



---

# **GCE A LEVEL MARKING SCHEME**

---

**SUMMER 2018**

**A LEVEL  
COMPUTER SCIENCE - COMPONENT 1  
A500U10-1**

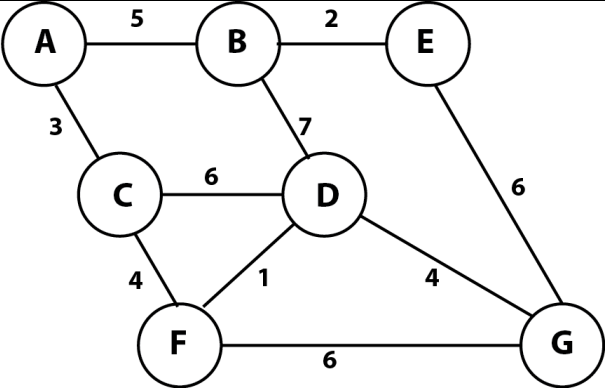
## **INTRODUCTION**

This marking scheme was used by WJEC for the 2018 examination. It was finalised after detailed discussion at examiners' conferences by all the examiners involved in the assessment. The conference was held shortly after the paper was taken so that reference could be made to the full range of candidates' responses, with photocopied scripts forming the basis of discussion. The aim of the conference was to ensure that the marking scheme was interpreted and applied in the same way by all examiners.

It is hoped that this information will be of assistance to centres but it is recognised at the same time that, without the benefit of participation in the examiners' conference, teachers may have different views on certain matters of detail or interpretation.

WJEC regrets that it cannot enter into any discussion or correspondence about this marking scheme.

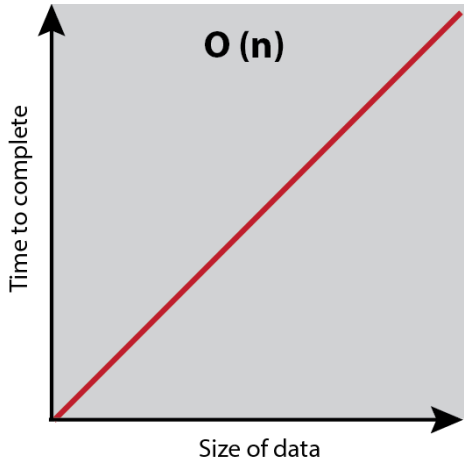
Qu	Answer	Mark	AO	Tot																														
1	<table border="1"> <tr> <td>A</td> <td>B</td> <td>A+B</td> <td>A.B</td> <td>(A+B).(A+B)</td> <td>(A.B)+(A.B)</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	A	B	A+B	A.B	(A+B).(A+B)	(A.B)+(A.B)	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1		2b	4
	A	B	A+B	A.B	(A+B).(A+B)	(A.B)+(A.B)																												
	0	0	0	0	0	0																												
	0	1	1	0	1	0																												
	1	0	1	0	1	0																												
1	1	1	1	1	1																													
1 mark for each of the last 4 correct columns.																																		
2(a) (i)	In optimisation, high-level general programming constructs are <b>replaced</b> by codes. The replacement codes are very efficient low-level programming codes.	1 1	1b	8																														
2(a)(ii)	The objectives of code optimization are to: Achieve the required output of the program. Increase the speed of the program Decrease demand on resources. Not delay the overall compilation process.	1 1 1 1																																
2(b)	This code involves repeated assignment of the identifier item. If the assignment of item is removed from the loop this will save CPU cycles.	1 1	2b																															
3(a)	One mark for each of the following up to a maximum of four A study leading to a preliminary report to the end user to advise on Technical practicality Cost effectiveness Time scale Budget To provide information required to support a decision to proceed.	1 1 1 1 1	1b	6																														
3(b)	Observation of a sample of operators as they use the current system. Document inspection, including business documents, user manuals and maintenance records.	1 1																																
4(a)(i)	$A + 1 = 1$	1	1a	8																														
4(a)(ii)	$A.\bar{A} = 0$	1																																
4(a)(iii)	$A + 0 = A$	1																																
4 (b)	$B.C.(\bar{C} + D) + C.D + C + \bar{A}$  $B.C.\bar{C} + B.C.D + C.D + C + \bar{A}$ $B.0 + B.C.D + C(D + 1) + \bar{A}$ $B.C.D + C + \bar{A}$ $C.(B.D + 1) + \bar{A}$ $C.1 + \bar{A}$ $C + \bar{A}$ Candidates may use more or fewer steps and correctly arrive at the answer – award full marks	1 1 1 1 1	2b																															

Qu	Answer	Mark	AO	Tot																																													
5	<p>A procedural programming language;</p> <ul style="list-style-type: none"> <li>• Supports a logical step-by-step process.</li> <li>• Allows the programmer to define precisely each step when performing a task.</li> <li>• Provides close control over the underlying operation of the hardware</li> <li>• Enables similar operations may be carried out at varying stages of the program execution</li> </ul>	1 1 1 1	1b	4																																													
6(a)	A shortest path algorithm will analyse a weighted network to identify the shortest route between two given vertices or nodes.	1 1	1b	7																																													
6(b)(i)	 <p>All correct connections shown No incorrect additional connections shown All values correct</p>	1 1 1	2b																																														
6(b)(ii)	<table border="1" data-bbox="256 1144 1166 1346"> <thead> <tr> <th>Step</th> <th>Vertex</th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> <th>G</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>A</td> <td>0</td> <td>5</td> <td>3</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> </tr> <tr> <td>2</td> <td>C</td> <td>X</td> <td>X</td> <td>0</td> <td>9</td> <td>X</td> <td>7</td> <td>X</td> </tr> <tr> <td>3</td> <td>F</td> <td>X</td> <td>X</td> <td>X</td> <td>8</td> <td>X</td> <td>X</td> <td>X</td> </tr> <tr> <td>4</td> <td>D</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>12</td> </tr> </tbody> </table> <p>Correct sequence A, C, F, D, G Correct total = 12</p>	Step	Vertex	A	B	C	D	E	F	G	1	A	0	5	3	X	X	X	X	2	C	X	X	0	9	X	7	X	3	F	X	X	X	8	X	X	X	4	D	X	X	X	X	X	X	12	1 1		
Step	Vertex	A	B	C	D	E	F	G																																									
1	A	0	5	3	X	X	X	X																																									
2	C	X	X	0	9	X	7	X																																									
3	F	X	X	X	8	X	X	X																																									
4	D	X	X	X	X	X	X	12																																									
7(a)	<p>One mark for each of the following up to a maximum of five.</p> <p>A stack is a container of objects that are inserted and removed according to the last-in first-out (LIFO) / first-in last-out (FILO) principle. It is a limited access data structure - elements can be added and removed from the stack only at the top push adds an item to the top of the stack, pop removes the item from the top. A stack can be used as a recursive data structure. A stack is either empty or it consists of a top and the rest which is a stack Underflow occurs when an attempt is made to pop an empty stack / overflow occurs when an attempt is made to add to a full stack</p>	1 1 1 1 1	1b	9																																													

Qu	Answer	Mark	AO	Tot
7(b)	<p>Indicative content</p> <pre>If stackPointer &lt; stackMaximum then     stackPointer = stackPointer + 1     stackArray(stackPointer) = dataItem Else Msg"Stack is full - your data has not been saved"</pre> <p>Adjust stack pointer Suitable output message</p>	<p>1 1</p>	3b	
7(c)	<p>Indicative content</p> <pre>If stackPointer &gt; 0 then     dataItem = stackArray(stackPointer)     stackPointer = stackPointer - 1 Else Msg"Stack is empty - no data can be retrieved"</pre> <p>Adjust stack pointer Suitable output message</p>	<p>1 1</p>		
8	<p>One mark for each as indicated up to a maximum of five.</p> <p>Initial set up &lt;letter&gt; ::= A B C. .Y Z &lt;digit&gt; ::= 0 1 2 . .8 9 &lt;Sletter&gt; ::= A B..G H</p> <p>Number definition &lt;Number&gt; ::= &lt;digit&gt;&lt;digit&gt; &lt;ShelfNumber&gt; ::= &lt;digit&gt; &lt;digit&gt;&lt;digit&gt; &lt;digit&gt;&lt;digit&gt;&lt;digit&gt;</p> <p>Code setup &lt;Man&gt; ::= &lt; letter&gt;&lt; letter&gt; &lt;letter&gt;&lt;Man&gt; &lt;Shelf&gt; ::= &lt;Sletter&gt;&lt;ShelfNumber&gt;</p> <p>&lt;Code&gt; ::= &lt;Man&gt;_&lt;Number&gt;&lt;Shelf&gt;</p> <p>Answer not correct if BNF notation not used correctly. Alternative solutions must involve recursion (x2) for full marks Must include _ for full marks.</p>	<p>1 1 1 1 1</p>	2b	5
9(a)	<p>A selection construct will use a logical condition to determine which line of code is to be processed next. If the condition is true then action 1 will be carried out. If the condition is false then action 2 will be carried out. Accept a correct example.</p>	<p>1 1</p>	1b	4
9(b)	<p>Nesting is when one selection statement is contained by another selection construct. If a logical condition is true, action 1 is carried out and then a second selection condition will govern whether action 3 or action 4 should be executed next. Accept a correct example.</p>	<p>1 1</p>		

Qu	Answer	Mark	AO	Tot
10	<p><b>Indicative algorithm</b></p> <pre> monthlyPay is real threshold is real upperEarnings is real lowerRate is integer upperRate is integer noEmployees is integer arrayEmployees() is real NI is real flag is Boolean  set threshold = 671.00 set upperEarnings = 3583.00 set lowerRate = 12 set upperRate = 2  output "enter number of employees" input noEmployees  for i = 1 to noEmployees  if monthlyPay(i) &lt;threshold then     NI(i) = 0.0 end if  if monthlypay(i)&gt; threshold and monthlyPay &lt;=upperEarnings then     NI(i) = (monthlyPay - threshold) * lowerRate end if  if monthlyPay(i) &gt; upperEarnings then     NI(i) = ((upperEarnings - threshold) * lowerRate) + ((monthlyPay)- upperEarnings))*upperRate end if next if  set flag = false for i = 1 to noEmployees - 1     for j = i+1 to noEmployees         set flag = false         if monthlyPay(i)&lt;monthlyPay(j) then             monthlyPay(i)= temp             monthlyPay(j) = monthlyPay(i)             monthlyPay(i) = temp             flag = true         endif     next j      if flag = false then i = noEmployees next i </pre>		3b	9

Qu	Answer	Mark	AO	Tot
	<pre> for i = 1 to noEmployees   output monthlyPay(i), NI(i) next i  end  declarations with sensible variable names correct numeric data types correct assignment of monthly wages and NI rates loop for NI calculation If statements for less than threshold and lower rate If statement for higher rate Sort in ascending order Use of flag in sort Suitable output </pre>	1 1 1 1 1 1 1 1 1 1		
11(a)	<pre> graph TD   Peugeot --&gt; Ford   Peugeot --&gt; Vauxhall   Ford --&gt; BMW   Ford --&gt; Mercedes   BMW --&gt; Audi   Vauxhall --&gt; Renault   Renault --&gt; Toyota </pre> <p>Correct root node Correct level 1 and correct level 2 Correct level 3</p>	1 1 1	2b	6
11(b)	<pre> graph TD   Peugeot --&gt; Ford   Peugeot --&gt; Vauxhall   Ford --&gt; BMW   Ford --&gt; Mercedes   BMW --&gt; Audi   Mercedes --&gt; Jaguar   Vauxhall --&gt; Renault   Renault --&gt; Toyota </pre>	1		
11(c)	<i>Audi, BMW, Jaguar, Mercedes, Ford, Toyota, Renault, Vauxhall, Peugeot</i> 1 mark for correct position of root, 1 mark for correct order and all nodes	1 1		

Qu	Answer	Mark	AO	Tot
12(a)(i)	$O(1)$	1	1a	10
12(a)(ii)	Accessing an array Accept: Memory complexity of a correct example	1	1b	
12(b)	 <p>Time axis correctly labeled. Input (size) axis correctly labeled Correct straight line graph</p>	1 1 1	2b	
12 (c)	<p>Looping through a list - as the size of the list increases the time taken increases in direct proportion. Nested statements – increase in time is directly proportional to the increase in input size (number of statements). Correct example Directly proportional relationship</p>	1 1		
12(d)	<p><math>O(\log_2 N)</math> Binary chop – time increase becomes an exponent e.g if <math>x = 16</math>, <math>y = \log_2 16 = 4</math>. As data size increases the process cuts the data set in 2 each time and therefore less data is searched.</p> <p>Reference to binary chop Logarithmic relationship Logarithmic relationship well explained or exemplified</p>	1 1 1	1b	



Qu	Answer	Mark	AO	Tot																										
13(a)	<p>Indicative Content</p> <table border="1"> <thead> <tr> <th>Reserved word</th> <th>Token (Hex)</th> </tr> </thead> <tbody> <tr> <td>Input</td> <td>3A</td> </tr> <tr> <td>=</td> <td>3B</td> </tr> <tr> <td>*</td> <td>3C</td> </tr> <tr> <td>Output</td> <td>3D</td> </tr> <tr> <td>(</td> <td>3E</td> </tr> <tr> <td>)</td> <td>3F</td> </tr> </tbody> </table> <p>All reserved words and symbols Unique Hex tokens</p> <table border="1"> <thead> <tr> <th>User Identifier</th> <th>Type</th> <th>Token (Hex)</th> </tr> </thead> <tbody> <tr> <td>Area</td> <td>Real</td> <td>2A</td> </tr> <tr> <td>Pi</td> <td>Real</td> <td>2B</td> </tr> <tr> <td>Radius</td> <td>Real</td> <td>2C</td> </tr> </tbody> </table> <p>All user identifiers Correct type and unique Hex tokens</p>	Reserved word	Token (Hex)	Input	3A	=	3B	*	3C	Output	3D	(	3E	)	3F	User Identifier	Type	Token (Hex)	Area	Real	2A	Pi	Real	2B	Radius	Real	2C	1 1  1 1	2b	5
Reserved word	Token (Hex)																													
Input	3A																													
=	3B																													
*	3C																													
Output	3D																													
(	3E																													
)	3F																													
User Identifier	Type	Token (Hex)																												
Area	Real	2A																												
Pi	Real	2B																												
Radius	Real	2C																												
13(b)	<p>Indicative Content</p> <p>2A. 3B. 2B. 3C. 2C. 3C. 2C All correct as shown in tables.</p>	1																												
14	<p>Indicative content</p> <p><b>Analysis</b>, descriptions of;</p> <ul style="list-style-type: none"> <li>• Abstraction / reduce problem to essential features</li> <li>• Decomposition / top down approach</li> <li>• DFD's / illustration of data flows</li> </ul> <p><b>Design</b> of,</p> <ul style="list-style-type: none"> <li>• Data structures / data types / variables and constants</li> <li>• Algorithms / pseudo code / flowcharts of processes</li> <li>• Sub routines</li> <li>• HCI / inputs / outputs.</li> <li>• Test data - typical, extreme and erroneous.</li> <li>• Prototyping</li> </ul> <p><b>Implementation</b>; consideration of</p> <ul style="list-style-type: none"> <li>• Type and level of language and IDE</li> <li>• Translation method</li> </ul> <p>and writing / de-bugging of code</p> <p><b>Documentation</b></p> <ul style="list-style-type: none"> <li>• Description of an ongoing process</li> <li>• User instructions, maintenance manuals</li> </ul> <p><b>Testing</b>, when and by whom</p> <ul style="list-style-type: none"> <li>• Alpha</li> <li>• Beta</li> </ul>		1b	15																										

3	<p style="text-align: center;"><b>11-15 marks</b></p> <p>The candidate has:</p> <ul style="list-style-type: none"> <li>• written an extended response which is coherent, relevant, and logically structured</li> <li>• shown clear understanding of the requirements of the question and a clear knowledge of the topics as specified in the indicative content. Clear knowledge is defined as responses that provide relevant detailed points about program development, which relate to an extensive amount of the indicative content.</li> <li>• addressed the question appropriately with minimal repetition and no irrelevant material</li> <li>• has presented detailed description</li> <li>• effectively drawn together different areas of knowledge, skills and understanding from all relevant areas across the course of study</li> <li>• used appropriate technical terminology confidently and accurately.</li> </ul>			
2	<p style="text-align: center;"><b>6 - 10 marks</b></p> <p>The candidate has:</p> <ul style="list-style-type: none"> <li>• written a response that has an adequate line of reasoning with elements of coherence, relevance, and logical structure</li> <li>• shown adequate understanding of the requirements of the question and a satisfactory knowledge of the topics as specified in the indicative content. Satisfactory knowledge is defined as responses that provide relevant points about the stages of program development, which relate to the indicative content.</li> <li>• presented descriptions with some detail</li> <li>• drawn together different areas of knowledge, skills and understanding from a number of areas across the course of study</li> <li>• used appropriate technical terminology.</li> </ul>			
1	<p style="text-align: center;"><b>1- 5 marks</b></p> <p>The candidate has:</p> <ul style="list-style-type: none"> <li>• written a response that lacks sufficient reasoning and structure</li> <li>• produced descriptions that lack detail</li> <li>• attempted to address the question but has demonstrated superficial knowledge of the topics specified in the indicative content. Superficial knowledge is defined as responses that provide limited relevant points about the stages of program development</li> <li>• used limited technical terminology.</li> </ul>			
0	Response not credit worthy or not attempted.			