Programmazione Avanzata

Durata: 4h
Data: 08/01/2024

Chiama i tuoi progetti con MATRICOLA_ESX con X il numero dell'esercizio. Per esempio, 2574188_ES1 conterrà la soluzione dell'esercizio 1 dello studente con matricola 2574188.

Per la consegna crea uno zip unico caricare il file su Moodle nella cartella `` Programmazione Avanzata 2024-01-08''.

Punteggio Massimo: 26 PUNTI.

1. Funzione in C (4 punti)

Scrivi una funzione **mix_array** che dati in ingresso due array di interi array1 e array2, **restituisce** un array array3. Il contenuto delle celle di array3 in posizione pari corrisponde al contenuto delle celle di array1 nelle medesime posizioni. Il contenuto delle celle di array3 in posizione dispari corrisponde al doppio del contenuto delle celle di array2 nelle medesime posizioni.

Se un array è più lungo, la parte avanzata viene tagliata.
Ad esempio
mix_array([0 5 9 6 7], [9 5 7 6 2 5 8 7]) = [0 10 9 12 7]
mix_array ([8 5 1 2 9 6 5 3],[8 4 6 3 2 5 1]) = [8 8 1 6 9 10 5]

(a) Scrivi tre versioni:
- Una iterativa non ricorsiva
- Una ricorsiva senza tail call
- Una ricorsiva con tail call.

(b) Scrivi anche un main di esempio in cui chiami le funzioni con almeno gli array di cui sopra.

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. Record di Attivazione (4 punti)

Si consideri la seguente funzione:

```
int f(int a, int b) {
    if (b==0)
        return a;
    else
```

```
        return f(b,a%b);
}
```

Si mostrino i record di attivazione nello stack quando chiamata con a=15, b=10: si crei un file .txt composto come segue (si aggiungano righe al file se necessario

Label	Indirizzo	Contenuto	Note

3. Tipi opachi (3 punti)

Definisci il tipo opaco Set che rappresenta un insieme (non ordinato) di interi. Nota: un insieme non contiene elementi ripetuti. Definisci l'operazione di unione. La dimensione del set deve aumentare se necessario. Implementa il set utilizzando un array. Definisci eventuali funzioni addizionali se necessario. Nota: Lo studente è libero di mantenere memorizzati il set di interi a piacimento (eventualmente anche ordinati).

costruttore: crea una set e inserisce l'insieme di interi contenuti in un array passato come argomento (facendone una copia).
union: prende due set e restituisce un set contenente l'unione degli elementi.
print: stampa il set di interi.
cancella: distrugge il set e libera la memoria

Fai un esempio in cui:

- costruisci il set con gli elementi {1,2,3} e lo stampi
- Calcola l'unione dei set contenenti gli elementi {1,4,5,7} e {8,4,3,1}
- cancelli tutti i set.

4. Inizializer list in C++ (1.5 punti)

Definire una classe Motore con un attributo intero che rappresenta la Cilindrata. Definire una classe Automobile con un attributo di tipo Motore. Sviluppare tre implementazioni diverse della classe Automobile in cui l'attributo Motore è rispettivamente un (a) puntatore a una porzione di memoria nello stack, (b) puntatore a una porzione di memoria nell'Heap come puntatore, (c) reference a una porzione di memoria nell'Heap. Per ciascuno di questi tre casi discutere se (a) è possibile utilizzare l'Inizializer list e (b) è possibile inizializzarlo all'interno del costruttore.

5. Smart pointers in C++ (1.5 punti)

Definisci una classe Persona con un attributo privato nome di tipo string. Crea una persona utilizzando un raw pointer, uno smart pointer, e uno shared smart pointer. Illustra e spiega le differenze creando un metodo main e commentando opportunamente il codice.

6. Java Generics (4 punti)

Implementa la struttura dati "Set" dell'esercizio 3 in java, che però sia generica in modo che si possa memorizzare ogni tipo che estenda Object. Come sottostante NON usare un array list o altre collezioni, ma utilizza un array puro []. Implementa i metodi dell'esercizio 2. Crea un main in cui illustri l'utilizzo del Set creando un set di Integer e un set di caratteri.

7. Visitor pattern (4 punti)

Si vuole gestire il menu di un ristorante. Il ristorante offre varie pietanze. Ogni pietanza ha un quantitativo di calorie. Considerare due pietanze di esempio: le carote (carrot) e l'agnello (lamb). A differenza delle carote, l'agnello ha un peso ed è possibile selezionare la cottura mediante un opportuno metodo: l'agnello offre il metodo configuraCottura(String cotturaSelezionata). Le carote offrono un metodo configuraPrezzemolo che prende un booleano e permette di inserire o meno il prezzemolo nelle carote.

(a) Si vuole utilizzare il pattern Visitor per
    1) stampare il menu in varie lingue (considerare per esempio l'inglese e l'italiano) e
    2) stampare le calorie dei vari piatti

(b) Deve essere possibile ordinare le pietanze in ordine crescente in base alle loro calorie. Utilizzare l'interfaccia Comparable e mostrare un esempio nel quale viene creata una lista di pietanze e vengono ordinate in base alle calorie.

(c) Creare una Lista di elementi di tipo Agnello contenente Agnelli di peso differente. Discuti se è possibile assegnare questa lista ad un Lista di elementi di tipo Pietanza e presenta le relative motivazioni.

8. Programmazione funzionale (4 punti)

(a) Definire in Scala la funzione mult6 che dato un numero intero lo moltiplica per sei, e la funzione mult che dati due numeri interi li moltiplica.
(b) Definire la funzione magic che data una funzione f:Integer -> Integer e una funzione f: Integer X Integer -> Integer restituisce la funzione g(f(x),f(y)). Dati due numeri interi, utilizzare la funzione magic in combinazione con le funzioni definite in precedenza per moltiplicare i numeri interi per sei e poi moltiplicarli tra di loro.
(c) Definire una list di interi contenente gli interi 1,2,3. Moltiplicare ogni elemento della lista per due, utilizzando una funzione anonima
(d) Utilizzare map-reduce per sommare la lista dei valori ottenuti al punto (c).