Question			Answer/Indicative content	Marks	Guidance
1	a		1 mark per bullet • Queue outputs data in a First In First Out fashion • It will retrieve the temperature values in the order they were recorded or • Stack outputs the data in a Last In First Out fashion • It will retrieve the temperature values in the reverse of the order they were recorded	2 AO1.2 (1) AO2.2 (1)	Mark Point 1 is the definition Mark Point 2 is for context of the temperature values <u>Examiner's Comments</u> The majority of candidates were able to define a queue as a FIFO structure. Most of those who did so were able to successfully go on to relate this to the context of the question, which was to be able to retrieve the temperature values in the same order as they had originally been recorded. A number of candidates answered equally well by explaining why a stack would not have been appropriate.
	b	i	It returns a value	1 AO2.1 (1)	<u>Examiner's Comments</u> Most candidates had learnt that a function always returns a value and scored full credit for recall of knowledge. Some candidates gave an ambiguous response stating that functions produce an output (procedures can produce outputs too, e.g. printing to the screen), which lacked the precision of definition required to gain credit.

		ii	1 mark per completed word
			
			`processedData[0] = 0`
			
			`firstDay = dequeue ()`
			
			`for count = 1 to 6`
			
			`    processedData[count] = dequeue() - firstDay`
			
			`next count`
				3
				
				AO2.2 (1)
				AO3.2 (2)
					Exact answers only
					
					<u>Examiner's Comments</u>
					
					Many candidates found it difficult to complete the code successfully – most often finding the application of the *dequeue()* function to return the first item in the queue the most difficult mark. Candidates also need to remember that variable identifiers need to be constructed correctly. Identifier names cannot have spaces in them, so those who answered *first Day* instead of *firstDay* did not provide a correct solution.

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td></td><td></td><td>iii</td><td>1 mark per bullet to max 5.
Max 3 if no application to data in <code>processedData</code>

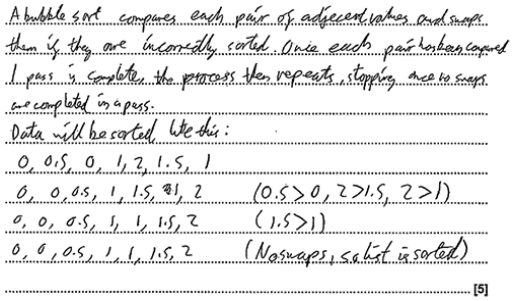
• Compares each pair of data e.g. 0 and 0.5
• If they are in the correct order it moves to the next pair e.g. 0.5 and 0
• If they are in the wrong order it swaps them e.g. 0.5 and 0 becomes 0 and 0.5
• Continues to the end of the array e.g. Pass 1 complete
• If there has been a swap it checks again e.g. Pass 2 complete
• If there have been no swaps it is sorted

<table><tr><td>0</td><td>0.5</td><td>0</td><td>1</td><td>2</td><td>1.5</td><td>1</td><td rowspan="4">Pass 1</td></tr><tr><td>0</td><td>0</td><td>0.5</td><td>1</td><td>2</td><td>1.5</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0.5</td><td>1</td><td>1.5</td><td>2</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0.5</td><td>1</td><td>1.5</td><td>1</td><td>2</td></tr><tr><td>0</td><td>0</td><td>0.5</td><td>1</td><td>1</td><td>1.5</td><td>2</td><td>Pass 2</td></tr><tr><td>0</td><td>0</td><td>0.5</td><td>1</td><td>1</td><td>1.5</td><td>2</td><td>No swaps</td></tr></table></td><td>5

AO1.1 (2)
AO1.2 (1)
AO2.2 (1)</td><td>Allow (full) credit for tables showing the bubblesort being completed.

<u>Examiner's Comments</u>

Exemplar 1

 
A bubble sort compares each pair of adjacent values and swaps them if they are incorrectly sorted. Once each pair has been compared 1 pass is complete, the process then repeats, stopping once no swaps are completed in a pass.
Data will be sorted like this:
0, 0.5, 0, 1, 2, 1.5, 1
0, 0, 0.5, 1, 1.5, 2, 2 (0.5 > 0, 2 > 1.5, 2 > 1)
0, 0, 0.5, 1, 1, 1.5, 2 (1.5 > 1)
0, 0, 0.5, 1, 1, 1.5, 2 (No swaps, so list is sorted)
[5]

Some candidates gave a purely theoretical description of bubble sort and were limited to a maximum of three marks for doing so. Where questions provide a specific data set to be used, candidates must make reference to it in their answers.

Candidates who gave diagrams in their answers (such as the exemplar) often demonstrated all of the steps of the algorithm in the clearest fashion. Candidates should be encouraged to use annotated diagrams rather than verbose prose where possible.</td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance
		iv	1 mark per bullet O(n) • Linear • Best time grows at the same rate as the number of elements • This is the case when the data is already in order $O(n^2)$ • Polynomial / Quadratic • Worst and average time is proportional to the square (polynomial) of the number of elements • Worst case is when the data is initially in the reverse order O(1) • Constant • Will always take the same amount of memory (in addition to the list itself).	6 AO1.1 (3) AO1.2 (3)	Note: First Mark Point is for the identification, second Mark Point is for the description Note: Do not allow descriptions relating to time complexity for 'Worst Space O(1)' Note: Do not allow 'equal to' in descriptions, O(n) and $O(n^2)$ grow in *proportion* to the number of items Examiner's Comments Many candidates had not learnt the appropriate terminology for linear, polynomial and constant complexity. Stronger candidates recognised that bubble sort had a best case when the items were already sorted, and a worse case when the items were in reverse order. Candidates struggled most with the concept of space complexity. Many scored marks for recognising that O(1) was constant, but then confused space complexity with time complexity. Few appreciated that O(1) meant that no additional storage was required in addition to the initial data set itself. Misconception A number of candidates equated O(n) linear complexity to mean the number of steps was equal to n. The number of steps is proportional to n since there are many other factors at play. It should be noted that ½ n and 2n would still give time complexity of O(n).
			Total	17	

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>2</td><td>a</td><td>i</td><td><u>Binary Tree</u> / <u>Binary Search Tree</u></td><td>1

AO2.1
(1)</td><td><u>Examiner's Comments</u>

Most candidates had the knowledge to recall the correct term.</td></tr>
<tr><td></td><td></td><td>ii</td><td>1 mark per bullet

• 1st layer: Lily
• 2nd layer: Daisy, Sunflower
• 3rd layer: Begonia, Hosta, Peony
• 4th layer: Rose</td><td>3
AO1.2
(1)
AO2.1
(1)
AO2.2
(1)</td><td><u>Examiner's Comments</u>

Most candidates could successfully show the result of executing a breadth-first search on the tree. Where candidates did not achieve the marks, they often confused depth-first with breadth-first.</td></tr>
<tr><td></td><td></td><td>iii</td><td>1 mark per bullet to max 4. Max 2 marks for no application to the tree.

• Depth first starts at the root (Lily)
• …and goes all the way down one branch to the bottom (Begonia)
• It stores which nodes it has visited/pushes nodes visited onto a stack
• When it cannot go any further
• …It then backtracks/returns to the previous node
• And continues to backtrack until a node is reached with unvisited children.
• …and checks down that branch
• In the tree shown, after visiting Begonia, the algorithm would backtrack to Daisy…
• …and would then visit Hosta (Accept any other example)</td><td>4

AO1.1
(1)
AO1.2
(1)
AO2.1
(1)
AO2.2
(1)</td><td><u>Examiner's Comments</u>

Candidates who gave generic answers limited themselves to a maximum of two marks. Where questions ask for a specific dataset to be used, candidates will need to do so if they are to access all of the marks available. While the general principles of a depth-first traversal were often understood in terms of traversing down the leftmost branches from Lily to Daisy to Begonia, the order in which the nodes would be output was less often appreciated.</td></tr>
<tr><td></td><td>b</td><td>i</td><td>1 mark per bullet

• Correct NextPointer values
• Suitable end/null pointer

<table>
<tr><th>Data item</th><th>Data</th><th>NextPointer</th></tr>
<tr><td>0</td><td>Begonia</td><td>1</td></tr>
<tr><td>1</td><td>Daisy</td><td>2</td></tr>
<tr><td>2</td><td>Hosta</td><td>3</td></tr>
<tr><td>3</td><td>Lily</td><td>4</td></tr>
<tr><td>4</td><td>Peony</td><td>5</td></tr>
<tr><td>5</td><td>Rose</td><td>6</td></tr>
<tr><td>6</td><td>Sunflower</td><td>null</td></tr>
</table></td><td>2

AO2.1
(2)</td><td>Exact values only. Allow -1 for null pointer or equivalent such as Φ.
Do not allow a blank or 0.

<u>Examiner's Comments</u>

Most candidates displayed some understanding of the use of pointers in a linked list and successfully gave the correct values. Some candidates erroneously gave 0 as a null pointer value or left the NextPointer value for Sunflower empty. Null, -1 and Ø were all accepted null pointer values.</td></tr>
</table>

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td></td><td></td><td>ii</td><td>1 mark per bullet

• Lavender added in position 7
• …Hosta points to 7
• …Lavender points to 3

<table><tr><th>Data item</th><th>Data</th><th>NextPointer</th></tr><tr><td>0</td><td>Begonia</td><td>1</td></tr><tr><td>1</td><td>Daisy</td><td>2</td></tr><tr><td>2</td><td>Hosta</td><td>7</td></tr><tr><td>3</td><td>Lily</td><td>4</td></tr><tr><td>4</td><td>Peony</td><td>5</td></tr><tr><td>5</td><td>Rose</td><td>6</td></tr><tr><td>6</td><td>Sunflower</td><td>null</td></tr><tr><td>7</td><td>Lavender</td><td>3</td></tr></table></td><td>3

AO1.2
(1)
AO2.2
(2)</td><td>Do not credit answers that do not place lavender in position 7 and then update pointer positions

<u>Examiner’s Comments</u>

The majority of candidates appreciated that a new entry could be made to the list by using the last available space, and then updating the relevant pointers. It was pleasing to see few candidates tried to shift the data items down, rather than updating the pointers.</td></tr>
<tr><td></td><td></td><td>iii</td><td>1 mark per bullet

• Traverse the list to the item immediately prior to the item to be removed (1)
• … which is DataItem 1 - Daisy
• Find the value of the NextPointer of the item to be removed
• … which is the NextPointer of DataItem 2 - Hosta, value 7
• Set the nextPointer of the item prior to the item to be removed to the NextPointer value of the DataItem to be removed
• … update the NextPointer of DataItem 1 - Daisy from 2 to 7 (Lavender)</td><td>4

AO1.1
(1)
AO1.2
(1)
AO2.2
(2)</td><td>Find the item before item to be deleted (Daisy)
Find nextPtr of item to be deleted (Hosta)
Update nextPtr of the item before (Daisy) to the nextPtr of item to be deleted (Hosta)
i.e. Daisy 2 is updated to Daisy 7

Allow FT from 2b(ii/iii) if candidate has used table in fig 2.1 (e.g. Daisy would now point to Lily at position 3)

<u>Examiner’s Comments</u>

Candidates sometimes struggled to break their explanations into a clear sequence of steps, using the specific data items given. Those candidates who scored well made clear reference to the relevant data items Daisy, Hosta and Lavender. A few candidates gave annotated diagrams that explained the process particularly clearly.</td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance

		iv	1 mark per bullet
			
			• Start at the `firstElement` in the list
			• Correctly looping until null pointer found / end of list
			• Outputting the data element
			• Accessing the pointer to the next element
			• Appropriate comment(s)
			
			e.g.
			`currentElement = firstElement`
			`while(currentElement != null)`
			`//Continue until last node`
			
			`print(plantList[currentElement,0])`
			`currentElement = plantList[currentElement,1]`
			`endwhile`
				5
				
				AO2.1 (1)
				AO2.2 (2)
				AO3.2 (2)
					Note: Solution must utilise pointers in a linked list; it cannot use a FOR loop as the number of elements is not known and the data is not in order by index number
					
					Note: identifiers given in the question as `plantList` and `firstElement` should be used accurately in the solution
					
					Note: allow credit for answers that interpret the data structure as an array of records/structures with data/pointer fields
					
					<u>Examiner's Comments</u>
					
					Candidates need to be familiar with a range of data structures and the associated algorithms for performing basic functions on the data structures. Many candidates erroneously thought that the individual elements of a linked list could be indexed and accessed with a for loop. Relatively few candidates appreciated that the items in the linked list could only be traversed by following each item's next pointer until the end of the list was located.
					
					Many candidates continue to show limited ability to write pseudocode.

			Total	22	

Question			Answer/Indicative content	Marks	Guidance
3	a	i	if	1 AO1.1 (1)	Examiner's Comments A number of candidates overthought the answer and did not identify the *if* branching statement used in the code.
		ii	1 mark per bullet • Branching decides which code is run / only runs code once • Iteration repeatedly runs the same code in the same sequence	2 AO1.2 (2)	Examiner's Comments Many candidates found difficulty in defining either branching or iteration clearly. Often the two terms were confused, with the clear differences between branching and iteration not expanded on.
		iii	num1, num2	1 AO2.1 (1)	Exact identifier names required Examiner's Comments Most candidates had little difficulty correctly identifying the parameters num1 and num2.
		iv	1 mark per bullet • By Value • … the original values do not need to be modified • … byRef would not work / would cause the routine to crash	2 AO2.2 (2)	Examiner's Comments Most candidates identified that the parameters should be passed by value, but fewer could successfully justify their choice, and hence show a deeper understanding of the reason for the choice they made.
		v	1 mark per bullet • Gives the remainder after division • E.g. 10 MOD 3 = 1	2 AO1.1 (1) AO1.2 (1)	Examiner's Comments Most candidates could successfully define the MOD function. A few confused it with the absolute() function or with integer division, and some did not illustrate their answer with the required example.
	b		1 mark per bullet to max 3 • Num2 != 0 therefore return GCD(20,10) • Num2 != 0 therefore return GCD(10,0) • Final return value = 10	3 AO2.1 (1) AO2.2 (2)	Allow FT for numerical errors Examiner's Comments Many candidates successfully traced the algorithm given.

Question			Answer/Indicative content	Marks	Guidance
	c	i	1 mark for benefit, 1 mark for drawback Benefit: • The program can/might run faster • Cannot run out of stack space/memory • Easier to trace/follow Drawback: • Iteration can lead to lengthier code • Iteration can lead to code that looks more complex / is harder to understand • some problems are more elegantly coded with a recursive solution	2 AO1.1 (1)	Examiner's Comments Most candidates achieved some credit, but many answers were often too vague.

		ii	1 mark for each correct statement
			
			`function newGCD(num1, num2)`
			
			`  temp = 0`
			
			`  while (num2 != 0)`
			
			`  temp = num2`
			
			`  num2 = num1 MOD num2`
			
			`  num1 = temp`
			
			`  endwhile`
			
			`  return num1`
			
			`endfunction`
				4
				AO2.2
				(2)
				AO3.2
				(2)
					Examiner's Comments
					
					Most candidates attempted the question and achieved some credit. Relatively few achieved full marks for this question.

			Total	17	

Question			Answer/Indicative content	Marks	Guidance
4	a		1 mark per bullet • She can split the problem down into sub problems • It will creates a more manageable problem / simpler to understand / maintain • can tackle each sub problem independently	2 AO1.2 (1) AO2.2 (1)	<u>Examiner's Comments</u> Decomposition was clearly understood and most candidates achieved at least one of the two marks. Those that did not gain full marks did not go beyond decomposition breaking the task into sub-tasks.
	b	i	1 mark per bullet, max 2 per sub-procedure e.g. • Select character (name, gender) • Gives the user options for choosing a character • Choose level • Give the user the choice of level (easy, normal, challenging) and take the user input • Touch enemy • Called to determine if the character touches an enemy • Lose life • Remove a life, if <0 then game over • End level • Move onto next level One mark for identifying sensible subroutine, 1 mark for description	6 AO2.1 (2) AO2.2 (2) AO3.2 (2)	Do not award any user *input* related procedures e.g. Left/Right input (but character movement *output* on screen left/right would be valid) Allow other reasonable responses from the scenario e.g. generate enemy() <u>Examiner's Comments</u> Most candidates presented reasonable procedures within the context of the question. Relatively few made the mistake of reiterating procedures to handle character movement that was specified as an example in the question.
		ii	1 mark per bullet • Decision based on what the user has input • E.g. If they click left move the character left // if they click right move the character right // if they click space bar make the character jump	2 AO2.1 (1) AO2.2 (2)	<u>Examiner's Comments</u> A number of answers lacked the clarity of a clear decision and subsequent action that would be performed. A number of answers given did not relate to the user input sub-procedure as specified.
		iii	1 mark per bullet • The result from one process / procedure feeds into the next • E.g. the result of detecting a character touching an enemy feeds into reducing the number of lives	2 AO1.2 (1) AO2.2 (1)	Note: 1 Mark Max for a generic description of pipelining <u>Examiner's Comments</u> Many candidates could define the concept of pipelining. Fewer were able to apply it within the context of the given scenario.

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td></td><td>c</td><td></td><td>1 mark for final solution, max 5 for showing the stages

• Mark A as the current node initially
• Record B = 1, C = 2 (mark A as visited)
• Record E = 5 (and mark B as visited)
• (Record D = 3, F = 5 (and mark B as visited)
• Change E to 4 (overriding previous value, and mark D as visited)
• Record G = 6 (and mark E as visited)
• …Do not change G as greater than current (mark F as visited)
• (G as visited) H = 10 (Mark G as visited)
• Solution: A-C-D-E-G-H path length 10

<table>
<tr><th>Node</th><th>Visited</th><th>From A</th><th>Previous Node</th><th></th></tr>
<tr><td>A</td><td>✓</td><td>0</td><td>-</td><td>1 Mark</td></tr>
<tr><td>B</td><td>✓</td><td>1</td><td>A</td><td rowspan="2">1 Mark</td></tr>
<tr><td>C</td><td>✓</td><td>2</td><td>A</td></tr>
<tr><td>D</td><td>✓</td><td>3</td><td>C</td><td rowspan="2">1 Mark</td></tr>
<tr><td>F</td><td>✓</td><td>5</td><td>C</td></tr>
<tr><td>E</td><td>✓</td><td>~~5~~ 4</td><td>~~B~~ D</td><td>2 Marks Initial visit, plus override values</td></tr>
<tr><td>G</td><td>✓</td><td>6</td><td>E</td><td>1 Mark</td></tr>
<tr><td>H</td><td></td><td>10</td><td>G</td><td>1 Mark</td></tr>
</table></td><td>6

AO1.2 (1)
AO2.2 (3)
AO2.2 (2)</td><td>Guidance – 1 mark only for stating the solution of A-C-D-E-G-H length 10

<u>Examiner's Comments</u>

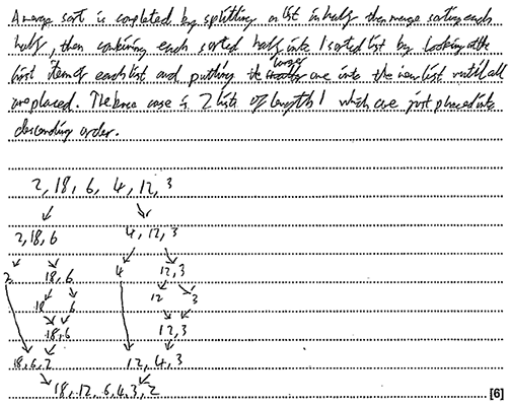
Exemplar 2

Show how Dijkstra's algorithm would find the shortest path from A to H.
<table>
<tr><th>node</th><th>visited</th><th>shortest distance from start</th><th>Previous node</th></tr>
<tr><td>A</td><td>✓</td><td>0</td><td>–</td></tr>
<tr><td>B</td><td>✓</td><td>1</td><td>A</td></tr>
<tr><td>C</td><td>✓</td><td>2</td><td>A</td></tr>
<tr><td>D</td><td>✓</td><td>3</td><td>C</td></tr>
<tr><td>E</td><td>✓</td><td>~~5~~ 4</td><td>~~B~~ D</td></tr>
<tr><td>F</td><td>✓</td><td>5</td><td>C</td></tr>
<tr><td>G</td><td>✓</td><td>6</td><td>E</td></tr>
<tr><td>H</td><td>✓</td><td>10</td><td>G</td></tr>
</table>
Shortest path is A C D E G H
and takes a total of 10 moves
[6]

Dijkstra's shortest path algorithm has appeared in previous sessions and candidates generally had a good grasp of the principles of the algorithm. Verbose text often made some responses difficult to follow. The clearest responses (as exemplified) tabulated the steps in the algorithm.</td></tr>
</table>

Question	Answer/Indicative content	Marks	Guidance
d	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of IDEs; 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 is able to weigh up the context which results in a supported and realistic judgment as to whether IDEs are useful in this context. *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 IDEs; 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 makes a reasonable attempt to come to a conclusion showing some recognition of influencing factors that would determine whether IDEs are useful in this context. *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 IDEs 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 nothing more than an unsupported assertion. *The information is basic and comunicated in an unstructured way. The information is*	9 AO1.1 (2) AO1.2 (2) AO2.1 (2) AO3.3 (3)	**AO1: Knowledge and Understanding** **Indicative content** Tools to aid writing • Coloured font • Predictive text • Auto-correct Tools to aid de-bugging • Stepping • Break points • Variable watch window **AO2: Application** e.g. • Can write subroutines for the program and it will tell you what parameters are needed • Allow you to run the program without exiting the software / having to load a separate compiler • Integrates other tools such as version control. • Can reduce spelling errors • Can use to fix errors that might occur / debug **AO3: Evaluation** e.g. • User friendly for novices • Increase speed of writing • Fewer mistakes • Increase speed of testing / finding errors • Collaborative team working facilitated Examiner's Comments AfL

Question			Answer/Indicative content	Marks	Guidance
			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.		Many candidates would benefit from being able to present a clear and logical structure to their answers. Candidates must show knowledge, application and evaluation. First, relevant knowledge (definitions) should be introduced. Secondly, examples that are relevant to the given scenario should be covered. Finally, evaluative comments should be made to compare the different methods in terms of relative advantages / disadvantages or impacts / consequences. Candidates had first-hand experience of using IDEs and features found within IDEs. Most candidates were able to give a Level 2 response giving a range of features found within an IDE along with descriptions of how they could be used. The evaluation commentary was often limited to generic comments such as increased productivity. Few candidates could offer more insightful depth to their evaluatory comments linking features to categories of users such as those features particularly beneficial to inexperienced programmers (such as predictive text, error correction suggestions) or experienced teams of programmers (collaborative online features, version control).
			Total	27	

Question			Answer/Indicative content	Marks	Guidance
5	a		1 mark per bullet to max 6. Max 4 if generic description given with no application Max 4 if a diagrammatic solution is given with no description • Splits the list in half repeatedly… • … until it is in independent arrays / elements e.g. 2, 18, 6, 4, 12, 3 • Compare the first two items (index 0 and 1) e.g. 2, 18 • … and combine to create a new array in descending order i.e. 18, 2 • Repeat with indexes 2 and 3 (6, 4), then 4 and 5 (12, 3) • Compare the first element in the first two new arrays • …Choose the largest element, writing this to the new array first • …repeat until no elements left • Combine the two remaining lists into one list e.g. e.g. [2, 18, 6, 4, 12, 3] [2, 18, 6, 4, 12, 3] [2, 18, 6] [4, 12, 3] [2, 18, 6] [4, 12, 3] [2, 18] [6] [4, 12] [3] [2, 18] [6] [4, 12] [3] [2] [18] [6] [4] [12] [3] [2] [18] [6] [4] [12] [3] [18, 2] [6, 4] [12, 3] [18 2] [6] [12 4] [3] [18, 6, 4, 2] [12, 3] [18 6 2] [12 4 3] [18, 12, 6, 4, 3, 2] [18, 12, 6, 4, 3, 2]	6 AO1.2 (3 AO2.2 (3)	Allow max 5 if correct description but in ascending order. <u>Examiner's Comments</u> **Exemplar 3**  A number of candidates simply presented learnt definitions without application to the given dataset, while other candidates presented diagrammatic solutions to the workings of the merge sort for the given data, but did not explain what was happening. Good responses such as the exemplar had explanatory text alongside a clear diagram, showing the steps that took place when the algorithm was run.
	b		1 mark per bullet • Merge sort might create a new array each time it splits and merges / often implemented recursively which places additional data on the stack • Insertion sort does not use any additional arrays//Insertion sort is an in-place algorithm.	2 AO1.2 (2)	<u>Examiner's Comments</u> Many candidates recognised that merge sort could generate an additional array each time a list was split. Fewer could explain that insertion sort worked in-situ and has a space complexity O(1).
			Total	**8**	

Question	Answer/Indicative content	Marks	Guidance
6	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of data mining; 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 is able to weigh up the context which results in a supported and realistic judgment as to whether it is possible to use data mining in this context. *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 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. The candidate makes a reasonable attempt to come to a conclusion showing some recognition of influencing factors that would determine whether it is possible to use data mining. *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 nothing more than an unsupported assertion. *The information is basic and comunicated in an unstructured way. The information is*	9 AO1.1 (2) AO1.2 (2) AO2.1 (2) AO3.3 (3)	**AO1: Knowledge and Understanding** **Indicative content** • Extracting data from databases • Using large data sets • Looking for patterns/specific occurrences of data • Gathering data that can be analysed and used to inform decisions **AO2: Application** e.g. • Use to find out what his users do • Find features that are used most often • Find features that are not used • Find out what people in his target age group do on other sites • Find out characteristics of people who use the site **AO3: Evaluation** e.g. • Can identify areas to focus attention • Save time and money by identifying areas that are not popular/used • New features targeted at specific groups could bring in new business e.g. advertising • But care would need to be applied to privacy issues / GDPR and potential impact on the users Examiner's Comments

Question			Answer/Indicative content	Marks	Guidance
			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.		Many candidates were familiar with the context of social networking and could provide examples of how data mining could present useful data to assist in facilitating customer connections or targeted advertising. However, on occasion, a number of candidates did forget to actually define what they actually meant by data mining or illustrate the type of data that could be collected for processing. A pleasing number of candidates did go on to discuss and evaluate the computational resources that would be required and/or the impact on users' personal liberties and freedoms that needed to be weighed against the potential benefits to the company.
			Total	9	

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>7</td><td>a</td><td>i</td><td>1 mark per bullet to max 4

• Class declaration
• 3 attributes declared
• Constructor
• …taking parameters
• …setting the attributes to the parameters

e.g.
<pre>class GardenItem
  private itemName
  private length
  private width
  public procedure
new(pItemName,
  pLength, pWidth)
    itemName = pItemName
    length = pLength
    width = pWidth
  endprocedure
endclass</pre></td><td>4

AO1.1
(1)
AO2.1

(1)
AO2.2
(1)
AO3.2
(1)</td><td>Note that example answers are given in the specification pseudocode. Any pseudocode answer that can be understood by a ‘competent programmer’ should be accepted

<u>Examiner’s Comments</u>

Many candidates struggled to differentiate between defining a class and defining a constructor method – with many passing parameters to the class definition rather than to the constructor method itself. While candidates are not required to use the pseudocode standard that appears in the specification, they are required to use pseudocode that is understandable. Many candidates need further practical experience of using OOP.</td></tr>
<tr><td></td><td></td><td>ii</td><td>1 mark per bullet

• Class declaration inheriting from gardenItem
• Additional 3 properties declared as private
• Constructor takes all 5 parameters
• Use of super (or equivalent) to set super class parameters
• Remainder of properties set to parameters

e.g.
<pre>class Tree inherits GardenItem
 private height
 private sun
 private shade
 public procedure new(pName,
pHeight, pLenWidth, pSun,
pShade)
  super.itemName = pName
  super.length = pLenWidth
  super.width = pLenWidth
  height = pHeight
  sun = pSun
  shade = pShade
 endprocedure
endclass</pre></td><td>5

AO1.1
(1)
AO2.2
(1)
AO3.2
(3)</td><td>Accept solutions that call the parent’s constructor.

<pre>class Tree inherits GardenItem
 private height
 private sun
 private shade
 public procedure new(pName,
pHeight, pLenWidth, pSun,
pShade)
  height = pHeight
  sun = pSun
  shade = pShade

super.new(pName,pLenWidth,
pLenWidth)
 endprocedure
endclass</pre>
<u>Examiner’s Comments</u>

Few candidates had the knowledge and skills required to answer this question well. Common misconceptions included an assumption that only the additional properties of height, sun and shade needed to be passed to the Tree class. Very few candidates set the parameters itemName, Length and Width of the super class successfully.</td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance
		iii	1 mark per bullet • Declaration of instance of tree (i.e. new Tree) • Storing result in firstTree • All parameters included and in the same order as 7bii • …with appropriate data types e.g. `firstTree = new Tree("Common Oak",40,40,true, true)`	4 AO1.1 (1) AO2.2 (1) AO3.2 (2)	Examiner's Comments Very few candidates achieved full credit, but many candidates did achieve some credit for assigning an instance of the *Tree* class to the *firstTree* identifier. Few went on to correctly pass the required parameters in the correct order.

		iv	1 mark per bullet
			
			• Get method declaration
			• Returns itemName
			• Set method declaration
			• …takes value as a parameter
			• …assigns parameter to itemName
			
			e.g.
			`function getItemName()`
			`  return itemName`
			`endfunction`
			
			`procedure setItemName(newname)`
			`  itemName = newname`
			`endprocedure`
				4
				
				AO1.2 (2)
				AO2.2 (2)
					Examiner's Comments
					
					Those candidates who gained credit in the initial parts of the question generally went on to successfully define the get and set methods. A number of candidates erroneously passed parameters into the get function or did not return the *itemName* parameter. Another common misconception was to ask for a user input to set the value of *itemName* in the set procedure, rather than passing a parameter for the assignment.
					
					Where candidates had experience of OOP is was apparent that they knew how to use getters and setters to access the private attributes of a class. Those who had no experience of this performed poorly.

Question			Answer/Indicative content	Marks	Guidance

		v	1 mark per bullet to max 6
			
			• Procedure declaration with all four parameters
			• Looping 1000 times / to end of array
			• Checking if height and width are less than or equal to the maximum height and width
			• Checking if sun and shade match
			• Outputting value(s) using get methods
			• Output an appropriate message
			• Outputs a message if no matching tree is found
			
			e.g.
			`procedure findTree(pHeight, pWidth, pSun, pShade)`
			`flag = false`
			`for i = 0 to 999`
			`if treeArray[i].getHeight() <= pHeight`
			`then`
			`if treeArray[i].getWidth() <= pWidth`
			`then`
			`if treeArray[i].getSun() == pSun`
			`then`
			`if treeArray[i].getShade() == pShade then`
			`flag = true`
			`print(treeArray[i].getItemName() + " height: " + treeArray[i].getHeight + " width: "+ treeArray[i].getWidth() + " Sun?: " + treeArray[i].getSun() + " Shade?: " + treeArray[i].getShade()`
			`endif`
			`endif`
			`endif`
			`endif`
			`next count`
			`if flag == false then`
			`print("No suitable trees")`
			`endif`
			`endprocedure`
				6
				
				AO1.2 (1)
				AO2.2 (2)
				AO3.2 (3)
					<u>Examiner's Comments</u>
					
					Many candidates successfully defined the *findTree* procedure declaration with the relevant parameters and then went on to correctly form a loop that would iterate the correct number of times. However, many candidates with poor knowledge of OOP then found it difficult to use the *Tree* attributes correctly through the use of the get methods. A sizeable number of candidates thought that the properties of the class could be addressed directly and thus showed a lack of appreciation of the encapsulation imposed by the OOP paradigm. Where candidates had a solid grounding in OOP they had little difficulty in obtaining full marks.

Question			Answer/Indicative content	Marks	Guidance
	b	i	1 mark per example e.g. • No actual images shown • Items are named / labelled • Simplified layout with shapes	2 AO2.1 (2)	Allow any reasonable examples, but they must be for different aspects <u>Examiner's Comments</u> Candidates understood the concept of abstraction and had little difficulty giving examples of how abstraction had been applied to the layout of the garden given.
		ii	1 mark per bullet to max 3 e.g. • Reduces complexity of design • Reduces complexity of programming • Reduce memory/processing requirements • Could involve a large number of images that would take excessive memory • Reality contains things that aren't relevant to a computer program	3 AO1.1 (1) AO1.2 (1) AO2.1 (1)	Note: do not allow answers related to the user experience / user interpretation, the question is about the production of the system <u>Examiner's Comments</u> Many candidates confused defining abstraction with the actual need to use abstraction. Candidates also often related their answers to the potential users of the system, rather than to the production of the program, as specified by the question.
		iii	1 mark per example e.g. • Garden dimensions/width/length • Number of items in the garden • Name of items in the garden • Location of items in the garden	3 AO2.1 (3)	<u>Examiner's Comments</u> Most candidates had little difficulty identifying suitable inputs to the system. There were occasional instances of repetition, and candidates need to guard against this.

Question	Answer/Indicative content	Marks	Guidance
c	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of caching and reuseable components; 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 is able to weigh up the use of both caching and reusable components which results in a supported and realistic judgment as to whether it is possible to use them in this context. *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 caching and reuseable components; 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 makes a reasonable attempt to come to a conclusion showing some recognition of influencing factors that would determine whether it is possible to use caching and reusable componments in this context. *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 caching and reuseable components 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.	9 AO1.1 (2) AO1.2 (2) AO2.1 (2) AO3.3 (3)	**AO1: Knowledge and Understanding** **Indicative content** Caching: • Data that has been used is stored in cache/RAM in case it is needed again • Allows faster access for future use Reusable components • One piece of code can be used in multiple places / called many times • Use of subroutines / procedures / functions • Use of classes • Use of external libraries **AO2: Application** • Store items in cache • Store requirements in cache • Store garden layout in cache • Reuse shapes / designs • The use of a class allows replication **AO3: Evaluation** e.g. • Faster development • Faster/easier future adaptation • Better performance of program • Takes more time to plan/design to make use of both **Examiner's Comments**

Question			Answer/Indicative content	Marks	Guidance
			The candidate provides nothing more than an unsupported assertion. *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.		Most candidates were able to identify generic points about re-usable code / modules. Fewer could apply their knowledge of re-usable components in the context of the given scenario, and more context was required to achieve higher scores. It was pleasing to see a number of strong responses that referenced the use of classes and inheritance and how it could be applied within the scenario to achieve benefits from generalisation. Caching responses were limited to cache memory on the processor by less able candidates. Better responses explained how commonly accessed components like particular graphic items could be stored in RAM to make retrieval faster. The best responses evaluated at a level where the effectiveness of caching was measured in terms of hit ratios versus the extra development in terms of time and complexity were weighed. As with the other level of response questions candidates achieved higher scores for demonstrating critical thinking with relevant examples evaluated in depth.
			Total	40	