AQA

Specimen material

A-level
COMPUTER SCIENCE

Paper 1
Additional Questions

IMPORTANT NOTES

These questions focus primarily on topics that were not covered by the AQA AS and A-level Computing specifications, introduced in 2009. It is hoped that teachers will find questions on these new topics to be particularly useful.

Many example questions on topics that are common to the new and old specifications can be found on past papers for COMP1, 2 and 3 on our website. Past papers that are more than three years old can be accessed via e-AQA.

This document contains additional questions; it is not intended to be treated as a complete paper. The questions do not provide balance coverage of the specification or the assessment objectives in the same way that a fully live paper would do.

7517A/B/C/D/E

0 1 **Table 1** contains a list of problems.

Table 1

Letter	Problem
A	Finding the shortest route that visits all nodes in a graph exactly once
B	Finding the shortest path between two nodes in a graph
C	Finding an item in a large unordered list
D	Finding a route between two nodes in a graph
E	Finding out if any program will eventually stop if given a particular input

0 1 . 1 State the letter (**A-E**) that corresponds to the problem that Dijksta's Algorithm is designed to solve.

Write the letter corresponding to the correct answer in the box provided in your Electronic Answer Document.

[1 mark]

0 1 . 2 State the letter (**A-E**) that corresponds to an intractable problem.

Write the letter corresponding to the correct answer in the box provided in your Electronic Answer Document.

[1 mark]

0 1 . 3 Explain the significance of problem **E** for computation.

[1 mark]

0 2 **Figure 1** contains pseudo-code for a recursive merge sort algorithm. **Figure 2** contains pseudo-code for an algorithm called `Merge` that is called by the merge sort algorithm in **Figure 1**.

Figure 1

```
FUNCTION MergeSort(L, S, E)
  IF S < E THEN
    M ← (S + E) DIV 2
    L1 ← MergeSort(L, S, M)
    L2 ← MergeSort(L, M + 1, E)
    RETURN Merge(L1, L2)
  ELSE
    RETURN Append([], L[S])
  ENDIF
ENDFUNCTION
```

Figure 2

```
FUNCTION Merge(L1, L2)
  L3 ← []
  WHILE Len(L1) > 0 AND LEN(L2) > 0
    IF L1[1] < L2[1] THEN
      L3 ← Append(L2[1], L3)
      L2 ← RemoveFirstItem(L2)
    ELSE
      L3 ← Append(L1[1], L3)
      L1 ← RemoveFirstItem(L1)
    ENDIF
  ENDWHILE
  WHILE Len(L1) > 0
    L3 ← Append(L1[1], L3)
    L1 ← RemoveFirstItem(L1)
  ENDWHILE
  WHILE Len(L2) > 0
    L3 ← Append(L2[1], L3)
    L2 ← RemoveFirstItem(L2)
  ENDWHILE
  RETURN L3
ENDFUNCTION
```

The `RemoveFirstItem` function takes a list and returns a list that contains all the items in the original list except the first one. For example, if `Names` is the list `["Gemma", "Richard", "Georgina", "Margaret"]` then the function call `RemoveFirstItem(Names)` will return the list `["Richard", "Georgina", "Margaret"]`.

The `Len` function takes a list and returns the number of items that are in the list. For example, if `Names` is the list `["Gemma", "Richard", "Georgina", "Margaret"]` then the function call `Len(Names)` will return the value of 4.

The `Append` function takes an item and a list and returns a list that has all the items from the original list followed by the item. For example, if `Names` is the list `["Gemma", "Richard", "Georgina", "Margaret"]` then the function call `Append("Matt", Names)` will return the list `["Gemma", "Richard", "Georgina", "Margaret", "Matt"]`.

The first item in the list has an index of 1.

0 2 . 1 What is meant by a recursive subroutine?

[1 mark]

0 2 . 2 What is the base case for the subroutine `MergeSort`?

[1 mark]

0 2 . 3 Complete **Table 1** to show the result of tracing the `MergeSort` algorithm shown in **Figure 1** with the function call `MergeSort(ListToSort, 1, 5)`. `ListToSort` is the list `[6, 3, 4, 8, 5]`. The first six rows and the **Call number** column have been completed for you.

Copy your answer in **Table 1** into the Electronic Answer Document.

Table 1

Call number	S	E	M	List returned
1	1	5	3	
2	1	3	2	
3	1	2	1	
4	1	1		[6]
3	1	2	1	
5	2	2		[3]
3				
2				
6				
2				
1				
7				
8				
7				
9				
7				
1				

[6 marks]

0 2 . 4 What is the time complexity for the `MergeSort` algorithm shown in **Figure 1**?

[1 mark]

A stack frame is used with subroutine calls.

0 2 . 5 State **two** items that will be stored in a stack frame for a subroutine call.

[2 marks]

When the subroutine call `MergeSort(ListToSort, 1, 5)` is made four is the largest number of stack frames, generated by this subroutine call, that will be on the stack at any one time.

0 2 . 6 Explain why there will be **three** occasions when there will be four stack frames on the stack when the subroutine call `MergeSort(ListToSort, 1, 5)` is made.

[2 marks]

Turn over for the next question

0 3 Three different types of relationship that can exist between objects in object-orientated programming are **aggregation**, **composition** and **inheritance**.

0 3 . 1 State **one** reason why many programmers follow the design principle "favour composition over inheritance".

[1 mark]

An object-oriented program is being written to store details about clients at an estate agency. Clients can be either sellers or prospective buyers.

A class `Client` has been created and two subclasses, `Seller` and `Buyer` are to be developed. A `Location` class has been created to store details about an address (eg postcode and street name)

The `Client` class has data fields `Name`, `Address` and `DOB`.

Part of the class definition for `Client` class is:

```
Client = Class {
  Private:
    Name: String
    Address: Location
    DOB: Date
  Public:
    Function GetName
    Function GetDOB
    Function GetAddress
    Procedure SetDetails
}
```

A `Buyer` has the following additional data fields:

- `NoOfBedroomsRequired`: stores the minimum number of bedrooms that the buyer requires in the house they purchase.
- `OffStreetParking`: stores a value indicating if the buyer requires off-street parking or not.
- `AreaDesired`: the name of the town/village/estate that the buyer is looking to purchase a house in.

0 3 . 2 Write the class definition for `Buyer`.

[4 marks]

0 3 . 3 Describe the relationships **aggregation** and **composition**, making it clear what the difference between the two is.

[2 marks]

0 3 . 4 Explain how the `Client` class uses aggregation and why it was considered more appropriate to use this type of relationship than composition.

[2 marks]

0 4 **Figure 3** contains a list of different sets.

Figure 3

S1 = {a, b}
S2 = {a, b, c}
S3 = {0, 1, 2}
S4 = {a, ab}
S5 = {a, b, c}
S6 = {c}

0 4 . 1 What is a set?

[2 marks]

0 4 . 2 State the name of a set listed in **Figure 3** that has the same cardinality as set S2.

[1 mark]

0 4 . 3 State the name of a set listed in **Figure 3** that is a subset of set S2 but not a proper subset of S2.

[1 mark]

0 4 . 4 Describe how set S6 could be created using the difference set operation together with two of the other sets listed in **Figure 3**.

[1 mark]

0 4 . 5 What is the Cartesian product of sets S1 and S6?

[1 mark]

A regular expression is a way of describing a set.

0 4 . 6 Write down **two** different regular expressions that describe the set S4.

[2 marks]

0 5 A designer jewellery company have a scheme where people can pay a fee to borrow one or more items of jewellery for a week. This scheme is often used by actors and musicians attending awards ceremonies and parties.

A computer program running on the jewellery company's server uses a dictionary to store details about who is currently borrowing which item of jewellery. A dictionary is a data structure that contains a collection of key-value pairs where the value is accessed via the associated key. The key used in the dictionary is the catalogue number of the item of jewellery and the key is the name of the client.

The current state of the dictionary is shown in **Figure 4**.

Figure 4

{1045 : 'Clark Gable', 1050 : 'Ingrid Bergman', 1052 : 'Katherine Hepburn', 2012 : 'Lauren Bacall'}

0 5 . 1 What value will be returned by a lookup operation using the key 1052?

[1 mark]

A hash table has been used to implement the dictionary. The hashing algorithm that has been used is `Catalogue Number MOD 100`.

0 5 . 2 What value is returned by the hashing function when it is applied to the catalogue number 1052?

[1 mark]

0 5 . 3 Explain how a value is stored in a hash table.

[4 marks]

When a large number of the possible positions in a hash table are filled it becomes inefficient; rehashing is often used to sort out this problem.

0 5 . 4 Describe the steps involved in rehashing.

[3 marks]

The jewellery company would also like to be able to complete lookup operations based on the client name to find out which items of jewellery are being borrowed by a particular client.

0 5 . 5 Explain how the jewellery company could set up a dictionary with the name of the client as the key and the catalogue numbers as the value.

[1 mark]