Bilet #4 / (150 min) | **11.06.2021**

Implementaţi o aplicaţie în limbajul C care implementează soluţii la probleme de gestionare a a conturilor bancare deschise la o bancă comercială.

1. Scrieţi secvenţa de cod sursă pentru crearea unei structuri de date de tip ***Tabelă de Dispersie*** ce conţine date aferente conturilor bancare***.*** Cheia de căutare utilizată este ***nume_client***, iar mecanismul de tratare a coliziunilor este ***Chaining***. Inserarea unui cont bancar se implementează într-o funcţie care se apelează în secvenţa de creare a structurii ***Tabelă de Dispersie***. Tabela de dispersie conţine minim 10 conturi bancare incarcate in aplicatie dintr-un fisier de intrare.
   Structura ***ContBancar*** se va defini astfel încât să conţină minim 7 câmpuri, astfel: ***nume_client (char*)***, ***sold (float), valuta_cont (char*)***; celelalte 4 campuri sunt definite la alegere.

   Cerinţe de implementare:
   - Definire structură ***ContBancar***. ***(0,25p)***
   - String-urile preluate din fişier trebuie să accepte prezenţa simbolului ***blank***. ***(0,25p)***
   - Absenţă memory leaks. ***(0,25p)***
   - Implementare logică de creare structură ***Tabelă de Dispersie*** cu ***Chaining***. ***(0,75p)***
   - Populare completă şi corectă a structurii ***Tabelă de Dispersie*** cu date de intrare din fisier. ***(0,25p)***
   - Testare implementare cu afisarea la consola a continutului structurii ***Tabelă de Dispersie***. ***(0,25p)***

2. Scrieţi şi apelaţi funcţia pentru determinarea conturilor bancare din structura creată la cerinta 1) care au valuta specificata ca parametru de intrare al functiei. Conturile bancare identificate sunt salvate într-un vector şi ***NU*** partajează zone de memorie heap cu structura ***Tabelă de Dispersie***. Vectorul se returnează în ***main()*** prin tipul de retur sau lista de parametri ai funcţiei.

   Cerinţe de implementare:
   - Definire funcţie cu parametri de I/O definiţi complet şi corect. ***(0,25p)***
   - Realizare deep-copy a conturilor bancare în vector. ***(0,25p)***
   - Implementare logică de determinare şi salvare a conturilor în vector. ***(1,00p)***
   - Populare completă şi corectă a vectorului. ***(0,25p)***
   - Testare implementare prin apel de functie si afisare la consola a rezultatului obtinut la apel. ***(0,25p)***

3. Scrieţi şi apelaţi funcţia pentru determinarea numarului si a dimensiunilor (exprimate ca numar de conturi bancare) pentru toate cluster-ele de coliziuni din ***Tabela de Dispersie***. Cluster-ele identificate sunt salvate într-un vector in care fiecare element contine perechea de valori ***(dimensiune_cluster, index_tabela).*** Vectorul si dimensiunea acestuia se returnează în ***main()*** prin tipul de retur sau lista de parametri ai funcţiei.

   Cerinţe de implementare:
   - Definire funcţie cu parametri de I/O definiţi complet şi corect. ***(0,25p)***
   - Determinare numar cluster-e. ***(0,25p)***
   - Determinare dimensiuni si indecsi cluster-e. ***(0,25p)***
   - Implementare logică de determinare şi salvare a cluster-elor în vector. ***(1,00p)***
   - Populare completă şi corectă a vectorului. ***(0,25p)***
   - Testare implementare prin apel de functie si afisare la consola a rezultatului obtinut la apel. ***(0,25p)***

4. Scrieţi şi apelaţi funcţia pentru determinarea soldurilor bancare totale la nivel de client. Se iau in considerare conturile bancare salvate in vectorul de la la cerinta 2). Un client poate avea deschise mai multe conturi bancare avand aceeasi valuta. Perechile de valori ***(nume_client, sold_total)*** sunt salvate intr-un vector. Vectorul si dimensiunea acestuia se returnează în ***main()*** prin tipul de retur sau lista de parametri ai funcţiei.

   Cerinţe de implementare:
   - Definire funcţie cu parametri de I/O definiţi complet şi corect. ***(0,25p)***
   - Determinare valori ***(nume_client, sold_total)*** pentru conturile bancare obtinute la cerinta 2). ***(0,75p)***
   - Implementare logică de creare vector cu valori ***(nume_client, sold_total)***. ***(1,25p)***
   - Populare completă şi corectă a vectorului. ***(0,25p)***
   - Testare implementare prin apel de functie si afisare la consola a rezultatului obtinut la apel. ***(0,25p)***

5. Scrieţi şi apelaţi funcţiile care dezalocă structurile ***Tabelă de Dispersie*** si ***3xVectori*** precum şi toate structurile auxiliare utilizate în implementarea cerinţelor (dacă este cazul).

   Cerinţe de implementare:
   - Definire funcţii cu parametri de I/O definiţi complet şi corect. ***(0,15p)***
   - Absenţă memory leaks. ***(0,15p)***
   - Actualizare variabile de gestionare a structurilor în funcţia ***main()***. ***(0,20p)***
   - Implementare logică de dezalocare a structurilor de date. ***(0,30p)***
   - Testare implementare, dezalocare completă şi corectă a structurilor prin apel de functii si afisare la consola a rezultatelor obtinute la apel. ***(0,20p)***
   - Absenţă dezalocări structuri auxiliare utilizate. ***(-0,20p)***

Bilet #4 / (150 min) | **11.06.2021**

MENTIUNI:

- **Proiectele cu erori de compilare nu vor fi evaluate.**
- **Implementările nu trebuie să conţină variabile definite la nivel global sau variabile statice.**
- **Implementarile nu trebuie sa contina structuri predefinite (ex STL, 3rd party libraries etc).**
- **Implementările plagiate vor fi evaluate cu 0 puncte, indiferent de sursă.**
- **Toate cerinţele trebuie apelate şi demonstrate in functia main() pentru a fi evaluate.**
- **Art. 72 (1) Pentru următoarele fapte, studenţii vor fi exmatriculaţi fără drept de reînmatriculare în Academia de Studii Economice din Bucureşti:**
  (c) încercarea de promovare prin fraudă a examenelor sau a altor evaluări;