

6.hét

Az óra témája

Az óra célja, hogy a hallgató megismerkedjen a pointer fogalmával, képes legyen műveleteket végezni ponterekkel és értse a viszonyukat a hagyományos változókhoz. A hallgatók legyenek képesek függvények deklarálására és hívására cím szerinti paraméterátadással, és tömbök, valamint tömbelemek címzésére ponterekkel.

Példaprogram

A következő programban bemutatásra kerül a címképző operátor, a pointer és a pointeraritmetika.

Forráskód

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

int main()
{
    int a;
    int* papi;

    a = 10; // a erteke 10
    papi = &a; // papi erteke az a valtozo cime

    printf( "Ertek\t Cim\n" );

    //kiirjuk a erteket es a cimet
    printf( "%d\t%X\n", a, &a );
    //kiirjuk papi által mutatott valtozo erteket, es papi erteket
    printf( "%d\t%X\n\n", *papi, papi );

    a++;
    printf( "%d\t%X\n", a, &a );
    printf( "%d\t%X\n\n", *papi, papi );

    (*papi)++;
    printf( "%d\t%X\n", a, &a );
    printf( "%d\t%X\n\n", *papi, papi );

    papi++;
    printf( "%d\t%X\n", a, &a );
    printf( "%d\t%X\n\n", *papi, papi );

    return 0;
}
```

Példaprogram

A következő példa cím szerinti paraméterátadást mutat be. A függvény az átadott memóriacímen tárolt értéket emeli a négyzetére, eredménye a hívó függvényben is látszani fog.

Forráskód

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

void square( int* pa );

int main()
{
    int a;
    a = 3;
    printf( "%d\n", a );
    square( &a );
    printf( "%d\n", a );
    return 0;
}

void square( int* pa )
{
    *pa = (*pa) * (*pa);
}
```

Példaprogram

A következő program tömb és tömbelemek címzését mutatja be pointerekkel.

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

void feltolt( int* p );
void kivonas( int* q );
void print( int* z, int size );

int main()
{
    int t[10];

    feltolt( t );
    // feltolt( &t[0] );
    print( t, 10 );
    // print( t, sizeof(t)/sizeof(t[0]));
    printf( "---\n" );
    kivonas( t );
    print( t, 10 );

    return 0;
}

void feltolt( int* p )
{
    int f;
```

```
    for( f=0; f<10; f++ )
    {
        *p = f * f;
        p++;
    }
}

void kivonas( int* q )
{
    int f;

    for( f=0; f<10; f++ )
    {
        (*q) = (*q) - 1;
        q++;
    }
}

void print( int* z, int size )
{
    int f;

    for( f=0; f<size; f++ )
    {
        printf( "%2d\n", *z );
        z++;
    }
}
```