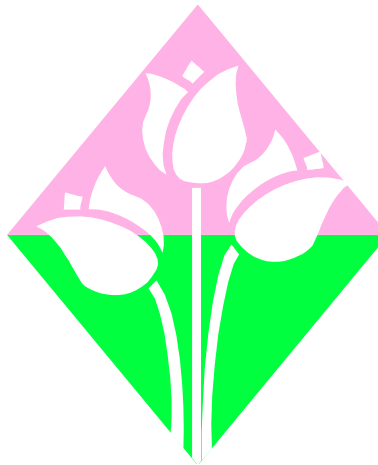
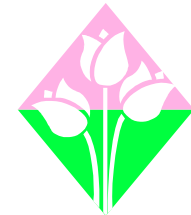


□ Könyvtári függvények I.

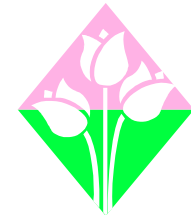
- Input / output kezelés függvényei
- Képernyőkezelés karakteres üzemmódban
- Adatkonverziós függvények



□ Input / output kezelés függvényei

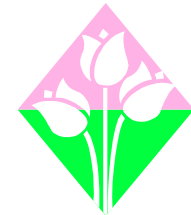


A C nyelvben az input / output perifériák, mint pl. a billentyűzet, karakteres képernyő, karakteres nyomtató, egyéb párhuzamos és soros kapcsolódási felületet használó egységek (szalagegység, lyukszalagolvasó / lyukasztó, rajzgép, stb.) kezelése a szabványos szinten nem tér el a fájlok **karakterfolyamos** (stream) elvű kezelésétől, azaz az adatátvitel a számítógép és a periféria között fájlok írásával és olvasásával valósul meg. DOS operációs rendszeren üzemeltetve, a C program **indítása pillanatában definiál öt szabványos fájlnevet** és megnyitja a hozzájuk rendelt "fájlokat", azaz perifériákat, amelyeket azonnal használhatunk a megszokott karakterfolyam-kezelő függvényekkel.



Ezek a standard logikai fájlnevek a következők:

Név	Periféria	Irány
□ stdin	billentyűzet (átirányítható)	input
□ stdout	képernyő (átirányítható)	output
□ stderr	képernyő (hibaüzenetek)	output
□ stdaux	1. soros kommun. csat. (COM1)	i / o
□ stdprn	1. nyomtató port (PRN, LPT1)	output



A szabványos fájlnevek szöveges állományokat jelölnek. Ezen szabványos input és output adatfolyamok kezelésére a már korábban megismert **printf()**, **scanf()** függvényeken, valamint a fájlok folyamszintű kezelésének függvényein túl ismerjünk meg két pár, a billentyűzet és a képernyő kezelésére szánt lehetőséget:

A **getchar()** és a **putchar()** makrók egyetlen karakternek a billentyűzetről való bekérésére, ill. a képernyőre való kiírására alkalmasak.

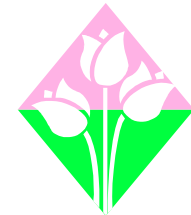
Formájuk:

<karakterváltozó> = getchar()
putchar(<karakterváltozó>)



A `getchar()`, szemben a `getch()` függvénnyel, puffertelt inputot végez, sorvégként kezeli az Enter karaktert és a Ctrl+Z, Enter billentyű-kombinációra `-1 == EOF` értéket ad, ily módon alkalmas szövegfájloknak a billentyűzetről történő bevitelére. Pl.:

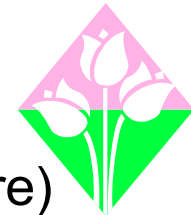
```
#include <stdio.h>                                     // Karakterek másolása a
main()                                                  // billentyűzetről a képernyőre
{ int bill;                                           // Ctrl+Z, Enter nyomásáig
  while ( (bill = getchar() ) != EOF ) // EOF == -1
    putchar( bill );
}
```



Az előző ciklus leállításához a getch() függvény alkalmazása esetén az EOF karaktert, azaz a '\x1A' karaktert kell megadnunk, és a bill változót elegendő **char** típusúra definiálni:

```
#include <stdio.h>
main()
{ char bill;
  while ( ( bill = getch() ) != '\x1A' )
    putchar( bill );
}
```

*// Karakterek másolása a
// billentyűzetről a képernyőre
// Ctrl+Z nyomásáig*



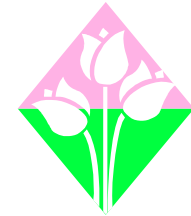
Sztringek beolvasása a szabványos inputról (billentyűzetről) a `gets()` függvénnyel, kiírása a szabványos outputra (képernyőre) a `puts()` függvénnyel végezhető. Bár a kiírásra alkalmas a `printf()` függvény is, szóközökkel tagolt sor beolvasására a `scanf()` függvény nem alkalmas.

Eltérés még, hogy a `gets()` a sort lezáró Enter hatására keletkező `'\n'` karaktert `'\0'` karakterré, a `puts()` pedig vissza, a `'\0'` karaktert `'\n'` karakterré alakítja, azaz a kiírás végén a következő sor elejére lép.

Formájuk:

`gets(<karaktervektor_név>)`

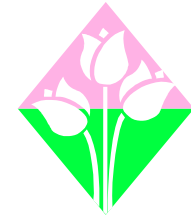
`puts(<szöveg>)`



Példa:

```
#include <stdio.h>
main()
{
    char svekt[128];           // +1 hely kell a '\0' karakternek
    puts("Legfeljebb 127 karakteres szöveg \
        beadása, lezárás Enter-rel:");
    gets(svekt);
    puts(svekt);             // a szöveget azonosíthatja a vektornév is
}
```


□ Adatkonverziós függvények

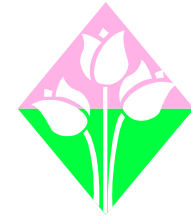


Különböző típusú számok és szöveges alak közötti átalakításra alkalmazható függvénypáros az **sprintf()** és az **sscanf()**, melyek alakja:

sprintf(<karaktervektor_név>, <formátumsztring>, <adatok_listája>)

sscanf(<számok_szövegalakban>, <formátumsztring>,
<változók_címeinek_listája>)

Pl.:

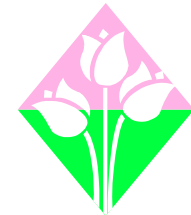


```
#include <stdio.h>
main()
{
    float szam = -123.456 ;
    char szovegalak[ 8 ] = "98.765";
    char string[ 28 ];
    sprintf(string, "%d %f", 12, szam );
    sscanf(szovegalak, "%f", &szam );
    printf( "szám=%f\n", szam );           // 98.765
    puts(string );                       // 12 -123.456
}
```

Az angol kisbetűk nagybetűvé alakítására alkalmas a

toupper(<kisbetű>)

függvény.



Pl.: `nagyp = toupper('p');`

Szövegalakban adott számértéket konvertálnak int, long int, ill. double típusú numerikus alakúvá a következő függvények:

atoi(<szövegalak>)

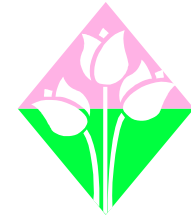
atol(<szövegalak>)

atof(<szövegalak>)

Pl.: `egesz = atoi ("-23");`

`hosszuegesz = atol("123456");`

`char s[12] = "-45.987"; duplapontos = atof(s);`



A fordított, numerikus alakból szövegalakra konvertálás függvényei az

itoa(<egész>, <szövegalak>, <alapszám>);

ltoa(<hosszúegész>, <szövegalak>, <alapszám>);

gcvt(<duplapontos>, <digitek>, <szövegalak>);

Pl.: **itoa**(egesz, s1, 10);
 ltoa(987456, s2, 10);
 gcvt(-645.65, 8, s3);