

Implementați o aplicație în limbajul C care implementează soluții la probleme de gestionare a stocurilor unei companii de distribuție produse în comerțul retail.

1. Scrieți secvența de cod sursă pentru crearea unei structuri de date de tip **Tabelă de Dispersie** ce conține date aferente produselor distribuite. Cheia de căutare utilizată este **numar\_magazine**, iar mecanismul de tratare a coliziunilor este **Linear Probing**. Inserarea unui produs se implementează într-o funcție care se apelează în secvența de creare a structurii **Tabelă de Dispersie**. Tabela de dispersie conține minim 10 produse încarcate în aplicație dintr-un fișier de intrare.

Structura **Produs** se va defini astfel încât să conțină minim 7 câmpuri, astfel: **numar\_magazine (unsigned char)**, **stocuri\_magazine (float\*)**; celelalte 5 câmpuri sunt definite la alegere, din care minim unul este de tip **char\***.

Cerințe de implementare:

- Definire structură **Produs**. (0,25p)
- String-urile preluate din fișier trebuie să accepte prezența simbolului **blank**. (0,25p)
- Absență memory leaks. (0,25p)
- Realocarea vectorului suport la atingerea capacității maxime. (0,25p)
- Implementare logică de creare structură **Tabelă de Dispersie** cu **Linear Probing**. (0,50p)
- Populare completă și corectă a structurii **Tabelă de Dispersie** cu date de intrare din fișier. (0,25p)
- Testare implementare cu afisarea la consola a conținutului structurii **Tabelă de Dispersie**. (0,25p)

2. Scrieți și apelați funcția pentru determinarea produselor din structura creată la cerința 1) care sunt prezente într-un număr de magazine peste un prag specificat ca parametru de intrare al funcției. Produsele identificate sunt salvate într-o listă simplă și **NU** partajează zone de memorie heap cu structura **Tabelă de Dispersie**. Lista simplă 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 produselor în lista simplă. (0,25p)
- Implementare logică de determinare și salvare a produselor în lista simplă. (1,00p)
- Populare completă și corectă a listei simple. (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 numărului și a dimensiunilor (exprimate ca număr de produse) pentru toate cluster-ele de coliziuni din **Tabela de Dispersie**. Cluster-ele identificate sunt salvate într-un vector în care fiecare element conține dimensiunea unui cluster. Vectorul și dimensiunea acestuia 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)
- Determinare număr cluster-e. (0,25p)
- Determinare dimensiuni cluster-e. (0,25p)
- Implementare logică de determinare și salvare a cluster-elor 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)

4. Scrieți și apelați funcția pentru determinarea produselor cu cele mai mici 3 valori ale stocurilor. Se iau în considerare produsele salvate în lista simplă de la cerința 2). Produsele al căror stoc se încadrează în cele mai mici 3 valori se vor salva într-o listă simplă diferită și **NU** partajează zone de memorie heap cu structura **Lista Simplă** sursa conform cerința 2). Lista simplă 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 produselor în lista simplă cu stocuri minime. (0,25p)
- Determinare cele mai mici 3 valori ale stocurilor. (0,50p)
- Implementare logică de creare listă simplă cu produse în categoria **stoc minim**. (1,25p)
- Populare completă și corectă a listei simple cu produse în categoria **stoc 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 **Tabelă de Dispersie**, **Liste Simple** 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 dezașocări 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;