

```
/*
 * Corso di Fondamenti di Informatica
 * Esercizio:
 * Versione modificata dell'esercizio es04_03.
 * In questa versione, in cui la frase da analizzare viene inserita
 * dall'utente sotto forma di argomento da linea di comando. Se il
 * separatore tra le parole è uno spazio, l'argomento comando (come
 * di consueto con gli argomenti che includono spazi) va racchiuso
 * tra doppi apici ("").
 * Se il file eseguibile ottenuto compilando il programma e' stato
 * chiamato (ad esempio) "test", il programma andra' chiamato
 * con un comando del tipo
 *
 * test " frase da analizzare ."
 *
 * Il programma stampa un messaggio di errore e termina nel caso in
 * cui non siano presenti argomenti; se gli argomenti sono piu' di
 * uno, solo il primo viene preso in considerazione.
 */

#include <stdio.h>

#include <string.h>
/* necessario per poter usare la funzione strcpy */

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

#define SEPARATORE_PAROLE ' '
/* Nota: SEPARATORE_PAROLE non puo' essere '.' perche' gia' usato per terminare la frase. ' ' (il carattere spazio) e' un
separatore accettabile qualora la frase sia fornita al programma (tramite linea di comando) racchiusa tra doppi apici ("").
*/

typedef enum {false, true} boolean;

int main(int argc, char *argv[])
{
    char Frase[LUNG_MAX_FRASE];
```

```
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 */
} TipodscrParola;
/* usato per rappresentare una singola parola */

struct
{
    TipodscrParola DatiParole[NUM_MAX_PAROLE];
    /* contiene le informazioni sulle parole contenute nella frase da
    * analizzare */
    int NumElementi;
    /* numero elementi dell'array effettivamente contenenti dati */
} ElencoParole;
/* 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 */
boolean ParolaGiàChiusa;
/* vale true se il prossimo carattere diverso da SEPARATORE_PAROLE e '.'
* e' l'inizio di una nuova parola, owerosia quando la parola precedente e'
* già stata "chiusa" */

if (1 == argc)
{
    printf("\n***ERRORE*** Nessun argomento e' stato passato al programma!\n\n");
    return (1);
}
```

```
strcpy(Frase, argv[1]);
/* 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;
ParolaGiachiusa = true;

while ( Frase[IndiceCorrFrase] != ' ' )
/* finché ci sono ancora caratteri da esaminare */
{
    if ( SEPARATORE_PAROLE != Frase[IndiceCorrFrase] )
/* Se il carattere corrente di Frase fa parte della parola
 * corrente in esame, aggiunge il carattere all'ultimo
 * elemento di ElencoParole */
    {
        ElencoParole.DatiParole[ElencoParole.NumElementi].Parola
        [IndiceCorrParola] = Frase[IndiceCorrFrase];
        ++IndiceCorrParola;
        ParolaGiachiusa = false;
    }
    else
/* il carattere corrente è un separatore di parole, dunque separa l'ultima
 * parola rilevata dalla successiva o da un altro separatore */
    {
        if ( false == ParolaGiachiusa )
/* se l'ultima parola rilevata non è stata ancora terminata */
        {
            /** 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;
ParolaGiachiusa = true;
}
/* se il carattere corrente è un separatore e ParolaGiachiusa è 'true'
 * tale carattere viene semplicemente saltato */
}
++IndiceCorrFrase;
} /* fine ciclo while */
}
/* il fatto che siamo usciti dal ciclo significa che il carattere
 * corrente di Frase è '.'. 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 */
/* La "chiusura" viene eseguita solo se l'ultima parola è effettivamente
 * aperta, ovvero se il '.' non era preceduto da spazi */
if ( false == ParolaGiachiusa )
{
    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");
```

```
IndiceCorreElemento = 0;

while (IndiceCorreElemento < ElencoParole.NumElementi)
{
    printf("\n%s\t%d", ElencoParole.DatiParole[IndiceCorreElemento].Parola,
    ElencoParole.DatiParole[IndiceCorreElemento].Lunghezza);
    ++IndiceCorreElemento;
}

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

/*****

Esempio di esecuzione:

`$. /test " 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

*****/