

## Esercitazione 10

Prof. Riccardo Zese - [riccardo.zese@unife.it](mailto:riccardo.zese@unife.it)

Ing. Alice Bizzarri - [alice.bizzarri@unife.it](mailto:alice.bizzarri@unife.it)

Ing. Elisabetta Gentili - [elisabetta.gentili1@unife.it](mailto:elisabetta.gentili1@unife.it)

Laboratorio Fondamenti di Informatica - Modulo B



**DE** Department of  
Engineering  
Ferrara

## Esercitazione su Alberi I

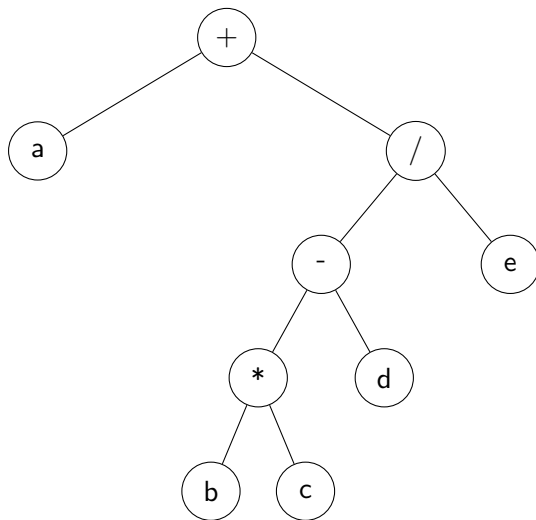
Si vuole creare un albero binario di ricerca (BST) contenente dei caratteri (`char`) in modo da poter eseguire funzioni su esso.

Dato l'albero in figura corrispondente all'espressione

$$a+(b*c - d)/e$$

realizzare le funzioni che permettano di costruirlo, di stamparlo e di calcolarne altezza e numero di nodi.

## Esercitazione su Alberi II



## Esercitazione su Alberi III

In particolare, implementare le funzioni:

- `consTree` per l'inserimento di un elemento nella radice dell'albero avente come sotto-alberi `r` (destro) e `l` (sinistro)
- `showTree` per la stampa preorder dell'albero
- `showTreePost` per la stampa postorder dell'albero
- `showTreeIn` per la stampa inorder dell'albero
- `height` per il calcolo dell'altezza dell'albero
- `nnodi` per il calcolo del numero di nodi dell'albero
- eventuali funzioni di supporto

Al netto delle parentesi, quale visita stampa l'espressione nell'ordine corretto?

## Esercitazione su Alberi IV

Le funzioni di manipolazione di alberi richieste, da inserire nei file `tree.c` e `tree.h` secondo le solite modalità, sono già state introdotte a lezione. Di seguito un rapido ripasso:

```
1  typedef struct nodo {
2      element value;
3      struct nodo *left, *right;
4  } NODO;
5
6  typedef NODO *tree;
7
8  tree consTree(char e, tree l, tree r);
9  void showTree(tree t);
10 void showTreePost(tree t);
11 void showTreeIn(tree t);
12 int nnodes(tree t);
13 int height(tree t);
14 // ...
```