

### Esercizio 3

Si tratta di fare pattern matching di un array  $\text{int } P[0..\text{dimP}-1]$  sui campi info dei nodi di una lista concatenata  $L(n)$ . Vediamo subito un esempio. Consideriamo solo match completi e contigui.

**Esempio:** sia  $L(n)=3 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 0 \rightarrow 0 \rightarrow 1$  la lista, sia  $\text{dimP}=3$  e  $P=[3,0,0]$ , allora c'è un match di  $P$  nella lista  $L(n)$  e inizia dal nodo di indice 3 della lista (gli indici dei nodi di una lista sono  $0,1,2,\dots$ ). Si richiede in un caso di successo del match come questo, di restituire col return la parte della lista  $L$  in cui abbiamo trovato il match completo e contiguo, cioè la sottolista  $3 \rightarrow 0 \rightarrow 0$  di  $L(n)$ , e attraverso il parametro  $n$ , passato per riferimento, si deve restituire quello che resta della lista  $L(n)$  originale, cioè:  $3 \rightarrow 2 \rightarrow 1 \rightarrow 1$ . La notazione che useremo per indicare queste 2 liste sarà:  $\text{match}(L(n), P[3,0,0])$  e  $\text{resto\_match}(L(n), P[3,0,0])$ .

In caso  $P$  fosse  $P=[3,0,1]$ , non ci sarebbe match completo e contiguo in  $L(n)$  e quindi col return si dovrebbe restituire 0 e col parametro passato per riferimento, andrebbe restituita la lista  $L(n)$  originale. In questo caso  $\text{match}(L(n), [3,0,1])=0$  e  $\text{resto\_match}(L(n), [3,0,1])=L(n)$ . In caso ci fossero vari match di  $P$  in  $L(n)$ ,  $\text{match}(L(n), P[0..\text{dimP}-1])$  è quello che inizia prima in  $L(n)$ .

**Attenzione:** nessun nodo diverso da quelli presenti in  $L(n)$  va allocato. Nessun nodo va deallocato.

E' dato un main che compie le letture, fa le stampe e costruisce la lista  $L(n)$  su cui tentare il match. Esso invoca la funzione ricorsiva  $\text{match}$  che deve rispettare il seguente prototipo ed essere corretta rispetto alle seguenti pre- e post-condizioni:

PRE=( $L(n)$  è una lista corretta,  $P$  ha  $\text{dimP}$  elementi, chiamiamo  $vL(n)=L(n)$ )

nodo\*  $\text{match}(\text{nodo* } \&n, \text{int* } P, \text{int } \text{dimP})$

POST=(in  $L(n)$  c'è un match di  $P$ , allora la funzione restituisce col return  $\text{match}(vL(n), P[0..\text{dimP}-1])$  e  $L(n)=\text{resto\_mach}(vL(n), P[0..\text{dimP}-1])$ , se invece non c'è il match allora la funzione restituisce 0 e  $L(n)=vL(n)$ )

**Consiglio:** conviene introdurre anche almeno un'altra funzione ausiliaria che deve essere anch'essa ricorsiva.

**Correttezza:** scrivere pre- e post-condizione della funzione ausiliaria e poi dimostrare induttivamente la correttezza della funzione  $\text{match}$  assumendo che la funzione ausiliaria sia corretta rispetto alle vostre pre e post. .