PROGRAMMING CHALLENGE 11: COUNTRY BINARY TREE

The task is to store a dataset (maximum size 20 items) as a binary tree structure. You should assume that the data items are unique.

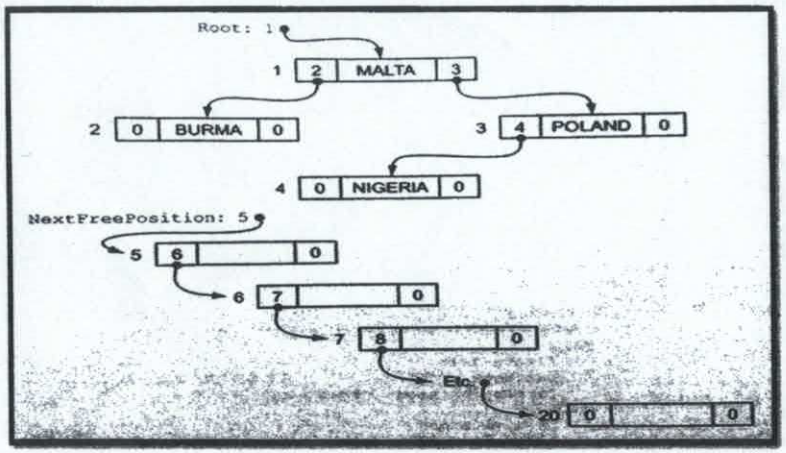
The program will use a user-defined type Node for each node defined as follows:

Identifier	Data Type	Description
LeftP	INTEGER	The left pointer for the node
Data	STRING	The node's data value
RightP	INTEGER	The right pointer for the node

A linked list is maintained of all the unused nodes which do not form part of the tree. The first available node which is used for a new term is indicated by NextFreePosition. Items in the unused list are linked using their left pointers.

The binary tree and linked list are implemented using variables as follows:

Identifier	Data Type	Description
ThisTree	ARRAY[20] : Node	The tree data
Root	INTEGER	Index for the root position of the ThisTree array
NextFreePosition	INTEGER	Index for the next unused node



The diagram shows the binary tree and linked list after four values have been added

PROGRAMMING CHALLENGE 11: COUNTRY BINARY TREE

Task 3.1
Write the program code to declare all the required variables and create the initial linked list which contains all 20 nodes.
Add statement(s) to initialise the empty tree.

Evidence 7:
Your program code for Task 3.1. **[11]**

The following (incomplete) pseudocode inserts a data value into the binary tree structure.

The LastMove variable holds the direction of the previous traversal move as follows:

X - no move yet made
L - move was to the left
R - move was to the right

```
PROCEDURE AddItemToBinaryTree(NewTreeItem)

    IF Root = 0
        THEN
            Root ← NextFreePosition
        ELSE
            //traverse the tree to find the position for the new value
            CurrentPosition ← Root
            LastMove ← 'X'
            REPEAT
                PreviousPosition ← CurrentPosition
                IF NewTreeItem < ThisTree[CurrentPosition].Data
                    THEN
                        // move left
                        LastMove ← 'L'
                        CurrentPosition ← ThisTree[CurrentPosition].LeftP
                    ELSE
                        // move right
                        LastMove ← 'R'
                        CurrentPosition ← ThisTree[CurrentPosition].RightP
                ENDIF
            UNTIL CurrentPosition = 0
    ENDIF

    IF LastMove = 'R'
        THEN
            ThisTree[PreviousPosition].RightP ← NextFreePosition
        ELSE
            ThisTree[PreviousPosition].LeftP ← NextFreePosition
    ENDIF
    NextFreePosition ← ThisTree[NextFreePosition].LeftP
ENDPROCEDURE
```

PROGRAMMING CHALLENGE 11: COUNTRY BINARY TREE

Task 3.2
Write non-recursive code to implement the `AddItemToBinaryTree` procedure.

The given pseudocode is incomplete as:

- it does not initially test that there is space available for a new item
- it does not assign `NewTreeItem` to the data field of the `ThisTree` array

Add these requirements to your program solution.

Evidence 8:
Your program code for Task 3.2. **[6]**

Task 3.3
Write a procedure `OutputData` which displays the value of `Root`, the value of `NextFreePosition` and the contents of `ThisTree` in index order.

Evidence 9:
Your program code for Task 3.3. **[5]**

Task 3.4
Write a main program to:

- Input new data items and add them to the binary tree by calling procedure `AddItemToBinaryTree`. The input is terminated with value `"XXX"`.
  Do not attempt to validate the input of the country names.
- Your program will then call procedure `OutputData`.

Run the program with the input of the single value `"XXX"`.

Evidence 10:
Screenshot showing the output from running the program in Task 3.4. **[3]**

Task 3.5
Test your program using the following data items input in the order shown:

```
INDIA, NEPAL, MALAYSIA, SINGAPORE, BURMA, CANADA, LATVIA, XXX
```

Evidence 11:
Provide screenshot test evidence for Task 3.5. **[5]**

Further program code is required to carry out an **in-order traversal**.

PROGRAMMING CHALLENGE 11: COUNTRY BINARY TREE

Task 3.6
Write a recursive procedure to carry out an in-order tree traversal.
Include a call to the procedure from your main program.

Evidence 12:
Your program code. **[8]**

Evidence 13:
Produce a screenshot for the Task 3.5 dataset confirming the output of the countries in alphabetical order. **[2]**