```
// Stephen M. Jones // 05/22/2024

// Coda-C example #2:  Dictionary and Char objects.

// Task:  Report the number of times each word from <stdin> is used.

staticCharget_Word(FILE*is) {
    charbuf[128];intpos=0;
    while(1) { intcc=tolower(fgetc(is));if(cc==EOF)break;
        if(cc>='a'&& cc<='z') { buf[pos++]=cc; if(pos>=100)break; }
        else if (pos)break;
        }
    buf[pos]=0;return(pos ? Char_Value(buf) : 0);
    }

static void codax02() {
    cleanO Dictionary dict=newO(Dictionary);
    while(1) {
        cleanO Char word=get_Word(stdin);if(!word)break;
        Shortcount=Dictionary_subKey(dict,word);
        if(!count) Dictionary_setKey(dict,word,Short_Value(1));else++(*count);
        }
    for(Keywordkey=Dictionary_scan(dict);key;key=Dictionary_next(key)) {
        Shortcount=key->item; printf("%s=%d\n",key->word,*count);
        }
    }

codax_register(codax02)

// Purpose:  To demonstrate how a Dictionary can simplify C programming.
```

```
// Coda-C adds the following:

// Dictionary          - C data type
// Dictionary_setKey() - add a key to a Dictionary
// Dictionary_subKey() - get a key's value in a Dictionary

// Keyword             - C data type
// Dictionary_scan()   - get the first internal element of a Dictionary
// Dictionary_next()   - get next internal element of a Dictionary

// Char                - C data type
// Char_Value()        - create a string object

/*<stdin> test input
All work and no play ...
All work and no play ...
All work and no play ...
All work and no play ...
Redrum.
**********************/

/*<stdout> example's output
redrum=1
play=4
no=4
and=4
work=4
all=4
**********************/
```