DATA STRUCTURE

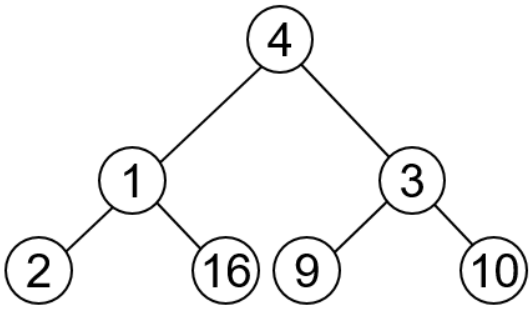
Type	Tree	**LAB PERFORMENCE**
Deadline	**IN CLASS**	
Weighting	TBA	**2**

OBJECTIVES

This assessment item is designed to test your skills on constructing a Tree structure and traversing tree through BFS, DFS and tree traversal techniques like pre-order, post-order, in-order etc.

ASSESSMENT TASK

1. The first part of this lab aims to construct the below tree manually:



Sample node structure for the tree:

```
struct node{
	int item;
	node* left;
	node* right;

	node(int item){
		this->item = item;
		this->left = NULL;
		this->right = NULL;
	}
};
```

2. The second aim of this lab is to traverse the above tree using BFS and/or DFS

ALGORTHM [BFS]:

```
BFS(node){ //node ≈ the root node of a tree/subtree
        Queue.append(node);
        while(!Queue.empty()){
                nodeToExpand = Queue.front()
                Queue.serve()
                Queue.append(child nodes of nodeToExpand)
        }
}
```

Output: 4 1 3 2 16 9 10

ALGORTHM [DFS]:

```
DFS(node){ //node ≈ the root node of a tree/subtree
        Stack.push(node);
        while(!Stack.empty()){
                nodeToExpand = Stack.top()
                Stack.pop()
                Stack.push (child nodes of nodeToExpand)
        }
}
```

Output: 4 3 10 9 1 16 2

3. Now you need to traverse the above tree in **pre-order**, **post-order** and **in-order** traversal techniques:

ALGORTHM [Pre-Order]:

```
preOrder(node){
        do print the item of node
        preOrder(node->left)
        preOrder(node->right)
}
```

Output: 4 1 2 16 3 9 10

ALGORTHM [Post-Order]:

```
postOrder(node){
        postOrder(node->left)
        postOrder(node->right)
        do print the item of node
}
```

Output: 2 16 1 9 10 3 4

ALGORTHM [In-Order]:

```
inOrder(node){
        inOrder(node->left)
        do print the item of node
        inOrder(node->right)
}
```

Output: 2 1 16 4 9 3 10

WHAT & HOW TO SUBMIT

You need to upload through your **VUES** account. You can find the upload link under "*Courses/ DATA STRUCTURE/Lab Performance/*"

SUBMISSION STEPS:

1. Create a Directory/Folder as following format:

   ***<Your ID>*_PERFORMENCE-*< *Performence Number*>**

   Ex: 14-10380-1_PERFORMENCE-1
2. If you update your code then the format should be following:

   ***<Your ID>*_PERFORMENCE-*< *Performence Number*>*_UPDATE-*<Update Number*>**

   Ex: 14-10380-1_PERFORMENCE-1_UPDATE-1
3. Put all the files into that Folder and upload the **zipped** format of that Folder

NOTES

- Your submission will be rejected if uploaded in wrong format
- Only ".zip" file will be accepted.