1. **Liste simplu inlantuite**
2. Creati o lista - direct in forma ordonata, datele (valori intregi) sunt citite de la tastatura/din fisier. Afisati-o!
3. Scrieti o functie care determina cate elemente are o lista in care sunt stocate date de tip int. Se da adresa primului element.
4. Scrieti o functie care inverseaza ordinea elementelor dintr-o lista in care sunt stocate date de tip int. Se da adresa primului element. Nu se foloseste alta lista.
5. Pentru exemplul dat in curs - inlocuiti tipul Date din int cu:

struct Student {

char\*nume;

int varsta;

};

Mai functioneaza corect afisarile si atribuirile? Ce modificari trebuiesc facute? Implementati!

1. Fie o lista cu date de tip Produs:

struct Produs {

char id[6]; /\*id-ul e unic\*/

char\*nume;

int pret;

};

Implementati functii pentru:

1. Citirea/Afisarea datelor despre un produs de la tastatura/pe consola (d/intr-un fisier)
2. Adaugarea unui Produs in lista
3. Cautarea unui Produs in lista in functie de id-ul sau. Functia returneaza pozitia in lista sau -1 daca nu este gasit produsul
4. Adaugarea unui produs in lista dupa un alt produs care are un anumit id
5. Adaugarea unui produs in lista dupa un alt produs care are o anumita adresa
6. Modificarea datelor unui produs (ex: pretul sau numele) in functie de id-ul acestuia
7. Stergerea unui produs din lista in functie de id
8. Stergerea tuturor produselor cu pretul mai mare decat un pret\_max
9. Stergeti produsele cu aceelasi nume care apar in lista pastrand doar elementul unde a fost gasit prima data un nume nou
10. Sortati lista in functie de pretul produselor folosind merge sort
11. Inversati ordinea elementelor din lista fara a folosi spatiu in plus (adica alta lista)

1. Interclasati doua liste ce contin valori intregi ordonate (fara a pastra listele originale <=> fara a folosi memorie in plus) si afisati rezultatul. Folositi pentru a implementa merge-sort. Care e complexitatea spatiala?
2. Creati o functie pentru eliminarea elementelor care se repeta intr-o lista de intregi si afisati lista rezultata. Care e complexitatea solutiei pe care ati propus-o?
3. Scrieti o functie care determina mijlocul unei liste (optim).

1. Polinoamele cu multi coeficienti nuli se numesc rare. Stocati informatiile utile pentru un astfel de polinom sub forma de lista.

Ex: x^16+2x^5+8x e stocat intr-o list cu 3 elemente care contin fiecare date de tip

struct monom{

int grad;

double coefficient;

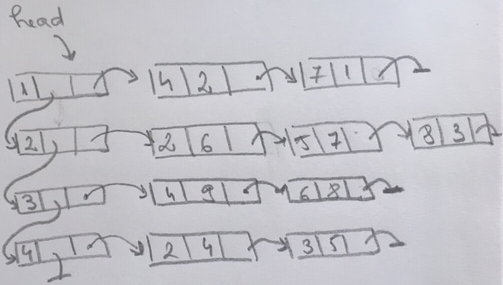
};

Implementati

1. O functie care creaza un polinom cu date citite de la tastatura/fisier
2. O functie care calculeaza suma a doua polinoame rare
3. O functie care afiseaza polinomul
4. Fie a o lista ce contine date de tip int. Spunem ca un element x este majoritar in lista daca apare de cel putin n /2 +1. Implementati un algoritm de complexitate minima care sa decida daca exista un element majoritar, si, daca da, sa il afiseze.
5. O matrice A de dimensiuni n × m ( linii x coloane) e rara daca majoritatea elementelor sunt egale cu 0. Pentru a economisi memorie, putem reprezenta o astfel de matrice folosind liste simplu inlantuite: pentru fiecare linie i nenula vom avea o lista ce contine doar elemente nenule, fiecare nod avand 3 campuri: indicele coloanei - j; valoarea elementului nenul Xij; o legatura catre urmatorul element nenul de pe linie.

Pentru a memora primul element nenul dintr-o linie avem nevoie de o lista aditionala in care fiecare nod stocheaza urmatoarea informatie: indicele i al unei linii nenule; o legatura catre urmatoarea linie nenula; o legatura catre primul element nenul din lista corespunzatoare liniei

Fie matricea de dimensiuni 4 × 8:



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

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

1. **Liste simplu inlantuite circulare**
2. Fie un poligon convex. Se citeste numarul de varfuri si apoi coordonatele lor, in ordine trigonometrica intr-o lista. Sa se determine perimetrul poligonului.
3. Jocul lui Joseph: N copii numerotati de la 1 la N sunt asezati in cerc (un copil are un numar si un nume asociat). Copilul 1 alege un numar p si incepand cu el se numara pana la p, copilul p e eliminat. Incapand cu p+1 se reia procedura, pana sunt eliminati toti copii. Afisati numele copiilor in ordinea in care sunt dati afara.

**(Tema dupa C4)**

1. **Liste dublu inlantuite**
2. Implementati problema I.5. folosind o lista dublu inlantuita cu santinele la inceput si sfarsit. Care sunt beneficiile?
3. Implementati toate functionalitatile necesare unei liste dublu inlantuite, circulare, cu santinela (creare, stergere, adaugare, stergere, modificare element de pe o anumita pozitie, de la o anumita adresa).