

5. hét

Az óra témája

A gyakorlat célja, hogy a hallgatók megismerjék és képesek legyenek használni a sztring adatstruktúrát, valamint képesek legyenek függvényekbe szervezni a kódjukat.

Példaprogram

A következő program megmutatja, hogy lehetséges sztringek számára tárhelyet foglalni, illetve sztringet inicializálva definiálni. A program szintén bemutatja, hogyan lehetséges tudunk sztringeket másolni C nyelven.

Forráskód

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

int main()
{
    char s[16];
    char s1[20];
    int len, siz, i = 0;

    s[0] = 'K';
    s[1] = 'o';
    s[2] = 'v';
    s[3] = 'a';
    s[4] = 'c';
    s[5] = 's';
    s[6] = '\0';
    s[7] = '\0'; // string lezaro nulla

    while(i < 8) // kiiratas ciklussal
    {
        printf("s[%d]\t%c\t%3d\n", i, s[i], s[i]);
        i++;
    }
    printf("\n");
    printf( "%s\n", s ); // string kiiratas printf()-el

    strcpy( s, "Kando Kalman" ); // "Kando Kalman"-t masolja s-be
    // s[0]='K';s[1]='a';s[2]='\n'; ...
    strncpy( s1, s, sizeof(s1) ); // sizeof(s1) szamu bajtot masol
    // s kezdocimtol -> s1 kezdocimre
    // s[0]=s1[0];s[1]=s1[1]; ...

    printf( "%s\n", s );
    printf( "%s\n", s1 );

    len = strlen( s1 ); // egy string hosszat adja vissza
```

```

    siz = sizeof( s1 );           // tarolo terület merete

    printf( "length: %d\n", len);
    printf( "size:   %d\n", siz);

    return 0;
}

```

Magyarázat

A program két megoldást is mutat sztringek másolására, először a gyakran használt ám nem biztonságos `strcpy` függvény segítségével, másodsor az `strncpy` függvény használatával, amelynek a maximálisan átmásolható bájtok számát is meg kell adni, ami garantálja, hogy hiányzó sztringzáró `'\0'` karakter nélkül sem fog a másolást végző függvény a program számára nem lefoglalt memóriaterületre írni.

Példaprogram

A következő programban egy téglalap kerületét és egy téglalap területét számoló függvényt fogunk írni, amiknek bemenő paramétere az oldalak hossza. A függvények két `int` típust várnak bemenő paraméterként, visszatérési értékük pedig `int`.

Forráskód

```

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

int kerulet( int a, int b );           // függvény deklaracio
int terület( int a, int b );

int main( void )
{
    int x, y;
    int k, t;
    x = 9; y = 5;

    k = kerulet( x, y );
    t = terület( x, y );
    printf( "k: %d, t: %d\n", k, t );
    // ugyanaz mint...
    printf( "k: %d, t: %d\n", kerulet( x, y ),
            terület( x, y ));

    return 0;
}

int kerulet( int a, int b )           // függvény definicio
{
    int r;
    r = 2 * ( a + b );
    return r;
}

```

```
int terület( int a, int b )  
{  
    return ( a * b );  
}
```

Magyarázat

A függvények hívásakor érték szerinti paraméterátadás történik, szemben a későbbiekben tanulandó címszerinti paraméterátadással.