

# Esercitazione 12

## Argomento: alberi binari + liste e code

Per questa esercitazione vengono forniti i file

- **esercizio.h** ed **esercizio.c** , contenenti la definizione e l'implementazione dei tipi di dato **TipoAlbero**, **TipoListaSCL** e **Coda**
- **test\_esercizio.c**, contenente un main di prova, che potete modificare a vostro piacimento per testare gli esercizi proposti.

### Esercizio 12.1

Implementare la funzione C

**int ricercaLivello(TipoAlbero a, TipoInfoAlbero v, int livello);**

che, dati un albero binario **a** e un valore **v**, restituisca il livello dell'albero dove si trova **v**.

NB: **livello** è il livello corrente dell'albero

### Esercizio 12.2

Implementare la funzione C

**int verificaNodi(TipoAlbero a, int livello);**

Che, dato albero binario, restituisca la somma del numero di nodi di valore pari che si trovano nei livelli pari e di quello dei nodi di valore dispari che si trovano nei livelli dispari.

### Esercizio 12.3

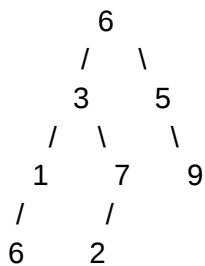
Implementare la seguente funzione C

**int singleChildSum(TipoAlbero a);**

che, dato in input un albero binario, restituisca la somma dei valori dei nodi che hanno un solo successore.

### Esempio

Dato il seguente albero binario **src** in ingresso:



singleChildSum( **src** ) dovrà restituire:  $1 + 7 + 5 = 13$

## Esercizio 12.4

Implementare la seguente funzione C

**TipoListaSCL\* listaNodiFoglia(TipoAlbero a);**

Che, dato un albero binario, restituisca una lista con i valori contenuti nelle foglie.

## Esercizio 12.5

Implementare la seguente funzione C

**Coda\* codaNodiDueFigli(TipoAlbero a);**

Che, dato un albero binario, restituisca una coda contenente i valori dei nodi dell'albero che hanno esattamente due figli.

**N.B. Coda** e le sue funzioni sono disponibili in **esercizio.h** e **esercizio.c**.

## Esercizio 12.6

Implementare la seguente funzione C

**TipoListaSCL\* listaPercorso(TipoAlbero a);**

Che, dato un albero binario, restituisca la lista dei nodi contenuti nel percorso più lungo dalla radice a una delle foglie. Se esistono diversi percorsi di dimensione massima, si restituisca quello più a sinistra.

## Esercizio 12.7

Implementare la seguente funzione C

**void aggiungiFratello(TipoAlbero a);**

Che, dato un albero binario, aggiunga una foglia ad ogni nodo avente solo un nodo figlio. Il valore del nuovo nodo deve essere 0.