Programmazione Avanzata

Durata: 4h
Data: 05/04/2023

Chiama i tuoi progetti con MATRICOLA_ESX con X il numero dell'esercizio.

Per la consegna fai uno zip unico caricare il file su Moodle nella cartella `` Programmazione Avanzata XX-XX-XXXX''.

Punteggio Massimo: 26 PUNTI.

1. Funzione in C (4 punti)

Scrivi una funzione **mix_strings** che date in ingresso due stringhe A e B, **restituisce** una stringa che fa il mix di A e B mettendo un carattere di A e uno di B alternati. Se una stringa delle due è più lunga, la parte avanzata viene tagliata.
Ad esempio
mix_strings("cane","nero") = "cnaenreo"
mix_strings("ciao","marianna") = "cmiaaroi"
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 almeno le stringhe 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 N) {
    if (N<=0)
        return 5;
    else
        return F(N-1)+F(N-3);
}
```

Si mostrino i record di attivazione nello stack quando chiamata con N=3: 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 CircularString che rappresenta una stringa circolare di una data lunghezza fissa e con contenuto fisso (ma che può girare).
costruttore: crea una CircularString e la inizializza alla stringa passata come argomento (facendone una copia).
gira: prende una CircularString e fa scorrere tutti i caratteri di una posizione in avanti e il carattere in ultima posizione diventa il primo. Esempio: gira("pippo") = "opipp". La CircularString tiene anche memoria del numero di giri che sono stati applicati
toString che restituisce una stringa (array di caratteri) che rappresenta la CircularString
n_gira che restituisce il numero di volte che il gira è stato applicato
cancella che distrugge la CircularString e libera la memoria
Fai un esempio in cui:

- costruisci la CircularString "ciao"
- fai circolare la string un paio di volte e ogni volta stampi la stringa e il numero di gira
- cancelli la string

4. inizializer list in C++ (1.5 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).

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

Fai un esempio di una funzione **setA** che prende una string S (come puntatore a char) e setta la prima lettera di S a 'A'.
Fai una versione setA con i puntatori raw (char*) e tutte quelle che riesci con i puntatori smart.
Fai poi un main in cui chiami tutte le funzioni con stringhe di prova.

6. Java Generics (4 punti)

Usando i generics in Java, scrivi una classe **Dizionario** che rappresenta una mappa da un insieme di un tipo T (chiave) ad un altro tipo S (valore). Il primo tipo T è un comparable (cioè estende in modo opportuno Comparable). Come sottostante usa due arraylist (uno per le chiavi di tipo T e uno per i valori di tipo S tali che chiave e valore stiano nella stessa posizione nei due array. Devi definire i seguenti metodi:

- un metodo per inserire una relazione tra due valori (T e S).
- un metodo che restituisce il valore (S) per una certa chiave (T).

Scrivi un main in cui fai un po' di prove. Se riesci sfrutta il fatto che T sia comparable.
Se ho Persona che è Comparable, Studente estende Persona (ma non ha un suo metodo compareTo), si può fare una **Dizionario** di Studenti e loro voto intero in programmazione avanzata? Fai qualche prova.

7. Visitor pattern (4 punti)

Un Animale può essere un Pesce. Per vedere se un animale nuota oppure no, invece di usare un metodo, voglio usare il visitor pattern. Assumo che ogni pesce nuoti e che ogni animale che nuota è un pesce – cioè un animale che non è un pesce non nuota. Come faresti?
Se volessi scrivere un metodo statico che data una lista di animali, conta quanti nuotano, come lo implementeresti?
Se riesci, evita l'uso di instaceof

8. Programmazione funzionale (4 punti)

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