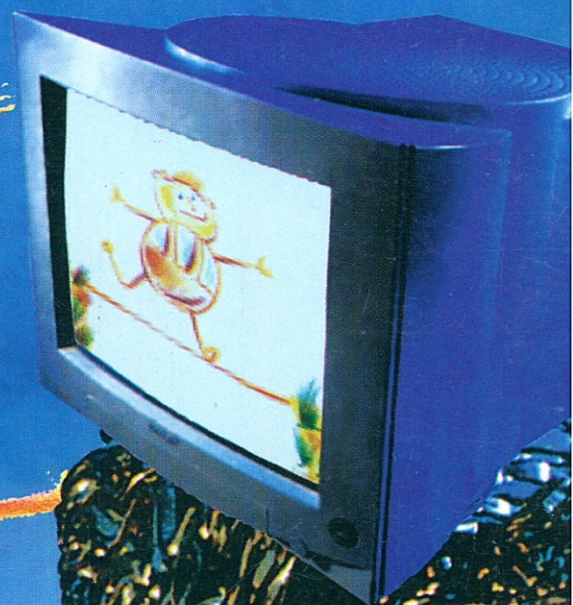


Horváth Tibor

Számítástechnika I. iskolásoknak




Informatika

Nemzeti Tankönyvkiadó

HORVÁTH TIBOR

SZÁMÍTÁSTECHNIKA
ISKOLÁSOKNAK I.

Nemzeti Tankönyvkiadó, Budapest

A tankönyv engedélyszáma: 59 476/1994. IX.

Bírálták:

Ódor Péter

főiskolai adjunktus

Zsinkó Erzsébet

főiskolai adjunktus

Haramia László

szakértő

Felelős szerkesztő:

Szalay Sándor

ISBN 963 18 8023 0

Nemzeti Tankönyvkiadó Rt.

A kiadásért felel: dr. Ábrahám István vezérigazgató

Raktári szám: 539

Műszaki vezető: Babicsné Vasvári Etelka igazgatóhelyettes

Tipográfus: Szalay Sándor

Műszaki szerkesztő: Wéber Andrea

Anyanyelvi lektor: Szilágyi Edit

Utánnnyomásra előkészítette: Borsányi Katalin

Terjedelem: 16,2 (A/5) ív

2. kiadás, 1997

Nyomtatta és kötötte a Kaposvári Nyomda Kft. – 170470

Felelős vezető: Mike Ferenc

Kedves Olvasó!

Érdekel a számítástechnika, de a megvásárolt IBM számítógépeddel játékon kívül nem tudsz mit kezdeni. Szeretnéd megtanulni a gép használatát, de a hozzá megvásárolt könyvekben sok ismeretlen fogalommal találkozol. Végre az iskolában is megvettek ilyen gépeket, s órákon foglalkoztok vele, de a tanári magyarázatokon kívül nincs miből tanulnod.

Neked írtam a könyvet!

Neked, aki elhatározta, hogy megtanulja az alapoktól kezdve a számítógép felépítését és működését is. Ehhez azonban még előismereteid az elektromosság területéről. a könyv is csak *tájékoztató jellegű anyagot* témakörökből. Az ilyen anyagrészeknél a cím alatt találsz meg. Ezeket a részeket figyelmesen olvasd tananyag.



nincsenek megfelelő Mindezek alapján ez tartalmaz ezekből a az itt is látható jelet el, de nem kötelező

A fejezetekben nagyon sok angol nyelvű kifejezést találsz. Ezek mellé – angolul nem tudók számára – dőlt betűkkel odaírtam az adott szó magyar kiejtését, a pontos fordítás nem mindig jelenti ugyanazt, így a kifejezések jelentését a szövegkörnyezetből ismerheti meg minden érdeklődő. Például: **BACKUP** (*bekáp*). A kiejtés leírása csak megközelítően azonos a hivatalos angol kiejtési szabállyal, ezért ha módodban áll, kérj segítséget egy angoltanártól.

A számítástechnika alapjai a gép bekapcsolása után a *DOS*-ismeretekkel kezdődnek. A könyvben az *MS-DOS 6.2* parancsaival ismerkedhetsz meg. Ennek megfelelő elsajátítása után lehetőséged nyílik egy, a *DOS*-parancsok és a számítógép használatát megkönnyítő keretrendszer, a *NORTON COMMANDER 4.0* használatának megismerésére is.

Levelet szeretnél írni a barátodnak, barátnődnek! Mindezt egy szövegszerkesztővel is megteheted, hiszen ezt is megtanulhatod a könyvből a *NORTON EDITOR* segítségével.

Gyakran kapsz barátaitól lemezen programokat, de hallottál már a számítógépvírusokról? Kiirthatók a számítógépből? Mindezekre a kérdésekre választ kapsz a *VÍRUSOK* című fejezetben. A *SCAN-CLEAN*, illetve a *CHKVIR* víruskereső és -irtó programok leírásai segítséget adnak a gépedbe jutott vírusok eltávolításához.

A felsoroltak elsajátítása után biztos felmerül benned a kérdés: Segíthet a számítógép a matematika házi feladat elvégzésében is? A *Quick BASIC* fejezetben megtanulhatod az egyszerűbb műveletvégzést is, de a *Quick BASIC* alapismeretek elsajátítása után már kisebb programokat is meg tudsz írni.

Remélem, sok örömed telik majd a számítástechnikában, s ehhez kívánok sok kitartást, sikert, s nem utolsósorban hasznos időtöltést:

a szerző

A SZÁMÍTÁSTECHNIKA TÖRTÉNETE



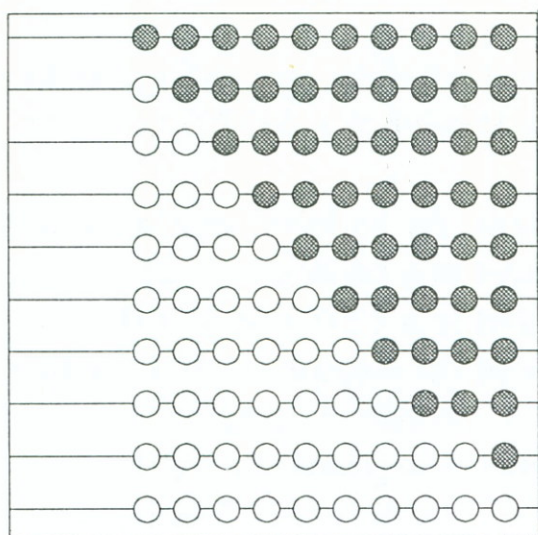
Az emberiség fejlődéstörténetéből tudjuk, hogy az emberré válás döntő mozzanata volt, amikor az ősember elkezdte fejleszteni szerszámaikat. A 8–20 ezer éves csontmaradványok némelyikén látható rovátkák valószínűleg számolási célból készültek. Talán az első sziklarajzokat tekinthetjük az információ egyfajta tárolásának is esztétikai jelentőségük mellett.

A számítástörténetben lényeges változást jelentett az írás megjelenése. 4000 évnél is régebbiek a babilóniai számolótáblák. Ezekkel a négy alapműveletet kitűnően el lehetett végezni.

Az egyiptomi piramisok belső falának rajzai között is találhatunk táblázatokat, sőt *számítási szabálykönyvet* is, de ezekkel csak a papok tudtak számolni.

A régen használt számírásokkal, a régi számrendszerekkel nagyon bonyolult volt számításokat végezni. Gondoljunk csak a római számokra! A régi számrendszerek *maradványait* örzi még ma is az idő és a szögek felosztása (60-as számrendszer), az év hónapokra tagolása (12-es számrendszer).

A mindennapi számításokhoz valami könnyebben kezelhető eszközre volt szükség. Hérodotosz mintegy 2500 éve azt írta, hogy az egyiptomiak *vonalakat húztak, és rajtuk kavicsokkal számoltak*. (A kavics szó latin megfelelője a *calculus*, ebből a szóból ered a *kalkulátor* neve is.)



Az abakuszt már ismerheted az első osztályból is.

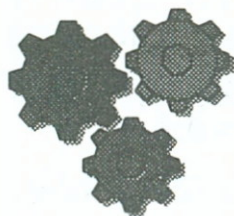
Kínában a számolótáblán nem kavicsokat, hanem vékony drótokon golyókat mozgattak. Ezt a segédeszközt csak a XVII. században ismerte meg a világ, s *abakusz* néven a mai napig használják.

Valószínű, hogy a számolást elősegítő eszközök egymástól függetlenül fejlődtek ki Európa és a Távol-Kelet országaiban.

Az arab kereskedők Indiában ismerték meg a ma *arab számok*ként ismert számjegyeket, és ezeket Európában is elterjesztették. *Muhammad ibn Muza al-Kvarizmi* a IX. században írt könyvében – latin címe: *Algoritmi dicit* – számolási módszereket, eljárásokat adott meg.

A tudomány fejlődésével egyre több és bonyolultabb számolási műveletet lehetett elvégezni, elsősorban a csillagászat és a térképészet számára. 1623-ban Schi-

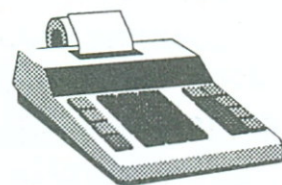
kard (*sikárd*), majd 1642-ben Pascal (*pászkál*) készített fogaskerekes összeadó-kivonó gépet.



Ezekben a gépekben a fogaskerekek tárolták, különböző karok (mechanizmusok) vezérelték a műveletvégzést. (Többszörös összeadással szorzást, többszörös kivonással osztást lehetett elvégezni.)

1671-ben Leibnitz (*lejbnic*) mind a négy alapművelet elvégzésére alkalmas gépet szerkesztett.

A XIX–XX. század motoros meghajtású számológépei elvükben nem különböztek a kézi meghajtású gépektől, csupán gyorsabbak voltak. E számológépeket csak a mai modern, zseb-
ben is hordható elektromos számológépek szorították ki az irodák világából.



1679-ben jutottak a matematikusok arra a következtetésre: a mechanikus rendszer legnagyobb hibája, hogy nem – vagy csak nagyon nehezen – lehet vele műveletsorokat, számítási programokat, eljárásokat végrehajtani. Az okot a fogaskerekes rendszerben találták meg. Túl sok fog vitte tovább az információt!

A tízes számrendszerrel (vagyis 10 fogú fogaskerekes rendszerrel) az említett problémát nem lehetett megoldani. Egyszerűsíteni kellett az információ átvitelét.

A megoldást a kettes számrendszer használatában látták. Arról, hogy ez milyen módon segítette a számítógépek tökéletesítését, a későbbiekben még lesz szó.

A XVIII. században Babbage (*bebecs*) egy lyukakkal ellátott szalaggal vezérelt számológépet készített Angliában. A gépet azonban nem sokáig használták, jelentőségét csak a XX. század számítógép-tervezői ismerték fel.

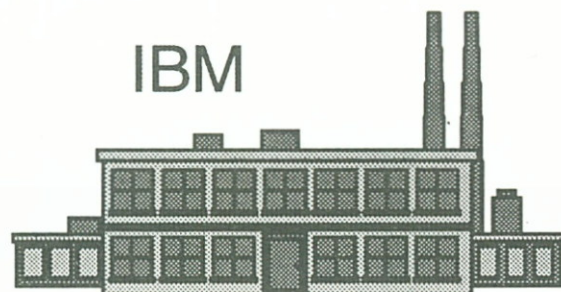
1890-ben népszámlálást tartottak az Amerikai Egyesült Államokban. Az adatok feldolgozását Hollerith (*holerit*) a Babbage által használt lyukkártyarendszer segítségével 3 hónap alatt elvégezte. (Az adatfeldolgozás kézi eljárással 500 embernek legalább két évi munkájába került volna.)



A sikeres főpróbán felbuzdulva – Hollerith – megalapította adatfeldolgozással foglalkozó cégét 1896-ban. 1911-ben még további két hasonló beállítottságú vállalkozás csatlakozott hozzá.

Az így létrejött cég az International Business Machines Company (*intönésönl biznisz mesinz cámpöni*) nevet vette fel, melyből az ismert nevű IBM (*áj bi em*) vállalat fejlődött ki.

A XX. század első felében a rádiózás fejlődése indította be az elektronika fejlődését. Az amerikai Atanasoff volt az első, aki számítógépet épített elektronikus elemekből. 1943 és 1945 között épült meg az ENIAC számítógép a philadelphiai (*filadelfiai*)



egyetemen. A gép 30 méter hosszú, 3 méter magas, 1 méter széles és 30 tonna tömegű volt. A megépítése 10 millió dollárba került. A szerkezet egy másodperc alatt 333 szorzást és 5000 összeadást tudott elvégezni, amikor jó volt. A sok alkatrész közül gyakran meghibásodott valamelyik. Programozása igen nehéz volt, mivel huzalokkal és dugaszolásokkal kellett létrehozni a műveletsorok elvégzéséhez szükséges programot.

Neumann (*najmann*) János az ENIAC továbbfejlesztésén dolgozva rájött, hogy a számítások algoritmusát (a programot) ugyanúgy is lehet tárolni a gépben, mint magukat az adatokat.

A Neumann által megfogalmazott logikai elveknek és követelményeknek megfelelően készítik még a mai számítógépek jelentős részét is. Neumann János egy 1946-ban tartott előadásában – napjainkban is érvényes módon – fogalmazta meg *a modern számítógépek 5 legfőbb alapelvét*:

1. **Soros működés:** teljesen elektronikus számítógép. A gép egyszerre csak egy műveletet vesz figyelembe és hajt végre.
2. **Kettes számrendszer használata:** elektromos berendezések gyors működéséhez nem célszerű a tízes számrendszert használni.
3. **Belső memória használata:** ebben tárolhatók – a számítógép gyors működése alatt – a részeredmények. Egy bizonyos műveletsorozatot a gép automatikusan is el tud végezni.
4. **Tárolt program elve:** a számítások menetére vonatkozó utasítások kifejezhetők számmal, azaz adatként kezelhetők. Így ezek éppúgy a belső memóriában tárolhatók, mint bármilyen adat.
5. **Univerzális gép:** a számítógép különféle feladatainak elvégzéséhez nem kell speciális gépeket készíteni. Az 1930-as években Turing (*tjúring*) angol matematikus mutatta ki, hogy az olyan gép, amely el tud végezni néhány alapvető műveletet, elvileg bármely más számítás végrehajtására is képes. Neumann ezért javasolta, hogy a számítógépek ilyen univerzális Turing-gépek legyenek.

Számítógép-generációk

Az első számítógép megépítése óta több alapvető változás történt a számítógépek felépítésében. Az egymástól lényegesen különböző műszaki megvalósítási módokat generációknak hívjuk.

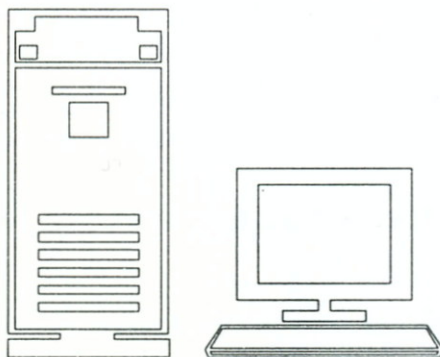
Az **első generációs számítógépek**et nagy energiaigény mellett kis teljesítmény (10 ezernél kevesebb művelet másodpercenként), kis tárolókapacitás, alacsony megbízhatóság (néhány órás működés) jellemezte. A gépet a memóriájában (gépi kódban) programozták. Az 1960-as évekig használták ezeket.

A **második generációs számítógépek** kevesebb energiát fogyasztottak, megbízhatóságuk jobb volt, mint az elődeiké. Teljesítményük elérte az 500 ezer műveletet másodpercenként. A programot speciális gépnyelveken (Algol, Basic) lehetett írni. Az 1970-es évek elejéig működtek ezek a gépek.

A **harmadik generációs számítógépek** már egy íróasztalon is elfértek. A számítógépek terminálok (billentyűzet, monitor), hálózaton keresztül érték el. A gépek teljesítménye elérte az 1 millió műveletet másodpercenként.

A **negyedik generációs számítógépek** a jelen gépei. Mikroprocesszorok és chip (*csip*)-ek vezérlik. A számítógépekhez rendkívül sok eszköz csatlakoztatható. Megjelentek a termelésben, a hétköznapi munkában, az oktatásban, az élet szinte minden területén.

Az **ötödik generációs számítógépek** több processzorral működő – nem Neumann-elvű – számítógépek. 1987-ben kezdték el fejlesztésüket.



SZÁMRENDSZEREK



Decimális (10-es) számrendszer

A mindennapi munkánk során szinte mindig ebben a számrendszerben dolgozunk. Így vannak kiírva az árak az üzletekben, s az órákon is ebben a számrendszerben számolunk.

Egy számrendszer alapszámát az adott számrendszerben használható számjegyek száma adja meg. A 10-es számrendszerben összesen 10 darab számjegyet írhatunk le a 0-tól a 9-esig.

Mit jelent a decimális számrendszerben leírt 478?

1000	100	10	1	<i>helyiértékek</i>
	4	7	8	<i>alakiértékek</i>
	$4 \cdot 100$	$+ 7 \cdot 10$	$+ 8 \cdot 1$	<i>valódiértékek</i>
	$= 478$			

A leírt módszer nemcsak a decimális számrendszer esetében alkalmazható. Ha más számot választunk alapszámnak, a szám értékét ugyanígy megkaphatjuk.

A helyiértékeket szorzat alakjukból egy egyszerűbb leírással is megadhatjuk. A szám után felső indexként odaírjuk, hányszor vesszük az adott számot szorzótényezőként. Például: $1000 = 10 \cdot 10 \cdot 10 = 10^3$; $100 = 10 \cdot 10 = 10^2$

Az ilyen alakban leírt számokat hatványoknak nevezzük.

Feladatok

1. Írd fel a helyiértékeknek megfelelő összeg alakban a következő számokat:

$$32 = \dots\dots\dots$$

$$547 = \dots\dots\dots$$

$$8016 = \dots\dots\dots$$

$$54\ 004\ 002 = \dots\dots\dots$$

2. Mennyit érnek a következő, hatvány alakban felírt számok?

$10^2 = \dots\dots\dots$	$3^2 = \dots\dots\dots$	$7^1 = \dots\dots\dots$
$8^3 = \dots\dots\dots$	$5^3 = \dots\dots\dots$	$7^2 = \dots\dots\dots$
$6^4 = \dots\dots\dots$	$2^7 = \dots\dots\dots$	$12^3 = \dots\dots\dots$

Bináris (2-es) számrendszer

Ebben a számrendszerben tehát csak két számjegyet írhatunk le a 0-át és az 1-et. Ezeket az angol **BI**nary **di**gi**T** (kétértékű jegy) betűiből **BI**T-nek nevezzük.

Vizsgáljuk meg, mit jelent a 11 100 101 kettes számrendszerben felírt szám?

2^7	2^6	2^5	2^4	2^3	2^2	2^1	1	<i>Helyiértékek</i>
1	1	1	0	0	1	0	1	<i>Alakiértékek</i>
128	64	32	16	8	4	2	1	<i>A helyiértékek 10-es számrendszerbeli megfelelője</i>
128 +	64 +	32 +	0 +	0 +	4 +	0 +	1 =	229 <i>Valódiérték a 10-es számrendszerben</i>

Rövidebben leírva:

$$11\ 100\ 101_{(2)} = 229_{(10)}$$

A leírt példából is látszik, hogy a decimálisan három helyiértéken felírt szám binárisan több helyet igényel.

Írjuk át a 142 tízes számrendszerben felírt számot kettes számrendszerbe!

$$142_{(10)} = \dots\dots\dots_{(2)}$$

Ebben a számban 128-as helyiértékű számunk van: 1 Maradt: 14.

A 14-ben már csak 8-as helyiértékű számunk van: 1 Maradt: 6.

A 6-ban van 4-es helyiértékű számunk: 1 Maradt: 2.

A 2-ben van 2-es helyiértékű számunk: 1 Maradt: 0.

Írjuk be a helyiérték-táblázatba a kapott számokat, majd az üresen maradt helyeket 0-val pótoljuk.

128	64	32	16	8	4	2	1
1	0	0	0	1	1	1	0

A 142 decimálisan felírt szám kettes számrendszerbeli alakja: $10\ 001\ 110_{(2)}$

$$142_{(10)} = 10\ 001\ 110_{(2)}$$

Feladatok

3. Írd át decimális számrendszerbe a következő bináris számokat!

$$10\ 110_{(2)} = \dots\dots\dots_{(10)}$$

$$11\ 110_{(2)} = \dots\dots\dots_{(10)}$$

$$10\ 000_{(2)} = \dots\dots\dots_{(10)}$$

4. A Függelékben található karakterkészlet-táblázatból keresd meg az **IBM** szó betűinek kódszámát, majd írd át bináris számrendszerbe azokat! Használd a következő oldalom lévő helyiérték-táblázatot!

$$\mathbf{I} \text{ betű kódja} = \dots\dots_{(10)} = \dots\dots\dots_{(2)}$$

$$\mathbf{B} \text{ betű kódja} = \dots\dots_{(10)} = \dots\dots\dots_{(2)}$$

$$\mathbf{M} \text{ betű kódja} = \dots\dots_{(10)} = \dots\dots\dots_{(2)}$$

128	64	32	16	8	4	2	1
...
...
...

5. Mennyit ér tízes számrendszerben az a kettes számrendszerbeli szám, melyet 8 helyiérték mindegyikén csupa 1-es jelöl?

$$11\ 111\ 111_{(2)} = \dots\dots\dots_{(10)}$$

Más számrendszerek

Az említett decimális, illetve bináris számrendszeren kívül a számítástechnikában még oktális (8-as) és hexadecimális (16-os) számrendszereket is alkalmaznak. Milyen számjegyek írhatók egy-egy helyiértékre:

oktális számrendszerben :

hexadecimális számrendszerben :

A hexadecimális számrendszerben egy helyiértéken a decimálisan kétjegyű számokat az *ABC* nagybetűivel helyettesítjük:

$$10 = A; \quad 11 = B; \quad 12 = C; \quad 13 = D; \quad 14 = E; \quad 15 = F$$

A számítástechnikában egy adott számrendszerben leírt számban egy kisbetűvel is jelölhetjük a számrendszer alapszámát aszerint, hogy *bináris*, *oktális*, vagy *hexadecimális* számokról van szó. A számrendszer nevének első betűjét írjuk a szám után. Így az előzőekben leírt szám

$$142_{(10)} = 10\ 001\ 110_{(2)} \quad \text{helyett rövidebben } 142d = 10\ 001\ 110b \text{ is írható;}$$

$$11\ 100\ 101_{(2)} = 229_{(10)} \quad \text{helyett rövidebben } 11\ 100\ 101b = 229d \text{ is írható.}$$

Feladat

6. A tanultak szerint készítsd el a hexadecimális számrendszer helyiérték-táblázatát 3 helyiértékre, majd keresd meg a karakterkód-táblázatban a keresztneved betűinek kódszámát! Írd át a kapott számokat a hexadecimális számrendszerbe!

Betű	Helyiértékek		
.....	_____	_____	_____
.....
.....
.....
.....
.....
.....
.....

A keresztneved hexadecimálisan leírt alakja :

Az információátvitel egységei

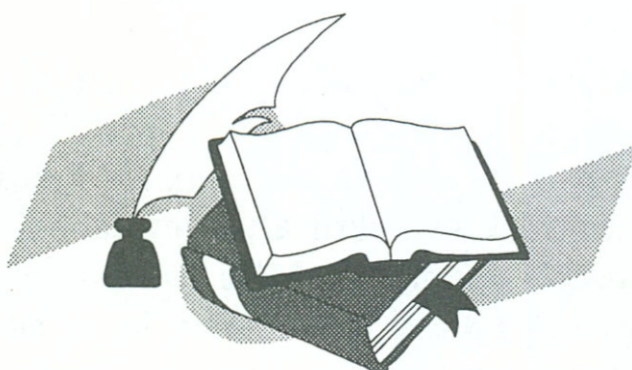
Az információ alapegysége a bit. Ez azt jelenti, egy kérdésre kétféle válasz adható: az IGEN vagy a NEM. Képzeld el, ha azt mondjuk: IGEN, akkor a gép egy bizonyos pontján van áram (mintha bekapcsoltunk volna egy kapcsolót), ha pedig NEM-et mondunk, akkor nincs áram az adott pontban.

Egy 8 bitből álló csoport neve BYTE (*bájt*). Egy 8 bites kód 0 és 255 között összesen 256 különböző számot jelenthet. Ez a 256 féle lehetőség már alkalmas szinte az összes karakter (betű, szám, írásjel) megkülönböztetésére. A különböző országokban különböző karaktereket, betűket használnak, így a számítógépekben is lehetőség van ezek közül válogatni. Az adott gépen használható 256 karakter együttesét **kódlap**nak nevezzük. A Függelékben három kódlapot adtunk meg, mert Magyarországon ezek fordulnak elő leggyakrabban.

Léteznek a byte-ok nagyobb csoportjai is. Ezeket foglaltuk össze a következő táblázatban. A karakter (character) szó helyett a C betűt használtuk!

	8 bit	= 1 byte (bájt)	= 1 C
2^{10} byte	= 1024 byte	= 1 kbyte (kilobájt)	= 1024 C
2^{10} kbyte	= 1024 kbyte	= 1 Mbyte (megabájt)	= 1 048 576 C
2^{10} Mbyte	= 1024 Mbyte	= 1 Gbyte (gigabájt)	= kb.1 milliárd C

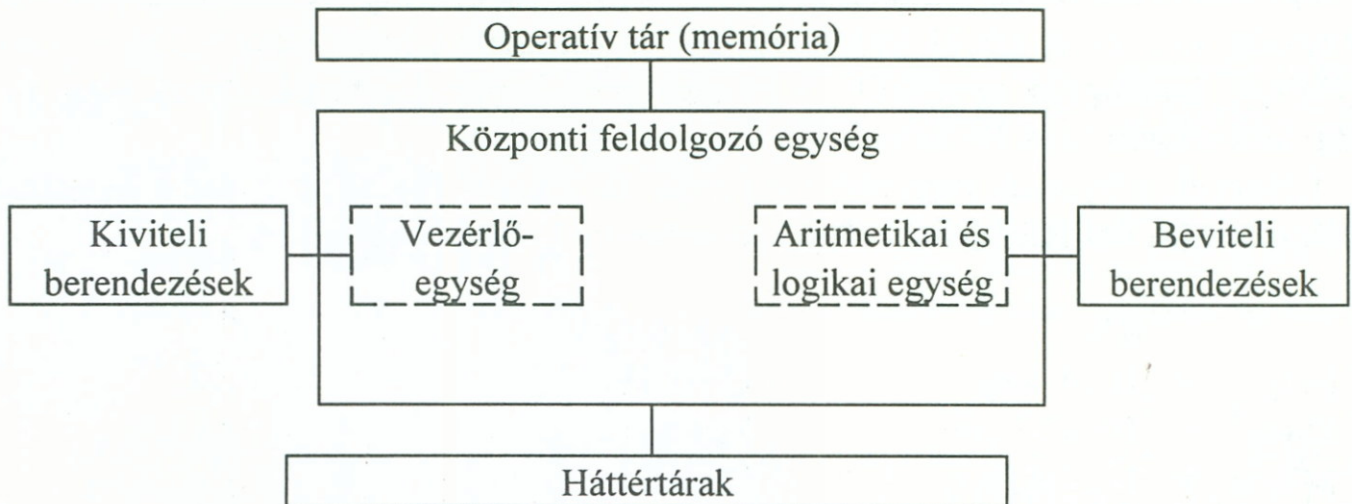
(Pl. 1 jelsorozathoz 1 karaktert rendelünk hozzá: 01 000 001b = „A”= 65d.)



A SZÁMÍTÓGÉP ELVI FELÉPÍTÉSE



A számítógép mechanikus és elektronikus részeinek együttesét, mindazt, ami a gépben fizikailag jelen van, kézzelfogható, HARDWARE-nek (*haadueönek*) nevezük. A mikroszámítógép (hardware) elvi felépítése a következő:



A számítógép központi egysége tehát a következő főbb részekből áll:

- vezérlőegység;
- aritmetikai és logikai egység;
- belső memória, vagy más néven operatív tár.

A vezérlőegység, valamint az aritmetikai és logikai egység együttes neve PROCESSZOR, vagy központi feldolgozó egység.

Az elektronikus számítógépek az elemi műveleteket (négy alapműveletet, logikai műveleteket) nagyon gyorsan el tudják végezni az aritmetikai és logikai egység segítségével.

A gyorsaságot egy olyan kvarckristály befolyásolja, mint amilyen a kvarcórában van. A rezgés sebességének mértékegysége a HERTZ (*herc*). (*1 hertz azon inga lengésének sebessége, amely 1 oda-vissza lendülést 1 másodperc alatt végez el.*)

Az AT típusú számítógépek kvarckristályai legalább 12 millió rezgést végeznek másodpercenként, rezgésszámuk 12 MHz (*megaherc*) értéktől kezdődik. A nagy sebességet csak úgy tudja a gép kihasználni, ha az adatok azonnal rendelkezésre állnak. Erre szolgál a gép memóriája, amely a számítások kezdő adatait, részeredményeit, valamint a számítások elvégzését irányító vezérlőprogramot tárolja.

A vezérlő-, az aritmetikai- és a logikai egységet együtt a számítógép **központi feldolgozó egységének** nevezzük. Angol nevének rövidítése CPU (Central Processing Unit, *szentrl prouszeszin(g) júnit*). Ez biztosítja a számítógép részegységeinek összehangolt működését.

VEZÉRLŐEGYSÉG: A CPU része. Feladata a program utasításainak értelmezése, végrehajtása. Ez vezérli a számítógép többi egységének működését.

Az **ARITMETIKAI- ÉS LOGIKAI EGYSÉG** áramköreiben történik a relációk (kisebb, nagyobb, egyenlő, stb.), illetve a logikai és számítási műveletek kiértékelése. Általában minden programutasítás végrehajtásában részt vesz.

A **MEMÓRIA** egy része kizárólag a CPU által használt rész. Ebbe gyárilag beleégetik a gép működéséhez elengedhetetlen információkat. Ezt a **csak olvasható memóriát** nevezzük **ROM**nak (Read Only Memory, *ríd onli memöri*). A ROM tartalmát a számítógép készítésekor egy speciális géppel rögzítik, így tartalma mindaddig megmarad, amíg a gép működőképes. Itt tárolják azokat az utasításokat, amelyek a gép bekapcsolása utáni életre keltéséhez szükségesek. A memória másik része a felhasználó által módosítható, adatokkal feltölthető, bármikor írható és olvasható. Ezt a **változtatható tartalmú memóriát** nevezzük **RAM**nak (Random Access Memory, *rendöm ekszesz memori*). Ide töltjük be azt a programot, amelyre szükségünk van, s ide kerülnek a program futása során keletkezett adatok is.

FIGYELEM: A gép kikapcsolásával a RAM tartalma elvész!

A **BUSZRENDSZER** egy vezetékhálózat, amely a CPU, a memória és a beviteli-kiviteli egységek között tartja fenn a kapcsolatot. Megkülönböztetünk cím-, adat- és vezérlőbuszt.

A perifériális eszközök többnyire különálló berendezéseket jelentenek, amelyek illesztőegységeken, interface-eken (*intörfész*) keresztül kapcsolódnak a számítógéphez, s a CPU irányítja tevékenységüket.

Adatbeviteli (input) egységek

A számítógéphez kapcsolható olyan eszközök csoportja, melyek az adatok gépbe vitelét szolgálják. Az alábbiakban csak néhány ilyen típusú eszközt sorolunk fel.

BILLENTYŰZET: A gép és a felhasználó közötti közvetlen kapcsolatot biztosítja. Kis mennyiségű adat bevitelére szolgál.

BOTKORMÁNY (Joystick, *dzsojsztik*): Főleg játékok irányítására használjuk.

EGÉR (Mouse, *máusz*): A programok vezérlésére, menüpontjainak kiválasztására használjuk.

FÉNYCERUZA: A képernyőre lehet vele írni, vagy a képernyő adott pontjának helyzetét lehet vele érzékelni.

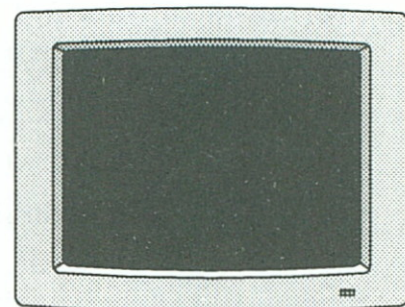
LAPOLVASÓ (Scanner, *szenö*): Rajzok, fényképek, szövegek számítógépbe vitelére használatos eszköz. A kézi scanner általában 8-10 cm széles, az asztali kivitelűek gépírópapír méretű lapok letapogatását teszik lehetővé. A gép memóriájában keletkezett képet speciális program segítségével lehet feldolgozni. Jelenleg már a kézzel írt szöveg felismerése is megoldott.

MIKROFON: Napjainkban dolgoznak az emberi hanggal irányított számítógépek kifejlesztésén. Már létezik olyan eszköz, amelynek segítségével a mozgássérültek elektromos kocsiját a vezető pár szó felhasználásával irányítja.

OPTIKAI LEMEZ (CD): Hatalmas mennyiségű adat tárolására alkalmas. Kapacitása 600 Mbyte, ami 200 ezer gépelt oldal tárolásának felel meg. Ilyen lemezen hanganyagokat (zenét), illetve a nagyobb méretű korongokon filmet is tárolnak.

Adatkiviteli (output) egységek

MONITOR (Display, *diszpléj*): A mikroszámítógépek elsődleges kiviteli eszköze. Más parancs hiányában a számítógép minden kimenő információja a képernyőre kerül. A számítógép képernyőjére 25 sorba, soronként 40–80 karaktert lehet írni. Egy karakter több kis pontból (*pixel*) áll. A karakterek ezen képpontok kifényesedésével, illetve elsötétedésével jelennek meg.



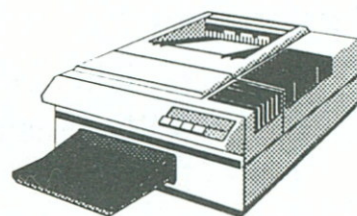
Ezeket a karaktereket a gép a memóriájában tárolja, s egy billentyű leütésekor vagy kódszámának beütésével jeleníti meg a képernyőn. A grafikai alkalmazások során ezeket a pixeleket kezeli az adott program. Attól függően, hogy a számítógép monitora hány darab ilyen pixelt tud megjeleníteni, csoportosíthatók a monitorok. A monitorok fajtái a képernyő grafikus felbontóképessége alapján:

HERCULES	720 × 348 pixel, sötét alpra világos betűkkel ír;
CGA	320 × 200 pixel; színes monitor (4 szín); 640 × 200 pixel, színes monitor (2 szín);
EGA	640 × 350 pixel, színes monitor (4 szín);
VGA	640 × 480 pixel, mono vagy színes monitor, 1024 × 768 pixel, SVGA monitor;
LCD	LAP-TOP számítógépek monitora, általában monochrom, de színes is készül (mint a kvarcjáték képernyője);
Plazma kijelző	Hasonlít az LCD-hez, csak erősebb a fénye.

A TV képernyőjét a PC-k csak megfelelő csatlakozókártyával tudják alkalmazni.

NYOMTATÓ (Printer): Segítségével az információt papíron is megjeleníthetjük. Megkülönböztetünk hő-, sor-, mátrix-, tintasugaras- és lézernyomtatókat.

RAJZGÉP (Plotter): térképek, diagramok, rajzok elkészítéséhez használják.



HANGSZÓRÓ: a számítógép által előállított hangokat lehet rajta megszólaltatni. Speciális hangkártyával sztereo minőségben hallgathatjuk a zenét.

Háttértárolók

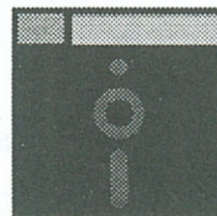
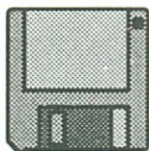
A gyakorlatban a mágneses adattároláson alapuló eszközöket használjuk háttértárolónak, de terjedőben vannak az optikai elven alapuló elektronikus háttértárolók is.

MÁGNESZALAGOS TÁR (streamer, *sztrímő*): mágneses anyaggal bevont szalagra rögzíti az adatokat.

MÁGNESLEMEZES TÁR: egy mágneses anyaggal bevont korong. Működés közben ez a korong nagy sebességgel forog az író-olvasó-törlő fejek között. Ezen a mágnesezhető rétegen található meg az oda felvitt adatok, innen olvashatjuk, illetve törölhetjük le azokat. Alapvetően két különböző fajta mágneslemezes tár van:

1. WINCHESTER (*uincsesztő*): Merevlemezes tároló vagy **Hard Disk Drive** (*haad diszk drájev*) vagy röviden **HDD**. Általában a számítógépek dobozán belül található. Tárolókapacitásuk 10–2000 Mbyte körüli. Működését egy kis lámpa jelzi a számítógépen. A zárt házban elhelyezett mágneslemezeket jóval nagyobb sebességgel lehet forgatni, mint a hajlékony tokban lévőket, így az adatok elérési ideje is lényegesen kisebb ebben az esetben. Jelenleg már kaphatók cserélhető lemezes winchesterek is.

2. FLOPPY DISK DRIVER (*flopi diszk drájevö*) vagy **(FDD)**, magyarul hajlékonylemezes tároló. Méretüket inch (*incs*)-ben adják meg. Ez angolszász mértékegység. 1" \approx 2,54 cm. A lemezek borítója négyzet alakú, de benne egy korong található, ennek az átmérőjét adják meg a lemez jellemző méreteként. A kereskedelemben kapható 3 és fél inch-es lemez (kb. 9 cm széles a tokja), valamint 5 és egynegyed inch-es lemez (kb. 13 cm széles tokban). A kis és közepes nagyságú adatfeldolgozó rendszerek könnyen kezelhető kiegészítő háttértára.



A hajlékony mágneslemez elkészültét az IBM cég 1970-ben jelentette be, akkor még csak olvasható adattárolóként. Kezdetben egyoldalas, majd 1976-tól már kétoldalas mágneslemezeket gyártottak.

A **lemez hordozóanyaga** hajlékony, rugalmas műanyag, melyre mágnesezhető réteget visznek fel. A lemezt a meghajtóba való helyezés után a középben levő felületen két tárcsa szorítja össze és megforgatja. A kb. 300 percenkénti fordulat elérése esetén lehet a lemezt használni. A gyors forgás következ-

tében a tok és a lemez között légpárna képződik, ami megakadályozza közöttük a súrlódást.

Az *öntapadós címkére* filc- vagy golyóstollal írjunk a felragasztás előtt, mert más esetben megsérülhet a lemez is!

Az *írásvédő kivágást* leraasztva a lemezre nem lehet írni.

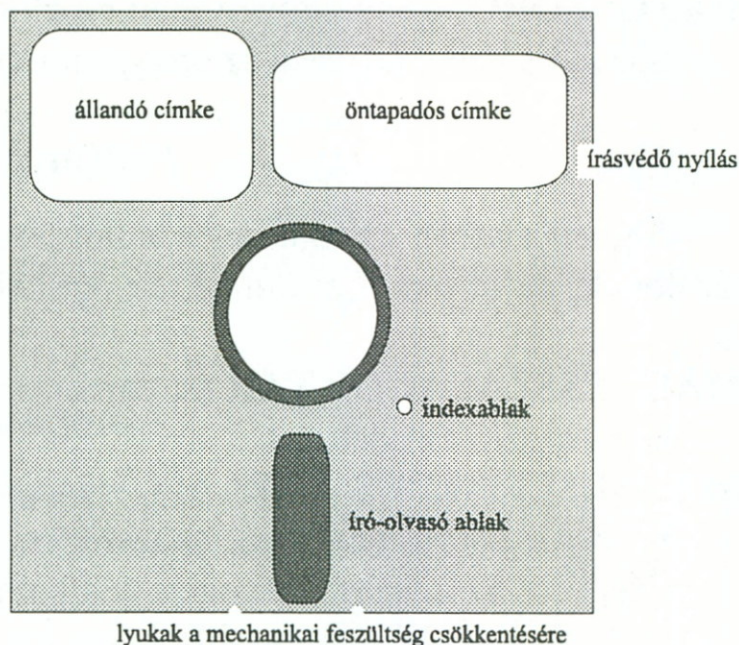
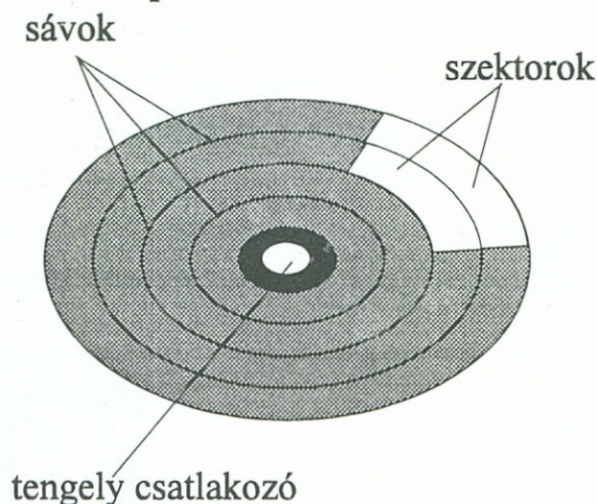
Az *indexablak* a borítóban forgó lemezen való tájékozódást szolgálja. Minden fordulat közben az indexablak és a lemezen található lyuk egymás fölé kerül. Ekkor a

lemez egyik oldalán elhelyezett fényforrásból a fény átjut a másik oldalon levő fényérzékelőbe. A jeleket a vezérlőegység áramkörei dolgozzák fel, így érzékelik a lemez típusát. Az egy-, illetve kétoldalas lemezek indexnyílásai máshol helyezkednek el.

A *központi ablakba* illeszkednek a lemezegység forgató csonkjai.

Az *író-olvasó ablak* fölött (kétoldalas lemeznél alatta is) található(k) a lemezegység író-olvasófeje(i).

A hajlékonylemez belső felépítése



Az adatok sávokban, sávonként 8 vagy 9 szektorban helyezkednek el. A sáv egy körgyűrű alakú felület a lemezen, mely az író-olvasófej előtt halad el, miközben a lemez egy teljes fordulatot tesz meg, és a fej nem mozdul el. Az író-olvasó szerkezet a fejet típustól függő számú helyre tudja mozgatni a lemez fölött. Így ennek megfelelő számú *koncentrikus kör (sáv)* mentén lehet adatokat rögzíteni. Kétoldalas lemez esetén a két oldalon az *azonos fejálláshoz tartozó sávpárok egy cilindert alkotnak*. A sávok számozása kívülről befelé történik.

A legelső (00 sorszámú) cilindert indexcilindernek nevezzük. Ezen a lemez tartalmára vonatkozó információk találhatóak.

A két legutolsó cylinder tartalék, melyet oldalanként két-két meghibásodott sáv pótlására használhatunk.

A *szektor* a sávnak az a része, amely egy 5 és negyed inch-es lemez esetében 512 byte (*bájt*) adat tárolására alkalmas. Így az említett méretű kétoldalas lemezre 80 sáv fér el egy oldalra. (A lemezeket jobbra kétoldalas kivitelben gyártják, csak az FDD-től függ, hogy mindkét oldalon van-e író-olvasó feje.)

A hajlékonylemezre írt betűjelek

Aszerint, hogy a lemez egyik vagy mindkét oldalát használja-e a lemezegység

SS (Single Side, *szingl szájd*) egyoldalas;

DS (Double Side, *dábl szájd*) kétoldalas;

lemezeket különböztetünk meg.

A jelrögzítésnél az **írás sűrűsége szerint** megkülönböztetünk

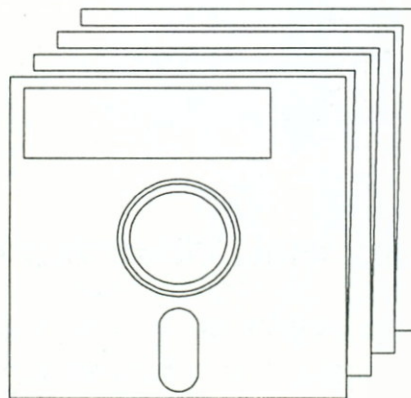
SD (Single Density, *szingl densziti*) egyszeres írássűrűségű;

DD (Double Density, *dábl densziti*) kétszeres írássűrűségű;

HD (High Density, *háj densziti*) nagy írássűrűségű;

hajlékonylemezeket.

A lemezeket az első felhasználás előtt elő kell készíteni, formázni, hogy az FDD ki tudja alakítani rajta a sávokat, ezen belül pedig a szektorokat. Eközben megtörténik a sávok felhasználhatóságának ellenőrzése is. A hibásnak bizonyult sávokat az FDD más, tartalék sávokkal helyettesíti. Ezt a folyamatot elvégezhetik a gyártók is, ekkor a boltban már eleve formattált lemezt árusítanak.



A SZEMÉLYI SZÁMÍTÓGÉP ALAPKONFIGURÁCIÓJA

A személyi számítógép angol nevének rövidítése, a PC kifejezés a Personal Computer (*pösznl kompjútö*) első betűiből származik.

Magyarországon a IBM PC vagy azzal kompatibilis számítógépek terjedtek el. Az iskolákban még viszonylag kis számban található meg az Apple által gyártott Macintosh gépek, így ezekről egyelőre nem írunk. Kompatibilis egymással az a két számítógép, amelyek egymás programjait változtatás nélkül képesek futtatni. A PC-k felülről kompatibilisek egymással, ami azt jelenti, hogy egy régebbi típusú gépen megírt program futtatható egy újabb típusú gépen, de fordítva már nem feltétlenül.

Alapkonfiguráción a számítógép központi egységét, billentyűzetét, lemezegységét, monitorát értjük, de gyakran egeret és nyomtatót is kapcsolunk hozzá.

Ez, a már említett, kézzelfogható rendszer, a **hardver**.

A **szoftver** valamely számítógéprendszerhez tartozó programok, programrendszerek, valamint eljárások, szabályok és mindezekre vonatkozó dokumentációk összefoglaló elnevezése. Szoftveren értjük emellett mindazon szellemi termékek összességét, mely a hardver mellett, illetve azzal kölcsönhatásban a számítógép működését, felhasználását lehetővé teszi.

Az IBM PC központi egységében található **mikroprocesszor legjellemzőbb típusai** a következők:

INTEL 80286-os vagy 80386-os vagy 80486-os típusú

vagy ezzel kompatibilis, más gyártók által készített chipek. A legutóbbi fejlesztések eredményeként napjainkban kezdenek elterjedni a piacon a **Pentium** névre keresztelt mikroprocesszorral működő gépek is.

A kvarckristály 12 MHz-nél nem kisebb órajelű, a RAM legalább 1 Mbyte.

A központi egység egy fémdobozban kapott helyet, több más kiegészítő egységgel együtt.

A számítógép dobozának külső burkolatán található

K/LOCK (*kí lok*): egy kulcs segítségével megszakítható a kapcsolat a gép és billentyűzete között, így azzal kulcs hiányában nem dolgozhat más.

POWER (*páuo*): az áramellátást visszajelző lámpa.

HDD-lámpa: a winchester működését jelzi. Csak akkor világít, ha a HDD működik.

TURBO-lámpa: a gyorsabb működést jelzi vissza.

TURBO-kapcsoló: kapcsolásával a gép gyorsabb, illetve lassabb működését állíthatjuk be.

RESET (rízset) -gomb: ezzel lehet a gép bekapcsolt állapotában újraindítani a számítógépet.

3.5"-es vagy 5.25"-es FDD nyílása: a két FDD közül legalább az egyiket beépítik a számítógépbe.

POWER kapcsoló: ezzel kapcsolható be a gép.

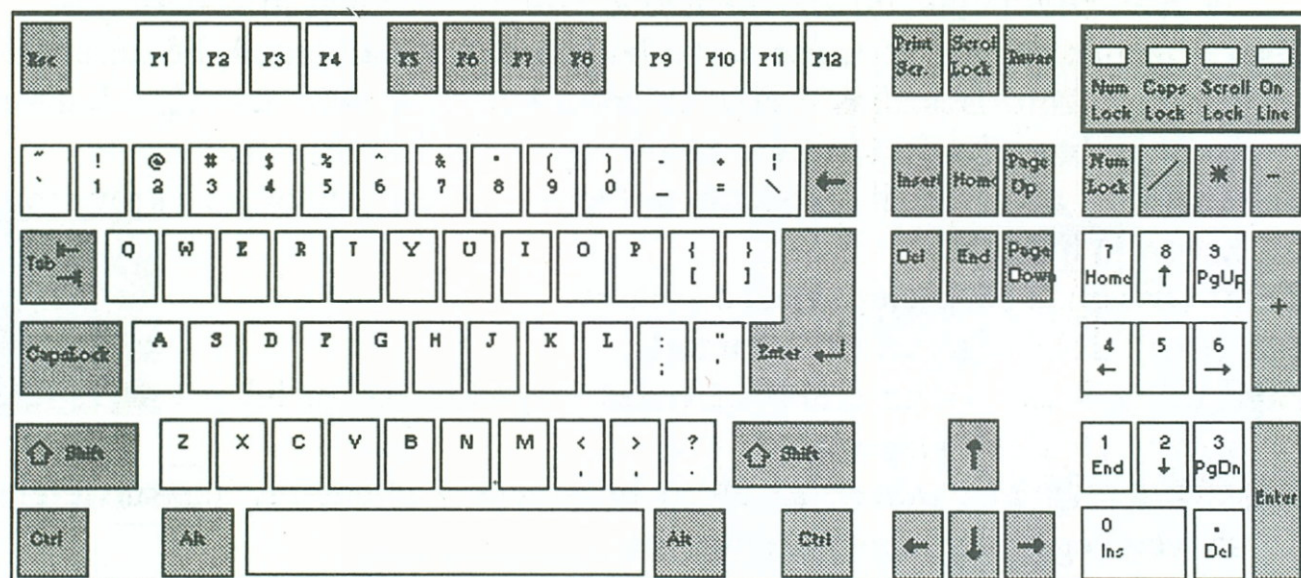
Monitor

Általában a számítógép fölé helyezük a monitort. A monitoron található a bekapcsoló-, a fényteltetés-, illetve a fényerősség-állító gombok. Gyakran van a monitoron néhány olyan gomb is, amelyekkel a kép helyzete állítható.

Egér (mouse)

A számítógép mellé helyezhetjük a kurzor mozgatását segítő egeret és a hozzá tartozó egérpadat. Az egeret célszerű ezen a lapon mozgatni, mert az egerben elhelyezett golyó így nehezebben csúszik meg, s kevésbé szennyeződik is.

Billentyűzet



Ezek többsége angol karaktereket tartalmaz. Kaphatók magyar ékezetes betűket tartalmazó billentyűzetek is.

F1 ... F12 a funkcióbillentyűk:

ezek működése a számítógépen futó programtól függ.

ESC (Escape, *iszkép*) gomb:

általában a programokból való „kimenekülésre” tudjuk használni.

PRINT SCREEN (*print szkrín*):

a billentyűt megnyomva a képernyő tartalmát nyomtatóra lehet kiíratni.

SCROLL LOCK (*szkroul lok*):

a billentyű használata a képernyő adott területének mozgatását segíti elő.

PAUSE (póz):

a gomb megnyomása felfüggeszti a program futását, majd egy újabb billentyű megnyomására a program tovább fut. Ugyanezt a gombot a **CTRL** gombbal együtt megnyomva a program futása megszakad.

ÍRÓGÉPBILLENTYŰK:

ezen számok, betűk, illetve egyéb írásjelek találhatók. Amelyik billentyűn két jel is található, azokon a felső jelet a **SHIFT** és az adott gomb egyidejű lenyomásával érhetjük el.

MÓDOSÍTÓ BILLENTYŰK:

az **ALT**, a **CTRL** és a **SHIFT** billentyűk önállóan nem hatnak a gép működésére, csak más billentyűkkel együtt megnyomva módosítják azok jelentését.

KURZORVEZÉRLŐ BILLENTYŰK:

A \uparrow , \downarrow , \leftarrow , \rightarrow billentyűk a numerikus és az írógépbillentyűk között helyezkednek el. Segítségükkel irányítható a képernyőn villogó kis vonal, a **KURZOR**.

EGYÉB VEZÉRLŐBILLENTYŰK:

A kurzorvezérlők fölött helyezkednek el. Jelentésüket az egyes programok megváltoztathatják, de legjellemzőbb szolgáltatásaikat itt ismertetjük.

Insert (inzót): szöveg gépelése során lenyomásával válthatunk beszúrás és felülírás között. Beszúrás esetén az adott helytől a beírt szöveg a kurzornál levőt jobbra tolja és a helyére kerül, egyébként felülírja.

Delete (dilit): a kurzornál lévő karaktert letörli, a tőle jobbra levő szövegrészeket balra húzza.

Home (houm): a sor elejére viszi a kurzort.

End (end): a sor végére viszi a kurzort.

Page Up (péidzs ap): hosszabb szövegek megtekintésekor lehet vele felfelé (a szöveg eleje felé) lapozni.

Page Down (péidzs dáun): az előző billentyűvel ellentétes hatású, lefelé (a szöveg vége felé) lehet vele lapozni.

BACKSPACE (bekszpéisz) vagy \leftarrow : a kurzortól balra lévő karaktert törli és a szöveg többi részét balra húzza.

ENTER (entö) vagy \downarrow : a billentyűt minden parancs begépelése után meg kell nyomni, hogy azt a gép végrehajthassa.

CAPS LOCK (kepsz lok): a billentyű a kis- és a nagybetűs írásmód közötti váltást teszi lehetővé.

TAB vagy $\overrightarrow{\leftarrow}$: ezzel lehet gépeléskor kb. 5 karakternyit jobbra ugrani. A **SHIFT** billentyűvel együtt lenyomva általában balra ugrást eredményez.

SPACE (szpéisz): rendszerint a szóköz beírására szolgáló billentyű.

JELZŐLÁMPÁK: egy-egy speciális gomb működését jelzik vissza.

Nyomtató

Egyes **elektromos írógépek** is alkalmasak a számítógép által küldött információk kiírására, de nagyon lassúak.

A **sornyomtató** egy sor betűit nem egymás után, hanem egyszerre veti papírra, így lényegesen gyorsabb.

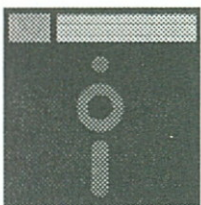
A **mátrixnyomtató** minden karaktert pontokból állít össze. A nyomtatás oszloponként történik 7-24 tű segítségével, festékszalagon keresztül. Lassabak, mint a sornyomtatók, de lényegesen olcsóbbak, ezért a legelterjedtebbek.

A **tintasugaras nyomtatónál** is pontokból állnak a karakterek, de az igen kis méretű pontokat festékszemcsék alkotják, amelyeket a fej ráfecskendez a papírra. Gyorsabban és sokkal csendesebben dolgozik, mint a mátrixnyomtatók és az írásképe is szebb.

A **lézernyomtatónál** a kinyomtatandó kép először egy szelénhenger palástjára kerül rá (lézerrel). Itt a különböző színű helyeknek megfelelően különböző mértékű az elektrosztatikus töltés, amely a festékszemcséket magához köti. A hengerről ezután kerül a papírra az információ. A legjobb minőségű nyomtatást végzi, ára azonban egyelőre meglehetősen borsos.

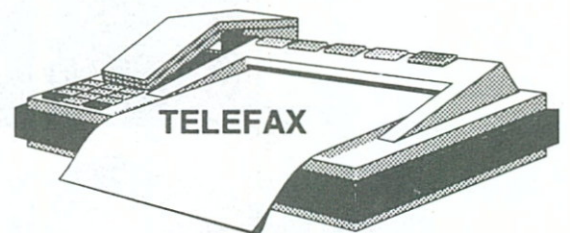
Feladatok

1. Kösd össze kék vonallal az adatbevitelhez, pirossal az adatkivitelhez szükséges eszközök és a központi egység rajzát.



CD-ROM

MOUSE



TELEFAX

HANGSZÓRÓ

KÖZPONTI EGYSÉG

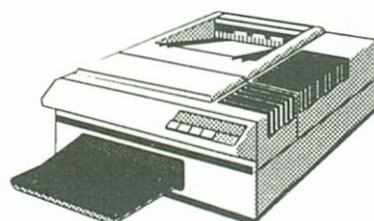


TV



MONITOR

RAJZGÉP

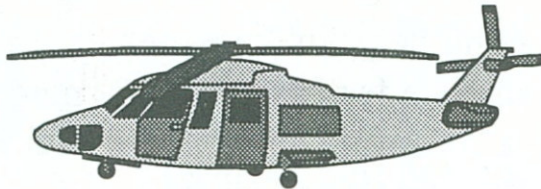


BILLENTYŰZET

2. Melyek voltak az előző feladatban azok az eszközök, melyek mindkét cél-
nak megfelelnek?
.....
3. Az 1. feladatban felsorolt eszközök közül melyiket nem lehet közvetlenül a
számítógéphez kapcsolni?
.....
4. Az alábbi tárgyak mellé írd oda röviden, milyen célra használnád fel hoz-
zájuk a számítógépet!



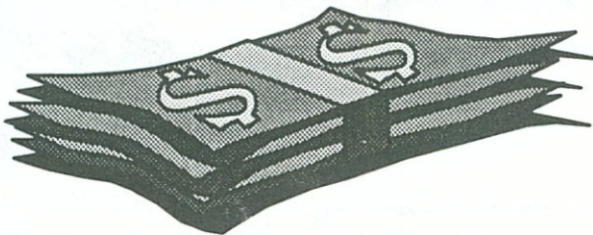
.....



.....



.....



.....



.....

A SZÁMÍTÓGÉP OPERÁCIÓS RENDSZERE

Mielőtt bekapcsoljuk számítógépünket, illik tudni, hogy csak akkor indul, ha van úgynevezett operációs rendszer (Operating System, *opöréting szisztöm*) a háttértárolón.

Az operációs rendszer olyan programrendszer, amely egy számítógép összes hardver és szoftver erőforrásának kezelését vezérli, felhasználását összehangolja, biztosítja hatékony együttműködésüket.

Az operációs rendszer sebessége, szolgáltatásai jelentősen befolyásolják egy számítógép működését. Minősége gyakran fontosabb a felhasználónak, mint maga a számítógép hardver része. A legelterjedtebbek a DR-DOS, a PC-DOS, az MS-DOS, a UNIX, a XENIX.

A DOS az IBM számítógépek lemezes operációs rendszere (Disk Operating System). Ezt a Microsoft cég készítette, ezért MS-DOS-nak nevezik. Az állandó fejlesztés mellett egyre több verziója alakult ki. Ezeket a verziókat sorszámokkal jelölik. Így létezik a legegyszerűbb MS-DOS 1.00-tól a MS-DOS 3.3 verzióig át a jelenlegi legnagyobb és legtöbbet tudó MS-DOS 6.22 verzió is. Minden újabb verziójú DOS kifejlesztésekor az a mérvadó, hogy többet tudjon, mint az előző, de a már meglévő programok az új MS-DOS alatt is működjenek.

A könyvben a DOS 6.2 parancsait használjuk és ismertetjük.

A számítógép a bekapcsolásakor a ROM-ba égetett két kis programot futtatja le (hardvertesztelés, DOS indítása). Az „élesztés” alkalmával a beégetett program megvizsgálja, milyen eszköztől tudja beolvasni a rendszert irányító COMMANDERt (*komandö*). A beállítástól függően először az FDD-ben (A: meghajtó) keresi. Ha a PC-ben HDD is van, akkor legutoljára általában ott keresi az úgynevezett rendszerfájlokat. Ha az FDD-ben talált ugyan lemezt, de az nem tartalmazza a megfelelő fájlokat, akkor

Non system disk error! (*non szisztöm diszk erö*)

hibaüzenettel leáll, hiába vannak a megfelelő fájlok a winchesteren. A számítógépek egy részében beállítható, hogy induláskor először a winchesteren keresse a rendszerfájlokat.

Ez az indítóprogram tehát elolvassa a lemez egy előre meghatározott részét. Amennyiben ott rendszerfájlok vannak, akkor a gép feléled, különben nem. A későbbiekben az itt talált programok felelősek az egész számítógép tökéletes működéséért.

Ha minden rendben volt az operációs rendszerrel, s a HDD-ről indult a gép, a monitoron általában a

C:\>

felirat látszik, vagy a felhasználó által beállított más jelsorozat (**Ez a prompt.**), esetleg azonnal elindul egy felhasználói program. A > jel utáni villogó kis vonal a már említett kurzor. A számítógépünk kész fogadni a parancsokat.

Az **A** és **B** betűvel a számítógép hajlékonylemezes meghajtóit jelölik, a **C**, **D**, ... betűkkel pedig a winchestereken található – esetleg logikai– meghajtókat.

A könyv későbbi részében egy olyan számítógépre vonatkozóan adjuk ki a parancsokat, amelyik egyetlen winchestert és egy 1,2 Mbyte-os FDD-t tartalmaz háttértárként. Amennyiben az általad használt gép nem ilyen, akkor a parancsok részletes leírásánál találhatsz segítséget a szükséges kiegészítésekhez, illetve kérdezd meg bátran tanárodat.

Írjuk be a következő parancsot:

DIR

(nem baj, ha kisbetűkkel írtuk, hiszen a gép nem tesz különbséget a kis- és a nagybetűk között).

[Ha a parancs betöltése vagy végrehajtása akadályba ütközik, akkor legtöbbször a következő üzenetet kapjuk:

Abort (öbót), Retry (ritráj), Ignore (ignó)?

Kilépésre, megisméltésre, illetve a figyelmen kívül hagyásra van lehetőségünk. Válaszolni a megfelelő kezdőbetű leütésével lehet: A vagy R vagy I.]

A DIR parancs hatására a képernyőn végigfut egy lista, amely a winchesteren (lemezen) levő állományokat tartalmazza.

Egyes nevek után egy hárombetűs szócška áll. Ezeket fájlloknak nevezzük. Máshol a név után a <DIR> felirat látszik. Ezeket alkönyvtáraknak hívjuk.

A DIR paranccsal tehát megvizsgálhatjuk, mi van az adott lemezen.

Tegyünk be egy üres lemezt a meghajtóba, de előtte vizsgáljuk meg, mi van a címkéjére írva. A PC megvásárlása óta tudjuk, milyen típusú a benne lévő FDD. Ha az FDD 360 kbyte-os, a DS/DD-s lemezt használhatjuk, ha 1.2 Mbyte-os, a DS/HD típusú lemez az ideális, de használható a DS/DD-s is. Amennyiben a felhasználni kívánt lemezünk nem formattált, akkor nekünk kell a DOS számára is használhatóvá tenni. A parancs – 1.2 Mbyte-os lemez esetében – a következő:

FORMAT A:

(ha 360 kbyte-os a lemez, akkor a helyes parancs **FORMAT A: /4**)

Ekkor a gép megkeresi és elindítja az operációs rendszerben található **FORMAT** (*fómet*) parancsot, majd kiírja:

Insert new diskette for drive A:

(*inzőt nyú diszket for drájv A*) tegye az új lemezt az A meghajtóba

and press ENTER when ready

(*end presz entö ven redi*) és nyomja meg az **ENTER**t, ha kész.

Nyomjuk meg a gombot. Kiírja:

Checking existing disk format

majd

Formatting...

... percent completed.

és kattogva dolgozni kezd.

Ha kész, megjelenik a következő felirat:

Format complete (*fómet kömplít*)

Volume label (11 characters, ENTER for none)

(*voljúm lébl 11 keriktöz, ENTER for non*) lemeznév 11 betűs, ENTER, ha nincs.

Írd be a monogramodat, majd nyomd meg az **ENTER** gombot.

Ekkor a gép többek között kiírja:

- a teljes lemez területét (ennyi byte-ra formáztuk a lemezt);
- az esetleg hibás területet;
- az elérhető területet byte-okban.

Ezután kérdést tesz fel.

Format another (Y/N)? (*fómet önádö Yes /No*) Formázunk másik lemezt Igen/Nem?

Nemleges választ (N) adjunk! Ekkor újra kiírja a promptot: C:\>_

Ha új lemezt kívánunk formázni, a **YES** (*jesz*) parancs **Y** betűjét válasszuk! A gép ismét bekéri a lemezt, s a folyamat az előzőek szerint újra lezajlik.

Ha olyan lemezt szeretnénk készíteni, amely alkalmas egy számítógép elindítására is (rendszerlemez), akkor a parancsot ki kell egészítenünk:

FORMAT A: /S

Ezzel a kiegészített paranccsal formáztunk egy lemezt, illetve rá is másoltuk a 3 rendszerfájlt.

Írd be:

DIR A:

Most az **A** meghajtó tartalomjegyzékében a 3 indítóprogram közül egy látszik.

Neve: **COMMAND.COM**

Azt a programot vagy állományt, amely a háttértáron tárolt információ, fájlnak, állománynak is szoktuk nevezni. A másik két programot a

DIR A: /A:H

paranccsal írathatod a képernyőre.

Nevük: **IO.SYS**, illetve **MSDOS.SYS**. Ezek úgynevezett rejtett állományok.

A fájl neve (maximum 8 betű) után mindig egy pont áll, majd egy (maximum 3 betűs) karaktársorozat, amit kiterjesztésnek is nevezünk.

Hagyd benn az **A** meghajtóban a lemezt, s nyomd meg a gép előlapján található **RESET-gombot** (újra indítjuk a gépet, **hideg indítás**).

A gép a beégetett két program végrehajtása után először az **A** meghajtóban keresi a lemezt. Mivel megtalálja, arról indul.

Ha valami hiba következett be a gép bekapcsolásakor, a következő üzenet jelenik meg a monitoron:

Non-system disk or disk error

Replace and press any key when ready

Ha a bent levő lemezen minden rendben volt, a képernyőn megjelenik:

Current date is...

Enter new date (mm-dd-yy): nyomd meg az **ENTER**t.

Current time is...

Enter new time: ismét **ENTER**t üss.

(Ezek jelentését a későbbiekben fogjuk tárgyalni a DOS-parancsoknál.)

Most a gép: **A:\>_** jellel jelentkezett be, s várja az újabb parancsokat. Így a gépünk a hajlékonylemez-es egységről indult el.

Vedd ki a lemezt az FDD-ből, s nyomd meg együtt a következő három gombot:

CTRL ALT DEL

A gép újraindul, a folyamat nagyon hasonlít ahhoz, mintha a **RESET** gombot nyomtuk volna meg (**meleg indítás**). A PC most ismét a HDD-ről indul.

Feladatok

1. Milyen sorrendben keresi meg a számítógép a meghajtók között az indító fájlokat?
2. Mely fájlokat kell az MS-DOS indításához a számítógépnek megtalálni a lemezen?
3. Figyeld meg a számítógépen, mi a különbség a **RESET**, illetve a **CTRL-ALT-DEL** hatása között?
4. Milyen betűkkel jelöljük az FDD-ket, illetve a HDD-ket?
5. Mit kell tenni a boltban vásárolt lemezzel, ha fel akarjuk használni a számítógépben? Ezt milyen paranccsal tehetjük meg?
6. Mi a hatása a **DIR** parancsnak?

Alkönyvtárak (subdirectory, *szábdirektri*) kezelése

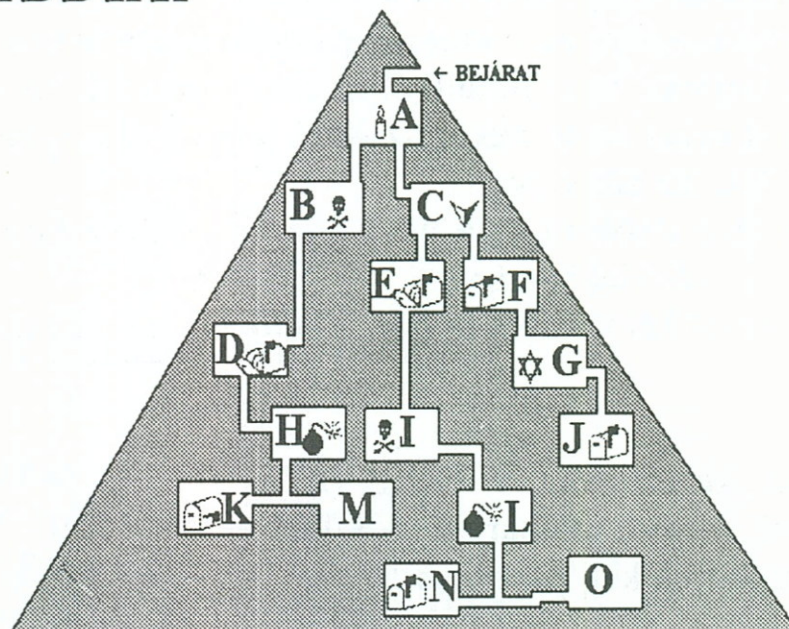
A következő oldal ábrája egy piramis rejtett kamráit mutatja. A kamrákat folyosók kötik össze. Egy kutató nagybetűkkel jelölte vázlatrajzán az egyes kamrákat. „Titkosírással” rögzítette, hogy a bejáratról milyen módon juthat el az **N**-nel jelölt kincstárterembe:

\\A\C\E\I\L\N

A vázlatrajzot megtekintve, kis gondolkodás után Ti is rájöhetek az egyszerű rejtjelre. Ezután írjátok le a fenti kódolással, milyen kamrákon át juthatunk el:

- a **BEJÁRAT**tól **F**-ig:
- a **BEJÁRAT**tól **H**-ig:
- a **BEJÁRAT**tól **K**-ig:
- a **BEJÁRAT**tól **J**-ig:

Ha az N-el jelölt teremben tartózkodunk, s át akarunk jutni a K-val jelölt másik kincseskamrába, vissza kell másznunk az A terembe, majd onnan a K-ba juthatunk. Ezt így írjuk le: \A\B\DH\K



Írd le jelekkel, hogyan juthatunk el:

E-ből D-be:

M-ből O-ba:

I-ből J-be:

A fent említett rendszert számítógépen is megvalósíthatjuk. Tegyük be egy formázott lemezt az FDD-be! A bekapcsolt gépbe írjuk be:

A:

és nyomjuk meg az **ENTER** billentyűt. A gép áttér az FDD-re.

Elkészítjük a lemezen a piramis kamrarendszerének mását. A lemez a piramis, de kamrák még nincsenek benne.

A kamrák neve a számítástechnikában alkönyvtár, a kamrák rendszere pedig „fastruktúra”.

A következő parancsokat fogjuk használni:

MKDIR név	létrehozás	röviden	MD név
CHDIR név	váltás	röviden	CD név
RMDIR név	törlés	röviden	RD név
DELTREE	alkönyvtárrendszer törlése		
MOVE név1 név2	átnevezés		
TREE	fastruktúra kiírása		

A leírtak szerint, ahol látszik, szünetet kell tartani, illetve a parancs beírása után meg kell nyomni az **ENTER** gombot, hogy a gép végrehajtsa az utasítást.

MKDIR (MaKe DIRectory, *mék direktri*) alkönyvtár létrehozása**MKDIR [meghajtó:] útvonal név**

meghajtó: ebben a meghajtóban van a kérdéses lemez. Ha az aktuális meghajtóval dolgozunk, akkor nem kell megadni (erre utal a [] a leírásában).

útvonal: ezen a bejárési úton jutunk el az adott nevű alkönyvtárba. Ha több alkönyvtáron át vezet az út, akkor azokat \ jellel kell elválasztani egymástól.

név: ilyen nevű lesz az alkönyvtár.

A piramis rajzát figyelve elsőként az **A** nevű kamrát (alkönyvtárat) hozzuk létre.

MKDIR A, rövidebben írva: **MD A**

Az **A** nevű alkönyvtárból két további nyílik, a **B** illetve a **C**. Ezeket is létrehozzuk:

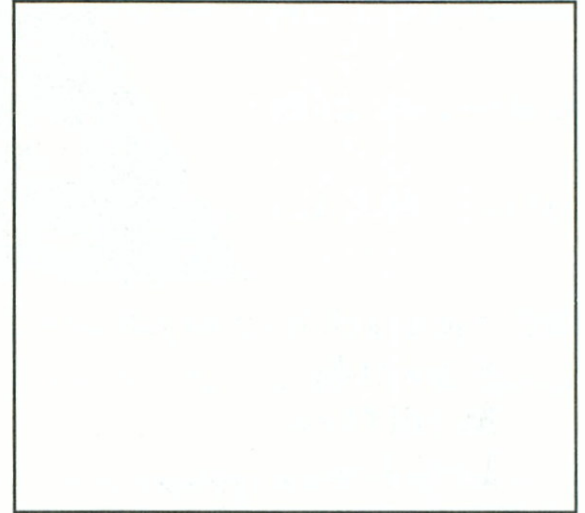
MKDIR A\B

MKDIR A\C

Az alkönyvtárak kezelésénél a **TREE** (*tri*) parancs segítségével megvizsgálhatjuk a teljes könyvtárszerkezetet. Adjuk ki a parancsot!

TREE

Mit tapasztaltatok? Ide rajzoljátok le!

**Feladat**

7. A piramis rajzát figyelve készítsétek el a teljes alkönyvtárrendszert! Írjátok le a felhasznált parancsokat!

D	E
F	G
H	I
J	K
L	M
N	O

A kialakult fastruktúrát a **TREE** parancssal ellenőrizd le!

RMDIR (ReMove DIRectory, *rimív direktri*) könyvtártörlés**RMDIR [meghajtó:][útvonal] név**

A parancs törli az adott bejárési út végén levő alkönyvtárat.

A parancs kipróbálásához hozzunk létre a **B** alkönyvtárban egy **Q** alkönyvtárat!

MD \A\B\Q

A **TREE** parancssal vizsgál meg a fastruktúrát!

Ezt az alkönyvtárat kitörölni a következő módon lehet:

RD \A\B\Q

Ismét megvizsgálva a fastruktúrát látjuk, hogy a **Q** nevű, előbb létrehozott alkönyvtárat eltávolítottuk.

A fenti példa alapján a hibásan létrehozott alkönyvtárat törölhetjük, s kijavíthatjuk a könyvtárszerkezetet, ha szükséges!

Feladatok

8. Töröld ki a **J** alkönyvtárat:

9. Töröld ki az **L** alkönyvtárat:

Mit tapasztalsz a parancs kiadása után?

.....

10. Mikor lehet egy alkönyvtárat kitörölni?

.....

MOVE (*múv*) könyvtár átnevezése

MOVE [meghajtó:][útvonal] név1 név2

A parancs a **név1** alkönyvtárat **név2** névűre változtatja.

Nevezzük át az **A** nevű alkönyvtárat **BEJARAT** névűre!

MOVE A:\A BEJARAT

A **TREE** paranccsal ellenőrizd az eredményt!

Feladat

11. Nevezd át az **N** alkönyvtárat **KINCS**-re, majd az **I** nevűt **MUMIA** névűre:

.....

.....

CHDIR (CHange DIRectory *cséndzs direktri*) könyvtársváltás

CHDIR [meghajtó:][útvonal] név

Ilyen nevű alkönyvtárba akarunk belépni.

CD hatására kiírja jelenlegi helyzetünket a fastruktúrán belül.

CD ** hatására a fastruktúra kezdetéhez, a **FŐKÖNYVTÁRba jutunk.

CD .. hatására egy szinttel feljebb lépünk.

Mielőtt elindulnánk a „kincskeresésre” a piramisban, a számítógépet olyan bejelentkezési jelre (*prompt*) állítjuk, hogy azonnal a képernyőre írja a fastruktúrán belüli helyzetünket.

PROMPT \$PSG

Ezután belépünk a **J** alkönyvtárba:

CHDIR \A\C\F\G\J

A képernyőn a következő felirat látszik:

A:\A\C\F\G\J>_

Lépünk be a **K** alkönyvtárba:

CD \A\B\D\HK

Feladatok

12. Mi a szerepe a CD utáni \ jelnek?

.....

13. Az alábbi alkönyvtárnevek után írd le azt a parancsot, amellyel beléphetsz az adott könyvtárba! Próbáld ki a számítógépen is!

D
 E
 MUMIA
 O
 N
 G
 BEJARAT
 B

TREE (tri) könyvtárstruktúra lekérdezése

Külső parancs, tehát a DOS alkönyvtár elérése nélkül nem működik. Mivel mi a winchesterről dolgozunk, a gépben már biztosítva van a DOS elérése, így azt nyugodtan használhatjuk. A MIÉRTre majd később visszatérünk. A parancs hatására a gép kirajzolja a képernyőre a könyvtárszerkezetet. Formája:

TREE [meghajtó:][F][A]

meghajtó: ebben a meghajtóban van a vizsgálandó lemez.

Lépünk vissza a főkönyvtárba, a gyökérbe: **CD **

TREE A:

majd a

TREE C: /f

illetve a

TREE A: /a

parancsokat próbáld ki!

Feladatok

14. Mit tapasztalsz, ha a **TREE C: /F** parancsot használjuk?

.....

15. Mit ír ki a számítógép ha a **TREE C: /F /A** parancsot használjuk?

.....

DELTREE (deltri) könyvtárstruktúra törlése

Külső parancs, tehát a DOS alkönyvtár elérése nélkül nem működik.

Formája:

DELTREE [/Y][meghajtó:] útvonal

/Y A törléshez nem kér megerősítést a prompt után.

A parancs kipróbálásához hozz létre a C alkönyvtárban egy **KUKTA**, majd abban egy **TYUK** alkönyvtárat!

A felhasznált parancsok:

A TREE paranccsal vizsgálj meg a fastruktúrát!

Most letöröljük mindkettőt egyszerre :

DELTREE /Y A:\C

A TREE paranccsal vizsgálj meg a fastruktúrát!

További MS-DOS parancsok

A parancsok egy része külső parancs (a DOS alkönyvtár elérése nélkül nem használható), más része belső (a DOS alkönyvtár elérése nélkül is használható, a COMMAND.COM fájl felismeri).

A parancsok paraméterei, kapcsolói a Függelékben található. A parancs ismertetésekor csak a leggyakrabban használt paramétereiket soroljuk fel. A Függelékben megtalálhatók az MS-DOS 6.2 olyan parancsai is, melyeket nem tárgyalunk a könyvben részletesen, de használatuk segítheti munkánkat.

Minden DOS-parancsnál lehetőségünk van a paraméterek lekérdezésére. A következőt kell tenni: **DOS_parancs /?**. Hatására a gép a képernyőre írja az összes lehetséges beállítási módot angol nyelvű magyarázó szöveggel!

A DOS összes parancsának leírását a HELP.COM tartalmazza.

Írd be:

HELP

Megjelennek az MS-DOS 6.2 parancsai. A képernyőn a **PgUp** és **PgDn** gombokkal lapozhatsz. A sorokban a kurzormozgató billentyűkkel állhatsz rá a kívánt parancsra. Az **ENTER** billentyűvel tudod kiválasztani a keresett parancs leírását. Ezek után jelenik meg az adott parancs angol nyelvű leírása.

A tartalomjegyzékhez visszatérni az **ALT** és **C** gombok együttes megnyomásával lehet.

Az **ALT** és **N** gombok megnyomására további leírások jelennek meg.

Az egymás után megjelenő leírásokból az **ALT** és **B** gombok megnyomásával léphetünk vissza egy szinttel.

A **menüsorba** az **ALT** gomb megnyomásával kerülhetünk. Itt a **FILE** menüt az **ENTER** gombot megnyomva választhatjuk ki. A leírásokat a **PRINT** paranccsal kinyomtathatjuk. Az **EXIT** parancsot választva kilépünk a **HELP**ből.

A **SEARCH** (szócs) menüben a **FIND** (fájnd) pontot választva megjelenik egy keret, melyben a kurzor villog. Ebbe beírva a keresett parancs nevét és az **OK** vagy az **ENTER** gombot választva a gép megkeresi, majd kiírja a képernyőre az adott parancs angol nyelvű leírását. A **Repeat Last Find** (ripít lászt fájnd) vagy az **F3** hatására megismétli a keresést.

DIR (DIRectory, *direktri*) **tartalomjegyzék kiíratása** (belső parancs)

Már használtuk a **DIR** parancsot tartalomjegyzék kiíratására. Most az egyéb lehetőségeket próbáljuk ki. Gépeled be: **DIR /?**

A képernyőn megjelennek a lehetséges beállítási módok. Ezek magyar nyelvű magyarázata a Függelékben megtalálható.

DIR

Paraméterek nélkül megadva a parancsot, az az aktuális könyvtárra vonatkozik. Kiírja a képernyőre a könyvtárat tartalmazó lemez címkéjét és sorozatszámát, a könyvtárban található állományok és alkönyvtárak nevét, kiterjesztését, méretét byte-ban, és az utolsó módosításuk idejét. Ezután megjeleníti az állományok számát, az általuk elfoglalt helyet, és a lemezen még meglévő szabad hely méretét. Például:

```
COMMAND COM 54500 93-09-27 6:20
      név      kit      byte  év-hó-nap  óra:perc
```

(A dátum és idő kijelzési formája a gép beállításától függ.) A <DIR> szócska utal arra, hogy ez egy alkönyvtár név, ami fájlokat tartalmaz, ha nem üres.

Például:

```
BASIC <DIR>          92-11-17 12:08
```

A dátum és az idő a létrehozás időpontját jelöli.

Feladatok

Próbáld ki a következő parancsokat!

16. Vizsgáld meg a képernyőn a kiírás eredményeit, majd írd a pontozott részre a parancs hatását!

```
dir a: .....
dir c:\dos /p .....
dir c:\dos /w .....
dir c: /s .....
dir a:\e*. sys .....
dir c:\*.exe .....
dir a: /a:h .....
dir c:\dos /o:n .....
dir c:\dos /o:-s .....
dir c: /o:g/s .....
dir c:\dos\?c*.* .....
dir c:\dos\??a*.* .....
dir k: .....
```

17. Írassd ki a képernyőre a DOS alkönyvtárban található minden C betűvel kezdődő,

.....
COM kiterjesztésű,

.....
C betűvel kezdődő, SYS kiterjesztésű fájlt!

.....

18. Írasd a képernyőre az FDD minden alkönyvtárának nevét!

.....

19. Képernyőnként lapozva szeretnénk megnézni a HDD minden olyan fájlját, amelynek második betűje C és EXE kiterjesztésű!

.....

Az eddigiekből már tudjuk, hogy elől áll a fájl neve, ami maximum 8 karakter hosszú lehet (a könyvtárnév is). A végén egy legfeljebb 3 betűs szócska áll, amit a fájl kiterjesztésének nevezünk.

Indítható fájlok:

COM	command	(<i>kömand</i>)
EXE	execute	(<i>ekszikjút</i>)
BAT	batch	(<i>becs</i>)

Egy programot (a fent említett kiterjesztéseket is figyelembe véve) elindítani úgy lehet a DOS-ból, hogy begépeljük a fájl nevét, és megnyomjuk az **ENTER** billentyűt. A többi már a számítógép dolga. Először megnézi, van-e ilyen nevű BAT kiterjesztésű fájl. (Többször előfordul, hogy csak a kiterjesztésben különböznek a fájlok, mert nevükben azonosak.) Ha talált ilyet, azt indítja. Ha nem talált, a COM, majd az EXE kiterjesztésűek között keres. Ha rosszul írtuk be a nevet, vagy nincs ilyen fájl, **Bad command or file name** (*baed kömand o fájl ném*) helytelen parancs vagy állománynév hibajelzést ad.

A konkrét fájl elindításakor meg kell adni a kiterjesztését is:

FÁJLNÉV.KITERJESZTÉS

Ezekon kívül még sok más kiterjesztés létezik. Számunkra lényegesek a következők:

DOC	programleíró állományok
TXT	szövegfeldolgozó állomány
BAS	BASIC nyelv állományai

A DOS-parancsok kiadásánál többször szükség van arra, hogy egy parancsot ismételten beírjunk, vagy az előzőleg beírt parancs minimális módosítását szeretnénk elérni. A számítógép, amikor beírtunk egy parancsot és megnyomtuk az **ENTER** billentyűt, egy úgynevezett átmeneti tárolóban is elhelyezi az általunk beírtakat. **Egy parancsot ismételten a képernyőre írni az F3, az F1 vagy a → billentyűvel lehet.**

FORMAT (*fómet*) lemezformázás, törlés (külső parancs)

E parancs nélkül nem minden megvásárolt lemezt lehetne használni. A parancs hatására a gép előkészíti a lemezt arra, hogy a DOS használhassa. Létrehozza a gyökérkönyvtárat, a FAT (információs) területet, felderíti a hibás lemezterületeket,

törli a lemezen esetleg tárolt adatokat. Az előzőekben már végrehajtottunk egy ilyen műveletet. Az utasítás összes paraméterét a

FORMAT /?

paranccsal írathatjuk ki a képernyőre. A magyar nyelvű magyarázat a Függelékben található.

Feladat

20. A következő parancsokat az FDD-n próbáld ki! A pontozott részre írd oda a megfelelő parancs hatását!

format a:

format a: /s

format a: /s/q

format a: /q/4

format a:/s/b

Ha a winchestert formázzuk, akkor a következő figyelmeztető üzenetet kapjuk.

WARNING! ALL DATA ON NON-REMOVABLE DISK

DRIVE C: WILL BE LOST!

Proceed with Format (Y/N)?

(A HDD teljes tartalma elveszhet! Mégis formázzuk a lemezt?) Az Y (Yes) válasza megkezdja a formázást, ha az N (No) betűt választjuk, nem hajtja végre a parancsot. Vigyázzunk arra, nehogy olyan lemezt formázzunk újra, amelyen esetleg pótolhatatlan adataink vannak. Ugyanez a helyzet a winchester esetén is. Ha mégis sikerült így letörölni egy „pótolhatatlan” lemezt, a DOS egy nagyon jó lehetőséget kínál a lemez tartalmának visszaállítására.

UNFORMAT (ánfómet) lemeztartalom visszaállítása a FORMAT után (külső parancs)

A parancs segítségével visszaállíthatjuk a FORMAT paranccsal újra formázott lemez tartalmát. A parancs a merevlemez hibás partíciós adatait is korigálja. Formája:

UNFORMAT meghajtó:

Az adott meghajtón levő lemez adatait visszaállítja . Próbáld ki!

unformat a:

Hatására az előzőleg formattált lemezünk adatait állítja vissza.

Ha a lemezt a FORMAT /U paranccsal formáztuk, akkor az UNFORMAT hatástalan marad.

VOL (VOLume, *voljúm*) lemeznév lekérdezése (belső parancs)

Hatására a gép kiírja az adott meghajtóban levő lemez nevét. Formája:

VOL meghajtó:

LABEL (*lébl*) lemeznév megváltoztatása (külső parancs)

Az új nevet ugyanúgy kell megadni, mint a FORMAT parancsnál a /v kapcsoló esetén. Ha csak ENTERt ütünk, a lemezek nem változik meg a neve.

Feladatok

21. Változtasd meg a lemez nevét a saját keresztnevedre!

.....

22. Változtasd meg a HDD nevét WINCHESTERre, majd kérdezd le!

.....

DISKCOPY (*diszkopi*) teljes lemezek másolása (külső parancs)

Ehhez nem kell formattálni a lemezt, mert az a másolás közben úgymint megtörténik. Csak hajlékonylemezek esetében használható. Nem lehet vele egy lemezt winchesterre másolni. A parancs speciális kapcsolóit a Függelékben találhatjuk meg. Tedd be a formázott lemezt az A meghajtóba, s készíts elő egy másikat!

diskcopy a: b:

Ekkor egy felszólítást kapunk:

Insert source diskette in drive A: (*inzőt szousz diszket in drájev A*)

Már betettük a másolandó lemezt az A meghajtóba, így csak ENTERt kell ütnünk. A gép beolvassa a lemez tartalmát a memóriájába, majd ismét felszólít bennünket:

Insert target diskette in drive B: (*inzőt taagit diszket in drájev B*)

Ha csak egy FDD van a gépbe építve, akkor ki kell cserélni az előző lemezt a céllemezre. Kis idő múlva a másolás megtörtént, amit a gép jelez is. 1.2 Mbyte-os lemezek másolása esetén a cserére felszólítást – a gép memóriájának méretétől függően – több esetben is megkaphatjuk, mert a másolás idején a másolandó lemez tartalma a memóriából kerül a céllemezre. Addig cserélgessük a lemezeket, míg a teljes másolat el nem készül. A másolás végét a következő felirat jelzi:

Diskcopy another (Y/N) (*diszkopi önádö*)

Az N betűt megnyomva befejeződik a másolás, Y-ra újabb másolást indítunk.

COPY (*kopi*) fájlok másolása (belső parancs)

Ez talán az egyik leggyakrabban használt DOS-parancs. Egy vagy több fájl másolására használható. Lehetőség van a fájlok összefűzésére is. Más rendszeregyeségről is lehet másolni (pl.: billentyűzetről nyomtatóra). Formája:

COPY [útvonal] mit cél [név]

útvonal: meg kell adni azt a könyvtárnevet, ahol a másolandó fájl(ok) található(k), ha abban a könyvtárban állunk, ahol a másolandó van, akkor elmaradhat.

mit: ide írjuk a másolandó fájl(ok) nevét és a kiterjesztését.

cél: megadjuk azt a könyvtárnevet, ahova másolunk.

név: ez lesz a másolt fájl új neve (csak akkor adjuk meg, ha meg akarjuk változtatni a fájl nevét a célkönyvtárban).

Ha nem adjuk meg a célfájl nevét, akkor az MS-DOS a másolatot az eredeti fájlnevével, dátumával és idejével készíti el. Ha nem adjuk meg a célkönyvtárat, az új fájl az aktuális meghajtón az aktuális könyvtárba másolja. Az állományokat nem lehet önmagukra másolni. A nevek megadásánál a * és a ? jelek ugyanúgy használhatók, mint a DIR parancsnál.

A honnan és a hova részek helyén nemcsak fájlnevek, hanem standard egységnevek is állhatnak:

Standard egységnevek a következők:

A, B	hajlékonylemezes egységek
C, D, ...	winchesterek
CON	konzol (billentyűzet)
LPT1	nyomtató

MOVE (mív) fájlok átmozgatása (külső parancs)

A parancsszó ugyanaz, mint a könyvtár átnevezése esetén, de most más paraméterekkel, fájlok mozgatására használjuk.

A COPY parancsnál másolatot készítettünk a fájlról. A fájl mindkét helyen szerepel a tartalomjegyzékben. A MOVE parancssal átmozgathatjuk a fájlt. A parancs az eredeti helyéről törli az állományt. A használata megegyezik a COPY parancsnál tanultakkal, ezért nem részletezzük.

Feladatok

- 23.** Formázd a lemezed úgy, hogy ne kerüljenek rá a rendszerfájlok, majd hozd létre a lemezen a PELDA1, a PELDA2, a PELDA3, illetve a SYS alkönyvtárakat! Írd a pontozott részre a használt parancsokat!

.....

- 24.** Mi a hatása a következő parancsnak:

copy c:\command.com a:

.....

- 25.** Az alábbi parancsok hatását vizsgálj meg a számítógépen, majd írd a pontozott részre a tapasztaltakat!

copy c:\dos\a*.* a:\pelda1

.....

c:\dos\?a*.* a:\pelda2

.....

copy c:\dos*.* a:\pelda3

.....

26. A winchester DOS alkönyvtárából minden COM kiterjesztésű fájlt át szeretnénk másolni az FDD SYS alkönyvtárába. Írd le az ezt megvalósító parancsot!
-

27. Mi a hatása az alábbi parancsoknak?

move a:\pelda1*.* a:\pelda2

.....

move a:\pelda3*.* a:*.*

.....

XCOPY fájlok csoportos másolása (külső parancs)

A COPY paranccsal ellentétben itt lehetőség van akár egy teljes alkönyvtár összes alkönyvtárának, illetve fájljának átmásolására. Formája:

XCOPY forrás cél[kapcsolók]

forrás: ide írjuk azt az alkönyvtárnevet, ahonnan másolunk;

cél: azt az alkönyvtárnevet írjuk be, ahova másolni szeretnénk;

kapcsolók: az összes kapcsoló és hatása a Függelékben megtalálható. Itt a leggyakrabban használtak adjuk meg csupán:

/S: hatására az adott alkönyvtár minden alkönyvtárát is átmásolja. Nélküle megadva a parancsot, csak az adott alkönyvtár fájljait másolja át.

Feladatok

28. Írd le, mit tapasztalsz, ha a COPY paranccsnál használt lemezre kiadjuk a következő parancsot!

xcopy a:\pelda1 a:\pelda2\pelda5

.....

29. Mi a hatása a következő parancsoknak?

XCOPY A: B: /D:13/01/92 /V

.....

30. Másold át a winchester egy olyan alkönyvtárát lemezre, melynek további alkönyvtárai vannak! (Vedd figyelembe a lemez méretét!)
-

SYS rendszerfájlok átmásolása (külső parancs)

A FORMAT /B paranccsnál helyet biztosítottunk a két rejtett rendszerfájlnak, de még nem tudtuk átmásolni őket erre a helyre a COPY paranccsal. Erre jó a SYS parancs. Formája:

SYS c: a:

Így átmásolásra kerülnek az IO.SYS és az MSDOS.SYS rejtett fájlok is.

DEL (DElete, *dilit*) vagy **ERASE** (*iréz*) fájl törlése (belső parancs)
NEM TÖRÖLHETŐK:

- a DOS-rendszer rejtett fájljai;
- a csak olvasható fájlok;
- a lemez tartalomjegyzéke.

Az előzőleg a COPY illetve az XCOPY parancsnál használt lemezen dolgozunk.

Feladat

31. Az alábbi parancsok hatását írd a pontozott részre!

del a:\pelda1*.sys

.....

erase a:\pelda2\backup.*

.....

del a:\pelda3*.txt /p

.....

erase a:\sys\a*.sys /p

.....

UNDELETE (*ándilit*) törölt fájlok visszaállítása (külső parancs)

Feladat

32. Add ki a következő parancsokat, majd írd a pontozott részre hatásukat!

undelete a: /list

.....

undelete a:\pelda1 /all

.....

undelete a:\sys*.* /all

.....

REN (RENAME, *riném*) fájl átnevezése (belső parancs)

REN [útvonal] régi.név új.név

[útvonal] **régi.név:** az átnevezendő állomány(ok) helyét és nevét adjuk meg.

új.név: az állomány(ok) új neve.

Ha az új.név már létező állomány, vagy az átnevezendő állományt nem találja, akkor a következő üzenetet kapjuk:

Duplicate file name or file not found (*djuplikit fájl ném or fájl nát fáund*)

A ? és a * használható, de csak a régi névben.

Feladatok

33. Másold a lemez főkönyvtárába a **command.com** fájlt, majd nevezd át **parancs.nok** nevére!

.....

34. Módosítsd a lemezen található **sys** kiterjesztésű fájlok nevét **vas** kiterjesztésűre!

.....

35. Az FDD minden olyan állományát, melyek második betűje **A**, nevezd át úgy, hogy az említett betű **C** legyen.

.....

TYPE (tájp) szövegfájl kiírása (belső parancs)

Ha meg szeretnénk nézni, mit tartalmaz a fájl, ezt a parancsot használjuk. Természetesen csak a már említett TXT, illetve BAT kiterjesztésű fájlok megtekintésének van értelme, mert ezek szöveget tartalmazó fájlok. Vizsgáljuk meg a winchesteren levő AUTOEXEC.BAT fájl tartalmát!

type c:\autoexec.bat

Írd ide, mit tartalmaz!

.....

.....

.....

Ha a kiírás hosszabb, mint amennyi a képernyőre fér, használjuk a billentyűzet PAUSE gombját vagy a CTRL és az S billentyűkombinációt. Bármely billentyűre folytatódik a lista. A kiírást megállítani a CTRL és BREAK gombokkal lehet.

Feladat

36. Az alábbi parancsok kiadása után írd le röviden a pontozott részre, mit tartalmaz az adott fájl!

type c:\dos\readme.txt

.....

type c:\command.com

.....

CLS (CLear Screen, kliö szkrín) képernyőtörlés (belső parancs)

Hatására a képernyő tartalma törlődik.

DATE (dét) a gépi dátum lekérdezése, módosítása (belső parancs)

A parancs hatására a gép kiírja az aktuális dátumot, és kéri az újat. (A dátum alakját is kiírja). Az IBM AT gépek beépített belső órával rendelkeznek, így a dátumot, s később az időt sem kell állandóan beírni, mert ezeket a gép kikapcsolásakor sem felejtje el. Formája:

DATE

Az új dátumot a kijelzett régivel azonos formában kell beírni. Egy lehetséges formátum a következő:

dd-mm-yy,

ahol **d** nap, **m** hónap, **y** év.

Ha nem írunk be semmit, és **ENTER**rel válaszolunk, az előző dátum marad meg. A dátum kiírásánál a napok angol neveinek rövidítése is kiíródik, de ezt nem kell megadni az újnál.

SU	SUNDAY	(számdi)	Vasárnap
MO	MONDAY	(mámdi)	Hétfő
TU	TUESDAY	(tjúzdi)	Kedd
WE	WEDNESDAY	(venzdi)	Szerda
TH	THURSDAY	(szőzdi)	Csütörtök
FR	FRIDAY	(frájdí)	Péntek
SA	SATURDAY	(szetődi)	Szombat

TIME (*tájm*) a gépi idő lekérdezése, módosítása (belső parancs)

Ugyanúgy működik, mint a **DATE** parancs, csak itt a gép által adott pontos időt lehet módosítani. Formája:

TIME

Felírja az általa mért időt századmásodperc pontosan, majd új idő megadását várja. **ENTER**rel elfogadhatjuk vagy beírhatjuk az új időt. Egy lehetséges formátum a következő:

hh:mm:ss,

ahol **m** a perc, **h** az óra, **s** a másodperc rövidítése.

VER (*VER*sion, *vőzsn*) a gépen futó DOS-verzió lekérdezése (belső parancs)

Hatására a képernyőre írja a gépen futó DOS-verzió nevét és számát. Formája:

VER

Írjuk be, és figyeljük meg az eredményt a képernyőn!

PROMPT (*prompt*) a bejelentkezési szöveg átírása (belső parancs)

A gép parancsfogadási készségét a bejelentkezés (a prompt) jelzi. Ez alaphelyzetben a winchesterről történt indítás esetén **C:\>** alakú. Ha más bejelentkezési szöveget szeretnénk, akkor át lehet írni:

PROMPT bejelentkezési szöveg és paraméterek

Az így beírt szöveg lesz a gép kikapcsolásáig a bejelentkezés.

(Nem lehet a bejelentkezési szövegben az **ENTER**, illetve a **\$** jel sem. A szövegben a gép különbséget tesz a kis és nagybetűk között.) Próbáld ki:

PROMPT Kutyuli

Most viszont fogalmunk sincs , melyik meghajtón vagyunk. Írd át:

PROMPT Kutyuli \$t\$h\$h\$h\$h\$h\$h\$h\$h\$

Így a bejelentkezési szövegben szerepel a „Kutyuli” szó, a pontos idő, a meghajtó neve, valamint a **:** jel.

A leggyakrabban használt bejelentkezési jelet megmutatjuk:

PROMPT \$PSG

A **PROMPT** egyéb paramétereit a Függelékben megtalálod.

Indítólemez készítése

Feladatok

37. Az eddig használt lemezt tedd be az FDD-be, majd formázd úgy, hogy a rendszerfájlok is rákerüljenek!
38. Másold rá a winchester DOS könyvtárából az alábbi fájlokat a lemezen létrehozott DOS alkönyvtárba:

CHKDSK.EXE	DISKCOPY.COM	FDISK.EXE
FORMAT.COM	SYS.COM	TREE.COM
UNDELETE.EXE	UNFORMAT.COM	XCOPY.EXE

Ha az indítólemez főkönyvtárában van

AUTOEXEC.BAT

nevű fájl, akkor ennek lefuttatása is megtörténik az indítás során. Ebben a fájlban állítjuk be a gép indulásakor megadható paraméterek egy részét. Többek között a következőket szerepeltethetjük benne:

- a DOS verzió számának kiírása;
- a PROMPT beállítása;
- a lehetséges keresési utak megadása;
- az indítani kívánt programok (keresési úttal).

Ezt a rövid programot közvetlenül a lemezre írjuk.

COPY CON rövid program lemezre írása

Formája:

COPY CON név.kiterjesztés

Az előbbi parancsot alkalmazva az A meghajtóban lévő lemez főkönyvtárába fogjuk másolni a fájlt.

copy con a:\autoexec.bat

@echo off

cls

ver

date

time

prompt \$p\$g

path=a:\;a:\dos;a:\virus

^Z

letiltjuk a gép szokásos kiírásait;

töröljük a képernyőt;

ezzel íratjuk ki a DOS-verziót;

dátum kiírása, módosítási lehetősége;

idő kiírása, módosítási lehetősége;

a már tanult bejelentkezési szöveg beállítása

a keresési utak beállítása. A VIRUS alkönyvtárat majd később használjuk.

a CTRL és a Z megnyomásával vetünk véget a beírásoknak. E parancs kiadásával a lemezre is másoljuk a programot.

A „szöveg vége” jel előtt még célszerű megadni azon fájlok neveit is, amelyek elindítása szükséges a használat során.

Kiegészítések az autoexec.bat fájlok írásához

Szövegsorok megjelenítéséhez:

ECHO szövegsor célszerű maximum egysornyi szöveget írni

Várakoztatás:

PAUSE

hatására a program futása egy billentyű megnyomásáig megszakad.

A BAT kiterjesztésű állományokban futtatható programok egymást követő indítását rögzítjük. Az ilyen fájlokat batch-programoknak is hívjuk. Próbáld meg más batch-programok írásánál felhasználni az említett parancsokat! Írj saját batch-programot, amellyel egy játékprogramot tudsz elindítani! A következő minta a GAMES alkönyvtár WOLF alkönyvtárában a WOLF.EXE fájllal indítja a játékot. Az ehhez szükséges indító fájlt a HDD főkönyvtárába írjuk meg.

COPY CON C:\WOLF.BAT

CD\GAMES\WOLF

WOLF

^Z

Az AUTOEXEC.BAT fájl PATH sorában szerepel a HDD főkönyvtárának elérési útja. Így a játékprogramot WOLF néven indíthatjuk anélkül, hogy belépnénk az adott alkönyvtárba. További utasításokat és példákat a DOS-kézikönyvekben találtak.

Ezzel tehát már van önálló indítólemezünk. Nézd meg, mit mutat a tartalomjegyzék, s írd a pontozott részre!

.....

 A lemezről történő indításhoz nyomd meg a RESET gombot a gépen, ha a lemez az FDD-ben van, vagy a **CTRL-ALT-DEL** billentyűkombinációt! A dátum és az idő beírásánál a tanultak szerint járj el!

A számítógép elindulásakor – amennyiben a főkönyvtárban megtalálható – lefut a CONFIG.SYS fájl is. Ebben a fájlban a gép környezetét beállító parancsokat sorolhatjuk fel. A pontos részleteket, ha érdekel tanárodtól megtudhatod.

Feladat

39. Hozz létre a lemezen egy VIRUS alkönyvtárat! Másold bele a HDD-ről például a CHKVIR (vagy más vírusölő) rendszer összes fájlját!
40. Járj utána mit jelentenek a számítógéped CONFIG.SYS fájljában található sorok! (A fájl tartalmát csak tanárod engedélyével módosítsd!)

VÍRUSOK

A számítógép-programozótól függ, hogy az összegyűjtött tudásával milyen feladatokat ad a gépnek. A könyvelés, a rajzolóprogramok, a játékok, a háborús előkészületek tervei, a repülőgépek navigációs anyagai, a dokumentumok jól megférnek egy winchesteren. A számítógépvírusok története főleg a háborús előkészületekkel függ össze.

A vírusok története

1957-ben ismerték fel először, hogy az egy pontból kiinduló járványok kitűnően modellezhetők számítógép segítségével. Ezt a matematikai tanulmányt, amely messzemenő következtetéseket vont le az önreprodukáló (önmagát megismétlő) programokra vonatkozóan, egyszerűen elfelejtették. 1974-ben jelent meg a programvírus, illetve vírusprogram fogalma.

1980-81-ben J. Klauss Dortmundban foglalkozott azzal, hogyan lehetne háborús előkészületek esetén megbénítani az ellenség számítógépes rendszerét. (Jellemző a téma komolyságára, hogy az Öböl-borúban az amerikai ügynökök számítógépvírusok segítségével tették ártalmatlanná az iráni légvédelmi, illetve távközlési rendszert. E vírusok megírására még tíz évvel korábban a PENTAGON írt ki pályázatot.)

1983-ban Friedrich Cohennek a Dél-Kaliforniai Egyetem számítógéphálózatán végzett kísérlete megdöbbentő eredményt hozott. A vírus elindítása után azonnal megfertőződött 33 rendszerállomány, 18 másodperc múlva minden felhasználói állomány. A 600-adik másodpercben teljessé vált a fertőzés.

1984-ben a Der Spiegel című magazin hívta fel a figyelmet az önreprodukáló programok léteire. Ugyanebben az évben jelent meg az első vírus Magyarországon, Commodore 64-es számítógépen. Terjesztője egy kisiparos volt, aki így akarta forgalmát növelni. (A vírus a lemezegység író-olvasó fejét állította olyan pályára, hogy azt visszaállítani csak szétszereléssel lehetett.) Ezt a vírust használták először másolásvédelemre egy könyvelőprogram esetében.

1989-ben már 30 vírust ismertek. Ebben az évben jelent meg Magyarországon a *Péntek 13.* nevű vírus, mely az izraeli eredetű, *Jerusalem* nevű vírus átírt változata. Az utóbbit a palesztinok csempészték be a Hebrew University számítógéphálózatába azzal a céllal, hogy onnan az összes megfertőzött állományt törölje, pénteken, 13-án. E program átírt változata lett a *Reboot* vírus, melyről nem tudni, miként került át Oroszországba. Innen már *Péntek 13.* néven került egy magyarországi bányavállalat számítóközpontjába, majd több intézmény számítógépparkját is megfertőzte. Több magyar átírata is van. Pl: *Kedd 1.*

1990–91-ben már Magyarországon is pusztított a *Stoned* vagy más néven *Marihuana*, illetve a *Yankee Doodle*. Az ismert vírusok száma 150-re nőtt, melyek közül a legtöbb tajvani eredetű volt. A nyugati fejlesztésű „remekművek” (*Michelangelo*, *Dir-2*) mellett már magyar fejlesztésű vírusok is megjelentek (*Pif-Paf*, *Phantom*). A ismert *lopakodó* vírusok mellett megjelentek az *alakváltó*, *szimbiózisban* (szerves együttműködésben) *élő*, illetve *többkomponensű* vírusok. A legújabbak az önállóan *fejlődni képes rendszerek*.

A számítógépek és a szoftverek elterjedése óriási hasznot hozott a fejlesztőknek. Akik nem léptek időben, olyan hátrányba kerültek a világpiacon, hogy azt már nem tudták volna behozni. Az egyre nagyobb energiákat (pénzt) felemésztő szoftvereket legtöbbször nem a „boltokban” vásárolták meg, hanem egyszerűen ellop-ták. Így a forgalmazók óriási hasznának jelentős része eltűnt a tolvajok kezében. A szoftverfejlesztők a programok megvédése céljából olyan kis programrészeket fejlesztettek ki, melyek az illegális másolatok esetében, az új „tulajdonosok” gépeinek memóriájába befészkeltek magukat, s egy idő múlva a gép használhatatlanná vált. Az ilyen programok kivétele a gép memóriájából természetesen új programot igényelt. Kis idő múlva már külön „iparág”-ként működött a vírusírás, illetve a megírt vírust „megölő” programok készítése. A világon eddig ismert vírusok száma több ezer, s ezek közül kb. 1500 azoknak a száma, melyek igazán veszélyesek.

A vírusölő programok a folyamatos fejlesztés ellenére alig ezer vírus felismerésére képesek. Az igazi probléma ott van, hogy egyes vírusok a számítógépekbe kerülve megváltoznak, úgynevezett „mutáns” alakot öltenek. Az eredeti vírust megölő program a mutáns alakot már nem ismeri fel, s ismeretlen vírusként terjed.

Vírusnak nevezünk tehát minden olyan programot, amely valamilyen kárt okoz a számítógép működésében, a háttértárain elhelyezett programok, adatok között. Más megfogalmazásban: A vírus önmagát más programok módosításával sokszorozó program. A lényeg tehát más programok megfertőzésén, s nem a kár-okozáson van.

A vírusok terjedésének megakadályozására minden lemezt, programot, ami kikerült a kezedből, vagy nem kellően ismert helyről származik, meg kell vizsgálni.

Mikor kell vírusra gyanakodni?

- A gépen levő programok vagy a kapott programok közül több nem, vagy szokatlan módon indul, működik.
- A COMMAND.COM fájl hossza megváltozik .
- Az operációs rendszer betöltése nem a szokásos sebességgel történik.
- Szokatlan hibaüzenetek jelennek meg a DIR parancs hatására és ismeretlen nevű, sokszor krikzkraksz állománynevek.
- Rendellenes jelenségek történnek a képernyőn (potyognak a betűk, labdák ugrálnak, a gép zenélni kezd stb.).

A vírusok fajtái

- Boot-vírusoknak nevezzük a vírusok azon fajtáját, amely a rendszer feléledésekor aktivizálódik. Ezek a vírusok azt használják ki, hogy a DOS induláskor betölti és elindítja az operációs rendszert. A betolakodó elhiteti az éledő rendszerrel, hogy ő az igazi operációs rendszer, majd miután megtette a magáét, betölti az eredeti operációs rendszert, azaz leplezi a jelenlétét.

Az ellenük való védekezés egyik hatásos eszköze, ha bekapcsoláskor, rendszerindításkor nincs lemez az FDD-ben.

- A fájl indításakor aktivizálódó vírusok valamely futtatható program elindításával kezdik el hatásukat kifejteni. Előfordul, hogy ezek is lefuttatják az eredeti programot is – azaz hatásuk szinte észrevehetetlen –, de gyakran teszük tönkre az eredeti programot, ami viszont teljesen észrevehető.

Mi a teendő vírusfertőzés esetén?

- Indítsd el a CHKDSK parancsot, mely információt ad a lemez kapacitásáról, a foglalt és a szabad lemezterületről, rejtett állományok méretéről, hibásnak minősített helyekről!
- E számok ismeretében, megfelelő gyakorlattal el lehet dönteni, van-e vírus a gépben.
- Indíts el egy vírusölő, vagy egy víruskereső rendszert! **Fontos tanács!** Egyetlen vírusölő rendszert sem célszerű winchesterről indítani, mivel ezen rendszerek nagy része a RAM-ba töltődik, ahol esetleg már tárolt programok vannak, s memóriahiány miatt használhatatlanná válnak. Előfordulhat az is, hogy maga a rendszer már vírussal fertőzött. Ezért célszerű ezeket a vírusölő rendszereket külön lemezen, az írásvédő nyílást leraasztva, indítólemezként tárolni.

Mi itt a SCAN rendszert, a FÜGGELÉKben pedig a CHKVIR vírusölő rendszert ismertetjük. Természetesen e két rendszeren kívül több más is létezik (CHKSEQ, SYSDOKI, TNT, NAV, F-PROT stb.), s a fejlesztések nyomán egyre újabbak jelennek meg.

A VIRUSCAN programcsomag használata

A SCAN általános víruskereső rendszer. Folyamatosan fejlesztik, így kb.: 2–3 havonta jelenik meg egy új verzió. A víruskereséshez lehetséges beállításokat a SCAN /? parancs segítségével írathatjuk a képernyőre. A programcsomag egy korábbi verziójának lényegesebb paramétereit adjuk meg az alábbiakban:

- \ a gyökérfájlyvtárat, illetve a bootszekort nézi;
- /A az adatfájlylokat is vizsgálja;
- /BELL hangjelzést ad, ha vírust talál;

- /CHKHI 0 -tól 1088 Kbyte-ig ellenőrzi a RAM tartalmát;
- /D felülírja és törli a fertőzött programokat;
- /DATE feljegyzi a vizsgálat időpontját;
- /MANY több floppy ellenőrzése;
- /NOMEM nem ellenőrzi a memóriát;
- /SUB a megadott könyvtárban levő alkönyvtárakban is keresi a vírusokat;

Példa**SCAN C:****SCAN C: /CHKHI /BELL**A megtalált vírusokat a **CLEAN.EXE** segítségével törölhetjük.**Példa****CLEAN C: D: [vírusnév]**

A szögletes zárójelbe a SCAN által megadott nevet kell beírni.

A vírusvédelmi rendszer tagja a **VSHIELD** nevű program is. A memóriába töltődik. Hatására vírus esetén nem lehet a fertőzött programot indítani, másolni. Kisebb változata a **VSHIELD1**, amely nem vizsgálja a bootszektor.

A **VSHIELD /COPY** figyel a DOS-parancsok segítségével végrehajtott másolásokat és a **DIR** parancsnál a bootszektor.

Hálózatok ellenőrzésére a **NETSCAN** programot használjuk .**A CHKVIR rendszer**

A hazai eredetű vírusok közül többet felismer, mint a SCAN. Használható teljes képernyős, menürendszerrel ellátott üzemmódban, de parancs üzemmódban is a menürendszer meghívása nélkül.

CHKVIR /? hatására a képernyőre írja a lehetséges paraméterekeket:

- /A minden meghajtón keres.
- /D a fertőzött fájlokat kérdés nélkül törli.
- /F minden fájlban keres.
- /J a mutáns alakú vírusokat is keresi.
- /K a talált vírusokat kiirtja, ha tudja.

Használata:

Parancs üzemmódban:**CHKVIR A: /J/K****CHKVIR A:\DOS*.EXE /K**

A menürendszer segítségével: a teljes lemez vizsgálatára beindítjuk a rendszert:

CHKVIR

megjelenik egy menüsor (részletes leírás a *Függelékben*). Itt a teljes lemezre kiadjuk a víruskeresés, vagy vírusirtás parancsot. Vírus esetén egy ablakban hangjelzés kíséretében megjelenik a talált vírus neve. Eltávolítására a számítógép megerősítést vár. Vírus esetén általában megjelenik a vírus neve, melyet sok esetben azonnal el lehet távolítani.

Az említett rendszerek úgy is indíthatók, hogy a lemez AUTOEXEC.BAT programjában megadjuk a szükséges rendszer elérési útját. Így a rendszert indító fájl nevének begépelésével ugyan nem jelenik meg a menü, de a szükséges paraméterek beállításával a rendszer megvizsgálja az adott lemezt, illetve az adott alkönyvtárat.

Feladat

1. Vizsgáld meg, van-e vírus a lemezen! Lépj be a lemez VIRUS alkönyvtárába, s indítsd el a benne levő programot! Az előbb leírt példák alapján parancs, illetve menü üzemmódban is vizsgáld meg az általad választott meghajtók adott alkönyvtárainak fájljait!

Védekezés a vírusok bekerülése ellen

Hardvervédelem

A beépített kártya a gép bekapcsolásával működésbe lép, figyeli a lemezműveleteket, s minden gyanús jelre figyelmeztet.

Már egyre több olyan alaplap kapható, amely tartalmaz valamilyen védekező mechanizmust, amely a vírusok jelentős részének bekerülését meg tudja akadályozni.

Szoftervédelem

A rendszer bizonyos jellemzőit elmentő, s később az elmentett állapottal összehasonlító programok. Például: CPAV, NAV, SCAN, VSHIELD stb. A védelem megszervezéséhez általában minden könyvtárba elhelyez egy-egy ellenőrző fájlt a program, ami lehetővé teszi az esetleges fertőzés esetén a visszaállítás könnyebb elvégzését.

A NORTON COMMANDER HASZNÁLATA

A tankönyv elején foglalkoztunk a DOS-parancsokkal. Az operációs rendszer használatakor sokat kell gépelni, és sok parancsot kell megjegyezni. Több olyan segédrendszert készítettek (DOS NAVIGATOR, DOS SHELL), amelyek egyszerűsítik a DOS fájl- és könyvtárkezelését. Ezeket a rendszereket keretrendszereknek nevezzük.

Az alábbiakban bemutatásra kerülő **NORTON COMMANDER 4.0** az egyik leggyakrabban használt keretrendszer.

A rendszer felépítése olyan, hogy indítása után a DOS-parancsok továbbra is használhatók, de a menüsor segítségével egyszerűbbé válik sok egyéb, a DOS-ban csak hosszú „pötyögés” után használható parancs.

Feladat

1. Készíts elő egy lemezt, formázd úgy, hogy a rendszerfájlok is rákerüljenek, majd hozz létre egy NORTON alkönyvtárat! Másold rá a Norton Commander rendszer fájljait! Készíts olyan AUTOEXEC.BAT fájlt, mely a lemezről történő indítás esetén induláskor betölti a keretprogramot!

A program indítása után az itt láthatóhoz hasonló lesz a képernyő.

Left		Files	Commands	Options	Right	18:17		
		C:\			D:\GAMES			
Name		Name		Name	Name	Size	Date	Time
LOTUS					..	▷ UP-DIR ◀	92-11-22	5:47
DOS					AMOBA	▷ SUB-DIR ◀	93-12-20	11:17
WINDOWS					BLOCK	▷ SUB-DIR ◀	94-01-20	12:47
WINWORD					WOLF	▷ SUB-DIR ◀	94-11-10	15:47
autoexec	bat							
config	sys							
command	com							
gmouse	com							
time	exe							
treeinfo	ncd							
time.exe		6573	90-12-29	5:47	..	▷ UP-DIR ◀	92-11-22	5:47

C:\>

1 Help 2 Menu 3 View 4 Edit 5 Copy 6 Remove 7 Mkdir 8 Delete 9 PullDn 10 Quit

A képernyőn két, jól elkülönülő ablak látható. A képernyő alsó sorában a PROMPT nem feltétlenül azonos a DOS-ban beállítottal. Valamelyik ablakban egy széles sáv van. Ezt a kurzormozgató nyilak segítségével mozgathatjuk, s széles kurzornak nevezzük. A NORTON COMMANDER 4.0-ás változatának részletes leírás-

sát a Függelékben találhatod meg. Az alábbiakban a legfontosabb parancsokat ismertetjük.

Meghajtóváltás

Amíg a DOS-ban egyszerre csak egy meghajtó tartalomjegyzékét láthatuk, addig itt a két ablak lehetőséget biztosít két lista kiíratására.

Az **ALT-F1** billentyűkombináció hatására a bal oldali ablakban megjelenik a gépen található meghajtók névsora. A kurzorral ráállva az egyikre és az **ENTER**t megnyomva, a választott meghajtó tartalomjegyzéke jelenik meg az ablakban.

Az **ALT-F2** lenyomásával a jobb oldali ablakban módosítható a kijelzésre kerülő meghajtó.

Ablakváltás

A **TAB** billentyűvel végezhető el.

Tartalomjegyzék

A **DIR** parancs helyett a megjelenő ablakban nagybetűkkel a könyvtárak, kisbetűkkel a főkönyvtár programjai láthatók.

Program indítása

Ráállunk a kurzorral az adott fájl névre, majd megnyomjuk az **ENTER**t.

Könyvtárváltás

A belépéshez a kurzorral ráállunk az adott alkönyvtár nevére, majd megnyomjuk az **ENTER** billentyűt. Ekkor az adott ablakban felül, középen megjelenik a keresési út, majd beállítástól függetlenül az ablak első sorában 2 pont.

Az alkönyvtárban mozogni a **PgUp**, a **PgDn**, a **Home**, az **End**, illetve a kurzormozgató billentyűkkel lehet.

A kilépéshez a tartalomjegyzék elején lévő két pontra állunk, és megnyomjuk az **ENTER** billentyűt.

Alkönyvtár létrehozása

Nyomjuk meg az **F7** billentyűt! A megjelenő ablakba írhatjuk a létrehozandó alkönyvtár nevét.

Alkönyvtár törlése

A törölni kívánt alkönyvtár nevére állunk, és az **F8** billentyűt lenyomjuk.

Feladat

2. Lapozz vissza a DOS-parancsokhoz (26. oldal)! Az ott található, könyvtárkezelésre vonatkozó feladatokat most a keretprogrammal hajtsd végre!

A menü aktivizálása

Nyomjuk meg az **F9** billentyűt! Az ablakból eltűnik a széles kurzor, és a képernyő legfelső sorába ugrik (menü). Visszalépni az **ESC** billentyűvel lehet. A Függelékben részletesen ismertetjük a keretrendszer menüpontjait.

Feladat

3. Állítsd be a bal oldali ablakot úgy, hogy a winchester tartalomjegyzéke – a legtöbb állományt tartalmazza:

.....

– a fastruktúrát tartalmazza:

.....

– névsorba legyen rendezve:

.....

– minden adatot tartalmazzon:

.....

Állományok másolása

A Norton Commander a következő módon segíti az említett parancs végrehajtását.

Egy fájl másolásához rá kell állunk a másolásra kiszemelt fájlra, majd a másik ablakban beállítjuk a célkönyvtárat (**ALT-F1** vagy **ALT-F2**), ha szükséges. Amennyiben a másik ablakba át kellett lépnünk (**TAB**), visszaugrunk a másolandó fájlra, majd megnyomjuk az **F5** billentyűt, és értelemszerűen válaszolunk a gép által feltett kérdésekre.

Több fájl csoportos másolásához álljunk rá az első másolásra szánt fájlra, majd nyomjuk meg az **INSERT** billentyűt! Hatására a fájl nevének kiírása megváltozik (pl.: más lesz a kijelzés színe stb.), ezáltal a fájlt kijelöltük. Ugyanezt tegyük a többi fájllal is, majd nyomjuk meg az **F5** billentyűt, és az **ENTER**t követően adjunk választ a feltett kérdésekre!

A szelektálást megszüntetni az **INSERT** billentyű ismételt megnyomásával lehet.

Ha **adott típusú fájlokat** szeretnénk másolni, akkor nyomjuk meg a numerikus billentyűzet **+** gombját. A megjelenő ablak mezőjébe beírhatjuk a szelektálás módját.

Feladat

4. Írd be a **+** gomb lenyomása után megjelenő ablakba a ***.???** jelsorozatot, és üsd le az **ENTER**t! Figyeld meg, milyen változások történtek a képernyőn!

Egy **alkönyvtár minden fájljának másolása** annyiban különbözik az előzőtől, hogy most nem írunk a numerikus billentyűzet **+** gombjának megnyomása után megjelenő ablak felirata (***.***) helyébe semmit, hanem

ENTERt ütünk. Hatására minden fájl nevének kijelzése megváltozik. Ezt a folyamatot is **SZELEKTÁLÁS**nak (SELECT GROUP) nevezzük. Megkezdhetjük a másolást az **F5** billentyű lenyomásával.

A szelektálást megszüntetni (UNSELECT) a numerikus billentyűzet - jelének lenyomásával lehet.

Fájlok mozgatva másolása

A DOS MOVE parancsához hasonló lehetőséget találunk a fájlkezelő keretprogramunk szolgáltatásai között. A másolás után az eredeti helyéről törli az adott fájlt vagy fájlokat. A parancsot az **F6** billentyűvel aktivizálhatjuk. Az eljárás egyébként megegyezik az **F5**-nél leírtakkal.

Feladat

5. Lapozz vissza a DOS-parancsok fejezetéhez (36. oldal)! Az ott található átmozgatási feladatokat most ezzel a keretprogrammal hajtsd végre!

Fájlok átnevezése

A fájlok mozgatva másolásánál (**F6** billentyű) megjelenő ablakba beírható a fájl új neve is. Célszerű, ha a jobb és bal oldali ablakban azonos könyvtárak vannak nyitva, mert más esetben (ha nem írjuk be az elérési útvonalat) a fájlt átmozgatjuk a másik oldalon nyitott könyvtárba.

Feladat

6. Lapozz vissza a DOS-parancsok című fejezethez (38. oldal)! Az ott található fájlatnevezési feladatokat most ezzel a keretprogrammal hajtsd végre!

Fájlok tartalmának megtekintése

A fájl nevére állva s megnyomva:

- az **F3** billentyűt, a fájl tartalmát a képernyőre írja;
- az **F4** billentyűt a fájl tartalmát a képernyőre írja, de szövegfájl esetén annak tartalma módosítható is;
- a **Ctrl-J** billentyűkombinációt együtt lenyomva a fájl tartalmát a képernyő másik ablakába írja.

A Norton Commander használata egérrel

A NORTON COMMANDER program használatának felgyorsítására is lehet használni az egeret. A két, illetve háromgombos egér bal, illetve jobbkezesek számára külön-külön beállítható. A kétgombos, jobbkezes egerek NORTON alatti használatát ismertetjük, de a balkezes egér használata ettől csupán annyiban különbözik, hogy a bal és a jobb gombok fel vannak cserélve. Aki tehát balkezes, annak nincs más tennivalója, mint következetesen az ellentétes oldali gombot megnyomni az egéren, ha a programot előtte beállították a balkezes használati módra.

A Norton bejelentkezése után a két ablakot elválasztó vonalon egy kis téglalap (színes képernyőn piros) jelenik meg. Az egeret mozgatva ezt a kis négyszöget irányíthatjuk.

A mozgatás során bármelyik fájlt **kijelölhetjük a bal oldali gomb** megnyomásával, amikor a nevéen állunk. Ha kétszer egymás után nyomjuk meg ezt a gombot, az adott fájlt elindítjuk (ha indítható). Ha alkönyvtár nevére mutatunk rá, s megismételjük a bal gomb megnyomását, bele tudunk lépni a könyvtárba. Ha az alkönyvtár olyan hosszú, hogy tartalma nem fér bele az ablakba, egyszerű dolgunk van. Álljunk rá az adott ablak alsó szélére, majd nyomjuk meg az egér bal oldali gombját. Az alkönyvtár tartalma felfelé végigfut az ablakban. Lefelé futtatni az ablak felső szélére állva lehet. Közben természetesen bármikor megállhatunk.

Ha egy fájl tartalmát (pl: *.TXT) meg szeretnénk nézni, akkor az **F3** billentyűt kell megnyomnunk. Ha a megtekintett fájl tartalma hosszabb, mint a képernyő mérete (ezt egyébként ekkor a felső sorban százalékban írt szám jelzi), akkor ugyanaz a teendők, mint az előbb.

A két ablak között az egér segítségével gyorsabban átléphetünk, mint a **TAB** billentyűvel, mert a kurzort csak át kell húzni és megnyomni a bal oldali gombját.

A **jobb szélső gomb** megnyomása esetén ugyanúgy kijelölhetjük a fájlt, mint az **Insert** gombbal. Ismét megnyomva a gombot, a kijelölés megszűnik. Különösen egy fájl vagy bizonyos fájlok másolásakor gyorsabban meg tudjuk oldani a kijelölést, mint az eddigiekben.

A beépített szövegszerkesztő használata

A szövegfájlok tartalmának vizsgálatára, módosítására az **F4** billentyűt használjuk, illetve egy eddig nem létező fájl létrehozására a **SHIFT-F4** billentyűket. Ekkor egy, a teljes képernyőt kitöltő ablakban megjelenik a szövegfájl tartalma, s lehetőségünk van annak módosítására. Ezt a **szerkesztőablakot EDITORnak nevezzük**. Az editort más módon is elérhetjük. A NORTON COMMANDER tartalmaz egy különálló programot (NCEDIT), amely segítséget ad szövegfájlok módosítására. Ezt a programot **egyszerű szövegszerkesztőnek** önállóan is használhatjuk.

A szövegszerkesztők olyan programok, melyek alkalmasak egy írógép helyettesítésére úgy, hogy a program az írógépnél több lehetőséget kínál a felhasználónak. A szövegszerkesztő programok (CHIWRITER, WORDSTAR, WORD for WINDOWS stb.) sokasága áll a felhasználók rendelkezésére. Mi a későbbiekben a WINDOWS (*uindóz*) rendszer WRITE (*rájt*) és WORD (*vőd*) szövegszerkesztőivel fogunk megismerkedni. Amíg erre sor kerül, egyszerűsége miatt a NORTON COMMANDER szövegszerkesztőjét fogjuk használni.

Az ékezetes betűk amerikai billentyűzeten való nehézkes begépelési lehetősége miatt, használjunk **magyar billentyűzetre alakító programot** is. (Például: a KEYBhu.COM vagy a CWI.COM fájlt.) Ezek az eddig hiányzó ékezetes betűket

megfelelő billentyűkhöz rendelik, ami jelentősen megkönnyíti a gépelésünket. Az ékezetes billentyűket be-, illetve kikapcsolni a használt programtól függő billentyűkombinációval lehet.

Feladat

7. Mely karakter megnyomására jelenik meg az ékezetes karakter? Töltsd ki a táblázatot! Ha valamely betűt nem tudod megjeleníteni, akkor maradjon üres a téglalap.

Á		É		Í	
á		é		í	
Ó		Ö		Ő	
ó		ö		ő	
Ú		Ü		Ű	
ú		ü		ű	

A program indítása

A NORTON COMMANDER keretrendszeren belülről történő indításról már volt szó, így most a DOS parancssoránál rendelkezésünkre álló lehetőségeket is ismertetjük. Célunk a PROBA.TXT szövegfájl létrehozása az A: meghajtóban lévő lemezre. Adjuk ki a következő parancsot, miután elindítottuk az ékezetes betűk írását megkönnyítő programunkat!

NCEDIT a:\proba.txt

Ha az A: meghajtón álltunk volna, akkor a parancs kicsit rövidebb is lehetne:

NCEDIT proba.txt

Amennyiben más néven akarunk létrehozni fájlt, akkor természetesen csak a fájl nevét kell megváltoztatnunk a parancssorban.

A NORTON COMMANDERben ugyanezt a **SHIFT-F4** billentyűk lenyomásával tehetjük meg, és a megjelenő ablakba beírjuk a fájl nevét.

Egyszerűsége miatt a továbbiakban a NORTON COMMANDERből indítjuk a szövegszerkesztőnket.

Hozzunk létre egy fájlt a lemezen:

SHIFT F4,

majd a megjelenő ablakba írjuk be:

a:\vers.txt!

Megjelenik az EDITOR ablaka, melynek bal felső sarkában a kurzor látszik.

A **legfelső sorban** a következők jelennek meg:

Edit: A:\vers.txt ezt a fájlt szerkesztjük;

Line 1 a kurzor az első sorban van;

Col 1 a kurzor az első oszlopban van;

53 000 Free ennyi szabad hely van még a szöveg bevitelére;

EOF a fájl végén állunk.

Gépeld be az alábbi szöveget !

**Egész úton – hazafelé –
Azon gondolkodám:
Miként fogom szólítani
Rég nem látott anyám?**

Az ékezetes karaktereket a tanult módon gépeld be! A sorok végén az **ENTER** leütésével tudsz új sort nyitni. Ha hiba történt a beírásnál, a kurzort mozgató billentyűk, a **DEL**, a **BACKSPACE** (*bekszpész*) felhasználásával javíthatsz. A gépelés befejeztével mentéssel lépj ki (**F10**).

Nézd meg a tartalomjegyzéket! Valóban ott szerepel a **VERS.TXT** fájl? Állj rá a széles kurzorral erre a fájlra, majd nyomd meg az **F4** billentyűt! A megjelenő ablakban látszik az általad begépelte vers.

Az eddigiekben leírt parancsokat megvizsgálva kevés olyan dolgot tapasztaltunk, mely különbözne egy írógép nyújtotta szolgáltatásoktól. Ha az írógépen javítani szeretnénk, az általában a papíron megjelent szöveg felülírását jelenti. Itt a felülírás könnyen elvégezhető, s már csak a hibátlan szöveget nyomtatjuk.

A lényeges különbség egy írógép és egy szövegszerkesztő között az úgynevezett **blokkműveletek**ben rejlik. Egy hagyományos írógépen nincs lehetőségünk az adott szöveg átszervezésére, az egyes részek sorrendjének (újraírás nélküli) megváltoztatására, bizonyos szövegrészek más helyre történő másolására. A szövegszerkesztő lehetőséget ad erre is. Állj rá a kurzorral a versszak első betűjére, majd nyomd meg az **F3** billentyűt! Ekkor az egész szöveg inverzbe vált. Állj a kurzorral a versszak végére, és ismét nyomd meg az **F3** billentyűt! Ezzel kijelöltünk egy **blokkot**. Most menj le a kurzorral a szöveg végére új sorba, majd nyomd meg az **F5** billentyűt! A blokk tartalma itt is megjelenik.

A szerkesztés közben kiadható parancsok felsorolását a Függelékben találod.

Feladatok

8. A szerkesztőt használva gépeld be a vers további versszakait!
9. Írj levelet a barátodnak, a nagyszüleidnek! A szöveget vedd fel lemezre, majd nyomtasd ki!
10. Ha van kedved, készítsd el az órarendedet a szövegszerkesztővel!

PROGRAMOZÁSI NYELVEK



A számítógépprogramokat speciális nyelveken írják. A különböző feladatok megoldásához különböző programnyelveket lehet használni a feladattól függően. Amit az egyik nyelven könnyedén megírhatunk, az a másikon nehéz is lehet. Valamennyi programozási nyelv közös tulajdonsága, hogy számítógépek vezérlésére alkalmas.

Egy lehetséges felosztás aszerint csoportosítja a nyelveket, hogy mennyire kell a természetes nyelven (pl.: angol) megfogalmazott utasításokat átdolgozni ahhoz, hogy ugyanolyan utasítást kapjunk az adott nyelven.

Gépikódú nyelv

Az adott számítógép mikroprocesszora által közvetlenül értelmezett utasítások halmaza. A gépi utasításokat bináris kódok formájában kell megadni. Ez adja az adott gép utasításkészletét (gépi kódját). Minden géptípus csak a saját gépikódú programját érti. (Ezért nem lehet pl. egy Commodore-programot IBM PC-n futtatni.) Futtatási idejük töredéke a magasszintű nyelvekének. Hátránya a mikroprocesszor-orientáltság és a nehézkes tesztelés.

Programozása nehézkes, a gyakorlatban nem használatos. Más nyelveken írt programokat fordítóprogramokkal alakítják át e nyelven futó változatra.

Assembly (eszömbli) nyelvek

A számkódok helyett szöveges szimbólumok formájában adják meg az utasításokat. A megírt programot a processzor átalakítás nélkül nem érti, ezért le kell fordítani gépi nyelvre. Jellemzője az egy az egyben fordítás. A fordító programot **assemblernek**, a visszaalakítót **disassemblernek** nevezzük. Olyan segédnyelvnek tekinthető, amely átmenet a gépikód és a magasszintű nyelvek között.

Magasszintű nyelvek

Gyakorlatilag függetlenek az alkalmazott géptől, s így a programozó a problémára koncentrálhat, és nem a gép sajátosságaira. A megírt **forrásprogramot** a programozási nyelv környezetében levő programmal kell **lefordítani** futtatható **tárgyprogrammá**. Ismertebb programnyelvek a következők:

FORTRAN

műszaki-tudományos problémák megoldására;

ALGOL

publikációs nyelv;

COBOL	kereskedelmi, ügyvitel-szervezési feladatok megoldására;
PL-1	tudományos és üzleti életben használatos nyelv;
BASIC (<i>bézik</i>)	kezdők általános célú programozási rendszere;
PASCAL (<i>pászkal</i>)	jól használható, általános célú programozási nyelv.

Logikai programozási nyelvek

Ezek olyan programnyelvek, amelyekben a program nem feltétlen lépésenkénti megoldási algoritmus, inkább a bonyolult probléma leírásának menete. Ilyen nyelvek:

PROLOG;
LISP;
MODULA.

Mi először a BASIC sajátosságaival ismerkedünk meg. Elsődlegesen a feladatok megoldási algoritmusait fogjuk vizsgálni, majd ezt követően fogjuk a programot megírni BASIC nyelven. A későbbiekben a PASCAL nyelv sajátosságaira is kitérünk.

A két nyelv szabályai, illetve a megismert algoritmusok segítségével könnyebb lesz az önálló problémák megoldása.

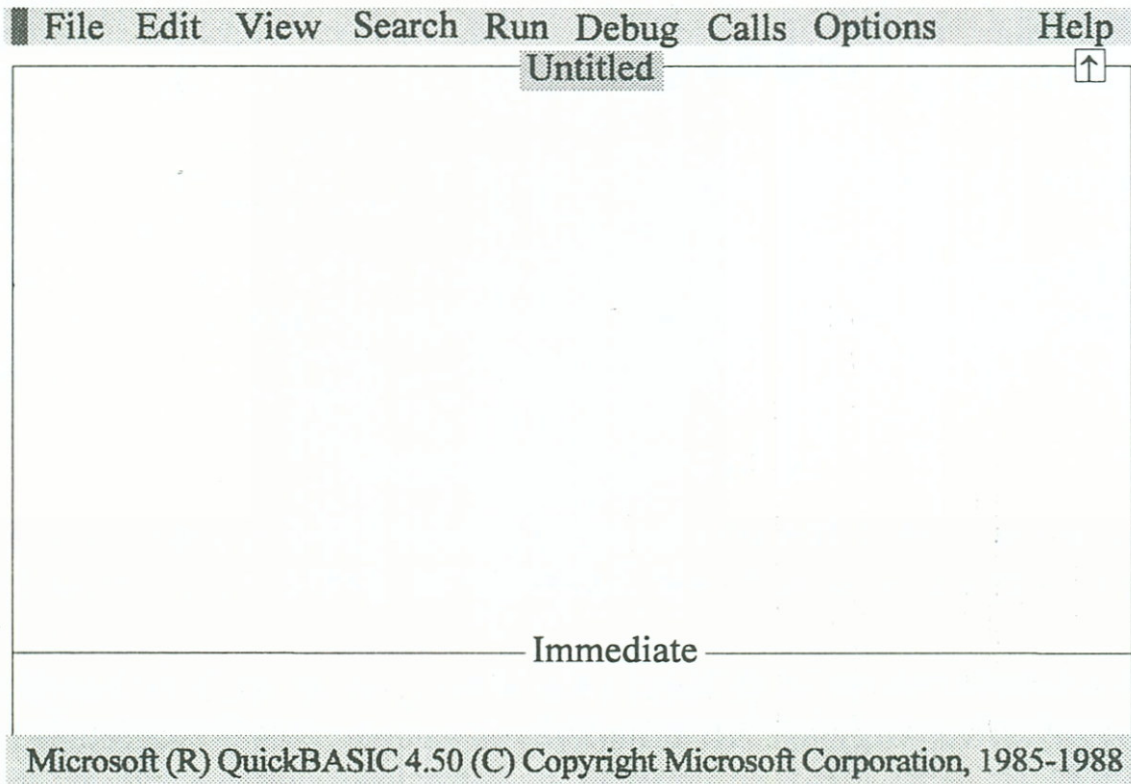
A BASIC nyelv IBM PC-s változatai közül a Quick BASIC-kel ismerkedünk meg. Az indító fájl a QB.EXE. A program egyszerűsített változata ma már az MS-DOS operációs rendszerrel együtt is megvásárolható. Elképzelhető, hogy az általad használt gépen más neve van (pl.: QBASIC.EXE) a teljes program indítófájljának, mert az egyszerűsített változatot nem célszerű átnevezni a DOS-könyvtárban.

Feladatok

1. Készíts elő egy új lemezt, s formázd úgy, hogy a rendszerfájlok rákerüljenek!
2. Másold rá erre a lemezre a Quick BASIC indításához és használatához szükséges fájlokat! Ha bizonytalan vagy, kérdezd meg tanárodat!
3. Írj olyan BASIC.BAT fájlt a lemezre, amely az erről történő indítás esetén azonnal betölti a QBASIC programot!
4. Hozz létre a lemezen egy alkönyvtárat, amelyben a későbbi programjaidat fogod tárolni!

QUICK BASIC

A program indítása a QB.EXE fájljal történik. A megjelenő képernyő az alábbiakat tartalmazza:



A menürendszer részletes leírását megtalálod a Függelékben!

A menüsor kiválasztása

Egérrel: A bal felső sarokban az egér kurzora látszik. A menüpontra állunk, s megnyomjuk az egér bal oldali gombját.

Billentyűkkel: A menüpontot az **ALT** és a menü kezdőbetűjének megnyomásával teheted aktívvá.

Almenük kiválasztása

A fő menüpont kiválasztása után legördülnek az almenük. Az itt található menüpontok kiválasztása történhet:

Egérrel: Ráállunk a megadott pontra, és megnyomjuk az egér bal oldali gombját.

Billentyűkkel: A **↑**, illetve a **↓** billentyűkkel ráállunk az adott menüpontra, majd lenyomjuk az **ENTER**t.

Program betöltése

Menüből: Válasszuk a menü **File** pontját, majd a legördülő almenüből az **Open ...** alpontot!

Open program

File **N**ame:

<p>Files</p> <div style="border: 1px solid black; height: 100px; width: 100%;"></div>	<p>C:\QBASIC</p> <p>Dirs/Drives</p> <div style="border: 1px solid black; padding: 5px;"> <p>.....</p> <p>PELDA</p> <p>[-A-]</p> <p>[-B-]</p> <p>[-C-]</p> </div>
----------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

< Ok > < Cancel > < **H**elp >

A szürke háttérrel jelölt karakterek billentyűit az **ALT** billentyűvel együtt megnyomva, vagy a **TAB** billentyűvel lépegetve és a megfelelő ponton az **ENTER**t leütve választhatók ki az adott alpontok. Azokban mozogni a kurzormozgató billentyűkkel lehet. A megjelenő ablakba írjuk be a fájl nevét és kiterjesztését, végül nyomjuk meg az **ENTER**t!

Egérrel: A **File** menü **Open ...** pontjának kiválasztásakor megjelenő ablakban álljunk rá az adott fájlra, majd nyomjuk meg egymás után kétszer az egér bal oldali gombját!

Program futtatása

A programot kétféle módon indíthatjuk, illetve futtathatjuk.

Menüből: Kiválasztjuk a **RUN** menü **START** alpontját.

Shift+F5: Ezt a billentyűkombinációt megnyomva a program elindul.

A kiválasztott program szerkesztett változata a **szerkesztőképernyőről** a futtatás alkalmával eltűnik, s helyette megjelenik egy új képernyőterület, amely a futtatás eredményét tartalmazza. Ezt **futóképernyőnek** hívjuk. A két képernyőt az **F4** billentyű lenyomásával váltogathatjuk.

A program futását a **Ctrl** és a **Pause** gombok együttes megnyomásával lehet megállítani. Ekkor a futóképernyő szerkesztőképernyőre vált. A program egy sora eltérő árnyalatú lesz, ez mutatja a programfuttatás megszakításának helyét. Az **F5** billentyű hatására innen folytatódik a program végrehajtása.

Segítségkérés

Ha a menüsor nem aktív, akkor a **SHIFT** és az **F1** billentyűk együttes megnyomásakor megjelenik egy angol nyelvű leírást tartalmazó ablak. Ebben a kurzormozgató, illetve a **TAB** billentyűkkel mozogva lehet – az **ENTER** lenyomásával – kiválasztani a nekünk szükséges segítő anyagot.

Ha a menüsor aktív – ezt valamelyik menüpont sötétebb árnyalata jelezi –, akkor az **F1** billentyű leütése az adott menüponthoz tartozó magyarázatokat hozza a képernyőre.

Kilépni mindkét esetben az **ESC** billentyűvel lehet. Ekkor a szerkesztőablakhoz térünk vissza.

A megjelenő magyarázatok között az egér segítségével is mozoghatunk. A keresett menüpontra kétszer megnyomva az egér bal oldali gombját, megjelenik a hozzá tartozó segítség.

Kiíratás a képernyőre

A szerkesztőképernyő két részből áll. A menüsor alatt új program szerkesztésekor az **Untitled** (*ántájtld*) felirat, program betöltésekor a program neve látszik. Itt villog a kurzor is.

A képernyő alsó részén az **Immediate** (*imídjöt*) felirat szerepel. Az alatta lévő területet **számlóterületnek** nevezzük. Ebben végezhetőek el a sürgősen szükséges számítások. A számlóterület méretét az egér segítségével meg tudjuk változtatni. Álljunk rá az Immediate szöveget tartalmazó vonalra, majd az egér bal oldali gombját lenyomva tartva mozgassuk a vonalat föl-le! Az egérgombot elengedve rögzíthetjük a számlóterület méretét.

A két terület között az **F6** billentyűvel tudunk váltani.

Állítsd be az Immediate terület méretét! Aktív állapotát a feliratot övező terület kifényesedése jelzi. A számlóterületre beírt, hibátlan parancsokat az **ENTER** billentyűvel hajthatjuk végre. A végrehajtás után a parancs eredménye a futóképernyőre kerül, ahonnan egy tetszőleges billentyű megnyomásával jutunk vissza a számlóterületre. A beírás helyét a képernyő jobb alsó sarkában látható **Cn** számláló mutatja.

A szükséges parancs formája:

PRINT változólista (a **PRINT** szó helyett lehet a **?** billentyűt is használni).

változólista: ide kell beírni a kiíratni szánt adatokat, illetve szöveget.

Ha a gép a beírtakat nem tudja értelmezni (hibás beírás), akkor a következő felirat jelenik meg:

Expected: statement	
<OK>	<Help>

Az **Ok** üzenetet elfogadva a hibát nekünk kell észrevenni s kijavítani. A **Help** segítséget ad a hiba kijavításában, a hiba okának megkeresésében.

Szöveg megjelenítése

PRINT "szöveg". Bármit írunk idézőjelek közé, azt a gép szövegnek értelmezi (tehát műveletet idézőjelek között nem tud elvégezni). Ha egy idézőjelet megnyitottunk, azt be is kell zárni.

Képernyőtörlés

Minden itt kiadott parancs előtt célszerű a futóképernyőt letörölni. A parancs: **CLS**

Feladatok

A pontozott részre írd a válaszokat!

1. Töröld le a képernyőt! Parancsa:
Írassd egy sorba, felváltva a kis- és nagybetűket! Hány darab betűt tudtál beírni?
2. Töröld le a képernyőt! Írassd egymás alá az ABC nagybetűit! Hány betű fért el?
3. Hány karaktert írhatunk az üres képernyőre?
4. Töröld a képernyőt, majd írassd egy sorba egymás után 10 darab A betűt! A parancs egyszeri végrehajtása után javítsd ki a parancsot úgy, hogy minden A betű elé egy B betűt írj!
5. Az alábbi parancsok kiadása után mi került a képernyőre?
PRINT 3+5
PRINT "3+5"
PRINT "3","5"
PRINT "3","5"
PRINT "3*5";3*5
PRINT "3-5",3-5
PRINT "3+5=";3+5
PRINT "Buda","Pest"
PRINT "Buda";;"Pest"
6. Mi a szerepe a **PRINT** parancsban
 – a pontosvesszőnek:
 – a vesszőnek:
 – az idézőjelnek:

Műveletek a BASIC programokban

A fejezet azokról a matematikai műveletekről szól, amelyeket a számítógép BASIC nyelven megért, azaz el is tud végezni.

Összeadás

jele: +

Két ilyen billentyű is van a gépen. Az egyik a numerikus billentyűzeten található, a másik az = jel fölött. A numerikus billentyűzet számait kényelmesebb használni, így a + jelet is innen választjuk gyakrabban.

Kivonás

jele: –

Ugyanígy jelölik a negatív számok előjelét is, csak azt a szám elé írjuk. A számítógép csak a negatív előjelet írja ki.

Szorzás

jele: *

Azért nem pontot írunk, mint a matematikában, mert az nem található a billentyűzeten.

Osztás

A számítógép háromféle osztást ismer.

hagyományos osztás

jele: /

Ez is eltér a matematikában megszokott jeltől, de gondoljatok a törtvonalra, ami szintén osztást jelöl, és az írógépeken csak így lehet leírni.

egészosztás 

jele: \

maradékos osztás 


jele: $a \text{ mod } b$


Ahol a az osztandót, b pedig az osztót jelöli. A **mod** szó (modulus szó rövidítése) maradékot jelent.

Hatványozás

jele: ^

A könyv első fejezeteiben már tanultunk a hatványozásról. Ha egy a számot n -szer vesszük szorozótényezőként $\underbrace{a \cdot a \cdot \dots \cdot a}_n$, akkor azt rövidebben

 Az elnevezés azt jelenti, hogy az osztás eredményeként azt az egész számot kapjuk meg, amelyet az osztóval megszorozva az osztandónál nem nagyobb egész számot kapunk.

 Ellentétben a matematikai szóhasználattal, itt csak az osztás maradékát kapjuk meg.

leírva az a n -edik hatványának nevezzük. A számítástechnikában az ilyen kifejezéseket így írjuk: a^n

Műveletek sorrendjének megváltoztatása

jele: ()

A matematikából ismert műveleti sorrendet itt is megváltoztathatjuk a zárójelekkel. A zárójelek közül csak az íves zárójeleket használjuk. Többszörös zárójelezésnél ahány zárójelet megnyitottunk, annyit be is kell zárni.

Feladatok

7. Az alábbi példákat számítsd ki fejben, majd a pontozott részre írd az eredményt! Számítógéppel ellenőrizd munkád eredményét!
- PRINT** $18-(2-(4-8))$
- PRINT** $18*9$
- PRINT** $-8*(-9)$
- PRINT** $72/9$
- PRINT** $73/8$
- PRINT** $46.4/3$
- PRINT** $19 \bmod 4$
- PRINT** $45 \setminus 7; 45 \bmod 7$
- PRINT** $"45:7=";45 \setminus 7;"$ **maradt** $";45 \bmod 7$
- PRINT** 2^8
8. Írd be az alábbi két parancsot!
- PRINT** $(17+9)*6$
- PRINT** $17+9*6$
- Miért más a két eredmény?
-
9. Milyen sorrendben végeznéd el a műveleteket?
- a) $(3+5)*12-24/6$
-
- b) $3^2+(5+8-11)*13-7$
-
10. Számítsd ki a számítógéppel a következő feladatok eredményeit!
- $1345+2945-175$
- $456-(234-45)$
- $645+45*107$
- $(56-34)*81$
- $305+2045:35$
- $6789:(43+57)+1000$
- $715:15+25*204$
- $2567:(75*4-23)^2$

11. Mennyi a 6325 és a 171 különbségének a 100-szorosa?

.....
Mennyi a különbsége 6325-nek és a 171 százszorosának?

12. Számítsd ki a számítógéppel az egészosztások eredményét! A maradékot is írasd a képernyőre!

58:17

1345:175

69257:1576

13. Számítsd ki a géppel a következő hatványozásokat!

3^3 12^4 $(-5)^3$ $(-7)^2$

.....

A műveleti jelek összefoglalása:

- = egyenlőségjel (csak idézőjelek között használva kerül kiírásra);
- + az összeadás jele;
- a kivonás jele;
- * a szorzás jele;
- / az osztás jele (ha nem egész az eredmény, tizedestört-alakban írja ki);
- \ az egészosztás jele;
- ^ a hatványozás jele.

Egyéb jelek:

- ? a PRINT utasítás rövidebb formája;
- () zárójelek (műveleti sorrend megváltoztatására);
- ; a PRINT után használva az eredményeket, szövegeket, egymás után írja (nem emel sort)
- , a PRINT után használva „zónázást” eredményez. Többszörözhető.
- " a PRINT utasításban az idézőjelek közé írt szöveg képernyőre kerül.

Programozás Quick BASIC-ben

A programozás, programok készítése elképzelhetetlen az adott nyelv utasítás-készletének biztos ismerete nélkül. Ha ezeket kiválóan megtanultuk, akkor sem biztos, hogy el tudunk készíteni egy programot, mert nem ismerjük az algoritmusát. Mi elsődlegesen az algoritmusok, eljárások készítésével foglalkozunk, s csak másodsorban a programok megírásával. Egy probléma megoldása során a megtanult algoritmusokat fogjuk alkalmazni, beépíteni a problémát megoldó algoritmusba. A program megírása az algoritmus ismeretében már szinte önmagától menni fog. Semmi esetre sem lesz nehezebb, mint lefordítani egy szövegrészt magyarról valamilyen idegen nyelvre. Először azonban néhány utasítással ismerkedjünk meg, hogy legyen valamekkora „szókincsünk” a fordításhoz!

A Quick BASIC nyelvben egymás alá írt utasítások sorozatával írjuk be a programot. A gép az egymást követő sorok alapján hajtja végre az utasítások sorozatát. Egy sorban több parancs is lehet, ha kettősponttal elválasztjuk őket!

A szerkesztőképernyőn dolgozunk! Minden további magyarázat nélkül gépeld be a következő – vastagon írt – sorokat!

COLOR 4,3,2 : PRINT "IBM AT" A számítógép nem hajtotta végre a beírt parancsot, mert a szerkesztőképernyőn dolgoztunk, s ezt már programnak értelmezi. A mi „programunk” csak egy sorból áll, de lehet „futtatni”.

Futtasd a programot (**SHIFT-F5**)! A képernyőn az IBM AT felirat minden színe (színes monitor esetén) megváltozott.

COLOR (kálör): a színek beállításának parancsa

COLOR a[,b[,c]]

- a** – a kurzor színe (16-ot hozzáadva villogást eredményez);
- b** – a háttér színe (erre a területre tudunk írni);
- c** – a keret színe (nem szükséges beírni).

A Függelékben megtalálod az egyes színek jelentését! A futtatás után bármely billentyű megnyomásával visszatérhetünk a szerkesztőképernyőre. Futtasd többször a programot, mindig más színeket megadva!

A szerkesztőképernyőn szúrd be a parancs elé:

CLS

és futtasd! Ekkor ugyanúgy végrehajtnak egymás után a parancsok. A számítógép letörli a képernyőt minden induláskor. Megváltoztatja a kiírások színét, majd kiírja a szöveget. Figyeld meg, hogy az alsó sorban a **Press any key to continue** felirat színe is megváltozott! Próbáld ki minél több színbeállítási lehetőséget!

Jó lenne, ha a képernyőre kerülnének a változások leírásai is! A képernyőre kiírni a **PRINT** utasítással lehet, a kiírandó szöveget idézőjelek közé kell tenni.

Új program indítása (File menü ⇒ New program almenü)

A menüpont kiválasztása után az alábbihoz hasonló ablak jelenik meg a képernyőn.

One or more file are not saved. Save Them Now?			
<i>(Uán ó mó fájl á not szévd. Szév dem náu?)</i>			
(Egy vagy több fájl nics elmentve. Mentsük ezeket?)			
<Yes>	<No>	<Cancel>	<Help>
<i>(Jesz)</i>	<i>(No)</i>	<i>(Kenszl)</i>	<i>(Help)</i>
(igen)	(nem)	(Visszavon)	(Segítség)

Nem mentjük el (Yes)! A képernyő tartalma törlődik.

Módosítsd tehát a programot!

CLS : COLOR 4,3,2 : PRINT "IBM AT"

PRINT "A kurzor piros"

PRINT "A hatter ciankek"

PRINT "A keret zold"

Futtasd le!

A programok mentése (File menü ⇒ Save as pontja)

A megírt programot BEALL1.BAS néven mentsd el!

(szév ez) másként ment

(fómet) formátum

(kvik bézik, fászt lód end szév)
gyors mentés és betöltés

(text, rídbl báj ádo program)
szöveg, más programmal is olvashatóan

A szürke háttérrel jelölt karaktereket az **ALT** billentyűvel együtt megnyomva választhatók ki az adott pontok. Azokban mozogni a kurzormozgató billentyűkkel lehet. A fájlnev sorában villog a kurzor. Írd ide a meghajtó nevét és az elérési utat, valamint a fájl nevét és kiterjesztését!

Például : C:\QBASIC\PELDA\BEALL1.BAS

Alkönyvtárat, meghajtót váltani az egérrel is lehet. Állj rá az adott alkönyvtár nevére, majd kattints kétszer! A **Dirs/Drives** felirat fölött az aktuális elérési út található. Ilyenkor a **File name** után csak a fájl nevét s kiterjesztését kell beírni.

A mentés formátumát az adott lehetőségek szerint válasszuk ki !

Az **ENTER**t ütve vagy az egérrel az **OK** felíratra rákattintva az adott nevű fájl a megadott alkönyvtárba kerül felvételre.

Módosítsd a programban a színeket úgy, hogy a kiírás is mindig helyes legyen! Ez a javítgatási eljárás nagyon nehézkes. Mennyivel egyszerűbb lenne, ha a gép kérné tőlünk az egyes beállítások kódjait, s a megadott értékeknek megfelelően végeznél el a színek beállításait.

Ahhoz, hogy ezt meg tudjuk oldani, elsőként meg kell ismerkedni a változókkal, illetve az értékadással.

Változók

Matematikából ismerős lehet a következő feladat:

$$q + 10 = \odot, \quad \text{ha } q = 5.$$

A \odot jel értékét keressük (nyitott mondat). A megoldást így írjuk:

$$q + 10 = 5 + 10 = 15.$$

A \odot jel helyett bármilyen jelet, betűt, szót írhattunk volna. Így az eredeti nyitott mondatnál egyenértékűek a következők is:

$$A + 10 = ? \quad \text{ha } A = 5$$

$$\text{BOCI} + 10 = ? \quad \text{ha } \text{BOCI} = 5$$

Minden esetben 15-öt kapunk eredményül.

Összefoglalva: a feladatokban a q és az A betű, valamint a **BOCI** szó mindig 5-öt jelentett, s így olvastuk ki:

Q egyenlő 5-tel.

A egyenlő 5-tel.

BOCI egyenlő 5-tel.

A számítástechnikában ugyancsak használjuk a jelöléseket, de bizonyos jeleket nem lehet alkalmazni.

A betűk, szavak összefoglaló neve: VÁLTOZÓ. A változók az IBM PC gépeken is mindig betűvel kezdődnek, de lehet utána már betű-szám kombinációkat is alkalmazni. Egy változó hossza maximum 40 karakter lehet (nem lehet benne szóköz).

Így például **változó nevek lehetnek:**

A ... Z *egy-egy betű*

A0 ; A1 ; A2 *betű-szám kombináció*

KZX34HUGO.

Nem lehet változóként használni a következő jelsorozatokat:

1BC *mert számmal kezdődik*

J K *mert szóköz van benne*

K,C *mert speciális jelet (itt a vessző) tartalmaz.*

Értékadás

Az a folyamat, amikor egy változó konkrét értéket kap: az értékadás. A konkrét formája:

VÁLTOZÓNÉV=ÉRTÉK

A változónév után közvetlenül az = jel áll, majd azt követően szünet nélkül az érték. A helyesen megadott változóknak értéket lehet adni.

C = 63 *kiolvasva C legyen 63;*

HUGO = -23.4 *kiolvasva HUGO legyen 23 egész 4 tized;*

D4 = C - HUGO *kiolvasva a D4 értéke legyen a C értékéből kivonva a HUGO értéke.*

Feladatok

14. Írd be az alábbi utasítást:

A=7:B=9

majd írasd ki a képernyőre az alábbiakat! A pontozott részre először a parancsot, majd a kiírt eredményt írd le!

A+B
A-B
A*B
A/B
A\B
A^B
A;B
A,B
(A+B)*(A\B)

15. Készítsd el azt a programot, amely az előző műveletek eredményeit kiírja!

Írjuk át a BEALL1.BAS programot úgy, hogy változókkal adjuk meg a beállításokhoz szükséges értékeket!

kurzor=4

hatter=8

keret=2

CLS : COLOR kurzor,hatter,keret : PRINT "IBM AT"

PRINT "A kurzor szine :";kurzor

PRINT "A hatter szine :";hatter

PRINT "A keret szine :";keret

Megtehetjük volna, hogy egy sorban, kettősponttal elválasztva adjuk meg az értékadásokat. A program utolsó három sorában hivatkozunk az első háromban megadott változókra. A pontosvesszőről már tanultuk, hogy „egymás után írás”-t jelent. Az idézőjelek közé írt *kurzor* szó kiírandó szöveget jelent, míg a pontosvessző utáni a változónév, melynek értéke jelen esetben 4.

Futtasd a programot (**SHIFT-F5**), és vizsgáld meg az eredményt a képernyőn! Mit tapasztalsz?

.....

Az egyéb lehetőségek kipróbálása most már sokkal könnyebben megy, mint az előbb. Elegendő felülírni a változók értékeit, s a számítógép a futtatás után bemutatja az eredményt a képernyőn. (Az értékek felülírásainál vigyázzunk a megbeszélt értékhatárok betartására! Nem lehet például negatív értékeket, illetve a táblázatban megadott legnagyobb számnál, a 15-nél nagyobbakat megadni a **b** és **c** értékeinél. A tanultak szerint próbáljuk ki a „villogtatást” is úgy, hogy az **a** értékéhez 16-ot hozzáadunk.)

Az adatok beszúrásánál az **INSERT** és a **DELETE** gombokat használd a felülíráshoz, illetve a törléshez! Próbáld ki minél több lehetőséget! Vedd fel a programot **BEALL2.BAS** néven!

Feladatok

16. Az alábbi példaprogramok alá – a pontozott részre – (azok begépelése nélkül) írd le, mit írnak ki a képernyőre! Ezután gépeled be egyenként a programokat, s ellenőrizd a megoldásaidat!

A=9	A=7	A=2	A=4
B=12	B=3	B=4	B=8
B=A	A=A+1	B=B/2	A=B
C=B-A	B=A	A=B-8	A=A+3
PRINT C	C=A+B	B=B+1	B=A
	PRINT C	C=A-B	C=A+B
		PRINT C	PRINT C
.....

Adatbevitel

INPUT ["szöveg";]változónév1[,változónév2, ...]

"szöveg"; Tájékoztató jellegű szöveg arról, hogy milyen adatot kérünk. Ugyanúgy megjelenik a képernyőn, mint a **PRINT** utasítás esetén. Elhagyható! Ha szöveget írtunk, a pontosvesszőt kötelező használni. Hatására a szöveg után nincs soremelés. Ugyanabban a sorban kéri a változó beírását.

változónév A tanult változónév-lehetőségek kerülhetnek ide. Egymás után több változó is megadható, de ebben az esetben vesszővel kell elválasztani őket egymástól.

Töltsd be a **BEALL2.BAS** programot! Módosítsd a programot az **INPUT** utasítás felhasználásával! A változók értékadásait írd felül!

INPUT "Kurzor szine=";kurzor

INPUT "Hatter szine=";hatter

INPUT "Keret szine=";keret

Ha minden sor végén megnyomtasd az **ENTER** billentyűt, a sorok felülírása megtörtént. Futtasd le! A képernyőn megjelenik az első sor szövege:

Kurzor szine=

majd utána egy villogó kurzor (a pontosvessző miatt). Írj be egy számot, amilyen színűre kívánod változtatni a kiírások szövegét!

Ekkor a következő sor szövege jelenik meg. A teendőd ugyanaz, mint az előbb. A harmadik beviteli sor végrehajtása után a képernyő (kiírás, háttér, keret) színei a beírtak szerint változnak meg.

Feladatok

17. Írj programot, amely két számot kér, majd kiírja
 - az összegüket;
 - a különbségüket;
 - a szorzatukat;
 - a maradékos osztással kapott eredményt a maradékkal együtt!
18. Írj programot, amely a téglalap két oldalának mérőszámát kéri, majd kiírja a területét és a kerületét!
19. Készíts olyan programot, amely az órák számát kéri, majd azt másodpercekben írja ki!

Mintaprogramok

1. Írj programot, amely a képernyőre írja a 6 és a 4 számokkal végzett alapműveleteket és azok eredményét!

A négy alpművelet: +, –, *, /. A program tehát a következő lesz:

```
CLS          letöröljük a képernyőt;
PRINT "6 + 4 = ";6+4  az idézőjelek között szereplő rész a képernyőn is
                    meg fog jelenni. Utána írjuk (a pontosvessző miatt
                    lesz a kiírás az idézőjeles rész után) a két szám
                    összeadásának eredményét.
```

Ha gyorsabban szeretnéd beírni az egymástól alig különböző sorokat, ismerned kell a szövegszerkesztés **blokk**-parancsait az **EDIT** menüpontból.

Blokk kijelölése: Ráállunk a kijelölésre szánt szöveg elejére, majd a **Shift** és a **kurzor jobbra** nyilat használjuk. (Az egér bal oldali gombját benyomva tartva és az egeret húzva a szövegen szintén kijelölhető a szükséges szövegrész.)

Blokk kivágása: A szövegrész kijelölése után megnyomjuk a **Shift** és a **Del** gombokat. (A szövegrész a memóriába kerül, s az adott helyről eltűnik.)

Blokk másolása: A szövegrész kijelölése után megnyomjuk a **Ctrl** és az **Insert** gombokat. (A szövegrész a memóriába kerül, s az adott helyről nem tűnik el.)

Blokk megjelenítése: A kívánt helyre állunk a kurzorral, megnyomjuk a **Shift** és az **Insert** gombokat.

A fentieket alkalmazva sokszorozd meg az elsőként leírt sort, majd módosítsd a sorokat a szükséges műveleti jelek szerint!

```
PRINT "6 - 4 =";6-4
PRINT "6 * 4 =";6*4
```

PRINT "6 : 4 =" ;6/4

Itt a matematikában használatos osztásjelet írjuk ki a képernyőre, s utána az osztás eredményét. Ez az eredmény természetesen tizedes tört lesz.

PRINT "6 : 4 =" ;6\4;" es maradt a " ;6 mod 4

Itt az osztást maradékosan végezzük a tanultak szerint. Először kiírjuk az egészosztás eredményét, majd utána a maradékot a mod segítségével.

Módosítsd a programot úgy, hogy a műveletek más-más színnel jelenjenek meg!

2. Írj programot, amely két tetszőlegesen megadott szám között elvégzi a négy alapműveletet!

A megoldás szinte ugyanaz, mint az előző feladatban. A változtatások a következők:

INPUT "Írj be egy számot";X

INPUT "Írj be egy masikat";Y *Beírhattuk volna egymás után is a két INPUTot :-tal elválasztva, vagy egy INPUTtal két adatot olvashattunk volna be.*

PRINT X;"+";Y;"=";X+Y

Először kiírjuk az X értékét, majd a + jelet (ezért kellett idézőjelbe tenni). Utána írjuk az Y értékét, majd az = jelet (idézőjelek). Ezt követően a két szám összeadásának eredményét jelenítjük meg.

A további kiírásoknál ugyanígy kell eljárni, csak a műveleti jeleket kell átírni. Használd az előző feladatnál leírt gyorsabb beírási módot (az előző sor szükséges javítása után **ENTER**)!

Ékezetes karakterek

A szövegek kiírásánál gyakran van szükség ezekre a karakterekre, amelyeket a számítógép billentyűzetén általában[■] nem találhatunk meg. Két lehetőségünk van.

I. Az **ALT** billentyű benyomása mellett a numerikus billentyűzetten kell beírni az adott betű kódszámát. A billentyűk elengedése után a kurzor helyén megjelenik az ékezetes betű.

á	160	é	130	í	161	ó	162	ö	148
Á	143	É	144					Ö	153
ô	147	ú	163	ü	129	Û	154	û	150

A hiányzó helyeken azért nem szerepeltetünk semmit, mert a leggyakrabban használatos operációs rendszerek és kódlap (437) ezeket a kódokat ismeri. A

[■] A számítógépek egyre gyakrabban kerülnek forgalomba magyar billentyűzettel. Aki ilyennel rendelkezik, annak ez a fejezet nem létező problémát tárgyal. Érdemes azonban megismerkedni vele, mert nem minden számítógép rendelkezik magyar billentyűzettel.

könyv végén megtalálható a Magyarországon előforduló kódkészletek teljes táblázata. A kódtáblák segítségével mindenki megtalálhatja a számítógépe beállításának megfelelő kódszámokat.

- II. A QBASIC indítása előtt egy magyar billentyűzetre átalakító programot (pl.: CWI.COM) indítunk. Ezek felhasználásával írd át a mintaprogramok kiírásra kerülő szövegét ékezetesre!

Szövegváltozók

Az eddig tanult változók mindegyike számértéket kaphatott csupán a program futtatása során. Ahhoz, hogy egy változó szövegváltozó legyen a \$ (dollár) jelet írjuk a változónév után. Az így felírt változónak csak szöveget adhatunk értéként.

változónév\$="érték"

Például:

A\$="Kukorica"

Horog\$="HAL"

cel\$="Elmegyek a boltba !"

Az idézőjelek közé írt karaktersorozatot jelenti tehát a változónév.

A pontozott részre írd oda, mi került a képernyőre a program indítása után!

A\$="Megkötöm lovamat"

B\$="szomorú fűzfához"

PRINT A\$,B\$

.....
PRINT A\$;B\$

.....
PRINT A\$+B\$

.....
Menj fel a kurzorral az utolsó sor + jelére, s javítsd át például – jelre! Futtasd a programot!

A gép a

Type mismatch (tájp miszmecs)

üzenettel leáll! Ez az üzenet azt jelenti, olyasmit kíséreltünk meg végrehajtani, ami a BASIC nyelvben nincs értelmezve.

Szövegváltozókkal (STRINGekkel) csak egyféle műveletet lehet végezni, az összeadást. Ez összefűzést jelent. A stringeket össze is lehet hasonlítani. (<; >; <=; >=; =)

Az utolsó sorban tehát az **A\$+B\$** művelet eredménye a két string egymás után írt szövege. Írj egy új sort úgy, hogy most a **B\$+A\$** string kerüljön kiírásra, s indítsd el a programot!

Feladatok

20. A programban legyen adott:

A\$="Alma" : B\$="Körte" : A=5 : B=12.

Írasd ki a változók segítségével a képernyőre!

a) **5 Alma 12 Körte;**

b) **12 kg Alma 5 kg Körte;**

c) **Almalé Körtelé.**

21. Írj programot, mely a BUDA, illetve a PEST szavakat a változók segítségével a következő módon jeleníti meg a képernyőn:

a) **BUDA PEST**

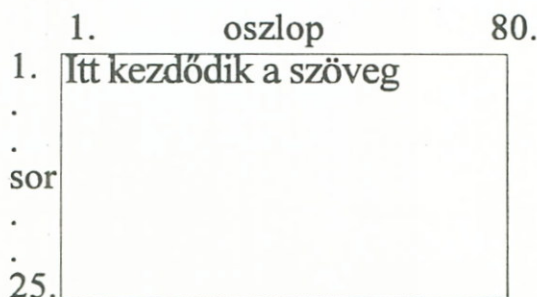
b) **PEST BUDA**

c) **BUDAPEST!**

A kurzor helyének beállítása

Több esetben szükség lesz arra, hogy a kiírások megadott helyre kerüljenek a képernyőn. Ehhez ismerni kell a képernyő méretét.

A képernyőn egymás mellett 80 karakter fér el. Úgy is mondhatjuk, hogy 80 oszlop van. Egymás alá 25 karaktert írhatunk. Ezeket sorszámozzuk, s a (sor, oszlop) koordinátákkal hivatkozunk rá.



A képernyő bal felső sarkának koordinátája (1,1); az ugyanabban a sorban jobbra mellette levő hely (1,2), majd (1,3) és így tovább.

A parancs formája:

LOCATE a,b

a – a sor értéke 1-től 25-ig,

b – az oszlop értéke 1-től 80-ig,

Az alábbi mintaprogramot gépeljétek be! Az indítás után az **X** betűk az adott koordinátán vannak.

CLS

LOCATE 5,11:PRINT "X Ez az 5;11 hely"

LOCATE 7,45:PRINT "X Ez pedig a 7;45"

LOCATE 13,64:PRINT "X A 13;64 helyre írtunk"

Készíts egy új sort, melyben a 8. oszlop 24-edik sorába íratod ki a nevedet!

.....

Futtatás után mit tapasztalsz?

.....

Feladatok

22. A programban legyen adott: `NEV$="saját neved"`! Készíts olyan programot, amely a képernyő azonos sorszámú sorába illetve oszlopába [pl.: (1,1), (2,2), (3,3),] írja ki a `NEV$` értékét!
Hányszor tudtad kiírni ?
-
23. Írj programot, amely egy szót, majd a képernyő egy sorának, illetve oszlopának számát kéri! Írasd ki az adott helyre a szót!

Mintafeladat

Írjunk programot, amely a képernyő közepén megjelenít az általunk megadott adatok alapján egy névtáblát!

A begépelésnél használjuk a tanult ékezetes karaktereket is! A programsorok után a parancsok magyarázata is megtalálható.

<code>CLS</code>	<i>Képernyőtörlés;</i>
<code>COLOR 4,8,2</code>	<i>Színek beállítása;</i>
<code>INPUT "Írd be a neved";N\$</code>	<i>A beírt nevet N\$-ban tároljuk, hiszen szöveget szöveges változóba kell tenni;</i>
<code>PRINT "A lakcímedet kérem!"</code>	<i>Ez a szöveg jelenik meg;</i>
<code>INPUT "Város: ";V\$</code>	<i>A városnév V\$-ba kerül;</i>
<code>INPUT "Utca és házsám: ";C\$</code>	<i>A cím pedig C\$-ba;</i>
<code>CLS</code>	<i>Letöröljük a képernyőt a kiírásokhoz;</i>
<code>COLOR 4,7</code>	<i>Beállítjuk az első kiírás színét;</i>
<code>LOCATE 12,30:PRINT N\$</code>	<i>A nevet a (12,30) pozícióba fogjuk írni;</i>
<code>COLOR 7,9</code>	<i>A város neve is más színű lesz;</i>
<code>LOCATE 13,30:PRINT V\$</code>	<i>Az előző név alá akarunk írni, ezért növeljük meg a sor értékét 1-gyel, hagyjuk változatlanul az oszlop értékét;</i>
<code>COLOR 10,5</code>	<i>Más színű lesz a cím is;</i>
<code>LOCATE 14,30:PRINT C\$</code>	<i>Ide kerül a cím;</i>
<code>COLOR 4,8</code>	<i>Visszaállítjuk az eredeti színeket.</i>

A programban csak nagyjából került a képernyő közepére a kész névtábla. A program indításakor még nem tudjuk, milyen hosszú neveket fogunk beírni, s a név hosszához kellene viszonyítani a kiírások kezdetét. (Ezt a problémát a stringfüggvényeknél oldjuk majd meg.)

Feladatok

24. Keretezd be a névtáblát csillagokkal úgy, hogy minden második csillag villogjon!

25. A képernyőre írasd ki az alábbi műveleteket s eredményeiket!

$$\frac{2}{3} + \frac{4}{5} =$$

$$\frac{6}{8} \cdot 12 =$$

$$\frac{7}{2} - \frac{8}{3} =$$

$$\frac{18}{27} : 9 =$$

26. Írj programot, amely egyenként bekéri legkedvesebb versed egy versszakának sorait, s utána más-más színnel kiírja a képernyő kiszámított helyére!

Gyakorlófeladatok

Az eddig tanultakat foglaljuk össze egy-egy feladat megoldásával! A feladatok hibátlan megoldása esetén haladj csak tovább! A válaszokat a pontozott részre írd!

27. Miért más az eredmény?

a) PRINT (34+7)*3

PRINT 34+7*3

.....

28. Számítsd ki a számítógéppel a következő műveleteket !

a) $9 - (6+5) * 4$

b) $(9-6) * 5 + 4$

c) $(9*6-5:4)*7$

d) $(9:6-5\backslash 4) - (14+7)\backslash 3$

29. Írasd ki a géppel az összes lehetséges sorrendben a következő szavakat!

ALOM; JEGYZÉK; TART

30. Egy téglalap két oldala 7,8 cm és 13 cm. Írjunk programot, amely kiszámítja a téglalap kerületét és területét! Írassuk ki a két oldal hosszát (mértékegységgel), valamint a „Terület=” és a „Kerület=” szavak után a számított értékeket!

31. Legyen $A = 9,7$ és $B = -3$. Írassuk ki a következő műveleteket, illetve azok eredményét!

$A+B$; $A-B$; $A*B$; A/B ; $A\backslash B$; A a B-ediken; A;B; A,B és az „A,,B

32. Legyen $A\$ = "ALMA"$ és $B\$ = "KÖRTE"$. Kérje a gép az almák és a körték számát! Írassuk ki a képernyőre más-más színnel egymás után, hány darab almánk, illetve körténk van!

33. Mi a hiba a következő programrészletekben? A pontozott részen javíts!

$X=2A+3*3,76$

$A=(A+B)*2+A+B^2$

$8=SZ+2*(Z+X)/(A+B)$

$A=7*(C+10):39$

$504A=5*3.14$




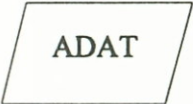



PROGRAMTERVEZÉS

Aki eddig figyelmesen olvasta a könyvet, már bizonyára képes arra, hogy néhány programot önállóan is elkészítsen. Hiszen milyen egyszerű az egész – néhány programsor, ezekhez néhány BASIC-parancs, és már kész is a program – gondolhatná bárki.

A dolog azonban nem ilyen egyszerű. A feladatok megoldása során bizonyos programozási szabályokat be kell tartani. A programozás egyik előfeltétele, hogy az elgondolásainkat a feladat megoldásával kapcsolatban valamilyen módon rögzíteni tudjuk. A problémát sok apró részletre kell bontani, s meg kell tervezni a részek összekapcsolásának rendszerét.

E munkát segíti, ha elképzeléseinket lerajzoljuk. A rajzok sokat elárulnak a majdani program működőképességéről, felhasználhatóságáról is. Mindezeket azért most mondjuk el, mert az „igazi programozás” gyakorlatilag most kezdődik.

A feladat megoldási algoritmusának rajzát **BLOKKDIAGRAM**nak, vagy más néven **FOLYAMATÁBRÁ**nak nevezzük. A blokkdiagram speciális rajzi jelek vonalakkal, irányított szakaszokkal összekötött rendszere. Egy programot lerajzolni, leírni más módon is lehet, de ezekről a későbbi tanulmányok során lesz szó. A leggyakrabban használatos jeleket foglalják össze a következő ábrák.

	A program kezdete (csak egy lehet)
	Általános műveletvégzés (pl.: értékadás)
	Adatbevitel
	Adatkivitel
	Elágazás kétfelé
	Kapcsolódási pont (hosszabb ábrák összekötéséhez)
	Program megállítása (több is lehet)

A Mellékletben megtalálod Petőfi Sándor: *Arany Lacinak című versének folyamatábráját*. Az ábra eredetije Dr. Szűcs Pál: *Számítógépes oktatási programok című könyvének 119. oldalán található*.

Mintafeladat

Számítsuk ki egy téglatest felszínét és térfogatát, ha ismerjük az oldalainak hosszát!

Vizsgáljuk meg a programtervezés lépéseit ezen a konkrét példán!

Bemenő adatok: a téglatest három éle, jelölje őket

A; B; C.

Kimenő adatok: Felszín (F), térfogat (V).

Képletek: $F=(A*B+A*C+B*C)*2$, valamint

$$V=A*B*C.$$

A felszín kiszámításánál nem használhatjuk a szokásos *A* betűt, mert az már az adatok bevitelénél szerepelt.

Először az adatokat kell bevinnünk az üres képernyőre:

CLS

INPUT "A=";A

INPUT "B=";B

INPUT "C=";C

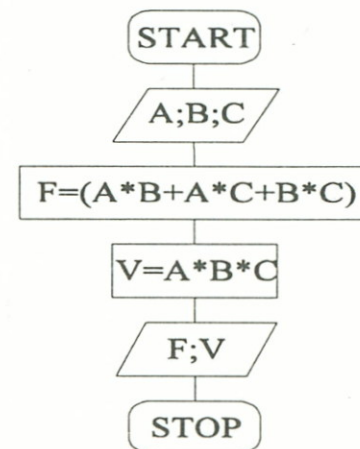
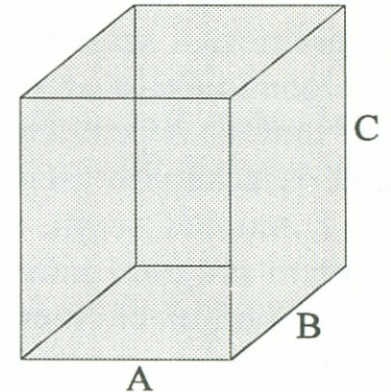
$F=(A*B+A*C+B*C)*2$

$V=A*B*C$

PRINT " Felszín = ";F

PRINT " Térfogat = ";V

END



Feladatok

Módosítsd a programot az alábbiak szerint!

1. A képernyő közepére írja ki az eredményeket a gép!
2. Az adatokat cm-ben kérje, majd a végeredmények után írja oda a mértékegységeket is a program!
3. A program más-más színnel kérje, illetve írja ki az eredményeket!

Változók értékadása

Az értékadásnál már említettük, hogy egy változó felveheti más változók értékeit is. Vizsgáljuk meg, mit ír ki a következő program!

A=8

az A értéke legyen 8;

B=10 *a B értéke legyen 10;*
A=B *az A vegye fel a B értékét. Eddig az A értéke 8 volt, most pedig a B értékét kapja meg, a 10-et;*
C=A+B *a C értéke legyen az A + B értéke, vagyis $10 + 10 = 20$ lesz.*
PRINT "C=";C *Végül a C értékének 20-at fog kiírni a program.*

Az ehhez hasonló értékadásoknál tehát nem a matematikában használatos egyenlőségjelet jelenti például az $A = 8$ kijelentés, hanem azt, hogy az A értéke legyen 8. Még furcsább lesz a következő példa:

A=8 *az A értéke legyen 8;*
B=10 *a B értéke legyen 10;*
A=A + 2 *A értéke legyen az előző értékénél 2-vel több, vagyis 10;*
B=B - 4 *B értéke legyen az előző értékénél 4-gyel kevesebb, azaz 6;*
C=A + B *összeadjuk az A és B értékeket ($C = 10 + 6 = 16$);*
PRINT "C=";C *itt pedig kiíratjuk. ($C = 16$)*

A mintaprogramból is látszik, hogy nem matematikai értelemben vett egyenlet az $A = A + 2$, a $B = B - 4$, illetve a $C = A + B$ kifejezés. A program futtatása után kiderül, hogy a gép elfogadja az ilyen kifejezéseket is, mint értékadást.

Feladatok

4. A programok begépelése és futtatása nélkül állapítsd meg, mit írna ki a gép a C értékének a program végén!

a) **A=B+8**

A=B:A=A+3

B=A

C=A+B

PRINT C

C=

b) **A=2:B=4**

B=B-2:A=B-8

B=B+1

C=A-B

PRINT C

C=

5. Géped be a programokat, s ellenőrizd a megoldásodat!

Változók értékeinek cseréje

Legyenek adva a következő értékek: **A=5** és **B=9**. Cseréljük fel a változók tartalmát! Nekünk persze könnyű, de ezt a gépnek is meg kellene tanítani!

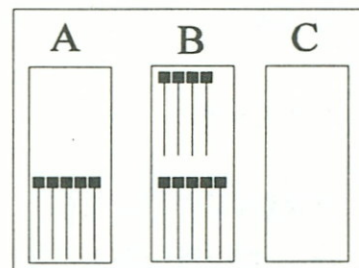
A=5:B=9

PRINT "A=";A,"B=";B

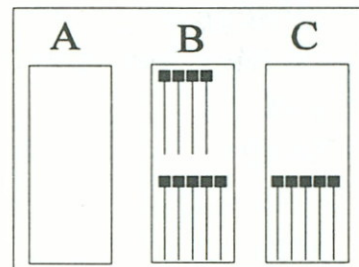
E kis program még csak az eredetileg adott számokat fogja kiírni. A képernyőn az 5 és a 9 számok ilyen sorrendben szerepelnek. Mi ezt fordítva szeretnénk: ha az A értékét íratjuk ki, akkor 9-et, a B értékénél pedig 5-öt kell kiírnia.

Az eljárás a következő:

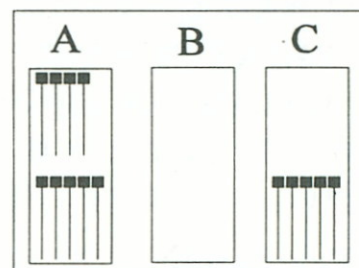
Képzeld el, hogy van 3 gyufásdobozunk. Az elsőben 5, a másodikban 9 szál gyufa van. A harmadikban nincs semmi. A gyufásdobozokat nevezzük el az előző sorrendben **A**, **B** és **C** betűkkel!



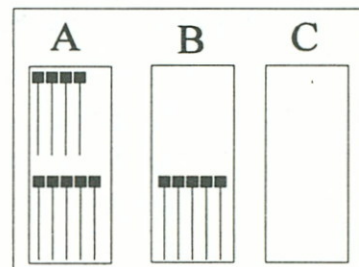
Az A-ból tegyük át az 5 gyufaszálat a C-be! Így ott 5 lesz, míg az A-ban egy sem.



Ezután a B-ből tegyük át a 9 gyufaszálat az A-ba! Így az A-ban lesz 9, a B-ben semmi, a C-ben pedig marad az 5.



Most már áttehetjük a B-be a C-ből az 5 gyufaszálat.



Az eljárással tehát felcseréltük az A és a B doboz tartalmát. Ugyanez programban:

```
C=A           a C értéke legyen az A értéke (C = 5);
A=B           az A értéke legyen a B értéke (A = 9);
B=C           a B értéke legyen a C értéke (B = 5);
PRINT "A=";A,"B=";B kiíratjuk a változók értékeit.
```

A teljes program indítása után a változók (A és B) értékei felcserélődnek.

Feladatok

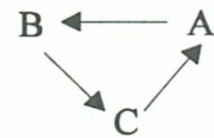
6. Adott A\$="GOLYÓ" és B\$="PUSKA" szövegváltozók. Írj programot, amely felcseréli a két változó értékét!

7. Mit csinál a program?

```
A$="NADRÁG":B$="KABÁT"
C$=B$:B$=A$:A$=C$:PRINT A$,B$
```

.....

8. Legyen $A = 5$; $B = 9$; $C = 11$. Cseréld fel a változók értékét az ábra szerint!



A gépi idő kiírása

Az AT számítógépekben beépített kvarcóra van. A gépi idő lekérdezésével a DOS-parancsoknál ismerkedtünk meg. Ennek BASIC nyelvű változata:

TIMES

Próbáld ki :

PRINT TIMES

Hatására a képernyőre kerül a pontos idő! A kiírt idő értékét módosítani úgy lehet, hogy szükség szerint a programban vagy a programon kívül beírjuk:

TIMES\$="hh:mm:ss"

– ahol a betűk jelentése a DOS-ban tanultakkal megegyező!

A dátum kiírása

Használatával a DOS-parancsoknál már megismerkedtünk. A BASIC nyelvben egy \$ jelet teszünk a DOS-parancs mögé.

Próbáljuk ki:

PRINT DATES

Az eredmény a képernyőn látszik.

Feltétel nélküli vezérlésátadás

Készítsünk programot, amely a képernyő közepére folyamatosan kiírja a pontos időt!

Ehhez szükséges egy olyan utasítás, amely a program működését folyamatosan a kiírásra irányítja. A parancs formája:

GOTO címke:

Ezek ismeretében a program a következő:

CLS

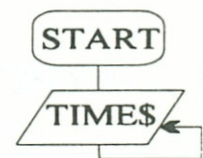
IDO:

LOCATE 12,35

PRINT TIMES

GOTO IDO

A képernyőn folyamatosan módosul a pontos idő kiírása. A programot megállítani a **Ctrl-Break** billentyűkkel lehet.



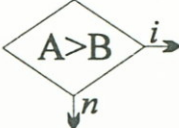
Feladatok

9. Módosítsd a programot úgy, hogy a mai dátum is a képernyőre kerüljön!
10. Rajzolj egy olyan kvarcórát a képernyőre, melyben a dátum és a pontos idő látszik! Az idő folyamatosan jelenjen meg a keretben!

17:56:38
1994.12.16

Feltételes utasítások

A feladatok megoldása során nagyon sokszor előfordul, hogy egy kérdésről véleményt kell mondania a gépnek. Ez legtöbbször azt jelenti, hogy egy kifejezésben összehasonlítást kell végezni. A számítógép a hasonlítás eredményéről csak annyit tud „mondani”, hogy igaz-e vagy sem. Gondoljunk például arra, hogy ha adott két szám, az **A** és a **B** (értékadás után), el kell döntetnünk a géppel, igaz-e a két szám közötti reláció. Az eldöntéshez egy új utasítás szükséges:

IF logikai kifejezés	<i>HA a logikai kifejezés igaz</i>	
THEN utasítás	<i>AKKOR milyen utasításunk van</i>	
[ELSE utasítás]	<i>[EGYÉBKÉNT mit tegyen]</i>	
END IF	<i>IF vége</i>	

A logikai kifejezés általában valamilyen relációt jelent:

<	kisebb	<=	kisebb egyenlő
>	nagyobb	>=	nagyobb egyenlő
=	egyenlő	<>	nem egyenlő

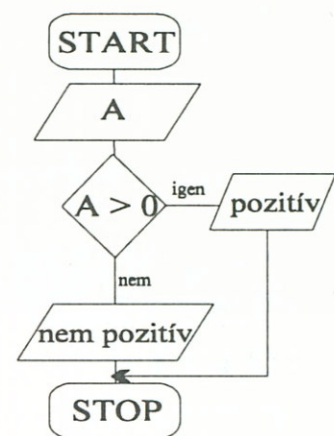
Ezekon kívül még más logikai műveletek is előfordulhatnak, de ezekről majd később lesz szó.

Mintaprogram

Írjunk programot, mely egy beadott számról eldönti pozitív-e vagy nulla! Egy szám akkor pozitív, ha nagyobb, mint nulla.

```
CLS
INPUT "Kérek egy számot";A

IF A>0 THEN
    PRINT "A szám pozitív"
ELSE
    PRINT "A szám nem pozitív"
END IF
```

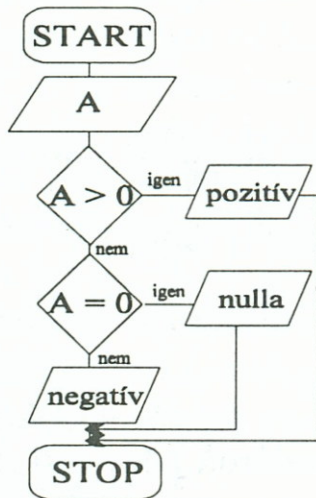


Felülről lefelé haladva tehát először beadunk egy **A** számot a gépnek (INPUT), majd egy feltételt vizsgálunk meg a géppel.

Ha a megadott **A** szám **nagyobb, mint 0**, azaz az állítás igaz, kiíratjuk vele azt, hogy „**POZITÍV**”, majd a vonalat követve leállítjuk a program futását. Ha az állítás hamis (nulla vagy negatív a szám), a megfelelő szöveget írjuk ki a géppel.

Módosítsuk a rajzot, illetve a programot úgy, hogy most a 0 beírását is vizsgálja meg a program, s jelezze a képernyőn!

Lerajzoltuk a folyamatábrát. Pótold a program hiányzó részeit!



CLS

INPUT "Kérek egy számot";A

IF ... THEN

PRINT "A szám pozitív !"

ELSE IF ... THEN

PRINT "A szám nulla!"

ELSE

.....

.....

.....

Futtasd a programot különböző értékekkel (nullával is)! Mit tapasztalsz?

.....

.....

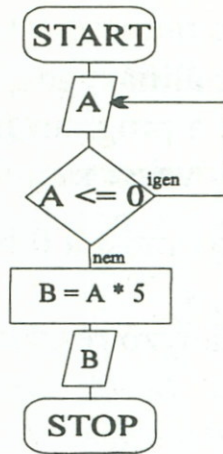
Mintafeladat

Írj programot, amely pozitív számok 5-szörösét írja ki! Ha a beadott szám negatív vagy nulla, kérjen új értéket a gép!

Ha a beadott érték kisebb vagy egyenlő nullával az nekünk a feladat szerint rossz adat, mert csak pozitívokat fogadhatunk el. Ezért ágazik vissza a vonal az adatbevitelhez. Ha a feltétel nem igaz (tehát pozitív), akkor számíthatjuk a szám 5-szörösét (ezt a B változóba tesszük), majd ezt írjuk ki.

Megjegyzés

A következő oldalom megtalálod a folyamatábrát és a programot. Az **END IF** kifejezés használata nem kötelező, ha az utasítás egyetlen sor csupán (pl.: IF A<=0 THEN GOTO ADATBE), de a könnyebb áttekinthetőség érdekében mindenütt kiírjuk.



Adatbe:
 CLS
 INPUT "A=";A
 IF A<=0 THEN
 GOTO Adatbe
 END IF
 B=A*5 : PRINT B

Feladatok

11. Vizsgáld meg, mit ír ki a program negatív, illetve pozitív értékeknél!

12. Próbáld meg legalább 10 karakter hosszú számokat is beírni!
 Mit tapasztalsz?

Több feltétel vizsgálata

Többször előfordul, hogy kizárólag bizonyos értékhatárok közé eső számot kívánunk a számítógéppel elfogadtatni. Ebben az esetben úgynevezett logikai műveletet kell a feltételek esetében megadnunk. Például: a -3 -nál nagyobb, de 12 -nél kisebb X számokat szeretnénk csak a gépbe engedni. Ebben az esetben úgy is fogalmazhatunk, ha $-3 \geq X$ vagy $X \geq 12$, akkor az nekünk rossz adat. A BASIC nyelvben is adottak ezek a lehetőségek.

Logikai műveletek: A kötőszavak BASIC (angol) és magyar megfelelői.

AND (*and*) ÉS
 OR (*or*) VAGY

Ezek segítségével lehet egyszerre több feltételt is megvizsgáltatni. Íme egy rövid mintaprogram a fentiek alkalmazására:

Adatbe:
 CLS:INPUT "X=";X
 IF X<= -3 OR X>=12 THEN
 GOTO Adatbe
 END IF
 B=X*5 : PRINT B

A lényeg a 3. sorban van. **HA** a $-3 \geq X$ vagy $X \geq 12$, **AKKOR** a szám nem felel meg a feltételeknek, ezért visszairányítjuk az adatbevitelre. Ha az állítás hamis (nekünk pontosan az ilyen számok kellene), egy egyszerű műveletvégzés után kiírjuk a művelet eredményét.

Feladatok

13. Próbáld ki a programot az AND szót alkalmazva is! Mit tapasztalsz?
.....

14. Töltsd be egyenként az eddig készített olyan programokat, melyekben INPUT utasítás szerepelt! Módosítsd ezeket úgy, hogy egy adott feltételnek feleljenek meg az elfogadott számok!

Az egészrészfüggvény

Készítsünk programot, amely csak egész számokat fogad bemenő adatként! A feladat megoldásához nélkülözhetetlen egy – a matematikában még nem használt – függvény, az egészrész ismerete.

Egy szám egészrészének a számnál nem nagyobb egész számok legnagyobbikát értjük. Másként fogalmazva: egy szám egészrésze egyenlő a számhoz a számegyenesen balról (lefelé) legközelebb eső egész számmal. Matematikai jele: $[x]$

A számítástechnikában ezt így írjuk: $INT(x)$ ahol az INTEGER (*intidzső*, egész) szó első három betűjét használjuk. A zárójelben szereplő x értéknek veszi az egészrészét.

Mintafeladat

Készítsünk programot, amely 1-től 100-ig kiírja a négyel osztható számokat!

CLS

Ismet:

```
IF J/4=INT(J/4) THEN
  PRINT J;
```

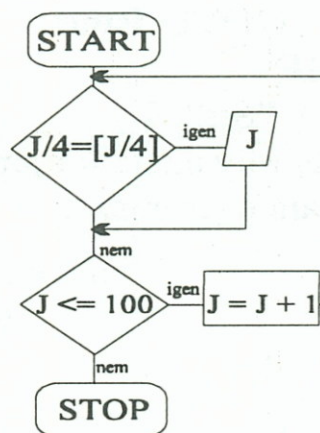
```
END IF
```

```
IF J<=100 THEN
```

```
  J=J+1
```

```
  GOTO Ismet
```

```
END IF
```

**Feladatok**

15. Gépeld be az alábbi parancsokat, majd írd a pontozott részre a kapott eredményt!

PRINT INT(2)

PRINT INT(-2)

PRINT INT(2.12)

PRINT INT(-2.12)

PRINT INT(0.001)

PRINT INT(-0.001)

PRINT INT(8.12*10)/10

16. Készíts folyamatábrát, majd programot, amely csak egész számokat fogad el bemenő adatként!

17. Egy dolgozat értékelésénél a következő ponthatárok vannak.

0 – 9 elégtelen;

10 – 15 elégséges;

16 – 22 közepes;

23 – 27 jó;

28 – 30 jeles.

A program kérje a tanuló nevét és elért pontszámát, majd értékelje a teljesítményét!

18. Írj programot, amely egy tanuló nevének és életkorának begépelése után eldönti, hogy hányadikos! (Feltételezzük, hogy nem ismételt osztályt és nem évvesztes!)

19. Az alábbi program begépelése nélkül állapítsd meg, mit ír ki a képernyőre!

CLS

J=100

Ismet:

IF J/2=J" THEN S=S+J

IF J>0 THEN

J=J-1

GOTO Ismet

END IF

PRINT "S=";S

20. Készíts folyamatábrát, majd írd meg a programot, amely megadja tetszőleges szám egészrészét!

SZÁMOK KERÉKÍTÉSE

Egész számok kerekítése

Kerekítsd a következő számokat!

	Tízesre	Százásra	Ezresre
45
745
4145
82 345

Készítsünk programot, amely egy beadott pozitív egész számot tízesre, százásra kerekít!

A feladat két részből áll. Először egy pozitív egész számot kell beadni a gépnek. Ezt a részt megoldottuk korábban, most nem tárgyaljuk újra. A másik rész a kerekítés. A matematikában tanult módszer – a jelenlegi ismeretek szerint – nem használható, mert nagyon nehézkes lenne az adott helyen a számjegyek vizsgálata. Ezért az egészrészfüggvényt fogjuk felhasználni. Legyen a szám például 46! Ezt tízesekre kerekítve 50-et kell kapnunk.

Nézzük az algoritmust!

Adjunk hozzá a 46-hoz 5-öt: $46+5=51!$

Osszuk el az 51-et 10-el: $51:10=5.1$ (5 egész 1 tized)!

Vegyük az 5.1 egészrészét: $\text{INT}(5.1)=5!$

Szorozzuk meg az 5-öt 10-zel: $5*10=50!$

Mindezt egyetlen utasítással elintézhethetjük a programban:

$$\text{INT}((X+5)/10)*10$$

Feladatok

1. Vizsgáljuk meg a kerekítés algoritmusát más számok esetén is:

X	X+5	(X+5)/10	$[(X+5)/10]$	$[(X+5)/10]*10$
72	72+5= ...	$\Rightarrow \dots/10= \dots$	$\Rightarrow \text{INT}(\dots)= \dots$	$\Rightarrow \dots*10= \dots$
16	16+5= ...	$\Rightarrow \dots/10= \dots$	$\Rightarrow \text{INT}(\dots)= \dots$	$\Rightarrow \dots*10= \dots$
699	699+5= ...	$\Rightarrow \dots/10= \dots$	$\Rightarrow \text{INT}(\dots)= \dots$	$\Rightarrow \dots*10= \dots$

Ha százásra akarunk kerekíteni, akkor ugyanezzel az eljárással dolgozunk, csak a számhoz 50-et adunk hozzá, és 100-zal osztunk, majd szorzunk.

2. Az előző leírás alapján írd le a százásra kerekítés utasítását!

.....

3. A leírtak szerint kerekítsd százásra!

3281

4. Az ezresre kerekítés utasítása:

.....

5. Kerekítsd ezek szerint ezresre a következő számot :

6219

6. Az alábbi programot bővítsd ki úgy, hogy ezresre is tudjon kerekíteni!

T\$="Tízre " :S\$="Századra " :E\$=.....:K\$="kerekítve"

Adatbe:

CLS

INPUT "A=";A

IF A<>INT(A) OR A<=0 THEN GOTO Adatbe

K1=INT((A+5)/10)*10

K2=INT((A+50)/100)*100

K3=.....

PRINT T\$+K\$;K1,S\$+K\$;K2 ,

Mi a feladata az IF A<>INT(A) OR A<=0 THEN GOTO Adatbe sornak a programban?

.....

Tizedestörtek kerekítése

Több esetben tizedestörtek kerekítését is el kell végezni tizedre, századra stb. Ilyenkor is ugyanezzel az eljárással dolgozunk.

Kerekítsük tizedre a 2,874-et!

$$2.874 + 0.05 = 2.924$$

$$2.924 * 10 = 29.24$$

$$\text{INT}(29.24) = 29$$

$$29 / 10 = 2.9$$

Kerekítsük századokra a 2,874-et!

$$2.874 + 0.005 = 2.879$$

$$2.879 * 100 = 287.9$$

$$\text{INT}(287.9) = 287$$

$$287 / 100 = 2.87$$

A tizedestörtek kerekítésekor így járunk el:

– tizedre kerekítésnél: $\text{INT}((X+0.05)*10)/10$

– századra kerekítésnél: $\text{INT}((X+0.005)*100)/100$

Összehasonlíttjuk a most leírt utasítást az előzővel.

- tízesre kerekítésnél: $\text{INT}((X+5)*10)/10$
- tizedre kerekítésnél: $\text{INT}((X+0.05)*10)/10$
- százásra kerekítésnél: $\text{INT}((X+50)*100)/100$
- századra kerekítésnél: $\text{INT}((X+0.005)*100)/100$
- ezresre kerekítésnél: $\text{INT}((X+500)*1000)/1000$
- ezredre kerekítésnél: $\text{INT}((X+0.0005)*1000)/1000$

Feladatok

7. Írj programot, mely két nevet kér, majd névsorban kiírja azokat! (A stringeket is ugyanúgy össze lehet hasonlítani, mint a számokat. Használd a relációkat, illetve a logikai műveleteket!)
8. Készíts programot, amely egy megadott racionális szám tizedre, századra, illetve ezredre kerekített értékét írja ki! (Használd az előző programot!)
9. Kérjen a gép két számot, majd írja ki a hányadosukat! (Vigyázz! Nullával nem lehet osztani! Ilyen számot nem fogadhat másodikként a gép!)
10. Három megadott egész számot állítson a gép növekvő sorrendbe!
11. Döntse el a gép két megadott számról, hogy azonos előjelűek-e!
12. Készíts programrészletet, amely nem engedi másoknak futtatni a programjaidat!
13. Írj programot, amely kér két természetes számot és megvizsgálja, hogy a kisebbik osztója-e a nagyobbiknak! (Vigyázz, a 0 is természetes szám!)
14. Készíts programot, amely kiírja egy pozitív egész szám osztóit!
15. Készíts programot, amely két pozitív egész szám közös osztóit írja ki!
16. Megadjuk egy tört számlálóját és nevezőjét. Készíts programot, amely egyszerűsíti a törtet, és kiírja az új számlálót, illetve nevezőt!
17. Adott egy téglalap területének és egyik oldalának mérőszáma. Készíts programot, amely kiszámítja a téglalap kerületét!

MENÜVEZÉRELT PROGRAMOK

A billentyűzet figyelése

A feltételes (IF ... THEN ... [ELSE ...]), illetve a feltétel nélküli (GOTO) vezérlésátadó utasítások segítségével szép programokat lehet készíteni. Egy billentyű megnyomásával más és más programrészekre irányíthatjuk a programot. Mindezekhez szükséges egy billentyűleütést figyelő eljárás:

X\$=INKEY\$

az INKEY\$ beolvas egy karaktert, és azt az X\$-ban tárolja. A \$ jel is mutatja, hogy itt szóveges változóról van szó. A beolvasás a billentyűzetről olyan gyorsan történik, hogy nem lehet pontosan kiszámítani azt az időpillanatot, amikor a billentyű megnyomását ellenőrzi a program, ezért itt „várakoztatni” kell a programot. Ezt a következők szerint tehetjük meg:

Varakozas:

X\$=INKEY\$

beolvas egy karaktert;

IF X\$="" THEN

ha ez semmi,

GOTO Varakozas

akkor vissza a VARAKOZAS sorra.

END IF

A két idézőjelet közvetlenül egymás után írtuk. Ezt követően, ha volt billentyűleütés, a feltétel hamis lesz, és a következő sorban folytatódik a program futása.

Elágazás (szelektálás) többfelé

A lenyomott billentyűtől függően más-más feladatot szeretnénk elvégeztetni a programmal!

CASE SELECT kifejezés

CASE kifejezés1 *mit csináljon?*

CASE kifejezés2 *mit csináljon?*

....

....

[CASE ELSE] *mit csináljon egyébként? nem szükséges beírni!*

END SELECT *Szelektálás vége*

Egy mintaprogramon vizsgáljuk meg a parancs használatát! Gépeld be, majd futtasd a következő programot különböző számokkal!

CLS:K\$="kerekítve"

LOCATE 1,9:PRINT"Számok kerekítését fogom elvégezni"

Adatbe:

LOCATE 3,9:INPUT"A kerekíteni kívánt számot kérem! ";X

IF X<=0 THEN GOTO Adatbe

```

LOCATE 5,9:PRINT"Válassz az alábbi betúk közül!"
LOCATE 6,13:PRINT"A. . . . . 100-asra"
LOCATE 7,13:PRINT"B. . . . . 10-esre"
LOCATE 8,13:PRINT"C. . . . . 1-esre"
LOCATE 9,13:PRINT"D. . . . . tizedre"
LOCATE 10,13:PRINT"E. . . . . századra"
LOCATE 11,13:PRINT"V. . . . . VÉGE"

```

Varakozas :

```
A$=INKEY$
```

```
IF A$="" THEN GOTO Varakozas
```

```
SELECT CASE A$
```

```
  CASE "a"
```

```
    LOCATE 6,35:PRINT K$; INT((X+50)/100)*100
```

```
    GOTO Varakozas
```

```
  CASE "b"
```

```
    LOCATE 7,35:PRINT K$;INT((X+5)/10)*10
```

```
    GOTO Varakozas
```

```
  CASE "c"
```

```
    LOCATE 8,35:PRINT K$;INT(X+0.5)
```

```
    GOTO Varakozas
```

```
  CASE "d"
```

```
    LOCATE 9,35:PRINT K$;INT((X+0.05)*10)/10
```

```
    GOTO Varakozas
```

```
  CASE "e"
```

```
    LOCATE 10,35:PRINT K$;INT((X+0.005)*100)/100
```

```
    GOTO Varakozas
```

```
  CASE "v"
```

```
    END
```

```
  CASE ELSE
```

```
    GOTO Varakozas
```

```
END SELECT
```

A kerekítendő szám megadása után egy úgynevezett „menü” jelenik meg a képernyőn. A kurzor nem látszik a várakozás sora miatt. Ha leütünk egy billentyűt, a gép megvizsgálja, hogy ez neki megfelel-e.

Ha megadott karakterek közül nyomunk meg egyet, a program vezérlése az adott műveletsort hajtja végre. Ezután a program vezérlése ismét áttevődik a várakozás billentyűbeolvasó eljárásra.

Ha a lenyomott billentyű nem szerepel a menüben, a CASE szerkezetben, akkor a GOTO VARAKOZAS sor segítségével visszairányítjuk a programot a billentyűleütést figyelő eljárásra.

A programból kilépni a v betű megnyomásával lehet .

Alaposan tanulmányozd át a program működését, majd vedd fel KERESKIT.BAS néven a lemezre!

Feladatok

1. Készítsd el a KERESKIT.BAS program folyamatábráját!
2. Módosítsd a programot úgy, hogy színes képernyőn, más-más színnel jelenjenek meg a menü sorainak kezdőbetűi, valamint a kezdőbetűvel megegyező színnel történjen a válasz kiírása is!
3. Készíts programot, mely a következőket tudja:
 - bekér két negatív egész számot, de a két szám nem lehet azonos;
 - menüből választhatóan összeadja, kivonja, összeszorozza, illetve elosztja a megadott számokat;
 - a nullával való osztást a gép nem végzi el, de jelezze, hogy az osztás nem végezhető el!

Karakter mozgatása a képernyőn

Mintafeladat

Készítsünk programot, amely egy karaktert mozgat a képernyőn az általunk kívánt irányban.

A feladat megoldásában az O betű mozgatását készítettük el.

Elsőként két string értékeit adjuk meg, melyek a karakter megjelenítését, illetve eltüntetését szolgálják. Az *s* illetve az *o* értékei a karika megjelenítésének helyét (sor,oszlop) adják meg.

Ezt követően – egy billentyűleütésre várakozó rész után – megvizsgáljuk, milyen billentyűt nyomtunk meg.

A *v* leütésére befejeződik a program futása.

Az *a*, illetve az *s* betű megnyomására a karikát balra, illetve jobbra mozgatjuk.

Ha a karika balra mozdul, akkor a képernyőn az oszlopszám értéke csökken, jobbra mozdulás esetén az oszlopszám értéke nő. Ugyanekkor a sor értéke változatlan marad.

```

a$ = "O":s$ = " "
s = 12: o = 40
CLS
golyo:
  LOCATE s, o: PRINT a$
varakozas:
  x$ = INKEYS
  IF x$ = "" THEN
    GOTO varakozas
SELECT CASE x$
CASE "v"
  END
CASE "s"
  LOCATE s, o
  PRINT s$
  o = o + 1
  IF o > 79 THEN
    o = 79
    PRINT CHR$(7)
  END IF
GOTO golyo

```

Minden elmozdulás esetén figyel-nünk kell a határértékekre, vagyis „nem mehet le” a képernyőről a karakter. A „határfalnak ütközést” hangjelzéssel is jelezzük: a CHR\$(7) hatására egy csen-gő szólal meg.

A k, illetve az l betűkkel pedig felfe-lé, illetve lefelé mozgathatjuk az alakza-tot.

Függőleges elmozdulás esetén a sor értéke változik meg, míg az oszlop érté-ke ugyanaz marad.

```

CASE "a"
  LOCATE s, o
  PRINT s$
  o = o - 1
  IF o < 1 THEN
    o = 1
    PRINT CHR$(7)
  END IF
  GOTO golyo
CASE "l"
  LOCATE s, o
  PRINT s$
  s = s + 1
  IF s > 22 THEN
    s = 22
    PRINT CHR$(7)
  END IF
  GOTO golyo
CASE "k"
  LOCATE s, o
  PRINT s$
  s = s - 1
  IF s < 1 THEN
    s = 1
    PRINT CHR$(7)
  END IF
  GOTO golyo
CASE ELSE
  GOTO varakozas
END SELECT

```

Feladat

4. Módosítsd a programot úgy, hogy a képernyő alsó sorában jelenjen meg a mozgató billentyű leírása! Ebbe a sorba ne lehessen a karaktert bevinni!
5. A program begépelése és kipróbálása után próbálkozz meg egy „pálcika-emberke” kirajzolásával, majd annak mozgásával! Vigyázz a képernyő méreteire!
6. Készíts programot, amely két különböző karakter mozgását teszi lehetővé! A mozgáshoz szükséges billentyűket a billentyűzet két széléről válassz, hogy ketten is játszassanak! Találj ki olyan játékszabályt, amit a gép ellenőriz, és lehetővé teszi az egyik játékos győzelmét a másik ellenében!

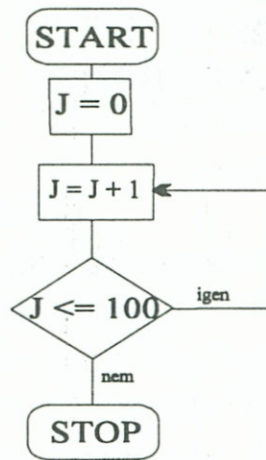
ISMÉTLÉSI SZERKEZETEK

Írassuk ki 1-től 100-ig a természetes számokat a képernyőre!

Az eddig tanultak alapján ezt PRINT utasításokkal megtehetnénk, de gondoljunk arra is, hogy például 50 ezerig ezt már értelmetlen lenne így megoldani. Az eddigiek során már tanult utasításokat fogjuk felhasználni.

Tudjuk, hogy a $J = J + 1$ utasítás azt jelenti, hogy a J értéke legyen a J előző értékénél 1-gyel nagyobb. Ezt egy olyan feltételsorral egészítjük ki, amely megvizsgálja J értékét, elérte-e például a feladatban adott 100-at. Hamis válasz esetén ismét megnövelnénk a J értékét és ki is írjuk a képernyőre.

Készítsük el a blokkdiagramot és írjuk meg a programot!



```

CLS
J=0
Leptet:
    J = J + 1
    IF J < 100 THEN
        PRINT J;
        GOTO Leptet
    END IF
  
```

Összefoglalva: a $J = 0$ a kezdőérték, a $J = J + 1$ esetében a +1 a növekedés mértéke, az **IF J < 100 THEN ...** sor pedig a tesztelő rész. Ezt az adott értékhatárig ismétlődő programot **VÉGES CIKLUS**nak nevezzük. J a ciklusváltozó, melynek kezdőértéke 0, végértéke 100.

A megírt program egy növekvő számsort írt ki a képernyőre. Módosítsuk úgy, hogy 100-tól 0-ig kettessel csökkenő számsorozatot írjon ki! A folyamatábra szinte ugyanaz, mint az előbb. A program mellé rajzold le!

```

CLS
I=100
Leptet :
    PRINT I;
    IF I>0 THEN
        I=I-2
        GOTO Leptet
    END IF
  
```

Feladatok

1. Készíts programot, amely teleír egy sort csupa X betűvel!
2. Írj programot, amely telerajzolja a képernyőt csillagokkal!
3. Írj olyan programot, amely telerajzolja csillagokkal a képernyőt úgy, hogy minden második után szünetet tesz, illetve minden harmadik más színű lesz!
4. Készíts olyan programot, amely 200-ig kiírja a 12-vel osztható természetes számokat!
5. Írj olyan programot, amely más-más színnel írja ki a 3-mal vagy 7-tel, valamint a 3-mal és 7-tel osztható természetes számokat 200-ig!
6. Legyen adott $A=72$ és $B=48$. Írj olyan programot, mely megkeresi az adott számok közös osztóit!
7. Kérjen a gép két természetes (2-nél nagyobb) számot, majd keresse meg a közös osztóikat!
8. Legyen adott $X=12$, illetve $Y=18$. Készíts olyan programot, amely megkeresi azt a legkisebb számot, amelyet maradék nélkül el lehet osztani az adott számokkal!
9. Készíts olyan programot, amely egy adott törtet egyszerűsít!
10. A program begépelése nélkül állapítsd meg, mi jelenik meg a képernyőn!

```
CLS : i = 1:
```

```
25 : j = 1
```

```
30 : LOCATE j, i: COLOR i, j: PRINT "o"
```

```
k = 1
```

```
leptet:
```

```
IF k < 200 THEN
```

```
    k = k + 1
```

```
    GOTO leptet
```

```
END IF
```

```
IF j < 15 THEN
```

```
    j = j + 1
```

```
    GOTO 30
```

```
END IF
```

```
IF i < 31 THEN
```

```
    i = i + 1
```

```
    GOTO 25
```

```
END IF
```

```
COLOR 4, 2
```

Gépeld be a programot, s ellenőrizd sejtésedet!

FOR ... NEXT ciklus

Az előző ciklusszervezésnél rövidebb, s egyszerűbb BASIC-utasítást ismertettünk, amelynek a feladata szintén végesszoros ismétlődő lépéssorozat szervezése. Az előző ciklus kapcsán már szükség volt a ciklus kezdő-, illetve végértékére, valamint a ciklusváltozóra.

Az új utasítás formája:

FOR ciklusváltozó=mettől TO meddig [STEP lépésköz]

utasítások

ez a rész fog ismétlődni

NEXT [ciklusváltozó]

Az utasítás felhasználási módját példán keresztül mutatjuk meg. Mit írnak ki a képernyőre az alábbi programok? Válaszodat a pontozott részre írd!

CLS

J=0

Leptet:

J=J+1

IF J<100 THEN

PRINT J;

GOTO Leptet

END IF

.....
.....

CLS

FOR J=1 TO 100

PRINT J;

NEXT J

.....
.....

Feladatok

11. Mi a különbség a két program kiírásai között?

.....

12. Mit tapasztalsz, ha a második program **FOR**ral kezdődő sorának végére beírjuk: **STEP 2**?

.....

13. Mit ír ki a képernyőre az alábbi két program?

CLS

I=100

Kiir:

PRINT I;

IF I>0 THEN

I=I-2

GOTO Kiir

End IF

.....

14. Módosítsd a második programot úgy, hogy 0-tól 210-ig hetesével írja ki a számokat! Módosítás:

.....

Mintafeladat

Készítsünk programot, mely 1-től 100-ig összeadja a számokat!

A BASIC-ben azok a változók, amik nem kapnak külön értéket, a program indításakor automatikusan nulla kezdőértéket vesznek fel. Az $S = S + K$ értékadás azt jelenti, hogy az S értéke legyen az S előző értékénél K -val több.

```
CLS
```

```
FOR K=0 TO 100
```

```
    S=S+K
```

```
NEXT K
```

```
PRINT "Az összeg =";S
```

Feladatok

15. Módosítsd az előző programot úgy, hogy a részösszegeket is írja ki a gép!
16. Számítsd ki a -1000 és az 1000 értékhatárok közé eső, egyesével növekvő számsorozat összegét!
17. Írj olyan programot, amely kéri két tört számlálóját és nevezőjét, majd kiírja a velük végzett összeadás műveletét és annak eredményét! A végeredményt egyszerűsítse is!
18. Készíts programot, mely menüből vezérelve lehetőséget ad a két törttel végezhető négy alapművelet választására! A gép végezze el a műveletet, majd a végeredményt egyszerűsítse!

DO (du) ... LOOP (lúp) ... UNTIL (ántil) ciklus

Az eddig tanult ciklusszervezéseknél előre adott kezdő-, illetve végértékek között történt az ismétlés végrehajtása. Nagyon sok esetben előfordul, hogy egy ismétlést addig kell végrehajtani, amíg valamilyen logikai feltétel igazgá nem válik. Az ehhez szükséges utasítás:

DO UNTIL logikai kifejezés

utasítások

ez a rész fog ismétlődni (ciklusmag)

LOOP

Az utasítás először mindig megvizsgálja a logikai kifejezés értékét, mielőtt végrehajtaná az utasításokat, ezért **előltesztelő ciklusnak** is nevezzük.

Egy hasonló parancs előbb egyszer végrehajtja az utasításokat, majd megvizsgálja az ismétlés további folytatásának lehetőségét.

DO

utasítások

ez a rész fog ismétlődni (ciklusmag)

LOOP UNTIL logikai kifejezés

Ezt a szerkezeti megoldást **hátultesztelő ciklusnak** nevezzük.

Feladat

19. Módosítsd a **FOR ... NEXT** ciklusban megadott mintafeladatok programját úgy, hogy most a **DO ... LOOP ... UNTIL** utasításokat használod!

20. Mit írnak ki a képernyőre az alábbi programok?

CLS	CLS
J=0	J=0
DO UNTIL J=100	DO
J=J+1:PRINT J;	J=J+1:PRINT J;
LOOP	LOOP UNTIL J=100
PRINT : PRINT"Vége=";J	PRINT : PRINT"Vége=";J

A *billentyűre várakozást* az eddigiekben egy címkére ugrással oldottuk meg. A tanult ciklusszervezés alapján mindezt egyszerűbben is megtehetjük.

```
DO
  X$=INKEY$
LOOP UNTIL X$=""
```

Mindezt programba beépítve a program csak a **SPACE** billentyű megnyomása után fut tovább.

DO (du) ... LOOP (lúp) ... WHILE (uájl) ciklus

Az előző ciklusszervezésnél addig hajtódott végre az utasítások sorozata, amíg az adott feltétel igazgá nem vált. Az alábbiak sokban hasonlítanak az előzőre. A különbség annyi, hogy az utasítások végrehajtása a logikai kifejezés hamissá válása esetén szakad meg. Az ehhez szükséges utasítás formálisan a következő:

```
DO WHILE logikai kifejezés
  utasítások ez a rész fog ismétlődni (ciklusmag)
LOOP
```

Az utasítás mindig megvizsgálja a logikai kifejezés értékét, mielőtt végrehajtná az utasításokat, ezért *előtesztelő ciklusnak* is nevezzük.

A következő utasítás előbb egyszer végrehajtja a parancsokat, majd megvizsgálja az ismétlés további folytatásának lehetőségét.

```
DO
  utasítások ez a rész fog ismétlődni (ciklusmag)
LOOP WHILE logikai kifejezés
```

Ezt a szerkezeti megoldást *hátulatesztelő ciklusnak* nevezzük.

Feladat

21. Mit írnak ki a képernyőre az alábbi programok?

CLS	CLS
J=0	J=0
DO WHILE J<100	DO
J=J+1:PRINT J;	J=J+1:PRINT J;
LOOP	LOOP WHILE J<100
PRINT : PRINT"Vége=";J	PRINT : PRINT"Vége=";J

22. Módosítsd a **DO ... LOOP** ciklusnál használt mintafeladatokat **DO ... WHILE** ciklusra úgy, hogy azonos eredményt adjanak a programok!

Pattogó golyó

Korábban készítettünk egy programot, amelyben karaktert mozgathattunk a képernyőn. Most készítsük el ennek önműködő változatát!

A képernyő bal felső sarkában jelenjen meg egy „golyó”, majd induljon el át-
lósan lefelé. A képernyő széléhez érve „pattanjon vissza”.

Nézzük a programot részleteiben!

```
CLS : x = 1 : y = 1 : xe = 1 : ye = 1
```

```
COLOR 4, 8, 2
```

```
a$="O": s$ = " "
```

```
golyo:
```

```
    LOCATE y, x
```

```
    PRINT a$
```

Az első „visszapattanásnak” akkor kell bekövetkeznie, amikor a golyó a képernyő alsó szélére ér. Ekkor az **X** (oszlop)változó értékeinek tovább kell növekedni, míg az **Y** (sor)változó értékeinek csökkenniük kell (ezután a golyó felfelé halad). Ezt úgy érjük el, hogy az **YE** (az **Y** változásának mértéke) nevű változó értékét (induláskor **YE** = 1) ellentettjére vesszük. A vizsgálatok végén tehát ezt az ellentett (**YE** = -1) értéket adjuk hozzá az **Y** értékéhez.

```
IF Y > 22 AND X > 1 AND X < 80 THEN YE = -YE
```

Ugyanez a teendőnk akkor is, amikor a képernyő felső széléről kell visszapatannia a golyónak. Ekkor ismét ellentettre változtatjuk az **YE** értékét (ismét 1 lesz).

```
IF Y <= 1 AND X > 1 AND X < 80 THEN YE = -YE
```

A képernyő jobb szélén az **XE** értékét vesszük ellentettjére, s ezt adva az **X** értékéhez az csökkenni fog.

```
IF X >= 80 AND Y > 1 AND Y < 22 THEN XE = -XE
```

A képernyő bal szélén ismét ugyanez a teendőnk.

```
IF X <= 1 AND Y > 1 AND Y < 22 THEN XE = -XE
```

A gép a fent leírtakat nagyon gyorsan elvégzi, így a golyó letörlése, illetve az új **X**, **Y** értékek kiszámítása előtt célszerű egy kissé lelassítani ezt a folyamatot. Ezt úgy érhetjük el, hogy a géppel elszámoltatunk egyesével 150-ig. Ha ennél nagyobb számot adnánk meg, akkor még tovább lassulna a golyó mozgása, kisebb értékekre pedig gyorsulna.

```
FOR J = 1 TO 150
```

```
NEXT J
```

```
LOCATE Y, X: PRINT S$
```

```
X = X + XE: Y = Y + YE
```

Az új értékek ismeretében ismét kiíratjuk a képernyőre az **O** betűt. A programnak ezt a műveletsort kell ismételnie, míg meg nem szakítjuk a futását.

GOTO golyó

A programot megállítani csak a **Ctrl-Break** billentyűkombinációval lehet.

Feladatok

23. Módosítsd a programot úgy, hogy a képernyő alsó sorában rajzoljon ki egy ütőt, majd azt két billentyűvel lehessen mozgatni. Alakítsd át a programot úgy, hogy a golyó csak akkor pattanjon vissza, ha az ütővel találkozik!
24. Számláld meg, hányszor találtad el a golyót! Ezt jelezze a gép a képernyőn is. Bizonyos számú hiba után a program írja ki: A játéknak vége!
25. Alakítsd át az előző programot úgy, hogy egy menüből lehessen választani a játék nehézségi fokát!
26. Módosítsd az előző programot úgy, hogy adott számú rossz ütés esetén (nem találsz el a golyót) írja ki a képernyőre „Pancser!”!
27. Készíts olyan programot, mely két ütőt mozgat a képernyő két szélén föl-le! Az ütőket két játékos tudja mozgatni úgy, hogy a golyó a két ütőről pattanhat csak vissza.
28. A program kérjen két egész számot, majd írja ki a két számmal végzett négy alapműveletet mindaddig, amíg a másodikként beadott szám nulla nem lesz!
29. Készíts programot, amelyben X értékei 1-től 1000-ig változnak egyesével, miközben a gép kiszámítja az $Y = 1/X$ értékét! A számítást addig folytatja, amíg az Y értéke nem kisebb 0,018-nál. A program írja ki a kapott Y értékeket!
30. Készíts programot, melyben az X értékei 1-től 1000-ig változnak egyesével, miközben a gép az $Y = X^2$ szabály szerint kiszámítja az Y értékét! Az eljárást addig folytassa, amíg Y értéke nem nagyobb 200 000-nél. Ekkor írja ki az X értékét!

VÉLETLENSZÁM ELŐÁLLÍTÁSA

A véletlen számokat mindenki ismeri. Ha egy kockával dobunk, nem tudjuk előre megmondani, milyen számot kapunk. A lottó esetében is ugyanilyen véletlen számok kerülnek „kihúzásra”. Ilyen számok előállítása számítógéppel is lehetséges. (A lottót is számítógéppel sorsolják.)

A BASIC-parancs formája:

RND(X) , ahol a **RANDOMIZE** (*rendömájjz*) szó 3 betűjét használjuk, s az **X** csak egész szám lehet.

Próbáld ki:

PRINT RND(0), RND(1), RND(10), RND(100)

Milyen számokat írtunk a képernyőre?

.....

Tudjuk, hogy a gépben beépített óra is van. Minden véletlen szám előállítása előtt vizsgáltsuk meg a géppel az időt, s ettől tegyük függővé a számok előállítását! A megfelelő parancs:

RANDOMIZE TIMER (*rendömájjz tájmö*) a véletlen számok előállítása az időtől is függ.

Feladatok

1. Próbáld meg 1 és 6 közötti egész számokat előállítani, most már programban!
2. A pontozott részre írva jellemezd a program lefuttatása után a képernyőn megjelenő számot!

CLS

RANDOMIZE TIMER

A = RND(1)

PRINT "A véletlen szám = "; A

.....

Szorozzuk meg 6-tal az előállított véletlen számot!

PRINT " annak 6-szorosa ="; A*6

.....

Vegyük az utolsó szám egészrészét!

PRINT "az előző egészrésze = "; INT(A*6)

.....

Többször egymás után futtasd a programot!

Módosítsd az utolsó sort a következők szerint:

PRINT INT(RND(1)*6+1)

Mit tapasztalsz többszöri futtatás után?

.....

Több véletlen szám előállításához töröld az előző programot, majd géped be a következőt!

```
CLS
RANDOMIZE TIMER
FOR I = 1 TO 20
  A = RND(1)
  PRINT INT(A*6 + 1);
NEXT I
```

Most már elég egyszer elindítani, a program egyszerre 20 véletlenszámot állít elő. Jegyezd meg a két legfontosabb lépést!

A véletlen szám előállítása előtt: **RANDOMIZE TIMER**
Az 1 ... 6 véletlen számok előállítása: **INT(RND(0)*6+1)**

Módosítsd az előző programot úgy, hogy 2 és 7 közé eső véletlenszámokat állítson elő!

A 2; 3; 4; 5; 6; 7 számokból álló tartomány összesen 6 számot tartalmaz, így ennek előállítási utasítása az előzőhöz nagyon hasonló. Írd meg a programot egyedül!

A tartomány hosszát így számítottuk ki:

$$(7-2)+1=6,$$

az előző esetben pedig

$$(6-1)+1=6.$$

Add meg a számítás általános szabályát!

A tartomány elnevezés helyett a matematikában is használatos INTERVALLUM szót is használhatjuk. Általánosítsuk az előző eljárást!

Tetszőleges intervallumba eső véletlen egész számok előállítása:

INT(RND(0)*H)+A vagy

INT(RND(0)*(B-A+1))+A

– ahol a betűk jelentése:

A az intervallum kezdete;

B az intervallum vége;

H=(B-A)+1 az intervallum hossza. Lehet **H=B-A+1** alakban is írni.

Feladatok

- Írass ki 100 darab véletlenszámot a 0 ... 100 intervallumban!
- Készíts olyan programot, amely 200 darab, az 1...6 intervallumba eső véletlenszámot állít elő! Számoldtasd meg a géppel, hányszor állított elő 6-os számot, s azt írasd ki! Mit tapasztalsz többszöri futtatás után?

.....

5. Állítsass elő a számítógéppel 500 darab olyan véletlen számot, amely a $-250 \dots 250$ intervallumba esik! Adja össze a gép az előállított számsorozatot, s csak a végeredményt írasd ki! Mit tapasztalsz többszöri futtatás után?

.....

5-ös lottó

A lottón az $1 \dots 90$ intervallumba eső számokat húznak ki. Így az intervallum kezdete 1, vége 90, hossza 90 lesz. Egymás után 5 lottószámot húznak ki, így ismétlésre is szükség van, ami nekünk ciklusszervezést jelent.

CLS	<i>letöröljük a képernyőt;</i>
RANDOMIZE TIMER	<i>az időtől tesszük függővé a véletlenszámok előállítását;</i>
FOR I=1 TO 5	<i>1-től 5-ig indítunk egy ciklust;</i>
X=INT(RND(0)*90+1)	<i>előállítjuk a véletlenszámot;</i>
PRINT X	<i>majd kiíratjuk a képernyőre;</i>
NEXT I	<i>vesszük a következő I-t.</i>

Mit tapasztalsz többszöri futtatás után?

.....

A program csak egyszer 5 számot ír ki, pedig mi esetleg több lottón is játszani szeretnénk. Több sorozat előállításához az 5 számot előállító ciklust egy újabb ismétlődő szerkezetbe tesszük. Ennek végértékét azonban mi adjuk meg. A program futtatása alatt a kívánt mennyiségű lottószámsor jelenik meg. A növekvő sorrendbe rendezéssel majd később fogunk megismerkedni, akkor próbáljuk meg kiküszöbölni az egy sorozatban megjelenő azonos számokat is.

CLS	
Adatbe:	
CLS	
INPUT "Hány lottót akarsz kitölteni? "; n	
IF n < 1 OR n <> INT(n) OR n > 22 THEN GOTO Adatbe	
FOR j = 1 TO n	<i>külső ciklus kezdete;</i>
FOR i = 1 TO 5	<i>belső ciklus kezdete;</i>
RANDOMIZE TIMER	
x = INT(RND(1)*90+1)	<i>ciklusmag;</i>
PRINT x;	
NEXT i	<i>belső ciklus vége;</i>
PRINT	
NEXT j	<i>külső ciklus vége.</i>

A külső ciklusban a **J** értéke először 1-et vesz fel. Ezt követően 5-ször egymás után végrehajtódik a belső ciklus, soremelés történik, majd a **NEXT J** hatására ismét a külső ciklus következik. Az **N** értékétől függ, hogy a belső ciklus hányszor hajtódik végre. Összesen tehát $N \cdot 5$ véletlen szám kerül a képernyőre.

A **NEXT I** a belső, míg a **NEXT J** a külső ciklushoz tartozik. Először mindig a belső ciklust kell lezárni, majd ezután a külsőt.

Feladatok

6. Készíts olyan programot, amely a 6-os lottó 45 számából állítja elő a kívánt mennyiségű sorozatokat!
7. Menüből vezérelve lehessen az 5-ös, illetve a 6-os lottó számait előállítani! Törekedj a szépen kivitelezett munkára!
8. Készíts olyan programot, amely bekéri az intervallum kezdetét és a végét, valamint az ismétlések számát, s az adott intervallumba eső, adott mennyiségű véletlen szám kerül a képernyőre!

A SZÁMÍTÓGÉP KARAKTERKÉSZLETE

Korábban már szó volt az ékezetes karakterek előállításáról. A számítógép olyan karakterek előállítására is képes, melyek a billentyűzeten nem találhatóak meg. Az ALT billentyű megnyomása mellett egy számkódot gépeltünk be, amelynek egy ékezetes karakter volt az eredménye. Más számkód begépelésére más-más karakter jelenik meg. A CHARACTER (*keriktő*) szó rövidítéséből (CHR) származik a következő BASIC-utasításunk:

A CHR\$(X) függvény előállítja az adott X ($0 \leq X \leq 255$) értékhez tartozó karaktert.

A HELP a Contents menü ASCII Character Codes alpontjában tartalmazza ezeket a karaktereket. A Függelékben szintén megtalálod az egyes számkódokhoz kapcsolódó karaktereket. A kód ismeretében tehát egy karaktert tudunk előállítani.

Ha egy karakter kódját szeretnénk megtudni, akkor a következő lehetőségünk van:

ASC("X\$") , ahol a X\$ helyére valamilyen karakter kerül.

A PRINT ASC("a") parancs hatására megjelenik az a betű kódja. Próbáld ki más karaktereket is!

Feladatok

1. Készíts olyan programot, mely kódszámok alapján megjeleníti az ABC nagy-, illetve kisbetűit!
2. A karaktertáblázat segítségével készíts egy keretezett névtáblát!
3. Készíts táblázatot, melyet az $Y=2*X - 3$ szabály alapján tölts ki a gép, ha az X értékei a -5-től 5-ig terjedő egész számok lesznek!
4. Készíts koordináta-rendszert a képernyőre a kódok felhasználásával!
5. A koordináta-rendszert megrajzoló programot felhasználva készíts olyan programot, amely kéri egy pont koordinátáit, majd az adott helyre kis kerestet tesz!

SZÖVEGKEZELŐ FÜGGVÉNYEK

Tanultunk már a PC karakterkészletéről. Az egyes karaktereket a **CHR\$(x)** függvénnyel jeleníthettük meg. Arra azonban eddig nem volt lehetőségünk, hogy egy szövegrészben megvizsgálhassuk a karaktereket. Legyen adott a következő string:

A\$="számítógép"

Milyen szavakat képezhetünk ebből a szóból annak feldarabolásával?

.....

.....

A felsorolt szavak előállításához új ismeretek szükségesek.

A string bal oldali *n* darab karakterének vizsgálata

A „szám” szó az adott string bal oldali négy karaktere. A „bal” szó angolul **LEFT** (*left*). A BASIC nyelvben az

X\$=LEFT\$(A\$,n)

függvény segítségével az **A\$** string bal oldali **n** darab karakterét kapjuk. A programunk második sora legyen:

BAL1\$=LEFT\$(A\$,4):PRINT BAL1\$!

Indítsd el a programot! A „számít” szó esetében az első 6 karaktert kell megkapnunk. Írd be a szükséges parancsot a harmadik sorba! Indítás után meggyőződhetünk a beírtak helyességéről.

Az **A\$** hossza 10 karakter. Próbáljuk kiírni a string bal oldali 11 karakterét! Mit tapasztalsz?

.....

Mi lesz az eredmény, ha negatív darabszámú (pl.: -3) karaktert szeretnénk megkapni?

.....

Írnod ki a „számító” szövegrészt is a képernyőre!

A string jobb oldali *n* darab karakterének vizsgálata

A „gép” szó kiírása az előbbieket szerint már nem megy. Ez a szöveg végén található három karakter. Úgy is mondhatnánk, hogy a jobb oldali 3 karaktert szeretnénk kiírni. A „jobb” szó angolul **RIGHT** (*rájt*), ezért a

Y\$=RIGHT\$(A\$,n)

függvény hatására az adott string jobb oldali **n** darab karakterét kapjuk meg.

JOBBS\$=RIGHT\$(A\$,3):PRINT JOBBS

Írnod ki a jobb oldali 12 karaktert!

JOBBS1\$=.....

Mit tapasztalsz?

.....

Mi a hatása a parancsnak negatív darabszám esetén?

Szövegrészek vizsgálata

Az „ámít” szövegrész már a string belsejében van, a második karaktertől 4 karakter hosszan. Az erre használható függvény:

Z\$=MID\$(A\$,n,m),

ahol a Z\$ értéke az A\$-ből az n-edik karaktertől m darab karakter.

A MID kifejezés az angol MIDDLE (*midl*), szóból származik, jelentése középső.

KOZEPS=MID\$(A\$,2,4):PRINT KOZEPS

Írasd ki az „ámító” majd a „gép” szövegrészeket a képernyőre!

K1\$=.....

K2\$=.....

Mit adnak eredményül a következő parancsok?

PRINT MID\$(A\$,6,15)

PRINT MID\$(A\$,6,-2)

PRINT MID\$(A\$,-3,6)

A string hossza

Az előzőekben adott stringben egyszerűen megszámozhattuk a karakterek számát. Hosszabb szöveg esetén erre nem mindig van lehetőségünk, ezért a string hosszának meghatározására a

H=LEN(A\$)

függvényt adja segítségül a BASIC nyelv. A H értéke lesz a string hossza.

Írasd ki a képernyőre az A\$ hosszának értékét !

H=.....

Mintafeladat

Készítsünk programot, mely egy adott szöveget a képernyőn karakterenként egymás alá ír!

A megoldásnál szükségünk van a string hosszára, és a MID\$ függvényt fogjuk használni. Az alábbiakban a programot mutatjuk meg.

CLS

INPUT "Kérem a szöveget";X\$

H=LEN(X\$)

FOR I=1 TO H

Y\$=MID\$(X\$,I,1)

LOCATE 5,I : PRINT Y\$

NEXT I

Itt határozzuk meg a string hosszát.

Indítunk 1-től H-ig egy ciklust.

Az X\$ string I-edik karakterétől egy karakter lesz az Y\$ értéke.

Az I-edik oszlop 5. sorába kiíratjuk az előállított karaktert.

Vesszük a következő karaktert.

Módosítsd a programot úgy, hogy alulról fölfelé írja ki az adott szöveget!

Alakítsd át a programot úgy, hogy a képernyőn átlóban írja ki a megadott szöveget!

Ha „írógépszerűen” szeretnénk kiírni a szöveget, akkor célszerű az egyes betűk megjelenése között egy kis szünetet tartani. Ezt a tanultak szerint egy számláló ciklussal megvalósíthatjuk:

```
FOR Q=1 TO 1500 : NEXT Q
```

Számváltozó átalakítása szövegváltozóvá

Több esetben szükségünk lesz erre a műveletre. A függvény formája:

```
S$=STR$(szám)
```

– hatására az S\$-ba kerül a számból képzett string.

Próbáld ki az alábbi programot, majd írd le a tapasztaltakat!

```
CLS
```

```
A=15 : B=-8
```

```
X$=STR$(A) : Y$=STR$(B)
```

```
PRINT A+B
```

```
PRINT X$+Y$
```

```
PRINT A*2+B/3
```

```
PRINT X$*2+Y$
```

String számmá alakítása

Ha a stringben számot tároltunk, akkor lehetőségünk van a szövegváltozóból számváltozó előállítására.

```
S=VAL(S$)
```

– hatására S-be kerül az S\$-ből előállított szám. Az utasítás balról jobbra keresi a stringben a számokat. Ha más karaktert talál, akkor befejezi a keresést.

Mit írnak a képernyőre a következő parancsok ?

```
PRINT VAL("AB24")
```

```
PRINT VAL("23C6")
```

```
PRINT VAL("-1.34")
```

Mely esetekben adott számot eredményül a VAL(X\$) függvény?

Feladatok

1. Készíts programot, mely egy adott szövegről megállapítja, hogy hány karaktert tartalmaz! (A szóközöket itt nem tekintjük karakternek.)
2. Egy tetszőlegesen megadott szövegben (maximum 10 betű) keressük meg az ABC szerinti első, illetve utolsó betűt! (Az ékezet nélküli betűk a karaktertáblázatban abc-sorrendben vannak. Így az adott betű CHR\$ értékei döntik el a sorrendet.)
3. Készíts programot, amely a gépi dátumból kiválasztja a hónap sorszámát, majd ez alapján a dátumban kiírja a hónap nevét!

HANG ELŐÁLLÍTÁSA

BEEP (*bíp*)

Hatására a hanggenerátor megszólaltat és 1/4 másodpercig kitart egy 800 hertzes hangot. Helyettesíthető a **PRINT CHR\$(7)** utasítással is. Mi a hatása a következő programnak?

```
BEEP : BEEP
FOR J=0 TO 200:NEXT J
PRINT CHR$(7)
FOR J=0 TO 200:NEXT J
BEEP
```

SOUND (*száund*)

SOUND frekvencia, időtartam

frekvencia 0-tól 32767 Hertzig

időtartam alapegysége 1/1092 perc. Egy másodperc kb. 18,2 egység. Értéke 0–65535 tartományban lehet.

A **SOUND 800,4** megfelel a **BEEP**nek. Az időtartam értéke nem haladhatja meg a 65535 értéket. A 0 érték megszakítja az éppen kitartott hangot, minden más esetben befejezi az előző utasítást, mielőtt feldolgozná a következőt. A hangjegyek frekvenciatáblázatából kiolvashatók a szükséges értékek. A táblázatban és a számítógépes zenében B-vel jelöljük a H hangot.

Frekvenciatáblázat

Hangjegy	1.oktáv	2.oktáv	3.oktáv	4.oktáv
C	130.81	261.63	523.15	1046.50
D	146.83	293.66	587.33	1174.70
E	164.81	329.63	659.26	1318.50
F	174.61	349.23	698.46	1396.90
G	196.00	392.00	789.99	1568.00
A	220.00	440.00	880.00	1760.00
B	246.94	493.88	987.77	1985.50

Az időtartamok számértéke:

egész	fél	negyed	nyolcad	tizenhatod
16	8	4	2	1

A pontozott fél értéke 12, a pontozott negyedé 6.

Mintafeladatok

1. Készítsünk programot, amely bemutatja a számítógép zenei képességeit!

Ciklusban szólaltatjuk meg a hangokat. A végértéket most csak 2000-re vesszük. Az időtartamot 1-re állítjuk.

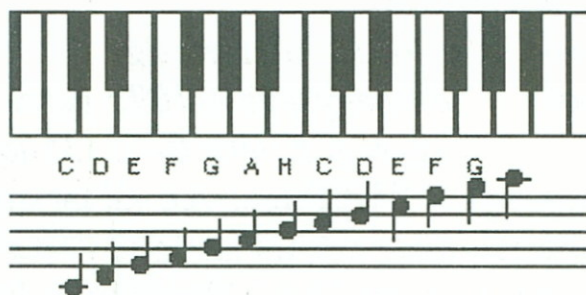
```
FOR J=37 TO 2000 STEP 5
  SOUND J,1
NEXT J
```



2. Készítsünk programot, amely eljuttassa a „Boci, boci tarka” című gyermekdalt!

Kottázzuk le a dalt, majd írjuk oda, mennyi ideig szóljon az adott hang!

A zongorabillentyűk ABC-s jelei



és a nekik megfelelő hangok a kottában.

Szöveg:	Bo	ci	bo	ci	tar	ka	se	fü	le	se	far	ka
Hang:	C	E	C	E	G	G	C	E	C	E	G	G
Időtartam:	8	8	8	8	16	16	8	8	8	8	16	16

Szöveg:	O	da	me	gyünk	lak	ni	a	hol	te	jet	kap	ni
Hang:	C'	B	A	G	F	A	G	F	E	D	C	C
Időtartam:	8	8	8	8	16	16	8	8	8	8	16	16

Ezek után következhet a program! Egyenként szólaltatjuk meg az adott hangokat. A frekvenciatáblázatból a 2. oktávot választjuk. Két azonos hang megszólaltatása egymás után a programban egy kis szünet közbeiktatásával történik, amit a **SOUND 0,1** utasítással érhetünk el. A C' hang nem más, mint a 3. oktáv első hangja. A programsorok után odaírtuk az adott sorban egymás után megszólaló hangok jelét.

SOUND 261.63,8:SOUND 329.63,8

C; E

SOUND 261.63,8:SOUND 329.63,8

C; E

SOUND 392,16:SOUND 0,1:SOUND 392,16

G; szünet; G

SOUND 261.63,8:SOUND 329.63,8

C; E

SOUND 261.63,8:SOUND 329.63,8

C; E

SOUND 392,16:SOUND 0,1:SOUND 392,16	<i>G; szünet; G</i>
SOUND 523.15,8:SOUND 493,8	<i>C'; B</i>
SOUND 440,8:SOUND 392,8	<i>A; G</i>
SOUND 349.23,16:SOUND 440,16	<i>A; F</i>
SOUND 392,8:SOUND 349.23,8	<i>G; F</i>
SOUND 329.63,8:SOUND 293.66,8	<i>E; D</i>
SOUND 261.63,16:SOUND 0,1:SOUND 261.63,16	<i>C; szünet; C</i>

Félhangok előállítása

A zene programozásánál gyakran előfordul, hogy nemcsak egész, hanem félhangokat is elő kell állítani. Megközelítőleg pontos eredményt kapunk, ha a két egész hang frekvenciaértékének átlagát vesszük. Pl.: C# vagy *cisz* értéke a 2. oktávban: $(261.63+293.66)/2$. A félhangokat természetesen csak az adott oktávban zeneileg lehetséges módon állíthatjuk elő. Például nem lehet félhang az E és az F hang között.

Feladatok

1. Szólaltasd meg egymás után a négy oktáv A hangját 2-2 másodpercig!
2. Készíts programot, amely negyedhangonként szólaltatja meg a C dúr skálát (CEGC') a szóköz billentyű megnyomásáig folyamatosan!
3. Szólaltasd meg egymás után a C-dúr skálát mind a négy oktávon egy billentyű megnyomásáig!
4. Válassz ki egy számodra kedves dallamot, majd szólaltasd meg számítógépen a szükséges program megírásával!
5. Készíts programot, amely csak pozitív egész számokat fogad el bemenő adatként! Rossz szám megadásakor a gép „berregő” hangon jelezze a hibás adatbevitelt!
6. Töltsd be a pattogó golyó programodat és módosítsd úgy, hogy minden „fálnál” más-más hanggal jelezze a visszapattanást!

ADATOK TÁROLÁSA PROGRAMBAN

A program begépelésekor mennyivel egyszerűbb lenne, ha a rengeteg **SOUND** utasítást nem kellene ennyiszor begépelni. A frekvencia, illetve az időtartam értékeit a program tárolná, s indítás után csak „elolvasná” a gép a tárolt adatokat. Erre a **DATA** sorokban van lehetőség.

DATA adat,adat,adat,...

A számítógép a **DATA** sorokat mindaddig figyelmen kívül hagyja, míg egy az adatok elolvasására felszólító utasítással nem találkozik. Így ezek a sorok a program bármely részén szerepelhetnek, mert futtatás közben ezeket átugorja. A **DATA** utasítás után az adatokat mindig vesszővel kell elválasztani. Tároljuk tehát **DATA**-sorokban egymás után a megszólaltatni kívánt hangok frekvenciáját, illetve időtartamát!

DATA 261.63,8,329.63,8,261.63,8,329.63,8,392,16,0,1, 392,16

DATA 261.63,8,329.63,8,261.63,8,329.63,8,392,16,0,1,392,16

DATA 523.15,8,493,8,440,8,392,8,349.23,16,440,16

DATA 392,8,349.23,8,329.63,8,293.66,8,261.63,16,0,1,261.63,16

A teljes sor 27 hangból, illetve a hozzájuk tartozó időtartamokból áll.

Adatok kiolvasása

Formája :

READ változó[,változó[,...]]

Az előzőekben beírt **DATA**-sorok 27 adatpárt tartalmaznak. Ezek egymás utáni elolvasását ciklussal oldjuk meg.

FOR I=1 TO 27

READ H,T

A H változóba kerül az adatpár első értéke, vagyis a frekvencia, míg a T változóba az időtartam.

SOUND H,T

A kiolvasott értékek segítségével előállítjuk a hangot.

NEXT I

Adatmutató visszaállítása

A **READ ... DATA** utasítások működése közben egy speciális mutató ellenőrzi, hogy mindig a következő adatot olvassuk ki. Ha egy adatsort újra ki akarunk olvasni, akkor a mutatót az adott sor elejére kell visszaállítani. Az adott zenét újra meghallgatni csak a program újraindításával lehet. Ha ugyanezt programból szeretnénk elérni, akkor a **DATA** mutatóját is vissza kell állítani.

RESTORE

hatására az első **DATA**-sorra állítódik vissza a mutató;

RESTORE címke hatására az adott címkéjű DATA-sorra állítjuk a mutatót.

Ezek után próbáljuk egymás után 5-ször meghallgatni a zenét! Ezt egy újabb ciklus létrehozásával tehetjük meg. Írd a program első sora elé:

FOR J=1 TO 5

Az előző program **NEXT**-je után írd be:

RESTORE *mielőtt újraolvasnánk az adatokat, a mutatót a legelső DATA-sorra állítjuk.*

NEXT J

Módosítsd a programot úgy, hogy gyorsabban játssza le a zenét! Ezt úgy teheted meg, hogy a hang előállításakor a kiolvasott időtartam értékét a felére veszed.

SOUND H,T/2

Feladatok

1. Készíts programot, amely az első 20 természetes számot DATA-sorokban tárolja, majd olvasd be ebből az első 10 számot és írasd ki azokat, illetve azok összegét!
2. Készíts programot, amely bármely hónapról megmondja hány napos!
3. Készíts olyan programot, amely bármely pénzösszegről megmondja, milyen címletű pénzekből lehet a legcélszerűbben kifizetni!
4. Egy családban a legfontosabb telefonszámokat noteszbe írják. Készítsd el azt a programot, amely ezeket a leggyakrabban használt telefonszámokat és címeket DATA-sorokban tárolja!

AZ MS-DOS 6.20 PARANCSAI

Ebben a fejezetben megadjuk a leggyakrabban használt MS-DOS-parancsok részletes paramétereit és kapcsolóit. Felsorolásunk elsősorban a kezdők számára készült, így nem tartalmazza az összes parancsot. A hiányzó utasítások leírásai a HELPben megtalálhatók, így azokat a gyakorlottabb felhasználók ott megtalálhatják. Nem írtuk le azokat a parancsokat sem, amelyeket az operációs rendszerhez külön megvásárolható kiegészítő lemezeken lehet megtalálni, továbbá nem foglalkoztunk az operációs rendszer korábbi verzióiban meglévő, de ebből kihagyott parancsokkal sem.

A felhasznált jelölések magyarázatai

[] A szögletes zárójelben lévő kifejezések használata elhagyható, de ez rendszert módosítja a parancsot.

| A két oldalán álló kifejezések egyikét vagy másikat lehet választani.

(K) Külső parancs.

(B) Belső parancs.

CHDIR | CD (CHange DIRectory, cséndzs direktri) Könyvtárváltás (B)

CD [meghajtó:][útvonal] név

CD

hatására kiírja jelenlegi helyzetünket a fájsztruktúrán belül;

CD név

hatására belép a megadott nevű könyvtárba;

CD..

hatására egy szinttel visszalép a könyvtárstruktúrában;

CD\
hatására a főkönyvtárba léphetünk.

CLS (Clear Screen, kliö szkrín) Képernyőtörlés (B)

A parancs törli a képernyőt, a kurzor a bal felső sarokba ugrik.

COPY (kopi) Másolás (B)

COPY [/A|/B] forrás [/A|/B] cél [/A|/B][/v]

forrás

Honnan, mit másolunk.

cél

Hova, milyen néven másolunk.

/V

Ellenőrzéssel másol.

/A

A fájlt mint szövegfájl értelmezi, és a szöveg vége jelig olvassa, illetve írásnál a végére szöveg vége jelet tesz.

/B

A fájl mint bináris fájl értelmezi, a fájl vége jelig olvassa, írásnál nem tesz utána szöveg vége jelet.

DATE (*dét*) **Dátum lekérdezése, módosítása**
DATE

(B)

Az új dátumot a gép által kijelzett régivel azonos módon kell beírni. Egy lehetséges példa: dd-mm-yy, ahol d (DAY) a nap, m (MONTH) a hónap, y (YEAR) az év rövidítése.

DEFRAG (DEFRAGmentation, *difregmentésn*) **Összerendezés**

(K)

A fájlműveletek közben a törlés-másolás hatására „lyukak” keletkezhetnek a lemez szektorai között, mivel nem biztos, hogy az adott fájl befér a még szabadon levő helyre. Ezt a gép úgy oldja meg, hogy a „lyuk” egy részét feltölti, a program további részeit a még szabadon maradt helyekre rakja. A parancs összerendezi a lemezen levő fájlkat, kitölti a lyukakat, ezáltal a későbbi fájlműveletek alkalmával gyorsabbá válik a parancs végrehajtása.

DEFRAG [mh:][/F] [/S:kód] [/B] [/SK] [/LCD] [/BW] [/GO] [/H] [/U]

mh

A meghajtó neve.

/F

Teljes optimalizálást végez.

/U

A fájlkat nem rendezi össze, de megszünteti a fájlkat közötti szüneteket, „lyukakat”.

/S:kód

A kódok lehetséges értékei:

n

– nevek (name, *ném*) szerinti sorrendbe;

e

– kiterjesztés (extension, *ikszstensön*) szerinti sorrend. A legrégebbi a legelső;

d

– az utolsó módosítás ideje (date and time, *dét end tájm*) szerinti sorrend;

s

– méret (size, *szájz*) szerinti sorrend. A legkisebb a legelső;

–

– az előző négy kód elé rakva fordított sorrendet alakít ki.

/B

Az optimalizálás után újra indítja a számítógépet.

/SKIPHIGH

Az optimalizálás során a memória kiterjesztéseit (pl.: extended) használja.

/LCD	LCD-monitort használ.
/BW	Kétszínű monitort használ.
/GO	Nem használ grafikus karaktereket.
/H	A rejtett fájlokat is átmozgatja.
DEL (DELeTe, <i>dilit</i>) Fájl törlése	(B)
DEL [meghajtó:][útvonal]fájlnév [/P]	Jóváhagyást kér a törlés megkezdéséhez.
/P	A képernyőn a következő üzenet jelenik meg :
Fájlnév DELETE (Y/N)?	Törlés (Igen/Nem)?
DELTREE (DELeTe TREE, <i>dilit tri</i>) Alkönyvtárak törlése	(K)
DELTREE[/Y] [meghajtó:][útvonal]	hatására nem kér jóváhagyást a törlés megkezdéséhez.
/Y	
DIR (DIRectory, <i>direktri</i>) Tartalomjegyzék kiírása	(B)
DIR [megh:][útvonal][fájlnév] [/P][/W][A[:]<i>attr</i>][O[:]<i>rend</i>][/S][/B][/1]	A meghajtó és a könyvtár neve.
[meghajtó:][útvonal]	Az állomány neve, melyre a parancsot végrehajtjuk.
[fájlnév]	A kapcsoló hatására egy képernyőnyi kiírás történik.
/P	A kapcsoló hatására egy képernyőnyi kiírás történik.
/W	A kapcsoló megadása esetén a parancs egy sorba öt fájl vagy könyvtár nevét írja ki.
/A[:] <i>típus</i>	Csak a megadott típusú állományokat listázza. A lehetséges típusok:
H	– rejtett (hidden, <i>hidn</i>);
S	– rendszer (system, <i>szisztöm</i>);
D	– alkönyvtár (directory, <i>direktri</i>);
A	– archiválandó;
R	– csak olvasható (read-only, <i>rid-onli</i>);
–	– az előzőek elé helyezve azok jelentését megfordítja.
/O[:] <i>sorrend</i>	Meghatározott sorrendben írja ki a listát.
	A lehetséges sorrendek:
N	– fájlnevek (name, <i>ném</i>) szerinti sorrend;
E	– kiterjesztések (extension, <i>iksztensön</i>) szerinti sorrend;

D	– az utolsó módosítás ideje (date and time, <i>dét end tájm</i>) szerint rendez. A legrégebbi az első;
S	– méret (size, <i>szájz</i>) szerint. A legkisebb a legelső;
G	– a könyvtárak és az állományok külön csoportba (group, <i>grup</i>) kerülnek. Az első csoport a könyvtáraké;
C	– a fájlok tömörítettsége szerint. A legkevésbé tömörített a legelső;
–	– az előző kapcsolók elé helyezve azok jelentését megfordítja.
/S	A megadott alkönyvtárban levő minden alkönyvtár tartalmát kijelzi.
/B	Minden név külön sorba kerül, de a fejléc és az összegző rész elmarad. Használata esetén a /W hatástalan.
/L	A parancs nem tesz különbséget a kis- és nagybetűk között. Célszerű, ha mindent kisbetűvel írsz.

DISKCOMP (*diszkomp*) Lemezek összehasonlítása (K)

Összehasonlítja két hajlékonylemez tartalmát. Azonosság esetén *Compare Ok*, hiba esetén *Compare error on side n, track m* (*Kömpeö errö on szájd en, trek em*) hibaüzenetet ad. A második meghajtónév nélkül az aktuális meghajtóval dolgozik. A **COPY** paranccsal másolt lemezek esetében hibajelzést ad.

DISKCOMP meghajtó1: meghajtó2 [/1] [/8]

meghajtó1:

Ezt a lemezt hasonlítjuk,

meghajtó2:

az ebben a meghajtóban levő lemezhez.

/1

Csak a lemez első oldalának ellenőrzése történik.

/8

Csak a lemez első 8 szektorát ellenőrzi.

DISKCOPY (*diszkopy*) Teljes lemezek másolása (K)

A két lemeznek azonos típusúnak kell lenni, mert másként nem használható hibamentesen.

DISKCOPY meghajtó1: meghajtó2: [/1][/V]

meghajtó1:	Erről a meghajtóról másolunk;
meghajtó2:	erre a meghajtóra.
/1	Csak a lemez első oldalának másolása.
/V	Ellenőrzéssel történik a másolás (ennek hatására lelassul a folyamat);
/M	A másoláshoz a memóriát használja.

EDIT Egyszerű szövegszerkesztő**(K)**

A parancs nem használható, ha nincs vele azonos könyvtárban a QBASIC.EXE fájl!

EDIT [meghajtó:] [útvonal] fájlnev] [/B] [/G] [/H] [/NOHI]

/B	Kétszínű kijelzéssel működik.
/G	CGA típusú monitor esetén a leggyorsabb képernyőfrissítést alkalmazza.
/H	Kijelzi a monitoron megjeleníthető sorok maximális számát.
/NOHI	Csak 8 színű kijelzést használ az alapértelmezés szerinti 16 helyett.

ERASE (iréz) Fájl törlése**(B)**

Ugyanaz a hatása, mint a **DEL** parancsé.

ERASE [meghajtó:][útvonal]fájlnev] [/P]**FORMAT (fómet) Lemez formázása****(K)**

A parancs előkészíti a lemezt arra, hogy a DOS használhassa. Létrehozza a gyökérkönyvtárat, felderíti a lemez esetleg hibás területeit, törli a lemezen eddig tárolt adatokat. A parancs beírásának egy lehetséges formája:

FORMAT [meghajtó:][/V[:címke]][/Q][/U][/F:méret][/B][/S][/C]

[meghajtó:]	Ebben a meghajtóban van a formálandó lemez.
/V	Lemeznevet akarunk adni.
[:címke]	Előre megadjuk a lemez nevét. E nélkül a formázás végén kéri a maximum 11 betűs lemeznevet.
/Q	Gyorsformázás. Törli a lemez információs részét, nem vizsgálja a lemezhibákat.

/U		A kapcsoló használatával visszaállíthatatlanul leromboljuk a lemezen előzőleg tárolt információkat.
/B		Helyet biztosít a rendszerfájlok későbbi átmásolásához.
/S		A rendszerfájlokat átmásolja a megformált lemezre.
/F:méret		A leggyakoribb lemezméretetek:
	360	– a DD-s vagy HD-s 5.25"-os lemez 360 kB kapacitású lesz;
	1200	– a HD-s 5.25"-os lemez 1.2 MB kapacitású lesz;
	720	– a DD-s vagy HD-s 3.5"-os lemez 720 kB kapacitású lesz;
	1440	– a HD-s 3.5"-os lemez 1.44 MB kapacitású lesz.
		A parancs kiadásánál a /S mellett nem szabad /B-t írni és viszont. További paraméterek a HELPben találhatóak.

MKDIR | MD (MaKe DIRectory, *mék direktri*) Alkönyvtár készítése (B)

MD [meghajtó:][útvonal] név

meghajtó:

Ebben a meghajtóban van a kérdéses lemez. Ha az aktuális meghajtóval dolgozunk, akkor nem kell megadni.

útvonal

A fastruktúrán belül egy út, aminek a végén a létrehozandó alkönyvtár lesz.

MSAV Vírusellenőrzés (B)

MSAV [meghajtó:][/S | /C][/R][/A | /L][/N][/P][/F][/VIDEO]

meghajtó:

Ezt a meghajtót szeretnénk ellenőrizni.

/S

Ellenőrzi az adott meghajtót, de nem törli a talált vírust.

/C

Ellenőrzi az adott meghajtót, és törli a talált vírust.

/R

Egy MSAV.RPT fájlt hoz létre, melyben rögzíti:

– a megtalált és vírussal fertőzött fájlok számát;

– a talált vírusok számát;

– az eltávolított vírusok számát.

/A	Az összes meghajtót ellenőrzi, kivéve az A és a B meghajtót.
/L	Az összes meghajtót ellenőrzi, kivéve a hálózati meghajtót;
/N	megvizsgálja az MSAV.TXT fájlt, ha ez üres, és abban az alkönyvtárban van, ahonnan az MSAV.EXE fájlt indítottuk. Ha vírust talál, kiírja a képernyőre.
/P	Parancs üzemmódban használjuk.
/F	Nem írja ki az ellenőrzött fájlok nevét. Csak a /N vagy /P kapcsolóval együtt használjuk.

LABEL (*lébl*) Lemeznév megadása (K)
LABEL meghajtó:

PATH (*pász*) Keresési út megadása (B)
 Pontosvesszővel választjuk el az egyes elérési utakat. Maximum 127 karakter hosszúságú lehet.

PATH [meghajtó:\]útvonal;[[meghajtó:\]útvonal;]
PATH kiírja az érvényes keresési utat;
PATH ; törli a jelenlegi keresési utat.

PRINT (*print*) Nyomtatás indítása (K)
PRINT [meghajtó:][útvonal]fájlnév [/C][/T][/P]

[meghajtó:][útvonal]fájlnév Ezt a fájlt akarjuk kinyomtatni. (Egymás után, szóközzel elválasztva több fájlnév is megadható keresési úttal együtt.)

/C A kijelölt fájlokat törli a várakozólistából.
/T Az összes fájlt törli a várakozólistából, s befejezi a nyomtatást.
/P Nyomtatás (ez az alapértelmezés).

PROMPT (*prompt*) Bejelentkezési szöveg megadása (B)
 A parancsfogadási készség (prompt) jelzősekor megjelenő felirat megadására szolgáló utasítás. Ez alaphelyzetben: **A\:>** vagy **C\:>** alakú az MS-DOS ezen verziójánál. A lehetséges paraméterek és jelentésük:

\$\$ – a \$ jel;
\$ – soremelés;

\$B	– a jel;
\$D	– az aktuális gépi dátum;
\$E	– az ESC billentyű kódja;
\$G	– a > jel;
\$H	– az utolsónak kiírt karakter törlése;
\$L	– a < jel;
\$N	– az aktuális meghajtó betűjele;
\$P	– az érvényes elérési út;
\$Q	– az = jel;
\$T	– az aktuális gépi idő;
\$V	– az MS-DOS verziószáma.

REN | RENAME (*riném*) **Fájl átnevezése** (B)

REN [meghajtó:][útvonal] régi új
[meghajtó:][útvonal] régi

Az átnevezendő állomány(ok) helyét és nevét adjuk meg.

új

Az állomány(ok) új neve.

RD | RMDIR (ReMove DIRectory, *rimúv direktri*) **Könyvtártörlés** (B)

A törlendő, üres alkönyvtár mindig a keresési út (útvonal) végén van.

RD [meghajtó:][útvonal] név

SYS **Rendszerfájlok átmásolása** (K)

A **COPY** paranccsal nem lehet átmásolni a rendszerfájlokat. A **SYS** parancs ezt a hiányosságot pótolja. A rendszerfájloknak a meghajtó1 főkönyvtárában kell lenni.

SYS meghajtó1: meghajtó2:

TIME (*tájm*) **Gépi idő lekérdezése, módosítása** (B)

Kiírja a gép órája által mutatott időt századmásodperc-pontosan, majd új idő megadását várja. Az új idő megadásának a kiírt régivel azonos formájúnak kell lennie. Egy lehetséges minta: hh:mm:ss, ahol h (heure) óra, m (minute) perc, s (second) másodperc.

TIME

TREE (*tri*) **Könyvtárstruktúra lekérdezése** (K)

A parancs kirajzolja a képernyőre a könyvtárszerkezet hálózatát.

TREE [meghajtó:][F][A]
[meghajtó:]

/F

/A

Ebben a meghajtóban van a vizsgálandó lemez.

A fájlok neveit is kiírja a képernyőre.

A kirajzolásához nem használ grafikus karaktereket.

TYPE (*tájp*) Szövegfájl kiíratása
TYPE [meghajtó:][útvonal] fájlnev

(B)

UNDELETE (*ándilit*) Törölt fájlok visszaállítása

(K)

UNDELETE [[meghajtó:][útvonal][fájlnev] [/LIST[/ALL] [/DOS]
[meghajtó:][útvonal]fájlnev

/LIST

/ALL

/DOS

A visszaállítandó állomány, vagy állománycsoport helyét és nevét adjuk meg. Alapértelmezése az aktuális könyvtár.

Listázza azokat a törölt állományokat, melyekről elegendő információval rendelkezik a visszaállításhoz.

Visszaállítja a törölt állományokat, kérdés nélkül.

Minden olyan visszaállítandó fájlról jóváhagyást kér, amelyek az MS-DOS könyvtárbejegyzésében törölt fájlként szerepelnek.

UNFORMAT (*ánfómet*) Lemeztartalom visszaállítása

(K)

A parancs segítségével visszaállíthatjuk a **FORMAT** paranccsal formázott lemez tartalmát. A parancs a merevlemezek hibás partíciós adatait is korrigálja. Az adott meghajtón levő lemez adatait visszaállítja. Ha a lemezt a **FORMAT /U** paranccsal formáztuk, akkor az **UNFORMAT** hatástalan marad.

UNFORMAT meghajtó: [/L][TEST][P]

meghajtó:

/L

/TEST

/P

Ebben a meghajtóban van a lemez.

Az összes megtalált fájl és alkönyvtár nevét kilistázza.

A képernyőre írja a visszaállításra váró tartalmat, de visszaállítás nem történik.

Az LPT1 portra csatlakoztatott nyomtatóra írja a tartalmat.

VER (VERsion <i>vőzsn</i>) DOS verzió lekérdezése	(B)
VER	A kiírja a DOS verzió nevét és számát.
VERIFY (<i>verifáj</i>) Lemezműveletek ellenőrzése	(B)
VERIFY	A lemezműveleteknél ellenőrzi, hogy az adatokat hiba nélkül el lehet-e olvasni.
VERIFY ON	Kiírja a kapcsoló állását (ON / OFF).
VERIFY OFF	Bekapcsolja az ellenőrzést. Kikapcsolja az ellenőrzést.
VSAFE Vírusvédelmi rendszer	(K)
	A program indítás után a memóriába kerül, ha ott legalább 22 Kbyte szabad hely van. A WINDOWS futtatása esetében nem szabad használni!
VSAFE [/opciók [+ -]] [/NE][/NX][/N][/D][/U]	
opciók	A + vagy a – jelet használjuk a számok után az opció be- vagy kikapcsolásához. Minden opció után zárójelben szerepel az alapbeállítás.
1	– előre figyelmeztet, ha winchester törlés következne be (+);
2	– előre figyelmeztet, ha a memóriába kerülne egy vírus (–);
3	– megghiúsítja a vírus lemezre kerülését (–);
4	– megakadályozza fertőzött program indítását a DOS-ból (+);
5	– megvédi a teljes lemezt a BOOT-vírusoktól (+);
6	– előre figyelmeztet, ha a vírus a winchester BOOT-szektorát, illetve a partícióstáblát veszélyezteti (+);
7	– figyelmeztet, ha a vírus a hajlékonylemez BOOT-szektorát veszélyezteti (–);
8	– előre figyelmeztet, ha a végrehajtható fájl fertőződött (–).
/NE	Az expanded memóriába töltődik.
/NX	Az extended memóriába töltődik.
/N	A program a hálózati gépet védi.
/D	Az ellenőrző összeg figyelésének kikapcsolása.
/U	A vírusvédelem kikapcsolása.

XCOPY Fájlok csoportos másolása**(K)**

A **COPY** parancs kiegészítéseként itt lehetőség van akár teljes alkönyvtárak átmásolására is.

XCOPY forrás [cél][/A | /M][/D:dátum][/P][/S][/E][/V][/W][/Y]

forrás

Innen másolunk.

cél

Ide másolunk. Ha a cél nem létező könyvtár tartalmaz vagy hiányzik a \ jel, akkor a következő üzenetet kapjuk: **Does destination specify a file name or directory name or target (F=file, D=directory)?** Azaz meg kell mondanunk, hogy könyvtár vagy fájl nevét adtuk meg. Válaszunktól függően tekinti fájl-, illetve könyvtárnévnek a megadott karaktersorozatot.

F

A fájlokat az aktuális alkönyvtárba,

D

az adott néven létrehozott alkönyvtárba másolja. Ez biztosítja a könyvtárstruktúrák másolását.

/P

Minden fájlnál rákérdez, másolja vagy ne.

/S

A forrás tartalomjegyzék minden, nem üres könyvtárát és alkönyvtárát átmásolja.

/E

Az üres alkönyvtárakat is átmásolja.

/V

Ellenőrzi a másolást.

/W

Megerősítést vár másolás előtt.

/A

Csak az archív állományokat másolja át.

/M

Átmásolja az archív állományokat, de másolás után törli az archív beállításukat.

/D:dátum

Az adott időpontban, vagy ezután módosított állományokat másolja.

/Y

Ha a célfájl már létezik, a fájl átírására nem vár megerősítést;

/-Y

Ha a célfájl már létezik, a fájl átírására megerősítést vár a prompt után.

MS-DOS HIBAÜZENETEK

Perifériákkal kapcsolatos üzenetek

[típus] ERROR READING (WRITE) [meghajtó]	Abort, Retry, Ignore?
[típus] hiba az olvasás (írás) során az adott [meghajtón].	
Kilépés, újra ismételjem, hagyjam figyelmen kívül?	
Válaszolni a megfelelő kezdőbetű leütésével kell: A vagy R vagy I .	
A hiba típusa lehet:	
BAD COMMAND (<i>baed kömand</i>)	Hibás parancs a perifériának. Javítsuk ki a parancsot, s utána futtatható!
DATA (<i>détö</i>)	Adathiba. A lemezen hibás rész található. Formázzuk újra!
DISK	Lemezhiba.
DRIVE NOT READY (<i>drávj not redi</i>)	A lemezegység nem használható. Nincs benne lemez. A lemezegység író/olvasó feje elakadt.
GENERAL FAILURE (<i>dzsenöröl féljör</i>)	Súlyos hiba (azonosíthatatlan). Hibás (formázatlan) lemez esetében is így jelez.
NON SYSTEM DISK (<i>non szisztem diszk</i>)	Nem rendszerlemez! A lemezen nincs operációs rendszer.
NOT READY	A berendezés nem működőképes. A hibaokok azonosak lehetnek a DRIVE NOT READY -nél említettekkel.
READ FAULT (<i>rid fólt</i>)	Olvasási hiba. A DOS nem tudja elolvasni az adatokat.
SECTOR NOT FOUND (<i>szektor not fánd</i>)	A lemezen hibás szektorok találhatók.
WRITE PROTECT (<i>rájt prötekt</i>)	Írásvédelmi hiba. Írásvédett (leragasztott) lemezre akarunk írni.
WRITE FAULT (<i>rájt fólt</i>)	Íráshiba. Írásvédett fájlban akarunk adatot módosítani.

Rendszerhibák

Acces denied (<i>ekszesz dinájd</i>)	Érvénytelenített művelet.
Are you soure (Y/N)? (<i>ár ju suö</i>)	Biztos benne (Igen/Nem)? Az Y és az ENTER megnyomására végrehajtja a parancsot.

Attempted write-protect violation (<i>ötemptid rájt-prötekt violésn</i>)	Írásvédelem megsértésére történt kísérlet.
Bad command or file name (<i>baed kömand or fájl ném</i>)	Hibás parancs vagy állománynév.
Bad or missing command interpreter (<i>baed or miszing kömand</i>)	Rossz vagy hibás parancsértelmező. Nem találja meg a COMMAND.COM fájlt.
Cannot find system files (<i>kenöt fájnd szisztöm fájlz</i>)	A rendszerfájlok nem találhatóak a lemezen.
Cannot load COMMAND, system halted (<i>kenöt lód kömand szisztem háltid</i>)	A COMMAND.COM nem tölthető be, a rendszer leállt. Lehet, hogy megsérült a lemez vagy vírusos.
Current drive is no longer valid (<i>karönt drájev in no longö veelid</i>)	Az aktuális lemezegység elérhetetlen. Használjuk a másik egységet, írjuk be annak a nevét.
Disk boot failure (<i>diszk bút féljö</i>)	A lemezegység boot-szektora hibás. A betöltéskor hiba történt, vagy vírus van benne.
Disk full (<i>diszk full</i>)	A lemez megtelt. Ellenőrizzük a forrás méretét és a célmeghajtó szabad bájttjainak számát.
Duplicate file name or file not found (<i>djuplikit fájl ném ó fájl not fáund</i>)	Két fájl neve azonos vagy nem található a fájl. Változtassuk meg a nevet, amit megadtunk, vagy az elérési út megadása volt hibás.
Entry error (<i>entri erö</i>)	Parancsmegadási hiba. Rosszul írtuk be a parancsot.
Error in .EXE file (<i>erö in exe fájl</i>)	Hiba van az adott .EXE fájlban. Lehet, hogy megsérült vagy vírusos.
Error loading operating system (<i>erö lóding opöréting szisztöm</i>)	Hibás az operációs rendszer betöltése. Másik lemezeről ismételjük meg a betöltést.
Error writing to device (<i>erö rájting to divájsz</i>)	Hiba történt a meghajtóra való íráskor.

File cannot be copied onto itself

(fájl kenöt bi kopid ántu itszelf)

File creation error

(fájl kriésön error)

File not found

(fájl not fánd)

Format failure

(fómet feilö)

Incorrect DOS version

(inkorekt DOS vőzsn)

Incorrect numbers of parameters

(inkorekt námböz öv paramétőz)

Insufficient disk space

(inszöfisönt diszk szpész)

Insufficient memory

(inszöfisönt memori)

Invalid date

(invelid deit)

Invalid disk change

(invelid diszk cséndzs)

Invalid drive in search path

(invelid drájv in szőcs pász)

Invalid drive or file name

(invelid drájv or fájl neim)

Invalid drive specification

(invelid drájv szpeszifikésön)

Invalid file name or file not found

(invelid fájl ném or fájl not found)

Invalid number of parameters

(invelid námbö öv paramétőz)

Az állomány önmagára nem másolható. Másik alkönyvtárba vagy másik néven kell másolni.

Fájl előállítási hiba. A fájl védett vagy betelt a lemez.

A fájl nem található. Hibás az elérési út megadása, vagy a fájl nem létezik.

Formázási hiba. A lemez hibás, másik lemezt használjunk.

Hibás verziójú DOS-t használtunk. A megadott parancs nem felel meg az indításkor használt DOS-verziónak.

Rossz a paraméterek száma. Kifelejtettünk, vagy hibásan írtunk (szóköz) egy paramétert.

Nincs elég hely a lemezen. Betelt a lemez.

Kevés a memória. A parancs vagy program végrehajtásához kevés a szabad memóriaterület.

Hibás dátumbeírás. A beírás formátuma, vagy értéke nem egyezik az előírással.

Lemez csere tilos! A lemezen még nyitott állományok vannak, tegyük vissza a lemezt!

Helytelen a keresési út megadása. A PATH sorában hibáztunk.

Érvénytelen lemezegység vagy állománynév.

Érvénytelen meghajtónév. A rendszer nem ismerte fel a megadott meghajtót.

Érvénytelen állománynév, vagy a fájl nem található. Hibás az átnevezési, másolási parancs.

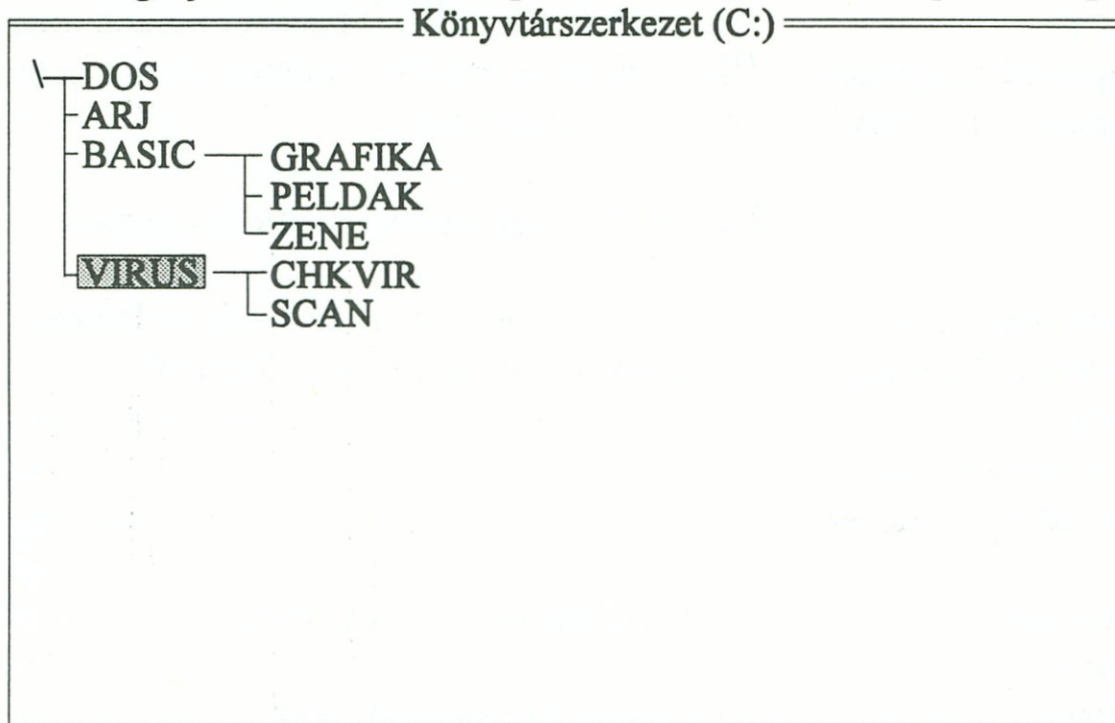
Érvénytelen számú paramétert adtunk meg. Túl kevés, vagy túl sok.

Invalid path (<i>invalid pász</i>)	Érvénytelen a keresési út megadása.
Invalid path, not directory or directory not empty (<i>...not direktri or ... not empty</i>)	Érvénytelen a keresési út megadása, nem alkönyvtár, vagy az nem üres.
Invalid path or file name (<i>invalid pász or fájl név</i>)	Helytelen keresési út vagy állománynév.
Invalid time (<i>invalid tájm</i>)	Érvénytelen idő megadása. Az idő megadása helytelen, vagy érvénytelen időpontot adtunk meg.
Missing operating system (<i>missing opöreiting szisztem</i>)	Hiányzik az operációs rendszer. A rendszerfájlok hiányoznak a lemeztől .
No path (<i>no pász</i>)	Nem adtuk meg az elérési utat.
Non-System disk error. Replace and strike a key when ready. (<i>Non-szisztem diszk erő. Riplész end sztrájk e kí uen redi</i>)	Nem rendszerlemez, vagy hibás a lemez. Cseréljük ki, és nyomjunk meg egy billentyűt!
Path not found (<i>pász not fánd</i>)	Nem található meg az elérési út .
Path too long (<i>pász tú long</i>)	Az elérési út túlságosan hosszú.
Syntax error (<i>szintex erő</i>)	A hiba a parancs beírásakor történt .
System transfered (<i>szisztöm trenszfőd</i>)	A rendszer átvitele megtörtént. A rendszerfájlok rákerültek a formázott lemezre.
Unrecognized command in CONFIG.SYS (<i>ánrekögnájzd kömand in...</i>)	Ismeretlen parancs a CONFIG.SYS fájlban. Javítsuk ki, majd indítsuk újra a számítógépet!

A CHKVIR VÍRUSÖLŐ RENDSZER

A rendszer bejelentkező képernyőjén megjelenik a „Chkvir töltése...” felirat, majd a programcsomag indulóképe. Tetszőleges billentyű lenyomására továbblép. Elsőként a memóriát vizsgálja meg. Ha ebben ismert vírust talál, automatikusan kiírtja onnan. Ismeretlen vírus esetén „ledermed”. Ezután beolvassa az adott meghajtó könyvtárszerkezetét, majd ezt a képernyőn megjeleníti.

≡ Meghajtók Ellenőrzés Opciók Információ Kilépés Help



F1 Help | Ellenőrzés engedélyezése / letiltása az alkönyvtárban

A legfelső sorban (menü) megjelenő széles sávot a balra, illetve jobbra mutató nyíllal tudjuk mozgatni, és **ENTER**rel választhatunk. Egérrel is használható a program.

A menüpontok leírása

≡ Megjelenik a rendszer rövid leírása.

MEGHAJTÓK

A megjelenő ablakban egymás alatt a lehetséges meghajtók közül a föl-le nyíl és **ENTER** megnyomására választhatunk. A kiválasztás után az egység könyvtárszerkezetének feltérképezése történik.

ELLENŐRZÉS

Víruskeresés **F2**

Víruskeresés és irtás **F3**

Több lehetőségünk van:

csak keresés;

keres és kiirt;

Memóriaellenőrzés	F4	az induláskor történt ellenőrzés megismétlése lehetséges;
Eredmények	F5	az utolsó keresésre vonatkozik.

OPCIÓK

A keresési és kiölési paraméterek állíthatók be:

Keresési eljárások**Vírusok**

Általános keresés

normál keresés;

Alapos szekvenciakeresés

a teljes állományt átvizsgálja;

Gyors szekvenciakeresés

csak az állományok elején és végén keres;

Mutációellenőrzés

ismert vírusok mutánsainak keresése;

Ismeretlen víruskeresés

ismeretlen vírusok keresése.

Külső adatfile-ok

A vírusok leírását tartalmazó külső fájlkat lehet itt megadni. A fájlok speciális szabályok szerint szerkesztendők!

Byte-szám

Itt adható meg, hogy a gyors szekvenciakeresés során az állományok elején és végén hány bájtot ellenőrizzen.

Tömörített file-ok

Nincs jelzés

a megtalált tömörített fájlkat átlépi a kereséskor;

Üzenet

üzenetet küld, ha tömörített állományt talál;

Analízis

ellenőrzi, hogy a tömörítés előtt tartalmazott-e az állomány ismert vírust.

Tönkretett file-ok

Nincs jelzés

nem történik semmi;

Üzenet

üzenetet küld a Chkvir, hogy mi történjen. A keresés folytatását, leállítását vagy a fájl törlését választhatjuk;

Törlés

a tönkretett fájlkat automatikusan törlésre kerülnek.

Immunizált file-ok

Nincs jelzés

nem történik semmi;

Üzenet	üzenetet küld a Chkvir. A fájl visszaállítása, a keresés folytatása, illetve a keresés leállítása választható;
Visszaállítás	megkísérli az eredeti állományt visszaállítani.
Keresési terület	
Szektorok	Beállítható, hogy a program mely szektorokban végezzen keresést:
Partíciós tábla	
Boot-szektor	
Egyéb szektorok	
File kijelölési minták	A megadott fájlokat fogja ellenőrizni a program. Ezen fájlok nem kerülnek ellenőrzésre. A felsorolásokban használható a * és a ? jel. A lista elemeit ; választja el egymástól.
File elhagyási minták	
Egyéb	
Report file	Megadható, hogy készüljön-e tájékoztató fájl a program működéséről:
Nincs	nem készül;
Normál	csak a vizsgálat eredményét tartalmazza;
Részletes	az ellenőrzött területek, alkönyvtárak nevei mellett a rendszerinformációk is benne lesznek az emlékeztetőben.
Vírusirtás	
Eredeti eljárás	a programban megadott eljárás alapján;
VirSec adatai alapján	a kiöléshez felhasználja a VirSec által elmentett információkat.
Egyéb	
Alkönyvtárak	a könyvtárszerkezetben beállított jelölések az alkönyvtárakra is öröklődnek;
Hangjelzés	bármilyen találat vagy hiba esetén hangjelzést ad;
Mutációk kiölése	a mutánsokat is kiölje-e;

TREINFO.NCD**Eredménygyűjtés****Ideiglenes könyvtár****Opciók mentése****INFORMÁCIÓ****Vírusok****Szekvenciák****Tönkretevők****Tömörítők****Immunizálók****KILÉPÉS****HELP****ALT-X****F1**

az induláskor a könyvtárszerkezetet a Norton Change Directory program által létrehozott fájlból olvassa be;

készüljön-e részletes statisztika a keresésről és irtásról.

A keresés és irtás során szükség lehet ideiglenes fájlok létrehozására. Ezek helyét adjuk meg itt.

A beállított opciók mentése a ChkVir.CFG fájlba, amely minden indításkor be fog tölteni.

Ismert vírusok listája.

A beépített szekvenciák listája.

Azon vírusok jellemzői, amelyek tönkretesznek állományokat, területeket.

Futtatható állományokat tömörítő programok listája.

Futtatható állományokat immunizáló programok információi.

NORTON COMMANDER 4.0

A menü leírása

Az **F9** billentyűt megnyomva vagy az egérrel a felső sor adott pontját kijelölve gördülnek le az alábbi menük. Minden menüpontban az aláhúzással jelölt betű billentyűjét vagy az adott ponton az egér megfelelő gombját lenyomva lehet a funkciót kiválasztani.

LEFT, illetve RIGHT

A **LEFT** (*left*) és a **RIGHT** (*rájt*) pontokhoz csaknem azonos menüsor tartozik, így ismertetésük is egyszerre történik.

√ <u>B</u> rief		<u>B</u> rief	
F <u>u</u> ll		√ <u>F</u> ull	
I <u>n</u> fo		I <u>n</u> fo	
T <u>r</u> ee		T <u>r</u> ee	
Q <u>u</u> ick v <u>i</u> ew		Q <u>u</u> ick v <u>i</u> ew	
C <u>o</u> mpressed F <u>i</u> le		C <u>o</u> mpressed F <u>i</u> le	
L <u>l</u> ink		L <u>l</u> ink	
O <u>n</u> / O <u>ff</u>	Ctrl - F1	O <u>n</u> / O <u>ff</u>	Ctrl - F2
<u>N</u> ame	Ctrl - F3	<u>N</u> ame	Ctrl - F3
√ <u>E</u> xtension	Ctrl - F4	√ <u>E</u> xtension	Ctrl - F4
T <u>i</u> me	Ctrl - F5	T <u>i</u> me	Ctrl - F5
S <u>i</u> ze	Ctrl - F6	S <u>i</u> ze	Ctrl - F6
U <u>n</u> sorted	Ctrl - F7	U <u>n</u> sorted	Ctrl - F7
<u>R</u> e - read		<u>R</u> e - read	
F <u>i</u> lter ...		F <u>i</u> lter ...	
D <u>r</u> ive ...	Alt - F1	D <u>r</u> ive ...	Alt - F2

Brief (*brif*)

Csak az állományok nevét írja ki az ablak 3 oszlopába.

Full (*ful*)

Az állományok részletesebb leírását adja:

Name
Size
Date
Time

– név;
– méret;
– utolsó módosítás napja;
– utolsó módosítás időpontja.

Info (*infó*)

A másik ablakban kijelzett meghajtóra vonatkozó adatok felsorolását végzi, valamint megadja a memória és a winchester adatait is.

Tree (*tri*)

A teljes könyvtárszerkezet megjelenik az ablakban. Kurzorral ráállva egy könyv-

Quick View (*kvik vjú*)

Compressed file (*kompreszd fájl*)

Link (*link*)

tárra, a másik ablakban – beállítástól függően automatikusan – megjelenik az adott könyvtár tartalomjegyzéke.

A kiválasztott fájl tartalma megjelenik az editor ablakban, illetve a kiválasztott könyvtár jellemzői is megjelennek.

Az ablakban a tömörített fájlokat mutatja. Az adott csatlakozó portokon keresztül kapcsolatot létesíthetünk más számítógéppel, ha a két gép között a kiválasztott kimeneten keresztül vezetékes kapcsolat is van.

Commander Link

Choose wich mode and port
you want for this panel

Serial ports <input checked="" type="radio"/> COM 1: <input type="radio"/> COM 2: <input type="radio"/> COM 3: <input type="radio"/> COM 4:	Paralel ports <input type="radio"/> LPT1: <input type="radio"/> LPT2: <input type="radio"/> LPT3: <input type="radio"/> LPT4:
----------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------

Select communication mode

Master Slave

On/Off CTRL-F1, illetve Ctrl-F2

Name (*ném*)

Extension (*iksztenšn*)

Time (*tájm*)

Size (*szájz*)

Unsorted (*ánszótíd*)

Re-read (*ri-ríd*)

A bal, illetve jobb oldali ablak eltűnik, s láthatóvá válnak a DOS-képernyőre kiírt feliratok.

Névsor szerint rendezi a fájlokat.

Kiterjedés szerint rendezi a fájlokat.

Az utolsó módosítás ideje szerint rendez.

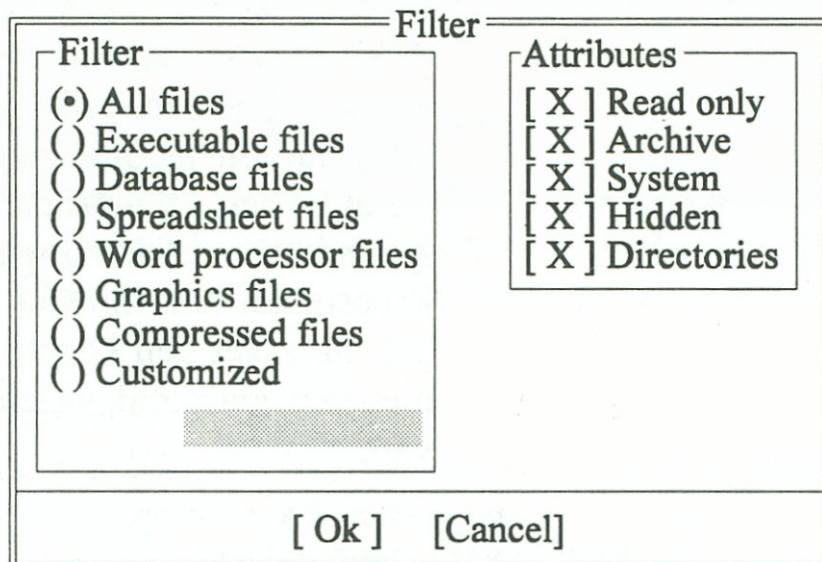
Méret szerint rendez.

Rendezetlenül írja ki a fájlokat, a lemezen található fizikai sorrendben.

Újraolvassa az aktuális panelban szereplő alkönyvtárat és új tartalomjegyzéket készít róla. Erre akkor lehet szükség, ha a hajlékonylemez tartalomjegyzéke szerepel a panelon, de a lemezt közben kicseréltük.

Filter... (filtő)

Szűrőparancs a tartalomjegyzékhez. Kiválasztva megjelenik egy új ablak



All files (*ól fájlz*)

Executable files (*ekszikjutöbl fájlz*)

Database files (*détöbész fájlz*)

Spreadsheet files (*szpredsít fájlz*)

Word processor files (*vöd proszeszö fájlz*)

Graphics files (*grefiksz fájlz*)

Compressed files (*kömpreszd fájlz*)

Customised (*kásztömizd*)

Minden fájl.

Futtatható fájlok.

Adatbáziskezelő programmal írt fájlok.

Táblázatkezelők fájljai.

Szövegszerkesztők fájljai.

Rajzolóprogramok fájljai.

Tömörített fájlok.

A felhasználó által megadott fájlok. A beállítást a szürke sávba kell írni.

Drive ALT-F1, illetve Alt-F2

Itt lehet meghajtót váltani. Az adott betűre ráállva és az **ENTER**t vagy az adott betűt megnyomva a kívánt meghajtó tartalomjegyzéke látszik.

FILES

<u>H</u> elp	F1
<u>U</u> ser menu	F2
<u>V</u> iew	F3
<u>E</u> dit	F4
<u>C</u> opy	F5
<u>R</u> ename or move	F6
<u>M</u> ake directory	F7
<u>D</u> elete	F8
File attributes	
<u>S</u> elect group	Gray +
<u>D</u> eselect group	Gray -
<u>I</u> nvert selection	Gray *
<u>R</u> estore selection	
<u>Q</u> uit	F10

Help (*help*)

Az adott utasításhoz tájékoztató szöveget ad angol nyelven.

User menu (*júzó menü*)

Egy ablakban megjeleníthető az általunk előre megírt, s NC.MNU fájlban tárolt felhasználói menü.

View (*vjú*)

Egy ablakban megtekinthető a fájl tartalma, de nem szerkeszthető.

Edit (*edit*)

Az editorban javítható módon mutatja a fájl tartalmát. A DOS szövegfájlok módosítására használjuk.

Copy (*kopi*)

Átmásolja a kijelölt fájl(oka)t a másik ablakban látható könyvtárba.

Rename or move (*riném or múv*)

Átnevezhetjük a kijelölt fájlt, vagy átmozgathatjuk a kijelölt fájlt a másik ablakban látható könyvtárba.

Make directory (*mék direktri*)

Új könyvtárat hozhatunk létre.

Delete (*dilit*)

Törölhetjük a kijelölt fájl(oka)t, vagy könyvtárat.

File attributes (*fájl atribútsz*)

Egy megjelenő ablakban a **SPACE** billentyűvel vagy egérrel beállíthatjuk, illetve törölhetjük a fájl jellemzőit. A következő két kép közül a bal oldali akkor jelenik meg, ha egy fájlt választottunk ki; a jobb oldali pedig akkor, ha többet.

Attributes	
Change file attributes for " . "	
<input type="checkbox"/>	Read only
<input type="checkbox"/>	Archive
<input type="checkbox"/>	Hidden
<input type="checkbox"/>	System
[Set] [Cancel]	

Attributes	
Change file attributes	
Set	Clear
<input type="checkbox"/>	<input type="checkbox"/> Read only
<input type="checkbox"/>	<input type="checkbox"/> Archive
<input type="checkbox"/>	<input type="checkbox"/> Hidden
<input type="checkbox"/>	<input type="checkbox"/> System
[Set] [Cancel]	

Read only (*rid onli*)

Archive (*arhiv*)

Hidden (*hidn*)

System (*szisztem*)

Select group (*szélekt grup*)

Deselect group (*diszélekt grup*)

Invert selection (*invőt szileksn*)

Restore selection (*risztór szileksn*)

QUIT (*kvit*)

COMMANDS

NCD tree	Alt - F10
Find file	Alt - F7
History	Alt - F8
EGA lines	Alt - F9
System information	
Swap panels	Ctrl - U
Panels on / off	Ctrl - O
Compare directorys	
Terminal emulation	
Menu file edit	
Extension file edit	

NCD tree (*enszίδi trί*)

– csak olvasható

– archivált

– rejtett

– rendszer

A numerikus billentyűzet + gombját megnyomva megjelenik egy ablak. A szürke mezőbe írhatjuk a kiválasztási szabályt.

A numerikus billentyűzet – gombját megnyomva, a kiválasztás megszüntetésére szolgál. Egyedi esetben inkább az **INSERT** billentyűt használjuk.

Az adott alkönyvtár eddig kijelölt állományainál a kijelölést megszünteti, az eddig ki nem jelölteket pedig kijelöli.

Az utolsónak kiadott kiválasztási parancsot megismétli.

Kilépés a programból.

Kiírja az aktuális meghajtó könyvtárszerkezetét, és megadja, hogy melyik alkönyv-

Find file (*fánd fájl*)**Chdir****New search****Cange drive****View****QUIT FF****History** (*hisztri*)**EGA lines** (*EGA lájnz*)**System information** (*szisztöm infomésön*)**Swap panels** (*szvep pénlz*)**Panels On/Off** (*pénlz on/off*)**Compare Directories** (*kömpeör direktriz*)**Terminal Emulation** (*terminl emjulésn*)**Menu file edit** (*menü fájl edit*)

tárban vagyunk. Az alsó sorban megjelenik a gyorskereső ablak. A kurzormozgató billentyűkkel változtathatjuk helyünket. Az adott fájlt vagy bármely jelölését beírva az ablakba megkeresi az összes ilyet, majd kiírja. A keresésnél a ? és a * karakterek is használhatók. A megtalált fájlokat listázza, majd a következő lehetőségek közül választhatunk:

- beléphetünk az általunk kijelölt fájl alkönyvtárába;
- újabb fájl keresését végezhetjük el;
- meghajtóváltás;
- megnézhetjük a fájl tartalmát;
- kilépünk ebből a funkcióból.

Az előzőleg kiadott DOS-parancsokat jeleníti meg. A kurzormozgató billentyűkkel tudunk közülük választani, majd **ENTER**rel ismételhetjük meg a parancs kiadását.

Csak EGA vagy VGA monitorok esetében működik. A képernyőt 24, 43 vagy 50 sorra lehet beállítani.

A számítógép paramétereit nézhetjük meg:

Felcseréli a két ablak tartalmát.

Megjeleníti, vagy eltünteti a NORTON ablakait a képernyőn.

Összehasonlítja a két ablak tartalmát. Eredményeként mindkét ablakban kifényesedve vannak azok a fájlok, amelyek nem találhatók meg a másik könyvtárban.

A számítógéphez csatlakoztatott eszközök és a számítógép közti üzenetváltást teszi lehetővé.

Itt definiálhatunk saját parancsokat a különböző billentyűkombinációkhoz. A definiálást sorszámmal kell kezdeni, majd az adott nevet beírni. Egy lehetséges menü leírást közlünk:

1. Rendszerlemez

format a:/s

2. Lemezformázás

format a:

Az **F2** billentyűt megnyomva felvesszük az NC.MNU fájlba, majd az **ESC**re kilépünk ebből a menüpontból. A saját menü előhívásához elegendő az **F2** billentyűt megnyomni, s az általunk írt lehetőségek közül máris választhatunk.

A nem futtatható fájlok esetén megadhatjuk, hogy kiválasztásuk esetén mi történjék. A hozzárendeléseket az NC.EXT fájl tartalmazza. Az ARJ-vel tömörített állományok kicsomagolásához jól jöhet a következő bejegyzés:

arj: arj x !.! c:\valami

Extension file edit**OPTIONS**

<u>C</u> onfiguration ...	
E <u>d</u> itor ...	
<u>C</u> onfirmation ...	
<u>C</u> ompression ...	
<u>A</u> uto menus	
√ <u>P</u> ath prompt	
√ <u>K</u> ey bar	Ctrl - B
√ <u>F</u> ull screen	
<u>M</u> ini status	
√ <u>C</u> lock	
<u>S</u> ave setup	Shift - F9

Configuration (könfigjurésn)

A beállítások elkészítéséhez egy ablak jelenik meg.

Configuration	
<p>Screen colors</p> <p><input checked="" type="radio"/> B&W <input type="radio"/> Laptop</p> <p><input type="radio"/> Color 1 <input type="radio"/> Color 2</p>	<p>Panel options</p> <p><input checked="" type="checkbox"/> Show hidden files</p> <p><input checked="" type="checkbox"/> Ins moves down</p> <p><input checked="" type="checkbox"/> Select directoris</p> <p><input checked="" type="checkbox"/> Auto change directory</p>
<p>Screen blank</p> <p><input checked="" type="checkbox"/> On</p> <p>03 Minutes</p>	<p>Other options</p> <p><input checked="" type="checkbox"/> Menu bar always visible</p> <p><input checked="" type="checkbox"/> Auto save setup</p> <p><input checked="" type="checkbox"/> Left - handed mouse</p> <p><input checked="" type="checkbox"/> Fast mouse reset</p> <p><input checked="" type="checkbox"/> Copy newer files only</p>
<p>Printer port</p> <p><input type="radio"/> COM1 <input checked="" type="radio"/> LPT1</p> <p><input type="radio"/> COM2 <input type="radio"/> LPT2</p> <p><input type="radio"/> COM3 <input type="radio"/> LPT3</p>	
<p>Press Space to change an option, ↑ and ↓ to move between options</p>	
<p>[Ok] [Cancel]</p>	

Screen colors (*szkrín kálörsz*)

Black & White

Color 1

Color 2

Laptop

Screen blank delay (*szkrín blenk diléj*)

Képernyő színek:
– kétszínű monitor;
– színes monitor kétféle színkiosztási lehetőséggel;
LCD monitorhoz.

A beírt szám a képernyő kikapcsolásának, „csillagossá” válásának várakozási idejét jelenti percben. (Ha a megadott időn túl nem történik képernyőművelet, akkor a rendszer lekapcsolja a képernyőt.) Ha az egér a jobb felső sarokba kerül, ugyanez történik.

Panel options (*péni opsnz*)

Show hidden file

Ins moves down

Select directories

Auto change directory

a rejtett fájlokat is mutatja a panelon.

Az **INSERT** gombbal történő kijelölésnél automatikusan továbbugrik a következő fájlra (vagy beállítástól függően a következő alkönyvtárra).

Az alkönyvtárakat is kiválasztja.

Automatikus tartalomjegyzékcseré. Így az alkönyvtárat kiválasztva belelép, s mutatja a benne levő fájlokat.

Other options (*ádö opsnz*)

Menu bar always visible

A menüsor mindig látszik.

Auto save setup

A beállításokat automatikusan menti. Ezt választva, milyen beállítással kilépünk, ugyanazzal a képpel fog ismét bejelentkezni a NORTON.

Left handed mouse

Az egér balkezes beállítású.

Fast mouse reset

Az előző egérbeállítás törlése.

Printer port

A nyomtató csatlakozási helyét adhatjuk meg.

Editor (*editő*)

Kiválaszthatjuk, melyik szerkesztőt használjuk az F4 billentyű lenyomásakor.

Built-in

Beépített.

External

Más szerkesztőt. Ekkor az alatta lévő sorba be kell írni a megfelelő fájl nevét és utána a !.! jelsorozatot. (pl.: ne !.!)

Confirmation (*könfömésn*)

Mely esetekben kérjen a rendszer megerősítést a parancs végrehajtása előtt?

Confirmations	
<input checked="" type="checkbox"/> Copy	
<input checked="" type="checkbox"/> Move	
<input checked="" type="checkbox"/> Delete	
<input checked="" type="checkbox"/> Delete subdirectories	
<input checked="" type="checkbox"/> Exit	
[Ok] [Cancel]	

Másoláskor

Fájl (vagy alkönyvtár) mozgásakor

Törlés esetén

Alkönyvtár törlésekor

Kilépéskor

Compression (*kömpresn*)

Tömörítési beállítások

Compressions Options	
Storage method	
<input checked="" type="radio"/> Automatically select best method	
<input type="radio"/> Create smallest file	
<input type="radio"/> Use fastest method	
General	
<input type="checkbox"/> Confirm with password	
[Ok] [Cancel]	

Tömörítési módszer

Automatikusan a legjobbat választja

A legkisebb méretre tömörít

A leggyorsabb módszert használja

Általánosan

Jelszóval tömörít

A következő beállításokat **ENTER**rel, vagy egérrel kiválasztva \surd jel jelenik meg a felirat bal szélén.

Auto menus (*auto menjuz*)

A felhasználói menü automatikusan megjelenik minden indításkor.

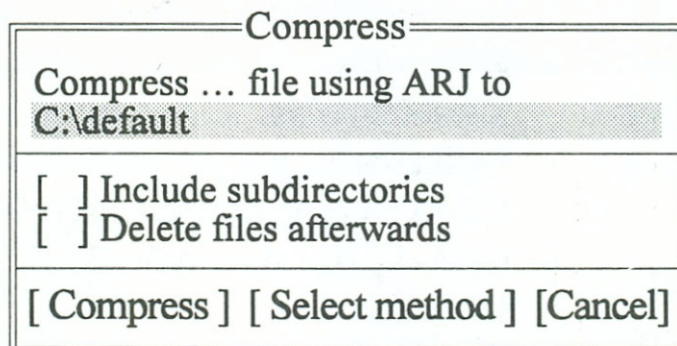
Path prompt (*pász prompt*)

A prompt tartalmazza-e az elérési utat vagy sem?

Key bar (<i>kí bár</i>)	A funkcióbillentyűk jelentése látszódjon-e a legelső sorban vagy sem?
Full screen (<i>ful szkrín</i>)	Teljes képernyőt használunk.
Mini status (<i>mini sztétjös</i>)	Az összesítő ablak látszódjon-e a panelek alján vagy sem?
Clock (<i>klok</i>)	A jobb felső sarokba óra kerül.
Save setup (<i>szév szetáp</i>)	Menthetjük az elvégzett beállításokat, így később már úgy jelenik meg a NORTON COMMANDER.

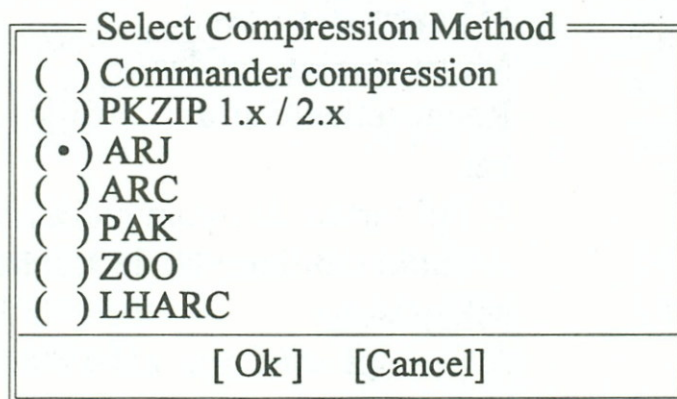
Fájlok, alkönyvtárak tömörítése

Az egyik ablakban a tömörítésre szánt állományok legyenek, a másik ablakban legyen az az alkönyvtár, ahová tömöríteni szeretnénk. Megnyomjuk az **Alt-F5** billentyűket, melyre az alábbi ablak jelenik meg:



A *default* nevet a szükségesre javítsuk. A két [] közé a **SPACE** billentyű lenyomásával tehetünk X-et, ha az alkönyvtárakat is tömöríteni akarjuk, vagy a tömörítés után le akarjuk törölni a tömörített állományokat.

A tömörítés módját egy újabb ablakban választhatjuk ki:



Tömörített állományok kicsomagolása

A NORTON egyik ablakában legyen a tömörített fájl, a másik ablakban pedig az az alkönyvtár, ahova ki szeretnénk kicsomagolni. A tömörített fájl nevére állva, majd az **Alt-F6** billentyűt megnyomva a kicsomagolás automatikusan megtörténik.

Norton Commander billentyűparancsok

F1	Segítség megjelenítése.
F2	Saját menü előhívása.
F3	Fájl tartalmának megtekintése.
F4	Szövegfájl szerkesztése.
F5	Másolás.
F6	Átmozgatás.
F7	Alkönyvtár készítése.
F8	Fájl, illetve könyvtár törlése.
F9	Menüsor aktiválása.
F10	Kilépés.
Alt-F1	Meghajtóváltás a bal panelen.
Alt-F2	Meghajtóváltás a jobb panelen.
Alt-F3	Fájl tartalmának megtekintése.
Alt-F4	Szövegfájl szerkesztése.
Alt-F5	Tömörítés.
Alt-F6	Kicsomagolás.
Alt-F7	Fájlkeresés.
Alt-F8	Előző DOS-parancsok aktiválása.
Alt-F9	EGA-VGA képernyőn 24 / 43 / 50 soros kijelzés beállítása.
Alt-F10	Gyors alkönyvtárelérés.
Ctrl-F1	Bal panel ki-bekapcsolása.
Ctrl-F2	Jobb panel ki-bekapcsolása.
Ctrl-F3	Névsorba rendezés.
Ctrl-F4	Kiterjesztés szerinti névsor.
Ctrl-F5	Idő szerinti sorrend.
Ctrl-F6	Méret szerinti rendezés.
Ctrl-F7	Rendezetlenül írja ki a fájlokat és az alkönyvtárakat.
Ctrl-F9	A fájl tartalmát nyomtatóra másolja.
Ctrl-B	A funkcióbillentyűk leírásának megjelenítését ki-bekapcsolja.
Ctrl-J	A prompt után írja azt az állománynevet, amin a kurzor áll.
Ctrl-L	A másik ablakba írja a lemezinformációkat (váltó).
Ctrl-O	Panel (ablak) ki-bekapcsolása.
Ctrl-P	Eltünteti a kurzorral ellentétes ablakot (váltó).
Ctrl-Q	Szövegfájl nevéen állva, a másik ablakba kiírja annak tartalmát (váltó).

Ctrl-U	Panel(ablak)cseré.
Shift-F9	Beállítások mentése.
Shift-F4	Új szövegfájl létrehozása.
CTRL -\	Hatására a meghajtó főkönyvtárába kerülünk.

NCEDIT parancsok

Kurzormozgató parancsok

→	Egy karakterrel jobbra	←	Egy karakterrel balra
Ctrl-→	Egy szóval jobbra	Ctrl-←	Egy szóval balra
↓	Egy sorral le	Ctrl-↑	Egy sorral fel
PgDown	Egy lappal le	PgUp	Egy lappal fel
Ctrl-Home	Szöveg elejére	Ctrl-End	Szöveg végére
Home	Sor elejére	End	Sor végére

Törlési parancsok

Del	Egy karaktert jobbra.	Backspace	Egy karaktert balra.
Ctrl-t	Egy szót jobbra.	Ctrl-w	Egy szót balra.
Ctrl-k	A kurzortól a sor végéig.	Ctrl-y	A teljes sort.

Fájlkezelési parancsok

F1	Segítségkérés.	F2	Mentés munka közben.
F9	Nyomtatás.	Shift-F2	Mentés új néven.
Alt-F5	Fájl beszúrása.	Shift-F10	Kilépés mentéssel.

Kilépés

ESC vagy **F10**

Blokk parancsok

F3	Blokk kezdete, vége.	Shift-F3	Kijelölés megszüntetése.
F5	Másolás a kurzorhoz.	F6	Mozgatás a kurzorhoz.
F8	Blokk törlése.	Alt-F10	Blokk kiírása lemezre.

Egyéb parancsok

Alt-F1	Windows-formátum.	Alt-F6	Sorszám beszúrása.
Alt-F2	ASCII-formátum.	Alt-F8	Ugrás sorszám szerint.
Alt-F3	Dátum beírása.		

A QUICK BASIC

A Quick BASIC szerkesztőparancsok

Ctrl + Q Y	Törlés a kurzortól a sor végéig.
Ctrl + Y	Kurzor sorának törlése.
End Enter	Egy sor beszúrása a kurzor alatti sor alá.
Home Ctrl+N	Egy sor beszúrása a kurzor alatti sor fölé.
Insert	Beszúró üzemmód ki-bekapcsolása.
Shift+Insert	Kijelölt blokk beszúrása a kurzor után.
Shift + →	Betűk kijelölése a kurzor után.
Shift + ←	Betűk kijelölése a kurzortól balra.
Shift + Delete	Kijelölt blokk törlése.
F1	Segítő (HELP) képernyő meghívása.
F2	Választás a szubrutinok között.
F3	FIND-funkciók szerinti keresés.
F4	Váltás a futó- és a szerkesztőképernyő között.
F5	Program futtatása a megszakítási ponttól.
F6	Áttérés a következő ablakra.
F7	Program végrehajtása a kurzortól kezdve.
F8	Lépésenkénti végrehajtás.
F9	Megszakítási pont elhelyezése.
F10	Lépésenkénti eljárásos végrehajtás.
Shift + F5	Program újraindítása.

A Quick BASIC menürendszere

File

New Program	A memóriában levő aktív programok törlése.
Open Program	A program betöltése.
Merge ...	A memóriában levő programhoz egy új program hozzákapcsolása.
Save	A memóriában levő program kimentése.
Save as ...	A program kimentése más névvel.
Save all	A programot csak akkor menti, ha abban változás történt.
Create File	Új fájl létrehozása.
Load File ...	Más kiterjesztésű fájl betöltése a memóriába.
Unload File ...	A memóriában levő programok törlése.

Print		Az aktív képernyő, blokk vagy az egész program nyomtatása.
DOS Shell		Kilépés a DOS-ba. (Visszatérni az EXIT parancs segítségével lehet.)
Exit		Kilépés a Quick Basicből.
Edit		
Undo	Alt+Backspace	A véletlen törlés után az eredeti visszaállítása.
Cut	Shift+Delete	A kijelölt blokk kivágása.
Copy	Ctrl+Insert	A kijelölt blokk tartalma a memóriába kerül.
Paste	Shift+Insert	A memóriában levő blokkot a kurzortól kezdve bemásoljuk.
Clear Delete		A kijelölt blokk törlése.
New SUB ...		A programhoz külön hozzáfűzhetünk alprogramokat, melyeket ebben a funkcióban kell megírni.
New FUNCTION		Olyan alprogramot készíthetünk, amely a főprogramból történő meghívása estén végrehajt egy értékadó műveletet, majd ezzel az értékkel visszatér a főprogramba.
Syntax Checking		Nyelvi ellenőrzés ki-bekapcsolása.
View		
SUBs	F2	Több SUB-funkcióval készített szubrutin közül választhatunk egyet törlésre, szerkesztésre, átmozgatásra vagy a második ablakban való megjelenítésre.
Next SUB	Shift+F2	A következő szubrutin megjelenítése.
Split		Osztott képernyő ki-bekapcsolása.
Next Statement		A következő modul vagy lap hívása. (Vissza: F2)
Output Screen	F4	Váltás a futó és a szerkesztőképernyő között.
Include File		A REM \$INCLUDE 'file' utasítással kijelölt fájl beszúrása a programba.
Include Lines		Az Include File menüben betöltött fájl sorai látszanak-e vagy sem?
Search		
Find ...		Keresendő szó, betű, kifejezés beállítása.
Selected Text	Ctrl+\	Kurzor alatti keresés.
Repeat last find	F3	További keresés.
Change ...		A keresett kifejezések adottra cserélése.
Label		Címkekeresés.

Run		
Start	Shift+F5	A program futtatása előlről.
Restart		A program megakasztása után futtatás a program elejéről.
Continue	F5	Futtatás a megakadás helyétől kezdve.
Modify COMMANDS		Az adott változó értékadása.
Make .EXE File ...		A .BAS fájlból futtatható .EXE fájlt állít elő.
Make Library		A .BAS kiterjesztésű fájlból .LIB kiterjesztésű fájlt állít elő.
Set Main Module		A memóriában tárolt több program közül az egyik kiválasztása szerkesztésre.
Debug		
Add Watch ...		Az általunk kijelölt változó értékének figyelése.
Watchpoint ...		Ha a figyelt változó kapott értéket, akkor TRUE, ha nem, akkor FALSE eredményt kapunk.
Delete Watch ...		Figyelés törlése.
Delete all Watch		Az összes változó figyelésének törlése.
Trace on		Nyomkövetés ki-bekapcsolása.
History on		Eseménykövetés ki-bekapcsolása.
Toggle Breakpoint	F9	Megszakítási pont elhelyezése a kurzor helyén.
Clear All Breakpoints		Az összes megszakítási pont törlése.
Set Next Statement^a		Megszakítási pont áthelyezése a kurzor alatti utasításra. A Continue esetében innen folytatódik a program futása.
Calls		
		Az aktuális program nevét tartalmazza.
Options		
Display		A képernyő színeit, a gördítősávokat, a tabulátorok beállításait módosíthatjuk.
Set Paths ...		A Quick BASIC használatához szükséges fájlok elérési útjait lehet módosítani.
Right Mouse		Az egér jobb gombjának megnyomására mi történjen. Két lehetőségünk van: <ul style="list-style-type: none"> – egy parancson megnyomva azonnal kiírja a segítséget; – végrehajtja az adott sort.
Syntax Checking		Kiválasztva: hiba esetén azonnal jelez. Formázza a sort. Végrehajthatóvá fordítja a hibamentes sort.
Full menus		A teljes menüsört használja.

A Quick BASIC utasításai

BEEP		hatására a hanggenerátor megszólaltat és 1/4 másodpercig kitart egy, 800 hertzes hangot.																																
BEEP	BEEP	Helyettesíthető a PRINT CHR\$(7) utasítással is.																																
CHDIR		Az adott alkönyvtárba lép a parancs végrehajtása után.																																
CHDIR	CHDIR "útvonal"	A megadott útvonal végén levő, megadott nevű alkönyvtárba lép.																																
CLEAR		Változók értékeinek nullázására használjuk.																																
CLEAR	CLEAR	Hatására minden változó tartalma nulla vagy üres lesz.																																
CLS (CLear Screen)		Képernyőtörlés.																																
CLS	CLS																																	
COLOR		A színek beállításának parancsa.																																
COLOR a,b,c	COLOR a,b,c																																	
	a	A kurzor színe (16-ot hozzáadva villogás).																																
	b	A háttér színe (ide írunk).																																
	c	A keret színe (nem szükséges beírni).																																
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 5%;">0</td> <td style="width: 25%;">fekete</td> <td style="width: 5%;">4</td> <td style="width: 25%;">piros</td> <td style="width: 5%;">8</td> <td style="width: 25%;">sötétszürke</td> <td style="width: 5%;">12</td> <td style="width: 25%;">világospiros</td> </tr> <tr> <td>1</td> <td>kék</td> <td>5</td> <td>bíbor</td> <td>9</td> <td>világoskék</td> <td>13</td> <td>világosbíbor</td> </tr> <tr> <td>2</td> <td>zöld</td> <td>6</td> <td>barna</td> <td>10</td> <td>világoszöld</td> <td>14</td> <td>sárga</td> </tr> <tr> <td>3</td> <td>ciánkék</td> <td>7</td> <td>világosszürke</td> <td>11</td> <td>világoscián</td> <td>15</td> <td>fehér</td> </tr> </table>	0	fekete	4	piros	8	sötétszürke	12	világospiros	1	kék	5	bíbor	9	világoskék	13	világosbíbor	2	zöld	6	barna	10	világoszöld	14	sárga	3	ciánkék	7	világosszürke	11	világoscián	15	fehér		
0	fekete	4	piros	8	sötétszürke	12	világospiros																											
1	kék	5	bíbor	9	világoskék	13	világosbíbor																											
2	zöld	6	barna	10	világoszöld	14	sárga																											
3	ciánkék	7	világosszürke	11	világoscián	15	fehér																											
CONST		Hatására ezek a nevek állandó értéket kapnak, melyet a programban használhatunk fel.																																
CONST név = kifejezés	CONST név = kifejezés	Például: CONST szam = 5; CONST honap\$ = "Október"																																
CSRLIN		A változó értéke a kurzor sorértéke lesz.																																
adat=CSRLIN	adat=CSRLIN	Például: SOR=CSRLIN hatására a SOR változó a kurzor pillanatnyi helyének értékét kapja.																																
DATA		Adatok tárolása programban.																																
DATA adat,adat,adat, ...	DATA adat,adat,adat, ...																																	

DATES		A dátum értékének beállítása.
	DATES = "HH-NN-ÉÉ"	Ahol H a hónapot, N a napot, az É az évet jelenti.
DIM		Az adott néven a megadott tartományban tömböt foglalhatunk le a memóriában.
	DIM név(végérték)	Például: DIM A(15) hatására 0..15 sorszámmú tömböt foglal le, amely számokat kaphat értékül. Pl: A(1)=8; A(2)=-3.14; Maximum 60 tömböt definiálhatunk.
DO LOOP UNTIL		Az utasításokat addig hajtja végre, amíg az adott feltétel IGAZZá nem válik.
	DO UNTIL logikai kifejezés	
	utasítások	
	LOOP	
		vagy
	DO	
	utasítások	
	LOOP UNTIL logikai kifejezés	
DO WHILE LOOP		Az utasításokat addig hajtja végre, amíg az adott feltétel HAMISSá nem válik.
	DO WHILE logikai kifejezés	
	utasítások	
	LOOP	
		vagy
	DO	
	utasítások	
	LOOP WHILE logikai kifejezés	
END		A program befejezése, megszakítása.
FILES		A fájlok neveinek megjelenítése.
	FILES	Az aktuális alkönyvtár tartalomjegyzéke.
	FILES "C:\QBASIC\PELDA*.BAS"	
FOR ... TO ... NEXT ...STEP		Ismétlés előre megadott lépésszámmal.
	FOR ciklusváltozó=mettől TO meddig [STEP lépésköz]	
	ez a rész fog ismétlődni	
	NEXT [ciklusváltozó]	
GOTO		Feltétel nélküli vezérlésátadás
	GOTO címke:	A programot a címke: feliratú sornál folytatja

IF ... THEN ... ELSE	feltételes vezérlésátadás
IF logikai kifejezés	
THEN Igaz esetben végrehajtandó utasítások	
[ELSE hamis esetben végrehajtandó utasítások]	
END IF	
INPUT	adatbevitel
INPUT ["szöveg";]változónév1[,változónév2, ...]	
"szöveg";	Tájékoztató jellegű szöveg például arról, hogy milyen adatot kérünk. Ugyanúgy megjelenik a képernyőn, mint a PRINT utasításban. Elhagyható! Ha szöveget írunk, a pontosvesszőt kötelező használni. Hatására a szöveg után nincs soremelés.
változónév	A tanult változónév-lehetőségek kerülhetnek ide. Egymás után több változó is megadható, de ebben az esetben vesszővel kell elválasztani őket egymástól.
KEY	A funkcióbillentyűk használatához beállított események figyelésének be-, illetve kikapcsolása.
KEY(n) ON	Bekapcsolja az n. funkcióbillentyűt.
KEY(n) OFF	Kikapcsolja az n. funkcióbillentyűt.
KEY(n) STOP	Felfüggeszti az n. funkcióbillentyű figyelését.
n lehetséges értékei:	
1...10	F1..F10
11	↑
12	←
13	→
14	↓
15 - 25	Definiált billentyűk;
30	F11
31	F12
KEY n, kifejezés	Az n. funkcióbillentyű átdefiniálásához használhatjuk. Például:
KEY 1,"FILES"	
KILL	Fájl törlésére használjuk.
KILL "C:\QBASIC\PELDA\proba.bas"	törli a proba.bas programot a megadott alkönyvtárból.

<p>LOCATE</p> <p style="padding-left: 2em;">LOCATE a,b</p> <p style="padding-left: 4em;">a</p> <p style="padding-left: 4em;">b</p>	<p>Kurzor adott helyre állítása</p> <p>A sor értéke 1-től 23-ig (a teljes képernyőn KEY OFF esetén 25 sor van).</p> <p>Az oszlop értéke 1-től 80-ig (a WIDTH parancs segítségével ez 40-re is állítható).</p>
<p>LPRINT</p> <p style="padding-left: 2em;">LPRINT változólista</p>	<p>A nyomtatóra írja a változólistában adottakat.</p> <p>Használata megegyezik a PRINT parancsnál leírtakkal.</p>
<p>MKDIR</p> <p style="padding-left: 2em;">MKDIR "útvonal"</p>	<p>Létrehozza az adott alkönyvtárat.</p> <p>Létrehozza az útvonal végén lévő alkönyvtárat.</p>
<p>PCOPY</p> <p style="padding-left: 2em;">PCOPY oldal1 oldal2</p>	<p>Az adott képernyő másolása.</p> <p>Az oldal1 és oldal2 olyan numerikus érték, amellyel meghatározhatjuk, melyik oldalt (OLDAL1) melyik oldalra (OLDAL2) kívánjuk másolni. Értékét a monitorvezérlő kártya határozza meg.</p> <p>Például:</p> <pre> PRINT "Ez a másolandó oldal: 2+3="; 2 + 3 PCOPY 0, 3 DO XS = INKEYS LOOP UNTIL XS = " " CLS DO XS = INKEYS LOOP UNTIL XS = " " PRINT "Most másoltuk vissza !" PCOPY 3, 0 </pre>
<p>POS</p> <p style="padding-left: 2em;">POS(kifejezés)</p>	<p>Annak az oszlopnak a számát adja, amelyikben a kurzor áll.</p> <p>A kifejezés bármennyi lehet.</p>
<p>PLAY</p> <p style="padding-left: 2em;">PLAY szövegváltozó</p> <p style="padding-left: 4em;">On</p> <p style="padding-left: 4em;">+ vagy -</p>	<p>Dallam lejátszása.</p> <p>Használható jelölések:</p> <p>Oktáv jelölése ($0 \leq n \leq 6$) Pl: O3 a 3.oktáv</p> <p>C...B betűk mögé írva fél hanggal följebb vagy lejjebb szólaltathatjuk meg a hangot.</p> <p>Pl: D#-nek megfelel a D+ (az E+ és B+</p>

	nem megengedett, mert az E és az F hangok között, valamint B és C hangok között csak félhangnyi távolság van).
<n vagy >n	Átválthatunk az aktuálisnál eggyel alacsonyabb vagy magasabb oktávba.
Nn	Ahol $0 \leq n \leq 84$ paraméterrel hét oktáv terjedelemben egy adott oktáv egy adott hangjegyét szólaltathatjuk meg. Az $n = 0$ a szünet jele.
Ln	Ahol $1 \leq n \leq 64$ segítségével meghatározhatjuk a soron következő hang időtartamát. $n = 1$: egész hang; $n = 2$: félhang; $n = 4$: negyedhang és így tovább.
Pn	Ahol $1 \leq n \leq 64$ segítségével adott tartamú szünetet jelölhetünk.
Tn	Az $1 \leq n \leq 255$ az egy perc alatt megszólaló negyedhangok számát jelöli. Alapértelmezése T120.
MB	Lejátssza a SOUND és a PLAY utasításokkal lekottázott dallamot, de közben a program megszakítás nélkül tovább fut. Egyszerre legfeljebb 32 hangjegyet tud egymás után lejátszani.
MF	Felfüggeszti a program futását, amíg a BASIC az utolsó hang- vagy szünetjelet fel nem dolgozta.
ML	Minden hangot egymáshoz kötve az Ln által megszabott ideig tart ki.
MN	Normál időtartam; a hangot az idő 7/8-án tartja ki. Ez az alapértelmezés.
MS	A hangot az idő 3/4-én tartja ki.
X+VARPTR\$(string)	A rendszer feldolgozza a <i>string</i> ben megadott karakterláncot.
PLAY ON	Ezzel kezdjük el lejátszani a zenét.
PLAY OFF	Megszakítjuk a zene lejátszását.
PLAY STOP	Felfüggeszti a zene lejátszását.
PRINT	A képernyőre írja a változólistában adott értékeket.
PRINT változólista változólista	Ide kell beírni a kiíratni szánt adatokat, illetve szöveget.

PRINT USING	Az adatok formázott kiírása a képernyőre.
PRINT USING forma; változólista	
forma	
!	Adatlánc első karakterének megjelenítése.
\ n \	2+n db szóköz megjelenítése.
&	Változó hosszúságú adat megjelenítése.
#	Számjegyek helyének meghatározása.
.	Tizedespont helyének meghatározása.
+	Előjel megjelenítése.
**	Értékes jegyek előtti karakterek feltöltése csillaggal.
\$\$	A szám előtt egy \$ jel megjelenítése, számok nagyságrendi bontása.
változólista	Itt kell megadni a kiíratni szánt adatokat. Például:
A=43324.42 : PRINT USING "+###,###.#" A	parancs hatására a képernyőn +43,324.4 jelenik meg.
RANDOMIZE TIMER	A véletlenszámok előállítása a gép órajeletől függ.
RANDOMIZE TIMER	
READ	A DATA-sorokban tárolt adatok olvasása.
READ változó[,változó[,...]]	
REM	Magyarázó szöveg elhelyezése a programban.
REM szöveg	
RESTORE	A DATA-sorokban tárolt adatok adatmutatójának visszaállítása.
RESTORE [címke]	Címke nélkül a legelső DATA-sorra, címkével az adott címkéjű DATA-sorra állítja a mutatót.
RMDIR	Alkönyvtár törlése, ha az üres.
RMDIR "útvonal"	A megadott útvonal végén levő alkönyvtárat törli.
SHELL	DOS-parancs végrehajtása.
SHELL	Kilép a DOS-ba, ahol végrehajtható a szükséges parancs. Visszatérni az EXIT paranccsal lehet.
SHELL "DOS_parancs"	Végrehajtja a DOS-parancsot, majd visszatér a BASICbe.

SELECT CASE	Elágazás többfelé.
SELECT CASE kifejezés	
Kifejezés1 Utasítás1	
.	
.	
.	
Kifejezés N Utasítás N	
END SELECT	
SLEEP	Felfüggeszti a program futását.
SLEEP [másodperc]	
másodperc	A felfüggesztés időtartama másodpercben.
SOUND	Adott frekvenciájú hang előállítása
SOUND frekvencia,időtartam	
frekvencia	37 hertztől 32767 hertzig;
idő tartam	alapegysége 1/192 perc. Egy másodperc kb. 18.2 egység.
STOP	Megállítja a program végrehajtását.
STOP	
SWAP	Felcseréli a változók tartalmát.
SWAP változó1 , változó2	A két változó értékeit felcseréli.
TIMES	Az aktuális rendszeridő beállítása.
TIMES="ÓÓ:PP:SS"	Ahol az ÓÓ az óra, a PP a perc, az SS a másodperc értéke.
TROFF	Nyomkövetés kikapcsolása.
TROFF	Az előzőleg kiadott TRON utasítást hatástalanítjuk.
TRON	Nyomkövetés bekapcsolása.
TRON	Bekapcsoljuk a nyomkövetést. Elsősorban hibakeresésnél használjuk. Az éppen végrehajtás alatt álló sor világos csíkba kerül.

A Quick BASIC függvényei

Aritmetikai függvények

ABS(X)	Az X abszolútértéke.
CBDL(X)	Az X értéke 15 jegyűre konvertálódik. $-32\,768 \leq X \leq 32\,767$
CINT(X)	Az X egész számmá kerekített értéke.
FIX(X)	X tizedesjegyeinek levágásával kapott szám.
HEX\$(X)	X decimális szám hexadecimális értéke.

INT(X)	Az X egészrésze.
OKT\$(X)	Az X decimális szám oktális alakja.
RND(X)	Véletlenszám.
Általános szövegkezelő függvények	
ASC(X\$)	Az X\$ első karakterének kódja.
CHR\$(x)	Az X kódhoz tartozó karakter.
INSTR([n,] A\$,B\$)	A\$ hányadik karaktertől tartalmazza aB\$-t (0, ha nem tartalmazza). Ha n-t is megadjuk, akkor csak az n. karaktertől kezdi a vizsgálatot.
LCASE\$(X\$)	Az X\$-t csupa kis betűssé alakítja.
LEFT\$(X\$,Y)	Az X\$ bal oldali Y darab karaktere.
LEN(X\$)	Az X\$ hossza.
LTRIM\$(X\$)	Levágja az X\$ bal oldalán álló szóközeit.
MID\$(X\$,Y,Z)	Az X\$ Z darab karaktere az Y karakterhelytől jobbra.
RIGHT\$(X\$,Y)	Az X\$ jobb oldali Y darab karaktere.
RTRIM\$(X\$)	Az X\$ végéről levágja a szóközöket.
SPACE\$(X)	X darab szóközből álló lánc.
STRING\$(X,Y)	Y kódú karakterből álló X hosszú lánc.
STRING\$(Y,X\$)	Az X\$ első karakteréből álló Y hosszú lánc.
UCASE\$(X\$)	Az X\$-t csupa nagy betűssé alakítja.
VAL(X\$)	Az X\$ numerikus értéke.
Egyéb karakteres függvények	
DATE\$	A rendszer dátumát adja.
STR\$(X)	Az X értékét tartalmazó karakterlánc.
TIME\$	A rendszeridő értéke.
Szöveg beolvasást és kiírást támogató függvények	
CSRLIN	A kurzor sorát adja.
INKEY\$	Egy karakter beolvasása a billentyűzetről.
POS(N)	A kurzor oszlopát adja.
SPACE\$(N)	N darab szóközből álló karaktorsorozatot állít elő.
SPC(X)	X darab szóköz kiírása a PRINT illetve az LPRINT végrehajtása közben.
TAB(X)	Ugrás az X tabulátorhelyzetbe a PRINT illetve az LPRINT esetében.

MINTAPROGRAMOK

1. Képernyőkeretek rajzolása.

```
COLOR 12, 0, 14 : CLS
```

```
a = 0: k = 14: CLS : GOTO bev
```

```
bev:
```

```
y = 2: x = 2: z = 20: GOSUB ker1
```

```
COLOR 10, a, k
```

```
LOCATE 4, 9: PRINT "A képernyőn olyan keretet akarunk rajzoltatni, amely a függőleges"
```

```
LOCATE 6, 9: PRINT "középvonalra szimmetrikus. Meg kell adni bal felső sarkának"
```

```
LOCATE 8, 9: PRINT "sorát és oszlopát, valamint függőleges méretét (y,x,z). A színét"
```

```
LOCATE 10, 9: PRINT "külön utasítással határozzuk meg. A 'GOSUB KER1' utasításra a "
```

```
LOCATE 12, 9: PRINT "keret kirajzolódik. (A színeket meg kell adni.)"
```

```
LOCATE 14, 9: PRINT "Ha a keret nem szimmetrikusan helyezkedik el, akkor bal felső "
```

```
LOCATE 16, 9: PRINT "sarkának sorát-oszlopát és jobb alsó sarkának sorát-oszlopát kell"
```

```
LOCATE 18, 9: PRINT "megadni (y,x,y1,x1). A 'GOSUB KER2' utasításra rajzolódik ki."
```

```
LOCATE 20, 9: PRINT "A KER3,KER4,KER5,KER6 más kereteket rajzol.(adatok mint a KER2-ben)"
```

```
GOSUB all
```

```
CLS
```

```
ker:
```

```
y = 2: x = 12: z = 4
```

```
GOSUB ker1
```

```
COLOR 15, a, k
```

```
LOCATE 4, 21: PRINT "Ide kerülhet például a programunk címe."
```

```
COLOR 7, a, k
```

```
LOCATE 4, 2: PRINT "ker1"
```

```
COLOR 5, a, k: y = 7: x = 10: y1 = 11: x1 = 52
```

```
GOSUB ker2
```

```
COLOR 14, a, k: LOCATE 9, 15: PRINT "A keret nemcsak középre kerülhet."
```

```
COLOR 7, a, k
```

```
LOCATE 9, 2: PRINT "ker2"
```

```

COLOR 6, a, k: y = 7: x = 56: y1 = 11: x1 = 70
GOSUB ker2
COLOR 10, a, k
y = 13: x = 12: x1 = 35: y1 = 17: GOSUB ker3
COLOR 7, a, k
LOCATE 15, 2: PRINT "ker3"
COLOR 14, a, k
LOCATE 15, 17: PRINT "Másfajta keret."
COLOR 12, a, k
y = 13: x = 43: x1 = 68: y1 = 19: GOSUB ker4
COLOR 15, 11, k
LOCATE 16, 47: PRINT "Kiemelt színes mező"
COLOR 7, a, k
LOCATE 16, 76: PRINT "ker4"
COLOR 14, a, k
y = 19: x = 10: y1 = 23: x1 = 38: GOSUB ker5
COLOR 12, a, k
LOCATE 21, 17: PRINT "Egyszerű keret"
COLOR 7, a, k: LOCATE 21, 2: PRINT "ker5"
COLOR 11, a, k
y = 21: x = 42: y1 = 23: x1 = 70: GOSUB ker6
COLOR 10, a, k: LOCATE 22, 47: PRINT "Legegyszerűbb keret"
COLOR 7, a, k: LOCATE 22, 76: PRINT "ker6"
GOSUB all
END
ker1:
t1 = 40 - x: x1 = 40 + t1: v = (2 * t1) - 1: y1 = y + z: m = z - 1
LOCATE y, x: PRINT CHR$(201)
LOCATE y, x1: PRINT CHR$(187)
FOR i = 1 TO v: LOCATE y, x + i: PRINT CHR$(205): NEXT
LOCATE y1, x: PRINT CHR$(200)
LOCATE y1, x1: PRINT CHR$(188)
FOR i = 1 TO v: LOCATE y1, x + i: PRINT CHR$(205): NEXT
FOR i = 1 TO m: LOCATE y + i, x: PRINT CHR$(186): NEXT
FOR i = 1 TO m: LOCATE y + i, x1: PRINT CHR$(186): NEXT
RETURN
GOTO ker
ker2:
v = x1 - x - 1: m = y1 - y - 1
LOCATE y, x: PRINT CHR$(201)
LOCATE y, x1: PRINT CHR$(187)

```

```

FOR i = 1 TO v: LOCATE y, x + i: PRINT CHR$(205): NEXT
LOCATE y1, x: PRINT CHR$(200)
LOCATE y1, x1: PRINT CHR$(188)
FOR i = 1 TO v: LOCATE y1, x + i: PRINT CHR$(205): NEXT
FOR i = 1 TO m: LOCATE y + i, x: PRINT CHR$(186): NEXT
FOR i = 1 TO m: LOCATE y + i, x1: PRINT CHR$(186): NEXT
RETURN

```

ker3:

```

v = x1 - x: m = y1 - y
FOR i = 1 TO v: LOCATE y, x + i: PRINT CHR$(178): NEXT
FOR i = 1 TO v: LOCATE y + m, x + i: PRINT CHR$(178): NEXT
FOR i = 0 TO m: LOCATE y + i, x: PRINT CHR$(178): NEXT
FOR i = 0 TO m: LOCATE y + i, x1: PRINT CHR$(178): NEXT
RETURN

```

ker4:

```

v = x1 - x: m = y1 - y
FOR i = 1 TO v: LOCATE y, x + i: PRINT CHR$(219): NEXT
FOR i = 1 TO v: LOCATE y + m, x + i: PRINT CHR$(219): NEXT
FOR i = 0 TO m: LOCATE y + i, x: PRINT CHR$(219): NEXT
FOR i = 0 TO m: LOCATE y + i, x1: PRINT CHR$(219): NEXT
COLOR 4, a, k
FOR i = 0 TO m: LOCATE y + 1 + i, x1 + 1: PRINT CHR$(219): NEXT
FOR i = 0 TO v: LOCATE y + m + 1, x + 2 + i: PRINT CHR$(219): NEXT
FOR i = 0 TO m: LOCATE y + i + 1, x1 + 2: PRINT CHR$(219): NEXT
COLOR 3, 15, k
FOR i = 1 TO v - 1: FOR j = 1 TO m - 1: LOCATE y + j, x + i: PRINT
CHR$(219): NEXT: NEXT
RETURN

```

ker5:

```

v = x1 - x: m = y1 - y
FOR i = 1 TO v - 1: LOCATE y, x + i: PRINT CHR$(220): NEXT
FOR i = 1 TO v - 1: LOCATE y + m, x + i: PRINT CHR$(223): NEXT
FOR i = 1 TO m - 1: LOCATE y + i, x + 1: PRINT CHR$(219): NEXT
FOR i = 1 TO m - 1: LOCATE y + i, x1 - 1: PRINT CHR$(219): NEXT
RETURN

```

ker6:

```

v = x1 - x - 1: m = y1 - y - 1
LOCATE y, x: PRINT CHR$(218)
LOCATE y, x1: PRINT CHR$(191)
FOR i = 1 TO v: LOCATE y, x + i: PRINT CHR$(196): NEXT
LOCATE y1, x: PRINT CHR$(192)

```

```
LOCATE y1, x1: PRINT CHR$(217)
FOR i = 1 TO v: LOCATE y1, x + i: PRINT CHR$(196): NEXT
FOR i = 1 TO m: LOCATE y + i, x: PRINT CHR$(179): NEXT
FOR i = 1 TO m: LOCATE y + i, x1: PRINT CHR$(179): NEXT
RETURN
```

all:

```
a$ = INKEY$
IF a$ = "" THEN GOTO all
RETURN
```

2. Kör és négyzet kerülete, területe.

```
COLOR 12, 1, 11
```

```
CLS
```

```
pi = 3.14159
```

```
y = 2: x = 2: z = 20: GOSUB keret
```

```
LOCATE 12, 2: PRINT CHR$(199): LOCATE 12, 79: PRINT CHR$(182)
```

```
FOR i = 1 TO 76: LOCATE 12, 2 + i: PRINT CHR$(196): NEXT
```

```
COLOR 14, 1, 11
```

```
LOCATE 3, 8: PRINT "Legyen adott egy kör és egy négyzet. Tételezzük fel, hogy  
vagy"
```

```
LOCATE 5, 18: PRINT "1) kerületeik egyenlőek, vagy"
```

```
LOCATE 7, 18: PRINT "2) területeik egyenlőek."
```

```
LOCATE 9, 8: PRINT "Adjuk meg a közös kerületet, illetve területet. Számítsuk ki  
ebből a"
```

```
LOCATE 11, 8: PRINT "négyzetoldalt, a körsugarat és a hiányzó kerületet, vagy  
területet."
```

```
COLOR 15, 1, 11
```

ujra:

```
COLOR 11, 1, 11
```

```
LOCATE 14, 8: PRINT "Mit választ: közös kerület (1), közös terület (2):"
```

```
LOCATE 14, 66: PRINT "(0=vége)": LOCATE 14, 60: INPUT v
```

```
IF v = 1 THEN GOTO ker
```

```
IF v = 2 THEN GOTO ter
```

```
IF v = 0 THEN END
```

ker:

```
LOCATE 16, 8: INPUT "Kérem a közös kerület hosszát: "; k
```

```
r = k / (2 * pi): o = k / 4: kt = r ^ 2 * pi: nt = o ^ 2
```

```
COLOR 15, 1, 11
```

```
LOCATE 18, 8: PRINT "A négyzet oldala ="; o; "a négyzet területe ="; nt
```

```
LOCATE 20, 8: PRINT "A kör sugara ="; r; ", a kör területe ="; kt
```

```
GOSUB all
```

```

CLS : GOTO ujra
ter:
LOCATE 16, 8: INPUT "Kérem a közös terület nagyságát:"; t
r = SQR(t / pi): o = SQR(t): kk = 2 * r * pi: kn = 4 * o
COLOR 15, 1, 11
LOCATE 18, 8: PRINT "A négyzet oldala ="; o; "a négyzet kerülete ="; kn
LOCATE 20, 8: PRINT "A kör sugara ="; r; ", a kör kerülete ="; kk
GOSUB all
CLS : GOTO ujra
END
all:
a$ = INKEY$
IF a$ = "" THEN GOTO all
RETURN
keret:
t = 40 - x: x1 = 40 + t: v = x1 - x: m = z: y1 = y + m + 1
LOCATE y, x: PRINT CHR$(201)
LOCATE y, x1 + 1: PRINT CHR$(187)
FOR i = 1 TO v: LOCATE y, x + i: PRINT CHR$(205): NEXT
LOCATE y1, x: PRINT CHR$(200)
LOCATE y1, x1 + 1: PRINT CHR$(188)
FOR i = 1 TO v: LOCATE y1, x + i: PRINT CHR$(205): NEXT
FOR i = 1 TO m: LOCATE y + i, x: PRINT CHR$(186): NEXT
FOR i = 1 TO m: LOCATE y + i, x1 + 1: PRINT CHR$(186): NEXT
RETURN

```

3. Szám értékének kiírása a képernyőre

```

bev:
COLOR 15, 1, 14
CLS
LOCATE 3, 19: PRINT CHR$(201): LOCATE 3, 60: PRINT CHR$(187)
LOCATE 7, 19: PRINT CHR$(200): LOCATE 7, 60: PRINT CHR$(188)
FOR i = 1 TO 3: LOCATE 3 + i, 19: PRINT CHR$(186): NEXT
FOR i = 1 TO 3: LOCATE 3 + i, 60: PRINT CHR$(186): NEXT
FOR i = 1 TO 40: LOCATE 3, 19 + i: PRINT CHR$(205): NEXT
FOR i = 1 TO 40: LOCATE 7, 19 + i: PRINT CHR$(205): NEXT
LOCATE 9, 23: PRINT "A szám legfeljebb 6 jegyű lehet!"
LOCATE 10, 26: PRINT "Ha 0-t ír, a program megáll."
LOCATE 5, 25: INPUT "Kérem a kiírandó számot:"; x
x$ = STR$(x): j = LEN(x$)
IF x = 0 THEN END

```



```

IF j = 7 THEN GOTO hat
IF j = 6 THEN GOTO ot
IF j = 5 THEN GOTO negy
IF j = 4 THEN GOTO hara
IF j = 3 THEN GOTO kete
IF j = 2 THEN GOTO egye

```

hat:

```

f$ = LEFT$(x$, 2): f = VAL(f$)
e$ = MID$(x$, 3, 1): e = VAL(e$)
d$ = MID$(x$, 4, 1): d = VAL(d$)
c$ = MID$(x$, 5, 1): c = VAL(c$)
b$ = MID$(x$, 6, 1): b = VAL(b$)
a$ = MID$(x$, 7, 1): a = VAL(a$)

```

GOTO hatnn

ot:

```

e$ = LEFT$(x$, 2): e = VAL(e$)
d$ = MID$(x$, 3, 1): d = VAL(d$)
c$ = MID$(x$, 4, 1): c = VAL(c$)
b$ = MID$(x$, 5, 1): b = VAL(b$)
a$ = MID$(x$, 6, 1): a = VAL(a$)

```

GOTO otnn

negy:

```

d$ = LEFT$(x$, 2): d = VAL(d$)
c$ = MID$(x$, 3, 1): c = VAL(c$)
b$ = MID$(x$, 4, 1): b = VAL(b$)
a$ = MID$(x$, 5, 1): a = VAL(a$)

```

GOTO negynn

hara:

```

c$ = LEFT$(x$, 2): c = VAL(c$)
b$ = MID$(x$, 3, 1): b = VAL(b$)
a$ = MID$(x$, 4, 1): a = VAL(a$)

```

GOTO harnn

kete:

```

b$ = MID$(x$, 2, 1): b = VAL(b$)
a$ = MID$(x$, 3, 1): a = VAL(a$)

```

GOTO ketnn

egye:

```

a$ = x$: a = VAL(a$)

```

GOTO egyenn

hatnn:

```

IF f = 1 THEN s6$ = "száz"

```

```
IF (f = 1 AND e = 0 AND d = 0) THEN s6$ = "százezer": GOTO harnn
IF (f = 1 AND e = 0) THEN GOTO negynn
IF f = 2 THEN s6$ = "kétszáz"
IF (f = 2 AND e = 0 AND d = 0) THEN s6$ = "kétszázezer": GOTO harnn
IF (f = 2 AND e = 0) THEN GOTO negynn
IF f = 3 THEN s6$ = "háromszáz"
IF (f = 3 AND e = 0 AND d = 0) THEN s6$ = "háromszázezer": GOTO harnn
IF (f = 3 AND e = 0) THEN GOTO negynn
IF f = 4 THEN s6$ = "négy száz"
IF (f = 4 AND e = 0 AND d = 0) THEN s6$ = "négy százezer": GOTO harnn
IF (f = 4 AND e = 0) THEN GOTO negynn
IF f = 5 THEN s6$ = "ötszáz"
IF (f = 5 AND e = 0 AND d = 0) THEN s6$ = "ötszázezer": GOTO harnn
IF (f = 5 AND e = 0) THEN GOTO negynn
IF f = 6 THEN s6$ = "hatszáz"
IF (f = 6 AND e = 0 AND d = 0) THEN s6$ = "hatszázezer": GOTO harnn
IF (f = 6 AND e = 0) THEN GOTO negynn
IF f = 7 THEN s6$ = "hétszáz"
IF (f = 7 AND e = 0 AND d = 0) THEN s6$ = "hétszázezer": GOTO harnn
IF (f = 7 AND e = 0) THEN GOTO negynn
IF f = 8 THEN s6$ = "nyolcszáz"
IF (f = 8 AND e = 0 AND d = 0) THEN s6$ = "nyolcszázezer": GOTO harnn
IF (f = 8 AND e = 0) THEN GOTO negynn
IF f = 9 THEN s6$ = "kilencszáz"
IF (f = 9 AND e = 0 AND d = 0) THEN s6$ = "kilencszázezer": GOTO harnn
IF (f = 9 AND e = 0) THEN GOTO negynn
GOTO otnn
```

otnn:

```
IF (e = 1 AND d > 0) THEN s5$ = "tizen"
IF (e = 1 AND d = 0) THEN s5$ = "tízezer": GOTO harnn
IF (e = 2 AND d > 0) THEN s5$ = "huszon"
IF (e = 2 AND d = 0) THEN s5$ = "húszezer": GOTO harnn
IF e = 3 THEN s5$ = "harminc"
IF (e = 3 AND d = 0) THEN s5$ = "harmincezer": GOTO harnn
IF e = 4 THEN s5$ = "negyven"
IF (e = 4 AND d = 0) THEN s5$ = "negyvenezer": GOTO harnn
IF e = 5 THEN s5$ = "ötven"
IF (e = 5 AND d = 0) THEN s5$ = "ötvenezer": GOTO harnn
IF e = 6 THEN s5$ = "hatvan"
IF (e = 6 AND d = 0) THEN s5$ = "hatvanezer": GOTO harnn
IF e = 7 THEN s5$ = "hetven"
```

```

IF (e = 7 AND d = 0) THEN s5$ = "hetvenezer": GOTO harmn
IF e = 8 THEN s5$ = "nyolcvan"
IF (e = 8 AND d = 0) THEN s5$ = "nyolcvanezer": GOTO harmn
IF e = 9 THEN s5$ = "kilencven"
IF (e = 9 AND d = 0) THEN s5$ = "kilencvenezer": GOTO harmn
GOTO negynn

```

negynn:

```

IF d = 1 THEN s4$ = "egyezer"
IF d = 2 THEN s4$ = "kétezer"
IF d = 3 THEN s4$ = "háromezer"
IF d = 4 THEN s4$ = "négyezer"
IF d = 5 THEN s4$ = "ötezer"
IF d = 6 THEN s4$ = "hatezer"
IF d = 7 THEN s4$ = "hétezer"
IF d = 8 THEN s4$ = "nyolcezer"
IF d = 9 THEN s4$ = "kilencezer"
IF d = 0 THEN GOTO ketnn
GOTO harmn

```

harmn:

```

IF c = 1 THEN s3$ = "száz"
IF c = 2 THEN s3$ = "kétszáz"
IF c = 3 THEN s3$ = "háromszáz"
IF c = 4 THEN s3$ = "négy száz"
IF c = 5 THEN s3$ = "ötszáz"
IF c = 6 THEN s3$ = "hatszáz"
IF c = 7 THEN s3$ = "hétszáz"
IF c = 8 THEN s3$ = "nyolcszáz"
IF c = 9 THEN s3$ = "kilencszáz"
IF c = 0 THEN GOTO ketnn
GOTO ketnn

```

ketnn:

```

IF (b = 1 AND a > 0) THEN s2$ = "tizen"
IF (b = 1 AND a = 0) THEN s2$ = "tíz"
IF (b = 2 AND a > 0) THEN s2$ = "huszon"
IF (b = 2 AND a = 0) THEN s2$ = "húsz"
IF b = 3 THEN s2$ = "harminc"
IF b = 4 THEN s2$ = "negyven"
IF b = 5 THEN s2$ = "ötven"
IF b = 6 THEN s2$ = "hatvan"
IF b = 7 THEN s2$ = "hetven"
IF b = 8 THEN s2$ = "nyolcvan"

```

```
IF b = 9 THEN s2$ = "kilencven"
GOTO egynn
egynn:
IF a = 1 THEN s1$ = "egy"
IF a = 2 THEN s1$ = "kettő"
IF a = 3 THEN s1$ = "három"
IF a = 4 THEN s1$ = "négy"
IF a = 5 THEN s1$ = "öt"
IF a = 6 THEN s1$ = "hat"
IF a = 7 THEN s1$ = "hét"
IF a = 8 THEN s1$ = "nyolc"
IF a = 9 THEN s1$ = "kilenc"
GOTO felir
felir:
COLOR 15, 4, 14
FOR h = 1 TO 3: FOR f = 0 TO 79: LOCATE 11 + h, 1 + f: PRINT " ": NEXT:
NEXT
GOTO feliro
feliro:
IF j = 7 THEN GOTO felir6
IF j = 6 THEN GOTO felir5
IF j = 5 THEN GOTO felir4
IF j = 4 THEN GOTO felir3
IF j = 3 THEN GOTO felir2
IF j = 2 THEN GOTO felir1
felir6:
LOCATE 13, 6: PRINT "A szám betűkkel :"; s6$; s5$; s4$; "-"; s3$; s2$; s1$
x = 0
END
felir5:
LOCATE 13, 10: PRINT "A szám betűkkel kiírva :"; s5$; s4$; "-"; s3$; s2$; s1$
x = 0
GOSUB all: GOTO bev
felir4:
LOCATE 13, 15: PRINT "A szám betűkkel kiírva :"; s4$; "-"; s3$; s2$; s1$
x = 0
END
felir3:
LOCATE 13, 18: PRINT "A szám betűkkel kiírva :"; s3$; s2$; s1$
x = 0
END
```

felir2:

LOCATE 13, 21: PRINT "A szám betűkkel kiírva :"; s2\$; s1\$

x = 0

END

felir1:

LOCATE 13, 24: PRINT "A szám betűkkel kiírva :"; s1\$

x = 0

END

all:

a\$ = INKEY\$

IF a\$ = "" THEN GOTO all

RETURN

4. Számok egész osztóinak kiírása

prog:

COLOR 5, 0, 6

CLS

LOCATE 2, 9: FOR i = 1 TO 62: PRINT CHR\$(220); : NEXT

FOR j = 0 TO 1

FOR i = 1 TO 9: LOCATE 2 + i, 10 - j: PRINT CHR\$(219): NEXT

FOR i = 1 TO 9: LOCATE 2 + i, 69 + j: PRINT CHR\$(219): NEXT: NEXT

LOCATE 12, 9: FOR i = 1 TO 62: PRINT CHR\$(223); : NEXT

COLOR 6, 0, 6

LOCATE 19, 1: FOR i = 1 TO 80: PRINT CHR\$(220); : NEXT

LOCATE 24, 1: FOR i = 1 TO 80: PRINT CHR\$(223); : NEXT

COLOR 1, 15, 6

FOR i = 0 TO 3

LOCATE 20 + i, 2: PRINT " "

"

NEXT

COLOR 4, 15, 6

LOCATE 3, 11: PRINT " "

"

LOCATE 4, 11: PRINT " SZÁMOK EGÉSZ OSZTÓI "

"

LOCATE 5, 11: PRINT " "

"

LOCATE 6, 11: PRINT " A program egy adott szám esetében megkeresi azokat az "

LOCATE 7, 11: PRINT " egész számokat, amelyekkel az adott szám maradék nél-
kül "

LOCATE 8, 11: PRINT " osztható. A számokat kiírja, s ha nem talál többet, ezt "

LOCATE 9, 11: PRINT " jelzi. (Jelzését várjuk meg !)

"

LOCATE 10, 11: PRINT " "

"

LOCATE 11, 11: PRINT " "

"

```

COLOR 14, 0, 6
LOCATE 13, 9: PRINT CHR$(201)
FOR i = 0 TO 59: LOCATE 13, 10 + i: PRINT CHR$(205): NEXT
LOCATE 13, 70: PRINT CHR$(187)
FOR i = 0 TO 3: LOCATE 14 + i, 9: PRINT CHR$(186); : NEXT
FOR i = 0 TO 3: LOCATE 14 + i, 70: PRINT CHR$(186); : NEXT
LOCATE 18, 9: PRINT CHR$(200)
FOR i = 0 TO 59: LOCATE 18, 10 + i: PRINT CHR$(205): NEXT
LOCATE 18, 70: PRINT CHR$(188)
COLOR 4, 3, 6
LOCATE 14, 11: PRINT " "
LOCATE 15, 11: PRINT " Kérem, adja meg a számot : "
LOCATE 16, 11: PRINT " (Ha 0-t ad, a program megáll.) "
LOCATE 17, 11: PRINT " "
LOCATE 15, 53: INPUT s
IF s = 0 THEN END
COLOR 1, 15, 6
FOR i = 0 TO 3
LOCATE 20 + i, 2: PRINT " "
NEXT
b = 1: x = 20: y = 3
GOTO szamol
szamol:
c = s / b: c1 = INT(c)
IF c = c1 THEN GOTO kiir2 ELSE GOTO kiir3
kiir2:
LOCATE x, y: PRINT b; : b = b + 1: y = y + 7
IF y > 70 THEN x = x + 1: y = 3
IF b > s THEN PRINT " Nincs több": GOSUB all: GOTO prog
GOTO szamol
kiir3:
b = b + 1: GOTO szamol
all:
a$ = INKEY$
IF a$ = "" THEN GOTO all
RETURN

```

5. Számjegyek összege egyenlő a négyzetük számjegyeinek összegével

```

COLOR 6, 15, 14
CLS
y = 1: x = 1: z = 21: GOSUB ker

```

COLOR 0, 15, 14

LOCATE 2, 3: PRINT "Vannak számok, amelyek számjegyeinek összege egyenlő négyzetük számjegyeinek"

LOCATE 4, 3: PRINT "összegével (pl.: 9, négyzete=81, 8+1=9). A program ezeket a számokat keresi"

LOCATE 6, 3: PRINT "meg és írja ki 10-től 999-ig. (Nyomj meg egy gombot!)"

GOSUB all

COLOR 4, 15, 14

g = 8: za = 5

FOR i = 10 TO 999

x = i: y = i * i

x\$ = STR\$(x): y\$ = STR\$(y)

x1 = LEN(x\$): y1 = LEN(y\$)

GOTO oax

GOTO oay

oax:

IF x1 = 3 THEN GOTO oax2

IF x1 = 4 THEN GOTO oax3

oax2:

a\$ = LEFT\$(x\$, 2): a = VAL(a\$)

b\$ = RIGHT\$(x\$, 1): b = VAL(b\$)

o1 = a + b: GOTO oay

oax3:

a\$ = LEFT\$(x\$, 2): a = VAL(a\$)

b\$ = RIGHT\$(x\$, 1): b = VAL(b\$)

c\$ = MID\$(x\$, 3, 1): c = VAL(c\$)

o1 = a + b + c: GOTO oay

oay:

IF y1 = 4 THEN GOTO oay3

IF y1 = 5 THEN GOTO oay4

IF y1 = 6 THEN GOTO oay5

IF y1 = 7 THEN GOTO oay6

IF y1 = 8 THEN GOTO oay7

oay3:

a\$ = LEFT\$(y\$, 2): a = VAL(a\$)

b\$ = RIGHT\$(y\$, 1): b = VAL(b\$)

c\$ = MID\$(y\$, 3, 1): c = VAL(c\$)

o2 = a + b + c

IF o2 = o1 THEN LOCATE g, za: PRINT x; y; o1: g = g + 1: GOTO vi

oay4:

a\$ = LEFT\$(y\$, 2): a = VAL(a\$)

```

b$ = RIGHT$(y$, 1): b = VAL(b$)
c$ = MID$(y$, 3, 1): c = VAL(c$)
d$ = MID$(y$, 4, 1): d = VAL(d$)
o2 = a + b + c + d
IF o2 = o1 THEN LOCATE g, za: PRINT x; y; o1: g = g + 1: GOTO vi
oay5:
a$ = LEFT$(y$, 2): a = VAL(a$)
b$ = RIGHT$(y$, 1): b = VAL(b$)
c$ = MID$(y$, 3, 1): c = VAL(c$)
d$ = MID$(y$, 4, 1): d = VAL(d$)
e$ = MID$(y$, 5, 1): e = VAL(e$)
o2 = a + b + c + d + e
IF o2 = o1 THEN LOCATE g, za: PRINT x; y; o1: g = g + 1: GOTO vi
oay6:
a$ = LEFT$(y$, 2): a = VAL(a$)
b$ = RIGHT$(y$, 1): b = VAL(b$)
c$ = MID$(y$, 3, 1): c = VAL(c$)
d$ = MID$(y$, 4, 1): d = VAL(d$)
e$ = MID$(y$, 5, 1): e = VAL(e$)
f$ = MID$(y$, 6, 1): f = VAL(f$)
o2 = a + b + c + d + e + f
IF o2 = o1 THEN LOCATE g, za: PRINT x; y; o1: g = g + 1: GOTO vi
oay7:
a$ = LEFT$(y$, 2): a = VAL(a$)
b$ = RIGHT$(y$, 1): b = VAL(b$)
c$ = MID$(y$, 3, 1): c = VAL(c$)
d$ = MID$(y$, 4, 1): d = VAL(d$)
e$ = MID$(y$, 5, 1): e = VAL(e$)
f$ = MID$(y$, 6, 1): f = VAL(f$)
h$ = MID$(y$, 7, 1): h = VAL(h$)
o2 = a + b + c + d + e + f + h
IF o2 = o1 THEN LOCATE g, za: PRINT x; y; o1: g = g + 1: GOTO vi
vi:
IF g > 22 THEN g = 8: za = za + 25
NEXT
END
all:
a$ = INKEY$
IF a$ = "" THEN GOTO all
RETURN
ker:

```



```

t = 40 - x: x1 = 40 + t: v = x1 - x - 1: m = z: y1 = y + m + 1
LOCATE y, x: PRINT CHR$(201)
LOCATE y, x1: PRINT CHR$(187)
FOR i = 1 TO v: LOCATE y, x + i: PRINT CHR$(205): NEXT
LOCATE y1, x: PRINT CHR$(200)
LOCATE y1, x1: PRINT CHR$(188)
FOR i = 1 TO v: LOCATE y1, x + i: PRINT CHR$(205): NEXT
FOR i = 1 TO m: LOCATE y + i, x: PRINT CHR$(186): NEXT
FOR i = 1 TO m: LOCATE y + i, x1: PRINT CHR$(186): NEXT
RETURN

```

6. Törtek egyszerűsítése

```

COLOR 14, 8, 0
CLS
DIM e3(30), x(30)
LOCATE 2, 12: PRINT "A program törteket egyszerűsít. Meg kell adni a"
LOCATE 4, 12: PRINT "számlálót és nevezőt. A tört szokásos formában kiíródik,"
LOCATE 6, 12: PRINT "mellette megjelenik az egyszerűsített tört."
LOCATE 8, 12: PRINT "Kérem a (max.7-jegyű) számot. 0-t a gép nem fogad el.)"
FOR i = 1 TO 76: LOCATE 9, 2 + i: PRINT "_": NEXT
bev:
LOCATE 11, 15: INPUT "Számláló:"; a: LOCATE 11, 50: INPUT "Nevező:"; b
FOR i = 1 TO 76: LOCATE 12, 2 + i: PRINT "_": NEXT
IF (a = 0 OR b = 0) THEN GOTO bev
LOCATE 14, 20: PRINT CHR$(201): LOCATE 14, 60: PRINT CHR$(187)
FOR i = 1 TO 39: LOCATE 14, 20 + i: PRINT CHR$(205): NEXT
FOR i = 1 TO 3: LOCATE 14 + i, 20: PRINT CHR$(186): NEXT
FOR i = 1 TO 3: LOCATE 14 + i, 60: PRINT CHR$(186): NEXT
LOCATE 18, 20: PRINT CHR$(200): LOCATE 18, 60: PRINT CHR$(188)
FOR i = 1 TO 39: LOCATE 18, 20 + i: PRINT CHR$(205): NEXT
a$ = STR$(a): b$ = STR$(b): a1 = LEN(a$): b1 = LEN(b$)
h1 = a1 + 1: IF b1 > a1 THEN h1 = b1 + 1
FOR i = 1 TO h1: LOCATE 16, 24 + i: PRINT "-": NEXT
LOCATE 15, 25: PRINT a: LOCATE 17, 25: PRINT b
LOCATE 16, 24 + h1 + 2: PRINT "="
sza:
c = a / b
IF c = INT(c) THEN e = c: LOCATE 16, 24 + h1 + 5: PRINT e: GOTO uta
GOSUB sza1
sza1:
RESTORE: FOR i = 1 TO 15: READ x

```

```

e1 = a / x: e2 = b / x
IF (e1 = INT(e1) AND e2 = INT(e2)) THEN e3(i) = x
IF (e1 <> INT(e1) OR e2 <> INT(e2)) THEN e3(i) = 1
NEXT
FOR i = 1 TO 14: s = e3(i) * e3(i + 1): e3(i + 1) = s: NEXT
IF s = 1 THEN LOCATE 16, 24 + h1 + 4: PRINT "nem egyszerűsíthető": GOTO
oszt
ere:
a2 = a / s: b2 = b / s
IF (a2 / 2 = INT(a2 / 2) AND b2 / 2 = INT(b2 / 2)) THEN s = 2 * s: GOTO ere
IF (a2 / 3 = INT(a2 / 3) AND b2 / 3 = INT(b2 / 3)) THEN s = 3 * s: GOTO ere
IF (a2 / 5 = INT(a2 / 5) AND b2 / 5 = INT(b2 / 5)) THEN s = 5 * s: GOTO ere
IF (a2 / 7 = INT(a2 / 7) AND b2 / 7 = INT(b2 / 7)) THEN s = 7 * s: GOTO ere
LOCATE 15, 24 + h1 + 5: PRINT a2: LOCATE 17, 24 + h1 + 5: PRINT b2
a2$ = STR$(a2): b2$ = STR$(b2): a3 = LEN(a2$): b3 = LEN(b2$)
h2 = a3 + 1: IF b3 > a3 THEN h2 = b3 + 1
FOR i = 1 TO h2: LOCATE 16, 24 + h1 + 4 + i: PRINT "-": NEXT
LOCATE 20, 28: PRINT "("; s; "volt a közös osztó.)"
IF a2 > b2 THEN GOTO atal: ELSE GOTO oszt
atal:
t1 = INT(a2 / b2): t2 = t1 * b2: t3 = a2 - t2
LOCATE 16, 24 + h1 + h2 + 6: PRINT "="; t1;
t1$ = STR$(t1): h3 = LEN(t1$)
t2$ = STR$(t2): h4 = LEN(t2$): t3$ = STR$(t3): h5 = LEN(t3$)
z = h4: IF h5 > h4 THEN z = h5
FOR i = 1 TO z - 1: LOCATE 16, 24 + h1 + h2 + h3 + 9 + i: PRINT "-": NEXT
LOCATE 15, 24 + h1 + h2 + h3 + 10: PRINT t3
LOCATE 17, 24 + h1 + h2 + h3 + 10: PRINT b2
oszt:
LOCATE 22, 22: PRINT "A tört értéke tizedestörtben"; a / b
uta:
LOCATE 24, 9
PRINT "(A 'v' gombbal a program megáll, más gombbal új feladatot kér.);
GOSUB all : GOTO bev : DATA 1,2,3,5,7,11,13,17,19,23,29,31,37,39,41
all:
a$ = INKEY$ : IF a$ = "" THEN GOTO all
IF a$ = "v" THEN END : IF a$ <> "" THEN CLS : GOTO bev
RETURN

```

FELHASZNÁLT IRODALOM

Clarence B.Germain: IBM PC XT/AT programozói kézikönyv
Novotrade, 1988

Tóth László : Számítástechnika, informatika
Fejér Megyei Pedagógiai Intézet, 1992

Dr. Hetényi Pálné : Számítástechnika középfokon
Számítástechnikai Oktatási Füzetek, 1991

Nagyné Bodor Ezsébet–Pálmai Éva: DOS alapismeretek
Számítástechnikai Oktatási Füzetek, 1991

Csajbók Zoltán : Számítástechnikai alapismeretek
Számítástechnikai Oktatási Füzetek, 1992

L-né Dr. Kálmán Éva, Kiss Zoltán, Dr.Tamás Péter, Tóth Bertalan:
MS - DOS 5.0 felhasználói szemmel
COMPUTER BOOKS, 1992

IBM PC DOS I.
Alkalmazástechnikai Tanácsadó Szolgálat, 1989

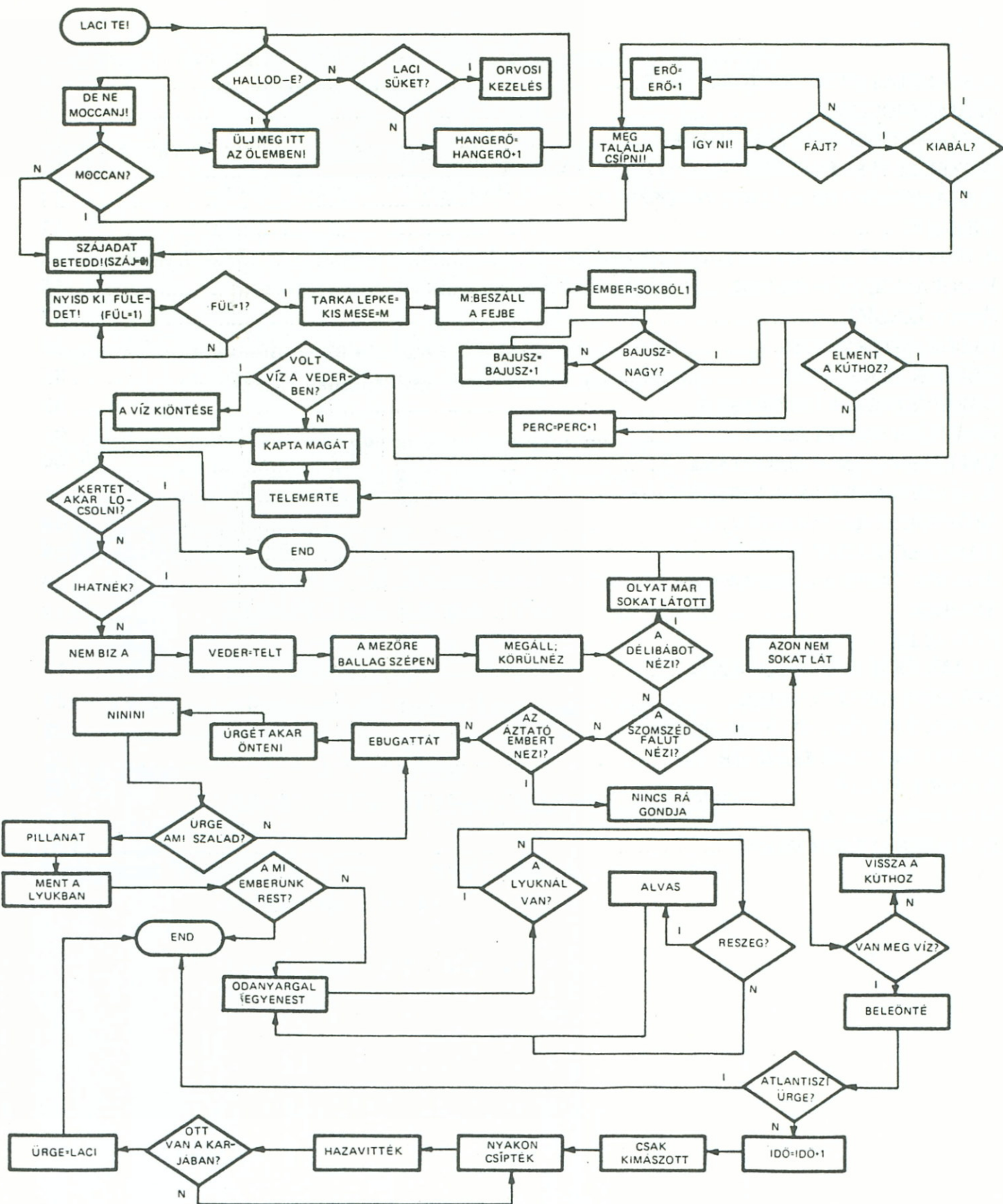
Baján Ferenc : A PC vírusai
Berzsenyi Dániel Tanárképző Főiskola, 1993

Hungarian VirusBuster Team: VirusBuster programcsomag
Felhasználói leírás

User's Guide MS-DOS Version 6.2
Microsoft

TARTALOMJEGYZÉK

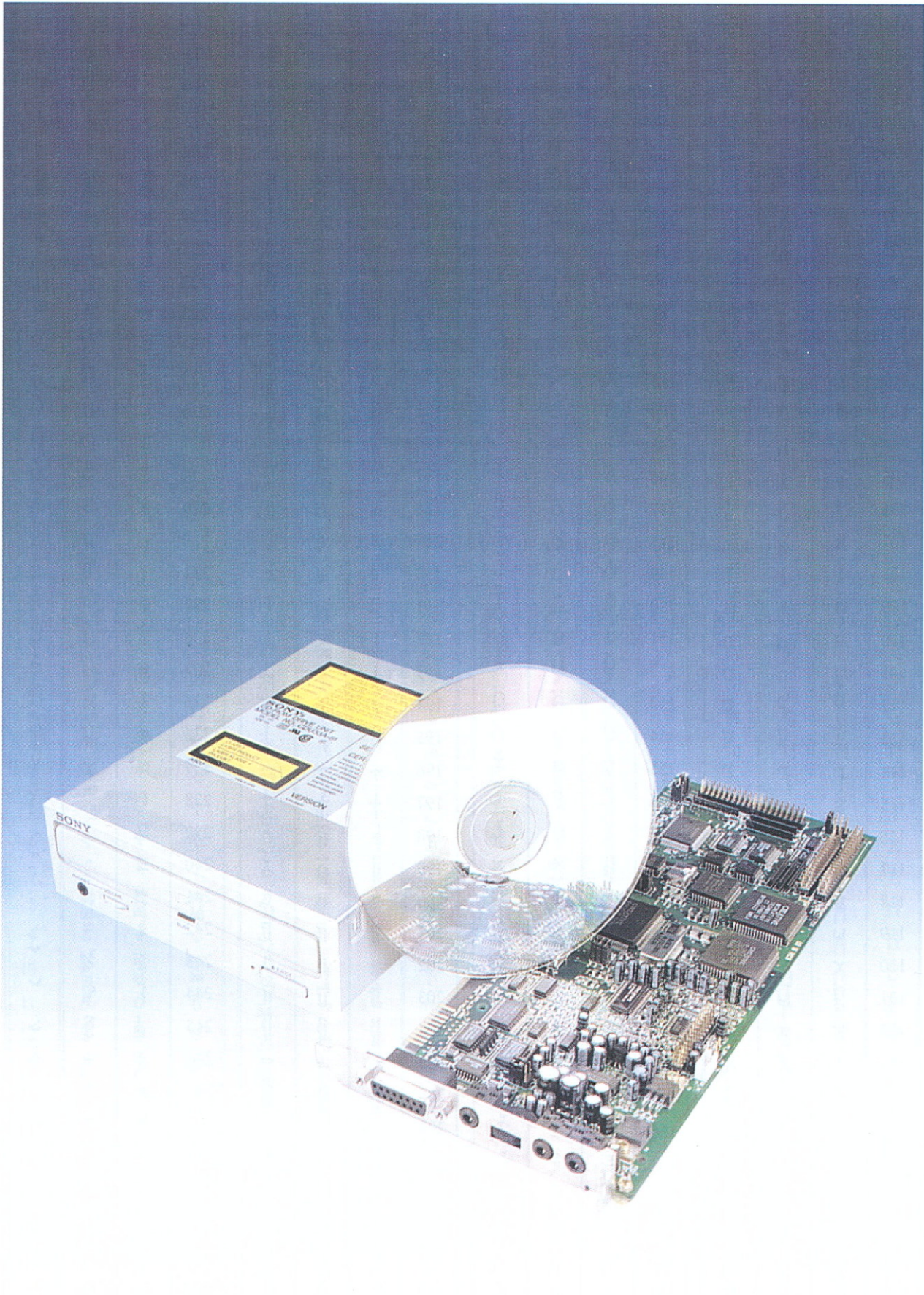
A SZÁMÍTÁSTECHNIKA TÖRTÉNETE.....	4
SZÁMRENDSZEREK	8
A SZÁMÍTÓGÉP ELVI FELÉPÍTÉSE	12
A SZEMÉLYI SZÁMÍTÓGÉP ALAPKONFIGURÁCIÓJA.....	18
A SZÁMÍTÓGÉP OPERÁCIÓS RENDSZERE	23
VÍRUSOK.....	43
A NORTON COMMANDER HASZNÁLATA	48
PPOGRAMOZÁSI NYELVEK.....	55
QUICK BASIC	57
PROGRAMTERVEZÉS	75
SZÁMOK KEREKÍTÉSE.....	85
MENÜVEZÉRELT PROGRAMOK	88
ISMÉTLÉSI SZERKEZETEK	92
VÉLETLENSZÁM ELŐÁLLÍTÁSA.....	99
A SZÁMÍTÓGÉP KARAKTERKÉSZLETE	103
SZÖVEGKEZELŐ FÜGGVÉNYEK.....	104
HANG ELŐÁLLÍTÁSA.....	107
ADATOK TÁROLÁSA PROGRAMBAN.....	110
TÖMBÖK.....	112
FÜGGELÉK	
AZ MS-DOS 6.20 PARANCSAI.....	113
MS-DOS HIBAÜZENETEK	124
A CHKVIR VÍRUSÖLŐ RENDSZER	128
NORTON COMMANDER 4.0	132
A QUICK BASIC	144
MINTAPROGRAMOK	155
FELHASZNÁLT IRODALOM.....	170



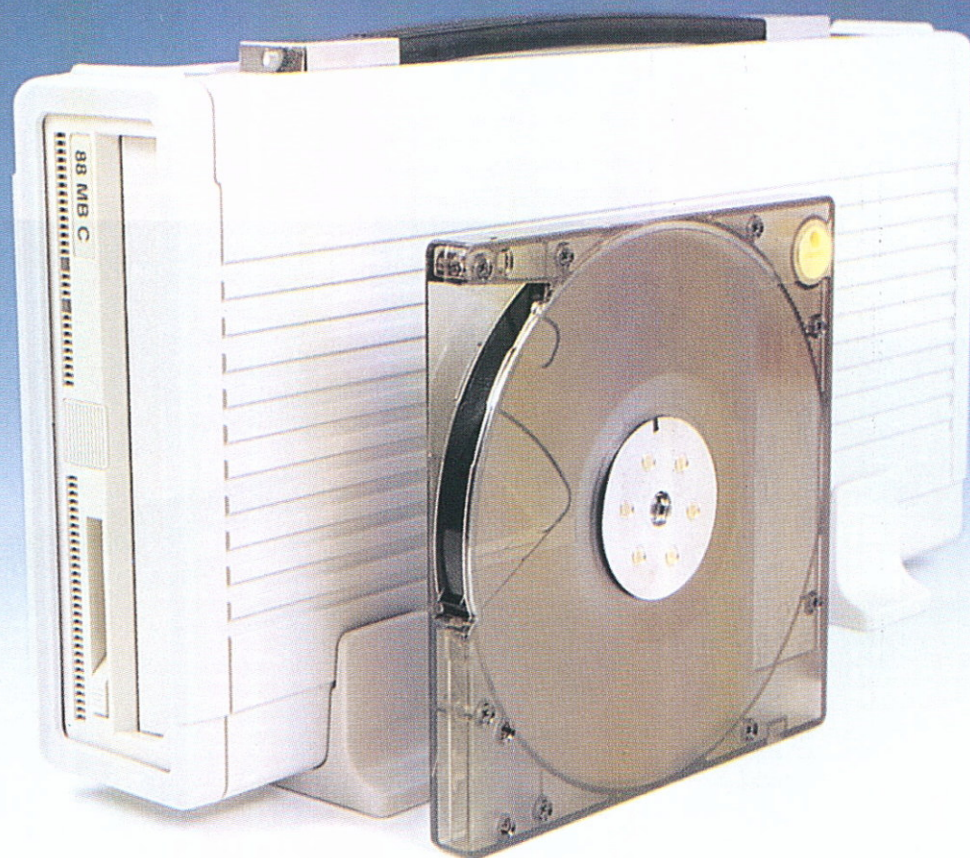
Folyamatábra



Egy lehetséges személyi számítógép konfiguráció

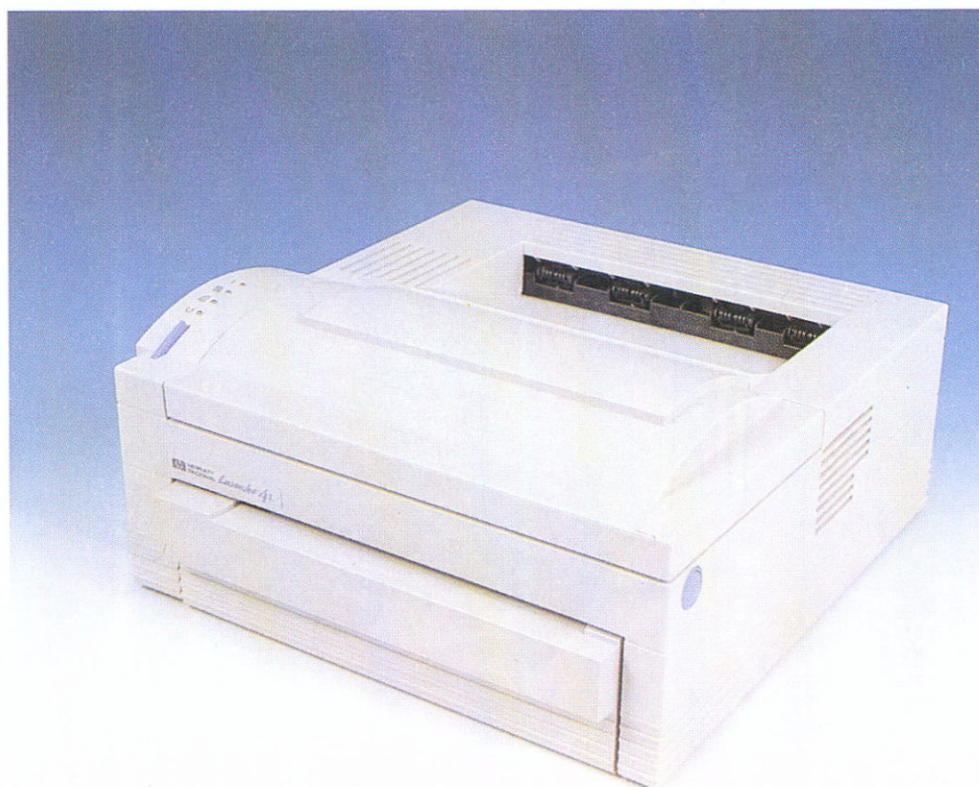
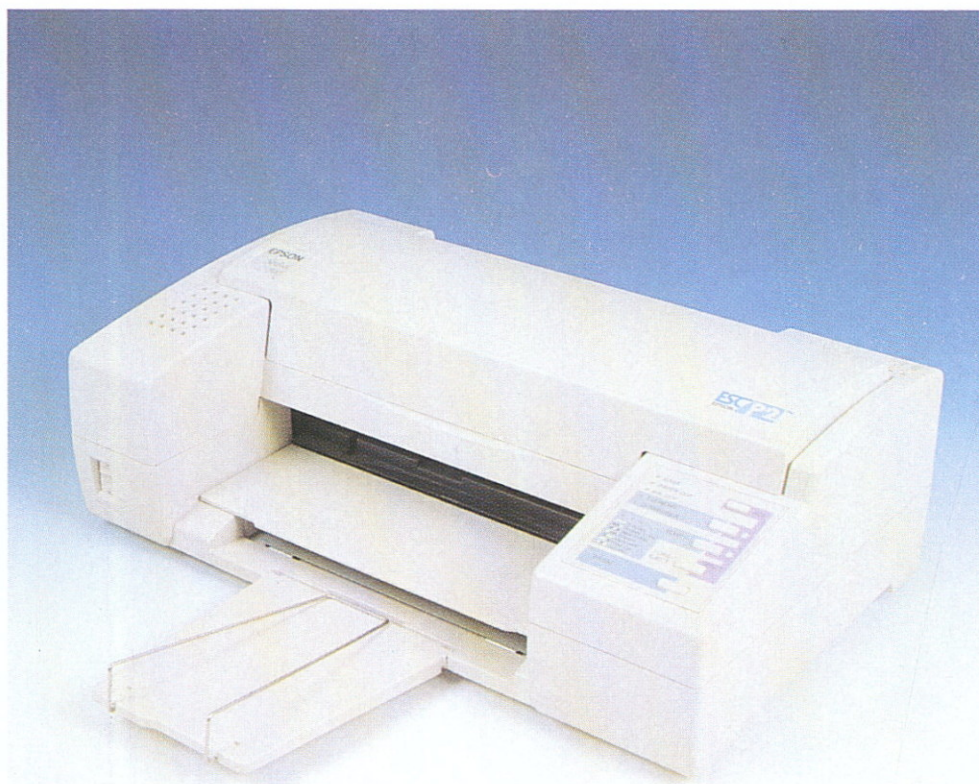


CD-lejátszó, CD-lemez és hangkártya



Cserélhető winchester és meghajtója

Tintasugaras
nyomtató



Lézernyomtató



Mátrixnyomtató



Lapolvasó



Hordozható számítógép (laptop)

További számítástechnikai kiadványaink:

Fekete Sándorné: Számítógép-kezelés az alapoktól
(Raktári száma: 53 282/1)

Fekete Sándorné: Windows-iskola
(Raktári száma: 53 288)

Fekete Sándorné: AutoCAD-iskola
(Raktári száma: 53 306)

Kökény Sándorné: Szövegszerkesztési alapismeretek
(Raktári száma: 53 307)

Új számítástechnikai tankönyvsorozatunk eddigi kötetei:

Varga Kornél: Bevezetés az adatfeldolgozásba
(Raktári száma: 73 530)

Varga Kornél: Az adatbázis-kezelés alapjai
(Raktári száma: 74 530)

Danitz Béláné: A táblázatkezelés alapjai
(Raktári száma: 73 510)

Szalayné Tahi Zsuzsa: Feladatgyűjtemény a táblázatkezeléshez
(Raktári száma: 73 513)

Tóth Tamás: A szövegszerkesztés alapjai
(Raktári száma: 73 410)

Mészáros Tamásné: Logo-világ
(Raktári száma: 72 311)

A Számítástechnika iskolásoknak I–II. sorozatunkat az általános iskola felső tagozatos és a korban ennek megfelelő kisgimnazista korosztálynak szánjuk.

A bevezető kötet a kezdőknek szól és szinte minden olyan alapvető ismeretet tárgyal, amit kezdőknek tanítani szokás (a számítógép elvi felépítése, története, DOS alatti gépkezelői ismeretek, vírusok), majd a programozás alapjaiba vezet be a Quick Basic nyelv segítségével. (Raktári száma: 539)

A második kötet szerves folytatása az előzőnek, a Quick Basic nyelvi ismeretek továbbfejlesztését tűzi ki célul – kiemelt figyelmet fordítva a grafikára. (Raktári száma: 639)

A programozás tanításához javasoljuk még Horváth Tibor: Turbo Pascal 7.0 című tankönyvét is.
(Raktári száma: 739)

669

