

## Esercizio per casa n. 8 3/3

consegna il 9/3 a mezzanotte

comando di consegna: **consegna prog8**

*Tutte le funzioni devono avere una pre- ed una post-condizione. Cercate di scrivere la dimostrazione di correttezza delle vostre funzioni seguendo lo schema induttivo.*

1) Si tratta di stampare i campi info dei nodi di un albero binario attraversando l'albero in ordine post-fisso. Però vogliamo stampare un valore ogni  $k$  nodi attraversati. Quindi i primi  $k-1$  nodi attraversati non vengono stampati. Si stampa il prossimo nodo, poi di nuovo  $k-1$  nodi attraversati senza stampa e poi 1 nodo stampato e così via. Si veda l'esempio seguente:

Esempio: si consideri il seguente albero binario (i cui nodi contengono caratteri),  
 $a(b(c(\_,\_),d(\_,\_)),e(f(l(\_,\_),\_),h(\_,\_)))$ . La lista in ordine post-fisso dei suoi nodi è: c d b l f h e a. La funzione richiesta con  $k=3$  deve stampare b h.

La funzione deve avere il seguente prototipo: `int f1(nodo* x, int salta, int k)`, dove  $k$  fissa ogni quanti nodi si deve stamparne uno ( $k=3$  nell'esempio precedente) e  $salta$  è il numero di nodi che si deve saltare prima di stampare il prossimo.

2) In un albero binario cerchiamo un nodo con campo informativo  $y$  e vogliamo restituire il cammino che dalla radice porta a questo nodo. Il cammino che porta dalla radice ad un nodo è una sequenza di 0 e 1, dove 0 significa "vado a sinistra" e 1 "vado a destra".

Esempio: si riconsideri l'albero dell'esempio precedente. Con  $y='d'$  il cammino da calcolare è 01.

La funzione richiesta costruisce il cammino in un array `int C[]` (di dimensione che si assume sufficiente) che riceve dal chiamante. Il prototipo della funzione deve essere:

`int F2(nodo* x, char y, int *C, int top)` in cui  $x$  punta alla radice dell'albero,  $y$  è il valore cercato,  $C$  serve a contenere il cammino costruito e  $top$  indica la prima posizione libera di  $C$ .

*AIUTO: F2 restituisce un valore intero che deve servire ad indicare se è stato trovato un nodo con campo informativo  $y$  oppure no. Nel primo caso il valore restituito deve indicare quanti elementi di  $C$  sono occupati dal cammino che conduce dalla radice al nodo trovato (e  $C$  conterrà quel cammino).*

### 3) ATTENZIONE, è OPZIONALE e non è facile

Il problema è come il (2), ma si vuole il cammino più corto che conduce dalla radice ad un nodo che contiene campo informativo uguale a  $y$ . Siete liberi di scegliere il prototipo e di usare anche passaggio per riferimento e/o memoria dinamica.

*Badate a quante copie di  $C$  create ! Alla fine deve restarne solo una se esiste almeno un nodo con  $y$  (e quindi esiste un cammino più corto) e nessun  $C$  altrimenti.*