

## 4. hét

### Az óra témája

Az óra célja, hogy a hallgatók megismerjék és képesek legyenek programozási feladatokban használni az egydimenziós és többdimenziós tömböket is, mint összetett adatstruktúrákat.

### Mintaprogram

A következő példában egy 100 elemű egész típusú tömböt töltünk fel négyzetszámokkal.

### Forráskód

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int t[100];
    int f;

    t[0] = 0;
    t[1] = 1;
    t[2] = 4;
    t[3] = 9;
    t[4] = 16;
    // ... es így tovább, vagy ciklussal...

    for ( f=0; f<100; f++ )
    {
        t[f] = f * f;
    }

    for ( f=0; f<100; f++ )
    {
        printf ( "%8d", t[f] );
    }

    printf ( "\ntomb merete: %d\n", sizeof(t) );
    printf ( "f erteke a ciklus utan: %d\n", f );
    printf ( "t tomb elemeinek merete: %d\n", sizeof(t[0]) );

    // printf ( "%d", t[65535] );          // tulindexeles
    printf ( "t tomb elemeinek max szama: %d\n", sizeof(t)/sizeof(t[0]) );

    return 0;
}
```

## Mintaprogram

A következő program egy két dimenziós egész típusú tömbben tárolja egy 10x10-es szorzótábla értékeit egymásba ágyazott ciklus segítségével.

## Forráskód

```
#include <stdio.h>
#include <stdlib.h>

int main ()
{
    int t[10][10];
    int f, g;

    for ( f=0; f<10; f++ )
    {
        for ( g=0; g<10; g++ )
        {
            t[f][g] = (f+1) * (g+1);
        }
    }

    for ( f=0; f<10; f++ )
    {
        for ( g=0; g<10; g++ )
        {
            printf ( "%4d", t[f][g] );
        }
        printf ( "\n" );
    }
    return 0;
}
```