2.5 Trees

Name: ____________________

Class: ____________________

Date: ____________________

Time: **228 minutes**

Marks: **155 marks**

Comments:

Q1.

A computer program stores a list of integers in an array named `List`. The numbers in the array are to be sorted into ascending order so that a particular efficient search algorithm can be used to search for a number.

(a) One of the search algorithms in **Table 1** can only be used successfully on a sorted list.

Place **one** tick next to the name of the algorithm that requires a list to be sorted.

Table 1

Algorithm Name	**Requires Sorted List? (Tick one box)**
Binary search	
Linear search	

(1)

(b) The pseudo-code for a standard algorithm that can be used to sort the data in the array `List` into order is shown in **Figure 1**. The variable `ListLength` stores a count of the number of items in the array `List`.

Array indexing starts at 1.

Figure 1

```
For OuterPointer ← 2 To ListLength
  CurrentValue ← List[OuterPointer]
  InnerPointer ← OuterPointer - 1
  While InnerPointer > 0 And
        List[InnerPointer] > CurrentValue Do
    List[InnerPointer + 1] ← List[InnerPointer]
    InnerPointer ← InnerPointer - 1
  EndWhile
  List[InnerPointer + 1] ← CurrentValue
EndFor
```


Complete the empty (unshaded) cells in the trace table (**Table 2**) for an execution of the algorithm in **Figure 1** when the array `List` contains the values `9, 8, 5` and `6` in that order.

Table 2

				List			
List	**Outer**	**Current**	**Inner**	[1]	[2]	[3]	[4]

Length	Pointer	Value	Pointer	9	8	5	6
4	2		1				
			0				
	3		2				
			1				
			0				
	4		3				
			2				
			1				

(3)

(c) In the trace table (**Table 2**), when the variable `OuterPointer` contains the value `2` and then `3`, the value of the variable `InnerPointer` decreases to `0`. When `OuterPointer` contains `4`, `InnerPointer` stops decreasing when it reaches the number `1`.

Explain why `InnerPointer` does not decrease to `0` when `OuterPointer` contains `4`.

____

____

____

____

(1)

(d) Tick **one** box in **Table 3** to indicate the correct Order of **Time** Complexity of the standard algorithm in **Figure 1**.

Table 3

Order of Time Complexity	Tick one box
O(n)	
$O(n^2)$	
$O(2^n)$	

(1)

(e) State the name of the standard algorithm that is represented by the pseudo-code in **Figure 1**.

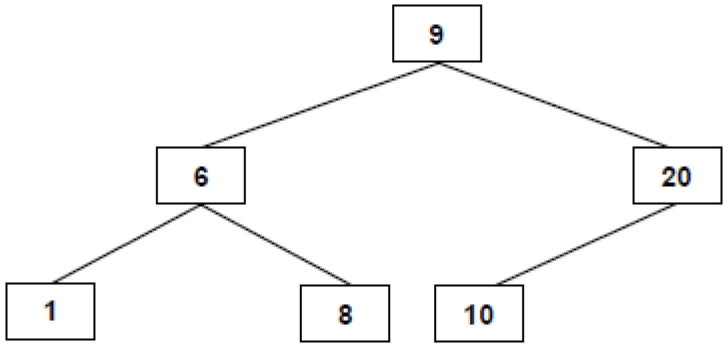
____

(1)

(f) Instead of storing a list of numbers in an array as in (b), the numbers could be stored in a binary search tree. This would also enable efficient searching.

The numbers 9, 6, 1, 8, 20 and 10 are put into a binary search tree in that order. **Figure 2** shows this binary search tree.

Figure 2



(i) A search of the binary tree is performed for the number `8`.

List the numbers, in the order that they would be checked, for the search to determine that the number `8` **is present** in the tree.

____

(1)

(ii) A search of the binary tree is performed for the number 11.

List the numbers, in the order that they would be checked, for the search to determine that the number `11` **is not present** in the tree.

____

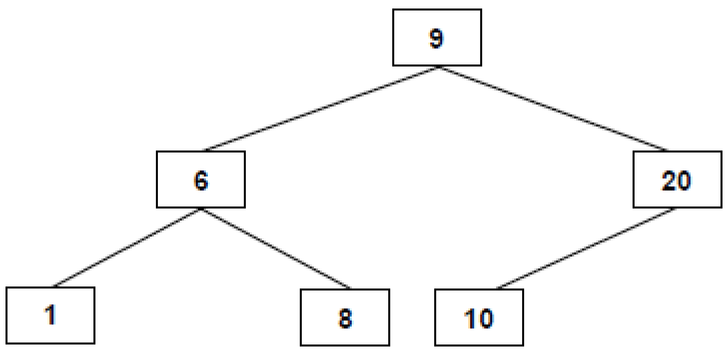
(1)

(g) The numbers 4, 5 and 3 are to be added into the binary search tree, in that order.

Figure 3 below is an identical copy of **Figure 2**.

Complete **Figure 3** below to show the binary search tree from **Figure 2** after the extra numbers have been added into it.

Figure 3



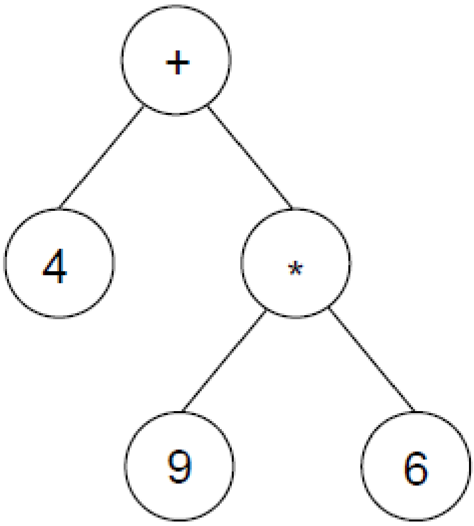
(2)

(Total 11 marks)

Q2.

A tree can be used to represent a mathematical expression. This is known as an expression tree. **Figure 1** is an expression tree for the infix expression 4 + 9 * 6.

Figure 1



(a) An expression tree is an example of a rooted tree.

State the contents of the root node:_____________________________

List the contents of **all** of the leaf nodes: _________________________

(2)

(b) The expression tree in **Figure 1** could be represented using three one-dimensional arrays named **A**, **B** and **C**. **Figure 2** shows a representation of **Figure 1** together with the array indices.

Figure 2
Arrays

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

Describe the role of each of the arrays **A**, **B** and **C**.

A:

____

B:

____

C:

____

(3)

(c) What does an entry of `0` in array **B** indicate?

____

____

(1)

(d) The procedure in **Figure 3** describes a type of tree traversal that can be carried out on the representation of the tree shown in **Figure 2**.

Figure 3

```
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)

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

____

(1)

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

____

(1)
(Total 12 marks)

Q3.

A graph can be drawn to represent a maze. In such a graph, each graph vertex represents one of the following:

- the entrance to or exit from the maze
- a place where more than one path can be taken
- a dead end.

Edges connect the vertices according to the paths in the maze.

Diagram 1 shows a maze and **Diagram 2** shows one possible representation of this maze.
Position 1 in **Diagram 1** corresponds to vertex 1 in **Diagram 2** and is the entrance to the maze. Position 7 in **Diagram 1** is the exit to the maze and corresponds to vertex 7.
Dead ends have been represented by the symbol ——| in **Diagram 2**.

Diagram 3 shows a simplified undirected graph of this maze with dead ends omitted.

Diagram 1

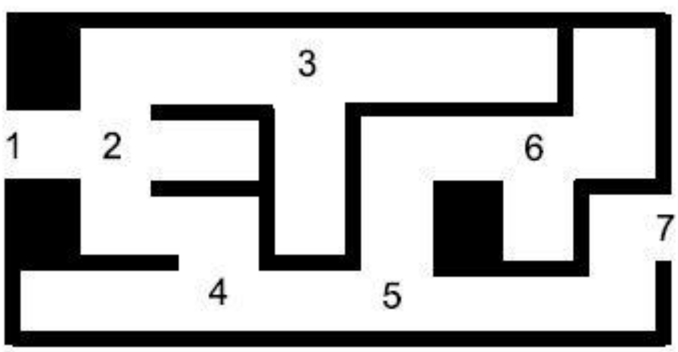
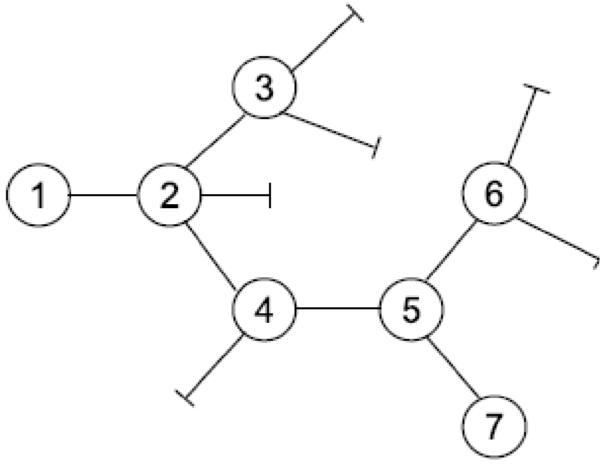
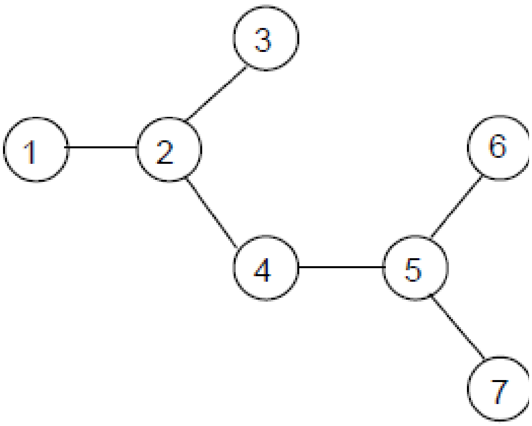


Diagram 2

Diagram 3



Representation of maze including dead ends



Graph representing maze with dead ends omitted

(a) The graph in **Diagram 3** is a tree.

State **one** property of the graph in **Diagram 3** that makes it a tree.

____

____

(1)

(b) The graphs of some mazes are not trees.

Describe a feature of a maze that would result in its graph **not** being a tree.

____

____

(1)

(c) Complete the table below to show how the graph in **Diagram 3** would be stored using an adjacency matrix.

(2)

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

____

____

(1)

(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.

____

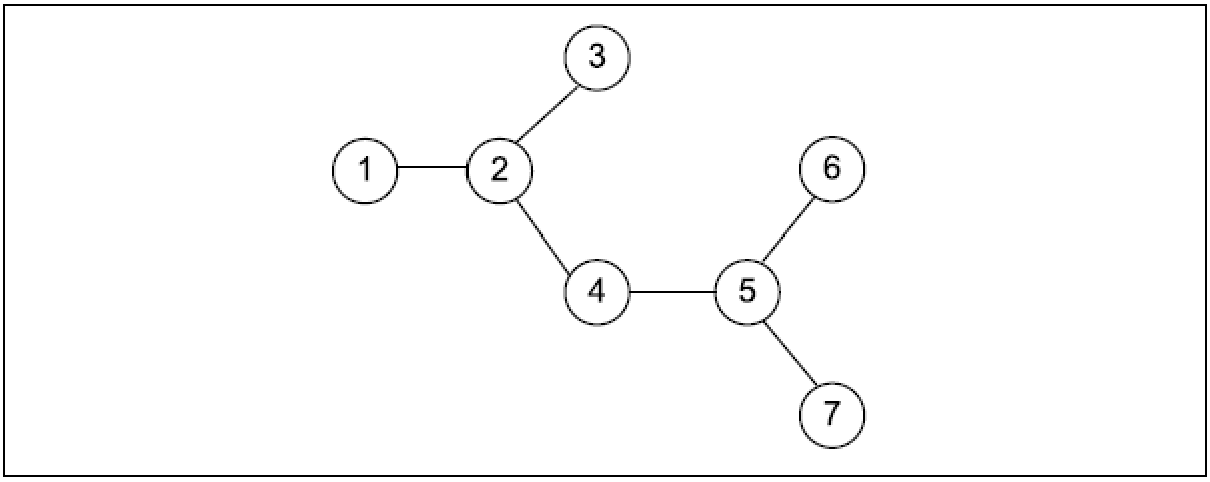
____

____

____

(2)

Diagram 3 is repeated here so that you can answer Question (e) without having to turn pages.



(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 the diagram below is to be used to explore the graph in **Diagram 3**. It has two parameters, `V` (the current vertex) and `EndV` (the exit vertex).

```
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`.

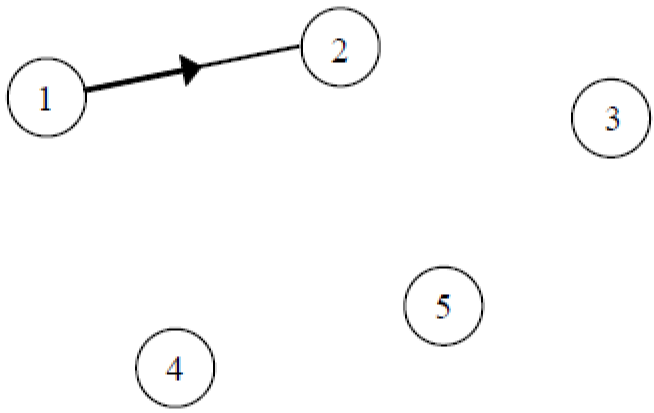
				Discovered							CompletelyExplored							
Call	V	U	EndV	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	Found
	-	-		F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
DFS(1,7)	1	2	7															
DFS(2,7)	2	1	7															
		3	7															
DFS(3,7)	3	2	7															
DFS(2,7)	2	4	7															
DFS(4,7)	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)
(Total 12 marks)

Q4.
The table below shows an adjacency matrix representation of a directed graph (digraph).

		1	**2**	**3**	**4**	**5**
From	**1**	0	1	0	1	0
	2	0	0	1	1	0
	3	0	0	0	0	0
	4	0	0	0	0	1
	5	0	1	0	0	0

(a) Complete this unfinished diagram of the directed graph.



(2)

(b) Directed graphs can also be represented by an adjacency list.

Explain under what circumstances an adjacency matrix is the most appropriate method to use to represent a directed graph, and under what circumstances an adjacency list is more appropriate.

____

____

____

(2)

(c) A tree is a particular type of graph.

What properties must a graph have for it to be a tree?

____

____

____

(2)

(d) Data may be stored as a binary tree.

Show how the following data may be stored as a binary tree for subsequent processing in alphabetic order by drawing the tree. Assume that the first item is the root of the tree and the rest of the data items are inserted into the tree in the order given.

Data items: Jack, Bramble, Snowy, Butter, Squeak, Bear, Pip

(3)

(e) A binary tree such as the one created in part (d) could be represented using one array of records or, alternatively, using three one-dimensional arrays.

Describe how the data stored in the array(s) could be structured for **one** of these two possible methods of representation.

____

____

____

____

____

____

____

____

(3)
(Total 12 marks)

Q5.

A binary tree has the following functions defined

`RootValue(T)`	Returns the contents of the root node of the tree T
`LeftChild(T)`	Returns the left child of the root node of the tree T
`RightChild(T)`	Returns the right child of the root node of the tree T

A recursively-defined procedure P with a tree as a parameter is defined below.

```
Procedure P(T)
  If RightChild(T) Exists
   Then P(RightChild(T))
  Output RootValue(T)
  If LeftChild(T) Exists
    Then P(LeftChild(T))
EndProc
```

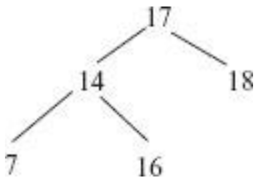
(a) What is meant by a recursively-defined procedure?

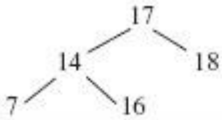
____

____

(1)

(b) (i) Complete the table below by dry running the procedure call P(T) for the tree T given below



Procedure Call	T
P_1	

Output					

(7)

(ii) What does the procedure P describe?

____

(2)
(Total 10 marks)

Q6.

A binary search tree is used by software to store and then search for user names on a college network.

The following are the first seven user names to join the tree:

PollardJ, AtkinsP, RogersG, AbbottJ, SearleF, CollinsK, RuddleA

(a) Sketch the tree structure.

(2)

(b) The tree is to be searched for various user names.

(i) The task is to search for the user name **CollinsK**. List in order the nodes visited.

____

(1)

(ii) A second search is done to find the user name **RuddleA**. How many comparisons does this require?

____

(1)

(Total 4 marks)

Q7.

A binary search tree has the following functions defined:

RootValue(T) Returns the value stored in the root node of the tree T
LeftChild(T) Returns the left child (subtree) of the root node of the tree T
RightChild(T) Returns the right child (subtree) of the root node of the tree T

A recursively-defined procedure P with a tree as a parameter is defined below.

```
Procedure P(T)
  If RightChild(T) exists
    Then P(RightChild(T))
  Output RootValue(T)
  If LeftChild(T) exists
    Then P(LeftChild(T))
EndProc
```

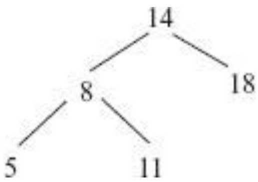
(a) What is meant by a recursively-defined procedure?

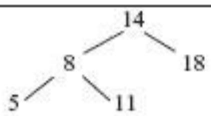
____

____

(1)

(b) (i) Complete the table below by dry running the procedure call P(T) for the tree T given below.



Procedure Call		Output
P_1		

(6)

(ii) What does the procedure P describe?

____

(2)

(Total 9 marks)

Q8.

A tree has the following functions defined:

RootValue(T)	Returns the contents of the root node of the tree T
LeftChild(T)	Returns the left child of the root node of the tree T
RightChild(T)	Returns the right child of the root node of the tree T

A recursively-defined procedure P with a tree as a parameter is defined below.

```
Procedure P (T)
     If LeftChild(T) exists
          then P(LeftChild(T))
     Output RootValue(T)
     If RightChild(T) exists
          then P(RightChild(T))
EndProc
```

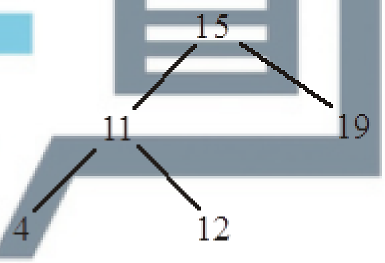
(a) What is meant by recursively-defined?

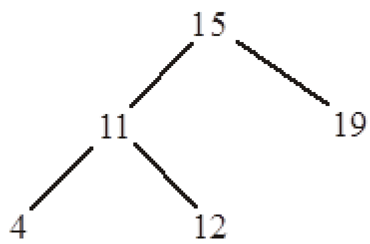
____

____

(1)

(b) (i) Complete the table below by dry running the procedure call P(T) for the tree T given below.



Procedure Call	T		Output
P_1			

(6)

(ii) What does procedure P describe?

____

(2)
(Total 9 marks)

Q9.

(a) In the context of data structures what is meant by the terms:

(i) FIFO; ____

(ii) LIFO? ____

(2)

(b) Queue and stack are examples of data structures. Tick in the following table to indicate whether they are FIFO or LIFO data structures.

	FIFO	LIFO
Queue		
Stack		

(2)

(c) Describe **one** example of the use of a stack.

____

____

____

(2)

(d) Describe **one** example of the use of a Binary Search Tree.

____

____

____

(2)

(Total 8 marks)

Q10.

A *recursively-defined* procedure **Process**, which takes an integer as its single parameter, is defined below.

(a) What is meant by recursively-defined?

____

____

(1)

(b) Describe how a stack is used in the execution of procedure **Process**?

____

____

____

____

(2)

(c) Dry run the procedure call **Process(1)**, using the data in the table below, showing clearly the order the values are printed.

```
 Procedure Process (P)
   Print (P)
   If Table[P].Left <> 0
      Then Process (Table[P].Left)
   EndIf
   Print (Table[P].Data)
   If Table[P].Right <> 0
      Then Process (Table[P].Right)
   EndIf
EndProcedure
```

		Table	
	Data	Left	Right
[1]	Jones	3	2
[2]	Smith	0	0
[3]	Bremner	5	4
[4]	Fortune	0	0
[5]	Bird	0	0

Printed Output:= ____

____

(6)

(d) What does procedure Process describe?

____

(1)

(Total 10 marks)

Q11.

A binary search tree is a data structure where items of data are stored such that they can be searched for quickly and easily.

The following data items are to be entered into a binary search tree in the order given:

Louise, Peter, Robert, Christine, Alan, Leslie, Maria

(a) Draw a diagram to show how these values will be stored in the tree.

(4)

(b) Circle the root node in your diagram.

(1)

(c) If Maria is being searched for in this binary tree, list the data items which have to be accessed.

____

(1)

(Total 6 marks)

Q12.

An algebraic expression is represented in a binary tree as follows.

(a) On the above diagram, circle and label the *root* of this tree, a *branch* and a *leaf node*.

(3)

(b) In the spaces below, draw the *left sub-tree* and the *right sub-tree* of this tree.

left sub-tree | right sub-tree

(2)

(c) What is the result if this tree is printed using in-order traversal?

____

(3)
(Total 8 marks)

Q13.

For the expression 3+x the binary tree stores + at the root, 3 at the left hand node and x at the right hand node. If the nodes of this tree are printed as the tree is traversed, what will be printed when the traversal is

(a) pre-order; ____

(b) in-order; ____

(c) post-order? ____

(Total 3 marks)

Q14.

(a) The series of characters J, F, H, U, S, X, T are to be entered into a binary search tree in the order given. Draw a diagram to show how these values will be stored.

(4)

(b) The following data are held in arrays Data, L and R:

	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Data	'J'	'F'	'H'	'U'	'S'	'X'	'T'
L	2	0	0	5	0	0	0
R	4	3	0	6	7	0	0

Using the arrays above, dry-run the following pseudo-code by completing the trace table opposite:

```
Item ← 'T'

Ptr ← 1
WHILE Data[Ptr] < > Item DO
  PRINT Data[Ptr]
  IF Data[Ptr] > Item

     THEN Ptr ← L[Ptr]

     ELSE Ptr ← R[Ptr]
  ENDIF
ENDWHILE
PRINT Data[Ptr]
```

Trace Table:

Item	Ptr	Printed Output
'T'	1	'J'

(6)

(Total 10 marks)

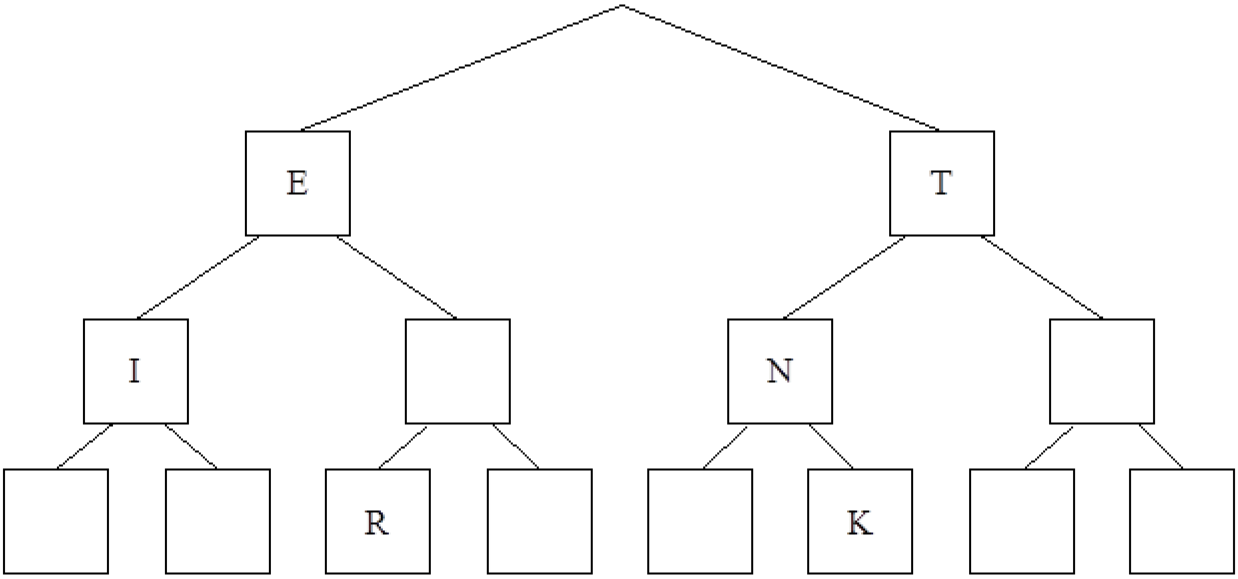
Q15.

A binary tree can be used to represent the alphabet in a code. Part of the tree is shown below. Starting at the root of the tree, branch left is a dot and branch right is a dash.

So N has the code: dash dot.

SOS has the code dot dot dot dash dash dash dot dot dot.

(a) Place the missing letters S and O into the correct positions in the diagram.



(2)

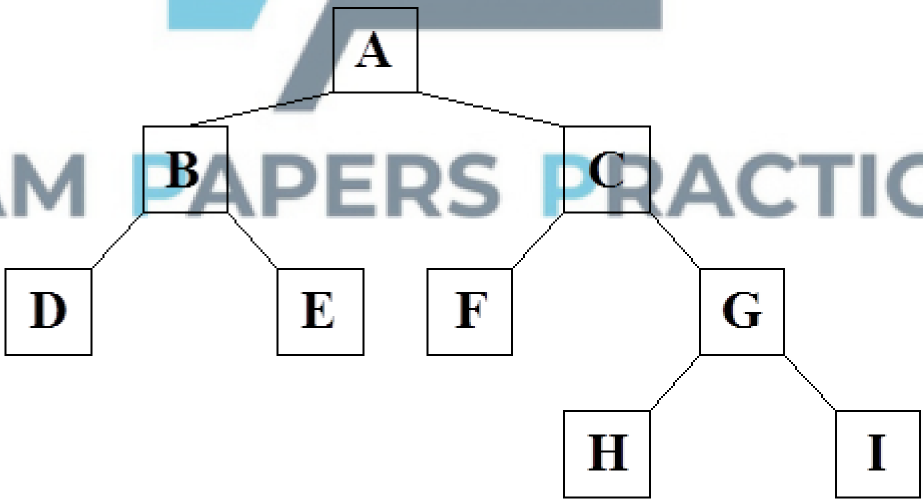
(b) What does the following 2 letter code spell: dot dot dash ?

____

(2)
(Total 4 marks)

Q16.

The diagram below shows the structure of a binary tree.



(a) Identify each of the following:

(i) the root node,

____

(ii) the parent nodes,

____

(iii) the leaf (terminal) nodes.

(5)

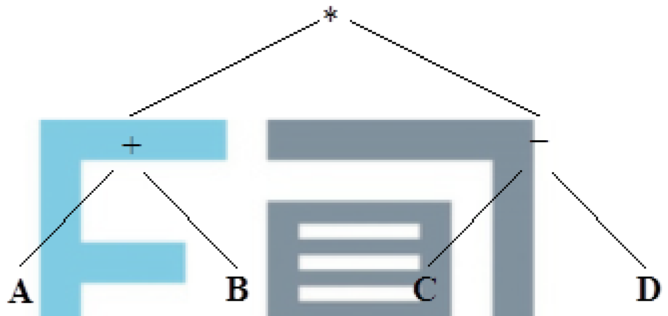
(b) Each node contains a data item. What else must a node contain?

(2)

(Total 7 marks)

Q17.

An algebraic expression is represented in a binary tree as follows:



(a) Label its *root*, a *branch* and a *leaf node.*

(3)

(b) Mark and label the *left sub-tree* and the *right sub-tree* of this tree.

(2)

A recursively-defined procedure T, which takes a tree structure, tree(x, y, z) as its single parameter, where x is the root, y is the left sub-tree and z is the right sub-tree, is defined below (<> means not equal to).

```
Procedure T (tree(x, y, z))

  If y <> empty
   Then
        PRINT ')'
        T(y)
 EndIf
   PRINT x
   If z <> empty
   Then
        T(z)
       PRINT ')'
   EndIf
 EndProc
```

(c) What is meant by *recursively-defined*?

(1)

(d) Explain why a stack is necessary in order to execute procedure T recursively.

(3)

(e) Dry run the following procedure call

T (tree('*', tree ('+', tree ('A', empty, empty), tree ('B', empty, empty)),
tree ('–', tree ('C', empty, empty), tree ('D', empty, empty))
)
)

showing clearly the PRINTed output and the values of the parameter omitted from the table (rows 4, 5, 6, 7) for the **seven** calls of T.

Call Number	**Parameter**
1	tree('*', tree('+', tree('A',empty,empty), tree('B',empty,empty)), tree('–', tree('C',empty,empty), tree('D',empty,empty)))
2	tree(' +', tree('A',empty,empty), tree('B',empty,empty))
3	tree('A',empty,empty)
4	
5	
6	
7	

(10)

(f) What tree traversal algorithm does procedure T describe?

(1)

(Total 20 marks)