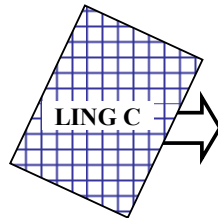
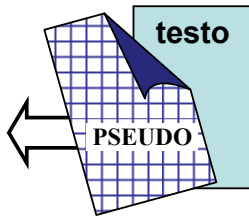


Algoritmi e Strutture di Dati – D.M. 270/04
Appello del 1° febbraio 2013 – 2 ore e 15 minuti
Libri e appunti chiusi

Ho bisogno di una correzione veloce in quanto _____

Cognome: _____ **Nome:** _____ **Matricola:** _____



**CONSEGNA PSEUDOCODIFICA E LINGUAGGIO C SU
DUE FOGLI PROTOCOLLO SEPARATI**

PUOI SCRIVERE A MATITA

LA BRUTTA LA PUOI TENERE

PSEUDOCODIFICA

Esercizio 1

Una lista L è un riferimento ad un oggetto che ha il solo campo $L.head$, che è un riferimento al primo elemento della lista. Ogni elemento di L ha i campi $prev$, $info$ e $next$.

Scrivi lo pseudocodice della procedura $MERGE-LIST(L1,L2)$ che prende in input due liste di interi $L1$ ed $L2$ ordinate in senso crescente e produce in output un array contenente tutti gli elementi di $L1$ ed $L2$ ordinati in senso crescente. Assumi che i valori delle liste $L1$ ed $L2$ siano tutti distinti e che non ci sia alcun valore in comune tra le liste.

Esercizio 2

Un grafo non orientato è rappresentato con un array A in cui ogni elemento $A[u]$ è un riferimento al primo elemento della lista di adiacenza doppiamente concatenata del nodo u (con i campi $prev$, $info$ e $next$). Essendo il grafo non orientato esiste un arco (u,v) per ogni arco (v,u) .

Una componente connessa è un insieme massimale di nodi mutualmente raggiungibili.

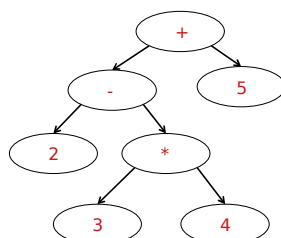
Scrivi lo pseudocodice della procedura $COMPONENTI-CONNESSE-BANALI(A)$ che prende in input un grafo non orientato A e produce in output il numero delle componenti connesse che hanno meno di tre nodi.

Esercizio 3

Discuti la complessità computazionale nel caso peggiore (in termini di O -grande, Ω e Θ) delle procedure che hai proposto per gli esercizi precedenti, utilizzando $v1$ e $v2$ per denotare il numero totale degli elementi delle liste $L1$ ed $L2$, n per denotare il numero dei nodi del grafo ed m per gli archi.

LINGUAGGIO C

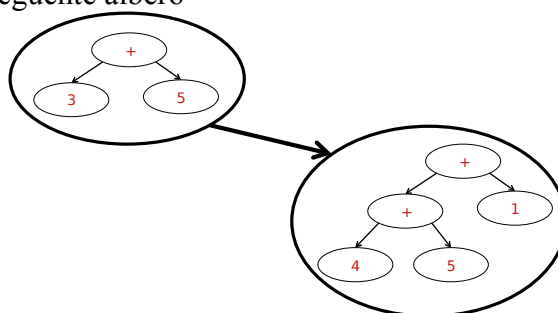
Si considerino espressioni aritmetiche (composizioni di numeri interi tramite gli operatori $[+,-,*,/]$) rappresentate da alberi binari la cui struttura è determinata dall'ordine con cui vengono svolte le operazioni dell'espressione, senza utilizzare le parentesi. Il valore numerico di una espressione aritmetica è un numero intero corrispondente alla sua risoluzione. Ad esempio l'espressione aritmetica $((2-(3*4))+5)$ ha associato il valore numerico -5 ed è rappresentata dal seguente albero binario:



Si vuole gestire una lista di espressioni aritmetiche (alberi binari). Si definisca un progetto in linguaggio C in cui siano implementate le seguenti funzionalità:

- 1) le strutture dati più adeguate per rappresentare la singola espressione aritmetica, la lista di espressioni aritmetiche ed eventuali altre strutturate dati di supporto, e i file **header** del progetto (**.h**)
- 2) la funzione **contExpr** che data la lista **L** di espressioni e un numero intero **v**, restituisca il numero di espressioni **x** in **L** che abbiano un valore numerico strettamente maggiore di **v**.
- 3) la funzione **createP_ABR** che data la lista **L** di espressioni, crei un albero binario di ricerca **T**, tale per cui ogni nodo di **T** contenga come info una espressione **x** di **L**, tale per cui il valore numerico di **x** sia un numero pari, e
 - a. il figlio destro di **n** contenga una espressione il cui valore numerico è maggiore di quello dell'espressione contenuta in **n**
 - b. il figlio sinistro di **n** contenga una espressione il cui valore numerico è minore di quello dell'espressione contenuta in **n**

Si può immaginare di passare un albero vuoto come parametro formale della funzione. Ad esempio si consideri la lista **L** composta dalle seguenti tre espressioni $((2-(3*4))+5)$, $3+5$ e $(4+5)+1$; otterremo il seguente albero



- 4) la funzione **createTextFileV** che dato l'albero binario di ricerca **T** creato al punto precedente e un numero intero **v**, stampi su di un file di testo "**divisori.txt**" tutte le espressioni contenute in **T** (una per riga) che abbiano un valore numerico divisibile da **v**. Si stampi di una espressione solamente numeri e operatori separati da spazi (ignorare le parentesi). Ad esempio considerando l'espressione $((2-(3*4))+5)$, su file la versione stampata sarà $2 - 3 * 4 + 5$.

E' possibile definire metodi di supporto e utilizzare ogni libreria conosciuta.