OCR
Oxford Cambridge and RSA

Practice Paper 1

GCE Computer Science

H446/02 Algorithms and Programming

Duration: 2 hour/s 30 minutes

MAXIMUM MARK 140

This document consists of 34 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

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 - Only AO3.3 is assessed in the external assessment
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.

The breakdown of Assessment Objectives for A Level in Computer Science:

	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.

MARK SCHEME:

Question			Answer	Marks	Guidance
1	a		1 mark per data item, accept any appropriate, sensible suggestions • Number of other planes that could be in the sky (1) • Speed(1) • Flight path(1) • Altitudes(1) • Rate of acceleration(1)	**3** **AO2.2 (3)**	
1	b		Max 1 for explanation of concurrent programming. Max 3 for each example. Concurrent processing: • One process does not have to finish before the other starts(1) Example e.g. • Each plane can move independently(1) • All move at the same time (1) • All need to react to different events(1) • The weather(1) • Wind, rain, direction of air etc. (1) • Each element needs to be run simultaneously(1) • It will react to its own stimulii(1)	**4** **AO1.2 (1)** **AO2.1 (3)**	Accept any reasonable suggestion for concurrent programming in the simulator For examples: 1 mark for identifying example. 1 mark for saying how they act concurrently. 1 mark for saying why this is necessary.
1	c		1 mark per bullet e.g. • It is safer... (1) • ...Real planes/lives are not put at risk by testing it in reality(1) • Time can be sped up/decreased ... (1) • ... do not need to wait to see what happens, can view changes immediately(1) • It will cost less... (1) • ...Can make multiple changes/test all possibilities(1)	**4** **AO1.1 (2)** **AO2.1 (2)**	

Question			Answer	Marks	Guidance
1	d		1 mark per bullet to max 3 • Removing unneeded complexities (1) • Saves memory/resources (1) • E.g. remove passengers, other planes, other obstacles, landscaping to reduce memory needed (1)	**3** **AO1.2 (2)** **AO2.2 (1)**	
2	a		2 marks, 1 for defining visualisation, 1 for application to the 2-d array and grid • Presents data in an easy-to-grasp way(1) • An array is not actually a grid/table(1)	**2** **AO1.1 (1)** **AO2.1 (1)**	
2	b	i	1 mark per bullet to max 2 • Declares a function called `generateobstacle`(1) • Has parameter `diceNumber` (1)	**2** **AO1.1 (1)** **AO2.1 (1)**	
2	b	ii	• 08(1)	**1** **AO2.1 (1)**	
2	b	iii	max 3 marks for benefit, max 3 for drawback, max 4 marks overall Benefit • More natural to read (1) • Quicker to write/less lines of code.. (1) As some functions are naturally recursive(1) • Suited to certain problems (1) For example those using trees (1) • Can reduce the size of a problem with each call.(1) Drawback • Can run out of stack space/memory(1) (due to too many calls (1)) causing it to crash(1) This can be avoided with tail recursion (1) • More difficult to trace/follow(1) as each frame on the stack has its own set of variables(1) • Requires more memory than the equivalent iterative algorithm. • Usually slower than iterative methods (1) due to maintainence of the stack (1)	**4** **AO1.1 (2)** **AO1.2 (2)**	

Question			Answer	Marks	Guidance

2	b	iv	1 mark per bullet
			• Loop start and end in correct positions(1)
			• With correct number of iterations(1)
			• Returns a value(1)
			• All other code correct, in the right place(1)
			
			e.g.
			`function generateobstacle(diceNumber)`
			`  for count = 0 to diceNumber`
			`  x = randomNumber(0, 7)`
			`  y = randomNumber(0, 7)`
			`  board(x, y) = new obstacle()`
			`  next count`
			`  return true`
			`endfunction`
				4
				AO2.2 (1)
				AO3.2 (3)	

2	b	v	1 mark per bullet, to max 3
			• Appropriate declaration of function, taking 2 parameters(1)
			• Checks position in board against "" correctly(1)
			• Returns `false` and true correctly(1)
			e.g.
			`function checkFree(x, y)`
			`  if board(x, y) == "" then`
			`  return true`
			`  else`
			`  return false`
			`  endif`
			`endfunction`
				3
				AO2.1 (1)
				AO3.2 (2)	

2	c		Max 2 marks for explanation of benefits. Max 2 marks for example related to this scenario • Code can be re-used(1) • …Saves time (1) • Can use subroutine(s) in other programs(1) • …saves time(1) • Can test independently…(1) • … may make finding errors easier(1) • Any suitable example, e.g. the code for rolling dice can be written once (1), then called whenever needed in the game (1)	**4** **AO1.1 (1)** **AO1.2 (1)** **AO2.1 (2)**	The question states there is only 1 programmer, so splitting the code and giving it to different programmers is not relevant to this scenario
2	d	i	1 from • Evaluate the complexity of the algorithm(1) • Show how the time/memory/resources increase as the data size increases (1) • Evaluate worst case scenario for the algorithm (1)	**1** **AO1.1 (1)**	
2	d	ii	• As the dice no increases, the time the function takes to run increases proportionally / linearly (1)	**1** **AO2.1 (1)**	
3			**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of data; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context proided. 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 knoledge and understanding of data mining; 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.	**9** **AO1.1 (2)** **AO1.2 (2)** **AO2.1 (2)** **AO3.3 (3)**	**AO1: Knowledge and Understanding** **Indicative content** • Data mining looks through vast quantities of data • Searches for relationships between facts/components/events that may not be obvious • May include pattern matching algorithms • May involve anomaly detection algorithms • Used for business modelling • Used to plan for future

			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 data mining 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.		eventualities **AO2: Application** • Can look for links between a customer's purchases • Give recommendations for future purchases • Check for days/times/months where increases are likely and what the increase will be purchasing • Look at matching sales, when people buy one product what else do they buy with it **AO3: Evaluation** Candidates will need to evaluate the benefits and drawbacks of using data mining. e.g. • Can improve marketing • Can improve quantity of stock needed • Ensure demand is met • Increase sales/profit • Takes vast processing requirements • Need powerful computers • Privacy concerns from customers • Misuse of information • Inaccurate information

					can produce false results
4	a		1 mark for linear search, 2 for justification Justification: • The array is not sorted(1) • Linear does not need ordered/linear goes through all elements from beginning/binary needs a sorted array(1)	**3** **AO1.1 (2)** **AO2.1 (1)**	
4	b	i	1 mark for each set of 2 moves • 02174356 01274356(1) • 01247356 01234756(1) • 01234576 • 01234567(1)	**3** **AO1.1 (2)** **AO2.1 (1)**	Allow follow through if one move is incorrect

<table>
<tr>
<td>4</td>
<td>b</td>
<td>ii</td>
<td>1 mark per bullet to max 6
• Uses divide-and-conquer(1)
• First item becomes pivot / 2 is the pivot(1)
• Compare each item to the pivot (e.g. compare 0 to 2, then 1 to 2)(1)
• Make two lists, 1 with less than the pivot... (0, 1) (1)
• ... 1 with more than the pivot (7,4,3,5,6) (1)
• Quick sort the new lists(1)
• Recombine the sub-lists(1)

OR
Example of alternative quicksort method
<table>
<tr><td></td><td>2 →</td><td>0</td><td>1</td><td>7</td><td>4</td><td>3</td><td>5</td><td>6 ←</td></tr>
<tr><td></td><td>2</td><td>0 →</td><td>1</td><td>7</td><td>4</td><td>3</td><td>5</td><td>6 ←</td></tr>
<tr><td></td><td>2</td><td>0</td><td>1 →</td><td>7</td><td>4</td><td>3</td><td>5</td><td>6 ←</td></tr>
<tr><td></td><td>2</td><td>0</td><td>1</td><td>7 →</td><td>4</td><td>3</td><td>5</td><td>6 ←</td></tr>
<tr><td>Swap 7 and 6</td><td>2</td><td>0</td><td>1</td><td>6 →</td><td>4</td><td>3</td><td>5</td><td>7 ←</td></tr>
<tr><td></td><td>2</td><td>0</td><td>1</td><td>6 →</td><td>4</td><td>3</td><td>5 ←</td><td>7</td></tr>
<tr><td>Swap 5 and 6</td><td>2</td><td>0</td><td>1</td><td>5 →</td><td>4</td><td>3</td><td>6 ←</td><td>7</td></tr>
<tr><td></td><td>2</td><td>0</td><td>1</td><td>5</td><td>4 →</td><td>3</td><td>6 ←</td><td>7</td></tr>
<tr><td></td><td>2</td><td>0</td><td>1</td><td>5</td><td>4</td><td>3 →</td><td>6 ←</td><td>7</td></tr>
<tr><td></td><td>2</td><td>0</td><td>1</td><td>5</td><td>4</td><td>3</td><td>6 ← →</td><td>7</td></tr>
</table>
marks for:
• Uses divide-and-conquer(1)
• Highlight first list element as start pointer, and last list element as end</td>
<td>6
AO1.1 (1)
AO1.2 (2)
AO2.1 (3)</td>
<td>If no description i.e. the candidate has just shown the quick sort, max 4 marks.</td>
</tr>
</table>

			pointer • Repeatedly compare numbers being pointed to... • ...if incorrect, swap and move end pointer • ...else move start pointer • Split list into 2 sublists • Quick sort each sublist • Repeat until all sublists have only 1 number • Combine sublists		

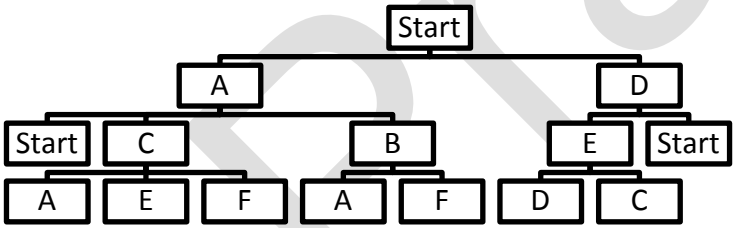
4	c		**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of merge and bubble sorts; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. *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 merge and bubble sorts; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate 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 merge and bubble sorts with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides 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**	**9** **AO1.1 (2)** **AO1.2 (2)** **AO2.1 (2)** **AO3.3 (3)**	**AO1: Knowledge and Understanding** **Indicative content** • Merge sort uses sub-lists • Bubble sort uses a temp element • Bubble sort moves through the list repeatedly • Merge sort divides the list into smaller lists • Merge is a recursive algorithm • Worst case is logarithmic, scales up well • Worst case is exponential, does not scale up well **AO2: Application** • Small data set • Few changes are needed • Demonstrates use of merge and/or bubble on the array • Calculations of average speed/best speed/worse speed **AO3: Evaluation** Candidates will need to evaluate the benefits and

			No attempt to answer the question or response is not worthy of credit.		drawbacks of each sorting algorithm e.g. • Merge is fast on large data sets • Bubble is intuitive (easier to program) • Both are fast (or even) on smaller data sets • Bubble's average speed is worse than merge • Bubble will be easier to write for such a small data set • Accept argument for either way as long as justified
5	a		max 1 mark for purpose and max 2 for application to line 8. • Case/convert numeric value to a string • Result of `x MOD 2` will be a number... • ...needs to be concatenated to string y	**3** **AO1.2 (1)** **AO2.2 (2)**	**Accept "avoid type mismatch error" for 1 mark as part of application to line 8**
5	b	i	1 mark for `flag = false`. 1 mark for showing y's values correctly through loop 1 mark for showing x's values correctly through loop	**4** **AO1.2 (1)** **AO2.1 (3)**	If only the result is shown, 1 mark only. Award bod if no "" with y

			1 mark for the correct answer
			
			• `10 > 0` (1)
			• `y = "0"`
			`x = 5`
			`y = "10"`
			`x = 2`
			`y = "010"`
			`x = 1`
			`y = "1010"` (1 for y at each stage)
			`x=0` (1 for x at each stage)
			• `01010` (or `y = "01010"`)
					Accept any suitable answer, e.g. trace table

5	b	ii	1 mark for `flag = true` and `x=13`
			1 mark for showing y's values correctly through loop
			1 mark for showing x's values correctly through loop
			1 mark for the correct answer
			
			• `flag = true`
			`x = 13`
			• `y = "1"`
			`x = 6`
			`y="01"`
			`x=3`
			`y="101"`
			`x=1`
			`y="1101"` (1 for y at each stage)
			`x=0` (1 for x at each stage)
			• `result=11101 (or y="11101")`
				4
				AO1.2 (1)
				AO2.1 (3)
					If only the result is shown, 1 mark only.
					Award bod if no "" with y
					
					Accept any suitable answer, e.g. trace table

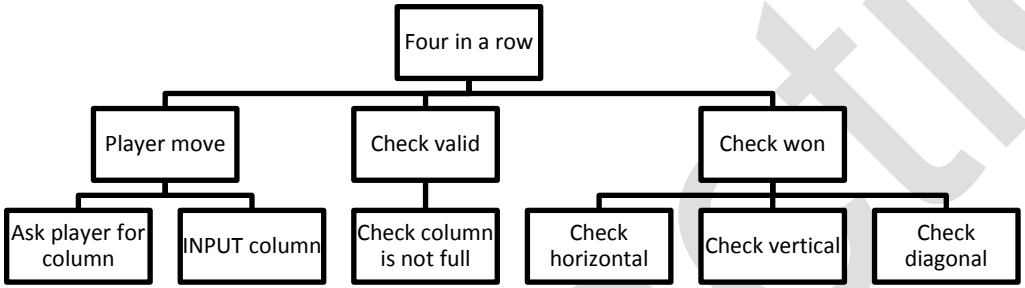
5	b	iii	• Converts a denary number into sign and magnitude	**1** **AO2.1 (1)**	cao
5	c	i	1 mark per bullet to max 3 • Past values will remain(1) • If the last value was a negative number it will remain `true`... (1)	**3** **AO1.2 (1)** **AO2.1 (2)**	

			• ...future positive values will still have the flag `true` / will be treated as a negative number(1)		
5	c	ii	1 from • Put an else statement in/another IF to set flag to `false` if needed(1) • Make `flag` be `false` in the first line of the procedure(1)	**1** **AO2.2 (1)**	
6	a		1 mark per bullet to max 2 e.g. • Places have been replaced with variables ... (1) • ...e.g. a place has been replaced with A(1) • Irrelevant information has been removed... (1) • ... e.g. only the routes and places are shown(1) • Time is given as a numeric value(1)... • ...e.g. 1 rather than 1 hour, or 1 minute(1) • Relative geographic location may not be accurate (1) • ... e.g. positions of the towns may not be proportional to actual distance (1)	**2** **AO1.2 (2)**	
	b	i	1 mark for completing A,E,F below C 1 mark for completing A, F below B 1 mark for completed D, C below E 	**3** **AO2.2 (3)**	

6	b	ii	In a binary tree a node can only have two children	**1** **AO1.2 (1)**	
6	c	i	1 mark per bullet to max 2 • Collection of data nodes/vertices(1) • Connections/edges are set between nodes/vertices(1) • Graph (edges) can be directional or bi-directional(1) • Graphs (edges) can be directed or undirected(1)	**2** **AO1.1 (2)**	

6	c	ii	1 mark each
			
			`markAllVertices (notVisited)`
			
			`createQueue()`
			
			`start = currentNode  (1)`
			
			`markAsVisited(start)  (1)`
			
			`pushIntoQueue(start)`
			
			`while QueueIsEmpty() == false  (1)`
			
			`  popFromQueue(currentNode)`
			
			`  while allNodesVisited() == false`
			
			`  markAsVisited(currentNode)  (1)`
			
			`  //following sub-routine pushes all nodes connected to`
			`  //currentNode AND that are unvisited`
			`  pushUnvisitedAdjacents()`
			
			`  endwhile`
			
			`endwhile`
				4
				AO1.2 (2)
				AO2.1 (1)
				AO3.2 (1)	

6	d		Max 6. 1 mark for final solution, max 5 for showing the stages • Mark A as the current node(1) • Record B is 5, C is 3, D is 3(1) • Mark A as visited(1) • C is shortest distance from A(1) • (C as current) Record E as 6, F as 6(1) • Mark C as visited(1) • (D as current) Record E as 5(1) • Mark D as visited(1) • (B as current) Record F as 7, do not update table as longer(1) • Mark B as visited(1) • (E as current) Record D as 8, do not update table as longer and E as visited(1) • A-C-F found as shortest(1)	**6** **AO1.2 (3)** **AO2.1 (3)**	
7	a	i	1 mark per bullet to max 2 • Breaks a problem down into its component parts(1) • Game can be divided into subprograms(1) • Subprograms can then be programmed as subroutines(1)	**2** **AO1.1 (2)**	

7	a	ii	1 mark for identifying all of the sub-modules in each sub-module. Allow any appropriate additions, do not award a mark for any inappropriate or clearly incorrect sub-modules. • Player move must ask for the move to be input(1) • Check valid must check if the column is full(1) • Check won must check horizonal, vertical and diagonal(1) e.g.  	**3** **AO2.2 (3)**	
7	a	iii	1 mark for the example, 1 mark for explanation of why it is pipelining e.g. • The results from the player move subroutine ... (1) • ...will feed into the check valid subroutine(1)	**2** **AO1.2 (1)** **AO2.1 (1)**	Accept any valid example

7	b		Programming steps to award marks for, max 6
			• Function declaration taking 2 parameters(1)
			• Looping through all 6 elements in the array…(1)
			• …in the correct order (bottom to top, 5 to 0) (1)
			• Place player in correct position…(1)
			• … return the position(1)
			• Return 999 if no space in that column(1)
			
			Example pseudocode
			e.g.
			`function gameMove(player, column)`
			`  for x = 5 to 0 step -1`
			`  if grid(x,column) == "" then`
			`  grid(x,column) = player`
			`  return x`
			`  endif`
			`  next x`
			`  return 999`
			`endfunction`
				6
				AO2.2 (2)
				AO3.2 (4)	

7	c		The algorithm can be tackled by either a) Checking all possible positions, b) Just checking what is around the last move c) checking the entire row. Full marks can be awarded for all possible methods, if correct.
			
			Programming steps to be awarded as follows, to max 6 marks.
			• Appropriate procedure declaration…
			• … taking at least player parameter(1)
			• Checking each element in the row(1)
			• Only checking valid options (e.g. if checking row-3, row-2, row-1, 0 then they need to check all rows are within the grid) (1)
			• Updating a counter/checking for four-in-a-row(1)
			• Appropriate output message(1)
			
			Example pseudocode:
			
			`procedure checkHorizontal (player, row)`
			
			`counter = 0`
			`for x = 0 to 6`
			`if grid(row, x)== player then`
			`counter = counter + 1`
			`if counter >= 4 then`
			`print "Player " + player + " has won"`
			`endif`
			`else`
			`counter = 0`
			`endif`
			`next x`
			
			`endprocedure`
				6
				AO2.2 (2)
				AO3.2 (4)	

7	d	i	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of queues and arrays; 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 knoledge and understanding of queues and arrays; 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 queues and arrays 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**	**9** **AO1.1 (2)** **AO1.2 (2)** **AO2.1 (2)** **AO3.3 (3)**	**AO1: Knowledge and Understanding** **Indicative content** • Arrays are static (size cannot change) • Queues are dynamic (size can change) • Queues use pointers to identify the first element (to be removed) **AO2: Application** • Array will need all elements to be moved 'down 1' each time a disc is removed • Queue will allow the front element to be taken out and then the pointer will move • Algorithms for queues can be more complex, especially as the language may use an array to implement the queue **AO3: Evaluation** Candidates will need to evaluate the benefits and drawbacks of using queues and arrays and suggest an appropriate solution e.g. • Size does not need to

			No attempt to answer the question or response is not worthy of credit.		change (Static is needed as grid is fixed size) so that benefit of queues is not necessary • Programmer has already written a program using arrays, may be less time consuming to edit it for arrays • Language may need a queue to be programmed in an array, therefore an array may be more straight forward to use • Queue does not need to move all elements each time a counter is removed, only pointers change
7	d	ii	Max 2 • Stack is last-in-first-out(1) • This game the first-in needs to be first-out(1)	**2** **AO1.1 (1)** **AO2.1 (1)**	

7	d	iii	1 mark per bullet to max 3
			• Appropriate loop to move through the column (bottom to top, so 5 to 1)
			• Replacing the grid with the value above (5 with 4 etc.)
			• Replacing row 0 with ""
			e.g.
			`procedure playDisc (removeColumn)`
			`  for x = 5 to 1 step -1`
			`  grid(x, removeColumn) = grid(x-1, removeColumn)`
			`  next x`
			`  grid(0, removeColumn) = ""`
			`endprocedure`
				3
				AO2.2 (1)
				AO3.2 (2)	

7	e		1 mark per bullet to max 7 • Stores/considers a range of next moves(1)	**7** **AO1.2 (2)**	Allow a diagrammatic answer with appropriate annotation

			• Creates branches with possible further moves (from both players) (1) • Continues branching(1) • Ranks possible moves based on success down the branches(1) • Uses a branching algorithm/step to decide which direction to follow(1) • Can increase ranks on stored moves based on past moves(1) • The current board position is at the top of the tree (1) • The possible moves are at the next level (1) • Each level gives every possible next move (1) • Can add weightings as to probability of winning (1) • Searching algorithm can find which set of moves leads to/has greatest possibility of winning (1) • Heuristics can help in the actual ranking / probability generation for each game state. (1)	**AO2.1 (2)** **AO2.2 (3)**	that meets the bullets

Assessment Objectives (AO) Grid

Question	Assessment Objectives							Total
	AO1.1	AO1.2	AO2.1	AO2.2	AO3.1	AO3.2	AO3.3	
1a				3				3
1b		1	3					4
1c	2		2					4
1d		2		1				3
2a	1		1					2
2bi	1		1					2
2bii			1					1
2biii	2	2						4
2biv				1		3		4
2bv			1			2		3
2c	1	1	2					4
2di	1							1
2dii			1					1
3*	2	2	2				3	9
4a	2		1					3
4bi	2		1					3
4bii	1	2	3					6
4c**m*	2	2	2				3	9
5a*m*		1		2				3
5bi*m*		1	3					4
5bii*m*		1	3					4
5biii			1					1
5ci		1	2					3
5cii				1				1
6a		2						2
6bi				3				3
6bii		1						1
6ci	2							2
6cii		2	1			1		4
6d*m*		3	3					6
7ai	2							2
7aii				3				3
7aiii		1	1					2
7b*m*				2		4		6
7c*m*				2		4		6
7di*	2	2	2				3	9
7dii	1		1					2
7diii				1		2		3
7e		2	2	3				7
Totals	**24**	**29**	**40**	**22**	**0**	**16**	**9**	**140**

*** = extended response**

m = mathematical content

BLANK PAGE

BLANK PAGE