

Si chiede di definire una funzione ricorsiva `patt_match` col seguente prototipo:

```
nodo* patt_match(nodo*&L, int*A, int nA)
```

che cerca di matchare il pattern A sulla lista L, restituendo nel parametro L i nodi che fanno parte del match e col return i nodi di che non fanno parte del match.

**Esempio:** sia  $L = 4 \rightarrow 2 \rightarrow 0 \rightarrow 3 \rightarrow 11 \rightarrow 2 \rightarrow 10$  e  $A = [4, 0, 11, 0]$ . E' possibile matchare solo i primi 3 elementi di A con il primo, il terzo e il quinto nodo di L. Quindi `patt_match` deve restituire  $4 \rightarrow 0 \rightarrow 11$  nel parametro L e  $2 \rightarrow 3 \rightarrow 2 \rightarrow 10$  col return. Ovviamente si deve cercare di matchare elementi successivi di A su nodi successivi di L e di matchare più elementi di A possibile. Non è richiesto di matchare tutto A.

La PRE e POST di `patt_match` sono le seguenti:

PRE=(Lista(L) bel formata, A ha nA elementi,  $vL=L$ )

POST=(L è la lista che consiste dei nodi di  $vL$  che matchano con A e col return restituisce la lista degli altri nodi di  $vL$ )

**Correttezza:** dare la prova induttiva della correttezza di `patt_match` rispetto alla PRE e POST data.