

Tema săptămânii 3
17 octombrie 2014

Problemă obligatorie

Termen de predare : Laboratorul din săptămâna 5 (31 octombrie 2014)

(2 p) **1.** Sa se creeze o listă ordonată folosind inserția elementelor în listă.

Probleme suplimentare

Termen de predare : Laboratorul din săptămâna 5 (31 octombrie 2014)

(1 p) **2.** Să se creeze o listă simplu înlănțuită alocată dinamic în care se vor reține numere întregi. Să se scrie un program pentru numărarea elementelor pozitive din listă și pentru calcularea mediei aritmetice a celor negative.

(2 p) **3.** Să se creeze o listă simplu înlănțuită alocată dinamic în care se vor reține numere întregi. Să se scrie o funcție care să modifice lista, adăugând între fiecare 2 elemente consecutive în listă media lor aritmetică.

(1 p) **4.** Să se creeze o listă simplu înlănțuită alocată dinamic în care se vor reține numere întregi. Să se scrie o funcție pentru ștergerea elementului de pe poziția k din listă, unde k se va transmite ca parametru.

(1 p) **5.** Să se implementeze cu ajutorul unei liste simplu înlănțuite un polinom de grad n . Fiecare nod se va considera că reține gradul fiecărui monom, precum și coeficientul său.

Structura poate fi definită astfel :

```
struct pol {  
    int gr, coef;  
    pol *next;  
};
```

Scrieți un program care creează un polinom implementat prin liste și calculează și afișează coeficienții polinomului obținut prin înmulțirea cu un scalar a , dat de la tastatură.

(2+2 p) **6.** Dată o listă simplu înlăuită A, scrieți o funcție care inversează ordinea elementelor în listă.

Se vor aborda doua metode: prima, în care lista A nu este modificată, rezultatul afișându-se într-o nouă listă B; a doua, în care nu se va aloca memorie suplimentară, inversarea făcându-se schimbând legăturile între nodurile listei A.

(3 p) **7.** Fie doua liste simplu înlătuite A și B, cu elementele întregi în ordine crescătoare.

Să se creeze o nouă listă simplu înlătuită C ce conține elementele din A și din B în ordine crescătoare. **Nu se va folosi memorie suplimentară.**

(3 p) **8.** Fie lista simplu înlătuită C. Să se distribuie elementele din C în doua liste simplu înlătuite A și B, astfel: A conține elementele de pe pozițiile impare din C, iar B conține elementele din C de pe pozițiile pare. **Nu se va folosi memorie suplimentară.**

(2 p) **9.** Un *vector rar* este un vector care are cel puțin 80% dintre elemente egale cu 0. O reprezentare eficientă a vectorilor rari se poate face cu ajutorul listelor simplu înlătuite alocate dinamic, reținând pentru fiecare componentă atât valoarea, cât și indicele din vector. Să se scrie un program pentru adunarea, respectiv, produsul scalar a doi vectori rari reprezentați cu ajutorul listelor simplu înlătuite.

(4 p) **10.** Spunem că o matrice X de dimensiuni $n \times m$ (n linii și m coloane) este rară dacă există “foarte multe” elemente egale cu 0. Pentru a economisi memoria, putem reprezenta o astfel de matrice prin liste simplu înlătuite. Pentru fiecare linie i nenulă vom avea o listă ce conține doar elemente nenule, în care fiecare nod are 3 câmpuri:

- (a) indicele coloanei, j ;
- (b) valoarea elementului nenul x_{ij} ;
- (c) o legătură către următorul element nenul de pe linie.

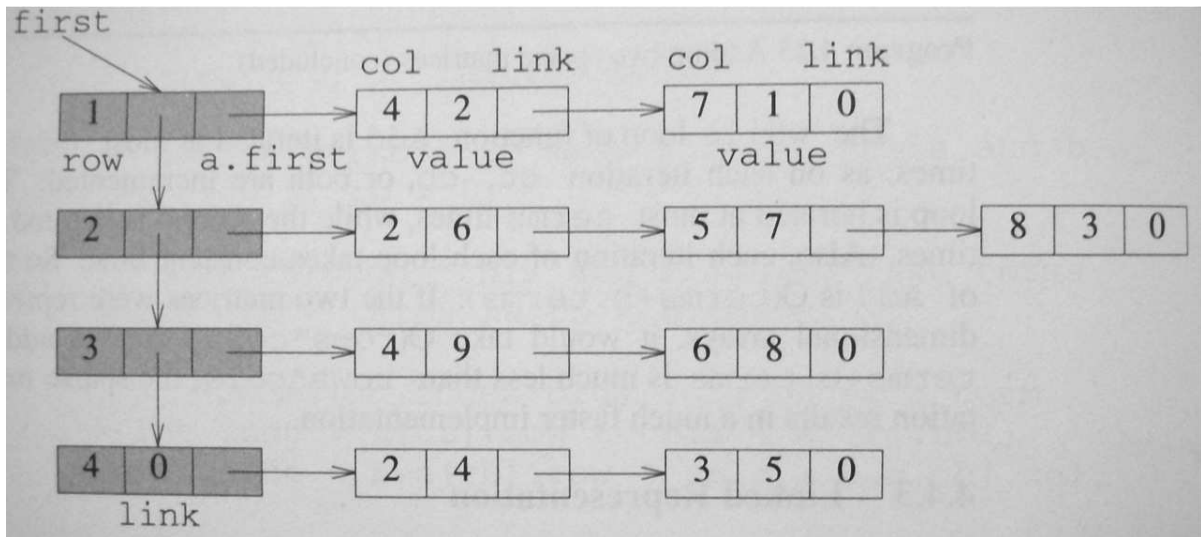
Pentru că trebuie să memorăm și care este primul element nenul dintr-o linie, vom avea o listă adițională în care fiecare nod conține următoarele informații:

- (a) indicele i al unei linii nenule;
- (b) o legătură către următoarea linie nenulă;
- (c) o legătură către primul element nenul din lista corespunzătoare liniei i .

Spre exemplu, matricea 4×8

$$\begin{pmatrix} 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 6 & 0 & 0 & 7 & 0 & 0 & 3 \\ 0 & 0 & 0 & 9 & 0 & 8 & 0 & 0 \\ 0 & 4 & 5 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

va fi reprezentata ca:



Folosind aceasta reprezentare pentru matrice rare, sa se scrie un program care citeste doua matrice, le reprezinta ca mai sus si face suma lor, reprezentata tot ca matrice rara.

Probleme facultative

Termen de predare : Laboratorul din săptămâna 5 (31 octombrie 2014)

(10ps) 1. Fie a un vector de n componente intregi, neordonate. Spunem ca un element x este majoritar in a daca apare de cel putin $\lceil n/2 \rceil$ ori in a . Descrieti si implementati un algoritm ce ruleaza in timp $O(n)$ care sa decida daca exista un element majoritar, si, daca da, sa il afiseze.