



Examiners' Report

Principal Examiner Feedback

Summer 2024

Pearson Edexcel International GCSE in
Computer Science Paper 2 (4CP0_02)

Edexcel and BTEC Qualifications

Edexcel and BTEC qualifications are awarded by Pearson, the UK's largest awarding body. We provide a wide range of qualifications including academic, vocational, occupational and specific programmes for employers. For further information visit our qualifications websites at www.edexcel.com or www.btec.co.uk. Alternatively, you can get in touch with us using the details on our contact us page at www.edexcel.com/contactus.

Pearson: helping people progress, everywhere

Pearson aspires to be the world's leading learning company. Our aim is to help everyone progress in their lives through education. We believe in every kind of learning, for all kinds of people, wherever they are in the world. We've been involved in education for over 150 years, and by working across 70 countries, in 100 languages, we have built an international reputation for our commitment to high standards and raising achievement through innovation in education. Find out more about how we can help you and your students at: www.pearson.com/uk

Summer 2024

Publications Code 4CP0_02_2406

All the material in this publication is copyright

© Pearson Education Ltd 2024

Introduction

- There were three multiple-choice questions included in the question paper.
 - 1(a) and 1(d) were answered well with most candidates knowing the meaning of abstraction and that a record contains different data types.
 - 2(a)(i) was also quite well answered, but some candidates did confuse pseudocode with the definition for an algorithm.
- Single mark questions were also generally well answered.
 - Question 1(c) was mostly well done with almost all candidates giving correct responses for 1(c)(i) and 1(c)(ii), although candidates were less able to accurately identify the selection keyword in 1(c)(iii), confusing it with 'while'. Some also lost marks by confusing the logical operator 'and' with a relational operator in 1(c)(iv).
 - Question 1(e); very few candidates lost marks as almost all could identify the two data types correctly.
 - In questions 2(a)(ii), 4(a)(i) and 4(a)(ii) most candidates were able to identify a variable, and understood the difference between global and local scope.
 - Question 2(b)(i) was not as well done as most candidates simply repeated the question about a **runtime error** e.g. "an error that occurs when code is running". Few candidates added any further detail. Most candidates were however able to give an example of a runtime error for 2(b)(ii), most identifying a divide by zero error.
- Multiple mark questions had mixed outcomes, with one question particularly poorly done while another was very well done.
 - Question 2(d) was quite well done, and most candidates were able to give correct examples of erroneous, normal and boundary test data. Where marks were commonly lost, candidates had repeated the same type of erroneous data e.g. two entries that were both too many characters or two entries that both contained data that was not just letters. Few candidates scored less than 3 marks in this question.
 - Question 3(a), where candidates were asked to complete a truth table, was very well done with most candidates scoring all 4 marks. This topic is clearly being well taught by centres.
 - Candidates performed generally poorly in Question 4(b) with few candidates gaining more than 1 of the 2 available marks, and many scoring zero. Very few candidates were able to identify that a nested selection statement ended as soon as a condition was found to be true, whereas the multiple if statements in Figure 5 would all be tested regardless of the outcome of previous tests. Most candidates understood that the two algorithms performed the same task, but their responses focused on the complexity of the code in Figure 5 e.g., 'more coding needed' or 'harder to understand the code' rather than the efficiency of executing the code as required by the question.
 - Candidates were able to understand the poor design of the code in figure 6 with most gaining at least 1 of the 2 marks available in question 5(a). Candidates knew that iteration was required, but not how to achieve it (through an array), or knew that an array would be a better structure, but not why (allow iteration). Higher ability candidates were able to put both parts together and achieve both marks.

REPORT ON INDIVIDUAL QUESTIONS

Examiners found that candidates responded very well to the coding challenges presented in the question paper. There were many candidates who scored full, or close to full marks on these questions. Most candidates made some attempt at every question.

- 1 (b) This question was generally well answered with most candidates obtaining most marks. The most often lost mark in this question was through candidates not initiating the `myAge` variable to zero. This happened more regularly than might have been expected.
- 2 (c) Most candidates scored all 3 marks in this question. Almost all candidates found the two syntax errors, and where a mark was lost it was almost always in failing to identify the logical error where modulus was used instead of multiplication.
- 3 (b) This question often acted as an indicator of performance in the remaining code questions. Most candidates were able to iterate over the array and output star names to the console. Many were able to print out every other name, and of those there was a relatively even split between those who completed this through adding further code, either by adding 2 to the counter or by adding a selection statement, and those who knew how to implement a step within the iteration itself.
- 3 (c) Another question that was generally well done. Candidates were able to take input data and change data types as well as identifying that the `if` was missing from the selection statement. Most were able to accurately complete the remaining tests with relational and logical operators. A few candidates lost marks by missing the edge values when completing the relational tests e.g. `>31` rather than `>=31`. However, the most common error was not using the correct logical operators, with some candidates using 'and' for both number ranges.
- 4 (c) Although many candidates gained full marks in this question, many others struggled to obtain more than one or two marks. A significant number of candidates did not seem to be familiar with the creation and calling of subprograms, with many candidates ignoring the given function altogether and trying to implement the string manipulation from within the main program. Others who did attempt to use the subprogram failed to use the given parameters, omitted the return, or called the function using the parameter names as arguments rather than the global variables given in the main program. For the string manipulation element of the question, many candidates were able to slice the string into the two-character elements required. For those who performed slicing, most were able to complete the concatenation successfully. Some candidates used four separate indexes to obtain the individual characters, many of whom did successfully concatenate the required key. Others used f-strings with the four indexes to create the final key. Most candidates were able to obtain at least one of the characters through the use of an index, but many were not able to move beyond that point.

- 5 (c) This question proved to be the most challenging question for most candidates. File handling still proves to be a difficult task for many. Most candidates were able to open and close the file, and knew to iterate over the file pointer, but many were not able to progress beyond this point. Some candidates struggled to match their code to the existing code given e.g. 'line' was used as the string that needed to be split, but candidates often did not use this variable name when iterating over the file. Some candidates did know that they had to split the string, but only used the split() default (whitespace) without specifying "," or the constant COMMA. For those candidates who were able to split the string and create the stringSales array, most were able to complete the rest of the program. Few candidates used the constant and variable names given in the program as required by the question thus losing a simple mark.
- 6 The final question was done relatively well, with almost all candidates making a reasonable attempt and many gaining all or most available marks. There are 9 level-based marks available in this question, with one of the considerations being that candidate code should be able to work on different data sets to that given in the question. The first task was to overwrite the two empty strings in the two-dimensional array. Many candidates hard-coded the replacement of the empty strings to index 10 in the array. While this did get the first mark, it may have impacted the levels-based marks awarded. Other candidates overwrote the last element in the array or searched for the empty strings as part of an iteration, both of which would work with alternate data sets. Most candidates were able to complete all tasks required by the question within a single iteration of the two-dimensional array. This will also be considered in the level-based marks awarded. Most candidates were able to use the length of the array to iterate over it and to use two-dimensional indexing to access the individual words within the inner arrays. Many also successfully used length functions to identify and output the longest word in each pair. Fewer candidates were able to compare the two words alphabetically and reorder them as required. Many candidates seemed unaware that relational operators will work with strings, based on alphabetical order, and created complicated loops where the first letters of each word were used to iterate over all letters of the alphabet searching for matches. Some candidates seemed to think that relational operators compared string length rather than order, resulting in them comparing alphabetically but incorrectly treating the result as if it were the length.

