Qualification Accredited

Oxford Cambridge and RSA

GCSE (9–1)

Examiners' report

COMPUTER SCIENCE

J276

For first teaching in 2016

J276/02 Autumn 2021 series

Version 1

ocr.org.uk/computerscience

Introduction

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

i	Reports for the November 2021 series will provide a broad commentary about candidate performance, with the aim for them to be useful future teaching tools. As an exception for this series they will not contain any questions from the question paper nor examples of candidate responses.

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. 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.

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

Would you prefer a Word version?

Did you know that you can save this PDF as a Word file using Acrobat Professional?

Simply click on **File > Export to** and select **Microsoft Word**

(If you have opened this PDF in your browser you will need to save it first. Simply right click anywhere on the page and select **Save as . . .** to save the PDF. Then open the PDF in Acrobat Professional.)

If you do not have access to Acrobat Professional there are a number of **free** applications available that will also convert PDF to Word (search for PDF to Word converter).

© OCR 2022

Paper 2 series overview

J276/02 (Computational thinking, algorithms and programming) is one of two examination components for the GCSE Computer Science. This component focuses on:

- algorithms
- programming techniques
- producing robust programs
- computational logic
- translators and facilities of languages
- data representation.

Candidates who did well on this paper were able to write, complete and use algorithms using pseudocode and/or flowcharts confidently. This involves applying their knowledge to unfamiliar contexts.

Confident responses reflected the candidate's experience of practical programming in a high-level language. For example, practical programming tasks allow candidates to practise their programming skills in a range of scenarios.

Candidates' confidence in programming allowed them to apply their knowledge within the contexts given in the questions.

Candidates who did less well on this paper produced answers that were generic or lacked the detail needed to clarify the steps needed for algorithms to be successful.

Candidates generally performed very well on Questions 3(b), 3(c), 3(d), 5(a), 5(c), 5(e) and 6(a).

Candidates did not seem to be so successful on Questions 2(b), 3(a), 3(c)(ii), 4(b), 4(c), 5(d) and 6(d).

Centres should be aware of the appendices at the rear of the specification. Section 5f shows the examination format of pseudocode and Boolean logic.

Candidates' answers are accepted in any logical form. Candidates do not have to present answers in a specific form. Candidates should be aware of these conventions to successfully understand and access examination questions.

The number of candidates taking this examination was much smaller than in a standard summer series and smaller than the equivalent examination in 2020.

This series is the last sitting of the J276 specification. All future GCSE Computer Science entries must be for the new J277 specification. Please carefully read the new J277 specification to familiarise yourself with this for the summer 2022 series.

Candidates who did well on this paper generally did the following:	***Candidates who did less well on this paper generally did the following:***
• Used the scenario given where appropriate, such as Question 2(b) where candidates were asked about a binary search on a specific list. • Were able to clearly define and communicate the steps that algorithms take to solve a problem, such as Questions 3(a), 3(c)(ii) and 6(d). • Showed a strong understanding of data representation in Questions 1 and 5 and were able to communicate this understanding succinctly and with clarity.	• Gave generic answers that did not focus on or link to the scenario given. • Struggled to decompose algorithm questions, leading to responses that repeated the question or gave a very high-level explanation rather than showing how each step could be solved. • Showed poor understanding of string manipulation in Question 4(b) and 4(c).

© OCR 2022

Comments on responses by question type

Questions where candidates write or follow algorithms

Question 3(a) required candidates to implement an algorithm. The algorithm takes a numerical input, adds it to a total and repeats this until the total is over 100.

Most candidates were comfortable with basic input and output of values. Many candidates were able to attempt to implement a loop. Fewer candidates were familiar with how to continually add to a total.

A number of candidates did not appear to appreciate that while `print(total + x)` may **display** the sum of the two variables, it does not add the values together and then store them.

?	**Misconception**	`print(total + x)` will output the sum of the two variables without changing either. Instead `total = total + x` (or in some languages, `total += x`) will modify the variable total, which can then be printed if needed.

Question 3(c)(i) asked candidates to complete a flowchart for an experiment. Many candidates were able to gain some marks for:

- initialising the 'Count' variable
- outputting this at the end
- repeating if 10 values had not been asked.

Some candidates did not complete the flowchart. They did not successfully join the elements together correctly from top to bottom.

Only high achieving candidates were able to correctly decide where the NO output from the first decision box should lead to. The correct response was to skip over the increment but then complete the second decision.

	AfL	The use of flowcharts to communicate algorithms should be taught as an alternative to pseudocode and students assessed on the logical flow expressed in these.

Question 3(c)(ii) utilised similar understanding to Question 3(a), but utilised the loop in a different way.

This time, the algorithm was required to repeat a set number of times and so a count-controlled loop would be most appropriate. Many candidates did this successfully and were credited for their answers.

Where candidates used condition-controlled loops or other methods (such as recursive calls to a function), these were also credited as long as they were logically correct.

The loop had to repeat 10 times to be correct. Benefit of doubt is given where languages such as Python are used, where defining this precisely could be ambiguous.

A small number of candidates attempted to write out the algorithm ten times which did not meet the requirements of the question.

	AfL	Centres should be encouraged to teach about iteration using both count-controlled and condition-controlled loops, comparing and contrasting the use of these for various situations. Rewriting a count-controlled loop to use a condition instead (and therefore manually incrementing the counter variable) is a relevant challenge for students.

© OCR 2022

Question 6(d) asked candidates to write an algorithm to allow a player to take part in a block-based game. Part of the algorithm required the use of a pre-defined function to decide if a block was free or not.

The use of this function was established in the previous four parts to the question, so ensuring that candidates understood the purpose of the function.

The majority of candidates were unable to use the function provided for its intended purpose however.

Some candidates ignored this requirement (and were able to gain other marks on the question, such as inputting values and looping).

Most candidates attempted to rewrite the function, copying out the code given in the stem of the question and this did not meet the requirements of the question.

Only a small number of candidates were able to call the function with the values inputted by the user and then store (or use) the returned value.

?	**Misconception**	Although many candidates can explain what a function is and define functions that return values, very few appear confident when using a pre-defined function. Candidates can practise using pre-defined functions as part of the programming skills requirements. Candidates are often unaware that they use pre-defined functions regularly. For example print() or len() could be used to introduce the topic of pre-defined functions.

i	**OCR support**	Appendix 5f of the current J276 specification covers the pseudocode guide for this examination. Candidates do not need to use this in their responses but they should be aware of this as questions will be presented using this format. For the new J277 specification, this has been replaced by OCR Exam Reference Language. This is detailed from page 25 onwards.

Other questions

Question 3(c) (part i, ii and iii) asked candidates about sound sampling. Most candidates gained some marks for this questions. Very few candidates gave answers that achieved full marks.

Teachers should make sure that candidates are aware that a question worth (for example) 4 marks would require 4 separate points or follow on points to gain all 4 marks. An answer to 3(c)(iii) such as "it could be compressed" would be correct, yet only achieve 1 mark.

Another common issue with sound sampling is the data that is actually sampled (Question 3(c)(i)). A number of candidates continue to claim that frequency is sampled, which is incorrect. Frequency refers to either how often the wave form oscillates or how often the sample is taken and this is not recorded. Instead, it is the amplitude or height of the waveform that is measured for one sample.

?	**Misconception**	When sound is sampled, the **height** or **amplitude** of the waveform at that point is recorded – "the frequency" would be incorrect.

© OCR 2022

Question 2(b) asked candidates to explain how a binary search would work to find the value "house" in the given list. Where candidates did this well and explained each step, using the list given, full marks were often attained.

However, a number of candidates instead ignored this list and gave a generic answer such as "pick the middle value, if this is not the one you want, check if this value is larger or smaller than the one you are looking for". This did not hit many points on the mark scheme.

A much better answer would refer to the middle value in the given list ("pumpkin"), noting that "house" is alphabetically before this so the right half of the list could be removed / ignored, leaving the new list as ["flour", "house"]. A candidate could then go on from this point to achieve full marks.

A small number of candidates confused linear and binary search. Other candidates referred to splitting the list in half without discussing any form of comparison.

	AfL	It is important to link scenarios to responses when required. This is particularly important if a question specifically asks for this as part of the answer (such as Question 2(d) above, "Explain how a binary search would be used to try to find whether the word "**house"** appears in **this list**.")

Question 3(d) asked candidates to describe computational thinking techniques that had been used to develop the algorithm. This was done extremely well by the majority of candidates and it is clear that this has been taught very well nationally.

Most candidates were able to name two suitable techniques and many were also able to describe what these techniques are used for.

Question 4(c) asked candidates to use string manipulation to output specific words or sentences from given values. The mark scheme given shows answers using `substring()` which is the method given in the specification pseudocode guide.

Candidates are able to use any valid string manipulation techniques from any relevant high-level language or any sensible pseudocode. For example, where the variable `second` contains the string "`is great`", the output "`great`" could be produced with `right(second, 5)` , `second[3:]`, or any other sensible manipulation.

Examiners attempted to be very generous with their marking of this question. However, candidates generally appeared to be unfamiliar with substring operations.

?	**Misconception**	String manipulation can join together (concatenate) strings using the + sign. However, the – sign cannot be used to remove values from strings; it would not be appropriate to expect that `is great - is` would give "great".

Question 5(a) and 5(c) were generally answered very well. Candidates and teachers are now well used to converting values into binary and hexadecimal and this has obviously been taught very well. Many candidates achieved full marks on these questions.

Question 5(d) showed that many candidates did not understand the use of hexadecimal. All values in a computer system are stored/processed using electronic switches (transistors) and these can be represented for humans as binary, with 1 and 0 being switches that are on or off.

Hexadecimal is a short cut for programmers to represent large binary values in a shorter amount of space. For example, a large binary number 1010 1001 1111 0011 could instead be represented as A9F3. This does not change how the underlying value is stored in the computer, but simply makes it easy for the human user to communicate or use.

© OCR 2022

?	**Misconception**	All data in a computer system is stored/processed using on/off switches known as transistors. Hexadecimal is an alternative way of representing the state of these switches, **NOT** a different method of storing data in a computer system.

Key teaching and learning points – comments on improving performance

To improve performance across this paper, teachers and students should focus on:

- Ensuring enough programming practice takes place as part of the curriculum delivery so that candidates are confident with both writing their own algorithms and tracing through algorithms given in pseudocode.
- For the updated J277 specification, candidates must be prepared to complete design, write, test and refine algorithms. Some of these questions require use of a high-level language (or OCR Exam Reference Language).
- Ensuring that candidates and teachers are familiar with how algorithms are presented in examinations. This style used is given within the specification to help support students and teachers.
- Where contexts or scenarios are given in questions, candidates should use or refer to this in their responses if appropriate.
- Ensuring that the purpose of hexadecimal as a human-focused shortcut is understood by students.

Guidance on using this paper as a mock

When marking the answers take note of the annotations in the mark scheme.

- Marks are usually given as 1 per bullet point, unless otherwise stated.
- If a candidate gives two answers that meet the same bullet point, then only 1 mark is given.
- The use of a // on a line indicates an alternative answer for that bullet point. Only 1 mark can be given even if multiple answers that hit this point are given.
- The use of a / on a line indicates an alternative word e.g. Question 1(c)(i) height / amplitude. "measures the height" and "records the amplitude" would both hit this mark point, but a maximum of 1 mark should be given even if both answers are present.
- The use of bold indicates that this idea must be present, although it could be expressed in a variety of ways. For example in Question 2(b) 'Compare this to house, **no match**'. The words 'no match' does not need to be present but the idea that the value is compared with house and that it doesn't match is important.

For algorithm questions, the following guidance should be considered:

- Candidate answers do not have to be in any specified language or format. Do not mark something wrong because "it wouldn't work in Python / VB / Java".
- Answers must show the basic construct to use. For example, Question 3(c)(ii) requires candidates to use iteration; this could be a FOR loop, a WHILE loop or another sensible method of demonstrating that code is repeated. It would not be enough to simply answer "loop around 10 times" – an examiner would quite rightly ask "how?" to this.
- Structured English and flowcharts are both acceptable methods of answering algorithm questions, but the candidate answer **must** show the steps to be taken to be credited with marks. Too often, candidates using these methods simply repeat the question and do not show how the problem will be solved. Be aware that J277 papers will include a Section B where candidates are required to answer in a high-level language or OCR Exam Reference Language.
- Ignore any superfluous or additional code given as long as it does not contradict other parts of the answer.

© OCR 2022

Supporting you

Review of results

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

Supporting you through 2021-2022

Our priority is supporting you and your students this autumn and to support you as you prepare for summer 2022 exams. We'll update our website information regularly with resources, guidance and key information.

Take a look at our support for:

- Teachers
- Students
- Exams officers

Keep up-to-date

We are sending a weekly roundup 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 CPD courses to hear directly from a senior assessor or drop in to a Q&A session. All our courses for the academic year 2021-2022 are being delivered live via an online platform, so you can attend from any location.

Please find details for all our courses on the relevant subject page on our website or visit OCR professional development.

Signed up for Exambuilder?

ExamBuilder is the question builder platform for a range of our GCSE, A Level, Cambridge Nationals, Cambridge Technicals and Functional Skills qualifications. See the full list of available qualifications in the sign up form.

ExamBuilder is **free for all OCR centres** with an Interchange account and gives you unlimited users per centre. We need an Interchange username to validate the identity of your centre's first user account for ExamBuilder.

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

Supporting you

Active Results

Review students' exam performance with our free online results analysis tool.

For the Autumn 2021 series, results analysis is available for GCSE English Language, GCSE Mathematics and Cambridge Nationals (moderated units) only.

It allows you to:

- review and run analysis reports on exam performance
- analyse results at question and/or topic level
- compare your centre with OCR national averages
- identify trends across the centre
- facilitate effective planning and delivery of courses
- identify areas of the curriculum where students excel or struggle
- help pinpoint strengths and weaknesses of students and teaching departments.

Find out more at ocr.org.uk/activeresults.

Need to get in touch?

If you ever have any questions about OCR qualifications or services (including administration, logistics and teaching) please feel free to get in touch with our customer support centre.

Call us on
01223 553998

Alternatively, you can email us on
support@ocr.org.uk

For more information visit

- **ocr.org.uk/qualifications/resource-finder**
- **ocr.org.uk**
- **/ocrexams**
- **/ocrexams**
- **/company/ocr**
- **/ocrexams**

We really value your feedback

Click to send us an autogenerated email about this resource. Add comments if you want to. Let us know how we can improve this resource or what else you need. Your email address will not be used or shared for any marketing purposes.

I like this **I dislike this**

CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT

OCR is part of Cambridge University Press & Assessment, a department of the University of Cambridge.

For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored. © OCR 2021 Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee. Registered in England. Registered office The Triangle Building, Shaftesbury Road, Cambridge, CB2 8EA. Registered company number 3484466. OCR is an exempt charity.

OCR operates academic and vocational qualifications regulated by Ofqual, Qualifications Wales and CCEA as listed in their qualifications registers including A Levels, GCSEs, Cambridge Technicals and Cambridge Nationals.

OCR provides resources to help you deliver our qualifications. These resources do not represent any particular teaching method we expect you to use. We update our resources regularly and aim to make sure content is accurate but please check the OCR website so that you have the most up to date version. OCR cannot be held responsible for any errors or omissions in these resources.

Though we make every effort to check our resources, there may be contradictions between published support and the specification, so it is important that you always use information in the latest specification. We indicate any specification changes within the document itself, change the version number and provide a summary of the changes. If you do notice a discrepancy between the specification and a resource, please contact us.

You can copy and distribute this resource freely if you keep the OCR logo and this small print intact and you acknowledge OCR as the originator of the resource.

OCR acknowledges the use of the following content: N/A

Whether you already offer OCR qualifications, are new to OCR or are thinking about switching, you can request more information using our Expression of Interest form.

Please get in touch if you want to discuss the accessibility of resources we offer to support you in delivering our qualifications.