



4.3.1 Graph and Tree Traversal

1) **Figure 1** shows a binary tree containing six nodes.

Figure 2 shows how the binary tree in **Figure 1** could be represented using three one-dimensional arrays: Data, Left and Right.

Figure 1

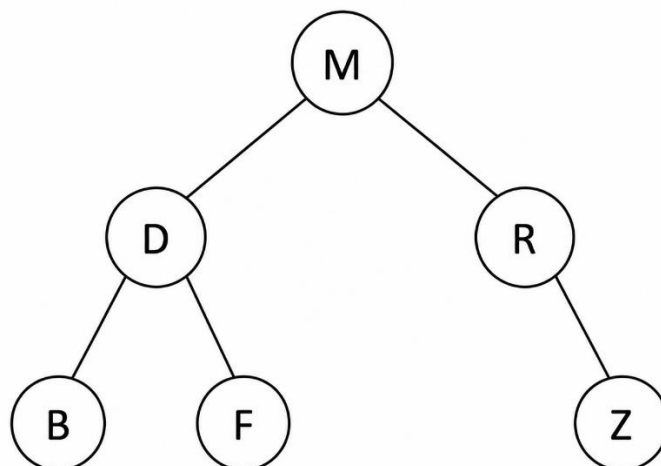


Figure 2

Index	Data	Left	Right
0	M	1	4
1	D	2	3
2	B	-1	-1
3	F	-1	-1
4	R	-1	5
5	Z	-1	-1

a) Define the term binary tree.

[2 marks]

- **Rooted (tree);**
- **Where each node has at most two child nodes;**

b) The output of an in-order traversal algorithm used to print the data item at each node in the binary tree shown in **Figure 1** would be B, D, F, M, R, Z.

State the output that would be produced by a pre-order traversal algorithm.

[2 marks]

- **M, D, B, F, R, Z;**





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

Figure 3

```
SUBROUTINE Traversal()  
    Current  $\leftarrow$  0  
    Pos  $\leftarrow$  -1  
    LastVisited  $\leftarrow$  -1  
  
    WHILE Current  $\neq$  -1 OR Pos  $\neq$  -1  
  
        IF Current  $\neq$  -1 THEN  
            Pos  $\leftarrow$  Pos + 1  
            Stack[Pos]  $\leftarrow$  Current  
            Current  $\leftarrow$  Left[Current]  
  
        ELSE  
            Current  $\leftarrow$  Stack[Pos]  
  
            IF Right[Current]  $\neq$  -1 AND Right[Current]  $\neq$  LastVisited THEN  
                Current  $\leftarrow$  Right[Current]  
            ELSE  
                OUTPUT Data[Current]  
                LastVisited  $\leftarrow$  Current  
                Pos  $\leftarrow$  Pos - 1  
                Current  $\leftarrow$  -1  
            ENDIF  
  
        ENDIF  
  
    ENDIF  
  
    ENDWHILE  
ENDSUBROUTINE
```





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

Table 1

			Stack					Output
Current	Pos	V	[0]	[1]	[2]	[3]	[4]	
0	-1	-1						
1	0		0					
2	1			1				
-1	2				2			
2								B
-1	1	2						
1								
3								
-1	2				3			
3	1	3						F
-1								
1	0	1						D
-1								
0								
4								
-1	1			4				
4								
5								
-1	2				5			
5	1	5						Z
-1								
4	0	4						R
-1								
0	-1	0						M
-1								

[7 marks]

