OCR
Oxford Cambridge and RSA

Practice Paper 2

GCE Computer Science

H446/02 Algorithms and Programming

Duration 2 hours 30 minutes

MAXIMUM MARK 140

FINAL

This document consists of 27 pages

MARKING INSTRUCTIONS

PREPARATION FOR MARKING

SCORIS

1. Make sure that you have accessed and completed the relevant training packages for on-screen marking: *scoris 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 scoris and mark the **required number** of practice responses ("scripts") and the **required number** of standardisation responses.

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

TRADITIONAL

Before the Standardisation meeting you must mark at least 10 scripts from several centres. For this preliminary marking you should use **pencil** and follow the **mark scheme**. Bring these **marked scripts** to the meeting.

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 scoris 50% and 100% (traditional 50% 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, email or via the scoris 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 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 telephone, email or the scoris 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
C	Subordinate clause / consequential error
	Incorrect point
E	Expansion of a point
FT	Follow through
NAQ	Not answered question
NBOD	No benefit of doubt given
P	Point being made
REP	Repeat
	Correct point
TV	Too vague
	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:

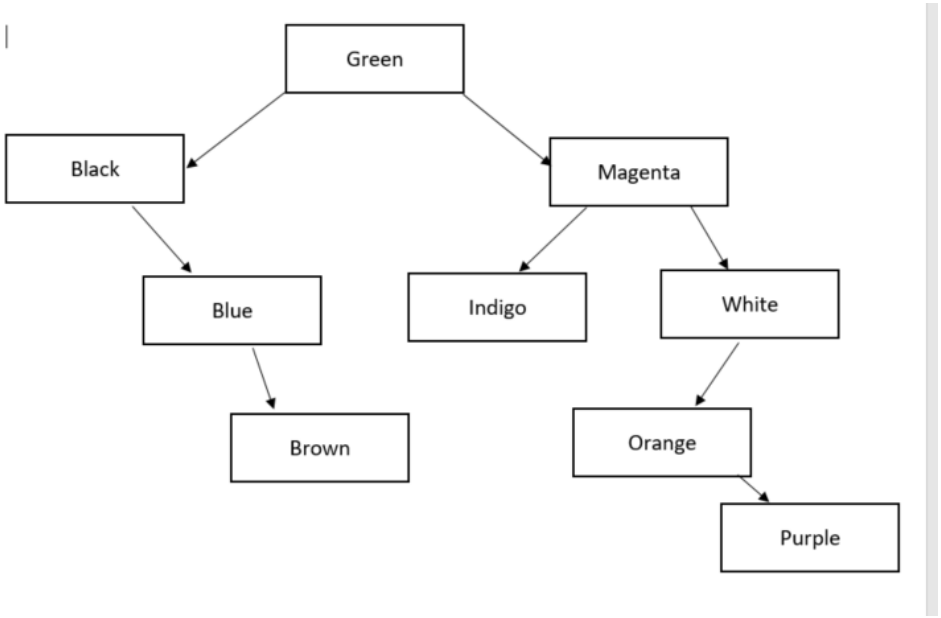
The indicative content indicates the expected parameters for candidates' answers, but be prepared to recognise and credit unexpected approaches where they show relevance.
Using 'best-fit', decide first which set of BAND DESCRIPTORS best describes the overall quality of the answer. Once the band is located, 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
1	a	i	1 mark per bullet to max 3 • A data structure • Consists of nodes • That have sub nodes (children) • First node is called the root • Lines that join nodes are called branches	**3** **AO1.1** **(3)**	
1	a	ii	1 mark per bullet to Max 1 • In a binary search tree, each node only has max. 2 sub nodes • If a child node is less than a parent node, it goes to the left of the parent. • If a child node is greater than a parent node, it goes to the right of the parent.	**1** **AO1.2** **(1)**	
1	b		1 mark for each node as a correct sub-node  	**4** **AO2.1** **(4)**	Allow follow through e.g. if white is incorrect, but orange follows through in a logical position.

Question			Answer	Marks	Guidance
1	c	i	2 marks for correct order • 5, 31, 20, 48, 45, 60 • 92, 88, 98, 76, 50 1 mark per bullet to max 3 for explanation • Visit all nodes to the left of the root node • Visit right • Visit root node • Repeat three points for each node visited	**5** **AO1.1 (1)** **AO1.2 (2)** **AO2.1 (2)**	
1	c	ii	2 marks for correct order • 50, 45, 76 • 20, 48, 60, 98, 5, 31, 88, 92 1 mark per bullet to max 3 for explanation • Visit root node • Visit all direct subnodes (children) • Visit all subnodes of first subnode • Repeat three points for each subnode visited	**5** **AO1.1 (1)** **AO1.2 (2)** **AO2.1 (2)**	
1	d	i	1 mark for the correct pointers for nodes 1, 2 and 3	**3** **AO2.1 (3)**	Ignore any change to FP, additional nodes and Right Pointers to node 4 and 5

Question	Answer	Marks	Guidance

Array Index	Left Pointer	Data	Right Pointer
0	1	68	2
1	3	30	
2	6	73	5
3	4	22	
4		1	
5		90	
6		70	

<table>
<tr><td>1</td><td>d</td><td>ii</td><td>To identify where the next element will be placed</td><td>1
AO1.2 (1)</td><td></td></tr>
<tr><td>1</td><td>d</td><td>iii</td><td>1 mark per bullet
• Adding data 6 and 100
• …In the correct order
• Updating the Right Pointers of nodes 4 and 5
• Updating the Free Pointer to 9
FP: 9
<table><tr><th>Array Index</th><th>Left Pointer</th><th>Data</th><th>Right Pointer</th></tr><tr><td>0</td><td>1</td><td>68</td><td>2</td></tr><tr><td>1</td><td>3</td><td>30</td><td></td></tr><tr><td>2</td><td>6</td><td>73</td><td>5</td></tr><tr><td>3</td><td>4</td><td>22</td><td></td></tr><tr><td>4</td><td></td><td>1</td><td>7</td></tr><tr><td>5</td><td></td><td>90</td><td>8</td></tr><tr><td>6</td><td></td><td>70</td><td></td></tr><tr><td>7</td><td></td><td>6</td><td></td></tr><tr><td>8</td><td></td><td>100</td><td></td></tr></table></td><td>4
AO2.1 (4)</td><td>Allow follow through for errors from 1di</td></tr>
<tr><td>2</td><td>a</td><td></td><td>1 mark per bullet
• Weighted/Undirected
• Graph</td><td>2
AO1.2 (2)</td><td></td></tr>
<tr><td>2</td><td>b</td><td>i</td><td>E.g. Weighting/cost based on estimated distance from final node</td><td>1
AO2.1</td><td></td></tr>
<tr><td>2</td><td>b</td><td>ii</td><td>Used to speed up process of finding solution</td><td>1
AO1.1</td><td></td></tr>
</table>

<table>
<tr>
<td>2</td>
<td>b</td>
<td>iii</td>
<td>1 mark per bullet, max 7 for calculations/explanation, max 1 for correct final path

• Visiting H with correct heuristic
• Visiting G and N from H
o Calculating correct distance+heuristic for G and N
• Identifying G as the smallest value
• Visiting L and M from G
o Calculating distance+heuristic for L and M
• Identifying L as the smallest value
• Visiting E
o Calculating distance+heuristic for E

• Final path: H-G-L-E

e.g.
<table>
<tr><th>Node</th><th>Distance travelled</th><th>Heuristic</th><th>distance+heuristic</th><th>previous node</th></tr>
<tr><td>H</td><td>0</td><td>80</td><td>80</td><td>-</td></tr>
<tr><td>G</td><td>25</td><td>70</td><td>95</td><td>H</td></tr>
<tr><td>N</td><td>210</td><td>90</td><td>300</td><td>H</td></tr>
<tr><td>L</td><td>51+25=76</td><td>50</td><td>126</td><td>G</td></tr>
<tr><td>M</td><td>176+25=201
<s>233+210=443</s></td><td>20</td><td>221
<s>463</s></td><td>G
<s>N</s></td></tr>
<tr><td>E</td><td>307+76=383</td><td>0</td><td>383</td><td>L</td></tr>
</table>
</td>
<td>8
AO1.2
(4)
AO2.1
(4)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>iv</td>
<td>1 mark for decision, 2 marks for effect
e.g.
Decision:
• Choosing which node to take next
• The shortest distance+heuristic is taken

Effect:
• All adjoining nodes from this new node are taken
• Other nodes are compared again in future checks
• Assumed that this node is a shorter distance</td>
<td>3
AO2.1
(3)</td>
<td></td>
</tr>
</table>

			• Adjoining nodes may not be shortest path … • ….may need to backtrack to previous nodes		
2	**c**		**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of concurrent processing; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided (searching algorithms). Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well balanced. Evaluative comments are consistenly 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 concurrent processing; 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. (searching algorithms) 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 concurrent processing with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired	**9** **AO1.1** **(2)** **AO1.2** **(2)** **AO2.1** **(2)** **AO3.3** **(3)**	**AO1: Knowledge and Understanding** **Indicative content** • Carrying out more than one task at a time • Multiple processors • Each processor performs simultaneously • Each processor performs tasks independently and/or • A program has multiple threads • Each thread starts and ends at different times • Each thread overlaps • Each thread runs independently **AO2: Application** • Each processor/thread performs a search in a different direction • Rather than going down one path, go down 2+ • E.g. apply different searches simultaneously - perform breadth-first and depth-first simultaneously • E.g. A* take the two shortest routes at each decision point, update same table • Linear search can have multiple processors searching different areas at the same time. • Binary search doesn't benefit from an increase in speed with additional processors

			knowledge and understanding to the context provided (searching algorithms). 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.		**AO3: Evaluation** Candidates will need to evaluate the benefits and drawbacks of concurrent processing in searching e.g. • Possibly find solution faster • Takes up more memory • Increase program throughput • May waste time investigating inefficient solutions • More difficult to program especially to cooperate • More memory intensive • Linear search scales very with additional processors • Binary search can perform better on large data sets with one processor than linear search with many processors
3	**a**		1 mark for each feature e.g. • Involves calculations • Has inputs, processes and outputs • Involves logical reasoning	**2** **AO1.2** **(2)**	Allow any suitable feature
3	**b**		**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of decomposition and abstraction; 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 consistenly relevant and well considered *There is a well-developed line of reasoning which is clear and logically*	**9** **AO1.1** **(2)** **AO1.2** **(2)** **AO2.1** **(2)** **AO3.3** **(3)**	**AO1: Knowledge and Understanding** **Indicative content** Decomposition: • splits problem into sub-problems • splits these problems further • until each problem can be solved • Allows the use of divide and conquer Abstraction

			structured. The information presented is relevant and substantiated. **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knoledge and understanding of decomposition and abstraction; 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 decomposition and abstraction 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.		• Removing unnecessary elements using symbols • Removing unnecessary design/programming/ computational resources **AO2: Application** • Split the simulation into subparts • E.g. generating rooms, patients, people, scenarios, interaction • E.g. replacing how instruments look with shapes, minimise features of human body **AO3: Evaluation** e.g. • Increase speed of production • Assign areas to specialities • Allows use of pre-existing modules • Allows re-use of new modules • Need to ensure subprograms can interact correctly • Can introduce errors • Reduces processing/memory requirements • Increases response speeds of programs
3	**c**		2 marks for definition, max 2 for application Caching: • Data that has been used is stored in cache/ram in case it is needed again • Allows faster access for future use	**4** **AO1.1** **(2)** **AO2.1** **(2)**	Allow any reasonable example A well-developed example can gain two marks

			Application: e.g. • Store patients' details/conditions/appearance • Design of people in the simulation • Design of specific rooms		
3	**d**	**i**	1 mark per bullet • How the times scales as data size increases • O(n) = linear complexity • Increases at the same rate as the number of data items increases	**3** **AO1.1** **(1)** **AO2.1** **(2)**	

<table>
<tr><td>3</td><td>d</td><td>ii</td><td>1 mark each
A = exponential
B = logarithmic</td><td>2
AO1.1
(2)</td><td></td></tr>
<tr><td>3</td><td>d</td><td>iii</td><td>1 mark for recommendation, max 3 for explanation
• Recommend: Solution B
Justification
• A’s space does not scale well when increased in number of items
• B’s space scales well / does not increase significantly with number of items
• As n increases at some point a will require significantly more space than B
• Both have same time complexity so need to look at space</td><td>4
AO1.2
(2)
AO2.1
(2)</td><td></td></tr>
<tr><td>3</td><td>e</td><td></td><td>1 mark per bullet, max 2 for each tools
Breakpoints
• Use to test the program works up to/at specific points
• Check variable contents at specific points
• Can set a point where the program stops running

Stepping
• Can set the program to run line by line
• Slow down/watch execution
• Find the point where an error occurs</td><td>4
AO1.1
(2)
AO1.2
(2)</td><td></td></tr>
<tr><td>4</td><td>a</td><td></td><td>1 mark for each correct swap identified/described

<table><tr><td>sheep</td><td>rabbit</td><td>dog</td><td>fox</td><td>cow</td><td>horse</td><td>cat</td><td>deer</td></tr><tr><td>sheep</td><td>rabbit</td><td>fox</td><td>dog</td><td>cow</td><td>horse</td><td>cat</td><td>deer</td></tr><tr><td>sheep</td><td>rabbit</td><td>fox</td><td>dog</td><td>horse</td><td>cow</td><td>cat</td><td>deer</td></tr><tr><td>sheep</td><td>rabbit</td><td>fox</td><td>dog</td><td>horse</td><td>cow</td><td>deer</td><td>cat</td></tr><tr><td>sheep</td><td>rabbit</td><td>fox</td><td>horse</td><td>dog</td><td>cow</td><td>deer</td><td>cat</td></tr><tr><td>sheep</td><td>rabbit</td><td>fox</td><td>horse</td><td>dog</td><td>deer</td><td>cow</td><td>cat</td></tr><tr><td>sheep</td><td>rabbit</td><td>horse</td><td>fox</td><td>dog</td><td>deer</td><td>cow</td><td>cat</td></tr></table></td><td>6
AO1.2
(3)
AO2.1
(3)</td><td></td></tr>
<tr><td>4</td><td>b</td><td></td><td>1 mark per bullet to max 7
• Correct function declaration, taking string1 and string2 as parameters
• Use of a flag</td><td>7
AO2.1
(1)
AO2.2</td><td>Allow reversed true and false</td></tr>
</table>

			• Looping through string2 by some means
			• Using string manipulators to check either letters or substrings of string2
			• Correctly setting return value to `true`
			• Returning `true` or `false` accordingly
			• Comments that explain how the algorithm works
			
			e.g.
			`function contains(string1, string2)`
			`  wordInside = false`
			`  for i = 0 to (string2.length – string1.length)`
			`  if string2.substring(i, string1.length) == string1 then`
			`  wordInside=true`
			`  endif`
			`  next i`
			`  return wordInside`
			`endfunction`
				(2)
				AO3.2
				(4)	

4	**c**	**i**	1 mark per bullet to max 4 • Merge sort splits the data • Merge sorts the split data as it is put back together • Bubble moves through the data in a linear way • Bubble moves through the data repeatedly • Merge is more efficient with larger volumes of data to sort • Merge may require more memory space	**4** **AO1.1** **(1)** **AO1.2** **(2)** **AO2.1** **(1)**	Allow points by demonstration/example
4	**c**	**ii**	1 mark per example e.g. • Insertion • Quick	**2** **AO1.1** **(2)**	
4	**d**		1 mark for each bullet • Duck is smaller than goat • Duck is less than frog/elephant • Duck is equal to duck/less than elephant so only duck left	**3** **AO2.1** **(3)**	

5	**a**	**i**	1 mark per bullet to max 3
			• Declaring the procedure and taking a player ID as parameter
			• Setting `playerID` to parameter
			• Setting `boardPosition` to 0 and money to 2000
			
			e.g.
			`public procedure new(thePlayerID)`
			`  playerID = thePlayerID`
			`  boardPosition = 0`
			`  money = 2000`
			`endprocedure`
				3
				AO2.2
				(2)
				AO3.2
				(1)	

5	**a**	**ii**	1 mark per bullet to max 4
			• Declaration as public and procedure, named constructor/new and Taking all correct parameters (missing currentLevel)
			• Sets currentLevel to 0
			• Setting all the data…
			• …to the matching parameters taken
			e.g.
			`public procedure new(theName, theCost, theL0, theL1,`
			`theL2, theL3, theImageLink, theSetSquare, theOwned)`
			`  name = theName`
			`  currentLevel = 0`
			`  cost = theCost`
			`  L0 = theL0`
			`  L1 = theL1`
			`  L2 = theL2`
			`  L3 = theL3`
			`  imageLink = theImageLink`
			`  setSquare = theSetSquare`
			`  owned = theOwned`
			`endprocedure`
				4
				AO2.2
				(2)
				AO3.2
				(2)	

5	**a**	**iii**	1 mark per bullet to max 2 • Creating new instance e.g. squirrel = new Animal • Parameters matching part (b)(i) e.g.	**2** **AO2.1** **(2)**	

			squirrel = new Animal(“Squirrel”, 1000, 10, 50, 100, 500, “squirrel.bmp”, 6, “free”)		

5	**b**	**i**	1 mark for each correctly completed space
			
			function playerMove(currentPlayer)
			dice1 = random(1,6)
			dice2 = random(1,6)
			position = **currentPlayer.getPosition()** + dice1 + dice2
			if **dice1** == dice2 then
			pickDeck(currentPlayer)
			endif
			
			if position > 25 then
			currentPlayer.setMoney(currentPlayer.getMoney() + **500)**
			position = position - **26**
			endif
			if position == **13** then
			missAGo(currentPlayer)
			elseif position != 0 then
			checkAnimal(currentPlayer)
			endif
			return position
			endfunction
				6
				AO2.2
				(3)
				AO3.2
				(3)	

5	**b**	**ii**	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of passing values by reference and by value; 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	**9** **AO1.1** **(2)** **AO1.2** **(2)** **AO2.1** **(2)** **AO3.3**	**AO1: Knowledge and Understanding** **Indicative content** By Value • sends the actual value • if changes are made then only the local copy is amended By Reference • sends a pointer to the value

			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 knoledge and understanding of passing values by reference and by value; 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 passing values by reference and by value 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.	**(3)**	• The actual value is not sent/received • If changed the original is also changed when the subroutine ends **AO2: Application** • Send by value • The `currentPlayer` value is not /does not need to be changed in the subprogram • Send by reference • The `currentPlayer` value is updated **AO3: Evaluation** • ByValue creates new memory space… • ByReference means existing memory space is used • Depends if original variable is local/global • If local and just referenced, send by value • If original value needs editing send by reference • If passing by reference then instead of returning `position` the code could just amend `currentPlayer.position` • If passing by value there could be inconsistencies when `currentPlayer` is passed to other methods, for example `pickDeck`

<table>
<tr>
<td>5</td>
<td>c</td>
<td></td>
<td>1 mark per bullet to max 6
• Procedure declaration with correct name and parameter
• Outputting the correct text from deck at <code>headPointer</code>
• Sending to <code>currentPlayer.setMoney</code> …
• <code>getMoney</code> + <code>deck</code> at head pointer amount
• Increase the head pointer
• Set <code>headPointer</code> to 0 if position 40 or greater

<pre>
procedure pickDeck(currentPlayer)
     output(deck[headPointer].getTextToDisplay())
     amount = deck[headPointer].getMoney()
     currentPlayer.setMoney(currentPlayer.getMoney() +
amount)
      headPointer = headPointer + 1
      if headPointer == 40 then
          headPointer = 0
      endif
endprocedure
</pre>
</td>
<td>6
AO2.2 (4)
AO3.2 (2)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>d</td>
<td></td>
<td>• Declaring the procedure with correct parameters
• Check if the space/animal is free
o If free, outputting name and cost
o Checking if they want to buy
▪ Calling purchase with current player and animal
• If they own the animal, checking if they can upgrade
▪ If they can, asking if they want to upgrade outputting the cost
▪ If they want to, calling the <code>upgrade</code> method
• If they don’t own the animal
o Calling <code>chargeStay</code> with the amount to charge and the current player</td>
<td>10
AO2.2(5)
AO3.2(5)</td>
<td>Allow follow through for incorrect accessing of methods</td>
</tr>
</table>

e.g.

```
procedure checkAnimal(currentPlayer)

   if board[currentPlayer.getPosition()].owned == “free”
      answer = input(“Would you like to purchase “,
                board[position].getName(), “? It costs “,
                board[position].getCost()

      if answer = “yes” then
         purchase(currentPlayer, board[position])
      endif

   elseif board[currentPlayer.getPosition()].getOwned() == currentPlayer

       if board[currentPlayer.getPosition()].getCurrentLevel()
!= “L3”
          answer = input(“Upgrade? It costs “,
board[position].getCost())

           if answer == “yes” then
              board[currentPlayer.getPosition()].upgrade(currentPlayer)
           endif

        endif

   else
      amount = board[position].getAmountToCharge()
      chargeStay(amount, currentPlayer)
   endif
endprocedure
```

Question	Assessment Objectives							Total
	AO1.1	AO1.2	AO2.1	AO2.2	AO3.1	AO3.2	AO3.3	
1ai	3							3
1aii		1						1
1b			4					4
1ci	1	2	2					5
1cii	1	2	2					5
1di*m*			3					3
1dii		1						1
1diii*m*			4					4
2a		2						2
2bi			1					1
2bii	1							1
2biii*m*		4	4					8
2biv			3					3
2c*	2	2	2				3	9
3a		2						2
3b*	2	2	2				3	9
3c	2		2					4
3di	1		2					3
3dii	2							2
3diii*m*		2	2					4
3e	2	2						4
4a		3	3					6
4b*m*			1	2		4		7
4ci	1	2	1					4
4cii	2							2
4d			3					3
5ai				2		1		3
5aii				2		2		4
5aiii			2					2
5bi*m*				3		3		6
5bii*	2	2	2				3	9
5c*m*				4		2		6
5d*m*				5		5		10
TOTALS	22	29	45	18	0	17	9	140

* = extended response

m = mathematical content