

Implementați o aplicație în limbajul C ce rezolvă probleme de gestionare a facturilor de platit la nivelul unei gospodării.

1. Scrieți secvența de cod sursă pentru crearea unei structuri de tip **Heap (coadă de priorități)** ce conține date aferente unor facturi. Inserarea unei noi facturi se implementează printr-o funcție care se apelează în secvența de adăugare elemente în structura **Heap**.

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

Coadă de priorități va conține datele a cel puțin 10 facturi care se preiau ca input dintr-un fișier text, prioritatea fiind dată de numărul de zile rămase până la scadență. **(2p)**

2. Scrieți și apelați funcția pentru eliminarea tuturor facturilor din structura **Heap** care mai au mai puțin de trei zile până la scadență. Pentru verificare, structura **Heap** este afișată înainte și după stergere. **(2p)**
3. Scrieți secvența de cod care copiază o parte dintre facturile din structura **Heap** creată anterior într-o structură **Arbore Binar de Căutare**. Filtrarea facturilor copiate se realizează pe baza unui câmp definit în structura **Factura**. Cele două structuri de date **NU** partajează zone de memorie heap. **(3p)**
4. Scrieți și apelați funcția pentru salvarea într-o **Listă simplu înlănțuită** a nodurilor din **Arborele Binar de Căutare** plasate de la rădăcina până la un anumit nod identificat pe baza cheii de căutare a arborelui din structura **Factura**. **(2p)**
5. Scrieți secvența de cod care dezalocă structura **Heap**, **Arborele Binar de Căutare** și **Listă simplu înlănțuită** 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.