

### Esercizio 3 del 26/3/2013 da consegnare corretto entro il 3/3 a mezzanotte

Questo esercizio è molto simile all'esercizio 2 di questa stessa settimana. La differenza è che si staccano i nodo del match da L solo quando il match è completo, cioè quando tutti i valori di P sono trovati nella lista L (non necessariamente in nodi contigui). La funzione da realizzare si chiama GC per G Completo. Vediamo un esempio.

**Esempio:** supponiamo che  $L = 2 \rightarrow 3 \rightarrow 3 \rightarrow 1 \rightarrow 4$  e che  $P = [3, 1, 2]$  con  $\dim P = 3$ . In questo caso non c'è match completo e quindi L resta invariata e la funzione GC restituisce la lista vuota (di tipo `nodoG*`). Se invece  $L = 2 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 1 \rightarrow 4$  e  $P = [3, 1, 4]$ , allora  $(L-P)$  è  $2 \rightarrow 2 \rightarrow 2 \rightarrow 1 \rightarrow 1$  mentre  $\text{Comp}(L-P)$  ha 3 nodi che puntano al secondo, quarto e ultimo nodo della lista originale L.

La funzione GC deve soddisfare la seguente pre- e post-condizione.

$\text{PRE\_GC} = (L \text{ è lista corretta e } L = vL, P \text{ ha } \dim P \text{ elementi definiti con } \dim P \geq 0)$

`nodoG* GC(nodo*&L, int*P, int dimP, ...)`

$\text{POST\_GC} = (\text{se c'è un match completo (non necessariamente contiguo) di } P \text{ in } L, \text{ allora GC restituisce col return } \text{Comp}(vL-P) \text{ e } (vL-P) \text{ attraverso il parametro L passato per riferimento, se invece non c'è un match completo di } P \text{ in } L, \text{ allora GC restituisce la lista vuota col return e la lista originale col parametro}).$

I puntini nella lista dei parametri di GC indicano che è possibile aggiungere qualche parametro. Nel caso si dovranno aggiungere opportune condizione alla pre- e post-condizione.

Dopo l'invocazione di GC, il main stampa opportunamente i nodi delle 2 liste prodotte da GC. Si richiede la dimostrazione induttiva di correttezza della funzione GC.

**Attenzione:** GC non deve creare né distruggere alcun nodo di tipo `nodo`, e non deve creare nodi di tipo `nodoG` se non c'è match completo. Insomma crea nodi `nodoG` solo se deve restituire col return una lista non vuota.