

Implementați o aplicație în limbajul C ce rezolvă probleme de gestionare a autobuzelor dintr-un parc auto.

1. Scrieți secvența de cod sursă pentru crearea unei structuri de tip **Lista Dublu Înlanțuită** ce conține date aferente autobuzelor din cadrul parcului auto. Inserarea în listă o realizați astfel încât elementele în cadrul listei să fie sortate crescător după un atribut al structurii Autobuz.

Structura **Autobuz** se va defini, astfel încât, să conțină minim 5 câmpuri, dintre care minim două sunt declarate ca pointeri la tipuri diferite.

Lista dublu înlanțuită va conține datele a cel puțin 10 autobuze care se preiau ca input dintr-un **fișier text**. În fișierul text autobuzele nu sunt sortate. **(2p)**

2. Scrieți și apelați funcția pentru ștergerea unui Autobuz din **Lista Dublu Înlanțuită**. Modul de alegere a nodului se face după un atribut al structurii Autobuz. Pentru verificare, structura este afișată înainte și după modificare. **(2p)**
3. Scrieți secvența de cod care copiază o parte dintre autobuzele din **Lista Dublu Înlanțuită** creată anterior într-o structură **Arbore Binar de Căutare**, cheia fiind unul dintre atributele autobuzului ce se considera a fi unic pentru fiecare autobuz (alt atribut față de cel ales pentru inserarea în listă). Cele două structuri de date **NU** partajează zone de memorie heap. **(3p)**
4. Scrieți și apelați funcția pentru determinarea numărului de noduri care au atât fiu stânga cât și fiu dreapta. **(2p)**
5. Scrieți secvența de cod care dezalocă structurile **Arbore binar de căutare**, **Lista dublu înlanțuită** și alte structuri suport create la punctele anterioare. **(1p)**

#### MENTIUNI:

- Proiectele cu erori de compilare nu vor fi evaluate.
- Implementările plagiate vor fi evaluate cu 0 puncte, indiferent de sursă.
- Toate cerintele trebuie apelate si demonstrate in functia main() pentru a fi evaluate.