

Seconda Esercitazione

1 Esercizi

1. Data una lista l di numeri naturali, progettare una funzione che costruisce una nuova lista che contiene tutti i numeri pari di l ordinati in ordine crescente
2. Data una lista l di numeri naturali, progettare una funzione che costruisce una nuova lista che contiene tutti i numeri dispari di l ordinati in ordine decrescente.
3. Dato un albero binario di ricerca contare il numero di nodi maggiori di un valore n dato in input.
4. Dato un albero binario contare le occorrenze dei numeri pari.
5. Progettare una struttura che simuli il database di un negozio. Ogni prodotto è rappresentato da una struttura con i seguenti campi:
 - nome prodotto
 - quantità
 - peso
 - anno di scadenza
 - (a) Controllare se il negozio possiede almeno due prodotti scaduti
 - (b) Data una lista di prodotti controllare se sono tutti presenti nel database del negozio (controllare sia la presenza del prodotto che la quantità)
 - (c) Aggiungere una lista di prodotti data in input al database del negozio (es. aggiungere prodotti nuovi, o modificare quantità e anno di scadenza per prodotti già esistenti)
 - (d) Contare la quantità complessiva presente in negozio di una lista di nomi di prodotti dati in ingresso.
6. Dato un albero binario controllare che abbia solo una foglia dispari.
7. Progettare una funzione che preso in input un albero binario restituisce la lista dei nodi multipli di 3

2 Esercizi più difficili

1. Progettare una funzione che prende in input un albero binario e due numeri min e max e restituisce un albero i cui nodi compresi fra min e max sono rimpiazzati con 0 e tutti gli altri sono lasciati inalterati
2. Progettare una funzione che preso in input un'albero binario restituisce il corrispondente albero binario di ricerca
3. Data una lista l di nomi restituire la lista ordinata in ordine alfabetico. (Sugg. Utilizzare la funzione di confronto *string <?*)