

9. Gyakorlat

Célkitűzések:

- Pointerek
- Dinamikus helyfoglalás: [malloc](#), [calloc](#), [free](#), [realloc](#)
- [qsort](#)

Mintafeladatok:

1. Írjunk programot, amely kiírja egy természetes számot tartalmazó változó címét és értékét.

```
unsigned int number;
printf("Enter a number:");
scanf("%ui", &number);
printf("The reference of the number is: %i\n", &number);
printf("The value of the number is: %i\n", number);
```

2. Foglaljunk le egy egész számokat tartalmazó egydimenziós tömböt dinamikusan, töltsük fel véletlenszerű értékekkel, írjuk ki azokat, majd szabadítsuk fel a memóriát.

```
//deklaráció
unsigned int n;
int *items;
//beolvasás
printf("Enter the number of items:");
scanf("%ui", &n);
//dinamikus helyfoglalás
items = calloc(n, sizeof (int));
//items = malloc(n * sizeof(int));
//helyfoglalás ellenőrzése
if(!items) {
    printf("Error when allocating memory!\n");
    exit(-1);
}
//véletlen szám generálás
srand(time(0));
```

```

for (int i = 0; i < n; ++i) {
    items[i] = rand() % 100;
}
//kiíratás
printf("The generated numbers are: ");
for (int i = 0; i < n; ++i) {
    printf("%i ", items[i]);
}
//felszabadítás
free(items);

```

3. Foglaljunk le egy valós számokat tartalmazó kétdimenziós tömböt dinamikusan, olvassuk be a tartalmát állományból, írjuk ki a mátrixot a képernyőre, majd szabadítsuk fel a memóriát.

```

//deklaráció
unsigned int rows, cols;
float **items;
//fájl megnyitás és ellenőrzés
if(!freopen("input.txt", "r", stdin)){
    printf("Error when opening file!\n");
    exit(1);
}
//beolvasás és dinamikus helyfoglalás
scanf("%ui", &rows);
scanf("%ui", &cols);
items = (float**)calloc(rows, sizeof (float*));
//items = (float**)malloc(rows * sizeof (float*));
if(!items) {
    printf("Error when allocating memory!\n");
    exit(-1);
}
for (int i = 0; i < rows; ++i) {
    items[i] = (float*)calloc(cols, sizeof (float));
    //items[i] = (float*)malloc(cols * sizeof (float));
    if(!items[i]) {
        printf("Error when allocating memory!\n");
        exit(-1);
    }
}
}

```

```

for (int i = 0; i < rows; ++i) {
    for (int j = 0; j < cols; ++j) {
        scanf("%f", &items[i][j]);
    }
}
//kiíratás
for (int i = 0; i < rows; ++i) {
    for (int j = 0; j < cols; ++j) {
        printf("%6.2f ", items[i][j]);
    }
    printf("\n");
}
//fájl átirányítás és memória felszabadítása
freopen("CON", "r", stdin);
for (int i = 0; i < rows; ++i) {
    free(items[i]);
}
free(items);

```

Kitűzött feladatok:

1. (*) Írj programot, amely **összead két valós számot** pointererek segítségével.
2. (*) Írj programot, amely **kicserél két egész számot** pointererek segítségével.
3. (**) Írj programot, amely kiszámolja egy egész számokat tartalmazó egydimenziós tömb **elemeinek összegét** pointererek felhasználásával.
4. (**) Írj programot, amely kiszámolja egy egész számokat tartalmazó egydimenziós tömb **elemeinek átlagát** pointererek felhasználásával.
5. (**) Írj programot, amely kiírja egy valós számokat tartalmazó egydimenziós tömb elemeit **fordított sorrendben**.
6. (***) Írj programot, amely rendezi növekvő sorrendbe egy egydimenziós tömb elemeit (qsort).
7. (***) Írj programot, amely rendezi csökkenő sorrendbe egy egydimenziós tömb elemeit (qsort).
8. (***) Írj programot, amely kiírja az **angol ABC betűit** pointererek segítségével.

9. (***) Írj programot, amely **tükrözi egy szó karaktereit**.
10. (***) Írj programot, amely feltölt egy kétdimenziós tömböt véletlen egész számokkal két határ között, majd kiírja a mátrix tartalmát a képernyőre. Használd a dinamikus helyfoglalást.
11. (***) Írj programot, amely feltölt egy kétdimenziós tömböt állományból, majd kiírja a mátrix **alsó háromszögét** a képernyőre. Használd a dinamikus helyfoglalást.
12. (***) Írj programot, amely feltölt egy kétdimenziós tömböt állományból, majd kiírja a mátrix **felső háromszögét** a képernyőre. Használd a dinamikus helyfoglalást.
13. (***) Írj programot, amely kiszámolja egy mátrix **soronkénti összegeit**. Használd a dinamikus helyfoglalást. Az elemeket állományból olvasd be.
14. (***) Írj programot, amely meghatározza egy mátrix **oszloponkénti legnagyobb elemeit**. Használd a dinamikus helyfoglalást. Az elemeket generáld véletlenszerűen két adott határ között.
15. (***) Írj programot, amely kiszámolja egy max 3x3 - as mátrix **determinánsának** értékét. Használd a dinamikus helyfoglalást.
16. (***) Írj programot, amely kiszámolja két tetszőleges **mátrix szorzatát**. Ellenőrizd, hogy alkalmazható-e a szorzás művelet a két mátrixon. Használd a dinamikus helyfoglalást.

További példafeladatok: [itt](#)

További gyakorló feladatok:

1. Oldj meg minden egydimenziós tömbbel kapcsolatos feladatot a dinamikus helyfoglalás segítségével.
2. Oldj meg minden kétdimenziós tömbbel kapcsolatos feladatot a dinamikus helyfoglalás segítségével.