

Esercizio 2 del 7/5/2018

L'esercizio è molto simile all'esercizio 1 di questa settimana (e al 3 della settimana scorsa). Unica differenza il modo in cui vengono restituite la lista del match e quella restante. Ora per farlo useremo questa struttura:

```
struct doppialista{nodo* listamatch, * restante};
```

In questo esercizio si richiede di scrivere la funzione ricorsiva match2 col seguente prototipo:

```
doppialista* match2(nodo* T, char* P, int dimP)
```

PRE=(L(n) è corretta, dimP>=0, P contiene dimP elementi)

POST=(se esiste match di P in L(n) allora match2 restituisce il puntatore X ad un valore di tipo doppialista tale che X->listamatch sia la lista del match trovato e X->restante sia la lista restante, se invece non c'è match, match2 restituisce il puntatore 0 e L(n) resta inalterata).

Si osservi che input e output sono realizzate dal main che è dato.

Correttezza: dimostrare induttivamente la correttezza di match2 rispetto a PRE e POST.