

PROGRAMMAZIONE E STRUTTURE DATI - MAT DIVISO 3 RESTO 0

ESERCIZI

1. Implementare un algoritmo **iterativo** che misuri l'altezza di un albero binario. Implementare anche la versione ricorsiva e verificare che produca lo stesso risultato (nel main).
2. Progettare e implementare un ADT per le espressioni aritmetiche (ad es. $((6.1-7)+((5/3)*4))$). Rappresentare le espressioni mediante un albero binario.
 - I nodi intermedi che rappresentano gli operatori.
 - Le foglie rappresentando gli operandi.
 - Fornire operatori che consentano di definire somma, sottrazione, moltiplicazione divisione, numero.
 - Consentire la stampa dell'espressione.
 - Consentire il calcolo del valore dell'espressione.
3. Verificare se una data espressione aritmetica è ben bilanciata rispetto a tre tipi di parentesi: $()$, $[]$, $\{ \}$
 - per semplicità supponiamo che non esista un ordine di priorità fra i tre tipi di parentesi: $(a + \{b - 1\}) / [b + 2]$ è ammessa come valida.
4. Realizzare l'ADT Lista utilizzando array dinamici. Nota: è necessario riallocare l'array quando necessario.
 - Fare lo stesso per le ADT Stack e Coda.
5. Realizzare delle funzioni che copi il contenuto di uno Stack in una Lista (mantenendo il contenuto dello Stack).
 - Utilizzare solo le funzioni proprie dell'ADT Stack.
 - Fare lo stesso per l'ADT Coda.
6. Realizzare SelectionSort, BubbleSort, InsertionSort per le liste linkate. Farlo anche per le liste basate su array.
7. Stampare gli elementi di uno Stack in maniera ricorsiva.
8. Aggiungere all' ADTLista l'operatore ListnewRandomList(int n) che restituisce una lista con n elementi casuali.
 - Fare in modo che possa funzionare con qualsiasi tipo di Item.
9. Dato un intero n , implementare un algoritmo che generi un albero binario con n nodi, casuale per contenuto dei nodi e struttura.
10. Scrivere una funzione che, dato un albero in input, inserisca in una lista solo i nodi di livello pari.