**Laborator 14**

**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. Pentru un graf memorat ca o matrice de adiacență determinați nodul cu grad maxim și afișați gradul acestuia. **printMaxDegree()**
2. Având un graf memorat ca o matrice de costuri implementați algoritul Floyd-Warshal pentru determinarea drumului minim între oricare două noduri. **floydWarshall()**
3. Având matricea de costuri generată mai devreme afișați [diametrul grafului](https://en.wikipedia.org/wiki/Distance_(graph_theory)). (diameter d, mai jos in pagină). **getGraphDiameter()**
4. Pentru același graf implementați algoritmul lui Dijsktra. **dijsktra()**

**Exercițiile de la 1 la 4** 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. Rezolvați algoritmul [Dijsktra](https://www.infoarena.ro/problema/dijkstra) pe infoarena.
2. Pentru același graf implementați algoritmul Kruskal. **kruskal()**

**Exemplu afișare:**

**Node 1 has maximum degree 3**

**0 1 3 2 3 4 999 5 6 6**

**1 0 2 1 2 3 999 4 5 5**

**3 2 0 1 1 1 999 2 3 3**

**2 1 1 0 2 2 999 3 4 4**

**3 2 1 2 0 2 999 3 4 4**

**4 3 1 2 2 0 999 1 2 2**

**999 999 999 999 999 999 0 999 999 999**

**5 4 2 3 3 1 999 0 1 1**

**6 5 3 4 4 2 999 1 0 1**

**6 5 3 4 4 2 999 1 1 0**

**Diameter of the graph is 6**

**Dijsktra rezult:**

**Node: 0 1 3 2 4 5 7 8 9 6**

**D: 0 1 2 3 3 4 5 6 6 999**

**Parent: -1 0 1 3 1 2 5 7 7 -1**

**Kruskal rezult:**

**0 -1-> 1**

**7 -1-> 9**

**1 -1-> 3**

**7 -1-> 8**

**2 -1-> 3**

**2 -1-> 4**

**2 -1-> 5**

**5 -1-> 7**