

Implementați o aplicație în limbajul C care implementează soluții la probleme colaborative din cadrul unei echipe de dezvoltare software.

1. Scrieți secvența de cod sursă pentru crearea unei structuri de tip **Arbore AVL** care reprezintă structura de mapare pe task-uri a unui proiect software. Cheia de inserare se alege din lista de câmpuri a structurii **Task**. Inserarea unui task se implementează în funcția principală pentru un număr de minim 10 elemente citite dintr-un fisier de intrare. **(2p)**

Structura **Task** se va defini astfel încât să conțină minim 5 câmpuri din care minim două câmpuri sunt definite ca variabile pointer.

2. Scrieți și apelați funcția pentru „spargerea” structurii creată la punctul 1) în doi arbori binari de căutare. Nodul în care are loc „spargerea” se identifică pe baza unui câmp din structura **Task**. După spargere, se calculează înălțimile celor doi arbori rezultați. **(2p)**
3. Scrieți secvența de cod care copiază task-urile din unul din cei doi arbori creați la cerința 2) într-o structură **Lista Dublă**. Inserarea task-urilor în **Lista Dublă** are loc astfel încât să se păstreze nodurile sortate după un câmp al structurii **Task** (diferit față de cel utilizat la punctul 2)). Cele două structuri de date **NU** partajează zone de memorie heap. **(3p)**
4. Scrieți și apelați funcția pentru extragerea într-un vector a tuturor task-urilor din **Lista Dublă** care îndeplinesc un anumit criteriu. Criteriul utilizat este diferit față de cele utilizate anterior (punctele 2) și 3)) **(2p)**
5. Scrieți secvența de cod care dezalocă structurile **Arbori binari de căutare**, **Lista Dublă**, **Vector** și toate structurile auxiliare utilizate în implementarea cerințelor (dacă este necesar). **(1p)**

MENTIUNI:

- Proiectele cu erori de compilare nu vor fi evaluate.
- Implementările plagiate vor fi evaluate cu 0 puncte, indiferent de sursă.
- Toate cerințele trebuie apelate și demonstrate în funcția main() pentru a fi evaluate.