

Compito di Programmazione

09 Luglio 2021

Nota Bene. Consegnare solamente la bella copia. Ogni esercizio deve essere svolto su una pagina diversa. Alla fine del compito, usando una applicazione che faccia la scansione, tipo CamScanner, fare una foto a tutto il compito col cellulare, il file deve chiamarsi `Cognome_Nome.pdf`. Inviare le scansioni in pdf per email a `cosimo.laneve@unibo.it`, `giuseppe.lisanti@unibo.it`, `adele.veschetti2@unibo.it`.

1. (**punti 8**) Scrivere una funzione che prende in input un array di interi dei quali metà sono pari e metà sono dispari (i pari e i dispari sono mischiati) e modifica l'array in maniera tale che nelle posizioni pari (0, 2, 4, ...) ci siano gli interi pari ordinati in maniera decrescente e nelle posizioni dispari (1, 3, 5, ...) ci siano gli interi dispari ordinati in maniera crescente.

Non si possono utilizzare array ausiliari.

2. (**punti 8**) Un'agenzia di viaggi utilizza una lista per salvare le vacanze che offre. Ogni vacanza è caratterizzata dal numero di giorni, il prezzo totale e la tipologia ("mare", "montagna", "città"). Definire la struttura dati necessaria a rappresentare la lista e le seguenti funzioni:

- Una funzione che presa una tipologia come parametro ("mare", "montagna", "città"), restituisce una nuova lista contenente solamente le vacanze della tipologia scelta, ordinata in ordine crescente rispetto al prezzo giornaliero delle vacanze.
- Una funzione che preso un numero di giorni, restituisce la tipologia di vacanza disponibile per il numero di giorni inserito. Se esistono più vacanze disponibili, la funzione ritorna la vacanza più economica.

3. (**punti 8**) Un distributore di benzina può essere Self oppure Servito. Nel caso Self un cliente può effettuare in maniera autonoma un rifornimento di benzina o di diesel. Nel caso di Servito è possibile effettuare rifornimenti di benzina, diesel e metano. E' possibile utilizzare una struttura dati che raccoglie tutte le informazioni legate al tipo di rifornimento (nome, litri e litri massimi). Si definisca la classe **Self** e si implementi il costruttore e i seguenti metodi:

- `aggiungi_litri()`, il quale riceve come parametri il tipo e i litri, controlla che non sia stata superata la soglia di litri massimi che è possibile immagazzinare per quel tipo e aggiunge i litri passati come parametro. Il metodo restituisce un boolean.
- `decrementa_litri()`, il quale riceve come parametri il tipo e i litri e controlla se ci sono abbastanza litri per effettuare un rifornimento di quel tipo. In caso positivo li scala dai litri disponibili per quel tipo. Il metodo restituisce un boolean.

Successivamente si definisca la classe **Servito** e si implementi il costruttore e i metodi sopra elencati opportunamente modificati sfruttando l'ereditarietà.