



## Laborator 11

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

Construiți **Trie**. Simplificare: fiecare nod va conține o literă. Rădăcina va conține caracterul ' ' - spațiu. Parcurgând arborele de la rădăcină la o frunză vom putea afișa un cuvânt. Un nod nu poate avea doi copii cu aceeași literă.

1. Implementați afișarea unui arbore în care fiecare nod poate să aibă oricâți copii. Recursiv, parcurgere în adâncime. `triePrint()`
2. Implementați introducerea unui cuvânt în trie. Atenție ca un nod să nu aibă 2 copii cu aceeași literă. `trieInsert()`
3. Implementați afișarea unui arbore de tip trie care să printeze pe fiecare linie câte un cuvânt introdus în trie. **Hint:** Puteți folosi un string pe care să îl trimiteți în recursivitate alături de mărimea sa. În acest string veți adăuga câte o literă, la final veți avea un cuvânt. `triePrintWords()`
4. Implementați căutarea unui cuvânt într-un trie. `trieSearch()`
5. Implementați o funcție care să returneze lungimea celui mai lung cuvânt din trie. `getSizeLongestWord()`

**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:**

Arbori **Shannon–Fano**. Se va lucra într-un nou fișier **.c**.

6. Implementați funcția care creează un arbore Shannon–Fano.
  - a. Dintr-un fișier se citesc caracterele.
  - b. Pentru fiecare caracter se numără câte apariții are în fișier (vector structură număr apariții și caracter).
  - c. Se sortează după numărul de apariții.
  - d. Se creează arborele Shannon–Fano împărțind vector în două bucăți în mod recursiv. Inițial rădăcina va avea tot vectorul, apoi cei 2 copii vor avea fiecare câte jumate, și tot așa până ce fiecare nod va avea maxim un caracter. Împărțirea se va face nu după numărul de elemente ci după suma probabilităților. (**Hint reprezentare:** căutare binară)
7. Implementați funcția care să afișeze un arbore Shannon–Fano.



### Exemplu afișare:

```
bob ina canc  
bob  
bobina  
bocanc
```

```
boca is NOT in the trie  
bob is in the trie  
bocanc is in the trie3
```

```
Longest Word has 6 characters
```