14.5 Somma tra liste

Consideriamo ora il problema di memorizzare due sequenze di numeri interi in due liste lineari, visualizzare le due liste e assegnare la somma delle corrispondenti informazioni delle due liste in una terza lista lineare che infine è anch'essa da visualizzare.
La lista somma avrà lunghezza uguale alla minore delle altre due: se la prima sequenza è: 2, 100, 20, 0 e la seconda sequenza è: 7, 300, 9, 527, 0, la lista somma sarà composta dai valori 9, 400, 29. Suddividiamo il problema nei seguenti sottoproblemi:

- creare la prima lista;
- creare la seconda lista;
- visualizzare la prima lista;
- visualizzare la seconda lista;
- creare la terza lista inserendo nei suoi elementi il risultato della somma delle altre due;
- visualizzare la terza lista.

Per creare le due liste possiamo utilizzare la funzione `crea_lista2` vista nel paragrafo precedente, chiamandola due volte nel `main`; la prima volta il puntatore restituito verrà assegnato a `punt_lista1`, la seconda a `punt_lista2`:

```
printf("\n PRIMA LISTA \n");
     punt_lista1 = crea_lista2();

     printf("\n SECONDA LISTA \n");
     punt_lista2 = crea_lista2();
```

Procederemo analogamente per visualizzare le due liste:

```
visualizza_lista(punt_lista1);
     visualizza_lista(punt_lista2);
```

Per costruire la lista risultato della somma delle prime due costruiremo la funzione `somma_liste`,
Quest'ultima riceve in ingresso i puntatori alle due liste da sommare e restituisce il puntatore alla nuova lista:

```
punt_lista3 = somma_liste(punt_lista1, punt_lista2);
```

Utilizzeremo la procedura `visualizza_lista` per visualizzarla:

```
visualizza_lista(punt_lista3);
```

Nel Listato 14.3 viene riportato il `main` e la funzione `somma_liste`.
Si tratta di scandire in parallelo le due liste, sommare il campo informazione degli elementi corrispondenti, creare un elemento della terza lista e inserivi il risultato ottenuto. Si effettueranno somme finché una delle due liste non termina. I parametri attuali `punt_lista1`, `punt_lista2` diventano i parametri formali `p1`, `p2` e il valore di ritorno `punt_lista3` è il puntatore alla lista somma delle precedenti:

```
punt_lista3 = somma_liste(punt_lista1, punt_lista2);
```

Si utilizzano due puntatori `p1` e `p2` per visitare le due liste, `p3` per creare il primo elemento della terza lista e `paus3` per creare i successivi. Il ciclo ha termine quando `paus1` o `paus2` diventa uguale a `NULL`. Il puntatore `p3` alla terza lista viene restituito al programma chiamante.

```
/* Crea due liste e le visualizza. Somma gli elementi
corrispondenti delle due liste, inserisce il risultato
in una terza lista e la visualizza                    */

#include <stdio.h>
#include <malloc.h>

struct elemento {
  int inf;
  struct elemento *pun;
};

struct elemento *crea_lista2();
void visualizza_lista(struct elemento *);
struct elemento *somma_liste(struct elemento *,struct elemento *);

main()
{
struct elemento *punt_lista1, *punt_lista2, *punt_lista3;

printf("\n PRIMA LISTA \n");
punt_lista1 = crea_lista2();
```

```
printf("\n SECONDA LISTA \n");
punt_lista2 = crea_lista2();

visualizza_lista(punt_lista1);
visualizza_lista(punt_lista2);

/* Invocazione della funzione per la somma delle liste */
punt_lista3 = somma_liste(punt_lista1, punt_lista2);

/* Visualizzazione della lista somma delle precedenti  */
visualizza_lista(punt_lista3);
}


/*  Somma gli elementi corrispondenti di due liste
e inserisce il risultato in una terza lista        */
struct elemento *somma_liste(struct elemento *p1, struct elemento *p2)
{
struct elemento *p3 = NULL, *p3aus;

if(p1!=NULL && p2!=NULL) {
                      /* Creazione primo elemento  */
 p3 = (struct elemento *)malloc(sizeof(struct elemento));
 p3->inf = p1->inf + p2->inf;             /* somma */
 p3aus = p3;              /* p3aus punta III lista */
 p1 = p1->pun;            /* Scorrimento I lista   */
 p2 = p2->pun;            /* Scorrimento II lista  */

 /* Creazione elementi successivi */
 for(; p1!=NULL && p2!=NULL;) {
  p3aus->pun = (struct elemento *)malloc(sizeof(struct elemento));
  p3aus = p3aus->pun;     /* Scorrimento III lista */
  p3aus->inf = p1->inf + p2->inf;         /* Somma */
  p1 = p1->pun;           /* Scorrimento I lista   */
  p2 = p2->pun;           /* Scorrimento II lista  */
 }
 p3aus->pun = NULL;     /* Marca di fine III lista */
}
return(p3);           /* Ritorno del puntatore alla III lista */
}

/* ATTENZIONE: devono essere aggiunte le definizioni delle
funzioni per creare - crea_lista2() - e visualizzare
- visualizza_lista() - la lista presenti nel precedente Listato 14.2
*/
```

Listato 14.3 Somma dei valori di due liste