

Implementați o aplicație în limbajul C care implementează soluții la probleme de gestionare a unei rețele de calculatoare.

1. Scrieți secvența de cod sursă pentru crearea unei structuri de tip **Graf** implementată prin **Listă de Adiacență (Listă de Liste)**. Graful este orientat și ponderat, conține minim 6 hosturi și 13 muchii/arce. Hosturile și muchiile/arcele sunt preluate dintr-un fișier de intrare.
Structura **Host** se va defini astfel încât să conțină minim 5 câmpuri din care minim două câmpuri sunt definite ca variabile pointer. Cel puțin unul din cei doi pointeri este de tip **char**.

Cerințe de implementare:

- Definire structură **Host**. (0,25p)
- String-urile preluate din fișier trebuie să accepte prezența simbolului **blank**. (0,25p)
- Absență memory leaks. (0,25p)
- Implementare logică de creare structură **Listă de Adiacență** pentru un graf orientat și ponderat. (1,25p)
- Testare implementare, populare completă și corectă a structurii **Listă de Adiacență**. (1,00p)

2. Scrieți și apelați funcția pentru determinarea arcelor din structura creată la punctul 1) și a căror pondere este mai mare decât o valoare specificată prin parametru al funcției. Arcele identificate sunt salvate într-un vector ca elemente de tipul (**varf_src**, **varf_dst**), unde **varf_src** este vârf sursă al arcului, iar **varf_dst** este cel destinație al arcului. Vectorul se returnează în **main()** prin tipul de retur sau lista de parametri ai funcției.

Cerințe de implementare:

- Definire funcție cu parametri de I/O definiți complet și corect. (0,25p)
- Alocare dinamică a vectorului de arce. (0,25p)
- Implementare logică de determinare a arcelor. (1,00p)
- Testare implementare, populare completă și corectă a vectorului. (0,50p)

3. Scrieți și apelați funcția pentru determinarea host-urilor din structura creată la punctul 1) și care îndeplinesc un criteriu specificat (câmp structură **Host**) prin parametru de intrare al funcției. Host-urile identificate sunt salvate într-un vector care **NU** partajează zone de memorie heap cu structura **Listă de Adiacență**. Vectorul se returnează în **main()** prin tipul de retur sau lista de parametri ai funcției.

Cerințe de implementare:

- Definire funcție cu parametri de I/O definiți complet și corect. (0,25p)
- Alocare dinamică a vectorului de host-uri. (0,25p)
- Realizare deep-copy a departamentelor în vectorul rezultat. (0,25p)
- Implementare logică de determinare a host-urilor fără partajare de zone de memorie heap. (0,75p)
- Testare implementare, populare completă și corectă a vectorului. (0,50p)

4. Scrieți și apelați funcția pentru eliminarea unor host-uri din vectorul obținut la punctul 3). Condiția de eliminare este ca cel puțin un arc incident spre exteriorul host-ului să aibă pondere minimă în graful stocat prin **Listă de Adiacență** creată la punctul 1). Vectorul se returnează în **main()** prin tipul de retur sau lista de parametri ai funcției.

Cerințe de implementare:

- Definire funcție cu parametri de I/O definiți complet și corect. (0,25p)
- Realocare dinamică a vectorului de host-uri. (0,25p)
- Absență memory leaks. (0,25p)
- Implementare logică de eliminare a host-urilor din vectorul de intrare. (0,75p)
- Testare implementare, populare completă și corectă a vectorului. (0,50p)

5. Scrieți și apelați funcțiile care dezalocă structurile **Listă de Adiacență (Listă de Liste)** și **Vectori**, precum și toate structurile auxiliare utilizate în implementarea cerințelor (dacă este cazul).

Cerințe de implementare:

- Definire funcții cu parametri de I/O definiți complet și corect. (0,15p)
- Absență memory leaks. (0,15p)
- Actualizare variabile de gestionare a structurilor în funcția **main()**. (0,20p)
- Implementare logică de dezalocare a structurilor de date. (0,30p)
- Testare implementare, dezalocare completă și corectă a structurilor. (0,20p)
- Absență dezalocări structuri auxiliare utilizate. (-0,20p)

MENTIUNI:

- Proiectele cu erori de compilare nu vor fi evaluate.
- Implementările nu trebuie să conțină variabile definite la nivel global sau statice.
- Implementările plagiate vor fi evaluate cu 0 puncte, indiferent de sursă.
- Toate cerințele trebuie apelate și demonstrate în funcția **main()** pentru a fi evaluate.
- Art. 72 (1) Pentru următoarele fapte, studenții vor fi exmatriculați fără drept de reinmatriculare în Academia de Studii Economice din București:
 - (c) încercarea de promovare prin fraudă a examenelor sau a altor evaluări;