tree Type</td><td>Size1 In</td><td>Size2 In</td><td>Size 3 In</td><td></td></tr><tr><td>Table:</td><td>TREETAB</td><td>TREETAB</td><td>TREETAB</td><td>TREETAB</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>=No</td><td>=No</td><td>=No</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>