Oxford Cambridge and RSA

GCE

Computer Science

H446/02: Algorithms and programming

A Level

Mark Scheme for June 2022

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.

MARKING INSTRUCTIONS

PREPARATION FOR MARKING
RM Assessor

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 posted on the RM Cambridge Assessment Support Portal http://www.rm.com/support/ca

3. Log-in to RM Assessor and mark the **required number** of practice responses ("scripts") and the **required number** of standardisation responses.

YOU MUST MARK 5 PRACTICE AND 10 STANDARDISATION RESPONSES BEFORE YOU CAN BE APPROVED TO MARK LIVE SCRIPTS.

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 50% and 100% 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, email or via the RM messaging system.

5. Work crossed out:

a. where a candidate crosses out an answer and provides an alternative response, the crossed out response is not marked and gains no marks

b. if a candidate crosses out an answer to a whole question and makes no second attempt, and if the inclusion of the answer does not cause a rubric infringement, the assessor should attempt to mark the crossed out answer and award marks appropriately.

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 isn't an attempt at the question.

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

8. The RM **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 telephone, email or the RM messaging system.

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:
a. **To determine the level** – start at the highest level and work down until you reach the level that matches the answer
b. **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
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

12. **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.

	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	Marks	Guidance
1ai	The array/data must be in order/sorted	**1**	
1aii	1 mark per bullet • Compare the search item with the first value • ….then compare the search item with the next value • ….repeat the above process until either • ….the end of the array has been reached or • ….the search item is found and then stop • ….then return the array position // return -1 / False if not found	**4**	Not all mark points are dependent, but points awarded must following logically in sequence.
1bi	1 mark for each variable • `contents` • `count` • `numberOfWords` • `words / words[]`	**2**	Accept exact spelling only Do not award numberOfWords if there are obvious spaces in 'number of Words'. It must be a valid identifier.
1bii	1 mark per bullet • By reference the function receives the memory location of the data • By value the function receives a copy of the variable • By reference will make changes to the original variable • By value will make changes to the copy of the variable • By reference will overwrite data in the original variable • By value will not overwrite the data in the original variable • By reference will keep the changes after the function ends • By value will not keep the changes after the function ends	**2**	Must cover byVal and byRef for 2 marks to be awarded. Must be clear that byVal <u>is a copy</u> of the original value.

<table>
<tr>
<td>1biii</td>
<td>1 mark per bullet
• initialising a counter
• looping between 0 and numberOfWords -1
• incrementing counter inside loop
• remainder of algorithm correct (initialisation, concatenation and return)

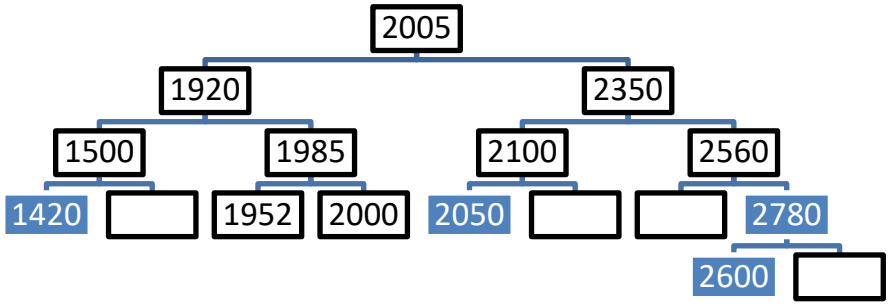
e.g.
<pre>contents = ""
count = 0
while count < numberOfWords
  contents = contents + words[count] + " "
  count = count + 1
endwhile
return contents</pre></td>
<td>4</td>
<td>Accept:
while count <= numberOfWords - 1

Accept other combinations for example counting from 1 and then subtracting 1 for the array element (but do not credit off by one errors)

Accept:
len(words) for numberOfWords</td>
</tr>
<tr>
<td>1c</td>
<td>1 mark for benefit, 1 mark for drawback
e.g.
Benefits:
• Variable doesn’t need passing as a parameter (byref)
• You don’t need to return a value
• Can be accessed from any function / anywhere in the program

Drawback:
• Increases memory usage (as it is used until full program execution is over)
• Alterations within the function may have unwanted side effects elsewhere in the program.</td>
<td>2</td>
<td></td>
</tr>
</table>

1d	1 mark per identification 1 mark for expansion, max 2 each. Write: e.g. • Auto-complete • Start typing an identifier/command and it fills in the rest • Auto-indent • Indents code automatically within structures to avoid errors • Coloured command words // pretty printing // syntax highlighting • Shows which commands are correct // help identify key elements Test e.g. • Breakpoints • Stop the program running at a set point to check variables • Variable watch window • Display the values of the variables while the program is run • Stepping • Run one line at a time and check variables Unit Testing • Automated tests to be run to check changes ensure changes haven't introduced errors.	**4**	
1e	1 mark per bullet to max 2 • Saves time from having to write the same algorithm repeatedly • Reduced testing requirements • Can be taken and used in different programs as well as the program they are written in // can be used in a program library	**2**	Allow other suitable answers

2a	2005	**1**	
2b	1 mark for each to max 2 • 1500 • 1952 • 2000 • 2100 • 2560	**2**	
2c	1 mark for each in the correct place • 1420 • 2050 • 2780 • 2600 	**4**	

2d	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of search traversals; 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. *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 search traversals; 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.* **Mark Band 1 – Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of search traversals 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 a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *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)** **AO2.1** **(2)** **AO3.3** **(3)**	**AO1: Knowledge and Understanding** **Indicative content** • Breadth first takes first value then visits all nodes connected to it. It then takes all nodes connected to those nodes. • Depth first goes to the left node, this becomes a new tree. It continues going to the left until a leaf. It then returns this, then goes right and repeats from the start. Follow left, follow right, take root. **AO2: Application** • Breadth will return 2005 1920 2350 1500 1985 2100 2560 (1420) 1952 2000 (2050) (2780) (2600) • Post-order / Depth will return (1420) 1500 1952 2000 1985 1920 (2050) 2100 (2600) (2780) 2560 2350 2005 **AO3: Evaluation** **Evaluations may vary and include one or more of the following points:** • Breadth is more efficient when the data searched for is closer to the root. • Depth is more efficient when data to be search for is further down. • Depth memory requirement is linear • Depth can be written recursively to aid understanding. • Breadth in general is better time complexity • In large trees depth may never return a value Candidates are not expected to know the complexities for the search traversals, however credit should be awarded if candidates choose to include these. Limit to band 2 if there is no evaluation of BFS/DFS

<table>
<tr><td>3a</td><td>1 mark for:
• Can refer to all 50 only using one identifier // all values can be indexed in one array
• The numbers can be passed as a single parameter
• Does not need 50 variables to be declared/passed</td><td>1</td><td></td></tr>
<tr><td>3b</td><td>1 mark for each completed statement

arrayLength = 50 // numberArray.length

tempValue =0

do

  flag = false

  for y = 0 to arrayLength - 2

    if numberArray[y] > numberArray[y + 1] then

      tempValue = numberArray[y]

      numberArray[y] = numberArray[y + 1]

      numberArray[y + 1] = tempValue

      flag = true

    endif

  next y

until flag == false</td><td>5</td><td>Note if numberArray - 1 // 49 used, then for loop for y will need to be 0 to arrayLength - 1

Allow other suitable valid identifier in place of tempValue e.g. temp</td></tr>
</table>

3c	1 mark for each stage shown • Splitting into individual items 2 12 1 9 3 5 15 7 • Combining in pairs 2 12 1 9 3 5 7 15 • Merge pairs 1 2 9 12 3 5 7 15 Merge for final 1 2 3 5 7 9 12 15	**4**	Do not award a mark for the final stage, unless candidate has shown the previous sorting stage(s).

3d	**Mark Band 3 – High level** **(9-12 marks)** The candidate demonstrates a thorough knowledge and understanding of sorting algorithms; 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. *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** **(5-8 marks)** The candidate demonstrates reasonable knowledge and understanding of sorting algorithms; 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.* **Mark Band 1 – Low Level** **(1-4 marks)** The candidate demonstrates a basic knowledge of sorting algorithms 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 a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *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.	**12** **AO1.1** **(3)** **AO1.2** **(3)** **AO2.1** **(3)** **AO3.3** **(3)**	**AO1: Knowledge and Understanding** **Indicative content** • Merge sort splits data into individual lists and merges • Insertion makes first value sorted list, then inserts each item into the sorted list • Bubble sort looks through each item in turn, number of items times **AO2: Application** • Merge uses more memory as new lists are needed. Insertion and Bubble need constant memory. • Bubble and Insertion have the best best-times, both O(n) because they run through the data once. merge sort requires a minimum number of stages so best case is longer (O(n log(n)) • Merge average is the same as best. Insertion and Bubble has average o(n^2). • Worst time merge has same has best and average because same number of stages are needed. Bubble sort and insertion all have worse O(n^2) **AO3: Evaluation** • There are a small number of elements (10) therefore a bubble sort of insertion would be better space wise because no further space is needed. • Merge would not need excessive amounts of more memory as there are only a small number of elements. • Time complexity, there is a small number of elements therefore Bubble and Insertion may be preferable. Differences are unlikely to be significant, so either would be more appropriate.

<table>
<tr><td>4ai</td><td>Max 1 mark for each definition

e.g.<table><tr><th>Term</th><th>Definition</th></tr><tr><td>Abstraction</td><td>Removal of unnecessary components // focus on only necessary components</td></tr><tr><td>Decomposition</td><td>Breaking down a problem into subproblems</td></tr></table></td><td>2</td><td></td></tr>
<tr><td>4aii</td><td>1 mark for each
e.g.
• Removal of visual elements such as buildings on the ground
• Simplification of controls
• Focus on important elements such as weather, height, speed</td><td>3</td><td></td></tr>
<tr><td>4aiii</td><td>1 mark for each to max 2
e.g.
• Reduce memory requirements
• Reduce processing requirements
• Simplify the problem being solved</td><td>2</td><td></td></tr>
<tr><td>4b</td><td>1 mark per bullet
e.g.
• Store data that has been used in cache/RAM in case needed again
• e.g. store design of the weather/a cloud/external environment</td><td>2</td><td>Allow 2 valid examples for 2 marks</td></tr>
<tr><td>5ai</td><td>1 mark
e.g.
In directed arcs/edges may only go in 1 direction // in undirected arcs/edges can go in both directions</td><td>1</td><td></td></tr>
<tr><td>5aii</td><td>1 mark
• More than one path is allowed in a graph
• Graphs do not have a root node
• Graphs can be weighted
• Graphs can have loops/cycles</td><td>1</td><td>Allow answers where candidates have given the reverse. e.g. a tree does not have loops.</td></tr>
</table>

<table>
<tr><td>5b</td><td>1 mark e.g.
• Symbols are used to represent the address
• The edges represent possible connections between addresses not the actual physical routes</td><td>1</td><td>Allow other suitable answers that are in context of the problem</td></tr>
<tr><td>5ci</td><td>
<table>
<tr><th>Node</th><th>Distance travelled</th><th>Previous node</th><th>Marking Guidance</th></tr>
<tr><td>A</td><td>0 / -</td><td>N/A / -</td><td>1 Mark</td></tr>
<tr><td>B</td><td>3</td><td>A</td><td rowspan="2">1 Mark</td></tr>
<tr><td>C</td><td>13</td><td>E</td></tr>
<tr><td>D</td><td>10</td><td>B</td><td rowspan="2">1 Mark</td></tr>
<tr><td>E</td><td>6</td><td>B</td></tr>
<tr><td>F</td><td>9</td><td>E</td><td rowspan="2">1 Mark</td></tr>
<tr><td>G</td><td>16</td><td>F</td></tr>
<tr><td>H</td><td><s>21</s> 19</td><td><s>D</s> G</td><td>1 Mark</td></tr>
</table>
Final Path = A,B,E,F,G,H, Distance = 19 (1 Mark)</td><td>6</td><td>Order of previous nodes visited must be clear

Note that nodes in the table do not have to be given in alphabetical order by candidates</td></tr>
<tr><td>5cii</td><td>1 mark per bullet
Similarity:
• Both always find the shortest route
• Both are pathfinding algorithms

Differences:
• A* is (usually) more efficient // dijkstra’s is (usually) slower
• A* uses heuristics to find a solution faster // Dijkstra’s does not use heuristics</td><td>2</td><td>Must contain a similarity and a difference for both marks.</td></tr>
</table>

5di	1 mark per bullet to max 1 • Simulate/model behaviour of the system (before it is) used under load • Because it would be too expensive/unsafe/time critical to test the real system	**1**	
5dii	1 mark per bullet to max 2 e.g. • Test with large and small values • e.g. largest number of deliveries • e.g. largest number of possible routes • Model how well the system scales with increasing use.	**2**	
6a	1 mark for each component e.g. • Allocating cards to each player • Generating the deck • Managing whose turn it is to play • Checking won	**3**	Accept any reasonable component
6bi	It returns a value	**1**	
6bii	1 mark per bullet • If the players card is the same suit return true • if the players card is the same number return true • if neither is true, return false	**3**	Allow 1/0, 'T'/'F', "Yes"/"No" or any sensible alternative as return values.
6c	1 mark per bullet • identifier `cards`… • …with 2 dimensions	**2**	
7ai	Line number `5`	**1**	
7aii	1 mark per feature • A function that calls itself // a function that is defined in terms of itself • …has a base case (that terminates the recursion)	**2**	

<table>
<tr>
<td>7b</td>
<td>
<table>
<tr><th>Function call</th><th>number</th><th>return</th><th>Marking Guidance</th></tr>
<tr><td>calculate(5)</td><td>5</td><td>15</td><td>1 Mark</td></tr>
<tr><td>calculate(4)</td><td>4</td><td>10</td><td>1 Mark</td></tr>
<tr><td>calculate(3)</td><td>3</td><td>6</td><td>1 Mark</td></tr>
<tr><td>calculate(2)</td><td>2</td><td>3</td><td>1 Mark</td></tr>
<tr><td>calculate(1)</td><td>1</td><td>1</td><td>1 Mark</td></tr>
</table>
</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>7c</td>
<td>calculate(10)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8ai</td>
<td>1 mark for correct data
1 mark for correc top of stack pointer
<table>
<tr><td>pointerValue</td><td>6</td></tr>
</table>
<table>
<tr><th>Index</th><th>Data</th></tr>
<tr><td>8</td><td></td></tr>
<tr><td>7</td><td></td></tr>
<tr><td>6</td><td></td></tr>
<tr><td>5</td><td>7</td></tr>
<tr><td>4</td><td>6</td></tr>
<tr><td>3</td><td>3</td></tr>
<tr><td>2</td><td>6</td></tr>
<tr><td>1</td><td>5</td></tr>
<tr><td>0</td><td>10</td></tr>
</table>
</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>8aii</td>
<td>1 mark per bullet
• Point to the next free space in the array
• Points to the top of the stack</td>
<td>1</td>
<td></td>
</tr>
</table>

<table>
<tr>
<td>8bi</td>
<td>1 mark per correctly completed statement

e.g.
<pre>pubic function pop()
  if pointerValue == 0 then
    return -1
  else
    pointerValue = pointerValue -1
    returnValue = stackArray[pointerValue]
    return returnValue
  endif
endfunction</pre></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>8bii</td>
<td>1 mark per bullet to max 6

function header
..taking parameter (ignore byval/byref)
checking if stack is full (pointerValue at 100)…
…and returning false
(otherwise) adding value to top of stack
…incrementing top of stack pointer
return true

e.g.
<pre>function push(value)
  if pointerValue < 100 then
    stackArray[pointerValue] = value
    pointerValue = pointerValue + 1
    return true
  else
     return false
  endif
endfunction</pre></td>
<td>6</td>
<td>Ignore additional parameters in function definition

Do not accept the return of string values

FT following a reasonable attempt to check if the stack is full</td>
</tr>
</table>

8biii	1 mark per bullet • instantiation of new object of type stack • assigned to variable mathsStack `mathsStack = new stack()`	**2**	**Accept** `mathsStack = stack()` allow missing brackets this time only e.g. mathsStack = stack

8biv	1 mark for each completed statement
	
	`returnValue = true`
	`while returnValue == true`
	`  returnValue = mathsStack.push(input("Enter Number"))`
	`  if returnValue == false then`
	`  print("Stack full")`
	`  endif`
		4	Accept equivalent for print e.g. output

<table>
<tr>
<td>8bv</td>
<td>1 mark per bullet to max 8
• initialise a total to 0 outside of loop
• looping
• removing an item from the stack using the method pop
• check if stack is empty
• (if not) add value returned to total
• …outputting total
• counting how many values are returned
• stopping loop when either 20 items removed or no items left

<pre>
total = 0
quantity = 0
returnValue = 0

while quantity<20 and retunValue!=-1
  returnValue = mathsStack.pop()
  if(returnValue != -1) then
    quantity = quantity + 1
    total = total + returnValue
    print(total)
  endif
endwhile
</pre>
</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>8ci</td>
<td>1 mark per bullet to max
• Queue has head pointer and tail pointer
• When an item is enqueued the tail pointer increments
• When an item is dequeued the head pointer increments</td>
<td>3</td>
<td>Max 1 mark for Enqueue/Dequeue operations if description of effect on tail/head pointers not given</td>
</tr>
</table>

8cii	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of object-oriented and procedural programming; 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. *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 object-oriented and procedural programming; 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.* **Mark Band 1 – Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of object-oriented and procedural programming 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 a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *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)** **AO2.1** **(2)** **AO3.3** **(3)**	**AO1: Knowledge and Understanding** **Indicative content** • OOP defines an object as an independent entity • OOP defines the attributes of the object and the methods that can be applied to it • attributes could be private to restrict accidental changes • Procedural the statements are executed in the order they are written **AO2: Application** • OOP allows for an object to be created from the queue • Many instances of this queue can then be declared in the main program. • Procedural will need each queue to be declared individually • Procedural will need to make use of subroutines where the queue will need to be sent and returned each time. **AO3: Evaluation** • OOP you can create multiple instances of the queue as required by the program without having to re-write all of the declarations etc. In procedural you would have to write separate code for each new stack • OOP reduces amount of code needed therefore fewer errors are likely as code is written once and then used multiple times • OOP can reduce mistakes because the subroutines are self-contained in procedural it would need to make sure the correct values are passed and returned, or global variables may be required which uses excess memory.