Informatica III / programmazione avanzata

Durata 2 h 07/02/2023

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 aggiungi file, NON upload file zip.

(RA 4 punti) – scritto pesa 40% del voto finale circa.

1. Funzione in C (5 punti)

Scrivi una funzione sum_product in C che dato in ingresso due array di interi a1 e a2, di lunghezza uguale, calcola la somma dei prodotti tra i due elementi di a1 e di a2 aventi uguale posizione.

di lunghezza diversa, considera quella più corta come se venisse allungata con degli spazi.

Per esempio
sum_product({2,3},{3,4}) → 2 * 3 + 3 * 4 = 18

Scrivi tre versioni:

- Una iterativa non ricorsiva
- Una ricorsiva senza tail call
- Una ricorsiva con tail call.

Scrivi anche un main di esempio in cui chiami le funzioni con qualche array.

Non usare alcuna variabile globale. Cerca di tenere il più semplice possibile la segnatura delle funzioni ricorsive, ma se non riesci fai una funzione con segnatura semplice che chiami quella ricorsiva.

2. Tipi opachi in C (6 punti)

Definisci il tipo opaco IntArray che rappresenta un array di interi a cui possono accodare altri array. La lunghezza dell'array deve aumentare se necessario e deve aumentare la memoria a disposizione).

costruttore: crea una IntArray e lo inizializza all'array di interi passato come argomento (facendone una copia).

accoda: prende un IntArray X e un altro array Y e accoda X a Y cambiandone lunghezza.

toString che restituisce una stringa (array di caratteri) che rappresenta l'IntArray. Puoi assumere che ogni intero occupi al massimo 10 caratteri. Il formato della stringa è del tipo "{1,2,3}".

cancella che distrugge l'IntArray e libera la memoria

Fai un esempio in cui:

- costruisci la IntArray con l'array {1,2,3} e lo stampi
- accodi l'array {4,5,6} e lo stampi
- cancelli l'array

3. inizializer list in C++ (2 punti)

A cosa serve la inizializer list in C++? Fai un piccolo esempio e spiega il tutto tramite commenti ad un esempio di codice. Ci sono alcuni casi in cui il suo uso è facoltativo (mostra le alternative nel caso) e invece casi in cui risulta essere necessario (mostra eventualmente un esempio).

4. Smart pointers in C++ (2 punti)

Fai un esempio in cui usi gli smart pointers (di diverso tipo se riesci) e fai vedere la differenza rispetto all'uso dei raw pointers (in particolare anche quando si passano come argomenti di funzioni).

5. ArrayT in Java (4 punti)

Implementa la struttura dati dell'esercizio 2 in java, che però sia generica in modo che si possa memorizzare ogni tipo che estenda Object in questo array variabile. Come sottostante NON usare un array list o altre collezioni, ma usa un array puro []. Implementa i metodi dell'esercizio 2.

Nel main fai un paio di esempi con interi o char.

6. Visitor pattern (4 punti)

Un Mobile ha un solo attributo che è il suo nome ed è Comparable rispetto ad esso. Un Tavolo e un Armadio sono due tipi di mobile. Per calcolare il prezzo di un mobile, invece di definire un metodo prezzo, voglio usare il visitor pattern (possibilmente generico rispetto il tipo restituito). Come faresti? Per semplicità assumo che il mobile abbia prezzo 100, un tavolo 200 e un armadio 300.

Se volessi scrivere un metodo statico che data una lista di oggetti Comparable generici, mi restituisce il maggiore, come lo implementeresti? Posso passargli anche una lista di Mobile? Anche una lista di Tavolo?

7. Programmazione funzionale (3 punti)

Salva anche questo in un cognome nome txt e allegalo alla consegna.

```
Scrivere in Haskell il codice ricorsivo di una funzione chiamata hth ("head to head")
che, date in input due liste, restituisce in output l'inversa della prima lista a cui
viene concatenata la seconda lista.
Es. hth [a b c] [1 2 3] = [c b a 1 2 3]
```