

**Prova d'esame di Laboratorio di Programmazione Strutturata
per il corso di laurea in Scienze e Tecnologia dei Media
Prova scritta – 2 Luglio 2013**

• **Esercizio 1**

Si scriva una propria versione della *function* di libreria `strchr`. Per fissare le idee, il prototipo della *function* richiesta è il seguente:

```
char *prima_pos_car_in_stringa(char stringa[], char lettera);
```

esattamente come la *function* di libreria `strchr`, `prima_pos_car_in_stringa` deve restituire il primo indirizzo di memoria (a partire dall'inizio di `stringa`) corrispondentemente al quale si trova proprio il carattere `lettera`, che viene passato in argomento.

Qualora il carattere `lettera` non sia tra quelli presenti tra l'inizio e il carattere di fine `stringa`, allora `prima_pos_car_in_stringa` dovrà restituire l'indirizzo `NULL`.

• **Esercizio 2**

Si consideri una *lista concatenata* che è implementata in linguaggio **C**, in modo tale che ogni suo nodo è rappresentato da una `struct` costituita da due campi: il primo è di tipo `int` e il secondo è un *puntatore* alla struttura che rappresenta il nodo successivo della sequenza. Per fissare le idee, ogni nodo di tipo `NodoLista` sia definito come segue:

```
struct NODO {  
    int info;  
    struct NODO *next;  
};  
typedef struct NODO NodoLista;
```

Come al solito, la *lista concatenata* è definita grazie a un *puntatore* al primo elemento della sequenza. È quindi utile introdurre il tipo `Lista` definito come segue:

```
typedef struct NodoLista* Lista;
```

Si scriva una *function* il cui prototipo è il seguente:

```
int totale(Lista L);
```

essa deve restituire la somma di tutti i numeri `info` che compaiono nei nodi della *lista concatenata*, il cui primo elemento è identificato dal puntatore `L` passato in argomento. Come al solito, quando `L` è uguale a `NULL`, allora si intende che la *lista* è vuota e, quindi, la *function* `totale` dovrà semplicemente restituire il valore 0.

• **Esercizio 3 (facoltativo)**

Si scriva una *function ricorsiva* il cui prototipo è il seguente:

```
NodoLista *trova_somma_parziale(Lista L, int somma_parz);
```

che ha lo scopo di restituire il puntatore `p` all'**ultimo elemento** che gode della seguente proprietà: *la somma parziale di tutti i numeri `info` che compaiono nei nodi della lista concatenata, a partire dall'elemento corrispondente all'indirizzo `L` fino a quello individuato da `p`, è inferiore al valore di `somma_parz` che viene passato in argomento.*

Nella scrittura della *function* `trova_somma_parziale` si tengano presenti le seguenti convenzioni:

1. quando la *lista* individuata da `L` è vuota, allora la *function* dovrà semplicemente restituire `NULL`;
2. quando il numero `info` del *primo elemento della lista* (cioè quello corrispondente all'indirizzo `L`) è maggiore o uguale al valore di `somma_parz`, allora la *function* dovrà restituire `NULL`;
3. quando la somma di tutti i numeri `info` che compaiono nei nodi della *lista concatenata* è inferiore al valore di `somma_parz`, allora la *function* deve restituire il puntatore all'*ultimo elemento della lista concatenata*;
4. siccome la *function* `trova_somma_parziale` deve soddisfare la richiesta di essere di tipo *ricorsivo*, allora essa deve *richiamare se stessa*, a meno che non siano verificate *quelle condizioni banali che "sciolgono la ricorsione"*, tali che l'indirizzo `p` da restituire può essere determinato senza ulteriori *richiami alla function stessa*.