

Implementați o aplicație în limbajul C care gestionează evaluările studenților în cadrul unei sesiuni de examinare.

1. Scrieți secvența de cod sursă pentru crearea unei structuri de date de tip **Tabelă de Dispersie** ce conține date aferente evaluărilor folosind ca mecanism de tratare a coliziunilor, **Chaining**. Tabela de dispersie stochează elemente de tip **Evaluare\***. Cheia de cautare utilizată este **denumireExamen(char\*)**. Alte atribute necesare în cadrul structurii Evaluare: **numeStudent(char\*)**, **semestru** (valori valide 1/2), **notaFinala(float)** și alte 2 atribute la alegere. Tabela de dispersie conține minim 10 evaluări încărcate în aplicație dintr-un fișier de intrare. (2p)

**Cerințe de implementare:**

- Definire structură **Evaluare**. (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 **Tabelă de Dispersie** cu **Chaining**. (0,75p)
  - 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 evaluărilor din structura creată la cerința 1) prin filtrarea acestora pe baza valorii unuia din atributele optionale. Valoarea este trimisă ca parametru funcției. Evaluările sunt salvate într-un vector fără ca acesta să partajeze zone de memorie heap cu elementele din tabela de dispersie. Vectorul și dimensiunea acestuia sunt returnate 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 produselor în vector. (0,25p)
  - Implementare logică de determinare și salvare a produselor în vector. (1p)
  - 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 numărului de examene promovate pentru fiecare cluster de coliziuni din **Tabela de Dispersie**. Rezultatul se va stoca într-un vector în care fiecare element conține perechea de valori (**index cluster, numar examene promovate**). 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 vector. (0,25p)
  - Implementare logică de determinare a elementelor în vector. (1,25p)
  - Populare completă și corectă a vectorului. (0,25p)
  - Testare implementare prin apel de funcție și afisare la consola a rezultatului obținut la apel. (0,50p)
4. Scrieți și apelați funcția pentru crearea unei matrice care stochează evaluările grupate după atributul semestru. Structura **partajează** zone de memorie heap cu tabela de dispersie la nivelul evaluărilor stocate. Structura rezultată este returnată în **main()** iar conținutul este afisat la consola. (2.5p)

**Cerințe de implementare:**

- Definire funcție cu parametri de I/O definiți complet și corect. (0,25p)
  - Determinare elementelor ce trebuie inserate în matrice. (0,25p)
  - Implementare logică de creare a matricei; fiecare din cei 2 vectori (pentru sem.1/2) sunt alocați dinamic. (1p)
  - Populare completă și corectă a matricei cu partajarea memorie heap cu tabela de dispersie. (0,50p)
  - Testare implementare prin apel de funcție și afisare la consola a rezultatului obținut la apel. (0,50p)
5. Scrieți și apelați funcțiile care dezalocă structura principală **Tabelă de Dispersie**, **vectorii rezultati** în implementarea cerințelor 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)

**NOTES:**

- Projects with compilation errors won't be evaluated.
- Implementations must not contain globally defined or static variables.
- Implementations must not use predefined structures such as STL or 3rd party libraries.
- Plagiarized implementations will be evaluated with 0 points, regardless of the source.
- All requirements must be called and demonstrated in the main () function to be evaluated.
- **Art. 72 (1) For the following facts, students will be expelled without the right to re-enroll in the Academy of Economic Studies in Bucharest:**
  - **(c) attempting to fraudulently pass examinations or other assessments;**