

Implementați o aplicație în limbajul C care implementează soluții la probleme de gestionare a resursei umane din cadrul unei companii.

1. Scrieți secvența de cod sursă pentru crearea unei structuri de date de tip **Lista de liste** ce conține date aferente angajaților companiei, grupați pe departamente. Inserarea unui angajat se implementează într-o funcție care se apelează în secvența de creare a structurii **Lista de liste**. Structura **Lista de liste** conține minim 10 angajați încarcăți în aplicație dintr-un fișier de intrare.

Structura **Angajat** se va defini astfel încât să conțină minim 6 câmpuri, astfel: **cod_departament (unsigned int)**, iar celelalte 5 câmpuri sunt definite la alegere, din care minim unul este de tip **char***.

Cerințe de implementare:

- Definire structură **Angajat**. (0,25p)
- String-urile preluate din fișier trebuie să accepte prezența simbolului **blank**. (0,25p)
- Absență memory leaks. (0,25p)
- Alocare corectă a nodurilor aferente structurii **Lista de liste**. (0,25p)
- Implementare logică de creare structură **Lista de liste**. (0,50p)
- Populare completă și corectă a structurii **Lista de liste** cu date de intrare din fișier. (0,25p)
- Testare implementare cu afisarea la consola a conținutului structurii **Lista de liste**. (0,25p)

2. Scrieți și apelați funcția pentru determinarea angajaților din structura creată la cerința 1) care au un salariu peste un prag specificat ca parametru de intrare al funcției. Angajații identificați sunt salvați într-un vector alocat dinamic și **NU** partajează zone de memorie heap cu structura **Lista de liste**. Vectorul 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 angajaților în vectorul alocat dinamic. (0,25p)
- Implementare logică de determinare și salvare a angajaților 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 la apel. (0,25p)

3. Scrieți și apelați funcția pentru determinarea angajaților dintr-un departament specificat ca parametru. Angajații identificați sunt salvați într-o coadă, care 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)
- Definire funcție inserare în coadă. (0,25p)
- Utilizare pointeri prim și ultim specifici structurii de tip coadă. (0,25p)
- Implementare logică de determinare și salvare a angajaților în coadă. (1,00p)
- Populare completă și corectă a structurii de tip coadă. (0,25p)
- Testare implementare prin apel de funcție și afisare la consola a rezultatului obținut (consultare fără extragere). (0,25p)

4. Scrieți și apelați funcția pentru determinarea angajaților cu cele mai mici 3 salarii. Se iau în considerare angajații salvați în structura de tip coadă de la cerința 3). Angajații cu cele mai mici 3 salarii se vor salva într-un vector diferit și **NU** partajează zone de memorie heap cu structura **coada**. Vectorul cu rezultate 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 angajaților în vectorul cu salarii minime. (0,25p)
- Determinare angajați cu cele mai mici 3 salarii. (0,50p)
- Implementare logică de creare vector cu angajați cu **salariu minim**. (1,25p)
- Populare completă și corectă a vectorului de angajați cu **salariu minim**. (0,25p)
- Testare implementare prin apel de funcție și afisare la consola a rezultatului obținut la apel. (0,25p)

5. Scrieți și apelați funcțiile care dezalocă structurile **Lista de liste**, **Coadă** și **Vector** 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 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 statice.
- 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;