```c
// Yannique Hecht
// HARVARD CS50 Week 5 - Speller - Implement a program that fastly
 spell-checks files with a hash table

// Include libabries to ensure dictionary functions properly
#include <stdbool.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#include <strings.h>
#include <stdio.h>
#include "dictionary.h"

// Represent a node in a hash table
[redacted]

// Number of buckets in hash table (originally = 1)
[redacted]

// Hash table
[redacted]

// return true if word is in dictionary, else false
[redacted]

    // compare words case insensitive
    [redacted]

    // keep traversing linked list until it finds the word or it
 finishes
    [redacted]
```

```
            [redacted]

    [redacted]

// Hashes word to a number
// Used one bucket for every letter, a = 0, b = 1 ... z = 25
[redacted]

// Loads dictionary into memory, returning true if successful else
 false
[redacted]
    // open dictionary and initializes temporary space to hold the
 words
    [redacted]

    // read file until the end
    [redacted]
        // allocate memory for a node in which the word will be
 inserted
        [redacted]

        // copies the word in the chunk of memory allocated and then
 updates the words count
        [redacted]

        // set next to point at beginning of list
        [redacted]
        // point array at n which becomes new beginning of the list
        [redacted]

    [redacted]

// Return number of words in dictionary if loaded, else 0 if not yet
 loaded
[redacted]

// Unload dictionary from memory, return true if successful, else
 [redacted]
    // create two pointers to traverse the linked list and cancel its
 element without losing its address
    [redacted]

    // repeat for every index in the table
    [redacted]

        [redacted]

        // free allocated memory in load until the end of list
        [redacted]
```

[redacted]
[redacted]