1(a) **Fig. 8** shows a binary search tree that contains the names of different towns in Ireland.

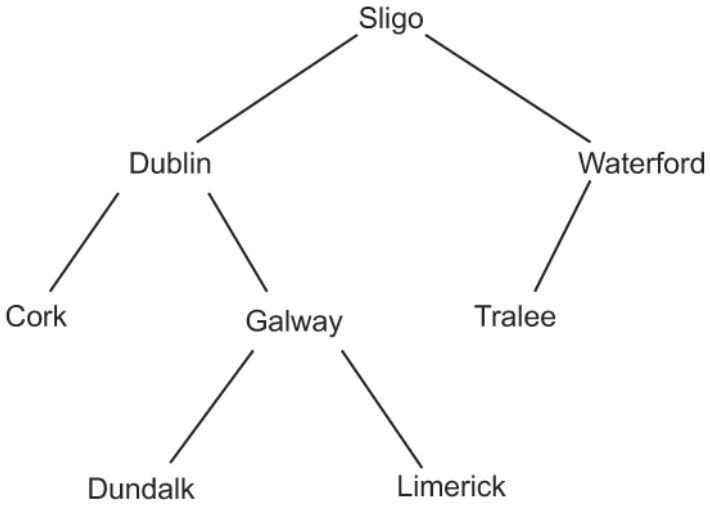


Fig. 8

The binary search tree is held in a 2-dimensional array called `towns` with 8 rows and 3 columns.

Write a line of program code or pseudocode to declare the array `towns`.

[2]

(b)

i. In the 2-dimensional array `towns`:

- the **first** column contains a pointer to the left side
- the **second** column contains the data
- the **third** column contains a pointer to the right side.

Leaf nodes have the pointer `null`.

Complete the table showing the contents of the `towns` array to store the binary search tree shown in **Fig. 8**.

	Left	Data	Right
0		Sligo	
1		Dublin	
2		Cork	
3		Waterford	
4		Galway	
5		Limerick	
6		Tralee	
7		Dundalk	

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

[4]

ii. **Four** more towns are added to the binary search tree shown in **Fig. 8** in this order:

Mallow

Cavan

Tuam

Wexford

Complete this binary search tree by adding the new towns to it.

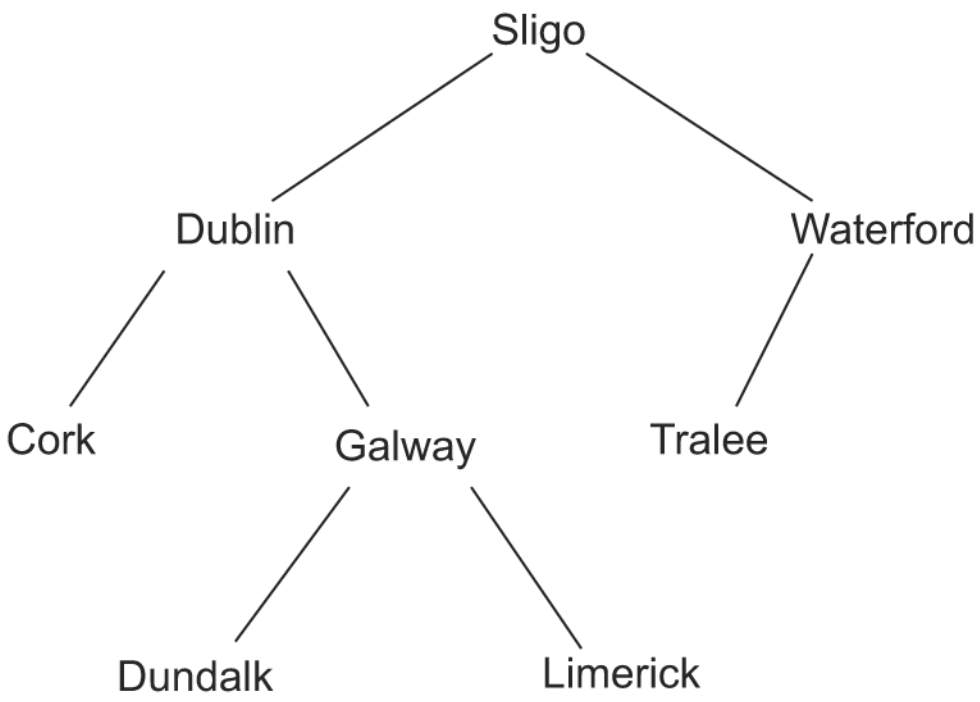


Fig. 8

[4]

2(a) Sundip writes an algorithm to carry out addition and subtraction. The algorithm will use an initially empty stack with the identifier `numbers` and will take input from the user.

The action the algorithm takes depends on the value input by the user. These actions are listed in **Fig. 2**.

Value input	Action to take
A	• Pop two values from the `numbers` stack • Add the two values • Push the result back onto the `numbers` stack
S	• Pop two values from the `numbers` stack • Subtract the first popped value from the second • Push the result back onto the `numbers` stack
E	• Pop one value from the `numbers` stack • Output this value • End program
Any other value	• Push the input value to the `numbers` stack

Fig. 2

Complete the pseudocode here to implement Sundip's algorithm.

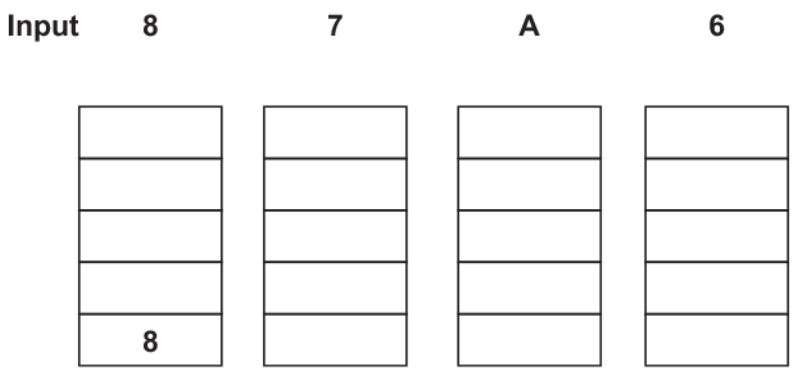
```
do
    value = input("Enter a value")
    if …… then
        num = numbers.pop()
        print(num)
    elseif value == "A" or …… then
        numone = numbers.pop()
        numtwo = numbers.pop()
        if value == "A" then
            numbers.push……
        elseif value == "S" then
            numbers.push(numtwo – numone)
        endif
    else
        numbers.push(……)
    endif
until value == ……
```

[5]

(b)

i. Complete the diagram to show the state of the stack after each value is entered into the algorithm. The letters will complete an action stated in **Fig. 2**.

The state of the stack after the first value, 8, has been completed for you.



[3]

ii. Complete the following table to give the output from this algorithm when the following set of inputs are entered by the user. The letters will complete an action stated in **Fig. 2**.

Input data (from left to right)	Output
9 3 A E	
10 5 A 8 S E	
25 5 S 2 3 A S E	

[3]

iii. If the user enters **4 2 S A E** , the algorithm will not work correctly.

Explain what problem this input data will cause and why the problem occurs.

---

[3]

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

(c) A stack is one data structure that is available for Sundip to use. She could also use a queue, list, linked list, array or tuple.

i. Describe **one** difference between a stack and a queue.

[2]

ii. Describe **one** difference between an array and a list.

[2]

iii. State **how a** tuple is different to a list.

[1]

iv. Describe how the **second** item in a linked list would be accessed using pointer values.

[3]

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

3 Describe how an array can be used to store and access data in a stack data structure.

[4]

4(a) A programmer uses a queue data structure to store data.

i. Tick **one** box that describes how a queue operates.

☐ Last In First Out

☐ First In First Out

[1]

ii. The figure below shows a queue data structure that contains a list of names. Alex is at the front of the queue.

Alex	Kofi	Ben	Sundip	Tom			

The operations that can be used on the queue are:

- `enqueue()` – This will add data that is passed in as a parameter to the queue.
- `dequeue()` – This will return the first element in the queue.

Show the contents of the queue after these operations have been performed:

```
enqueue("Charlie")
dequeue()
enqueue("Ling")
dequeue()
enqueue("Sara")
```


[2]

(b) A stack is a type of data structure.

A stack is implemented using these variables:

- `items` – This is used to store an array that contains the data.
- `top` – This is an integer value pointing to the last item of data that was inserted.

`pop()` is one operation that can be performed on a stack. This will remove an item from the top of the stack, or –1 if the stack is empty.

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

i. Complete the pseudocode function for the `pop()` operation.

```
function pop()
  if top == …… then
     return –1
  else
     item = items[……]
     top = top – ……
     return ……
  endif
endfunction
```

[4]

ii. A function called `reverse` uses a stack called `theStack` to reverse data that is passed in as a parameter called `name`. For example, the `name` "Jack" would be returned as "kcaJ" by the function.

`theStack` uses these operations which are already defined as global scope in the program:

- `push()` – This will add data that is passed in as a parameter to the stack.
- `pop()` – This will remove and return the item on top of the stack.

Write the function `reverse` so that it:

- accepts the `name` as a parameter
- uses `push()` to add each character in the `name` to `theStack` separately
- uses `pop()` to return each character from theStack and add it to a variable called `reverseName`
- outputs the variable `reverseName` once all characters have been popped from `theStack`.

You should write your function using pseudocode or program code.

---

[7]

5 **Fig. 5** shows a graph data structure representing a small section of a parcel delivery network. Each node represents an address where deliveries need to be made. The edges show the possible routes and distances between these deliveries.

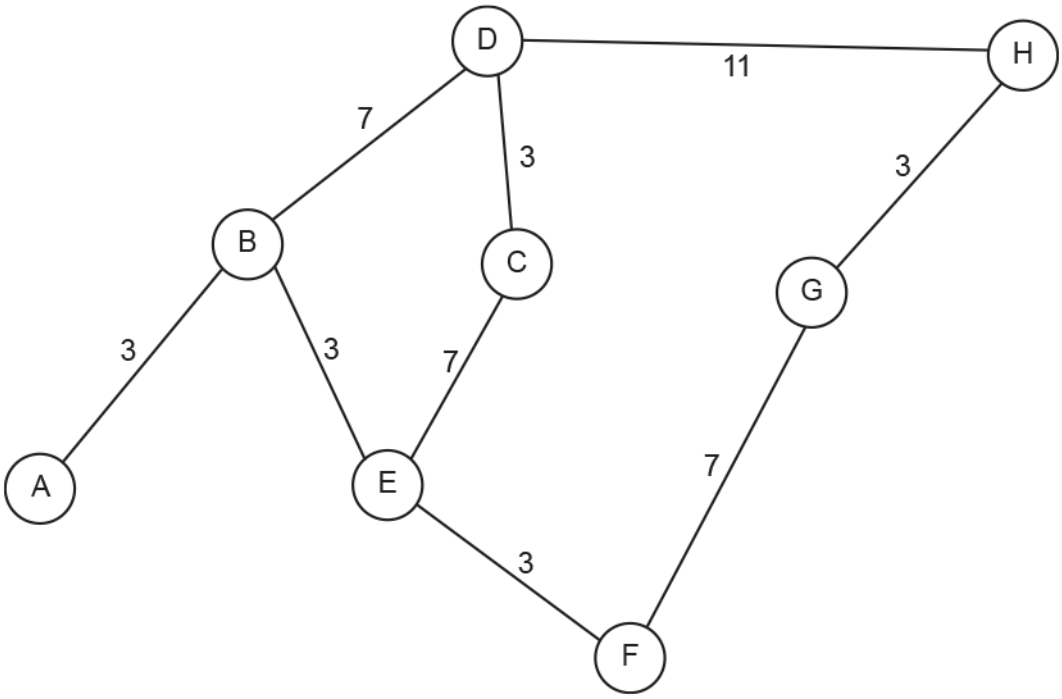


Fig. 5

i. State **one** way a directed graph is different to an undirected graph.

---

[1]

ii. State **one** way a graph data structure is different to a tree data structure.

---

[1]

6 Give **one** benefit and **one** drawback of declaring an array as a global variable instead of a local variable.

Benefit ---

Drawback ---

---

[2]

7(a) A veterinary surgery uses a two dimensional array to store bookings for customers to bring in their animal to see the vet. There are ten possible booking slots during each day.

An example of the two dimensional array is shown in **Fig. 1**.

- The first column stores the booking slot number, ranging between 1 and 10.
- The second column stores the time of the appointment.
- The third column stores the customerID of the customer who has booked that slot.

1	9:00	5877RC
2	9:30	9655AS
3	10:00	
4	10:30	8754TT
5	11:00	
6	11:30	8745SD
7	13:00	9635GH
8	13:30	
9	14:00	9874PL
10	14:30	9658SV

Fig. 1

If a customerID has been entered for a booking slot then the booking slot has been taken. If no customerID has been entered then the booking slot is available for booking.

A function `findFirst`, is used to find the first available appointment.

Write the function `findFirst` that will find the first available appointment and return the booking slot number. If no appointments are available then the function should return "`-1`".

You should write your function using pseudocode or program code.

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

[7]

(b) When customers make an appointment they often ask for the first available booking slot.

Describe how a linear search could be used for this purpose.

[3]

8(a)

A business uses an array with the identifier `wNames` to store workers' names. A variable with the identifier top is used to store the index of the last element to be added to the array, which is also the element which will next be removed.

`wNames`

0	1	2	3	4	5	6
Kirstie	Martyn	Louise	Alex	Anna		

`top`

4

i. State the name of the type of data structure described above.

---

[1]

ii. Using pseudocode, write an algorithm that allows the user to enter a name which is then pushed onto the data structure above, checking first that the data structure is not full.

---

[4]

(b) The same workers' names are stored in a binary search tree which is ordered alphabetically.

Kirstie is set as the root node, with Martyn, Louise, Alex and Anna added one by one.



i. Complete the tree diagram above to show where Martyn, Louise, Alex and Anna would be added to this binary search tree.

[4]

ii. Describe the process of using the binary search tree above to search for the name "Zoe".

---

[3]

iii. Compare the efficiency of a binary search tree to a linked list when searching for data.

---

[2]

© OCR 2026. You may photocopy this page.

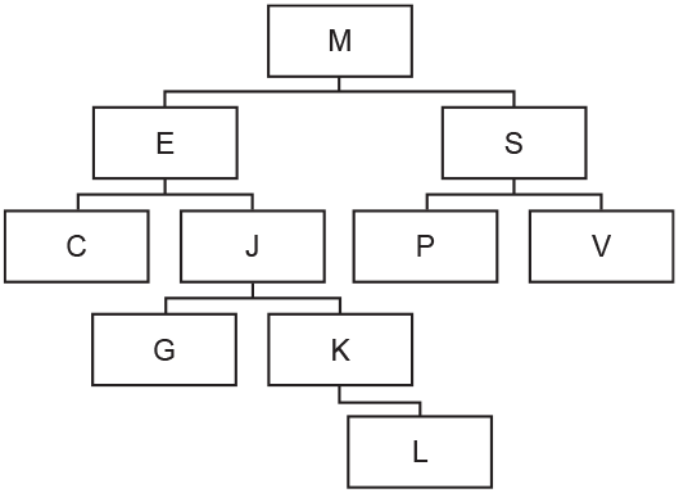
Created in ExamBuilder

iv. Compare the efficiency of a binary search tree to a hash table when searching for data.

[2]

9(a) A breadth-first traversal can be performed on both a tree and a graph.

Show how a breadth-first traversal is performed on the following binary tree.



[6]

(b) Some of the characters in the game will move and interact independently. Taylor is going to use graphs to plan the movements that each character can take within the game.

DancerGold is one character. The graph shown in **Fig. 1** shows the possible movements that DancerGold can make.

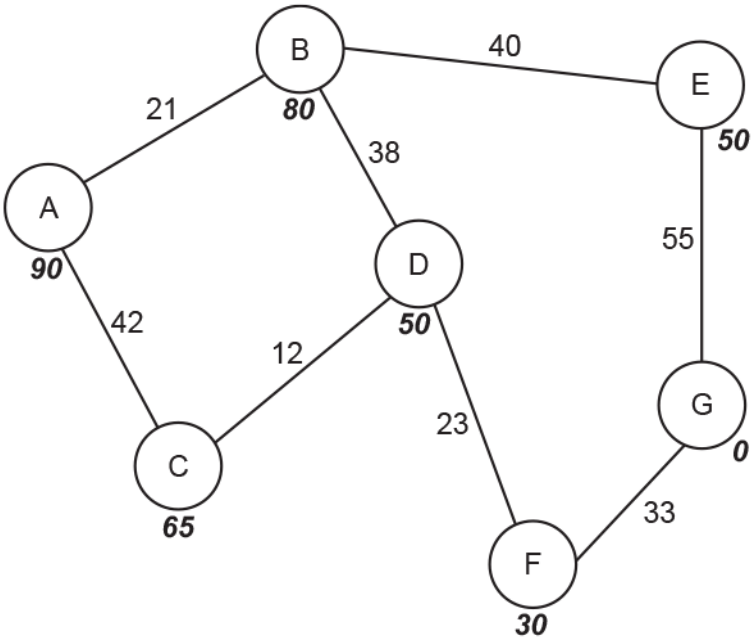


Fig. 1

DancerGold's starting state is represented by node A. DancerGold can take any of the paths to reach the end state represented by node G.

The number on each path represents the number of seconds each movement takes.

The number in bold below each node is the heuristic value from A.

i. Define the term heuristic in relation to the A* algorithm.

---

[2]

ii. Perform an A* algorithm on the graph shown in **Fig. 1** to find the shortest path from the starting node to the end node. Show your working, the nodes visited and the distance. You may choose to use the table below to give your answer.

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Node	Distance travelled	Heuristic	Distance travelled + Heuristic	Previous node

Final path: _____

Distance: _____

[8]

10(a) Kira is creating a computer game where the user can play against the computer.

In each turn, each character can make one move from a selection of possible moves.

Kira uses a tree data structure shown in **Fig. 1** to identify the range of possible moves the computer can make from starting position A. Each connection is a move, with each node representing the result of the move.

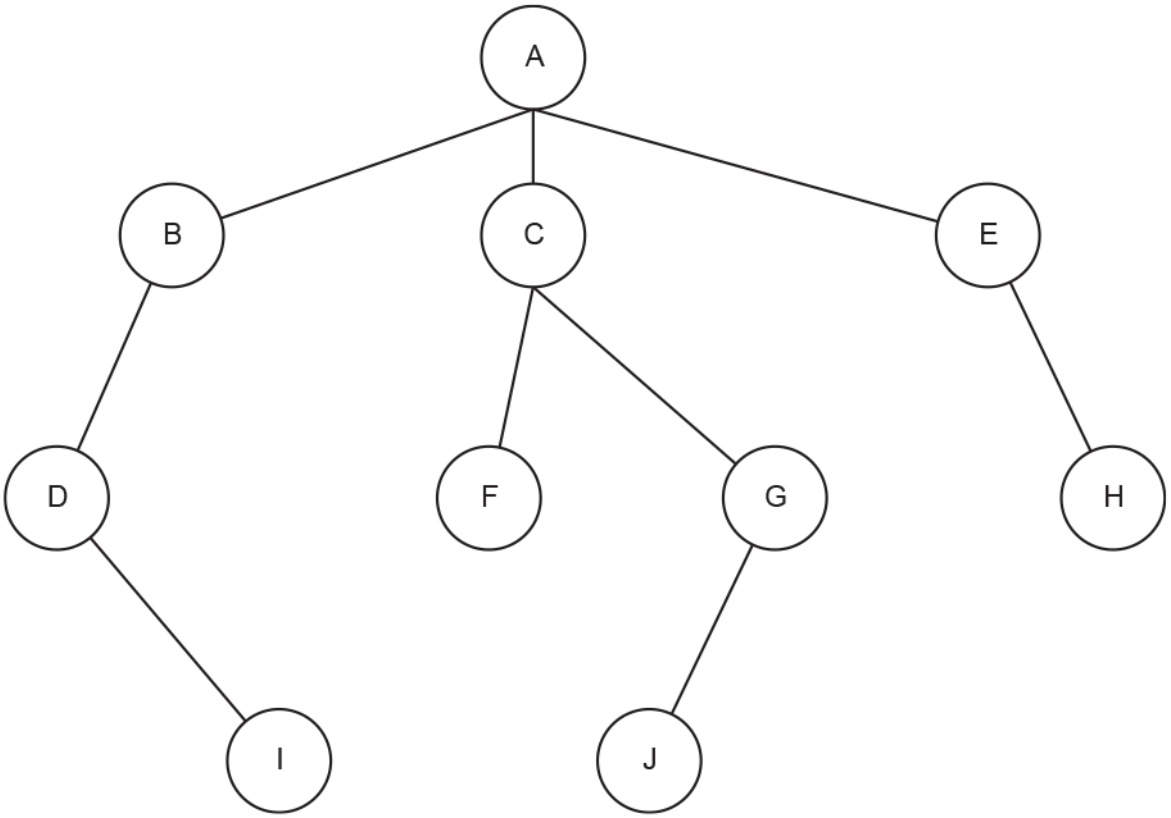


Fig. 1

Kira wants to make some changes to the data that is stored in the tree structure shown in **Fig. 1**.

i. The move represented by node 'E' needs to be deleted.

Describe the steps an algorithm will follow to delete node 'E' from the tree.

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

[3]

ii. The move represented by the node 'K' needs to be added. Node 'K' needs to be joined to node 'G.'

Describe the steps the algorithm will follow to add node 'K' to the right of node 'G'.

[3]

(b) Kira could have used a graph data structure to represent the moves in her game.

Give **two** similarities and **two** differences between a tree and a graph data structure.

Similarity 1

Similarity 2

Difference 1

Difference 2

[4]

(c) State why the tree shown in **Fig. 1** is **not** an example of a binary search tree.

[1]

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

(d) State what type of pointers are used to store nodes I, F, J and H so they do not point to any other nodes.

---

[1]

(e) Kira wants the program to traverse the tree to evaluate the range of possible moves. She is considering using a breadth-first traversal or a depth-first (post-order) traversal.

Show how a breadth-first traversal would traverse the tree shown in **Fig. 1**.

---

[4]

11(a) A programmer is writing software for a firewall. She is writing code so that it keeps a track of websites that users are permitted to visit. The software stores the websites' addresses along with details about who can view them and when.

The following data is also stored about each website:

- Access level needed (`1-4`)
- If it is available all the time (`true`) or just lunch times and out of work hours (`false`).

So a website which is available to users of access level 2 and above, all the time, would have the details `[2, true]` stored.

A website accessible to users of access level 3 and above, only outside of work hours, would have the details `[3, false]` stored.

A flaw with the current hash function is it tends to generate lots of collisions (addresses that compute to the same hash). Below is a diagram of part of the hash table. The address www.rnd.com with details `[2, true]` is being added to the hash table.

Explain how a hash table can be used to handle collisions, referring to the example below.

227	
228	www.ocr.org.uk : [1, true]
229	
230	www.ppf.nz : [2, false]
231	
232	www.ntf.biz : [4, true]
234	
235	

---

[4]

(b) The hash function is changed so there are no longer high numbers of collisions.

During busy periods the firewall is expected to check several addresses a second. It is anticipated that roughly 10 new addresses will be added to a whitelist (list of acceptable addresses) each day.

There is a debate as to whether a hash table (with the new hash function) is the best approach, or if the whitelist would be better stored in a linked list.

* Discuss whether a hash table or linked list is better to store acceptable websites. You should compare how each structure can be searched and has data added and come to a recommendation as to which is better for the whitelist.

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

[12]

(c) State the name of a data structure that could be used to store a single site's details.

[1]

12 Stacks and queues are both data structures.

State which of a stack or queue would be considered as a 'First In First Out' data structure.

[1]

END OF QUESTION PAPER

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>1</td><td>a</td><td></td><td>

- Typed as array/string and set to the variable <code>towns</code>
- 2 dimensions set to 8 and 3 **in brackets**

Examples

<pre>array towns(8,3)
string towns[][] = new string [8][3]
array towns [8][3] as string</pre>

</td><td>2</td><td>

For MP1 accept any suitable attempt at typing as an array, for example <code>towns = [], towns( )</code>

For MP2 accept any **suitable** attempt to set to 2 dimensions of the correct size

Allow 7, 2 as array index

Allow column major notation
i.e. <code>array towns [3,8]</code>

<u>Examiner's Comments</u>

Many candidates were able to gain full marks with most being able to gain one. Many candidates did not specify the number of elements in the array which was given in the question and tended to give responses such as towns = [ ] and as such only gained one mark.

</td></tr>
<tr><td></td><td>b</td><td>i</td><td>

<table>
<tr><th></th><th>Left</th><th>Data</th><th>Right</th><th>Marking Guidance</th></tr>
<tr><td>0</td><td>1</td><td>Sligo</td><td>3</td><td rowspan="2">1 Mark</td></tr>
<tr><td>1</td><td>2</td><td>Dublin</td><td>4</td></tr>
<tr><td>2</td><td>null</td><td>Cork</td><td>null</td><td rowspan="2">1 Mark</td></tr>
<tr><td>3</td><td>6</td><td>Waterford</td><td>null</td></tr>
<tr><td>4</td><td>7</td><td>Galway</td><td>5</td><td rowspan="2">1 Mark</td></tr>
<tr><td>5</td><td>null</td><td>Limerick</td><td>null</td></tr>
<tr><td>6</td><td>null</td><td>Tralee</td><td>null</td><td rowspan="2">1 Mark</td></tr>
<tr><td>7</td><td>null</td><td>Dundalk</td><td>null</td></tr>
</table>

</td><td>4</td><td>

Allow -1 or Ø for <code>null</code>

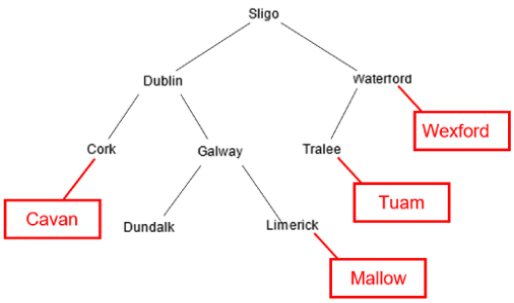
FT for 0/blank used as null **after** first error.

<u>Examiner's Comments</u>

The question states that the first and third column contain a pointer and the second contains the data. The biggest error by far was candidates putting data in the first and third column instead of a pointer which meant that many only gained a mark for having null in the last two rows. More successful candidates were able to gain full marks.

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

1 mark for each correct town.



</td><td>4</td><td>

If same town added more than once BOD if one in the right place

Ignore basic spelling mistakes and case

Towns must be to left or right of the node above - DNA nodes straight below.

<u>Examiner's Comments</u>

This was generally well answered and many candidates gained full marks with many candidates writing out the alphabet beneath the question to aid then in placing the new towns.

</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>10</td><td></td></tr>
</table>

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance

2	a		• value == "E"
			• value == "S"
			• (numone+numtwo)
			• value
			• "E"
				5	All string values must be in quotes. Allow single or double quotes.
					
					Don't allow single = for MP1&2. Penalise once and FT for multiple occurrences.
					
					Case needs to match that used in the question
					
					Needs to have brackets for MP3
					
					do
					value = input("Enter a value")
					if value == "E" then
					num = numbers.pop()
					print(num)
					elseif value == "A" or value == "S" then
					numone = numbers.pop()
					numtwo = numbers.pop()
					if value == "A" then
					numbers.push(numone + numtwo)
					elseif value == "S" then
					numbers.push(numtwo - numone)
					endif
					else
					numbers.push(value)
					endif
					until value == "E"
					
					Examiner's Comments
					
					Generally well answered with most candidates being able to gain between 2 and 5 marks on this question.

	b	i	• 8, 7 • 15 • 15,6	3	One mark per stack diagram Examiner's Comments This question was generally well answered although some candidates struggled to understand the concept of a stack and how data is pushed on to it and popped from it.
		ii	• 12 • 7 • 15	3	Examiner's Comments This was generally well answered, with the biggest misunderstanding being the way the subtraction occurs.

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance
		iii	• S causes the two values inputted to be popped and only one value to be pushed back // 4 and 2 are popped and 2 is pushed • A causes an attempt to pop **two values but only one present / not two values there** • Causing a **stack** underflow	3	Examiner's Comments Many candidates were able to gain at least one mark on this question for stating that the addition would only be able to pop one number. The candidates who gained full marks were able to state the type of error correctly and explain why there was only one value able to be popped after the subtraction.
	c	i	• Stack is LIFO / FILO • Queue is FIFO / LILO • Stack uses one pointer (for head) • Queue uses two pointers (head and tail) • Stack, data is popped/pushed from the top • Queue, data is dequeued from the from the startand enqueued onto the back // a queue can be circular	2	Mark in pairs Accept descriptions of LIFO / FIFO for MP1 and 2 Examiner's Comments Generally well answered with the most popular answer being that a stack is LIFO and a queue is FIFO.
		ii	• Array is of fixed/defined size // static • List size can be changed // no defined size //dynamic • Array holds data of single data type • List can hold data of multiple / different types	2	Mark in pairs Examiner's Comments Generally well answered, although some candidates confused a list with a linked list.
		iii	• A tuple cannot be changed **at runtime** // a tupleis immutable	1	Examiner's Comments Generally well answered, with a tuple being immutable being the most common answer.
		iv	• **Go to** the first position indicated by the startpointer • From the first position, **read** the next pointervalue… • …follow this pointer value and **access** the data item	3	Accept answers relating to locations given by pointers Allow acceptable diagram illustrating the same points Examiner's Comments For this question candidates were asked how the second item in a linked list would be accessed using the pointers but many just gave descriptions of a linked list having pointers and data without saying how the data in the second item would be accessed.

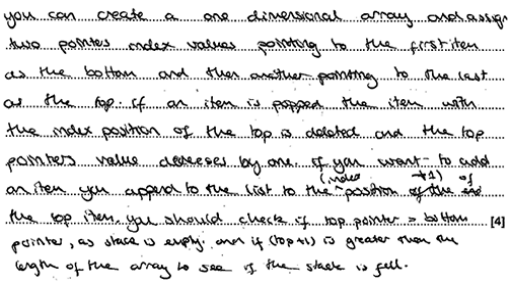
© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance
			Total	22	

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question	Answer/Indicative content	Marks	Guidance
3	1 mark per bullet to max 4 • (Stack) Pointer points to the last element added to the stack / top of the stack • New data is added to the pointer position // pointer+1… • …check for overflow condition • …pointer is then incremented • Data is removed from pointer/pointer-1 position… • …check for underflow condition • …pointer is then decremented • Elements can be accessed through Push() and Pop() methods that are implemented	4	Note: Answers must relate to an array implementation which means that a stack pointer must be implemented. <u>Examiner's Comments</u> Candidates need to be mindful of the fact that an array is a static data structure with a predetermined size. This meant that responses that explained how lists could be used to append and remove items were not given marks. This is an area where candidates who only have programming experience of using lists in Python often struggle. Implementing a stack in an array requires a stack pointer. Those candidates who appreciated this and who clearly had practical experience of modelling push()/pop() operations on a stack were able to articulate how the stack would access array locations at the stack pointer. Exemplar 3  you can create a one dimensional array and assign two pointer index values pointing to the first item as the bottom and then another pointing to the last or the top. If an item is popped the item with the index position of the top is deleted and the top pointers value decreased by one. If you want to add an item you append to the list to the position of the (index -1) of the top item. you should check if top pointer = bottom [4] pointer, as stack is empty. and if (top+1) is greater than or equal to the length of the array to see if the stack is full. This exemplar was one of the few responses to show a clear insight as to how an array can be used to store and access a stack. It makes it clear that a stack pointer is required and goes on to explain how the stack pointer is manipulated when push()/pop() operations are performed. **Misconception** Many candidates were not clear about the difference between an array and a list: • An array is a static structure whose

Question			Answer/Indicative content	Marks	Guidance
					size cannot be changed. • Lists are dynamic and support items being removed or appended.
			Total	4	

Question			Answer/Indicative content	Marks	Guidance
4	a	i	☑ First In First Out	AO1.1 (1)	Examiner's Comments Most candidates correctly identified that a queue is a 'First in First Out' data structure.
		ii	<table><tr><td>Ben</td><td>Sundip</td><td>Tom</td><td>Charlie</td><td>Ling</td><td>Sara</td><td></td><td></td></tr></table> 1 Mark: Adding, Charlie, Ling and Sara in correct order 1 Mark: Exclusively removing Alex and Kofi.	AO2.1 (2)	Accept: <table><tr><td></td><td></td><td>Ben</td><td>Sundip</td><td>Tom</td><td>Charlie</td><td>Ling</td><td>Sara</td></tr></table> Examiner's Comments This question was generally answered well with most candidates achieving both marks. Some candidates did not correctly dequeue from the front of the queue, or enqueue to the rear.

	b	i	`function pop()`
			`  if top == 0 then`
			`  return -1`
			`  else`
			`  item = items[top]`
			`  top = top – 1`
			`  return item`
			`  endif`
			`end function`
				AO3.2 (4)	Examiner's Comments
					
					Many candidates were given full marks for this question, although some did not correctly return 'item' and incorrectly stated 'items' would be returned.

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance

		ii	• Correctly declaring the function reverse to include passing in name as a parameter
			• Correct logic to calculate the number of pushes required
			• Correct use of a loop to push all characters onto the stack separately
			• Creating a local variable reverseName to hold the reversed string
			• Correct use of a loop to pop all characters from the stack (in the reverse order)
			• Correct logic to add each popped character to the reverseName variable
			• Correctly returning the reverseName variable
				AO3.1 (3)
				AO3.2 (4)
					Example solution
					`function reverse(name)`
					`  reverseName = ""`
					
					`  for nameCount = 0 to name.Length-1`
					
					`theStack.push(name[nameCount])`
					`  next nameCount`
					
					`  for nameCount = 0 to name.Length-1`
					`  reverseName = reverseName + theStack.pop()`
					`  next nameCount`
					
					`  return reverseName`
					`end function`
					
					Give full marks for alternative solutions that would work fully.
					
					Allow FT for any duplicate identifiers named incorrectly or using the incorrect case
					
					Examiner's Comments
					
					Although some candidates answered this question well, others were only given 1 or 2 of the available 7 marks. Many candidates that lost marks did not push and/or pop characters correctly, and many did not return a value for the final mark. Candidates should also be reminded that the correct spelling and case of identifiers are required in questions of this type. Some candidates added spaces in identifier names or used a different case to the identifiers stated in the question and therefore, were not given marks.
					
					Assessment for learning
					
					Candidates need to understand the importance of maintaining the spelling, case and use of spaces etc in identifiers in questions which ask for functions to be written.

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance
			Total	14	

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance
5		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).
			Total	2	

© OCR 2026. You may photocopy this page.

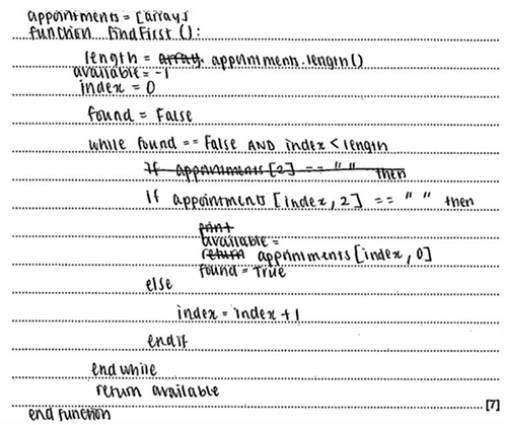
Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance
6			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.
			Total	2	

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

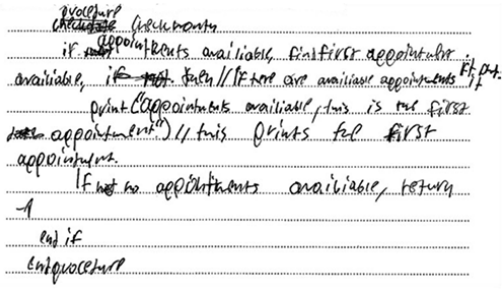
Question			Answer/Indicative content	Marks	Guidance

7	a		1 mark per bullet up to a maximum of 7 marks, e.g.:
			
			• Defining the `findFirst` function correctly
			• Suitable logic for checking the first time slot
			• Suitable logic for checking the next time slot…
			• Suitable loop to check all time slots
			• Suitable logic for returning the available time slot
			• Suitable logic for returning -1 if no time slots available
			• Suitable use of variable names and indentation
				7
				(AO3.1)
				(7)
					Example solution:
					`function findFirst()`
					`    count = 0`
					`    do`
					`        found = false`
					`        if customerID[2, count] == "" then`
					`            found = true`
					`        else`
					`            count = count + 1`
					`        endif`
					`    until count == 10 or found == true`
					`    if found == True`
					`        return customerID[0,count]`
					`    else`
					`        return -1`
					`    endif`
					`endfunction`
					There are many different ways that this function could have been achieved. Therefore other alternative methods should be given credit.
					
					<u>Examiner's Comments</u>
					A number of candidates started by defining a procedure rather than defining a function. Indentation was not always consistent with the constructs being used. Candidates seemed to have difficulty with referencing two dimensional structures. Algorithms that require two dimensional data frequently appear on this paper and candidates need to have extensive practical programming experience solving problems using these structures.
					
					Exemplar 2
					
					appointments = [arrays]
					function findFirst():
					length = ~~array~~ appointment.length()
					available = -1
					index = 0
					found = false
					while found = false AND index < length
					~~if appointments[0] = " " then~~
					if appointments[index,2] = " " then
					~~found~~
					available = ~~return~~ appointments[index, 0]
					found = true
					else
					index = index + 1
					end if
					end while
					return available
					end function [7]
					
					A well-structured pseudocode response

Question	Answer/Indicative content	Marks	Guidance

that uses indentation and variable naming well. The logic of the loop to check each item in sequence for an empty entry is clear, as is the indexing into the 2-Dimensional table structure.

Exemplar 3



To contrast with Exemplar 2, this exemplar shows a lack of pseudocode with a response written mostly in prose English, with inconsistent indentation and a lack of clear variable naming.

Assessment for learning

Candidates frequently benefit from having extensive practical programming experience when answering pseudocode questions.

Past paper questions can provide a context for problems for implementation. For this question candidates could either be asked to code a two dimensional table structure for the data in **Fig 1** and to then implement the linear search, or they code be provided with a scaffolded partially complete solution with the table defined and the stem of a function given.

Routinely taking past paper questions that can be practically implemented is an effective way to make sure that candidates have relevant experience.

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

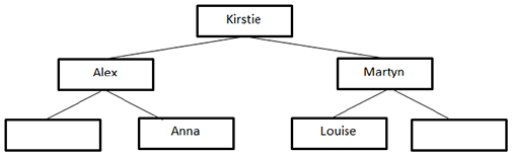
Question			Answer/Indicative content	Marks	Guidance
	b		1 mark per bullet up to a maximum of 3 marks, e.g.: • To start from the beginning // first booking slot • Search each slot in order // sequentially • Search until the first empty slot is found	3 (AO2.2) (3)	Examiner's Comments This question was generally not well answered. Candidates found it difficult to articulate a clear and concise set of steps that take place during a linear search. Many responses were vague and not directly related to a linear search being performed on the data presented in the table.
			Total	10	

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance
8	a	i	Stack	1 AO1.1	Correct answer only

		ii	• Input name from user
			• Check if stack is full e.g.(top <=5)
			• If not, update top pointer
			• Correctly push name to index referenced by top
				4 AO3.2	Example answer
					`new = input("enter a name : ")`
					`if top <=5 then`
					`  top = top + 1`
					`  wNames[top] = new`
					`end if`

	b	i	1 mark per name inserted in correct place in diagram	4 AO2.1	
		ii	• Compare with Kirstie – Zoe is larger so go right • Compare with Martyn - Zoe is larger so go right • No right element so stop/not found	3 AO2.2	Allow FT from b(i)
		iii	• Binary tree more efficient than linked list • Do not need to check every value / tree removes half values each time	2 AO1.2	Allow reference to big O for second mark. Linked List O(n), Binary Tree O(log n)
		iv	• Binary tree less efficient than hash table • Hash table can find data immediately / without checking other values.	2 AO1.2	Allow reference to big O for second mark. Binary tree O(log n) Hash table O(1)
			Total	16	

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>9</td><td>a</td><td></td><td>1 mark per bullet to max 6

• Visit root node M
• Visit E and S
• Visit C and J (from E)
• …then P and V (from S)
• Visit G and K (from J)
• Visit L (from K)</td><td>6 AO1.2 (1)
AO2.1 (3)
AO2.2 (2)</td><td></td></tr>
<tr><td></td><td>b</td><td>i</td><td>1 mark per bullet to max 2
e.g:

• A rule of thumb / estimate / guess
• That estimates the distance / cost from each node to the destination node
• To speed up the process of finding a solution
• …by identify which paths to follow first</td><td>2 AO1.1 (1)
AO1.2 (1)</td><td></td></tr>
<tr><td></td><td></td><td>ii</td><td>
<table>
<tr><th>Node</th><th>Distance travelled</th><th>Heuristic</th><th>Distance travelled + Heuristic</th><th>Previous node</th><th>MARKING GUIDANCE</th></tr>
<tr><td>A (✓)</td><td>0</td><td>90</td><td>90</td><td></td><td>1 MARK</td></tr>
<tr><td>B (✓)</td><td>∞ 21</td><td>80</td><td>101</td><td>A</td><td>1 MARK</td></tr>
<tr><td>C (✓)</td><td>∞ 42</td><td>65</td><td>107</td><td>A</td><td>1 MARK</td></tr>
<tr><td>D (✓)</td><td>∞ 42+12=54</td><td>50</td><td>104</td><td>C</td><td>1 MARK</td></tr>
<tr><td>E</td><td>∞ 21+40=61</td><td>50</td><td>111</td><td>B</td><td>1 MARK</td></tr>
<tr><td>F (✓)</td><td>∞ 42+12+23=77</td><td>30</td><td>107</td><td>D</td><td>1 MARK</td></tr>
<tr><td>G</td><td>∞ 42+12+23+33=110</td><td>0</td><td>110</td><td>F</td><td>1 MARK</td></tr>
</table>

Final path = A,C,D,F,G and Distance = 110 (1 Mark)</td><td>8 AO1.2 (3)
AO2.1 (3)
AO2.2 (2)</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>16</td><td></td></tr>
</table>

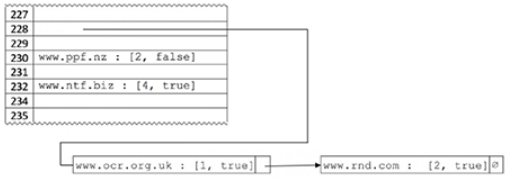
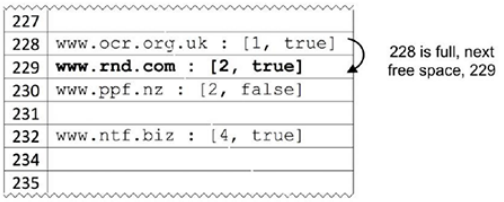
© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance
10	a	i	1 mark per bullet to max 3 • Search the tree to find the location of Node E // by example of search • Replace the content of node E with blank/null/equivalent • Make node A point to the node H • Add node E to the empty node list	3 AO1.2 (3)	
		ii	1 mark per bullet to max 3 • Search the tree to find the location of node G // by example of search • Create a new node with value K • Add a pointer from node G to the new node • Make node K point to null/equivalent	3 AO1.2 (3)	
	b		1 mark per similarity to max 2 • Both consists of nodes • Both are connected by edges/links • Both are non-linear data structures • Both are dynamic data structures 1 mark per difference to max 2 • Tree is 1-directional whereas a graph is 2-directional • Tree has a root node whereas a graph does not have a (clear) root node • Tree will not have cycles whereas graphs can contain cycles • Tree will not be weighted whereas edges in a graph can be weighted	4 AO1.1 (4)	
	c		One node (node A) has more than 2 connections Nodes aren't ordered (e.g. F is C's left child)	1 AO2.1 (1)	
	d		1 mark for identification • Null pointers	1 AO2.1 (1)	
	e		1 mark per bullet • Take **A** as starting node • Visit **B**, **C** and **E** • Visit **D, F, G** and **H** • Visit **I** and **J**	4 AO1.2 (2) AO2.2 (2)	Allow the reverse ordering from right to left e.g. A; E, C, B; H, G, F, D; J, I
			Total	16	

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance
11	a		– rnd.com would cause a collision with ocr.org.uk / would has to the same position as ocr.org.uk (228) – Linear probing could be used – Move through the structure one space at a time – …to find the next free space/229 – Chaining could be used – Each location points (to the start of) a <u>linked</u> list. – The new item is added to the end of the linked list / free. – points to an overflow area – The new item is stored with the other values in the same area Accept a separate or annotated diagram showing a method on given example (1 per -, max 4)	4 AO1.2	Example diagrams Chaining  Linear Probing  <u>Examiner's Comments</u> It was pleasing to see that many candidate responses did refer to the given example. Where this was evident, the candidate scored well. Candidates who did not refer to the example, gained little to no credit on the question.

Question			Answer/Indicative content	Marks	Guidance
	b		**Mark Band 3–High Level (9-12 marks)** The candidate demonstrates a thorough knowledge and understanding of storing and retrieving data from hash tables and linked lists. 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 both sides of the argument which results in a supported and realistic judgment as to which data structure is suitable. 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 storing and retrieving data from hash tables or linked lists; 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 which data structure is suitable. 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 storing and retrieving data from hash tables or linked lists; the material is basic and contains some	12 AO1.1 (2) AO1.2 (2) AO2.1 (3) AO3.3 (5)	Searching of a Linked list involves starting at the first node and following the pointers until either the desired value is found, or the end of the list is reached, meaning the item isn't in the list. The bigger the linked list grows, the longer it takes to search. If a linked list doubles in size it will, on average, take twice as long to search. A list of size n takes on average n/2 checks. In Big O this is O(n), or linear complexity. Searching of a hash table requires the key to be hashed and the correct location accessed. The time this takes is largely dependent on the time to create the hash. If we ignore collisions, the time to find an item will stay the same regardless of the size of the white list. In other words it has O(1) or constant complexity. Unfortunately as the white list grows collisions become more likely. Linear probing and chaining means that once a location has been found the time taken grows linearly with the number of collisions that have occurred for that location, Nonetheless this is still going to perform significantly better than a linked list. If items are added to the end of the linked list then if the location of the last node is stored, that location can be ready made to point at the new item. The time to add items is constant. If they are added in some sort of order then the time to add items grows linearly due to the time spent searching for the right position. (Storing in order has the advantage that it is if an item isn't in the list this can be deduced once its location is passed, rather than waiting until the end.) Adding items to a hash table involves hashing the key and placing it in the correct location. This takes a constant amount of time.. ..unless there are collisions then there is

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance
			inaccuracies. The candidate 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 communicated 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.		an overhead which grows with the number of collisions for that location. Overall a hash table is likely to be the best option (assuming it has enough space and a good hashing algorithm which produces a hash quickly and with few collisions). It will give very consistent performance even as the whitelist grows. *NB Big O Notation isn't necessary for full marks as it is paper 2 content (though many of the more able candidates are likely to include it). The question is assessing candidate's knowledge of traversing and adding to the two data structures and their ability to analyse this to determine their suitability for the scenario.* Examiner's Comments Candidates were assessed on the quality of their extended response in this question. Most candidates correctly cited direct access as the main advantage of hash tables and access times as the disadvantage of linked lists. The level of discussion varied with most candidate responses being given Level 2.
	c		tuple / record / list	1 AO2.1	Don't accept array Examiner's Comments Many candidates cited a suitable structure but some incorrectly cited an array. This demonstrates a clear misconception or lack of understanding that an array can only hold one type of data.
			Total	**17**	

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance
12			A queue	1 (AO1.1)	
			Total	1	

© OCR 2026. You may photocopy this page.

Created in ExamBuilder