GCE A LEVEL MARKING SCHEME

SUMMER 2017

A LEVEL (NEW)
COMPUTER SCIENCE - UNIT 3
1500U30-1

INTRODUCTION

This marking scheme was used by WJEC for the 2017 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.

GCE A LEVEL COMPUTER SCIENCE

SUMMER 2017 MARK SCHEME

<table>
<tr><th>Qu</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>TOT</th></tr>
<tr><td>1(a)</td><td>Any one of the following up to a maximum of two:

Two benefits of using a touch screen human computer Interface are:
• No need for another pointing device such as a stylus
• Can pinch and expand to scale images/text
• Screen can be used for input as well as output so device can be small
• Intuitive, so easy for beginners to learn to use
• Limits number of peripherals needed</td><td>2</td><td></td><td>2a</td><td></td><td>2</td></tr>
<tr><td>1(b)</td><td>Any one of the following up to a maximum of four:

Difficulties creating a natural language interface include difficult to recognise:

• two words that sound the same such as ‘two’ and ‘to’
• user’s voice changes due to a cold or sore throat or other impediment
• colloquialisms or local dialect
• speech where there is background noise
• proper nouns that might not be in dictionary
• strong accents
• words from other languages in common use such as ‘sacré bleu’ in English</td><td>4</td><td>1b</td><td></td><td></td><td>4</td></tr>
<tr><td>2</td><td>The term ‘class’ refers to the written code which is used to define a template for an object
An object is an instance of a class (an actual thing created using the template)
Methods are actions (behaviours) that an object can perform or can be performed on the object</td><td>1

1
1</td><td>1b

1b
1b</td><td></td><td></td><td>3</td></tr>
<tr><td>3(a)</td><td><table><tr><th>A</th><th>B</th><th>C</th><th>Ā</th><th>Ā.B</th><th>P</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr></table>
Marking
one mark for all possible values of A, B and C correct
one mark for each correct column if column P correct award full marks</td><td>

1
3</td><td></td><td>

2a
2a</td><td></td><td>4</td></tr>
<tr><td>3(b)</td><td><table><tr><th>Bit number</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr><tr><td>Register contents</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>Mask</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>Result</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr></table>
Marking
Correct mask – one mark
Identify AND operation – one mark
produce correct result – one mark</td><td>

1
1
1</td><td></td><td>

2a
2a
2a</td><td></td><td>3</td></tr>
</table>

Qu	Answer	Mark	AO1	AO2	AO3	TOT
4(a)	One mark for each point up to a maximum of 6. Must address all three sections to gain maximum marks, otherwise maximum of 5. **Lexical analysis** • Comments and unneeded spaces are removed • Keywords, constants and identifiers are replaced by 'tokens' • A symbol table is created which holds the addresses of variables, labels and subroutines **Syntax analysis** • Tokens are checked to see if they match the spelling and grammar expected, using standard language definitions. This is done by parsing each token to determine if it uses the correct syntax for the programming language. • If syntax errors are found, error messages are produced **Semantic analysis** • Variables are checked to ensure that they have been properly declared and used • Variables are checked to ensure they are of the correct data type, e.g. real values are not being assigned to integers • Operations are checked to ensure that they are legal for the type of variable being used e.g. you would not try to store the result of a division operation as an integer	6	1b			6
4(b)	One mark for each of the following up to a maximum of two: Advantage of using a language that requires compiling compared with a language that requires interpreting are: Once compiled the program will run quickly / the object code will be efficient because the compiler will translate directly to the native code of the specific machine / optimise the code for the target hardware. Protection of intellectual property	 1 1 1	 1b 1b 1b			2
4(c)	Two advantages for a program developer of using a language that requires interpreting compared with language that requires compiling are: Debugging can be easier as interpreter will stop translation at the point where the error occurred and highlight the error for the programmer to deal with. Code is more portable as it is not machine dependent and will run on different hardware or in a browser (java script) For security when downloading code from the Internet so it can be checked before interpreting on the local machine (2 marks)	 1 1 1 1	 1b 1b 1b 1b			4
4(d)	The purpose of an assembler is to translate assembly language into machine (executable) code An assembler's source code is low level code, compliers translate high level source code. An assembly instruction which will translate to one machine code instruction, whereas single lines of high level code compile to many machine code instructions.	1 1 2	1a 1b 1b			4

Qu	Answer	Mark	AO1	AO2	AO3	TOT
5(a)	One possible solution is: $A \cdot (\overline{A \cdot B})$ $A \cdot (\overline{A \cdot B})$ use de Morgan´s Theorem on $(\overline{A \cdot B}) \rightarrow$ $\overline{A} + \overline{B}$ $A.(\overline{A} + \overline{B})$ use distributive law $\rightarrow$ $A.\overline{A} + A.\overline{B}$ $A.\overline{A} + A.\overline{B}$ use $A.\overline{A} = 0 \rightarrow$ $0 + A.\overline{B}$ $0 + A.\overline{B}$ use $0 + A = A \rightarrow$ $A.\overline{B}$ $A.\overline{B}$ Marking Correctly applying rules/identities to arrive at correct answer – 4 marks Correctly applying rules/identities but arriving at incorrect answer then one mark per correct rules/identity applied – **max** 3 marks NOTE Candidate must use De Morgan's law, however may use more or fewer rules and correctly arrive at the answer – award full marks	4		2a		4
5(b)	One possible solution is: $A.B + A.(B + C) + B.(B + C)$ $A.B + A.C + B.B + B.C$ $A.B + A.C + B + B.C$ $A.B + A.C + B$ $A.C + B$ NOTE Candidate may use more or fewer rules and correctly arrive at the answer – award full marks	4		2a		4

<table>
<tr><th>Qu</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>TOT</th></tr>
<tr><td>6</td><td>

Indicative Content

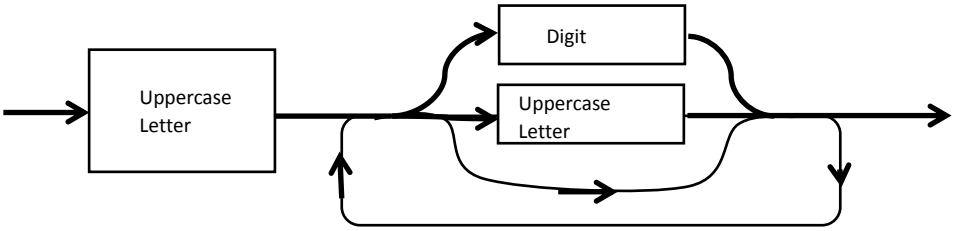
```
Start Procedure SortMyArray
 n is integer
 temp is integer
 swapped is boolean

 set n = length(myArray){returns the length of myArray}
 repeat
    set swapped = FALSE
    for i = 0 to (n - 1)
          if myArray[i] <= myArray[i + 1] then
            temp = myArray[i + 1]
            myArray[i + 1] = myArray[i]
            myArray[i] = temp
            swapped = TRUE
          end if
    end for
 until (swapped = FALSE)

End Procedure
```

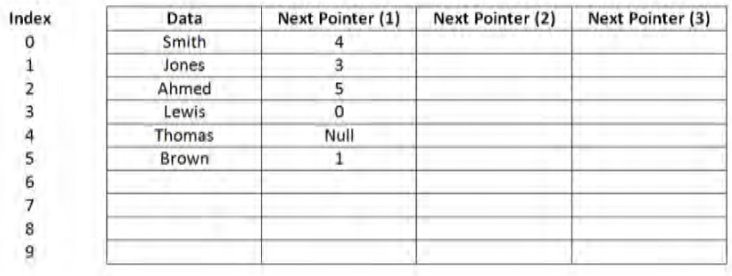
Marking
Initialise n
Terminating outer loop with n = 0
Use of an inner loop
Initialising swapped
Correct number of iterations for inner loop (reducing in size by one after each pass)
Comparing on of consecutive array elements
Move to temp
Swap array elements
Updating swapped

NOTE this algorithm swaps largest element with last in array but it could have swapped smallest with first element

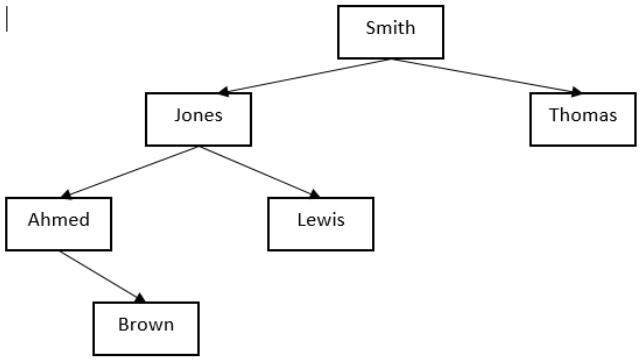
NOTE
A flowchart clearly showing all the marking points above is acceptable.
A programming language clearly showing all the marking points above is acceptable.

</td><td>1
1
1
1
1
1
1
1
1</td><td></td><td></td><td>3b
3b
3b
3b
3b
3b
3b
3b
3b</td><td>9</td></tr>
</table>

Qu	Answer	Mark	AO1	AO2	AO3	TOT
7(a)	Constant  Marking One mark for compulsory uppercase letter One mark for optional uppercase letter or digit One mark for null (skip) arrow One mark for repeating uppercase letter or digit (must have arrow)	4		2b		4
7(b)	<uppercase letter> ::= A|B|C . . . Y|Z <digit> ::= 0|1|2 . . . 8|9 <string> ::= Null | <uppercase letter> | <digit> |< uppercase letter> < string > | <digit><string> <constant> ::= <uppercase letter> <string> Marking: one mark for recursion: same item Left and Right are needed Cannot gain full marks unless completely correct Answer not correct if BNF notation used incorrectly Alternative correct solutions will be accepted	1 1 1 1		2b 2b 2b 2b		4
8	The assignment appears in the j loop and the j loop will execute N * N times = N^2 As N gets very big then N^2 gets bigger far quicker Therefore the number of assignments will tend to N^2 as N → ∞ The order of the algorithm for time efficiency is said to be $O(N^2)$ Marking: 1 mark for identifying i loop will execute n times 1 mark for identifying j loop will execute N^2 operations / N^2 times 1 mark for correct number of calculations N^2 + N 1 mark for determining that the order will be dominated by $N+N^2$ as N gets very big giving $O(N^2)$ for the algorithm	 1 1 1 1			 3c 3c 3c 3c	4

Qu	Answer	Mark	AO1	AO2	AO3	TOT
9(a)	An algorithm is an (unambiguous) set of instructions to solve a problem An alternative method to pseudo code of expressing an algorithm is a flowchart / **structured** English	1 1	1a 1a			2
9(b)	A recursive algorithm is one which calls itself (with a parameter which changes) and It must have a terminating condition (base case) to stop the calls	1 1	1a 1a			2
9(c)	One mark for each of the following up to a maximum of two: Memory overheads of stack use with many recursive procedural calls Difficult to program / dry run/debug if there is a fault Stack overflow Infinite loop	 1 1 1 1	 1b 1b 1b 1b			2
10	Design review tasks: Checking the correspondence between the actual design and its specification / user requirements / objectives / safety issues Confirming that the most appropriate techniques have been used Confirming the HCI is appropriate for the application	 1 1 1	 1b 1b 1b			3
11(a)	One mark for each of the following up to a maximum of two: When searching an ordered list the search can be terminated when an item greater than the search value (or less than) is reached When searching an unordered list the search cannot be terminated until the last item has been reached. For an ordered list a binary search can be used.	 1 1 1	 1b 1b 1b			2
11(b)(i)	 Start pointer – one mark End pointer – one mark All pointers correct – one mark	 1 1 1	 	 2b 2b 2b		3

Index	Data	Next Pointer (1)	Next Pointer (2)	Next Pointer (3)
0	Smith	4		
1	Jones	3		
2	Ahmed	5		
3	Lewis	0		
4	Thomas	Null		
5	Brown	1		
6				
7				
8				
9				

<table>
<tr><th>Qu</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>TOT</th></tr>
<tr><td>11(b)(ii)</td><td>
<table>
<tr><th>Index</th><th>Data</th><th>Next Pointer (1)</th><th>Next Pointer (2)</th><th>Next Pointer (3)</th></tr>
<tr><td>0</td><td>Smith</td><td></td><td>4</td><td></td></tr>
<tr><td>1</td><td>Jones</td><td></td><td>3</td><td></td></tr>
<tr><td>2</td><td>Ahmed</td><td></td><td>5</td><td></td></tr>
<tr><td>3</td><td>Lewis</td><td></td><td>6</td><td></td></tr>
<tr><td>4</td><td>Thomas</td><td></td><td>Null</td><td></td></tr>
<tr><td>5</td><td>Brown</td><td></td><td>7</td><td></td></tr>
<tr><td>6</td><td>Murphy</td><td></td><td>0</td><td></td></tr>
<tr><td>7</td><td>Collins</td><td></td><td>1</td><td></td></tr>
<tr><td>8</td><td></td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td><td></td></tr>
</table>
Murphy in location 6 and Collins in location 7 – one mark
Murphy pointer points at 0 – 1 mark
Brown pointer changed to point at 7 – one mark
Collins pointer points at 1 – one mark
</td><td>1
1
1
1</td><td></td><td>2b
2b
2b
2b</td><td></td><td>4</td></tr>
<tr><td>11(b)(iii)</td><td>
<table>
<tr><th>Index</th><th>Data</th><th>Next Pointer (1)</th><th>Next Pointer (2)</th><th>Next Pointer (3)</th></tr>
<tr><td>0</td><td>Smith</td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>Jones</td><td></td><td></td><td>3</td></tr>
<tr><td>2</td><td>Ahmed</td><td></td><td></td><td>5</td></tr>
<tr><td>3</td><td>Lewis</td><td></td><td></td><td>6</td></tr>
<tr><td>4</td><td>Thomas</td><td></td><td></td><td>Null</td></tr>
<tr><td>5</td><td>Brown</td><td></td><td></td><td>7</td></tr>
<tr><td>6</td><td>Murphy</td><td></td><td></td><td>4</td></tr>
<tr><td>7</td><td>Collins</td><td></td><td></td><td>1</td></tr>
<tr><td>8</td><td></td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td><td></td></tr>
</table>
Marking
Murphy pointer changed to point at 4 instead of 0 – one mark
Smith retained in the original list – 1 mark
</td><td>
1
1</td><td></td><td>
2b
2b</td><td></td><td>2</td></tr>
<tr><td>11(c)</td><td>


Correct root node
Correct level 1 and correct level 2
Correct level 3
</td><td>
1
1
1</td><td></td><td>
2b
2b
2b</td><td></td><td>3</td></tr>
</table>

Qu	Answer	Mark	AO1	AO2	AO3	TOT
12	**Stepping.** Execution of code one line at a time. Allows the programmer to examine each line of code in isolation to check that it is behaving as intended**.** **Break points.** A special marker that pauses execution of code at a preset position. Whilst paused the programmer inspects the test environment (registers, memory, files etc) to check that the program is functioning correctly. **Variable watch.** Used to view values in global and local variables as the code is executed in debug mode. Can be set to continually inspect variables which will be updated as the code is stepped through.	1 1 1 1 1 1	1b 1b 1b 1b 1b 1b			6

Qu	Answer	Mark	AO1	AO2	AO3	TOT
13	**Indicative content** **The Waterfall approach** • Sequential design process, in which various developers draft up all of the requirements for a system up front • By having all the requirements beforehand, everyone knows exactly what's needed • Client knows what to expect, including time frame, size, and cost of the project, and they know exactly what their product will do • If employees leave or join the development team, the strong documentation allows bringing new people up to speed quickly • Because the process is sequential, once a stage of development has been completed, you can't go back to a previous stage to make changes • If the initial requirements of the project are faulty in any way, the project is almost guaranteed to fail • The product is only tested once it is completed and if bugs were made early on, a large amount of code will be affected • If the client's needs change as the project goes on, the project will take longer than predicted **The Agile approach** • Incremental approach to development, in which developers start off with a simple project design instead of a huge document, and work on small modules at a time • It can be hard to employ new people into a team when you have less of a clearly defined structural process • It can be difficult to predict when the project will be completed, or how much it will ultimately cost • Changes can be made after the initial planning phase, and as the client makes changes the program can be re- written • Testing is done as the product is developed, ensuring that bugs are found earlier in the process • A smaller team can work on the product because you are removing the upper layers of project managers • There can be a closer relationship between the customer and the developer • When the end goal of the product is not clearly defined, Agile development is the most suitable approach • Sprints of work on the project are done and priorities of the project are discussed, evaluated, and tested Then, a simple product is released to the consumer and they are now able to use it and provide feedback **Which different programming paradigms would be suitable for each approach** • Procedural languages are suitable for both the Agile and Waterfall approach • Scripting Languages are suitable for both the Agile and Waterfall approach • Non-Procedural languages would be suitable for the Waterfall approach but some might not work as well with the Agile approach • Non-procedural programming languages require programmers to specify rules and facts which is more suitable for Waterfall • Object Orientated languages are suitable for both the Agile and Waterfall approach • Visual languages would be more suitable for Agile • 4th Generation languages are suitable for both the Agile and Waterfall approach		1b			10

Band	AO1b - Max 10 marks					
3	**8 - 10 marks** **The candidate has:** • written an extended response that has a sustained line of reasoning 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 of the implications of agile and waterfall approaches and suitable paradigms, which relate to an extensive amount of the indicative content. • addressed the question appropriately with minimal repetition and no irrelevant material • has presented a balanced discussion and justified their answer with examples • 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	**4 - 7 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 of the implications of agile and waterfall approaches and suitable paradigms, which relate to the indicative content. • presented a discussion with limited examples • 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- 3 marks** **The candidate has:** • written a response that that lacks sufficient reasoning and structure • produced a discussion which is not well developed • 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 of the implications of agile and waterfall approaches or suitable paradigms which relate to a limited amount the indicative content. • used limited technical terminology.					
0	Response not credit worthy or not attempted.					