```c
//Programma per usare il programma di ricerca in un dizionario

#include <stdio.h>
#include <stdbool.h>


struct entry{           //entry diventa un "contenitore"

    //2 membri / componenti di tipo char    -->     definiscono SOLO come è fatta entry

    char word[15];
    char definition[50];
};


int main (void){

    const struct entry dictionary[100] = {

        { "aardvark", "a burrowing African mammal" },       // { "word",
"definizion" }     -->     questo per via di struct entry
        { "abyss", "a bottomless pit" },
        { "acumen", "mentally sharp; keen" },
        { "addle", "to become confused" },
        { "aerie", "a high nest" },
        { "affix", "to append; attach" },
        { "agar", "a jelly made from seaweed" },
        { "ahoy", "a natural call of greeting" },
        { "aigrette", "an ornamental cluster of feathers" },
        { "ajar", "partially opened" }
    };


    char word[16];          //la parola che sto cercando    -->     inserita dall'utente
    int entries = 10;       //numero di riche nel dictionary    -->     numero di { }   --
>     come gli array parte da 0
    int entry;              //il numero della parola cercata  ( es: word = abyss     --
>     entry = 1 )

    int lookup(const struct entry dictionary[], const char search[], const int
entries);        //Prototipo


    printf("Digita la parola: ");
    scanf("%15s", word);
    entry = lookup(dictionary, word, entries);

    if( entry != -1 ){
        printf("%s\n", dictionary[entry].definition);       //  deifinition = per accedere
alla parte definizion della stringa

    } else{
        printf("La parola %s NON è nel dizionario.\n", word);
    }

    return 0;
}



//Funzione per ricercare una parola in un dizionario

int lookup(const struct entry dictionary[], const char search[], const int entries){

    int i;
    bool equalStrings(const char s1[], const char s2[]);        //Prototipo
```

```
    for( i = 0; i < entries; ++i ){

        if ( equalStrings(search, dictionary[i].word) == true ){        // dictionari[i] = 1 singola {struct}   . word = per accedere alla word della stringa
                                                                        //controlla se la parola inserita ( search ) è uguale alla parola word nelle varie entries[i]

            return i;
        }
    }

    return -1;

}



bool equalStrings(const char s1[], const char s2[]){

    int i = 0;
    bool areEqual;


    while( s1[i] == s2[i] && s1[i] != '\0' && s2[i] !='\0' ){

        ++i;
    }


    if( s1[i] == '\0' && s2[i] == '\0' ){

        areEqual = true;

    } else{
        areEqual = false;
    }

    return areEqual;

}
```