Chhetri Academy

chhetriacademy.com

GCSE COMPUTER SCIENCE

1.2

Edexcel 1CP2 | Specification point 1.2.6

How standard algorithms (bubble sort, merge sort, linear search, binary search) work

Untiered

Answers and examiner comments

3	2020-2025	Untiered
question groups	available papers	tier

Using this booklet

1. Answer numbers match the companion question booklet.
2. Follow the original question number on marking tables.
3. Examiner comments are included where available.

Name _____ **Date** _____

Some entries use full source marking material in an appendix. Any unavailable mark schemes are identified beside the relevant answer.

6(a)	A description to include **four** from a response such as: URL is clicked/typed into a web browser address bar (1). The page is requested (by the web browser) (1) so HTML and files are fetched from a web server (1) and the browser collates the page (1), which is then displayed on the monitor (1). OR URL is clicked/typed into a web browser address bar (1). The browser locates the IP address (1) either in cache or using a DNS server (1) and the browser sends a request to the web server with that IP address (1); web server uses HTTP to send back the requested web page (1) and the browser displays the content of the page (1).	Expected answer structure: - Request - Response - Build - Render	**4**

Question Number	Answer	Additional Guidance	Mark
6(b)	A description to include **four** from: Finds the midpoint of the list (1), divides/splits the list in two (1) until each list contains one item (1). Compares the first item in each list with the first item in the adjacent list (1) until all have been compared and placed in order (1) and recombined/merged into one list (1).		**4**

Question Number	Answer	Additional Guidance	Mark
6(c)i	6		**1**

Question Number	Answer	Additional Guidance	Mark
6(c)ii	5		**1**

Question Number	Answer	Additional Guidance	Mark
6(c)iii	4		**1**

Question Number	Answer	Additional Guidance	Mark
6(c)iv	No swaps will have been performed.	Do not accept: The list is sorted.	**1**

Question Number	Answer	Additional Guidance	Mark
6(d)	Any **one** of: • Faster • More efficient • Uses fewer comparisons		**1**

Question Number	Answer	Additional Guidance	Mark

Question Number	Answer	Additional Guidance	Mark
1(a)	The only correct answer is C *A is not correct because abstraction is hiding or removing detail* *B is not correct because computation is the act of computing* *D is not correct because evaluation is reaching a conclusion*		(1)

Question Number	Answer	Additional Guidance	Mark
1(b)	Award one mark for any of the following up to a maximum of two marks: • The subprogram code only has to be written once / saves development time (1) • The subprogram can be called many times (reusability) (1) • The subprogram only has to be debugged once (1) • It is easier to locate and debug errors (1) • Hides the details of how a function/procedure/code works (abstraction) (1) • Can be used by programmers who don't have the knowledge to write them (1) • Can be grouped into libraries/can be shared with other programmers (1) • Makes programs easier to read/understand (1) • Reduces the overall size of code (1) • Problem can be decomposed into smaller sub-problems (1) • Allows a team to work on a project (1)	Do not accept generalisations such as easier/faster/quicker without qualification	(2)

Question Number	Answer	Additional Guidance	Mark
1(c)(i)	Award one mark for any of the following: • 15 (1)		(1)

Question Number	Answer	Additional Guidance	Mark
1(c)(ii)	Award one mark for any of the following: • 11 (1)		(1)

Question Number	Answer	Additional Guidance	Mark
1(c)(iii)	Award one mark for any of the following: • 16, 17, 18, 19 (1) • 16 - 19 (1) • 16, 18 (1)	Do not accept 16 on its own.	(1)

Question Number	Answer	Additional Guidance	Mark
1(d)(i)	Award one mark for any of the following: • Code that breaks/violates the rules/grammar of the programming **language** (1)	Do not award credit for an example in isolation such as missing colon, there must be a definition.	(1)

Question Number	Answer	Additional Guidance	Mark
1(d)(ii)	Award up to two marks for a linked explanation, such as: • The program crashes/stops (1) because the operation the computer is asked to do is impossible (1) • The program crashes (1) because the CPU cannot execute one of the instructions in the code (1)	For both marks, the expansion must follow/associate with the statement. Do not award credit for an example in isolation such as division by zero. Be careful not to award marks for syntax/logic error	(2)

Question Number	Answer	Additional Guidance	Mark
1(e)(i)	Award one mark for each: • True (1) • False (1)	Do not award 0, 1, Yes, No Allow T/F for True/False	(2)

Question Number	Answer	Additional Guidance	Mark
1(e)(ii)	The only correct answer is C *A is not correct because 0.5 is the fractional part of division* *B is not correct because 1 is the remainder after integer division, i.e. the result of modulus* *D is not correct because 2.5 is the result of division*		(1)

<table>
<tr><th>Question Number</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>1(f)(i)</td><td>Award one mark for each correct cell:

<table>
<tr><th>Algorithm type</th><th>Characteristic</th><th>Algorithm name</th></tr>
<tr><td>Search</td><td>Is a divide and conquer algorithm</td><td>Binary search (1)</td></tr>
<tr><td>Sort</td><td>Is not a divide and conquer algorithm</td><td>Bubble sort (1)</td></tr>
</table></td><td>Ignore exclusion of the words 'search' and 'sort' in the last column</td><td>(2)</td></tr>
</table>

Question Number	Answer	Additional Guidance	Mark
1(f)(ii)	Award up to two marks for a linked explanation, such as: • A sorting algorithm executes more quickly (1) because a small number of comparisons are made (1) • A sorting algorithm executes more slowly (1) because a large number of comparisons are made (1)	For both marks, the expansion must follow/associate with the statement.	(2)

<table>
<tr><th>Question Number</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>5(a)</td><td>Award one mark for each of:

• S = A AND C (1)
• T = B OR C (1)
• Correct values in column Q (1)

<table>
<tr><th>A</th><th>B</th><th>C</th><th>S</th><th>T</th><th>Q</th></tr>
<tr><td></td><td></td><td></td><td>A AND C</td><td>B OR C</td><td>NOT S AND T</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
</table>
</td><td>Ignore capitalisation</td><td>(3)</td></tr>
</table>

<table>
<tr><th>Question Number</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>5(b)(i)</td><td>Award one mark for each correct cell:

<table>
<tr><th></th><th>Line numbers</th></tr>
<tr><td>Traverses the records</td><td>17 (1)</td></tr>
<tr><td>Traverses the fields</td><td>19 (1)</td></tr>
</table>
</td><td></td><td>(2)</td></tr>
</table>

<table>
<tr><th>Question Number</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>5(b)(ii)</td><td>Award one mark for any of the following:

• if (item > currMax): (1)
• item > currMax (1)</td><td>Do not award relational symbol only, as there is no indication which line the error is on

Allow equivalent expressions:
item >= currMax
currMax < item
currMax <= item</td><td>(1)</td></tr>
</table>

<table>
<tr><th>Question Number</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>5(c)</td><td>Award one mark for each of the following:

• Number column is correct (goes from 1-6) (1)
• Rows where number is 1, 2, 3 are correct (1)
• Rows where number is 4, 5, 6 are correct (1)
• Single accurate output only (1)

<table>
<tr><th>number</th><th>remainder</th><th>total</th><th>output</th></tr>
<tr><td></td><td></td><td>0</td><td></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td></td></tr>
<tr><td>2</td><td>0</td><td>3</td><td></td></tr>
<tr><td>3</td><td>0</td><td>6</td><td></td></tr>
<tr><td>4</td><td>2</td><td>6</td><td></td></tr>
<tr><td>5</td><td>1</td><td>6</td><td></td></tr>
<tr><td>6</td><td>0</td><td>12</td><td></td></tr>
<tr><td></td><td></td><td></td><td>12</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>
</td><td>Interpret empty cells as unchanged values

Ignore spacing/empty cells on first bullet as long as numbers in sequence

Other formats of trace table are acceptable</td><td>(4)</td></tr>
</table>

Question Number	Answer	Additional Guidance	Mark
5(d)(i)	Award up to **two** marks for a linked explanation, such as: • When the target is the last item/is not in the array (1) because the number of comparisons equals the number of items in the array / every item in the array must be compared to the target (1)	For both marks, the expansion must follow/associate with the statement.	**(2)**

Question Number	Answer	Additional Guidance	Mark
5(d)(ii)	Award up to **two** marks for a linked explanation, such as: • Only one pass of the array is needed (1) because the array is already in order (1) • A single pass will be done (1) because after one pass it can detect if the array is already sorted (1) • No swaps are needed/detected (1) because the array is already sorted (1)	For both marks, the expansion must follow/associate with the statement.	**(2)**

Question Number	Answer	Additional Guidance	Mark
5(e)	Award **one** mark for each point for a maximum of **six** marks: • 0 and 10 in first row (1) • Evidence of a calculation for midpoint that would result in 5 shown in first row (1) • N and lower in first row (1) • 6, 10, calculation/8, N, higher in second row (1) Rounding down method • 6, 7, calculation/6, N, lower in third row (1) • 7, 7, calculation/7, N, higher/none/<blank> in fourth row (1) Rounding up method • 6, 7, calculation/7, N, higher in third row (1) • 6, 6, calculation/6, N, lower/none/<blank> in fourth row (1)	Penalise only once on bullet 2 if mid-point value given instead of calculation shown For BP2 there needs to be some evidence of a calculation that would yield the result 5 for the first midpoint. Allow midpoint calculation method to be either rounding down or rounding up.	**(6)**

Method 1: DIV 2 rounding down

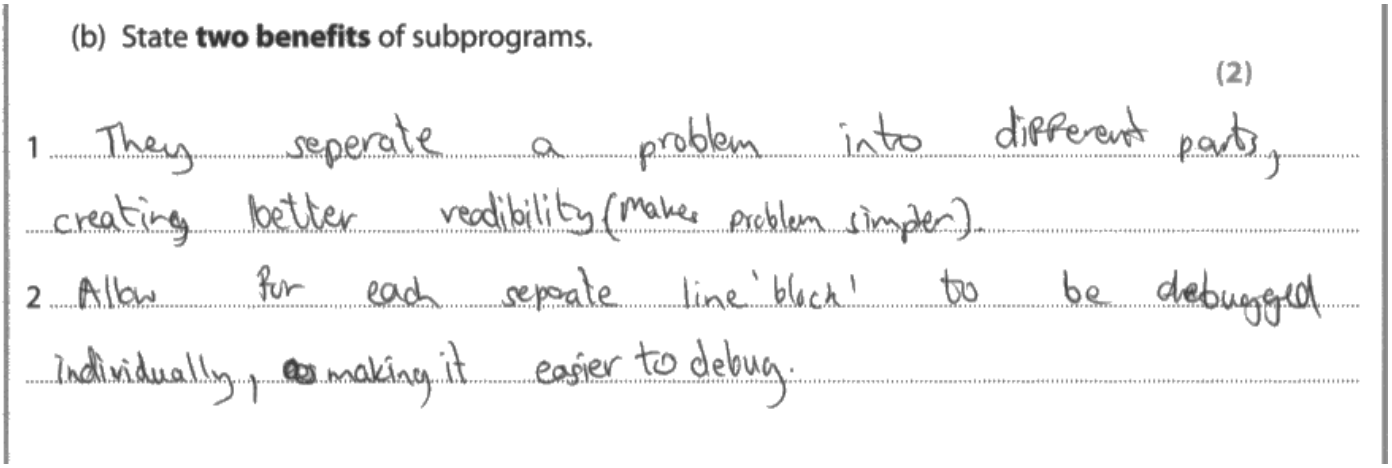
Start index	End index	Calculation of mid-point	Found Y or N	Discard lower, higher or none
0	10	(0 + 10) // 2 = 5	N	lower
6	10	(6 + 10) // 2 = 8	N	higher
6	7	(6 + 7) // 2 = 6	N	lower
7	7	(7 + 7) // 2 = 7	N	(higher) none/<blank>
7	6			

Method 2: rounding up midpoint calculation

Start index	End index	Calculation of mid-point	Found Y or N	Discard lower, higher or none
0	10	(0 + 10) / 2 = 5	N	lower
6	10	(6 + 10) / 2 = 8	N	higher
6	7	(6 + 7) / 2 = 6.5 → 7	N	higher
6	6	(6 + 6) / 2 = 6	N	(lower) none/<blank>
7	6			

Question 1 (b)

Many candidates demonstrated a good knowledge of the benefits of subprograms. Most answers related to code being written once or subprograms being easier to debug. Occasionally answers were an unqualified 'easier' or 'quicker' rather than a qualified example such as 'easier to debug'.



(b) State **two benefits** of subprograms.

(2)

1 They seperate a problem into different parts, creating better readibility (makes problem simpler).

2 Allow for each seperate line 'block' to be debugged individually, making it easier to debug.

ResultsPlus
Examiner Comments

This response scores 2 marks for stating improved code readability and then making code easier to test and debug.

Question 1 (c)(i)

Most candidates were able to correctly identify the line where a conditional loop started.

Question 1 (c)(ii)

Fewer candidates could identify an example of iteration compared to a conditional loop in the previous question. The most common mistake was not identifying the entire selection process and instead only listing the IF command on line 16.

Question 1 (c)(iii)

Many candidates were unable to identify selection or gave just 16 rather than the full range of lines that contained the selection.

Question 1 (d)(i)

Many candidates gave definitions that were vague or ambiguous. Stating spelling mistakes or mistakes in grammar in isolation is insufficient, and candidates need to make it clear that a syntax error means that the rules of a programming language have been broken. Candidates are required to give accurate technical definitions. Some confused syntax errors with logic or runtime errors, or gave ambiguous responses that could equally have applied to other categories of error. Where candidates gave examples of syntax errors they did not answer the question as they need to provide a definition and not an example.

(d) Programs can have syntax errors and runtime errors.

(i) Define the term 'syntax error'.

(1)

an error caused when the rules of a programming language are not followed.

ResultsPlus
Examiner Comments

This response scores 1 mark for defining that the rules of the programming language have not been followed.

Question 1 (d)(ii)

Many candidates confused a runtime error with a syntax error stopping execution of an interpreted program. A number of candidates erroneously thought that a runtime error meant that the computer had insufficient time to complete the execution of the program or that it took too long to run. Some candidates gained partial credit for identifying that the program would crash, but were then unable to give the reason why. An identification of a potential cause of a runtime error such as division by zero is not enough as an expansion. Candidates need to explain that such operations cannot be executed and that is why the error occurs.

(ii) Runtime errors happen when a program is executing.

Explain a runtime error.

(2)

Runtime errors occur when the computer is asked to perform an impossible task, such as dividing by zero or opening a non-existent file. They usually result in crashes, because the computer is unable to perform the task.

ResultsPlus
Examiner Comments

This response scores 2 marks for explaining that the computer will crash when it is asked to do an impossible task.

Question 1 (e)(i)

Candidates either tended to score 2 marks for knowing that a condition evaluates to a Boolean value of True/False or scored 0. A number of candidates erroneously described the greater than relational operator, or gave the output as yes/no rather than as a Boolean.

Question 1 (f)(i)

Many candidates identified the correct search and sort algorithm. Where candidates did not understand the term 'divide and conquer' they often cited linear search or merge sort which were incorrect responses.

(f) Programmers consider algorithm efficiency when they write code.

(i) Sorting and searching use algorithms.

Complete the table with the name of a search algorithm and a sort algorithm.

(2)

Bub sort
Merge sort
linear search
Binary search

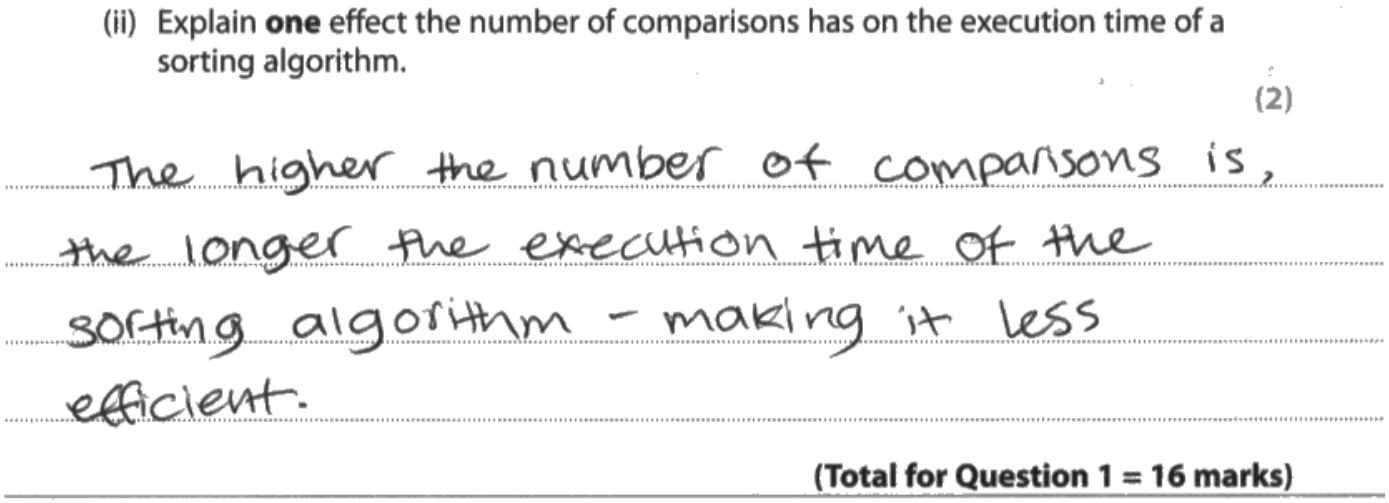
Algorithm type	Characteristic	Algorithm name
Search	Is a divide and conquer algorithm	Binary search
Sort	Is **not** a divide and conquer algorithm	Merge sort

ResultsPlus
Examiner Comments

This response scores 1 mark for correctly identifying binary search as a divide and conquer algorithm. Merge sort was an erroneous answer because it is also a divide and conquer algorithm.

Question 1 (f)(ii)

Most candidates scored full marks for this question for correctly identifying the relationship between the number of comparisons and execution time.



(ii) Explain **one** effect the number of comparisons has on the execution time of a sorting algorithm.

(2)

The higher the number of comparisons is, the longer the execution time of the sorting algorithm - making it less efficient.

(Total for Question 1 = 16 marks)

ResultsPlus
Examiner Comments

This response scores 2 marks for correctly explaining the relationship between an increased number of comparisons and increased execution time.

Question 5(a)

Many candidates found it difficult to derive the correct expressions for S and T in the truth table. Where candidates struggled to derive the expressions, they were often able to complete the actual truth table values.

Question 5(b)i

Most candidates attempted this question, but relatively few could identify both lines of code that were required.

Question 5(b)ii

Many candidates who attempted most of Q5 were able to identify the correct expression, but there were many candidates who omitted a response to this question.

Question 5(c)

It was encouraging to see many candidates attempt the trace table at this stage in the paper. When candidates attempted the question, it was noticeable that many iterated the loop too many times and traced *number* from 1 to 7. Remainders were more accurately given in rows 1 to 3 where numbers divided exactly, but in rows 4 to 6 more errors occurred. Some candidates gave outputs on each line rather than giving the final output which would only be printed at the end of the algorithm.

Complete the trace table.

You may not need to fill in all the rows in the table.

(4)

number	remainder	total	Output
		0	
1	0	1	
2	0	3	
3	0	6	
4	2		
5	1		
6	0	12	
7	-1		
			12

ResultsPlus Examiner Comments

This response scored 3 marks.

It did not get mark point 1 because it iterated numbers from 1 to 7 instead of 1 to 6. It did get mark points 2 and 3 for correctly tracing the intervening rows. It also achieved mark point 4 for giving the final output of 12 once only at the end.

Question 5(d)i

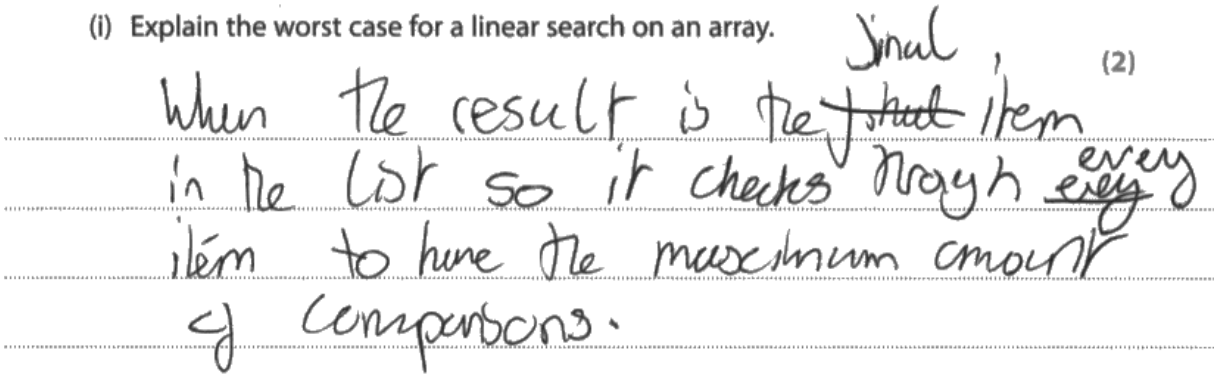
Whilst many candidates identified that the worst case for a linear search was when the item to locate was either the last item or not in the list, fewer were able to give the reason for this.

(d) The efficiency of algorithms can be analysed.

The worst case is when the most comparisons are required.

The best case is when the fewest comparisons are required.

(i) Explain the worst case for a linear search on an array. (2)



ResultsPlus
Examiner Comments

This response scored 2 marks for giving a clear linked explanation that justified the worst case being that the item was the last in the list because all preceding items had to be compared first.

Question 5(d)ii

Answers to this question tended to mirror Q5d(i) with many identifying that the best case for a bubble sort was when the data was already in order. Fewer were able to give the reason for this. Many candidates reiterated the wording of the question stating that the best case was when the fewest comparisons were needed. Responses that reiterated the question were not creditworthy.

(ii) Explain the best case for a bubble sort on an array.

(2)

The best case is that it only needs one pass to be in order.

ResultsPlus
Examiner Comments

This response scored 1 mark for stating that the best case was when the list was already in order, but there was no linked explanation to explain why this was the case.

Question 5(e)

Encouragingly, most candidates attempted the last question on the paper. Many achieved the first mark or two for tracing the initialisation of the start and end index and stating whether the item was found or not. Fewer were then able to update the start/end indexes correctly to reflect the part of the list that would be discarded and fewer still were able to trace the algorithm to the point when the algorithm would stop. Another common error was to continue to include the midpoint rather than discarding it when deciding on the bounds for the new sublist.

Start index	End index	Calculation of mid-point	Found Y or N	Discard lower, higher or none
[0]	[10]	(0+10)÷2 = [5]	N	discard lower
[5]	[10]	(5+10)÷2 = 7.5 → [8]	N	discard higher
[8]	[6]	(8+6)÷2 = 7 [7]	N	discard higher
[7]	[6]	(7+6)÷2 = 6.5 → [6]	N	discard none. (error)

ResultsPlus Examiner Comments

This response, like many, scored the first 3 marks.

Where candidates had an idea as to how a binary search calculated a midpoint and then discarded the half of the list that did not contain the target item, they scored 3 marks. More detailed knowledge of the specific way in which the upper/lower bounds are updated in a binary search was required to update the start/end indexes accurately to continue the algorithm.

Paper Summary

Based on their performance on this paper, candidates are offered the following advice:

- Learn and use clear definitions for technical terms within the specification.
- Read and interpret the question prompt to focus on the specific requirements and provide more relevant responses that address the critical aspects of the question.
- Where a context or scenario is provided, respond by using the contextual information in the question.
- Expand and explain answers using examples and reasons, especially where more than a simple statement or list is requested.
- Understand that characteristics are properties or features.
- Clarify understanding of the differences between high and low level languages.
- Develop clear understanding between the difference between different searching and sorting algorithms.

Grade boundaries

Grade boundaries for this, and all other papers, can be found on the website on this link: https://qualifications.pearson.com/en/support/support-topics/results-certification/grade-boundaries.html