



Cambridge IGCSE™

COMPUTER SCIENCE

0478/21

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.

This document consists of **11** 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 Value Use <code>NoCourts // NoSessions // NoBookingSlots</code> <code>8 // 10 // 80</code> Storing the number of courts / sessions available / times and courts available for booking	3
1(b)	Any four from: Any three from MP1 Identifier / name of array used MP2 Description of purpose of an identified array MP3 Length of an identified array used MP4 Type of data in an identified array MP5 Explanation of number of arrays used, must be capable of storing all data required MP6 Sample data for an identified array One mark MP7 Identifying more than one array E.g.4 arrays of ten elements for each squash court, for example for squash court 1 <code>Availability1</code> of type Boolean, <code>Guest1, Mobile1</code> and <code>Code1</code> all type string	4
1(c)	Any three from: MP1 Setting the first code number MP2 How to find subsequent code numbers MP3 How to ensure they were unique MP4 How to ensure they were exactly 4 digits	3

Question	Answer	Marks
1(d)	<p>Any six from:</p> <p>MP1 Input time MP2 ... with prompt MP3 Check each court MP4 If court not booked ... MP5 ... set as booked MP6 Input name, mobile number MP7 Calculate unique 4-digit code MP8 Store name, mobile number and 4-digit code in an array MP9 Display mobile number and 4-digit code with suitable message MP10 Change mobile number if necessary</p> <p>Sample</p> <pre> OUTPUT ("Which time do you want to book a squash court") INPUT BookTime Booked ← FALSE CourtNumber ← 0 IF BookTime >= 8 AND BookTime <= 17 THEN BookTime ← BookTime - 7 IF Available1[BookTime] THEN Available1[BookTime] ← FALSE; CourtNumber ← 1; Booked ← TRUE ENDIF IF NOT Booked AND Available2[BookTime] THEN Available2[BookTime] ← FALSE; CourtNumber ← 2; Booked ← TRUE ENDIF IF NOT Booked AND Available3[BookTime] THEN Available3[BookTime] ← FALSE; CourtNumber ← 3; Booked ← TRUE ENDIF IF NOT Booked AND Available4[BookTime] THEN Available4[BookTime] ← FALSE; CourtNumber ← 4; Booked ← TRUE ENDIF IF NOT Booked AND Available5[BookTime] THEN Available5[BookTime] ← FALSE; CourtNumber ← 5; Booked ← TRUE ENDIF IF NOT Booked AND Available6[BookTime] THEN Available6[BookTime] ← FALSE; CourtNumber ← 6; Booked ← TRUE ENDIF IF NOT Booked AND Available7[BookTime] THEN Available7[BookTime] ← FALSE; CourtNumber ← 7; Booked ← TRUE ENDIF </pre>	6

Question	Answer	Marks
1(d)	<pre> IF NOT Booked AND Available8[BookTime] THEN Available8[BookTime] ← FALSE; CourtNumber ← 8; Booked ← TRUE ENDIF IF NOT Booked OUTPUT("No court available at ", BookTime + 7) ENDIF IF Booked THEN OUTPUT ("Please enter your name") INPUT Name Code ← Code + 1 Correct ← "N" WHILE Correct = "N" DO OUTPUT ("Please enter your mobile number") INPUT Mobile OUTPUT ("Is Your mobile number correct Y/N? ", Mobile, " Your court number is ", CourtNumber, " Your entry code is ", Code) INPUT Correct ENDWHILE CASE CourtNumber OF 1 : Guest1[BookTime}← Name; Mobile1[BookTime] ← Mobile; Code1[BookTime] ← Code 2 : Guest2[BookTime}← Name; Mobile2[BookTime] ← Mobile; Code2[BookTime] ← Code 3 : Guest3[BookTime}← Name; Mobile3[BookTime] ← Mobile; Code3[BookTime] ← Code 4 : Guest4[BookTime}← Name; Mobile4[BookTime] ← Mobile; Code4[BookTime] ← Code 5 : Guest5[BookTime}← Name; Mobile5[BookTime] ← Mobile; Code5[BookTime] ← Code 6 : Guest6[BookTime}← Name; Mobile6[BookTime] ← Mobile; Code6[BookTime] ← Code 7 : Guest6[BookTime}← Name; Mobile7[BookTime] ← Mobile; Code7[BookTime] ← Code 8 : Guest8[BookTime}← Name; Mobile8[BookTime] ← Mobile; Code8[BookTime] ← Code ENDCASE NoCoutsBooked[BookTime] ← NoCoutsBooked[BookTime] - 1 ELSE OUTPUT ("Court not booked") ENDIF </pre>	

Question	Answer	Marks
1(e)	Explanation Any four from MP1 How the program totalled the number of bookings MP2 How the program displayed the total number of bookings MP3 How the program calculated the number of times each court was booked MP4 How the program selected the highest value for times a court was booked MP5 How the program displayed the court that was most booked Programming statements must be included and must be explained.	4

Question	Answer	Marks
Section B		
2(a)	<p>Line 1 should be Counter ← 0 Line 3 RandNum[Counter] ← Rand(1, 100) should be RandNum[Counter] ← Rand(1, 101) Line 4 Counter ← Counter + 2 should be Counter ← Counter + 1 Line 5 UNTIL Count <= 50 should be UNTIL Counter >= 50 // UNTIL Counter = 50</p> <pre> 1 Counter ← 0 2 REPEAT 3 RandNum[Counter] ← Rand(1, 100) 4 Counter ← Counter + 1 5 UNTIL Counter >= 50 </pre> <p>Or</p> <p>Line 3 RandNum[Counter] should be RandNum[Count] Line 3 Rand(1, 100) should be Rand(1, 101) Line 4 Counter ← Counter + 2 should be Count ← Count + 1 Line 5 UNTIL Count <= 50 should be UNTIL Count >= 50 // UNTIL Count = 50</p> <pre> 1 Count ← 0 2 REPEAT 3 RandNum[Count] ← Rand(1, 100) 4 Count ← Count + 1 5 UNTIL Count >= 50 </pre>	4
2(b)	<p>One mark for each correct line FOR Count ← 0 TO 49 // FOR Count ← 1 TO 50 RandNum[Count] ← Rand(1, 101) / Rand(0, 101) NEXT // NEXT Count</p>	3
2(c)	Precondition loop // WHILE ... DO ... ENDWHILE	1

Question	Answer	Marks
3(a)(i)	<p>one mark for sample, one mark for reason max four</p> <p>Normal Sample any positive value with three decimal places e.g. 5.682 Reason to test that normal data is accepted and processed correctly Erroneous Sample any value that would be rejected e.g. 5.6 or -1.345 or seven Reason to test that erroneous data is rejected</p>	4
3(a)(ii)	<p>Reason to test that 0.000 / -0.001 / highest possible non-positive is rejected and 0.001 / 0.000 / lowest positive number is accepted</p> <p>Sample 1 0.000 Sample 2 0.001</p>	3
3(b)	<p>One mark To check that values are entered as intended // to prevent incorrect values that meet the validation criteria being accepted</p> <p>Two marks Asking the user to enter the value twice and comparing the values // double entry (1) only accepting a value if both entries are identical (1)</p> <p>or</p> <p>Displaying the value as it is entered (1) so the user can put right errors have been made as the value was entered (1)</p>	3

Question	Answer				Marks
4	One mark for each correct column				5
Counter	Distinction	Mark	Award	OUTPUT	
0	0				
1	1	88			
2		74			
3		60			
4	2	90			
5	3	84			
6	4	87			
7	5	95			
8		72			
9	6	84			
10		66			
		-1	0.6	Highly Commended	

https://xtremepape.rs/

Question	Answer	Marks																														
5(a)	CATEGORY – Text characters / words only used PRICE – Currency, the price is in dollars / money CODE – Text no calculations required, could be numbers or characters STOCK – Number, comparisons and calculations may be required	4																														
5(b)	One mark for correct rows Field and Table One mark for correct Show row One mark for correct Criteria row <table border="1" data-bbox="443 517 1624 944"> <tr> <td data-bbox="344 517 443 582">Field:</td> <td data-bbox="443 517 739 582">CATEGORY</td> <td data-bbox="739 517 1034 582">MANUFACTURER</td> <td data-bbox="1034 517 1330 582">CODE</td> <td data-bbox="1330 517 1624 582">ECONOMYRATING</td> </tr> <tr> <td data-bbox="344 582 443 647">Table:</td> <td data-bbox="443 582 739 647">APPLIANCE</td> <td data-bbox="739 582 1034 647">APPLIANCE</td> <td data-bbox="1034 582 1330 647">APPLIANCE</td> <td data-bbox="1330 582 1624 647">APPLIANCE</td> </tr> <tr> <td data-bbox="344 647 443 713">Sort:</td> <td data-bbox="443 647 739 713"></td> <td data-bbox="739 647 1034 713"></td> <td data-bbox="1034 647 1330 713"></td> <td data-bbox="1330 647 1624 713"></td> </tr> <tr> <td data-bbox="344 713 443 778">Show:</td> <td data-bbox="443 713 739 778"><input checked="" type="checkbox"/></td> <td data-bbox="739 713 1034 778"><input checked="" type="checkbox"/></td> <td data-bbox="1034 713 1330 778"><input checked="" type="checkbox"/></td> <td data-bbox="1330 713 1624 778"><input type="checkbox"/></td> </tr> <tr> <td data-bbox="344 778 443 876">Criteria:</td> <td data-bbox="443 778 739 876"></td> <td data-bbox="739 778 1034 876"></td> <td data-bbox="1034 778 1330 876"></td> <td data-bbox="1330 778 1624 876">="A"</td> </tr> <tr> <td data-bbox="344 876 443 944">or:</td> <td data-bbox="443 876 739 944"></td> <td data-bbox="739 876 1034 944"></td> <td data-bbox="1034 876 1330 944"></td> <td data-bbox="1330 876 1624 944"></td> </tr> </table>	Field:	CATEGORY	MANUFACTURER	CODE	ECONOMYRATING	Table:	APPLIANCE	APPLIANCE	APPLIANCE	APPLIANCE	Sort:					Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Criteria:				="A"	or:					3
Field:	CATEGORY	MANUFACTURER	CODE	ECONOMYRATING																												
Table:	APPLIANCE	APPLIANCE	APPLIANCE	APPLIANCE																												
Sort:																																
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																												
Criteria:				="A"																												
or:																																