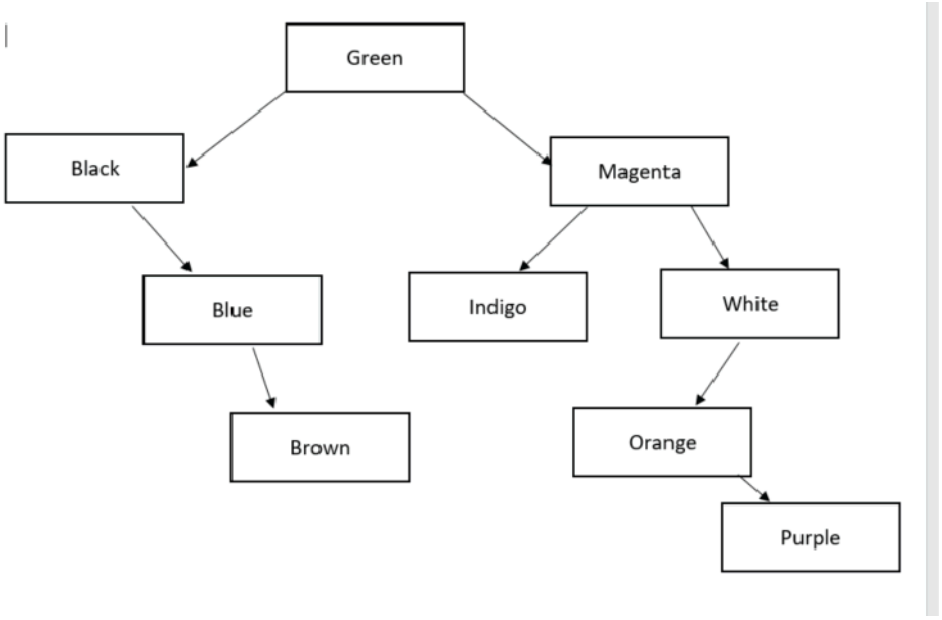
Mark scheme

Question			Answer/Indicative content	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	
		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	
	b		1 mark for each node as a correct sub-node  	4	Allow follow through e.g. if white is incorrect, but orange follows through in a logical position.
	c	i	2 marks for correct order	5	

<table>
<tr><td></td><td></td><td></td><td>

- 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

</td><td></td><td></td></tr>
<tr><td></td><td></td><td>ii</td><td>

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

</td><td>5</td><td></td></tr>
<tr><td></td><td>d</td><td>i</td><td>

1 mark for the correct pointers for nodes 1, 2 and 3

<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></td></tr>
<tr><td>5</td><td></td><td>90</td><td></td></tr>
<tr><td>6</td><td></td><td>70</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

</td><td>3</td><td>Ignore any change to FP, additional nodes and Right Pointers to node 4 and 5</td></tr>
<tr><td></td><td></td><td>ii</td><td>To identify where the next element will be placed</td><td>1</td><td></td></tr>
</table>

		iii	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 Array Index / Left Pointer / Data / Right Pointer 0 / 1 / 68 / 2 1 / 3 / 30 / 2 / 6 / 73 / 5 3 / 4 / 22 / 4 / / 1 / 7 5 / / 90 / 8 6 / / 70 / 7 / / 6 / 8 / / 100 /	4	Allow follow through for errors from 1di
			Total	**26**	
2	a		1 mark per bullet • Weighted/Undirected • Graph	2	
	b	i	E.g. Weighting/cost based on estimated distance from final node	1	
		ii	Used to speed up process of finding solution	1	

		iii	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
			  ○ Calculating correct distance+heuristic for G and N
			• Identifying G as the smallest value
			• Visiting L and M from G
			  ○ Calculating distance+heuristic for L and M
			• Identifying L as the smallest value
			• Visiting E
			  ○ Calculating distance+heuristic for E
				8	

- Final path: H-G-L-E

e.g.

Node	Distance travelled	Heuristic	distance+heuristic	previous node
H	0	80	80	-
G	25	70	95	H
N	210	90	300	H
L	51+25=76	50	126	G
M	176+25=201 ~~233+210=443~~	20	221 ~~463~~	G ~~N~~
E	307+76=383	0	383	L

iv

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
- Adjoining nodes may not be shortest path …
- ….may need to backtrack to previous nodes

3

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

9

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

			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 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.		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. **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
			Total	**24**	
3	a		1 mark for each feature e.g. • Involves calculations • Has inputs, processes and outputs • Involves logical reasoning	2	Allow any suitable feature
	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 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.	9	**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 • 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
	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 Application: e.g. • Store patients' details/conditions/appearance • Design of people in the simulation • Design of specific rooms	4	Allow any reasonable example A well-developed example can gain two marks
	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	
		ii	1 mark each A = exponential B = logarithmic	2	
		iii	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	4	

<table>
<tr><td></td><td></td><td></td><td>

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></td><td></td></tr>
<tr><td></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</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>28</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</td><td></td></tr>
<tr><td></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
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</td><td>7</td><td>Allow reversed true and false</td></tr>
</table>

			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`		

	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	Allow points by demonstration/example
		ii	1 mark per example e.g. • Insertion • Quick	2	
	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	
			Total	**22**	
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	3	

			e.g.
			`public procedure new(thePlayerID)`
			`  playerID = thePlayerID`
			`  boardPosition = 0`
			`  money = 2000`
			`endprocedure`		

		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	

		iii	1 mark per bullet to max 2
			• Creating new instance e.g. squirrel = new Animal
			• Parameters matching part (b)(i)
			e.g.
			`squirrel = new Animal("Squirrel", 1000, 10, 50, 100, 500,`
			`"squirrel.bmp", 6, "free")`
				2	

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	

	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 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*	9	**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 • The actual value is not sent/received

			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.		• 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
	c		1 mark per bullet to max 6 • Procedure declaration with correct name and parameter • Outputting the correct text from deck at headPointer • Sending to currentPlayer.setMoney …	6	

			• getMoney + deck at head pointer amount
			• Increase the head pointer
			• Set headPointer to 0 if position 40 or greater
			
			`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`		

	d		• Declaring the procedure with correct parameters
			• Check if the space/animal is free
			  ○ If free, outputting name and cost
			  ○ 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 upgrade method
			• If they don't own the animal
			  ○ Calling chargeStay with the amount to charge and the current player
				10	Allow follow through for incorrect accessing of methods

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
```

			Total	**40**	