

Tömbök és ciklusok

Egydimenziós tömbök:

Van olyan eset a programozás során, amikor nem egy különálló változót szeretnénk használni, hanem változók sorozatát. Ilyenkor azt mondjuk, hogy egy új típust szeretnénk definiálni, az egydimenziós tömböt. A típus bevezetésénél meg kell adnunk a tömb hosszát, azaz hogy hány elemből áll a sorozat, valamint a tömböt alkotó változók típusát. (Egész, valós, szöveg.) A változó deklarációnál ezt a típust használom fel. A következőképpen tehetem meg:

Pl.:

```
Program tombok;
uses crt, dos;
type tomb=array[1..100] of integer;
var x: tomb;
begin
  ...
end.
```

Számlálós ciklus:

Az egydimenziós tömbök használatakor egy új utasítást használunk fel, a számlálós ciklust.

Pl.:

```
for i:=1 to N do
  writeln(x[i]);
```

Az i változót ciklusváltozónak hívjuk és mindig integer típus. Ciklusváltozónak az i, j, k, l betűket szokás felhasználni. A fenti ciklus 1-től N-ig számol, és az x tömb tartalmát írhatja ki.

Ha a cikluson belül több utasítást szeretnénk használni, akkor itt is begin – end közé kell zárni.

Pl.:

```
for i:=1 to N do
  begin
    write('Kerem a(z) ',i,'. elemet'); readln(x[i]);
  end;
```

Kétdimenziós tömbök:

Van olyan eset, hogy az adatainkat mátrix formában szeretnénk tárolni, (pl.: könyvespolc), ekkor kétdimenziós tömböket használunk. A típus definiálása:

Pl.: type tomb2=array[1..10,1..20] of string;

A kétdimenziós tömbökhöz dupla számlálós ciklust használunk fel.

Pl.:

```
for i:=1 to N do
  for j:= 1 to M do
    writeln(x[i,j]);
```

Vagy

```
for i:=1 to N do
```

```
  for j:=1 to M do
```

```
    begin
```

```
      write('Kerem a(z) ',i,'. sorban és a(z) ',j,'. oszlopban szereplo erteket! '); readln(x[i,j]);
```

```
    end;
```