$\boxed{\text{Burt Rosenberg}}$

Problem Set 6

OUT: 9 NOVEMBER 1992
DUE: 18 NOVEMBER 1992

Reading Assignment

Read:

- Finish reading Chapter 10, *Tables and Trees.*
- Read Chapter 8, *Stacks*, and Chapter 9, *Queues.*

Goals

Practice programming with trees. Improve our style of user interface.

Assignment

Add deletion to the program you wrote for the previous assignment. The user interface will be as follows. The program accepts the commands,

- **Insert** — places the program in insert mode.
- **Delete** — places the program in delete mode.
- **Pretty** — pretty-prints the tree.
- **List** — lists the tree elements in order.
- **Quit** — exits the program.
- n, where n is any integer — if in insert mode, the program inserts the number n in the tree. If in delete mode, the program checks to see if n is in the tree and deletes it if it is. Else a message is printed informing the user that n cannot be deleted.

Initially the program starts in insert mode. If an input is not one of these six forms, you can simply ignore it. You can also assume that only the first letter of the command is important. That is, instead of "Delete" the user could just as well type "Dinosaur".

Useful programming techniques for accomplishing this user interface are illustrated in the program below.

The input is read first as a string s. The first character is checked using the pascal *case* construct to separate into cases. If the first character is a digit, the sign $+$ or the sign $-$, the string is reread as an integer using *readv*. The other entries in the case statement separate individual commands.

Define the flag *insert_mode* and initially set it true. The case statement entry for the delete command sets this flag false. Likewise, the case statement entry for the insert command sets it true. The entry for a digit not only converts the input to an integer but references this flag to find out what to do with the number.

Extra Credit

Implement the command *file* which prompts the user for a file name and then accepts input from the file, using the same format as described above, until a quit command is encountered. Then the program returns to the user for interactive input. To make things simple, you can assume that the file command is not active during file input.

So far we have used *external files*, whose names are fixed at compile time, appearing in the program header next to *input* and *output*. The file command requires you to use an *internal file* whose name is chosen at run time. Such files do not appear in the program header and need be opened by you before reading. The program below shows how the *open* statement accomplishes this.

The most important reason for having input from files is to allow thorough testing of your programs. File input means that you can record exactly how your program was tested and the test can be verified by others. Also, other programs can build test files and challenge your program to run them.

If you implement the file command, you should also implement *comments*. If "!" is the first character of a line, or if the line is empty, the line is discarded by the program. The only use of such lines is to document the file.

For extra credit, write your test input in a file, document it with comments, and submit it with your programming rsolution.

Sample Program

```
{}
{ A program to demonstrate how to }
{ reread the input under a different}
{ format.}
{}
{ Prof. Rosenberg}
{ Univ. Miami}
{ 6 November, 1992}
{}

program inp(input,output) ;

type
  string = varying [50] of char ;

function FirstChar( s: string ) : char ;
var
  i : integer ;
begin
  if s.length=0 then FirstChar:=chr(0)
  else
  begin
    i := 1 ;
    while (s.body[i]=' ') and (i<s.length) do i := i + 1 ;
    FirstChar := s.body[i]
  end
end ;

var
  f : text ;
  s : string ;
  c : char ;
  i : integer ;
  insert_mode : boolean ;
```

```
begin
  insert_mode := true ;
  writeln('Welcome to the test!') ;
  c := ' ' ;
  repeat
    case c of
      '0'..'9','+','-' :  {number command, insert or delete i}
         begin
           readv(s,i) ;
           writeln('The number you entered is: ',i)
         end ;
      'i','I' : {insert command, change to insert mode}
         insert_mode := true ;
      'd','D' : {delete command, change to delete mode}
         insert_mode := false ;
      'f','F' : {file command, get an input file}
         begin
           write('  filename>') ;
           readln(s) ;
           open( f, s, history:=old) ;
           reset(f) ;
           readln(f,s) ;
           writeln('From the file I read: ',s) ;
           close(f)
         end ;
      '!','#' ,' ': {comment, disregard line} ;
      'q','Q' :{exit command, never really gets here.} ;
      otherwise writeln('What kindda lousy input is that?')
    end ;
    write('What do you want?>') ;
    readln(s) ;
    c := FirstChar(s)
  until (c='q') or (c='Q') ;
  writeln('Bye-bye')
end .
```