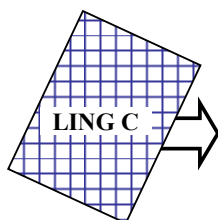
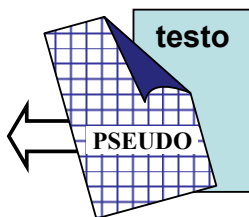


---

**Algoritmi e Strutture di Dati – A.A. 2012-2013– D.M 270/04**  
**Seconda prova intermedia del 1° febbraio 2013**  
**Libri e appunti chiusi – tempo: 1 ora e 15 minuti**

---

Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_



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

**PUOI SCRIVERE A MATITA**

**LA BRUTTA LA PUOI TENERE**

## **PSEUDOCODIFICA**

Negli esercizi seguenti 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  $(v,u)$ .

### **Esercizio 1**

Scrivi lo pseudocodice della procedura `CONTA-COMPONENTI-CONNESSE(A)` che prende in input un grafo non orientato  $A$  e produce in output il numero delle componenti connesse del grafo.

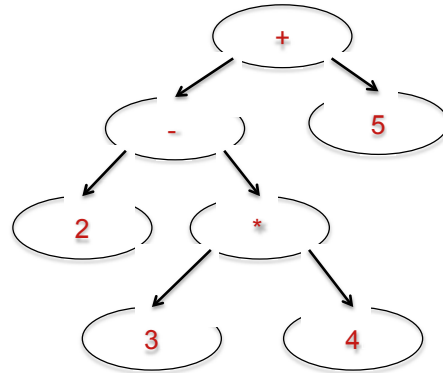
Una componente connessa è un insieme massimale di nodi mutualmente raggiungibili.

### **Esercizio 2**

Discuti la complessità computazionale nel caso peggiore (in termini di  $O$ -grande,  $\Omega$  e  $\Theta$ ) delle procedure che hai proposto per gli esercizi precedenti, utilizzando  $n$  per denotare il numero totale dei nodi del grafo ed  $m$  per gli archi.

## LINGUAGGIO C

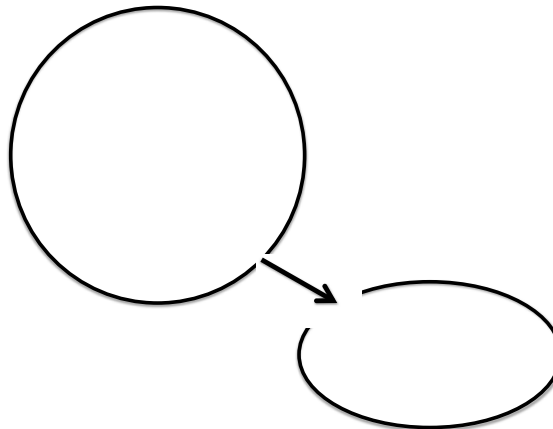
Si considerino espressioni aritmetiche (composizioni di numeri interi tramite gli operatori  $[+, -, *, /]$ ) rappresentate da alberi binari la cui struttura è determinata dall'ordine con cui vengono svolte le operazioni dell'espressione, senza utilizzare le parentesi. Il valore numerico di una espressione 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 implementino in linguaggio C le seguenti funzionalità:

- 1) le strutture dati più adeguate per rappresentare la singola espressione aritmetica, la lista di espressioni aritmetiche ed eventuali altre strutture dati di supporto
- 2) 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  $L$ , tale che:
  - a. il figlio destro di  $n$  contenga un'espressione il cui valore sia maggiore di quello dell'espressione contenuta in  $n$
  - b. il figlio sinistro di  $n$  contenga un'espressione il cui valore sia maggiore di quello dell'espressione contenuta in  $n$

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



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