

Implementați o aplicație în limbajul C ce rezolvă probleme de gestionare a examenelor la nivelul unei universități.

1. Scrieți secvența de cod sursă pentru crearea unei structuri de tip **Tabelă de Dispersie** ce conține date aferente unor examene. Cheia de căutare un câmp al structurii **Examen**, iar mecanismul de tratare a coliziunilor este **Linear Probing**. Inserarea unui examen se implementează într-o funcție care se apelează în secvența de creare a structurii **Tabelă de Dispersie**.

Structura **Examen** se va defini astfel încât să conțină minim 5 câmpuri, din care minim două sunt declarate ca pointeri.

Tabela de dispersie va conține datele a cel puțin 10 examene care se preiau ca input dintr-un fișier text. **(2p)**

2. Scrieți și apelați funcția pentru copierea într-o listă simplă a examenelor din **Tabela de Dispersie** care îndeplinesc un criteriu specificat. Criteriul este un câmp din **Examen** și nu poate fi cheia de căutare. **Lista simplă NU** partajează zone de memorie heap cu **Tabela de Dispersie**. **(2p)**
3. Scrieți și apelați funcția pentru determinarea clusterelor de coliziune din **Tabela de Dispersie**. Un cluster de coliziune se identifică prin perechea **(index1 vector suport, index2 vector suport)**. Perechile de cluster de coliziune se salvează într-un vector. **(3p)**
4. Scrieți și apelați funcția pentru realocarea tabelii de dispersie pe o lungime de 2 ori mai mare față de lungimea curentă. **(2p)**
5. Scrieți secvența de cod care dezalocă structurile **Tabelă de Dispersie**, **Listă Simplă**, **Vector** 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.