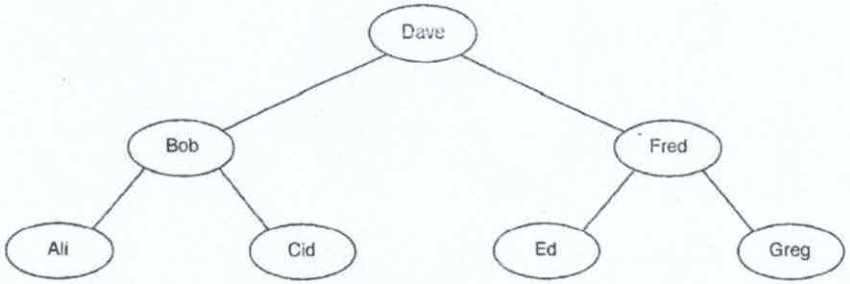
PROGRAMMING CHALLENGE 12: NAME BINARY TREE

A binary tree Abstract Data Type (ADT) has commands to create a new tree, add unique data items to the tree and print the tree.
The sequence of commands:

```
CreateNewTree
AddToTree("Dave")
AddToTree("Fred")
AddToTree("Ed")
AddToTree("Greg")
AddToTree("Bob")
AddToTree("Cid")
AddToTree("Ali")
```

would create the following binary tree:



The program to implement this ADT will use the classes `Tree` and `Node` designed as follows:

Tree
tree : ARRAY of Node root : INTEGER
constructor() add(newItem) print()

Node	
data	: STRING
leftPtr	: INTEGER
rightPtr	: INTEGER
constructor()	
setData(s	: STRING)
setLeftPtr(x	: INTEGER)
setRightPtr(y	: INTEGER)
getData()	: STRING
getLeftPtr()	: INTEGER
getRightPtr()	: INTEGER

PROGRAMMING CHALLENGE 12: NAME BINARY TREE

The program code must:

- Create a new tree, which has:
  - no nodes
  - the root set to –1
- Use the root as a pointer to the first node in the tree
- Add a new node to the tree in the appropriate position
- Use the `print()` method to output , for each node, in array order:
  - the data item
  - the left pointer
  - the right pointer.

Task 1
Write program code to define the classes `Tree` and `Node`.

Evidence 1
Your program code. **[30]**

Task 2
The program is to be tested.
Write a sequence of program statements to:
- Create a tree
- Add the data items shown in the original list of ADT commands
- Print the array contents.

Evidence 2
Your program code.
Screenshot of test run. **[3]**

Task 3
A method `inOrderTraversal()` is to be added, which outputs the data stored in the tree in alphabetical order.
Write program code to:
- Implement this method
- Test the program code with the data from Task 3.2.

Evidence 3
Your program code.
Screenshot of test run. **[7]**