

7. hét

Az óra témája

Az óra célja, hogy a hallgatók képesek legyenek összetett adattípusokat, tömböket és sztringeket pointerrek segítségével kezelni.

Mintafeladat

A következő mintaprogram sztringek (karakter tömbök) pointerrel való kezelését mutatja be egyszerű műveleteket végző függvényeken keresztül. Három függvény szerepel a kódban:

<code>void reset_str(char* str)</code>	– egy sztring tartalmát törli ('x'-ekre cseréli),
<code>void swap(char* s1, char* s2)</code>	– a két pointeren tárolt sztring tartalmát kicseréli,
<code>int szamol(char* s, char c)</code>	– a paraméterben átadott sztringben számolja meg a megadott karaktert.

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

void reset_str( char* str );
void swap( char* s1, char* s2 );
int szamol( char* s, char c );

int main()
{
    int c;
    char s1[50], s2[50];

    strncpy( s1, "Kando Kalman", 50 );
    strncpy( s2, "s2 string", 50 );

    c = szamol(s1, 'a');
    printf( "a vizsgalt karakterek szama: %d\n\n", c );

    printf( "%s <-> %s\n", s1, s2 );
    swap( s1, s2 );
    printf( "%s <-> %s\n\n", s1, s2 );

    printf( ">>>%s<<<\n", s2 );

    reset_str( s2 );
    // reset_str( &s1[0] );
    printf( ">>>%s<<<\n", s2 );

    return 0;
}

int szamol( char* s, char c )
```

```

{
    int f, n, len;
    len = strlen(s);
    n = 0;
    for( f=0; f<len; f++ )
    {
        if ( s[f] == c )
        {
            n++;
        }
    }

    return ( n );
}

void swap( char* s1, char* s2 )
{
    char tmp[50];

    strncpy( tmp, s1, 50 );
    strncpy( s1, s2, 50 );
    strncpy( s2, tmp, 50 );
}

void reset_str( char* str )
{
    int f;

    /*
    for( f=0; f<strlen(str); f++ )
    {
        str[f] = 'x';
    }
    */

    while( *str != '\0' )
    {
        *str = 'x';
        str++;
    }
}

```

Mintaprogram

A következő program *brute force* támadással próbál megtalálni egy, a program elején megadott három karakterből álló jelszót. A meghívott függvény fix három karakterre vizsgál, ha megtalálta a jelszót, kiírja a képernyőre, megjeleníti hány lépésből találta meg, majd visszatér.

Forráskód

```

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

```

```

void bruteforce(char* p);

int main()
{
    char pw[4] = "aSd";
    printf( "The password is: %s\n", pw );
    bruteforce(pw);
    return 0;
}

void bruteforce(char* p) {
    unsigned char a, b, c;
    char psw[] = { 0, 0, 0, 0 };
    int count = 0;
    for( a=' '; a<=128; a++ )
    {
        psw[0]=a;

        for( b=' '; b<=128; b++ )
        {
            psw[1] = b;

            for( c=' '; c<=128; c++ )
            {
                count++;
                psw[2] = c;
                // check the password
                if(!strcmp(p,psw,4))
                {
                    printf( "bruteforce psw:  %s\n", psw);
                    printf( "count = %d\n", count);
                    return;
                }
            }
        }
    }
}

```

Gyakorlófeladat

Oldja meg rekurzióval az előző feladatot, úgy, hogy a program nem fix karakterszámú kódot keres, hanem tetszőlegesen hosszú jelszóra működjön.