GCE

Computer Science

H046/02: Algorithms and problem solving

Advanced Subsidiary GCE

Mark Scheme for June 2019

OCR (Oxford Cambridge and RSA) is a leading UK awarding body, providing a wide range of qualifications to meet the needs of candidates of all ages and abilities. OCR qualifications include AS/A Levels, Diplomas, GCSEs, Cambridge Nationals, Cambridge Technicals, Functional Skills, Key Skills, Entry Level qualifications, NVQs and vocational qualifications in areas such as IT, business, languages, teaching/training, administration and secretarial skills.

It is also responsible for developing new specifications to meet national requirements and the needs of students and teachers. OCR is a not-for-profit organisation; any surplus made is invested back into the establishment to help towards the development of qualifications and support, which keep pace with the changing needs of today's society.

This mark scheme is published as an aid to teachers and students, to indicate the requirements of the examination. It shows the basis on which marks were awarded by examiners. It does not indicate the details of the discussions which took place at an examiners' meeting before marking commenced.

All examiners are instructed that alternative correct answers and unexpected approaches in candidates' scripts must be given marks that fairly reflect the relevant knowledge and skills demonstrated.

Mark schemes should be read in conjunction with the published question papers and the report on the examination.

© OCR 2019

Annotations

Annotation	Meaning
	Omission mark
	Benefit of the doubt
	Incorrect point
	Follow through
	Not answered question
	No benefit of doubt given
	Repeat
	Correct point
	Too vague
	Zero (big)
	Blank Page – this annotation must be used on all blank pages within an answer booklet (structured or unstructured) and on each page of an additional object where there is no candidate response.
	Level 1
	Level 2
	Level 3

Subject - specific Marking Instructions

INTRODUCTION

Your first task as an Examiner is to become thoroughly familiar with the material on which the examination depends. This material includes:

- the specification, especially the assessment objectives
- the question paper and its rubrics
- the mark scheme.

You should ensure that you have copies of these materials.

You should ensure also that you are familiar with the administrative procedures related to the marking process. These are set out in the OCR booklet Instructions for **Examiner**s. If you are examining for the first time, please read carefully Appendix 5 Introduction to Script Marking: **Notes for New Examiners**.

Please ask for help or guidance whenever you need it. Your first point of contact is your Team Leader.

USING THE MARK SCHEME

Please study this Mark Scheme carefully. The Mark Scheme is an integral part of the process that begins with the setting of the question paper and ends with the awarding of grades. Question papers and Mark Schemes are developed in association with each other so that issues of differentiation and positive achievement can be addressed from the very start.

This Mark Scheme is a working document; it is not exhaustive; it does not provide 'correct' answers. The Mark Scheme can only provide 'best guesses' about how the question will work out, and it is subject to revision after we have looked at a wide range of scripts.

The Examiners' Standardisation Meeting will ensure that the Mark Scheme covers the range of candidates' responses to the questions, and that all Examiners understand and apply the Mark Scheme in the same way. The Mark Scheme will be discussed and amended at the meeting, and administrative procedures will be confirmed. Co-ordination scripts will be issued at the meeting to exemplify aspects of candidates' responses and achievements; the co-ordination scripts then become part of this Mark Scheme.

Before the Standardisation Meeting, you should read and mark in pencil a number of scripts, in order to gain an impression of the range of responses and achievement that may be expected.

In your marking, you will encounter valid responses which are not covered by the Mark Scheme: these responses must be credited. You will encounter answers which fall outside the 'target range' of Bands for the paper which you are marking. Please mark these answers according to the marking criteria.

Please read carefully all the scripts in your allocation and make every effort to look positively for achievement throughout the ability range. Always be prepared to use the full range of marks.

LEVELS OF RESPONSE QUESTIONS:

The indicative content indicates the expected parameters for candidates' answers, but be prepared to recognise and credit unexpected approaches where they show relevance.

Using 'best-fit', decide first which set of BAND DESCRIPTORS best describes the overall quality of the answer. Once the band is located, adjust the mark concentrating on features of the answer which make it stronger or weaker following the guidelines for refinement.

- **Highest mark**: If clear evidence of all the qualities in the band descriptors is shown, the HIGHEST Mark should be awarded.
- **Lowest mark**: If the answer shows the candidate to be borderline (i.e. they have achieved all the qualities of the bands below and show limited evidence of meeting the criteria of the band in question) the LOWEST mark should be awarded.
- **Middle mark**: This mark should be used for candidates who are secure in the band. They are not 'borderline' but they have only achieved some of the qualities in the band descriptors.

Be prepared to use the full range of marks. Do not reserve (e.g.) high Band 3 marks 'in case' something turns up of a quality you have not yet seen. If an answer gives clear evidence of the qualities described in the band descriptors, reward appropriately.

	AO1	**AO2**	**AO3**
High (thorough)	Precision in the use of question terminology. Knowledge shown is consistent and well-developed. Clear appreciation of the question from a range of different perspectives making extensive use of acquired knowledge and understanding.	Knowledge and understanding shown is consistently applied to context enabling a logical and sustained argument to develop. Examples used enhance rather than detract from response.	Concerted effort is made to consider all aspects of a system/problem or weigh up both sides to an argument before forming an overall conclusion. Judgements made are based on appropriate and concise arguments that have been developed in response resulting in them being both supported and realistic.
Middle (reasonable)	Awareness of the meaning of the terms in the question. Knowledge is sound and effectively demonstrated. Demands of question understood although at times opportunities to make use of acquired knowledge and understanding not always taken.	Knowledge and understanding applied to context. Whilst clear evidence that an argument builds and develops through response there are times when opportunities are missed to use an example or relate an aspect of knowledge or understanding to the context provided.	There is a reasonable attempt to reach a conclusion considering aspects of a system/problem or weighing up both sides of an argument. However the impact of the conclusion is often lessened by a lack of supported judgements which accompany it. This inability to build on and develop

			lines of argument as developed in the response can detract from the overall quality of the response.
Low (basic)	Confusion and inability to deconstruct terminology as used in the question. Knowledge partial and superficial. Focus on question narrow and often one-dimensional.	Inability to apply knowledge and understanding in any sustained way to context resulting in tenuous and unsupported statements being made. Examples if used are for the most part irrelevant and unsubstantiated.	Little or no attempt to prioritise or weigh up factors during course of answer. Conclusion is often dislocated from response and any judgements lack substance due in part to the basic level of argument that has been demonstrated throughout response.

	Assessment Objective
AO1	Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation.
AO1.1	Demonstrate knowledge of the principles and concepts of abstraction, logic, algorithms, data representation or other as appropriate.
AO1.2	Demonstrate understanding of the principles and concepts of abstraction, logic, algorithms, data representation or other as appropriate.
AO2	Apply knowledge and understanding of the principles and concepts of computer science including to analyse problems in computational terms.
AO2.1	Apply knowledge and understanding of the principles and concepts of computer science.
AO2.2	Analyse problems in computational terms.
AO3	Design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions.
AO3.1	Design computer systems that solve problems.
AO3.2	Program computer systems that solve problems.
AO3.3	Evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions.

Question			Answer/Indicative content	Mark	Guidance
1	a	i	Abstraction	**1** **AO1.1**	
1	a	ii	e.g. • Reduces the amount of memory / processing required • Reduces complexity • Reality contains things that aren't relevant to a computer program • Reduces design / programming effort	**1** **AO1.2**	Need to allow FT from (a)(i) e.g. if decomposition is given as answer
1	a	iii	1 mark per bullet to max 4 e.g. • Remove details of the furniture • E.g. design elements • Remove details of the room • E.g. light switches • Replace objects with shapes/identifiers • E.g. set room shape as a rectangle	**4** **AO2.1 (2)** **AO2.2 (2)**	Need to allow FT from (a)(i) e.g. if decomposition is given as answer
1	b	i	1 mark per bullet to max 2 e.g. • Room size (width / length / height) • Room shape • Position of items in room (windows / door) • Furniture items required	**2** **AO2.1 (1)** **AO2.2 (1)**	Allow width / length / height as separate mark points Allow position of windows / door as separate mark points

1	b	ii	1 mark per bullet to max 2 e.g. • 3D image of the room • Image of the furniture items • Furniture to choose from / menu of library items • Error messages	**2** **AO2.1 (1)** **AO2.2 (1)**	Do not allow angles / perspective of room as well as image of room
1	c		1 mark per bullet to max 4 • Efficiency • Small sub-programs are easier to read / understand / modify • Write once and call repeated times • Avoids repeated code • Subroutines can be tested individually then added to the main program. • Can reuse in other programs • Can give procedures to different programmers to build	**4** **AO1.1 (1)** **AO1.2 (1)** **AO2.1 (1)** **AO2.2 (1)**	
1	d		1 mark per bullet to max 2 e.g. • Room dimensions must be greater than 0 • Furniture must fit in room • Width, length and height of room must have been entered	**2** **AO2.2 (2)**	The precondition must refer specifically to a condition that can be tested

1	e		**Mark Band 3–High Level** **(7-9 marks)** The candidate demonstrates thorough knowledge and understanding of reasons for the use of different testing strategies; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well-balanced. Evaluative comments are consistently relevant and well-considered. There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated. **Mark Band 2-Mid Level** **(4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of reasons for the use of different testing strategies; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are for the most part appropriate, although one or two opportunities for development are missed. There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.	**9** **AO1.1 (2)** **AO1.2 (2)** **AO2.1 (2)** **AO3.3 (3)**	**AO1: Knowledge and Understanding** The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: e.g. • Blackbox Use to test the functionality without knowledge of the inner workings • Whitebox Test the algorithms to ensure they do what they were designed to. Does not test functionality • Alpha Internal testing by the programmers before showing to end user • Beta Testing by third party/end users to ensure it meets requirements and is functional. Helps test usability. **AO2.1: Application** The selected knowledge/examples should be directly related to the specific question. The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive. • Discussion of how each strategy would be used in the program • She should use alpha testing during

			Mark Band 1-Low Level **(1-3 marks)** The candidate demonstrates a basic knowledge of testing strategies, with limited understanding shown; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgments if made are weak and unsubstantiated. The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear. **0 marks** No attempt to answer the question or response is not worthy of credit.		the creation to check the processes she is creating. • She should use black box testing to make sure the program produces the expected outputs i.e. entering the height, width and depth of a room and it producing these exact dimensions • She should use white box testing to make sure the algorithms work, e.g. entering the dimensions of a piece of furniture and then an algorithm calculating if it will fit • She should use acceptance testing by showing the software to the end user and walking them through the use of the software to prove it meets all the requirements **AO3.3: Evaluation** Candidates will need to consider a variety of viewpoints in relation to testing strategies and will make evaluative comments about the issues and solutions they are discussing. Candidates should suggest a combination of testing strategies to test a range of features.

<table>
<tr><td>2</td><td>a</td><td>i</td><td>08</td><td>1

AO2.1 (1)</td><td></td></tr>
<tr><td>2</td><td>a</td><td>ii</td><td>1 mark per bullet to max 3
• Compare the size/number of elements in the two arrays <code>data</code> and <code>nextData</code>
• To set the number of times the loop will run // set the value of <code>loopCount</code>
• … as many times as the array with the fewest items
• Otherwise it will attempt to add an empty value
• … which could cause a logic error</td><td>3

AO2.1 (1)
AO2.2 (2)</td><td></td></tr>
<tr><td>2</td><td>b</td><td>i</td><td><code>data, nextData</code></td><td>2

AO2.1 (2)</td><td></td></tr>
<tr><td>2</td><td>b</td><td>ii</td><td>The actual data stored in the array will be changed</td><td>1

AO1.2 (1)</td><td></td></tr>
<tr><td>2</td><td>c</td><td>i</td><td>1 mark per set of swaps

<table>
<tr><td>10</td><td>95</td><td>5</td><td>33</td><td>100</td><td>77</td><td>45</td><td rowspan="3">1</td></tr>
<tr><td>10</td><td>5</td><td>95</td><td>33</td><td>100</td><td>77</td><td>45</td></tr>
<tr><td>10</td><td>5</td><td>33</td><td>95</td><td>100</td><td>77</td><td>45</td></tr>
<tr><td>10</td><td>5</td><td>33</td><td>95</td><td>77</td><td>100</td><td>45</td><td rowspan="2">1</td></tr>
<tr><td>10</td><td>5</td><td>33</td><td>95</td><td>77</td><td>45</td><td>100</td></tr>
<tr><td>5</td><td>10</td><td>33</td><td>95</td><td>77</td><td>45</td><td>100</td><td rowspan="3">1</td></tr>
<tr><td>5</td><td>10</td><td>33</td><td>77</td><td>95</td><td>45</td><td>100</td></tr>
<tr><td>5</td><td>10</td><td>33</td><td>77</td><td>45</td><td>95</td><td>100</td></tr>
<tr><td>5</td><td>10</td><td>33</td><td>45</td><td>77</td><td>95</td><td>100</td><td>1</td></tr>
</table>
</td><td>4

AO1.1 (1)
AO1.2 (1)
AO2.1 (2)</td><td>Pair swaps may be shown individually or at the end of each traversal (award mark point 1 and 2 if first traversal is shown on one line)</td></tr>
</table>

2	c	ii	1 mark per bullet max 8
			• Procedure declaration
			• …taking array as parameter by reference
			• Outer loop, looping the number of times there are elements…
			• Inner loop, looping through all elements of the loop (or one less element each iteration)
			• Comparing two elements
			• Swapping two elements if out of order
			• …using an appropriate temp memory space or equivalent
			• Efficiency: stopping as soon as no swaps have been made
			
			e.g.
			`procedure sortData(data[]:byRef)`
			
			`  swapped = true`
			`  while swapped == true`
			`   swapped = false`
			
			`   for innerCount = 0 to data.length - 2`
			`      if data[innerCount] > data[innerCount + 1] then`
			`         temp = data[innerCount]`
			`         data[innerCount] = data[innerCount + 1]`
			`         data[innerCount+1] = temp`
			`         swapped = true`
			`       end if`
			`    next innerCount`
			
			`  endwhile`
			
			`endprocedure`
				8
				
				AO1.1 (2)
				AO3.2 (6)	

<table>
<tr>
<td>2</td>
<td>c</td>
<td>iii</td>
<td>1 mark per row(s) as shown:

<table>
<tr><td>95</td><td>10</td><td>5</td><td>33</td><td>100</td><td>77</td><td>45</td><td></td></tr>
<tr><td>10</td><td>95</td><td>5</td><td>33</td><td>100</td><td>77</td><td>45</td><td>1</td></tr>
<tr><td>5</td><td>10</td><td>95</td><td>33</td><td>100</td><td>77</td><td>45</td><td>1</td></tr>
<tr><td>5</td><td>10</td><td>33</td><td>95</td><td>100</td><td>77</td><td>45</td><td rowspan="2">1</td></tr>
<tr><td>5</td><td>10</td><td>33</td><td>95</td><td>100</td><td>77</td><td>45</td></tr>
<tr><td>5</td><td>10</td><td>33</td><td>77</td><td>95</td><td>100</td><td>45</td><td rowspan="2">1</td></tr>
<tr><td>5</td><td>10</td><td>33</td><td>45</td><td>77</td><td>95</td><td>100</td></tr>
</table>
</td>
<td>4

AO1.2 (2)
AO2.1 (1)
AO2.2 (1)</td>
<td>In the 3rd mark point the line may appear just once</td>
</tr>
</table>

<table>
<tr>
<td>3</td><td>a</td><td></td>
<td>1 mark per bullet to max 2
• both are pointers
• front gives the (array) position/index of the first element in the queue
• end gives the (array) position/index of the last element in the queue</td>
<td>2

AO1.1 (1)
AO1.2 (1)</td>
<td></td>
</tr>
<tr>
<td>3</td><td>b</td><td>i</td>
<td>1 mark per bullet
• elements in the queue (correct four colours)
• …in the correct positions
• front points to the first element in queue
• end points to the last elements in queue

<table><tr><td></td><td></td><td></td><td>blue</td><td>grey</td><td>orange</td><td>maroon</td><td></td></tr></table>front = 3
end = 6</td>
<td>4

AO1.1 (1)
AO1.2 (1)
AO2.1 (1)
AO2.2 (1)</td>
<td>Allow FT for values for front/end</td>
</tr>
<tr>
<td>3</td><td>b</td><td>ii</td>
<td>Function has to return a value, procedure does not</td>
<td>1

AO1.1 (1)</td>
<td></td>
</tr>
<tr>
<td>3</td><td>b</td><td>iii</td>
<td>1 mark per bullet to max 4
• Check if the queue is full…
• … e.g. if the start and end pointers are equal // are at index 0 // description of checking circular queue
• …If the queue is full, return full / error message
• Otherwise, increment the end pointer…
• … and add the new item at the position indicated by the end pointer
• …return added/equivalent</td>
<td>4

AO1.1 (2)
AO2.1 (2)</td>
<td></td>
</tr>
</table>

4	a		1 mark per bullet to max 4
			• Function declaration with parameter passed
			• Opening a file to read
			• Reading the data from a file
			• Closing the file and then returning the string
			
			`function getText(filename)`
			`  file = openRead(filename)`
			`  dataString = file.readLine()`
			`  file.close()`
			`  return dataString`
			`endfunction`
				4
				
				AO3.2
				(4)	

4	B		1 mark per bullet to max 7 • Procedure declaration • Reading file name as input • Calling getText() function with filename • Looping through all characters in the data • Checking if there is a "." (ASCII code 46) • … checking if the next character is a space " " (ASCII code 32) • … looping until there is no " " (ASCII code 32) • … checking if the character is lowercase • … changing the character to uppercase • Writing output to text file **procedure fullStop()** **filename = input("Enter filename")** **textString = getText(filename)** **output = ""** **i = 0** **newSentence = True** **while i < textString.length - 1:** **nextChar = textString.substring(i, 1)** **if newSentence == True:** **if nextChar == " ":** **output = output + " "** **else:** **output = output + upper(nextChar)** **newSentence = False** **else:** **if nextChar == ".":** **newSentence = True** **output = output + char** **endif**	**7** **AO2.2 (1)** **AO3.2 (6)**	

			`    i = i + 1`
			
			`  endwhile`
			
			`  file = openWrite(filename)`
			`  file.write(output)`
			`  file.close()`
			`endprocedure`		

Assessment Objectives (AO) Grid

Question	AO1.1	AO1.2	AO2.1	AO2.2	AO3.1	AO3.2	AO3.3	Total
1ai	1							1
1aii		1						1
1aiii			2	2				4
1bi			1	1				2
1bii			1	1				2
1c	1	1	1	1				4
1d				2				2
1e*	2	2	2				3	9
2ai			1					1
2aii			1	2				3
2bi			2					2
2bii		1						1
2ci*m*	1	1	2					4
2cii	2					6		8
2ciii*m*		2	1	1				4
3a	1	1						2
3bi*m*	1	1	1	1				4
3bii	1							1
3biii	2	2						4
4a						4		4
4b*m*				1		6		7
Total	12	12	15	12	0	16	3	70

*** = extended response**
***m* = mathematical content**

OCR (Oxford Cambridge and RSA Examinations)
The Triangle Building
Shaftesbury Road
Cambridge
CB2 8EA

OCR Customer Contact Centre

Education and Learning
Telephone: 01223 553998
Facsimile: 01223 552627
Email: general.qualifications@ocr.org.uk

www.ocr.org.uk

For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored

Oxford Cambridge and RSA Examinations
is a Company Limited by Guarantee
Registered in England
Registered Office; The Triangle Building, Shaftesbury Road, Cambridge, CB2 8EA
Registered Company Number: 3484466
OCR is an exempt Charity

OCR (Oxford Cambridge and RSA Examinations)
Head office
Telephone: 01223 552552
Facsimile: 01223 552553

© OCR 2019