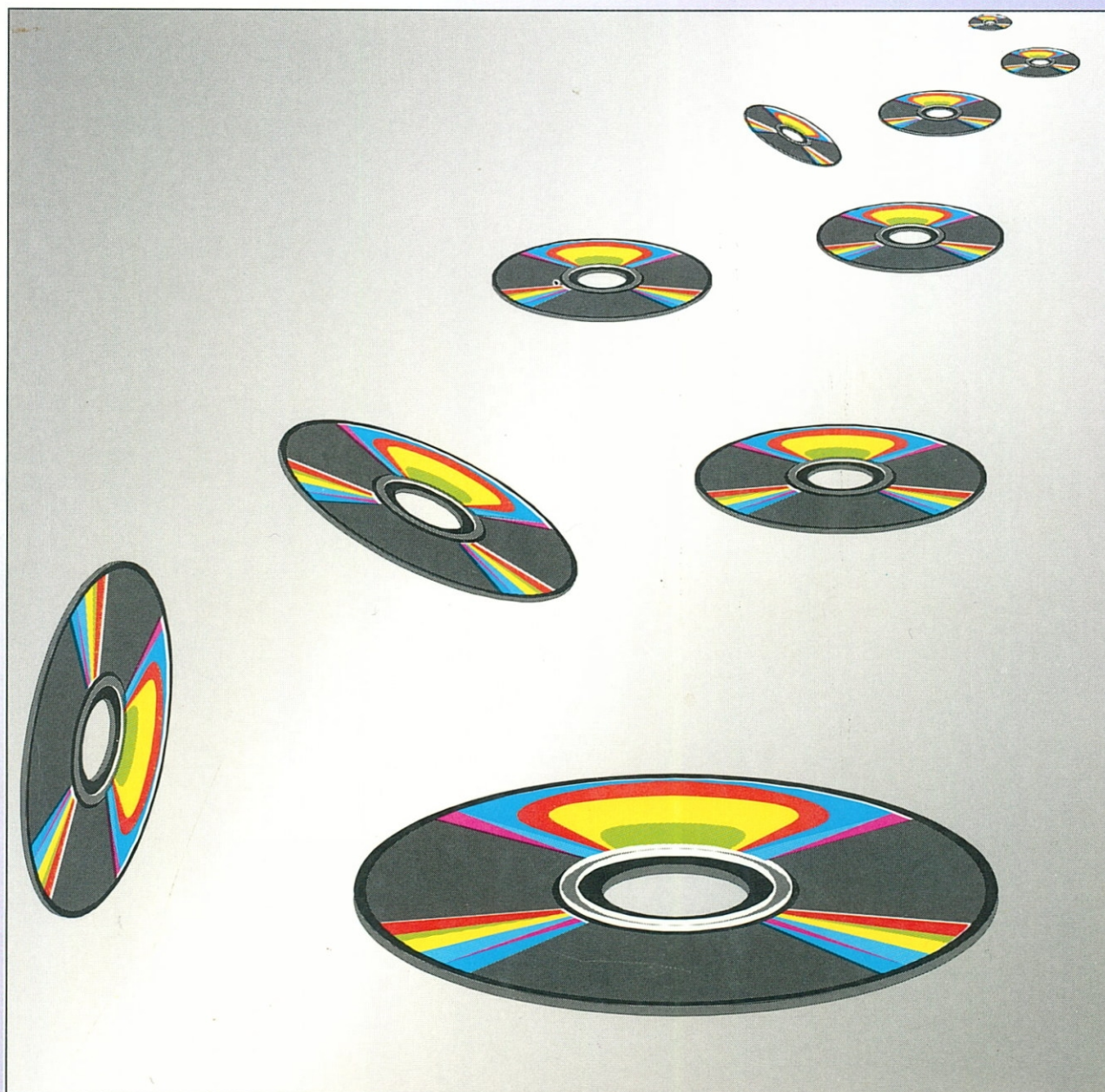
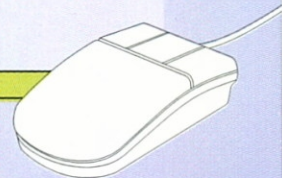


# SZÁMÍTÁSTECHNIKA

12-13 ÉVESEKNEK



Végh András

---

# **SZÁMÍTÁSTECHNIKA**

**12–13 éveseknek**

---

**Műszaki Könyvkiadó,  
Budapest**

E tankönyv használatát az Oktatási Minisztérium  
a 47.632/1997. XV. számon engedélyezte.

Sorozatszerkesztő:  
MEZŐSI KÁROLY

Szakmai és pedagógiai lektorok:  
REGELE GYÖRGY  
Dr. ZSAKÓ LÁSZLÓ

Grafikus:  
TAX ZSÓFIA

A fényképeket készítette:  
KOVÁTS TIBOR

© Végh András, 1997, 1999

© Hungarian edition Műszaki Könyvkiadó

ISBN 963 16 2343 2

# Tartalomjegyzék

<b>Az informatika mint tantárgy .....</b>	<b>5</b>
Néhány szó az informatikáról, és arról, amit tanulunk .....	5
Néhány jó tanács a tanuláshoz .....	5
Az informatikaterem rendje .....	7
Emlékeztető .....	7
<b>Informatikai alapok .....</b>	<b>10</b>
Mi az információ? .....	10
Az információ mennyisége .....	13
A jelek világa .....	17
Az adat .....	21
Elvárásolt jelek .....	24
Az ASCII-kód .....	27
Adatátvitel .....	29
<b>Hardvereszközök és kezelésük .....</b>	<b>34</b>
A személyi számítógép részei .....	34
Mit kell tudni a monitorokról? .....	38
A billentyűzet .....	40
Az egér .....	44
Nyomtatók .....	45
Szkenner (Scanner) .....	47
Lemezegységek és lemezek .....	48
Egyéb hardvereszközök .....	53

<b>Operációs rendszer</b> .....	<b>54</b>
Néhány szó a szoftverekről .....	54
Az operációs rendszerekről .....	55
Az MS-DOS használata .....	56
<b>Rendszerközeli programok</b> .....	<b>67</b>
Norton Commander .....	68
Windows 3.1 .....	74
A Windows kezelése .....	76
A Windows részei .....	78
<b>Algoritmusok, programok</b> .....	<b>87</b>
Az algoritmus .....	87
Algoritmusok leírása .....	89
Adattípusok .....	90
Mondatszerű algoritmusleírás .....	91
Röviden a Turbo Pascal fejlesztőrendszerrel .....	95
Elágazás, döntés .....	100
Hátultesztelő ciklus .....	103
Számlálóciklus .....	107
Rajzolás .....	110

# Az informatika mint tantárgy

## Néhány szó az informatikáról, és arról, amit tanulunk...

Lassan belépünk – vagy már be is léptünk – az **informatikai társadalmak** körébe. Sokat tudunk az információ tárolásának régi és új formáiról (könyv, könyvtár, adatbank, film, mágnesszalag, mágneslemez, CD, chip). Föld méretű információs és hírközlő hálózataink vannak (rádió, televízió, telefon, számítógépes hálózat), az információ feldolgozására pedig számítógépek állnak a rendelkezésünkre. Az emberiség kifejlesztett egy olyan technikát, amely a korábbinál sokkal több adat gyors feldolgozására és a gépek irányítására is alkalmas. Ez megkönnyítheti a munkát, a tanulást, és ráadásul még szórakoztató is lehet.

Az informatikaórán az információ kezelésével és használatával foglalkozunk. Megtanuljuk, hogy hogyan kell az informatikai eszközökkel dolgozni, játszani. Sok szó esik majd a számítógépről, a programok használatáról és a programozásról is.

Feltételezzük, hogy valamit már tudsz, ill. tanultál a számítógépről, de ha nem, akkor se keseredj el! Az *Emlékeztető* c. fejezetben összefoglaljuk a fontosabb gyakorlati ismereteket.

## Néhány jó tanács a tanuláshoz

A legfontosabb munkaeszközöd a tankönyv, a füzet és a számítógép (számítógép-hálózat) a benne lévő vagy vele elérhető programokkal, információkkal.

Ha elolvastál egy tananyagrészt, válaszolj minden ellenőrző kérdésre! Ha valamelyikre nem tudod a választ, olvasd el újra figyelmesen a megfelelő bekezdést! A példákat alaposan tanulmányozd! Ezek ugyanis segítenek Neked abban, hogy hogyan kell megoldani a feladatokat és elvégezni a gyakorlatokat. Lehetőleg mindet oldd meg, végezd el! Az olvasmányok általában magasabb szintű, nehezebb anyagok, de lehet, hogy csak kiegészítő érdekességek. A tanárodtól függ, hogy kötelezőek-e! Az emlékeztetők a tanult dolgokat foglalják össze.

Egy-egy fejezet végén készíts összefoglalást, és írd le a füzetbe! Reméljük, hogy az ellenőrző kérdések (és a tanárod) segítenek ebben a munkában.

Most pedig néhány olyan jelölés, amellyel többször is találkozol majd a könyvben:

## Bekeretezett szöveg

Alapvetően fontos ismeret.

### **Vastag betű**

Fontos ismeret, avagy fontos fogalom.

### *Dőlt betű*

Emlékeztető, megjegyzés, érdekesség.

A könnyebb tájékozódás érdekében a következő rajzokat használjuk:



**Olvasmány, kiegészítő anyag.**



**Ellenőrző kérdés, könnyű feladat.**



**Számolni vagy programozni kell.**



**Egérrel dolgozz!**



**Billentyűzettel dolgozz!**



**Fontos példa.**



**Figyelem, figyelmeztetés.**



**Írni vagy rajzolni kell.**



**Veszélyes dolog.**



**Lemezt kell használni.**

## Az informatikaterem rendje

A következőkben néhány fontos dolgot mondunk el arról, hogy mit kell tudnod, ill. milyen szabályokat kell betartanod, ha az informatikateremben dolgozol.

- 1.) A terem fokozottan balesetveszélyes, ezért fegyelmezetten viselkedj!
- 2.) A gépeket csak a tanárod engedélyével kapcsold be, ill. ki!  
Ha bármilyen rendellenességet tapasztalsz, azonnal szólj a tanárodnak!
- 3.) Ne javíts meg semmit önállóan, mert az balesetveszélyes, inkább jelentsd a hibát a tanárnak!
- 4.) Ha a gép csatlakozózsínorja be van dugva a konnektorba (esetleg maga a gép is be van kapcsolva), akkor ne nyisd ki a tetejét, és ne nyúlj bele, mert ez veszélyes! Áramütés érhet, vagy pedig tönkremehet a gép valamelyik alkatrésze. Ha számítógéppel akarsz dolgozni, először minden részegységet csatlakoztatnod kell a dobozhoz, és csak utána szabad bekapcsolnod a gépet!
- 5.) Ügyelj arra, hogy a programokban és az operációs rendszerben ne okozz kárt, pl. ne törölj ki fontos állományokat!
6. A teremben ne étkezz!
7. Munka után az eszközöket és az asztalt hagyd rendben!
8. *Figyelj ne beszélgess!*

## Emlékeztető

Az informatikaórákon gyakran használjuk majd a számítógépet, ezért összefoglaljuk a legegyszerűbb, de fontos elméleti és géphasználati ismereteket.

A sokféle számítógép közül mi a személyi számítógéppel (az angol **personal computer** kifejezés rövidítéseként PC-vel) dolgozunk. A számítógép kézzelfogható részeit **hardvernek** nevezzük. A hardver életre keltéséhez egyrészt elektromos energiára, másrészt a gépet működtető programrendszerre van szükség.

A számítógépen programokkal és adatokkal dolgozunk. A **program** a gép számára „érthető”, végrehajtható utasítások sorozata. A programok, adatok összefoglaló neve: **szoftver**.

Tudod-e, hogy mi mindenre alkalmas a számítógép? Sokféle feladatra! Általános, „univerzális” szellemi eszköz, amely egyre több szakmában nélkülözhetetlen. Sorold fel, milyen fő részekből áll a Te iskolai vagy otthoni géped!

A PC központi egysége a lemezegységekkel együtt egy dobozban van. Ehhez csatlakozik a monitor, a billentyűzet és az egér. Nagy mennyiségű adat hosszú távú tárolását szolgálják a lemezek. **A hajlékonylemezes meghajtók jele**



**A: és B:**, a merevlemezes meghajtóé (winchester), **C:**. A merevlemez sokkal több adatot tud tárolni, mint a hajlékonylemez (floppy).

A **billentyűzet** fontos adatbeviteli eszköz, ezen gépeled be a parancsaidat, szövegeidet, adataidat. A begépelte számok, betűk, jelek, összefoglaló néven **karakterek**, a monitoron egy kis villogó jelnél, a **kurzornál** jelennek meg.

Néhány fontos billentyű:

<b>Enter</b>	Ezzel zárod a begépelte parancsot.
<b>Esc</b>	Hatástalaníthatod az utoljára kiadott utasítást, ha lehet.
<b>Backspace</b>	A kurzortól balra lévő jelet törölheted vele (az Enter fölött található).
<b>Delete</b>	A kurzortól jobbra lévő karaktert törli.
<b>Space</b>	Szóköz.
<b>Shift</b>	Ha nyomva tartod, és leütsz egy betűt, nagybetűt kapsz.

Érdemes megtanulnod az **egér** kezelését is, sok munkától kímélhet meg. Ahogy mozog a kezdeddel együtt az egér az asztalon, úgy mozog az egérkurzor a monitoron. Miután rámutattál valamire az egérkurzossal a képernyőn, egy alkalmas programban a következő fontosabb műveleteket végezheted el:

- a bal oldali gombbal egyet kattintva, pl. kijelölhetsz valamit;
- a bal oldali gombbal duplán kattintva, pl. elindíthatsz egy programot;
- a gombot nyomva az egér mozgásával valamit mozgathatsz.

Ennyi bevezető után akár be is kapcsolhatod a számítógépet. A géppel való kommunikációt az operációs rendszer teszi lehetővé. Az operációs rendszer feladatai közé tartozik a számítógép vezérlése is.

Ilyen operációs rendszer például a DOS és a Windows '95. A kettő közti leglátványosabb különbség az, hogy a DOS-nak szöveges képernyője van, és a parancsokat általában be kell gépelni, míg a Windows '95 „grafikus képernyőjén” az egér használatával lehet a megfelelő parancsot kiválasztani.

Nos, nézzük, hogyan indíthatod el kedvenc programodat!

- DOS esetében gépeld be a program nevét, vagy – ha segédprogramot is használsz –, jelöld ki a nevét egy listából, és nyomd le az **Enter**-t!
- Windows '95 esetében kattints rá (duplán) programod indító ikonjára!

Ha a DOS-t használod, a következőket kell tudnod: A programokat és az adatokat **állományokban**, fájlokban (file) tároljuk a lemezeken. Az állomány neve legfeljebb 8 karakter, ezután egy pont és maximum 3 karakter, ún. **kiterjesztés** következik. Az **exe**, **com** és **bat** kiterjesztésű állományok programok, ezek indíthatók, futtathatók.

Ha kíváncsi vagy a lemezed tartalmára, a **dir** paranccsal kilistázhatod a képernyőre. Az állományok nem ömlesztve, hanem meghatározott rendben – könyvtárrendszerben – helyezkednek el a lemezen. Ez névvel ellátott „dobozokból”, azaz alkönyvtárakból áll. Minden lemeznek van egy főkönyvtára, ebben lehetnek al-

könyvtárak és állományok is. Az alkönyvtárak további alkönyvtárakból és állományokból állhatnak. A **tree** parancs kirajzolja a könyvtárszerkezetet a képernyőre.

Ha keresel egy programot, esetleg be kell lépned egy alkönyvtárba. Ezt a **cd** paranccsal teheted meg. Például, ha a főkönyvtárban van egy TANULO alkönyvtár, akkor a **cd TANULO** paranccsal léphetünk bele. A **cd..** paranccsal egy szinttel visszaléphetünk.

Előfordulhat, hogy valamelyik programod rosszul működik, és a gép nem reagál, **lefagy**. Vajon, mit kell ilyenkor tenned? Nem szabad gyorsan ki-be kapcsolnod a gépet, inkább nyomd le egyszerre a **Ctrl + Alt + Del** billentyűket, és ilyenkor a gép újraindul! Ha ez sem segít, akkor nyomd be a dobozon a Reset gombot!



### Ellenőrző kérdések

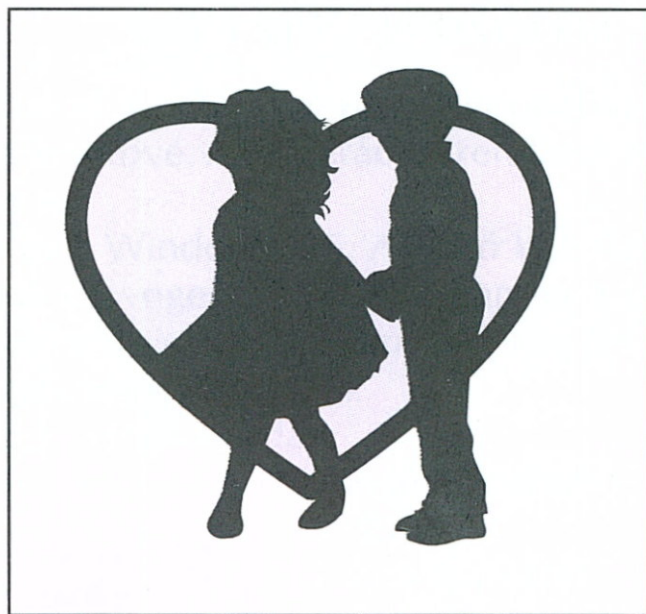
1. Mit jelentenek a következő fogalmak: hardver, szoftver, karakter, kurzor?
2. Mi a lemezek funkciója, mire szolgálnak?
3. Hogyan jelöljük a lemez meghajtókat?
4. Milyen egérműveleteket ismersz?
5. Mik az operációs rendszer feladatai?
6. Hogyan indíthatsz el egy programot?
7. Mit teszel, ha lefagy a gép?

## MI AZ INFORMÁCIÓ?

Figyeld meg a következő közléseket! Melyik jelent számodra információt?

Ma vasárnap van, holnap hétfő lesz.  
Holnap felelsz fizikából.  
Egyik iskolatársad szerelmes beléd.

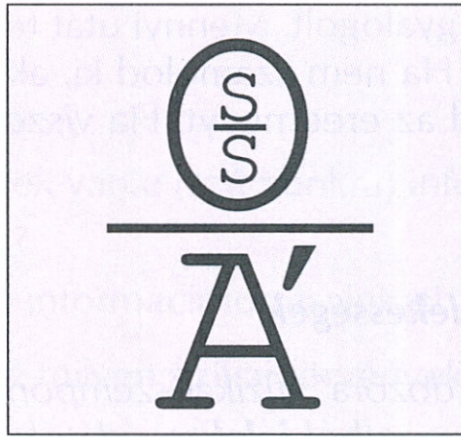
Lehetséges, hogy valamelyik közlés nem jelent számodra információt? Melyik közlés jelent számodra új ismeretet?



**Az információ új ismeretet jelent.**

Az információ tehát valaminek az észlelését, megtudását jelenti, olyan valamit, amit eddig nem észleltünk, vagy nem tudtunk. **Az információ tehát viszonylagos (relatív), függ előzetes ismereteinktől is.** Az mindegy, hogy milyen jelek hordozzák az információt, ha számodra érthetőek.

Mit mondanak számodra a következő jelek?



Fontos, hogy a jeleket megértsd, ismerd a jelentésüket, különben nem kapsz információt.

**Az információt mindig jelek (jelsorozatok, jelhalmazok) hordozzák.** Nem minden jelnek van számunkra információja; vagy azért, mert nem értjük, vagy azért, mert már ismertük a tartalmát.

Az információ lehet fontos, értékes, hasznos, de lehet nem fontos, értéktelen, haszontalan is. Lehet, hogy megmozgatja a gondolatainkat, érzéseinket, de lehet, hogy közömbösen fogadjuk. *(A köznapi életben gyakran akkor is információt mondunk, ha valójában csak ismeretekre gondolunk.)*

A tudatos, szándékos információcserét **kommunikációnak** is nevezik. Az emberek jellegzetes testtartással, távolságtartással, mozdulatokkal, arcmimikával, szemjátékkal, tagolatlan hanggal is cserélnek információt. Ezt az ősi, általában nem tudatos kommunikációs formát **metakommunikációnak** nevezzük.

A **nyelv** a legfontosabb információközlő eszközünk. Az emberiség népei igen sok nyelvet alkottak, formáltak az idők folyamán. A magyar nyelv a finnugor nyelvcsalád része, amely az uráli alapnyelvből származik.

A siketek **jelbeszéddel** kommunikálnak. Ez a „mutogatás”, a **jelelés** éppen olyan információdús lehet, mint a hangos beszéd. A vakok elsősorban hallás és tapintás útján szerzik ismereteiket.

Az információt hordozó jeleket nem öletszerűen, nem egymástól függetlenül szoktuk közölni (kimondani, mutogatni, leírni, rajzolni), hanem bizonyos szabályok szerint használjuk.

Legfontosabb **információforrásaink:**

- természetes és technikai környezetünk,
- embertársaink, mégpedig közvetlenül, telefonon, számítógép-hálózaton vagy tömegkommunikációs eszközökön keresztül,
- az írott és a nyomtatott szövegek (könyv, újság, jegyzet),
- a rögzített hangos beszéd és a zene (magnó, CD),
- a rajzolt, festett és nyomtatott képek, fényképek,
- a hangos mozgókép (film, videó, CD),
- a multimédia-számítógép és a virtuális valóság.

Információhoz juthatunk logikus **gondolkodással** is. Nézzünk egy egyszerű példát!

Julcsi 1,5 óra alatt 6 km-t gyalogolt. Mennyi utat tett meg 5 óra alatt? A válasz el van rejtve az adatokban. Ha nem számolod ki, akkor nem tudsz válaszolni a kérdésre, mert nem ismered az eredményt. Ha viszont kiszámolod, információhoz jutsz.



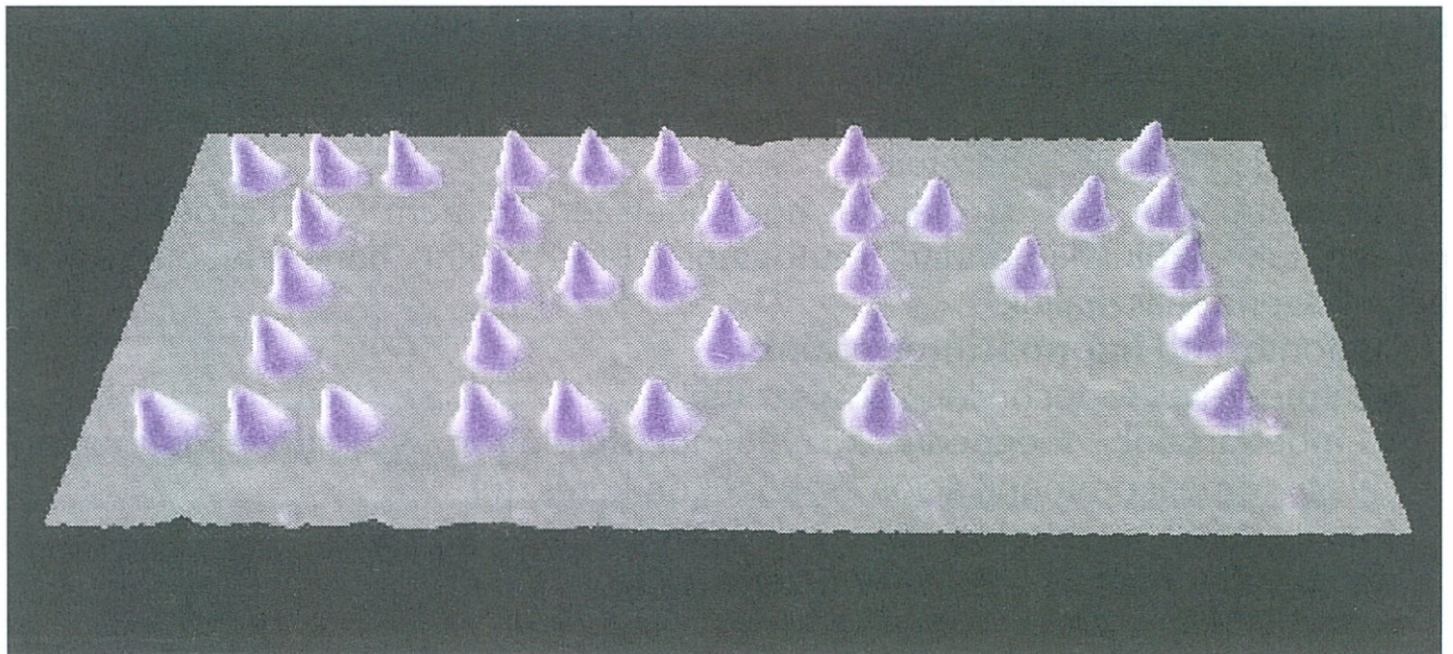
### Információfizikai érdekességek

Információ létezéséhez "hordozóra", fizikai szempontból valamilyen **anyagra és energiára**, informatikai szempontból **jelekre**, adatokra van szükség. Bár a jel (információ) nagyon gyorsan továbbítható, és kevés anyagban is sok tárolható, mégis korlátot jelentenek a következő törvények:

**A jel legfeljebb fénysebességgel (300 000 km/s) továbbítható.** Fénysebességgel közvetítjük a jeleket pl. a rádióhullámokkal és a lézerrel (ha légüres térben haladnak).

**Minden jelnek van anyaga (anyagi hordozója) és energiája. Vannak jelek, amelyeknek nagyon kicsi az energiája, de az energia nem lehet akármilyen kicsiny.** Nem lehet kisebb, mint a részecskék mozgási energiája (különben a véletlenszerűen mozgó részecskék azonnal "szétbombázzák", megsemmisítik a jelet). Persze a részecskék energiája nagyon-nagyon kicsiny (kb.  $10^{-21}$  J =  $1/10^{21}$  J szobahőmérsékleten). A gyakorlatban használt jelek általában sokkal nagyobb energiájúak.

A jelek sűrítésének is határai vannak, **semmilyen jel sem lehet tetszőlegesen kis méretű.** (Ehhez, a fizika törvényei szerint végtelen nagy energiára lenne szükség.) A jelek összesűrítése terén egyébként óriási technikai lehetőségek előtt állunk.



Egy nikkelkristály felszínén xenonatomokból rakták ki ezt a három betűt pásztázó elektronmikroszkóp segítségével



## Ellenőrző kérdések

1. Mi az információ?
2. Mi hordozza az információt?
3. Mitől függ, hogy egy jelnek van-e (számunkra) információja?
4. Mi a metakommunikáció?
5. Sorold fel a legfontosabb információforrásainkat!



A jelekre vonatkozóan milyen fizikai törvényeket ismersz?



## Feladatok

1. Közöld a padtársaddal rajzban, hogy holnapra uszodába vagy focizni hívod!
2. Rajzold le néhány vonallal, hogy milyen a tanárod arca, ha örül a feleletednek!
3. Fejezd ki jelekkel, hogy győztetek!
4. Hogyan kell érdeklődő arccal figyelni társaságban vagy az órán? Mondd el!
5. Olvasd el figyelmesen a következő történetet, és válaszolj!

Egy erdőben él Alice az oroszlánnal és az egyszarvúval. Az oroszlán minden hétfőn, kedden és szerdán hazudik, a hét többi napján igazat mond. Az egyszarvú csütörtökön, pénteken és szombaton hazudik, a hét többi napján igazat mond. Egy nap a következőt állították:

Oroszlán: Tegnap hazudós napom volt!

Egyszarvú: Tegnap nekem is hazudós napom volt!

Vajon milyen nap volt?

## AZ INFORMÁCIÓ MENNYISÉGE

Láttuk, hogy a jelek, közlések néha hordoznak információt, néha viszont nem. Lehet-e mérni az információt? Mi a véleményed, melyik mondatban van több információ?

- Péternek március 22-én van a születésnapja.
- Péternek márciusban van a születésnapja.

(Feltesszük, hogy egyiket sem tudtad.) Az első mondat 365 nap közül egyet jelöl meg. A második csak 12 hónap közül nevez meg egyet. Ezért úgy érezzük, hogy az első mondatban több az információ.

Ha egy eseményről vagy tényről értesülünk, a közleményre (jelekre, adatokra) igaz a következő két állítás:

**Annál nagyobb az információ mennyisége, minél nagyobb az egyformán lehetséges (azonos valószínűségű) esetek száma, mielőtt az információt megkaptuk.**

**Annak a közlésnek, amely nem ad (újabb) információt, 0 az információ-mennyisége.**

Visszatérve Péter születésnapjához, a példa szerinti 365 lehetőség (a lehetséges születésnapok száma) több, mint 12 (a lehetséges hónapok száma). Az első mondatban ezért több az információ.

Mit gondolsz, melyiket nehezebb véletlenül eltalálni: a hónapot vagy a napot? Jól sejtetted, a napot nehezebb! Ezt úgy mondjuk, hogy a nap eltalálásának kisebb a valószínűsége, mint a hónap eltalálásának. Összefoglalva:

**Minél kisebb egy esemény bekövetkezésének a valószínűsége, annál több információt jelent, ha megtudjuk.**

## AZ INFORMÁCIÓ EGYSÉGE



**Kísérletezz! Dobj fel egymás után sokszor egy pénzérmét!**

Számold össze az írások és a fejek számát! Számold ki, hogy ez hány %-a az összes kísérletnek!

Valószínűleg 50% körüli értéket kaptál, mégpedig azért, mert pontosan 50% az esélye a fejnek és 50% az írásnak. A pénzfeldobásnak kétféle, azonos valószínűségű kimenetele lehet. Az eredményt két jellel tudjuk leírni, a kettőt együtt **bináris jelnek** nevezzük. Például: 0 és 1, ↑ és ↓, O és + ...

Ebben az esetben azt mondjuk, hogy minden pénzfeldobás után egységnyi információt kaptunk.

Az információmennyiség mértékegysége a **bit**, az angol binary unit (kettes egység) alapján.

**Ha megtudjuk, hogy két azonos valószínűségű lehetőség (esemény) közül melyik következett be, akkor 1 bit információmennyiséghez jutunk.**

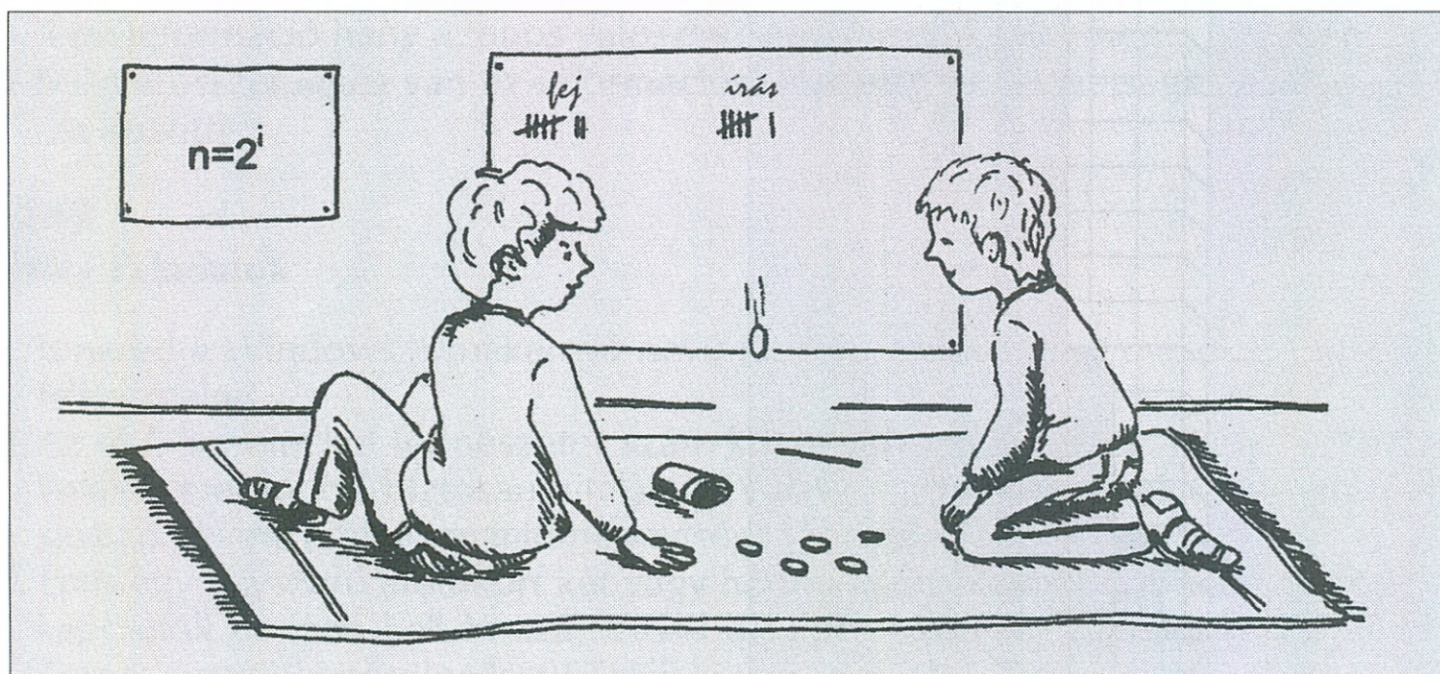


Tegyük fel, hogy egy áramkörben egy kapcsoló általában ugyanannyi ideig van bekapcsolva, mint kikapcsolva. Másként fogalmazva: azonos valószínűséggel van bekapcsolva, mint kikapcsolva. Ha megtudod, hogy éppen melyik állapotban van a kapcsoló, akkor 1 bit információt kapsz.

## ÖSSZEFÜGGÉS AZ INFORMÁCIÓMENNYISÉG ÉS A LEHETSÉGES ESETEK SZÁMA KÖZÖTT



Dobj fel kétszer egy pénzérmét, és írd le jelekkel az eredményt! Legyen a fej jele  $\uparrow$ , az írásé  $\downarrow$ . (A  $\uparrow\uparrow$  és a  $\downarrow\downarrow$  együtt bináris jel.) Egy kísérlet eredménye pl.:  $\downarrow\downarrow$  (kétszer írás).



Mennyi információmenységhez jutottál?  
Mennyi az összes lehetséges esetek száma?

Az információmenység

$2 \times 1 \text{ bit} = 2 \text{ bit}$ ,  
mert kétszer  
végeztük el a  
kísérletet

A lehetőségek száma

$(\downarrow, \downarrow)$   
 $(\downarrow, \uparrow)$  összesen  
 $(\uparrow, \downarrow)$  4 lehetőség  
 $(\uparrow, \uparrow)$





## Feladatok

1. Dobj fel háromszor egy pénzt, és írd le jelekkel az eredményt! Mennyi információmennyiséghez jutottál? Mennyi az összes lehetőség száma?
2. Négy pénzfeldobás esetén hány bithez jutottál? Mennyi az összes lehetőség száma?

Találtál-e szabályt?

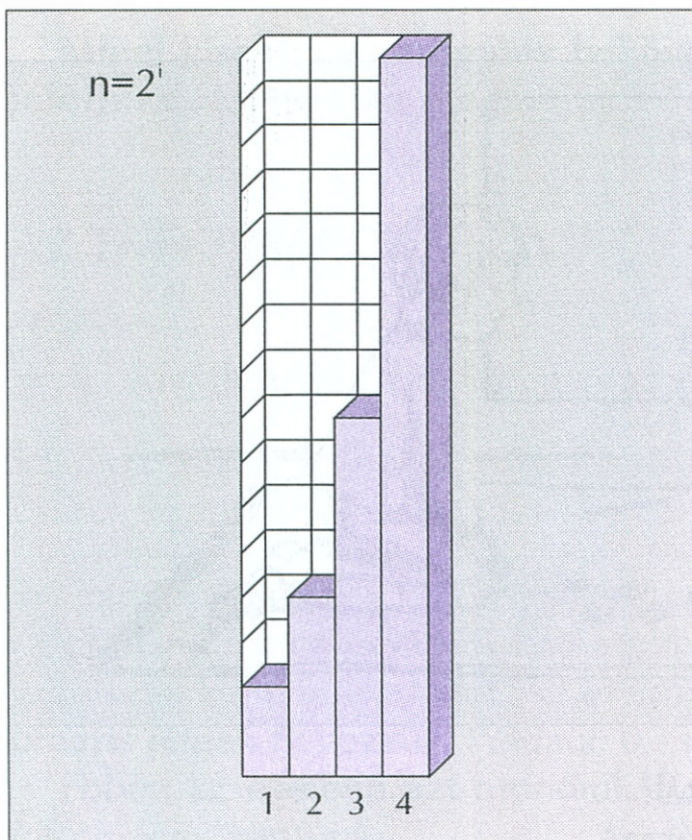
**Az információmennyiség annyi bit, ahányszor elvégeztük a kísérletet. Ezt jelöljük  $i$ -vel.**

**Az összes lehetőség száma:  $2^i$  (pl. 4 dobás esetén  $2^4$ ).**



## Érdekes

*Érdekes, hogy az összes lehetőség száma,  $2^i$  nem egyenesen arányos az információmennyiséggel, azaz a bitek számával. Annál sokkal gyorsabban nő.*



*Hány bit információmennyiséget hordozhat egy bináris jel?*

*Azt gondolhatnád, hogy éppen egyet. De ez nem mindig igaz! Képzeld el a következő játékot! Osztyártársad a sakktábla egyik mezőjére helyezte a vezért. Te nem láthatod! Hány bit információmennyiséget jelent, ha megtudod, hogy hol van? Találjuk ki barkochbakérdésekkel! Tudjuk, hogy **ha** olyan barkochbakérdést teszel fel, amelyre **az igen válasz éppen olyan valószínű, mint a nem, akkor***

**minden válasszal 1 bit információhoz jutsz.** (Ez biztos, ez az információmennyiség egységének definíciója.) Hogyan kell feltenni a kérdést, hogy az igen válasz éppen olyan valószínű legyen, mint a nem? Mindig felezni kell a lehetséges mezők számát. (Például az első kérdés: Az A vagy a B vagy a C vagy a D oszlopban van a vezér?) Így 6 kérdéssel biztosan kitalálhatod, hogy hol van a vezér. **Hat bináris jel (igen/nem) volt a válasz, és Te pontosan 6 bit információmennyiséghez jutottál.** Most figyelj! Ha rögtön az elsőre rákérdezel, és olyan szerencséd van, hogy eltalálsz, akkor **egyetlen igen válasszal jutsz** ugyanahhoz az információmennyiséghez, vagyis **6 bithez**. Ebben az esetben **egy bináris jel 6 bit információt hordoz.**

(Igaz az is, hogy egy bináris jel hordozhat 1 bitnél kevesebb információmennyiséget is. Van 1 bitnél kevesebb információ is.)



### Ellenőrző kérdések

1. Mikor nulla az információmennyiség?
2. Mi az információmennyiség egysége?
3. Mikor 1 bit az információ(mennyiség)?
4. 4 bit információ hány azonos valószínűségű esetnek felel meg?
5. Milyen összefüggés van az információmennyiség és a lehetséges esetek száma között?



### Feladatok

1. Ismered a Windows Aknakereső nevű játékát? Milyen információkat tudsz itt felhasználni?
2. Az első harminckét természetes szám közül gondoltam egyre. Hány barkochbakérdéssel tudod biztosan kitalálni? (Válaszolni csak **igennel** vagy **nemmel** szabad.) Hány bit információmennyiséget kapsz?
3. Építs egy egyszerű áramkört két vagy három kapcsolóval (zsebtelep, vezeték, kapcsolók és izzó kell hozzá)! Kösd a kapcsolókat sorba, és próbáld ki az összes esetet (kapcsolóállást)! Találj ki egy jelölést, és írd le az eseteket!

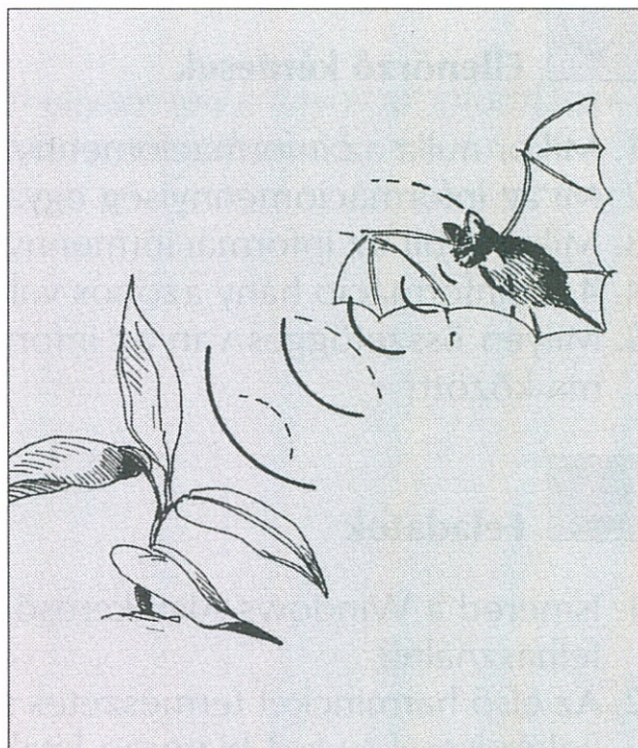
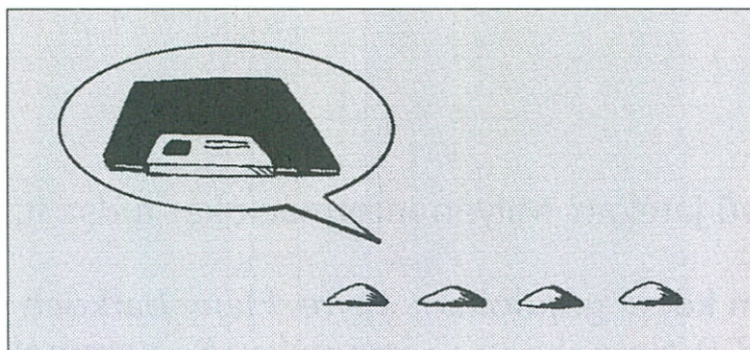
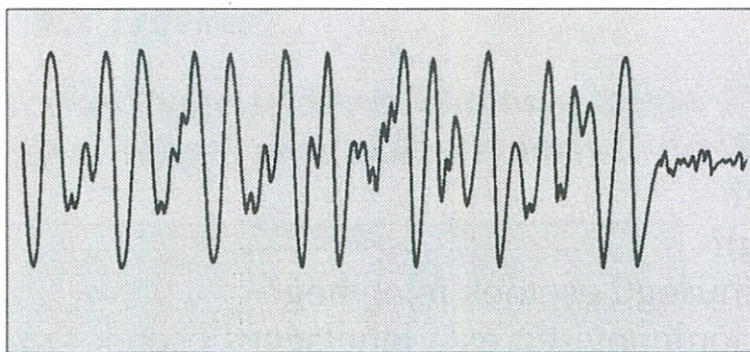
## A JELEK VILÁGA

Természeti és technikai környezetünkről érzékszerveink révén jutunk közvetlen információhoz (látjuk, halljuk, tapintjuk a tárgyakat, anyagokat). Az érzékelés szempontjából annyi jelfajtát különböztethetünk meg, ahány érzékszervünk van; így vannak **látható, hallható, tapintható, szagolható, ízlelhető jelek.**

A közösségekben élő élőlények (állatok, rovartársadalmak, sejthalmazok, növények), az emberi társadalmak és az ember által készített informatikai gépek is különféle jelek, jelrendszerek segítségével közölnek információt egymással.

Civilizációnkban leggyakrabban **kép, szó** és **írás** közvetítésével kommunikálunk.

Rengeteg, közvetlenül nem látható, nem érzékelhető technikai jel is létezik, amelyeket az ember nap mint nap használ, pl. ilyen a rádiójel, a radarjel, a mikrofonjel, a jel a CD lemezen, a mágneslemezen, a csipben. Az élőlények által használt jelek egy része sem érzékelhető az ember számára, pl. gyenge szag, illat, amely egy adott rovarfajra jellemző, a denevérek által használt ultrahang, az infravörös fény (van olyan kígyó, amelyik lát ebben a tartományban) vagy az ibolyántúli fényjel, mágneses jel (amely néhány állat tájékozódását segíti).



**A jel érzékszerveinkkel vagy (legalább) műszereinkkel felfogható, mérhető jelenség, amely egy másik dologra, jelenségre utal.**

Vizsgáljuk meg a könyv példáján keresztül, mi is a jel!

A könyv szó egyrészt öt betű egymás után: k, ö, n, y, v, ez a **jelölő**, ez jelöli a könyvet, másrészt jelent egy ismert tárgyat:



A jelölő mindig valamilyen anyagi dolog, jelenség; a jelölt pedig egy csoport, ill. közösség által elfogadott jelentés. Ez igaz mindenféle jelre.

**jel = jelölő + jelentés**

A jelek egy halmaza a használati szabályaikkal együtt **jelrendszer** alkot. A jelekkel részletesebben a jelentéstan és a nyelvten foglalkozik.

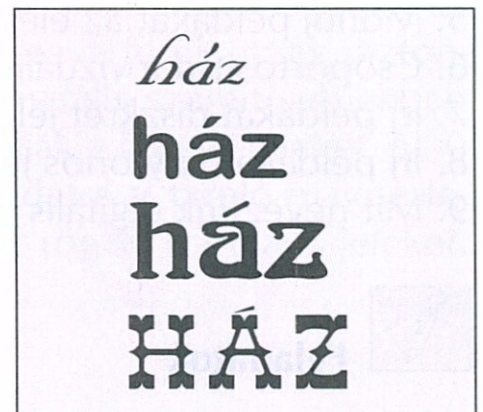
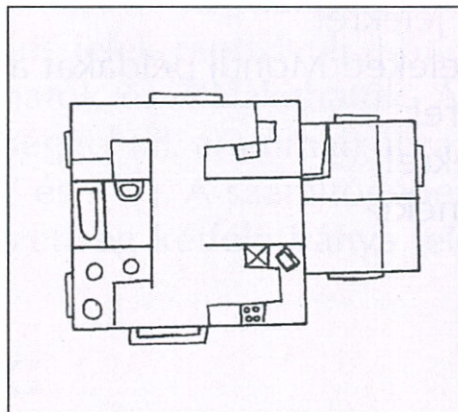
A kiejtett szó hangokból, a leírt szó betűkből áll. A kép is felbontható „pontokra”, képelemekre. Ezeket **elemi jeleknek** nevezzük. Az elemi jeleknek nincs feltétlenül önálló jelentésük, más elemi jelekkel együtt alkotnak értelmes egységet.

**A jelek elemi jelekre bonthatók.**

Milyen jeleket ismerünk?

A jelek egyik fontos halmazát a **látható**, idegen szóval **vizuális jelek** alkotják. Ezeket a jelölt és a jelölő közötti fizikai hasonlóság alapján könnyű csoportosítani; a jel lehet:

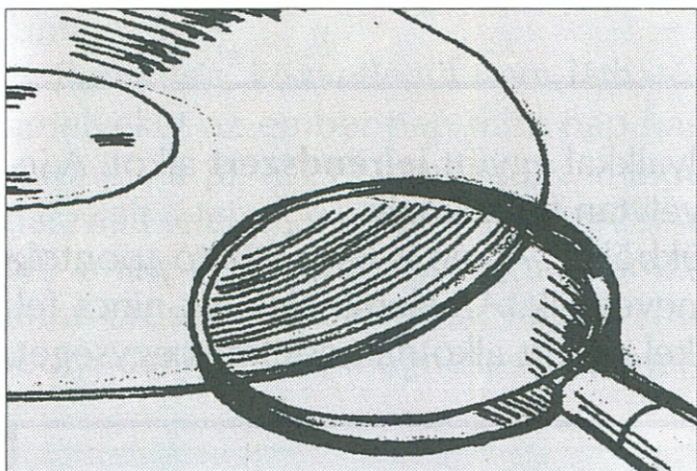
- **képszerű** (rajz, fénykép, film, hologram, képregény);
- **stilizált** (térkép, műszaki rajz, ikon, piktogram, képírás);
- **szimbolikus** a jelölő és a jelölt között nincs hasonlóság, a megfeleltetés önkényes (betűk, számok, írásjelek, morzejelek).



Más szempontból a jeleket két csoportra oszthatjuk:

- A jel lehet **diszkrét**, azaz különálló. A diszkrét jel véges sokféle lehet, véges sokféle értéket vehet fel. Diszkrét jelek a logikai jelek is, amelyek két értéket vehetnek fel: hamis (0) vagy igaz (1) értéket. Diszkrét jelek a betűk is. A diszkrét jelekből hosszabb sorozatokat, halmazokat alkothatsz, pl. ilyen ez a szöveg, amit most olvasol. Diszkrét jelsorozatok: a számjegyek (142873), morzejelek (... - - - . . .), a ki-be kapcsolható fényjelek (világos/sötét).

- A jel lehet **folytonos** is, azaz bármilyen értéket felvehet két határérték között, pl. az óramutató állása, kürtjel, hangbarázda a régi hanglemezeken.



A diszkrét jeleket számokkal, számjegyekkel is leírhatjuk, ezért **digitális jeleknek** is nevezzük őket (digit = számjegy). Látni fogjuk, hogy a digitális jelek nem minden esetben számok. Nemcsak mennyiségek leírására használhatók.



### Ellenőrző kérdések

1. Mondj példákat a kedvező és a kedvezőtlen természeti jelekre!
2. Leggyakrabban milyen jelekkel kommunikálsz?
3. Mi a jel?
4. Mi a jelrendszer?
5. Mondj példákat az elemi jelekre!
6. Csoportosítsd a vizuális jeleket! Mondj példákat a csoportok elemeire!
7. Írj példákat diszkrét jelekre!
8. Írj példákat folytonos jelekre!
9. Mit nevezünk digitális jelnek?



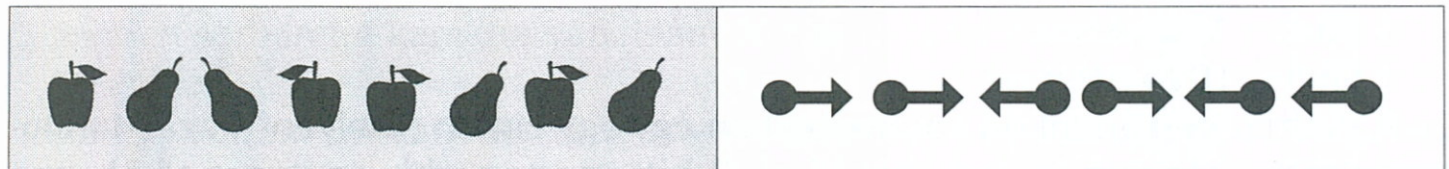
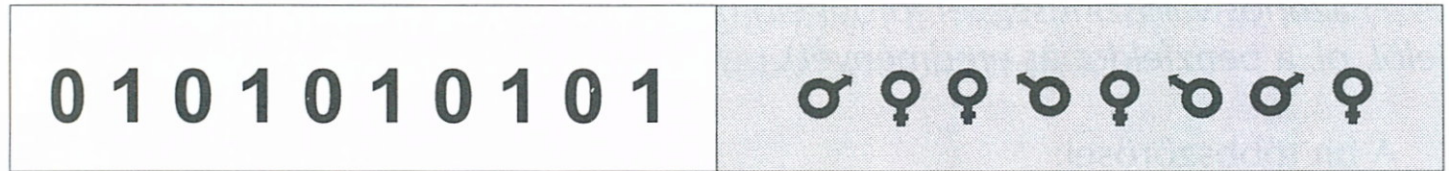
### Feladatok

Gyűjts különféle jeleket, írd, rajzold vagy másold le! Csoportosítsd őket! (Használhatsz fényképezőgépet, magnót, fénymásolót vagy számítógépet is. Ez hosszabb ideig tartó munka.)

1. Gyűjts példákat a növény- és állatvilágból, és írd le, hogy milyen jelekkel üzennek egymásnak? (Néhány ötlet: illatok, szagok, feromonok, hangok, fény, ének, fütty, elektromos jel, testmozgás, testtartás.)
2. Gyűjts jeleket az írás és nyomtatás történetéből!
3. Gyűjts technikai jeleket a hírközlés történetéből!

## AZ ADAT

Az adat olyan jelhalmaz, amit valakinek vagy valaminek a jellemzésére használunk. Az adatnak megfelelő jelhalmazt mindig valamilyen jelrendszerben ábrázoljuk, ezek közül a legegyszerűbbek azok a jelrendszerek, ahol csak két elemi jel van.



**Ha egy jelrendszerben mindössze két elemi jelet használunk (pl.: 0 és 1, vagy + és -, vagy ● és ○, ↑ és ↓), akkor bináris jelrendszerről beszélünk.**

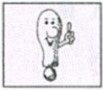
Ilyen bináris jelrendszer pl. a kettes számrendszer. Ha egy számot tízes és kettes számrendszerben is leírunk, pl.  $21_{10}$  és  $10101_2$ , akkor rögtön észrevesszük, hogy az utóbbi ábrázolás sokkal hosszabb. Nincs ez másként akkor sem, ha nem számokról van szó.

Ha ilyen hosszadalmas bináris jelrendszerben leírni az információt, akkor vajon miért használjuk? Azért, mert a mai számítógépek túlnyomó többsége bináris jelekkel dolgozik. A bináris jelek rendkívüli módon zavartűrők, nagy biztonsággal tárolhatók, továbbíthatók és átalakíthatók. A digitális számítógépekben az áramkörök eltérő feszültséggel (ill. árammal) ábrázolják a bináris jeleket, pl. 0 V és nem 0 V (általában 3 V és 5 V). A számítógépes adatokat tároló mágneslemezekeken pedig a mágnesezettség kétféle iránya jelöli, rögzíti a bináris jeleket.

## AZ ADATMENNYISÉG

Általában tudni kell, hogy valamely adat leírásához, tárolásához mennyi „helyre” van szükség, vagyis hány elemi jelből áll az adatunk. Ez a mérőszám a jelek számát, az **adathalmaz méretét** jelzi, ezért **adatmennyiségnek** nevezzük.

**Az adatmennyiség mértékegysége a bit (binary digit), amely egy bináris jel adatmennyiségét jelenti.**



Az adatmennyiség más, mint az információmennyiség, bár a mértékegységek neve ugyanaz. Egy bináris jelnek mindig 1 bit az adatmennyisége. Az információmennyiségről ez nem mondható el. Az információ csak akkor 1 bit, ha a két jel (0 és 1) azonos valószínűséggel fordul elő (azaz egyformán lehetséges két eseményt jelöl, pl. a pénzfeldobás eredményét).

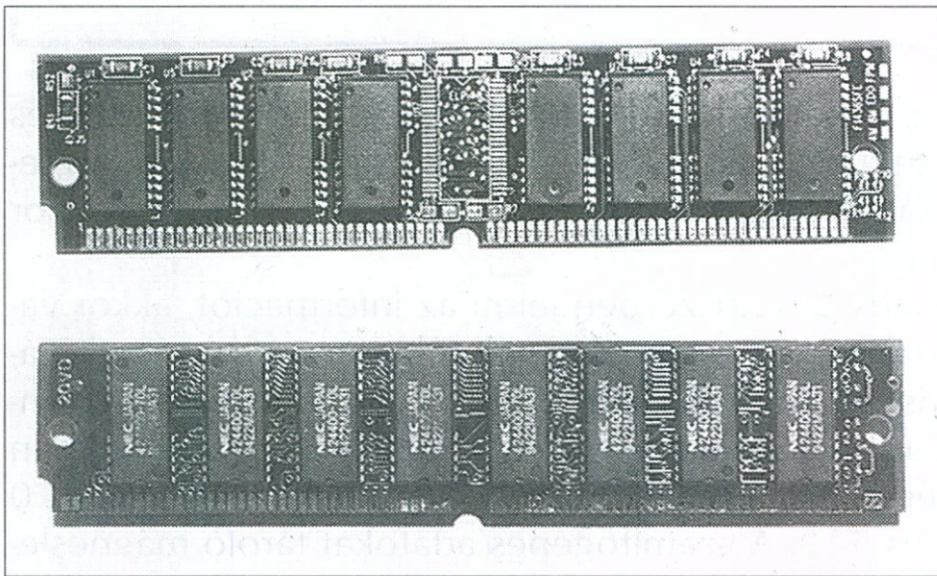
A bit többszörösei:

1 Kbit =  $2^{10}$  bit = 1024 bit;

1 Mbit = 1024 Kbit;

1 Gbit = 1024 Mbit.

A számítógép áramköri memóriájában egy-egy címen („helyen”), azaz **tárolóegységben** 8 bináris jel helyezhető el. **8 bit** egy nagyobb egységet alkot, amit **byte-nak** (bájt) nevezünk. Egy, számítógépben használt memóriaegységen nagyon sok byte-nak van hely, ilyen modulok láthatók a következő képen:



**1 byte = 8 bit**

A byte többszörösei:

1 Kbyte =  $2^{10}$  byte = 1024 byte;

1 Mbyte = 1024 Kbyte;

1 Gbyte = 1024 Mbyte.

Nyolc bináris jel felhasználásával 256 (azaz  $2^8$ ) db egymástól különböző jelet tudunk leírni, megkülönböztetni. Minden ilyen bitsornak, byte-nak – kölcsönösen és egyértelműen – megfeleltethetünk jeleket: betűket, számjegyeket, írásje-

leket, szimbólumokat, összefoglaló néven **karaktereket**. Ezt a lehetőséget alkalmazzuk a személyi számítógépekben, a különböző karaktereket különböző byte-okkal jelöljük. Pl. A  $\Leftrightarrow$  01000001, B  $\Leftrightarrow$  01000010, C  $\Leftrightarrow$  01000011. Ezért igaz a következő megállapítás:

### Egy karakter adatomennyisége 1 byte.

Tudni kell, hogy nem a karakter képét, formáját tárolják egy byte-on, hanem a karakter kódját. A karakter képét sokkal több byte-on tárolják.

Példa:

Péter levelet írt a barátjának. A levél 3 oldal és 48 sort tartalmaz oldalanként, és minden sorban 64 karakter van. Hány Kbyte az adatomennyisége?

Megoldás

Ha egy sorban 64 karakter van, akkor 48 sorban  $64 \cdot 48 = 3072$  karakter, három oldalon pedig  $3 \cdot 3072 = 9216$  karakter. Tudjuk, hogy 1 karakter adatomennyisége 1 byte, ezért 9216 karakter 9216 byte.  $9216 : 1024 = 9$  Kbyte.

*Figyeld meg!*

A **bit** jelenthet egy konkrét bináris jelet (pl. 0-t), és jelentheti az adatomennyiség egységét is. Ez általában nem okoz félreértést.



### Ellenőrző kérdések

1. Milyen bináris jeleket ismersz?
2. Mi az adat?
3. Egy byte hány bit?
4. Sorold fel az adatomennyiség egységeit és a többszöröseket!
5. Egy karakter hány byte-on tárolható?



### Feladatok

1. Gyakorold az átváltásokat!

5 byte = ..... bit	16 bit = ..... byte	9600 bit = ..... byte
50 byte = ..... bit	1024 bit = ..... byte	2400 bit = ..... byte
5 Kbyte = ..... bit	512 bit = ..... byte	14 400 bit = ..... byte
1,2 Mbyte = ..... byte	1,44 Mbyte = ..... byte	
1 Gbyte = ..... Mbyte	1 Gbyte = ..... Kbyte	
2. Egy titkos jelsorozat a következő jelekből áll: §@§§§@@§§§§@§@@. Hány bit adatomennyiséget tartalmaz?
3. Egy regény 250 oldal, egy-egy oldalon kb. 2000 karakter van. Mennyi karakter van a könyvben, és hány byte az adatomennyiség?



4. Ha egy merevlemez tárolókapacitása 1,08 Gbyte-os, hány karaktert tudnál rajta tárolni? Hány db könyvben férne el ennyi karakter, ha egy könyvben átlag 800 000 karakter van?
5. Hajlékony mágneslemezünk 1,2 Mbyte szöveges adatot tartalmaz. Hány gépelt oldalon fér el ennyi szöveg, ha egy oldalon 40 sor, és egy sorban 65 karakter van?

## ELVARÁZSOLT JELEK

### Emlékeztető

- Bizonyára olvastál már a titkosításokról. Az egyik titkosítás, a betűhelyettesítési módszer a következő: A titkosítandó szöveg minden betűjét helyettesítjük az ábécében mögötte álló harmadik betűvel. Az a helyett d-t, a b helyett e-t ... írunk. Az ábécé utolsó három betűje helyett pedig az első hármat írjuk. Megfejtéskor pedig fordítva: d helyett a-t, e helyett b-t ... olvasunk, ill. írunk. Az ilyen titkosítások lényege, hogy egy jelrendszerben (ábécében) minden elemi jelnek (betűnek) megfeleltetünk valamilyen más jelet, és ebben az új rendszerben írjuk le az információt.
- Valószínűleg tanultad matematikából, hogy a számok is átírhatók tízes számrendszerből más, pl. kettes számrendszerbe.

Egy kis segítség: kettes számrendszerben csak 0-t és 1-et használhatsz. Például a 13 így bontható fel:

1 db 8-as, 1 db 4-es, 0 db 2-es és 1 db 1-es. Ha ezeket összeadod, 13-at kapsz. Táblázatba rendezve:

8	4	2	1
1	1	0	1

Az ilyen és ezekhez hasonló megfeleltetéseket (titkosításokat) kódolásnak nevezzük.

**Amikor jeleket meghatározott szabályok szerint egy másik jelrendszerbeli jelekké alakítunk, kódolásról beszélünk. A visszaalakítást dekódolásnak nevezzük.**

A kettes számrendszer egyfajta bináris jelrendszer. Vajon mindenféle jel leírható, kódolható valamilyen bináris jelrendszerben? A számítógép karakterei 8 bináris jellel kódolhatók, egy karakter 1 byte-on. 8 bináris jel  $2^8 = 256$ -félekép-

pen rakható sorba, ezért 256-féle karaktert lehet velük kódolni. Kell-e 8 bit akkor, ha ennél kevesebb betűnk van?



### Példa

Legalább mennyi bitre van szükség az angol ábécé 26 betűjének titkosításához, ha azonos hosszúságú bitsorokat használunk?

Megoldás

4 bit nem elég, mert ennek csak 16 féle sorrendje van,  $2^4 = 16$ . 5 bit már megfelel, mert abból 32 féle sorrend állítható össze.

Készíthetünk egy konkrét megfeleltető táblázatot is:

A 00001

B 00010

C 00011

D

Folytasd!

Ebben a példában minden betűt (elemi jelet) 5 bittel írtunk le. Sok esetben hasonló a helyzet, csak az elemi jelek nem feltétlenül betűk, és lehet, hogy nem 5 bit kell a kódolásukhoz.



### Példa

Egy képet úgy képzelhetünk el, mintha apró képpontokból állna.

Ha egy képpont 16 féle színű lehet, akkor hogyan lehet kódolni egy képpontot? Milyen hosszú bitsorral lehet leírni (titkosítani) egy-egy színt? Készítsünk ilyen kódtáblázatot!

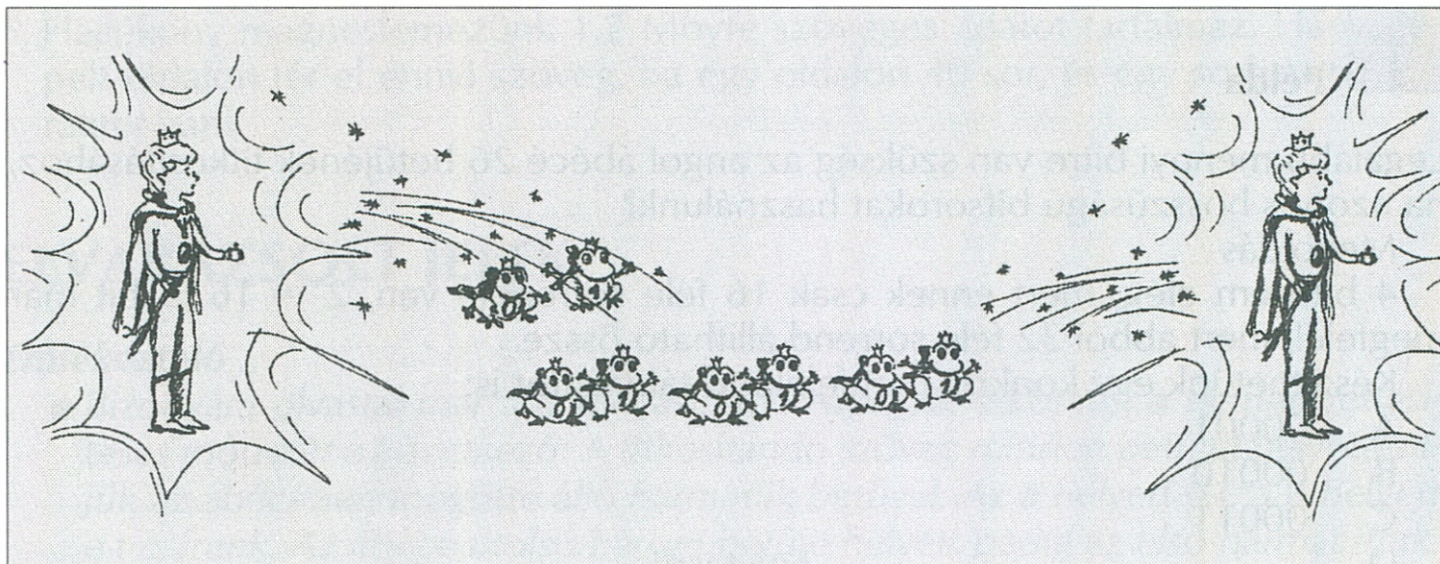
Megoldás:

Minden színhez tartozzon egy szám a következő táblázat szerint:

szín	színkód	bináris kód
fekete	0	0000
kék	1	0001
zöld	2	0010
piros	3	0011
.	.	.
.	.	.
.	.	.

Látható, hogy 4 bináris jelre van szükség, mert  $2^4 = 16$  féleképpen lehet variálni, sorba rakni a biteket.

Sokféle bináris kód létezik, akár következőt is mondhatjuk:  
„A **jel**királyfi elvarázsolható **bit**békák sorává, de vissza is varázsolható.”

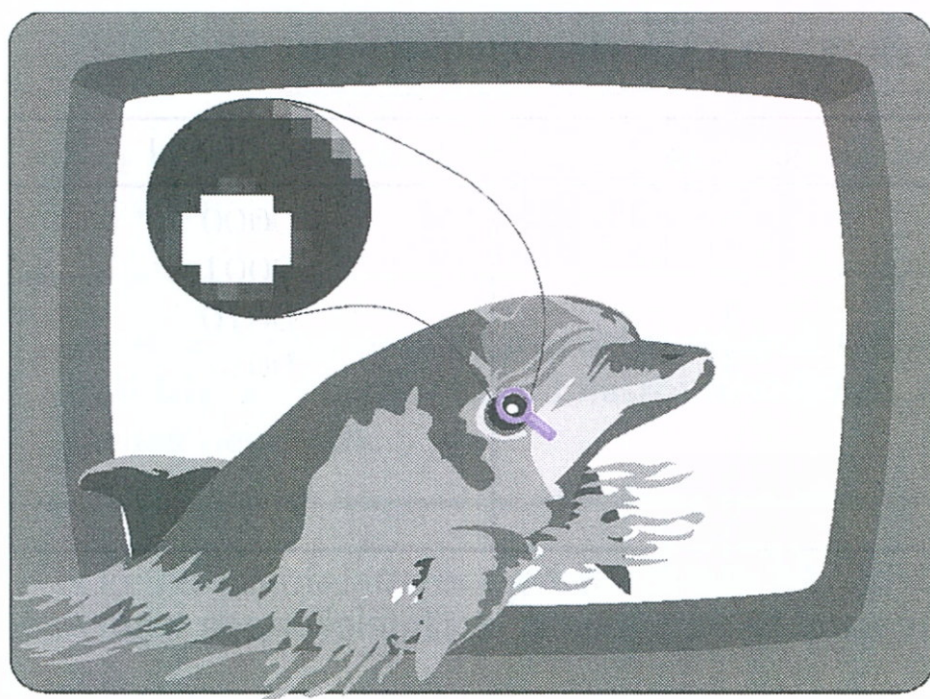


Bármilyen ismert jel, legyen az hang, zene, ének, beszéd, kép, grafika, fénykép, video-, hologram, szöveg vagy bármi más (pl. nem érzékelhető jel), kódolható bináris jelrendszerrel. Ez az alapja annak, hogy a bináris jelekkel, ill. az ezekből felépített jelekkel dolgozó számítógépek mindenféle emberi (érzékelhető) vagy gépi jelet képesek feldolgozni, tárolni, továbbítani.



### Példa

Egy színes kép a monitoron  $800 \times 600$  képpontból áll. Minden képpont 256 féle színű lehet. Hány Kbyte-on tárolható a kép, azaz hány Kbyte az adatmennyisége?



Megoldás:

A kép  $800 \cdot 600 = 480\,000$  képpontból áll. A példában 1 képpont adatmennyisége  $8 \text{ bit} = 1 \text{ byte}$ , mert 256 lehetőséget (színt) 8 biten lehet tárolni:  $256 = 2^8$ . Ezért  $480\,000$  képponté  $480\,000 \text{ byte}$ , ami  $468,75 \text{ Kbyte}$ .



### Ellenőrző kérdések

1. Mit nevezünk kódolásnak?
2. Milyen jelek kódolhatók bináris jelrendszerben?



### Feladatok

1. Egy fekete-fehér pontokból álló kép (grafika)  $1000 \times 1500$  képpontra bontható. (A képpontok bináris jelek, hiszen bármelyik pont kétféle lehet: fekete vagy fehér.) Hány bit a kép adatmennyisége?
2. Mennyi adatmennyiséget tartalmaz egy kép, ha  $1024 \times 768$  képpontból áll, és egy képpont 256 féle színű lehet, azaz 1 byte? Hányféle szín kódolható két byte-tal? (Nem 512!)
3. Egy színes monitor  $640 \times 480$  képpontból áll. Minden képpont színe 2 byte-tal van kódolva. Legalább hány Kbyte memória kell egy ilyen kép tárolásához? Hány színű a monitorkép?
4. Legfeljebb mennyi adatmennyiséget tartalmaz egy 200 oldalas könyv, ha 55 féle jelet használ, egy oldalon átlagosan 52 sor van, és egy sorban átlagosan 68 jel?
5. Mennyi adatmennyiséget tartalmaznak a muslica DNS-molekulái, ha együttes hosszuk  $1,2 \text{ cm}$ , és egy nukleinsav „hossza”  $3,4 \text{ nm} = 3,4 \cdot 10^{-7} \text{ cm}$ ? A DNS 4 féle nukleinsav jelsorozatának tekinthető. Ezeket A, T, G, C betűkkel jelölik.
6. Egy régi civilizáció 1024 féle jelet, ikont használt. Egy táblán 11 ilyen jelet számoltunk össze. Legfeljebb hány bittel lehetne kódolni azt, hogy milyen és mennyi ikont tartalmaz a tábla?

## AZ ASCII-KÓD

Ha a jeleket tárolni vagy továbbítani akarjuk, általában kódolni kell őket. A kódolást, amely mindig együtt jár kisebb-nagyobb jelátalakítással is, a feladat követelményeitől függően általában automata vagy program végzi.

A számítógépek és az adatátviteli rendszerek karakterkészletének kódolására szolgáló táblázat az **ASCII-kódrendszer** (American National Standard Code for

Information Interchange, amelynek jelentése: amerikai nemzeti szabványos kód információátvitelre). Minden karakternek megfelel egy 8 bites kód. Ez a bitsor a kettes számrendszerben egy szám. Ezt a számot írjuk le (tízes számrendszerben) és melléje a karakter jelét, így kapjuk a kódtáblázatot. Az ASCII-karakterkészlet betűkön, számjegyeken és írásjeleken kívül sokféle grafikai jelet, ill. vezérlőjelet is tartalmaz. A szabványos ASCII-készlet első 128 karakterét megtartva, de a többi közül néhányat módosítva, többféle kódtáblázat is született. Egyes programok, más-más nemzeti sajátosságokat is figyelembe vevő kódlapot használhatnak. A teljes magyar karakterkészletet pl. a 852-es kódlap tartalmazza. A következő táblázatban ennek a kódlapnak néhány ékezetes karakterét és annak kódját olvashatjuk.

á	Á	é	É	í	Í	ó	Ó	ö	Ö
160	181	130	144	161	214	162	224	148	153

ő	Ő	ú	Ú	ü	Ü	ű	Ű
139	138	163	233	129	154	251	235

Számítógépen az **Alt** billentyűt lenyomva a (jobb oldali) numerikus billentyűzetről begépetelt **kódszám** hatására a monitoron megjelenik a karakter.

**Alt + kódszám ⇒ karakter**

A teljes ASCII-táblázatot a könyv végén találod meg.



*A hagyományos és általánosan elterjedt kódrendszereken túlmenően a hírközlésben és a számítástechnikában rendkívül sokféle kódrendszer létezik (morzekód, telexkód, vonalkód, BCD-kód, számábrázolás, színekód, hangkód stb.).*



### Ellenőrző kérdések

1. Mi az ASCII-kódrendszer?
2. Melyik kódlap tartalmazza a magyar karakterkészletet?



## Feladatok

1. Kapcsold be a számítógépet a tanár utasítása szerint, és felhasználva az ASCII-kódtáblázatot, a számkódok segítségével írd ki a monitorra minél több karaktert! Például: **ALT + 65**  $\Rightarrow$  **A**. Eközben használd a numerikus billentyűzetet!
2. Írd le a neved ASCII-számkóddal!

## ADATÁTVITEL

A számítógépek nemcsak tárolják, átalakítják a jeleket (biteket, byte-okat), hanem mozgatják is egyik helyről a másikra, egyik eszközből a másikba. Ez a mozgatás igen gyorsan történik, általában elektromos jelvezetékeken. Gyorsaságát az adatátviteli sebességgel jellemezzük, amely a számítógépes hálózati kommunikációnak is fontos paramétere. Napjainkban az egyik legismertebb ilyen világméretű hálózat az **Internet**.



Egy vezeték valamely keresztmetszetén másodpercenként továbbított adatmennyiséget az adatátvitel sebességének nevezzük.

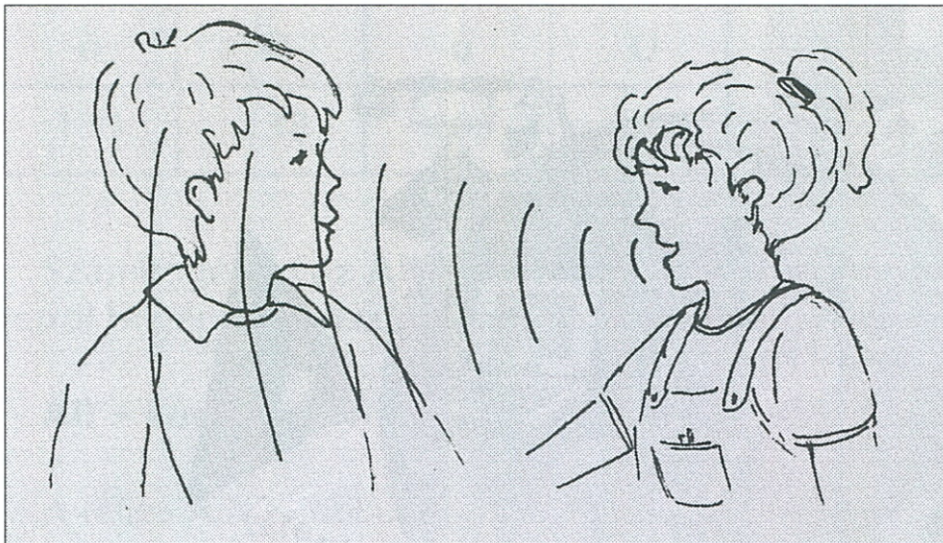
Az adatátvitel sebességét  $v$ -vel jelöljük:  $v = a/t$ , ahol  $a$  az adatmennyiség és  $t$  az idő.

Egysége a **bit/s**, azaz **bps**.

Az egyes hardvereszközök fizikai tulajdonságaitól függ, hogy egy bit továbbításához legalább mennyi idő szükséges. Minél kisebb ez az idő, annál nagyobb a vezeték adatátviteli sebessége.

Ha **egy** vezetéken küldjük a jeleket, **soros adattovábbításról** van szó, mert a jeleket sorban egymás után kell küldeni. Növekszik az adatátvitel sebessége, ha **párhuzamosan egyszerre több** vezetéken haladnak a jelek. Ekkor az átvitt adatmennyiségek összeadódnak, és így a sebességek is (pl. 8 vezeték használata esetén egyszerre 8 bit továbbítható, így az adattovábbítás sebessége is nyolcszoros lesz.)

A jelek valamilyen kézzelfogható anyaggal – pl. könyvvel, újsággal, filmen, mágnesszalagon, mágneslemezen – is továbbíthatók. Kevésbé megfogható, de igen fontos lehetőség az is, ha a jeleket valamilyen fizikai mező vagy közeg hordozza, pl. hang vagy fény a levegőben, elektromos jel a rézvezetékben, lézerfény az üvegszálban, ill. elektromágneses hullám (speciálisan rádióhullám).



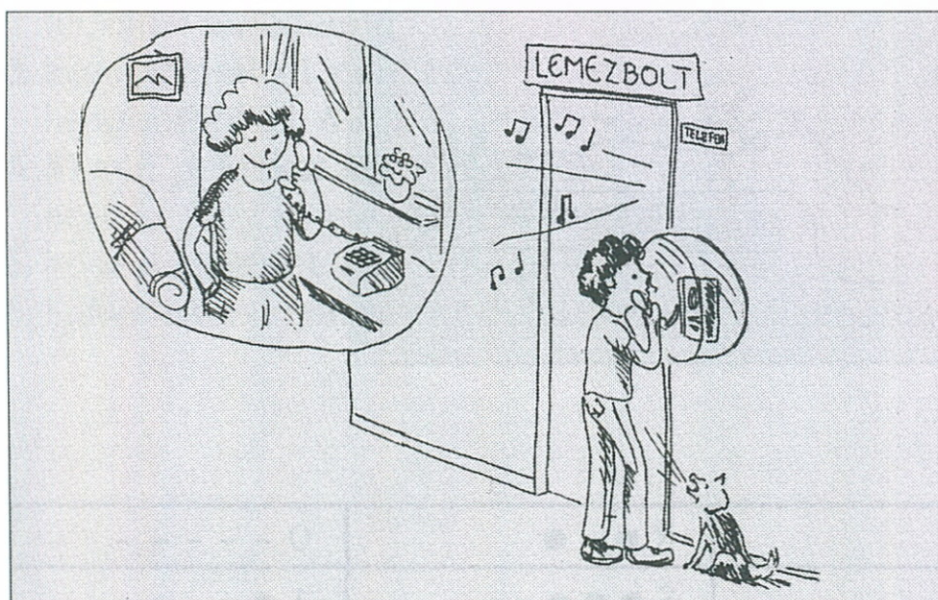
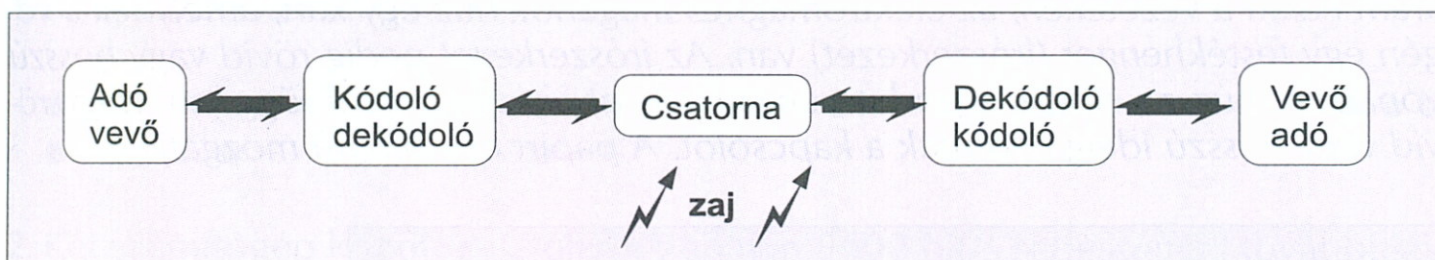
**Azt a mezőt, közeget, vezetéket, amely jeleket továbbít, információs csatornának (adatcsatornának) nevezzük.**

Egy csatorna lehet egyirányú, pl. a rádióadó és a rádióvevő között, ill. kétirányú, azaz kommunikációra alkalmas, pl. a telefonvonal.

Ha beszélgetsz a barátoddal telefonon, akkor a jeltovábbítás szempontjából a következő fontosabb dolgok történnek:

- Megfogalmazott gondolataidat, érzéseidet a mikrofonba mondod – Te vagy az **adó**.

- A mikrofon felveszi a hangot, elektromos jelekké alakítja, és azonnal továbbítja – ez a **kódoló**.
- Egy elektromos érpár (vezeték) a telefonközpont kapcsolóin és erősítőkön keresztül viszi a jelet a barátod készülékébe – ez az **információs csatorna**.
- A csatorna sokszor zajos környezetben van, és nincs elég jól szigetelve, ezért a jelhez több-kevesebb **zaj** (zavarójel: pl. mások beszéde) keveredhet. Valójában bárhol keveredhet zaj a hasznos jelhez, ha pl. az utcáról hívod barátodat, akkor már a beszédedhez (az adó jeléhez) is zaj keveredik.
- A barátod készülékében egy hangszóró átalakítja az elektromos jeleket beszéddé – ez a **dekódoló**.
- Végül a címzett hallja és felfogja a beszédet – ő a **vevő**.



Amikor a barátod beszél, fordított irányú, de hasonló folyamat játszódik le. A zaj ellen jó szigeteléssel, ill. zajmentes környezet választásával védekezhetünk. A legfontosabb az, hogy elég nagy legyen a jel-zaj arány, vagyis erős hangzavarban megfelel a hangos beszéd vagy a kiabálás is – ha már a zajt nem csökkenthetjük.

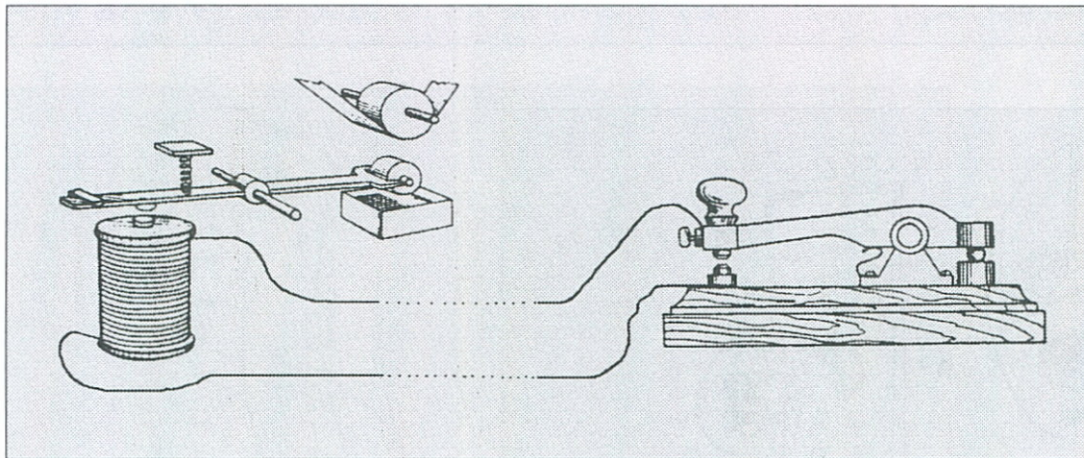
Egy másik módszer a zaj ellen, ha többször is elmondjuk ugyanazt, esetleg más módon. Így valószínű, hogy minden ismeret eljut a vevőhöz, korrigálni tudja a félrehallásokat, hibákat. Az ilyen közlést redundánsnak, terjengősnek nevezük. **A redundáns közleményben az információ nincs a legtömörebben megfogalmazva, de így nagyobb biztonsággal lehet venni.**





Samuel Morse (1791–1872) eredetileg festő volt, de 1837-től a távjelzéssel kezdett foglalkozni. 1844-ben elkészítette a Washington-Baltimore közötti villamos távíróvonalat. Morse készüléke pontokat és vonásokat írt papírszalagra. Ezeket a kódtáblázat alapján visszaírták betűkké. Némi gyakorlással a morzejelek olvasását éppen olyan jól meg lehet tanulni, mint a szövegolvasást. Morse találmánya bevált, kiszorította a régebbi berendezéseket.

A készülék működése egyszerű: Az adás pillanatkapcsolóval történik. Ha lenyomjuk a kapcsolót, zár az áramkör, ha elengedjük, megszakad. Egy hosszú vezeték végén van a vevő írókészülék, amelynek a fő része egy elektromágnes. Ha áram halad a vezetéken, az elektromágnes magához húz egy kart, amelynek a végén egy festékhenger (írószerkezet) van. Az írószerkezet pedig rövid vagy hosszú vonalakat húz az előtte egyenletesen mozgó papírcsíkra, attól függően, hogy rövid vagy hosszú ideig nyomjuk a kapcsolót. A papírt egy óramű mozgatja.



Morze-féle táblázat

A ● -	I ● ●	R ● - ●	0 - - - - -
Á ● - - ● -	J ● - - -	S ● ● ●	1 ● - - - -
B - ● ● ●	K - ● -	T -	2 ● ● - - - -
C - ● - ●	L ● - ● ●	U ● ● -	3 ● ● ● - -
D - ● ●	M - -	Ü ● ● - -	4 ● ● ● ● -
E ●	N - ●	V ● ● ● -	5 ● ● ● ● ●
É ● ● - ● ●	O - - -	W ● - -	6 - ● ● ● ●
F ● ● - ●	Ö - - - ●	X - ● ● -	7 - - ● ● ●
G - - ●	P ● - - ●	Y - ● - -	8 - - - ● ●
H ● ● ● ●	Q - - ● -	Z - - ● ●	9 - - - - ●



## Ellenőrző kérdések

1. Mi az adatátvitel sebessége?
2. Hogyan számolod ki az adatátviteli sebességet?
3. Mi a párhuzamos adatátvitel?
4. Mit nevezünk információs csatornának?
5. Sorold fel a kommunikáció folyamatának fontosabb elemeit!



## Feladatok

1. Mit jelent a következő morzeüzenet:

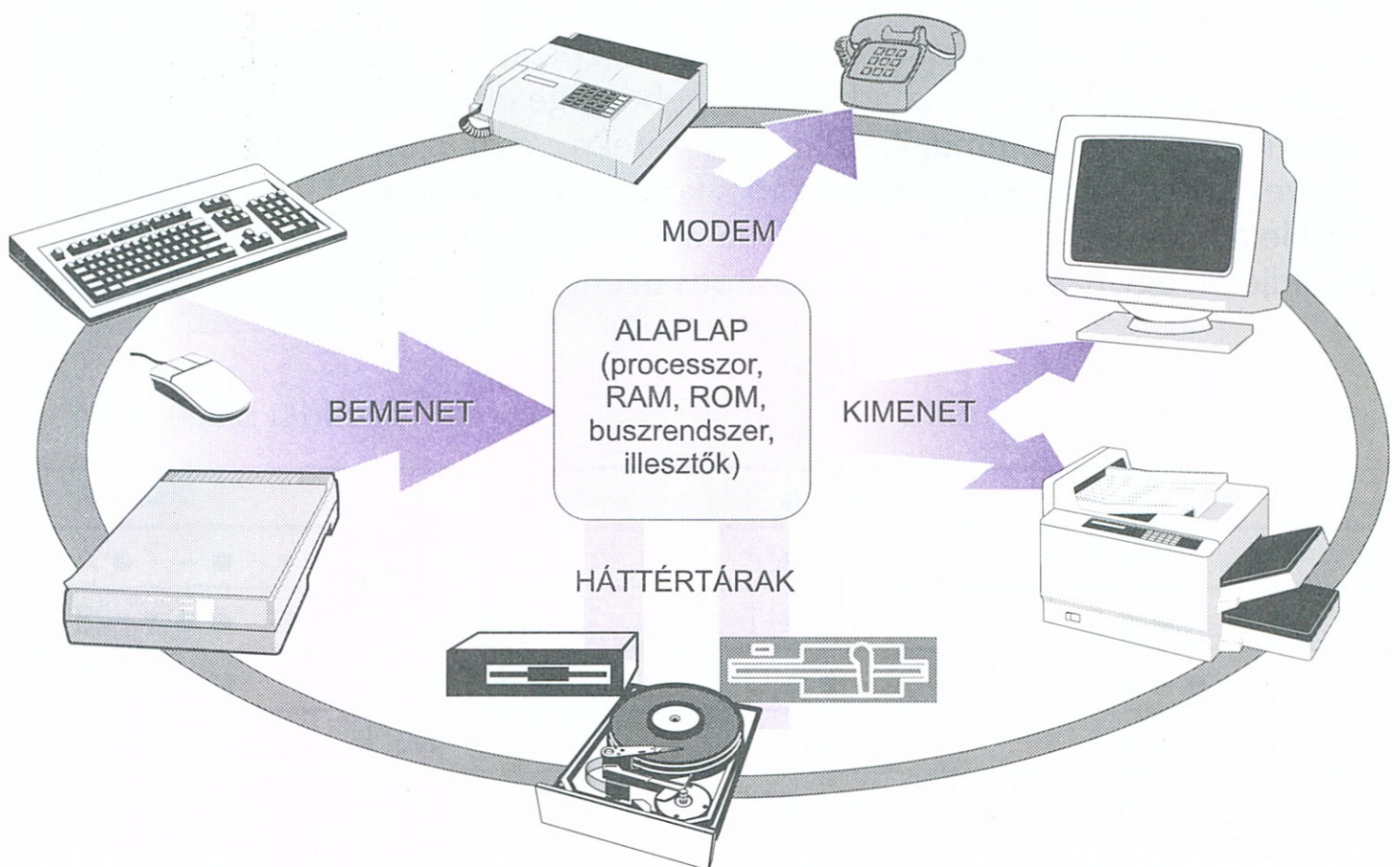


2. Két számítógép között egy telefonvonalon 9600 bit/s sebességgel továbbítják az adatokat. Legalább mennyi idő szükséges egy 100 oldalas 2000 karakter/oldal terjedelmű könyv továbbításához?
3. Egy nyomtató 1 perc alatt nyomtat ki egy rajzot, amely 320 x 200 fekete-fehér képpontból áll. Mennyi az adatközlés sebessége ebben az esetben?
4. Mondj példákat kommunikációs rendszerekre, eszközökre, módszerekre, és nevezd meg az egyes elemeket (adó, kódoló, csatorna stb.)!
5. Írd le a nevedet, vagy írd egy levelet a barátodnak morzejelekkel!
6. Küldjétek egymásnak kódolt (titkosított) üzeneteket!

## A SZEMÉLYI SZÁMÍTÓGÉP RÉSZEI

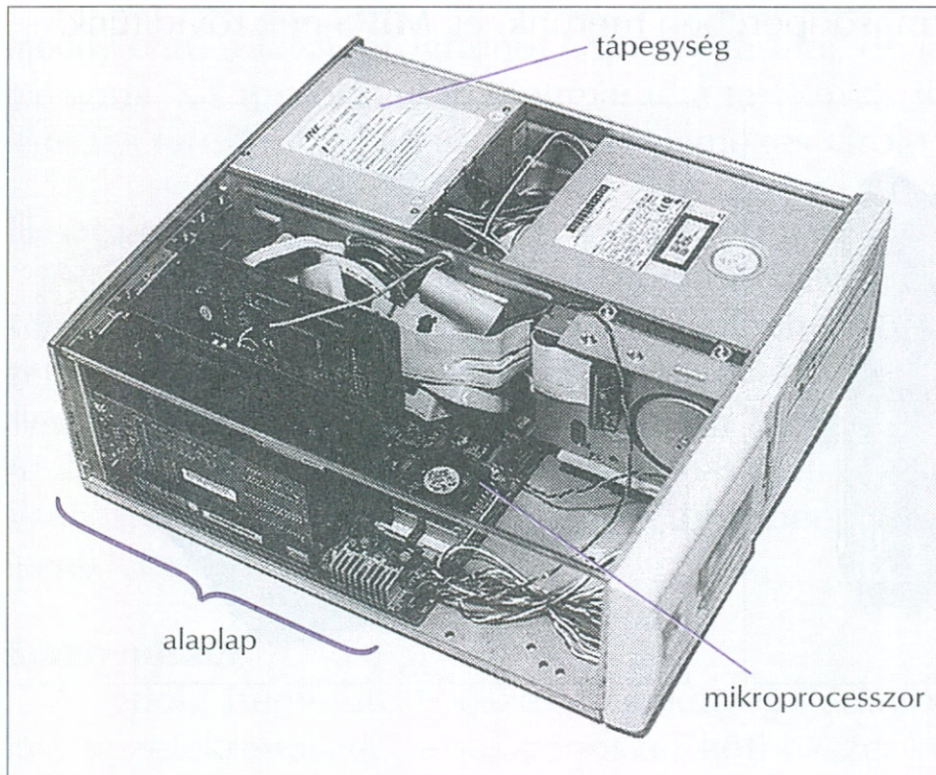
Az iskolában vagy otthon valószínűleg kapcsolatba kerültél már a személyi számítógépek valamelyikével. Ezek közül az IBM és az IBM-kompatibilis (az IBM márkájú személyi számítógépeknek megfelelő, de más által gyártott) gépek a legelterjedtebbek. A továbbiakban ezzel a típussal foglalkozunk.

Egy személyi számítógép felépítése:



Ha kívülről nézel meg egy személyi számítógépet, biztosan megtalálod a házat, a billentyűzetet, az egeret és a monitort.

## MI VAN A HÁZ BELSEJÉBEN?



### Tápegység

A tápegység biztosítja a megfelelő egyenfeszültségeket, ill. az elektromos energiát a számítógép különböző részei számára.



Soha ne nyisd fel a gép tetejét, és ne nyúlj bele, ha csatlakoztattad a gépet a konnektorhoz (esetleg be is kapcsoltad), mert veszélyes lehet Rád is, és a gépre is.

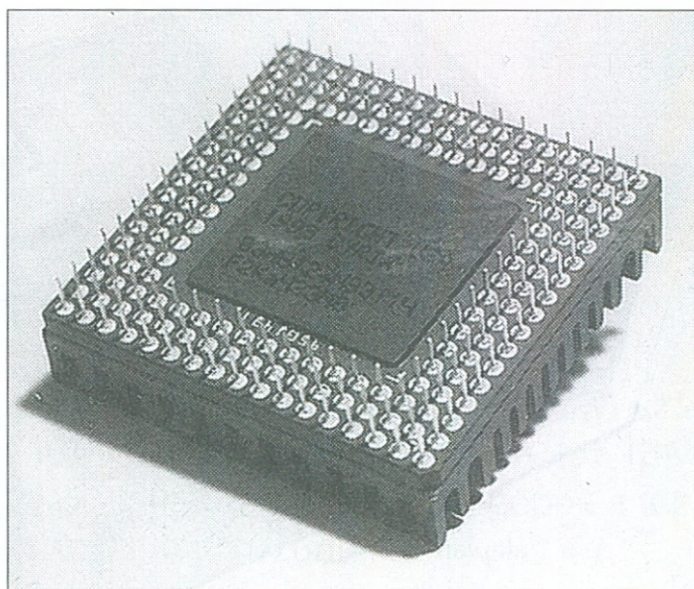
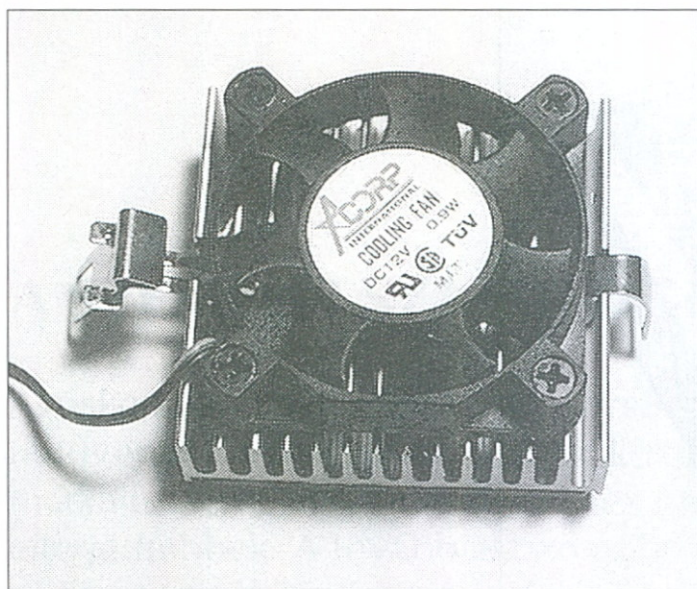
### Alaplap

Az alaplapon van a legfontosabb egység, a mikroprocesszor. Itt található a memória-áramkörök, a buszrendszer, és a különböző egységek illesztőáramkörei.

### Mikroprocesszor (vezérlőegység)

Vezérli a számítógépet, végrehajtja az utasításokat, műveleteket, számításokat. Általában a processzor típusával szoktuk jellemezni a gépet. Íme, a gépek néhány típusa fejlettségük sorrendjében: 386, 486, Pentium. A processzor jellemzője, hogy egyszerre hány bit adattal tud műveletet végezni (16, 32 vagy 64). A gyorsaságát főként az határozza meg, hogy milyen frekvencián működik. Egy 486-os típusú processzor 33 és 133 MHz közötti frekvenciával dolgozik. (A frekvencia a másodpercenkénti rezgések számát jelenti. A 33 MHz 33 millió rezgés másodpercenként.) A rezgéseket egy kvarcóra állítja elő, ezért órajelnek

is nevezik. A processzorok többségének egy művelethez egynél több rezgésre van szüksége. A mi szempontunkból a frekvenciánál fontosabb a **műveleti sebesség**, amit millió művelet/másodpercben mérünk, és **MIPS**-nek rövidítünk.

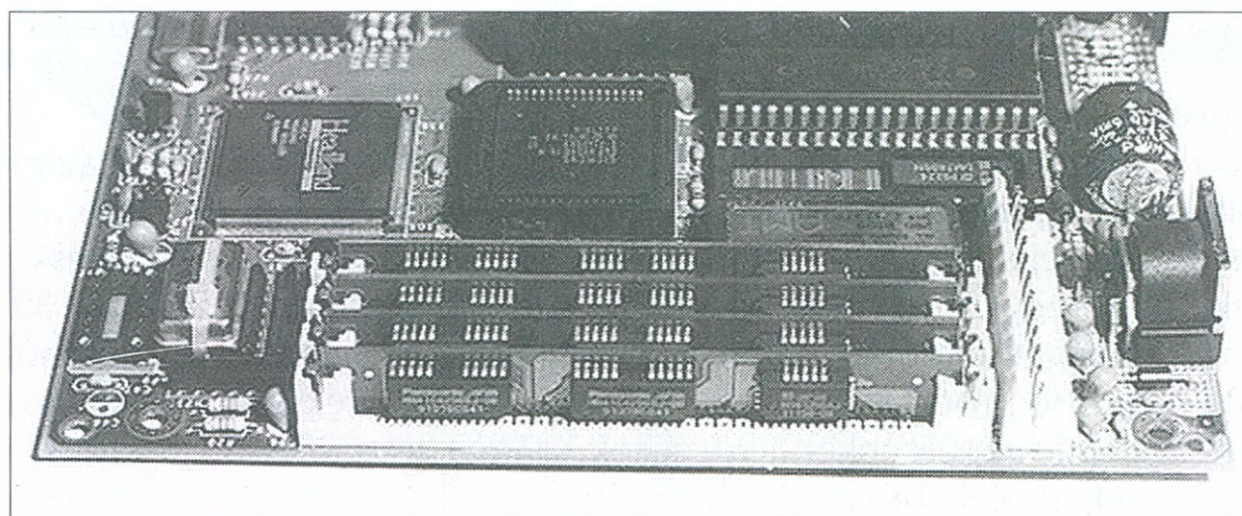


## Memória

A mikroprocesszor által közvetlenül és gyorsan elérhető adattároló a memória. Itt helyezkednek el az éppen futó programok és az adatok. A memória legfontosabb jellemzője a kapacitás, ami azt jelenti, hogy mennyi adatmennyiséget képes tárolni.

A csak olvasható memóriát ROM-nak nevezzük (Read Only Memory), pl. ilyen típusúba írják be gyárilag a gép működéséhez, bekapcsolás utáni „feléledéséhez” elengedhetetlen programokat, adatokat. Az írható-olvasható memóriát RAM-nak nevezzük (Random Access Memory). Egy átlagos PC 4–16 Mbyte RAM-mal rendelkezik.

A RAM az áram kikapcsolásakor elveszíti tartalmát. Ezért kikapcsolás előtt mentsd a háttértárra a munkáidat! Egyébként ajánlatos munkavégzés közben is többször menteni, mert a váratlan áramkimaradás esetén elvesznek a memóriában lévő adataink.



## Háttértárak

Ezeket a memória lényeges „meghosszabbításának” tekintheted. Nagy mennyiségű adatot, programot képesek tárolni, de lényegesen lassabbak, mint a memória. Kikapcsolás után megmarad a tartalmuk. Ilyen háttértár pl. a hajlékonylemezes tároló (floppy disk), a merevlemezes tároló (winchester) és a CD-ROM.

## Illesztők

A perifériák (külső egységek: monitor, billentyűzet, nyomtató) illesztőáramkörökön keresztül kapcsolódnak az alaplaphoz. Néhány illesztő (pl. a billentyűzetvezérlő) az alaplapba van beépítve. Más illesztők külön kártyákon, ún. illesztőkártyákon vagy vezérlőkártyákon helyezkednek el (pl. monitorvezérlő). Az illesztőkártyákat az alaplapon levő 6-8 csatlakozó valamelyikébe kell bedugni. Vannak olyan alaplapok, amelyeken több beépített illesztő, pl. hajlékonylemez, merevlemez és CD-vezérlő van.

## Buszrendszer

Az eddig bemutatott részegységeket egy adatforgalmat lebonyolító elektromos vezetékrendszer, a buszrendszer köti össze. Létezik címbusz, adatbusz és vezérlőbusz. A buszrendszerhez csatlakoznak az illesztőkártyák és az alaplapba épített illesztők is.

A buszrendszer bármikor bővíthető, így nem kell egyszerre minden eszközt (CD, szkennert, hangkártya stb.) megvenni. Mindenki úgy fejlesztheti a gépét, ahogyan az igényei nőnek, és a pénztárcája engedi.

A számítógép használatához szükségesek az adatok és programok ki- és bemenetét, tárolását megvalósító eszközök. A továbbiakban ezekkel foglalkozunk, elsőként a legfeltűnőbbel, a monitorral.



### Ellenőrző kérdések

1. Mi a mikroprocesszor feladata?
2. Mikor vesznek el a RAM-ban lévő adatok?
3. Mire szolgál a buszrendszer?

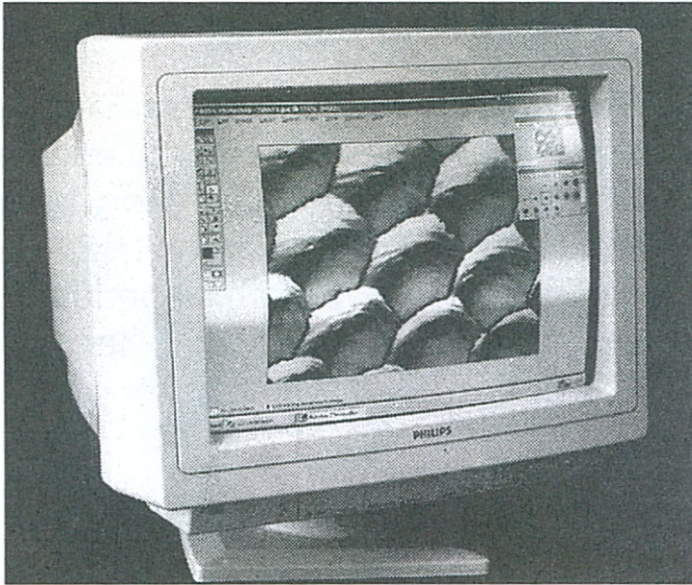


### Gyakorlat

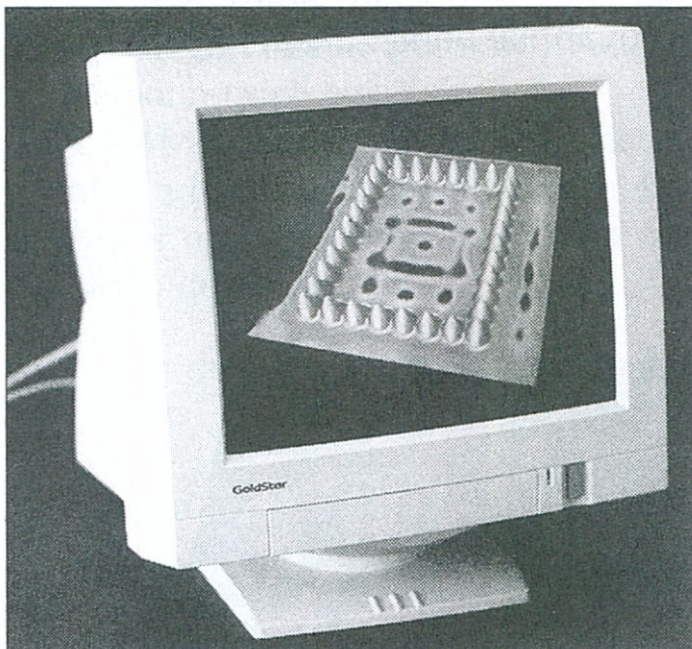
Kikapcsolás után, tanárod segítségével vizsgáljátok meg egy PC belsejét! Azonosítátok a különböző részeket!

# MIT KELL TUDNI A MONITOROKRÓL?

Naponta akár több órát is eltölthetsz a számítógép monitora (képernyője) előtt, fontos tehát, hogy alaposan megismerd!



A monitor a számítógép elsődleges kimeneti egysége. A működési elvet tekintve a monitor lehet (a tv-hez hasonló) katódsugárcsöves vagy a hordozható gépekben használt folyadékkristályos. A monitort jellemzi a képátmérő, a megjelenített színek száma, a felbontás, a sugárzás és a képfrekvencia.



## Képátmérő

A képátmérőt colban (2,54 cm) szokták megadni, a gyakoribb méretek: 12", 14", 15", 17" és 21". (A col jele a ", de használják az angol inch-et és a magyar hüvelyket is.)

## Felbontás

A képernyő apró pontokból áll. Minél több képpontból áll egy kép, annál részletesebb, annál jobb a felbontása.

**A képernyő felbontását a vízszintesen és a függőlegesen megjeleníthető képpontok számával adják meg (pl.: 600 × 800, 1024 × 768).**

## A megjelenített színek száma

A monokróm képernyőknek egy háttérszínük (fekete) és egy írószínük (pl.: fehér vagy zöld) van.

A színes képernyőkön előállítható színek száma a monitortól és a hozzá tartozó illesztőkártyától függ. Lehet pl. 4, 16, 256, 65 536, 16 777 216.

## Képfrekvencia

A képfrekvencia arra utal, hogy a monitor másodpercenként hányszor frissíti (rajzolja újra) a képernyő tartalmát. Ha elég nagy a képfrekvencia (több mint 70 frissítés másodpercenként, 70 Hz), akkor a kép nem villózik. Az ilyen monitorok jele **NI**.

Az alacsony sugárzású monitor (jele **LR**) képernyője elektrosztatikusan nem töltődik fel.



A hűtőnyílásokat ne takard le (pl. fűzettel), mert a monitor túlmelegszik!  
A képernyőt tartsd tisztán! Kikapcsolt állapotban töröld le róla a port!

A kép minőségét a monitor típusa és az illesztőkártya (ún. videokártya) együttesen határozza meg. A videokártyának saját memóriája (RAM-ja) van, ami a képet tárolja. A következő táblázat néhány elterjedt típus (a vezérlőkártya és a hozzá tartozó monitor) fontosabb paramétereit tartalmazza:

típus	felbontás	szín
VGA	640 × 480 720 × 400	2, 16, 256
SVGA	800 × 600 1024 × 768 1280 × 1024	2, 16, 256 65536 16 777 216
XGA	1600 × 1200	16 777 216





## Ellenőrző kérdések

1. Milyen adatokkal jellemezhetők a monitorok?
2. Mit jelent a felbontás?
3. Hány színű lehet az SVGA monitor, és mennyi lehet a felbontása?



## Feladatok

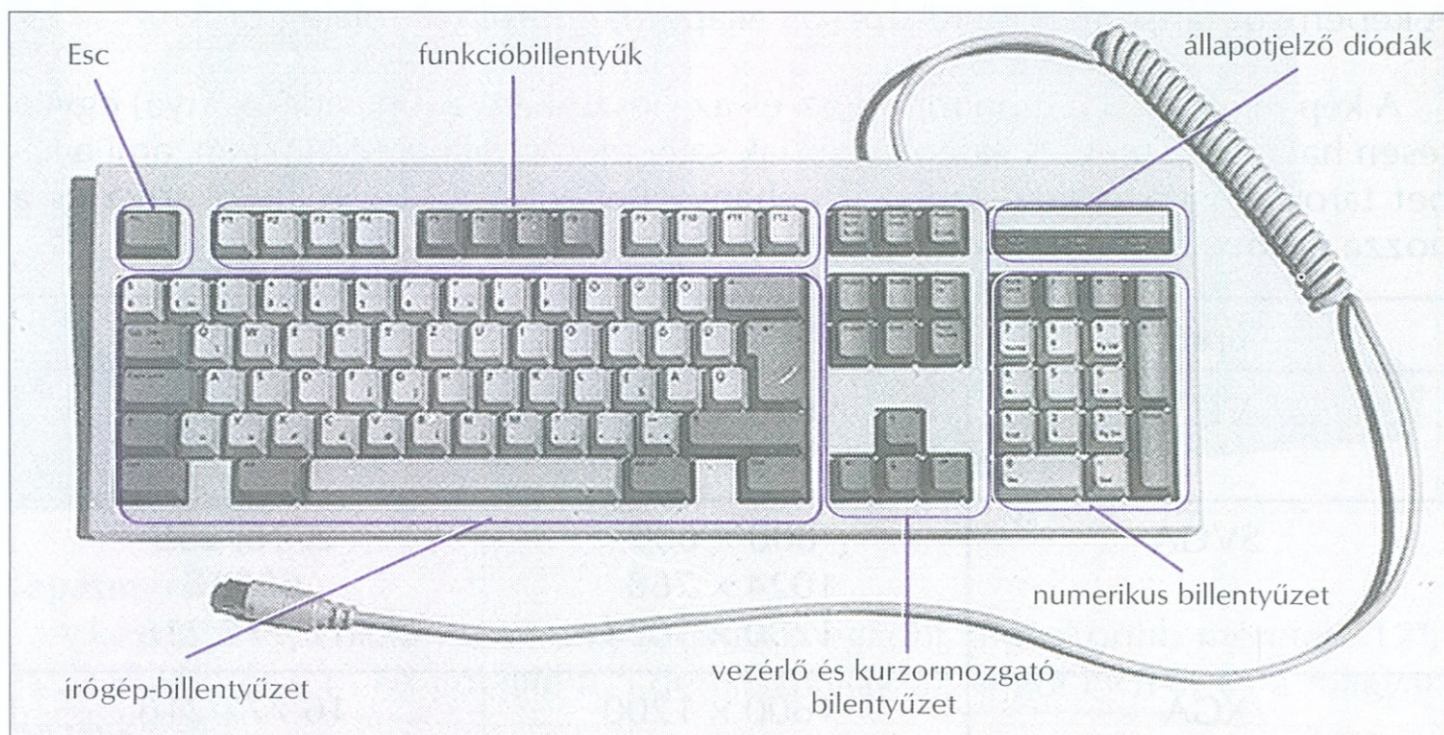
1. Hány Kbyte memória kell egy  $1024 \times 768$  képpontból álló, 256 színű kép tárolásához?
2. Hány Kbyte memóriát foglal el egy  $640 \times 480$  képpontból álló 16 színű kép?

## A BILLENTYŰZET

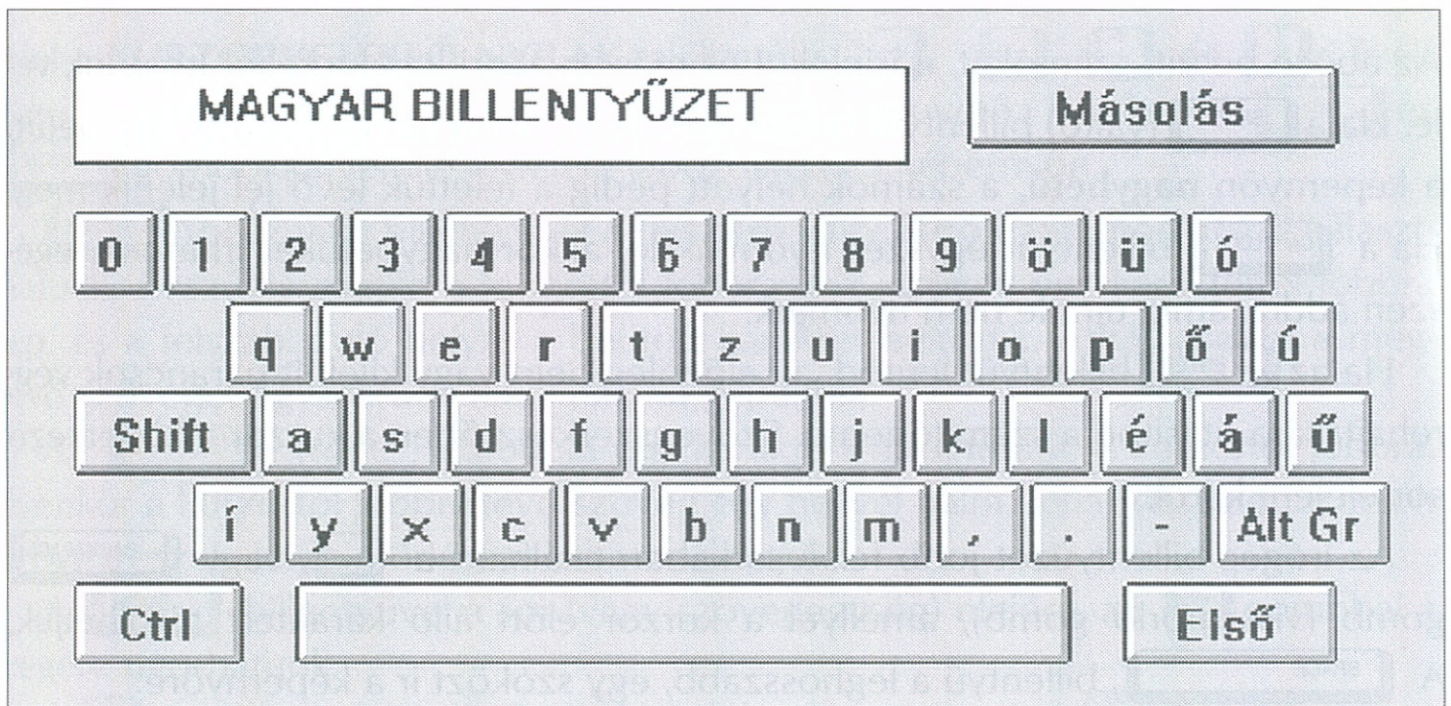
Amíg nem tudunk folyékonyan beszélgetni a számítógéppel, addig valószínűleg a billentyűzet marad a legfontosabb beviteli eszköz. Jelenleg is léteznek olyan programok, amelyekkel (korlátozottan) lehet beszélgetni: igen sok mondatot, szót azonosítanak és ennek megfelelően válaszolnak, ill. elvégzik a kiadott utasítást. Ma még azonban igen fontos a billentyűzet, ezért bővebben is foglalkozunk vele.

A billentyűzettel (klaviatúrával) parancsokat, programokat, adatokat írhatunk a gépbe.

Sokféle billentyűzet létezik. Míg a hordozható gépeknek többféle, speciális billentyűzete van, addig az asztali gépek billentyűzete leggyakrabban 101 vagy 102 gombos. Tekintsük át a főbb billentyűcsoportokat:



A következő ábrán az írógép-billentyűzet magyar változata látható:



Ha a képernyő szöveges üzemmódban van, ill. szövegszerkesztővel dolgozunk, a billentyűzeten begépeltek szöveg megjelenik a képernyőn. Az éppen érvényes pozíciót, a **kurzor** mutatja.

**A kurzor a monitoron megtalálható jel, általában villogó vonal. A kurzor mögött jelenik meg a begépeltek karakter.**

Ha egy billentyűt hosszabb ideig (kb. egy másodpercig) lenyomsz, a megfelelő karakter többször és gyorsan kiíródik a képernyőre.



### Figyelmeztetés!

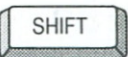
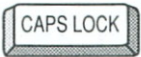
A gombokat pillanatszerűen, rövid ideig nyomd!  
A billentyűk kiosztása (jelentése) szoftverrel megváltoztatható.





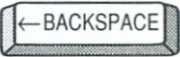

### Gyakorlat

Kapcsold be a számítógépet! Indíts el a tanár segítségével egy egyszerű szövegszerkesztőt, pl. a DOS-editort, az EDIT (Enter) paranccsal! Windows használata esetén futtasd a Kellékekből a Write szövegszerkesztőt! Próbáld ki a gyakorlatban a gombokat!



## AZ ÍRÓGÉP-BILLENTYŰZET


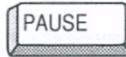
Az ábécé betűit, számokat, írásjeleket és más, gyakori karaktereket írhatunk vele. Ha a  (váltó) billentyűt lenyomjuk, és nyomva tartva leütünk egy betűt, a képernyőn **nagybetű**, a számok helyett pedig a felettük lévő jel jelenik meg. Ha a  billentyűt egyszer nyomjuk le, akkor nagybetűket írhatunk, egészen addig, amíg újra le nem nyomjuk.




Ha az  billentyűt leütöd, az előzőleg beírt vagy kijelölt parancsok végrehajtására utasítod a számítógépet. Szövegszerkesztőben a kurzor a következő sor elejére kerül.

Az írógép-billentyűzet jobb felső sarkában található a , vagy  gomb (visszatörölő gomb), amellyel a kurzor előtt álló karaktert törölhetjük. A , billentyű a leghosszabb, egy szóközt ír a képernyőre.



Ha leütöd a  (tabulátor) billentyűt, valamilyen ugrásra utasítod a kurzort, pl. 8 karakterrel jobbra ugrik, a **SHIFT**-tel együtt lenyomva, balra ugrik.


A  és az  a **SHIFT**-hez hasonlóan jelentésmódosító hatású, csak más billentyűkkel együtt fejtik ki hatásukat. Ez a hatás attól is függhet, hogy melyik programot használjuk. A következő két kombinációt **csak a tanár utasítására próbáld ki!**

 +  az éppen futó program leállítását és az operációs rendszerbe való visszatérést eredményezheti,





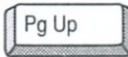
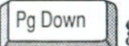
 +  +  hatására a számítógép a rendszert újraindítja (megindítás). Hasonló hatású a RESET gomb, de ez nem a billentyűzeten, hanem a dobozon található.

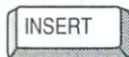
## A FUNKCIÓBILLENTYŰK ÉS AZ ESC


Az , ...  funkcióbillentyűk jelentése az alkalmazott programtól függ. (Pl. a Norton Commanderben **F1** a súgót, **F5** a másolást, **F8** a törlést jelenti.)

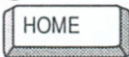

Az  (escape = menekülés) billentyű szintén a használt programtól függ, általában az utoljára kiadott utasítást vonja vissza, vagy a megjelenő ablakot tünteti el, vagy a begépelte sort érvényteleníti, de jelentheti a program befejezését is.



## KURZORMOZGATÓ ÉS -VEZÉRLŐ BILLENTYŰK


Alul a **KURZORVEZÉRLŐ NYILAK** találhatóak: , ,  és . Ezek igen fontosak, hiszen a kurzor mozgatása alapvető feladat. A  és  gombokkal felfelé és lefelé lapozhatunk a képernyőn.

A  (beszúrás) billentyűvel a **beszúró** vagy **átíró** üzemmód közül választhatunk. Beszúrás esetén a kurzortól jobbra lévő szöveg egy karakterrel jobbra lép, és a felszabaduló helyre a beütött karakter beíródik. Átírás esetén a meglévő szöveget egyszerűen felülírjuk, ha gépelünk.


A  (törlés) gombbal egy karaktert törölhetünk a kurzortól jobbra. Ilyenkor a kurzortól jobbra lévő szöveg egy hellyel balra, tehát visszalép. Kijelölt szövegrészek, képek törlésére is használhatjuk.

A  billentyűvel a sor (vagy szövegegység) elejére, az  gombbal a végére ugorhatunk.

A  gombbal a képernyő tartalmát lehet kinyomtatni. A  funkciója egyes programok esetében az, hogy a kurzormozgató billentyűk hatására nem a kurzor mozog fel-le, hanem a képernyőtartalom.

A  lenyomásakor a számítógép általában felfüggeszti a program futtatását, majd egy billentyű leütésére a program folytatódik.

## NUMERIKUS BILLENTYŰZET

Ha be van kapcsolva a , a számok és az alapszámjelölések itt kényelmesen begépelhetők. Ha nincs bekapcsolva, akkor a vezérlési és kurzormozgató funkciók érvényesek, ugyanúgy, mint a vezérlő, ill. a kurzormozgató billentyűknél. Itt van egy másik **Enter** billentyű is.

E terület fölött találhatóak (ha vannak) a világítódiódák, amelyek jelzik, hogy egyes jelentésmódosító billentyűk (**NUM LOCK**, **CAPS LOCK**, **SCROLL LOCK**) be vannak-e kapcsolva.

A billentyűzet használatának megtanulásához léteznek billentyűzetgyakorlatozó programok.



### Gyakorlat

1. Gépelj be legalább tíz sor szöveget a képernyőre! Gyakorold a begépelte szöveget az **Insert**, **Delete**, **Home**, **End** stb. billentyűk használatát!
2. Gépelj be a numerikus billentyűzeten számcsoportokat! Rendezzettek verseny! Mennyi idő alatt tudsz 100 számjegyet begépelni?

# AZ EGÉR

Az egér (mouse) sok program adatbeviteli eszköze. Kezelését könnyen elsajátíthatod, ahogyan mozgatod a kezdeddel együtt az egeret az asztalon, úgy mozog az egérkurzor is a képernyőn. Az egérben egy golyó van, amely a mozgatáskor elfordul. A golyó mozgását kis görgők veszik át. A görgők elmozdulását egy elektronikai eszköz alakítja át olyan jelekké, amit a számítógép képes értelmezni.

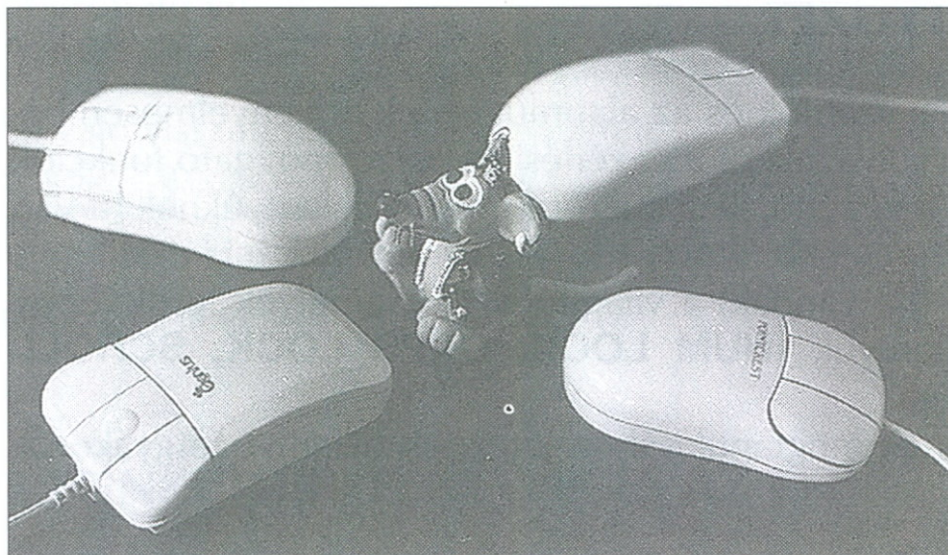


Tartsd tisztán az egérpadot, mert a golyó felhordja a piszkot a görgőkre, és ettől a görgők működése bizonytalanná válhat. Ezenkívül tisztítsd meg a golyót és a görgőket is!

Az egér burkolatán két vagy három kapcsoló van, amelyekkel különböző utasításokat adhatunk ki. Ha pl. a bal kapcsolóval **egyszer kattintunk**, kijelölhetünk a képernyőn egy ikont vagy egy szöveget, **duplán kattintva** egy program ikonjára, elindíthatjuk azt. A bal gombot lenyomva tartva „megfoghatunk” egy ikont, és tetszőleges helyre húzhatjuk.

Az egér helyes működéséhez program is szükséges, ilyen pl. **mouse.com** vagy **mouse.sys**.

Másmilyen egerek is vannak, pl. a hordozható gépeknél használt „hanyatt egér” (trackball), itt a golyót közvetlenül az ujjal lehet mozgatni.



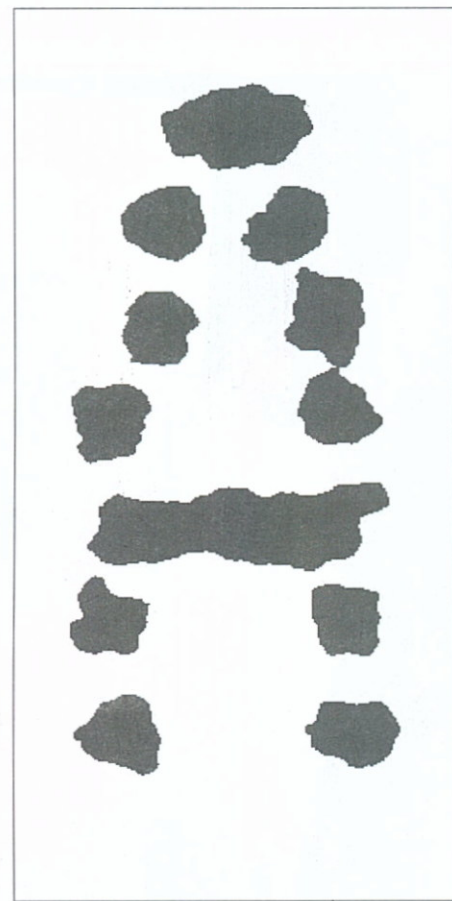
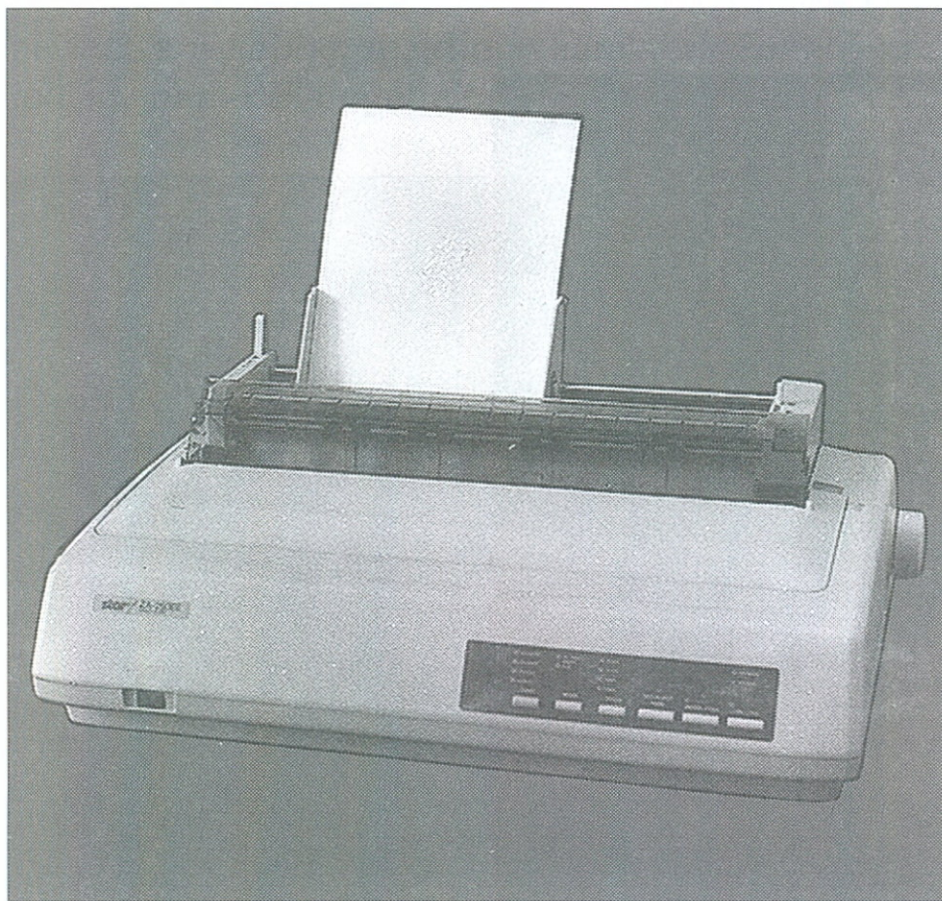
## Gyakorlat

Gyakorold az egér használatát pl. a Windows tankönyv programjával, vagy a Pasziánsz nevű kártyajátékprogrammal!

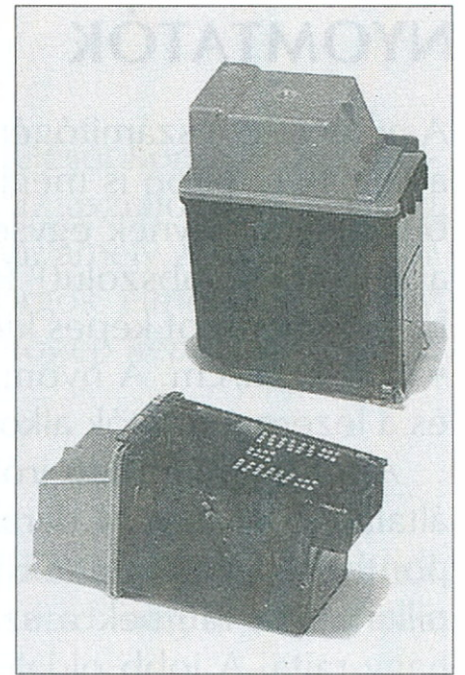
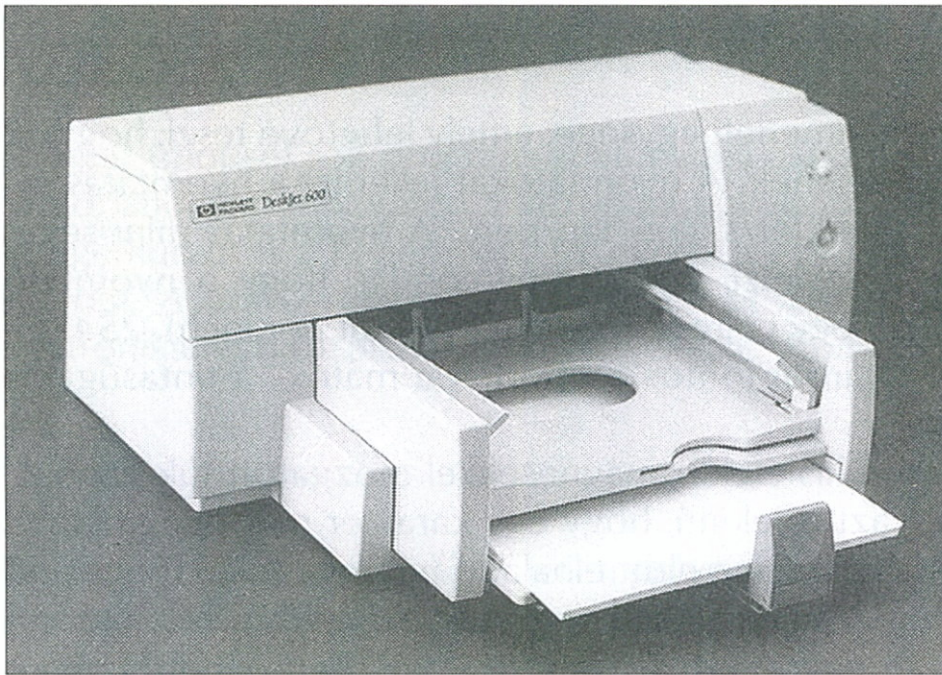
# NYOMTATÓK

A nyomtató a számítógép azon kimeneti egysége, amely lehetővé teszi, hogy az adataink papíron is megjelenhessenek. A nyomtatókat jellemzi a nyomtatás sebessége, amelynek egységei: karakter/s vagy lap/perc. A nyomtatás minőségét a nyomtató (abszolút) felbontása jelzi, amely megmondja, hogy a nyomtató hány képpontot képes kirajzolni egy colra. Egysége a dpi (dot per inch).  $254 \text{ dpi} = 100 \text{ pont/cm}$ . A nyomtatók három fontos csoportját a mátrix-, a tintasugaras és a lézernyomtatók alkotják.

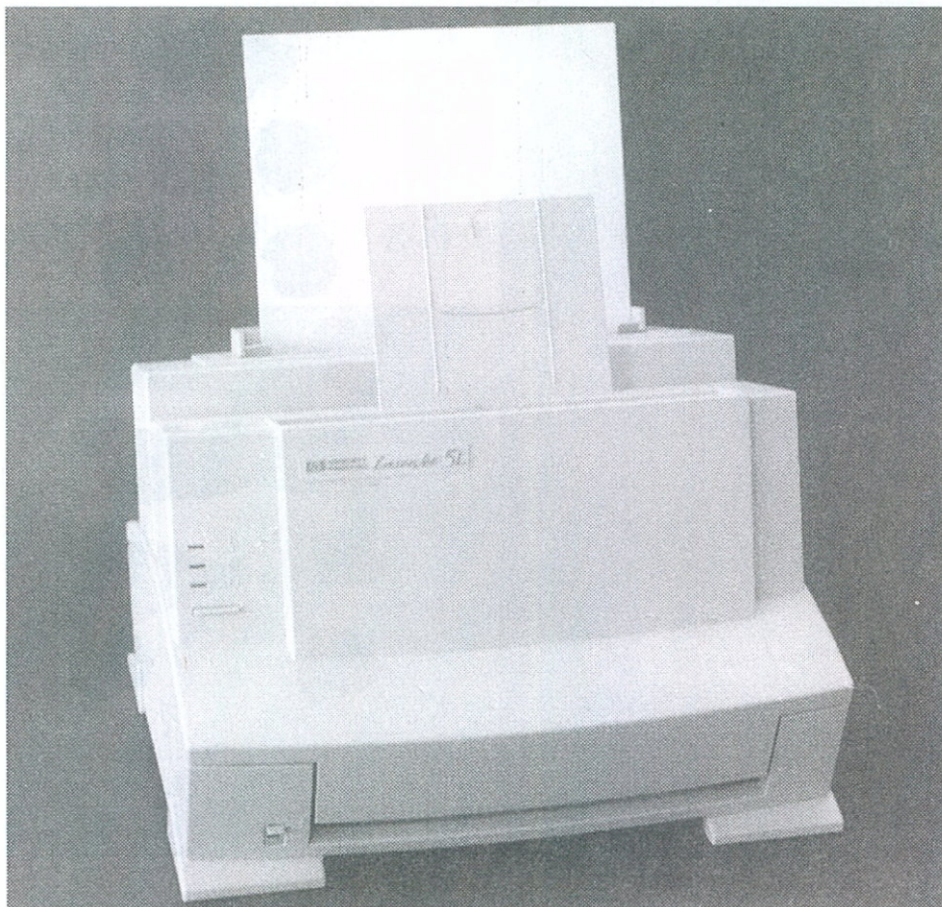
A **mátrixnyomtató** írófejében kis elektromágnesekkel mozgatott tűk vannak, általában 9 vagy 24 darab. Ez azt is jelenti, hogy egy karakter magassága ennyi pontból áll. Nyomtatáskor a festékszalag előtt elhaladó fejben a tűk a megfelelő pillanatban ráütnek a szalagra, amely hozzányomódik a papírhoz és nyomot hagy rajta. A jobb oldali ábrán egy mátrixnyomtatóval készített A betű látható sokszoros nagyításban.



A **tintasugaras nyomtatóban** egy apró lyukakkal teli fej mozog a papír előtt, és a megfelelő helyen igen kis tintacseppeket lő a papírra. Ez a nyomtató csendesebb, és a felbontása is jó (300–700 dpi). Gazdag választék van színesen nyomtató típusokból is. A következő oldalon, a tintasugaras nyomtató fényképe mellett, a hozzátartozó tintapatronok képe van.



A **lézernyomtatókra** az igen nagy felbontás (300–1200 dpi), a kiváló minőség és a gyors nyomtatás a jellemző.



A lézernyomtatót jól szellőző helyen ajánlatos használni, mert a gép működése során ózon és oldószergőzök keletkeznek, és ezek mérgező anyagok.



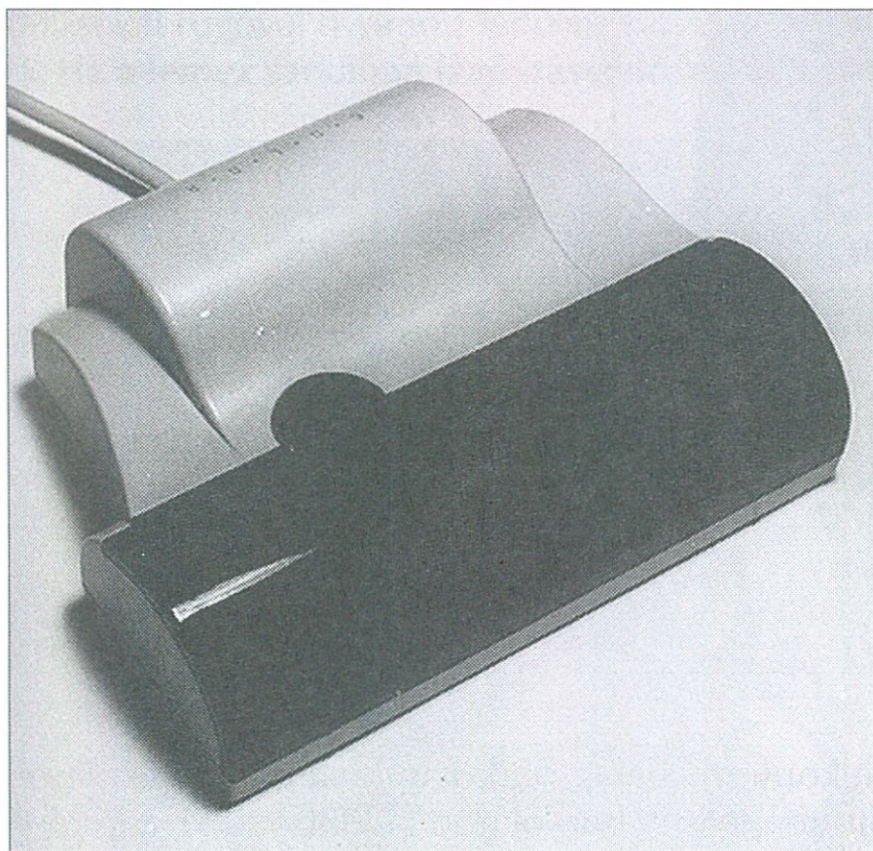
## Ellenőrző kérdések

1. Mit jelent a dpi?
2. Hogyan működik a mátrixnyomtató?
3. Hogyan működik a tintasugaras nyomtató?

## SZKENNER (SCANNER)

Ha papíron lévő rajzaidat, szövegeidet vagy képeidet szeretnéd gyorsan és egyszerűen bevinni a számítógép memóriájába, akkor szkenerre, képdigitalizálóra van szükséged. A szkener a papíron lévő képet, szöveget képpontonként, a szín és árnyalat szerint binárisan kódolja, azaz bitek sorozatára alakítja, ezt a számítógép megfelelő programmal képes értelmezni és megjeleníteni a képernyőn.

Érdekes és hasznos lehetőség, egy nyomtatott szöveget képként, szkenerrel bevihetsz a gépbe, majd egy karakterfelismerő programmal szöveggé alakíthatod, így nem kell begépelned.



Van kézi- és nagyobb, lapszkenner, színes vagy fekete-fehér képek beolvasására alkalmas változatban. A felbontás is széles skálájú, és általában beállítható (100–2400 dpi).

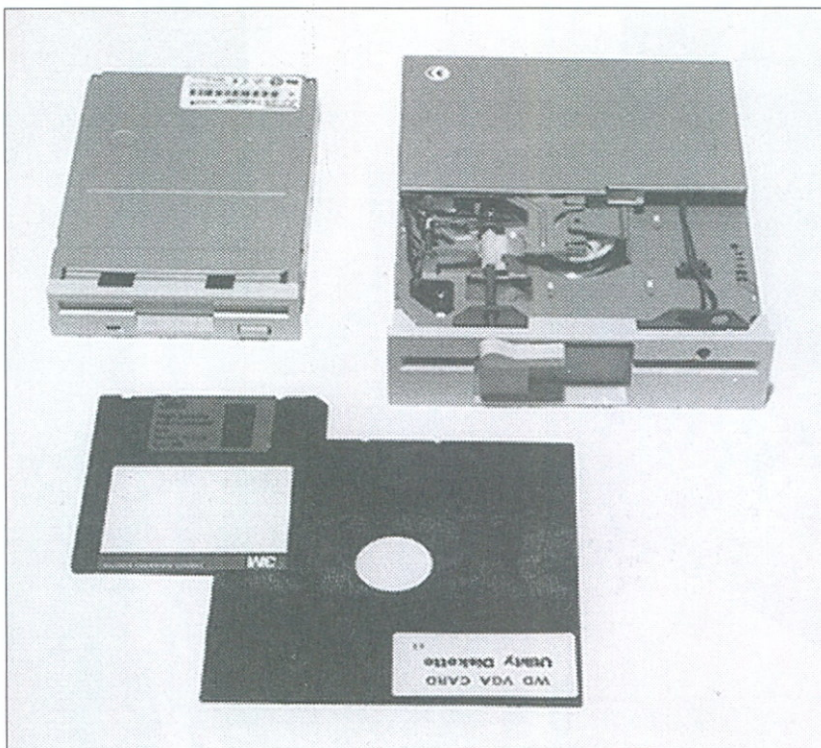


# LEMEZEGYSÉGEK ÉS LEMEZEK

Ha a programokat és a különféle adatokat (szövegeket, képeket, hangokat, táblázatokat) éppen nem használjuk, a háttértáron tároljuk. A leggyakrabban használt háttértárak a mágneses és az optikai lemezegységek. A legfőbb jellemzőjük a tárolókapacitás, vagyis az, hogy mennyi adatot képesek tárolni. Értékét Mbyte-ban, ill. Gbyte-ban adják meg.

## HAJLÉKONYLEMEZES TÁROLO

A hajlékonylemezes meghajtó vagy lemezegység (floppy disk drive, FDD) adathordozója a hajlékony mágneslemez (floppy disk). Működése a mágnesességen alapul. Többféle lemezegység létezik, a leggyakoribbak az 5,25"-os, 3,5"-os lemezmeghajtók. A számok azt jelentik, hogy mekkora hajlékonylemez lehet betenni. A kétféle lemez tárolókapacitása is különböző, az 5,25"-osé legtöbbször 1,2 Mbyte, a 3,5"-osé 1,44 Mbyte. A lemez egy vékony, hajlékony, műanyag korong, mindkét oldalán jól mágnesezhető anyaggal bevonva.



Az **5,25"-os lemezeknek** hajlékony műanyag burkolatuk van. A lemez közepén lévő lyukba illeszkedik a hajtótengely. A burkolaton különböző kivágások is vannak: nyílás az író-olvasó fejek számára, indexlyuk a lemez helyzetének érzékelésére és az írásvédő kivágás.

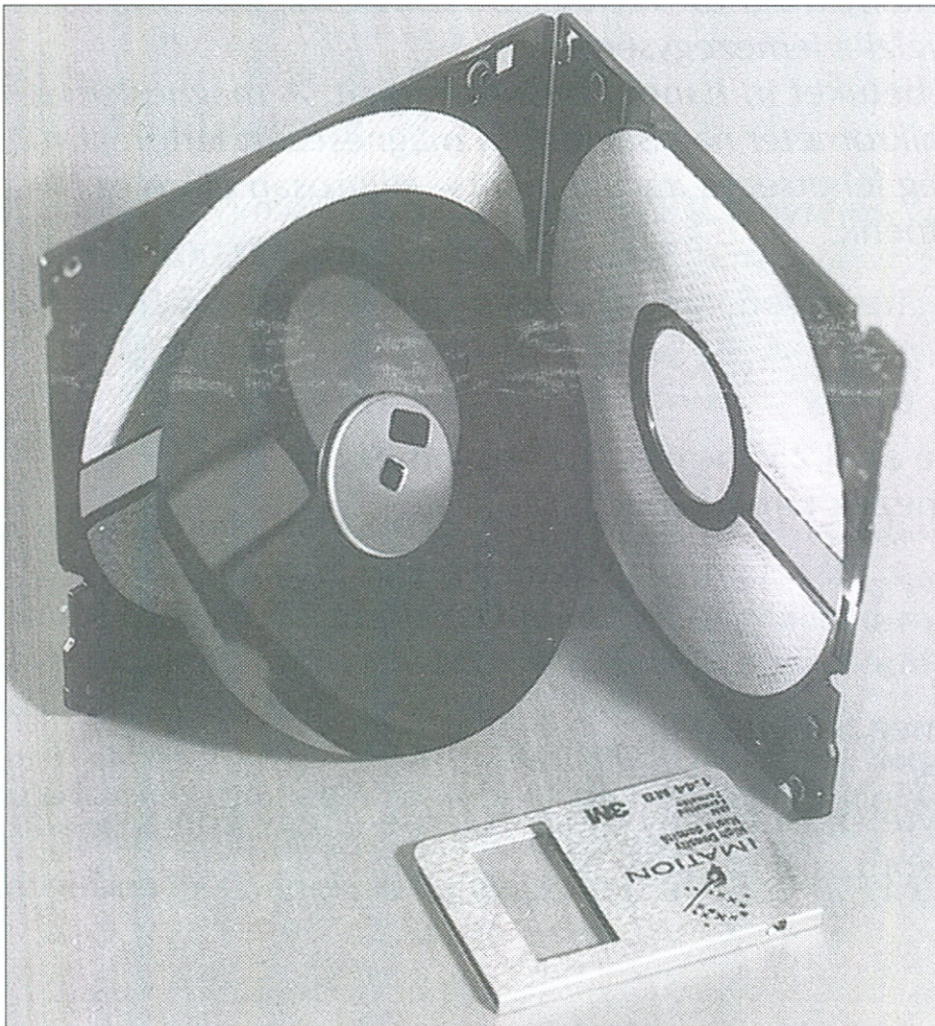
**Ha az írásvédő kivágást leragasztjuk, a lemezre nem lehet írni, de a véletlen törlés és a vírusok ellen is védelmet nyújt!**



A vírusok nemkívánatos rövid programok, amelyek önállóan szaporodnak és rombolnak.



A **3,5"-os lemezek** merev műanyag burkolatban vannak, a fejek számára kivágott nyílásokat rugóval nyomott fedőlap zárja el. Ez a meghajtóban automatikusan kinyílik. **Ha a lemez sarkában lévő írásvédő ablakot kinyitjuk, nem lehet írni a lemezre!**





A mágneslemezeket óvni kell a magas hőmérséklettől és az erős mágnesektől!

A lemezeken az adatok meghatározott logikai rendben helyezkednek el. Az új lemezre ezeket a fontos információkat rá kell írni a számítógéppel. Ezt a műveletet **formázásnak** nevezzük.

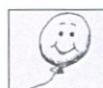


Ha a lemezen már voltak adatok, ezek a formázásakor (vagy a formázás utáni újabb íráskor) elvesznek. Különösen sok és értékes adat veszhet el a merevlemez formázásakor. **Ezért formázni csak megfontoltan szabad.**



Amikor behelyezzük a lemezt a meghajtóba, egy elektromotorral hajtott tengely megfogja. Ha a gép ír a lemezre vagy olvas róla, a motor gyorsan megforgatja a lemezt (360-at fordul percenként). A lemezegységben két író-olvasó fej veszi közre a lemezt, amelyek a forgó lemez felületéhez igen közel, sugárirányban mozoghatnak. Az író-olvasó fejek rendkívül kis méretű elektromágnesek, amelyek egyaránt tudnak írni és olvasni az érzékeny mágneses felületre. Az írást vagy az olvasást általában egy kis lámpa fénye jelzi a lemezegységen.

A fej természetesen nem betűket ír, hanem bitsorozatot. A mágneslemezen a bitek rendkívül kicsi (mikrométer nagyságrendű) mágnesezett tartományok, amelyeket – a mágnesezettség lehetséges irányai miatt – stílusosan előre és hátra mutató nyilacskákkal jelölhetnénk.



### Ellenőrző kérdések

1. Hogyan tehető írásvédetté az 5,25"-os és a 3,5"-os lemez?
2. Mekkora a különböző lemezek tárolókapacitása?



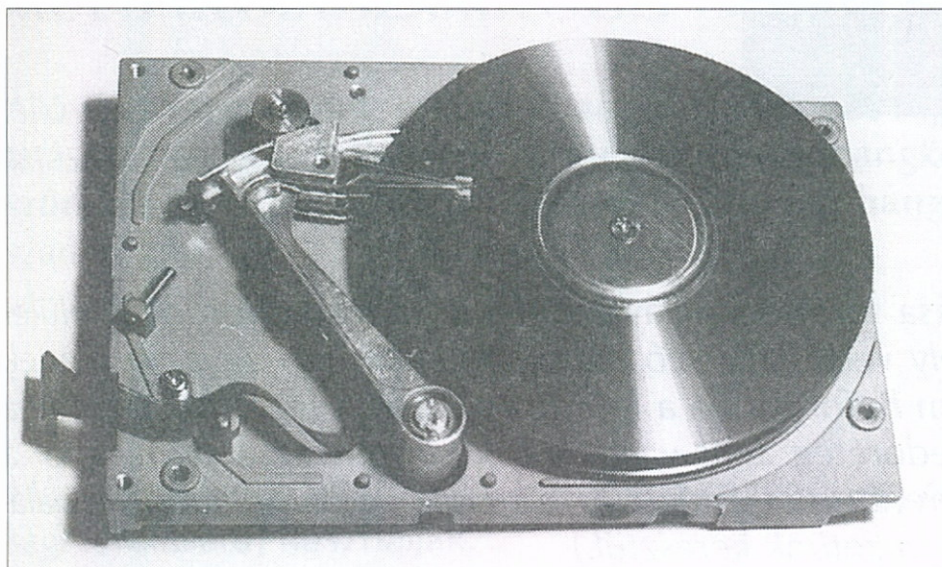
### Feladatok

1. Vizsgáld meg a kétféle lemez felépítését!
2. Mitől kell óvni a lemezeket?
3. Számold ki, hogy 1,2 Mbyte szöveg hány oldalon férne el egy könyvben, ha egy oldalra 50 sor és egy sorba 60 karakter fér!

## MEREVLEMEZ

A merevlemez, más néven winchester vagy HDD (hard disk drive) működése a mágnességen alapul, hasonlóan a hajlékonylemez meghajtóhoz, csak annál sokkal gyorsabb, és sokkal nagyobb a tárolókapacitása (1–10 Gbyte nagyságrendű). A lemezek itt nem cserélhetők, ugyanis az író-olvasó egységgel együtt egy légmentesen lezárt dobozban vannak, hogy a mágneses felület pormentes maradjon. Egy tengelyen több lemez van, és mindegyikhez két író-olvasó fej tartozik. A lemezek gyorsabban forognak (több ezret percenként).

Az írás-olvasás folyamatát egy kis világítólámpa (dióda) jelzi.



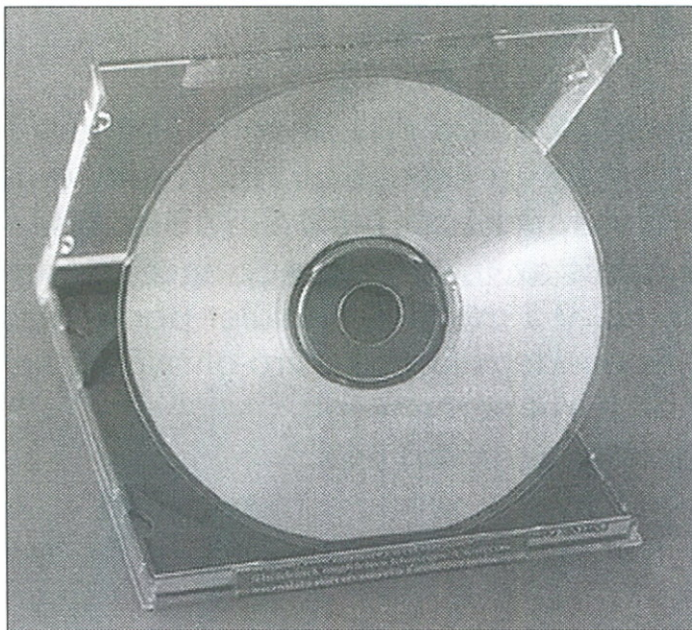
### Feladatok

1. Hány könyvnyi szöveg fér el egy 1,7 Gbyte-os winchesteren, ha egy átlagos könyv 800 000 karaktert tartalmaz?
2. Hány 0,5 Mbyte-os kép férne el a winchesteren?

## CD-ROM

A hangtechnikában a CD (Compact Disk = kompakt lemez) szinte teljesen kiszorította a hagyományos barázdás lemezt. Míg a „megfagyott hang” (a barázda) folytonos jel, addig a CD bitekben, digitális formában tárolja az adatokat, legyen az zene, kép vagy program. Jellegzetes csoportjaik az audio-CD, a video-CD és a CD-ROM.

A CD-ROM csak olvasható, cserélhető, nagy kapacitású (640 Mbyte) lemez, az adatokat csak egyszer lehet ráírni. Itt kell megjegyezni, hogy létezik magneto-optikai elven működő, írható-olvasható CD-RAM is. A CD lemezek is többféle méretben készülnek. A leggyakoribb átmérő 12 cm.



A CD lemez írása és olvasása lézertechnikával történik. Az adathordozó felület alumínium vagy arany, amely védő, átlátszó plasztikrétegbe van ágyazva. Ennek köszönhető, hogy a por nem befolyásolja a leolvasást, mert a lézer fókuszpontja az alumíniumfelületen, a védőréteg alatt van. (A fókuszpont az a pont, ahol a fénysugarak egy pontban keresztezik egymást. Ez történik, ha párhuzamosan haladó fénysugarat egy nagyítón vezetünk keresztül.)

A fókuszált lézer sokkal sűrűbben tudja a biteket beírni, mint a mágneses lemezmeghajtók író-olvasó feje. Ennek köszönhető, hogy a CD lemezek igen nagy kapacitásúak. A hagyományos CD lemezeknek tehát nincs közük a mágneses adattároláshoz.

Napjainkban (1996 végén) már létezik a **DVD** (Digital Video Disc), amely a CD-hez hasonlóan lézerlemez, de lényegesen nagyobb kapacitású. A DVD 2 x 0,6 mm vastag, 120 vagy 80 mm átmérőjű, összeragasztott, szendvicsszerkezetű polikarbonát lemez. Érdekes, hogy a lemez nem csak két oldalról olvasható. Egy oldalról nézve a bitek két rétegben (síkban) helyezkedhetnek el (összesen négy rétegben). A következő táblázat összefoglalja a különböző méretű és rétegszámú lemezek tervezett kapacitását.

struktúra	egy oldalról olvasható		két oldalról olvasható	
	egyrétegű	kétrétegű	egyrétegű	kétrétegű
120 mm	4,7 Gbyte	8,5 Gbyte	9,4 Gbyte	17 Gbyte
80 mm	1,4 Gbyte	2,6 Gbyte	2,8 Gbyte	5,2 Gbyte

A DVD-ROM a multimédia-alkalmazások ideális adathordozója lesz.



## Ellenőrző kérdések

1. Írhatunk-e adatokat a hagyományos CD-ROM-ra?
2. Hány Mbyte kapacitású egy CD-ROM, és hány cm átmérőjű?



Mit tudsz a DVD-ről?

## EGYÉB HARDVERESZKÖZÖK

### MULTIMÉDIA-SZÁMÍTÓGÉP

Álló- és mozgóképet, grafikát, hangot, zenét, szöveget integráló programok futtatására képes számítógép. Természetesen a hangkeltéshez szükség van **hangszórókra** is, amelyeket egy illesztőkártyán, ún. **hangkártyán** keresztül vezérel a számítógép.

### STREAMER

Mágnesszalagos adattároló, hasonló a magnóhoz, adatok archiválására (biztonsági tárolására) használják.

### MODEM

Segítségével a számítógépek telefonvonalon keresztül kommunikálhatnak egymással. Legfontosabb jellemzője az **adatátviteli sebesség**, mértékegysége a bit/s = bps.

A következő könyvben bővebben is foglalkozunk a modemmel, és a vele elérhető világméretű hálózatokkal. Modem segítségével kapcsolódhatunk pl. az Internetre is.



## Ellenőrző kérdések

1. Mi a multimédia-számítógép?
2. Mire használható a modem?

## NÉHÁNY SZÓ A SZOFTVEREKRŐL

A számítógép működéséhez nem elég a hardver, szoftverre is szükség van. Mielőtt az operációs rendszerekkel foglalkoznánk, tekintsük át a szoftvertípusokat és a kezelésükre vonatkozó főbb szabályokat!

### **Operációs rendszer**

Felügyeli a hardver működését, biztosítja a felhasználó és a gép közötti kommunikációt, és megfelelő környezetet teremt más programok működéséhez. (Ilyen rendszer pl. az MS-DOS, az OS/2, a WINDOWS NT, a WINDOWS 95, az OS/2 WARP.)

### **Rendszerközeli szoftver**

Kényelmesebbé teszi az operációs rendszer, ill. a hardver használatát, de nem része az operációs rendszernek. (Ilyen szoftver pl. a Norton Commander (NC), a Dos Navigátor (DN) és a Windows 3.1.)

### **Fejlesztői rendszer**

Segíti a programozást, a programok fejlesztését, a programok fordítását és tesztelését. A programok általában az ember számára könnyebben érthető magas szintű programnyelven (pl. Pascal, C++) készülnek, nem a számítógép (a mikroprocesszor) által közvetlenül érthető bináris kódban.

### **Alkalmazói szoftver**

Segíti valamilyen felhasználói feladattípus megoldását. (Ilyen pl. a szövegszerkesztő, a táblázatkezelő, a grafikai program, az oktatóprogram és a játékok.)

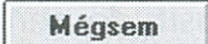
Egy-egy szoftver általában több állományból áll. Egy részük indítható fájl, más részük adatfájl. Általában van olyan fájl, amely a szoftver használatával kapcsolatos tudnivalókat tartalmazza: READ.ME, OLVASS.EL – ezeket az állományokat megfelelő paranccsal vagy szövegszerkesztő programmal olvashatod el, nézheted meg.

Ha a szoftvert Te használsz először a gépeden, akkor általában telepíteni (installálni, felrakni) kell. Ez egy, a szoftverhez tartozó ún. telepítőprogram elindí-

tásával (felrak.exe, setup.exe, install.exe) többnyire egyszerűen megoldható. (A telepítőprogram bemásolja a szoftver részeit a merevlemezre.)

Ha egy programot elindítottunk – pl. begéptük a nevét, és **Enter**-t ütöttünk, vagy pedig rákattintottunk (kétszer) az ikonjára –, akkor kommunikálni kell vele. Ezt megtehetjük **parancs üzemmódban** (begépt, ill. kiírt szövegekkel, esetleg beszéddel) vagy **menüvezérelt** módon. A menü mindig valamilyen véges számú választási lehetőséget kínál. Ha a menüpontok további menüsorokat tartalmaznak, akkor a menürendszert hierarchikus felépítésűnek nevezzük. A menürendszerek általában ilyenek. Napjainkban leggyakrabban olyan programokat írnak, amelyekben **ikonok, nyomógombok** vagy **képek** segítségével is lehet választani a lehetőségek közül.

A legtöbb programban van Súlyó (Help), amely elsősorban a használathoz nyújt segítséget. A ma használt programok többségének kezelőfelülete alig néhány típusba besorolható. Egy-egy típust nagyon hasonlóan kell kezelni. A felhasználók – érthető módon – nem szeretik, ha minden új szoftver használatát hosszasan kell tanulni.

Pl. sok programra igaz, hogy egy menüpontba eljuthatunk egérekattintással, vagy az **Alt +** (a menüpontban aláhúzott) betű leütésével. A menüpontok között a kurzormozgatókkal, vagy a **Tab** billentyűvel közlekedhetünk. A menüpontot a kijelölés után általában **Enterrel** vagy egérekattintással kell érvényesíteni ami lényegében egy parancs végrehajtását jelenti. Ha vissza akarod vonni a parancsot, általában megteheted az **Esc** billentyű megnyomásával, vagy ha a képernyőn  gombra kattintasz.

A kilépés többnyire az **Exit, Quit** menüponttal; **Alt+X, Ctrl+Q, Esc** billentyűkkel; **ajtó** ikonnal történhet.



### Ellenőrző kérdések

1. Hogyan csoportosítjuk a szoftvereket?
2. Mi a parancs üzemmód?
3. Mi a menüvezérlés?
4. Általában hogyan lépsz ki egy programból?

## AZ OPERÁCIÓS RENDSZEREKRŐL

A számítógépes rendszert – a memóriát, a háttértárat és más hardvereszközöket, valamint a programok működését – az operációs rendszer vezérli. Ez teszi lehetővé a hardver és a felhasználó (vagyis a gép és az ember) közötti kommunikációt; segítségével **parancsokat** adhatunk a gépnek és **üzeneteket** kapunk tőle. Az operációs rendszer vezérli a programok és adatok kezelését.



Az operációs rendszerek a gépek jobb használhatósága érdekében készültek. A személyi számítógépek elterjedésével egyre jobb, kényelmesebb (felhasználóbarát) rendszerek születtek, ill. a meglévők egyre fejlettebb változatai, verziói jelentek meg. Vannak olyanok, amelyek egyszerre több programot is képesek futtatni, vagy több felhasználót szolgálnak ki.

Az IBM PC eredeti operációs rendszere a **DOS** (Disk Operating System, vagyis lemezes operációs rendszer). Az első jelentősebb DOS-változatot (verziót) 1980-ban fejlesztette ki a **Microsoft** cég (MS-DOS). Azóta sok verzió jelent meg, amelyeket számokkal jelölnek, pl. MS-DOS 6.22. A DOS egy része – bekapcsolás után – lemezről töltődik be a gép memóriájába. Szöveges felhasználói felülettel rendelkezik, és parancs üzemmódban kommunikálhatunk vele. Egyszerre csak egy programot képes futtatni. Előnye, hogy lassabb processzor és kevesebb memória is megfelel a használatához. A felhasználónak igen pontosan kell begépelnie a parancsokat. Ezt megkönnyítendő az MS-DOS 4.0 verziótól kezdődően egy Shell program is része a rendszernek. Ennek az a célja, hogy a felhasználó menüvezérelt módon kezelhesse az operációs rendszert.

A gyorsabb processzorok és a nagyobb memória tette lehetővé a grafikus felületű operációs rendszerek kifejlesztését és elterjedését. Ez az operációs rendszer menüvezérelt, azaz nincs szükség a parancsok begépelésére. A grafikus felület ikonjai, ablakai segítségével áttekinthetőbb, kényelmesebb a program kezelése, és elég, ha csak nagy vonalakban tudjuk, hogy mit akarunk csinálni. Ilyen a Windows 95 és az OS/2 Warp.

A Windows 3.1 a DOS alatt indítható rendszer, amely 1992-ben jelent meg. Ablakos, ikonos, grafikus felületével kényelmes kezelést biztosít, de nem tekinthető önálló operációs rendszernek, működése a DOS-ra épül. A legtöbb helyen a magyar nyelvű változatát használják. Továbbfejlesztéseként 1995 augusztusában jelent meg a DOS-t már nem igénylő Windows '95 operációs rendszer.



### Ellenőrző kérdés

1. Miért használunk operációs rendszert?
2. Milyen rendszereket ismersz?

## AZ MS-DOS HASZNÁLATA

### A SZÁMÍTÓGÉP BEKAPCSOLÁSA

A számítógép bekapcsoláskor a ROM-ban lévő speciális programok „élesztik fel” a rendszert.

1. A gép először teszteli az áramköreit, pl. a memóriát, majd megvizsgálja, hogy melyik lemezről tudja beolvasni a memóriába a DOS ún. rendszerprog-

ramjait (az IO.SYS, MSDOS.SYS és a parancsértelmező COMMAND.COM programot). Azt a lemezt, amely ezeket a programokat tartalmazza és elindítható róla a DOS, **rendszerlemeznek** nevezzük.

A számítógép a beállítástól függően, általában először a hajlékonylemezt vizsgálja. Ha van lemez a meghajtóban, és nem találja rajta a rendszerprogramokat, akkor hibaüzenetet küld: (Non-system disk or disk error!). A lemezt ilyenkor ki kell venni, vagy a kallantyút felhajtani, és az **Enter-t** leütni!

2. A számítógép betölti: az IO.SYS, az MSDOS.SYS és a COMMAND.COM rendszerprogramokat. A COMMAND.COM a felhasználó parancsainak közvetlen értelmezője, végrehajtója. Az ún. **belső parancsokat** ez a program tartalmazza. A ritkábban használt, ún. **külső parancsokat** külön programok hajtják végre. Ezek csak akkor töltődnek a memóriába és lépnek működésbe, ha szükség van rájuk.

3. Utolsóként két speciális állomány beállításait és parancsait hajtja végre a számítógép. A CONFIG.SYS eszközmeghajtó programokat indít el. Az AUTOEXEC.BAT DOS-parancsokat tartalmaz, amelyek a rendszer indításakor végrehajtnak.

Az indítási folyamat végén a számítógép egy készenléti jelnél várja az utasításaidat.



### Ellenőrző kérdések

1. Milyen adathordozón van a DOS, és bekapcsolás után hová töltődik be?
2. Mi a feladata a COMMAND.COM állománynak?
3. Mit kell tenni, ha a gép nem találja a rendszerlemezt?

## AZ ELSŐ DOS-PARANCSONK

A DOS egy készenléti jellel, prompittal, (pl.: C:\>) jelentkezik be. A prompt mellett a kurzor villog, ide gépelheted be a parancsaidat. Gépeléskor kis- vagy nagybetűt egyaránt használhatsz.

Ha valamit rosszul gépelsz be, a **Backspace** billentyűvel törölheted.

**Minden begépelte parancs után üsd le az Enter billentyűt!**  
**Ekkor végrehajtnak a parancs.**

Ha egy parancsot a DOS nem tud végrehajtani, akkor általában a következő üzenetet kapod:

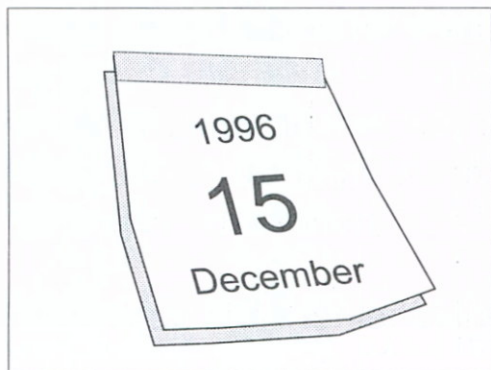
*Abort, Retry, Fail? Megszakítsuk, újrapróbáljuk, hagyjuk?*

Innen leggyakrabban az **F** billentyű leütésével tudsz továbbmenni, azaz elmarad a parancs végrehajtása.

Tekintsük át a DOS néhány parancsát! Minden parancsot célszerű kipróbálni, így könnyebben meg tudod jegyezni.

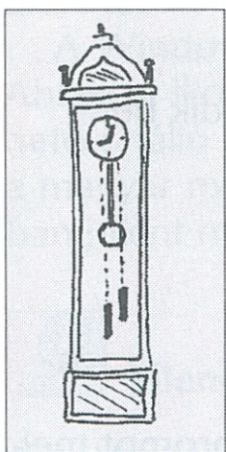
## DATE

A DOS kiírja a nyilvántartott dátumot az előre beállított formában (pl. év-hó-nap-nap), és lehetőséget ad a dátum megváltoztatására. Ha csak az **Enter** nyomod le, nem változik a dátum.



```
C:\>DATE
Current date is Sun 1980.03.09
Enter new date (yy-mm-dd): 1996.09.12
C:\>_
```

## TIME



Megjelenik az időpont: óra:perc:másodperc alakban, pl. 2:16:33,34, és – a dátumhoz hasonlóan – az idő is megváltoztatható.

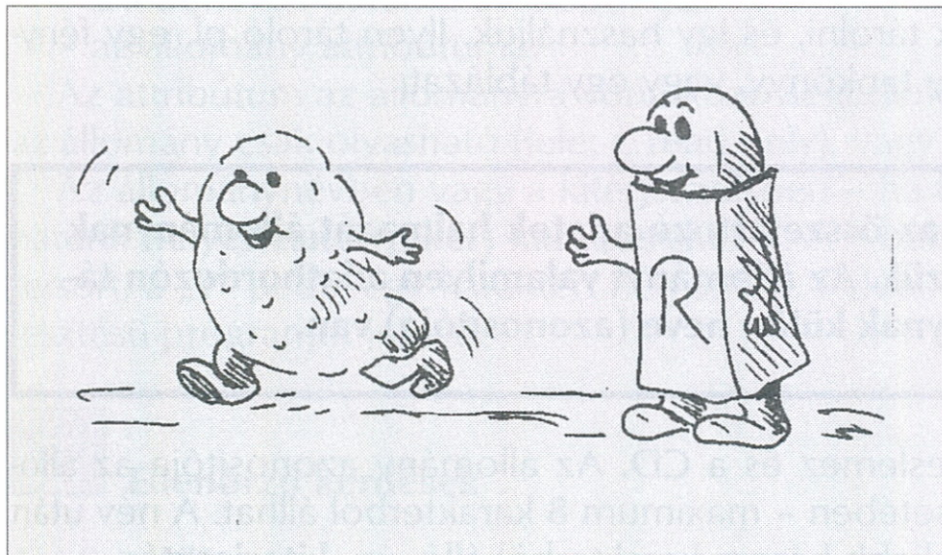
## PROMPT \$x

A DOS készenléti jelének alakját változtathatod meg, x helyére egy betűt kell írnod: a PROMPT \$p hatására kiíródik az aktuális könyvtár,  
a PROMPT \$d hatására az aktuális dátum,  
a PROMPT \$g hatására a > jel.

*(Ha nem találsz a \$ jelet a billentyűzeten, az **Alt** billentyűt nyomva, a jobb oldali numerikus billentyűkről begépelte 36-os kódszám segítségével is kiírathatjuk a monitorra!)*

## CLS

A képernyőt törölheted.



## VER

Megnézheted a DOS verziószámát, pl. *MS-DOS Version 6.22*.

Egy géphez általában több lemezegység is tartozik, amiket a DOS az ábécé betűivel jelöl.

**A hajlékonylemezek (floppy) meghajtóját A: vagy B:; a merevlemez tárolót (winchester-t) C: vagy D:; a CD meghajtót D: vagy E: jelöli.**

## Meghajtóváltás

A tanár utasítása szerint tégy egy hajlékonylemezt az A: vagy a B: meghajtóba! Írd be a megfelelő lemezegység jelét, és a DOS aktuális meghajtót vált.

A **C:** parancs hatására újra a C: jelű meghajtó lesz az aktuális.

A legtöbb parancs után **/betű** következhet. Ezzel pontosíthatod a parancsot. Ha egy parancs után **/?** áll, részletes tájékoztatást kaphatsz róla. Például: **VER/?**

A következő fejezetekben azzal foglalkozunk, hogy a programok milyen rendszerben helyezkednek el a háttértáron. Egy lemez tartalmát a **DIR** paranccsal nézheted meg.



### Ellenőrző kérdés

1. Sorold fel, hogy milyen DOS-parancsokat ismersz, és mi a jelentésük!
2. Hogyan töröld le a képernyőt?
3. Milyen lesz a prompt alakja a \$d hatására?

# ÁLLOMÁNYOK

Az információt hordozó szövegeket, adatokat általában összefüggő halmazokban, rendszerekben szoktuk tárolni, és így használjuk. Ilyen tároló pl. egy fénykép, egy füzet, egy levél, egy tankönyv, vagy egy táblázat.

**A számítástechnikában az összetartozó adatok halmazát állománynak vagy fájlnak (file) nevezzük. Az állományt valamilyen adathordozón tároljuk. Minden állománynak külön neve (azonosítója) van.**

Adathordozó pl. a mágneslemez és a CD. Az állomány azonosítója az állománynév, amely – a DOS esetében – maximum 8 karakterből állhat. A név után egy pont következik és legfeljebb három karakterből álló, ún. **kiterjesztés**.

Az állománynévben nem lehet használni a szöközt és a következő jeleket:

„ / \ < > + : = ; , .

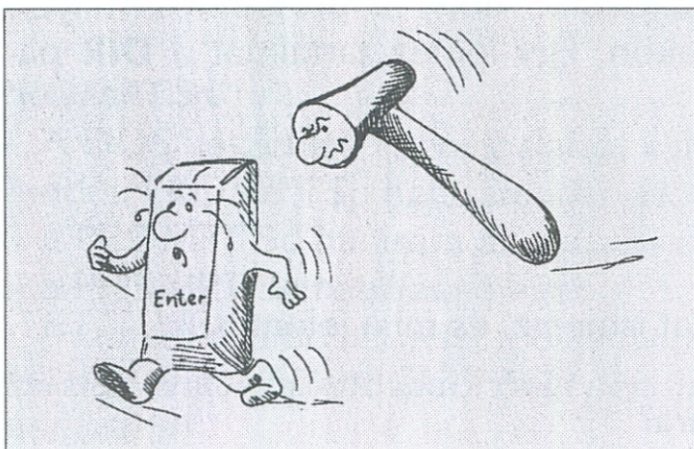
A kiterjesztés általában utal az állomány **formátumára**, fajtájára pl.:

- TXT, DOC, DOT szövegek;
- EXE, COM, BAT végrehajtható programok;
- XLS, DAT, DBF táblázatok, adatok;
- BMP, TIF, GIF képek;
- WAV, MID hangok.

A különböző felhasználói programok általában eltérő módon tárolják a velük létrehozott adatokat, állományokat, ezért léteznek különböző kiterjesztések. Ha egy fájlunk (később) mi adunk nevet, a tartalomnak és a kiterjesztésnek meg kell felelnie egymásnak. Ha pl. egy képfájl nevét exe kiterjesztéssel látjuk el, az még nem lesz működő (futtatható) program.

## **Emlékeztető**

*Egy programot a fájlnev begépelésével és az **Enter** billentyű leütésével indíthatasz el. A pontot és a kiterjesztést nem kell leírnod.*



Egy fájlról a nevéen kívül a DOS még a következőket tárolja:

- a fájl mérete, azaz adatmennyisége byte-ban,
- a létrehozás dátuma és időpontja,
- az állomány attribútuma.

Az attribútum az állományra vonatkozó segédinformációt tartalmazza: pl., ha az állomány csak olvasható (jele: r, read only), vagy ha rejtett fájl (h, hidden).

Az állománynévben vagy a kiterjesztésben – ha hivatkozunk rájuk – használhatunk **helyettesítő (joker) karaktereket** is. A \* bármilyen karaktert vagy karaktersort, a „?” pedig egy karaktert helyettesít. Például a \*.exe az összes exe kiterjesztésű programot jelenti.

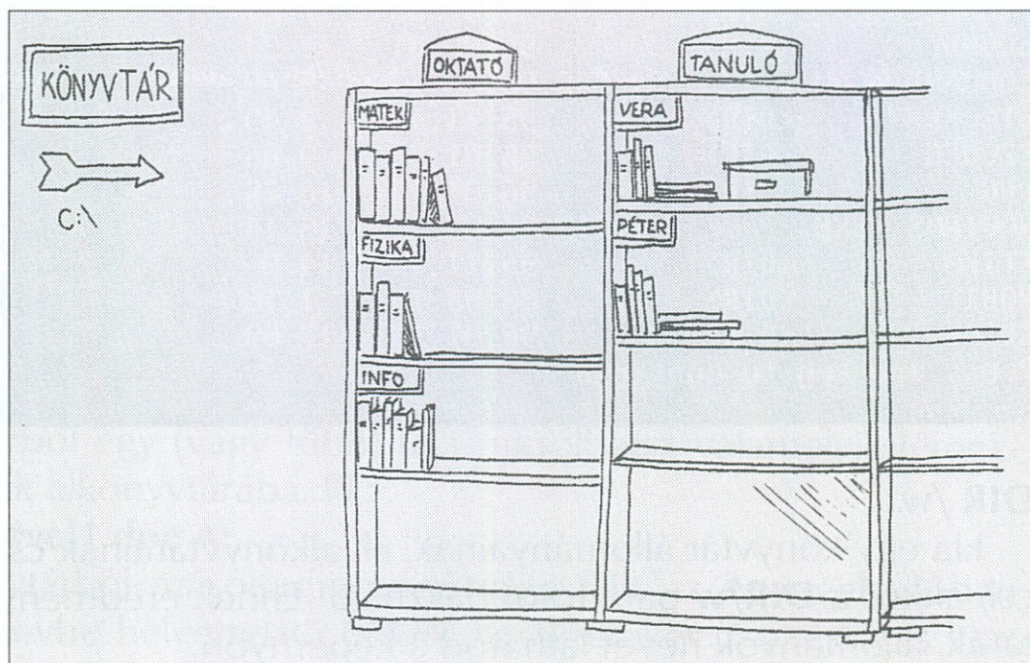
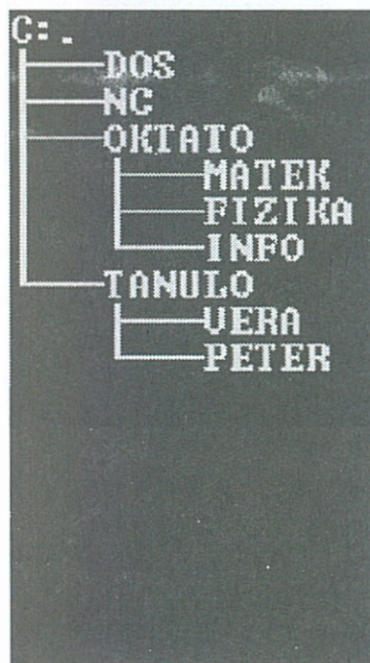


## Ellenőrző kérdések

1. Mi a fájl?
2. Legfeljebb hány betűből állhat az állománynév?
3. Milyen karaktereket tilos használni?
4. Milyen jellegű adathalmaz egy bmp kiterjesztésű állomány?

## KÖNYVTÁRSZERKEZET

A DOS a fájlok sokaságát **főkönyvtárban** és **alkönyvtárakban** tárolja a lemezen. Minden lemeznek van egy főkönyvtára, amit fordított törtvonallal (\) jelölünk. Ebben lehetnek alkönyvtárak is és állományok is. Minden alkönyvtárban lehetnek további alkönyvtárak és állományok. A főkönyvtár és az alkönyvtárak egy fához hasonló hierarchikus szerkezetet alkotnak, ahol a főkönyvtár a törzs, az alkönyvtárak az ágak, és a különböző fájlok a levelek.



Te is megnézheted a könyvtárszerkezetet a következő DOS-paranccsal.

## TREE

E parancs segítségével kiírathatod a képernyőre a könyvtárszerkezetet.

## KÖNYVTÁRKEZELÉS

### DIR

Egy adott könyvtár állományait és alkönyvtárait nézheted meg a **DIR** parancs segítségével a következő formában:

*név kiterjesztés adatmennyiség dátum idő*

Alkönyvtárak esetében a név után a <DIR> jelölés áll.

### DIR /P

A könyvtár vagy az alkönyvtár tartalmából annyit látsz, amennyi elfér egy képernyőn. Ha bármelyik billentyűt leütöd, megnézheted a további részeket.

```
Volume in drive C is POWER
Volume Serial Number is 1139-1504
Directory of C:\

DOS             <DIR>           95.06.01    14.21
WINDOWS        <DIR>           95.11.10    12.46
DN              <DIR>           95.06.01    14.49
VB             <DIR>           95.11.10    13.30
UTIL           <DIR>           95.06.01    14.50
ANTIVIR        <DIR>           95.06.01    14.59
EGER           <DIR>           95.06.01    16.40
GRAPHIC        <DIR>           95.06.01    16.41
CDROM          <DIR>           96.07.04     7.45
PICDIC_E       <DIR>           96.07.03    17.21
SOUND          <DIR>           95.06.01    17.34
COMMAND  COM      54 645 96.06.13     5.09
STACKER        <DIR>           95.11.13    12.49
SZOVEG         <DIR>           96.01.22     9.06
PRG            <DIR>           96.01.13    16.28
NOVIRUS  EXE        5 312 94.03.17    16.14
ETIKA          <DIR>           95.11.13    14.06
ROBI           <DIR>           80.02.16     1.16
CDTAN          <DIR>           96.01.15    17.09
Press any key to continue . . .
```

### DIR /w

Ha egy könyvtár állományainak, ill. alkönyvtárainak csak a nevére vagy kíváncsi, akkor a **DIR/w** parancsot használd! Ennek eredményeként csak az alkönyvtárak, állományok nevét láthatod a képernyőn.

## CD alkönyvtárnév

A könyvtárváltás parancsa. Ha pl. a TANULO alkönyvtárba akarsz belépni a főkönyvtárból (C:\), akkor a **CD TANULO** paranccsal teheted meg.

A **CD..** paranccsal egy szinttel feljebb, míg a **CD\** paranccsal rögtön a főkönyvtárba ugorhatsz. Ha a főkönyvtárból (C:\) több szinttel lejjebb akarsz ugrani, ezt is megteheted, ha megadod az **elérési utat**: pl. **CD TANULO\VERA**. Ezzel a paranccsal a főkönyvtár TANULO alkönyvtárának VERA alkönyvtárába léphetsz be.

## MD alkönyvtárnév

Az alkönyvtár-létrehozás parancsa. Ha pl. a TANULO alkönyvtárban vagy, és készíteni akarsz itt egy PROBA nevű alkönyvtárat, az **MD PROBA** parancsot gépd be.

```
C:\TANULO> MD PROBA
```

## RD alkönyvtárnév

Ezzel a paranccsal törölheted az alkönyvtárat, **ha üres**. Ha pl. ha a TANULO könyvtárban vagy, akkor az **RD PROBA** törli az ott található PROBA alkönyvtárt.



## Feladatok

1. Készíts a főkönyvtárban egy TANULO alkönyvtárt!
2. Lépj be a TANULO-ba, és készíts benne egy VERA, egy PETI és egy ADRI alkönyvtárt!
3. Hogyan tudsz a TANULO könyvtárba belépni a C:\TANULO\PETI alkönyvtárból?
4. Nézd meg (a DIR paranccsal), hogy a számítógép milyen nevű és kiterjesztésű állományokat tárol!

## FÁJLKEZELÉS

Az állományok kezelésében a két legfontosabb művelet a másolás és a törlés. Ezeket tekintjük most át.

### COPY mit hová

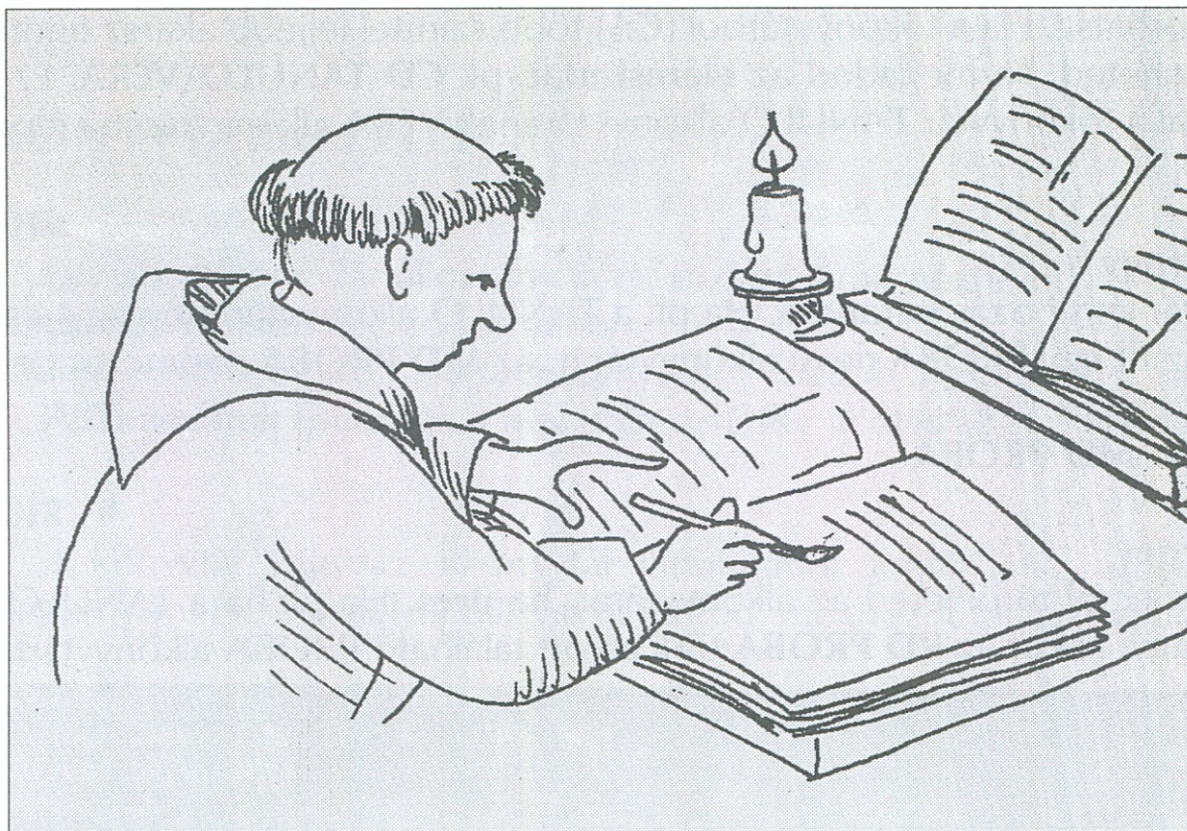
Az aktuális alkönyvtárból egy (vagy több) fájlt másolhatsz valamelyik lemezre, vagy annak valamelyik alkönyvtárába. Pl.:

```
C:\LEVELEK> COPY level1.doc A:
```

Itt a LEVELEK alkönyvtárból másoltad a level1.doc fájlt az A: meghajtóban lévő lemezre. Ha a fájlnevbe helyettesítő (joker) karaktereket (\* vagy ?) is írsz,



egyszerre több állományt is átmásolhatsz. Pl. a level1.doc helyett \*.doc-ot írva, minden doc kiterjesztésű fájlt átmásolsz a LEVELEK könyvtárból a hajlékonylemezre.



### DEL fájlnev

Az aktuális könyvtárból egy vagy több fájlt törölhetsz a **DEL** segítségével.

Pl.: C:\MONDATOK> **DEL level2.doc**

Ez esetben a MONDATOK alkönyvtárból törölted a level2.doc fájlt.

*Néha meggondolatlanul törölünk, és szeretnénk visszaállítani az eredeti állapotot. Erre is van lehetőség az **UNDELETE fájlnev** paranccsal.*

### TYPE fájlnev

Ezzel a paranccsal az aktuális könyvtárban lévő, megadott fájl tartalmát nézheted meg.

Pl.: C:\MONDATOK>TYPE olvass.el

Kiírja az olvass.el tartalmát, amit el is olvashatsz.



### Feladat

Lépj be egy gyakorlásra használható alkönyvtárba, és másolj át onnan egy tetszőleges fájlt a C:\TANULO-ba! Menj át a TANULO alkönyvtárba, és töröld ki az előbb átmásolt állományt!

# FORMÁZÁS

## FORMAT meghajtó jele

A meghajtó jele lehet: **A:**, **B:**, **C:** \_

Ahhoz, hogy egy új lemezt használni tudj, előbb formáznod kell. Az operációs rendszer információkat ír a lemezre, amelyek meghatározzák, hogy milyen szabályok szerint és hová kerüljenek majd az adatok, fájlok.



*A merevlemez formázásakor a lemezen levő összes adat elveszik, ezért ne a merevlemezen próbáld ki a formázást! Légy e téren körültekintő!*

## FORMAT A: /S

Erre a parancsra a DOS rendszerlemezt készít, azaz átmásolja a lemezre az operációs rendszer egy részét. A rendszerlemezről indítható el a számítógép.

## FORMAT A: /F: a lemez kívánt kapacitása

Beállíthatod, hogy a formázás hatására mekkora legyen a lemez tárolókapacitása. Ha az F: után 1.44-et írsz, akkor az 1,44 Mbyte, ha 720-at, akkor 720 Kbyte lesz a tárolókapacitás.

## FORMAT A: /U

Ezzel a paranccsal úgy formázhatod a lemezt, hogy ha volt a lemezen információ, nem lehet többé visszaállítani.

*Készíts Te is rendszerlemezt! Ha merevlemezről nem indul a gép, külső lemeztől kell indítanod. A merevlemez ilyenkor többnyire javítható, tartalma visszaállítható. Ha vírusos a géped, akkor is külső lemeztől célszerű indítanod.*



## Gyakorlat

Tegyéél egy üres lemezt az A: meghajtóba, és gépedbe a következő parancsot!

FORMAT A:

Az

*Insert new diskette for drive A: and press ENTER when ready...*

figyelmeztetésre újra üss **Enter**-t! A gép %-ban jelzi ki, hogy hol tart a folyamat. A

*Volume label...*

kérdésre írd be a **lemez nevét**, majd nyomd le az **Enter**-t!

```
C:\NC>FORMAT A:
Insert new diskette for drive A:
and press ENTER when ready...

Checking existing disk format.
Saving UNFORMAT information.
Verifying 1.44M
Format complete.

Volume label (11 characters, ENTER for none)? TANFOLYAM

    1 457 664 bytes total disk space
    1 457 664 bytes available on disk

        512 bytes in each allocation unit.
        2 847 allocation units available on disk.

Volume Serial Number is 3A39-0FDE

Format another (Y/N)?N

C:\NC>_
```

A gép többek között kiírja a teljes lemezkapacitást (*total disk space*) és a szabad lemezterületet (*available on disk*). Végül, a

*Format another (Y/N)?*

kérdésre válaszolj N betűvel, ha nem akarsz újabb lemezt formázni! (Az UNFORMAT paranccsal visszaállítható a lemez formázás előtti tartalma.)

# RENDSZERKÖZELI PROGRAMOK

Ahhoz, hogy a DOS lehetőségeit kihasználhasd, nagyon sok parancsot, ill. szabályt kellene megtanulnod és begyakorolnod, és ha mindezt már tudnád, akkor is nehéz lenne az utasításokat begépelni. Így érezte ezt **Peter Norton** amerikai elektromérnök (szerzetes és később PC specialista) is. Ezért kifejlesztett egy olyan programot, a **Norton Commandert**, amely jelentősen megkönnyítette a DOS-szal való kommunikációt. Ezen kívül is sok olyan program van, amely a nevét viseli, pl. a **Norton Disk Doktor**.

A következőkben olyan rendszerközeli programokról lesz szó, amelyek megkönnyítik a géppel (és a DOS-szal) való kommunikációt: a gyakori fájlkezeléssel, könyvtárkezeléssel kapcsolatos feladatok elvégzését, esetleg olyan eszközöket is biztosítanak (jegyzet, kalkulátor, szövegszerkesztő, rajzoló-festő, naptár, óra), amelyek a munkához gyakran szükségesek.

```
MS-DOS Shell
File Options View Tree Help
C:\DOS
[A:] [B:] [C:]

Directory Tree
[-] C:\
  [+ ] AAPLAYHI
  [+ ] ANTI VIR
  [+ ] CDROM
  [- ] CDTAN
  [+ ] CONVERT
  [+ ] COREL50
  [+ ] DAVE
  [+ ] DEMOS
  [- ] DN
  -> [- ] DOS
  [- ] EGER
  [+ ] ET IKA
  [+ ] EXCEL
  [+ ] FIZIKA
  [+ ] GAMES
  [+ ] GRAPHIC

C:\DOS\*. *
▶ ANSI .SYS 9,065 94.05.31
APPEND .EXE 10,774 94.05.31
ATTRIB .EXE 11,208 94.05.31
B4B5DOS .BAT 8 80.01.17
B430DOS .BAT 8 80.01.17
B871DOS .BAT 8 80.01.17
CHKDSK .EXE 12,241 94.05.31
CHKSTATE .SYS 41,600 94.05.31
CHOICE .COM 1,754 94.05.31
COUNTRY .SYS 26,936 94.05.31
COUNTRY .TXT 13,091 94.05.31
DBLSPACE .BIN 57,510 94.02.13
DBLWIN .HLP 8,444 94.05.31
DEBUG .EXE 15,718 94.05.31
DEFRAG .EXE 79,065 94.05.31
DEFRAG .HLP 9,227 94.05.31
DELOLDOS .EXE 17,726 94.05.31

F10=Actions Shift+F9=Command Prompt 07.55
```

Ilyen rendszerközeli program az **MS-DOS Shell**, amit az 5.0-ás DOS-szal már együtt adtak. Karakteres vagy grafikus felhasználói felülete, képernyője van, amely megkönnyíti a programok indítását, a lemezmeghajtók, a könyvtárszerkezetek és a fájlok kezelését (formázás, törlés, átnevezés, másolás, mozgatás, megjelenítés, nyomtatás, keresés, fájlattribútum-beállítás).

A programot a DOSSHELL-paranccsal indíthatjuk és egérrel is kezelhető.

Ezekon kívül is sok és sokféle szoftvereszköz készült a DOS operációs rendszerhez, amelyek megkönnyítik, kényelmessé teszik a munkát. Ilyen pl. a **Volkov Commander** vagy a **DOS Navigator**, amely hasonlít a **Norton Commander**-hez és egy kicsit az MS-DOS Shell-hez is, a panelek méretezhetők és mozgathatók, mint a Windows ablakai.

A következőkben a Norton Commanderrel ismerkedünk meg, amely olyan sikeres program volt, hogy hasonló felépítésű, ill. felhasználói felületű programok más operációs rendszerekhez is készültek, pl. a Windows '95-höz is. Így az erről tanultaknak máshol is hasznát veszed!



### Ellenőrző kérdés

1. Milyen rendszerközeli programokat ismersz, és mire használhatók?

## NORTON COMMANDER

Jellemző a Norton Commanderre – mint sok más szoftverre is –, hogy megjelenése a felhasználó saját igényei szerint állítható, szabályozható. Mivel az összes funkció áttekintésére nincs lehetőség, egy gyakran használt alapbeállításból indulunk ki.

Az elindításhoz lépj be a Norton Commandert tartalmazó alkönyvtárba (pl. CD NC), és gépeled be a program nevét (NC)!

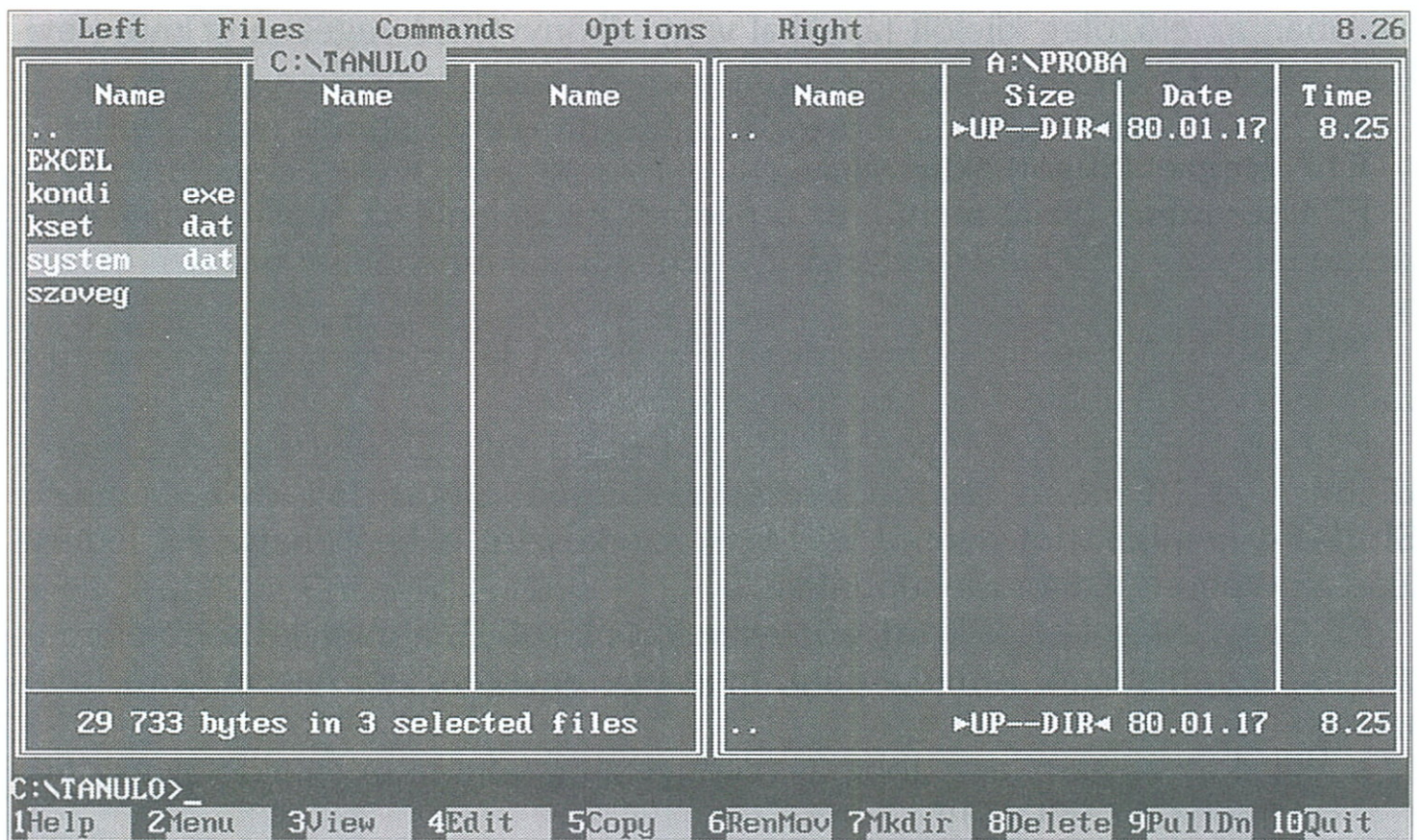
A program indítása után két **panel** tölti be a monitort. Alul van egy sor a DOS-parancsok esetleges begépelésére, itt látható a már jól ismert készenléti jel.

## A PANELEK

A panelek felső szélén a meghajtó jele, és az éppen mutatott alkönyvtár neve látható.

Az a panel aktív, amelyiken a meghajtó jele ki van jelölve, vagyis különböző színű, mint a háttér. Az aktív panelen egy mutatósáv is látható.

A mutatósávot a kurzormozgató billentyűkkel (nyilak, **Home**, **End**, ...) vagy az egérmutató és egy kattintás segítségével lehet a kívánt sorra állítani.



Az egyik panelről a másikra a **Tabulátor** billentyűvel ugorhatsz.

A panelen mutatott alkönyvtárak NAGYBETŰVEL, a fájlok kisbetűvel vannak kiírva.

Ha egy alkönyvtáron állsz, és leütöd az **Enter**-t, akkor belépsz az alkönyvtárba; ennek tartalma fog megjelenni a panelen. Ilyenkor az első sorban két pont (..) látható, itt léphetsz ki **Enter**-rel az alkönyvtárból, vissza az előző szintre. Ha egy **indítható fájl**on (exe, com vagy bat kiterjesztésű) áll a mutató, és leütöd az **Enter**-t (vagy az egérrel kétszer rákattintasz), akkor a program elindul.

A Norton Commander akkor is a memóriában marad, ha más programot indítasz el belőle. Ilyenkor a képernyőről eltűnik, helyet adva a programnak, de újra előjön, mihelyt kilépsz a programból.

Ha meghajtót akarsz váltani, az **Alt + F1** (bal panel) vagy **Alt + F2** (jobb panel) billentyűkombinációt alkalmazd! Ekkor a megjelenő kis ablakban, a kurzormozgató nyilak segítségével ráállhatsz a kívánt meghajtóra (A: B: C: D:...), majd **Enter**-rel érvényesítheted a parancsot. Ne felejts el lemezt tenni a floppy meghajtókba!

## A FUNKCIÓBILLENTYŰK

A képernyő alján egy menüsor van, itt a funkcióbillentyűknek (**F1, F2, F3** stb.) megfelelő számok láthatók és rövid utalások a kiváltható funkciókra. (Ha nem lenne itt, akkor a **Ctrl+B** kombinációval bekapcsolható.) A funkcióbillentyűkkel

általában az előzőleg kijelölt fájlokkal vagy alkönyvtárral végezhetsz műveleteket. Vegyük sorra őket!

**F1 A Help-et** (Súgót) aktivizálja.

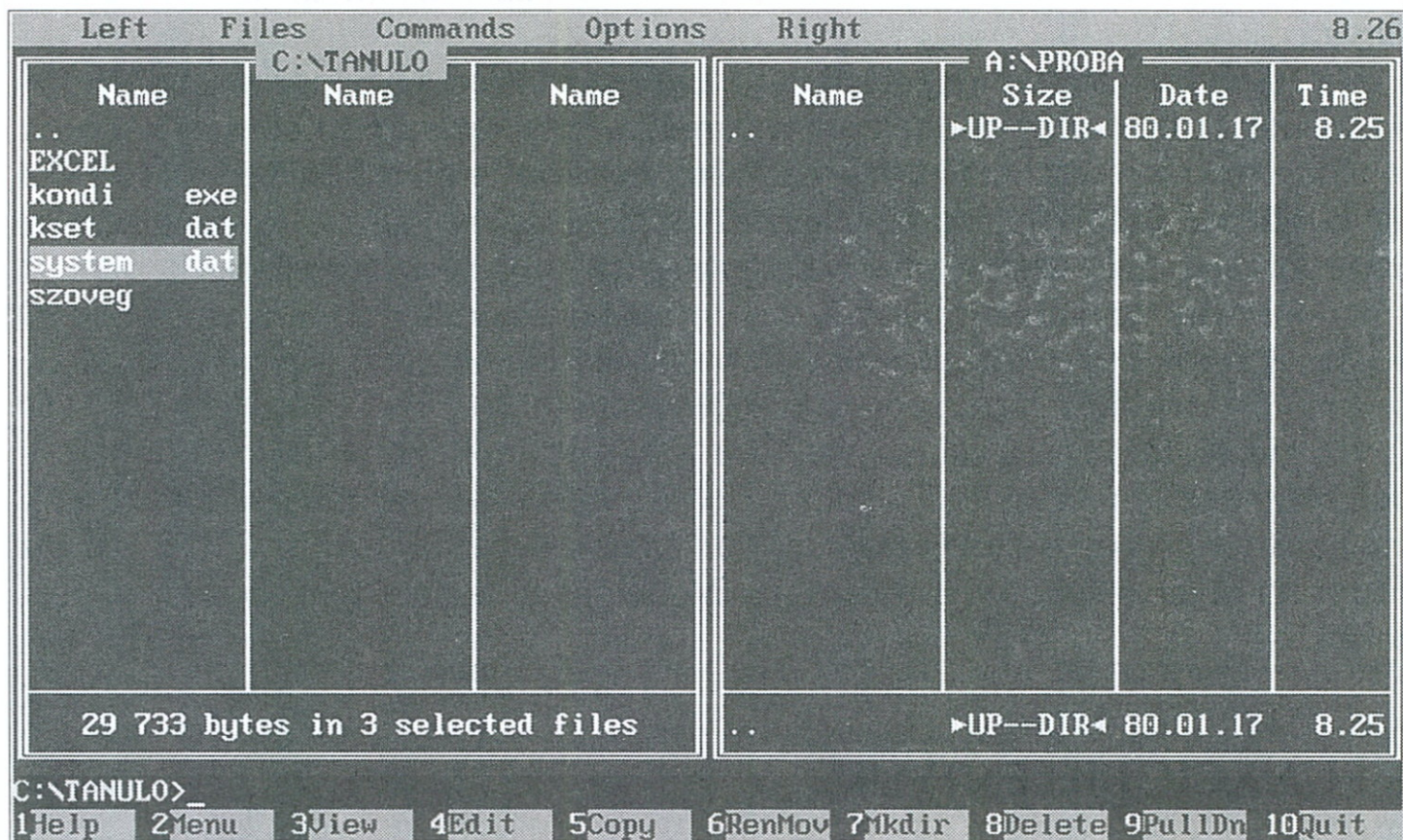
**F2 User menu** (Saját menü), ez a parancs megjeleníti az általad definiált Saját menüt, ezt az **F9** leütése után a Commands menüben lehet megszerkeszteni.

**F3 View** (Megjelenítés), megmutatja a kijelölt fájl tartalmát, pl. egy képet vagy szöveget. Többféle fájlformátumot felismer.

**F4 Edit** (Szerkesztés), betölti a Norton Commander szövegszerkesztőjébe a kijelölt fájlt. Itt módosíthatod, átszerkesztheted. Egy új fájl készítéséhez a **Shift+F4** kombinációt nyomd le! Másik szerkesztőt is beállíthatsz **F9** leütése után az Options, Editor menüponttal.

**F5 Copy** (Másolás), fájlokat, könyvtárakat másol. Ez a művelet többféle módon is végrehajtható, ismerkedjünk meg egy egyszerű, jól használható módszerrel!

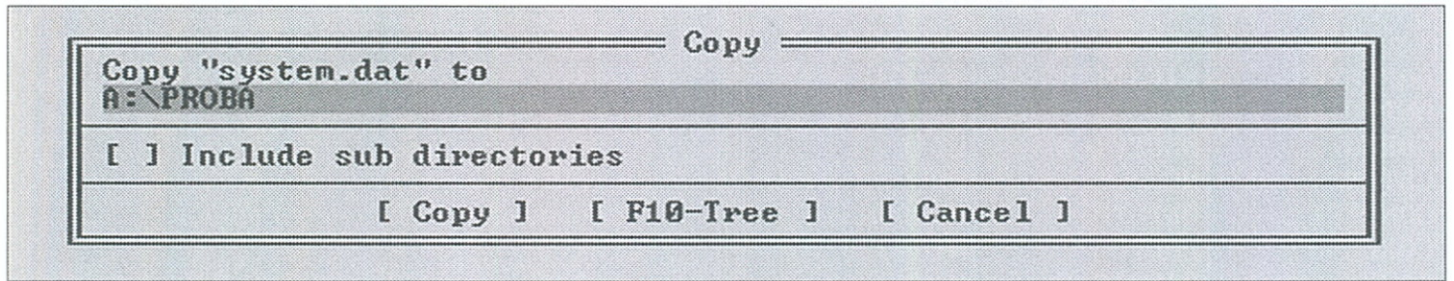
- Állítsd az egyik panelt abba az alkönyvtárba, **ahonnan** fájlokat akarsz másolni (pl. C:\TANULO)!
- Állítsd a másik panelt arra a meghajtóra, és abba az alkönyvtárba, **ahová** másolni szeretnél (pl. A:\PROBA)!



- Ugorj vissza az első panelre, és az **Insert** billentyűvel (vagy ha minden fájlt másolni akarsz, akkor a csillaggal) jelöld ki a másolni kívánt fájlokat!

- Nyomd le az **F5** gombot!

A megjelenő párbeszédablakban még módosíthatod a célkönyvtárat (vagy **F10** leütése után a megjelenő könyvtárrendszerből új helyet választhatsz), bekapcsolhatod az alkönyvtármásolást vagy visszaléphetsz.



- Nyomd le az **Enter** billentyűt!

Ezzel a módszerrel alkönyvtárakat is másolhatsz, állományokkal együtt (4.0 vagy annál újabb verziókban).



**F6 Rename or move** (Átnevezés/áthelyezés), a kijelölt fájl egy párbeszédablakban átnevezhető. Az áthelyezés a másoláshoz hasonlóan történik, csak a régi helyről kitörlődik a fájl.

**F7 Make directory** (Könyvtárlétrehozás), új alkönyvtárt hozhatsz létre az aktuális könyvtárban, ha a megjelenő párbeszédablakban begépeled az új könyvtár nevét. (Más meghajtón vagy könyvtárban is nyithatsz újat, ha megadod az elérési utat, pl. C:\TANULO\PROBA.)



**F8 Delete** (Törlés), az Insert gombbal kijelölt fájlokat vagy könyvtárat törölheted vele, a benne lévő állományokkal együtt.

**F9 PullDn** (Legördülőmenük), hatására a panel fölött megjelenik egy menüsor, ha nem volt bekapcsolva:

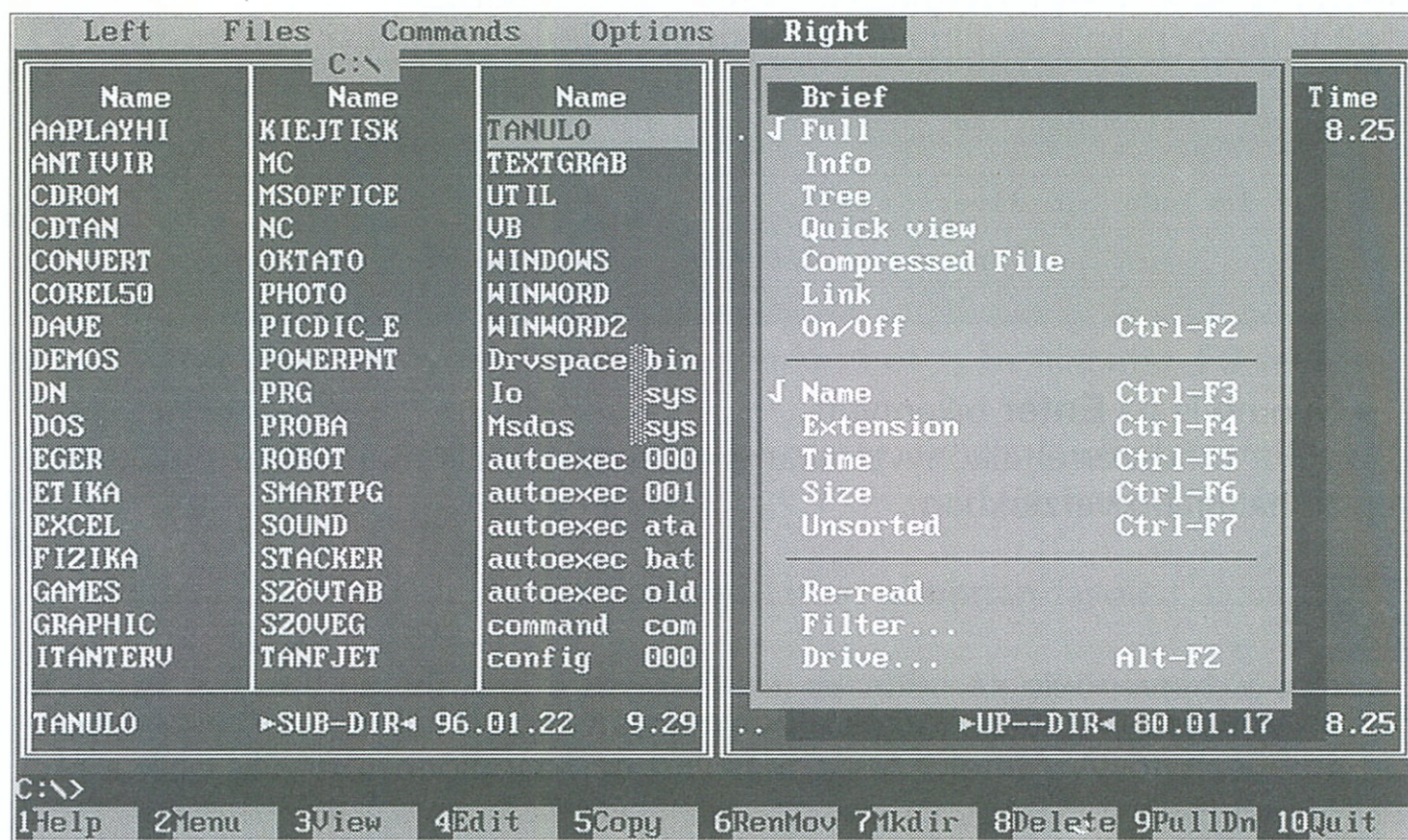
Left	Files	Commands	Options	Right
Bal	Fájlok	Parancsok	Beállítási lehetőségek	Jobb

Valamelyik menüpontra ráállva és **Enter**-t lenyomva, egy-egy legördülőmenü jelenik meg. Tekintsük át ezeket röviden, kiemelve néhány módosítási lehetőséget!

A **Left** és a **Right** a két panelre vonatkozik, itt állíthatod be, hogy milyen formában jelenjenek meg az adatok: pl. *Brief*: rövidített; *Full*: teljes megjelenítés;



*Info*: információk; *Tree*: fastruktúra; *On/Off*: Panel Ki/Be. Meghatározhatod, hogy a fájlok név, kiterjesztés, idő vagy méret szerint következzenek-e egymás után.



A **Files** menü tartalmazza azokat a feladatokat, amiket a funkciógombokkal is végre lehet hajtani (pl.: másolás, törlés, átnevezés).

A **Commands** menü néhány menüpontja: *NCD Tree*: megtekintheted az aktuális meghajtó könyvtárainak fastruktúráját, lehetőséged nyílik a gyors könyvtárcserére; *Find file*: fájlokat kereshetsz; *Compare directories*: összehasonlíthatod a két panel tartalmát; *Menu File Edit*: itt szerkesztheted a saját felhasználói menüdet (a szerkesztés során használd az **Insert** és a **Delete** gombokat is).

Az **Options** menüben elsősorban a rendszer működési paramétereit, megjelenését, ill. „viselkedését” állíthatod be, pl. *Configuration*: számos paramétert módosíthatsz a szóköz billentyűvel; *Editor*: beállíthatod a rendszer szövegszerkesztőjét; *Confirmation*: eldöntheted, hogy legyenek-e a fájlműveletek során ellenőrző kérdések; *Mini status*: ki/be kapcsolhatod a panel alján lévő Állapotsort.

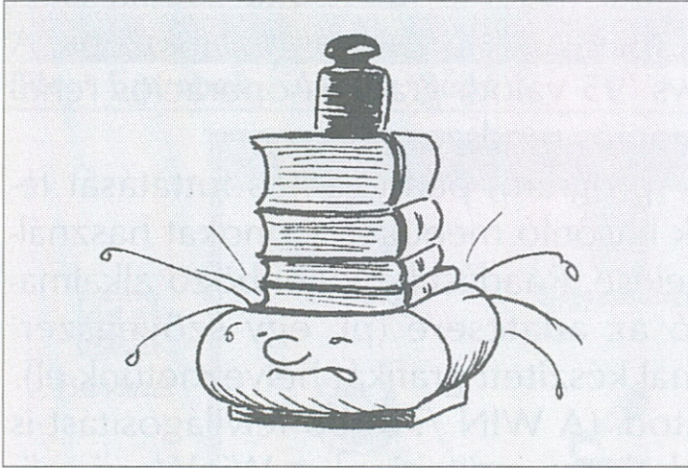
**F10 Quit** menüponttal kiléphetsz a Norton Commanderből.

Végül, érdemes megjegyezni, hogy az **Alt** vagy a **Ctrl** billentyűk lenyomásakor az alsó menüsor átvált, újabb menüpontok jelennek meg. Ezek közül kettőt emelnénk ki:

- az **Alt + F5** kombinációval fájlokat tömöríthetsz,
- az **Alt + F6**-tal pedig kicsomagolhatod őket.

## Emlékeztető

A tömörítés azt jelenti, hogy csökkentjük az adathalmaz (fájl) adatmennyiségét, anélkül, hogy veszítenénk az információból. Ez megfelelő kódolással érhető el, amit valamilyen tömörítőprogram végez. Kicsomagolás (dekódolás) után visszkapjuk az eredeti fájlt.



Ezzel be is fejeztük a Norton Commander áttekintését.

A rendszerközeli programokhoz sorolható a **Windows 3.1**, amely megjelenésében és szolgáltatásaiban más szemléletet hozott, bár a személyi számítógépek világában nem elsőként.



## Ellenőrző kérdések

1. Hasonlítsd össze a DOS és a Norton Commander könyvtár- és fájlkezelését!
2. Hogyan végzed el a
  - meghajtóváltást,
  - az egyik panelről a másikra ugrást,
  - a belépést egy alkönyvtárba,
  - a kilépést egy alkönyvtárból,
  - a programindítást, valamint
  - az alsó menüsor bekapcsolását?
2. Magyarázd el a funkciógombok (**F1, F2, F3 ...**) jelentését!
3. Hogyan másolnád át az összes állományt az a: meghajtóról a c:\jatek\verseny alkönyvtárba?
5. Mit tennél, ha az előző feladatban nem másolás, hanem áthelyezés szerepelne?
6. Készíts egy alkönyvtárrendszert fontosabb számítógépes feladataidnak (pl. a SAJAT alkönyvtáron belül van TANTARGY, JATEKOK, SZOVEGEK, RAJZOK alkönyvtár...)!
7. Mire használhatod az **F9** billentyűt?
8. Hogyan tömöríthetsz állományokat a Norton Commander segítségével?

## WINDOWS 3.1

A Windows (magyarul ablakok) a Microsoft cég terméke, grafikus felhasználói felületet biztosít az operációs rendszer kezeléséhez. A Windows 3.1x vagy az ennél régebbi verziók elindításához szükség van a DOS-ra, ezért még nem tekinthetők valódi operációs rendszereknek, bár hasonló funkciókat látnak el. A 3.1 verzió 1992-ben jelent meg, amelynek magyar változatával foglalkozunk. A Windows család fiatalabb tagjai a Windows '95 valódi grafikus operációs rendszer, valamint a Windows NT hálózati operációs rendszer.

A Windows egyszerre több alkalmazás (program) párhuzamos futtatását teszi lehetővé, hasonló ablakokban, amelyek hasonló menüket, ikonokat használnak, hasonló a billentyűzet és az egér kezelése. Ráadásul a különböző alkalmazások között egyszerűen megvalósítható az adatcsere (pl. egy szövegszerkesztővel írt meghívóban rajzolóprogrammal készített grafikát helyezhetünk el).

A Windows-t a WIN paranccsal indíthatod. (A WIN /?-re bő felvilágosítást is ad.) Ha rögtön egy másik alkalmazást is el akarsz indítani, pl. a WinWord szövegszerkesztőt, a WIN WINWORD parancs begépelésével teheted meg.

### A Windows rendszerben gyakran használt alapfogalmak:

- **Program, alkalmazás:** meghatározott feladatok elvégzésére (pl. szövegszerkesztésre, rajzolásra, számolásra, táblázatkezelésre) szolgáló program, ill. szoftver.
- **Programcsoport:** programok, alkalmazások egy halmaza.
- **Futó program, futó alkalmazás:** olyan program, amelyből nem léptünk ki.
- **Dokumentum:** valamely alkalmazással létrehozott munka (pl. szövegszerkesztővel írt levél).

Az eddig felsorolt fogalmakat objektumoknak is nevezzük.

- **Ikon:** valamilyen objektumot jelölő kis kép.
- **Ablak:** egy körülhatárolt, téglalap alakú tartomány a képernyőn. Kevés kivételtől eltekintve minden ablak mozgatható, méretezhető, ikonná zárható.

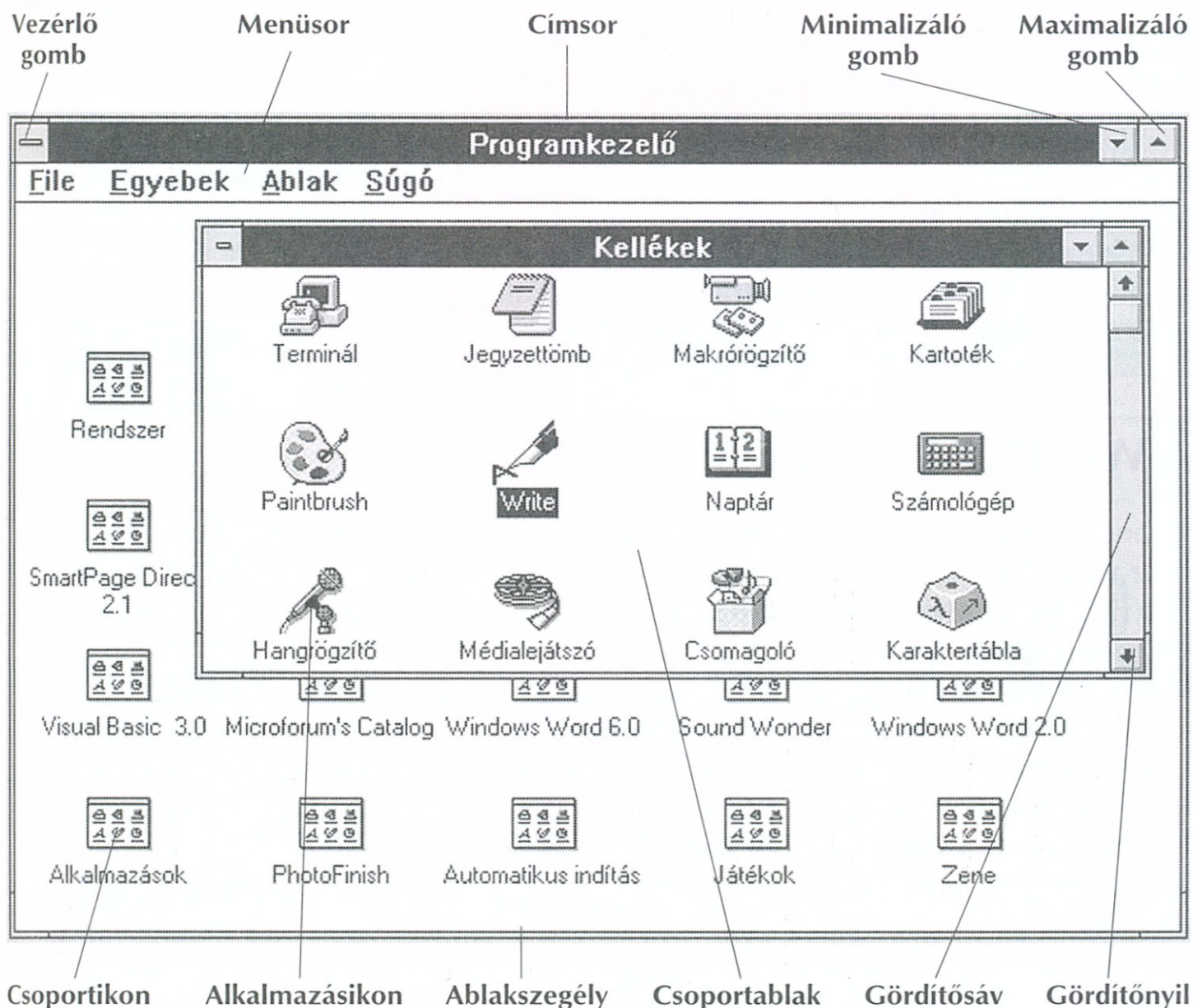
Az alkalmazásokat úgy indíthatod el, hogy a nekik megfelelő ikonra duplán kattintasz. Ha elindítasz egy alkalmazást, megnyílik egy ablak, ebben fut az alkalmazás. Ha a vezérlőgombra duplán kattintasz, bezárhatod a programot, ill. kiléphetsz. Ha az alkalmazásból kilépsz, akkor az **ablakot is ikonná zárod**.



Többféle ablak van a Windowsban. A **programcsoportablakban** az odatartozó programok ikonjai vannak, a **dokumentumablakban** pedig egy program, pl. a szövegszerkesztő dokumentuma.

Némileg eltér az eddigi ablakoktól a **párbeszédablak**, amely kommunikációra, adatok közlésére és megadására szolgál, és általában valamely menüpont kiválasztása után jelenik meg.

A képernyőn egyszerre több ablak is lehet. A pillanatnyilag használt ablakot **aktuálisnak** vagy aktívnek nevezzük, ennek a címsora más színű.

Az ablakok vagy ikonok között kattintással jelölheted ki az aktuálist. Egy Windows ablak a következő elemekből épül fel:



A minimalizálógombra  kattintva az ablakot ikonná tudod alakítani. A maximalizálógomb  a teljes képernyőméretre nagyítja az ablakot.

Érdekes, hogy az **ablakot ikonná csukhatod**, ha a minimalizálógombra kattintasz, úgy, hogy az alkalmazás továbbra is fut.



A Windows 3.1 legalább MS DOS 3.3-at igényel. (Gyakorlatilag ennél már mindenütt korszerűbb DOS-t használnak.) Hardverigénye ún. **standard** üzemmód esetén 80 286-os processzor, legalább 1 Mbyte memória, 9 Mbyte szabad me-

revlemez és hajlékonylemezes meghajtó; **386-os** üzemmód esetén pedig leg-  
alább 80 386-os processzor, 2 Mbyte memória, 10,5 Mbyte szabad merevlemez  
és hajlékonylemezes meghajtó. Ezek a feltételek ma már szerénynek nevezhetők.



## Ellenőrző kérdések

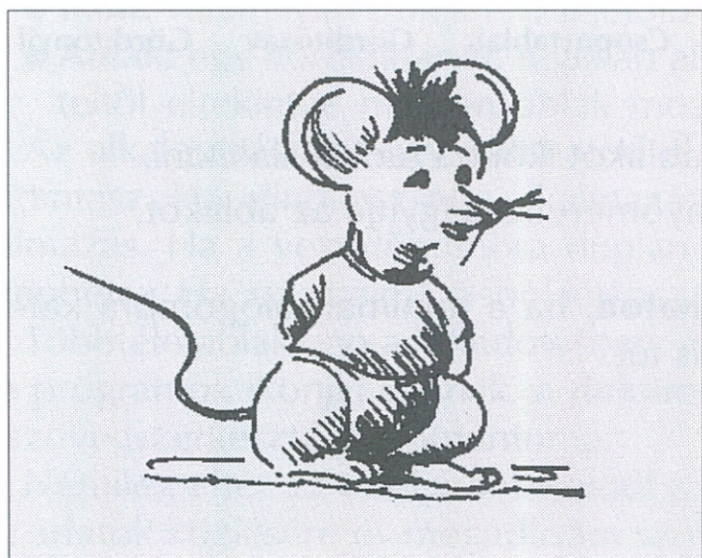
1. Mit nevezünk alkalmazásnak, csoportnak, ill. dokumentumnak?
2. Milyen ikonokat ismersz a Windowsban?
3. Milyen más ikonok léteznek?
4. Milyen elemekből áll a programkezelő ablaka?
5. Hogyan lehet az aktuális ablakot kijelölni?

## A WINDOWS KEZELÉSE

A rendszert egérrel vagy billentyűzettel kezelheted. Ha az egérkurzossal rámu-  
tatsz valamire a képernyőn, négy fontosabb műveletet végezhet az egérrel:

- ha **egyet kattintasz a bal oldali gombbal**, akkor ez jelenthet kijelölést, aktu-  
alizálást;
- ha a **bal oldali gombbal** gyorsan egymás után **duplán kattintasz**, akkor  
megnyitás, programindítás kezdődhet;
- ha a bal oldali gombot nyomva tartod, és húzod az egeret, a **mozgatási  
művelettel** megfoghatsz és elvihetsz valamit;
- ha a **jobb oldali gombbal kattintasz**, akkor egy menü jelenik meg.


A Windows kezelése az egérre épül, de mindent megcsinálhatsz a billentyű-  
zetről is.

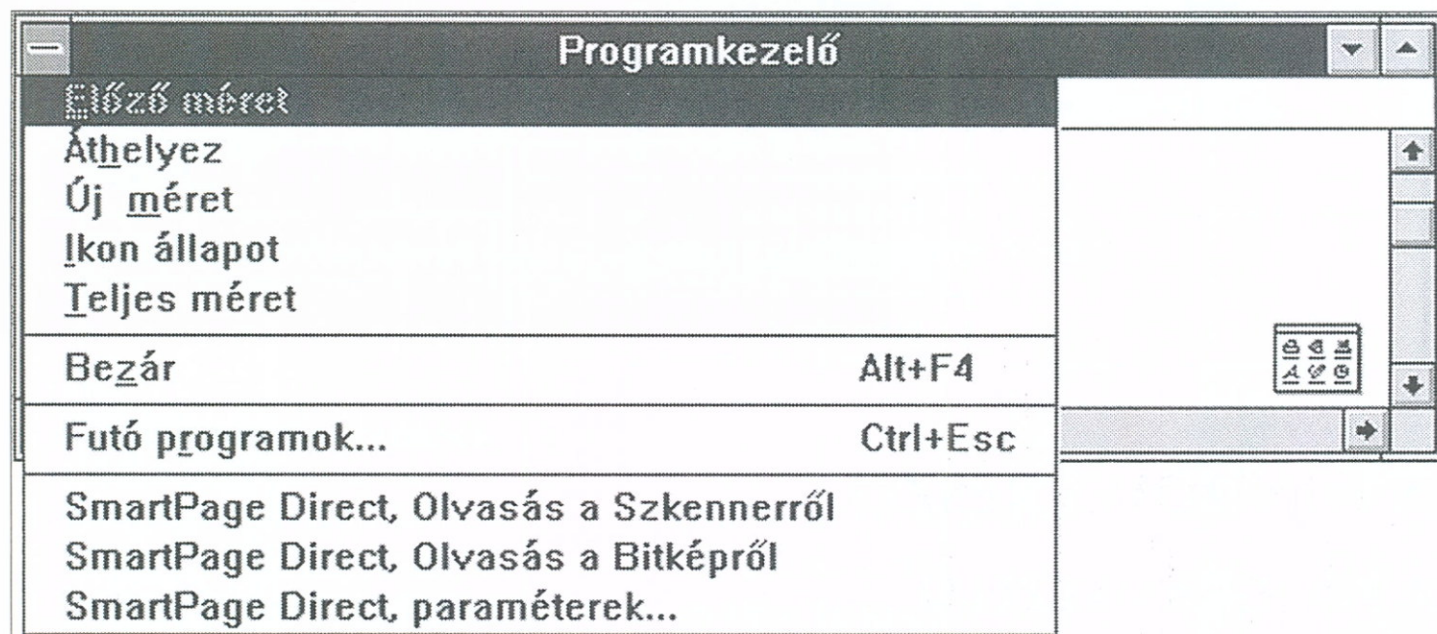


A menüsor egy menüpontja az **Alt** billentyű és a nevében aláhúzott betű  
együttes lenyomásával gördíthető le; pl. **Alt + F** legördíti a File menüt. Ha a le-

gördült listából egy újabb menüpontot akarsz kiválasztani, akkor üsd le az aláhúzott betűjét (pl. K kilépés), vagy vidd a kijelölőt, a kurzormozgatókkal a menüpontra, és üsd le az **Enter**-t!

Az **Esc** billentyű itt is menekülésre (pl. valamely ablakból való kilépésre) szolgál, hatása megfelel a **Mégsem** nyomógombnak, az **Enter** pedig az **OK** gombnak.

Ha egy ablak bal felső sarkában lévő **vezérlőgombra**  kétszer rákattintasz, **bezárhatod** az ablakot, ami alkalmazásablak esetén a programból (alkalmazásból) való **kilépést** jelenti, ha pedig egyszer kattintasz, az ablak **vezérlőmenüjét** gördítheted le. A megjelenő menüpontok az ablak állapotára vonatkoznak:



A **Futó programok...** menüpont kiválasztásával egy párbeszédablak jelenik meg, amelyben egy lista mutatja a megnyitott alkalmazásokat (futó programokat), ezekből bármelyiket aktualizálhatod.

Ha egy ablak **minimalizálógombjára**  kattintasz, az ablak ikon méretűvé csukódik, de az alkalmazás továbbra is fut (ill. ha dokumentum, továbbra is nyitva lesz). Ha pedig az ikonra duplán kattintasz, az ikon ablakká nyílik.

**Ne nyiss ki túl sok alkalmazást, mert lelassítja a rendszert, és a képernyő is túl zsúfolt lesz!**

**Ha nem találsz a futó alkalmazás ikonját, keresd a többi ablak mögött a tapétán!**



### Ellenőrző kérdések

1. Milyen műveletek végezhetők az egérrel, és mire szolgálnak ezek a műveletek?
2. Hogyan csuksz ikonná egy ablakot?

3. Hogyan lépsz ki egy alkalmazásból?

4. Mire használod a vezérlőgomb  menüjét?

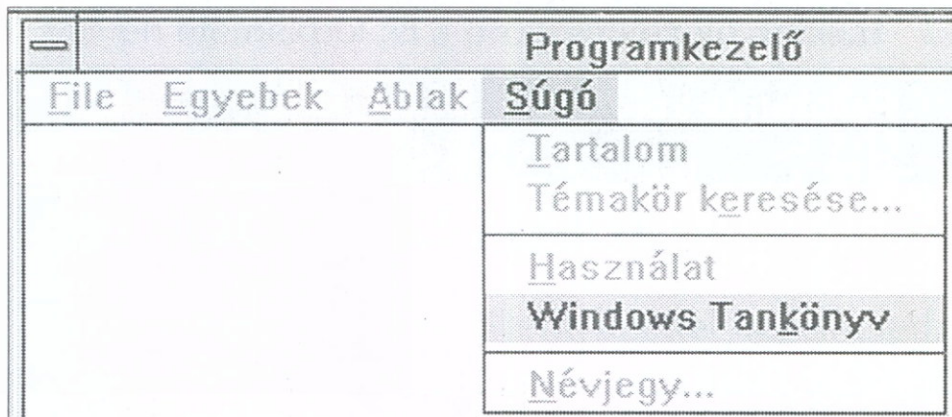
5. Hogyan tudsz átkapcsolni (aktuális ablakot váltani) a futó alkalmazások között?



## Gyakorlat

1. Az ellenőrző kérdésekre adott választ a gyakorlatban is hajtsd végre!

2. A Windows tankönyv segítségével gyakorold a Windows használatát!



## A WINDOWS RÉSZEI

### A PROGRAMKEZELŐ

A **Programkezelő** a Windows fő programja, indításkor ennek az ablaka vagy ikonja jelentkezik be, innen indíthatók pl. az alkalmazások. Ha bezárod az ablakot, kilépsz a Windows-ból.

A Programkezelő menüsora a következő pontokból áll:

File, Egyebek, Ablak, Súgó. Billentyűzettel vagy egérekattintással gördítsd le őket!

A legördülőmenük általános jelölései:

Ha a menüparancs

- sötét, akkor végrehajtható;
- halvány, akkor nem hajtható végre;
- után három pont (...) van, akkor párbeszédablakot hív;
- előtt pipajel van, akkor bekapcsolt állapotot jelöl.

Egyes pontok után billentyűk, ill. billentyűkombinációk állnak, amelyekkel közvetlenül is aktivizálhatod a kívánt menüpontot.

A Súgó segít Neked, ha valahol elakadsz és információra van szükséged.

Az Ablak menüben rendezheted az aktuális ablak ikonjait és a megnyitott ablakokat, vagy átkapcsolhatsz a csoportok között, aktívá téheted a kiválasztott alkalmazás ablakát.

Az Egyebekben az aktuális ablak állapotára vonatkozó opciókat kapcsolhatsz ki-be, a bekapcsolt állapotot egy pipa jelzi.

Az egyes pontok jelentése a File menüben:

File	
Új...	
Megnyit	Enter
Áthelyez...	F7
Másol...	F8
Töröl	Del
Jellemzők...	Alt+Enter
Futtat...	
Kilépés a Windowsból...	

új ikon létrehozása

a kijelölt csoportablak vagy alkalmazás megnyitása

ikon áthelyezése egy másik csoportba

ikon átmásolása egy másik csoportba

ikon törlése

ikon jellemzőinek megváltoztatása

alkalmazás indítása

a Windows befejezése



## Ellenőrző kérdések

1. Hogyan aktivizálhatók, ill. gördíthetők le a menüpontok?
2. Hogyan lehet érvényesíteni valamely választást?
3. Milyen egérműveleteket ismersz, és mi a hatásuk?
4. Hogyan tudsz váltani a futó programok között?
5. Hová „bújhat el” a futó alkalmazás ikonja?
6. Milyen általános jelöléseket ismersz a legördülőmenükben?



## Gyakorlatok

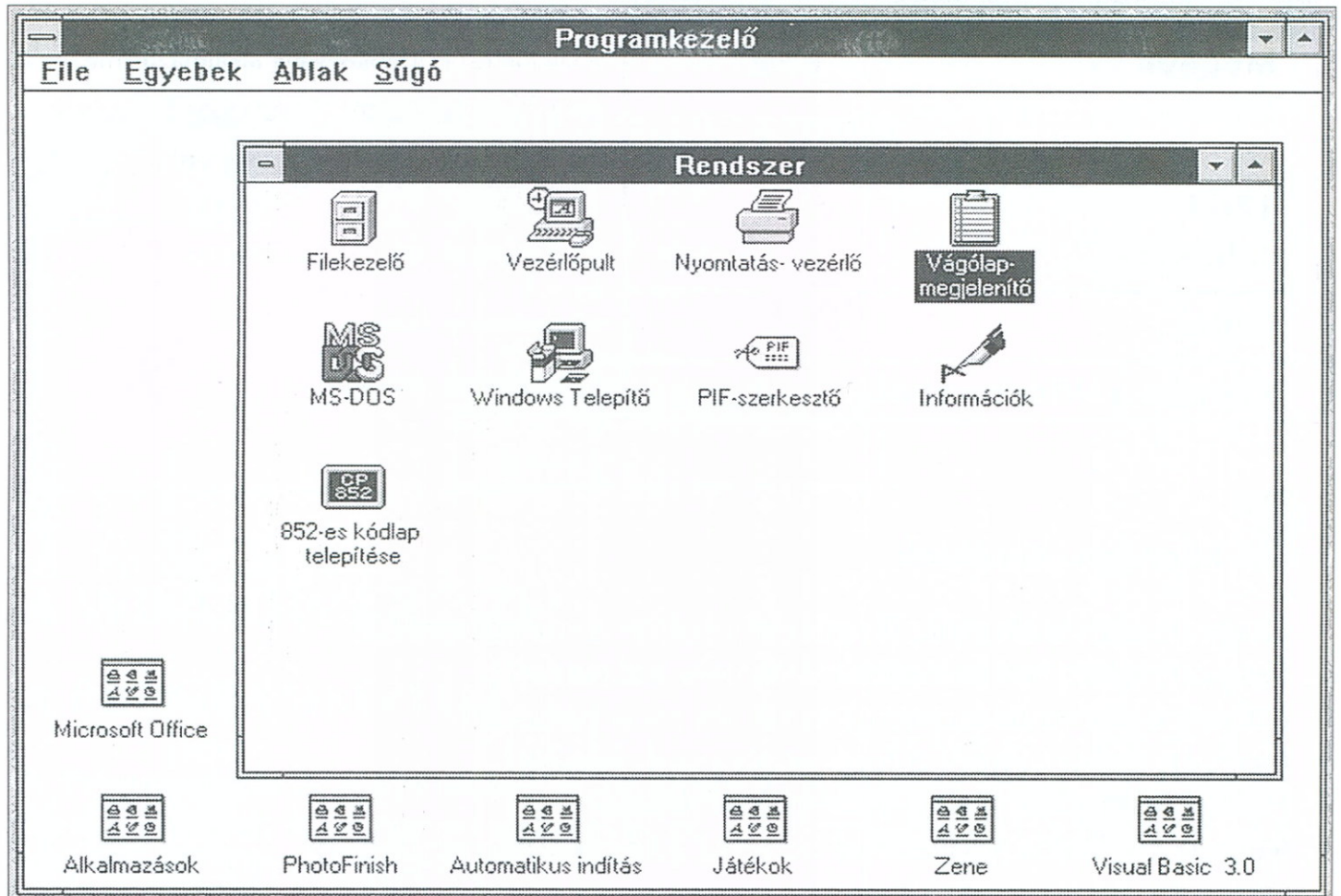
1. Nyiss ki néhány csoportikont (pl. kettős kattintással)! Válaszd ki az Ablak menüből a Lépcsőzetes, majd a Mozaik elrendezést, végül rendezd el az ikonokat!
2. A File Futtat parancsára megjelenő párbeszédablak parancssorába gépeld be a **write**-ot, és üsd le az **Enter**-t! Ezzel elindítod a Write szövegszerkesztőt. Hasonló módon tudsz más programokat is elindítani. Ha a vezérlőgombra duplán rákattintasz, kiléphetsz a szövegszerkesztőből.
3. Indítsd el a Kellékek csoportban található Paintbrush festőprogramot az ikonok segítségével, majd lépj ki a programból!
4. Próbáld végig a Programkezelő File menüjének pontjait! Vigyázz a törléssel! Egy átmásolt ikont törölj!



# FÁJLKEZELŐ



A Programkezelőben az egyik legfontosabb csoport a **Rendszer**. Általában a Rendszerben található a következő alkalmazások: Filekezelő (Fájlkezelő), Vezérlőpult, Nyomtatásvezérlő, Vágólap-megjelenítő, MS-DOS, PIF-szerkesztő, 852-es kódlap telepítése, Információk, Windows Telepítő.

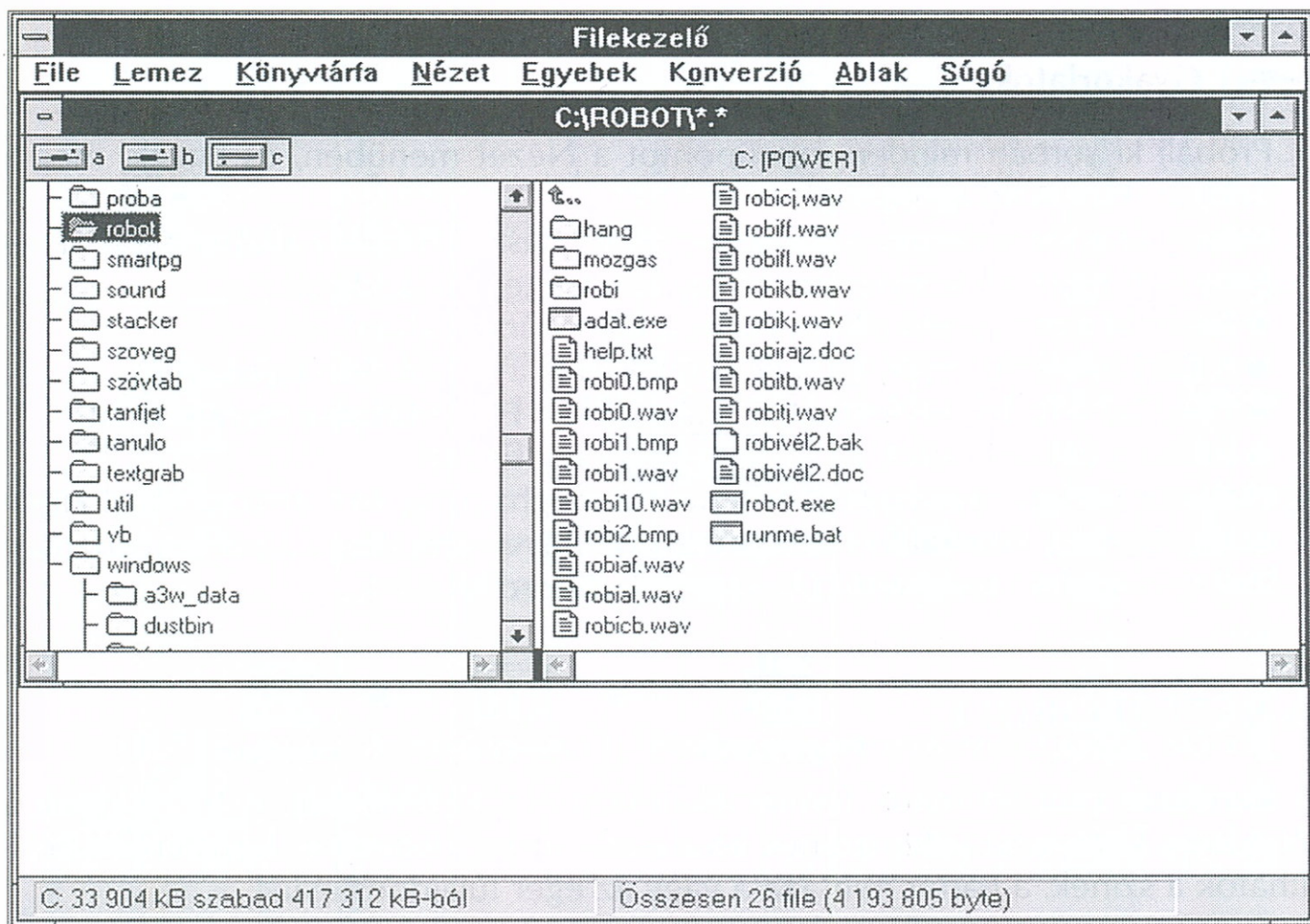


A **Fájlkezelő** a hagyományos MS-DOS fájl- és könyvtárkezelő szolgáltatásainak nagy részét hajtja végre egyszerű grafikus környezetben, hasonlóan a Norton Commanderhez.

A Fájlkezelő a háttértárak tartalmát a **könyvtárablakokban** jeleníti meg; a gyakori beállítás szerint ezek két részre osztottak: a bal oldali részben a könyvtár struktúrája, a jobb oldaliban az aktuális könyvtár tartalomjegyzéke látható. Ha a bal oldali részben kijelölsz egy alkönyvtárat, akkor a jobb oldali részben annak a tartalma jelenik meg. A Könyvtárfa és a Nézet menüben a Neked tetsző módon beállíthatod, hogy mennyi információt mutassanak a könyvtárablakok.

A könyvtárablakban a lemezegységeket ikonok jelzik, a kijelölt ikon mutatja az aktuális lemezegységet.

Ha több könyvtárablakra is szükséged van, akkor a megfelelő lemezegység kiválasztása után az **Ablak** menüből, az **Új ablak** paranccsal újabb könyvtárablakot nyithatsz. Az aktuális ablakot a szokásos módon, egérrel jelöld ki!



Egyszerűen **elindíthatsz** egy **alkalmazást** is, ha a program nevére duplán rákattintasz. Ha egy dokumentumra kétszer kattintasz, akkor a Windows előbb elindítja a megfelelő alkalmazást, majd megnyitja (benné) a dokumentumot. (Ha mégsem, akkor előbb ki kell jelölnöd, majd a File menüből a Tárσίít paranccsal társítanod a neki megfelelő alkalmazáshoz.)

A **File** menüben a párbeszédablakokon keresztül a szokásos fájlműveleteket végezheted el: áthelyezés, másolás, törlés, átnevezés, attribútumok állítása, nyomtatás – és alkönyvtárakat is itt hozhatsz létre.

A **Lemez** menüben lemezmásolást, címkézést, formázást végezheted, rendszerlemezt készíthetsz és meghajtóváltásra is módod nyílik.

**Állományt** nagyon egyszerűen áthelyezhetsz az **egérrel**: megfogod és a megfelelő könyvtárba mozgatod. Ha **másolni** akarsz, akkor tartsd nyomva a **Ctrl** billentyűt is!



### Ellenőrző kérdések

1. Hogyan néz ki egy könyvtárablak a fájlkezelőben?
2. Indíthatsz-e alkalmazást a fájlkezelőben?
3. Hogyan lehet két ablakot nyitni?



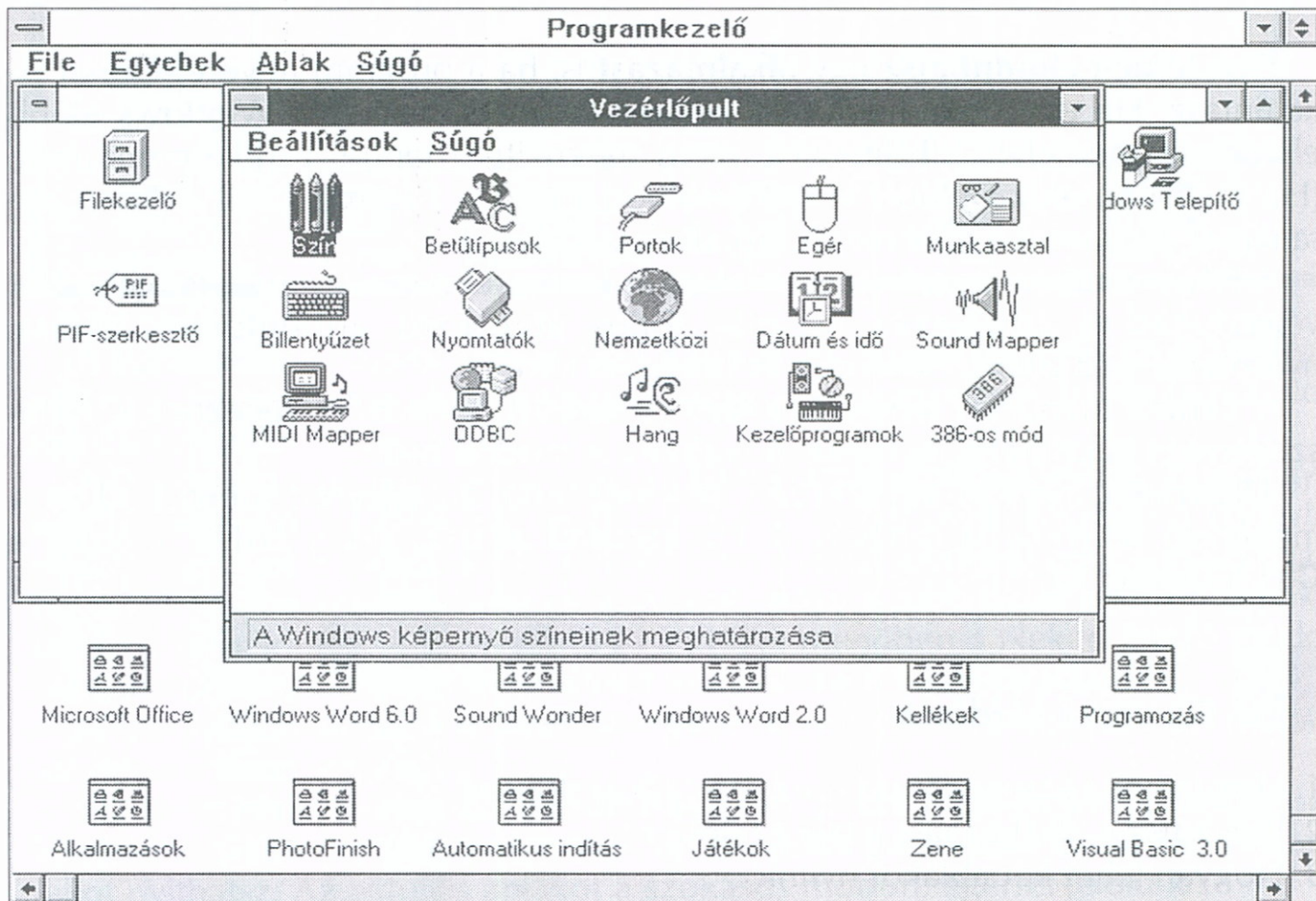
## Gyakorlatok

1. Próbáld ki sorban minden menüpontot a Nézet menüben, és figyelj meg a hatását! Használd az egeret!
2. Másolj át a merevlemez valamelyik (gyakorlásra szolgáló) alkönyvtárából néhány fájlt egy hajlékonylemezre! Egy lehetséges megoldás:
  - Jelöld ki a megfelelő alkönyvtárat a könyvtárlakban!
  - A **Ctrl** billentyűt nyomva (egérrel) jelöld ki a másolandó fájlokat!
  - Tedd be a hajlékonylemezt, és válaszd ki a **File, Másolás** menüpontot!
  - Írd be a célmeghajtót, és **OK** vagy **Enter**!
3. Helyezz be egy üres hajlékonylemezt a meghajtóba, és formázd meg! Használd a **Lemez, Formázás** menüpontokat! Ügyelj arra, hogy a párbeszédablakban a megfelelő meghajtót (A: vagy B:) válaszd ki!










## A VEZÉRLŐPULT



A **Rendszer** csoport másik fontos alkalmazása a **Vezérlőpult**. A Vezérlőpult a Windows környezet jellemzőinek kiválasztását teszi lehetővé. Egyszerűen beállíthatók a színek, a háttér mintázata vagy az egér tulajdonságai.



## A fontosabb ikonok magyarázata:

 Szín	A képernyőelemek színe állítható be egy párbeszédablakban.
 Betűtípusok	A Windows-alkalmazások által használható betűtípusok meghatározása, újabbak felvétele és törlése.
 Egér	Az egér működési paraméterei állíthatók (sebesség, bal/jobbs gomb csere ...).
 Munkaasztal	A monitorkép beállítása (képernyővédő, tapéta, ikonok, kurzor, ...).
 Billentyűzet	A billentyűzet sebességének beállítása.
 Nyomtatók	A nyomtatótípusok telepítése, törlése, a nyomtatók beállítása.
 Nemzetközi	A nyelv és az adott országra jellemző formátumok, billentyűzetkiosztás, mértékegységek beállítása.
 Dátum és idő	A dátum és az idő beállítása.
 Hang	A rendszereseményekhez rendelhető hangok beállítása.



## Gyakorlatok

1. Állítsd be a billentyűzet késleltetését közepesre, ismétlési ütemét gyorsra!
2. Állítsd be az egér dupla kattintásának sebességét lassúra! Cseréld fel a bal és a jobb gombot! A balkezeseknek így célszerű használni. (Ne felejtse el visszaállítani!)
3. Állítsd át a képernyővédőt és a tapétát!
4. Ellenőrizd a dátumot és az időt, szükség esetén korigáld!

## NYOMTATÁSVEZÉRLŐ



A telepített nyomtatók típusát és állapotát mutatja. A nézetre és a nyomtatási folyamatra vonatkozó beállításokat változtathatjuk, pl. megállíthatjuk vagy elindíthatjuk a nyomtatást. Itt is elérhető a nyomtatóbeállítás ablaka az összes funkciójával.

## VÁGÓLAP-MEGJELENÍTŐ



A Windows-alkalmazások dokumentumaiból, állományjaiból kimásolhatsz vagy kivághatsz részeket (kép, szöveg, táblázat, ...), amelyek a vágólapra kerülnek. A vágólap-megjelenítővel megnézheted a tartalmát, elmentheted egy .clp kiterjesztésű fájlba, vagy megnyithatsz egy ilyen. A vágólap tartalmát valamelyik Windows-alkalmazással beillesztheted egy tetszőleges dokumentumba. Az újabb ráírás törli a vágólapot.

## MS-DOS IKON



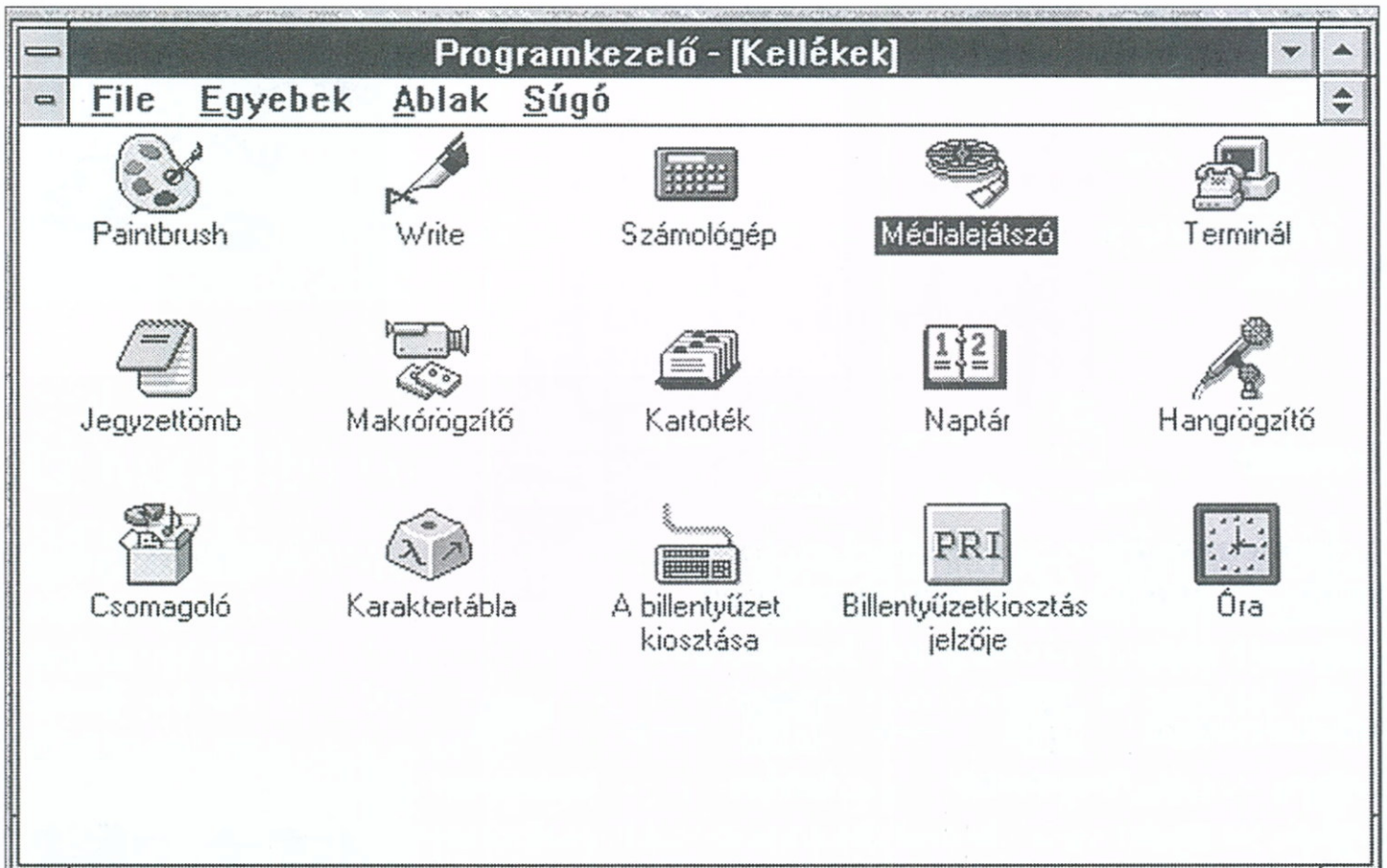
Ideiglenesen elhagyhatod a Windows grafikus felületét és visszatérhetsz a DOS szöveges képernyőjéhez, ahol a DOS-ban megszokott módon dolgozhatsz. Választhatsz a teljes képernyős vagy az ablakos megjelenítés között. Az EXIT parancs beírásával visszatérhetsz a kényelmes Windowsos felülethez.



## Kellékek



A Kellékek csoport a irodai munkához, tanuláshoz és szórakozáshoz használható eszközöket ad a kezvedbe.

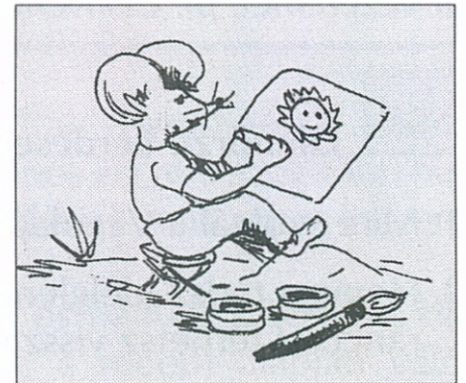


## Néhány alkalmazás

### Paintbrush



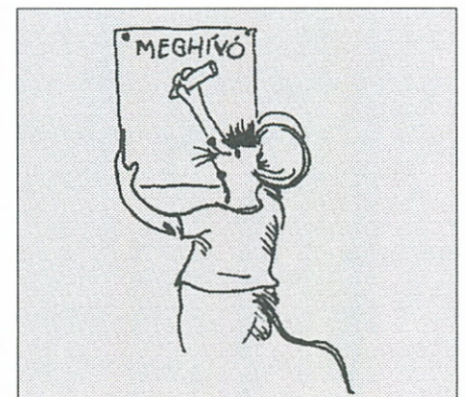
Kiseb, színes rajzok elkészítésére alkalmas rajzóprogram. A képeket szerkesztheted, feliratozhatod, kinyomtathatod, állományba mentheted, vagy a vágólap segítségével más dokumentumba illesztheted, másolhatod.



### Write



Egyszerű, de használható szövegszerkesztő, amely a Word for Windows elterjedésével veszített a jelentőségéből.



## Számológép



Választhatsz az egyszerű és a „tudományos” kalku-  
látor közül.



## Terminál



Modemen keresztül telefonkapcsolatba léphetsz más számítógépekkel. Ez a terület sokat fejlődött az elmúlt években.

## Médialejátzó



Jelentősége a multimédia gépek elterjedésével nő.  
A számítógép felszereltségétől függően többféle médi-  
át vezérelhet, pl. CD-ROM-ot, hang CD-t, video CD-t.



### Ellenőrző kérdések

1. Mire szolgál a Vágólap-megjelenítő?
2. Hogyan tudsz ideiglenesen, de gyorsan visszatérni a DOS-hoz, és milyen pa-  
ranccsal térhetsz vissza?



Mire használható a médialejátzó?

## AZ ALGORITMUS

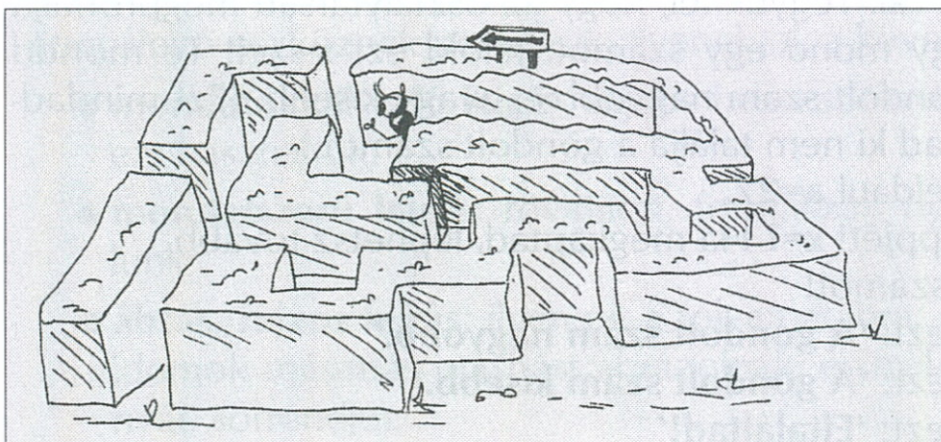
Elgondolkoztál már azon, hogy mi mindenre jó a számítógép? Röviden azt válaszolhatnánk, hogy feladatmegoldásra. **Megfelelő programmal** és megfelelő hardvereszközzel (pl. nyomtatóval, mikrofonnal, hangszórával, mérőberendezéssel) sokféle feladatot meg tudunk oldani. A hardverről már tanultál, de a programról még nem sokat.

Mi tehát a program? Lényegében a számítógép által végrehajtható algoritmus. Ha tanultál már informatikát, akkor bizonyára tudod, hogy mi az az algoritmus.

**Az algoritmus egy feladattípus véges számú lépésre bontott megoldása.**

- Fontos, hogy az algoritmus – véges számú lépés után – eredményt adjon.
- Elvárjuk azt is, hogy az algoritmus valamennyire általános legyen, azaz több hasonló, pl. a bemenő adatokban eltérő feladat megoldására is alkalmas legyen.

A mindennapi életben sokféle algoritmust ismerünk és hajtunk végre (pl. algoritmus a telefonálás, az automatából való vásárlás, a recept alapján való főzés, a számok összeszorzása).





Ha nyitott szemmel jársz a világban, észreveszed, hogy ma már szinte mindenütt ott van a számítógép – irodában, hivatalban, szervizben, sőt a boltokban is. És Te is fogod használni a gépet, akár egy irodában, akár gépek mellett dolgozol majd! A felhasználás tehát szinte végtelen: a tudományos kutató ugyanúgy számítógépet használ, mint a szórakozásra, kikapcsolódásra vágyó diák. A legtöbb alkalmazónak nem kell programoznia, de azért hasznos, ha ismeri az alapvető algoritmusokat. Te is könnyebben használhatod majd a korszerű alkalmazói programokat, ha tisztában vagy az algoritmikus gondolkodás alapjaival! Ehhez meg kell tanulni egy programnyelvet. Ahogy lassanként megismerkedsz a programozással, egyre több örömet találsz benne, hiszen az alkotás: öröm, a jó program a számítógépben „életre kel”, működik. Az izgalmas, érdekes játékprogramokat is programnyelveken írják. Ezek persze több ember összehangolt munkájával készülnek, és igen bonyolultak, de egyszerűbbeket mi is tudunk írni.

*Muhammed ibu Muza al-Khvarizmi arab matematikus 825-ben könyvet írt a fontosabb számolási módszerekről, eljárásokról. Az algoritmus szót az ő nevéből származtatják.*



Írjuk le néhány egyszerű feladat megoldásának menetét, algoritmusát! Számold ki a háromszög területét, ha adott az egyik oldala és a hozzá tartozó magasság!

Ha az oldalt **a**-val, a magasságot **m**-mel és a területet **t**-vel jelöljük, akkor  **$t = a \cdot m / 2$** .

1. Szükség van **a** és **m** konkrét értékére (**a**=?; **m**=?). Ezek a bemeneti adatok.
2. El kell végezni a számítást, ezzel **t**-nek értéket adsz:  **$t = a \cdot m / 2$** .
3. Közölni kell az eredményt: **t** értékét. Ez lesz a kimeneti adat.



Írj egy algoritmust a következő számkitaláló játékra! Gondolj egy egész számot 0 és 100 között. Jelöld ezt **a**-val. Tegyük fel, hogy az osztálytársad megpróbálja kitalálni, mégpedig úgy, hogy mond egy számot. Jelöld ezt **x**-szel! Te mondd meg neki, hogy az általad gondolt szám nagyobb-e, avagy kisebb. (Ezt mindaddig játszhatjátok, amíg a társad ki nem találja a gondolt számot.)

1. Gondolj egy számot! Például **a=27**.
2. Kérd az osztálytársad tippjét! **x**=? Ha megkaptad, léphetsz tovább.
3. Hasonlítsd össze a két számot!

**Ha  $a > x$ , akkor mondd ezt: 'A gondolt szám nagyobb.'**

**Ha  $a < x$ , akkor mondd ezt: 'A gondolt szám kisebb.'**

**Ha  $a = x$ , akkor mondd ezt: 'Eltaláltad!'**

Bővítsd ki az előző megoldást úgy, hogy mindaddig folytatódjon a játék, amíg a társad el nem találja a gondolt számot! Például:

4. **Ismételd a 2. ponttól** az előzőeket mindaddig, **amíg  $x=a$  nem teljesül!** Elég hosszadalmas, és néha nem is egyértelmű, ha az algoritmusokat ehhez hasonló módon akarjuk leírni. Szerencsére, kifejlesztettek néhány algoritmusleíró eszközt, amelyek segíthetnek a program tervezésében.



### Ellenőrző kérdések

1. Mi az algoritmus?
2. Mi az összefüggés a program és az algoritmus között?



### Feladatok

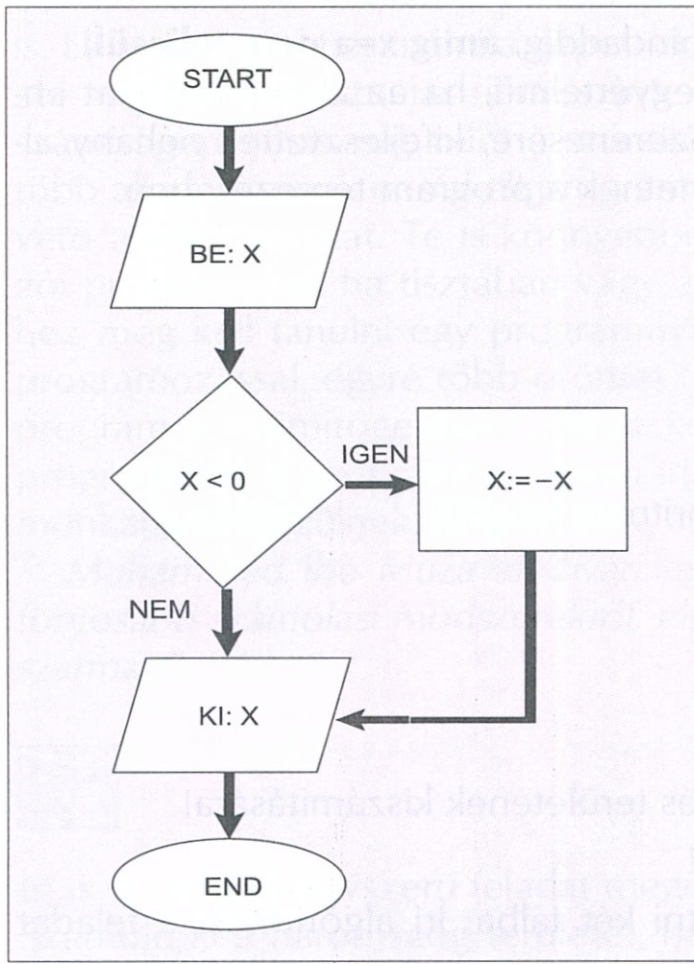
1. Írj algoritmust egy téglalap kerületének és területének kiszámítására!
2. Írj algoritmust egy téglalap lerajzolására!
3. Bab- és borsószemeket kell szétválogatni két tálba. Írj algoritmust a feladat megoldására!
4. Hat tojást kell megsütni. Írd le a tojássütés algoritmusát!
5. Egy osztályban kisorsolják a karácsonyi ajándékokat. Éppen annyi ajándék van, ahány gyerek. Írd le a sorsolás algoritmusát!
6. Hogyan kell pénzt felvenni az automatából? Írd le ennek az algoritmusát!

## ALGORITMUSOK LEÍRÁSA

A konkrét programnyelvektől többé-kevésbé független leírásokat általános algoritmusleíró eszköznek nevezzük. Ilyenek pl. a következők:

- **mondatokkal való leírás;** sajnos hosszadalmas és nem elég pontos; eddig ezt használtuk;
- **mondatszerű leírás;** rövidített, formalizált nyelv, erről bővebben is tanulunk;
- **ábrás, rajzos leírás;** ilyen pl. a folyamatábra; ebben különböző geometriai idomok más-más utasítást ábrázolnak, és nyilak mutatják a végrehajtás irányát, sorrendjét.

A mellékelt folyamatábra az abszolútérték-képzés algoritmusát mutatja:



Az algoritmusleíró eszközök hasznosak az áttekinthetőség, érthetőség szempontjából, de ahhoz, hogy a számítógép számára „érthető” legyen, az algoritmust mindig valamilyen programnyelven kell leírni.

Az algoritmusokhoz szorosan hozzátartoznak az adattípusok, hiszen ezeket alakítják át, ezekkel végeznek különféle műveleteket, ezeket fogadják és továbbítják.

## ADATTÍPUSOK

- **Egész szám;** adott értelmezési tartománnyal, általában 1–4 byte-on ábrázolhatók (kettes számrendszerben). Ilyenek pl. a 2 byte-os,  $-32768$ -tól  $+32767$ -ig terjedő egész számok.
- **Valós szám;** általában 4–10 byte-on ábrázolhatók, a pontosság 7–20 számjegynyi, a típustól függően. Valójában racionális számokról van szó. A legnagyobb leírható szám legalább  $10^{38}$ .
- **Logikai típus;** értéke IGAZ vagy HAMIS.
- **Karakter;** értéke egy karakter, az ASCII-kódrendszerből.
- **Karakterfüzér;** más néven string típus (string=szöveg), karakterek valamilyen sorozata, Pascalban max. 256 karakter.

A pontosabb típusdefiníciók függenek a használt programnyelvtől. Léteznek más adattípusok is (ezekkel később foglalkozunk), sőt mi is meghatározhatunk (definiálhatunk) adattípusokat, ha szükséges a feladat megoldásához.

## MONDATSZERŰ ALGORITMUSLEÍRÁS

Ahhoz, hogy rövid, áttekinthető, de azért pontos algoritmusokat tudj írni, meg kell ismerkedned néhány egyszerű szabállyal, szövegformulával.

*A program elejét és végét így add meg:*

**Program**

utasítások sorozata

**Program vége.**

*Értékadás:*

**változó:=értéke** vagy **változó:=kifejezés értéke**

Például:

$x:=5$ , az  $x$  legyen egyenlő öttel,

$x:=x+1$ , az  $x$  új értéke legyen egyenlő az  $x$  régi értéke + 1-gyel,

$t:=a*m/2$ .



Nagyon fontos, hogy ez nem a matematikából ismert egyenlet, hanem értékadás.

*Az adat be- vagy kivitel ilyen egyszerű:*

**Be: változók értékei**

**Ki: változók vagy kifejezések értéke**

Például: **Be: a, m**

**Ki:  $a*m/2$**

*Elágazások:*

1. **Ha feltétel akkor** utasítás

**Elágazás vége**

Például: **Ha  $a>x$  akkor** Ki: 'A gondolt szám nagyobb'

**Elágazás vége**

2. **Ha feltétel akkor** első utasítás

**különben** második utasítás

**Elágazás vége**

A feltétel egy logikai változó vagy egy kifejezés, amelynek igaz voltát vizsgáljuk. A feltétel egy állítás. Az 1. esetben, ha az állítás igaz, akkor az utasítást végre kell hajtani, ha nem, akkor nem kell elvégezni. A 2. esetben, ha az állítás igaz, akkor az első utasítást kell végrehajtani, különben pedig a második utasítást.

Például egy feladat szerint borsót és babot kell szétválogatni. A **mag** változóban adott, hogy az éppen kézbe vett mag milyen. El kell dönteni, hogy melyik tálba tesszük. Pl. így:

**Ha** *mag*="borsó", **akkor** Tedd a bal oldali tálba!

**különben** Tedd a jobb oldali tálba!

**Elágazás vége**



*Ciklusok:*

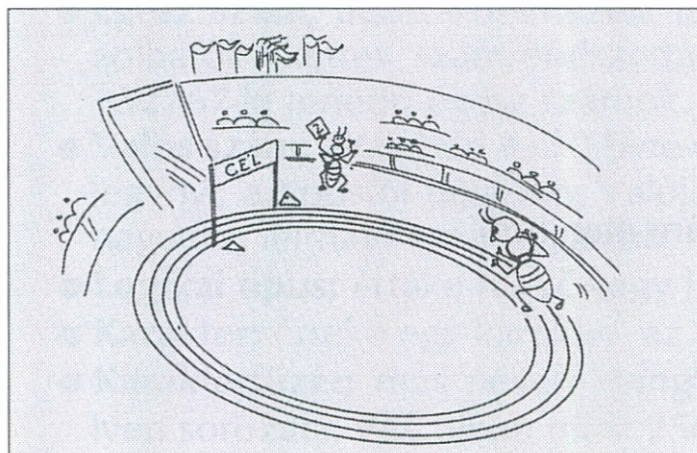
Ha egy feladatot sokszor kell egymás után elvégezni, **ciklust** használunk. Ennek egyik fajtája a számlálóciklus.

*Számlálóciklus:*

**Ciklus x:=x1-től x2-ig lépésköz delta**

ciklusmag utasításai (Ezek ismétlődnek.)

**Ciklus vége**



A számlálóciklusban szükség van egy ciklusváltozóra (**x**), ennek kezdő értékére (**x1**), és utolsó értékére (**x2**). (Általában megadható a lépésköz is (*delta*), ilyen lépésekben vesz fel értékeket a ciklusváltozó. Ha nem adunk meg lépésközt, akkor 1-nek kell venni.)

Példa:

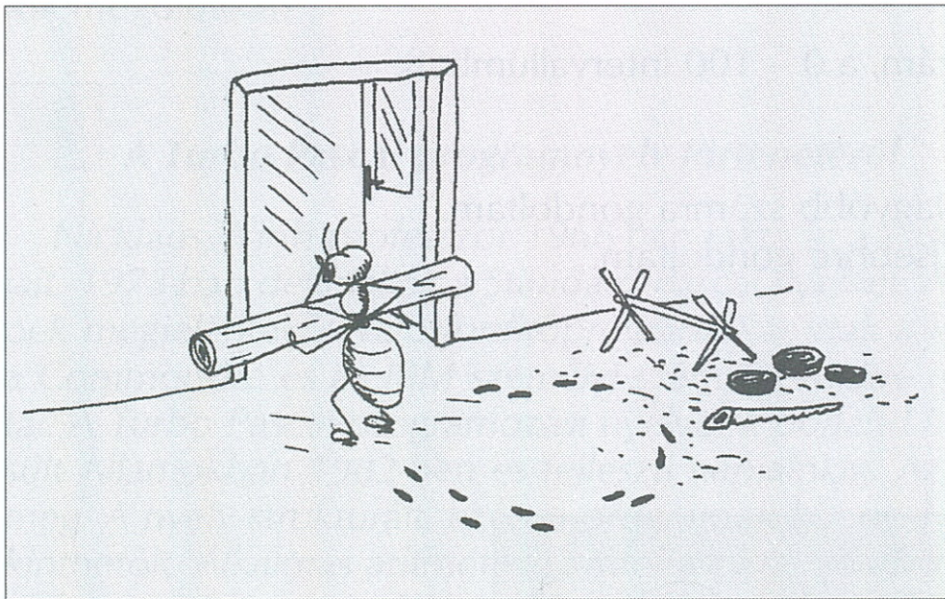
**Ciklus x:=1-től 10-ig**

Ki:  $x, x^2$  (Ezt az utasítást kell 10-szer elvégezni.)

**Ciklus vége**

Sokszor előfordul, hogy egy feladatot mindaddig ismételni kell, amíg egy logikai feltétel igazgá nem válik. Egyszer azonban mindenképpen végre kell hajtani. Erre szolgálhat a hátultesztelő ciklus.

*Hátultesztelő ciklus:*



**Ciklus**

ciklusmag utasításai (Ezek ismétlődnek, amíg a feltétel igazgá nem válik.  
Nem fontos, hogy hányszor.)

**amíg feltétel**

**Ciklus vége**

Példa:

**Ciklus**

Be: karakter a billentyűzetről

**amíg karakter='t'** (A 't' betű leütése után megy tovább.)

**Ciklus vége**

(Ez a ciklus arra „várt”, hogy valaki leüsse a t betűt a billentyűzeten.)



## Példák

Írjuk le mondatszerű leírással a „területszámítás” és a „gondoltam egy számot” algoritmusokat!

### Program

a, m, t: valós típusú adatok

Be: a, m

t:=  $a \cdot m / 2$

Ki: 'A háromszög területe:', t

### Program vége

Ugye milyen egyszerű? A „gondoltam egy számot” algoritmus sem nehéz!

### Program

a, x: egész számok

a:=véletlen egész szám, a 0 – 100 intervallumban.

### Ciklus

Be: x

**Ha**  $a > x$ , **akkor** Ki: 'Nagyobb számra gondoltam.'

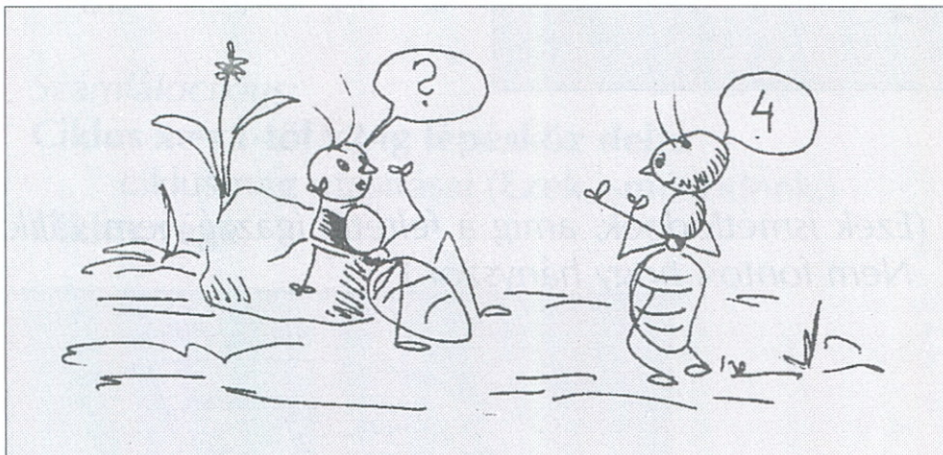
**Ha**  $a < x$ , **akkor** Ki: 'Kisebbre gondoltam.'

**amíg**  $a \neq x$

### Ciklus vége

Ki: 'Eltaláltad!'

### Program vége



Mondatszerűen megfogalmazott megoldásainkat a Pascal nyelv segítségével tesszük a gép számára is érthetővé. A Pascal magas szintű programozási nyelv, sok utasítása van, az utasítások angol szavak vagy pedig rövidítések. Minden magas szintű nyelvhez szükség van fordítóprogramra, amely le tudja fordítani gépi (bináris) kódra az utasításokat. A processzor csak ezeket az utasításokat ké-

pes „értelmezni”, ezeket tudja végrehajtani. Szükség van szövegszerkesztőre, amelyben megírhatjuk a programot. Szükség van a hibakeresést megkönnyítő szoftvereszközökre. Ilyen és ehhez hasonló feladatokat lát el egy program, a nyelv fejlesztőrendszere.



### Ellenőrző kérdések

1. Milyen általános algoritmusleíró eszközöket ismersz?
2. Milyen adattípusokról tudsz?
3. Milyen szabályai vannak a mondatszerű leírásnak?



### Feladat

Írd le mondatszerűen is az e fejezet elején lévő, a 89. oldalon található feladatok megoldását!



### A Turbo Pascal programnyelv történetéről

Nicklaus Wirth professzor 1968-ban rakta le a Pascal programozási nyelv alapjait. 1973-ban definiálták a Standard Pascal nyelvet. Amikor a személyi számítógépek megjelentek, kiváló fordítóprogramokat írtak a különböző egyetemeken, így a Commodore és az IBM személyi számítógépekre is elterjedtek a nyelv változatai. A Turbo Pascal programozási nyelvet a Borland Intézet fejlesztette ki az Egyesült Államokban 1982-ben és fejleszti napjainkig. Az első verzió 1984-ben jelent meg. A nyelv strukturált, ezért a programok könnyebben fejleszthetők és jól áttekinthetők. Alkalmas aritmetikai számításokra, adatfeldolgozásra, grafikus megjelenítésre. Az 5.5 verziótól ún. objektumorientált programozásra is.

A Turbo Pascal igen elterjedt az oktatásban, a nyelv fejlesztőrendszere (programja) interaktív, saját szövegszerkesztője van, amellyel a forrásprogram szövegét megszerkeszthetjük. Igen gyors a fordítója, amely a Pascal nyelven írt forrásprogramot lefordítja a processzor által végrehajtható gépi kódra.

## RÖVIDEN A TURBO PASCAL FEJLESZTŐRENDSZERRŐL

A rendszer a TURBO.EXE programmal indul. Ha a rendszert elindítod, a képernyő felső szélén megjelennek a fő menüpontok: **File** (állomány, fájl), **Edit** (szerkeszt), **Search** (keres), **Run** (futtat), **Compile** (fordít), **Debug** (javít), **Tools**



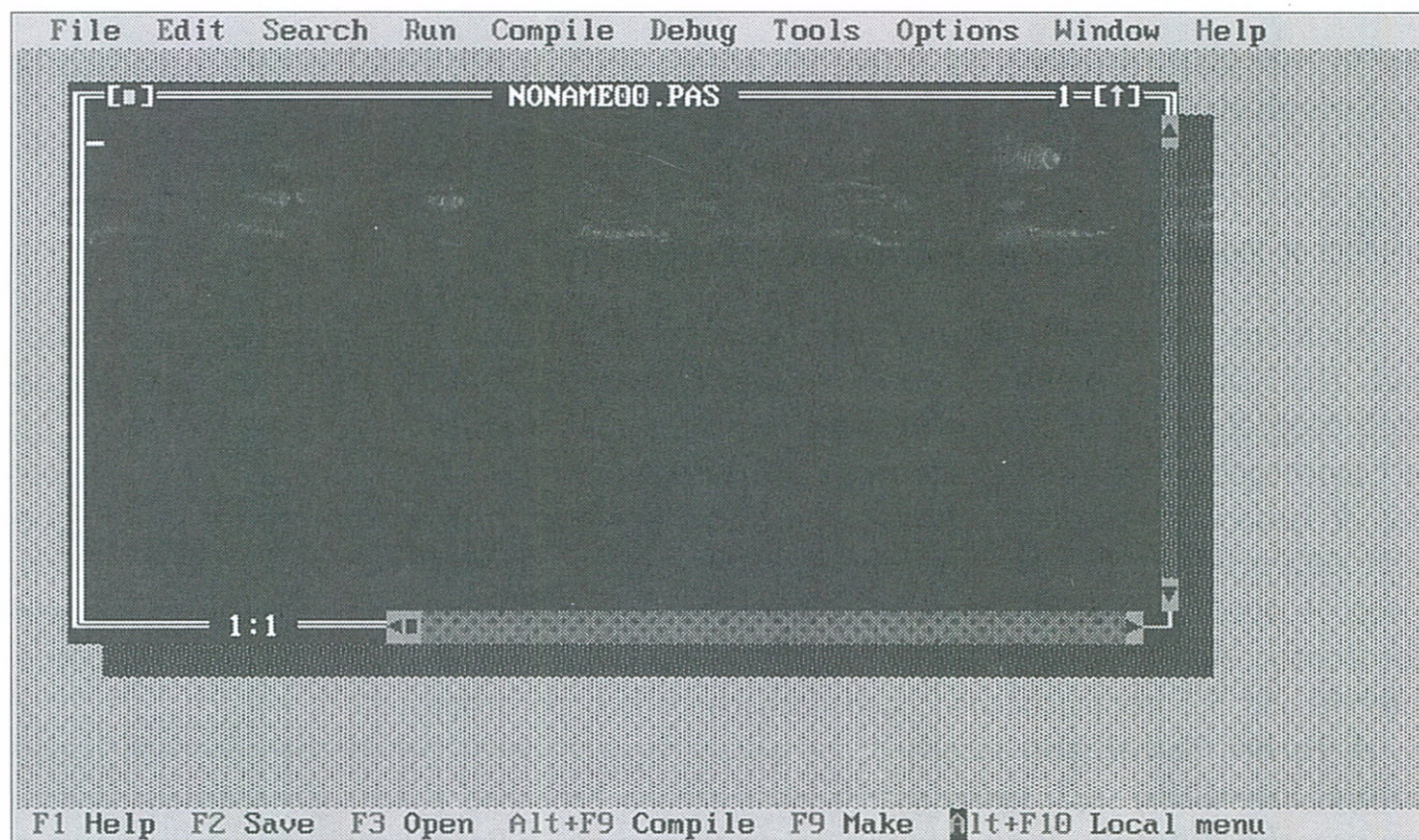
(szerszámok), **Options** (lehetőségek, opciók), **Window** (ablak), **Help** (segítség, súgó). Ezeket a szokásos módon érheted el: egérrel vagy az **Alt + betű** kombinációval, vagy pedig egyszerűen **F10** billentyűvel is. (A legördülő menüket általában úgy kell kezelni, mint más programokban.)

(Mivel a nyelv nem ismeri a magyar ékezetes karaktereket, a fejlesztőrendszer indítása előtt elindíthatsz egy karakterátdefiniáló programot (például: *mskeyb.com*, *keybhu*, 852-es kódlap), vagy végső esetben, az ASCII-kódtáblázatot is használhatod – a tanult módon (**Alt** + kódszám). Jó, ha fut az egérkezelő (*mouse.com*, *mouse.sys*) program is.)

Nézzünk egy példát a rendszer használatára!

Nyiss egy ablakot a begépelendő programod számára a **File, New** menüpontokkal!

Vess egy pillantást az ablakra! Talán kevésbé mutatós, de nagyon hasonlít a Windows-nál megszokottra, és egérrel jól kezelhető:



- Felül középen (a kereten) a program neve látható, és az a könyvtár, ahová utoljára mentettél. Ha még nem adtál nevet a programnak, akkor a NONAMEXY.PAS nevet kapja, ahol XY egy sorszám.
- A bal felső sarokban lévő jellel [X] bezárhatod az ablakot.
- A jobb felső sarokban a kereten van a teljes méret [↑], vagy az előző méret jele.
- Itt láthatod az ablak sorszámát is.
- Baloldalt alul (a kereten) a kurzor aktuális koordinátái jelennek meg (sor : oszlop).
- Alul és jobboldalt a gördítősávok vannak.



Írd be a háromszög területét számoló algoritmust, Pascal nyelven, a következő formában:

```
Program haromszog;  
Uses Crt;  
Var  
    a, m, t: real;  
Begin  
    ClrScr;  
    write('Kérem az alapot:');  
    read(a);  
    write('Kérem a magasságot:');  
    read(m);  
    t:=a*m/2;  
    writeln('A háromszög területe:',t:8:3);  
End.
```

A monitoron ezt fogod látni:

The screenshot shows a Turbo Pascal IDE window titled "BP\BIN\HAR.PAS". The code displayed is identical to the one in the previous block. The IDE interface includes a menu bar with "File", "Edit", "Search", "Run", "Compile", "Debug", "Tools", "Options", "Window", and "Help". At the bottom, there is a status bar with function key shortcuts: "F1 Help", "F2 Save", "F3 Open", "Alt+F9 Compile", "F9 Make", and "Alt+F10 Local menu". The cursor is positioned at the end of the "writeln" statement on line 15.

Ha összehasonlítod a mondatszerű leírással, beláthatod, hogy nem is olyan nehéz megérteni a programot, csupán meg kell tanulnod néhány jelölést! Előbb azonban futtasd le! Tényleg úgy működik, ahogy szeretnéd?

A programot futtatni a **Run** menüben található **Run** paranccsal tudod, vagy a **Ctrl + F9** billentyűkombinációval.

- Ha a rendszer hibát jelez, megjelenik a hiba kódszáma és a hiba fajtája. A kurzor ott villog, ahol a rendszer a hibát észlelte. Ilyenkor ellenőrizd, hogy mit írtál be helytelenül! Esetleg a program elején nem, vagy rosszul adtál meg egy változót.
- Ha minden rendben van, és a program fut, akkor kéri az alap hosszát. Beírsz egy számot (pl.: 12), és leütöd az Enter-t. Ezután kéri a magasságot, azt is megadod (pl.: 5). Erre visszatér a szerkesztőablak a programunkkal.
- Ha látni akarsz az eredményt, akkor válaszd ki a **Debug** (hibakeresés) menüben az **User screen** (felhasználói képernyő) pontot! Ezt gyorsabban is megteheted az **Alt+F5** billentyűkkel. Ugyanitt, a **Debug** menüben **Output** (kimenet) ablakot is beállíthatsz, és ekkor egyszerre láthatod az eredményt és a programot is.

Programod három részből áll: programfejből, deklarációs részből és végrehajtási részből. A programfejben a **Program** szó után kell megadnod a program nevet, amit pontosvessző választ el a mögötte lévőttől. Egyébként minden utasítást (eljárást, fejlécet) zárj pontosvesszővel, kivéve a program végét jelző **End.** utasítást, amit ponttal kell zárnod!

Add meg a deklarációs részben a használni kívánt szoftvereszközöket: konstansokat, változókat, unitokat! A Pascalban **unit**-nak (egységnek) nevezzük a programjainkban felhasználható eljárásokat, függvényeket, változókat tartalmazó csomagot. Ha használni akarsz egy unitot, akkor a deklarációs rész elején a **Uses** kulcsszóval meg kell nyitnod, pl.: `Uses Crt;` . A **Crt unit** a képernyőt (kivéve a grafikát), a billentyűzetet és a hangszórót kezelő eljárásokat, függvényeket tartalmazza.

A **Var** szó után kell felsorolni a változókat, megadva az adattípusukat.

A **real** valós típust jelent.

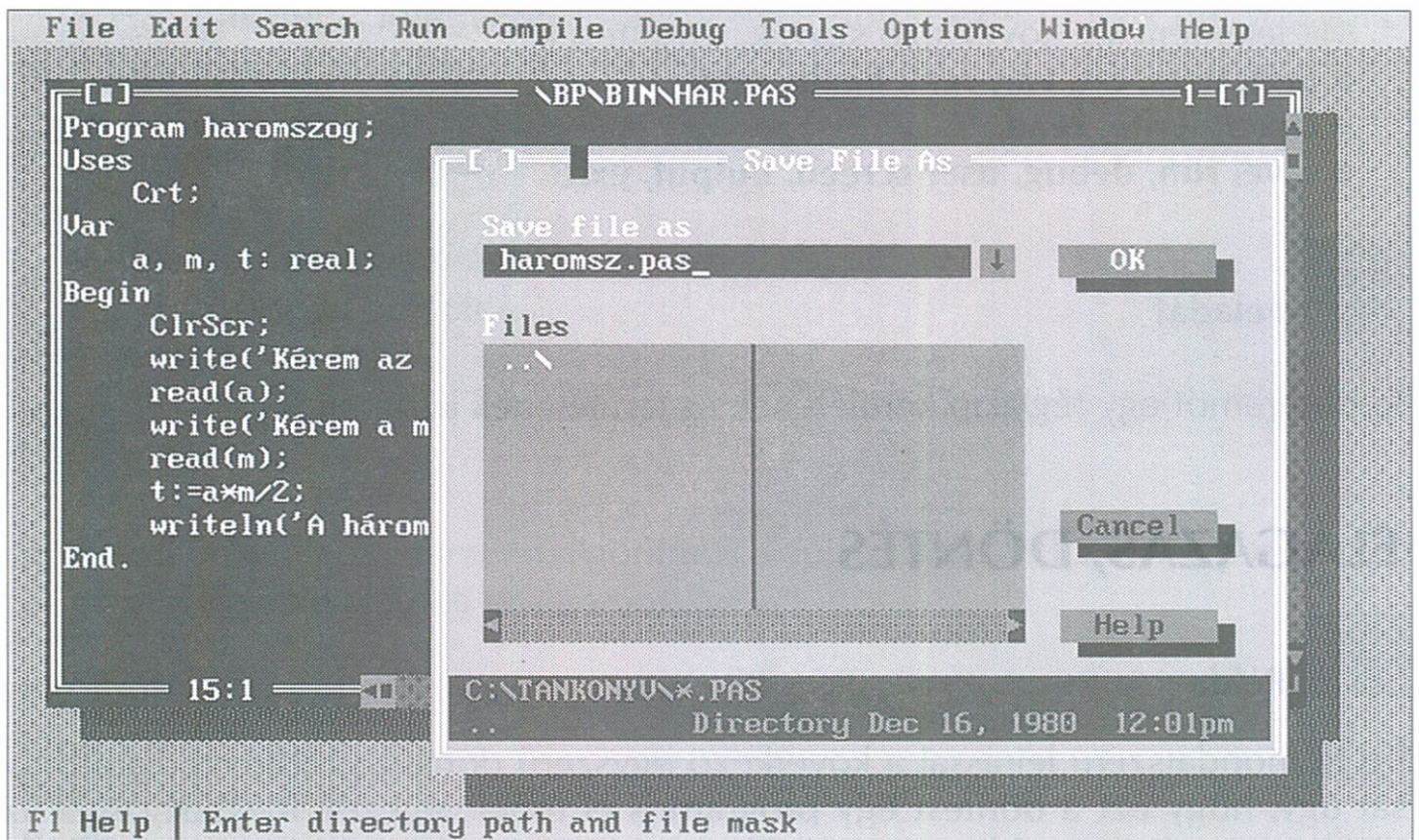
A végrehajtási rész a **Begin**-nel kezdődő és az **End.**-del záródó utasítássorozat. Programodban a **ClrScr** (Clear Screen) a képernyőt törlő eljárás, a **Crt unit**-ban van. A **write('Kérem az alapot');** eljárás kiírja a képernyőre az idézőjelek közötti szöveget. A **writeln** eljárás ugyanezt teszi, de sort is emel. A **read(a)** az a változó értékét olvassa be a billentyűzetről.

A **t:=a\*m/2** nem egyenlet, hanem egy értékadó utasítás, jelentése: **t legyen egyenlő** a jobb oldali kifejezés kiszámolt értékével.

A **writeln('A háromszög területe:',t:8:3);** utasítás a szöveget és a **t** változó értékét írja ki egy sorba. A **t** értékét 8 karakter hosszúságú mezőben 3 tizedes pontossággal kapod meg. Végül az **End.** utasítással zárul a program.

Jegyezd meg, hogy a kis- és nagybetűk felcserélhetők!

Mentsd el a programodat háttértárra! Ezt a **File** menü **Save** parancsával teheted meg. Ha új néven vagy új helyre kívánod menteni az újabb programváltozatot, akkor használd a **Save as...** parancsot!



Írd be a programállomány nevét a **Save file as...** alatti mezőbe! Legyen most az állomány neve `haromsz.pas`, amely maximum 8 karakter lehet a pont előtt. Ne használj ékezetes karaktereket!

Ha a **files** ablakban mutatott könyvtár megfelel a program tárolására, akkor kattints az **OK** gombra (vagy üss **Enter**-t). Ha másik alkönyvtárba akarod menteni, akkor ugorj le a **Tab** billentyűvel a **files** mezőbe, és a szokásos módon válaszd ki a megfelelő alkönyvtárt!

**Néhány megjegyzés:** A párbeszédablakban a mezők, ill. a gombok között előre a **Tab** billentyűvel, hátra a **Shift+Tab**-bal ugrálhatsz. Meghajtót vagy könyvtárat legkönnyebben a **File Change dir...** menüponttal válthatsz. A kezelési szabályok ott is hasonlóak.

Lépj ki a fejlesztőrendszerből! A **File** menü **Exit** pontját kell választanod, vagy az **Alt+X**-et. Ha kilépéskor még nincs elmentve valamelyik ablakban lévő programod utolsó változata, a rendszer figyelmeztet (*név.pas has been modified. Save?*), és lehetőséged nyílik arra, hogy elmentsd.



### Ellenőrző kérdések

1. Hogyan használhatsz ékezetes karaktereket a Turbo Pascal fejlesztőrendszerben, ha programot írsz?
2. Mit tudsz a fejlesztőrendszer programablakáról?
3. Hogyan lehet futtatni a programot és megnézni az eredményt?

4. Mit jelentenek a következő szavak, rövidítések:

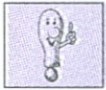
- unit, uses, crt, var, real, begin, end
- clrscr, write, read,
- save, run, debug, user screen, output, exit?



## Feladat

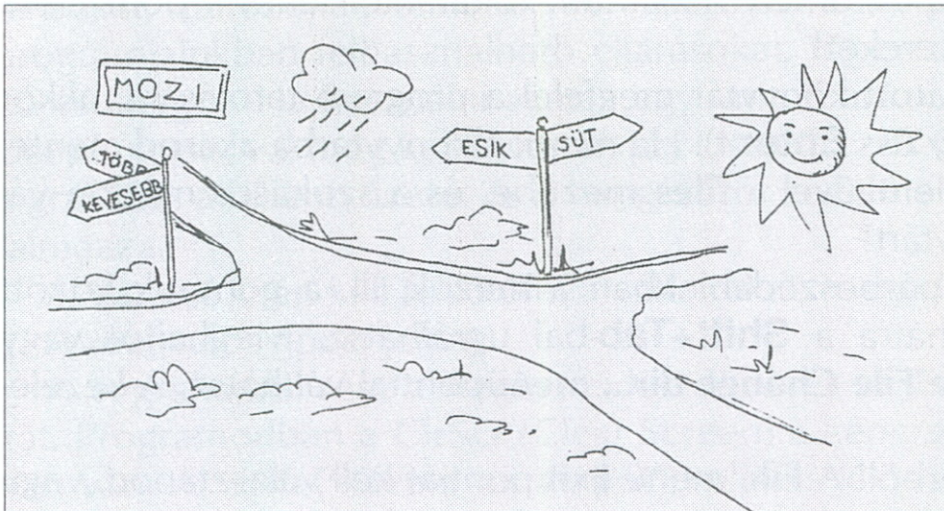
Írj programot egy téglalap kerületének és területének kiszámítására!

# ELÁGAZÁS, DÖNTÉS



## Példa

Írd le mondatszerű leírással a következő egyszerű döntési probléma algoritmusát úgy, hogy ezt a döntést egy program is el tudja végezni: meghallgatjuk az időjárásjelentést, és eldöntjük, mit fogunk csinálni. Ha napos időt jósolnak, kirándulni megyünk, ha várhatóan esni fog, akkor moziba.



### Program

időjárás: egy karakter

Ki: 'Válaszd ki az időjárásnak megfelelő karaktert!'

Ki: 'Sütni fog a nap: s'

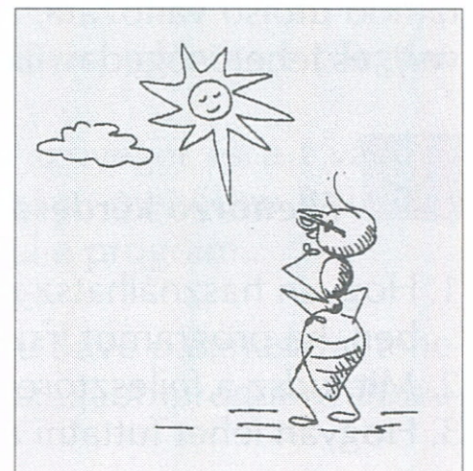
Ki: 'Esni fog az eső: e'

Be: időjárás

**Ha** időjárás= 's' **akkor** Ki: 'Menj kirándulni!'

**Ha** időjárás= 'e' **akkor** Ki: 'Menj moziba!'

**Program vége**



A **Ha ... akkor ... különben ...** szerkezettel döntéseket tudunk megfogalmazni, ennek alakja Pascalban a következő:

- I. **if feltétel then utasítás;**  
    vagy
- II. **if feltétel then utasítás1**  
    **else utasítás2;**



Írj Pascal programot az iménti algoritmus alapján! Indítsd el a fejlesztőrendszert, és gépeld be a következő programot!

```
Program tanacsado;  
Uses Crt;  
Var idojaras: char;  
Begin  
    ClrScr;  
    writeln('Válaszd ki az időjárásnak megfelelő karaktert!')  
    writeln('Sütni fog a nap: s ');  
    write('Esni fog az eső: e ');  
    readln(idojaras);  
    if idojaras='s' then writeln('Menj kirándulni!');  
    if idojaras='e' then writeln('Menj moziba!');  
End.
```

Futtasd le a programot néhányszor! Mentsd el (Save), majd nyisd meg a **File** menü **Open** parancsával!

A programban a kapcsos zárójelek közé magyarázatokat (kommentárokat) helyezhetsz el. Pl. a **ClrScr**; után a {*Letörli a képernyőt.*} szöveget írhatod. Az idojaras **Char** típusú változó, az értéke egy karakter lehet, pl. 's' vagy 'e' betű.

Lásd el magyarázatokkal a programod! Az egyszerű utasítások helyett összetett utasításokat is használhatsz. A **begin** és az **end** közé több utasítást is írhatsz, így összetett utasítást kapsz, amelynek alakja:

```
Begin  
    utasítások  
End;
```

Itt nincs pont az End után, hiszen ez nem a program vége!



Összetett utasítást találsz a következő programban. Mit csinál a program?

**Program** tanacsado2;

**Uses** Crt;

**Var** idojaras: **char**;

penz: **integer**;

**Begin**

**ClrScr**;

**writeln**('Válaszd ki az időjárásnak megfelelő karaktert!')

**writeln**('Sütni fog a nap: s ');

**writeln**('Esni fog az eső: e ');

**read**(idojaras);

**if** idojaras='s' **then writeln**('Menj kirándulni!');

**if** idojaras='e' **then**

**begin**

**writeln**('Hány Ft-od van?');

**read**(penz);

**if** penz>300 **then write**('Menj moziba!') *{Ha a pénzed több mint 300 Ft, menj*

**else write**('Maradj otthon!') *moziba, különben maradj otthon.}*

**end**;

**End.**

Gépeld be, és futtasd le néhányszor, és akkor meglátod, hogy mit csinál a program!



### Ellenőrző kérdések

1. Milyen elágazásformákat ismersz?
2. Mi lehet feltétel egy elágazásban?



### Feladatok

1. Írj programot, amely bekér egy valós számot, és kiírja az abszolút értékét!
2. Írj programot, amely bekér két számot, és kiírja őket növekvő sorrendben!

# HÁTULTESZTELŐ CIKLUS



Emlékszel még a „Gondoltam egy számot...” játékra? Leírtuk a játék algoritmusát is. Itt az ideje, hogy magad is programozz, és játssz a számítógéppel! Lapozz vissza az algoritmushoz, és hasonlítsd össze a következő kódolt formával!

**Program gondol;**

**Uses crt;**

**Var** a, x: **integer;**

**Begin**

**randomize;**

**a:=random(101);** *{Az a értéke egy 0 és 100 közötti véletlen egész szám lesz.}*

**clrscr;**

**writeln('Gondoltam egy számot, találd ki!');**

**repeat** *{Itt kezdődik a ciklus}*

**write('Kérem a tipped!');**

**readln(x);**

**if a>x then writeln('Nagyobbat gondoltam.');**

**if a<x then writeln('Kisebb számra gondoltam.');**

**until a=x;** *{Itt a ciklus vége. Ha a=x, akkor a következő utasítással folytatódik a program.}*

**writeln('Eltaláltad!');**

**End.**

A **repeat ... until** közötti utasításokat addig kell ismételni, amíg  $a=x$  egyszer csak igaz lesz. Ekkor a játékos eltalálta a gondolt számot. A mondatszerű algoritmusleírásnál már megtudtad, hogy ezt a „szerkezetet” hátultesztelő ciklusnak nevezzük.

Alakja Pascal nyelvben:

**repeat**

*ciklusmag utasításai*

**until feltétel**

A ciklusból való kilépés feltételét kell megadnod.

A ciklusmag utasításait a gép egyszer mindenképpen végrehajtja. Ha a feltétel teljesül, igazzá válik, akkor az ismétlés véget ér, és a program a következő utasítással folytatódik.

Akkor lehet nagyon hasznos a ciklus, ha egy utasítássorozatot sokszor kell



egymás után elvégezni. A következő programmal egy karaktert (csillagot) mozgathatsz, irányíthatsz a képernyőn. Gépeld be, és rajzolj vele valamit!



**Program** csillag;

**Uses** crt;

**Var** x, y: integer;  
k: char;

**Begin**

x:=40; y:=12;

clrscr;

write('Írányítsd a csillagot a nyilakkal! Kilépés: Esc');

**repeat**

gotoxy(x,y);

write('\*');

k:=readkey;

if k=#0 then k:=readkey;

if k=#77 then x:=x+1;

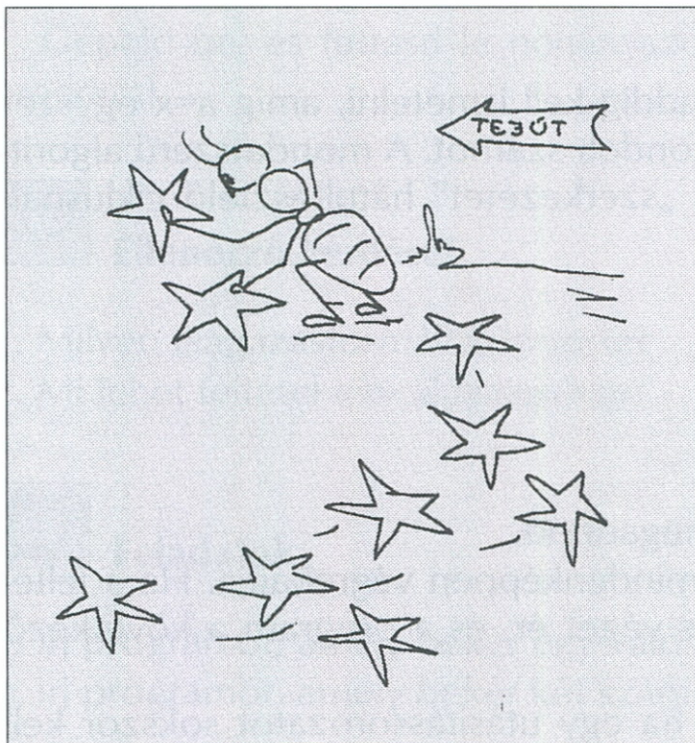
if k=#75 then x:=x -1;

if k=#80 then y:=y+1;

if k=#72 then y:=y -1;

**until** k=#27;

**End.**



A ciklusmag első utasítása a **gotoxy(x,y)**; ezzel a kurzort az x,y karakterkoordinátákra állíthatod. Mivel x=40, y=12, a kurzor vízszintesen a 40. oszlopba, függőlegesen a 12. sorba kerül. Írj ide egy csillagot a következő utasítással: **write('\*')**.

A **k:=readkey**; utasítás hatására a **k** változó értéke a billentyűzeten utoljára leütött karakter lesz. A **if k=#0 then k:=readkey**; utasításra elvileg azért van szükség, mert a vezérlőkarakterekhez (mint pl. a nyilak) két kódszám tartozik, ezek közül az első 0. (Egyébként, ha kihagyod, a program akkor is működik.)

A kurzormozgató nyilakra a második kódszámukkal lehet hivatkozni:  
jobbra nyíl #77; balra nyíl #75; le nyíl #80; fel nyíl #72.

Ha nyilak helyett leírható karaktereket használsz irányításra, egyszerűen tedd idézőjelbe őket! Az első elágazás pl. így nézhetne ki: **if k='j' then x:=x+1**;. Ekkor a **j** betű hatására a csillag jobbra megy.

Az #27 az **Esc** billentyűnek felel meg.

Hogyan lehetne úgy módosítani a programot, hogy radírozni is tudjunk?

## TITKOSÍTSUNK, KÓDOLJUNK!



Az *Elvárásolt jelek* c. részben szó volt a betűhelyettesítéses titkosírásról. A következő program a begépelte szöveget folyamatosan, karakterenként titkosítja, lényegében betűeltolásos módszerrel. Némi különbség azért van az itteni és a betűeltolásos módszer között. Mi az?



**Program** titok3;

**Uses** crt;

**Var** x, y, kodsza: **integer**;  
titkos, karakter: **char**;

**Begin**

x:=5; y:=25;

**clrscr**;

**writeln**('Írd be a szöveget! Kilépés: Esc');

**repeat**

**gotoxy**(x,y);

karakter:=**readkey**;

**write**(karakter);

**gotoxy**(x,y+2);

kodsza:=**ord**(karakter)+3;

titkos:=**chr**(kodsza);

```

write(titkos);
x:=x+1;
until karakter=#27;
End.

```



Az **ord(karakter)** megadja a zárójelben lévő karakter kódját. A **chr(kódszám)** pont az ellenkezőjét csinálja, a kódszámnak megfelelő karaktert adja.

Ez a módszer hány hellyel tolja el az ábécében (ASCII-kódban) a betűket? Az eredeti begépelte karakterhez viszonyítva hová írja ki a program a titkosított karaktert a képernyőn?

Gépeld be a programot, és ellenőrizd, hogy helyesen válaszoltál-e a kérdésekre! Írj a barátodnak (barátnődnek) titkosított üzenetet a program segítségével!



### Ellenőrző kérdések

1. Írd le a hátultesztelő ciklus szerkezetét!
2. Milyenek kell lennie a feltételnek?
3. Mit jelent a **gotoxy(20,10)** utasítás?



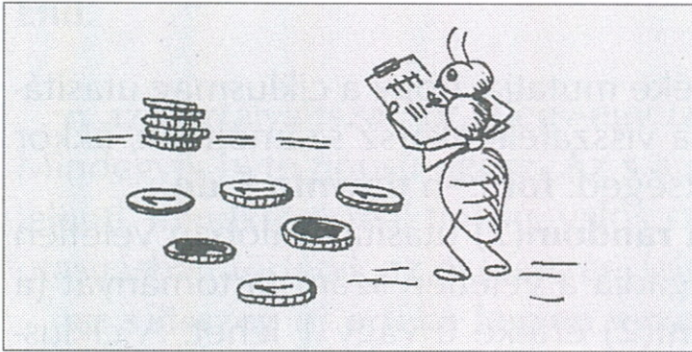
### Feladatok

1. Írj olyan programot, ami kiírja a begépelte szöveget és alája a karakterek kódjait!
2. Írj olyan programot, amely bekér egy byte típusú egész számot (ezt jelöld **i**-vel), beállítja az **i** kódszámnak megfelelő szint a következő utasítással: **text-color(i)**, végül kiírja a következőt: Ennek a színnek a kódja: **i**! Ezt a program mindaddig ismétli, amíg 0-t nem gépelsz.

# SZÁMLÁLÓCIKLUS



Sokszor előre tudjuk, hogy hányszor kell valamilyen tevékenységet elvégezni. A programozásban ilyenkor számlálóciklust célszerű alkalmazni. Már ismered a Pascal véletlenszám-függvényét (**random**). Írj olyan programot, amely a pénzfeldobást szimulálja! Add meg a kísérletek számát! Íme egy megoldás:



```
Program pénzfel;  
Uses crt;  
Var x: byte;  
    iras, fej, n, i: longint;  
    szazalek: real;  
begin  
    fej:=0; iras:=0;  
    clrscr;  
    gotoxy(1,10);  
    write('Hányszor végezzem el a pénzfeldobást?');  
    read(n);  
    randomize;  
  
    for i:=1 to n do  
        begin  
            x:=random(2);  
            if x=0 then fej:=fej+1;  
            if x=1 then iras:=iras+1;  
        end;  
    szazalek:=100*fej/n;  
    writeln('Fejek száma:',fej:9);  
    writeln('Írások száma:',iras:9);  
    writeln('A fejek százalékos aránya:', szazalek:5:2);  
End.
```

Az **x byte** típusú, 0 és 255 közötti egész szám. A **longint** típus szintén egész szám, a kb. -2 milliárd +2 milliárd (-2 147 483 648 ... +2 147 483 647) intervallumban. Azért választottunk ilyen nagy tartományon értelmezett változót, mert így nagyon sok kísérletre van lehetőségünk. (Egy átlagos számítógép, több mint 100 000 pénzfeldobási szimulációt végez másodpercenként.)

**n** az elvégzendő kísérletek számát jelenti, ezt Te adhatod meg a program futása során. Amint a példában látható, a számlálóciklus felépítése a következő:

```
for i:=1 to n do  
  begin  
    ciklusmag utasításai  
  end;
```

**i** a ciklusváltozó, egész szám, ennek értéke mutatja, hogy a ciklusmag utasításait hányszor hajtotta végre a program. Ha visszafelé akarsz számoltatni, akkor az első sor helyett a következőre lesz szükséged: **for i:=n downto 1 do**.

A **randomize** utasítás azért kell, hogy a **random(2)** utasítás valóban véletlen számot adjon. A zárójelben álló szám megadja a véletlen szám tartományát (a felső határ nem fordul elő), azaz a **random(2)** értéke 0 vagy 1 lehet. A ciklusmagban a pénzfeldobást szimuláltatod, és összeszámláltatod a fejek (ez a 0) és az írások (ez az 1) számát. (A két **if ... then ...** utasítást egyetlen **if ... then ... else ...** utasítással is helyettesítheted.)

Végül, kiszámoltatod a fejek százalékos arányát az összes kísérlethez viszonyítva, majd kiíratod az eredményeket.

Kísérletezz minél többet, és írd le az eredményeket! Először kis n-nel (10–100) próbálkozz, majd egyre nagyobbal (10 000–1 000 000). Mit tapasztalsz?



Nézzünk egy másik példát! Gyakori feladat, hogy számok összegét vagy átlagát kell kiszámolni. Pl. egy mérésorozat eredményeinek számtani közepét akarjuk kiszámolni, vagy egy-egy tantárgyból szeretnénk megkapni az osztály tanulmányi átlagát. A következő program erre ad megoldást:

```
Program atlag;  
Uses crt;  
Var a, s: real;  
    n, x, i: byte;
```

```
Begin  
  clrscr; s:=0;  
  write('Kérem az osztálylétszámot!');
```

```

read(n);
for i:=1 to n do
begin
  write('Kérem a következő osztályzatot!');
  read(x);
  s:=s+x;
end;
a:=s/n;
writeln('Az átlag:',a:5:2);
End.

```

**n** az osztálylétszámot, **x** egy-egy tanuló osztályzatát jelenti, **i** a ciklusváltozó. Mindegyik byte típusú egész. Az **s** a pillanatnyi összeget, az **a** az osztályátlagot jelenti, mindkettő real típusú, valós szám. A ciklusmagban lévő **s:=s+x** értékadó utasításban történik az összegzés. Jelentése:

**az s összeg új értéke legyen egyenlő s előző értéke + az (éppen bekért) x szám összegével.**

Az átlagot úgy kaphatod meg, hogy az osztályzatok összegét elosztod az osztályzatok (vagyis a tanulók) számával (**a:=s/n**).



### Ellenőrző kérdések

1. Milyen egy számlálóciklus?
2. Milyen egy visszafelé számláló ciklus?
3. Milyen utasítással hoznál létre véletlen számot (egészset) az 1–90 tartományban?



### Feladatok

1. Írj olyan programot, amely 1-től n-ig kiírja a számokat, négyzetüket és négyzetgyöküket!  $x$  négyzete **sqr(x)**, a gyöke **sqrt(x)**!
2. Írj lottósorsolást szimuláló programot!
3. Írj olyan programot, amely beolvas egy karakterláncot (szöveget, mondatot) és kiírja fordítva! Segítség: A **mondat** legyen egy **string** típusú változó, amely maximum 255 karakter lehet. Használhatod a **length(mondat)** függvényt, amely a **mondat** karaktereinek a számát adja (a hosszát). Végül a **mondat[i]** utasítás a **mondat** i-edik karakterét adja.

# RAJZOLÁS

Ahhoz, hogy rajzolni tudj, szükséged lesz a **graph unit**-ra, amelyet a **uses graph** utasítással nyithatsz meg. Ezután a *karakteres képernyőről* át kell váltanod a *grafikus képernyőre* az **initgraph** utasítással! Ehhez három adat kell:

- a grafikus meghajtó típusa; ezt a program automatikusan felismeri az **i:=detect**; utasítás hatására; ( az **i integer** típusú változó),
- a grafikus üzemmód; erre alkalmazhatod a **j integer** változót; ettől függ a felbontás (ezzel egyelőre nem kell törődnöd, mert a **j-t** a **detect** utasítás automatikusan beállítja a maximális felbontásnak megfelelő értékre.);
- végül meg kell adnod a grafikus üzemmód használatához szükséges `\bgi` kiterjesztésű állományok elérési útvonalát karakterfüzérként, idézőjelek között (pl. 'C:\TP\bgi').

Az utasítás pontos alakja (ha a \*.bgi állományok a C:\TP\bgi könyvtárban vannak): **initgraph(i, j, 'c:\tp\bgi')**.

Ha befejezted a rajzolást, zárd be a grafikus képernyőt a **closegraph** utasítással!

Összefoglalva a szükséges utasításokat:

```
uses graph;  
var i, j: integer;
```

```
begin  
  i:=detect;  
  initgraph( i, j, 'c:\tp\bgi' );
```

utasítások ...

```
closegraph;  
end.
```



Próbáld ki tudományodat! A következő egyszerű program fehér pontot, piros kört és zöld vonalat rajzol.

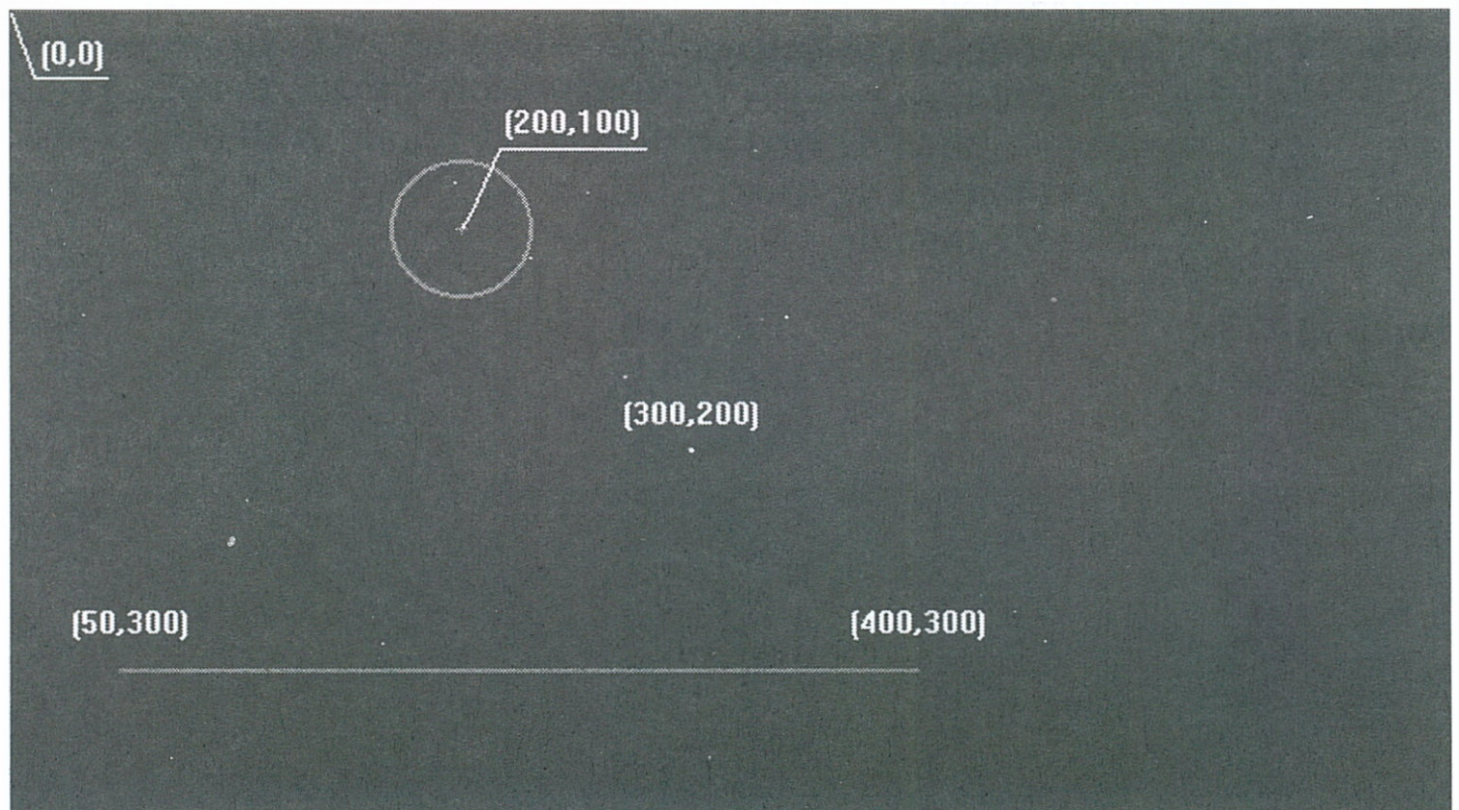
```
Program rajz1;  
Uses graph;  
Var i, j: integer;  
Begin  
  i:=detect;
```

```

initgraph(i, j, 'c:\tp\bgi');
putpixel(300,200,15);
setcolor(red);
circle(200,100,30)
setcolor(green);
line(50,300,400,300);
readkey;
closegraph;
End.

```

A **putpixel(300,200,15)** egy fehér pontot rajzol a vízszintesen 300, függőlegesen 200 koordinátájú helyre. A 15 a fehér szín kódszáma. A **setcolor(red)** utasítással a rajzolószínt adhatod meg. A **circle(200,100,30)** 30 egység sugarú kört rajzol (200,100) koordinátájú középponttal. A **line(50,300,400,300)** egyenes szakaszt húz, az (50,300) ponttól az (400,300) pontig.



Most pedig egy egyszerű játék következik! A következő program segítségével egy nyomvonalat rajzoló mozgó pontot irányíthatsz. Különböző irányokban úgy szabályozhatod a pont sebességét a kurzormozgató nyilakkal, mint a jégpályán csúszó korongra ütétnél. Ne engedd, hogy kiszaladjon a képernyőről a „jégkorong”! Ha mégis kicsúszna, az **Esc** billentyűvel leállíthatod a programot.



```

Program jegpalya;
Uses crt, graph;
Var i, j: integer;
    a, b, dx, dy: integer;
    x, y: real;
    k: char;
Begin
x:=100; y:=100; dx:=0; dy:=0;
i:=detect;
initgraph(i, j, 'c:\tp\bgi');
repeat
    a:=trunc(x); b:=trunc(y);
    putpixel(a,b,15);
    x:=x+dx*0.001; y:=y+dy*0.001;
    if keypressed then
        begin
            k:=readkey;
            if k=#0 then k:=readkey;
            if k=#77 then dx:=dx+1;
            if k=#75 then dx:=dx-1;
            if k=#80 then dy:=dy+1;
            if k=#72 then dy:=dy-1;
        end;
until k=#27;
closegraph;
End.

```



A **k:=trunc(x)** függvény **integer-real** átalakítást végez (egészrész-függvény). Egy kis fantáziával könnyedén továbbfejlesztheted a játékot és a programot. Íme, néhány ötlet:

- Az új pont kirajzolása előtt töröld le a régit (rajzold ki 0 kódszámú színnel)! Így a pont nem húz vonalat.
- Pattanjon vissza a pont a szélekről!
- Rajzolj a pont helyett kis kört!



## Ellenőrző kérdések

1. Milyen utasítások szükségesek a grafikus üzemmód használatához?
2. Hogyan tudsz zöld kört rajzolni a monitor közepére?



## Feladatok

1. Készítsd el a következő rajzokat:
  - Rajzolj koncentrikus (azonos középpontú) köröket a képernyő közepére!
  - Rajzolj egy kört, amelyet két billentyűvel nagyítani, ill. kicsinyíteni lehet!
  - Rajzolj egy koordináta-rendszert!
  - Rajzolj 1000 pontot véletlenszerűen a monitorra!

## ASCII táblázat (0-31-ig vezérlőjelek)

0 NUL	32 <i>szóköz</i>	64 @	96 `
1 SOH	33 !	65 A	97 a
2 STX	34 "	66 B	98 b
3 ETX	35 #	67 C	99 c
4 EOT	36 \$	68 D	100 d
5 ENQ	37 %	69 E	101 e
6 ACK	38 &	70 F	102 f
7 BEL	39 '	71 G	103 g
8 BS	40 (	72 H	104 h
9 HT	41 )	73 I	105 i
10 LF	42 *	74 J	106 j
11 VT	43 +	75 K	107 k
12 FF	44 ,	76 L	108 l
13 CR	45 -	77 M	109 m
14 SO	46 .	78 N	110 n
15 SI	47 /	79 O	111 o
16 DLE	48 0	80 P	112 p
17 DC1	49 1	81 Q	113 q
18 DC2	50 2	82 R	114 r
19 DC3	51 3	83 S	115 s
20 DC4	52 4	84 T	116 t
21	53 5	85 U	117 u
22 SYN	54 6	86 V	118 v
23 ETB	55 7	87 W	119 w
24 CAN	56 8	88 X	120 x
25 EM	57 9	89 Y	121 y
26 SUB	58 :	90 Z	122 z
27 ESC	59 ;	91 [	123 {
28 FS	60 <	92 \	124
29 GS	61 =	93 ]	125 }
30 RS	62 >	94 ^	126 ~
31 US	63 ?	95 _	127 *

## ASCII táblázat

128	Ç	160	á	192	Ł	224	α
129	ü	161	í	193	ł	225	β
130	é	162	ó	194	Ƨ	226	Γ
131	â	163	ú	195	┆	227	Π
132	ä	164	ñ	196	—	228	Σ
133	à	165	Ñ	197	†	229	σ
134	å	166	ª	198	‡	230	μ
135	ç	167	º	199	‡	231	τ
136	ê	168	¿	200	ℒ	232	Φ
137	ë	169	┌	201	℞	233	Θ
138	è	170	┐	202	⌚	234	Ω
139	ï	171	½	203	π	235	δ
140	î	172	¼	204	‡	236	∞
141	ì	173	ı	205	=	237	∅
142	Ä	174	«	206	‡	238	ε
143	Å	175	»	207	⌚	239	∩
144	É	176	▒	208	⌚	240	≡
145	æ	177	▒	209	Ƨ	241	±
146	Æ	178	■	210	π	242	≥
147	ô	179		211	ℒ	243	≤
148	ö	180	┆	212	ℒ	244	
149	ò	181	‡	213	Ƒ	245	
150	û	182	‡	214	π	246	÷
151	ù	183	π	215	‡	247	≈
152	ÿ	184	Ƒ	216	‡	248	°
153	Ö	185	‡	217	┘	249	•
154	Ü	186		218	┌	250	·
155	ç	187	┐	219	■	251	√
156	£	188	┘	220	■	252	n
157	¥	189	⌚	221	■	253	²
158	℞	190	┘	222	■	254	■
159	f	191	└	223	■	255	

Kiadja a Műszaki Könyvkiadó

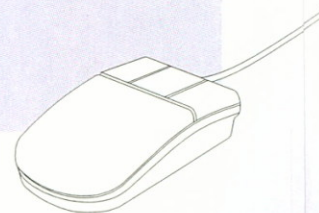
Felelős kiadó: Bérczi Sándor  
ügyvezető igazgató

Felelős szerkesztő: Szloboda Tiborné  
Szerkesztő: Erényi Tamás  
Műszaki vezető: Abonyi Ferenc  
Műszaki szerkesztő: Németh Csongor  
Terjedelem: 10,05 (A/5) ív  
Azonosító szám: MK-07091

Nyomta az Oláh Nyomda és Kiadó  
Felelős vezető: Oláh Miklós

A Műszaki Könyvkiadó új tankönyvcsaládja az alapszinttől a felhasználói szintig vezeti be a tanulókat az informatika világába. 10 éves korosztálytól kezdve bármikor elkezdhető az iskola lehetőségeihez és a tanulófelkészültségéhez igazodva.

A tankönyvhöz ingyenes tanmenetjavaslat készül. Valamennyi évfolyam számára feladatgyűjteményt is megjelentetünk, melynek segítségével gyakorolhatják a tankönyv ismeretanyagát.



Műszaki Könyvkiadó  
1033 Budapest, Szentendrei út 89–93.



MK-07091

ISBN 963-16-2343-2

