www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

```
/*
 * Corso di Fondamenti di Informatica
 * Esercizio:
 * esempio di uso di liste sequenziali
 */

/* Scrivere un programma C che:
 *
 **1. Definisce un array di caratteri Frase, di 60 elementi; in Frase va
 * inserita, a partire dal primo carattere, una frase a piacere di lunghezza
 * minore o uguale della capienza massima di Frase.
 * La frase deve contenere da 1 a 10 parole, separate da (singoli) spazi, e
 * va terminata con il carattere '.' (punto) non preceduto da spazio. Il
 * primo carattere non può essere uno spazio. Ogni parola della frase può
 * avere una lunghezza compresa tra 1 e 15 caratteri. Se necessario, gli spazi
 * possono essere sostituiti da un opportuno carattere non alfabetico.
 **2. Legge da tastiera una frase (che soddisfa le condizioni sopra indicate) e
 * la inserisce in Frase.
 **3. Crea una LISTA SEQUENZIALE, ElencoParole, i cui elementi vanno usati per
 * descrivere le singole parole presenti in Frase. Ciascuno di tali elementi
 * comprende: (1) un array di caratteri che serve a memorizzare la parola,
 * TERMINATA DAL CARATTERE '\0' (carattere di fine stringa); (2) la lunghezza
 * della parola (numero delle lettere che la compongono, escluso il '\0').
 **4. Esamina Frase e memorizza in ElencoParole i dati sulle singole parole
 * della frase ivi contenuta.
 **5. Stampa su righe separate le parole di ElencoParole, con a fianco le
 * relative lunghezze.
 *
 * Nota: un modo molto semplice per leggere da tastiera [stampare] una stringa
 * terminata con '\0' consiste nell'utilizzare scanf [printf] con la stringa di
 * controllo "%s". Nel caso di scanf, il '\0' viene inserito automaticamente in
 * coda ai caratteri inseriti da tastiera.
 * In realtà, per ragioni di sicurezza che vedremo in seguito nel corso, l'uso
 * di scanf per leggere stringhe è DA EVITARE: utilizzare invece la funzione
 * fgets. Inoltre la stringa non deve contenere spazi, che scanf interpreta
 * (al pari del carattere '\n' inserito dalla pressione del tasto <enter>) come
 * segnale che l'input e' terminato.
 */


#include <stdio.h>
```

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

```
#define LUNG_MAX_FRASE 60
#define LUNG_MAX_PAROLA 15
#define NUM_MAX_PAROLE 10

#define TERMINATORE_FRASE '.'

#define SEPARATORE_PAROLE '_'
/* Nota: SEPARATORE_PAROLE non puo' essere ' ' o '\n' perche' entrambi questi
 * caratteri sono interpretati da scanf("%s", ...) come segnale di fine
 * input, e non puo' essere TERMINATORE_FRASE perche' gia' usato per
 * terminare la frase. */

int main()
{
  char Frase[LUNG_MAX_FRASE];
  /* Nota: poiche' useremo scanf per inserire i caratteri in Frase, occorre
   * tenere conto che un elemento di Frase sara' utilizzato dal carattere '\0'
   * che scanf inserisce automaticamente quando l'inserimento termina. Quindi i
   * caratteri di Frase utili per contenere la frase sono 59. Non essendo
   * possibile imporre a scanf un limite sul numero di caratteri inseriti, tale
   * limite sarà gestito comunicandone l'esistenza all'utente tramite la stampa
   * di un apposito messaggio. */

  typedef struct
  {
    char Parola[LUNG_MAX_PAROLA+1];
    /* contiene una parola della frase da analizzare, terminata da '\0' */
    int Lunghezza;
    /* numero di lettere che compongono la parola */
  } TipoDescrParola;
  /* usato per rappresentare una singola parola */

  struct
  {
    TipoDescrParola DatiParole[NUM_MAX_PAROLE];
    /* contiene le informazioni sulle parole contenute nella frase da
     * analizzare */
    int NumElementi;
    /* numero elementi dell'array effettivamente contenenti dati */
  } ElencoParole;
```

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

```
  /* tipo di dato della lista sequenziale usata per contenere le parole */

  int IndiceCorrFrase;
  /* indice dell'elemento di Frase attualmente in esame durante la ricerca di
   * parole */
  int IndiceCorrParola;
  /* indice dell'elemento del campo Parola dell'ultimo elemento di ElencoParole
   * nel quale va scritto il carattere corrente letto da Frase */
  int IndiceCorrElemento;
  /* indice dell'elemento di ElencoParole attualmente in esame in fase di
   * stampa; variabile non indispensabile ma introdotta per chiarezza */

  printf("\nInserisci una frase terminata dal carattere '%c'. \nLa frase, '%c' incluso, puo' comprendere al massimo %d
caratteri. \nLe parole della frase devono essere separate da '%c'.\n", TERMINATORE_FRASE, TERMINATORE_FRASE, LUNG_MAX_FRASE-
1, SEPARATORE_PAROLE);
  scanf("%s", Frase);

  /* printf("\nDEBUG: la frase inserita e' '%s'.\n", Frase); */
  /* l'istruzione precedente e' stata usata in fase di debugging */

  /*** riempimento della lista sequenziale: ***/

  ElencoParole.NumElementi = 0;
  /* svuota la lista sequenziale */

  IndiceCorrFrase = 0;
  IndiceCorrParola = 0;

  while ( Frase[IndiceCorrFrase] != TERMINATORE_FRASE )
  /* finché ci sono ancora caratteri da esaminare */
  {
    if ( SEPARATORE_PAROLE != Frase[IndiceCorrFrase] )
    /* Se il carattere corrente di Frase fa parte della parola
     * correntemente in esame, aggiunge il carattere all'ultimo
     * elemento di ElencoParole */
    {
      ElencoParole.DatiParole[ElencoParole.NumElementi].Parola
        [IndiceCorrParola]  = Frase[IndiceCorrFrase];
      ++IndiceCorrParola;
    }
    else
```

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

```
  /* il carattere corrente è un separatore di parole, dunque separa l'ultima
   * parola rilevata dalla successiva */
  {
    /*** termina la memorizzazione della parola corrente e prepara
     *** quella della parola successiva: ***/

     ElencoParole.DatiParole[ElencoParole.NumElementi].Parola
       [IndiceCorrParola] = '\0';

     ElencoParole.DatiParole[ElencoParole.NumElementi].Lunghezza =
       IndiceCorrParola;

     IndiceCorrParola = 0;

     ++ElencoParole.NumElementi;
  }

  ++IndiceCorrFrase;

}  /* fine ciclo while */

/* il fatto che siamo usciti dal ciclo significa che il carattere
 * corrente di Frase è TERMINATORE_FRASE.  Occorre ancora "chiudere" la
 * parola  corrente, visto che all'interno del ciclo ciò viene fatto solo
 * quando il carattere che segna la fine della parola è un separatore */

ElencoParole.DatiParole[ElencoParole.NumElementi].Parola
[IndiceCorrParola] = '\0';

ElencoParole.DatiParole[ElencoParole.NumElementi].Lunghezza =
IndiceCorrParola;

++ElencoParole.NumElementi;

/*** stampa contenuto della lista sequenziale: ***/

printf("\n\nLe parole contenute nella frase sono:\n");

IndiceCorrElemento = 0;

while (IndiceCorrElemento < ElencoParole.NumElementi)
```

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

```
  {
    printf("\n%s\t%d", ElencoParole.DatiParole[IndiceCorrElemento].Parola,
    ElencoParole.DatiParole[IndiceCorrElemento].Lunghezza);

    ++IndiceCorrElemento;
  }

  printf("\n\n");
  return(0);
}


/*********************************************************************
Esempio di uso del programma:

---------
Inserisci una frase terminata dal carattere '.'.
La frase, '.' incluso, puo' comprendere al massimo 59 caratteri.
Le parole della frase devono essere separate da '_'.

Nel_mezzo_del_cammin_di_nostra_vita.


Le parole contenute nella frase sono:

Nel     3
mezzo   5
del     3
cammin  6
di      2
nostra  6
vita    4
---------


Esempio di uso SCORRETTO del programma (inserimento di una frase troppo lunga):

---------
Inserisci una frase terminata dal carattere '.'.
La frase, '.' incluso, puo' comprendere al massimo 59 caratteri.
Le parole della frase devono essere separate da '_'.
```

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

```
sadf lsdkfjlkj sadkljflksfdj asdlfkjsdflk sldkfjlsdkfj sdf sdfsdfa asdfasfd asdfasfd sdfffsd sadflksdafj asdflkj asdkjf
asdfasdf safdsdafaf.


Le parole contenute nella frase sono:

sadf    341

*** stack smashing detected ***:
LD_LIBRARY_PATH=/home/giulio/catkin_ws/devel/lib:/opt/ros/kinetic/lib:/opt/ros/kinetic/lib/x86_64-linux-gnu terminated
Aborted (core dumped)
---------

***********************************************

ESERCIZIO.
Come e' facile verificare, il programma non gestisce correttamente i casi
in cui sono presenti spazi in testa alla frase o prima del punto finale,
nonche' quelli in cui le parole sono separate da più di uno spazio.
Gli esempi seguenti illustrano quanto affermato:

---------
Inserisci una frase terminata dal carattere '.'.
La frase, '.' incluso, puo' comprendere al massimo 59 caratteri.
Le parole della frase devono essere separate da '_'.

_Nel_mezzo_del___cammin_di_nostra_vita.

Le parole contenute nella frase sono:

              0
Nel     3
mezzo     5
del         3
              0
              0
cammin  6
di          2
nostra  6
vita      4
```

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

```
---------

---------
Inserisci una frase terminata dal carattere '.'.
La frase, '.' incluso, puo' comprendere al massimo 59 caratteri.
Le parole della frase devono essere separate da '_'.

_Nel_mezzo_del___cammin_di_nostra_vita_.

Le parole contenute nella frase sono:

             0
---------

Modificare il programma in modo da consentire la presenza di più di uno
spazio tra le parole, nonché la presenza di spazi tra inizio della frase e
prima parola e/o tra ultima parola e punto.
**********************************************************************/
```

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari