0 1 **Figure 1** shows a binary tree containing seven nodes. **Figure 2** shows how the binary tree in **Figure 1** could be represented using three one-dimensional arrays: `Data`, `Dir1` and `Dir2`.

Figure 1

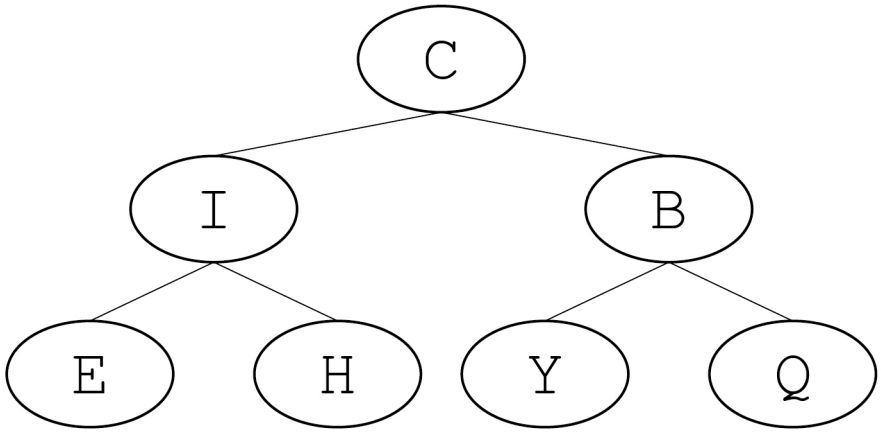

Figure 2

Index	Data	Dir1	Dir2
[0]	C	1	4
[1]	I	2	3
[2]	E	-1	-1
[3]	H	-1	-1
[4]	B	5	6
[5]	Y	-1	-1
[6]	Q	-1	-1

0 1 . 1 The output of a post-order traversal algorithm used to print the data item at each node in the binary tree shown in **Figure 1** would be `E`, `H`, `I`, `Y`, `Q`, `B`, `C`.

State the output that would be produced by an **in-order** traversal algorithm.

[2 marks]

Figure 3 shows pseudo-code for a subroutine called `Traversal` that uses the three arrays from **Figure 2**.

Figure 3

```
SUBROUTINE Traversal(StartNode)
  Current ← StartNode
  Pos ← 0
  Stack[Pos] ← Current
  WHILE Pos ≠ -1
    Current ← Stack[Pos]
    Pos ← Pos - 1
    OUTPUT Data[Current]
    IF Dir2[Current] ≠ -1 THEN
      Pos ← Pos + 1
      Stack[Pos] ← Dir2[Current]
    ENDIF
    IF Dir1[Current] ≠ -1 THEN
      Pos ← Pos + 1
      Stack[Pos] ← Dir1[Current]
    ENDIF
  ENDWHILE
ENDSUBROUTINE
```

0 1 . 2 Complete the unshaded cells in **Table 2** to show the result of the subroutine call `Traversal(0)`

Table 2

		Stack				
Current	**Pos**	**[0]**	**[1]**	**[2]**	**[3]**	**Output**

Copy the contents of the unshaded cells in **Table 2** into the table in your Electronic Answer Document.

[7 marks]

0 2 **Figure 3** shows a graph containing five nodes. **Figure 4** shows how the graph in **Figure 3** could be represented using three one-dimensional arrays: `Data`, `Dir1` and `Dir2`.

Figure 3

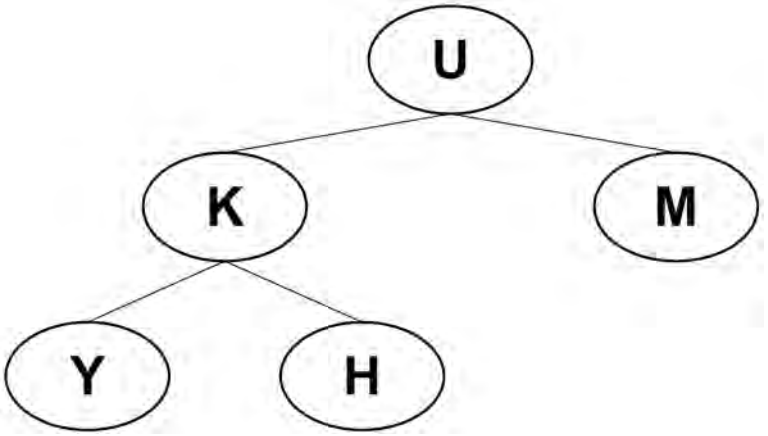

Figure 4

Index	Data	Dir1	Dir2
[0]	U	1	4
[1]	K	2	3
[2]	Y	-1	-1
[3]	H	-1	-1
[4]	M	-1	-1

0 2 . 1 The graph in **Figure 3** is a binary tree. A binary tree is a rooted tree where each node has at most two child nodes.

There are three properties that a graph needs to have for it to be a tree. One of those properties is that it contains no cycles.

State the other **two** properties of this graph that make it a tree.

[2 marks]

Figure 5 contains pseudo-code for an algorithm that uses the arrays in **Figure 4**.

Figure 5

```
Done ← False
Pos ← –1
Current ← 0
WHILE Done = False
  WHILE Current ≠ –1
    Pos ← Pos + 1
    Temp[Pos] ← Current
    Current ← Dir1[Current]
  ENDWHILE
  IF Pos = –1 THEN
    Done ← True
  ELSE
    OUTPUT Data[Temp[Pos]]
    Current ← Dir2[Temp[Pos]]
    Pos ← Pos – 1
  ENDIF
ENDWHILE
```

0 2 . 2 Complete the unshaded cells in **Table 1** to show the result of tracing the algorithm shown in **Figure 5** using the arrays in **Figure 4**.

Table 1

		Temp				
Done	**Pos**	**[0]**	**[1]**	**[2]**	**Current**	**OUTPUT**

Copy the contents of the unshaded cells in **Table 1** into the table in your Electronic Answer Document.

[7 marks]

0 2 . 3 The array `Temp` needs to be able to store three values when used with the binary tree shown in **Figure 3**, on page 6.

For some binary trees with only five nodes, the array `Temp` would need to be able to store five values.

Describe the structure of a five-node binary tree that would require `Temp` to be able to store five values.

[2 marks]

0 2 . 4 State the type of data structure the algorithm shown in **Figure 5** implements using the array `Temp`.

[1 mark]

0 2 . 5 Describe the changes that need to be made to the algorithm shown in **Figure 5** so that the order that the data values are output in is reversed.

In your answer, you should only describe changes to the existing lines of code in the algorithm; you must **not** suggest the addition of extra lines of code.

[1 mark]