

Proiect de lecție

Clasa: a X-a

Disciplina: Informatică

Subiectul lecției: Aplicații cu vectori

Unitatea de învățare: Tablouri unidimensionale

Competențe specifice:

- identificarea necesității structurării datelor în tablouri
- prelucrarea datelor structurate în tablouri
- utilizarea fișierelor text pentru introducerea datelor și extragerea rezultatelor

Obiective operaționale:

- O1. să identifice situațiile în care utilizarea tablourilor unidimensionale în anumite probleme propuse este necesară și situațiile în care nu este necesară utilizarea vectorilor
- O2. să citească/scrie elementele unui vector din/în fișiere text
- O3 .să utilizeze eficient structurile de control studiate în aplicațiile cu tablouri unidimensionale

Strategia didactică

a) Tipul experienței de învățare

- legarea teoriei de practică
- formarea temeinică a cunoștințelor
- accesibilă, individualizată

b) Metode și procedee

- metode de comunicare: explicații, problematizare, conversația
- metode de acțiune: exerciții, experimentarea

c) Mijloace de învățământ

- mijloace vizuale: manual, computer, videoproiector/tabla, medii de programare

d) Forme de organizare

- individual: realizarea de sarcini practice
- frontal: lecția

1. Moment organizatoric

Conținut:

- verificarea prezenței
- pregătirea computerelor

2. Captarea atenției

Conținut:

- elevii vor căuta pe grupe diverse aplicații în lumea reală a utilizării vectorilor (ex. ordonarea unor obiecte după o anumită trăsătură), la final urmând să fie discutate cu restul clasei

Metode și procedee didactice:

- metode de comunicare: conversația
- metoda brainstormingului

Forma de organizare:

- frontală
- în grup

Mijloace didactice:

- computer
- internet

3. Anunțarea subiectului lecției și a obiectivelor

Conținut:

- se vor anunța obiectivele lecției într-un mod accesibil elevilor

4. Reactualizarea cunoștințelor anterioare

Conținut:

- se reamintește maniera de declarare a unui tablou unidimensional și elementele/variaibilele care permit gestionarea/prelucrarea unui vector
- se reamintesc cele 2 moduri în care se poate citi un vector dintr-un fișier: cel în care fișierul conține dimensiunea reală a vectorului și cel în care fișierul nu conține acest număr
- se verifică tema, iar dacă este cazul se reiau exercițiile din tema

Metode și procedee didactice:

- metode de comunicare: conversația
- metode de acțiune: exercițiu

Forma de organizare:

- frontală

Mijloace didactice:

- computer

5. Dirijarea învățării

Conținut

- elevii vor rezolva, în grupe de 2-3 persoane următoarele probleme

Exercițiu	Enunț	Rezolvare
1 O2	<p>Fișierul text "intrare.txt" conține pe mai multe linii, cel mult 100 de valori reale separate prin spații. Determinați pozițiile din șirul de valori pe care se află valoarea maximă din șir (pentru construirea șirului se parcurg liniile fișierului de la stânga la dreapta și de sus în jos iar elementele acestuia vor fi numerotate de la 1). Rezultatele se vor scrie în fișierul text "iesire.txt", pe prima linie a acestuia, separate prin câte un spațiu.</p> <p>Exemplu: intrare.txt conține: 3 4 1 4 2 3 4 iesire.txt va conține: 2 4 7</p>	<pre>#include <iostream> #include <fstream> using namespace std; int main() { float x[100]; int n=0,i; ifstream f("intrare.txt"); while(f>>x[n]) n++; f.close(); float maxim=x[0]; for(i=1;i<=n-1;i++) if(x[i]>maxim) maxim=x[i]; ofstream g("iesire.txt"); for(i=0;i<=n-1;i++) if(x[i]==maxim) g<<i+1<<' '; g.close(); return 0; }</pre>
2 O1,O2	<p>Fișierul text "intrare.txt" conține pe prima linie două valori naturale nenule n și k ($1 \leq k \leq n \leq 20$). Pe a doua linie a fișierului sunt scrise n valori întregi. Scrieți în fișierul text "iesire.txt", pe prima linie a acestuia, toate elementele de pe a doua linie, exceptând elementul de pe poziția k, (elementele șirului sunt numerotate de la 1) separate prin câte un spațiu.</p> <p>Exemplu: intrare.txt conține: 5 3 1 2 3 4 5 iesire.txt va conține: 1 2 4 5</p> <p>Obs. Se explică de ce nu se utilizează vectori (deoarece se poate parcurge o singură dată șirul este mai eficient să nu se rețină un vector)</p>	<pre>#include <iostream> #include <fstream> using namespace std; int main() { int n,i,k; ifstream f("intrare.txt"); ofstream g("iesire.txt"); f>>n>>k; while(f>>x) { if(k!=1) g<<x<<' '; k--; } f.close(); g.close(); return 0; }</pre>

<p>3 O1,O2,O3</p>	<p>Fișierul text "intrare.txt" conține pe prima linie o valoare naturală nenulă n ($1 \leq n \leq 50$). Pe a doua linie a fișierului sunt scrise n valori întregi. Verificați dacă elementele de pe a doua linie a fișierului text pot forma o mulțime (nu există duplicate). Scrieți apoi în fișierul text "iesire.txt", pe prima linie a acestuia, mesajul "multime" dacă testul este afirmativ, respectiv mesajul "eroare", în caz contrar.</p> <p>Exemplu: intrare.txt conține: 5 1 2 3 4 5 iesire.txt va conține: multime intrare.txt conține: 5 1 2 3 3 5 iesire.txt va conține: eroare</p>	<pre>#include <iostream> #include <fstream> using namespace std; int main() { int x[50],n,i,j; ifstream f("intrare.txt"); f>>n; for(i=0;i<=n-1;i++) f>>x[i]; f.close(); int test=1; for(i=0;i<=n-2 && test;i++) for(j=i+1;j<=n-1 && test;j++) if(x[i]==x[j]) test=0; ofstream g("iesire.txt"); if(test) g<<"multime"; else g<<"eroare"; g.close(); return 0; }</pre>
<p>4 O1,O2,O3</p>	<p>Se dau două mulțimi A și B de numere reale, cu maximum 100 de elemente fiecare. Determinați mulțimile A reunit cu B, A intersectat cu B și diferența lui A și B ($A-B$). Elementele mulțimii A sunt scrise în fișierul "A.txt" pe mai multe linii, separate prin spații; elementele mulțimii B sunt scrise în fișierul "B.txt" pe mai multe linii, separate prin spații. Elementele mulțimii A reunit cu B vor fi scrise în fișierul "reuniune.txt", pe prima linie a acestuia, separate prin câte un spațiu; elementele mulțimilor A intersectat cu B și $A-B$ vor fi scrise în aceeași manieră în fișierele text "intersectie.txt" respectiv "diferenta.txt".</p> <p>Exemplu: intrare.txt conține: 1 2 3 4 5 10 3 4 5 6 7</p>	<pre>#include <iostream> #include <fstream> using namespace std; int main() { float A[100], B[100]; int m,n,i,j; m=n=0; ifstream f("A.txt"); while(f>>A[m]) m++; f.close(); ifstream h("B.txt"); while(h>>B[n]) n++; h.close(); //reuniune ifstream r("reuniune.txt"); for(i=0;i<=m-1;i++) r<<A[i]<<' '; for(i=0;i<=n-1;i++) {</pre>

	<p>reuniune.txt va conține: 1 2 3 4 5 10 6 7</p> <p>intersectie.txt va conține: 3 4 5</p> <p>diferenta.txt va conține: 1 3 10</p>	<pre> int gasit=0; for(j=0;j<=m-1 && !gasit;j++) if(A[j]==B[i]) gasit=1; if(!gasit) r<<B[i]<<' '; } r.close(); //intersectie ofstream q("intersectie.txt"); for(i=0;i<=n-1;i++) { int gasit=0; for(j=0;j<=m-1 && !gasit;j++) if(A[j]==B[i]) gasit=1; if(gasit) q<<B[i]<<' '; } q.close(); //diferenta ofstream d("diferenta.txt"); for(i=0;i<=m-1;i++) { int gasit=0; for(j=0;j<=n-1 && !gasit;j++) if(B[j]==A[i]) gasit=1; if(!gasit) d<<A[i]<<' '; } d.close(); return 0; } </pre>
--	---	---

Metode și procedee didactice:

- metode de comunicare: conversația
- metode de acțiune: exercițiu

Forma de organizare:

- în grup

Mijloace didactice:

- computer

6. Obținerea performanței

Conținut:

- pe măsură ce exercițiile sunt rezolvate, va ieși câte o grupă la tablă (preferabil să nu iasă de mai multe ori o singură grupă, pentru a permite tuturor să prezinte câte o problemă) și va prezenta exercițiul (unde este posibil se va folosi videoproiector)
- de asemenea se discută soluțiile alternative ale celorlalte grupe

Metode și procedee didactice:

- metode de comunicare: conversația, problematizarea

Forme de organizare:

- în grup

Mijloace didactice:

- tabla

7. Asigurarea feedbackului / Evaluarea

Conținut:

- în timp ce elevii rezolvă exercițiile, profesorul va merge la fiecare grupă pentru a oferi feedback, explica enunțul ori problemele de sintaxa sau de logică
- în timp ce elevii prezintă, profesorul va coordona prezentarea, oferind feedback asupra prezentării, în același timp ghidându-i pe elevi în prezentare pentru a atinge punctele importante

8. Încheierea lecției

Conținut:

- tema va fi compusă din exercițiile rămase nerezolvate
- reamintirea obiectivelor lecției