**Laborator 13**

**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**

1. Implementați citirea unui graf reținut ca o matrice de adiacență. **readGraphMatrix()**
2. Implementați parcurgerea **Deep First Search** a unui graf reținut ca o matrice de adiacență. Nodurile vor fi afișate în timpul parcurgerii. **dfs\_matrix()**
3. Implementați parcurgerea **Breadth First Search** a unui graf reținut ca o matrice de adiacență. Nodurile vor fi afișate în timpul parcurgerii. **bfs\_matrix()**
4. Implementați citirea unui graf reținut ca o listă de adiacență. **readGraphEdgeList()**
5. Implementați parcurgerea **Deep First Search** a unui graf reținut ca o listă de adiacență. Nodurile vor fi afișate în timpul parcurgerii. **dfs\_edges()**
6. Implementați parcurgerea **Breadth First Search** a unui graf reținut ca o listă de adiacență. Nodurile vor fi afișate în timpul parcurgerii. **bfs\_edges()**

**Exercițiile de la 1 la 6** 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:**

1. Implementați citirea unui graf reținut ca o matrice de adiacență. **readGraphVertex()**
2. Implementați parcurgerea **Deep First Search** a unui graf reținut ca o matrice de adiacență. Nodurile vor fi afișate în timpul parcurgerii. **dfs\_vertex()**
3. Implementați parcurgerea **Breadth First Search** a unui graf reținut ca o matrice de adiacență. Nodurile vor fi afișate în timpul parcurgerii. **bfs\_vertex()**
4. Implementați [BFS](https://www.infoarena.ro/problema/bfs) și [DFS](https://www.infoarena.ro/problema/dfs) de pe infoarena. Va trebui să faceți cont. În timpul laboratorului veți prezenta punctajul primit pe aceste rezolvări.

**Exemplu afișare:**

**0 1 3 2 4 5 7 8 9**

**0 1 9 3 4 7 8 2 5**

**0 1 3 2 4 5 7 8 9**

**0 1 9 3 4 7 8 2 5**

**0 1 3 2 4 5 7 8 9**

**0 1 9 3 4 7 8 2 5**