Problem Set #9 - Due 04/16/03 Total 100

The purpose of this problem set is to:

- Help you become familiar with manipulating binary trees and finite state machines.

Please turn in each problem on a separate page. Each page should have your Name, email id, and the problem number clearly printed/written on it. Keep track of how long time it takes to complete each problem. The time taken for each problem should be printed on the first page. If you use more than one page for one problem, please STAPLE the pages together. You will lose points if you do not document the time taken for each problem, which at the same time means that you will get points for documenting "time taken" A template (in PDF form) is available on the web.

Problem 1 - 60 points

Part a. Write pseudo-code to

- Create a tree
- Insert a node into a tree.
- Traverse the tree
  - Inorder
  - Preorder
  - Postorder

The insertion algorithm works as follows

- If the tree is empty, create a new root node.
- If the tree is not empty then
  - if the element has value less than that of the root, insert into the left subtree
  - else, insert into the right subtree.

Part b. Implement the pseudo-code above in Ada95 i.e. create an .ads and .adb file to implement the pseudo-code you wrote.

Part c. Write a test program to accept the an input string, create the tree using the input, and display the result of traversing the tree

- Inorder
- Preorder
- Postorder

Hint: The node for the tree is a record with the structure shown below:

```
Type Node is Record

        Element : Character
        Left_Child: NodePtr
        Right_Child : NodePtr
End Record
```

Assume that the string contains only single character elements.

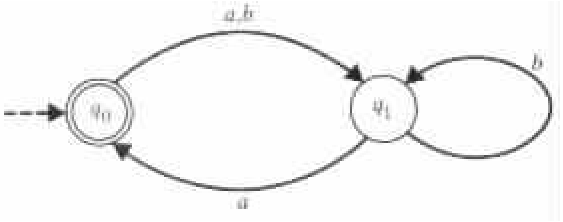
Turn in a **hard copy** of the code listing for **part b and part c,** and turn in **your code** for **part b and part c electronically**. Feel free to reuse any of the code you have written / received so far. If you are reusing material, make a note of it in the header of your program ☺.

Problem 2 - 30 points

Part a. What is a Finite State Machine?

Part b. Draw a finite state machine that can accept all strings generated from an alphabet {0,1}, with the substring 0010.

Part c. What is the language accepted by the finite state machine shown below?



Problem 3 - 10 points

Part a. What did you learn while working on the project this week? Summarize in 3 lines.

Part b. Have you split the work on the project? Explain your portion of the project.

Part c. Is there anything else you would like to add?