**Tablouri cu o dimensiune (denumite vectori sau siruri) - continuare**

**1. Generarea unui vector**

a)Sa se genereze sirul numerelor naturale pana la un numar natural n dat.

**cin>>n;**

**for(i=1;i<=n;i++)**

**a[i]=i;**

b)Sa se genereze sirul numerelor pare <= cu n

**cin>>n;**

**m=0;**

**for(i=2;i<=n;i=i+2)**

**{m++; a[m]=i;}**

c)Sa se genereze sirul numerelor Fibonacci <= cu n dat

**cin>>n;**

**fib[1]=fib[2]=1;**

**for(i=3;i<=n;i++)**

**fib[i]=fib[i-1]+fib[i-2];**

Alte aplicatii:

d) Sa se genereze numerele prime <= cu un numar natural n dat.

e) Sa se genereze primele n numere prime, unde n este un numar natural dat.

**Numere aleatoare**

Aruncarea succesiva a unui zar. Sa se genereze un sir de numere aleatoare.

**Folosim functia rand();**

Exemple:

**v= rand() % 10; // v primeste o valoare intre 0 si 9  
v = rand() % 10 + 1; // v primeste o valoare intre 1 si 10**

**v = rand() % 30 + 1999; // v primeste o valoare intre 1999 si 2028**

insa functia **rand()** trebuie sa fie precedata de un apel al functiei **srand(time(NULL));** care initializeaza generatorul de numere.

Aceasta are nevoie de librariile:

**#include <time.h>**

**#include <cstdlib>**

Aruncarea succesiva a unui zar. Sa se genereze un sir de perechi de numere aleatoare.

Avem doua implementari posibile:

a) sa folosim doi vectori, cate unul pentru fiecare din cele doua zaruri

1. sa folosim un singur vector, in care sa asezam succesiv numerele generate

Cerinta: Sa afisam perechile a caror suma este para.

Se vor implementa ambele solutii.

1. Cu doi vectori:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Z1 | 6 | 1 | 2 | 3 | 4 | 5 | 6 |
|  |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Z2 | 6 | 5 | 1 | 6 | 4 | 2 | 5 |
|  |  |  |  |  |  |  |  |

1. cu un singur vector, pozitiile impare fiind pentru primul zar, celelalte pentru al doilea zar.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** |
| **6** | **6** | **1** | **5** | **2** | **1** | **3** | **6** | **4** | **4** | **5** | **2** | **6** | **5** |

**2. Perechi si triplete**

Exista 3 situatii:

A. perechi de valori aflate pe pozitii succesive in vector A

1. perechi de valori aflate pe orice pozitii

B1. orice combinatie

B2. numai combinatii in care se pastreaza ordinea din vector

1. **exemplul 1**: solutia 2 a problemei anterioare

**exemplul 2**: Fiind dat un vector de numere intregi. Sa se afiseze de cate ori gasim doua elemente aflate pe pozitii consecutive in vector care sa fie egale intre ele.

Exemple. Pentru vectorul (1,2,2,3,4,4,4,5,6) se va afisa valoarea 3.

**Exemplul 3**: Triplete. Sa se afiseze toate tripletele de numere crescatoare de pe pozitii consecutive din vectorul x de numere reale.

Exempul. Pentru x=(2,9,41,61,6,24,84,1,21) se vor afisa tripletele (2,9,41) , (9,41,61) si (6,24,84).

Solutie: cu o singura parcurgere

**for(i=1;i < n;i++)**

**compar a[i] cu a[i+1] ...**

**B1. exemplul 1**: Se citesc de la tastatura cele n elemente ale unui sir de numere intregi. Sa se afiseze toate perechiile de elemente ale sirului (nu neaparat consecutive) cu proprietate ca elemente ale perechii au aceeasi suma de cifre.

Exemplu. Pentru vectorul (12,9,54,32,222,49,34,6,1,91) se vor afisa perechile (9,54) (54,9) (222,6) (6, 222) .

**exemplul 2**: Sa se afiseze toate tripletele de numere din vector a caror suma este > decat o valoare data p.

Solutie: 2 for-uri, compar fiecare cu fiecare

**for(i=1;i <= n;i++)**

**for(j=1;j <= n;j++)**

**compar a[i] cu a[j] ...**

**B2. exemplul 1**: Se citesc de la tastatura cele n elemente ale unui sir de numere intregi. Sa se afiseze toate perechiile de elemente ale sirului (nu neaparat consecutive) cu proprietate ca elemente ale perechii au aceeasi suma de cifre.

Exemplu. Pentru vectorul (12,9,54,32,222,49,34,6,1,91) se vor afisa perechile (9,54) si (222,6) .

**exemplul 2**: Sa se afiseze toate tripletele de numere din vector a caror suma este > decat o valoare data p si care se afla in ordine in vector.

**exemplul 3**: Sa se afiseze toate tripletele de numere pitagoreice <= cu un n dat (atentie nu folosim vectori)

Solutie: 2 for-uri, compar fiecare cu fiecare dar j sa fie tot timpul mai mare ca j.

**for(i=1; i < n; i++)**

**for(j=i+1 ; j <= n ; j++)**

**compar a[i] cu a[j] ...**

**3. Secvente de elemente aflate pe pozitii succesive (extinderea perechilor de tip A)**

Exemplu: Secvente de elemente identice (cel putin 2)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** |
| **a:** | **6** | **6** | **1** | **5** | **5** | **5** | **5** | **4** | **4** | **4** | **4** | **4** | **4** | **4** |

Secventele detectate aici sunt:

**6 6** secventa identificata de pozitia de inceput: 1 si lungime:2

**5 5 5 5** secventa identificata de pozitia de inceput: 4 si lungime:4

**4 4 4 4 4 4 4** secventa identificata de pozitia de inceput: 8 si lungime:7

Concluzie:

Orice secventa se identifica prin **inceput** si **lungime** (sau prin sfarsit).

Cand cautam secventa de lungime maxima, este necesar sa pastram suplimentar 2 variabile pentru inceputul secventei maxime si lungimea aceasteia.

Aplicatii:

1. Se citeste un vector a cu n elemente naturale. Sa se imparta vectorul a intr-un numar minim de subsiruri strict crescatoare de lungime minim 2, de valori consecutive si sa se afiseze aceste subsiruri.   
   Exemplu: daca a=3 10 4 5 4 6 11 5, atunci subsirurile sunt:   
   3 10

4 5

4 6 11

1. Sa se determine cea mai lunga secventa de numere succesive in vector ce sunt in ordine crescatoare. Daca sunt mai multe, se afiseaza prima. Apoi sa se modifice astfel incat sa se afiseze ultima.
2. Se citesc valori aflate in fisierul date.in in ordine crescatoare. Numerele nu sunt neaparat distincte. Sa se afiseze valoarea ce este imediat mai mica decat cea aflata la pozitia n/2. N se calculeaza in momentul citirii. Daca nu exista se va afisa NU.
3. Fisierul date.in contine pe prima linie, separate prin cate un spatiu, cel mult 1000 de numere naturale, fiecare dintre ele avand maximum 9 cifre. Determinati cea mai lunga secventa ordonata strict descrescator formata din valori consecutive in fisier.

Sa se rezolve utilizand un vector.

Sa se rezolve apoi si fara vectori.

1. Se citesc din fisier numere nu neaparat distincte dar date in ordine crescatoare. Sa se creeze un vector cu numerele distincte si altul cu numarul de aparitii ale fiecarui astfel de numar.
2. Se citesc cifre separate cu spatii. Sa se elimine secventele de cifre egale aflate pe pozitii consecutive. Se repeta procedeul pana cand nu mai exista astfel de secvente.
3. Sa se determine cea mai lunga secventa de elemente aflate pe pozitii consecutive care au paritate diferita.