



Programmazione Avanzata

Durata: 4h

Data: 18 Aprile 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 2024-04-18''.

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 alla somma delle celle di array1 e array2 nelle medesime posizioni. Il contenuto delle celle di array3 in posizione dispari corrisponde al doppio del contenuto delle celle di array2 nella posizione precedente.

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]) = [9 9 16 7 9]

mix_array([8 5 1 2 9 6 5 3],[8 4 6 3 2 5 1]) = [16 8 7 6 11 2 6]

(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.

Non usare variabili globali.

2. Record di Attivazione (4 punti)

Si consideri la seguente funzione:

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



```
        return a;  
    else {  
        c=a%b;  
        return f(b,c);  
    }  
}
```

Si mostrino i record di attivazione nello stack quando chiamata con a=9, b=6: 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 OrderedSet che rappresenta un insieme (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 aggiuntive se necessario. Nota: Lo studente è libero di mantenere memorizzati il set di interi a piacimento.

costruttore: crea un set ordinato 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 (ordinata) 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 Auto con un attributo che rappresenta la sua targa. Definire una classe Persona che ha nome, cognome, ed auto come attributi. Sviluppare tre implementazioni diverse della classe Persona in cui l'attributo Auto 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 iniziarlo all'interno del costruttore.

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

Considerare la classe Auto dell'esercizio 4. Crea una Persona utilizzando un raw pointer, uno smart pointer, e uno shared smart pointer. Ricordarsi che alla distruzione della Persona deve essere distrutta anche la relativa auto. Illustra e spiega le differenze relative all'utilizzo dei vari tipi di puntatori creando un metodo main e commentando opportunamente il codice.

6. Java Generics (4 punti)

Implementa la struttura dati "OrderedSet" 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 array[]. Implementa i metodi dell'esercizio 2. Crea un main in cui illustri l'utilizzo dell' OrderedSet creando un set di Integer e un set di caratteri.

7. Visitor pattern (4 punti)

Si vuole sviluppare una applicazione per gestire una pasticceria. Ci sono tre tipologie di dolci: torte, pasticcini, cornetti. Ogni dolce ha un quantitativo di calorie. A differenza delle torte e dei pasticcini i cornetti hanno un ripieno. È possibile settare il ripieno del cornetto mediante un opportuno metodo setRipieno(String ripieno). Le torte offrono un metodo configuraTipologia che prende un parametro di tipo enum che permette di settare la tipologia della torta.

- (a) Si vuole utilizzare il pattern Visitor per
- 1) stampare il menu della pasticceria in diverse lingue (considerare per esempio l'inglese e l'italiano) e
 - 2) stampare le calorie dei vari dolci



- (b) Deve essere possibile ordinare i dolci 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 Torta contenente Torte di peso differente. Discuti se è possibile assegnare questa lista ad un Lista di elementi di tipo Dolci e presenta le relative motivazioni.

8. Programmazione funzionale (4 punti)

- (a) Definire in Scala la funzione f che data una stringa restituisce un'altra stringa contenente i caratteri in posizione 3, 6 e 9 o la stringa vuota se la stringa passata come parametro è nulla o a meno di 9 caratteri.
- (b) Definire la funzione g che data una stringa come parametro restituisce la stringa ottenuta concatenando la stringa a se stessa.
- (c) Definire la funzione $h: \text{String} \rightarrow \text{String}$ che data una funzione $l: \text{String} \rightarrow \text{String}$ e una funzione $m: \text{String} \rightarrow \text{String}$ restituisce la funzione $m(l(x))$.
- (d) Definire la funzione "k" utilizzando la funzione "h" passandogli come input le funzioni "g" e "g" definite in precedenza. Utilizzare la funzione "h" sulla stringa "oloboslorat".
- (e) Definire una lista di stringhe contenente le stringhe "oloboslorat", "olocosiorat", "olomosaor!t". Applicare ad ogni elemento della lista la funzione "k". Invece di utilizzare la funzione "k" una funzione anonima
- (f) Utilizzare map-reduce per concatenare la lista delle stringhe ottenute al punto (e).