```c
/*Completare il codice dove indicato tramite commenti in modo che implementi una ricerca
binaria di stringhe in un dizionario.
Il programma deve stampare la definizione della stringa se trovata o il messaggio "La
stringa non esiste" in caso contrario.

Alcuni esempi di funzionamento del codice sono i seguenti:

Inserire la chiave da ricercare: cane
La definizione di "cane" è la seguente: "Animale che abbaia spesso."

Inserire la chiave da ricercare: ca
La stringa inserita non esiste nel dizionario

*/


#include <stdio.h>

#define MAX 20
#define MAX_DEF 100

struct entry {
    char word[MAX];
    char definition[MAX_DEF];
};

int lookup_binaria(char key[], struct entry dictionary[], int length);


int main(void) {

    struct entry dictionary[] = {       //un array di tipo struct entry
        {"automobile", "Mezzo di trasporto su gomma."},
        {"cane", "Animale che abbaia spesso."},
        {"cattedra", "Tavolo di lavoro universitario."},
        {"computer", "Strumento dfi calcolo automatico."},
        {"gatto", "Animale peloso e miagolante."},
        {"lavagna", "Supporto per la scrittura a parete."},
        {"studente", "Persona che dovrebbe studiare molto."}
    };

    char key[MAX];
    int posizione;

    printf("Inserire la chiave da ricercare: ");
    scanf("%s", key);


    posizione = lookup_binaria(key, dictionary, 7);     //passo 7 perchè gli elementi word
sono in totale 7
    if(posizione >= 0){
        printf("La definizione di \"%s\" è la seguente: \"%s\"\n", key,
dictionary[posizione].definition);
    }
    else
        printf("La stringa inserita non esiste nel dizionario\n");
  return 0;
}


int compare_strings(char s1[], char s2[]){
    int i=0;
    while (s1[i] == s2[i] && s1[i] != '\0'){
        i++;
    }
```

```
    return s1[i] - s2[i]; //se <0, s1!= s2 e s1 viene prima di s2
    //se >0 s1!=s2 e s1 viene dopo di s2
    //se ==0 s1==s2
};



int lookup_binaria(char key[], struct entry dictionary[], int length){      //lenght =
numero di elementi word che posso cercare nel dizionario
    /*
    completare
    */

    int min_i = 0;
    int max_i = length;
    int compare;

    int mid_i;       //mi identifica la riga di riferimento dentro dictionary    -->
word + definitio

    do{
        mid_i = (min_i + max_i) / 2;

        printf("Min:%i \t Mid:%i \t Max:%i\n", min_i, mid_i, max_i );
        compare = compare_strings( key, dictionary[mid_i].word );

        if( compare == 0 ){
            return mid_i;

        }

        if( compare < 0 ){
            max_i = mid_i;

        }

        if( compare > 0 ){
            min_i = mid_i;

        }

    } while( (min_i + max_i) / 2 != mid_i );

    return -1;           //ritorno ERRORE    ??

};
```