University of Massachusetts Dartmouth
Department of Electrical and Computer Engineering

ECE 161 Due: see `http://ece161.viall.org`
Lab 2 Name: `isword.cpp`

The objective of this lab is to write a function `isword()` which, given a `char` array of letters will return a non-zero value if the sequence of letters is a word. If the sequence of letters is not a word, a value of zero should be returned. You also need to write other functions as needed and the main program to test the `isword()` routine such the program performs like the sample output below.

SAMPLE RUN

```
Word lookup 1.0
Enter a token: cat
cat is a word.
Enter a token: DOG
DOG is a word.
Enter a token: DoGgiE
DoGGie is a word.
Enter a token: viall
viall is not a word.
Enter a token: *
Exiting...
```

THE WORD LIST

The word list is located at `M:\ECE-161\public\sow-twl.txt`.
The file is `sow-twl.txt`. Download the file to a place where it will be useful to your program. The following program will read the wordlist into an array. Note the array is a global array. We will discuss tradeoffs of using a global array in class. You may not use any other global variables in your program. In your actual lab, you should put this functionality into a module (function) of its own.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char w[205000][14];    // up to 205,000 words, each with up to 12 letters (plus a null)
                       // note this is VERY wasteful.  If the average word length is 6
                       // bytes (5 chars + null), this wastes 7*205000 bytes or 1.4Meg
void main(void)
{
    FILE *wordfile;    // file handle
    char tmp[100];     // temp word
    int i;             // index
    int numwords;      // number of words
    wordfile = fopen("sow-twl.txt","rt"); // open file - returns null if file not found
    if (!wordfile) { printf("File not found...\nexiting..."); exit(0); }

    for (i=0;i<7;i++)  // skip over the first seven lines of file
        fgets(tmp,100,wordfile);

    i=0;               // loop to read words from file
    while (!feof(wordfile))
    {
        fgets(w[i],13,wordfile);  // get next word
        w[i][strlen(w[i])-1]='\0';// replace \n with \0
    //  printf("%s ",w[i]);
        i++;                      // increment index
    }
    numwords=i-1;                 // i is one greater than number of words read
    printf("%d words read\nFirst word: %s\nLast word: %s\n",numwords,w[0],w[numwords-1]);
    fclose(wordfile);             // tidy up
}
```

WHAT IS A "TOKEN"

A "token" as used in this program is a group of alphabetic characters that may or may not be a word.

THE "`isword()`" FUNCTION

You need to write an `isword()` function. The prototype should be:

```
int isword(char word[]);
```

The function should return a true (non-zero) value if the word[] appears in the w[] array, otherwise the function should return a zero. For this lab, you may do a brute force search to see if the user-entered token matches any word.

THINGS TO THINK ABOUT

- The word list contains words in both upper and lower case.
- The first seven lines of the file contain information about the file. You need to skip over these line.
- The user may enter tokens in any combination of upper and lower case.
- Your program should consider uppercase and lowercase characters equivalent ('A' equals 'a', 'B' equals 'b', etc)
- The variable `numwords` in the program above either needs to be passed back to the calling function, or you need another way to designate the last word in the wordlist (consider putting ****** as the last word)