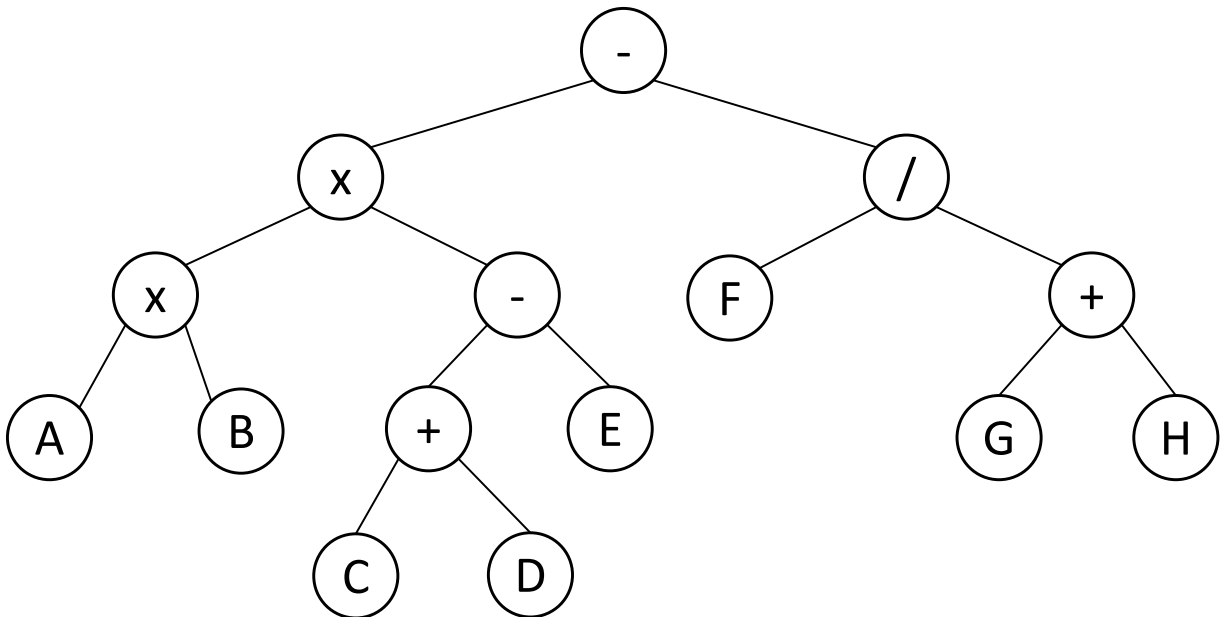


Assignment 2 – Trees

Question 1

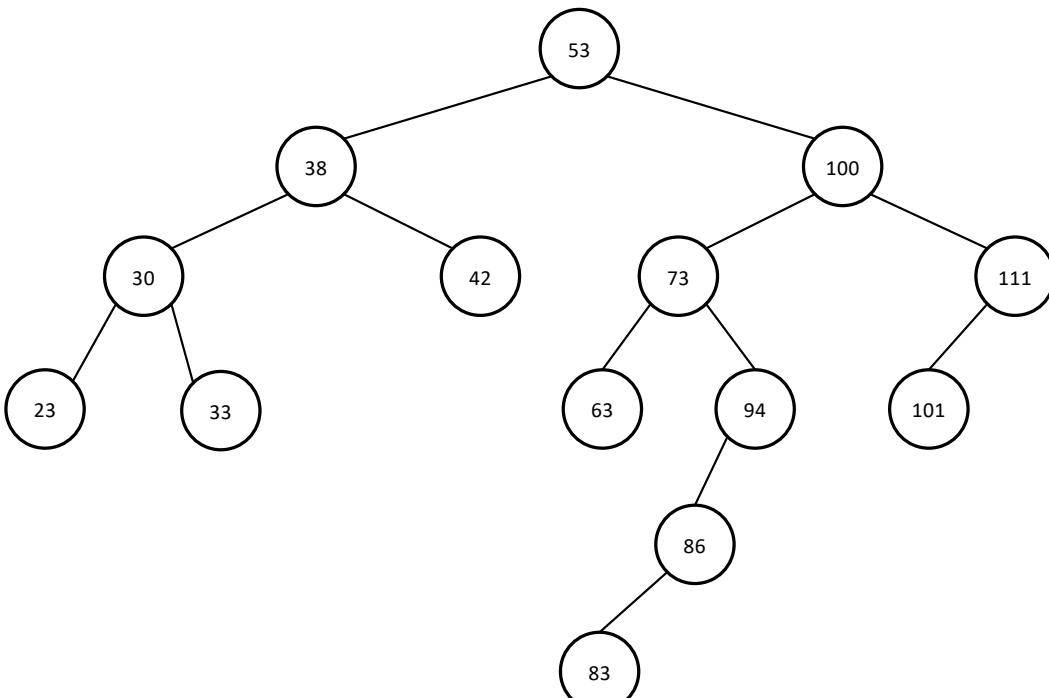
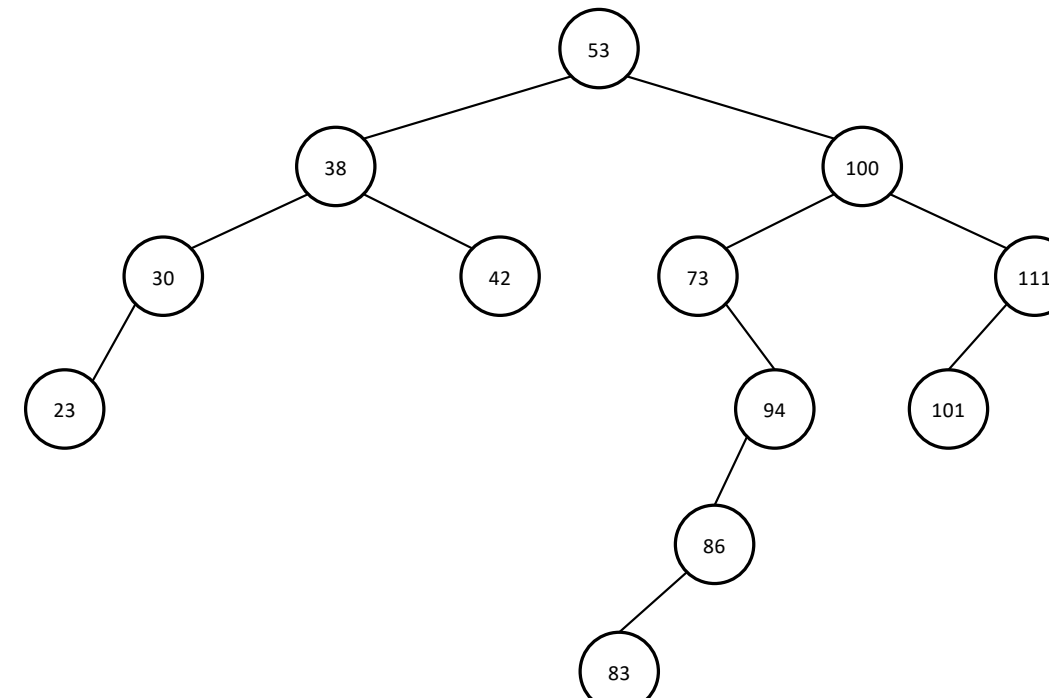
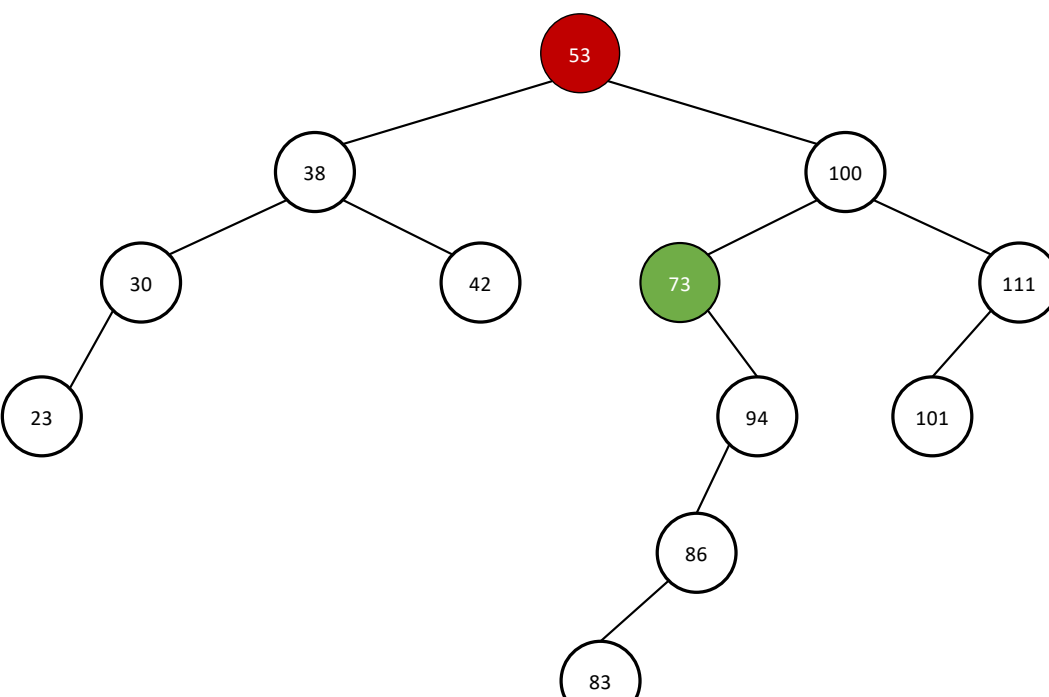
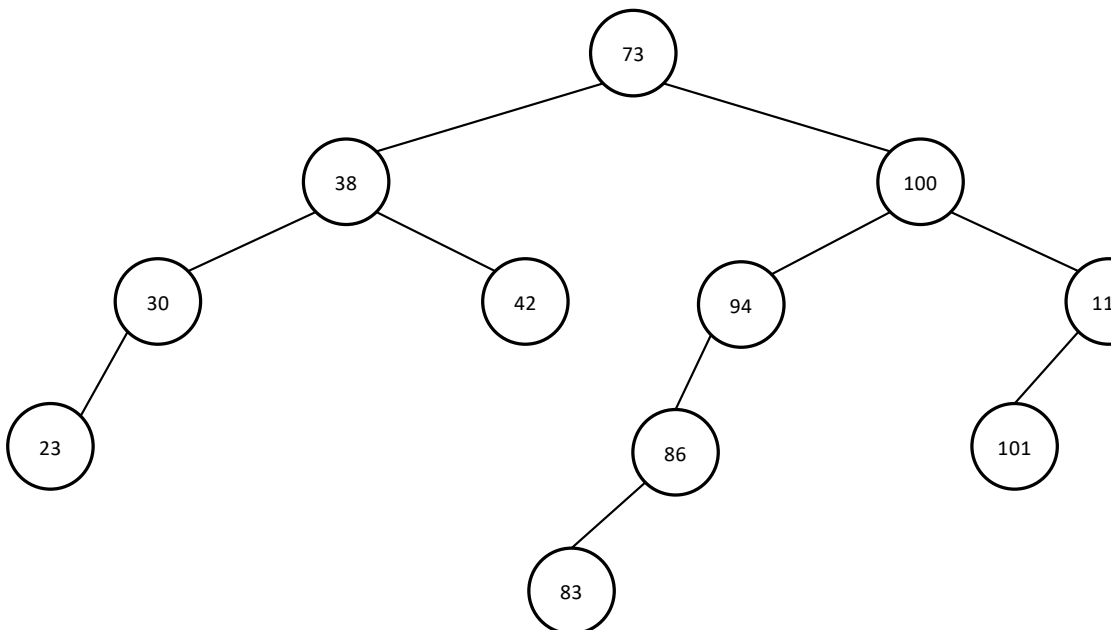


Preorder Traversal of Binary Tree: Prefix Expression													
-	x	x	A	B	-	+	C	D	E	/	F	+	H

Inorder Traversal of Binary Tree: Infix Expression														
A	x	B	x	C	+	D	-	E	-	F	/	G	+	H

Postorder Traversal of Binary Tree: Postfix Expression														
A	B	x	C	D	+	E	-	x	F	G	H	+	/	-

Question 2

After All Insertions	After Deleting 33 and 63 (leaf nodes)
	
Deleting 53 (node with two children), 73 is the inorder successor	The BST after all operations
	

Question 4

```
1 ostream &operator<<( ostream& out, const NgramTree& tree) {
2     tree.outHelper(out, tree.root);
3     return out;
4 }
5
6 void NgramTree::outHelper(ostream& out, TreeNode* treePtr) const {
7     if (treePtr != NULL) {
8         outHelper(out, treePtr->leftChildPtr);
9         out << "\"" << treePtr->ngram << "\"" << " appears " << treePtr->count << " time(s);\n";
10        outHelper(out, treePtr->rightChildPtr);
11    }
12 }
```

The code for the `operator<<` method and its helper method `outHelper` can be found above. The worst case for the `operator<<` method is also the best and average case, since for any given n -node tree, it has to traverse all n nodes (inorder) and print information about the given node. For that reason, the worst-case complexity of `operator<<` is $O(n)$.

```
1 void NgramTree::addNgram(string ngram) {
2     insertItem(root, ngram, 1);
3 }
4
5 void NgramTree::insertItem(TreeNode*& treePtr, const string& ngram, const int& count)
6     throw(TreeException) {
7     if (treePtr == NULL) {
8         treePtr = new TreeNode(ngram, count);
9     }
10    if (treePtr == NULL)
11        throw TreeException("NgramTreeException: Insertion Failed.");
12 }
```

```
11  else if (ngram < treePtr->ngram) {  
12      insertItem(treePtr->leftChildPtr, ngram, count);  
13  }  
14  else if (ngram > treePtr->ngram) {  
15      insertItem(treePtr->rightChildPtr, ngram, count);  
16  }  
17  else {  
18      treePtr->count += count;  
19  }  
20 }
```

The code for the `addNgram` method and its helper method `insertItem` can be found above. The worst case for the `addNgram` method is when an n -node tree has height $h = n$. In this case, the function has to traverse all nodes to add the required `ngram` to the node at height h . Hence, the worst-case time complexity of `addNgram` is $O(n)$.