

Implementați o aplicație în limbajul C care implementează soluții la probleme de gestionare a rezervarilor la o sala de spectacole.

1. Scrieți secvența de cod sursă pentru crearea unei structuri de date de tip **Arbore Binar de Cautare**, în continuare **ABC**, ce conține date aferente rezervarilor de tip **Spectacol**. Cheia de căutare utilizată este atributul **idSpectacol**, iar articolul este compus din următoarele atribute obligatorii **pretBilet(float)**, **numeClient(char\*)**, **dataSustinere(char[10])**, precum și din alte 2 atribute definite la alegere. Inserarea unei rezervari se implementează într-o funcție care se apelează în secvența de creare a structurii **ABC**. Structura conține minim 10 înregistrări încărcate în aplicație dintr-un fișier de intrare. (2p)

Cerințe de implementare:

- Definire structură **Spectacol**. (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 **ABC**. (0,75p)
- Populare completă și corectă a structurii **ABC** cu date de intrare din fișier. (0,25p)
- Testare implementare cu afisarea la consola a conținutului structurii **ABC**. (0,25p)

2. Scrieți și apelați funcția pentru determinarea rezervarilor din structura creată la cerința 1) care se desfășoară la aceeași dată specificată ca parametru de intrare al funcției. Rezervările identificate sunt salvate într-un vector și **NU** partajează zone de memorie heap cu structura **ABC**. Vectorul se returnează în **main()** prin tipul de retur sau lista de parametri ai funcției. (2p)

Cerințe de implementare:

- Definire funcție cu parametri de I/O definiți complet și corect. (0,25p)
- Realizare deep-copy a rezervarilor în vector. (0,25p)
- Implementare logică de determinare și salvare a rezervarilor în vector. (1,00p)
- Populare completă și corectă a vectorului. (0,25p)
- Testare implementare prin apel de funcție și afisare la consola a rezultatului obținut după apel. (0,25p)

3. Scrieți și apelați funcția pentru determinarea **costului total aferent fiecărui client** pentru **toate rezervările** atribuite. Valorile identificate sunt salvate într-un vector în care fiecare element conține perechea de valori (**numeClient**, **costTotal**). Vectorul și dimensiunea acestuia se returnează în **main()** prin tipul de retur sau lista de parametri ai funcției. (2,5p)

Cerințe de implementare:

- Definire funcție cu parametri de I/O definiți complet și corect. (0,25p)
- Determinare dimensiune masiv. (0,25p)
- Implementare logică de determinare clienți și cost total rezervări. (1,5p)
- Populare completă și corectă a vectorului. (0,25p)
- Testare implementare prin apel de funcție și afisare la consola a rezultatului obținut după apel. (0,25p)

4. Scrieți și apelați funcția pentru transformarea structurii **ABC** de la cerința 1) în două structuri complementare de tip arbore binar de căutare, pe baza unuia din campurile optionale definite la alegere. Campul utilizat trebuie să aibă semnificație binară pentru a putea crea cei doi arbori binari de căutare complementari. Structurile rezultate se returnează în **main()** prin tipul de retur sau lista de parametri ai funcției iar conținutul acestora este afisat la consola. (2,5p)

Cerințe de implementare:

- Definire funcție cu parametri de I/O definiți complet și corect. (0,25p)
- Determinare elemente pentru inserarea în structurile complementare. (0,25p)
- Implementare logică de creare arbori binari de căutare **fără noi alocări de memorie**. (1,25p)
- Populare completă și corectă a structurilor. (0,5p)
- Testare implementare prin apel de funcție și afisare la consola a rezultatului obținut după apel. (0,25p)

5. Scrieți și apelați funcțiile care dezalocă structurile **2xABC** și **2xVectori** precum și toate structurile auxiliare utilizate în implementarea cerințelor (dacă este cazul). (1p)

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 prin apel de funcții și afisare la consola a rezultatelor obținute la apel. (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 variabile statice.
- Implementările nu trebuie să conțină structuri predefinite (ex STL, 3rd party libraries etc).
- 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 reînmatriculare în Academia de Studii Economice din București:
  - (c) încercarea de promovare prin fraudă a examenelor sau a altor evaluări;