



---

# A level Computer Science

Paper 1  
Report on the Examination

---

7517/1  
June 2017

---

Version: v1.0

---

---

Further copies of this Report are available from [aqa.org.uk](http://aqa.org.uk)

Copyright © 2017 AQA and its licensors. All rights reserved.

AQA retains the copyright on all its publications. However, registered schools/colleges for AQA are permitted to copy material from this booklet for their own internal use, with the following important exception: AQA cannot give permission to schools/colleges to photocopy any material that is acknowledged to a third party even for internal use within the centre.

**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 unseen programming question) was found challenging by a significant number of students indicating that students at some centres would benefit from more practice with this type of question.

Students will not receive marks for screen captures that have not been produced by running their code. There were many students who provided screen captures that were not produced by their code for Question 7 as it was quite straightforward to work out what the output should be. Their time would have been better spent trying to get their program code working correctly.

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 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**

This question was generally well-answered by students with most able to identify which of the numbers in Table 1 were valid according to the rules defined in the syntax diagram. The second part of the question was harder with many students correctly defining the base case but not the recursive case; a number of answers were not written using BNF even though an example of a rule was given in the question.

**Question 2**

Most students were able to obtain some marks on this question but very few obtained full marks. Many answers for 2.2 simply stated that state S11 meant that the postcode was valid which was accurate but not enough to get the mark as this did not differentiate S11 from the states S10 and S21.

Many students were unaware of the order or precedence used in regular expressions and therefore dropped marks on question 2.4 due to not including brackets where they were needed. A sequence has higher priority than OR in regular expressions. The regular expression `ab|cd` would match with the strings `ab` and `cd` not the strings `abd` and `acd`; to obtain this latter functionality the regular expression `a(b|c)d` could be used.

**Question 3**

This was the Section A question that students found hardest, with very few getting full marks. Not many students were able to identify the time complexity of either merge sort or (to a lesser extent) bubble sort and a significant number of students thought that the binary search and/or the post-order tree traversal would not be used to solve tractable problems.

When students could state the time complexity of the bubble sort algorithm they were rarely able to clearly explain why  $O(n^2)$  was the correct answer.

**Question 4**

Most students could explain what was meant by a recursive subroutine though some answers showed that the difference between iteration and recursion was not always understood. The trace was reasonably well done with the most common error being to include additional function calls or outputs in the table.

**Question 5**

This was the first A Level exam question on the topic of vectors and most students were well-prepared for this topic though there were clear differences between centres with students from some centres showing no familiarity with this area of the syllabus. A significant number of students found calculations that involved negative numbers more difficult.

The explanations of what a heuristic technique is often showed some understanding but were too vague to be creditworthy.

The difference(s) between static and dynamic data structures was generally well understood. A common answer that was not awarded marks was to simply state that static data structures cannot be changed, this does not make it clear that the size cannot change.

**Question 6**

Most students had a general understanding of what a stack was but answers were often about how stacks work rather than about the use of stacks in the context given in the question. A common misconception was that the top pointer could be moved down to point to items lower down in the stack without popping data from the top of the stack.

**Question 7**

A wide variety of approaches were used to successfully answer this question showing with students often coming up with creative and unexpected approaches to the task set. Some of the more common errors from students who had made a good but not completely accurate attempt at answering the question were to count all instances of a character in the string (rather than all consecutive instances) and to fail to stop the program from checking a position outside the bounds of the string entered by the user.

It was disappointing that more than 10% of students either did not attempt to answer the question or obtained fewer than two marks when it was possible to obtain this mark by simply displaying an appropriate message on the screen and storing the string entered by the user in an appropriately-named variable.

**Question 8**

Almost all students obtained some marks on question 8 though very few got full marks. In question 8.1 the most common error was to state that there was a protected method present in Figure 11. Most students got the mark for 8.2 with `warren` being the most frequently seen incorrect answer. For question 8.3 the concept of a private attribute was better understood than a protected attribute. Many students thought that a protected attribute was an attribute that could not be changed. Students who did well on the exam paper overall normally had no issues answering 8.4 but students with less understanding of the code in the Skeleton Program often gave answers that explained why knowing the number of rabbits in a warren was useful instead of answering the question set.

Question 8.5 was not well answered with many students writing about the functionality of the program as a whole rather than the `CompressRabbitList` method. Answers often described rabbits being killed by other factors even though this was not done by this method.

Most students were able to get some marks for writing the class definition in question 8.6. The most common errors were to have `HDRabbit` inheriting from `Animal` instead of `Rabbit`, including the gender attribute in the `HDRabbit` definition and not overriding the `Inspect` method.

### Question 9

This was, for most students, the easiest of the programming questions on the paper with about half obtaining full marks. Less confident programmers often had the wrong logic in their conditions (either getting `AND/OR` mixed-up or `</>`). Some students did not write code to get the validation condition to continually repeat until a valid value was entered. A significant minority of students did not add the validation routine to the `InputCoordinate` routine and instead tried to add it the constructor for the `Simulation` class.

Some students used recursion instead of iteration and full marks could be obtained from using this method if it was done correctly however many of these students did not return the value from the recursive call to the calling routine in a way that it could then be used by the rest of the program.

### Question 10

The majority of students were able to get at least half the marks on this question and were clearly familiar with how to create a method that overrides a method in a base class in the programming language they were using. A significant minority of students did not attempt this question and had clearly not prepared for answering questions using OOP within the Skeleton Program.

A number of students did not identify the correct variable to use and wrote code that tried to change the default probability instead of the protected attribute inherited from the `Animal` class storing the probability for that animal.

Some students did not call the overridden method in the base class even though the question specified this should be done. The equivalent functionality could be obtained by copying the code in the `CalculateNewAge` method in the `Animal` class into the new `CalculateNewAge` method in the `Rabbit` class but this is poor programming practice as the original code would now be in two places in the program rather than reusing the existing code.

### Question 11

One fifth of students did not provide any evidence of their attempt to answer this question. All students should be encouraged to include any program code they have written as it may be worth some marks even if it doesn't work correctly.

The most common mistake in reasonable attempts at the tasks in this question was to have the incorrect logic (for example, getting muddled between `AND/OR`) when writing the code to prevent a warren/fox being placed in a river.

### Question 12

Many students came up with creative answers to this question that showed a high-level of programming and problem-solving skill. However, a large number of students did not include any evidence of their attempt at writing the program code. Some students showed good exam technique by including a very limited answer which they knew was nowhere near correct but would

allow them to get some marks (most frequently for creating a new subroutine with the name specified in the question).

The most challenging part of the question was to make sure that the solution worked irrespective of the relative position of the fox and the warren with a number of solutions working if the fox was to the left of and above the warren but not if it was to the right of and below the warren.

### **Mark Ranges and Award of Grades**

Grade boundaries and cumulative percentage grades are available on the [Results Statistics](#) page of the AQA Website.