Cambridge Assessment International Education

Cambridge IGCSE™

COMPUTER SCIENCE **0478/21**

Paper 2 Algorithms, Programming and Logic **October/November 2023**

MARK SCHEME

Maximum Mark: 75

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 2023 series for most Cambridge IGCSE, Cambridge International A and AS Level components, and some Cambridge O Level components.

Generic Marking Principles

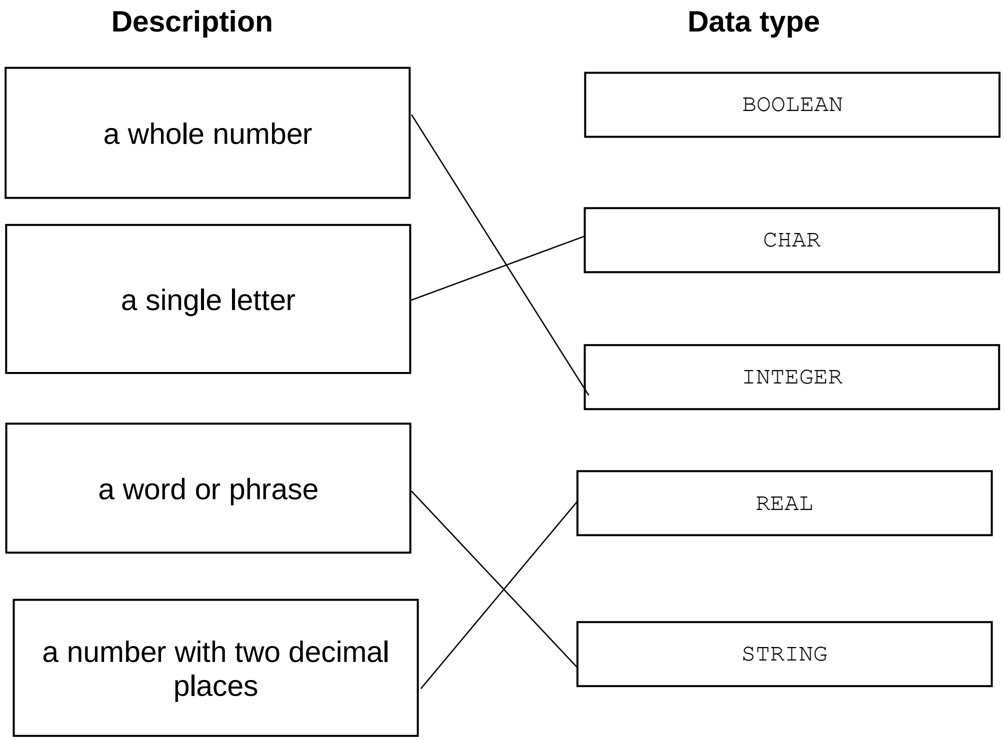
These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptors for a question. Each question paper and mark scheme will also comply with these marking principles.

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
1	C	**1**

Question	Answer	Marks
2	B	**1**

Question	Answer	Marks
3	One mark for each correct line from description to data type  	**4**

Question	Answer	Marks
4	One mark for each correct word • array • constant • variable	**3**

Question	Answer	Marks
5(a)	One mark for each point (max two) • simplifying the problem • removing unnecessary details from the problem // selecting elements required • filtering out irrelevant characteristics from those elements	**2**
5(b)	One mark for each point (max three) • inputs • processes • outputs • storage	**3**
5(c)	One mark for stage, one mark for matching description (max two) • design (1) details of solution set out (1) • coding (1) program is developed (1) • testing (1) program is tested for errors (1)	**2**

Question	Answer	Marks
6(a)	Displaying/sort 10 names in alphabetical order 9	**1**

Question	Answer	Marks
6(b)	One mark for each point (max four) • Initialisation • inputting 10 names • storing the names in an array • sorting the names in alphabetical order using a bubble sort • displaying the 10 names • iteration	**4**

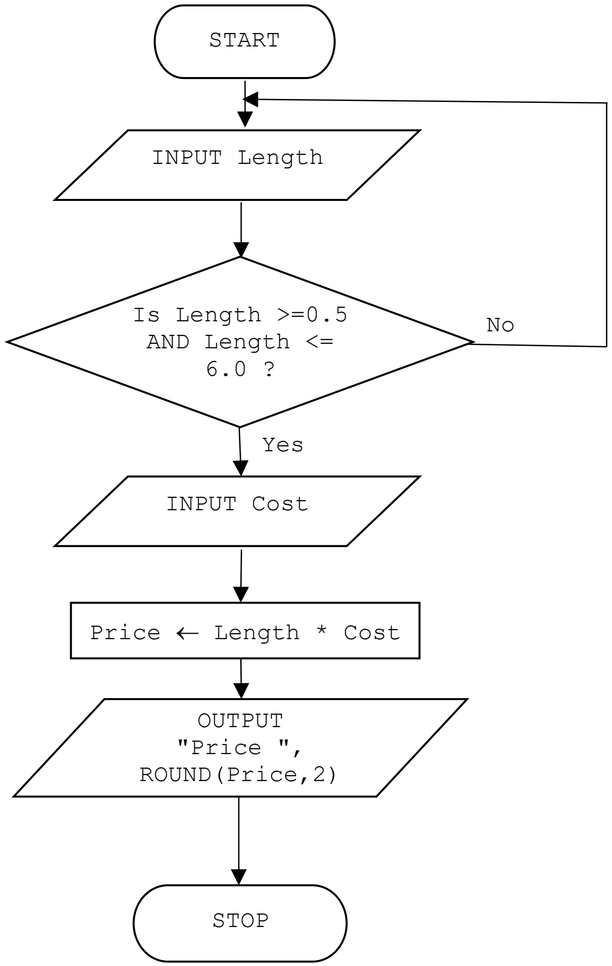
6(c)	**One** mark for a meaningful identifier for the array
	`A Names` // `ArrayNames`
	
	Two marks for 3 meaningful identifiers for variables
	One marks for 1 or 2 meaningful identifiers for variables
	`T Temp`
	`C Counter`
	`L Length`
		3

6(d)	One mark for each point (max two) • use of comments • use of procedures/functions • use of white space	**2**

Question	Answer	Marks
7(a)	One mark for each point • NOT A • AND B • OR NOT C • expression correct (NOT A AND B) OR NOT C	**4**

Question	Answer	Marks
7(b)	<table><tr><th>A</th><th>B</th><th>C</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table> 4 marks for 8 correct outputs 3 marks for 6/7 correct outputs 2 marks for 4/5 correct outputs 1 mark for 2/3 correct outputs	**4**

Question	Answer	Marks
8(a)	Range check	**1**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>8(b)</td><td>


One mark for each of the following points

correct use of flowchart symbols
working flow lines and complete
both inputs correct
working range check
working calculation
correct output rounded to two decimal places

</td><td>6</td></tr>
<tr><td>8(c)</td><td>One mark for set of test data, one mark for purpose (max four)

Example:
1 and 1 (1) normal data to ensure the algorithm accepts this test data (1)
–1 and 1 (1) abnormal data for length to ensure that it is rejected (1)</td><td>4</td></tr>
</table>

Question	Answer	Marks
8(d)	One mark for two correct headings Two marks for three correct headings Three marks for all headings correct and no other headings unless used in 8(b) <table><tr><td>**Length**</td><td>**Cost**</td><td>**Price**</td><td>**OUTPUT**</td></tr></table>	**3**
8(e)	**One** mark for each point (max two) • validate `Cost` … • … with a range/presence check • add another validation check for `Length`	**2**

Question	Answer	Marks
9(a)	**One** mark for each appropriate field name **and** correct data type `CatNo` text/alphanumeric `Description` text/alphanumeric `StockLevel` integer `Price` real `Painted` Boolean	**5**
9(b)(i)	`CatNo`	**1**
9(b)(ii)	Unique identifier	**1**
9(c)	`SELECT` lists the fields to be displayed `FROM` identifies the table `WHERE` identifies the search criteria	**3**

Question	Answer	Marks

10	• AO2 (maximum 9 marks)
	• AO3 (maximum 6 marks)
	
	Data Structures required names shown underlined must be used as given in the scenario
	2D Array or list `Evening[1:10, 1:20] / Evening[0:9, 0:19]`
	Variables `Counter, SeatCounter, NumSeats, Row, Column`
	
	Requirements (techniques)
	R1 Find number of seats available for each performance and output (searching, nested iteration, output)
	R2 Inputs and validates number of seats (input, iteration, and selection)
	R3 Checking if seats available (selection, assignment, output with appropriate messages)
	
	Example 15-mark answer in pseudocode
	
	`// meaningful identifier names and appropriate data structures to store the data required`
	`DECLARE Counter, SeatCounter, NumSeats, Row, Column : INTEGER`
	
	`CONSTANT HouseFull = 200`
	`CONSTANT MaxRow = 10`
	`CONSTANT MaxColumn = 20`
	
	`SeatCounter1 ← 0 // initialise seat counter for performance 1`
		15

Question	Answer	Marks

10	<pre>FOR Row ← 1 TO 10
	    FOR Column ← 1 TO 20
	        IF Evening[Row, Column]
	          THEN
	            SeatCounter ← SeatCounter + 1
	        ENDIF
	    NEXT Column
	NEXT Row
	
	// validate input
	OUTPUT "How many seats do you want to book? 1, 2, 3 or 4 "
	INPUT NumSeats
	
	WHILE 1 < NumSeats OR NumSeats > 4 OR NumSeats <> ROUND(NumSeats, 0)
	    OUTPUT "Please enter 1, 2, 3 or 4 for the number of seats "
	    INPUT NumSeats
	ENDWHILE
	
	IF SeatCounter + NumSeats > 200)// check for house full
	  THEN
	    OUTPUT "House full"
	  ELSE
	    IF SeatCounter + NumSeats > 200 // checks for not enough seats
	            THEN
	              OUTPUT "Only ", SeatCounter + NumSeats – 200, " seats left"
	            ELSE
	              FOR Counter ← 1 TO NumSeats // book required number of seats for performance
	                  Evening[MOD(SeatCounter + Counter, MaxColumn), DIV(SeatCounter +
	                          Counter), MaxColumn] ← TRUE
	                  OUTPUT "Row ", MOD(SeatCounter + Counter, MaxColumn), " seat ",
	                          DIV(SeatCounter + Counter, MaxColumn)," booked"
	              NEXT Counter
	           ENDIF</pre>	

<table>
<tr><td colspan="4">Marking Instructions in italics</td></tr>
<tr><td colspan="4">AO2: Apply knowledge and understanding of the principles and concepts of computer science to a given context, including the analysis and design of computational or programming problems</td></tr>
<tr><th>0</th><th>1-3</th><th>4-6</th><th>7-9</th></tr>
<tr><td rowspan="2">No creditable response.</td><td>At least one programming technique has been used.
Any use of selection, iteration, counting, totalling, input and output.</td><td>Some programming techniques used are appropriate to the problem.
More than one technique seen applied to the scenario, check the list of techniques needed.</td><td>The range of programming techniques used is appropriate to the problem.
All criteria stated for the scenario have been covered by the use of appropriate programming techniques, check list of techniques needed.</td></tr>
<tr><td>Some data has been stored but not appropriately.
Any use of variables or arrays or other language dependent data structures e.g. Python lists.</td><td>Some of the data structures chosen are appropriate and store some of the data required.
More than one data structure used to store data required by the scenario.</td><td>The data structures chosen are appropriate and store all the data required.
The data structures used store all the data required by the scenario.</td></tr>
</table>

Marking Instructions in italics			
AO3: Provide solutions to problems by:	**evaluating computer systems**	**making reasoned judgements**	**presenting conclusions**
0	**1-2**	**3-4**	**5-6**
No creditable response.	Program seen without relevant comments.	Program seen with some relevant comment(s).	The program has been fully commented
	Some identifier names used are appropriate *Some of the data structures used have meaningful names.*	The majority of identifiers used are appropriately named. *Most of the data structures used have meaningful names.*	Suitable identifiers with names meaningful to their purpose have been used throughout. *All of the data structures used have meaningful names.*
	The solution is illogical.	The solution contains parts that may be illogical.	The program is in a logical order.
	The solution is inaccurate in many places. *Solution contains few lines of code with errors that attempt to perform a task given in the scenario.*	The solution contains parts that are inaccurate. *Solution contains lines of code with some errors that logically perform tasks given in the scenario. Ignore minor syntax errors.*	The solution is accurate. *Solution logically performs all the tasks given in the scenario. Ignore minor syntax errors.*
	The solution attempts at least one of the requirements. *Solution contains lines of code that attempt at least one task given in the scenario.*	The solution attempts to meet most of the requirements. *Solution contains lines of code that perform most tasks given in the scenario.*	The solution meets all the requirements given in the question. *Solution performs all the tasks given in the scenario.*