GCE

Computer Science

H046/02: Algorithms and problem solving

Advanced Subsidiary GCE

Mark Scheme for November 2020

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.

Annotations

Annotation	Meaning
∧	Omission mark
BOD	Benefit of the doubt
✖	Incorrect point
FT	Follow through
NAQ	Not answered question
NBOD	No benefit of doubt given
REP	Repeat
✔	Correct point
TV	Too vague
0	Zero (big)
BP	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.
L1	Level 1
L2	Level 2
L3	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 Examiners**. 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.

Question			Answer	Mark	Guidance
1	**(a)**	**(i)**	1 mark per bullet up to a maximum of 3 marks, e.g: • Classroom displays have been removed • People have been removed / simplified with symbols • Wall/carpet colours/details have been removed • Diagram is not to scale.	3 A02.1 (3)	Allow other suitable responses that are applied to the scenario in the question.
1	**(a)**	**(ii)**	1 mark per bullet up to a maximum of 2 marks for each benefit (4 marks maximum in total), e.g: • Reduced development time (1) as factors that can detract from the program can be ignored (1) • Program more likely to solve the problem (1) as unnecessary aspects will not detract from the main purpose of the program (1) • Reduces complexity of programming code (1) therefore can run on lower spec computers(1)	4 A01.2 (2) A02.1 (2)	
1	**(b)**	**(i)**	• Selection/branching	1 A03.3 (1)	
1	**(b)**	**(ii)**	1 mark per bullet up to a maximum of 3 marks, e.g: • The number of user attempts is not known • The code will need to continue until the password entered is correct • A while loop will keep repeating until the correct password has been input // condition is met • A for loop will only repeat a certain number of times • A for loop may continually ask for password even though it's been entered correctly	4 A02.1 (4)	
1	**(b)**	**(iii)**	1 mark per bullet up to a maximum of 3 marks, e.g: • Correct use of `do` at the start of the loop. • Correct use of `until` at the end of the loop. • Correct logic for inputting password, checking the entered password and for setting check to `true`/checking the password within the condition of the loop.	3 A03.2 (3)	Example Solution ```do enteredPassword=input("Enter Password") if enteredPassword == correctPassword then check = true endif until check == true``` Alternative solution

| | | | | | ```
do
    enteredPassword=input("Enter Password")
until enteredPassword==correctPassword
``` |
1	**(c)**	**(i)**	1 mark for identifying a feature and 1 mark for stating how it can be used up to a maximum of 2 marks for each IDE feature (6 marks maximum in total.) For example: • Autocomplete (1) which will predict variable / built-in functions (1) • Auto indent (1) to automatically indent code when selection / iterative statements are used (1) • Colour coding (1) to be able to distinguish between the different parts of each statement/line (1)	6 A01.1 (3) A01.2 (3)	Allow other suitable responses that are appropriate to **writing** programming code such as automatic syntax analysis, automatic cross-referencing, line numbers and code comments, automated refactoring, automated code generation (e.g. creating templates for common patterns).
1	**(c)**	**(ii)**	1 mark per bullet up to a maximum of 2 marks for each benefit(4 marks maximum in total), e.g: • Reduced development time (1) due to time boxing / each subtask being given a strict time limit (1) • Increased user involvement (1) so issues can be identified and fixed early / more likely to meet client requirements (1) • The requirements do not all need to be stated at the start (1) so therefore it is more flexible (1)	4 A01.1 (2) A01.2 (2)	Allow other suitable responses.
1	**(c)**	**(iii)**	1 mark per bullet up to a maximum of 2 marks, e.g: • Black box is when the internal structure/ design is not known (to the tester) • Black box requires limited/no programming knowledge • White box is when the internal structure/ design is known (to the tester) • White box requires programming knowledge	2 A01.2 (2)	Response must cover both black box and white box testing for 2 marks.

2 (a)	**Mark Band 3 – High level (7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of bubble sorts; 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 is able to weigh up the use of bubble sorts within the context which results in a supported and realistic judgment as to whether it is suitable to use within the context.*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 bubble sorts; 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 makes a reasonable attempt to come to a conclusion showing some recognition of influencing factors that would determine whether it is possible to use bubble sorts in this context. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence* **Mark Band 1 – Low Level (1-3 marks)** The candidate demonstrates a basic knowledge of bubble sorts with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides nothing more than an unsupported assertion.*The information is basic and comunicated 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.	9 AO1.1 (2) AO1.2 (2) A02.1 (2) A03.3 (3)	**Knowledge and Understanding** • All adjacent items are compared against each other. • The biggest number in the adjacent pair is swapped over with the smallest number. A temp variable is used to hold the data while it's being moved. • When a swap is made a flag is set. • This is repeated for all adjacent values, known as one pass. • At the end of one pass, the largest item should appear at the end of the list. • If at the end of the list the flag has been set the flag is unset and the algorithm starts from the beginning of the list again. • When the algorithm gets to the end of the list and the flag is unset the list is sorted. **Application** • As there are 250,000 items a bubble sort would perform very slowly as a lot of passes will need to be made in order to sort the items. • Bubble sorts are better suited to data sets where the items are almost/partly sorted. However the smaller numbers are currently towards the end and the larger numbers are towards the start. • This will therefore increase the amount of comparisons / passes/swaps required which will therefore slow the performance of the sort down. **Evaluation** • The algorithm is easy to implement as the number of lines of code is less than other standard sorting algorithms. • Although a bubble sort will be able to sort the items into order, it will take longer than other sorting algorithms due to the number of items and the current order or items in the unsorted list.	

2	(b)	(ii)	• 249,999	1 A02.2 (1)	
2	(c)	(iii)	1 mark per bullet up to a maximum of 4 marks: • The inner loop will compare all of the adjacent items…. • ….in a single pass • The outer loop will repeat the process of checking adjacent items…. • …until all passes are complete / the items are sorted/no swaps have been made in a pass	4 A01.2 (4)	Allow other valid interpretations e.g. conditional while loop … used to compare against swap flag after each pass; counter controlled for loop … used to check adjacent items on each pass
2	(d)	(iv)	• Insertion sort	1 A02.1 (1)	Allow other sorting algorithms not listed in the specification (e.g. Merge Sort, Quick Sort etc)
3	(a)	(i)	1 mark per bullet up to a maximum of 4 marks: • A queue is First In First Out (FIFO) • Therefore bookings will be executed in the order they have received • A stack is Last In First Out (LIFO) • Therefore the bookings would be executed from the most recent booking	4 A02.2 (4)	
3	(a)	(ii)	`custNumber`	1 A03.3 (1)	
3	(a)	(iii)	1 mark per bullet up to a maximum of 2 marks: • Correct logic for incrementing the tail pointer by 1 (e.g. `tail = tail +1`) • Correct logic for adding `custNumber` to the tail pointer (e.g. `queue[tail]=custNumber`)	2 A03.2 (2)	

3	(a)	(iv)	• If the tail is greater than 10 / `maxElements`	1
				A02.2
				(1)
					Accept:
					`not((tail + 1) > maxElements)`
					Or
					`(tail + 1) <= maxElements`
					Or equivalent

3	**(b)**	**(i)**	1 mark per bullet up to a maximum of 3 marks: • 5 (Jimmy) • 8 (Siad) • 9 (Tommy)	3 A02.1 (3)	
3	**(b)**	**(ii)**	1 mark per bullet up to a maximum of 2 marks, e.g: • Efficient • ...as does not need to search every single element/uses divide and conquer	2 A01.2 (2)	
3	**(b)**	**(iii)**	• Linear Search / Serial Search	1 A02.1 (1)	
3	**(b)**	**(iv)**	• The items are in alphabetical order / the items are sorted	1 A02.1 (1)	

4			1 mark per bullet up to a maximum of 9 marks:
			• Defining the `createUsername` procedure correctly
			• Suitable logic for inputting the first name
			• Suitable logic for inputting the surname
			• Suitable logic for using the first letter of the first name
			• Suitable logic for joining the different sections of the username together
			• Suitable logic to pass the username into the function `existingUsers` (eg as a parameter or global variable)
			• Suitable logic for continually increasing the number by 1 …..
			• …until the username is unique
			• Sensible use of variable names and indentation throughout
				9
				
				A03.1
				(3)
				
				A03.2
				(6)
					Example solution:
					
					``
					procedure createUsername()
					    firstname = input("Enter First Name")
					    surname = input("Enter Surname")
					    number = 0
					    while unique == false
					        number = number + 1
					        username = surname + firstname.substring(0,1) +
					str(number)
					        unique = existingUsers(username)
					    endwhile
					    print("Username is unique")
					endprocedure
					``
					
					There are many different ways that this procedure could have been achieved. Therefore other alternative methods should be given credit (candidates may use `substring` or `mid` to access first character of `firstname`).

5	**(a)**		• 30	1 A03.2	

					(1)	
5	**(b)**		•	10	1 A03.2 (1)	
5	**(c)**		•	10	1 A03.2 (1)	
5	**(d)**		•	To round the `number` to the nearest 10	1 A02.1 (1)	

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

Cambridge
Assessment