AS
Computer Science

Paper 1
Report on the Examination

7516/1
June 2017

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) was 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.

When the evidence requested is amended program source code for a subroutine, the whole of the edited subroutine should be added to the Electronic Answer Document. Responses only showing the new code but not where in the subroutine it was inserted may not secure all the marks available.

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 three marks from this question. Some students did not follow the guidelines in the question that two labels have to be assigned to some of the events. Many students did not use all the labels.

Question 2

Some students struggled to complete the trace table for this question. Very few students gained the two marks available for explaining why the algorithm was not fully effective. Many students were able to state that G was an invalid character and produced -1. A small number of students then deduced that this should be used as an error code and not to calculate with it as this would give a misleading result.

Question 3

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 integer division instead of real/float division. A small number of students were not awarded marks because they did not use 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. Some students were not awarded the mark because they tested using different values to those provided in the question.

As stated above 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 number of students could identify the reason why the WHILE loop was written using `Temp1` and `Temp2` instead of `Number1` and `Number2`.

Question 4

The majority of students secured all of the marks for this question. A small number of 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 5

The majority of students gained one mark, usually for stating that if the specification for the field size changes, only the values at the beginning of the source code need to be changed. Another valid reason frequently given was that using named constants made the source code more understandable. Some students secured both marks for giving both these explanations. Students should be aware of the difference between 'understandable' and 'readable'; the latter does not necessarily include that the meaning is clearer.

Most students gained the mark for realising that the first selection structure in the `SeedLands` subroutine tested that the coordinates are within the field boundaries.

The majority of students were able to describe the changes needed to the subroutine in order to simulate a minor drought. Many students suggested that the code for the major drought should be replicated, another selection construct added and the rainfall tested for a different value. Some students replaced the major drought with a minor drought rather than provide this as an alternative. Some wrongly suggested using a random number generator to determine which plants should be killed off.

Some students correctly stated that the arithmetic operation was integer division. Division on its own was not precise enough to gain a mark. Many students were able to give the correct values for Row and Column. Some students clearly did not check their answer when they stated a decimal value, which is not possible for a coordinate.

Question 6

The hierarchy chart was well answered by the majority of students.

Many students did not refer to the Skeleton Program when explaining how data is shared between the separate subroutines. The `Field` data structure, and other values, were passed as parameters and returned from subroutines. The term 'parameter passing' did not appear to be familiar to some students.

Question 7

The majority of students secured all of the marks for this question by explaining that the extra rows and columns would be ignored. Some students wrongly thought that an exception would be generated and therefore a new field created.

Many students missed out on some marks because they did not explain what happens in each season. It was not sufficient to state that in spring every seed changed into a plant and nothing changed in summer. An acceptable explanation was that in spring every location would have a plant in it as there was no frost and there was no change in summer because there was no drought. Most students gained the two marks for explaining that in autumn the seeds dropped but could not land as there were no free locations and therefore in winter, when all the plants died, the field would be left totally empty.

Question 8

The majority of students used iteration to repeatedly input data until a valid value was entered. Many students did not use exception handling to deal with non-integer data, even though an example of exception handling was already in the Skeleton Program in another subroutine. The majority of students correctly tested for the range, although some excluded zero. Although it was clearly stated in the question that the message '`Invalid input`' should be output, a significant minority of solutions did not do this.

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

Question 9

A significant number of students were not able to write the correct formula to calculate the percentage, even though a description was given in the question. Many students used integer division rather than float division in the calculation. However, this rounds down, rather than to the nearest integer as required by the question. The testing mark was only awarded for the evidence of a correctly rounded percentage and the field shown to which the calculation referred.

Question 10

The majority of students secured most of the marks for this question. A small number of students did not include the `Field` parameter when defining the `SaveToFile` subroutine header. Many students realised that they could use the `ReadFile` subroutine as a template for the `SaveToFile` subroutine and that the formatting information for the end of each row was also given in the `Display` subroutine of the Skeleton Program. To gain both marks for the testing, students had to show the field contents to be saved, all the required user interaction and the field contents after the program loaded the saved file. The field to be saved and the field loaded did not match in some students' screen captures.

Question 11

Most students attempted this question and it was very pleasing to see that some students had carefully thought about the design before writing their code. The majority of students repeatedly copied the contents of the selection statement and adjusted the `Row` and `Column` values to move the seed for each of the possible wind directions. Some students did not use the random number generator correctly and ended up with fewer or more possible wind directions than required. Some students did not take note of the explanation about prevailing winds and adjusted seed positions for a wind blowing to a direction rather than from a direction. A significant minority confused the adjustments and the wind directions did not match where the seeds landed.

A design mark was available for those students whose solution did not increase the number of calls to the SeedLands subroutine. A particularly good design is shown here using pseudo-code (the additional code is shown in bold):

```
SUBROUTINE SimulateAutumn(Field)
  WindNS ← random(-1, 1)
  WindEW ← random(-1, 1)
  Wind ← [["SouthEast", "South", "SouthWest"]
          ["East", "None", "West"]
          ["NorthEast", "North", "NorthWest"]]
  FOR Row ← 0 TO FIELDLENGTH – 1
    FOR Column ← 0 TO FIELDWIDTH – 1
      IF Field[Row, Column] = PLANT THEN
        Field ← SeedLands(Field, Row – 1 + WindNS, Column – 1 + WindEW)
        Field ← SeedLands(Field, Row – 1 + WindNS, Column + WindEW)
        Field ← SeedLands(Field, Row – 1 + WindNS, Column + 1 + WindEW)
        Field ← SeedLands(Field, Row + WindNS, Column – 1 + WindEW)
        Field ← SeedLands(Field, Row + WindNS, Column + 1 + WindEW)
        Field ← SeedLands(Field, Row + 1 + WindNS, Column – 1 + WindEW)
        Field ← SeedLands(Field, Row + 1 + WindNS, Column + WindEW)
        Field ← SeedLands(Field, Row + 1 + WindNS, Column + 1 + WindEW)
      ENDIF
    ENDFOR
  ENDFOR
  WindDirection ← Wind[WindNS + 1, WindEW + 1]
  IF WindDirection = "None" THEN
    OUTPUT("There was no prevailing wind")
  ELSE
    OUTPUT("There was a prevailing wind from the ", WindDirection)
  ENDIF
  RETURN Field
ENDSUBROUTINE
```

Mark Ranges and Award of Grades

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