Problem 1 (20 points)

Implement a binary search tree

Create an ADT for a Node

Create an ADT for a BST with the following functions:

- Add Node(Value)
- Delete Node(Value)
- FindNode(Value)
- PrintTree()

Demonstrate you code by randomly generating an input set of size 5 to 50 for numbers between 1 and 1000

You must print out your input set, your initial tree and then exercise your methods add and delete printing your tree after every method invocation. You must also exercise your findNode method by randomly generating a number between 1 and 1000 and printing whether or not you found the node, you must have both positive and negative cases.

You should submit a readme.txt file with an explanation of your code and algorithms. You must provide exact instructions on how to run your code and you must submit screen shots of your running code.

Problem 2 (30 points) Implement from Class Discussion

Given is a sequence of n symbols, each of which is either a dot (.) or a dash (-). This can represent a sequence of letters in Morse code. However, since the separation between letters is not given, it can represent a number of different sequences. For example, . - - could represent ETT, AT, EM, or W.

Design an algorithm that computes the number of possible letter sequences containing only vowels (A,E,I,O,U) that can be derived from a given input sequence of dots and dashes of length n.

You should submit a readme.txt file with an explanation of your code and algorithms. You must provide exact instructions on how to run your code and you must submit screen shots of your running code. Your code should print the following text:

File Input: vowel_input<input file number>.txt

The Number of Vowel combinations is: <the number of combinations you calculate>

Problem 2 (30 points) con't

File Input: vowel_input1.txt

The Number of Vowel combinations is: 6

File Input: vowel_input2.txt

The Number of Vowel combinations is: 8

File Input: vowel_input3.txt

The Number of Vowel combinations is: 0

File Input: vowel_input4.txt

The Number of Vowel combinations is: 193050000

Problem 3 (50 points)

You are given two sequences of integers: A = $\{a_1, a_2, \ldots a_i\}$ and B = $\{b_1, b_2, \ldots b_j\}$.

Define an algorithm that returns the longest sequence of alternating increasing values from A and B where the sequence X = $\{x_1, x_2, \ldots x_n\}$ is as follows:

- The elements of X are increasing e.g. $x_i < x_{i+1}$ for all $1 <= I < n$
- The odd-indexed elements of X are a subsequence of one of the sequences either A or B and the even-indexed elements of X are a subsequence of the other sequence.
- Note a subsequence need not be consecutive elements of the original sequence but must maintain the relative order of the original sequence

For example A = {1,7,2} and B = {4,8,3,9} then the answer is X = {4,7,9} which has a length of 3.

Problem 3 (50 points)

Demonstrate you code by running your code with the provided input files. The input file will contain 4 input lines. The first line will be the size of the A array, the second will be the size of the B array the third line will be the A array values separated by a space and the fourth line will be the B array values separated by a space.

Sample input file:

5

6

8 2 9 1 4

4 1 5 3 2 6

You should submit a readme.txt file with an explanation of your code and algorithms. You must provide exact instructions on how to run your code and you must submit screen shots of your running code. The output should be of the format:

File Input: longest_seq<input file number>.txt
Longest Sequence: <the longest sequence you computed>

Problem 3 (50 points)

File Input: longest_seq1.txt

Longest Sequence: 4

File Input: longest_seq2.txt

Longest Sequence: 7

File Input: longest_seq3.txt

Longest Sequence: 10

File Input: longest_seq4.txt

Longest Sequence: 39

File Input: longest_seq5.txt

Longest Sequence: 192

File Input: longest_seq6.txt

Longest Sequence: 1