AQA

AS-Level
Computer Science

Paper 1
Report on the Examination

7516/1
June 2016

Version: v1.0

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

General

Most students were well prepared for this exam and had made good use of the time available between the release of the Preliminary material and the day of the exam.

Section B (the questions about the skeleton code) were often poorly answered with only a limited understanding of the Skeleton Program shown.

Students will not receive marks for screen captures that have not been produced by running their code. Students should also make sure, when pasting in screen captures, that they are readable. This is perhaps a skill that could be practised before the exam by making use of an Electronic Answer Document and a past paper.

A copy of the Skeleton Program used by the school/college should be included alongside the scripts sent to the examiner whether or not the Skeleton Program was modified. A significant number of school/colleges did not to do this. A few centres attached a copy of the Skeleton Program to each student Electronic Answer Document and, sometimes, also the exam paper which is not required.

Question 1

The majority of students gained the two marks available from question 1.

Question 2

Most students were able to identify whether or not a string would be accepted by the finite state machine. Describing the language was more of a challenge. Most students could identify that the string would always end with an 'x'. Describing a language needs to be more than just specifying some strings that will work.

Question 3

In question 3 students were asked to identify different types of test data and this proved to be quite challenging for students. Some students calculated the square root of each value, whilst others tried to identify what kind of number would be returned (for example integer / natural). Of those students providing types of test data 'boundary test' was the one most commonly identified correctly.

Question 4

Students struggled to complete the trace table for question 4. Whilst the code did involve a nested loop, marks were available for completing just one iteration of the inner loop. Of those students who completed the trace table the majority were able to identify the purpose of the algorithm, which was to make a copy of the array but remove duplicate values.

Question 5

This question was completed well by students with the majority of students achieving a high mark. There were a lot of full mark answers seen across all of the programming languages. A common mistake was to use real/float division instead of integer division. A few students also dropped marks by not using the exact messages and identifiers given in the pseudo-code.

Most students completed the testing section well and provided clear screen shots of their code working. A few students did not get the mark because they tested using different values to those provided in the question.

It is important that students consider the readability of their screen shots when pasting them into the Electronic Answer Document.

It was pleasing to see that a large group of students could identify the reason why a WHILE repetition structure was used.

Question 6

The majority of students secured all of the marks for question 6. A few students copied across more than just the identifier for each part and therefore did not gain the mark as it was not clear which part of the copied code they believed the identifier to be.

Question 7

The majority of students secured a few marks for question 7. Parts 7.2 and 7.3 allowed the majority of students access to one mark. For 7.2 they could explain that a FOR loop was used as the number of iterations was known as it depended on the length of a ship. It was less common to see a student talk about each iteration of the FOR loop being used to check whether a square was empty or not.

The topic of exception handling is new in this specification and when answered well it was clear that a student understood how it could be used. Some answers were really explaining validation checking rather than exception handling. Those who could explain that exception handling is used to capture errors that would cause a program to crash secured a mark. It was then common to see students talk about this being caused by a non-integer value being entered and the idea of being asked to try again.

Question 8

The hierarchy chart was attempted well by the majority of students. A few students responded with identifiers for variables rather than the identifier for a subroutine.

The idea of breaking down a program into subroutines allowed students to pick up one mark for part 8.4 but most struggled to gain the second mark although some did identify the main control structures used.
The majority of students could identify the two different variables as being local variables and secure a mark.

Question 9

The majority of students made a good attempt at question 9. Implementing a loop and checking against the two boundaries was completed successfully by the majority of students.

Some students missed out the loop but did gain marks for checking against the boundaries and having the correct prompt for an invalid entry; another common error was to only check against the upper bound.

With syntactically correct code most students could then proceed and pick up the testing mark.

Question 10

The majority of students made a good attempt at question 10. Students do need, however, to make sure that if a task is broken down into steps on the question paper that they follow these steps. Some students presented a different solution for CheckSunk that just utilised scanning across the whole board rather than decreasing the size of a ship. Whilst these students were rewarded for certain parts of their solution they did not gain full marks as they did not answer the question in the way described on the question paper.

Where a FOR loops was implemented to iterate across the ships array we saw some very efficient solutions presented. A common mistake was placing the sunk message in the wrong place, resulting in multiple messages being displayed as the game progressed.

Question 11

It was disappointing that a large number of students did not include any attempt at answering this question. There was a mark available for simply creating a correctly-named subroutine with a suitable parameter and for identifying that a variable would be needed to hold whether a torpedo had been fired.

Of those who submitted an attempt it was pleasing to see a variety of ways at implementing the torpedo into code. A large number of students identified the need for a variable to hold whether a torpedo had been fired or not. Students either used a Boolean value or manipulated an integer value to clearly represent this idea. They usually then asked the player whether they wished to fire a torpedo using correct logic but a few students lost a mark by not using the prompt giving in the exam paper. It did prove difficult, however, to get the calls to MakePlayerTorpedoMove and MakePlayerMove to be called under the correct circumstances.

In the MakePlayerTorpedoMove subroutine it was common to see an attempt at an iteration structure to control the movement of the torpedo. To correctly code the exiting of this iteration structure proved to be hard but we saw some students write criteria to be tested against and other students making use of break statements. The most common iteration structure was a WHILE statement with the torpedo being move upwards inside the loop but a few students did use a FOR loop to control the torpedo movement up the board, breaking out of the structure if necessary. In terms of moving the torpedo there was a bit of confusion as to direction with some students moving the torpedo down the board or even moving the torpedo across columns but the majority of students answering the question secured a mark for correctly moving the torpedo.

Able students were able to correctly adapt the PlayGame routine and made a good attempt at getting the torpedo to move up the board exiting appropriately. These students were also able to secure both testing marks although it was possible to achieve one of the testing marks just by altering the PlayGame subroutine.

Mark Ranges and Award of Grades

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