ANSI C NYELVEN

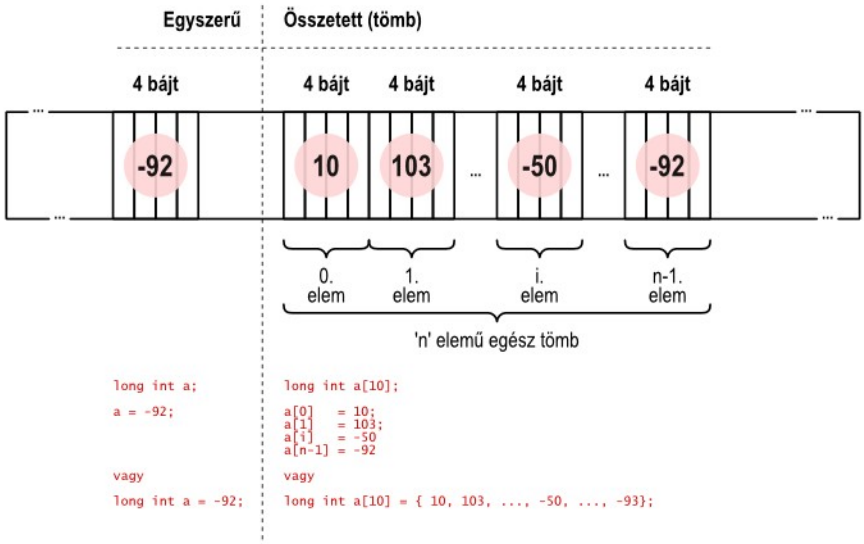
18-19. ÓRA

Tömbök
– a változók lehetnek egyszerűek, vagy összetettek (**ez már tudott**)
– az összetett változók esetében több egyszerű változót kapcsolhatunk egybe
– az összekapcsolás történhet
  – azonos típusú változók összekapcsolásával, vagy
  – eltérő típusú változók összekapcsolásával
– tömbről akkor beszélünk, ha azonos típusú változókat kapcsolunk össze úgy, hogy azok neve is azonos

Példa a tömbváltozó értelmezésére
– legyen egy hosszú egész változó '**bevetelek**' névvel, amely egy vállalkozás éves összes bevételét tartalmazza
– ha a bevételeket havonta is szeretnénk tárolni, akkor 'bevetelek1' … 'bevetelek12' változókat kellene definiálni
– ez több szempontból is helytelen megoldás lenne, mert
  – nehezen tudnánk megoldani az éves érték kiszámítását
  – túlságosan elbonyolítanánk a programunkat
  – a sok változó felesleges gépelést eredményezne (ami növelné a hibázás lehetőségét)
– megoldás a tömbváltozó alkalmazása

```
int bevetelek;     // egyszerű egész típusú változó
int bevetelek[12]; // összetett, 12 db egész típusú értéket tartalmazó változó
```

Tömb változók helyfoglalása



Változók címe és értéke újra
– ha van egy egész változónk (például: **int bevetel;**), akkor
  – nevét egy kifejezésben/képletben leírjuk, akkor annak értékét vesszük figyelembe
  – amennyiben a címére van szükségünk (scanf függvény), akkor &bevitel formában hivatkozhatunk rá
– egy tömbváltozó (például: **int bevetelek[10];**) esetén valójában több egyszerű változónk van egyszerre
  – az egyes elemeket **index** hivatkozásokkal érhetjük el
  – a tömb legelső értékére a bevtelek[0], vagyis a 0. indexű elem
  – a tömb nevének leírásával magára a tömbre hivatkozunk, de mivel
  – a tömbnek **nincs** értéke (csak ez egyes elemeinek), ezért ilyenkor a tömb címére utalunk

Tömbváltozók értékei és címei

```
int bevetelek[12];         // egész típusú tömb, 12 darab értékkel
int i;                     // ciklusváltozó
int *pi;                   // egész típusú értékre mutató változó
int havi;                  // egész típusú változó

pi = bevetelek;            // a 'pi' mutató a bevetelk tömbre mutat (0. elemre)
pi = &bevetelek[0];        // ugyan azt jelenti mint az előbbi
bevetelek[4] = 120000;     // a tömb ötödik elemének értéket adunk (120000)
*(pi+4) = 80000;           // ugyan az, mint az előbbi
```

Tömb feltöltése konstans adatokkal

```
#define HONAP 12

int main(void) {
int i;
int bevetelek[HONAP] = {                 // 12 elemu egesz tomb
    120, 240, 240, 123, 100, 210,        // elso hat elem ertekei
    200, 240, 210, 135, 100, 190         // masodik hat elem ertekei
};

    // adatok kiirasa kepernyore
    for (i=0; i<HONAP; i++) {
           printf("\n%2d. honap: %8d eFt", i+1, bevetelek[i]);
    }
    printf("\n");
    return (0);
}
```

Tömb feltöltése felhasználótól kapott adatokkal

```
#define HONAP 12

int main(void) {
int i;
int bevetelek[HONAP];
};

    // adatok beolvasasa billenytozetrol
    printf("\nAdja meg a havi beteleket\n");
    for (i=0; i<HONAP; i++) {
           printf("- %2d. honap bevetele [eFt]: ", i+1);
           scanf("%d", &bevetelek[i]);
    }

    // adatok kiirasa kepernyore
    for (i=0; i<HONAP; i++) {
           printf("\n%2d. honap: %8d Ft", i+1, bevetelek[i]);
    }
    printf("\n");
    return (0);
}
```

Házi feladat

– mik azok a programozási tételek