

# Cambridge IGCSE™

---

**COMPUTER SCIENCE****0478/21**

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><b>One mark for each correct line</b></p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding-bottom: 10px;"><b>Description</b></th> <th style="text-align: center; padding-bottom: 10px;"><b>Programming concept</b></th> </tr> </thead> <tbody> <tr> <td style="border: 1px solid black; padding: 5px;">a subroutine that may <b>not</b> return a value</td> <td style="border: 1px solid black; padding: 5px; text-align: center;">function</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">a value that is declared and used within a specific procedure</td> <td style="border: 1px solid black; padding: 5px; text-align: center;">procedure</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">a value that a procedure expects you to supply when it is called</td> <td style="border: 1px solid black; padding: 5px; text-align: center;">parameter</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">a subroutine that will always return a value</td> <td style="border: 1px solid black; padding: 5px; text-align: center;">global variable</td> </tr> <tr> <td></td> <td style="border: 1px solid black; padding: 5px; text-align: center;">local variable</td> </tr> </tbody> </table>	<b>Description</b>	<b>Programming concept</b>	a subroutine that may <b>not</b> 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
<b>Description</b>	<b>Programming concept</b>													
a subroutine that may <b>not</b> 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><b>One mark per mark point</b></p> <ul style="list-style-type: none"> <li>• Correct key word - CALL</li> <li>• Procedure name with correct parameters - Average( 25 , 50 )</li> </ul> <p>CALL Average( 25 , 50 )</p>	2												

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

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

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

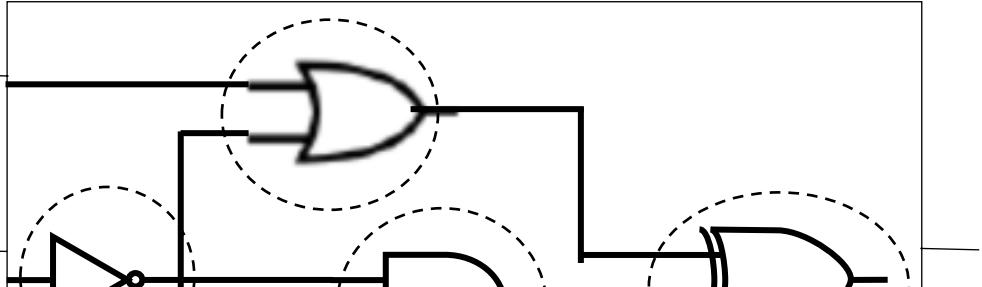
Question	Answer	Marks
4(b)	<p><b>One mark per mark point</b></p> <ul style="list-style-type: none"> <li>• Correct use of ROUND with 2 arguments separated by comma, for example ROUND( 5, 2) (in statement 13) ...</li> <li>• ... (Total / Limit,1) correct arguments</li> </ul> <p>For example:</p> <pre>OUTPUT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

Question	Answer	Marks
5	<p><b>One mark per mark point</b></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><b>MP1</b> Correct <b>Limit</b> column</p> <p><b>MP2</b> Correct <b>Count</b> column</p> <p><b>MP3</b> Array columns for input stage</p> <p><b>MP4</b> Array columns for sort stage</p> <p><b>MP5</b> Correct <b>Flag</b> column</p> <p><b>MP6</b> Correct <b>Swap</b> column</p> <p><b>MP7</b> Correct <b>Result</b> and <b>OUTPUT</b> columns</p> <table border="1" data-bbox="316 541 1882 1364"> <thead> <tr> <th colspan="14" data-bbox="316 541 1882 600">Numbers</th> </tr> <tr> <th data-bbox="316 600 444 659"><b>Limit</b></th><th data-bbox="444 600 572 659"><b>Count</b></th><th data-bbox="572 600 653 659">[1]</th><th data-bbox="653 600 734 659">[2]</th><th data-bbox="734 600 815 659">[3]</th><th data-bbox="815 600 898 659">[4]</th><th data-bbox="898 600 979 659">[5]</th><th data-bbox="979 600 1060 659">[6]</th><th data-bbox="1060 600 1140 659">[7]</th><th data-bbox="1140 600 1320 659"><b>Flag</b></th><th data-bbox="1320 600 1455 659"><b>Swap</b></th><th data-bbox="1455 600 1590 659"><b>Result</b></th><th data-bbox="1590 600 1882 659"><b>OUTPUT</b></th></tr> </thead> <tbody> <tr> <td data-bbox="316 659 444 717">7</td><td data-bbox="444 659 572 717">1</td><td data-bbox="572 659 653 717">47</td><td data-bbox="653 659 734 717"></td><td data-bbox="734 659 815 717"></td><td data-bbox="815 659 898 717"></td><td data-bbox="898 659 979 717"></td><td data-bbox="979 659 1060 717"></td><td data-bbox="1060 659 1140 717"></td><td data-bbox="1140 659 1320 717"></td><td data-bbox="1320 659 1455 717"></td><td data-bbox="1455 659 1590 717"></td><td data-bbox="1590 659 1882 717"></td></tr> <tr> <td data-bbox="316 717 444 776"></td><td data-bbox="444 717 572 776">2</td><td data-bbox="572 717 653 776"></td><td data-bbox="653 717 734 776">50</td><td data-bbox="734 717 815 776"></td><td data-bbox="815 717 898 776"></td><td data-bbox="898 717 979 776"></td><td data-bbox="979 717 1060 776"></td><td data-bbox="1060 717 1140 776"></td><td data-bbox="1140 717 1320 776"></td><td data-bbox="1320 717 1455 776"></td><td data-bbox="1455 717 1590 776"></td><td data-bbox="1590 717 1882 776"></td></tr> <tr> <td data-bbox="316 776 444 835"></td><td data-bbox="444 776 572 835">3</td><td data-bbox="572 776 653 835"></td><td data-bbox="653 776 734 835"></td><td data-bbox="734 776 815 835">52</td><td data-bbox="815 776 898 835"></td><td data-bbox="898 776 979 835"></td><td data-bbox="979 776 1060 835"></td><td data-bbox="1060 776 1140 835"></td><td data-bbox="1140 776 1320 835"></td><td data-bbox="1320 776 1455 835"></td><td data-bbox="1455 776 1590 835"></td><td data-bbox="1590 776 1882 835"></td></tr> <tr> <td data-bbox="316 835 444 894"></td><td data-bbox="444 835 572 894">4</td><td data-bbox="572 835 653 894"></td><td data-bbox="653 835 734 894"></td><td data-bbox="734 835 815 894"></td><td data-bbox="815 835 898 894">60</td><td data-bbox="898 835 979 894"></td><td data-bbox="979 835 1060 894"></td><td data-bbox="1060 835 1140 894"></td><td data-bbox="1140 835 1320 894"></td><td data-bbox="1320 835 1455 894"></td><td data-bbox="1455 835 1590 894"></td><td data-bbox="1590 835 1882 894"></td></tr> <tr> <td data-bbox="316 894 444 952"></td><td data-bbox="444 894 572 952">5</td><td data-bbox="572 894 653 952"></td><td data-bbox="653 894 734 952"></td><td data-bbox="734 894 815 952"></td><td data-bbox="815 894 898 952"></td><td data-bbox="898 894 979 952">80</td><td data-bbox="979 894 1060 952"></td><td data-bbox="1060 894 1140 952"></td><td data-bbox="1140 894 1320 952"></td><td data-bbox="1320 894 1455 952"></td><td data-bbox="1455 894 1590 952"></td><td data-bbox="1590 894 1882 952"></td></tr> <tr> <td data-bbox="316 952 444 1011"></td><td data-bbox="444 952 572 1011">6</td><td data-bbox="572 952 653 1011"></td><td data-bbox="653 952 734 1011"></td><td data-bbox="734 952 815 1011"></td><td data-bbox="815 952 898 1011"></td><td data-bbox="898 952 979 1011"></td><td data-bbox="979 952 1060 1011">63</td><td data-bbox="1060 952 1140 1011"></td><td data-bbox="1140 952 1320 1011"></td><td data-bbox="1320 952 1455 1011"></td><td data-bbox="1455 952 1590 1011"></td><td data-bbox="1590 952 1882 1011"></td></tr> <tr> <td data-bbox="316 1011 444 1070"></td><td data-bbox="444 1011 572 1070">7</td><td data-bbox="572 1011 653 1070"></td><td data-bbox="653 1011 734 1070"></td><td data-bbox="734 1011 815 1070"></td><td data-bbox="815 1011 898 1070"></td><td data-bbox="898 1011 979 1070"></td><td data-bbox="979 1011 1060 1070"></td><td data-bbox="1060 1011 1140 1070">70</td><td data-bbox="1140 1011 1320 1070"></td><td data-bbox="1320 1011 1455 1070"></td><td data-bbox="1455 1011 1590 1070"></td><td data-bbox="1590 1011 1882 1070"></td></tr> <tr> <td data-bbox="316 1070 444 1129"></td><td data-bbox="444 1070 572 1129">8</td><td data-bbox="572 1070 653 1129"></td><td data-bbox="653 1070 734 1129"></td><td data-bbox="734 1070 815 1129"></td><td data-bbox="815 1070 898 1129"></td><td data-bbox="898 1070 979 1129"></td><td data-bbox="979 1070 1060 1129"></td><td data-bbox="1060 1070 1140 1129"></td><td data-bbox="1140 1070 1320 1129">TRUE</td><td data-bbox="1320 1070 1455 1129"></td><td data-bbox="1455 1070 1590 1129"></td><td data-bbox="1590 1070 1882 1129"></td></tr> <tr> <td data-bbox="316 1129 444 1187"></td><td data-bbox="444 1129 572 1187">1</td><td data-bbox="572 1129 653 1187"></td><td data-bbox="653 1129 734 1187"></td><td data-bbox="734 1129 815 1187"></td><td data-bbox="815 1129 898 1187"></td><td data-bbox="898 1129 979 1187"></td><td data-bbox="979 1129 1060 1187"></td><td data-bbox="1060 1129 1140 1187"></td><td data-bbox="1140 1129 1320 1187">FALSE</td><td data-bbox="1320 1129 1455 1187"></td><td data-bbox="1455 1129 1590 1187"></td><td data-bbox="1590 1129 1882 1187"></td></tr> <tr> <td data-bbox="316 1187 444 1246"></td><td data-bbox="444 1187 572 1246">2</td><td data-bbox="572 1187 653 1246"></td><td data-bbox="653 1187 734 1246"></td><td data-bbox="734 1187 815 1246"></td><td data-bbox="815 1187 898 1246"></td><td data-bbox="898 1187 979 1246"></td><td data-bbox="979 1187 1060 1246"></td><td data-bbox="1060 1187 1140 1246"></td><td data-bbox="1140 1187 1320 1246"></td><td data-bbox="1320 1187 1455 1246"></td><td data-bbox="1455 1187 1590 1246"></td><td data-bbox="1590 1187 1882 1246"></td></tr> <tr> <td data-bbox="316 1246 444 1305"></td><td data-bbox="444 1246 572 1305">3</td><td data-bbox="572 1246 653 1305"></td><td data-bbox="653 1246 734 1305"></td><td data-bbox="734 1246 815 1305"></td><td data-bbox="815 1246 898 1305"></td><td data-bbox="898 1246 979 1305"></td><td data-bbox="979 1246 1060 1305"></td><td data-bbox="1060 1246 1140 1305"></td><td data-bbox="1140 1246 1320 1305"></td><td data-bbox="1320 1246 1455 1305"></td><td data-bbox="1455 1246 1590 1305"></td><td data-bbox="1590 1246 1882 1305"></td></tr> <tr> <td data-bbox="316 1305 444 1364"></td><td data-bbox="444 1305 572 1364">4</td><td data-bbox="572 1305 653 1364"></td><td data-bbox="653 1305 734 1364"></td><td data-bbox="734 1305 815 1364"></td><td data-bbox="815 1305 898 1364"></td><td data-bbox="898 1305 979 1364"></td><td data-bbox="979 1305 1060 1364"></td><td data-bbox="1060 1305 1140 1364"></td><td data-bbox="1140 1305 1320 1364"></td><td data-bbox="1320 1305 1455 1364"></td><td data-bbox="1455 1305 1590 1364"></td><td data-bbox="1590 1305 1882 1364"></td></tr> </tbody> </table>	Numbers														<b>Limit</b>	<b>Count</b>	[1]	[2]	[3]	[4]	[5]	[6]	[7]	<b>Flag</b>	<b>Swap</b>	<b>Result</b>	<b>OUTPUT</b>	7	1	47												2		50											3			52										4				60									5					80								6						63							7							70						8								TRUE					1								FALSE					2													3													4											
Numbers																																																																																																																																																																																								
<b>Limit</b>	<b>Count</b>	[1]	[2]	[3]	[4]	[5]	[6]	[7]	<b>Flag</b>	<b>Swap</b>	<b>Result</b>	<b>OUTPUT</b>																																																																																																																																																																												
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)			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	
6(b)	<p><b>One mark per mark point</b></p> <ul style="list-style-type: none"> <li>• A set of numbers is input / stored in an array</li> <li>• The numbers are sorted / Bubble sort into (ascending) order</li> <li>• The middle / median value is found / output</li> </ul>	3												

Question	Answer	Marks
7(a)	<p><b>One mark per mark point, max two</b></p> <ul style="list-style-type: none"> <li>• Data stored in a file gives a permanent copy / prevents the data from being lost</li> <li>• ... so it can be recalled / used again in a program / in another program</li> <li>• can be stored elsewhere / transferred to another computer</li> </ul>	2
7(b)	<p><b>One mark per mark point, max four</b></p> <p>MP1 Declaration of a string variable      MP2 Correct use of OPENFILE keywords for reading      MP3 Correct use READFILE with string variable      MP4 Correct use of UCASE and LENGTH functions <b>and</b> output of each      MP5 Correct use of CLOSEFILE</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>The circuit diagram shows three inputs: R, S, and T. Input R is connected to one input of a top-left NOT gate. The output of this NOT gate is connected to one input of a top-right OR gate. The other input of the OR gate is connected to the output of a middle-left NOT gate. The output of the middle-left NOT gate is connected to one input of a bottom-left AND gate. The other input of the bottom-left AND gate is connected to the output of a bottom-right NOT gate. The output of the bottom-right NOT gate is connected to the output Z.</p>	5

Question	Answer	Marks																																				
8(b)	<p><b>Four</b> marks for eight correct outputs. <b>Three</b> marks for six or seven correct outputs. <b>Two</b> marks for four or five correct outputs. <b>One</b> mark for two or three correct outputs</p> <table border="1" data-bbox="332 377 788 965"><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><b>One</b> 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><b>One</b> mark for each correct answer</p> <p>SUM          NumberInStock          SoftDrinks          Container          'Can'</p> <p>Correct code:</p> <pre>SELECT SUM (NumberInStock) FROM SoftDrinks WHERE Container = 'Can';</pre>	5
10	<p><b>Data structures required:</b></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><b>Requirements (techniques):</b></p> <p><b>R1</b> 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><b>R2</b> 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><b>R3</b> 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><b>Example 15-mark answer in pseudocode</b></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 &lt;= 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 &lt;&gt; 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] &gt; 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>	

<b>Marking Instructions in italics</b>			
<b>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</b>			
<b>0</b>	<b>1–3</b>	<b>4–6</b>	<b>7–9</b>
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>

<b>Marking Instructions in italics</b>			
<b>0</b>	<b>1–2</b>	<b>3–4</b>	<b>5–6</b>
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>