Informatica III / programmazione avanzata

Durata 2.15 h

25/01/22

Chiama i tuoi progetti con COGNOME_NOME_ESX con X il numero dell'esercizio. Per la consegna fai uno zip per progetto o uno zip unico come preferisci e caricalo su ilias sotto Consegna. Usa carica unico file, NON come file zip.

1. Funzione in C

Scrivi una funzione in C che dato in ingresso un array numeri interi lo ordina.

Scrivi tre versioni:

- Una iterativa non ricorsiva che ordine mediante il bubble sort
- Una ricorsiva (con tail call) che sia la versione ricorsiva di quella iterativa. Nota che il bubble sort ad ogni ciclo ha un elemento (o il primo o l'ultimo dipende da come hai scritto il bubble sort) che sicuramente non verrà più mosso.

Scrivi anche un main di esempio in cui chiami le funzioni con almeno i seguenti array:

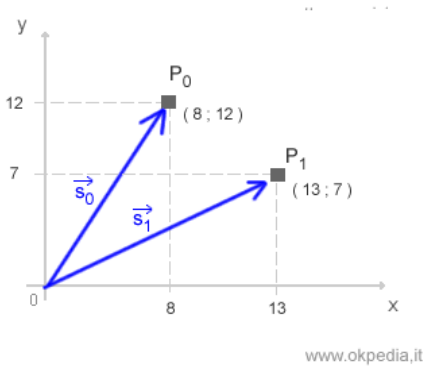
{-6,7, -9,1,3} e {0, 1, 2, 3} e {} (array vuoto).

Stampa l'array prima e dopo l'ordinamento con una funzione printIntArray che definisci tu in modo ricorsivo.

Non usare alcuna variabile globale.

2. Tipi opachi

Definisci il tipo opaco **vettore** che rappresenta un vettore matematico/fisico n-dimensionale.



Costruttore: inizializza il vettore con n che rappresenta la dimensione e un array di float che contiene i valori sulle n dimensioni (nella figura n =2)

toString che restituisce una stringa (array di caratteri) del vettore in formato "(x1, x2, …. xn)"

somma che dati due vettori fa la somma. Se hanno dimensioni diverse, considera quello con dimensione minore come se avesse 0 in tutte le dimensioni mancanti.

cancella che distrugge il vettore e libera la memoria

Fai un esempio in cui crei 3 vettori s0: [8,12], s1:[13,7] e s2:[0,-1,5], li stampi, fai la somma tra il s0 e s1 e tra s1 e v2, stampi le somme e liberi tutto.

3. Ereditarietà privata in C++

Che differenza c'è tra ereditarietà pubblica e privata in C++? Spiegalo per bene con un esempio ben commentato.

4. Smart pointers in C++ (solo AA 2122)

Fai un esempio in cui usi gli smart pointers e fai vedere la differenza rispetto all'uso dei raw pointers.

5. Visitor Pattern

Un **Pasto** può essere un **Pranzo** o una **Cena**

Si vuole, mediante visitor, calcolare il consumo e il costo al Km come nella tabella:

	Costo	Orario
Pranzo	10	12
Cena	20	20

Implementa il tutto e metti nel main un esempio in cui calcoli sia il costo che l'orario per un paio di pasti.

Nel main definisci un metodo **static sommaCosto** che data una lista (List) di **Pasto** calcola la somma die loro costi, utilizzando il visitor definito sopra e la stampa. Definisci **sommaCosto** in modo che possa prendere sia lista di Pasto che di Pranzo che di Cena. Nel main costruisci una lista e prova a chiamare il metodo di **sommaCosto**.

6. Programmazione funzionale

`Haskell (solo AA 2122):`

`Definire in maniera ricorsiva una funzione feo (fondi e ordina) (type signature Ord a ⇒ [a] → [a] → [a]) che prende in input due liste ordinate e le fonde in un'unica lista ordinata. Scrivo su un file .txt e consegnalo.`

Scala (prima di AA2122):

Scrivi in scala la funzione concat che data una lista di stringhe, le concatena tutte, tranne quelle che iniziano con la 'a'. Ad esempio la ["anno", "gio", "rno"] -> "giorno"

Definisci diverse versioni (in ordine di difficoltà)

- concat_for con un semplice ciclo for
- concat_foreach usando il foreach (e una funzione + var locale)
- concat_rec: usando la ricorsione
- concat_tail usando la tail recursion
- concat_filter usando anche filter
- concat_map usando anche map che converte le stringe che iniziano con 'a' in "" e poi le concatena tutte
- concat_reduce usando anche foldLeft o foldRight

Scrivi alcuni esempi con le seguenti liste e chiama le funzioni sopra definite:

["anno", "gio", "rno"] , ["", "ak", "mo"]

Se volessi generalizzare il metodo per filtrare non solo quelle che iniziano con la 'a' ma in generale quelle che soddisfano un certo criterio, come potrei usare le HOF? Ad esempio se volessi concatenare solo le stringhe con lunghezza == 10 come potrei fare con le HOF? Riscrivi un paio di metodi di cui sopra usando le higher order function e la funzione F opportuna. Usa il currying se riesci.