



# Programozás I.

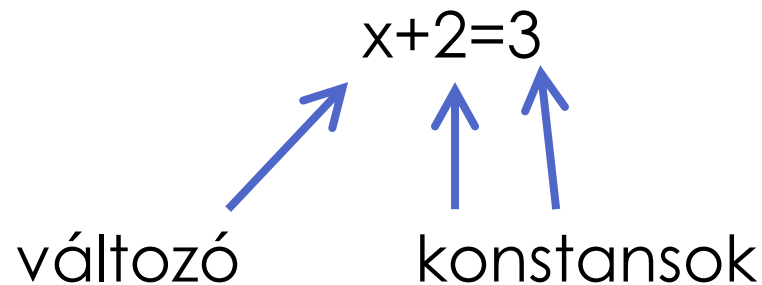
2.

Változók, műveletek,  
beolvasás, kiíratás

# Változók bevezetése

# A változókról általában

A változók a matematikából már ismertek

$$x+2=3$$


változó      konstansok

$$x=3-2=1$$


# Típus-független nyelvek


- Pl.: Python, Perl, Bash
- `x=3`
- `x=5.412`
- `x='z'`
- `x="alma"`

# Típusos nyelvek


- PL.: C, C++, C#, Java


- x egy egész szám:

- x=3 

- x='z' 

- x egy karakter:

- x=3 

- x='z' 

# Számábrázolás

- 32 bit: 11000011001111110100101001101001
  - Előjeles egész: -1019262359
  - Előjel nélküli egész: 3275704937
  - Két 16 bites egész: -15553, 19049
  - 4 karakter: ã, ?, J, i
- Nem egyértelmű, hogy mi a bitsorozat értéke
  - Egyértelművé válik ha definiáljuk az értelmezés módját → változó típusa

# Változók használata

# Változók adatai

- Fordító és programozó számára érdekes:
  - Típus (pl. int)
  - Név (pl. x)
- A program futása közben érdekes:
  - Érték (pl. 42)
  - Memóriacím (pl. 0x887af60)



# A program memóriája

- Futás előtt a teljes kódot betölti a RAM-ba
- A programba írt változók helye a futás közben nem változik
- A memória bájtokra van osztva, de szükség szerint kezelhető nagyobb egységekben

0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19

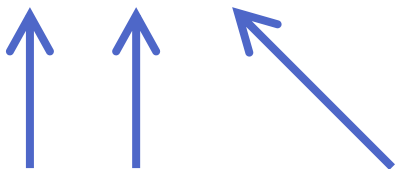
12	0	0	42	67	255	143	123	46	231
----	---	---	----	----	-----	-----	-----	----	-----

# Változók típusai

- char : 8 bit
- short : 16 bit
- int : 32 bit
- float : 32 bit
- double : 64 bit
- long int : 32/64 bit
- long long int : 64 bit
- long double : 80 bit (96)
- unsigned char : 8 bit
- unsigned short : 16 bit
- unsigned int : 32 bit

# Változók létrehozása

- `int a;`
- `int b=34;`
- `int apple=a*b+2;`
- `int x, y=5, z=2, w;`
- `float fl=3.45;`
- `char c='z';`



Típus      Név      Kezdeti érték

# Műveletek változókkal

- $x=y;$
- $x=3+4*5;$
- $x=((y+z)*3-12)/4;$
- $x=2*x;$
- $x=y*x+z-x+13;$
- $x=-y*y*y;$
- $x=y\%3;$
- $x=y=z=3;$

# Műveletek változókkal

○ $x+=3;$	[= $x=x+3]$
○ $x-=5;$	[= $x=x-5]$
○ $x*=2;$	[= $x=x*2]$
○ $x/=4;$	[= $x=x/4]$
○ $x+=(y-z)*3;$	[= $x=x+(y-z)*3]$
○ $x+=y+z-53;$	[= $x=x+(y+z-53)]$
○ $x\%=6;$	[= $x=x\%6]$
○ $x++;$	[= $x=x+1]$
○ $x--;$	[= $x=x-1]$

# Gyakorló feladatok

## ◦ 1.1 Feladat

- **x** legyen **a** negyedik hatványa osztva 3-mal.
- **y** legyen **b** háromszorosának és **c** kétszeresének különbsége.
- **z** legyen az előbb kiszámolt **x** és **y** összege.

## ◦ 1.2 Feladat

- Egy új változóba (**ab**) tárolja el **a** és **b** szorzatát, egy másikba (**bc**) **b** és **c** szorzatát, egy harmadikba pedig (**ac**) **a** és **c** szorzatát.
- **x** legyen a három szorzat összege
- **y** legyen a három szorzat szorzata
- **z** legyen **ab** és **ac** különbsége

# Beolvasás és kiíratás

# Bemenet és kimenet

- Minden, felhasználók számára készülő programnak kell kommunikálnia a felhasználóval.
- Bemenet: a felhasználó ezen keresztül tud beleszólni a program működésébe.
- Kimenet: a program által adott visszajelzések és információk.



# Bemenet és kimenet

- A kurzus során csak a konzolos bemenet és kimenet használatát tárgyaljuk.
- A megfelelő elemek használatához a következő sorra szükség lesz minden forrásfájl legelején:

`#include <stdio.h>`

# printf

- Szöveg és változók kiíratására alkalmas
- Szöveg kiírása:
  - `printf("Ez egy pelda szoveg.");`
- Speciális karakterek (escape sequences) kezelése:
  - `printf("\tEz egy beljebb kezdett szoveg uj sorral a vegen.\n");`

# Változók kiírása

- A változó típusát és helyét a szövegben meg kell adni
- A kiíratandó változó(ka)t a szöveg után, vesszővel elválasztva kell megadni
  - `printf("Az x változó értéke: %d\t az y változóé: %f\n", x, y);`

# Változók kiírása

- Egész (int) típusú változó: %d
  - `printf("Az egész változó értéke: %d\n", i);`
- Karakter (char) típusú változó: %c
  - `printf("A karakter változó értéke: %c\n", c);`
- Előjel nélküli egész (unsigned int): %u
  - `printf("Az előjel nélküli változó értéke: %u\n", u);`
- Egész szám hexadecimális formában: %x
  - `printf("Az egész változó értéke hexadecimálisan: %x\n", x);`

# Változók kiírása

- Lebegőpontos (float) változó: %f
  - `printf("A lebegopontos valtozo erteke: %f\n", f);`
- Duplapontosságú (double) változó: %lf
  - `printf("A duplapontossagu valtozo erteke: %lf\n", d);`

# Változók kiírása

- Azt is ki tudjuk írni, hogy egy változó hol helyezkedik el a memóriában
- A változó helyét a memóriában a & jel segítségével lehet lekérdezni
- Memóriacím kiírása minden típusnál: %p
  - `printf("A változó a %p címen található\n", &i);`
- A memóriacím egy egész szám, amely hexadecimális formában jelenik meg

# Formázott kiíratás

- Kiírás fix hosszú helyre, jobbra igazítva:
  - `printf("%10d\n", i);`
- Kiírás fix hosszú helyre, balra igazítva:
  - `printf("%-10d\n", i);`
- Kiírás fix hosszú helyre, elejét 0-val feltöltve:
  - `printf("%010d\n", i);`
- Írja ki a pozitív előjelet is:
  - `printf("%+d\n", i);`

# Formázott kiíratás

- Lebegőpontos szám normálalakban:
  - `printf("%e\n", f);`
- Lebegőpontos szám a jobbnak tűnő alakban:
  - `printf("%g\n", f);`
- Lebegőpontos szám fix tizedes jeggyel:
  - `printf("%.6f\n", f);`



# Kifejezések kiíratása

- A printf nem csak változók értékét tudja kiírni, hanem kifejezéseket is.
- Ha ki szeretnénk íratni a változó (**x**) háromszorosának értékét, elég ennyi:
  - `printf("%d\n", 3*x);`

# scanf

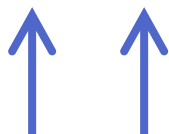
- Ezzel lehet az inputról beolvasni
- printf-hez hasonló szintaxis
- Lehet formázásokat megadni, de annak részleteit nem tárgyaljuk

## Beolvasás változókb

- A beolvasás valójában sosem változóba történik, hanem a memóriába
- Ha a memória megadott területén egy változó található, annak az értéke módosul
- A változó helyét a memóriában a & jel segítségével lehet lekérdezni

# Beolvasás változókba

- A változó típusát a printf-nél tanultaknak megfelelően jelölni kell
- A scanf-nek a változó címe kell, mivel a memóriába olvas be
  - `int i;`
  - `scanf("%d", &i);`



Változó típusa

Ez mindig kell a változó neve elé.

# Több változó beolvasása

- Több változót is lehet egyszerre beolvasni:
  - `int i, j, k;`
  - `char c;`
  - `scanf("%d %d %d %c", &i, &j, &k, &c);`
    - Beolvas három számot és egy karaktert
    - A számokat szóközzel elválasztva vagy új sorba kell írni, a karakter elé viszont nem kötelező a szóköz

# Szövegillesztés

- `scanf("%d,%d:%d");`
  - Csak akkor fogadja el a bemenetet, ha a megfelelő karakterek szerepelnek benne
  - 3,4:5 😊
  - 3 4 5 ❌
  - 3,4 5 ❌
  - 3:4:5 ❌
- Az új sor karaktert ne írjuk oda, az meg tudja zavarni a működést
  - `scanf("%d\n");` ❌

# Gyakorló feladatok

## ◉ 1.8 Feladat

- ◉ Olvasson be billentyűzetről két számot
- ◉ Írassa ki az első szám négyzetét
- ◉ Írassa ki a második szám ötszörösét

## ◉ 1.9 Feladat

- ◉ Olvass be a billentyűzetről három számot (**x**, **y**, **z**)
- ◉ Tárolja el és írassa ki  $3 * \mathbf{x} - \mathbf{y} * \mathbf{z}$  értékét
- ◉ Újabb változó felhasználása nélkül írassa ki a három szám szorzatát

# Gyakorló feladatok

## ◦ 4.1.1 Feladat

- Olvasson be két számot a billentyűzetről és tárolja el két változóba
- Cserélje meg a két változó értékét