

Algoritmi și programare - tema 2

Termen de realizare: săptămâna 5-9 noiembrie 2012

Obiective

- familiarizarea cu modul de lucru cu tablouri, pointeri și structuri;
 - parcurgerea tablourilor;
 - relația între tablouri și pointeri;
 - familiarizarea cu operațiile pe biți.
-

Probleme propuse

1. Se citește un tablou de numere întregi. Scrieți o funcție care să determine diferența maximă (în modul) dintre oricare două elemente ale vectorului.
Exemplu: pentru tabloul (-1, 3, 2, 7, 11, 3, 2), diferența maximă (în modul) este 12.
Observație: Punctajul maxim va fi acordat soluțiilor cu complexitatea timp $O(n)$.
2. Se citește un tablou de numere întregi. Să se determine, printr-o singură parcurgere a sa, poziția de început și lungimea celei mai lungi secvențe de elemente egale. Dacă există mai multe secvențe de lungimi egale, se va determina secvența cu poziția de început minimă.
Exemplu: pentru tabloul (1, 2, 3, 3, 3, 2, 1, 5, 5, 5), se vor determina poziția 2 și lungimea 3.
3. Se dă un vector de numere întregi. Să se determine subsecvența pentru care suma elementelor componente este maximă. Dacă există mai multe subsecvențe cu această proprietate, se va determina ultima subsecvență găsită.
Exemplu: pentru tabloul (2, 3, -4, -2, 9, -2, 1, 7, -3), subsecvența este (9, -2, 1, 7)
Observație: Punctajul maxim va fi acordat soluțiilor cu complexitatea timp $O(n)$.
4. Se citește un tablou de numere întregi. Să se rearanjeze elementele astfel încât cele pare să apară înaintea celor impare. În cadrul subsecvenței de numere pare, respectiv impare, elementele trebuie să apară în ordinea în care erau în tabloul inițial.
Exemplu: tabloul (2, 3, -4, -2, 9, -2, 1, 7, -3) va fi rearanjat în (2, -4, -2, -2, 3, 9, 1, 7, -3).
5. Se consideră un tablou cu n elemente. Să se decidă dacă există un element majoritar (cu numărul de apariții mai mare decât $n/2$).
Observație: Punctajul maxim va fi acordat soluțiilor cu complexitatea timp $O(n)$.
6. Scrieți un program C care citește elementele unei matrici pătratice și interschimbă liniile și coloanele acesteia astfel încât, în matricea finală, elementele de pe diagonala principală să fie în ordine descrescătoare. Să se afișeze matricea inițială și matricea după fiecare operație.
Exemplu: pentru matricea
1 4
3 2
se vor afișa, pe rând, matricile
1 4 4 1 2 3
3 2 2 3 4 1.
7. Scrieți un program C care citește elementele unei matrici pătratice de dimensiune n și afișează la ieșirea standard șirul format prin parcurgerea acesteia în spirală, pornind din colțul stânga sus către dreapta până în centrul matricii.
Exemplu: pentru matricea
1 2 3
4 5 6
7 8 9
se va afișa șirul 1 2 3 6 9 8 7 4 5.

8. Să se definească un tip pentru reprezentarea datelor calendaristice. Să se scrie o funcție care primește o dată calendaristică și returnează succesorul acesteia.
Exemplu: succesorul datei "31.10.2012" este "1.11.2012".
9. Să se determine ecuația unei drepte care trece prin două puncte în plan.
Exemplu: pentru punctele (2, 1) și (5, 7), ecuația dreptei este $2x - y - 3 = 0$.
10. Să se definească un tip Multime ce conține elemente numere întregi de la 0 la 31. Să se definească funcțiile de reuniune, intersecție, diferență și complementara unei mulțimi, utilizand operatii pe biți. Prototipurile acestor funcții sunt:

```
Multime reuniune(Multime X, Multime Y);  
Multime intersectie(Multime X, Multime Y);  
Multime diferenta(Multime X, Multime Y);  
Multime complementara(Multime X);
```

Exemplu: pentru mulțimile care conțin numerele întregi A(5, 7, 31) și, respectiv, B(3, 7), reuniunea este (3, 5, 7, 31), intersecția este (7), diferența A-B este (5, 31), iar complementara lui A este (0 ... 4, 6, 8 ... 30).