AQA Computer Science A-Level

4.1.1 Programming

Past Paper Questions

Additional Spec Qs AS Paper 1

0 3 A pseudo code representation of an algorithm is given in **Figure 2**.

Figure 2

```
OUTPUT "Enter value 1:"
INPUT Value1
OUTPUT "Enter value 2:"
INPUT Value2
IF Value1 < Value2 THEN
   OUTPUT "Value 2 is larger"
ELSE
   OUTPUT "Value 1 is larger"
ENDIF
```

0 3 . 1 It is found when testing an implementation of the pseudo code in **Figure 2** that with the inputs 007 and 06 it wrongly outputs that the second value is larger.

Explain why this might be the case.

[2 marks]

0 3 . 2 It is also found whilst testing that an implementation fails with the inputs 34 and 34, wrongly outputting that the first value is larger.

Write the corrected code necessary so that the implementation would work correctly with the inputs 34 and 34.

[2 marks]

) 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]

June 2012 Comp 3

10 (d) (i) What is a *recursive routine*?

.....

.....

(1 mark)

10 (d) (ii) To enable the use of recursion a programming language must provide a stack.

Explain what this stack will be used for and why a stack is appropriate.

.....

.....

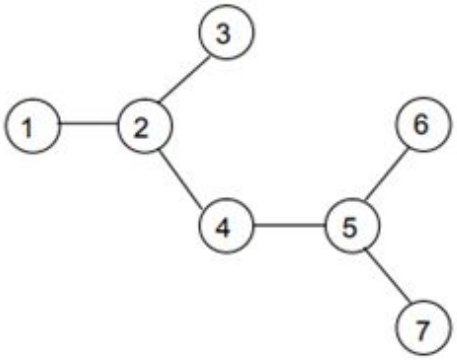
.....

.....

(2 marks)

Figure 8 from page 20 is repeated here so that you can answer Question 10(e) without having to turn back in the question booklet.

Figure 8 (repeated)



10 (e) A recursive routine can be used to perform a depth-first search of the graph that represents the maze to test if there is a route from the entrance (vertex 1) to the exit (vertex 7).

The recursive routine in **Figure 9** is to be used to explore the graph in **Figure 8**. It has two parameters, `V` (the current vertex) and `EndV` (the exit vertex).

Figure 9

```
Procedure DFS(V, EndV)
  Discovered[V] ← True
  If V = EndV Then Found ← True
  For each vertex U which is connected to V Do
    If Discovered[U] = False Then DFS(U, EndV)
  EndFor
  CompletelyExplored[V] ← True
EndProcedure
```

Complete the trace table below to show how the `Discovered` and `CompletelyExplored` flag arrays and the variable `Found` are updated by the algorithm when it is called using `DFS(1,7)`.

The details of each call and the values of the variables `V`, `U` and `EndV` have already been entered into the table for you. The letter `F` has been used as an abbreviation for `False`. You should use `T` as an abbreviation for `True`.

				Discovered							CompletelyExplored							
Call	V	U	EndV	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	Found
	-	-		F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
DFS(1,7)	1	2	7															
DFS(2,7)	2	1	7															
		3	7															
DFS(3,7)	3	2	7															
DFS(2,7)	2	4	7															
DFS(4,7)	4	2	7															
		5	7															
DFS(5,7)	5	4	7															
		6	7															
DFS(6,7)	6	5	7															
DFS(5,7)	5	7	7															
DFS(7,7)	7	5	7															
DFS(5,7)	5	-	7															
DFS(4,7)	4	-	7															
DFS(2,7)	2	-	7															
DFS(1,7)	1	-	7															

(5 marks)

June 2016 AS Paper 1

0 5 The algorithm, represented using pseudo-code in **Figure 4**, describes a method to calculate the additive or multiplicative persistence of a two-digit integer. The examples below illustrate how additive and multiplicative persistence are calculated.

Example: calculating the additive persistence of 87

8 + 7 = 15
1 + 5 = 6

After 2 steps the method results in a one digit answer so the additive persistence of 87 is 2.

Example: calculating the multiplicative persistence of 39

3 * 9 = 27
2 * 7 = 14
1 * 4 = 4

After 3 steps the method results in a one digit answer so the multiplicative persistence of 39 is 3.

Figure 4

```
OUTPUT "Enter integer (0-99): "
INPUT Value
OUTPUT "Calculate additive or multiplicative persistence (a or m)? "
INPUT Operation
Count ← 0
WHILE Value > 9
  IF Operation = "a" THEN
    Value ← (Value DIV 10) + (Value MOD 10)
  ELSE
    Value ← (Value DIV 10) * (Value MOD 10)
  ENDIF
  Count ← Count + 1
ENDWHILE
OUTPUT "The persistence is: "
OUTPUT Count
```

The `MOD` operator calculates the remainder resulting from an integer division, for example, `10 MOD 3 = 1`.

The `DIV` operator calculates integer division, for example `10 DIV 3 = 3`.

What you need to do

Task 1
Write a program for the algorithm in **Figure 4**.

Task 2
Test the program by showing the result of entering `47`, followed by `m` when prompted by the program.

Task 3
Test the program by showing the result of entering `77`, followed by `a` when prompted by the program.

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

0 5 . 1 Your PROGRAM SOURCE CODE.

[8 marks]

0 5 . 2 SCREEN CAPTURE(S) showing the tests described.

[1 mark]

The part of the program where the calculations are performed uses a `WHILE` repetition structure.

0 5 . 3 Explain why a `WHILE` repetition structure was chosen instead of a `FOR` repetition structure.

[1 mark]

June 2017 AS Paper 1

0 3 The algorithm represented using pseudo-code in **Figure 5** describes a method to find the greatest common factor (GCF) of two whole numbers (integers) entered by the user.

Figure 5

```
OUTPUT "Enter a whole number: "
INPUT Number1
OUTPUT "Enter another whole number: "
INPUT Number2
Temp1 ← Number1
Temp2 ← Number2
WHILE Temp1 ≠ Temp2
  IF Temp1 > Temp2 THEN
    Temp1 ← Temp1 - Temp2
  ELSE
    Temp2 ← Temp2 - Temp1
  ENDIF
ENDWHILE
Result ← Temp1
OUTPUT Result, " is GCF of ", Number1, " and ", Number2
```

What you need to do:

Task 1
Write a program to implement the algorithm in **Figure 5**.

Task 2
Test the program by showing the result of entering `12` and then `39`.

Evidence that you need to provide
Include the following in your Electronic Answer Document.

0 3 . 1 Your PROGRAM SOURCE CODE for **Task 1**.

[6 marks]

0 3 . 2 SCREEN CAPTURE(S) showing the test described in **Task 2**.

[1 mark]

The algorithm copies the values of `Number1` and `Number2` into `Temp1` and `Temp2` respectively.

0 3 . 3 Explain why the `WHILE` loop was written using `Temp1` and `Temp2` instead of `Number1` and `Number2`.

[1 mark]

June 2017 Paper 1

0 7

One method that can be used to compress text data is run length encoding (RLE). When RLE is used the compressed data can be represented as a set of character/frequency pairs. When the same character appears in consecutive locations in the original text it is replaced in the compressed text by a single instance of the character followed by a number indicating the number of consecutive instances of that character. Single instances of a character are represented by the character followed by the number 1.

Figure 9 and **Figure 10** show examples of how text would be compressed using this method.

Figure 9

Original text: `AAARRRRGGGHH`
Compressed text: `A 3 R 4 G 3 H 2`

Figure 10

Original text: `CUTLASSES`
Compressed text: `C 1 U 1 T 1 L 1 A 1 S 2 E 1 S 1`

What you need to do

Task 1
Write a program that will perform the compression process described above. The program should display a suitable prompt asking the user to input the text to compress and then output the compressed text.

Task 2
Test the program works by entering the text `AAARRRRGGGHH`.

Task 3
Test the program works by entering the text `A`.

Evidence that you need to provide
Include the following in your Electronic Answer Document.

0 7 . 1 Your PROGRAM SOURCE CODE.

[12 marks]

0 7 . 2 SCREEN CAPTURE(S) for the test showing the output of the program when `AAARRRRGGGHH` is entered.

[1 mark]

0 7 . 3 SCREEN CAPTURE(S) for the test showing the output of the program when `A` is entered.

[1 mark]

June 2011 Comp 1

Question 7

Create a folder/directory **Question7** for your new program.

The variable table, **Table 4**, and the Structured English algorithm, **Figure 4**, describe a linear search algorithm that could be used with a simplified version of the Dice Cricket game to find out if a particular player's name appears in the high score table.

In this simplified version only the names of the players getting a top score are stored. Their scores are **not** stored.

Table 4

Identifier	Data Type	Purpose
`Names`	`Array[1..4] of String`	Stores the names of the players who have one of the top scores
`PlayerName`	`String`	Stores the name of the player being looked for
`Max`	`Integer`	Stores the size of the array
`Current`	`Integer`	Indicates which element of the array `Names` is currently being examined
`Found`	`Boolean`	Stores `True` if the player's name has been found in the array, `False` otherwise

Figure 4

```
Names[1] ← 'Ben'
Names[2] ← 'Thor'
Names[3] ← 'Zoe'
Names[4] ← 'Kate'
Max ← 4
Current ← 1
Found ← False
OUTPUT 'What player are you looking for?'
INPUT PlayerName
WHILE (Found = False) AND (Current <= Max)
  IF Names[Current] = PlayerName
    THEN Found ← True
    ELSE Current ← Current + 1
  ENDIF
ENDWHILE
IF Found = True
  THEN OUTPUT 'Yes, they have a top score'
  ELSE OUTPUT 'No, they do not have a top score'
ENDIF
```

What you need to do

Write a program for the above algorithm.

Test the program by searching for a player named 'Thor'.

Test the program by searching for a player named 'Imran'.

Save the program in your new **Question7** folder/directory.

Evidence that you need to provide
Include the following in your Electronic Answer Document.

2 0	Your PROGRAM SOURCE CODE.	*(11 marks)*
2 1	SCREEN CAPTURE(S) for the test searching for 'Thor'.	*(2 marks)*
2 2	SCREEN CAPTURE(S) for the test searching for 'Imran'.	*(2 marks)*

June 2012 Comp 1

Create a folder/directory **Question6** for your new program.

The algorithm, represented as a flowchart in **Figure 4**, and the variable table, **Table 3**, describe the converting of a 4-bit binary value into denary.

Figure 4

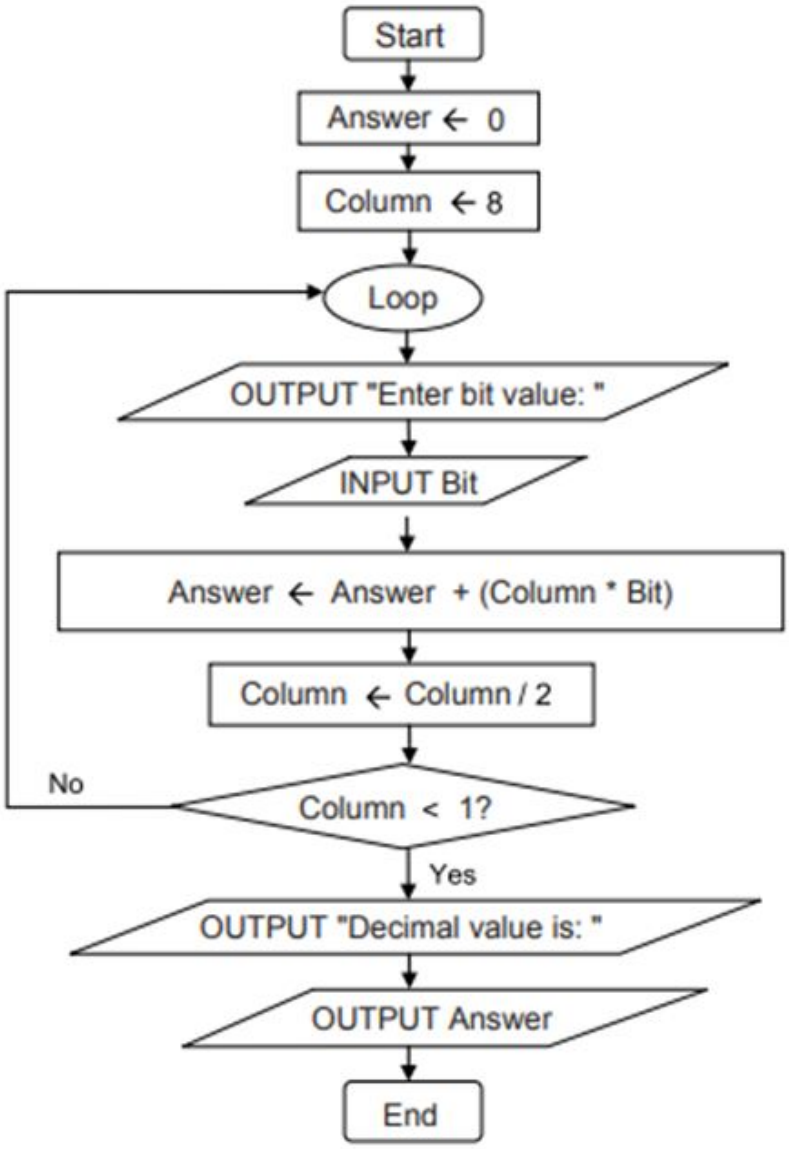


Table 3

Identifier	Data type	Purpose
`Column`	`Integer`	Stores the place value (column heading)
`Answer`	`Integer`	Stores the denary value equivalent to the bit pattern entered by the user
`Bit`	`Integer`	Stores a `0` or `1` entered by the user

What you need to do

Write a program for the above algorithm.

Test the program by showing the result of entering the values `1`, `1`, `0`, `1` (in that order).

Save the program in your new **Question6** folder/directory.

Evidence that you need to provide
Include the following in your Electronic Answer Document.

1 6 Your PROGRAM SOURCE CODE. *(11 marks)*

1 7 SCREEN CAPTURE(S) for the test described above. *(3 marks)*

1 8 What is the largest denary number that could be output by the algorithm represented by the flowchart in **Figure 4**? *(1 mark)*

June 2013 Comp 1

Section B

You are advised to spend no more than **25 minutes** on this section.

Type your answers to **Section B** in your Electronic Answer Document.
You **must save** this document at regular intervals.

The question in this section asks you to write program code **starting from a new program/project/file**.

- Save your program/project/file in its own folder/directory.
- You are advised to save your program at regular intervals.

Question 4

Create a folder/directory **Question4** for your new program.

The algorithm, represented using pseudo-code in **Figure 4**, and the variable table, **Table 3**, describe a simple two player game. Player One chooses a whole number between 1 and 10 (inclusive) and then Player Two tries to guess the number chosen by Player One. Player Two gets up to five attempts to guess the number. Player Two wins the game if they correctly guess the number, otherwise Player One wins the game.

Note that in **Figure 4**, the symbol <> means "is not equal to".

Figure 4

```
OUTPUT "Player One enter your chosen number: "
INPUT NumberToGuess
WHILE NumberToGuess < 1 OR NumberToGuess > 10 DO
  OUTPUT "Not a valid choice, please enter another number: "
  INPUT NumberToGuess
ENDWHILE
Guess ← 0
NumberOfGuesses ← 0
WHILE Guess <> NumberToGuess AND NumberOfGuesses < 5 DO
  OUTPUT "Player Two have a guess: "
  INPUT Guess
  NumberOfGuesses ← NumberOfGuesses + 1
ENDWHILE
IF Guess = NumberToGuess
  THEN OUTPUT "Player Two wins"
  ELSE OUTPUT "Player One wins"
```

Table 3

Identifier	Data type	Purpose
`NumberToGuess`	`Integer`	Stores the number entered by Player One
`NumberOfGuesses`	`Integer`	Stores the number of guesses that Player Two has made so far
`Guess`	`Integer`	Stores the most recent guess made by Player Two

What you need to do

Write a program for the above algorithm.

Test the program by conducting the tests **Test 1** and **Test 2**.

Save the program in your new **Question4** folder/directory.

Test 1
Test that your program works correctly by conducting the following test:

- Player One enters the number `0`
- Player One enters the number `11`
- Player One enters the number `5`
- Player Two enters a guess of `5`

Test 2
Test that your program works correctly by conducting the following test:

- Player One enters the number `6`
- Player Two enters guesses of `1, 3, 5, 7, 10`

Evidence that you need to provide
Include the following in your Electronic Answer Document.

1 9	Your PROGRAM SOURCE CODE.	*(13 marks)*
2 0	SCREEN CAPTURE(S) showing the result of **Test 1**.	*(4 marks)*
2 1	SCREEN CAPTURE(S) showing the result of **Test 2**.	*(3 marks)*

Part of the algorithm from **Figure 4** is shown in **Figure 5**.
Note that in **Figure 5**, the symbol <> means "is not equal to".

Figure 5

```
WHILE Guess <> NumberToGuess AND NumberOfGuesses < 5 DO
  OUTPUT "Player Two have a guess: "
  INPUT Guess
  NumberOfGuesses ← NumberOfGuesses + 1
ENDWHILE
```

2 2 Explain why a `WHILE` repetition structure was chosen instead of a `FOR` repetition structure for the part of the algorithm shown in **Figure 5**. *(1 mark)*

Specimen AS Paper 1

0 1 **Figure 1** contains the pseudo-code for a program to output a sequence according to the 'Fizz Buzz' counting game.

Figure 1

```
OUTPUT "How far to count?"
INPUT HowFar
WHILE HowFar < 1
  OUTPUT "Not a valid number, please try again."
  INPUT HowFar
ENDWHILE
FOR MyLoop ← 1 TO HowFar
  IF MyLoop MOD 3 = 0 AND MyLoop MOD 5 = 0
  THEN
      OUTPUT "FizzBuzz"
    ELSE
      IF MyLoop MOD 3 = 0
        THEN
          OUTPUT "Fizz"
        ELSE
          IF MyLoop MOD 5 = 0
            THEN
              OUTPUT "Buzz"
            ELSE
              OUTPUT MyLoop
          ENDIF
      ENDIF
  ENDIF
ENDFOR
```

What you need to do:

Write a program that implements the pseudo-code as shown in **Figure 1.**

Test the program by showing the result of entering a value of `18` when prompted by the program.

Test the program by showing the result of entering a value of `-1` when prompted by the program.

Evidence that you need to provide
Include the following in your Electronic Answer Document.

0 1 . 1 Your PROGRAM SOURCE CODE for the pseudo-code in **Figure 1**.

[8 marks]

0 1 . 2 SCREEN CAPTURE(S) for the tests conducted when a value of `18` is entered by the user and when a value of `-1` is entered by the user.

[1 mark]

The main part of the program uses a `FOR` repetition structure.

0 1 . 3 Explain why a `FOR` repetition structure was chosen instead of a `WHILE` repetition structure.

[1 mark]

Even though a check has been performed to make sure that the variable `HowFar` is greater than `1` there could be inputs that might cause the program to terminate unexpectedly (crash).

0 1 . 4 Provide an example of an input that might cause the program to terminate and describe a method that could be used to prevent this.

[3 marks]

0 6 Create a folder/directory called **Question6** for your new program.

One method for converting a decimal number into binary is to repeatedly divide by 2 using integer division. After each division is completed, the remainder is output and the integer result of the division is used as the input to the next iteration of the division process. The process repeats until the result of the division is 0.

Outputting the remainders in the sequence that they are calculated produces the binary digits of the equivalent binary number, but in reverse order.

For example, the decimal number 210 could be converted into binary as shown in **Figure 7**.

Figure 7

```
210 ÷ 2 =  105   remainder 0
105 ÷ 2 =  52    remainder 1
 52 ÷ 2 =  26    remainder 0
 26 ÷ 2 =  13    remainder 0
 13 ÷ 2 =  6     remainder 1
  6 ÷ 2 =  3     remainder 0
  3 ÷ 2 =  1     remainder 1
  1 ÷ 2 =  0     remainder 1
```

The sequence 0, 1, 0, 0, 1, 0, 1, 1 which would be output by this process is the reverse of the binary equivalent of 210 which is 11010010.

What you need to do

Task 1
Write a program that will perform the conversion process described above. The program should display a suitable prompt asking the user to input a decimal number to convert and then output the bits of the binary equivalent of the decimal number in reverse order.

Task 2
Improve the program so that the bits are output in the correct order, e.g. for 210 the output would be 11010010.

Task 3
Test the program works by entering the value 210.

Save the program in your new **Question6** folder/directory.

Evidence that you need to provide
Include the following in your Electronic answer document.

0 6 . 1 Your PROGRAM SOURCE CODE after you have completed both **Task 1** and **Task 2**.

If you complete **Task 1** but do not attempt **Task 2** then a maximum of 9 marks will be awarded.

[12 marks]

0 6 . 2 SCREEN CAPTURE(S) for the test showing the output of the program when 210 is entered.

The marks for this test will be awarded whether the binary digits are output in reverse order or in the correct order.

[2 marks]
