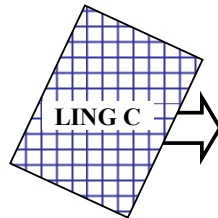
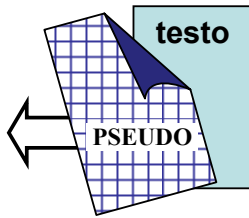


Algoritmi e Strutture di Dati – D.M. 270/04
Appello del 1° febbraio 2013 – 2 ore e 15 minuti
Libri e appunti chiusi

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 LISTA-ORDINATA(V) che prende in input un array V di interi e produce in output una lista ordinata in senso crescente con i valori di V . Puoi implementare qualsiasi strategia di ordinamento. Assumi che i valori dell'array V siano tutti distinti e che l'array V possa essere modificato.

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.

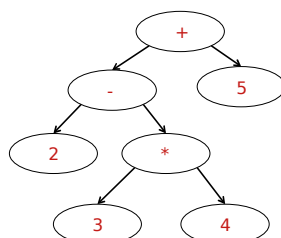
Scrivi lo pseudocodice della procedura COMPONENTI-CONNESSE-SIGNIFICATIVE(A) che prende in input un grafo non orientato A e produce in output il numero delle componenti connesse che hanno più di un nodo.

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 v per denotare il numero totale degli elementi dell'array, n per denotare il numero dei nodi del grafo ed m per gli archi.

CODIFICA C

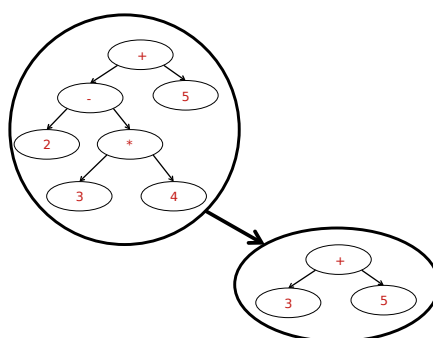
Si considerino espressioni aritmetiche (composizioni di numeri interi tramite gli operatori $[+, -, *, /]$) rappresentate da alberi binari la cui struttura è determinata dall'ordine con cui vengono svolte le operazioni dell'espressione, senza utilizzare le parentesi. Il valore numerico di una espressione 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 **insertExpr** che data la lista **L** di espressioni una espressione aritmetica **x**, inserisca **x** in **L** in coda (ultimo elemento).
- 3) la funzione **createABR** che data la lista **L** di espressioni crei un albero binario di ricerca **T**, tale per cui ogni nodo **n** di **T** contenga come info una espressione di **L**, 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 due espressioni $((2-(3*4))+5)$ e $3+5$; otterremo il seguente albero



- 4) la funzione **createTextFile** che dato l'albero binario di ricerca **T** creato al punto precedente e un numero intero **v**, stampi su di un file di testo **"espressioni.txt"** tutte le espressioni contenute in **T** (una per riga) che abbiano un valore numerico strettamente maggiore di **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.