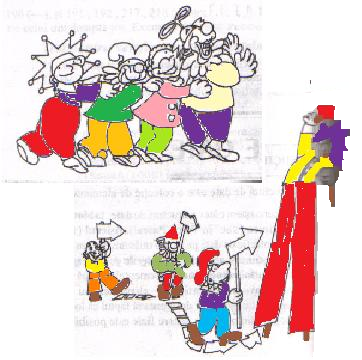
Orice colecție de date pe care o utilizăm într-un program are o anumită organizare internă. Indiferent de complexitatea modului de organizare a datelor, structurile de date se pot clasifica în două tipuri fundamentale: secvențiale (contigue) și aleatoare (non-contigue).

Termenul de contiguu desemnează faptul că locațiile de memorie sunt așezate una după cealaltă, continuu din punct de vedere logic (din punct de vedere fizic este posibil ca această continuitate să nu fie respectată).

În cazul structurilor secvențiale, componentele acestora sunt așezate în locații succesive de memorie.

De exemplu, în cazul unui tablou fiecare element este așezat lângă unul sau lângă două alte elemente ale tabloului. […]

Pentru un tablou toate elementele sunt de același tip, deci vor avea aceeași mărime. La o dată de tip înregistrare tipurile câmpurilor pot fi diferite, și în acest caz vor avea mărimi diferite. […]



În cazul structurilor memorate aleator, elementele structurii nu ocupă o zonă contiguă de memorie, ordinea lor fiind restabilită prin legături. […]

Unul dintre cele mai uzuale moduri de a avea o evidență pentru o mulțime de date este să facem o listă a acestora: lista de produse dintr-un depozit, lista de cumpărături, lista programelor care așteaptă la imprimantă, o listă de cărți etc.; orice program mai important folosește cel puțin o listă, declarată explicit sau nu.

Datorită importanței lor, se poate defini o structură de date – lista, cu forme particulare, de asemenea, foarte importante.

O listă văzută ca o structură de date este o colecție de elemente de același tip pentru care se definește un set de operații.

Elementele listei (nodurile) se notează cu a1, a2, …an, unde indicele ne arată poziția nodului în cadrul listei. Mărimea listei în acest caz este n și vom spune că lista este vidă dacă n=0. […]

Pentru orice listă nevidă, sunt valabile următoarele afirmații: pentru 1<k<n, ak este precedat de nodul ak-1 și este urmat de nodul ak+1; a1 este primul nod și el nu are predecesor; an este ultimul nod și el nu are succesor. […]

Întrebări cu răspunsuri multiple. […]

1. Stiva poate fi descrisă astfel:
   1. primul element introdus în listă este ultimul care poate fi extras;
   2. primul element introdus în listă este primul care poate fi extras;
   3. ultimul element introdus în listă este primul care poate fi extras.
2. La o stivă adăugarea unui nou element se poate face:
3. după elementul din vârf;
4. la baza stivei;
5. într-o poziție intermediară.
6. Eliminarea unui element din stivă se poate face astfel:
7. se elimină elementul din vârf;
8. se elimină elementul de la bază;
9. se poate elimina orice element din stivă. […]

|  |  |  |  |
| --- | --- | --- | --- |
| Nr. crt. | Tip operație | secvențial | înlănțuit |
| 1 | acces secvențial | avantajos | avantajos |
| 2 | acces aleator | avantajos | dezavantajos |
| 3 | redimensionare | dezavantajos | avantajos |
| 4 | rearanjare a elementelor | dezavantajos | avantajos |

(Adaptat după *Manualul de* *Informatică, clasa a X-a*, Livia Ţoca, Andreea-Ruxanda Demco, Cristian Opincaru, Adrian Sindile)