Questions matter

GCSE

Computer Science

8525/1A/B/C Computational thinking and problem solving

Report on the Examination

8525
June 2024

Version: 1.0

Further copies of this Report are available from aqa.org.uk

Introduction

In general, the students seemed well prepared for this examination paper and showed a good application of the problem solving and computational thinking skills they have been taught.

The coding questions were well answered overall, in all three languages, however there is still some confusion shown when using the indentation grids that have been provided for coded responses. It is especially important for Python centres to work with their students on the correct use of the indentation grids within the answer spaces as these are provided to support them in making their indentation clear, but in many cases it is not being used as intended and the student's indentation is not always clear as a result.

With the coding questions, students should be careful either to output the requested values/messages or return a value as required by the question. Failure to provide the required items could result in marks not being awarded. A common error continues to be using the word `OUTPUT` instead of a correct statement in all languages.

Students continue to use mathematical symbols such as ≥ incorrectly which would not work in programming languages. Some students were also not gaining marks due to confusing the symbols (using greater than instead of less than and vice versa) and/or misunderstanding what 'inclusive' means.

As with last year's paper, another key point to take forward would be to ensure students read the question carefully as some student responses indicated that this was not always happening.

Question 1

This question set was about variable assignment. It was encouraging to see that over 90% of students were able to identify a suitable data type for the variable (Q1.3) and nearly 90% were able to correctly apply the length function to a string (Q1.1).

Question 1.2 asked the students to complete a statement using a provided subroutine. Although the subroutine was explained and two examples given for students to follow, only just over 40% of students were able to correctly complete the statement. The question stated that the `POSITION` subroutine should be used in the answer which a significant portion of students failed to do.

Part 1.4 asked for a description of an assignment statement, this was well answered with just over 65% of students gaining the mark available. Many students who did not gain the mark gave answers that were too vague such as "assigning something to a variable"; students should be encouraged to learn the specific terms so that they can use them in their answers.

This question block also had the first of the coding questions for this year's paper, a straightforward assignment and print. It was pleasing to see over 90% of students gaining at least one of the two available marks with common mistakes being using `OUTPUT` rather than a language specific statement or the incorrect joining of a string within the output statement.

Question 2

This set of questions focused on testing different values within an algorithm and determining the output based on specific inputs. It was pleasing to see approximately 70% of the students getting these questions correct, the exception being question 2.4 where only 20% of students were able to rewrite a line of code from the algorithm. The question stated the answer should not use the `NOT` operator which many students still used. The most common mistake was excluding the value 1 in the response. Boundary values have proved to be an issue in previous papers and students should be encouraged to double check their data.

Question 3

Question 3 looked at a short program, this time focusing on errors and testing.

Just over half of the students gained both of the available marks when assigning a random number to a variable and 75% gained at least one. Once again, the format of the function was given in the question along with an explanation of the parameters, but many students used `(1, 100)` or `(a, b)` in their answer.

Most students were able to correctly identify some of the test data with the boundary test causing the most issues. Most were able to describe the different errors in the line of code. Whilst examiners saw some very good descriptions of both syntax and logical errors this did not answer the question that was asked, and answers of this nature were not creditworthy.

Question 4

Question 4 looked at some of the standard definitions and whilst it was pleasing to see over 80% of students successfully identifying decomposition, a lot of the descriptions for abstraction either referred to removing elements from a program/code rather than from the problem, or just stated removing unnecessary detail without qualifying what it is removed from. These definitions are listed in the specification and are regularly tested on the exam papers. Students should be familiar with the wording of the definitions to support their answering of these questions.

Question 5

This was the first of the trace tables on this year's paper and was well answered: 60% of students gained all three available marks and over 80% gained at least one.

Question 6

This was the first of the main language specific questions on the paper that asked the student to write a program. It was pleasing to see that over 50% of the students gained five or more of the available seven marks on this question. A common error was to miss out the selection statement to check if the value were less than 0 and then setting the value to 0 if the statement was true.

Question 7

This language specific question tested indexing of characters in a string: only 20% of students gained all four available marks, the most common issue was students not knowing the correct syntax for the substring function, but many of those who did know it used arguments starting from one, not zero. A

further issue is the incorrect use of a comma for concatenation. Students should understand the difference between true concatenation and the use of a comma when outputting strings in an output statement.

Question 8

Question 8 was the second trace table and tested the student's ability to work with lists/arrays. Over a third of the students gained all the available marks with over 80% getting at least 1. The most common mistake was to put the output for the week's total inside the loop.

Question 9

This was testing students' understanding of records using an algorithm. Teachers are advised to use the resources that are available to them through the AQA website as it was clear that few students were prepared for this type of question with only a third gaining full marks on Q9.3. Most often students only referenced B1 and B2 without specifying the price or name fields.

Question 10

Question 10 asked for students to work through three subroutines testing understanding of parameters and return values. Students were asked to identify the output based on inputs. However, as in past series, this type of question is still causing students some issues. Fewer than a third of students could correctly trace through two subroutines and that number dropped to fewer than a quarter when the trace involved three subroutines, with students tending to stop at the end of the first part and returning a string including the function call.

Question 11

For this question students were required to write code to ask the user to input a username and password repeatedly until they matched one of the given pairs. It was pleasing to see that over 90% of students attempted the question with more than 65% gaining four or more of the available marks. The most common mistake made, as with previous years, was students not using iteration to keep getting user input. This limits the marks they can gain in the question. Another common error was not using Boolean conditions correctly in the statement. Students did not check each point against the search criteria, for example, using

```
if username == "Yusef5" or "Mary80" and password == "33kk" or
"af5r"
```

instead of

```
if username == "Yusef5" and password == "33kk" or username ==
"Mary80" and password == "af5r"
```

These are errors that recurring each year and across all three languages.

Question 12

Question 12 was a series of questions based around subroutines and a program to slide tiles around a puzzle. Parts 12.3 and 12.4 asked the students to explain the purpose of the iteration structures in the

code: only a third of students were able to answer this correctly although over 90% of the students attempted the question. As with other questions seen on this paper many of the answers given were vague and lacking detail.

With part 12.6, it was disappointing to see that nearly 20% of students did not attempt this question which assessed their ability to index three squares in a grid. Those that did answer did well with over 50% gaining at least three of the four available marks and over 60% gaining at least two. Commonly students did not select the correct three tiles or checked if they were merely greater than each other rather than consecutive.

A similar number of students attempted part 12.7 with over 50% gaining at least four of the six available marks. As with other questions expecting an indefinite loop, this was often missed out or done as a loop that does not exist in a language (such as a `do until` in Python). The other common mistake was not to use the subroutines that were given as part of the question or to use them incorrectly. A significant number of students used `solved = solved()` and then used solved in their conditions.

Question 13

Question 13 was a description of a linear search and it was pleasing to see some very detailed answers with nearly 70% of students gaining at least two of the three available marks. Some common misconceptions were that you need to have an ordered list or students explaining a binary search. Students must also remember the item being searched for may not be in the list/array being searched.

Question 14

For part 14.1, a number of responses merely referenced subroutines but then did not provide enough detail to be credited with the mark; as with definition questions students need to be clear in their response.

For part 14.2, as with previous coding questions, a large number of students did not attempt the question; however 40% of the students gained four of the available six marks. Some students did not create the subroutine correctly – this was an issue in all three languages but especially where a function is specifically required. This question tested the use of a definite loop, but a lot of students did not gain the mark due to incorrect conditions being used in the loop. A significant number of students also printed the value rather than returning as the question asks.

Question 15

This was the last coding question on the paper and attempted by a similar number as the others. Those that did attempt it gave well structured answers with over 40% of students gaining five or more of the eight available marks. The biggest issue seen with this question was the list/array index being allowed to go out of bounds. Students who scored highly checked that `pos` had not gone beyond the length of the list/array and reset it to `0` before checking for `"X"`, this prevented the out of bounds error. The other common issue was not checking to see if the input was either 1 or 2.

Final note

A small number of students did not attempt the coding questions. Teachers should encourage students to attempt every question as often marks can be gained for straightforward variable assignment, output

statements and simple Boolean conditions. There are also design marks available for the coding questions which means that some marks can be gained even if the students code is incorrect. A tip for students might be to tick off each item on any list of requirements when they have included it in their answer.

AQA has provided resources on their website which show the coding guidance that the examiners use as well as programming guides and pseudocode guides to support the teaching of this paper which covers the areas that will be tested on the exam.

Mark Ranges and Award of Grades

Grade boundaries and cumulative percentage grades are available on the Results Statistics page of the AQA Website.