

Mutatók, dinamikus helyfoglalás

általános - típus *név
egész szám - int *poin1
valós szám - double *poin2
típus nélküli - void *poin3

```
In [6]: #include <stdio.h>
#include <stdlib.h>

int main() {
    int p = 12;
    int *poin;
    poin = &p;

    printf("\np - %d ", p);
    printf("\np címe - %p ", &p);
    printf("\npointer alapján - %d ", *poin);
}
```

p - 12
p címe - 0x7ffef3deec0c
pointer alapján - 12

Dinamikus helyfoglalás

Memóriafoglalás - **malloc()**

Szükséges mamóriaterület mérete - **sizeof()**

Foglalt memória felszabadítása - **free()**

```
In [8]: #include <stdio.h>
#include <stdlib.h>

int main() {
    double *ap = NULL;
    ap = (double*)malloc(sizeof(double));
    printf("%p\n", ap);
    if (ap != NULL) {
        *ap = 1.56;
    }
    printf("%lf", *ap);
    free(ap);
    ap = NULL;
}
```

0x55924468e880
1.560000

Tömb dinamikus létrehozása

```
In [13]: #include <stdio.h>
#include <stdlib.h>

int main() {
    int *xp = NULL;
    xp = (int*)malloc(5 * sizeof(int));
    if (xp != NULL) {
        xp[0] = 3;
        xp[4] = 12;
    }
    printf("\n%d. elem: %d", 0, xp[0]);
    printf("\n%d. elem: %d", 4, xp[4]);

    printf("\nTömb egyes elemeinek címe:");
    for (int i = 0; i < 5; i++){
        printf("\n&x[%d] = %p", i, &xp[i]);
    }
    printf("\nTömb címe: %p", &xp);

    free(xp);
    xp = NULL;
}
```

0. elem: 3
4. elem: 12
Tömb egyes elemeinek címe:
&x[0] = 0x55f9d82aa880
&x[1] = 0x55f9d82aa884
&x[2] = 0x55f9d82aa888
&x[3] = 0x55f9d82aa88c
&x[4] = 0x55f9d82aa890
Tömb címe: 0x7fff6cac0aa0

Mintafeladat

Hozzunk létre dinamikusan felhasználó által megadott méretű tömböt, töltsük fel véletlen számokkal és írjuk ki a képernyőre a tömb tartalmát.

```
In [18]: #include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main() {
    //elemek számának beolvasása
    int n;
    printf("Kérem a tömb elemeinek számát: ");
    scanf("%d", &n);
    //tömb dinamikus létrehozása
    int *a = (int*)malloc(n * sizeof(int));
    if (a == NULL) {
        printf("Hiba a memóriafoglalásnál!");
        exit(0);
    }
    //tömb feltöltése véletlen számokkal és tömb kiírása
    srand(time(NULL));
    for (int i = 0; i < n; i++) {
        a[i] = rand() % 51;
        printf("%d, ", a[i]);
    }
    printf("\n");

    //lefoglalt memória felszabadítása
    free(a);
}
```

Kérem a tömb elemeinek számát:

41, 29, 50, 32, 48, 23, 26, 44, 5, 46, 16, 41, 32, 41, 17, 26, 1, 24, 12, 37,

1. Feladat

Kérjük be a tömb elemeinek számát (n), majd hozzunk létre dinamikusan egy n-elemű, egész szám típusú tömböt! A tömbbe generáljunk ki -100 és +100 közötti véletlen számokat és írjuk ki a tömböt a képernyőre.

- számítsuk ki az elemek összegét, átlagát!
- számoljuk meg a tömbben található páros és páratlan számokat!

A program kimenete:

Kérem a tömb elemeinek számát: 12

71, -90, 90, 60, -69, 0, -46, 43, -77, -92, -45, 13

Számok összege: -142

Számok átlaga: -11,833

Páros számok darabszáma: 6

Páratlan számok darabszáma: 6

2. Feladat

Kérünk be egy n számot, majd hozzunk létre dinamikusan két n-elemű, valós szám típusú tömböt (a, b)! A tömbökbe generáljunk ki 0.00-9.99 közötti véletlen számokat és írjuk ki a tömb elemeit egymás alá, majd számoljuk ki a tömbök összegét és különbségét (a két tömb azonos indexű elemeit összeadva ill. kivonva)!

A program kimenete:

Kérem a tömbök elemeinek számát: 10

5.33, 4.54, 8.80, 8.09, 9.24, 3.17, 8.22, 0.95, 3.28, 9.32

2.28, 1.14, 2.61, 2.51, 5.62, 9.88, 4.70, 9.40, 2.70, 5.85

Számok összege:

7.61, 5.68, 11.41, 10.60, 14.86, 13.05, 12.92, 10.35, 5.98, 15.17

Számok különbsége:

3.05, 3.40, 6.19, 5.58, 3.62, -6.71, 3.52, -8.45, 0.58, 3.47

3. Feladat

Kérjük be a tömb elemeinek számát (n), majd hozzunk létre dinamikusan egy n-elemű, valós szám típusú tömböt! Olvassuk be a tömb elemeit, majd keressük meg a tömbben a legkisebb és a legnagyobb elemet!

- keresse meg a második legnagyobb elemet
- keresse meg a harmadik legkisebb elemet

A program kimenete:

Kerem a tomb elemeinek szamat: 7

1. elem: -28.17

2. elem: 3.5

3. elem: -62.8

4. elem: 37.05

5. elem: 28.1

6. elem: 72.66

7. elem: -52.4

Legkisebb elem: -62.80

Legnagyobb elem: 72.66

4. Feladat

Készítsen programot, ami létrehoz két tetszőleges méretű tömböt dinamikusan, majd feltölti azokat véletlen számokkal. Hozzon létre egy 3. tömböt, amiben **pontosan** elfér a két másik tömb tartalma, majd másolja át az elemeket.

5. Feladat

Kérjük be a tömb elemeinek számát (n), majd hozzunk létre dinamikusan kettő n-elemű, természetes szám típusú tömböt! A tömböket mindig csak az előző értéknél nagyobb számmal lehet bővíteni. A két tömb elemeit olvassa be egy harmadik tömbbe olyan módon, hogy azok rendezve kerüljenek új helyükre.

A program kimenete:

Kérem az első tömb elemeinek számát: 5

1. elem: 8

2. elem: 5

Hiba! Nem adhat meg az előzőnél kisebb értéket!

2. elem: 10

3. elem: 15

4. elem: 22

5. elem: 25

Kérem a második tömb elemeinek számát: 3

1. elem: 12

2. elem: 13

3. elem: 15

Az új tömb elemei: 8, 10, 12, 13, 15, 15, 22, 25

6. Feladat

Kérjük be a tömb elemeinek számát, majd hozzunk létre dinamikusan egy n-elemű, természetes szám típusú tömböt! A tömböt töltsük fel felhasználó által megadott intervallumról származó véletlen számokkal és írassuk ki azt. A páros és páratlan számokat különítse el két külön tömbben, majd írassuk ki őket!

7. Feladat

Kérjük be a tömb elemeinek számát, majd hozzunk létre dinamikusan egy n-elemű, karakter típusú tömböt! A tömböt töltsük fel az angol ABC véletlen nagy- és kis betűivel, majd írassuk ki azt. A nagy és kis betűket különítsük el két külön tömbbe, majd írassuk ki a szétválogatott tömbök tartalmát!

In []: