Qualification Accredited

OCR
Oxford Cambridge and RSA

Examiners' report

A LEVEL

COMPUTER SCIENCE

OCR Level 3 Advanced GCE in Computer Science

H446

For first teaching in 2015

H446/02 Summer 2025 series

Version 1

ocr.org.uk/computerscience

Contents

Introduction

Our examiners' reports are produced to offer constructive feedback on candidates' performance in the examinations. They provide useful guidance for future candidates.

The reports will include a general commentary on candidates' performance, identify technical aspects examined in the questions and highlight good performance and where performance could be improved. A selection of candidate responses is also provided. The reports will also explain aspects which caused difficulty and why the difficulties arose, whether through a lack of knowledge, poor examination technique, or any other identifiable and explainable reason.

Where overall performance on a question/question part was considered good, with no particular areas to highlight, these questions have not been included in the report.

A full copy of the question paper and the mark scheme can be downloaded from Teach Cambridge.

Paper 2 series overview

Paper 2 focuses on algorithms and problem solving. It assesses candidates' computational thinking ability to analyse and solve problems. Candidates are expected to be able to write algorithms fluently in either pseudocode or program code and to be able to trace and debug algorithms. To be successful in the paper, candidates require a solid grounding in standard data structures backed by practical experience. The question paper also covers Object Oriented Programming (OOP) principles and candidates need to have extensive experience of OOP to be able to tackle Section B of the question paper successfully.

Candidates who did well on this paper generally:	Candidates who did less well on this paper generally:
• displayed an ability to competently interpret, modify, debug and write pseudocode • demonstrated a clear understanding of OOP principles allied to an ability to write code that instantiated classes and provided methods for a given scenario • displayed a good knowledge of a range of data structures and standard algorithms for operations on the given data structures. The data structures covered were stacks, trees, graphs and linked lists • demonstrated more AO3 evaluative comparisons which were relevant to the scenarios given for the level of response questions.	• were less successful in interpreting or correcting code and were unable to write basic algorithms using pseudocode • showed limited understanding of OOP which suggested limited practical experience of relevant programming • displayed limited knowledge of standard data structures such as stacks and linked lists where AO1 knowledge was evident, but AO2 application was less evident • demonstrated very limited knowledge in extended response questions and were less successful in applying their learning to the scenarios given.

Section A overview

Section A of the question paper consisted of seven questions, each with a separate theme.

Question	Topics covered
1	Sorting algorithms
2	Searching algorithms and data mining
3	Stacks
4	Pipelining
5	Dijkstra's algorithm and A* algorithm
6	Procedural coding
7	Trees and tree traversals

Question 1 (a) (i)

1 Data in a computer program needs to be sorted.

(a)
(i) Describe the steps a bubble sort will take to sort items into **ascending** order.

.....

.....

.....

.....

.....

..... **[5]**

Most candidates were able to describe the steps that a bubble sort takes. The most given mark points were for giving the correct steps used in the first pass of a bubble sort. Fewer candidates were able to demonstrate the depth of knowledge required to achieve full marks.

The use of a temporary variable to swap two values and the use of a swap flag in a more efficient version of a bubble sort were far less frequently cited.

Question 1 (a) (ii)

(ii) Darcie and Felix have both written a program that will perform a bubble sort to put these numbers into **ascending** order.

2	1	3	4	5	6	7

Darcie's version will complete **six** passes. However, Felix's version will only need to complete **two** passes.

Explain how Felix may have made his version more efficient than Darcie's version.

.....

.....

.....

..... **[2]**

Few candidates demonstrated the depth of knowledge to be able to recognise that a swap flag would be required for a bubble sort to sort the given set of data within two passes.

Question 1 (b)

(b) A second type of sorting algorithm is an insertion sort.

Describe how an insertion sort can be used to put the following data into **ascending** order.

20	8	15	36

.....

.....

.....

.....

.....

..... **[5]**

Some candidates confused insertion sort with other sorting algorithms. Many candidates demonstrated the knowledge to apply the principles of an insertion sort to the given data set.

Some candidates gave generic descriptions for an insertion sort which was not given marks.

Where candidates used diagrams, the level of annotation was often not sufficient to give the associated level of description required. Many less successful responses did not state that 20 would be considered as sorted in the correct position at the start of the insertion sort, before 8 was compared to 20.

Question 1 (c) (i)

(c) A third type of sorting algorithm is a quick sort.

The following quick sort algorithm is made up of two functions called `partition()` and `quickSort()`. The algorithm will sort an integer array into **ascending** numerical order.

```
01   function partition(array, low, high)
02      pivot = array[high]
03      index = low - 1
04      for count = low to high-1
05         if array[count] < pivot then
06            index = index + 1
07            temp = array[index]
08            array[index] = array[count]
09            array[count] = temp
10         endif
11      next count
12      temp = array[index + 1]
13      array[index + 1] = array[high]
14      array[high] = temp
15      return index + 1
16   endfunction
17
18   function quickSort(array, low, high)
19      if low < high then
20         partitionIndex = partition(array, low, high)
21         quickSort(array, low, partitionIndex - 1)
22         quickSort(array, partitionIndex + 1, high)
23      endif
24   endfunction
```

(i) Identify which of the **two** functions is recursive **and** give all of the line numbers where recursive calls are made.

Recursive function name .....

Recursive call line numbers .....

[2]

Most candidates correctly identified `quickSort` as the recursive function name in lines `21` and `22`.

Question 1 (c) (ii)

(ii) Identify the type of iteration used in the function `partition()`.

..... **[1]**

Most candidates correctly identified count-controlled iteration. Conditional iteration was a common, less successful response.

Question 1 (c) (iii)

(iii) Describe the role of the function `partition()` in the quick sort algorithm.

.....

.....

.....

.....

.....

..... **[3]**

Many candidates were far less familiar with quick sort and thus less successful in describing the role of partitioning. Some successful responses identified that a pivot was selected. Fewer candidates then went on to state that values are compared to the selected pivot and that values less than the pivot are placed to the left of the pivot and values greater than the pivot are placed to the right.

Question 1 (c) (iv)

(iv) The algorithm needs to be changed to sort the integer array into **descending** numerical order.

Identify the line number that needs to be changed **and** write the amended line of code.

Line number .....

Amended line of code .....

.....

[2]

Many candidates successfully identified and corrected the relevant line of code.

Question 1 (c) (v)

(v) The quick sort is an example of a divide-and-conquer algorithm.

Describe what divide-and-conquer means.

.....

.....

.....

..... **[2]**

Some candidates confused divide-and-conquer as a problem solving technique with decomposition or stepwise refinement. Responses giving examples were credited.

Question 2 (a)*

2* A shopping website requires customers to create a username and password.

(a) The usernames and passwords are stored securely in a file. The file contains the username and passwords for 50 million users which are sorted in alphabetical order by username.

The file can be searched using a linear search or a binary search.

The big-O complexities for a linear search and a binary search are given in the table:

Search algorithm	Worst-case time	Average-case time	Worst-case space
Linear	O(n)	O(n)	O(1)
Binary	O(log n)	O(log n)	O(1)

Compare the use of a linear search and a binary search in this scenario.

You should include the following in your answer:

- the features and preconditions of a linear and binary search
- the benefits and drawbacks of each search in this scenario
- a conclusion justifying which search algorithm should be used in this scenario.

You should make reference to the big-O complexities in your answer. **[9]**

Most candidates were able to describe how a linear search and a binary search work, as well as their time and space Big O complexities.

Fewer responses made explicit links to the scenario to identify that the data was already ordered, but most candidates successfully gave some evaluation to justify binary search as the best option.

Most responses achieved a mark towards the top of Level 2 or the bottom of Level 3.

Question 2 (b) (i)

(b) The website collects data from customers' activities once they log on. For example, the items they view and purchase. The company that owns the website would like to use data mining to improve their website and customer experience.

(i) Describe what data mining means.

.....

.....

.....

.....

.....

..... **[3]**

Most candidates had some knowledge of data mining. However, fewer candidates were able to describe it accurately and in depth.

Many candidates displayed the misconception that data mining was simply collecting data or customer preferences rather than analysing large sets of data.

Some responses did not make it clear that data mining operates on large sets of data. Where candidates had learnt definitions, they were successful in gaining full marks for this question.

Misconception

Data mining is not simply collecting sales data or logging customer preferences. Data mining is specifically analysing very large sets of data to find patterns or trends that might otherwise not be obvious.

Question 2 (b) (ii)

(ii) Explain how the company can make use of data mining to improve their website and customer experience.

.....

.....

.....

..... **[2]**

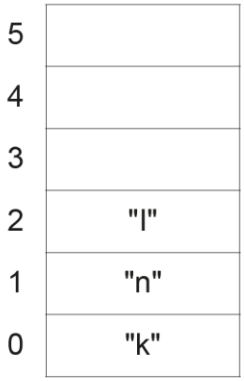
Most candidates made a successful attempt to give a response within the context of the scenario.

Candidates needed to identify the type of information or pattern that could be generated from data mining before giving a related expansion to explain how that would then be used.

Question 3 (a)

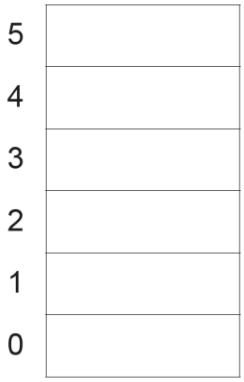
3 A program is designed to use a stack data structure.

The current contents of part of the stack and its current pointer value are shown:




```
pointer = 2
```

(a) Show the contents of the stack and its pointer value after the data items "t" and "s" are inserted in the order given.




```
pointer = .....
```

[2]

Most candidates successfully demonstrated the ability to add new values to the stack in the correct order.

Question 3 (b)

(b) The stack is stored using a 0-based 1-dimensional array called `theStack` with 20 elements.

The function `push()` stores its parameter into the next space in the stack if it is not full. The function returns `true` if it is successfully stored and returns `false` if the stack is full.

Complete the function `push()` using pseudocode or program code.

```
function push(data)
   if pointer == ..... then
      return false
   else
      pointer = .....
      theStack[pointer] = .....
      .....
   endif
endfunction
```

[4]

Most candidates made a successful attempt at this question. But many did not achieve full marks.

A common misconception was to compare the pointer with 20 instead of 19, showing limited understanding of zero-indexing and the purpose of the pointer in this code being to point to the last item inserted into the stack.

Question 3 (c)

(c) A `pointer` value of `-1` indicates that the stack is empty.

The function `pop()` returns the next element in the stack. If the stack is empty it returns `"false"`. The function updates the `pointer` value when appropriate.

Write the function `pop()` using pseudocode or program code.

.....

.....

.....

.....

.....

..... **[6]**

Many candidates made a good attempt to write a `pop()` function with many candidates being given 5 or 6 marks.

A common misconception for this question was to return the value of the stack at the pointer **without** sequencing the pointer decrement correctly.

Exemplar 1

```
def pop():
    if pointer == -1:
        return False
    else:
        return theStack[pointer]
        value = theStack[pointer]
        return value
        pointer = pointer - 1
```

This response was given 5 marks and exemplified a common misconception. Many candidates returned the value at the top of the stack and then decremented the pointer. This would mean that the line of code to decrement the pointer would not run. The pointer decrement needed to be placed before the return value line.

Question 3 (d)

(d) A program is designed to reverse a set of data using the stack.

The main program takes 10 characters separately as input from the user and uses the function `push()` to insert these into the stack.

Once all 10 characters have been inserted into the stack, the program then uses the function `pop()` to output all 10 characters from the stack.

Write the main program to achieve this using pseudocode or program code.

.....

.....

.....

.....

..... **[5]**

Many candidates were less successful in producing pseudocode to an open-ended problem. Many candidates did not create accurate loop structures that would iterate 10 times.

There were few completely successful responses. Less successful responses often did not output values popped from the stack or iterated an incorrect number of times.

Exemplar 2

Once all 10 characters have been inserted into the stack, the program then uses the function `pop()` to output all 10 characters from the stack.

Write the main program to achieve this using pseudocode or program code.

- For loop input & push
- for loop pop & concat
- print

```
FOR x =0      y = 0      ans = ''  ''
for x in range 0 to 9 :
    data = input ("Enter character: ")
    push (data)
For x = x + 1
for y in range 0 to 9 :
    ~~out of loop~~ ans = ans + pop()
    y = y + 1
print (ans)
```

[5]

This response gained 4 marks. The candidate wrote a mix of Python and pseudocode.

This response shows how the variable has been incremented inside of the loop. Where candidates produced an incorrect loop, follow through was applied and marks were gained for the subsequent input and push operations.

This solution appended the popped values before outputting to produce a valid solution. Again, the output push loop was incorrect, but it operated the same number of times as the input push loop, so follow through was applied.

Question 4*

4* A processor is being designed to use pipelining and will need to be able to process millions of instructions.

For each instruction the following stages will be performed:

A the instruction will be fetched from memory
B the instruction will be decoded
C the instruction will be executed

The first four instructions to be processed will be instructions 1, 2, 3 and 4. The design allocates 1-time interval for each of the three stages A, B, C.

Discuss the use of pipelining by this processor.

You should include the following in your answer:

- the features of pipelining
- how pipelining can be used to manage the first four instructions
- a conclusion justifying the use of pipelining in this scenario. **[9]**

Most responses to this question were given a Level 2 mark, with many towards the middle of the band.

Some candidates confused multiple cores and parallelism or threading with processor pipeline architecture.

Most candidates gave a description or annotated diagram showing how the overlapping of the fetch, decode and execute phases would enable greater throughput.

Fewer candidates were able to explain more nuanced factors such as the need for equally sized instructions or the need for more sequential programs to get the greatest gains from pipelining.

Level 3 responses frequently considered hazard events such as branching and the effect that this would have on pipelining.

Question 5 (a)

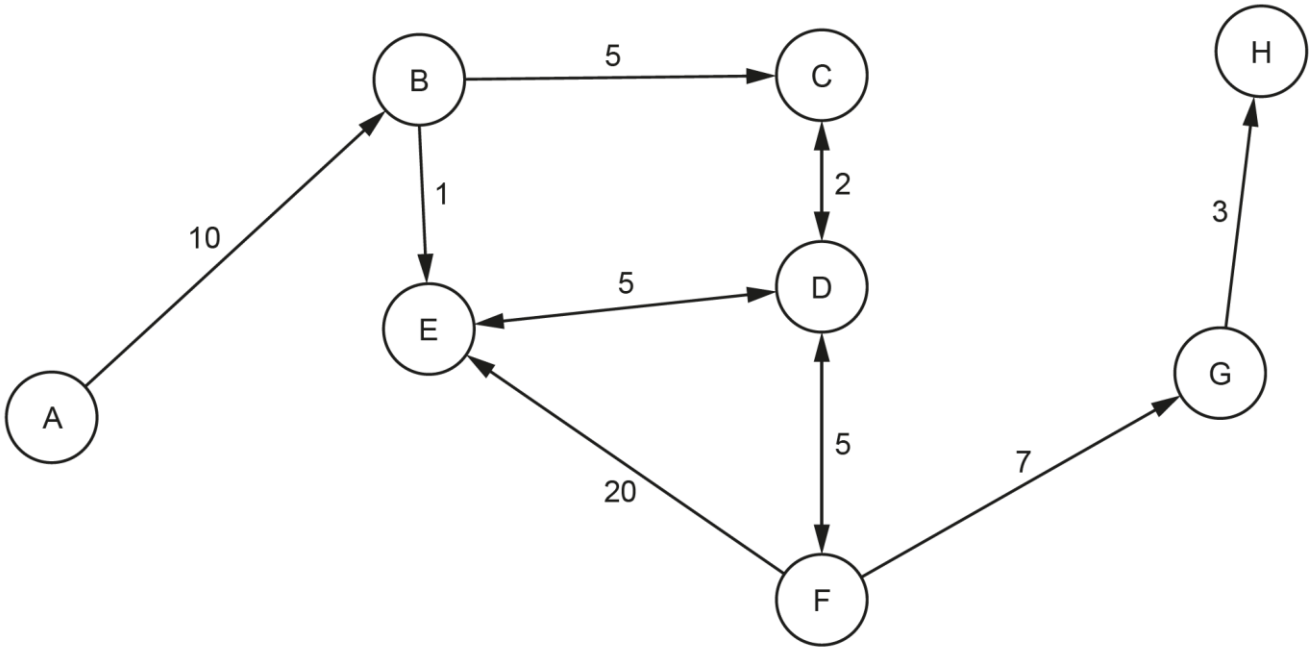
5 A program is being written that allows users to travel between different planets in a virtual world. Spaceships are used to travel between planets and there are only set routes that they use.

Fig. 5 shows a directed graph. The nodes represent the different planets and the edges represent the routes the spaceships can travel.

The arrows on the edges show which way the spaceships can travel. For example, a spaceship can travel from planet B to planet C, but **not** from planet C to planet B.

The value on each edge is the cost in space dollars to travel on that route.

Fig. 5



(a) Explain why this graph is a visualisation of the problem.

.....

.....

.....

..... **[2]**

Responses were evenly distributed across the mark range to this question. Some candidates confused visualisation with abstraction. Many candidates gave examples from the graph to explain how it simplified and aided understanding by diagrammatically showing the relationship between the different planets and the relative distances.

Question 5 (b)

(b) A user needs to travel from planet A to planet H. They need to travel the least expensive route.

Show how Dijkstra's algorithm can be used on the directed graph shown in **Fig. 5** to find the shortest path from start planet A to the end planet H.

You must state the nodes on the final path and the distance of this path. Show your working.

You may use the table to give your answer.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Node	Distance travelled	Previous node

Final path: .....

Distance: .....

[7]

Candidates demonstrated good understanding of Dijkstra's algorithm and its application to the given graph. Most candidates were given full marks.

Question 5 (c)

(c) The A* algorithm can also be used to find the shortest path in a graph.

Describe how an A* algorithm uses heuristics to be an efficient algorithm.

.....

.....

.....

..... **[2]**

Few candidates were able to identify the role of heuristics in the A* algorithm. Many candidates gave generic definitions for heuristics and did not respond to the set question.

Where candidates did recognise the role of the heuristic to determine an approximation of the distance of a node to the target node, fewer were then able to say how this would be applied or how it made the A* algorithm more efficient.

Assessment for learning

Reading the question set.

Many candidates see a keyword such as heuristic and write a definition without pausing to consider whether it responds to the set question.

This question saw many responses that did not respond the question and is therefore a good example to use when considering exam technique with candidates.

Question 6 (a)

6 A student is writing a program in a high-level programming language.

(a) The student writes the program using an Integrated Development Environment (IDE).

Complete the table by identifying **and** describing **three** features of an IDE that the student can use to **debug** their program.

Debugging feature	Description
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....

[6]

Marks were evenly distributed across the range with most candidates making a successful attempt at the question.

Some candidates did not read the question clearly and gave features of IDEs that were not specifically relevant for debugging programs - such as code auto-completion and pretty printing.

Features had to relate specifically to debugging.

Question 6 (b) (i)

(b) The program the student is developing includes both global and local variables.

(i) Explain the difference between a global variable and a local variable.

.....

.....

.....

..... **[2]**

Most candidates were able to demonstrate knowledge of the difference in scope between local and global variables and therefore gained full marks.

Question 6 (b) (ii)

(ii) Describe **one** drawback of using global variables in a program.

.....

.....

.....

..... **[2]**

Candidates appeared less confident on the drawbacks to using global variables . While many candidates gained 1 mark, far fewer were able to articulate a clearly linked expansion.

The most successful responses related to global variable use of memory or potential side effects.

Question 6 (b) (iii)

(iii) Describe **one** drawback of using local variables in a program.

.....

.....

.....

..... **[2]**

Candidates also seemed less familiar with the drawbacks of local variables. Many candidates identified that they were limited to the scope in which they were defined, but then unable to expand on this, for example, with the proviso that values would have to be passed/returned increasing program complexity.

Question 6 (c)

(c) A variable can be passed as a parameter into a function by value or by reference.

Describe what parameter passing by value means.

.....

.....

.....

..... **[2]**

Most candidates demonstrated clear understanding of passing by value with relatively few confusing it with passing by reference.

Question 6 (d)

(d) An algorithm is needed to count how many times each of the numbers from 1 to 9 appear in an array called `myArray`. Another array called `checked` is used to store each number found in `myArray` and the number of times it appears. For example, the contents of `myArray` are:

7	3	6	7	5	3	2	3	6

Based on these contents in `myArray`, the contents in `checked` would be:

7	2
3	3
6	2
5	1
2	1
–1	0
–1	0
–1	0
–1	0

–1 indicates an unused space in the array as the numbers 1, 4, 8 and 9 have **not** been found.

The student has written the following pseudocode for this program.

The pseudocode contains **several** errors.

```
01   array myArray[9]
02   myArray = [7,3,6,7,5,3,2,3,6]
03   array checked[9,2]
04   checked = [[-1,0],[-1,0],[-1,0],[-1,0],[-1,0],[-1,0],[-1,0],
               [-1,0],[-1,0]]
05   numbersFound = 1
06   for x = 0 to myArray.length
07      found = false
08      for y = 0 to checked.length
09         if checked[y,0] == myArray[y] then
10            checked[y,1] = checked[y,1] + 1
11            found = true
12         endif
13      next y
14      if found == true then
15         checked[numbersFound,0] = myArray[x]
16         checked[numbersFound,1] = 0
17         numbersFound = numbersFound + 1
18      endif
19   next y
```

Identify the line number of any **four** errors **and** write the corrected line for each error.

Error 1

Line number .....

Corrected line .....

Error 2

Line number .....

Corrected line .....

Error 3

Line number .....

Corrected line .....

Error 4

Line number .....

Corrected line .....

[4]

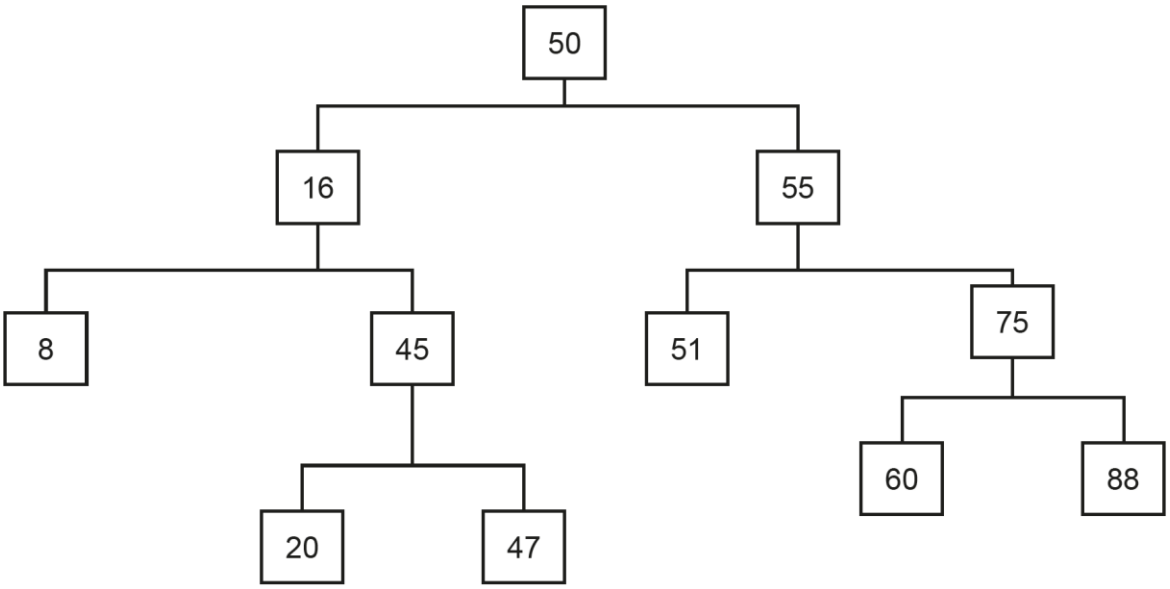
While marks were evenly distributed across the range it was noticed that many candidates were less successful in interpreting and debugging code. There were many less successful responses to this question.

Assessment for learning

This question can be set as a coding exercise. IDE tools such as breakpoints, stepping and variable watch windows can be used to see how variable values change during program execution. This can assist to identify how the algorithm is operating to determine where logical errors are present. Specific practical use of these debug features will help to reinforce which features of an IDE assist with debugging code.

Question 7 (a)

7 An ordered binary search tree contains the following data:



(a) Add the following data items in the given order to the binary search tree above.

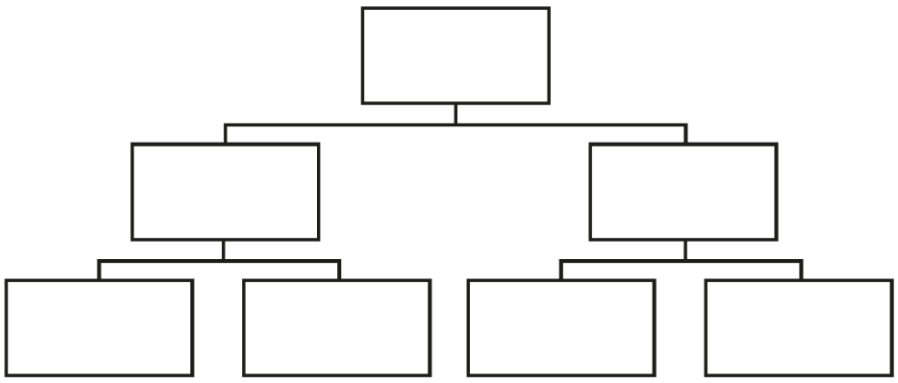
5 30 53 57 90

[3]

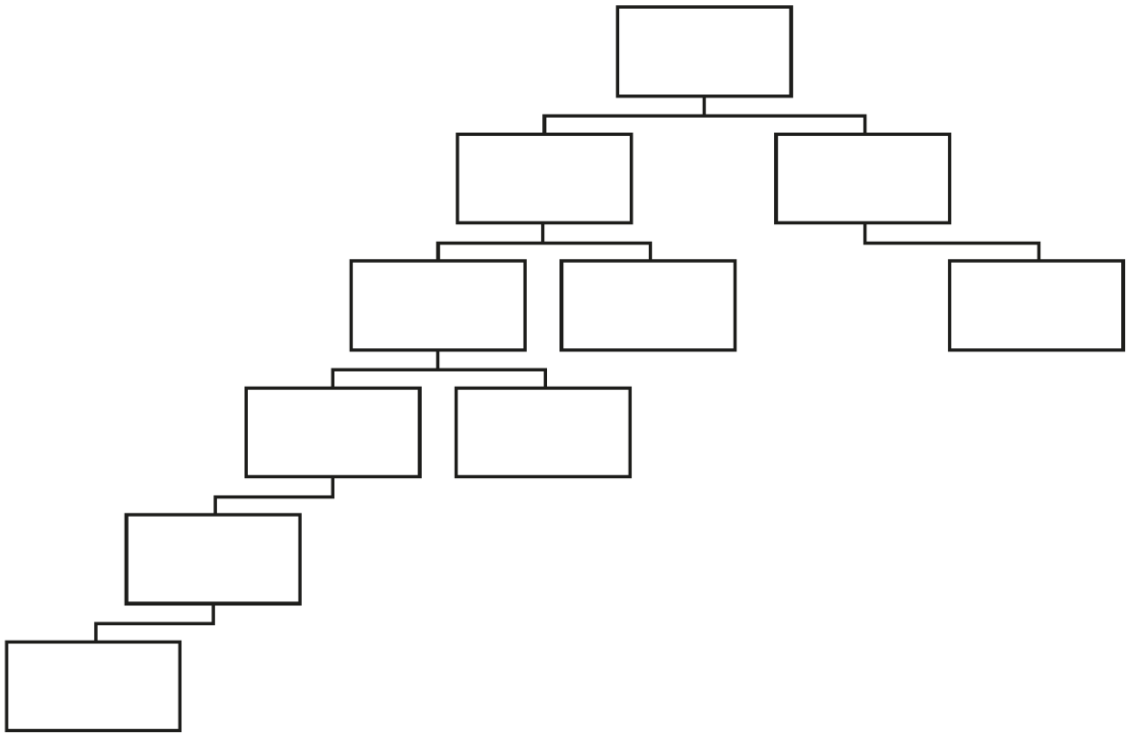
Most candidates were given full marks for this question by successfully inserting the given values.

Question 7 (b)

(b) A balanced binary search tree has data consistently at each level, for example:



An unbalanced binary search tree is uneven, for example:



Describe the drawback of an unbalanced binary search tree when searching for a value.

.....

.....

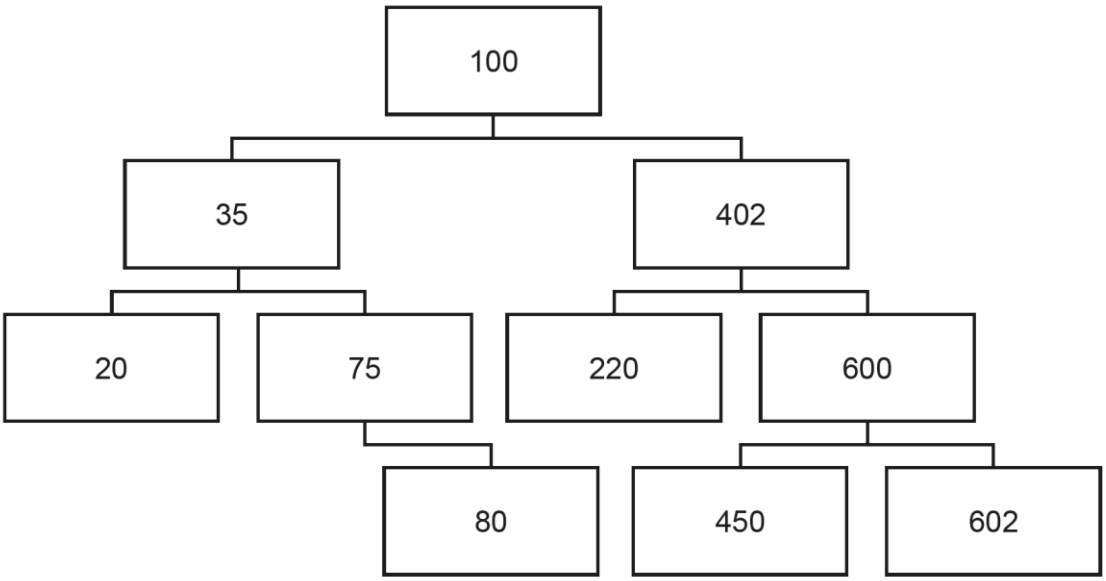
.....

..... **[2]**

Most responses were given 1 mark for identifying that it would take longer to locate values, but fewer candidates were able to clearly articulate why this would be the case. Some candidates also confused traversals with searches.

Question 7 (c)

(c) A different binary search tree stores the following data:



Give the output that will be produced from a depth-first (post-order) traversal and a breadth-first traversal of this binary search tree.

Depth-first (post-order) .....

.....

Breadth-first .....

.....

[4]

Many candidates were given full marks and demonstrated clear understanding of both depth-first and breadth-first traversals. Where candidates did not gain full marks the response to depth-first was often less successful.

Section B overview

Section B referred to one specific scenario that revolved around a problem that was solved by using Object Oriented Programming (OOP) techniques. Those candidates who had very limited practical experience of OOP were less successful in responding to the questions in Section B. Conversely, those candidates who demonstrated strong coding skills often gave successful responses.

Assessment for learning

The OOP scenario in Section B lends itself to implementation by candidates as a programming exercise to enhance the learning of associated concepts such as constructor methods, instantiation and class methods.

Question 8 (a)

8 A linked list data structure is designed using object-oriented programming.

The design includes two classes called `node` and `linkedList`.

(a) Describe the features of a linked list data structure.

.....

.....

.....

.....

.....

..... **[3]**

Most candidates were able to identify some features of a linked list but far fewer were able to identify three distinct features. Many candidates did not clearly articulate that nodes in a linked list contain both data and a pointer to the next node in a linked list.

Commonly cited features beyond nodes containing data and a pointer were that linked lists contain a head pointer and that linked lists are dynamic data structures.

Question 8 (b) (i)

(b) The class design for `node` is shown here.

class: node
attributes: private value : integer private nextNode : node
methods: new() getValue() getNextNode() setValue() setNextNode()

`new()` is the constructor method. The constructor method takes a parameter and assigns this to the `value` attribute.

(i) The method `getNextNode()` returns the attribute `nextNode`.

Write the method `getNextNode()` using pseudocode or program code.

.....

.....

.....

..... **[2]**

Many responses did not include a clear function definition, often omitting the word function and writing `getNextNode()` on its own as given in the question.

Pseudocode responses must be clear. Where programming languages are used, candidates should use common conventions and add comments to their code if the meaning might not be clear.

While candidates do not have to use the OCR pseudocode it does provide a mechanism for clear and unambiguous solutions to be given.

Question 8 (b) (ii)

(ii) The method `setValue()` takes an integer as a parameter and assigns this to the attribute `value`.

Write the method `setValue()` using pseudocode or program code.

.....

.....

.....

.....

.....

..... **[3]**

In this question, a common error was to omit a parameter, or to not assign the given parameter to the specific class attribute `nextNode`.

Question 8 (c) (i)

(c) The class design for `linkedList` is shown here.

class: linkedList
attributes: private headNode : node
methods: new() insertNode() outputList()

`new()` is the constructor method. The constructor initialises the attribute `headNode` to a `null` value.

(i) Write the constructor method using pseudocode or program code.

.....

.....

.....

..... **[2]**

Candidates generally found writing the constructor method more challenging than the previous getter and setter methods. The most common error was to pass a value as a parameter to assign to `headNode` rather than just assigning a `null` value to it.

Question 8 (c) (ii)

The class design for `node` and `linkedList` are shown again here for your reference.

class: node
attributes: private value : integer private nextNode : node
methods: new() getValue() getNextNode() setValue() setNextNode()

class: linkedList
attributes: private headNode : node
methods: new() insertNode() outputList()

(ii) The method `getValue()` will return the value of a node. The method `getNextNode()` will return the next node to be accessed in the linked list.

The method `outputList()` traverses the linked list starting at the node `headNode`. The method joins the data in each node and returns it.

Complete the method `outputList()` using pseudocode or program code.

```
function outputList()

   returnValue = ""

   start = .....

   while start ..... null

      returnValue = returnValue + start. .....() + " "

      start = start. ..... ()

   endwhile

   return .....

endfunction
```

[5]

Most candidates scored either 4 or 5 marks. There were a significant number of candidates who did not use a valid relational operator for not equal to, using =! Instead of !=.

Question 8 (d)

(d) The method `insertNode()` will insert a new node into the linked list.

The method accepts an integer as a parameter, creates a new node with that data and inserts it into the linked list, e.g. `insertNode(8)` creates a node object with the data 8 and adds it to the linked list.

The program needs to store the following data items in the linked list in the order they are given:

10 25 39

The program then needs to output all of the data in the list using the appropriate method.

Write an algorithm using pseudocode or program code to:

- create a new `linkedList` object with the identifier `createList`
- insert each of the data items into the linked list (10, 25, 39)
- output the content of the linked list.

.....

.....

.....

.....

.....

..... **[5]**

Many candidates found this question challenging and were given no marks.

Some candidates scored 4 or 5 marks. Candidates were less successful in writing a valid instantiation statement to create an instance of `linkedList` with the identifier `createList`.

Calling the correct method to `insertNode` to add the required values and then calling the `outputList` method proved problematic for many candidates.

Where candidates had completed all these steps the final requirement to print the value returned by the `outputList` method was often missed.

Question 8 (e)

(e) A new method is needed to remove a node from the linked list. The method needs to take the data to remove as a parameter and then find this data in the linked list before removing the node.

Describe the steps that this method will need to follow **and** the decisions that need to be made in the method to remove the correct node.

.....

.....

.....

.....

.....

..... **[5]**

Few candidates scored more than 1 or 2 marks because descriptions were too vague.

Candidates were expected to list the specific steps required. Many candidates did not articulate clearly how the traversing to the required value was done.

A description of the traversal process starting at the head node and then following each subsequent node via the next node pointer where the parameter value did not match the node's data value, was required. Where candidates had generalised, they could still achieve credit for a description of the pointer updates that were required.

Question 8 (f)

(f) The program developer made use of computational methods when designing the object-oriented programming solution. One computational method is divide-and-conquer.

Identify **three** other computational methods that the program developer could have used.

1 .....

2 .....

3 .....

[3]

Most candidates were less successful in identifying three computational methods. Where candidates gave less successful responses, they often erroneously gave programming constructs such as sequence, selection and iteration.

Question 8 (g)*

(g)* This program uses classes and objects to create a linked list.

An alternative method is creating a 2-dimensional array. In this design the first index stores the data and the second element stores the index of the next data item. Array elements not currently used are also linked together to form a chain of empty values to identify where the next data item can be inserted.

Compare the two different methods of creating a linked list that have been described.

You should include the following in your answer:

- the features of both methods including the reusability of program components
- benefits and drawbacks of both methods in this program
- a conclusion evaluating the efficiency of both methods when designing a program that needs a linked list. **[12]**

The final question required candidate to discuss technical content.

Many candidates did not appreciate how the alternative implementational approaches worked. This may demonstrate limited practical programming experience. Some candidates listed features of the OOP paradigm without specific linkage to the scenario. Very few Level 3 responses were seen.

Supporting you

Teach Cambridge

Make sure you visit our secure website **Teach Cambridge** to find the full range of resources and support for the subjects you teach. This includes secure materials such as set assignments and exemplars, online and on-demand training.

Don't have access? If your school or college teaches any OCR qualifications, please contact your exams officer. You can forward them this link to help get you started.

Reviews of marking

If any of your students' results are not as expected, you may wish to consider one of our post-results services. For full information about the options available visit the OCR website.

Access to Scripts

We've made it easier for Exams Officers to download copies of your candidates' completed papers or 'scripts'. Your centre can use these scripts to decide whether to request a review of marking and to support teaching and learning.

Our free, on-demand service, Access to Scripts is available via our single sign-on service, My Cambridge. Step-by-step instructions are on our website.

Keep up-to-date

We send a monthly bulletin to tell you about important updates. You can also sign up for your subject specific updates. If you haven't already, sign up here.

OCR Professional Development

Attend one of our popular professional development courses to hear directly from a senior assessor or drop in to a Q&A session. Most of our courses are delivered live via an online platform, so you can attend from any location.

Please find details for all our courses for your subject on **Teach Cambridge**. You'll also find links to our online courses on NEA marking and support.

Signed up for ExamBuilder?

ExamBuilder is the exam paper builder platform for a range of our qualifications. Find out more.

Free for all OCR centres with an Interchange account, it allows unlimited users per centre. We require an Interchange username to verify your centre's users for ExamBuilder.

If you do not have an Interchange account, please contact your centre administrator (usually the Exams Officer) to request a username.

Once you have an Interchange username, use the form on this page to sign up for ExamBuilder, or ask an existing user at your centre to create an ExamBuilder account for you.

Active Results

Review students' exam performance with our free online results analysis tool. It is available for all GCSEs, AS and A Levels and Cambridge Nationals (examined units only).

Find out more.

Online courses

Enhance your skills and confidence in internal assessment

What are our online courses?

Our online courses are self-paced eLearning courses designed to help you deliver, mark and administer internal assessment for our qualifications. They are suitable for both new and experienced teachers who want to refresh their knowledge and practice.

Why should you use our online courses?

With these online courses you will:

- learn about the key principles and processes of internal assessment and standardisation
- gain a deeper understanding of the marking criteria and how to apply them consistently and accurately
- see examples of student work with commentary and feedback from OCR moderators
- have the opportunity to practise marking and compare your judgements with those of OCR moderators
- receive instant feedback and guidance on your marking and standardisation skills
- be able to track your progress and achievements through the courses.

How can you access our online courses?

Access courses from Teach Cambridge. Teach Cambridge is our secure teacher website, where you'll find all teacher support for your subject.

If you already have a Teach Cambridge account, you'll find the link to the **Online courses** under the **Assessment** or **Training** tab – hover over **Assessment** or **Training** on the horizontal menu and select **Online courses** from the dropdown.

If you don't have a Teach Cambridge account yet, ask your exams officer to set you up – just send them this link and ask them to add you as a Teacher.

Access the courses **anytime, anywhere and at your own pace**. You can also revisit the courses as many times as you need.

Which courses are available?

We recommend all teachers complete the module on Online courses – NEA for their subject. These courses provide subject-specific information about course content; how to prepare for planning, marking and submitting; and support with understanding the marking criteria.

The subject-specific courses are tailored for each qualification that has non-exam assessment (NEA) units, except for AS Level and Entry Level. They cover the following topics:

- the structure and content of the NEA units
- the assessment objectives and marking criteria for the NEA units
- examples of student work with commentary and feedback for the NEA units
- interactive marking practice and feedback for the NEA units.

We are also developing courses for some of the examined units, which will be available soon.

How can you get support and feedback?

If you have any queries, please contact our Customer Support Centre on 01223 553998 or email support@ocr.org.uk.

We welcome your feedback and suggestions on how to improve the online courses and make them more useful and relevant for you. You can share your views by completing the evaluation form at the end of each course.

Tell us what you think

Your feedback plays an important role in how we develop, market, support and resource qualifications now and into the future. We want you and your students to enjoy and get the best out of our qualifications and resources, but to do that we need your honest opinions to tell us whether we're on the right track or not.

You can email your thoughts to resources.feedback@ocr.org.uk or visit our feedback page to learn more about how you can help us improve our resources.

Designing and testing in collaboration with you and your students

Helping young people develop an ethical view of the world

Equality, diversity, inclusion and belonging (EDIB) are part of everything we do

---

Contact the team at:

01223 553998

ocr.org.uk

To stay up to date with all the relevant news about our qualifications, register for email updates at ocr.org.uk/updates

Visit our Online Support Centre at support.ocr.org.uk

CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT