

```
/*
 * Corso di Fondamenti di Informatica
 * Esercizio:
 * modifica del programma precedente 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 o tra ultima parola e punto, come richiesto
 * dall'ESERCIZIO in coda al programma precedente.
 */

/* Le parti aggiunte o modificate rispetto all'esercizio precedente sono
 * seguite da un commento contenente @@@@bis@@@@ */

#include <stdio.h>

#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. */

typedef enum {false, true} boolean;

int main()
{
    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 */
    } TipodescrParola;
    /* usato per rappresentare una singola parola */
```

```
struct
{
    TipDescrParola 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 ParolaGiachiusa; /* @@@bis@@@ */
/* vale true se il prossimo carattere diverso da SEPARATORE_PAROLE e
* TERMINATORE_FRASE e' l'inizio di una nuova parola, ovvero sia quando la
* parola precedente e' gia' stata "chiusa" */

printf("\nInserisci una frase composta da non piu' di %d caratteri, terminata da %d. Le parole devono essere separate da
%c'.\n", LUNG_MAX_FRASE-1, TERMINATORE_FRASE, 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;
```

```

ParolagiChiusa = true;    /* @@@bis@@@ */
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;
    }
    ParolagiChiusa = false;    /* @@@bis@@@ */
}
else
/* il carattere corrente è un separatore di parole, dunque separa l'ultima
* parola rilevata dalla successiva o da un altro separatore */
/* @@@bis@@@ */
{
    if ( false == ParolagiChiusa ) /* @@@bis@@@ */
    /* 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;
    }
    ParolagiChiusa = true;    /* @@@bis@@@ */
}
/* se il carattere corrente è un separatore e ParolagiChiusa è 'true'

```

```
    * tale carattere viene semplicemente saltato */ /* @@@bis@@@ */
}
++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 */

/* La "chiusura" viene eseguita solo se l'ultima parola è effettivamente
 * aperta, ovvero se il TERMINATORE_FRASE non era preceduto da spazi */
/* @@@bis@@@ */

if ( false == ParolaGiachiusa ) /* @@@bis@@@ */
{
    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)
{
    printf("\n%s\t%d", ElencoParole.DatiParole[IndiceCorrElemento].Parola,
    ElencoParole.DatiParole[IndiceCorrElemento].Lunghezza);
    ++IndiceCorrElemento;
}
}
```

