```c
/*
 * -- M.G. Maffucci --
 * 2. Dato un elenco di nomi, controlla se un nome e' compreso
 * nell'elenco.
 *
 * NOTA: per fare l'input delle stringhe ho dovuto utilizzare
 * la funzione fgets() perche' la funzione gets() viene
 * considerata non sicura e, dato che il compilatore
 * che uso su Linux e' piu' esigente rispetto a quello che usiamo
 * a scuola, mi segnala
 * un errore. Avete comunque tutti gli elementi per capirne
 * il funzionamento, considerando che fgets() legge tutto
 * quello che scrivete, incluso l'invio finale che dovra' essere
 * eliminato. Per maggiorni dettagli andate a leggere
 * http://blacklight.gotdns.org/guidac.pdf a pag. 84.
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define NMAX 30     //numero massimo di nomi
#define LMAX 20     //lunghezza massima di un nome

int main()
{
    //input
    char nomi[NMAX][LMAX];
    char nome[LMAX];
    char risposta;  //permette l'inserimento dei nomi, puo'
                    //valere s/n
    //lavoro
    int i;          //indice del vettore dei nomi
    int n;          //numero di nomi inseriti nell'elenco
    int controllo;  //serve per verificare la presenza del
                    // nome nell'elnco
    int daButtare;  //variabile usata per ripulire lo stdin

    /*
     * Caricamento dei nomi nell'elenco fino a quando l'utente
     * lo desidera o, al limite, fino al completamento delle
     * posizioni nel vettore.
     * Osserva che la variabile n conterra' il numero di nomi
     * caricati.
     */
    n = 0;
    do {
        printf("Inserisci il nome nell'elenco: ");
        scanf("%s", nomi[n]);
        printf("Vuoi inserire un altro nome? (s/n): ");
        /*
         * Per evitare di utilizzare la fflush(), che non
         * risulta piu' fra le funzioni del C consigliate e,
         * dato che sul S.O. Linux ha un comportamento
         * imprevedibile, e' possibile aggiungere uno spazio
         * prima della specifica di conversione nella scanf(), in
         * modo tale che vengano ignorati eventuali invii
         * che non permettano il corretto inserimento dell'input.
         */
        scanf(" %c", &risposta);
        n++;
    } while( (n <= NMAX) && (risposta == 's') );
    /*
     * Questo while vuoto (notate il ; subito dopo) mi permette
     * di ripulire lo stdin senza usare la fflush() che non e'
     * piu' consigliato utilizzare dall'ultima versione del C.
     * Quella proposta e' una delle possibili soluzioni: la
     * funzione getchar() legge un carattere per volta dallo
```

```
68          * stdin, memorizzandolo nella variabile daButtare,
69          * fino a quando non trova l'invio che ne provoca la
70          * conclusione.
71          */
72         while( (daButtare = getchar()) != '\n');
73         /*
74          * Caricamento del nome da cercare nell'elenco.
75          * Provo ad usare fgets(); era possibile usarla anche
76          * durante l'input dell'elenco di nomi.
77          */
78         printf("Inserisci il nome da cercare nell'elenco: ");
79         fgets(nome, LMAX, stdin);
80         nome[(strlen(nome) - 1)] = '\0';
81         /*
82          * Imposto il ciclo di ricerca del nome nell'elenco.
83          */
84         controllo = 0;
85         for(i = 0; i < n; i++){
86             if( !strcmp(nomi[i], nome) ){
87                 printf("Il nome %s e' presente nell'elenco alla posizione %d\n"
    , nome, (i + 1));
88                 controllo = 1;
89             }
90         }
91         /*
92          * Segnalo l'eventuale assenza del nome nell'elenco
93          */
94         if(!controllo)
95             printf("Il nome %s non e' presente nell'elenco", nome);
96         return 0;
97     }
98
```