```
vip% cat tree.c
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#define MAXWORD 100

struct tnode {
   char *word;
   int count;
   struct tnode *left;
   struct tnode *right;
};

struct tnode *addtree(struct tnode *, char *);
void treeprint(struct tnode *);
struct tnode *talloc(void);
int getword(char *, int);
char *strdupl(char *);

/* word frequency count */
void main(void)
{
   struct tnode *root;
   char word[MAXWORD];
   root = NULL;
   while (getword(word, MAXWORD) != EOF)
      if (isalpha(word[0]))
         root = addtree(root, word);
   treeprint(root);
   exit(0);
}

/* addtree: add a node with w, at or below p */
struct tnode *addtree(struct tnode *p, char *w)
{
   int cond;
   if (p == NULL) {
      p = talloc();
      p -> word = strdupl(w);
      p -> count = 1;
      p -> left = p -> right = NULL;
   }
   else if ((cond = strcmp(w, p -> word)) == 0)
      (p -> count)++;
   else if (cond < 0)
      p -> left = addtree(p -> left, w);
   else
      p -> right = addtree(p -> right, w);
   return p;
}

/* treeprint: in-order print of tree p */
void treeprint(struct tnode *p)
{
   if (p != NULL) {
      treeprint(p -> left);
      printf("%4d %s\n", p -> count, p -> word);
      treeprint(p -> right);
   }
}

/* talloc: make a tnode */
struct tnode *talloc(void)
{
   return (struct tnode *)
      malloc(sizeof(struct tnode));
}

/* getword: get next word or charcter from input */
int getword(char *word, int lim)
{
   int c;
   char *w = word;
   while (isspace(c = getchar()))
      ;
   if (c != EOF)
      *w++ = c;
   if (!isalpha(c)) {
      *w = '\0';
      return c;
   }
   for ( ; --lim > 0; w++)
      if (!isalpha(*w = getchar())) {
         ungetc(*w, stdin);
         break;
      }
   *w = '\0';
   return word[0];
}

/* strdupl: make duplicate of s.(strdup builtin) */
char *strdupl(char *s)
{
   char *p;
   p = (char *) malloc(strlen(s)+1);
   if (p != NULL)
      strcpy(p, s);
   return p;
}
```

```
vip% lint -m -u tree.c

function returns value which is always ignored
    strcpy         printf         ungetc
vip% cc -o tree tree.c
vip% tree <tree.c
   2 EOF
   3 MAXWORD
   5 NULL
   3 a
   1 add
   6 addtree
   1 at
   1 below
   1 break
   1 builtin
   6 c
  13 char
   1 charcter
   3 cond
   5 count
   1 ctype
   1 d
   1 define
   1 duplicate
   3 else
   1 exit
   1 for
   1 frequency
   1 from
   1 get
   2 getchar
   4 getword
   4 h
   9 if
   1 in
   4 include
   1 input
   7 int
   1 is
   3 isalpha
   1 isspace
   5 left
   2 lim
   1 main
   2 make
   2 malloc
   1 n
   1 next
   1 node
   2 of
   2 or
   1 order
  27 p
   1 print
   1 printf
   5 return
   5 right
   5 root
   5 s
   1 sizeof
   1 stdin
   1 stdio
   1 stdlib
   1 strcmp
   1 strcpy
   1 strdup
   4 strdupl
   1 string
   1 strlen
  14 struct
   4 talloc
  15 tnode
   1 tree
   6 treeprint
   1 ungetc
   6 void
  13 w
   2 while
   1 with
  13 word
```