

Settimana 6 Esercizi per casa (liste)

MEMO. Pensare ricorsivo:

- definire i casi di terminazione (quando la risposta e' immediata)
- isolare un problema piu' piccolo.

Si parte dalla chiamata ricorsiva, che fornisce la soluzione nel caso ridotto, e si costruisce la soluzione completa intorno alla soluzione del problema ridotto.

Una lista e' la lista vuota, oppure un nodo seguito da una lista (coda).

La definizione di lista aiuta a costruire funzioni ricorsive sulle liste. Spesso il problema si riduce, in modo naturale, allo stesso problema sulla coda.

```
... risolvi (... lista)
if (lista == NULL)
    risolvi il problema per la lista vuota;
else
    {usa
      risolvi (coda);
      per risolvere il problema su lista}
```

ES 0 Scrivere una funzione *MuoviNodo*, che date due liste *l1* (non vuota) ed *l2*, rimuove il primo nodo di *l1*, e lo mette in testa alla seconda. (Questa *non* e' una funzione ricorsiva!!)

ES 1 Data una lista di interi, scrivere una *funzione ricorsiva* che riempie il campo *int* con la sua distanza dalla fine della lista (l'ultimo elemento ha distanza 0).

ES 2 Scrivi una *funzione ricorsiva alterna*, che prende due liste, e le mette insieme, prendendo alternativamente i nodi dall'una e dall'altra (non si creano nuovi nodi).

Se finiscono gli elementi di una delle liste, tutti i nodi vengono presi dall'altra.

Esempio. Se *l1* = 3->3->3->3 ed *l2* = 7->7,
alterna (*l1*,*l2*) restituisce la lista 3->7->3->7->3->3.

ES 3 Scrivere una delle seguenti *funzioni ricorsive* (a scelta):

- **copiaLista** che data una lista di interi *l1*, la copia creando una nuova lista *l2*:
l1 = 1->2->3->4 *l2* = 1->2->3->4
- **copiaRov**, che data una lista di interi *l1*, crea una nuova lista *l2* come segue:
l1 = 1->2->3->4 *l2* = 4->3->2->1

Test (ho davvero una copia?) Creare una lista *l1*, generare *l2*, moltiplicare i valori di *l1* per due, stampare *l1* ed *l2*.