

Implementați o aplicație în limbajul C care implementează soluții la probleme de gestionare a sejururilor turistice de către o agentie de turism.

1. Scrieți secvența de cod sursă pentru crearea unei structuri de date de tip **Lista Dublu Inlantuita** ce conține date aferente sejururilor turistice. Inserarea unui sejur turistic se implementează într-o funcție care se apelează în secvența de creare a structurii **Lista Dublu Inlantuita**. Lista dubla conține minim 10 sejururi turistice incarcate in aplicatie dintr-un fisier cu date de intrare.

Structura **SejurTuristic** se va defini astfel încât să conțină minim 7 câmpuri, astfel: **destinatie (char*)**, **suma_plata (float)**, **perioada_sejur** exprimata ca date calendaristice de inceput si de sfarsit ale sejurului; celelalte 4 campuri sunt definite la alegere, din care minim unul este de tip **char***.

Cerințe de implementare:

- Definire structură **SejurTuristic**. (0,25p)
- Definire funcție de inserare nod cu parametri de I/O definiți complet și corect. (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 **Lista Dubla**. (0,25p)
- Gestionare structura **Lista Dubla** cu adresele primului, respectiv ultimului nod. (0,25p)
- Populare completă și corectă a structurii **Lista Dubla** cu date de intrare din fisier. (0,25p)
- Testare implementare cu afisarea la consola a continutului structurii **Lista Dubla** prin traversarea in ambele sensuri. (0,25p)

2. Scrieți și apelați funcția pentru determinarea sejururilor turistice din structura creată la cerinta 1) a caror perioada de rezervare este de exact 7 zile. Sejururile identificate sunt salvate într-un vector și **NU** partajează zone de memorie heap cu structura **Lista Dubla**. Vectorul si dimensiunea acestuia se returnează în **main()** prin tipul de retur si/sau lista de parametri ai funcției.

Cerințe de implementare:

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

3. Scrieți și apelați funcția pentru gruparea sejururilor pe destinatii prin crearea unei structuri **Lista de liste**. Un nod din lista principala contine camp de tip **char*** aferent unei destinatii. O lista secundara contine sejururi turistice rezervate pentru aceeasi destinatie. **Lista de liste NU** partajeaza zone de memorie heap cu structura **Lista Dubla**. Lista de liste se returnează în **main()** prin tipul de retur sau lista de parametri ai funcției.

Cerințe de implementare:

- Definire funcție cu parametri de I/O definiți complet și corect. (0,25p)
- Realizare deep-copy a sejururilor în lista de liste. (0,25p)
- Implementare operatie de inserare nod in lista principala a structurii lista de liste. (0,25p)
- Implementare logică de creare structura lista de liste. (1,50p)
- Populare completă și corectă a structurii lista de liste. (0,25p)
- Testare implementare prin apel de functie si afisare la consola a rezultatului obtinut la apel. (0,25p)

4. Scrieți și apelați funcția pentru determinarea valorii totale a incasarilor pentru sejururile salvate in structura **Lista Dubla** creata la cerinta 1). Se iau in considerare sejururile cu o anumita destinatie si rezervate intr-o perioada de timp specificata. Destinatia si perioada de timp (date calendaristice de inceput si sfarsit sejur) sunt specificate in lista de parametri de intrare ai functiei. Valoarea totala a incasarilor se returnează în **main()** prin tipul de retur al funcției.

Cerințe de implementare:

- Definire funcție cu parametri de I/O definiți complet și corect. (0,25p)
- Filtrare sejur turistic pe perioada de timp specificata. (1,00p)
- Implementare logică de calcul a valorii totale a incasarilor. (0,75p)
- Testare implementare prin apel de functie si afisare la consola a rezultatului obtinut la apel. (0,25p)

5. Scrieți și apelați funcțiile care dezalocă structurile **Lista Dubla**, **Vector** si **Lista de Liste** precum și toate structurile auxiliare utilizate în implementarea cerințelor (dacă este cazul).

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 functii si afisare la consola a rezultatelor obtinute 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;