

13 Műveletek kétdimenziós tömbökkel

- legkisebb, legnagyobb elem megkeresése
- tömb tükrözése a függőleges tengelye szerint

13.1 Legkisebb, legnagyobb elem megkeresése

Készítsünk programot, amely egy 6x6-os kétdimenziós tömbbe kigenerál 10 és 70 közötti véletlen számokat, majd kiírja a kigenerált tömböt. A program második része keresse meg a tömb legkisebb és legnagyobb elemét, majd írja ki ezt a képernyőre.

A tömb kigenerálása - ahogy már megszoktuk - két egymásba ágyazott ciklus segítségével lesz megvalósítva.

A tömb legkisebb ill. legnagyobb elemét hasonlóan fogjuk megkeresni, mint az egydimenziós tömbnél. Vesszünk két változót - **min**, **max**, melyekbe a keresés előtt beállítjuk az **a[1,1]** elem értékét. Ez után végigmegegyünk a tömb összes elemén (két egymásba ágyazott ciklus segítségével), és ha találunk kisebb ill. nagyobb elemet, akkor ezt megjegyezzük a **min** ill. **max** változóknak. Végül a ciklusok lefutása után kiírjuk a **min** és **max** változók értékét.

```

program Pelda32a;
uses crt;
var a: array[1..6,1..6] of integer;
    i, j, min, max: integer;
begin
clrscr;
{ tömb kigenerálása }
randomize;
for i:=1 to 6 do
  for j:=1 to 6 do a[i,j]:=random(61)+10;
{ tömb kiírása }
for i:=1 to 6 do
  begin
  for j:=1 to 6 do write(a[i,j]:3);
  writeln;
  end;
{ min, max keresése }
min:=a[1,1];
max:=a[1,1];
for i:=1 to 6 do
  for j:=1 to 6 do
  begin
  if a[i,j]<min then min:=a[i,j];
  if a[i,j]>max then max:=a[i,j];
  end;
{ min, max kiírása }
writeln('Legkisebb elem: ',min);
writeln('Legnagyobb elem: ',max);
readln;
end.

```

Próbáljuk meg módosítani az előző feladatot úgy, hogy ne csak a maximum és minimum értékét írja ki, hanem azt is, hogy ezeket az elemeket melyik sor melyik oszlopában találta meg (tehát a maximum és a minimum tömbindexeit is).

Ehhez bevezetük négy új változót: **min_i**, **min_j**, **max_i**, **max_j**, melyekben mindig amikor változtatunk a **min** ill. **max** értékén, megjegyezzük az éppen elmentett elem tömbindexeit. Programunk ezekkel a módosításokkal így néz ki:

```

program Pelda32b;
uses crt;
var a: array[1..6,1..6] of integer;
    i,j,min,max,min_i,min_j,max_i,max_j:integer;
begin
  clrscr;
  { tömb kigenerálása }
  randomize;
  for i:=1 to 6 do
    for j:=1 to 6 do a[i,j]:=random(61)+10;
  { tömb kiírása }
  for i:=1 to 6 do
    begin
      for j:=1 to 6 do write(a[i,j]:3);
      writeln;
    end;
  { min, max keresése }
  min:=a[1,1];
  min_i:=1;
  min_j:=1;
  max:=a[1,1];
  max_i:=1;
  max_j:=1;
  for i:=1 to 6 do
    for j:=1 to 6 do
      begin
        if a[i,j]<min then begin
          min:=a[i,j];
          min_i:=i;
          min_j:=j;
        end;


        if a[i,j]>max then begin
          max:=a[i,j];
          max_i:=i;
          max_j:=j;
        end;
      end;
  { min, max kiírása }
  writeln('Legkisebb elem: ',min,' (' ,min_i,'. sor ',min_j,'. oszlop)');
  writeln('Legnagyobb elem: ',max,' (' ,max_i,'. sor ',max_j,'. oszlop)');
  readln;
end.

```

13.2 Tömb tükrözése a függőleges tengelye szerint

Készítsünk programot, amely egy 7x7-es kétdimenziós tömbbe kigenerál 100 és 999 közötti véletlen számokat, majd tükrözi a tömböt a függőleges tengelye szerint. A program írja ki az eredeti, majd a tükrözés utáni tömböt is.

	1	2	3	4	5	6	7
1	158	256	179	785	354	133	420
2	800	451	511	563	123	456	289
3	132	796	912	430	770	600	742
4	354	312	896	765	540	550	451
5	745	825	520	544	122	100	258
6	512	851	959	742	425	380	192
7	455	756	409	278	171	108	778



Ehhez előbb a tömb első, majd a második, harmadik, stb. sorait fogjuk tükrözni egymás után. Egy külső ciklus fogja megadni hogy éppen melyik sor tükrözését végezzük. Ezen belül lesz egy belső ciklus, amely az adott sort tükrözését végzi el – ez egy hasonló ciklus, mint amilyent az egydimenziós tömb tükrözésénél használtunk. Programunk így néz ki:

```

program Pelda33;
uses crt;
const n=7;
var a: array[1..n,1..n] of integer;
    i,j,x:integer;
begin
  clrscr;
  { tömb kigenerálása }
  randomize;
  for i:=1 to n do
    for j:=1 to n do a[i,j]:=random(900)+100;
  { tömb kiírása }
  for i:=1 to n do
    begin
      for j:=1 to n do write(a[i,j]:4);
      writeln;
    end;
  { tömb tükrözése }
  for i:=1 to n do
    begin
      { egy sor tükrözése }
      for j:=1 to n div 2 do begin
        x:=a[i,j];
        a[i,j]:=a[i,n-j+1];
        a[i,n-j+1]:=x;
      end;
    end;
  { tömb kiírása }
  writeln;
  for i:=1 to n do
    begin
      for j:=1 to n do write(a[i,j]:4);
      writeln;
    end;
  readln;
end.

```