GCE A LEVEL MARKING SCHEME

SUMMER 2018

A LEVEL
COMPUTER SCIENCE - COMPONENT 1
A500U10-1

© WJEC CBAC Ltd.

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.

© WJEC CBAC Ltd.

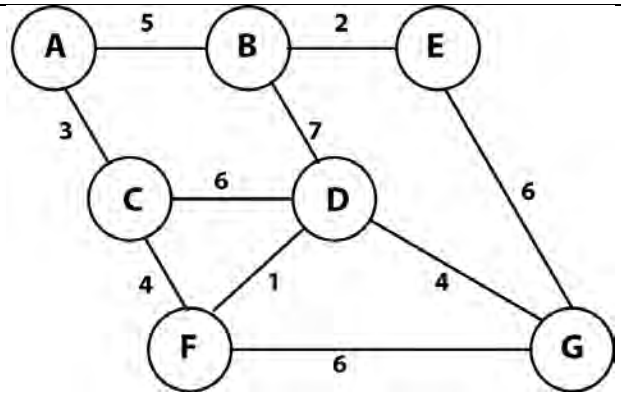
<table>
<tr><th>Qu</th><th>Answer</th><th>Mark</th><th>AO</th><th>Tot</th></tr>
<tr><td>1</td><td>
<table>
<tr><td>A</td><td>B</td><td>A+B</td><td>A.B</td><td>(A+B).(A+B)</td><td>(A.B)+(A.B)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</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>
1 mark for each of the last 4 correct columns.</td><td></td><td>2b</td><td>4</td></tr>
<tr><td>2(a) (i)</td><td>In optimisation, high-level general programming constructs are replaced by codes.
The replacement codes are very efficient low-level programming codes.</td><td>1
1</td><td>1b</td><td>8</td></tr>
<tr><td>2(a)(ii)</td><td>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.</td><td>
1
1
1
1</td><td>1a</td><td></td></tr>
<tr><td>2(b)</td><td>This code involves repeated assignment of the identifier item.
If the assignment of item is removed from the loop this will save CPU cycles.</td><td>1
1</td><td>2b</td><td></td></tr>
<tr><td>3(a)</td><td>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.</td><td>

1
1
1
1
1</td><td>1b</td><td>6</td></tr>
<tr><td>3(b)</td><td>Observation of a sample of operators as they use the current system.
Document inspection, including business documents, user manuals and maintenance records.</td><td>1
1</td><td>1a</td><td></td></tr>
<tr><td>4(a)(i)</td><td>$A + 1 = 1$</td><td>1</td><td>1a</td><td>8</td></tr>
<tr><td>4(a)(ii)</td><td>$A.\overline{A}$ = 0</td><td>1</td><td></td><td></td></tr>
<tr><td>4(a)(iii)</td><td>$A + 0 = A$</td><td>1</td><td></td><td></td></tr>
<tr><td>4 (b)</td><td>$B.C.(\overline{C} + D) + C.D + C + \overline{A}$

$B.C.\overline{C} + B.C.D + C.D + C + \overline{A}$
$B.0 + B.C.D + C\,(D + 1) + \overline{A}$
$B.C.D + C + \overline{A}$
$C.\,(B.D + 1) + \overline{A}$
$C.1 + \overline{A}$
$C + \overline{A}$
Candidates may use more or fewer steps and correctly arrive at the answer – award full marks</td><td>

1
1
1
1
1</td><td>2b</td><td></td></tr>
</table>

© WJEC CBAC Ltd.

<table>
<tr><th>Qu</th><th>Answer</th><th>Mark</th><th>AO</th><th>Tot</th></tr>
<tr><td>5</td><td>A procedural programming language;
• Supports a logical step-by-step process.
• Allows the programmer to define precisely each step when performing a task.
• Provides close control over the underlying operation of the hardware
• Enables similar operations may be carried out at varying stages of the program execution</td><td>1
1
1
1</td><td>1b</td><td>4</td></tr>
<tr><td>6(a)</td><td>A shortest path algorithm will analyse a weighted network
to identify the shortest route between two given vertices or nodes.</td><td>1
1</td><td>1b</td><td>7</td></tr>
<tr><td>6(b)(i)</td><td> 

All correct connections shown
No incorrect additional connections shown
All values correct</td><td>
1
1
1</td><td>2b</td><td></td></tr>
<tr><td>6(b)(ii)</td><td>
<table>
<tr><td>Step</td><td>Vertex</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td></tr>
<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>O</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>
</table>
Correct sequence A, C, F, D, G
Correct total = 12</td><td>

1
1</td><td></td><td></td></tr>
<tr><td>7(a)</td><td>One mark for each of the following up to a maximum of five.

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</td><td>

1
1
1
1
1
1</td><td>1b</td><td>9</td></tr>
</table>

© WJEC CBAC Ltd.

Qu	Answer	Mark	AO	Tot

7(b)	Indicative content
	
	`If stackPointer < stackMaximum then`
	`  stackPointer = stackPointer + 1`
	`  stackArray(stackPointer) = dataItem`
	`Else`
	`Msg"Stack is full – your data has not been saved"`
	
	Adjust stack pointer
	Suitable output message
		
		1
		1
			3b	

7(c)	Indicative content
	
	`If stackPointer > 0 then`
	`  dataItem = stackArray(stackPointer)`
	`  stackPointer = stackPointer - 1`
	`Else`
	`Msg"Stack is empty – no data can be retrieved"`
	
	Adjust stack pointer
	Suitable output message
		
		1
		1		

8	One mark for each as indicated upt to a maximum of five. Initial set up <letter> ::= A|B|C. .Y|Z <digit>::= 0|1|2 . .8|9 <Sletter>::= A|B..G|H Number definition <Number>::= <digit><digit> <ShelfNumber>::=<digit>|<digit><digit>|<digit><digit><digit> Code setup <Man> ::= < letter>< letter>|<letter><Man> <Shelf>::= <Sletter><ShelfNumber> <Code>::= <Man>_<Number><Shelf> Answer not correct if BNF notation not used correctly. Alternative solutions must involve recursion (x2) for full marks Must include _ for full marks.	 1 1 1 1 1 	2b	5
9(a)	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.	1 1	1b	4
9(b)	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.	1 1		

© WJEC CBAC Ltd.

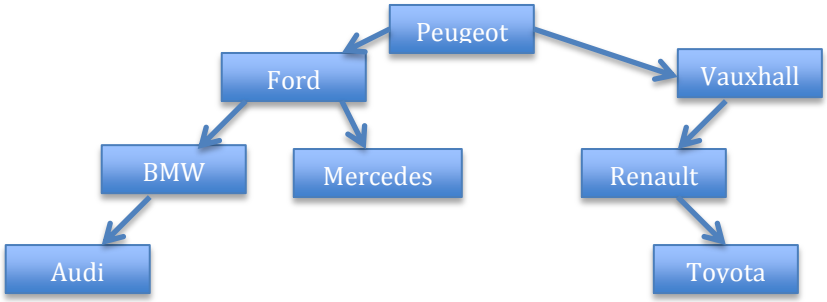
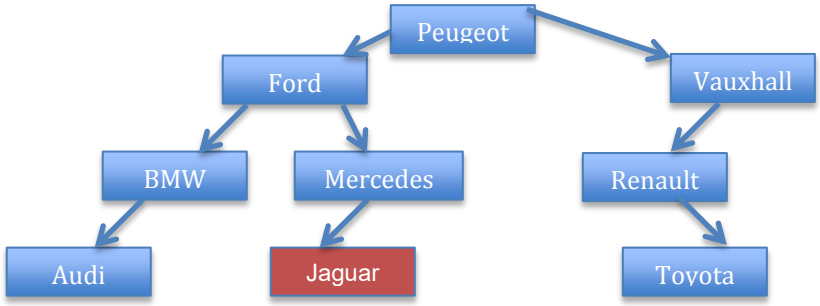
Qu	Answer	Mark	AO	Tot

10	**Indicative algorithm**
	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) <threshold then
	NI(i) = 0.0
	end if
	
	if monthlypay(i)> threshold and monthlyPay
	<=upperEarnings then
	NI(i) = (monthlyPay – threshold) *
	lowerRate
	end if
	
	if monthlyPay(i) > 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)<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
			3b	9

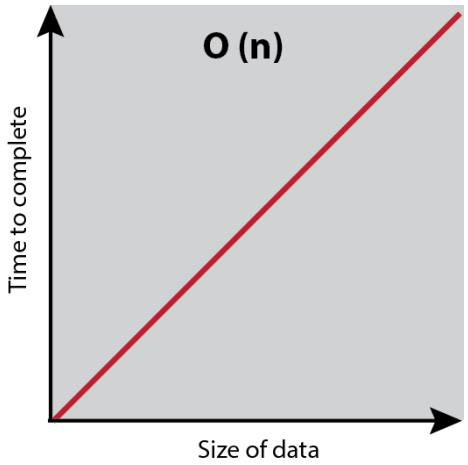
© WJEC CBAC Ltd.

Qu	Answer	Mark	AO	Tot

	`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
		
		1
		1
		1
		1
		1
		1
		1
		1
		1		

11(a)	 Correct root node Correct level 1 and correct level 2 Correct level 3	 1 1 1	2b	6
11(b)		1		
11(c)	*Audi, BMW, Jaguar, Mercedes, Ford, Toyota, Renault, Vauxhall, Peugeot* 1 mark for correct position of root, 1 mark for correct order and all nodes	1 1		

© WJEC CBAC Ltd.

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)	 Time axis correctly labeled. Input (size) axis correctly labeled Correct straight line graph	 1 1 1	2b	
12 (c)	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	 1 1		
12(d)	O(log_2N) Binary chop – time increase becomes an exponent e.g if x = 16, y = log2 16 = 4. As data size increases the process cuts the data set in 2 each time and therefore less data is searched. Reference to binary chop Logarithmic relationship Logarithmic relationship well explained or exemplified	 1 1 1	1b	

© WJEC CBAC Ltd.

<table>
<tr><th>Qu</th><th>Answer</th><th>Mark</th><th>AO</th><th>Tot</th></tr>
<tr><td>13(a)</td><td>Indicative Content

<table><tr><td>Reserved word</td><td>Token (Hex)</td></tr><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></table>All reserved words and symbols
Unique Hex tokens

<table><tr><td>User Identifier</td><td>Type</td><td>Token (Hex)</td></tr><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></table>All user identifiers
Correct type and unique Hex tokens</td><td>1
1

1
1</td><td>2b</td><td>5</td></tr>
<tr><td>13(b)</td><td>Indicative Content

2A. 3B. 2B. 3C. 2C. 3C. 2C
All correct as shown in tables.</td><td>

1</td><td></td><td></td></tr>
<tr><td>14</td><td>Indicative content

Analysis, descriptions of;
• Abstraction / reduce problem to essential features
• Decomposition / top down approach
• DFD’s / illustration of data flows
Design of,
• Data structures / data types / variables and constants
• Algorithms / pseudo code / flowcharts of processes
• Sub routines
• HCI / inputs / outputs.
• Test data - typical, extreme and erroneous.
• Prototyping
Implementation; consideration of
• Type and level of language and IDE
• Translation method
and writing / de-bugging of code
Documentation
• Description of an ongoing process
• User instructions, maintenance manuals
Testing, when and by whom
• Alpha
• Beta</td><td></td><td>1b</td><td>15</td></tr>
</table>

© WJEC CBAC Ltd.

3	**11-15 marks** The candidate has: • written an extended response which is coherent, relevant, and logically structured • 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. • addressed the question appropriately with minimal repetition and no irrelevant material • has presented detailed description • effectively drawn together different areas of knowledge, skills and understanding from all relevant areas across the course of study • used appropriate technical terminology confidently and accurately.			
2	**6 - 10 marks** The candidate has: • written a response that has an adequate line of reasoning with elements of coherence, relevance, and logical structure • 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. • presented descriptions with some detail • drawn together different areas of knowledge, skills and understanding from a number of areas across the course of study • used appropriate technical terminology.			
1	**1- 5 marks** The candidate has: • written a response that lacks sufficient reasoning and structure • produced descriptions that lack detail • 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 • used limited technical terminology.			
0	Response not credit worthy or not attempted.			

© WJEC CBAC Ltd.