<u>AQA Computer Science A-Level</u>

4.4.1 Abstraction and automation

Past Paper Questions

Additional Specimen AS Paper 1

A pseudo code representation of an algorithm is given in **Figure 1**.

Figure 1

```
A = 100
B = 60
WHILE B <> 0
   TEMP = B
   B = A MOD B
   A = TEMP
ENDWHILE
OUTPUT "The value is:", A
```

The `MOD` operator calculates the remainder resulting from an integer division.
For example, `12 MOD 5 = 2`.

0 2 . 1 Dry run the above segment of code by completing **Table 1**.

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

Table 1

A	B	TEMP	OUTPUT
100	60	60	

0 2 . 2 What does the above segment of code in **Figure 1** perform?

[1 mark]

0 1 In each question part below two statements are given followed by two conclusions numbered 1 and 2.

You must take the two given statements to be true.

Read the statements and then decide which combination of the given conclusions logically follows from the two given statements.

0 1 . 1 Statements: All computing students drink coffee.
No coffee drinkers fly kites.

Conclusion 1: A computing student flies a kite.
Conclusion 2: All kite fliers drink tea.

Give answer:

- A If only Conclusion 1 follows
- B If only Conclusion 2 follows
- C If either Conclusion 1 or Conclusion 2 follows
- D If neither Conclusion 1 nor Conclusion 2 follows
- E If both Conclusion 1 and Conclusion 2 follow.

[1 mark]

0 1 . 2 Statements: If it rains, the streets will be wet.
If the streets are wet, accidents will happen.

Conclusion 1: Accidents will happen when it rains.
Conclusion 2: The streets will never be dry.

Give answer:

- A If only Conclusion 1 follows
- B If only Conclusion 2 follows
- C If either Conclusion 1 or Conclusion 2 follows
- D If neither Conclusion 1 nor Conclusion 2 follows
- E If both Conclusion 1 and Conclusion 2 follow.

[1 mark]

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]

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 . 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]

June 2011 Comp 3

5 (c) The pseudo-code algorithm in **Figure 3** can be used to calculate the result of evaluating a Reverse Polish Notation expression that is stored in a string. The algorithm is designed to work only with the single digit denary numbers 0 to 9. It uses procedures and functions listed in **Table 1**, two of which operate on a stack data structure.

Figure 3

```
StringPos ← 0
Repeat
  StringPos ← StringPos + 1
  Token ← GetCharFromString(InputString, StringPos)
  If Token = '+' Or Token = '-' Or Token = '/' Or Token = '*'
    Then
      Op2 ← Pop()
      Op1 ← Pop()
      Case Token Of
        '+': Result ← Op1 + Op2
        '-': Result ← Op1 - Op2
        '/': Result ← Op1 / Op2
        '*': Result ← Op1 * Op2
      EndCase
      Push(Result)
    Else
      IntegerVal ← ConvertToInteger(Token)
      Push(IntegerVal)
  EndIf
Until StringPos = Length(InputString)
Output Result
```

Table 1

Procedure/Function	Purpose	Example(s)
GetCharFromString (InputString: String, StringPos: Integer): Char	Returns the character at position StringPos within the string InputString. Note that the leftmost letter is position 1, not position 0.	GetCharFromString ("Computing", 1) would return the character 'C'. GetCharFromString ("Computing", 3) would return the character 'm'.
ConvertToInteger (ACharacter: Char): Integer	Returns the integer equivalent of the character in ACharacter.	ConvertToInteger('4') would return the integer value 4.
Length(AString: String): Integer	Returns a count of the number of characters in the string AString.	Length("AQA") would return the integer value 3.
Push(ANumber: Integer)	Puts the number in ANumber onto the stack.	Push(6) would put the number 6 on top of the stack.
Pop(): Integer	Removes the number from the top of the stack and returns it.	X ← Pop() would remove the value from the top of the stack and put it in X.

5 (c) Complete the table below to trace the execution of the algorithm when `InputString` is the string: `64+32+*`

In the `Stack` column, show the contents of the stack once for each iteration of the `Repeat..Until` loop, as it would be at the end of the iteration.

The first row and the leftmost column of the table have been completed for you.

StringPos	Token	IntegerVal	Op1	Op2	Result	Stack
0	-	-	-	-	-	
1						
2						
3						
4						
5						
6						
7						

(5 marks)

Final output of algorithm: ....

(1 mark)

11 A particular Turing machine has states S_1, S_2, S_3 and S_4. S_1 is the start state and S_4 is the stop state. The machine uses one tape which is infinitely long in one direction to store data. The machine's alphabet is 1, □. The symbol □ is used to indicate a blank cell on the tape.

The transition rules for this Turing machine can be expressed as a transition function δ. Rules are written in the form:

δ(Current State, Input Symbol) = (Next State, Output Symbol, Movement)

So, for example, the rule:

$$\delta\,(S_1, 1) = (S_1, 1, \rightarrow)$$

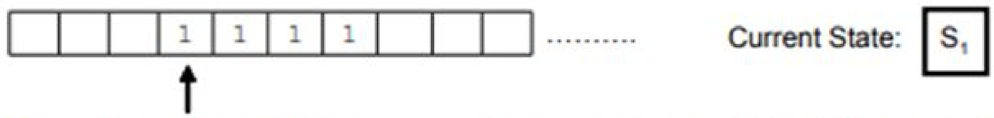
means:

IF the machine is currently in state S_1 AND the input symbol read from the tape is 1
THEN the machine should remain in state S_1, write a 1 to the tape and move the read/write head one cell to the right

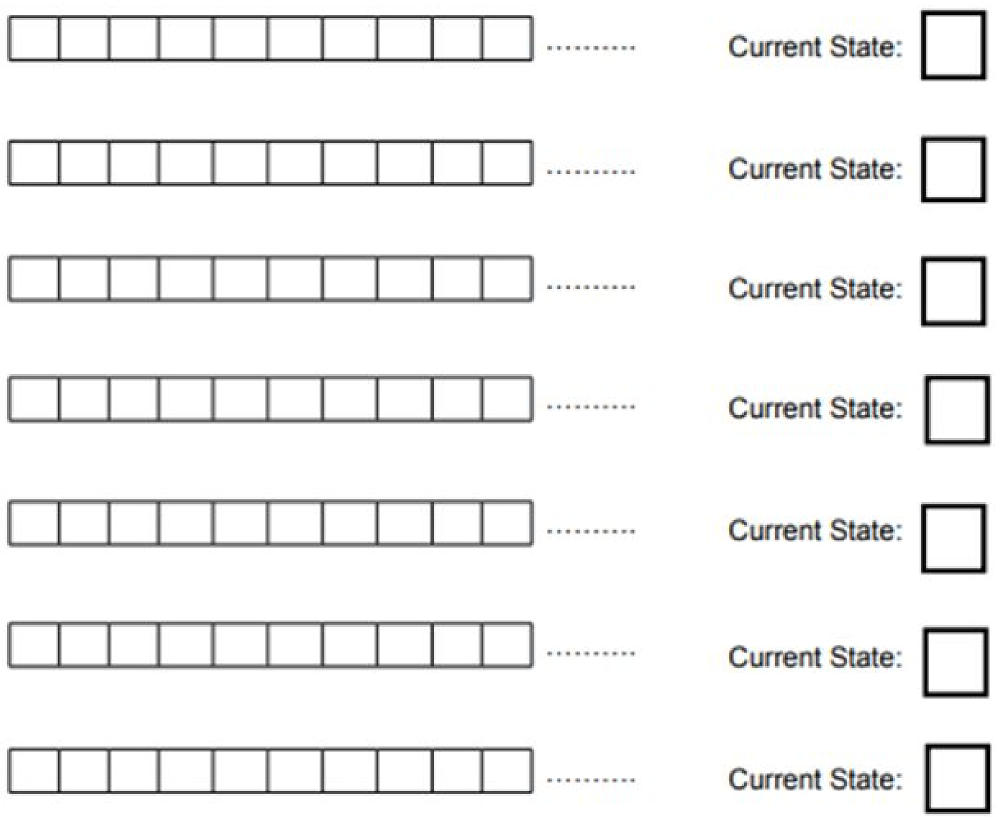
The machine's transition function, δ, is defined by:

$$\delta\,(S_1, 1) = (S_1, 1, \rightarrow)$$
$$\delta\,(S_1, \square) = (S_2, \square, \leftarrow)$$
$$\delta\,(S_2, 1) = (S_3, \square, \leftarrow)$$
$$\delta\,(S_3, 1) = (S_4, \square, \leftarrow)$$

11 (a) The Turing machine is carrying out a computation. The machine starts in state S_1 with the string `1111` on the tape. All other cells contain the blank symbol, □. The read/write head is positioned at the leftmost `1`, as indicated by the arrow.



Trace the computation of the Turing machine, using the transition function δ. Show the contents of the tape, the current position of the read/write head and the current state as the input symbols are processed.



(6 marks)

June 2012 Comp 3

3 (b) A student has been asked to write a program to list duplicate entries in a file containing a list of words. **Figure 2** shows her first attempt at planning an algorithm. The algorithm will not work in all circumstances.

Figure 2

```
Open file
N ← Number of items in file
For Pos1 ← 1 To N Do
  Read item at position Pos1 in file into variable W1
  For Pos2 ← 1 To N Do
    Read item at position Pos2 in file into variable W2
    If W1 = W2 And Not (Pos1 = Pos2)
      Then Output 'Duplicate: ' , W1
    EndIf
  EndFor
EndFor
Close file
```

The basic operation in the algorithm is the `If` statement that compares two words.

The contents of a particular file are shown in **Figure 3**.

Figure 3

File position	Item
1	Rope
2	Dagger
3	Rope

3 (b) (i) Complete **Table 2** below by tracing the execution of the algorithm in **Figure 2** when it is applied to the file in **Figure 3**.

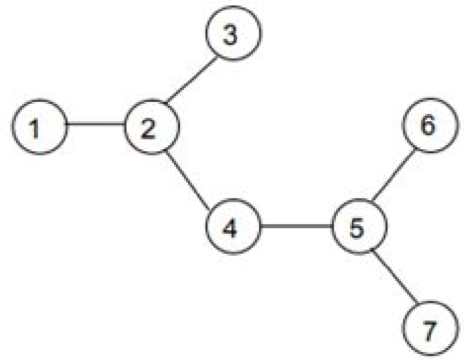
Table 2

N	Pos1	W1	Pos2	W2	Output

(3 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 1 In question parts **0 1 . 1** and **0 1 . 2** two statements are given followed by two conclusions numbered 1 and 2.

You must assume the two statements given in each question are true.

Read the statements and then decide which of the given conclusions logically follows from the two given statements.

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

0 1 . 1 Statements:

All programmers work at night.
Nobody who works at night earns lots of money.

Conclusion 1: All programmers earn lots of money.
Conclusion 2: Some night workers are programmers.

Give answer:

A If only **Conclusion 1** follows.
B If only **Conclusion 2** follows.
C If neither **Conclusion 1** nor **Conclusion 2** follows.
D If both **Conclusion 1** and **Conclusion 2** follow.

[1 mark]

0 1 . 2 Statements:

Some aardvarks are computing professors.
All computing professors love Java.

Conclusion 1: All aardvarks love Java.
Conclusion 2: All computing professors are aardvarks.

Give answer:

A If only **Conclusion 1** follows.
B If only **Conclusion 2** follows.
C If neither **Conclusion 1** nor **Conclusion 2** follows.
D If both **Conclusion 1** and **Conclusion 2** follow.

[1 mark]

0 4 The contents of the arrays `Items` and `NewItems` are shown in **Figure 2**.

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

Figure 2

Items			
[0]	[1]	[2]	[3]
12	25	12	53

NewItems			
[0]	[1]	[2]	[3]
0	0	0	0

Figure 3

```
ItemsCount ← 4
NewItems[0] ← Items[0]
NewItemsCount ← 1
FOR LoopA ← 1 TO ItemsCount - 1
  Done ← False
  FOR LoopB ← 0 TO NewItemsCount - 1
    IF Items[LoopA] = NewItems[LoopB] THEN
      Done ← True
    ENDIF
  ENDFOR
  IF Done = False THEN
    NewItems[NewItemsCount] ← Items[LoopA]
    NewItemsCount ← NewItemsCount + 1
  ENDIF
ENDFOR
```

0 4 . 1 Dry run the algorithm in **Figure 3** by completing **Table 3**. The first row has been completed for you. You may not need to use all of the rows provided in the table.

Copy the contents of all the unshaded cells in **Table 3** into your Electronic Answer Document.

Table 3

ItemsCount	NewItemsCount	LoopA	Done	LoopB	NewItems			
					[0]	[1]	[2]	[3]
4	1				12	0	0	0

[5 marks]

0 4 . 2 Explain the purpose of the algorithm in **Figure 3**.

[1 mark]

June 2017 AS Paper 1

0 2 The algorithm represented using pseudo-code in **Figure 3** describes a method to convert two hexadecimal numbers into decimal. The subroutine `ToDecimal` used in **Figure 3** is shown in **Figure 4** and the built-in subroutine `ASCII` is explained in **Table 2**.

Figure 3

```
FOR Count ← 1 TO 2
  INPUT HexString
  Number ← 0
  FOR EACH HexDigit IN HexString
    Value ← ToDecimal(HexDigit)
    Number ← Number * 16 + Value
  ENDFOR
  OUTPUT Number
ENDFOR
```

The `FOR EACH` command steps through each character in a string working from left to right.

Figure 4

```
SUBROUTINE ToDecimal(HexDigit)
  IF HexDigit = "A" THEN
    Value ← 10
  ELSEIF HexDigit = "B" THEN
    Value ← 11
  ELSEIF HexDigit = "C" THEN
    Value ← 12
  ELSEIF HexDigit = "D" THEN
    Value ← 13
  ELSEIF HexDigit = "E" THEN
    Value ← 14
  ELSEIF HexDigit = "F" THEN
    Value ← 15
  ELSEIF HexDigit IN ["0", "1", ..., "9"] THEN
    Value ← ASCII(HexDigit) - 48
  ELSE
    Value ← -1
  ENDIF
  RETURN Value
ENDSUBROUTINE
```

Table 2

Subroutine used in Figure 4	**Description**
`ASCII(Char)`	Returns the ASCII code of the character passed as a parameter. Example: `ASCII("1")` returns `49`

0 2 . 1 Complete **Table 3** by hand-tracing the algorithm in **Figure 3**. Use `"A2"` and `"1G"` as input strings. You may not need to use all the rows in **Table 3**.

Table 3

`Count`	`HexString`	`Number`	`HexDigit`	`Value`	`Output`

Copy the contents of all the unshaded cells in **Table 3** into your Electronic Answer Document.

[5 marks]

0 2 . 2 Explain how the algorithm in **Figure 3** has attempted to deal with the conversion of `"1G"` into decimal **and** why this method is not fully effective.

[2 marks]

June 2017 Paper 1

0 4 **Figure 4** shows the data Norbert, Phil, Judith, Mary, Caspar and Tahir entered into a binary search tree.

Figure 5 contains pseudo-code for a recursive binary tree search algorithm.

Figure 4

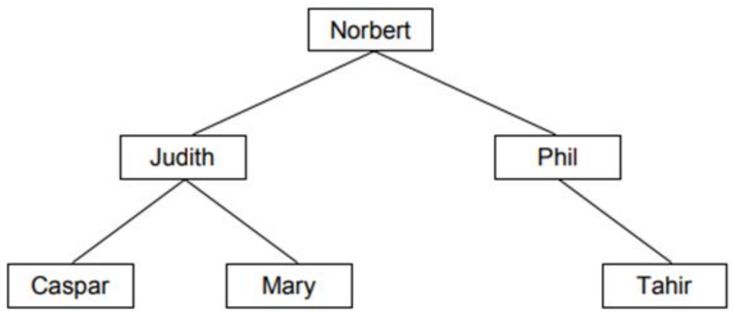


Figure 5

```
FUNCTION TreeSearch(target, node)
  OUTPUT 'Visited ', node
  IF target = node THEN
    RETURN True
  ELSE IF target > node AND Exists(node, right) THEN
    RETURN TreeSearch(target, node.right)
  ELSE IF target < node AND Exists(node, left) THEN
    RETURN TreeSearch(target, node.left)
  ENDIF
  RETURN False
ENDFUNCTION
```

The subroutine `Exists` takes two parameters – a node in the binary tree and a direction (`left` or `right`). It returns a Boolean value indicating if the node given as a parameter has a child node in the direction specified by the second parameter. For instance, `Exists(Mary, left)` will return a value of `False` as there is no node to the left of `Mary` in the binary tree.

`node.right` evaluates to the child node to the right of `node`, eg `Judith.right` is `Mary`.

`node.left` evaluates to the child node to the left of `node`, eg `Judith.left` is `Caspar`.

0 4 . 1 What is meant by a recursive subroutine?

[1 mark]

0 4 . 2 There are two base cases for the subroutine `TreeSearch`. State **one** of the base cases.

[1 mark]

0 4 . 3 Complete the unshaded cells of **Table 3** to show the result of tracing the `TreeSearch` algorithm shown in **Figure 5** with the function call `TreeSearch(Olivia, Norbert)`. You may not need to use all of the rows.

[3 marks]

Table 3

Function call	**Output**
`TreeSearch(Olivia, Norbert)`	

Copy the contents of the unshaded cells in **Table 3** into the table in your Electronic Answer Document.

June 2011 Comp 1

The contents of an array `Scores` are shown in **Figure 2**.

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

Figure 2

Scores							
[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
18	23	36	21	58	40	45	59

Figure 3

```
Max ← 8
FOR Count1 ← 1 TO (Max - 1) DO
  FOR Count2 ← 1 TO (Max - 1) DO
    IF Scores[Count2] > Scores[Count2 + 1]
      THEN
        Temp ← Scores[Count2]
        Scores[Count2] ← Scores[Count2 + 1]
        Scores[Count2 + 1] ← Temp
    ENDIF
  ENDFOR
ENDFOR
```

1 8 **One pass** is made through the outer loop of the algorithm in **Figure 3**.

Complete **Table 2** to show the changed contents of the array `Scores` after this single pass. You may use **Table 3** to help you work out your answer, though you are neither required to use **Table 3** nor to copy it into your Electronic Answer Document.

Copy the bottom row of your completed ***Table 2*** *into the Electronic Answer Document.*

Table 2

Scores							
[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]

Table 3

Max	Count1	Count2	Temp	Scores							
				[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
				18	23	36	21	58	40	45	59

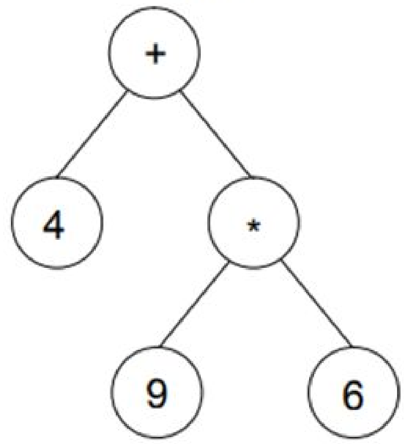
(4 marks)

1 9 What is the name of the standard algorithm shown in **Figure 3**? *(1 mark)*

June 2013 Comp 3

4 A tree can be used to represent a mathematical expression. This is known as an expression tree. **Figure 5** is an expression tree for the infix expression 4 + 9 * 6.

Figure 5



4 (b) The expression tree in **Figure 5** could be represented using three one-dimensional arrays named **A**, **B** and **C**. **Figure 6** shows a representation of **Figure 5** together with the array indices.

Figure 6

Arrays

Index	A	B	C
[1]	+	2	3
[2]	4	0	0
[3]	*	4	5
[4]	9	0	0
[5]	6	0	0

4 (d) The procedure in **Figure 7** describes a type of tree traversal that can be carried out on the representation of the tree shown in **Figure 6**.

Figure 7

```
Procedure Traverse(Pos:Integer)
  If B[Pos] > 0 Then Traverse(B[Pos])
  If C[Pos] > 0 Then Traverse(C[Pos])
  Output A[Pos]
End Procedure
```

Using the table below, trace the execution of the procedure when it is called using `Traverse(1)`. You may not need to use all of the lines provided in the table.

Pos	Output

(4 marks)

4 (e) Which type of tree traversal does the procedure `Traverse` carry out?

....

(1 mark)

4 (f) What does the output of the procedure represent?

....

(1 mark)

Specimen Paper 1

0 1 The famous detective John Stout was called in to solve a perplexing murder mystery. He determined the following facts.

(a) Nathan, the murdered man, was killed by a blow on the head.
(b) Either Suzanne or Martin was in the dining room at the time of the murder.
(c) If Peter was in the kitchen at the time of the murder, then Ian killed Nathan using poison.
(d) If Suzanne was in the dining room at the time of the murder, then Steve killed Nathan.
(e) If Peter was not in the kitchen at the time of the murder, then Martin was not in the dining room when the murder was committed.
(f) If Martin was in the dining room at the time the murder was committed, then Paul killed Nathan.
(g) If Kevin was in the hall at the time of the murder, then Suzanne killed Nathan by a blow to the neck with a saucepan.

0 1 . 1 Who murdered Nathan?

A Paul
B Steve
C Suzanne
D Ian
E It is not possible for John Stout to solve the crime.

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

[1 mark]

0 1 . 2 Explain how you know your answer to **0 1 . 1** is correct.

[2 marks]

Use the space below for rough working, then write your answer in your Electronic Answer Document.

0 3 The Cat transportation company (CTC) is a business that specialises in preparing cats for cat shows.

They need to take five cats to the AQA cat show. They will transport the cats in their van. CTC owns only one van.

They cannot put all the cats in their van at the same time because some of the cats get stressed when in the company of some of the other cats. The cats would not therefore arrive in top condition for the cat show if they were all in the van at the same time.

The graph in **Figure 3** shows the relationships between the five cats (labelled 1 to 5). If there is an edge between two cats in the graph then they **cannot** travel in the van together at the same time.

Figure 3

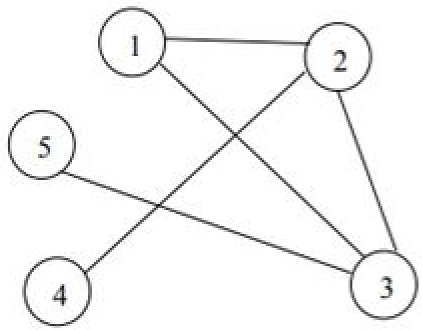


Figure 4 shows an algorithm, written in pseudo-code, that CTC use.

Figure 4

```
NoOfCats ← 5
Cat[1] ← 1
FOR A ← 2 TO NoOfCats
  B ← 1
  C ← 1
  WHILE B < A DO
    IF M[A, B] = 1
      THEN
        IF Cat[B] = C
          THEN
            B ← 1
            C ← C + 1
          ELSE B ← B + 1
        ENDIF
      ELSE B ← B + 1
    ENDIF
  ENDWHILE
  Cat[A] ← C
ENDFOR
```

The two-dimensional array, `M`, is used to store the adjacency matrix shown in **Table 4**.

0 3 . 4 Complete **Table 5** to show the result of tracing the algorithm in **Figure 4**.

[6 marks]

Copy your answer in **Table 5** into the Electronic Answer Document.

Table 5

				Cat				
NoOfCats	**A**	**B**	**C**	**1**	**2**	**3**	**4**	**5**

0 3 . 5 Explain the purpose of the algorithm in **Figure 4**.

[1 mark]