GCSE COMPUTER SCIENCE

1.2

Edexcel 1CP2 | Specification point 1.2.5

Types of errors that can occur in programs (syntax, logic, runtime) and be able to identify and correct...

Untiered

Answers and examiner comments

2	2019-2024	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.

Question Number	Answer	Additional Guidance	Mark
5(a)	An explanation that makes reference to the following point: • This is a runtime error (1) Plus **one** from: • Because it occurs when the computer tries to execute the code (1) • Because the (translated) code cannot be executed (1) • Because it crashes/stops (1) (the computer when attempting to execute) • Occur intermittently, depending on the data that is encountered (1) • Can be caused when a user inputs data of the wrong type (1)		**2**

Question Number	Answer	Additional Guidance	Mark
5(b)	Line 6 • Error: Only one short side is needed/two added to width (1) • Correction: SET totalWidth TO width (1) Line 8 • Error: Lengths of the sides are multiplied (1) • Correction: SET total TO totalWidth + totalLength (1)	• Do not penalise syntax errors as long as discernible	**4**

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:

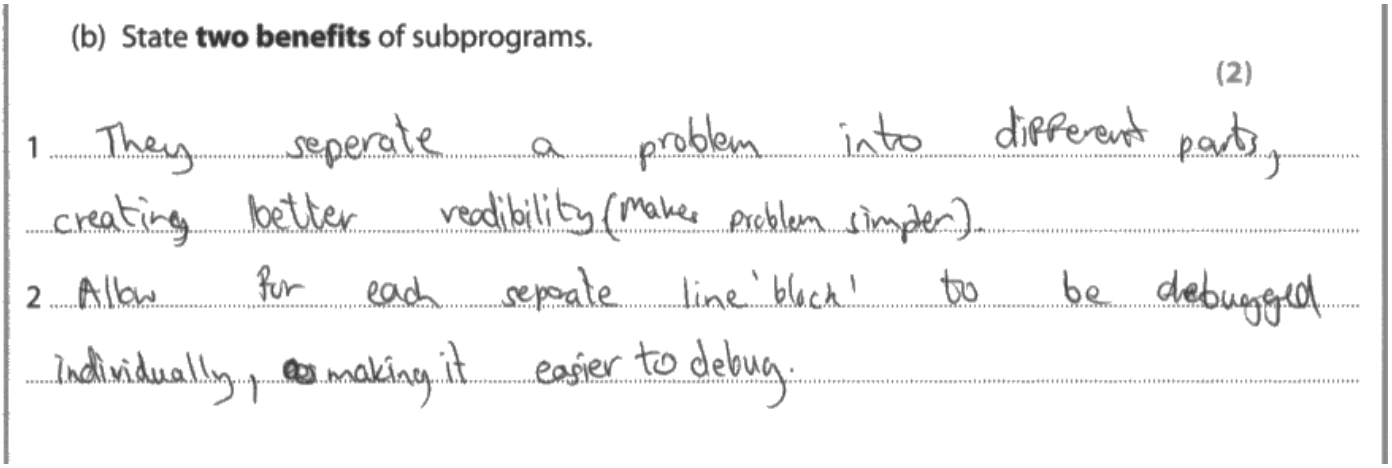
<table><tr><th>Algorithm type</th><th>Characteristic</th><th>Algorithm name</th></tr><tr><td>Search</td><td>Is a divide and conquer algorithm</td><td>Binary search (1)</td></tr><tr><td>Sort</td><td>Is not a divide and conquer algorithm</td><td>Bubble sort (1)</td></tr></table></td><td>Ignore exclusion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

<table>
<tr><th>Question Number</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>1(f)(ii)</td><td>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)</td><td>For both marks, the expansion must follow/associate with the statement.</td><td>(2)</td></tr>
</table>

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~~ 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

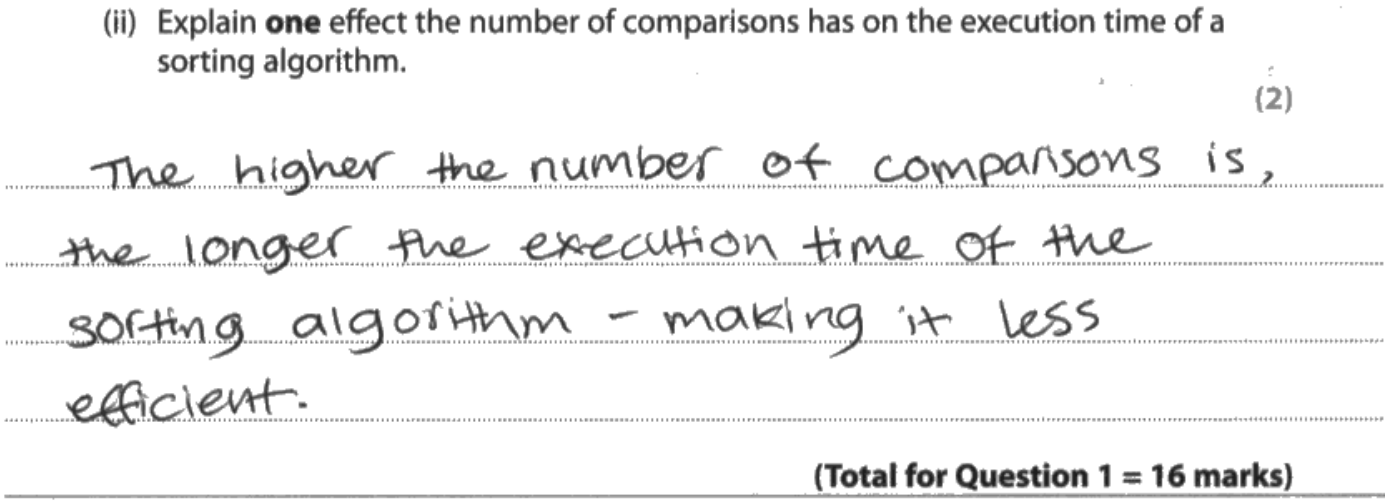
Algorithm type	Characteristic	Algorithm name
Search	Is a divide and conquer algorithm	Binary search
Sort	Is **not** a divide and conquer algorithm	Merge sort

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.