



Laborator 10

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 acest laborator veți construi și folosi arbori binari ce reprezintă expresii matematice. Se numesc arbori de expresii și sunt folosiți în compilatoare.

Simplificare: În expresie pot apărea doar `+`, `-`, `*`, `/` și cifre (nu numere). Este mult mai simplu dacă implementați fiecare exercițiu recursiv (tratezi subarbore stâng, tratezi subarbore drept, combini rezultatul folosind informații despre nodul curent și întorci rezultatul pentru arborele părinte). Ca să obțineți dintr-un caracter `c` un număr de o cifră puteți scrie `c - '0'`.

Toate funcțiile se vor implementa **recursiv**.

1. Se citește dintr-un fișier o expresie matematică cu notatie Poloneză (formă prefixată). Să se creeze arborele binar aferent expresiei. `readTree()`
2. Implementați funcția pentru afișarea arborelui obținut anterior. Afișare identică cu fișierul de intrare. `printPolish()`
3. Implementați funcția pentru afișarea arborelui în formă matematică normală. Nu uitați de paranteze. `printNormal()`
4. Implementați funcția pentru evaluarea arborelui obținut anterior. Va returna rezultatul calculului. `evaluate()`
5. Implementați o funcție care să determine înălțimea arborelui. `getHeight()`
6. Implementați o funcție care să determine numărul de constante (cifre) (Hint: Numărul de frunze). `getLeafs()`
7. Implementați o funcție care să determine dacă 2 arbori sunt echivalenți (aceeași structură și aceleași valori ca noduri). `areEqual()`

Exercițiile de la 1 la 7 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:

8. Extindeți programul anterior permițând folosirea numerelor cu mai mult de o cifră.
9. Extindeți programul anterior pentru a putea face un simplu limbaj de calcul ce folosește variabile (litere). Fiecare rând va începe cu o literă, rezultatul calculului va fi atribuit acelei variabile. În arborii următori poate apărea acea variabilă. În momentul evaluării unui arbore care conține variabile, acestea vor fi înlocuite cu valorile calculate.



Exemplu afișare:

```
+ 3 * 2 - 9 5
```

```
(3+(2*(9-5)))
```

```
11
```

```
4
```

```
4
```

```
Arborele din expresie.txt este identic cu cel din expresie.txt
```

```
Arborele din expresie.txt este DIFERIT de cel din expresie1.txt
```