CS 1723, Data Structures
Fall Semester, 1998
Programming Assignment 8

Concordance Program
Due November 11, 1998

The Assignment: On pages 139-143 (Section 6.5), your text gives the C code for a program which will print each word in an input text along with the number of times the word occurred. Using the program source itself as input, the book's program will produce output that starts as follows:

```
2 EOF
3 MAXWORD
5 NULL
3 a
1 add
6 addtree
. . .
```

For this assignment you are to modify the text's program so that it produces a concordance: a list of each word in the input text, and for each word a list of the line numbers where the word occurred in the text. (This is essentially just Exercise 6-3 on page 143.) Using the same program as source, your modified program should produce output like the following. (Much of it is deleted below.)

```
         EOF:  27  77
     MAXWORD:   6  25  27
        NULL:  26  39  43  57  97
           a:  35  64  92
         add:  35
     addtree:  15  29  35  36  48  50
          at:  35
       below:  35
       break:  86
     builtin:  92
           c:  73  75  77  78  79  81
        char:   9  15  18  19  25  36  71  74  93  95  96

  ...  ...  (a number of lines deleted)

       order:  54
           p:  35  36  39  40  41  42  43  45  46  48  50  51  54  55
               57  58  59  60  95  96  97  98  99
       print:  54
        ...  ...  (a number of lines deleted)
```

Details:

1. Relatively simple changes are needed to the source supplied in other handouts.
2. You should return the current line number from the `getword()` function as another parameter. You can keep track of the current line number by starting a static variable at 1 and incrementing it each time you see a `'\n'` on input.
3. Inside the main struct for the tree node (`tnode`) you must use a pointer to a linked list of integers, which will hold the line numbers. This will require creating another struct, and even writing another allocation function.
4. The `addtree()` function will have another parameter, the line number where the word occurred. `addtree()` will insert this line number into the linked list associated with the node for the word.
5. You should fix up the output so that for a given word it only prints a given line number once, as shown with the entry for `p` above.
6. You should have your program skip to a new line after printing 14 numbers (or so), again as shown with the entry for `p` above.
7. Print the line numbers in increasing order. There are several ways to arrange this.
8. Inside the directory `"~wagner/pub/CS1723"` is the text of the assignment as `"assign8.text"`, the source file as `"tree.c"`, and a non-recursive version of the source as `"treealt.c"`. (The file `"treeprint.c"` is a similar version with single chars at nodes.)