

GCE

Computer Science

H046/02: Algorithms and problem solving

AS Level

Mark Scheme for June 2025

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 2025

MARKING INSTRUCTIONS

PREPARATION FOR MARKING SCORIS

1. Make sure that you have accessed and completed the relevant training packages for on-screen marking: *RM Assessor Online Training: OCR Essential Guide to Marking*.
2. Make sure that you have read and understood the mark scheme and the question paper for this unit. These are available in RM Assessor
3. Log-in to RM Assessor and mark the **required number** of practice responses (“scripts”) and the **required number** of standardisation responses.

MARKING

1. Mark strictly to the mark scheme.
2. Marks awarded must relate directly to the marking criteria.
3. The schedule of dates is very important. It is essential that you meet the RM assessor 50% and 100% (traditional 40% Batch 1 and 100% Batch 2) deadlines. If you experience problems, you must contact your Team Leader (Supervisor) without delay.
4. If you are in any doubt about applying the mark scheme, consult your Team Leader by telephone or the RM assessor messaging system, or by email.
5. **Crossed Out Responses**
Where a candidate has crossed out a response and provided a clear alternative then the crossed out response is not marked. Where no alternative response has been provided, examiners may give candidates the benefit of the doubt and mark the crossed out response where legible.

Rubric Error Responses – Optional Questions

Where candidates have a choice of question across a whole paper or a whole section and have provided more answers than required, then all responses are marked and the highest mark allowable within the rubric is given. Enter a mark for each question answered into RM assessor, which will select the highest mark from those awarded. *(The underlying assumption is that the candidate has penalised themselves by attempting more questions than necessary in the time allowed.)*

Multiple Choice Question Responses

When a multiple choice question has only a single, correct response and a candidate provides two responses (even if one of these responses is correct), then no mark should be awarded (as it is not possible to determine which was the first response selected by the candidate).

When a question requires candidates to select more than one option/multiple options, then local marking arrangements need to ensure consistency of approach.

Contradictory Responses

When a candidate provides contradictory responses, then no mark should be awarded, even if one of the answers is correct.

Short Answer Questions (requiring only a list by way of a response, usually worth only **one mark per response**)

Where candidates are required to provide a set number of short answer responses then only the set number of responses should be marked. The response space should be marked from left to right on each line and then line by line until the required number of responses have been considered. The remaining responses should not then be marked. Examiners will have to apply judgement as to whether a 'second response' on a line is a development of the 'first response', rather than a separate, discrete response. *(The underlying assumption is that the candidate is attempting to hedge their bets and therefore getting undue benefit rather than engaging with the question and giving the most relevant/correct responses.)*

Short Answer Questions (requiring a more developed response, worth **two or more marks**)

If the candidates are required to provide a description of, say, three items or factors and four items or factors are provided, then mark on a similar basis – that is downwards (as it is unlikely in this situation that a candidate will provide more than one response in each section of the response space.)

Longer Answer Questions (requiring a developed response)

Where candidates have provided two (or more) responses to a medium or high tariff question which only required a single (developed) response and not crossed out the first response, then only the first response should be marked. Examiners will need to apply professional judgement as to whether the second (or a subsequent) response is a 'new start' or simply a poorly expressed continuation of the first response.

6. Always check the pages (and additional objects if present) at the end of the response in case any answers have been continued there. If the candidate has continued an answer there then add a tick to confirm that the work has been seen.
7. There is a NR (**No Response**) option. Award NR (No Response):
 - if there is nothing written at all in the answer space
 - OR if there is a comment which does not in any way relate to the question (e.g., 'can't do', 'don't know')
 - OR if there is a mark (e.g., a dash, a question mark) which is not an attempt at the question.

Note: Award 0 marks – for an attempt that earns no credit (including copying out the question).

8. The scoris **comments box** is used by your team leader to explain the marking of the practice responses. Please refer to these comments when checking your practice responses. **Do not use the comments box for any other reason.**
If you have any questions or comments for your team leader, use the phone, the scoris messaging system, or e-mail.

9. *Assistant Examiners will send a brief report on the performance of candidates to their Team Leader (Supervisor) via email by the end of the marking period. The report should contain notes on particular strengths displayed as well as common errors or weaknesses. Constructive criticism of the question paper/mark scheme is also appreciated.*
10. For answers marked by levels of response: Not applicable in F501
- To determine the level** – start at the highest level and work down until you reach the level that matches the answer
 - To determine the mark within the level**, consider the following:

Descriptor	Award mark
On the borderline of this level and the one below	At bottom of level
Just enough achievement on balance for this level	Above bottom and either below middle or at middle of level (depending on number of marks available)
Meets the criteria but with some slight inconsistency	Above middle and either below top of level or at middle of level (depending on number of marks available)
Consistently meets the criteria for this level	At top of level

11. Annotations

Annotation	Meaning
	Omission mark
	Benefit of the doubt
	Incorrect point
	Expansion of a point
	Follow through
	Not answered question
	No benefit of doubt given
	Point being made
	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

Question			Answer	Mark	Guidance
1	a		1 mark for each input to max 2. 1 mark for each output to max 2. Input • Membership ID • Date • Time Output • Error message • Confirmation message • Bleeping sound	4	
1	b		1 mark each to max 3 e.g. • Code can be written once and used many times // reused in other programs // saves programming time • Reduces the file size of the program • Reduces the time taken testing • Makes the code more efficient • Makes the program easier to maintain • Program can be divided between different programmers • ...to allow the solution to be developed faster • ...to allow programmers to work to their strengths	3	Expansion point 'allow the solution to be developed faster' is valid for MP1, MP3 and MP6

2	a		1 mark each to max 2 e.g. • The identifier/name is <code>p1Items</code> • It has 10 elements by 2 elements • It has 10 rows • It has 2 columns • It has a total of 20 elements • It stores data of type string	2	Responses must be specific to <code>p1Items[10, 2]</code> Do not accept generic answers such as the array is 0 indexed (which is also in the question)
2	b	i	It is a string/text	1	
2	b	ii	1 mark each to max 2: • Correct use of for loop to iterate 10 times • Correct indexing of each array declaration that is consistent with loop index variable Example: <pre>for counter = 0 to 9 p1Items[counter, 0] = "empty" p1Items[counter, 1] = "empty" next counter</pre>	2	Arrays may be rewritten e.g. <code>[counter][0]</code> Allow valid Python loop ranges: <pre>for counter in range (10): p1Items[counter, 0] = "empty" p1Items[counter, 1] = "empty"</pre> It should be clear that the counter will iterate through the values 0 to 9 inclusive. Do not award if <i>counter = counter + 1</i> code line is left inside the loop because it will not then iterate every index.
2	b	iii	1 mark from: • The updated array is saved when the procedure ends • The procedure can access the array and change it. • It is a procedure so cannot return the updated array	1	

2	c	1 mark each to max 6 • Suitable use of a loop to iterate through array • Comparing each name [counter, 0] in all 10 array elements to "empty" ... // while the item is not empty • ...outputting the name [counter, 0] if not "empty" • Suitable logic to count how many items are in the array // set flag... • Checking if the total is 0 // checking flag is set/not set... • ...outputting suitable message when there are no items in the array e.g. <pre>total = 0 for x = 0 to 9 if p1Items[x,0] != "empty" then print(p1Items[x,0]) total = total + 1 endif next x if total == 0 then print("There are no items") endif</pre>	6	Answer is example only. Arrays may be written e.g. [x][0] Allow coded examples e.g. <pre>itemsFound = False for i in range(10): if p1Items[i][0] != "Empty": print(p1Items[i][0]) itemsFound = True if itemsFound == False: print("No items found")</pre> Responses may assume items are added in sequence from the first index and that as soon as an "empty" value is found there are no more items. Alternative solutions that map to the mark points given are therefore possible e.g. <pre>count = 0 if p1Items[0,1] = "empty" then print("No items") else while p1Items[count,0] != "empty" print(p1Items[count,0]) count = count + 1</pre>
---	---	--	---	---

3

1 mark each to max 6:

- First storing 1 in *x*, 2 in *item* and 0 in *y*
- Swapping 10 and 2, *y* set to -1 and *x* incremented to 2
- Storing 5 in *item* and 1 in *y*
- Swapping 10 and 5, changing *y* to 0
- 3 in *x*, 20 in *item*, 2 in *y* (20 present/not repeated)
- 4 in *X* with swaps etc. and nothing superfluous after

			data				
x	item	y	0	1	2	3	4
			10	2	5	20	19
1	2	0		10			
2		-1	2				
	5	1					
					10		
		0		5			
3	20	2				20	
4	19	3					20
		2				19	

6

Ignore spacing between values in columns provided additional values are not included.

Shading shows marks.

If an error occurs, or additional statement/value included, do not award mark where this first occurs and then continue as mark scheme.

4		Mark Band 3–High Level (7-9 marks) The candidate demonstrates thorough knowledge and understanding of the agile methodology and waterfall lifecycle; 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 the agile methodology and waterfall lifecycle; 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.	9	AO1: Knowledge and Understanding e.g. Agile programming • Programming is the priority • Still requires planning, design, code and test • Testing focus • Focus on end user involvement and testing • Create time-box that is given to end user/customer for feedback Waterfall • Has structured stages which are followed in a linear fashion • Move back through processes if change required AO2.1: Application e.g. Agile • Students can start programming faster - more engaging, can see results faster • May be harder to divide a system up between them • Can test features on the younger children throughout to make sure the end product is useable/friendly • Requires team to work closely together and communicate. • More costly for small projects • Could go off-track due to not planning everything at the start. • Good for large projects and division of tasks. Waterfall • More systematic and structured • Can design whole project at once • Students may find it harder to identify all requirements//harder to design entire system • Harder to make changes to the requirements/design once started • Developers more likely to work independently and have their own responsibilities and then combine at a later stage. • Focus on documentation first before starting production. • Development time is drastically increased
---	--	--	---	---

		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 the agile methodology and waterfall lifecycle; 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.		There is limited involvement with the client so they will be unable to try trial the program with the children AO3.3: Evaluation Accept any recommendation if justification matches. This project has a small number of developers, is a small project, low risk. Developers may not have all the skills to plan entire project at start, but may need structure due to inexperience, therefore: Waterfall allows for whole project to be designed and divided at the start, may be easier for the students to refer back to plan and structured processes. Formal requirements/plans will help students to write individual components that will then work together. Students will know their role at the start, and this is unlikely to change, students can work independently. Agile will allow students to get a working prototype earlier that they can start evaluated. Easier for new programmers to visualise and develop the project in steps instead of having to design the whole system. Might be harder to divide the project into parts when the whole system has not been planned. Fewer late changes needed due to incompatibility with end users due to testing with them throughout. Students will have to work as a team more than individually and needs good communication. Cost does not impact this.
4	b	1 mark from e.g. Black-box: tester does not know the code/algorithms White-box: tester knows the code/algorithms Black-box: tests inputs against outputs White-box: tests each path through the algorithm	1	Both sides required for mark to be awarded

5	a		1 mark each to max 2: • Input • Print • Int • Str • Substring	2	Do not accept .length as this is an attribute and not a method Allow data.substring
5	b		1 mark each to max 2: • 01 – Iteration // Looping • 08 - Branching // Selection	2	Do not accept if statement, for loop etc. Type of construct not example of construct required.
5	c		1 mark for: • To change the data type of first to an integer • To cast the data type of first as an integer	1	

5	d	i	1 mark for each completed line to max 5 • value • < • true • +1 • return false <pre>function linearSearch(array, value) counter = 0 while counter < array.length if array[counter] == value then return true endif counter = counter + 1 endwhile return false endfunction</pre>	5	Do not accept <= because it is clear from the code given that it is zero-indexed
5	d	ii	1 mark each to max 4 • Calculate middle value • Compare search value to the middle value • If value is smaller, discard upper half of array// change the upper bound to the middle value -1 • If value is larger discard lower half of the array// change the lower bound to the middle value +1 • Repeat until either value is found or no values left to compare// lower bound>upper bound	4	Allow description through code/pseudocode

6	a		1 mark each to max 2 e.g. • Queue is FIFO • Stack is LIFO • Queue inserts data and removes data at different ends • Stack inserts and removes at the same end • Queues need two pointer values • Stacks (usually) need one pointer value • Queue can be circular • Stack can only be linear	2	Must be full comparison for 2 marks.
6	b	i	1 mark for: • Points to the index of the last value put into the stack // points to the item at the top of the stack • Points to the next item that will be output from the stack	1	Any equivalent description
6	b	ii	1 mark each to max 3: e.g. • Check if stack is empty // check if <code>TopPointer == -1/null</code> • If not, then take/remove/pop data at index <code>TopPointer</code> • Decrement <code>TopPointer</code>	3	Allow decrement <code>topPointer</code> followed by <code>return stack[topPointer+1]</code>

6	b	iii	1 mark for each answer <pre>function push(dataItem) if topPointer == 7 then return false else topPointer = topPointer + 1 stack[topPointer] = dataItem return true endif endfunction</pre>	3	Allow = instead of == Accept >= instead of ==
---	---	-----	---	---	--

7		1 mark each to max 8 • Take date and year as input • Casting of input year and extracted year as int • Accessing the year in date of birth • ...subtracting from year input • Accessing the month in date of birth • ...calling <code>getMonth()</code> with this value (or recreating) • ...storing/using return value • Accessing date in date of birth • Outputting age, day, month and year in message • ...that matches example format with suitable concatenation. e.g. <pre>date = input("Enter your date of birth") year = int(input("Enter the year")) yearInput = int(date.substring(6, 4)) age = year - yearInput monthInput = date.substring(3, 2) month = getMonth(monthInput) day = date.substring(0, 2) print("You will be " + age + " on the " + day + " " + month + " " + year)</pre>	8	String can be 0- or 1-indexed. Output message can have/not have spaces Allow variants of substring as long as used consistently (e.g. <code>word.substring(startPos, endPos exclusive)</code>) Or the Pythonic <code>word[startPos : endPos exclusive]</code> Additional guidance: Only penalise type casting in BP2. FT thereafter in all calculations. Candidates are not provided with 'syntax' for substring, so allow different interpretations if applied consistently e.g. <code>.substring(2,4)</code> could mean start at index 2 / letter 2 and mean take continue to index 4 / letter 4 or take 4 letters from the given position. If candidates have interpreted as DDMMYYYY allow this instead of DD/MM/YYYY i.e. do not penalise for not dealing with slashes in the formatting. It is not clear if <code>getMonth()</code> requires an integer or a string parameter. Accept either. The month argument must have been extracted from the input string.
---	--	---	---	--

8		1 mark for first pass resulting in: • sun friend orange night time earth 1 mark for second pass resulting in: • sun orange night time friend earth 1 mark for 3rd, pass resulting in: • 3rd sun orange time night friend earth 1 mark for 4th and 5th passes resulting in: • 4th sun time orange night friend earth • 5th time sun orange night friend earth	4	If Ascending order, mark as FT and max 3: Pass 1: F E O N S T Pass 2: E F N O S T Pass 3: E F N O S T If ascending, third mark is for showing pass 3 has no swaps and therefore the list is sorted.
---	--	--	---	---

Need to get in touch?

If you ever have any questions about OCR qualifications or services (including administration, logistics and teaching) please feel free to get in touch with our customer support centre.

Call us on

01223 553998

Alternatively, you can email us on

support@ocr.org.uk

For more information visit



ocr.org.uk/qualifications/resource-finder



ocr.org.uk



Twitter/ocrextams



/ocrextams



/company/ocr



/ocrextams



CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT

OCR is part of Cambridge University Press & Assessment, a department of the University of Cambridge.

For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored. © OCR 2025 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 operates academic and vocational qualifications regulated by Ofqual, Qualifications Wales and CCEA as listed in their qualifications registers including A Levels, GCSEs, Cambridge Technicals and Cambridge Nationals.

OCR provides resources to help you deliver our qualifications. These resources do not represent any particular teaching method we expect you to use. We update our resources regularly and aim to make sure content is accurate but please check the OCR website so that you have the most up-to-date version. OCR cannot be held responsible for any errors or omissions in these resources.

Though we make every effort to check our resources, there may be contradictions between published support and the specification, so it is important that you always use information in the latest specification. We indicate any specification changes within the document itself, change the version number and provide a summary of the changes. If you do notice a discrepancy between the specification and a resource, please [contact us](#).

Whether you already offer OCR qualifications, are new to OCR or are thinking about switching, you can request more information using our [Expression of Interest form](#).

Please [get in touch](#) if you want to discuss the accessibility of resources we offer to support you in delivering our qualifications.