



# Programozás I.

7.

Mutatók használata

# Memóriacím

# Memóriacím

- Minden változó egy adott címen szerepel a memóriában
- Nem csak a változó értékét, de annak címét is elérhetjük

0x10 0x11 0x12

char	char	int	char	short	char
------	------	-----	------	-------	------

0x16 0x17

0x19

# Mutatók

Változók

# Változók

- Mutatók létrehozása

`int* pi;`      `char* pc;`  
                  ↑                   ↑  
                  Típus

- Több változó deklarálása egyszerre

`int* p, q`  
`int *p, *q`  
`int p, *q`

//a **p** mutató, a **q** azonban nem  
//a **p** és a **q** is mutató  
//a **p** nem mutató, a **q** igen

# Operátorok

- &                      Címképző operátor
  - `int* intptr = &x;`
- \*                      Indirekció operátor
  - `int y = *intptr;`
  - `*intptr = 42;`

# Változók

- Eltárolhatjuk egy változó helyét

```
char a = 'a';
```

```
char b = 'b';
```

```
int i = 42;
```

```
int* p = &i;
```

0x10	0x11	0x12	0x16
'a'	'b'	42	0x12

# Változók

- Eltárolhatjuk egy változó helyét

```
char a = 'a';
```

```
char b = 'b';
```

```
int i = 42;
```

```
int* p = &i;
```

```
*p = 43;
```

0x10 0x11 0x12

0x16

'a'	'b'	43	0x12
-----	-----	----	------



# Változók

## • Értékadás

```
int *p, *q;
```

```
int i = 3;
```

```
p = &i;
```

```
q = p;
```

```
*p = 2;
```

```
*q = 1;
```

```
i = ?
```

//a **p** és a **q** is mutató

//a **p** tárolja az **i** címét

//a **q** tárolja azt a címet, amit **p**

//a mutatott címet szereplő érték legyen 2

//a mutatott címet szereplő érték legyen 1

# Változók

- Mutatóra mutató mutató

```
int *p, **q;
```

```
int i = 3;
```

```
int j = 42;
```

```
p = &i;
```

```
q = &p;
```

```
//vesszük a p címét
```

```
*p = 2;
```

```
**q = 1;
```

```
//*q) megegyezik p-vel, p pedig az i címe
```

```
*q = &j;
```

```
**q = 1;
```

```
// ?
```

```
i = ?
```

# Változók élettartama

- Minden változó a saját blokkjának végéig érhető el
- Egy mutató, amely egy megszűnt változó helyét címzi, hibás eredményhez, vagy programhibához vezet!

# Nulla értékű mutató

- Ha egy mutató érvénytelen címre mutat, akkor nem szabad használni
- A programban nem tudjuk lekérni, hogy a cím érvényes-e, így erről magunknak kell gondoskodni
- Ha a mutató nem mutat sehova, akkor mutasson a 0 címre (0, 0x0, NULL)

# Gyakorló feladatok

## ◦ 7.1.1 Feladat

- Hozzon létre egy egész számot tároló változót
- Hozzon létre egy mutatót, amely az előzőleg létrehozott változóra mutat
- Olvasson be billentyűzetről egy számot a mutatón keresztül
- Írassa ki a képernyőre a szám értékét

## ◦ 7.1.2 Feladat

- Hozzon létre két, egész számokat tároló változót
- Hozzon létre egy mutatót, amely az előzőleg létrehozott egyik változóra mutat
- Olvasson be billentyűzetről egy számot a mutatón keresztül
- Olvasson be billentyűzetről még egy számot a mutatón keresztül úgy, hogy az a másik változó értékét befolyásolja
- Írassa ki a képernyőre a két szám értékét

# Mutatók

Tömbök

# Tömbök

- A tömb hasonlóan viselkedik a mutatókhoz
- Ha a tömböt nem indexeljük, akkor az első elemének a címét kapjuk
  - `int* ptr = tomb;`
- A mutató is használható tömbként, ugyanúgy lehet indexelni
  - `ptr[3]=8;`

# Tömbök

- Ha egy mutató értékét növeljük, akkor a növelés mértékét megszorozza a típusának méretével
  - `int *ptr = tomb;`
  - `tomb + 3 == &tomb[3]`
  - `*(tomb + 3) == tomb[3]`
  - `ptr + 3 == &ptr[3]`
  - `*(ptr + 3) == ptr[3]`
  - `*(ptr + 3) == tomb[3]`



# Tömbök

```
int array[10] =  
    { 0, 1, 2, 3, 4,  
      5, 6, 7, 8, 9 }  
  
int* p = array;           //0x10  
int* p5 = p + 5;          //0x24  
int* p0 = p;              //0x10
```

0x10 0x14 0x18 0x1C 0x20 0x24 0x28 0x2C 0x30 0x34

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

# Tömbök

```
int array[10] =  
    { 0, 1, 2, 3, 4,  
      5, 6, 7, 8, 9 }  
  
int* p = array;           //0x10  
int* p5 = p + 5;          //0x24  
int* p0 = p;               //0x10  
*p0 = 10;
```

0x10	0x14	0x18	0x1C	0x20	0x24	0x28	0x2C	0x30	0x34
10	1	2	3	4	5	6	7	8	9

# Tömbök

```
int array[10] =  
    { 0, 1, 2, 3, 4,  
      5, 6, 7, 8, 9 }  
  
int* p = array;           //0x10  
int* p5 = p + 5;          //0x24  
int* p0 = p;              //0x10  
*p0 = 10;  
*p5 = 15;
```

0x10	0x14	0x18	0x1C	0x20	0x24	0x28	0x2C	0x30	0x34
10	1	2	3	4	15	6	7	8	9

# Tömbök

```
int array[10] =  
    { 0, 1, 2, 3, 4,  
      5, 6, 7, 8, 9 }  
  
int* p = array;           //0x10  
int* p5 = p + 5;          //0x24  
int* p0 = p;               //0x10  
*p0 = 10;  
*p5 = 15;  
*p  = 0;
```

0x10 0x14 0x18 0x1C 0x20 0x24 0x28 0x2C 0x30 0x34

0	1	2	3	4	15	6	7	8	9
---	---	---	---	---	----	---	---	---	---

# Tömbök

```
int array[10] =  
    { 0, 1, 2, 3, 4,  
      5, 6, 7, 8, 9 }  
  
int* p = array;           //0x10  
int* p5 = p + 5;          //0x24  
int* p0 = p;              //0x10  
*p0 = 10;  
*p5 = 15;  
*p  = 0;  
p[6] = 18;
```

0x10	0x14	0x18	0x1C	0x20	0x24	0x28	0x2C	0x30	0x34
0	1	2	3	4	15	18	7	8	9

# Tömbök

```
int array[10] =  
    { 0, 1, 2, 3, 4,  
      5, 6, 7, 8, 9 }  
  
int* p = array;           //0x10  
int* p5 = p + 5;          //0x24  
int* p0 = p;              //0x10  
*p0 = 10;  
*p5 = 15;  
*p = 0;  
p[6] = 18;  
array = p;
```

0x10 0x14 0x18 0x1C 0x20 0x24 0x28 0x2C 0x30 0x34

0	1	2	3	4	15	18	7	8	9
---	---	---	---	---	----	----	---	---	---

# Gyakorló feladatok

## ◦ 7.2.1 Feladat

- Olvasson be egy hatelemű tömböt, amely egész számokat tárol és töltse fel értékekkel
- Hozzon létre egy 10 elemű tömböt, amely mutatókat tárol
- Olvasson be tíz számot a képernyőről. Ha a szám szerepel a hatelemű tömbben, tárolja el annak a címét, ellenkező esetben tároljon el egy nullaértékű címet
- Jelenítse meg a 10 elemű tömb által mutatott értékeket és hozzájuk tartozó memóriacímeket

## ◦ 7.2.2 Feladat

- Olvasson be egy tízelemű tömböt, amely egész számokat tárol és töltse fel értékekkel
- Hozzon létre egy 10 elemű tömböt, amely mutatókat tárol az előző tömb megfelelő elemeire
- Rendezze a mutató tömböt a mutatott értékek szerinti növekvő sorrendbe
- Jelenítse meg a két tömb értékeit (a mutató tömb kiíratásakor jelenítse meg a címeket és a mutatott értékeket egyaránt)

# Mutatók

String



# String

- A szöveg reprezentálása egy karakter tömb segítségével valósítható meg
- A karaktersorozat végét egy '\0' speciális karakter jelzi
- A karaktertömb mérete a karaktersorozat hosszánál legalább egyel nagyobb kell, hogy legyen

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

'a'	'l'	'm'	'a'	'f'	'a'	'\0'			
-----	-----	-----	-----	-----	-----	------	--	--	--

# String

```
char text[10] = { 'a', 'l', 'm', 'a', 'f', 'a', '\0' };
```

```
char* p = text;
```

```
printf("%s", p)
```

```
scanf("%s", p);
```

```
printf("%p", p);
```

//6 karaktert jelenít meg

//a **scanf** mutatót vár,

de a string már eleve mutató

//**p** értékének kiírása memóriacímként

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

'a'	'l'	'm'	'a'	'f'	'a'	'\0'			
-----	-----	-----	-----	-----	-----	------	--	--	--

# String függvények

- String függvények használatához:
  - `#include <string.h>`
- Összehasonlítás:
  - `strcmp( str1, str2 )`
  - Eredménye
    - 1: **str1** lexikografikusan előrébb helyezkedik el
    - 0: **str1** és **str2** megegyezik
    - -1: **str2** lexikografikusan előrébb helyezkedik el

# String függvények

- Másolás:
  - strcpy( ide, innen )

```
char* innen = "almafa";  
char ide[10];  
strcpy(ide, innen);  
printf("%d: %s", strcmp(innen, ide), ide);
```

# Mutatók

Struktúrák

# Struktúrák

- Egy struktúra típusú változónak szintén le tudjuk kérni a címét
- Akár struktúrában is tárolhatunk memóriacímeket

# Struktúrák

```
typedef struct computer
{
    CyberMedic* Medic;
    int Id;
} Computer;
```

```
typedef struct it
{
    char* Name;
    char* IpAddress
    int Port;
} CyberMedic;
```

```
int main()
{
    CyberMedic cm;
    cm.Name = "Admin";
    cm.IpAddress = "127.0.0.1";
    cm.Port = 80;
```

# Struktúrák

```
Computer c1;  
c1.Id = 0;  
c1.Medic = &cm;
```

```
printf("%s:%d", c1.Medic->IpAddress, c1.Medic->Port);  
printf("%p", &c1);
```

```
}
```

-> Mutatón keresztüli tagelérés  
`C1.Medic->Port == (*c1.Medic).Port`



# Gyakorló feladatok

## ◦ 7.3.1 Feladat

- Hozzon létre egy struktúrát, amely négy memóriacímet tárol:
  - Minimum
  - Maximum
  - Első előfordulás
  - Utolsó előfordulás
- Olvasson be egy tízelemű tömböt
- Olvasson be egy számot
- Tárolja el a struktúrában a tömb legkisebb, legnagyobb elemének a címét és a beolvasott szám első és utolsó előfordulásának a címét (vagy nullát)
- Jelenítse meg a struktúrában tárolt címeket és a címeken tárolt értékeket

# Gyakorló feladatok

## ◉ 7.3.2 Feladat

- ◉ Készítsen egy struktúrát, amely koordinátákat tárol ( $x, y, z$ )
- ◉ Hozzon létre egy tízelemű tömböt, amely koordinátákat tárol
- ◉ Egy mutató segítségével töltsse fel a tömböt értékekkel
- ◉ A mutatón keresztül jelenítse meg a tömb elemeit

## ◉ 7.3.3 Feladat

- ◉ Készítsen programot, amely háromszögek tárolására alkalmas. Egy háromszög három térbeli pont mutatóját tartalmazza
- ◉ Hozzon létre három koordinátát és egy háromszöget
- ◉ A háromszög struktúráján keresztül töltsse fel a koordinátákat a billentyűzetről olvasott adatokkal
- ◉ Írassa ki a háromszög adatait

# Mutatók

Függvények

# Függvények

- A mutatók használata függvények paramétereként praktikus lehet
  - Lehetőséget ad egy függvényen kívüli változó értékének beállítására
  - Nagy méretű változók esetén gyorsabb egy kis méretű memóriacím másolása

# Függvények

```
int increment(int* a, int b)
{
    *a = 42;           //a mutatott cím értéke legyen 42
    return (*a + b);
}

int main()
{
    int a = 1;
    int b = 2;
    int c = increment(&a, b);

    //a = ?  b = ?  c = ?
}
```

# Függvények

- Függvény soha ne adjon vissza olyan mutatót, amely bármely, a függvényben létrehozott változóra mutat!

# Gyakorló feladatok

## ◦ 7.4.1 Feladat

- Írjon függvényt, amely két memóriacímet kap paraméterül és megcseréli az értékeiket, amennyiben az első számé nagyobb
- Olvasson be egy tízelemű tömböt
- Rendezze a tömböt növekvő sorrendbe az előzőleg megírt függvény segítségével
- Jelenítse meg a tömböt

# Gyakorló feladatok

## ◉ 7.4.2 Feladat

- ◉ Készítsen egy koordináta struktúrát
- ◉ Írjon függvényt, amely egy koordináta struktúra címét kapja meg és felölti azt billentyűzetről olvasott értékekkel
- ◉ Írjon függvényt, amely egy koordináta struktúra címét kapja és megjeleníti azt a képernyőn
- ◉ Olvasson be egy tízelemű tömbbe koordinátákat az előbb megírt függvény segítségével
- ◉ Írassa ki a tömb elemeit