



## Laborator 09

În **REPORT.txt** adăugați output-ul versiunii finale a programului.  
Dacă o parte din program nu e implementată, nu funcționează, face ca programul să dea seg fault atunci puteți comenta unele linii din main și să folosiți aceea afișare.

### Exerciții

Pentru un arbore binar de căutare implementați variantele iterative (NU recursive) a următoarelor funcții (schelet.c):

1. Afișarea în ordine descrescătoare a valorilor din arbore. `bstPrint()`
2. Inserarea unui nou nod în arbore. `bstInsert()`
3. Căutarea unui element într-un arbore (returnează 1 – găsit, sau 0). `bstSearch()`
4. Ștergerea unui nod din arbore. `bstDelete()`
5. Aflarea înălțimii unui arbore. `bstHeight()`

**Exercițiile de la 1 la 5 sunt obligatorii.** Conceptele explorate sunt esențiale pentru obținerea notei **minime** de promovare.

**Vă recomandăm, pentru a crește șansele de a obține o notă cât mai mare să explorați și următoarele exerciții:**

6. Inserarea unui nod în arbore de tip AVL.
7. Ștergerea unui nod din arbore de tip AVL.

### Exemplu afișare:

```
(5)
((3)5)
((3)5(7))
(((1)3)5(7))
(((1)3)5(7(9)))
(((1(2))3)5(7(9)))
(((1(2))3)5((6)7(9)))
*** Search node test: ***
Element 2 exists
Element 4 does NOT exist
*** Delete node test: ***
(((1)3)5((6)7(9)))
(((1)3)5((6)7(9)))
*** Get height test: ***
Inaltimea arborelui este 3.
```

În afișarea de mai sus fiecare subarbore (inclusiv de un nod) este înconjurat de paranteze. Este ideal să aveți aceleași paranteze, dar le puteți omite. În acest caz trebuie verificat cu Debugger ca arborele este construit corect.