

10 Tömbök (array of ...)

- tömbök (array of ...)
- konstansok használata
- tömb elemeinek generálása

10.1 Tömbök (array of ...)

Eddigi programjainkban egyedi változókkal dolgoztunk, mindegyik változónak külön neve volt. Pl

```
var a,b,c,i,n:integer;
```

a	b	c	i	n
9	1	5	7	3

Sok feladatot lehetetlen egyedi változókkal megoldanunk. Gondoljunk például egy névsor tárolására, melyben akár 100-200 név is lehet.

Ehhez szükségünk van olyan adattípusra, melyben több adatot tárolhatunk. Az eddig megismert típusok **elemi típusok** voltak. Ezekből az elemi típusokból adatszerkezeteket, **összetett típusokat** építhetünk. Ilyen összetett típus a tömb is.

A **tömb** típusát a következő képpen kell megadnunk:

```
var tömb_neve : array [ indexhatárok ] of alaptípus ;
```

Az *indexhatárok* megadják a tömb méretét (hogy hány elemet tárolhatunk benne), az *alaptípus* pedig az elemek típusát (ez lehet pl. **integer**, **string**, **char**, **boolean**, **byte**, stb).

Például:

```
var a:array [1..7] of string;
```

a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
Peter	Agnes	Ivan	Jozsef	Eva	Katalin	Timea

Ez létrehoz egy 7-elemű, **string**ekből álló tömböt. Ebben tárolhatunk például egy 7 elemből álló névsort. A tömb egyes elemeit az elemek indexén keresztül érhetjük el úgy, hogy a tömb neve után szögletes zárójelben megadjuk az elem **indexét**. A tömbünk tehát a következő elemekből áll: **a[1]**, **a[2]**, ..., **a[7]**.

A tömb beolvasása történhet például egy egyszerű ciklussal:

```
for i:=1 to 7 do begin
    write('Add meg a(z) ',i,'. nevet: ');
    readln(a[i]);
end;
```

Hasonlóan történhet a tömb kiírása, feldolgozása is ciklusok segítségével.

Feladat: Készítsünk programot, amely beolvas 10 egész számot egy tömbbe, majd kiírja őket fordított sorrendben.

Megoldás: A programunk a következő képen néz ki:

```

program Pelda25a;
uses crt;
var a:array[1..10] of integer;
    i:integer;
begin
  clrscr;
  for i:=1 to 10 do begin
    write(i, '. szam: ');
    readln(a[i]);
  end;
  writeln('A szamok fordított sorrendben:');
  for i:=10 downto 1 do write(a[i]:6);
  writeln;
  readln;
end.

```

10.2 Konstansok használata

Programunkban használhatunk konstansokat is. Ezeket a **var** előtt a **const** utasítással adhatjuk meg:

```
const konstansneve = értéke ;
```

Például ha az előző programban 10 elem helyett 20-at szeretnénk beolvasni majd kiírni, több helyen meg kéne változtatnunk a programunkat - a 10-et átírni 20-ra. Ha program elején bevezetünk egy konstans (állandó) a tömb elemeinek számára és a programban mindenhol ezt használjuk, akkor később ha meg akarjuk változtatni a programban a beolvasandó elemek számát elég ennek a konstansnak az értékét megváltoztatni a programunk elején. Az előző programunk például így is nézhetett volna ki:

```

program Pelda25b;
uses crt;
const tombmerete=10;
var a:array[1..tombmerete] of integer;
    i:integer;
begin
  clrscr;
  for i:=1 to tombmerete do begin
    write(i, '. szam: ');
    readln(a[i]);
  end;
  writeln('A szamok fordított sorrendben:');
  for i:=tombmerete downto 1 do write(a[i]:6);
  readln;
end.

```

10.3 Tömb elemeinek generálása

Készítsünk programot, melyben egy tömb elemeit véletlenszerűen kigenerálja majd kiírja a képernyőre. A tömb elemeinek értéke 1 és 100 közötti szám legyen.

Ehhez a már régebben megismert **random** függvényt fogjuk használni. Mivel egy tömbről van szó, a generálást ciklus segítségével fogjuk elvégezni, hasonlóan ahogy az elemek kiírását is a képernyőre. Programunk így néz ki:

```

program Pelda26;
uses crt;
const n=20; {a tömb elemeinek a száma}
var a: array[1..n] of integer;
    i: integer;
begin
  clrscr;
  randomize;
  for i:=1 to n do a[i]:=random(100)+1;
  for i:=1 to n do write(a[i], ', ');
  readln;
end.

```

Feladatok:

1. Készíts programot, melyben létrehozol egy 5 elemből álló tömböt, amelybe keresztneveket veszel fel, majd egymás mellé kiíratod pontosvesszővel elválasztva! (**array01**)

```
Add meg az 1. nevet: Eva
Add meg az 2. nevet: Cecil
Add meg az 3. nevet: Janos
Add meg az 4. nevet: Sandor
Add meg az 5. nevet: Endre
-----
Eva; Cecil; Janos; Sandor; Endre
```

2. Készíts programot, melyen karakterekből álló tömböt hozol létre! Az elején konstansban add meg, hogy 6 karaktert keljen megadni. Majd tükörszerűen írasd ki egymás mellé kétszer a következő sorban a minta alapján! (**array02**)

```
Add meg az 1. karaktert: A
Add meg az 2. karaktert: B
Add meg az 3. karaktert: C
Add meg az 4. karaktert: D
Add meg az 5. karaktert: E
Add meg az 6. karaktert: F
-----
ABCDEF | FEDCBA
```

3. Készíts programot a minta alapján, mellyel ötösöltő sorsolást generálsz! (tömb használatával) (**array03**)

```
A(z) 1. szam: 21
A(z) 2. szam: 35
A(z) 3. szam: 90
A(z) 4. szam: 13
A(z) 5. szam: 54
```

4. Olvassunk be egész számokat egy hét elemű tömbbe! Utána irassuk ki a tömb elemeit sorban, majd visszafelé, aztán írassuk ki a legkisebb elemet és a legnagyobbat. Alá pedig a tömb elemeinek az átlagát! (**array04**)

```
Add meg a szamot: 22
Add meg a szamot: 43
Add meg a szamot: 67
Add meg a szamot: 44
Add meg a szamot: 69
Add meg a szamot: 13
Add meg a szamot: 35
-----
22;43;67;44;69;13;35;35;13;69;44;67;43;22
A legkisebb: 13
A legnagyobb:69
Atlag: 41,85
```

5. Állítsunk elő egy 30 elemű tömböt véletlen egész számokból (0-tól 99-ig). Írjuk ki a tömböt a képernyőre. (**array05**)

```
22;34;68;97;41;12;2;78;33;;52;37;97;
23;45;67;89;12;32;4;21;54;72;48;36;
61;28;31;54;67;82
```