

Cambridge IGCSE™ (9–1)

COMPUTER SCIENCE**0984/21**

Paper 2 Algorithms, Programming and Logic

May/June 2024**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 May/June 2024 series for most Cambridge IGCSE, Cambridge International A and AS Level and Cambridge Pre-U components, and some Cambridge O Level components.

This document consists of **18** printed pages.

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 descriptions 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.

Mark scheme abbreviations

/ separates alternative words / phrases within a marking point

// separates alternative answers within a marking point

underline actual word given must be used by candidate (grammatical variants accepted)

max indicates the maximum number of marks that can be awarded

() the word / phrase in brackets is not required, but sets the context

Note: No marks are awarded for using brand names of software packages or hardware.

Question	Answer	Marks
1	D	1

Question	Answer	Marks												
2(a)	<p>One mark for each correct line</p> <table> <thead> <tr> <th>Description</th> <th>Programming concept</th> </tr> </thead> <tbody> <tr> <td>a subroutine that may not return a value</td> <td>function</td> </tr> <tr> <td>a value that is declared and used within a specific procedure</td> <td>procedure</td> </tr> <tr> <td>a value that a procedure expects you to supply when it is called</td> <td>parameter</td> </tr> <tr> <td>a subroutine that will always return a value</td> <td>global variable</td> </tr> <tr> <td></td> <td>local variable</td> </tr> </tbody> </table>	Description	Programming concept	a subroutine that may not return a value	function	a value that is declared and used within a specific procedure	procedure	a value that a procedure expects you to supply when it is called	parameter	a subroutine that will always return a value	global variable		local variable	4
Description	Programming concept													
a subroutine that may not return a value	function													
a value that is declared and used within a specific procedure	procedure													
a value that a procedure expects you to supply when it is called	parameter													
a subroutine that will always return a value	global variable													
	local variable													
2(b)	<p>One mark per mark point</p> <ul style="list-style-type: none"> • Correct key word - <code>CALL</code> • Procedure name with correct parameters - <code>Average (25, 50)</code> <p><code>CALL Average (25, 50)</code></p>	2												

Question	Answer	Marks
2(c)	<p>One mark per mark point, max three</p> <ul style="list-style-type: none"> • Procedures / functions divide the program into smaller manageable segments • ... making it more readable / easier to understand / easier to debug • Procedures / functions with meaningful names help to provide documentation for the program / enable/help/provide abstraction • Procedures and functions may be re-used (in the program / in other programs / as part of a library) / run from a single line • Procedures and functions can reduce / eliminate (repeated) code 	3

Question	Answer	Marks
3	<p>One mark for a correct statement about each data type one mark for a correct example of data for each data type.</p> <p>Example answer</p> <p>Integer A whole number [1] Example 27 [1]</p> <p>Real A number that contains a fractional part [1] ... Example 18.75 [1]</p>	4

Question	Answer	Marks
4(a)	<p>One mark per mark point</p> <ul style="list-style-type: none"> • Line 01 / DECLARE Loop : STRING should be DECLARE Loop : INTEGER • Line 07 / IF Loop ← 1 TO Limit should be FOR Loop ← 1 TO Limit • Line 09 / INPUT Loop should be INPUT Value • Line 10 / Total ← Total * Value should be Total ← Total + Value <p>Correct algorithm:</p> <pre> 01 DECLARE Loop : INTEGER 02 DECLARE Limit : INTEGER 03 DECLARE Value : REAL 04 DECLARE Total : REAL 05 Total ← 0 06 Limit ← ROUND(RANDOM() * 19,0) + 1 07 FOR Loop ← 1 TO Limit 08 OUTPUT "Enter a number" 09 INPUT Value 10 Total ← Total + Value 11 NEXT Loop 12 OUTPUT "The total of the numbers entered is ", Total 13 OUTPUT "The average number entered is ", Total / Limit </pre>	4

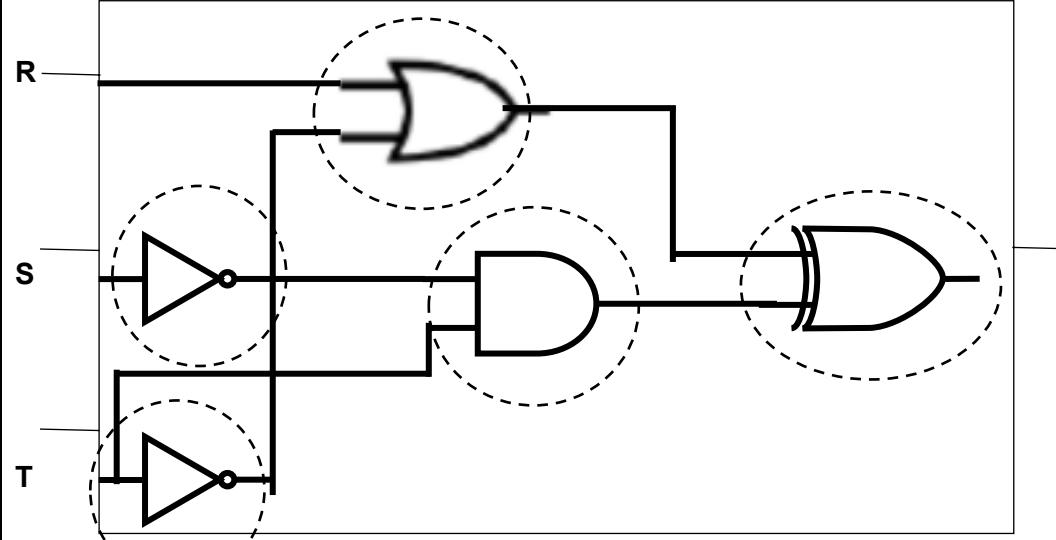
Question	Answer	Marks
4(b)	<p>One mark per mark point</p> <ul style="list-style-type: none"> Correct use of ROUND with 2 arguments separated by comma, for example ROUND(5, 2) (in statement 13) (Total / Limit,1) correct arguments <p>For example:</p> <pre>OUTPUT "The average of the numbers entered is ", ROUND(Total / Limit,1)</pre>	2
4(c)	<p>One mark per mark point, max four</p> <ul style="list-style-type: none"> After line 09 / after the input Insert a WHILE / pre-condition loop... ... to check if the value entered is between 1 and 500 inclusive If the value is not in range, output an error message ... and insert another input statement for re-input. <p>Or</p> <ul style="list-style-type: none"> Before line 08 / before the input message start a REPEAT / post-condition loop After line 09 / after the input ... close the REPEAT loop by checking if the value entered was between 1 and 500 inclusive If it wasn't, (the loop will repeat) for the number to be re-input / If it was, the program continues. 	4

Question	Answer	Marks
5	<p>One mark per mark point</p> <p>MP1 Correct input statement with appropriate variable MP2 Elements of selection statement present – CASE OF ENDCASE MP3 At least one correct branch in the case statement MP4 All branches from 1 to 4 correct MP5 Correct use of OTHERWISE with correct output.</p> <p>For example:</p> <pre>INPUT Number CASE OF Number 1 : OUTPUT Number 2 : OUTPUT Number 3 : OUTPUT Number 4 : OUTPUT Number OTHERWISE OUTPUT "ERROR" ENDCASE</pre> <p>Or</p> <pre>INPUT Number CASE OF Number 1 : OUTPUT 1 2 : OUTPUT 2 3 : OUTPUT 3 4 : OUTPUT 4 OTHERWISE OUTPUT "ERROR" ENDCASE</pre>	5

Question	Answer												Marks																																																																																																																																																																										
6(a)	<p>One mark per mark point</p> <p>MP1 Correct Limit column</p> <p>MP2 Correct Count column</p> <p>MP3 Array columns for input stage</p> <p>MP4 Array columns for sort stage</p> <p>MP5 Correct Flag column</p> <p>MP6 Correct Swap column</p> <p>MP7 Correct Result and OUTPUT columns</p> <table border="1" data-bbox="631 541 1275 1356"> <thead> <tr> <th colspan="12" data-bbox="631 541 1275 589">Numbers</th> </tr> <tr> <th data-bbox="339 605 451 652">Limit</th><th data-bbox="451 605 563 652">Count</th><th data-bbox="563 605 653 652">[1]</th><th data-bbox="653 605 743 652">[2]</th><th data-bbox="743 605 833 652">[3]</th><th data-bbox="833 605 923 652">[4]</th><th data-bbox="923 605 1012 652">[5]</th><th data-bbox="1012 605 1102 652">[6]</th><th data-bbox="1102 605 1192 652">[7]</th><th data-bbox="1192 605 1327 652">Flag</th><th data-bbox="1327 605 1462 652">Swap</th><th data-bbox="1462 605 1596 652">Result</th><th data-bbox="1596 605 1866 652">OUTPUT</th></tr> </thead> <tbody> <tr> <td data-bbox="339 652 451 716">7</td><td data-bbox="451 652 563 716">1</td><td data-bbox="563 652 653 716">47</td><td data-bbox="653 652 743 716"></td><td data-bbox="743 652 833 716"></td><td data-bbox="833 652 923 716"></td><td data-bbox="923 652 1012 716"></td><td data-bbox="1012 652 1102 716"></td><td data-bbox="1102 652 1192 716"></td><td data-bbox="1192 652 1327 716"></td><td data-bbox="1327 652 1462 716"></td><td data-bbox="1462 652 1596 716"></td><td data-bbox="1596 652 1866 716"></td></tr> <tr> <td data-bbox="339 716 451 779"></td><td data-bbox="451 716 563 779">2</td><td data-bbox="563 716 653 779"></td><td data-bbox="653 716 743 779">50</td><td data-bbox="743 716 833 779"></td><td data-bbox="833 716 923 779"></td><td data-bbox="923 716 1012 779"></td><td data-bbox="1012 716 1102 779"></td><td data-bbox="1102 716 1192 779"></td><td data-bbox="1192 716 1327 779"></td><td data-bbox="1327 716 1462 779"></td><td data-bbox="1462 716 1596 779"></td><td data-bbox="1596 716 1866 779"></td></tr> <tr> <td data-bbox="339 779 451 843"></td><td data-bbox="451 779 563 843">3</td><td data-bbox="563 779 653 843"></td><td data-bbox="653 779 743 843"></td><td data-bbox="743 779 833 843">52</td><td data-bbox="833 779 923 843"></td><td data-bbox="923 779 1012 843"></td><td data-bbox="1012 779 1102 843"></td><td data-bbox="1102 779 1192 843"></td><td data-bbox="1192 779 1327 843"></td><td data-bbox="1327 779 1462 843"></td><td data-bbox="1462 779 1596 843"></td><td data-bbox="1596 779 1866 843"></td></tr> <tr> <td data-bbox="339 843 451 906"></td><td data-bbox="451 843 563 906">4</td><td data-bbox="563 843 653 906"></td><td data-bbox="653 843 743 906"></td><td data-bbox="743 843 833 906"></td><td data-bbox="833 843 923 906">60</td><td data-bbox="923 843 1012 906"></td><td data-bbox="1012 843 1102 906"></td><td data-bbox="1102 843 1192 906"></td><td data-bbox="1192 843 1327 906"></td><td data-bbox="1327 843 1462 906"></td><td data-bbox="1462 843 1596 906"></td><td data-bbox="1596 843 1866 906"></td></tr> <tr> <td data-bbox="339 906 451 970"></td><td data-bbox="451 906 563 970">5</td><td data-bbox="563 906 653 970"></td><td data-bbox="653 906 743 970"></td><td data-bbox="743 906 833 970"></td><td data-bbox="833 906 923 970"></td><td data-bbox="923 906 1012 970">80</td><td data-bbox="1012 906 1102 970"></td><td data-bbox="1102 906 1192 970"></td><td data-bbox="1192 906 1327 970"></td><td data-bbox="1327 906 1462 970"></td><td data-bbox="1462 906 1596 970"></td><td data-bbox="1596 906 1866 970"></td></tr> <tr> <td data-bbox="339 970 451 1033"></td><td data-bbox="451 970 563 1033">6</td><td data-bbox="563 970 653 1033"></td><td data-bbox="653 970 743 1033"></td><td data-bbox="743 970 833 1033"></td><td data-bbox="833 970 923 1033"></td><td data-bbox="923 970 1012 1033">63</td><td data-bbox="1012 970 1102 1033"></td><td data-bbox="1102 970 1192 1033"></td><td data-bbox="1192 970 1327 1033"></td><td data-bbox="1327 970 1462 1033"></td><td data-bbox="1462 970 1596 1033"></td><td data-bbox="1596 970 1866 1033"></td></tr> <tr> <td data-bbox="339 1033 451 1097"></td><td data-bbox="451 1033 563 1097">7</td><td data-bbox="563 1033 653 1097"></td><td data-bbox="653 1033 743 1097"></td><td data-bbox="743 1033 833 1097"></td><td data-bbox="833 1033 923 1097"></td><td data-bbox="923 1033 1012 1097"></td><td data-bbox="1012 1033 1102 1097"></td><td data-bbox="1102 1033 1192 1097">70</td><td data-bbox="1192 1033 1327 1097"></td><td data-bbox="1327 1033 1462 1097"></td><td data-bbox="1462 1033 1596 1097"></td><td data-bbox="1596 1033 1866 1097"></td></tr> <tr> <td data-bbox="339 1097 451 1160"></td><td data-bbox="451 1097 563 1160">8</td><td data-bbox="563 1097 653 1160"></td><td data-bbox="653 1097 743 1160"></td><td data-bbox="743 1097 833 1160"></td><td data-bbox="833 1097 923 1160"></td><td data-bbox="923 1097 1012 1160"></td><td data-bbox="1012 1097 1102 1160"></td><td data-bbox="1102 1097 1192 1160"></td><td data-bbox="1192 1097 1327 1160">TRUE</td><td data-bbox="1327 1097 1462 1160"></td><td data-bbox="1462 1097 1596 1160"></td><td data-bbox="1596 1097 1866 1160"></td></tr> <tr> <td data-bbox="339 1160 451 1224"></td><td data-bbox="451 1160 563 1224">1</td><td data-bbox="563 1160 653 1224"></td><td data-bbox="653 1160 743 1224"></td><td data-bbox="743 1160 833 1224"></td><td data-bbox="833 1160 923 1224"></td><td data-bbox="923 1160 1012 1224"></td><td data-bbox="1012 1160 1102 1224"></td><td data-bbox="1102 1160 1192 1224"></td><td data-bbox="1192 1160 1327 1224">FALSE</td><td data-bbox="1327 1160 1462 1224"></td><td data-bbox="1462 1160 1596 1224"></td><td data-bbox="1596 1160 1866 1224"></td></tr> <tr> <td data-bbox="339 1224 451 1287"></td><td data-bbox="451 1224 563 1287">2</td><td data-bbox="563 1224 653 1287"></td><td data-bbox="653 1224 743 1287"></td><td data-bbox="743 1224 833 1287"></td><td data-bbox="833 1224 923 1287"></td><td data-bbox="923 1224 1012 1287"></td><td data-bbox="1012 1224 1102 1287"></td><td data-bbox="1102 1224 1192 1287"></td><td data-bbox="1192 1224 1327 1287"></td><td data-bbox="1327 1224 1462 1287"></td><td data-bbox="1462 1224 1596 1287"></td><td data-bbox="1596 1224 1866 1287"></td></tr> <tr> <td data-bbox="339 1287 451 1351"></td><td data-bbox="451 1287 563 1351">3</td><td data-bbox="563 1287 653 1351"></td><td data-bbox="653 1287 743 1351"></td><td data-bbox="743 1287 833 1351"></td><td data-bbox="833 1287 923 1351"></td><td data-bbox="923 1287 1012 1351"></td><td data-bbox="1012 1287 1102 1351"></td><td data-bbox="1102 1287 1192 1351"></td><td data-bbox="1192 1287 1327 1351"></td><td data-bbox="1327 1287 1462 1351"></td><td data-bbox="1462 1287 1596 1351"></td><td data-bbox="1596 1287 1866 1351"></td></tr> <tr> <td data-bbox="339 1351 451 1414"></td><td data-bbox="451 1351 563 1414">4</td><td data-bbox="563 1351 653 1414"></td><td data-bbox="653 1351 743 1414"></td><td data-bbox="743 1351 833 1414"></td><td data-bbox="833 1351 923 1414"></td><td data-bbox="923 1351 1012 1414"></td><td data-bbox="1012 1351 1102 1414"></td><td data-bbox="1102 1351 1192 1414"></td><td data-bbox="1192 1351 1327 1414"></td><td data-bbox="1327 1351 1462 1414"></td><td data-bbox="1462 1351 1596 1414"></td><td data-bbox="1596 1351 1866 1414"></td></tr> </tbody> </table>	Numbers												Limit	Count	[1]	[2]	[3]	[4]	[5]	[6]	[7]	Flag	Swap	Result	OUTPUT	7	1	47												2		50											3			52										4				60									5					80								6					63								7							70						8								TRUE					1								FALSE					2													3													4												7
Numbers																																																																																																																																																																																							
Limit	Count	[1]	[2]	[3]	[4]	[5]	[6]	[7]	Flag	Swap	Result	OUTPUT																																																																																																																																																																											
7	1	47																																																																																																																																																																																					
	2		50																																																																																																																																																																																				
	3			52																																																																																																																																																																																			
	4				60																																																																																																																																																																																		
	5					80																																																																																																																																																																																	
	6					63																																																																																																																																																																																	
	7							70																																																																																																																																																																															
	8								TRUE																																																																																																																																																																														
	1								FALSE																																																																																																																																																																														
	2																																																																																																																																																																																						
	3																																																																																																																																																																																						
	4																																																																																																																																																																																						

Question	Answer												Marks																																																																																																																																																																																							
6(a)	<table border="1" data-bbox="316 215 1882 1184"> <thead> <tr> <th colspan="2"></th> <th colspan="7">Numbers</th> <th colspan="3"></th> </tr> <tr> <th>Limit</th> <th>Count</th> <th>[1]</th> <th>[2]</th> <th>[3]</th> <th>[4]</th> <th>[5]</th> <th>[6]</th> <th>[7]</th> <th>Flag</th> <th>Swap</th> <th>Result</th> <th>OUTPUT</th> </tr> </thead> <tbody> <tr> <td></td> <td>5</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>80</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>63</td> <td>80</td> <td></td> <td></td> <td>TRUE</td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>6</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>80</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>70</td> <td>80</td> <td>TRUE</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>7</td> <td></td> </tr> <tr> <td></td> <td>1</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>FALSE</td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>2</td> <td></td> </tr> <tr> <td></td> <td>3</td> <td></td> </tr> <tr> <td></td> <td>4</td> <td></td> </tr> <tr> <td></td> <td>5</td> <td></td> </tr> <tr> <td></td> <td>6</td> <td></td> </tr> <tr> <td></td> <td>7</td> <td></td> </tr> <tr> <td></td> <td>4</td> <td>60</td> <td></td> </tr> </tbody> </table>				Numbers										Limit	Count	[1]	[2]	[3]	[4]	[5]	[6]	[7]	Flag	Swap	Result	OUTPUT		5									80								63	80			TRUE					6									80									70	80	TRUE						7													1								FALSE					2													3													4													5													6													7																						4	60	
		Numbers																																																																																																																																																																																																		
Limit	Count	[1]	[2]	[3]	[4]	[5]	[6]	[7]	Flag	Swap	Result	OUTPUT																																																																																																																																																																																								
	5									80																																																																																																																																																																																										
					63	80			TRUE																																																																																																																																																																																											
	6									80																																																																																																																																																																																										
						70	80	TRUE																																																																																																																																																																																												
	7																																																																																																																																																																																																			
	1								FALSE																																																																																																																																																																																											
	2																																																																																																																																																																																																			
	3																																																																																																																																																																																																			
	4																																																																																																																																																																																																			
	5																																																																																																																																																																																																			
	6																																																																																																																																																																																																			
	7																																																																																																																																																																																																			
										4	60																																																																																																																																																																																									
3																																																																																																																																																																																																				
<p>One mark per mark point</p> <ul style="list-style-type: none"> • A set of numbers is input / stored in an array • The numbers are sorted / Bubble sort into (ascending) order • The middle / median value is found / output 																																																																																																																																																																																																				

Question	Answer	Marks
7(a)	<p>One mark per mark point, max two</p> <ul style="list-style-type: none"> • Data stored in a file gives a permanent copy / prevents the data from being lost • ... so it can be recalled / used again in a program / in another program • can be stored elsewhere / transferred to another computer 	2
7(b)	<p>One mark per mark point, max four</p> <p>MP1 Declaration of a string variable MP2 Correct use of <code>OPENFILE</code> keywords for reading MP3 Correct use <code>READFILE</code> with string variable MP4 Correct use of <code>UCASE</code> and <code>LENGTH</code> functions and output of each MP5 Correct use of <code>CLOSEFILE</code></p> <p>Example:</p> <pre>DECLARE Words : STRING OPENFILE Quotation.txt FOR READ READFILE Quotation.txt, Words OUTPUT UCASE(Words), LENGTH(Words) CLOSEFILE Quotation.txt</pre>	4

Question	Answer	Marks
8(a)	<p>One mark for each correct gate, with the correct input(s) as shown.</p>  <p>z</p>	5

Question	Answer	Marks																																				
8(b)	<p>Four marks for eight correct outputs. Three marks for six or seven correct outputs. Two marks for four or five correct outputs. One mark for two or three correct outputs</p> <table border="1"><thead><tr><th>R</th><th>S</th><th>T</th><th>Z</th></tr></thead><tbody><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>1</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>0</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>1</td></tr></tbody></table>	R	S	T	Z	0	0	0	1	0	0	1	1	0	1	0	1	0	1	1	0	1	0	0	1	1	0	1	0	1	1	0	1	1	1	1	1	4
R	S	T	Z																																			
0	0	0	1																																			
0	0	1	1																																			
0	1	0	1																																			
0	1	1	0																																			
1	0	0	1																																			
1	0	1	0																																			
1	1	0	1																																			
1	1	1	1																																			

Question	Answer	Marks
9(a)	<p>One mark for an appropriate reason if no has been stated, for example: None of the fields are likely to contain unique data.</p>	1
9(b)	<p>One mark for each correct answer</p> <p>SUM NumberInStock SoftDrinks Container 'Can'</p> <p>Correct code:</p> <pre>SELECT SUM <u>NumberInStock</u> FROM SoftDrinks WHERE Container = 'Can';</pre>	5
10	<p>Data structures required:</p> <p>The names underlined must match those given in the scenario:</p> <p>Arrays or lists <u>Clubs[]</u>, <u>Statistics[]</u>, <u>Points[]</u>, <u>TieIndex[]</u></p> <p>Variables <u>Matches</u>, Won, Drawn, Lost, Counter, Maximum, TieCheck, OutLoop, Max, Count</p> <p>Requirements (techniques):</p> <p>R1 Input and store number of matches, cricket club names, number of matches won, drawn and lost. Validation of number of matches played and matches won, lost and drawn against number of matches played (with prompts, iteration, storing data in variables, validation, 1D and 2D arrays).</p> <p>R2 Calculate and store the number of points for each cricket club. Finding the index of the array with the highest score / highest score (calculation, finding the maximum, iteration).</p> <p>R3 Identifying cricket clubs with highest number of points and whether there is a tie for the top position. Output with appropriate messages (iteration, selection and output).</p>	15

Question	Answer	Marks
10	<p>Example 15-mark answer in pseudocode</p> <pre> // meaningful identifiers and appropriate data structures // similar variables grouped to save space as in HL languages DECLARE Clubs : ARRAY[1:12] OF STRING DECLARE Statistics : ARRAY[1:12, 1:3] OF INTEGER DECLARE Points : ARRAY[1 : 12] OF INTEGER DECLARE TieIndex : ARRAY[1 : 12] OF INTEGER DECLARE Matches, Won, Drawn, Lost : INTEGER DECLARE Counter, Maximum, TieCheck, OutLoop : INTEGER DECLARE Max, Count : INTEGER // input number of matches REPEAT OUTPUT "How many matches have been played (Maximum 22) ? " INPUT Matches UNTIL Matches <= 22 // input of data as a single loop - multiple loops for data entry // is also acceptable. FOR Counter ← 1 TO 12 OUTPUT "Enter the name of the cricket club" INPUT Clubs[Counter] // input of match results with validation REPEAT OUTPUT "Enter the number of matches won, drawn and lost for ", Clubs[Counter] INPUT Won, Drawn, Lost IF Won + Drawn + Lost <> Matches THEN OUTPUT "Your inputs must total ", Matches, " please try again" ENDIF UNTIL Matches = Won + Drawn + Lost Statistics[Counter, 1] ← Won Statistics[Counter, 2] ← Drawn Statistics[Counter, 3] ← Lost // calculating and storing points </pre>	

Question	Answer	Marks
10	<pre> Points[Counter] ← Statistics[Counter, 1] * 12 + Statistics[Counter, 2] * 5 NEXT Counter // finding the highest points Max ← -100 FOR Maximum ← 1 TO 12 IF Points[Maximum] > Max THEN Max ← Points[Maximum] ENDIF NEXT Maximum // finding the winner(s) Count ← 0 FOR TieCheck ← 1 TO 12 IF Points[TieCheck] = Max THEN Count ← Count + 1 TieIndex[Count] ← TieCheck ENDIF Next TieCheck // outputting the results FOR OutLoop ← 1 TO Count OUTPUT "Winning Club(s): ", Clubs[TieIndex[OutLoop]] OUTPUT "Number of wins: ", Statistics[TieIndex[OutLoop], 1] Next OutLoop OUTPUT "Winning Points: ", Max </pre>	

Marking Instructions in italics			
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			
0	1–3	4–6	7–9
No creditable response.	At least one programming technique has been used. <i>Any use of selection, iteration, counting, totalling, input and output.</i>	Some programming techniques used are appropriate to the problem. <i>More than one technique seen applied to the scenario, check the list of techniques needed.</i>	The range of programming techniques used is appropriate to the problem. <i>All criteria stated for the scenario have been covered by the use of appropriate programming techniques, check the list of techniques needed.</i>
	Some data has been stored but not appropriately. <i>Any use of variables or arrays or other language dependent data structures e.g. Python lists.</i>	Some of the data structures chosen are appropriate and store some of the data required. <i>More than one data structure used to store data required by the scenario.</i>	The data structures chosen are appropriate and store all the data required. <i>The data structures used store all the data required by the scenario.</i>

Marking Instructions in italics			
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. <i>Some of the data structures used have meaningful names.</i>	The majority of identifiers used are appropriately named. <i>Most of the data structures used have meaningful names.</i>	Suitable identifiers with names meaningful to their purpose have been used throughout. <i>All of the data structures used have meaningful names.</i>
	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. <i>Solution contains few lines of code with errors that attempt to perform a task given in the scenario.</i>	The solution contains parts that are inaccurate. <i>Solution contains lines of code with some errors that logically perform tasks given in the scenario. Ignore minor syntax errors.</i>	The solution is accurate. <i>Solution logically performs all the tasks given in the scenario. Ignore minor syntax errors.</i>
	The solution attempts at least one of the requirements. <i>Solution contains lines of code that attempt at least one task given in the scenario.</i>	The solution meets most of the requirements. <i>Solution contains lines of code that perform most tasks given in the scenario.</i>	The solution meets all the requirements given in the question. <i>Solution performs all the tasks given in the scenario.</i>