GCE A LEVEL MARKING SCHEME

SUMMER 2017

A LEVEL (NEW)
COMPUTER SCIENCE - COMPONENT 1
A500U10-1

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

Qu	Answer	Mark	AO1	AO2	AO3	TOT
1(a) (i)	Storage space wasted as would need to cater for every possible 7 digit component number.	1		2a		1
1(a) (ii)	A suitable hashing algorithm will map component numbers onto a smaller range of addresses, by generating fewer digit address references.	2		2b		2
1(b)	Use progressive overflow, if the location is occupied use the next available location if the end of the file is reached wrap around and start searching from the beginning again. OR Flag original block and move data into designated overflow area for subsequent linear search	2 2	1b 1b			2
2(a)	One mark for each of the following: Inheritance enables new objects to take on the properties of existing objects. A superclass is used as the basis for inheritance. A class that inherits from a superclass is called a subclass. Inheritance defines relationships between classes and organises classes into groups. Inheritance enables classes that are similar to existing classes to be created by indicating differences (rather that starting again) and thereby allows code to be organised and re-used effectively	 1 1 1 1	 1b 1b 1b 1b			4
2(b)	Award one mark for each of the following up to a maximum of two: Abstraction Encapsulation Polymorphism Object Hierarchy	 1 1 1 1	 1b 1b 1b 1b			2
3(a)	A compiler takes the entire program as input to produce a machine code version of the program. An interpreter takes a single source code instruction as input, translates and executes it. A compiled program can be re-run without further translation. Interpretation needs to be repeated each time the program is run	1 1 1	1a 1a 1a			3
3(b)	Award one mark for each of the following up to a maximum of two: Errors are reported after compilation has finished. One error may cause many related/spurious errors Recompiling after fixing an error adds time to the process.	 1 1 1	 1b 1b 1b			2
3(c)	One mark for identifying the error and one mark for the example. Maximum of four marks. Example must match the error. Syntax error e.g. IF without ENDIF or punctuation error or spelling error if correct words given Linking error e.g. calling a standard function where the correct library has not been linked to the program Semantic Error e.g. Variable declared illegally	 1 1 1 1 1 1	 1a 1b 1a 1b 1a 1b			4

Qu	Answer	Mark	AO1	AO2	AO3	TOT
3(d)	Lookup tables to be used during the translation of this source code are:					5

Reserved word	Hex Token
input	5C
=	5D
*	5E
+	5F
output	60

Award 1 mark for all reserved words and symbols — Mark: 1, AO2: 2b
Award 1 mark for unique hex tokens — Mark: 1, AO2: 2b

Identifier	Type	Hex Token
basicCost	real	2C
VAT	real	2D
totalCost	real	2E

Award 1 mark for all identifiers — Mark: 1, AO2: 2b
Award 1 mark for data type = real — Mark: 1, AO2: 2b
Award 1 mark for unique hex tokens — Mark: 1, AO2: 2b

Qu	Answer	Mark	AO1	AO2	AO3	TOT
4(a)	$A.(\bar{A}+\bar{B})$ $A.\bar{A} + A.\bar{B}$ $0 + A.\bar{B}$ $A.\bar{B}$ 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

Qu	Answer	Mark	AO1	AO2	AO3	TOT
4(b)	One possible solution is: $$\bar{A}.B.C + A.B.\bar{C} + A.B.C + \bar{A}.B.\bar{C}$$ $\bar{A}.B.C + A.B(\bar{C} + C) + \bar{A}.B.\bar{C}$ $\bar{A}.B.C + A.B(1) + \bar{A}.B.\bar{C}$ $\bar{A}.B.C + A.B + \bar{A}.B.\bar{C}$ $\bar{A}.B.C + \bar{A}.B.\bar{C} + A.B$ $\bar{A}.B.C + \bar{A}.B.\bar{C} + A.B$ $\bar{A}.B(C + \bar{C}) + A.B$ $\bar{A}.B(1) + A.B$ $\bar{A}.B + A.B$ $B(\bar{A} + A)$ $B(1)$ B **NOTE** Candidate may use more or fewer rules and correctly arrive at the answer – award full marks	4		2a		4
5(a)	Example must match the error Perfective maintenance – to improve a system in use, Making improvements that are not major enough to justify a new system. Adaptive maintenance – to change a system in use. Making changes to suit revised working requirements / OS versions / new hardware	 1 1 1 1	 1a 1a 1a 1a			4
5(b)	Award one mark for each of the following up to a maximum of two: Variable list Data dictionary Class diagram List of sub routines Entity Relationship Diagram	 1 1 1 1 1	 1a 1a 1a 1a 1a			2
6(a)	Variables – a, b, Hypotenuse or Answer Parameter - x	1 1		2a 2a		2
6(b)	Value parameter is used in a calculation within a subprogram when you want to retain the original values	1 1	1b 1b			2
6(c)	Rounding. The result is given to the nearest value Truncation. The approximation of a numeric data item by ignoring all information beyond a given number of significant figures. Accuracy: Working to 2 decimal places the rounded result will be 9.49, producing an error of 0.0032, whereas truncation produces a result of 9.48 with an error of 0.0068 More than twice the rounding error	1 1 1 1	1a 1a	 2b 2b		4

Qu	Answer	Mark	AO1	AO2	AO3	TOT

7(a)

A	B	C	A.B	A NAND B	(A NAND B) + C	(A NAND B) NOR C
0	0	0	0	1	1	0
0	0	1	0	1	1	0
0	1	0	0	1	1	0
0	1	1	0	1	1	0
1	0	0	0	1	1	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	0	1	0

P = (A NAND **B)** NOR **C**

Marking
one mark for all possible values of A, B and C correct
one mark for correct column **A**NAND**B**
one mark for correct column **(A**NAND**B) + C**
one mark for correct column **(A**NAND**B)** NOR **C**

7(a): Mark 1, 1, 1, 1 — AO2 2a, 2a, 2a, 2a — TOT 4

7(b)

Bit number	7	6	5	4	3	2	1	0
Register contents	1	1	1	0	0	0	1	1
Mask	0	0	1	1	0	0	0	0
Result	0	0	1	0	0	0	0	0

One mark for each of the following:

Correct mask – one mark
Identify AND operation – one mark
produce correct result – one mark

7(b): Mark 1, 1, 1 — AO2 2a, 2a, 2a — TOT 3

8(a)

Marking
One mark for compulsory letter
One mark for optional letters with loop
One mark 3 compulsory digits

8(a): Mark 1, 1, 1 — AO2 2b, 2b, 2b — TOT 3

8(b)

<letter> ::= a|b|c. . . y|z <digit> ::= 0|1|2 . . . 8|9
<digits> ::= <digit><digit><digit

<letters> ::= Null | <letter> |< letter> < letters >
<email> ::= <letter> <letters><digits>@parkwood.ac.uk

Answer not correct if BNF notation used incorrectly

8(b): Mark 1, 1, 1, 1 — AO2 2b, 2b, 2b, 2b — TOT 4

Qu	Answer	Mark	AO1	AO2	AO3	TOT

9	Quicksort algorithm
	Indicative content
	
	Declare subprocedure QuickSort (myArray is string, indexLow is integer, indexHi is integer)
	
	Pivot is string
	tmpSwap is string
	tmpLow is integer
	tmpHi is integer
	
	tmpLow = indexLow
	tmpHi =indexHi
	
	pivot = myArray (int(indexLow + indexHi)/2))
	
	while (tmpLow <= tmpHi)
	
	      while (myArray(tmpLow) < pivot and tmpLow < indexHi)
	         tmpLow = tmpLow + 1
	      wend
	
	      while (pivot < myArray(tmpHi=i) and tmpHi > indexLow)
	         tmpHi = tmpHi – 1
	      wend
	
	      if (tmpLow <= tmpHi) then
	         tmpSwap = myArray(tmpLow)
	         myArray(tmpLow) = myArray(tmpHi)
	         myArray(tmpHi) = tmpSwap
	         tmpLow = tmpLow + 1
	         tmpHi = tmpHi -1
	      end if
	wend
	
	if (indexLow < tmpHi) then QuickSort (myArray , indexLow, tmpHi)
	if (tmpLow < indexHi) then QuickSort (my Array, tmpLow, indexHi)
	
	end sub
	One mark for each of the following up to a maximum of nine:
						9

	Declare sub procedure 'Quicksort'	1			3b	
	Initialise pointers (high and low)	1			3b	
	Set pivot to data at mid point	1			3b	
	Outer loop with terminating condition	1			3b	
	Compare data with pivot	1			3b	
	Increment / decrement pointers	1			3b	
	Swap elements	1			3b	
	Recursion index low, temp high	1			3b	
	Recursion index high, temp low	1			3b	
	Alternative solutions acceptable if correct					

Qu	Answer	Mark	AO1	AO2	AO3	TOT
10(a)	Evaluation of algorithm Sort Loop i will iterate a total number of n - 1 times. Loop j will iterate a total number of n^2 - 1 times. There are 2 operations in i loop: swap and start = start + 1 therefore there are 2(n - 1) operations carried out in the i loop. There is one operation in the j loop - a comparison, therefore there are n^2 operations carried out in the j loop. Adding these we get a total of n^2 + 2(n - 1) Therefore, n^2 will dominate(1 mark) making growth rate for time performance $O(n^2)$ (1 mark) **NOTE: Calculations might include assignment operations but these will not affect overall time so ignore.** Marking: 1 mark for identifying i loop will execute 2(n - 1) times. 1 mark for identifying j loop will execute n^2 times. 1 mark for correct number of calculations n^2 + 2(n - 1) 1 mark for determining that the order will be dominated by n^2 1 mark for determining that growth rate for time performance $O(n^2)$	1 1 1 1 1			3c 3c 3c 3c 3c	5
10(b)	Algorithm will need to store one array that will require N elements Total storage therefore = 1 x N As N increases the storage requirements will increase by N. Constant 1 will be insignificant so storage requirements will be O (N) accept O(1)	1 1			3c 3c	2
11 (a)	 Correct root node Correct level 1 and correct level 2 Correct level 3 and level 4	1 1 1		2b 2b 2b		3

Qu	Answer	Mark	AO1	AO2	AO3	TOT
11 (b)		1		2b		1
11(c)	1 mark for correct position of root, 1 mark for correct order and all nodes Philips, Blaupunkt, Panasonic, LG, Hisense, Samsung, Toshiba, Techwood, Sony.	2		2b		2
12(a)	Ambiguity is an uncertainty of meaning in which different interpretations are possible. High level programming languages must be unambiguous so that there is only one way to interpret each program statement and therefore enable accurate translation into machine code.	1 1 1	1a 1b 1b			3
12(b)	Award one mark for each of the following up to a maximum of four: Two words that sound the same (two, to) homonyms Dialect / accents Use of proper nouns Words from other languages in common use Voice patterns	 1 1 1 1 1	 1b 1b 1b 1b 1b			4
13	**Indicative content** Procedural languages Procedural languages are used in traditional programming based on algorithms or a logical step-by-step process for solving a problem They obey (ordered) instructions They carry out actions / calculations etc. A procedural programming language provides the programmer a way to define precisely each step when performing a task Allows tight control over the underlying operation of the hardware Used in (large complicated) programs where similar operations may be carried out at varying stages of the program execution Non-Procedural languages Non-procedural programming languages allow programmers to specify the results they want without specifying how to solve the problem Non-procedural languages are to do with rules / making queries / facts Used in database interrogation where retrieving answers are more important than the exact steps required to calculate the result Artificial intelligence, grammar checking and language translation applications are often written in a non-procedural language	10	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 about procedural and non-procedural languages and their uses 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 about procedural and non-procedural languages and their uses 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 about procedural and non-procedural languages and their uses which relate to a limited amount the indicative content. • used limited technical terminology.					
0	Response not credit worthy or not attempted.					