Question			Answer/Indicative content	Marks	Guidance
1	a	i	The array/data must be in order/sorted	1	Examiner's Comments Most candidates successfully identified that data must be ordered as a precondition for a binary search. A few candidates were too vague giving unqualified answers such as 'must be sorted', without specifying what had to be sorted.
		ii	1 mark per bullet • Compare the search item with the first value • ….then compare the search item with the next value • ….repeat the above process until either • ….the end of the array has been reached or • ….the search item is found and then stop • ….then return the array position // return -1 / False if not found	4	Not all mark points are dependent, but points awarded must following logically in sequence. Examiner's Comments Many candidates scored some marks for describing the steps involved in a linear search, but relatively few presented a comprehensive and detailed description for full marks. Many responses used vague language such 'checking if the value is found' without explaining that a comparison between the current term and the target search value must be performed. Other examples of vague language use included responses such as 'keep going until you find what you're looking for', which begged the question, how do you know when you've found what you're looking for? Common errors included candidates who mistakenly described a binary search, and those who did not answer the question. Examples of not answering the question included giving properties of a linear search such as its linear run time or the fact that it can be run on an unordered data set.
	b	i	1 mark for each variable • `contents` • `count` • `numberOfWords` • `words / words[]`	2	Accept exact spelling only Do not award numberOfWords if there are obvious spaces in 'number of Words'. It must be a valid identifier. Examiner's Comments The majority of candidates answered correctly, with the most popular answers being `count` and `contents`. A few candidates incorrectly gave data values from the array rather than identifying variables in the function.

Question			Answer/Indicative content	Marks	Guidance
		ii	1 mark per bullet • By reference the function receives the memory location of the data • By value the function receives a copy of the variable • By reference will make changes to the original variable • By value will make changes to the copy of the variable • By reference will overwrite data in the original variable • By value will not overwrite the data in the original variable • By reference will keep the changes after the function ends • By value will not keep the changes after the function ends	2	Must cover byVal and byRef for 2 marks to be awarded. Must be clear that byVal is a copy of the original value. Examiner's Comments It was pleasing to see an improvement in responses to this topic this session. There were still some answers that were too vague that did not specify that by *value* uses a copy of the parameter and that by *reference* passes the memory address. Other examples of vagueness that were not given marks included answers such as 'by value can't change the value while by reference can' that did not qualify the scope within which changes can be made. A variable passed by value can clearly be changed in a function, but it is the local copy that is changed and then disregarded when the function finishes.

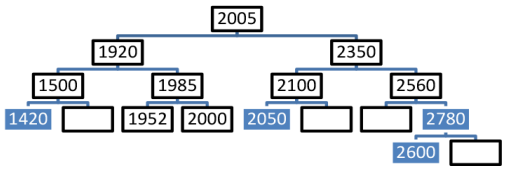
Question			Answer/Indicative content	Marks	Guidance

		iii	1 mark per bullet
			
			• initialising a counter
			• looping between 0 and numberOfWords -1
			• incrementing counter inside loop
			• remainder of algorithm correct (initialisation, concatenation and return)
			
			e.g.
			`contents = ""`
			`count = 0`
			`while count < numberOfWords`
			`  contents = contents + words[count] + " "`
			`  count = count + 1`
			`endwhile`
			`return contents`
				4	Accept:
					while count <= numberOfWords - 1
					
					Accept other combinations for example counting from 1 and then subtracting 1 for the array element (but do not credit off by one errors)
					
					Accept:
					len(words) for numberOfWords
					
					Examiner's Comments
					
					Many candidates struggled with this question. Some candidates rewrote the `for` loop putting the word *while* in place of *for* showing little understanding of the difference between a counter-controlled and a conditional loop.
					
					Common errors included not initialising the `count` variable before using it within the body of the while loop, off-by-one errors, and forgetting to increment the `count` variable within the loop. Poor indentation was often a problem, and a number of candidates erroneously placed the return statement inside the body of the loop.
					
					Off-by-One errors
					There were many off-by-one errors observed, e.g. `while count <= numberOfWords` rather than `while count < numberOfWords`.
					
					Candidates need to give code that is logically accurate, and, in this instance, it required the loop to run the correct number of times so that all the words in the array were processed.

Question			Answer/Indicative content	Marks	Guidance
	c		1 mark for benefit, 1 mark for drawback e.g. Benefits: • Variable doesn't need passing as a parameter (byref) • You don't need to return a value • Can be accessed from any function / anywhere in the program Drawback: • Increases memory usage (as it is used until full program execution is over) • Alterations within the function may have unwanted side effects elsewhere in the program.	2	<u>Examiner's Comments</u> Many candidates found it easier to describe a benefit than to give a drawback. The most commonly identified benefit was that the array would have global scope (and would therefore not need to be passed as a parameter), but often the descriptions given were too vague, e.g. 'can be accessed anywhere'. The correct technical vocabulary is required. Drawbacks were poorly described. Potential side effects and resultant complexity debugging were frequently alluded to as 'accidental change' but not fully developed into complete qualified points. There was also a frequent misconception that you cannot have multiple variables with the same name, which is not true. When a local variable is declared with the same name as a global variable that already exists, it takes precedence within the local scope.

Question			Answer/Indicative content	Marks	Guidance
	d		1 mark per identification 1 mark for expansion, max 2 each. Write: e.g. • Auto-complete • Start typing an identifier/command and it fills in the rest • Auto-indent • Indents code automatically within structures to avoid errors • Coloured command words // pretty printing // syntax highlighting • Shows which commands are correct // help identify key elements Test e.g. • Breakpoints • Stop the program running at a set point to check variables • Variable watch window • Display the values of the variables while the program is run • Stepping • Run one line at a time and check variables Unit Testing • Automated tests to be run to check changes ensure changes haven't introduced errors.	4	Examiner's Comments Most candidates achieved at least 2 marks by identifying two suitable features, but often the expansions to describe each feature were less clear. Developments in modern IDEs meant that there were many valid features, but auto-completion and breakpoints were particularly common answers.

Question			Answer/Indicative content	Marks	Guidance
	e		1 mark per bullet to max 2 • Saves time from having to write the same algorithm repeatedly • Reduced testing requirements • Can be taken and used in different programs as well as the program they are written in // can be used in a program library	2	Allow other suitable answers <u>Examiner's Comments</u> Many candidates scored 1 mark, but fewer gave two clear benefits. Unqualified statements that were too vague were often given. A frequent response was 'saves time', but as an unqualified point it did not state how time was saved, e.g. 'saves time as you don't have to retype the same code/routine again'. Candidates need to be reminded that at this level responses of a 'quicker' or 'easier' nature will not gain marks unless qualified. Exemplar 1 1 Saves time, as code has already been tested and written so does not need to be tested and written again (when it is reused). 2 Easier to debug and test, as each component can be tested individually ✓ (against its own well-defined requirements). Candidates are expected to fully qualify the points that they make. This response shows clear identification of points followed by qualifying statements.
			Total	21	

Question			Answer/Indicative content	Marks	Guidance
2	a		2005	1	Examiner's Comments Most candidates correctly identified the root node as 2005.
	b		1 mark for each to max 2 • 1500 • 1952 • 2000 • 2100 • 2560	2	Examiner's Comments Most candidates correctly identified valid leaf nodes, but a significant number erroneously gave 1920 and 2350 as the child nodes of the root instead of identifying leaf nodes.
	c		1 mark for each in the correct place • 1420 • 2050 • 2780 • 2600  	4	Examiner's Comments Many candidates scored at least 2 marks, but many erroneously inserted 2600 before 2780, or just put 2600 as the left child node of 2560. A common mistake was to use straight lines for new child nodes rather than clearly indicating whether the new child node was a left/right child of the parent node.
	d		**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of search traversals; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated*. **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of search traversals; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is	9 AO1.1 (2) AO1.2 (2) AO2.1 (2) AO3.3 (3)	**AO1: Knowledge and Understanding** **Indicative content** • Breadth first takes first value then visits all nodes connected to it. It then takes all nodes connected to those nodes. • Depth first goes to the left node, this becomes a new tree. It continues going to the left until a leaf. It then returns this, then goes right and repeats from the start. Follow left, follow right, take root. **AO2: Application** • Breadth will return 2005 1920 2350 1500 1985 2100 2560 (1420) 1952 2000 (2050) (2780) (2600) • Post-order / Depth will return (1420) 1500 1952 2000 1985 1920 (2050) 2100 (2600) (2780) 2560 2350 2005 **AO3: Evaluation** **Evaluations may vary and include one or more of the following points:** • Breadth is more efficient when the data searched for is closer to the root. • Depth is more efficient when data to

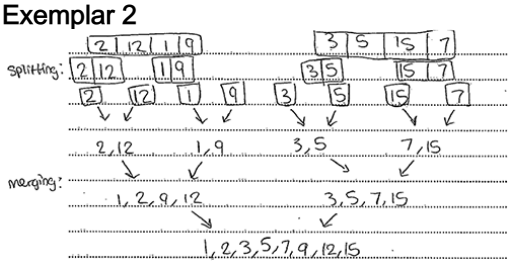
Question			Answer/Indicative content	Marks	Guidance
			focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence*. Mark Band 1 – Low Level (1-3 marks) The candidate demonstrates a basic knowledge of search traversals with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear*. 0 marks No attempt to answer the question or response is not worthy of credit.		be search for is further down. • Depth memory requirement is linear • Depth can be written recursively to aid understanding. • Breadth in general is better time complexity • In large trees depth may never return a value Candidates are not expected to know the complexities for the search traversals, however credit should be awarded if candidates choose to include these. Limit to band 2 if there is no evaluation of BFS/DFS <u>Examiner's Comments</u> Many candidates achieved some marks, but a few did confuse Breadth-first search (BFS) and Depth-first search (DFS), getting them the wrong way round. Most had a much clearer understanding of BFS than DFS and were able to score marks in Level 1 for showing how a BFS would be carried out. Fewer managed to accurately show how a DFS would work. While there was an understanding that DFS would traverse leftward to the lowest leftmost node, descriptions of back tracking were not given as many marks. A common mistake made with DFS was to output the nodes in the order they were first visited rather than the order in which they are output. Few could describe the mechanics of the algorithms with BFS using a queue and DFS using a stack. Those who did achieved the top of Level 2. Very few candidates could give exemplar uses of BFS or DFS (e.g. deleting nodes in a tree) or compare cases where they might be used (e.g. distance of target node(s) from root), with very little evaluation. This meant there were very few Level 3 responses seen.
			Total	16	

Question			Answer/Indicative content	Marks	Guidance
3	a		1 mark for: • Can refer to all 50 only using one identifier // all values can be indexed in one array • The numbers can be passed as a single parameter • Does not need 50 variables to be declared/passed	1	Examiner's Comments The concept of having a single indexable identifier seemed to be poorly understood with few candidates gaining the mark. While many candidates understood that an array can be iterated through, they did not link this to the fact that elements in an array are indexable. Similarly, unqualified points such as 'an array is easier to sort' were not given marks for the same reason. There was a common misconception that an array would use far less memory space than 50 separate variables. There would be minimal difference since each separate value would take the same amount of storage space as the corresponding value in an array. The only overhead for separate variables would be the extra pointers to the memory locations for each, compared to one pointer for an array.

Question			Answer/Indicative content	Marks	Guidance

	b		1 mark for each completed statement
			
			`arrayLength = 50 //`
			`numberArray.length`
			`tempValue =0`
			`do`
			`  flag = false`
			`  for y = 0 to arrayLength - 2`
			`  if numberArray[y] > numberArray[y + 1] then`
			`  tempValue = numberArray[y]`
			`  numberArray[y] = numberArray[y + 1]`
			`  numberArray[y + 1] = tempValue`
			`  flag = true`
			`  endif`
			`  next y`
			`until flag == false`
				5	Note if numberArray - 1 // 49 used, then for loop for y will need to be 0 to arrayLength - 1
					
					Allow other suitable valid identifier in place of tempValue e.g. temp
					
					Examiner's Comments
					
					Many candidates scored at least 1 mark for correctly initialising the `arrayLength`, although some erroneous `length.arrayLength` rather than `arrayLength.length` answers were seen. If candidates are assuming the existence of inbuilt methods they should reference them in the correct OOP way.
					
					Very few candidates achieved the second marking point for the loop. A very common off-by-one error was seen with the value 1 given. If `arrayLength` was set to 50 this would cause a run time error for an out-of-bounds reference when the loop ran.
					
					Some candidates tried to swap the array values directly in a Pythonic style that was not suitable for a pseudocode solution in context, and some used incorrectly formatted variable identifiers.
					
					Variable identifiers
					Valid identifiers must be single words. In a number of instances, it was clear that `tempValue` was given as `temp Value`. Where candidates give an answer that clearly has spaces within an intended identifier name no marks will be given.

	c		1 mark for each stage shown • Splitting into individual items 2 12 1 9 3 5 15 7 • Combining in pairs 2 12 1 9 3 5 7 15 • Merge pairs 1 2 9 12 3 5 7 15 Merge for final 1 2 3 5 7 9 12 15	4	Do not award a mark for the final stage, unless candidate has shown the previous sorting stage(s). Examiner's Comments Many candidates presented very clear solutions that used the values in `numberArray` to construct clearly annotated diagrams. Textual prose responses tended to either not refer at all to the given data set, or mentioned it only partially, so lost marks. There was occasional confusion with other sorting algorithms, but this was seen relatively

Question	Answer/Indicative content	Marks	Guidance
			infrequently, Where errors were made candidates did not split the data to the atomised level. Many did not fully understand how a merge sort works by merging two separate lists of ordered values together but performed in-place sorts in sub-lists. **Misconception** The merge phase in a merge sort takes two separate lists that are already ordered. Values in those two separate lists are taken and merged in sequence to form a new list. E.g. List 1 [1, 9] and List 2 [2, 10] are taken and merged 1 from List 1, 2 from List 2, 9 from List 1 and then 10 from List 2, into a new List. Values in lists [1, 9] and [2, 10] are not placed in a list [1, 9, 2, 10] and then sorted in-situ. A significant number of candidates thought that a merge sort split lists up until values were in pairs and showed [15, 7] going directly to [7, 15] (i.e. an in-situ sort) without first being atomised into two separate lists of [15] and [7] and then being merged to give [7, 15]. **Exemplar 2**  Candidate responses to questions that require 'showing' how data sets are ordered by sorting algorithms are best tackled by using clearly annotated diagrams such as this exemplar response.

Question	Answer/Indicative content	Marks	Guidance
d	**Mark Band 3 – High level** **(9-12 marks)** The candidate demonstrates a thorough knowledge and understanding of sorting algorithms; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated*. **Mark Band 2 – Mid level** **(5-8 marks)** The candidate demonstrates reasonable knowledge and understanding of sorting algorithms; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence*. **Mark Band 1 – Low Level** **(1-4 marks)** The candidate demonstrates a basic knowledge of sorting algorithms with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be*	12 AO1.1 (3) AO1.2 (3) AO2.1 (3) AO3.3 (3)	**AO1: Knowledge and Understanding** **Indicative content** • Merge sort splits data into individual lists and merges • Insertion makes first value sorted list, then inserts each item into the sorted list • Bubble sort looks through each item in turn, number of items times **AO2: Application** • Merge uses more memory as new lists are needed. Insertion and Bubble need constant memory. • Bubble and Insertion have the best best-times, both O(n) because they run through the data once. merge sort requires a minimum number of stages so best case is longer (O(n log(n)) • Merge average is the same as best. Insertion and Bubble has average o(n^2). • Worst time merge has same has best and average because same number of stages are needed. Bubble sort and insertion all have worse O(n^2) **AO3: Evaluation** • There are a small number of elements (10) therefore a bubble sort of insertion would be better space wise because no further space is needed. • Merge would not need excessive amounts of more memory as there are only a small number of elements. • Time complexity, there is a small number of elements therefore Bubble and Insertion may be preferable. Differences are unlikely to be significant, so either would be more appropriate. **<u>Examiner's Comments</u>** Most candidates could describe the basic elements of each of the bubble, merge and insertion sort, although some had difficulty remembering insertion sort and confused it with quick sort.

Question			Answer/Indicative content	Marks	Guidance
			clear. 0 marks No attempt to answer the question or response is not worthy of credit.		Many candidates struggled to show accurate knowledge of the Big O time complexity values for best/average/worst case for the three algorithms, which is a learnt response. Some did give very good examples of where bubble and insertion sort would give best/worst case times. Very few candidates could cite space complexity, and where they did a number thought bubble/insertion were space complexity O(n) because there are n elements rather than O(1) which means constant with no extra memory overhead. There was some evaluation of the number of items being sorted in most cases. Overall, many responses were clustered in Level 2, but a pleasing number of candidates achieved Level 3 with clear and detailed descriptions, accurate Big O values, and an evaluation of the size of the data set being sorted.
			Total	22	

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>4</td><td>a</td><td>i</td><td>Max 1 mark for each definition

e.g.

<table><tr><th>Term</th><th>Definition</th></tr><tr><td>Abstraction</td><td>Removal of unnecessary components // focus on only necessary components</td></tr><tr><td>Decomposition</td><td>Breaking down a problem into subproblems</td></tr></table></td><td>2</td><td><u>Examiner's Comments</u>

While many candidates accurately recited definitions for abstraction and decomposition there were equally many that presented very vague and unqualified responses. E.g. For abstraction: 'Simplifying a problem' without specifying how it was simplified was insufficient. For decomposition: 'Breaking a problem into smaller parts' – without specifying that the 'smaller parts' are sub-problems rather than saying sequences of instructions or loops was insufficient.</td></tr>
<tr><td></td><td></td><td>ii</td><td>1 mark for each
e.g.

• Removal of visual elements such as buildings on the ground
• Simplification of controls
• Focus on important elements such as weather, height, speed</td><td>3</td><td><u>Examiner's Comments</u>

Many candidates gave detailed and relevant examples that were suitable within the context of a flight simulator. However, there were many unqualified responses that were too vague. Unqualified responses such as 'terrain' did not go far enough to explain how the terrain in the simulation would differ from reality. To gain marks candidates had to make it clear exactly how the difference identified differed between reality and the simulation.</td></tr>
<tr><td></td><td></td><td>iii</td><td>1 mark for each to max 2
e.g.

• Reduce memory requirements
• Reduce processing requirements
• Simplify the problem being solved</td><td>2</td><td><u>Examiner's Comments</u>

It was noticeable that many candidates reiterated definitions of abstraction rather than identifying the actual reasons why abstraction is used, thus scoring no marks for not answering the question. Again, there were also many vague and unqualified responses. Responses such as 'focus on key aspects' did not identify the reason why abstraction is used in the design, whereas clearing specifying 'focus on key aspects allows coding/development time to be saved' would make the reason clear.</td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance
	b		1 mark per bullet e.g. • Store data that has been used in cache/RAM in case needed again • e.g. store design of the weather/a cloud/external environment	2	Allow 2 valid examples for 2 marks Examiner's Comments Responses to caching were better than those seen in recent papers and many candidates managed to either describe the concept of caching or give an example of something that would realistically be cached. Far fewer managed to do both in detail.
			Total	9	

Question			Answer/Indicative content	Marks	Guidance
5	a	i	1 mark e.g. In directed **arcs/edges** may only go in 1 direction // in undirected **arcs/edges** can go in both directions	1	Examiner's Comments Very few candidates had a clear grasp of the relevant terminology to clearly identify that an edge/arc is given a specific direction in a directed graph. Many responses either erroneously talked about paths between nodes (too vague as there can be many routes between two nodes located in different parts of a graph) or graphs being weighted.
		ii	1 mark • More than one path is allowed in a graph • Graphs do not have a root node • Graphs can be weighted • Graphs can have loops/cycles	1	Allow answers where candidates have given the reverse. e.g. a tree does not have loops. Examiner's Comments Most candidates gave clear responses that indicated either that a tree has a root whereas a graph does not, or graphs can contain cycles whereas trees do not. There were a surprising number of candidates who thought trees were limited to a maximum of two child nodes, showing a lack of understanding of generalised tree structures with hierarchies (e.g. directory tree).
	b		1 mark e.g. • Symbols are used to represent the address • The edges represent possible connections between addresses not the actual physical routes	1	Allow other suitable answers that are in context of the problem Examiner's Comments Very few candidates were able to give suitable answers within the context of the problem. The question was asking why the graph in Fig 5 was a visualisation. Few candidates identified that it was because the letters at the nodes represented delivery addresses, while the weights on the edges represented the road distances between the addresses. Most candidates gave descriptions of visualisation in general rather than answering in context.

<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>i</td><td>
<table>
<tr><th>Node</th><th>Distance travelled</th><th>Previous node</th><th>Marking Guidance</th></tr>
<tr><td>A</td><td>0 / -</td><td>N/A / -</td><td>1 Mark</td></tr>
<tr><td>B</td><td>3</td><td>A</td><td rowspan="2">1 Mark</td></tr>
<tr><td>C</td><td>13</td><td>E</td></tr>
<tr><td>D</td><td>10</td><td>B</td><td rowspan="2">1 Mark</td></tr>
<tr><td>E</td><td>6</td><td>B</td></tr>
<tr><td>F</td><td>9</td><td>E</td><td rowspan="2">1 Mark</td></tr>
<tr><td>G</td><td>16</td><td>F</td></tr>
<tr><td>H</td><td><s>21</s> 19</td><td><s>D</s> G</td><td>1 Mark</td></tr>
</table>
Final Path = A,B,E,F,G,H, Distance = 19 (1 Mark)
</td><td>6</td><td>Order of previous nodes visited must be clear

Note that nodes in the table do not have to be given in alphabetical order by candidates

<u>Examiner's Comments</u>

Most candidates gave the final path and the total distance correctly by inspection if nothing else. All nodes are initially set to infinity, so A is updated to 0 and has a distance 0 from A as the start node and many candidates missed this. Those who gave an answer by inspection wrote ABEFGH without knowing Dijkstra's algorithm but gained some marks by giving the distances to BEFGH along the way, but distances to nodes C and D were omitted. Few candidates clearly showed that the initial calculation for the path distance to H (from D, distance 21) was later updated and overwritten with the more optimal path length from G with distance 19.</td></tr>
<tr><td></td><td></td><td>ii</td><td>1 mark per bullet
Similarity:

• Both always find the shortest route
• Both are pathfinding algorithms

Differences:

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

<u>Examiner's Comments</u>

Those candidates who scored well in c(i) frequently gained full marks for describing the similarities and differences between Dijkstra's algorithm and the A* algorithm. The most common responses were that both give the shortest path and that A* uses heuristics.</td></tr>
<tr><td></td><td>d</td><td>i</td><td>1 mark per bullet to max 1

• Simulate/model behaviour of the system (before it is) used under load
• Because it would be too expensive/unsafe/time critical to test the real system</td><td>1</td><td><u>Examiner's Comments</u>

The concept of stress testing a system by simulating heavy loads seemed to be poorly understood by most candidates. Many candidates often confused this with the general testing of a system to find bugs.</td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance
		ii	1 mark per bullet to max 2 e.g. • Test with large and small values • e.g. largest number of deliveries • e.g. largest number of possible routes • Model how well the system scales with increasing use.	2	Examiner's Comments There was much confusion over testing a system in general to determine whether an individual route was effective in the graph given, and the actual performance of the system as a whole in terms of the overall time taken to actually perform the calculations required. The majority of candidates did not appreciate that performance modelling looks at the effect of escalating loads on a system.
			Total	14	

Question			Answer/Indicative content	Marks	Guidance
6	a		1 mark for each component e.g. • Allocating cards to each player • Generating the deck • Managing whose turn it is to play • Checking won	3	Accept any reasonable component <u>Examiner's Comments</u> The specification requires candidates to be able to identify elements of computational thinking. As such, candidates are expected to be able to think procedurally and to be able to identify the components of a problem. While analysis of the problem given in context led most candidates to identify valid components, many struggled to read the scenario and to give relevant points. For instance, many reiterated aspects of checking if a move was valid, which was already given in the question.
	b	i	It returns a value	1	<u>Examiner's Comments</u> Most candidate correctly identified that `checkValid()` was a function because it returned a value. Some candidates erroneously stated 'because it has an output' which did not differentiate a function from a procedure since both can print an output as a side effect.
		ii	1 mark per bullet • If the players card is the same suit return true • if the players card is the same number return true • if neither is true, return false	3	Allow 1/0, 'T'/'F', "Yes"/"No" or any sensible alternative as return values. <u>Examiner's Comments</u> Many candidates were able to think computationally to identify the points in a solution where a decision had to be taken. Occasionally some candidates did not specify the specific result that would be returned from the function after determining if a card was valid or not. There were also occurrences of incorrect logic such as returning valid using an AND condition on the same number and same suit clauses, rather than executing them in sequence.

Question			Answer/Indicative content	Marks	Guidance
	c		1 mark per bullet • identifier `cards`… • …with 2 dimensions	2	Examiner's Comments Many candidates gained a mark by initialising the identifier `cards`, but fewer gained the second mark for correctly setting it to be a two-dimensional structure. Many obscure forms of syntax were observed, but marks was given if it was clear that the structure was two-dimensional, however, for many responses, it was clear that a one-dimensional list had been initialised.
			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>Line number 5</td><td>1</td><td><u>Examiner's Comments</u>

Nearly all candidates gave the correct answer line 5.</td></tr>
<tr><td></td><td></td><td>ii</td><td>1 mark per feature

• A function that calls itself // a function that is defined in terms of itself
• …has a base case (that terminates the recursion)</td><td>2</td><td><u>Examiner's Comments</u>

Most candidates knew that a recursive algorithm is self-referential and calls itself, but some candidates were too vague specifying that it 'calls a function'. Candidates often found it harder to gain the second mark and some gave answers not related to the question such as explanations of how recursion uses stack frames during execution. Technical vocabulary is important, and some candidates did not make it clear that recursion has a base case. Those that stated a stopping/terminating condition needed to qualify their response to say that these conditions stopped/halted the recursion. Where candidates just wrote 'stopping condition' it was too vague as it was unqualified since they could have been talking about any conditional loop.</td></tr>
<tr><td></td><td>b</td><td></td><td><table><tr><th>Function call</th><th>number</th><th>return</th><th>Marking Guidance</th></tr><tr><td>calculate(5)</td><td>5</td><td>15</td><td>1 Mark</td></tr><tr><td>calculate(4)</td><td>4</td><td>10</td><td>1 Mark</td></tr><tr><td>calculate(3)</td><td>3</td><td>6</td><td>1 Mark</td></tr><tr><td>calculate(2)</td><td>2</td><td>3</td><td>1 Mark</td></tr><tr><td>calculate(1)</td><td>1</td><td>1</td><td>1 Mark</td></tr></table></td><td>5</td><td><u>Examiner's Comments</u>

Many candidates continue to struggle with recursion as a concept and so had little idea how to trace and unwind a call to a recursive function. Some got to the last call and the base case and could return 1 or unwind one step further to gain the first 2 marks. Fewer achieved all 5 marks.</td></tr>
<tr><td></td><td>c</td><td></td><td>calculate(10)</td><td>1</td><td><u>Examiner's Comments</u>

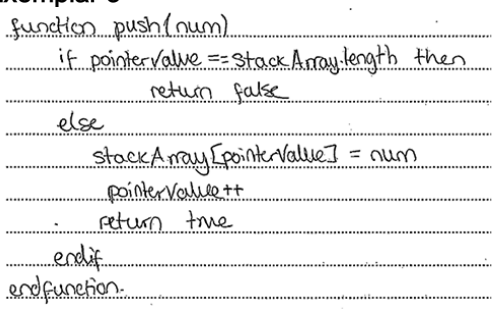
Those candidates who scored full marks in 7b had little difficulty giving the correct call calculate(10). Some candidates just wrote 10, which did not answer the question. Those who showed little understanding of recursion in 7b rarely gave the correct answer in 7c.</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>9</td><td></td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance
8	a	i	1 mark for correct data 1 mark for correc top of stack pointer pointerValue | 6 Index / Data 8 7 6 5 — 7 4 — 6 3 — 3 2 — 6 1 — 5 0 — 10	2	Examiner's Comments Most candidates scored full marks for this question, but the most common error was the omission of the `pointerValue` or the setting of the `pointerValue` to 5 to point to the last element in the stack instead of the next free location at the top of the stack.
		ii	1 mark per bullet • Point to the next free space in the array • Points to the top of the stack	1	Examiner's Comments Many candidates answered the question clearly specifying a pointer to the next free index in the stack, but some candidates gave vague responses such as 'location of next item'.

	b	i	1 mark per correctly completed statement
			
			e.g.
			`pubic function pop()`
			`  if pointerValue == 0 then`
			`  return -1`
			`  else`
			`  pointerValue = pointerValue -1`
			`  returnValue = stackArray[pointerValue]`
			`  return returnValue`
			`  endif`
			`endfunction`
				5	Examiner's Comments
					
					Most candidates gained some marks with many gaining full marks. The most common errors were writing identifiers for `pointerValue` and `returnValue` with spaces in, returning values as strings, or returning True/False instead of -1 as required in the question.

		ii	1 mark per bullet to max 6
			
			• function header
			• ..taking parameter (ignore byval/byref)
			• checking if stack is full (pointerValue at 100)…
			• …and returning false
			• (otherwise) adding value to top of stack
			• …incrementing top of stack pointer
			• return true
			
			e.g.
			`function push(value)`
			`  if pointerValue < 100 then`
			`  stackArray[pointerValue] = value`
			`  pointerValue = pointerValue + 1`
			`  return true`
			`  else`
			`  return false`
				6	Ignore additional parameters in function definition
					
					Do not accept the return of string values
					
					FT following a reasonable attempt to check if the stack is full
					
					Examiner's Comments
					
					Candidates produced a higher standard of pseudocode this session and many scored most if not full marks for a standard stack push routine.
					
					Common errors included omitting a parameter and/or getting a user input as the value to place into the stack, returning strings or printing the return values, and off-by-one errors when testing to see if the `stackPointer` was at the top of the stack to determine if the stack was full.

Question			Answer/Indicative content	Marks	Guidance

			`  endif`
			`endfunction`
					It was noticeable that a number of students who were only familiar with Python gave list append type solutions rather than using the array and pointers as per the implementation given.
					
					Another common error was incrementing the value of the stackPointer before the parameter was assigned to `stackArray` at the `stackPointer` index, which was frequently seen when candidates did not know that the `stackPointer` actually pointed to the index of the next free space in the stack.
					
					Exemplar 3
					 
					`function push(num)`
					`  if pointerValue == stackArray.length then`
					`  return false`
					`  else`
					`  stackArray[pointerValue] = num`
					`  pointerValue++`
					`  return true`
					`  endif`
					`endfunction.`
					
					Candidates are encouraged to present pseudocode solutions with clear indentation to aid readability. No specific language/syntax is expected, but the logic of the solution must be clear.

		iii	1 mark per bullet • instantiation of new object of type stack • assigned to variable mathsStack `mathsStack = new stack()`	2	**Accept** `mathsStack = stack()` allow missing brackets this time only e.g. mathsStack = stack **Examiner's Comments** Many candidates struggled to answer this question because they lacked practical experience of OOP that showed a lack of familiarity in terms of creating instances of a class. Some candidates tried to declare `mathsStack` as a procedure or class, and many got it the wrong way round and actually declared a new identifier called `stack` as an instance of the class `mathStack`.

Question			Answer/Indicative content	Marks	Guidance

		iv	1 mark for each completed statement
			
			`returnValue = true`
			`while returnValue == true`
			`  returnValue = mathsStack.push(input("Enter Number"))`
			`  if returnValue == false then`
			`  print("Stack full")`
			`  endif`
				4	Accept equivalent for print e.g. output
					
					Examiner's Comments
					
					The majority of candidates scored some marks but relatively few gained full marks. Return was seen instead of print/output when the question did, on this occasion ask for the message to be output (as the code was specified as being in the main program rather than a subroutine).

		v	mark per bullet to max 8
			
			• initialise a total to 0 outside of loop
			• looping
			• removing an item from the stack using the method pop
			• check if stack is empty
			• (if not) add value returned to total
			• …outputting total
			• counting how many values are returned
			• stopping loop when either 20 items removed or no items left
			
			`total = 0`
			`quantity = 0`
			`returnValue = 0`
			
			`while quantity<20 and retunValue!=-1`
			`  returnValue = mathsStack.pop()`
			`  if(returnValue != -1) then`
			`  quantity = quantity + 1`
			`  total = total + returnValue`
			`  print(total)`
			`  endif`
			`endwhile`
				8	Examiner's Comments
					
					The standard of pseudocode / programming code was better than in previous sessions and most candidates made a reasonable attempt to pop 20 values from the stack. Many candidates found it difficult to reference the class methods correctly or made assumptions about the existence of other methods that were not provided within the scenario (e.g. `.full(), .empty(), .length(), .remove()`) that should not be presumed to exist.
					
					Those who had little understanding of encapsulation often tried to access class attributes such as `stackPointer` directly to manipulate `mathsStack`, rather than using methods to interact with the instance of the stack.
					
					When candidates did use the `.pop()` method to retrieve a value from `mathsStack` they frequently did not store the result for later use to check whether then stack was empty.

Question			Answer/Indicative content	Marks	Guidance
	c	i	1 mark per bullet to max • Queue has head pointer and tail pointer • When an item is enqueued the tail pointer increments • When an item is dequeued the head pointer increments	3	Max 1 mark for Enqueue/Dequeue operations if description of effect on tail/head pointers not given <u>Examiner's Comments</u> Many candidates identified the need to have a head/tail pointer but struggled to gain more than 1 mark by expanding on how enqueue and dequeue operations would be implemented. Some candidates continued to talk about push/pop operations for a queue rather than enqueue/dequeue and often gave properties of a queue in general such as First In First Out rather than answering the question.
		ii	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of object-oriented and procedural programming; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated*. **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of object-oriented and procedural programming; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with*	9 AO1.1 (2) AO1.2 (2) AO2.1 (2) AO3.3 (3)	**AO1: Knowledge and Understanding** **Indicative content** • OOP defines an object as an independent entity • OOP defines the attributes of the object and the methods that can be applied to it • attributes could be private to restrict accidental changes • Procedural the statements are executed in the order they are written **AO2: Application** • OOP allows for an object to be created from the queue • Many instances of this queue can then be declared in the main program. • Procedural will need each queue to be declared individually • Procedural will need to make use of subroutines where the queue will need to be sent and returned each time. **AO3: Evaluation** • OOP you can create multiple instances of the queue as required by the program without having to re-write all of the declarations etc. In procedural you would have to write separate code for each new stack • OOP reduces amount of code needed therefore fewer errors are likely as code is written once and then used

Question			Answer/Indicative content	Marks	Guidance
			some structure. The information presented is in the most part relevant and supported by some evidence. Mark Band 1 – Low Level (1-3 marks) The candidate demonstrates a basic knowledge of object-oriented and procedural programming with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear*. 0 marks No attempt to answer the question or response is not worthy of credit.		multiple times • OOP can reduce mistakes because the subroutines are self-contained in procedural it would need to make sure the correct values are passed and returned, or global variables may be required which uses excess memory. <u>Examiner's Comments</u> Many candidates were able to identify some elements of OOP and Procedural programming to achieve a Level 1 response or were able to describe features in detail for a Level 2 response. Far fewer were able to apply this to the specific context to achieve a Level 3 response. Those with good knowledge of OOP stood out in terms of giving clear evaluations of multiple queues generated from instance of the class, encapsulation to reduce side effects and possibilities for inheritance for different types of queues. Many did not describe the necessary creation of enqueue and dequeue subroutines in procedural programming for each separate queue or the need to pass queues to or returning queues from subroutines. Misconception There was a lot of confusion between inheritance and instantiation, e.g. "when creating several queues you can use inheritance, so all queues inherit attributes and methods". Candidates need to be clear that each instance of a class is assigned the attributes of the class and has access to all associated methods.
			Total	40	