Compito di Programmazione
16 Giugno 2023

Nota Bene. Ogni esercizio deve essere svolto su un foglio diverso.
Scrivere Nome, Cognome e Matricola su ogni foglio.

1. **(punti 8)** Scrivere una funzione che prende in input un array di interi (non ordinato) e la sua lunghezza (più altri parametri a scelta dello studente) e ritorna il numero di occorrenze maggiore di un intero nell'array. È possibile usare funzioni ausiliarie.

2. **(punti 8)** Un *albero binario di ricerca* è un albero binario in cui, per ogni nodo, i valori memorizzati nel sotto-albero di sinistra sono minori del valore del nodo e i valori del sotto-albero di destra sono maggiori del valore del nodo. Un *albero binario di ricerca invertito* è quando i valori nei sottoalberi sono in una relazione d'ordine invertita (a sinistra i maggiori, a destra i minori).

   Definire le strutture dati necessarie per rappresentare gli alberi binari di ricerca (sia semplici che invertiti) e le liste di interi e implementare le seguenti funzioni:

   - `create_tree` che prende come parametri un albero binario di ricerca e una lista di interi e inserisce gli elementi della lista nell'albero.
   - `facimm_ammuina` che prende come parametri un albero binario di ricerca e ritorna l'albero binario di ricerca invertito che contiene gli stessi elementi. Non bisogna usare l'istruzione `new` (l'albero binario di ricerca va distrutto).
   - `max_depth` che prende come parametro l'albero e restituisce l'altezza dell'albero.

   [N.B. Si possono usare funzioni ausiliarie se definite.]

3. **(punti 8)** Un'agenzia offre la possibilità di costruirsi una visita su misura scegliendo da un catalogo di punti di interesse di una città (rappresentato come `array`). Un punto di interesse è caratterizzato da un identificativo, un nome e un prezzo. Si rappresenti un punto di interesse tramite `struct` e il suo identificativo tramite `int`. L'agenzia applica uno sconto crescente ogni N punti di interesse scelti (es. se lo sconto è del 5% ogni 3, applicherà un sconto del 5% per 3, 4 o 5 punti di interesse scelti, 10% per 6, 7 o 8 punti e così via). Si implementi la classe `Agenzia`, il relativo costruttore e i seguenti metodi:

   - `trova_prezzo`: il quale dato un identificativo cerca nel catalogo il prezzo del punto di interesse e lo restituisce;
   - `calcola_prezzo`: il quale dato l'insieme di identificativi scelti per la visita calcola il prezzo totale, scontato come descritto sopra, e lo restituisce.

   Esistono anche agenzie che offrono due servizi aggiuntivi. Un servizio di "salta coda" il quale ha un costo fisso applicato una sola volta per tutta la visita e un servizio di "guida turistica" che ha un costo fisso da moltiplicare per il numero di punti di interesse scelti. Si implementi la sottoclasse `AgenziaVIP` con il relativo costruttore. Sfruttando opportunamente l'ereditarietà, si calcoli il prezzo finale tenendo conto dei servizi aggiuntivi.
   In tutto l'esercizio 3 non è consentito usare le liste.