

ADAPTIV OKTATÓRENDSZEREK KÉSZÍTÉSE
CSOPORTOS SZÁMITÓGÉPES OKTATÁSHOZ

EGYETEMI DOKTORI ÉRTEKEZÉS

NAGY ELEMÉRNÉ

SZEGED
1985



TARTALOMJEGYZÉK

old.

BEVEZETÉS	1
1. A SZÁMITÓGÉPES OKTATÓRENDSZERÜNK ELŐZMÉNYEI....	7
1.1 A számítógépi rendszerrel szemben támasztott főbb követelmények	7
1.1.1 Alkalmazási követelmények.....	7
1.1.2 Az eszközökre vonatkozó követelmények, le- hetőségek.....	12
1.2 A megvizsgált oktatórendszerek tapasztalatai.	13
1.2.1 Számítástechnikai szaktárgyak tanulását se- gítő rendszerek.....	15
1.2.2 Általános célú oktatórendszerek.....	17
1.2.3 Tantárgy, témakör orientált oktatórendsze- rek.....	22
2. A SZÉF OKTATÓRENDSZER KIDOLGOZÁSA.....	25
2.1 A SZÉF oktatórendszer megtervezése.....	25
2.1.1 További feladatelemzés	25
2.1.1.1 Funkcionális elemzés.....	25
2.1.1.2 Adatmodell elemzés	31
2.1.1.3 Tananyag és oktatóprogram kapcsolata....	39
2.1.2 Rendszerterv.....	42
2.1.2.1 Adatrendszer, fájl strukturák.....	43
2.2 A programok funkciója és használata.....	49

2.2.1 A TAFEL program.....	49
2.2.2 Az oktató program	57
2.2.3 Elemző programok.....	65
2.2.4 A szervíz program	66
2.3 A tananyagok kidolgozása	67
3. AZ OKTATÓRENDSZER ALKALMAZÁSI MÓDJA, TAPASZTALATAI.	70
3.1 Az adaptáció tapasztalatai.....	70
3.2 Oktatási tapasztalatok.....	72
3.2.1 A kitűzött oktatási cél teljesülése.....	72
3.2.2 Az oktatórendszer általános használhatósága.	74
3.2.3 Tananyag optimalizáló alkalmazások.....	75
3.3 További célkitűzések.....	80
IRODALOMJEGYZÉK.....	81

BEVEZETÉS

A Pedagógia története során kevés jelenség váltott ki olyan élénk reakciót, mint a számítógépek megjelenése a tanítási folyamatban. A számítógéppel támogatott oktatásnak vannak hívei és heves ellenzői, de kevesen vannak, akik figyelmen kívül hagyják.

Az 1960-as évek elejétől - az első kísérletek megkezdése óta - a számítógépes oktatásról váltakozó hevességgel folyik a vita. Meggyőződésünk, hogy a számítógépes oktatás nem csodaszer, de a számítógép igen jó oktatási segédeszköz is lehet, ha megfelelő célra és formában alkalmazzák. E vélemény igazolására nyílt lehetőség az Élelmiszeripari Főiskolán, ahol olyan oktatási igények merültek fel, amelyekre a számítógéppel támogatott oktatás jó lehetőséget adott.

A konkrét igényeket és lehetőségeket megvizsgálva elkészítettünk egy általános célú, adaptív oktatóprogram rendszert és kidolgoztuk az alkalmazásához szükséges tananyagokat.

Az általunk kidolgozott programrendszer egyesíti a [17] , [22] -ben leírt oktatórendszerek előnyeit (pl. általános célú alkalmazhatóság, csoportos és egyéni tanulás biztosítása) és kiküszöböli azok hátrányait, (pl. hardver eszközök magas ára, helyhez kötöttség).

Oktatórendszerünk jellemző tulajdonsága az adaptivitás, amely mind hallgatói oldalról (pl. többszintű ismeretközlés), mind oktatói oldalról (pl. válaszértékelési stratégiák) dominál. Programrendszerünk elsődleges előnyét abban látjuk, hogy lehetővé teszi a számítógépes oktatást a laikus, számítástechnikai ismeretekkel nem rendelkező oktatók számára is. Az esetleges hibák javítása egyszerű és gyors. A teendőkre és a korrekcióra vonatkozó üzenetek, figyelmeztetések magyar nyelvűek. Ezzel a "user-friendly" oktatórendszerrel elérhető, hogy a számítógépes oktatás széles körben elterjedjen, mivel alkalmazása nem számítástechnikai, hanem szaktárgyi tudásra és pedagógiai tapasztalatokra épül. Ilyen paraméterekkel rendelkező megvalósítást a szakirodalom áttekintése során nem találtunk.

Jelen disszertáció az oktatórendszerrel kapcsolatos előkészítő és fejlesztő munkát, valamint az oktatási folyamatban való felhasználás tapasztalatait tartalmazza. Kézenfekvő volt, hogy az oktató programrendszer életképességének eldöntésére a Számítástechnika tantárgyat válasszuk, ami olyan speciális igényeket vetett fel, amelyeket a hagyományos módszerekkel nem lehetett kielégíteni.

Az alábbiakban röviden áttekintjük az Élelmiszeripari Főiskolán folyó számítástechnikai képzés - szempontunkból lényeges - jellemzőit.

Az Élelmiszeripari Főiskolán 1979 óta folyik számítástechnikai alkalmazói képzés az élelmiszertechnológus és élelmiszeripari gépész szakokon.

A képzés két lépcsőben történik. Az első lépcső célja a számítástechnikai ismeretek megalapozása a "Számítástechnika" tantárgy keretében két féléven át mintegy 30 óra előadás és 45 óra gyakorlat keretében. A második lépcsőben a számítástechnika nem önálló tantárgyként jelentkezik, hanem bizonyos szaktárgyak keretében mintegy 30 órában számítástechnikai alkalmazásokkal, alkalmazási lehetőségekkel ismerkednek meg a hallgatók.

A számítástechnika lényegének megértéséhez két alapvető fontosságú összetevő emelhető ki:

- empirikus kapcsolat egy számítógéppel,
- egy programozási nyelv (legalább passzív) megismerése.

Az Élelmiszeripari Főiskolán a számítástechnika oktatásához gépi bázisként 1979-1982-ig egy VTS 56100 típusú, a JATE R40 számítógéphez kapcsolódó terminál állt rendelkezésre. A számítástechnika tantárgy oktatási célja e feltételek mellett az alkalmazások lehetőségeinek megismerésére irányult.

Nem lehetett cél, hogy a hallgatók megtanuljanak programozni (még amatőr szinten sem). Ennek ellenére minden évfolyamon volt néhány érdeklődő hallgató, aki igényelte az önálló géphasználati lehetőséget, és a szabad-

idejét is szívesen fordította számítástechnikai ismereteinek gyarapítására. Az évek során egyre több hallgató tanulta meg a programkészítés alapjait is, így szakdolgozatok, tudományos diákköri dolgozatok keretében egyre szaporodtak a számítástechnikai alkalmazások. A hallgatók számára 1980-ban írtunk ki először számítástechnikai feladatmegoldó versenyt, mely azóta haladó hagyománnyá vált, hallgatói igényre minden évben megismételjük.

A számítástechnikai oktatás órarend szerinti formája mellett fokozatosan kialakult tehát egy "szabadabb" egyéni géphasználati lehetőség és számítástechnikai önképzés is.

1982-től kezdődően hazánkban is elérhetővé, megvásárolhatóvá váltak a - kis méretű és viszonylag olcsó - személyi számítógépek. A számítástechnikai eszköztár bővítése céljából az Élelmiszeripari Főiskola is vásárolt több ZX81, ZX Spektrum majd COMMODORE 64 típusú személyi számítógépet, amelyeket felhasználhattunk a számítástechnikai alapok oktatásában.

Az egyszerűen kezelhető, gyors válaszidejű számítógépek több munkahelyet, gépvezérlési lehetőséget jelentettek hallgatóink számára.

A géphasználati lehetőség jelentős növekedése minőségi változásokat érlelt. Egyre inkább kielégítővé vált

az a hallgatói igény, hogy a számítástechnika tantárgy keretében ne csak az alkalmazási lehetőségeket ismerjék meg, hanem megtanulhassák a programkészítés alapjait is; jussanak el az egyszerűbb alkalmazások (illetve az egyszerűsített alkalmazói modellek) programjainak elkészítéséig.

A módosított oktatási cél tananyag-ütemezési problémát vetett fel. Nevezetesen: míg a korábbi években a programozási nyelv megtanítása hosszú időt (körülbelül a rendelkezésre álló óraszám 2/3-át) töltött ki, az új oktatási célhoz ezt az időtartamot radikálisan le kell rövidíteni; legkésőbb az első félév közepére már ismerni kellene a hallgatóknak a programozási nyelv utasításkészletét és szintaxisát.

Az órarend szerinti hagyományos előadásokkal és gyakorlatokkal ezt a gyorsított ütemet nem lehet megvalósítani, ezért esett a választásunk a számítógéppel támogatott oktatási formára.

Elképzelésünk szerint a számítógéppel támogatott oktatási formában a hallgatók egyéni ütemezés szerint tanulják meg a nyelvi kurzus anyagát.

A tanulás úgy történik, hogy a hallgató a kijelölt határidőig az általa választott időpontokban a számítógép közvetítésével tanulja meg a BASIC nyelvet.

A számítógép funkciója az ismeretközlés és ismeretellenőrzés egysége; azaz a kurzus csak akkor tekinthető tel-



jesítettnek, ha a hallgató a folyamatos ismeretellen-
őrzéseken elérte a kívánt szintet.

Elképzelésünk kivitelezhetőnek tűnt. Eszközként rendelkezésre álltak az oktatásban amúgy is használt személyi számítógépek; a teachware-t pedig megszerezhetőnek illetve elkészíthetőnek ítéltük. Tervünk figyelembe vette a középiskolai számítástechnikai program beindítását, azaz azt a tényt, hogy hallgatóink differenciált számítástechnikai ismeretanyaggal kezdik főiskolai tanulmányaikat. Néhány közülük (s ez a szám a későbbiekben reményeink szerint erőteljesen növekszik) már jelentős számítástechnikai ismeretanyaggal rendelkezik; célszerű, ha e kör számára a "nyelvi" kurzus minimális időt igényel; míg másoktól alaposabb tanulást, több ráfordítást kíván.

Tervünk épít arra, hogy főiskolánkon hagyománya van az egyéni számítástechnikai önképzésnek, azaz összhangban van a korábban kialakult géphasználati szokásokkal.

1. A SZÁMÍTÓGÉPES OKTATÓRENDSZERÜNK ELŐZMÉNYEI

1.1 A számítógépi rendszerrel szemben támasztott főbb követelmények

Az alábbiakban áttekintjük azokat a követelményeket, amelyeket a számítógépi rendszernek - a korábban vázolt célok eléréséhez - ki kell elégítenie.

1.1.1 Alkalmazási követelmények

Alapvető szempont a csoportos egyéni tanulás biztosítása. A csoportos egyéni tanulás lényege az, hogy a tanulói csoport követelményrendszere, teljesítendő feladatköre azonos, ugyanakkor a teljesítés ütemezése eltérő, egyéni lehet.

Esetünkben csoportos követelmény a BASIC nyelv alapjainak elsajátítása egy adott határidőig. Ugyanakkor egyénileg választható meg, hogy a rendelkezésre álló idő (kb. 1.5 hónap) alatt az egyes hallgatók mikor kezdik el és milyen intenzitással végzik a tanulást.

Egyes hallgatóknál például előnyösebb az intenzív foglalkoztatás (rövid átfutási idő, magas óraszám) másoknál a hosszabb ideig tartó, kevésbé kumulált ütemezés a kedvezőbb.

A csoportos egyéni tanulás megvalósításához a számítógépes oktatórendszernek biztosítani kell mindkét funkcionális elemi összetevőjét, az ismeretközlést és az ismeret ellenőrzést.

Oktatási-alkalmazási céljaink megkívánják, hogy az ismeret ellenőrzés ne egyszeri (vizsga-jellegű) hanem folyamatos, vagy legalábbis rendszeres legyen. Másként fogalmazva: újabb ismeretek közlésére, csak akkor kerüljön sor, ha a korábbiak elsajátítása ellenőrzötten megtörtént.

A hallgatók különböző számítástechnikai ismeretanyaggal kezdik meg főiskolai tanulmányaikat. Vannak olyan hallgatók, (igaz nem túl nagy arányban) akik középiskoláskorukban kapcsolatba kerültek a számítógéppel, s az alapismeretek nagy részét már elsajátították. Számukra az alapozó szakaszt célszerű gyorsan teljesíthetővé tenni.

A tapasztalatok szerint az egyes hallgatók számítástechnikai ismeretbefogadó képessége is jelentősen eltérő.

Ez a jelenség valószínűleg abból származik, hogy az Élelmiszeripari Főiskola Élelmiszertechnológus szak felvételi tantárgyai a biológia és a kémia, így jelentős azon hallgatók aránya is, akik nem utolsósorban azért jelentkeztek a főiskolánkra, mert a matematika és fizika nem felvételi tárgy. Ugyanakkor a számítástechnika a matematikához és fizikához áll legközelebb. Az előismeretek illetve a befogadóképesség különbözősége miatt célszerű az ismeretközlés többszintűségét biztosítani.

Ez részletesebben azt jelenti, hogy az ismeretközlés először csak egy rövid, tömör megfogalmazásban történik. Ha a hallgatónak ez kevés, akkor mód legyen arra, hogy ugyanannak az ismeretkörnek egy bővebb, részletesebb példákkal illusztrált megfogalmazását is megkapja.

Ebben az oktatási rendszerben a tanár kapcsolata a hallgatókkal esetleges, a korábbi oktatási módoktól eltérően a tanulási folyamat nem kötődik a személyes kapcsolathoz. Az előadásokon, gyakorlatokon alapuló hagyományos oktatásban a tanárnak állandó visszajelzése van a hallgatóktól arra nézve, hogy állnak az anyag megértésében, elsajátításában. A számítógépes oktatási rendszerben sem képzelhető el az, hogy a tanár ne rendelkezék információval arról, hogy a tanulók meddig jutottak az anyag elsajátításában, mely részek okoztak nehézséget nekik, melyek esetleg a "túl tanított"részek. Így a számítógépes oktatási rendszerben is szükség van arra, hogy a tanár visszajelzéseket kapjon a tanulási folyamatról. Minimális információ igény az, hogy az ismeretek elsajátítására kijelölt időszak folyamán bármikor megtudhassa, hogy egyes hallgatói hol tartanak a feladatok teljesítésében. Ez az információ feltétele annak is, hogy az idejünkkel rosszul gazdálkodó hallgatóknál az oktató idejében közbeléphessen, intenzívebb haladásra készítse őket. A további tanári információk általában a gépi oktatási anyag minőségére, esetleg az adott hallgatói csoport

(évfolyam) és az oktatási anyag kapcsolatára vonatkoznak. A sikeres oktatási anyagok általában nem egy egyszeri akció eredményeként keletkeznek, hanem rendszeresen csiszolni, javítani, esetleg optimalizálni is célszerű a tananyagokat. A javítás, optimalizálás szempontjait a tanítási folyamat eredményességéből illetve hiányosságaiból lehetne a legtermészetesebb módon meríteni. A számítógéppel támogatott oktatás olyan lehetőségeket kínál a tanulási folyamat eredményeinek rögzítésére és kiértékelésére, amelyeket a hagyományos tanítás esetén lehetetlennek vagy megvalósíthatatlannak ítélnénk. Megoldható az, hogy az ellenőrző kérdésekre kapott válaszokat a számítógép tárolja, ezekből elemző (pl. statisztikai) programokkal igen fontos információkat nyerhetünk az ismeretátadó részek teljességére és az ellenőrző kérdésekkel való összhangjára. Ha több populáció (pl. évfolyam) eredményeit vetjük egybe, akkor még további, a hallgatói "évjáratra" vonatkozó információkat is nyerhetünk. A csoportos egyéni tanulás megvalósításának legfontosabb a számítógépes oktatórendszerre vonatkozó követelményeit tehát az alábbiakban foglalhatjuk össze:

- a.) Biztosítsa mind az ismeretközlést, mind az ismeretellenőrzést.
- b.) Az oktatási anyag (kurzus) önálló részekből (tananyagok) épüljön fel; a számítógépes oktatórendszer

mindkét egység kezelését biztosítsa.

- c.) Új tananyagra a hallgatót csak akkor engedje áttérni, ha a korábbi tananyagokat már megtanulta, azaz az ismeretellenőrzés pozitív eredménnyel járt.
- d.) A hallgatók egymástól (csoporttársaiktól) függetlenül haladhassanak előre a kurzus elvégzésében.
- e.) Az ismeretközlés különböző részletezettségben történhessen.
- f.) A tanár kapjon információt a számítógéptől arra, hogy az egyes hallgatók hol tartanak a kurzus teljesítésében.
- g.) A számítógép rögzítse a tanulási folyamat eseményeit a későbbi (különböző szempontok szerinti) tanári elemzésekhez.

A fenti követelményeket a konkrét oktatási cél és körülmények elemzésével állítottuk fel. Ugyanakkor elég általánosak ahhoz, hogy más oktatási területek, célok esetén is érvényesek legyenek;

Az "a" követelmény nélkül az alkalmazás vagy demonstrációs jellegű vagy "gépi vizsgáztatás".

A b), c), d), e) követelmények érvényesülnek általában minden kurzus-szerű (tananyagorientált) oktatási cél esetén. Az e) követelmény akkor hagyható el, ha a hallgatói populáció elegendően homogén.

Az f) és g) követelmények a tanár operativitásához kel-
lenek, enélkül a "tanár" csak passzív szemlélője lehet-
ne a tanulási folyamatnak.

1.1.2 Az eszközökre vonatkozó követelmények, lehető- ségek

Tervünk megvalósítása szempontjából alapvető köve-
telmény volt az is, hogy az oktatórendszert a rendelke-
zésre álló illetve beszerezhető hardver eszközökre épít-
sük. A "beszerezhetőség" nem csak a piaci viszonyokra
értendő, hanem egy hazai oktatási intézmény anyagi le-
hetőségeire is. A tervezési fázisban a Sinclair ZX
Spektrum személyi számítógépet vehettük számításba, táv-
latilag a COMMODORE 64 számítógépet. A márkás professzi-
onális számítógépeket (APPLE, IBM PC) részben az embargó
részben az anyagi lehetőségek miatt egyszerűen figyel-
men kívül kellett hagyni.

Megvizsgáltunk jó néhány (publikált és publikálatlan)
oktatási célú számítástechnikai alkalmazást, áttekin-
tettük a jelentősebb hazai és külföldi szoftver terjesztő
és forgalmazó intézmények termékismertetőit, de nem ta-
láltunk olyan rendszert, amely kielégítette volna a ko-
rábban részletezett funkcionális igényeket és megfelelt
volna a fenti hardver lehetőségeknek.

Gondolkodás nélkül ki kellett zárni pl. a WICAT Ltd - a szakirodalomban kedvezőnek tűnő - WISE rendszerét (WICAT Interactive System for Education [22]), vagy a DELTAK Ltd Pass nevű, APPLE II alapú kurzus készítő (course authoring) rendszerét [22] .

Igy amellettt döntöttünk, hogy a céljainkhoz szükséges szoftver eszközöket az Élelmiszeripari Főiskolán fejlesztjük ki.

1.1 A megvizsgált oktatórendszerek tapasztalatai

Röviden összefoglaljuk a fontosabb ismert oktatási célú alkalmazásokat, különös tekintettel azokra a szempontokra, amelyeket az 1.2. részben megfogalmaztunk.

Az oktatási célú számítástechnikai alkalmazásokat két csoportra oszthatjuk:

- számítógép mint az oktatás tárgya,
- számítógép mint az oktatás eszköze.

Ez a csoportosítás nem szó szerint fejezi ki az eltérés lényegét, ugyanis ha a számítógép az oktatás tárgya, akkor - néhány egészen kivételes esettől eltekintve - megjelenik az oktatás eszközeként is. Elvi és tapasztalati tény, hogy a számítástechnika pillérét alkotó alapismeretek jó része (programozási nyelvek, algoritmizálás, programozási módszerek) egyszerűen nem tanulható

meg a számítógépi gyakorlat nélkül.

"Nem tekintem programozónak azt, aki nem használt még fel ezer óra gépidőt" mondta Jean Dominique Warnier [94] .

A tárgy és az eszköz megkülönböztetés arra utal, hogy a számítógép nem csak a számítástechnikai gyakorlati ismeretek elsajátítására alkalmazható, hanem más, a számítástechnikától távol eső területeken (pl. zene- és nyelvoktatás) is felhasználható operatív oktatástechnikai eszközként [55] , [65] , [71] , [30] .

A fentiek figyelembevételével a számítógépre alapozott oktatási alkalmazásokat - az oktatás tárgya szerint - a következő csoportokra osztjuk.

- a.) Számítástechnikai ismeretek tanítására orientált rendszerek, ezen belül
 - a1) a számítógép a tanulás (gyakorlatszerzés) passzív tárgya
 - a2) a számítógép aktív, funkcionális szerepet játszik "önmaga megtanításában".
- b.) Általános oktató, tanító rendszerek, ezen belül;
 - b1) tantárgy, témakör orientált oktató rendszerek,
 - b2) általános célú (tantárgyra, témakörre nem szűkített) oktatórendszerek.

Nyilvánvaló, hogy az a2) a b1) speciális esetének tekinthető abban az esetben, ha az oktatás témaköre számítástechnikai.



Az Élelmiszeripari Főiskolán felmerült feladat esetén a számítástechnika egy fejezetének tanításához - eszközként - vettük figyelembe a számítógépet. Ennek megfelelően ez az alkalmazás az a2) csoportba tartozik. Az oktatási célú alkalmazások áttekintésénél tehát figyelmünket elsősorban az a2) és b1) típusú rendszerekre fordítottuk.

1.2.1. Számítástechnikai szaktárgyak tanulását segítő rendszerek

A legnagyobb al) típusú rendszer a Dartmouth College DTSS nevű rendszere [17]. A Dartmouth College vezetőinek az volt a célja, hogy a hallgatók a számítógépet, mint eszközt megismerjék, megszeressék és kezelni tudják. A College hallgatói számára a szabad, ingyenes terminálhasználat olyan jellegű szolgáltatás, mint pl. a könyvtárhasználat. A DTSS két nagyteljesítményű Honeywell számítógépre épül, és 1972 óta működik. A rendszer távadatátviteli hálózaton keresztül a központi géphez kapcsolja az Egyesült Államok és Kanada nagyvárosaiban elhelyezett több mint 200 terminált.

Hasonló típusú - természetesen méreteiben sokkal szerényebb rendszerrel hazánkban is találkozhatunk. Ilyen például a JATE regionális (R40 majd 1983-tól R55M alapú,

30-40 terminálos) számítógéprendszere [17] , amely 1978 óta tölt be a Dartmouth College-hez hasonló funkciót. A regionális rendszerbe 1979-ben főiskolánk is bekapcsolódott.

Ilyen széles eszközbázison általában találkozunk olyan kezdeményezésekkel, amelyek a2) vagy b) típusú alrendszerek létrehozására irányulnak.

Kisebb eszközigenyű (minigép alapú) hasonló rendszer a KFKI TEASYS [20] nevű szoftvere, amelyet pl. a Nyíregyházi Mezőgazdasági Főiskolán alkalmaznak számítástechnikai gyakorlati oktatás eszközeként.

A SZÍMOK PROLON [107] rendszere már fejlettebb (a2) típusú) alkalmazás. Nem az általános (szokásos) szoftver lehetőségeket biztosítja a tanuláshoz, hanem egy speciális (ESZR nagyszámítógépre és PDP 11/70-re készített) szoftver segítségével könnyíti meg a "programozás mesterségének" elsajátítását. A PROLON nyelvű programok kidolgozásához a rendszer igen magas szintű hibakezelési szolgáltatásokat nyújt, így lecsökkenti a "futóképes" programok előállításához felhasználandó géphez fordulások számát.

A PROLON rendszerhez hasonló a YALE egyetemen kifejlesztett MENO II, amely a PASCAL programozás tanulásában segíti a kezdőket [101] .

1.2.2 Általános célú oktató rendszerek

Az első széleskörű, államilag támogatott kísérletet 1960-ban kezdték meg az Illinois-i egyetemen. Ma már a - PLATO nevet viselő - multimédia oktatóhálózat egy CYBER típusú nagyteljesítményű központi számítógépre kapcsolt több ezer terminálos oktatórendszerként működik [17] .

A PLATO általános célú oktatórendszer, ami tükröződik a kialakításához felhasznált hardver, szoftver komponensek megválasztásában is.

A hardver jellegzetességek közül legszembeűnőbb a speciális "PLATO" terminál, ami nem csak a központi géptől érkező szöveges információkat és grafikákat tudja kezelni, hanem mikrofilm vetítőt, üzenetátadó készüléket is. Így a tananyagok tárolására nem csak a számítástechnikában szokásos mágneses rögzítés alkalmazható, hanem a fentiek is. A drága PLATO terminálok (kb 10 ezer \$) előnyei akkor jelentkeznek egyértelműen, ha a rendszerbe építendő oktatási anyag más médiumon (mikrofilm) már rendelkezésre áll, vagy például az oktatási célhoz jól illeszkedő hangfelvételek állnak rendelkezésre.

A PLATO szoftvere alapvetően két összetevőből áll. A rendszer "meghajtó" alapszoftverből és az oktatásra orientált alkalmazói szoftverből. Az alkalmazói szoftver nem tartalmazza az oktatási anyagokat, hanem adatként kezeli

a megfelelő számítógépi vagy más tárolású egységeket. Ez a megoldás az általános célú felhasználás mellett a széleskörű alkalmazásnak is feltétele; ha ugyanis az oktatási anyagokat programozástechnikailag beépítik a szoftverbe, akkor mind az új tananyagok létrehozása, mind a rendszerben levők módosítása magasszintű számítástechnikai, programozói szakismeretet igényel, azaz a tananyagkészítők csak egy igen szűk körből kerülhetnek ki. A PLATO rendszerben viszonylag egyszerűen, a TUTOR "szerzői" nyelven készítik a tananyagokat, így a potenciális tananyagkészítők köre igen nagy. A TUTOR nyelv egy egyszerű logikájú, könnyen megtanulható programozási nyelv, amelynek utasítás-készlete a PLATO célját és lehetőségeit tükrözi. Módot ad szöveges információközlésre, válaszelemzésre, ábrák megjelenítésére, a mikrofilm vetítő vagy más, speciális kiegészítő berendezések kezelésére.

A PLATO rendszer koncepciója szerint egyetlen közös tananyagbázisban gyűlnek össze a PLATO felhasználók által készített tananyagok. Mivel minden terminálról felhasználható a teljes tananyag-készlet és bármely felhasználó készíthet tananyagot, a rendszer kidolgozói azt várták, hogy a PLATO felhasználók között egy magasfokú, spontán munkamegosztás alakul ki. A tananyagbázisban ugyanis összegyűlnek a különböző oktatási területek eredményei, s az egyes felhasználók - ahelyett, hogy kidolgoznák a saját tananyagukat - válogathatnak a jobbnál jobb megoldások között.

Ez az elképzelés nem valósult meg. Bár a tananyagok mennyiségi szaporodása jelentős volt, egy tartalmilag és tematikailag rendezetlen, áttekinthetetlen tananyag "halmazzá" vált a PLATO tananyagbázisának jó része.

Az 1970-es évek elején kialakított és folyamatosan fejlesztett TICCIT rendszerben [17] a mikroelektronika - és általában a technikai eszközrendszer - fejlődését figyelhetjük meg.

A kisebb teljesítményű (mini kategóriájú) központi számítógép-párra kapcsolt, televízió-szerű terminálokból felépített rendszernél kiemelt célként szerepelt a reális ár. Egyes nézetek szerint ugyanis az árviszonyok okozták a PLATO relatív sikertelenségét.

A TICCIT rendszer már nem a globális, hanem a lokális hálózat koncepcióját követi. A központi géppárra ennél a rendszernél jóval kevesebb (128 db) terminál kapcsolható, mint a PLATO esetén; s a terminálokat a központ közelében (néhány száz méteres körzetben) kell elhelyezni. A TICCIT terminálok bővebb, korszerűbb oktatástechnikai lehetőségekkel rendelkeznek, mint a PLATO terminálok. A képernyő színes, az oktatási anyagba videofelvételek is beépíthetők, stb.

A kisebb méretek következtében jobban áttekinthető, megszervezhető a tananyagkészítés rendszere is.

A TICCIT tematikailag nem, de technológiailag segíti a tananyagok kialakításának szervezését.

Itt figyelhetjük meg - elsőként a pedagógiai - számítástechnikai munkamegosztást. Ez részletesebben azt jelenti, hogy a tananyagkészítés két különböző "szakmai fázisból" áll. Az oktató megtervezi a tananyagot és a tervet hagyományos eszközökkel (szöveggel, ábrákkal) rögzíti. A számítástechnikusok feladata a tervnek megfelelő tananyag gépi implementálása, az oktató által megtervezett "forgatókönyvhöz" a számítógéppel kezelt tananyag végleges elkészítése.

A PLATO és a TICCIT mind hardvert, mind szoftvert tekintve oktatási célalkalmazások.

Emellett találkozhatunk olyan kísérletekkel is, ahol az általános célú hardver-szoftver rendszeren belül bizonyos szoftver komponensekkel oktatási célalkalmazásokat lehet biztosítani.

Ebbe a csoportba tartozik pl. az IBM által forgalmazott COURSEWRITER szoftver rendszer [17]. Az IBM nagyteljesítményű számítógépeken használható COURSEWRITER "szerzői" nyelv segítségével konkrét ismeretközlő és számonkérő tananyagok programozhatók.

Bár sokan igen jó eszköznek tartják a COURSEWRITER-t, a szakirodalomban alig találunk hivatkozást a gyakorlati oktatásban szélesebb körben alkalmazott, COURSEWRITER alapú konkrét oktatórendszerre. Ugyanez mondható el a hasonló célú, korszerűbb PHOENIX rendszerről is [22].

Itt említjük a Havannai Állami Egyetem CID oktatórendszerét, amelynek célszoftvere 1980-ban elkészült, de használatba vétele azóta sem történt meg, mivel nem készülnek hozzá a tananyagok.

Ma már megállapíthatjuk, hogy az általános célú oktatórendszerek - szemben a hozzájuk fűzött reményekkel - mindeddig nem helyezték széleskörűen új alapokra az oktatást. Úgy tűnik, hogy ezen a területen előrébb jár a technikai potenciál, mint a tényleges széleskörű alkalmazási igény és lehetőség. A gyakorlati oktatás - részben anyagi részben, személyi okok miatt - nem tudta tömegméretekben felhasználni azt a technikai lehetőséget, amit a számítástechnika felkínált.

A nagyszámú, publikált, de nem funkcionáló "oktatórendszer" létéhez az is hozzájárul, hogy a számítógépek programozhatósága szinte kínálja a lehetőséget arra, hogy oktatógépként is alkalmazzuk. A tanulási- tanítási folyamat összetevőinek programozása ugyanis viszonylag egyszerű, egyszeri és mutatós feladat. A gyakorlati oktatásban való felhasználás viszont folyamatos magasszintű, (általában nem honorálható) pedagógiai munkát kíván, amiért rendszerint több kritikát lehet kapni, mint elismerést.

1.2.3. Tantárgy, témakör orientált oktatórendszerek

E hatalmas terjedelmű témakörből csak néhány érdekes alkalmazást emelünk ki és a megoldandó feladat szempontjából közvetlenül számításba vehető eszközökkel foglalkozunk részletesebben.

A különböző szakterületeken sajnos ma még csak egyedi, izolált jelenségként - igen figyelemreméltó alkalmazásokkal vagy alkalmazási kísérletekkel találkozunk. Ezek sorába tartozik a Stanfordi Egyetem GUIDON programja

[22] , amely egy intelligens, számítógépre alapozott oktatóprogram az orvosi diagnosztika tanítására. Figyelemre méltó a CAUGHT projekt [22] is, amely ULSTER földrajzának és történelmének tanításához készített számítógépes alkalmazásokat.

Érdekes alkalmazás az OREGON állami egyetemen készített, de középiskolai felhasználásra szánt PSOLVE program [58] , amely az egyenletmegoldás súlykolásában mentesíti a tanárt a mechanikus és sokszor idegölő munkától.

A vegyipari, kémiai alkalmazások tanítása igen korán jelent meg önálló számítástechnikai alkalmazási feladatként. A hazai fejlesztések főként ESZR nagyszámítógépre orientáltan folynak (CHEMISYS, UNISYS) [6] míg a Leeds-i egyetemen a CP/M alapú mikrogépeken használható STAF II-t fejlesztették ki [22] .

A feladatunk megoldása szempontjából figyelembe vehető számítógépek közül a Commodore (VC 20 és C64) gépekhez, több oktatási célú programot találtunk. Annak ellenére, hogy rendszerünk elkészítése ekkor már folyamatban volt, megvizsgáltuk ezeket a programokat is, nem utolsósorban azért, hogy felesleges volt-e a saját oktatórendszerre fordított munkánk.

A leglényegesebb funkcionális hiányosságokat az alábbiakban foglalhatjuk össze:

- 1.) Az említett programok nagy része nem fogja át a tanulási folyamatot, abban az értelemben, hogy vagy csak ismeretközlő, vagy csak számonkérő (ismeretlenőrző) funkciót tölt be.
 - Az ismeretközlő, demonstrációs programok (Nebuló, Easy Basic) nem ellenőrzik, hogy a tanuló valóban megértette-e a vele közölteket.
 - A számonkérő program ismeretközlés nélkül vizsgáztat. Kivételként említjük a Bits and Bytes nevű programot, amely ismeretközlő és ellenőrző egyszerre, ez viszont csak a címben jelzett szűk tematikai területet fogja át.
- 2.) A programokat nem szervezték rendszerré a következő vonatkozásokban:

- a programok egymástól független funkciókat töltenek be (a Nebulótól és az Easy Basic-től eltekintve) ad hoc módon ragadnak ki egy-egy részt a számítógép illetve a BASIC tanításából, tematikailag hiányosan, sok módszertani hibával.
- teljes mértékben hiányzik a tanulási folyamat eseményeinek rögzítése, így az 1.1.1-ben b, c, e, f, g-vel jelölt követelmények eleve kielégíthetetlenek.
- a tananyag és a kezelő algoritmus programozástechnikailag nincs szétválasztva, ami a következő hátrányokkal jár:
 - a) a tananyagok módosítása csak akkor jöhet szóba, ha a program forrásszöveg szinten rendelkezésre áll;
 - b) a metodikai jellegű változtatások általában a programszerkezetet érintik;
 - c) a programok más tananyagrészekre történő alkalmazása egyenértékű a program teljes újraírásával.

3.) A programok eredetileg angol nyelven oktatnak. Ez önmagában nem kizáró ok; ha egyébként kielégítőek, lényegesen egyszerűbb feladat a tananyagok nyelvi lefordítása, mint egy új rendszer kidolgozása.

2. A SZÉF OKTATÓRENDSZER KIDOLGOZÁSA

2.1 A SZÉF oktatórendszer megtervezése

2.1.1 További feladatelemzés

Az oktatórendszer megtervezésénél figyelembe kellett vennünk az 1.1.1 részben részletezett funkcionális igényeket, valamint az 1.1.2 részben leírt hardver lehetőségeket. Mivel a kidolgozás jelentős ráfordításokat igényel, a megoldandó feladatot további szempontok szerint kell elemezni, Warnier szavaival:

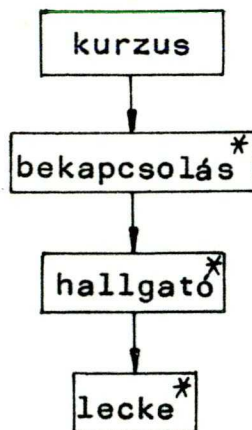
- "hogya a hasonló felhasználási igények szempontjait lehetőség szerint már korai fázisban felismerjük, s ezzel előre megteremtsük a potenciális további felhasználásokra alkalmas eszközt."

2.1.1.1 Funkcionális elemzés

Az alábbiakban részletezzük azokat a jellegzetességeket, amelyeket a BASIC nyelvet tanító (és más, hasonló célú) oktatóprogramnál jellemzőnek, feladat (illetve feladatosztály) specifikusnak tekintünk. Az elemzések eredményeként hozott döntéseket Jackson diagrammal ábrázoljuk.

a) Oktatási anyagunk (terjedelemben, gyakorlási igényben) túl nagy ahhoz, hogy egyszerre, egyetlen alkalommal meg lehessen tanulni; ezért logikai egységekre (ún. leckékre) kell bontanunk.

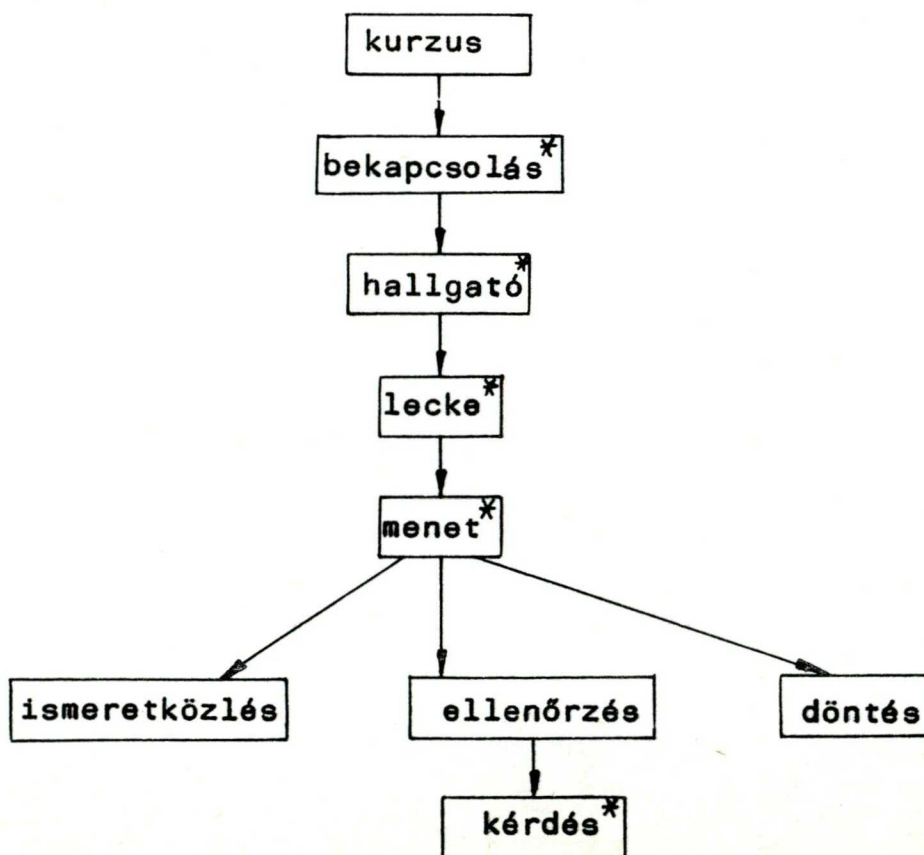
A kurzus során minden résztvevő hallgatónak meg kell tanulnia a leckéket. A kurzus teljesítésére több alkalom van, nevezzük ezeket az egyszerűség kedvéért "bekapcsolás"-nak.



1. ábra

b) A tanítás az ismeretközlés és ellenőrzés szerves egysége; így az oktatóprogramnak mindkét funkciót vezérelnie kell. A gyakorlati pedagógiai munkában az ismeretközlés és ellenőrzés a tanítási folyamat

két szimultán, egyszerre jelenlevő összetevője, amelyek közül egyes szakaszokban (pl. magyarázat, feleltetés) valamelyik oldal domináns (de sohasem kizárólagos). Mivel a szimultaneitás nem algoritmizálható, döntést kell hozni arra nézve, hogy mely pontokon és hogyan történjék az ismeretellenőrzés. A BASIC nyelv korábbi oktatási tapasztalatait és egyéb szempontokat figyelembe véve legcélszerűbbnek az a megoldás tűnt, hogy leckénként "megértést ellenőrző kérdésekkel" történjék ellenőrzés. Ha az ellenőrzés eredménye nem kielégítő, akkor újbóli (esetleg részletesebb) ismeretközlés majd újabb ellenőrzés következzen.

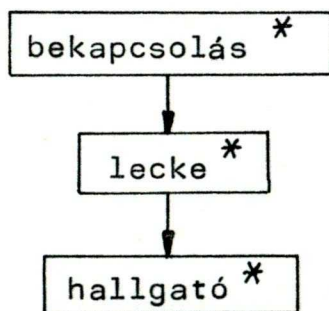


2. ábra

Annak megválaszolása, hogy hány kérdést kell feltenni és mi a "kielégítő tudásszint", nem programozási hatáskör, hanem a lecke anyagát és a kérdéseket összeállító oktató feladata. Programozási szempontból ezek csupán - a tapasztalatok birtokában menet közben módosítható - adatként jelentkeznek.

A diagramokból látható, hogy nem rögzítettük azt, hogy hány résztvevője van a kurzusnak, hány bekapcsolási alkalom lehet, mennyi a menetek maximális száma (illetve maximáljuk-e egyáltalán a menetek számát). Ebben a tervezési fázisban ezt még nem célszerű véglegesíteni.

Ugyanakkor hoztunk már olyan döntéseket, amelyek lényegesek az oktatás majdani menete szempontjából. A "bekapcsolás", "hallgató", "lecke" jelű elemek hierarchiájából következően a hallgatóknak nem kell azonos ütemben haladniuk. Előfordulhat, hogy ugyanazon bekapcsoláskor az egyik hallgató pl. a 10. és 11. leckét teljesíti, az utána következő a 5. leckét.



3. ábra



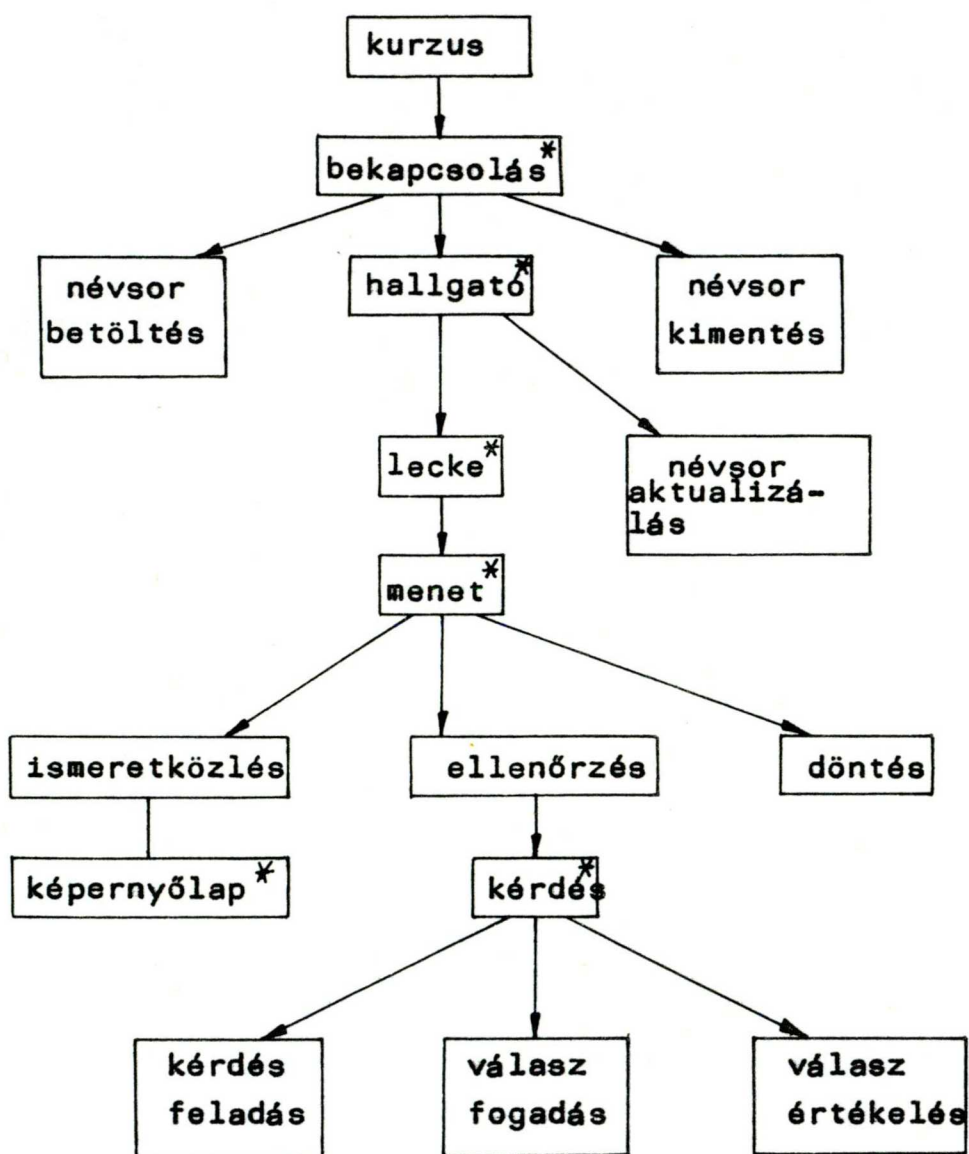
Ha a 3. ábrán adott hierarchiát választottuk volna, akkor egy leckén az összes hallgatónak túl kell jutnia ahhoz, hogy bárki is áttérhessen a következő leckére.

A 2. ábrán látható az is, hogy egy hallgató nem szakíthatja meg a tanulást két menet vagy az ismeretközlés és az ismeretellenőrzés között.

c) Tekintettel arra, hogy a hallgatók előrehaladása nem azonos, gondoskodni kell arról is, hogy a program adminisztrálja, hogy melyik hallgató hol tart a kurzus teljesítésében. E célra legcélszerűbb egy névsor alkalmazása, amely a nevek mellett egy bejegyzést is tartalmaz a teljesített leckeikről. A névsor az egész kurzushoz tartozó információkat tartalmazza, tehát a bekapcsolások között permanensen (mágneses adathordozón) kell tárolni.

d) Az "ismeretközlési funkció" az információkat tartalmazó képernyőlapok sorozatának megjelenítéséből áll (tekintettel arra, hogy más speciális oktatástechnikai eszközökre pl. video, dia a széleskörű használhatóság érdekében nem építettünk).

Egy-egy kérdés kezeléséhez összetevő tevékenységként a kérdés megjelenítése (feladása), a válasz fogadása majd értékelése tartozik (4. ábra).



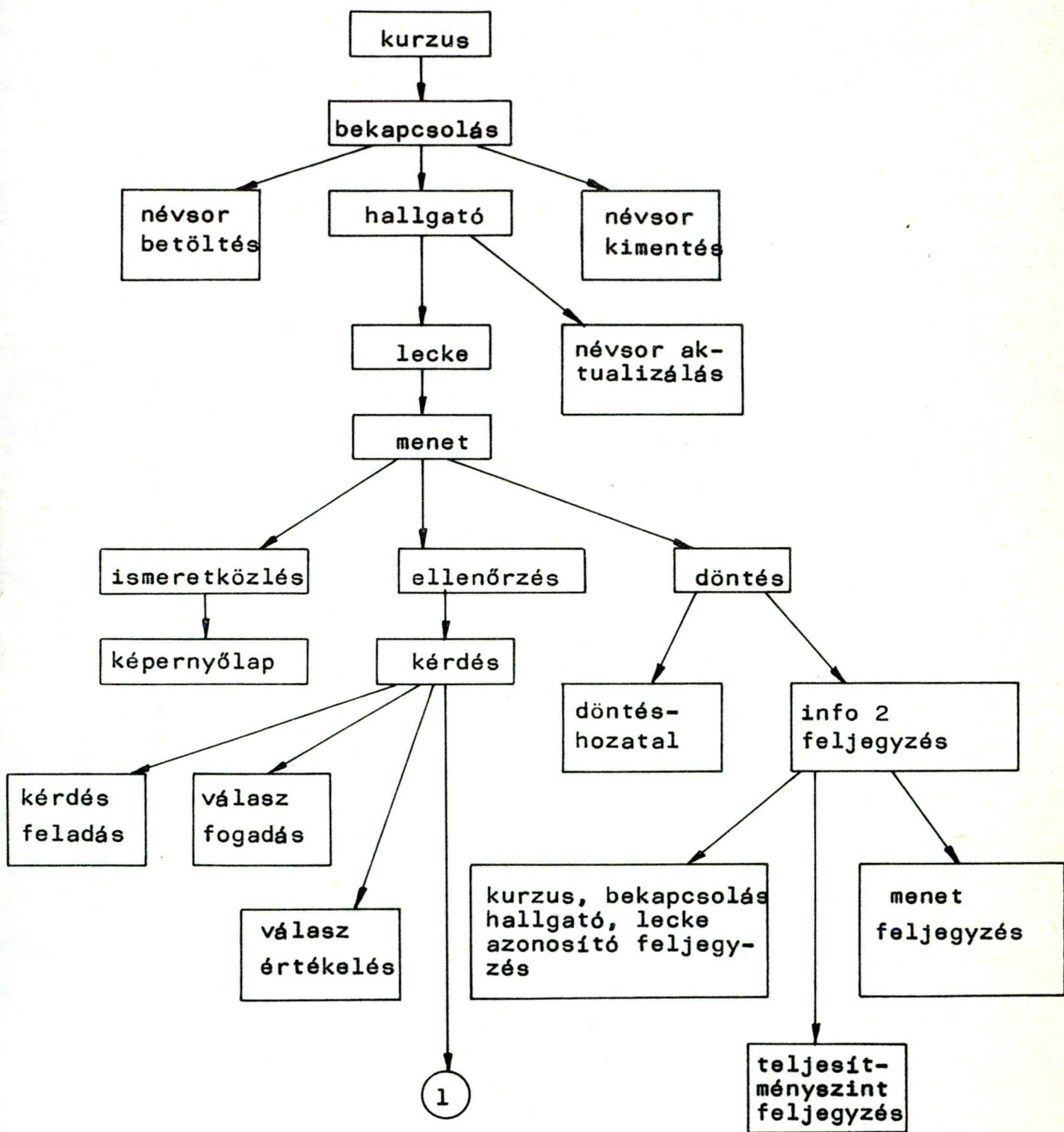
4. ábra

e) A kurzusról a legrészletesebb információkat úgy szerezhethetjük meg, ha hallgatónként és leckénként, azon belül menetenként összegyűjtjük a kérdésekre adott válaszokat és a tudásszintre vonatkozó információkat a későbbi értékelések és elemzések céljaira (5. ábra). Az értékelő elemzés várhatólag a kurzus tananyagának javítására vagy a résztvevő populáció vizsgálatára irányulhat.

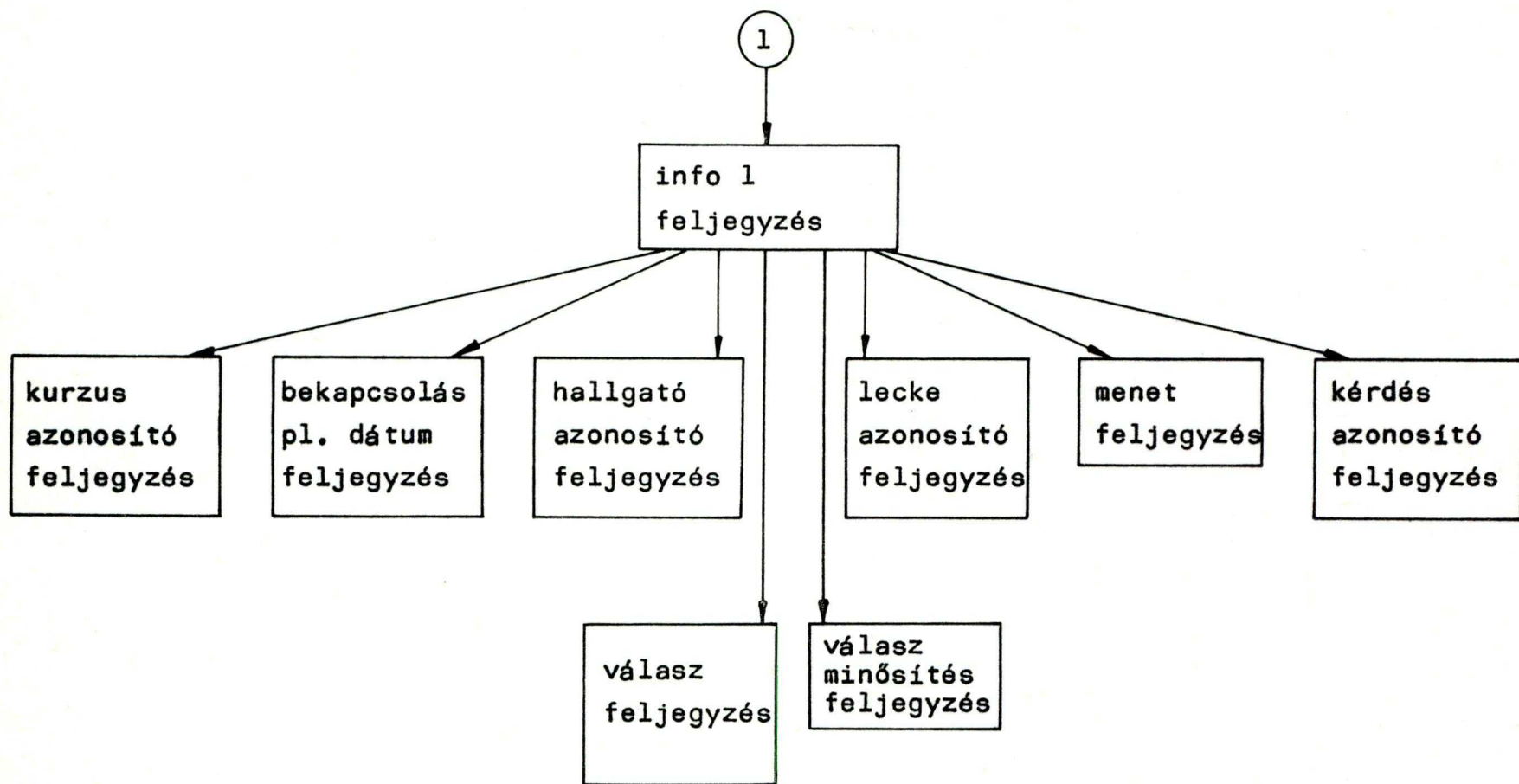
2.1.1.2 Adatmodell elemzés

A 2.1.1.1 részben funkcionálisan elemzett modell általános adatmodelljeként az alábbi Bachmann diagram adódik (6. ábra).

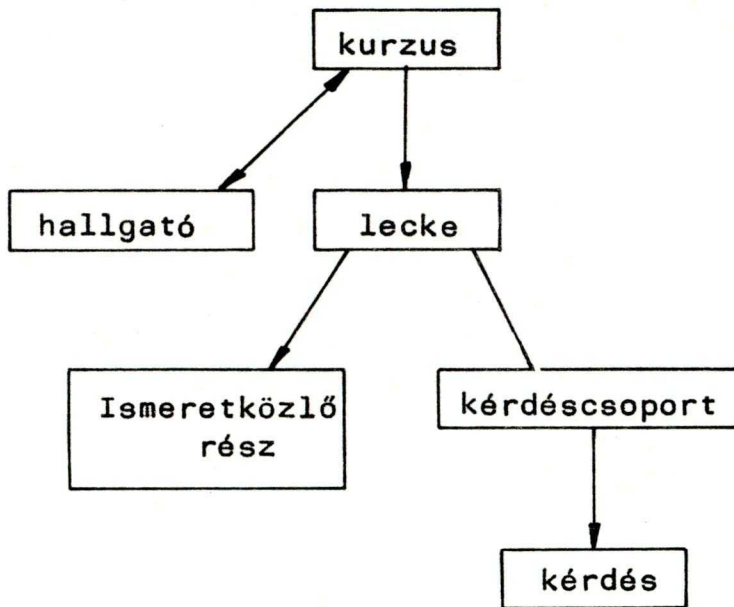
A lecke és "ismeretközlő rész" egyed típusok közötti 1 : N kapcsolattal feltételezzük, hogy ugyanabban a leckében ismeretközlő részként több (pl. nagyvonalúbb és részletesebb) szöveg tartozhat. Ha nincs többféle szöveg, akkor a fenti kapcsolat 1 : 1 típusú, ami speciális esete az általunk feltételezett 1 : N esetnek. Az implementációs modell származtatásához további döntéseket kell hozni a fenti logikai elemek és az implementációs egyed típusok (adathordozó, képernyő) közötti kapcsolatok tisztázására.



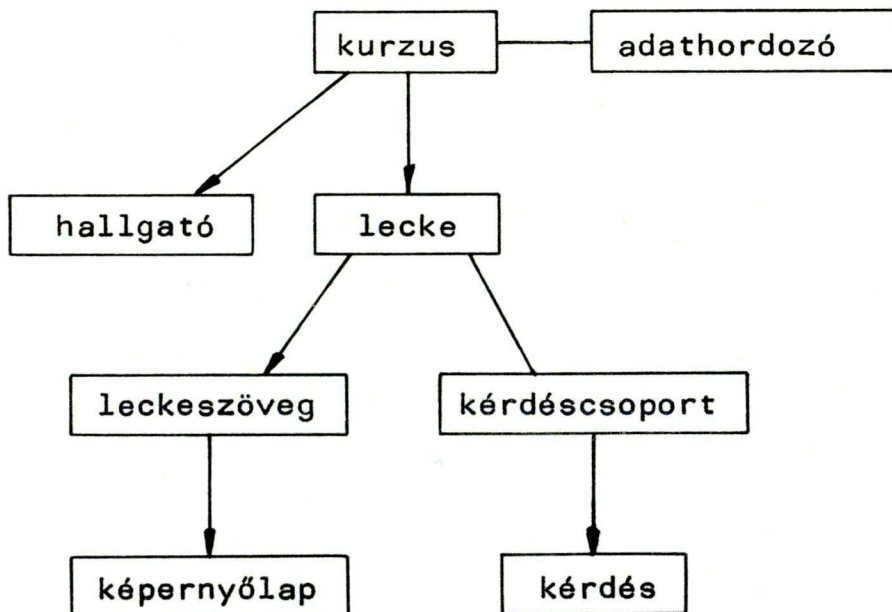
5. ábra



5. ábra



6. ábra

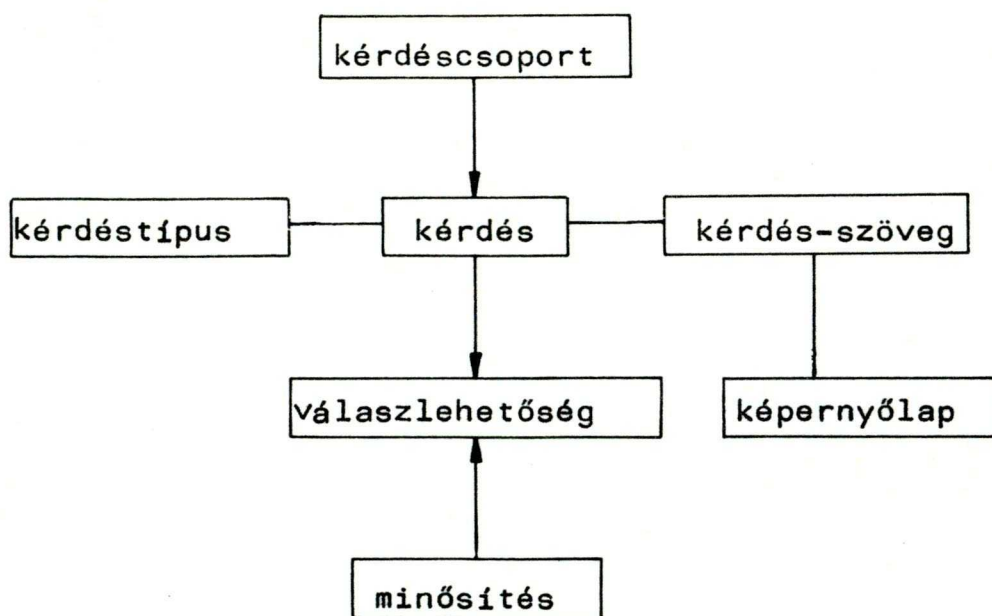


7. ábra

a.) A személyi számítógépek tároló kapacitása nem indokolja több kurzus közös adatrendszerben történő tárolását. Így a kurzus - adathordozó kapcsolat 1 : 1 típusú, a kurzus - hallgató kapcsolat 1 : N típus lesz. (Ezzel a megszorítással lemondunk arról a lehetőségről, hogy ugyanazon hallgató különböző kurzusokban mutatott teljesítményét közvetlenül egybevethessük. Közvetett igényhez elegendő a közvetett út.)

b.) Az ismeretközlő részek és a kérdések megjelenítésére a képernyőt alkalmazzuk, ami az alábbi következményekkel jár. Az ismeretközlő szövegeket vagy folyamatosan írjuk a képernyőre (ún. rollozással) vagy képernyőlapnyi egységekben jelenítjük meg. Céljainknak az utóbbi megoldás felel meg jobban; elsősorban azért, mert így nem a gépi kiírási sebesség, hanem a hallgatói döntés határozza meg, hogy mennyi ideig marad a képernyőn egy-egy tananyagrészt. A szövegek fizikai egységekre (képernyőlapokra) való felosztása ugyanakkor feltételezi a megfelelő logikai egységekre (bekezdésekre) történő tagolását is. Ez a szempont a tananyag előkészítésénél betartandó (nem mindig kényelmes) szabályként jelentkezik (7. ábra).

c.) A kérdéseket úgy célszerű megfogalmazni, hogy a szövege egy képernyőlapon elférjen. A több képernyős kérdések technikailag is áttekinthetetlenek, mivel egy időben a szövegnek csak egy részét látná a hallgató. Emellett ellentmondana annak a pedagógiai alapelvnek, hogy a megválaszolandó kérdés legyen világos, tömör, szabatos és rövid. Egy kérdéshez több válaszlehetőség tartozik, minden válaszlehetőséghez egy minősítés (pl. pontérték), mint a válasz helyességének értékmérője. Ezzel a közelítésmóddal a legáltalánosabb kapcsolatot fogalmaztuk meg a kérdés és a válasz között, amely magába foglalja a különböző kérdéstípusokat és értékelési stratégiákat (8. ábra).



8. ábra

A válaszlehetőség és a minősítés között az alábbi összefüggések lehetnek.

α .) Egy kérdésre egyetlen jó válasz van.

Ekkor a helyes válaszhoz a "jó" minősítés, az összes ettől eltérő válaszlehetőséghez a "hibás" minősítés tartozik.

β .) A kérdésre több jó válasz adható (esetleg jó, kevésbé jó, legjobb), az összes többi válasz hibás. A minősítés hozzárendelés egyértelmű.

γ .) A kérdéshez egy vagy több jó válasz, egy "nem tudom" típusú ún. neutrális válasz és hibás válaszok tartoznak; a minősítés itt is egyértelmű.

A minősítésként alkalmazott pontérték - hozzárendelés alapvetően két féle lehet:

- tudásértékelő; amikor a jó válaszáért pozitív pontszám, a hibás válaszáért $\emptyset$ pont jár;
- hibabüntető; amikor a jó válaszáért pozitív pontszám, a hibás válaszáért negatív pontszám, neutrális válaszáért (ha van) $\emptyset$ pont jár.

Mint látható, mindhárom fenti kérdéscsoporttal és mindkét válaszártékelési stratégiával összhangban van a fenti Bachman diagram részletünk. Modell szinten azonosnak tűnik a technikailag megkülönböztetendő három-

féle - feleletalkotó, feleletkiegészítő és feleletválasztó - kérdéstípus is.

A számítógépi lehetőségeket figyelembe véve a legnehezebb a feleletalkotó kérdések kezelése. Itt a fő nehézséget az jelenti, hogy a választ az intelligens, természetes nyelvi mondatokból mechanikusan, algoritmikusan kell kihámozni. Még a nagyteljesítményű számítógépeken sem sikerült megnyugtatóan megoldani e kérdés típus kiértékelését, s ez a tény nem a számítástechnikai eszközök gyengeségét, mint inkább a természetes nyelvek gazdagságát, intellektualitását igazolja.

A feleletalkotó kérdések kezelése nem csak a válaszkértékelő algoritmusban okoz gondot, hanem a tananyag készítői számára is - a helyesnek elfogadható illetve hibásnak minősítendő válaszok szétválasztásában [17] .

Megfigyelhető, hogy a gyakorlatban rohamosan szorítják ki a feleletválasztó kérdések a feleletalkotó típusokat, ott is, ahol a "kiértékelés" nem gépi, hanem emberi döntéssel történik (tesztlapok). A feleletválasztó kérdések (nem mindig szerencsés) előnybe részesítését általában az egységesebb, objektívebb és gyorsabb kiértékelési lehetőséggel indokolják.

A feleletkiegészítő kérdések kezelése alig bonyolultabb a feleletválasztó kérdések kiértékelésénél, mégis gyakori jelenség, hogy azokat a kérdéseket is átfogalmazzák feleletválasztó típusúra, amelyek a természetes gon-

dolkodás szerint feleletkiegészítő típusúak lennének. Modellünk szempontjából a feleletválasztó és a másik két kérdéstípus között ott van lényeges eltérés, hogy míg a feleletválasztó kérdések esetén a válaszlehetőségek halmaza véges és korlátos, addig a másik két típusnál gyakorlatilag végtelennek tekinthető; következésképp a válaszelemzés más programozástechnikai megoldásokat kíván. Ennek megfelelően a kérdésnek igen fontos attribútuma lesz, hogy feleletválasztó típusú vagy sem.

2.1.1.3 Tananyag és oktatóprogram kapcsolata

Oktatóprogramokat kétféle implementációban lehet készíteni.

Egyik esetben a tananyag és a vezérlő szoftver programozástechnikailag egybeépül; azaz az oktatott tananyag beépül a program forrásszövegébe. Ez a tény alkalmazási szempontból azt jelenti, hogy az ilyen tananyag nem használható másképp (csak az adott vezérlő szoftverrel együtt) és a kezelő szoftver sem alkalmazható más konkrét tananyag oktatására, legfeljebb a program átírásával. Ez a megoldás a programozói és oktatói munka integrációját tételezi fel, nevezetesen azt, hogy a tananyagkészítésre vállalkozó egyszemélyben "jó" pedagógus és "jó" programozó legyen.

Ha figyelembe vesszük a számítógéppel támogatott oktatás tömegméretű igényeit és a pedagógusi munka alkotó jellegét, beláthatjuk, hogy ez az implementációs stílus csak egyedi eredményeket hozhat, távlatilag csak zsákutcába vezethet.

A másik megoldási módnál programozástechnikailag elkülönítjük az oktatási folyamatot vezérlő szoftvert a tananyagoktól. Az önállósult, "általános célú" oktatóprogram önmagában semmire sem alkalmas, működni is csak a megfelelő szabályoknak eleget tevő tananyaggal együtt képes, mivel az oktatásnak csak az "algoritmusát" tartalmazza.

Az előző implementációhoz képest lassabban gyümölcsöző megoldás távlatilag a következő fő előnyöket nyújtja:

- Az egyszer jól kidolgozott általános célú oktatóprogram egy sor különböző konkrét tananyaghoz felhasználható anélkül, hogy programírással és a mindig kritikus belövással foglalkozni kellene. Ha a program elegendően általános, az is elérhető, hogy különböző esetekben (pl. paraméterezéssel) különböző oktatási vagy válaszértékelési stratégiákat alkalmazzon.
- Lehetővé válik az ésszerű munkamegosztás, nevezetesen az, hogy a tananyagokat ne programozók, hanem oktatók készítsék.

Ahhoz, hogy lehetőségből valóság legyen, természetesen biztosítani kell azt is, hogy a tananyagokat egyszerű, könnyen kezelhető formában lehessen megadni.

Igy elkerülhetők azok a - sajnos gyakori - mellékhatások, amelyek abból adódnak, hogy a programozók próbálnak oktatók is lenni az oktatók pedig programozóként is helytállni.

A tananyagok és az oktató szoftver elkülönítése felvet olyan feladatokat is, amelyek az egybeépített megoldásnál fel sem merülhetnek.

- Technikailag is meg kell oldani azt, hogy a tananyagokat az oktatók készíthessék, azaz meg kell határozni azt a formát, ahogyan a tananyagokat az oktató szoftver számára elő kell készíteni. Az előkészítés ne (vagy csak minimális mértékben) tételezzon fel számítástechnikai ismereteket.
- Egyszerű lehetőséget kell biztosítani a géppel tárolt tananyagok módosítására.
- Meg kell oldani a tananyagok nyilvántartását, a számítógéppel folytatott oktatás szervezését, menedzselését, továbbfejlesztését.

Röviden összefoglalva meg kell oldani azokat a részfeladatokat, amelyek a munkamegosztásból és széleskörű használatból adódó korszerű számítástechnikai alkalmazásoknál jelentkeznek.

2.1.2 Rendszerterv

Az eddigi elemzésekben adódóan rendszerünk két fő összetevőből épül fel.

- az általános célú oktatói-szerzői rendszer programjaiból;
- a BASIC nyelv szakalkalmazói szintű megtanításához szükséges tananyagokból.

Ez a felépítési mód összhangban van [22]-ben talált megoldások egy részével.

Maga az általános célú oktatói-szerzői programrendszer négy funkcionális egységből áll:

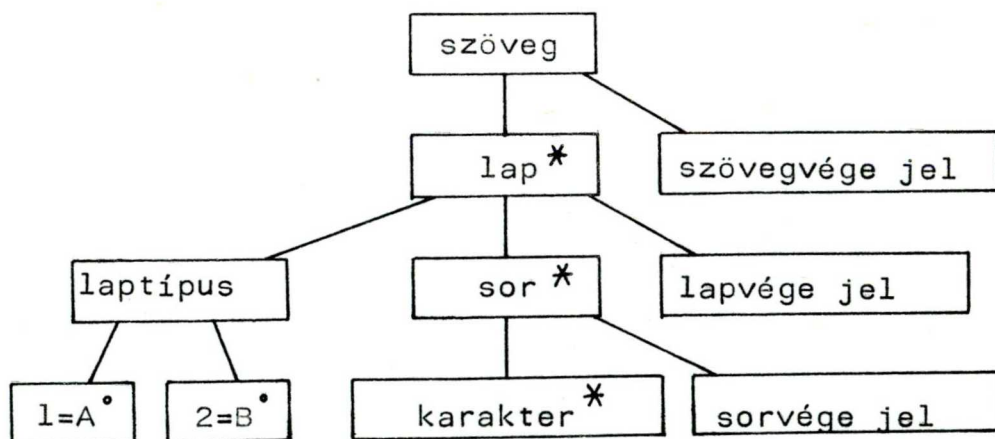
- a tananyagok előkészítését és módosítását segítő "szerzői" szoftver, amelyet a továbbiakban "TAFEL"-lel rövidítünk;
- az előkészített, géppel tárolt tananyagok kezelését, a gépi "oktatást" elvégző szoftver, amelyet röviden "OKTATO"-nak nevezünk;
- az oktatási folyamatra vonatkozó (az OKTATO által összegyűjtött) események elemzését, kiértékelését végző szoftver, amelyet röviden "ELEMZŐ"-nek nevezünk;
- a "technikai" tevékenységek (archiválás, helyreállítás, másolatkészítés) elvégzését biztosító szoftver elem, amelyet "INST"-nek nevezünk.

2.1.2.1 Adatrendszer, fájl struktúrák

A tananyagok gépi tárolása az alábbi módszer szerint történik.

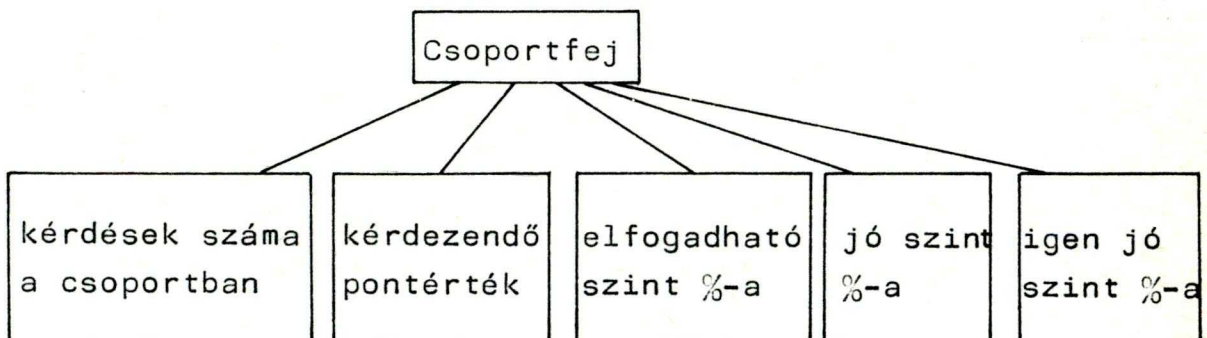
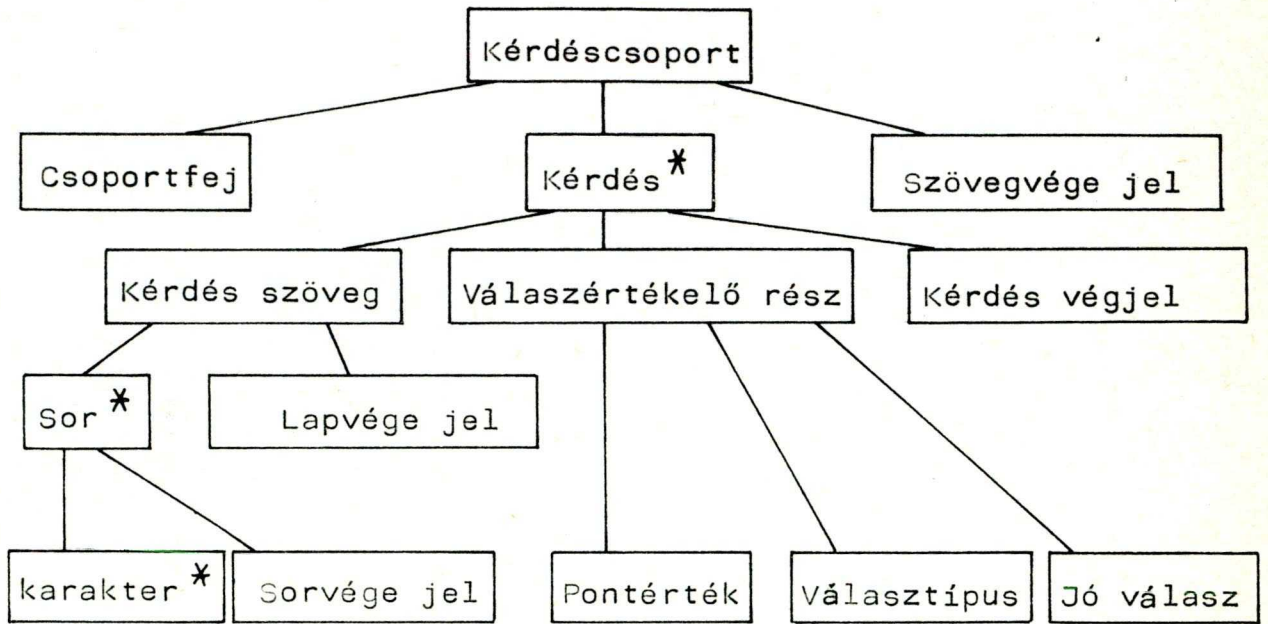
A kurzus minden leckéjéhez egy fájl-pár tartozik. Az egyik fájl az ismeretközlő részeket, a másik a kérdéseket tartalmazza. Az OKTATO program reális választidejének eléréséhez az ismeretközlő részeket és a kérdéscsoportot "takarékos" belső ábrázolási formában kell tárolni.

Az ismeretközlő részek tárolására alkalmazott adatstruktúra diagramját a 9. ábrán láthatjuk, értelmezésével a belső ábrázolási formát kialakító TAFEL program leírásánál foglalkozunk. A kérdéscsoport társtruktúráját a 10. ábra mutatja, értelmezését szintén a TAFEL programnál találjuk.

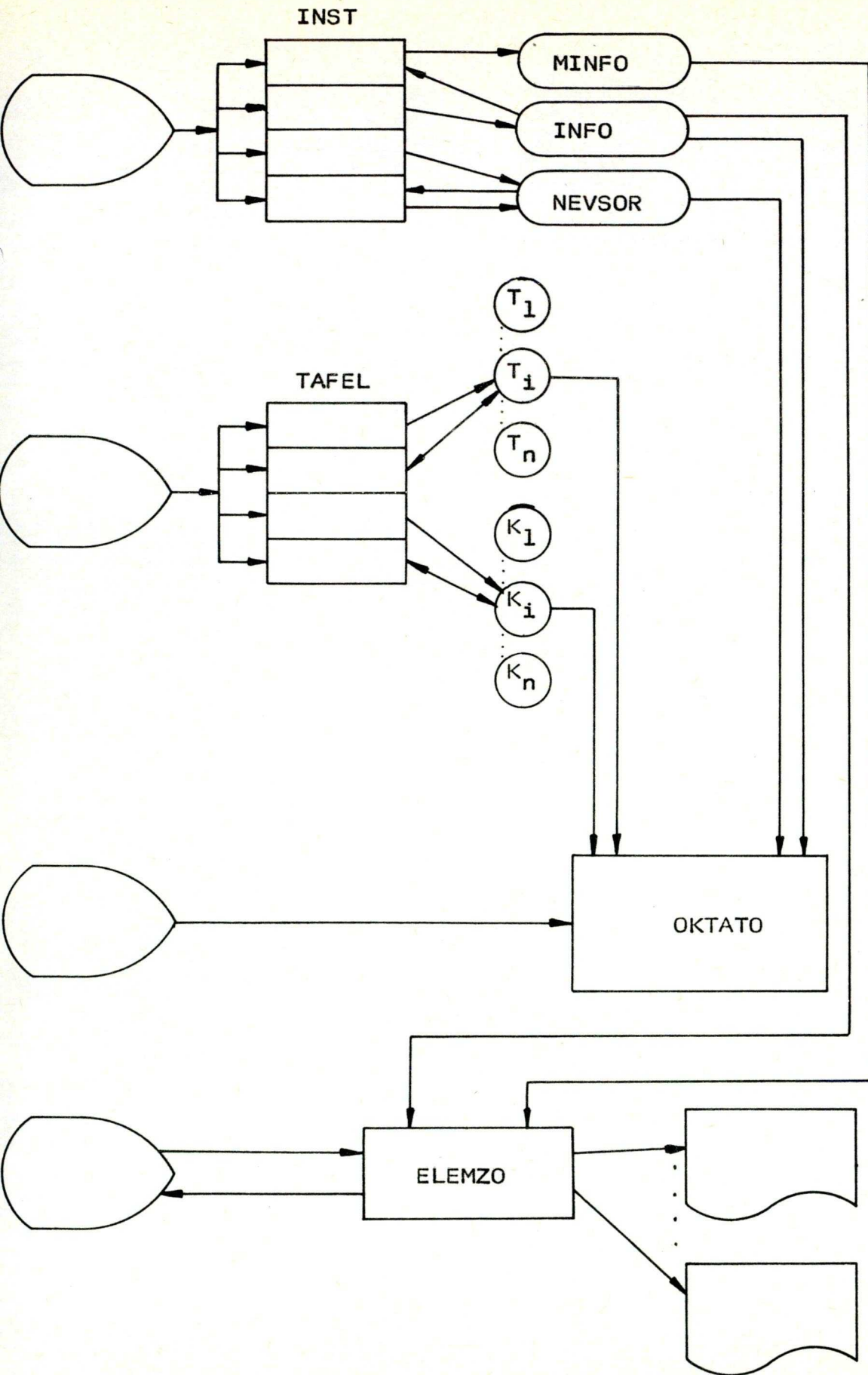


9. ábra

Kérdéscsoport
társtruktúrája



10. ábra



11. ábra

A rendszer áttekintő adatfolyamatábráját a 11. ábra tartalmazza. Itt $T_1, \dots, T_i \dots T_n$ -nel a kurzus-hoz tartozó ismeretközlő részek; $K_1, \dots, K_i \dots K_n$ -nel a megfelelő kérdéscsoportok fájlját jelöltük. A NEVSOR fájl a kurzus hallgatóira vonatkozó információkat tartalmazza. Minden hallgatóhoz egy bejegyzés tartozik, az alábbi jellemzőkkel:

- a hallgató neve
- azonosítója (kódja)
- az előrehaladást jelző szám, amelynek értéke a kurzus azon leckéjének sorszáma, amit a hallgató már teljesített.

A névsor fájlt az oktató hozza létre (az INST program segítségével) a kurzus indításakor; megadva a résztvevők nevét és azonosítóját. A továbbiakban az OKTATO program itt vezeti hallgatónként az előrehaladást, kettős célzattal. Egyrészt az oktató így bármikor megtudhatja a számítógéptől, hogy a hallgatók hol tartanak a kurzus teljesítésében (12. ábra); másrészt így biztosítottuk azt, hogy egy hallgató bejelentkezésekor automatikusan a számára következő leckét kapja az OKTATO programtól.

Az INFO fájl bejegyzései az oktatási folyamat során keletkeznek. Feljegyzésre kerül minden feltett kérdésnél



HALLG.SZAM=16 LECKE SZ.=13

KOD NÉV LECKE

501	Berg Maréit	2
503	Csáki Miklós	3
504	Dóra Tóth Anikó	3
505	Halász István	4
506	Hangácsai Viktor	3
507	Hevesi Mariann	4
508	Karchesz Ágnes	3
509	Kasza Beáta	2
510	Kocsis Beáta	4
511	Komornik Attila	3
512	Lantos Ildikó	3
513	Mészáros Eleonóra	4
514	Pálfi Andrea	3
515	Sándor József	5
516	Schmelzer Éva	4
517	Baksay Ágnes	3

12. ábra

INFORMACIÓS FILE /RÉSZLET/

HK. LEC KOR KSZ VAL ERT TELJ%

U11	00	1	0	C	-1	
U11	000	1	00	EE	0	
U11	0000	1	2	A	1	
U11	0000	1				43.7
U11	0000	NNN	4	D	0	
U11	0000	NNN	1	B	1	
U11	0000	NNN	7	DD	1	
U11	0000	NNN	0	D	1	
U11	0000	NNN				64.4
UU00	0000	1	U	E	0	
UU00	0000	1	U	EE	-1	
UU00	0000	1	7	A	1	
UU00	0000	1				5.5
UU00	0000	NNN	0	B	1	
UU00	0000	NNN	00	DD	1	
UU00	0000	NNN	4	DE	0	
UU00	0000	NNN	U	E	0	
UU00	0000	NNN	1	B	1	31.2
UU00	0000	NNN	00	EE	1	
UU00	0000	NNN	0	EE	0	
UU00	0000	NNN	0	B	1	
UU00	0000	1	11	C	1	73.3
UU00	0000	1	4	C	1	
UU00	0000	1				100.0

13. ábra

- a hallgató azonosítója;
- a lecke azonosítója;
- a kérdés azonosítója;
- hányadik menetben;
- milyen válasz érkezett rá;
- a választ hogyan minősítettük.

A menetek végén ezen kívül még feljegyzésre kerül az is, hogy a hallgató milyen teljesítményt ért el (százalékosan). Az INFO fájl egy részletének kiíratása a 13. ábrán látható.

2.2 A programok funkciója és használata

2.2.1 A TAFEL program

A TAFEL program rendeltetése az, hogy

- a leckék ismeretközlő részeinek illetve a kérdéscsoportoknak az OKTATO program által feltételezett belső ábrázolási formáját kialakítsa;
- a belső ábrázolási formában lévő leckék módosítását, javítását biztosítsa.

A TAFEL adataegységként a képernyőlapokat, ezen belül a képernyősorokat kezeli. Bár a tananyagot a számítógép mellett ülve is ki lehet dolgozni, nagymértékben

megkönnyíti a munkát, ha a 14. és 15. ábrán látható szöveglapokon és kérdéslapokon előre megtervezzük a leckéket.

Mindkét formalap rovatolt; a képernyő felosztásnak megfelelően 20 sort és soronként 32 pozíciót tartalmaz. A képernyő teljes (24 soros) tartományát nem használhatjuk fel, mert a válaszkezeléshez is szükség van 2-3 sorra. A formalapok valójában képernyő-tervek, így olyan tartalommal kell kitölteni a rovatokat, amelyet a képernyőn majdan látni kívánunk (16. ábra).

Az ismeretközlő részek kidolgozásánál figyelembe kell venni, hogy a tanuló mindig egy képernyőlapnyi szöveget lát egyszerre. Ennek megfelelően az ismeretközlést, logikai-gondolati egységekre (bekezdésekre) kell tagolni. A tagolás kialakításához és az esztétikus képernyőelhelyezés megtervezéséhez igen nagy segítséget nyújt, ha a tananyagot formalapokon szerkesztjük meg. A formalapok használata munkamegosztási szempontból is előnyös lehet.

Egyrészt a számítástechnikai ismeretekkel vagy számítógéppel egyáltalán nem rendelkező (a számítógép kezelésétől idegenkedő) oktatók számára is fenntartjuk a tananyagkészítés lehetőségét, ha megszervezzük, hogy a "gépkezelő" szintű laboráns a számítógépbe billentyűzze a formalapokon lévő tananyagaikat. Másrésztől rövidebb ideig kötjük le a számítógépet, ha a tananyagok előzetes kidolgozása formalapokon már megtörtént.

Szöveglap

Tananyag azonosító:

--	--	--	--	--

Lapszám:

--	--

Laptipus:

☐[illegible]

Dátum:

Készítette:

Tipus = 1

--	--	--	--	--

[illegible]Pontérték:

--	--

Készítette:

Szöveglap

Tananyag azonosító: MIR11

Lapszám: 03

Laptípus: A

1																			
2																			
3																			
4			N	y	á	r	i		n	a	p	n	a	k					
5														a	k	k	o	n	y
6														u	l	a	t	á	n
7			M	e	g	á	l	l	é	k		a		k	a	n	y	a	r
8														g	ó				
9																T	i	s	z
10			O	t	t	,		h	o	l		a		k	i	s		T	u
11																s	i	e	t
12																	b	e	l
13			M	i	n	t		a											
14																			
15																			
16																			
17																			
18																			
19																			
20																			

Dátum: 1985. III. 27.

Készítette: BAKO ISTVÁN

A felhasznált, bebillentyűzött formaközpontokat is érdemes archiválni részben a későbbi módosítások egyszerűsítésére, részben a megfelelő tananyag eszköztár áttekintésének egyszerűsítésére.

A TAFEL programnak az alábbi funkciói vannak:

- új anyagrész (ismeretközlő rész vagy kérdésrész) felvitele,
- meglévő tananyag betöltése mágneslemezről
- tananyag kiírása a képernyőre
- tananyag módosítás
- tananyag felírása mágneslemezre

Az új anyagrészek felvitele illetve a tananyag módosítás képernyőlaponként, ezen belül sor-egységenként történik.

Az ismeretközlő részek és a kérdések kezelése hasonló logika szerint történik, így csak az ismeretközlő részekkel foglalkozunk részletesebben.

Új lecke feladásakor az "új anyagrész felvitele" funkciót kell kérni az étlapról.

A program laponként kéri a tananyagot. Minden laphoz "laptípus" rendelhető; ezzel biztosítjuk az "OKTATO" programnál leírt többszintű ismeretközlés lehetőségét.

A program soronként kéri a lapok tartalmát. Ha a bebillentyűzött jelsorozat hibátlan, akkor elfogadja a következő sorként, ha hibás, akkor újra kéri ugyanazt a sort.

Kétféle hibalehetőség fordulhat elő;

- a sor túl hosszú (32-nél több jelet tartalmaz)
- a sor illegális jelet tartalmaz; amit a program vezérlőjelként értelmez a későbbi működés során (# , " , @ , &).

Ha az utolsó lap feladása megtörtént, laptípusként egy * jelet kell beütni. Ennek hatására visszatér a főmenü.

"A tananyag felírása mágneslemezre" funkció hatására a mágneses adathordozón rögzítésre kerül az eddig operatív memóriában tárolt tananyagrészt.

Ha valamely létező tananyagot meg akarunk tekinteni, akkor a "meglevő tananyag betöltése mágneslemezről", majd a "tananyag kiírása képernyőre" funkciókkal képernyőlaponként megtekinthetjük a tárolt anyagrészt.

Minden képernyőlap után várakozik a program, s csak akkor hozza a következő lapot, ha - a tartalmát elolvassa - lenyomjuk a RETURN gombot.

Létező tananyag módosítása, javítása esetén először a "meglevő tananyag betöltése mágneslemezről", majd a "tananyagmódosítás" funkciókat kell kérni.

A program laponként megjeleníti a tananyagrészt és lehetővé teszi a javítást a következő módon.

A "Sor beszúrás (i/n)" kérdésre adott igenlő válasz esetén megkérdezi a beszúrandó sor sorszámát és a sor tartalmát.

Ha a sorszám és a tartalom hibátlan, elvégzi a képernyőlap tartalmának módosítását, majd az újabb aktuális állapot jelenik meg a képernyőn.

A "beszúrás (i/n)" kérdésre adott nemleges válasz esetén a "sortörlés (i/n)" kérdés következik. Igenlő válasz esetén megtörténik a kijelölt sorszámú sor törlése és a képernyőtartalom felfrissítése.

Nemleges válasz esetén a "sormódosítás (i/n)" kérdés következik. Az erre adott igenlő válasz esetén kéri a cserélendő sorszámot és a sor új tartalmát. Ha hibátlan adatokat kapott, megtörténik a módosítás és a képernyőtartalom felfrissítése.

Ha a "sorcsere i/n" kérdésre is nemleges választ adtunk, akkor a program áttér a következő lapra, ahol hasonló módon aktualizálható a képernyőlap tartalma.

Az utolsó létező lap kezelése után a "folytató lap felvétele" kérdés következik. Nemleges válasz esetén a funkció lezárul és a főmenübe tér vissza a program.

Igenlő válasz esetén új lapokat lehet feladni a tananyag kiegészítéseként, az "új anyagrész felvétele" funkcionál leírt működés szerint. A módosítások végzetével a "tananyag felírás mágneslemezre" funkciót kell végrehajtani, hogy az operatív memória tartalmán átvett javítások a háttér-tárolón permanensen is megőrződjenek.

A "tananyag felírás mágneslemezre" funkció vég-

rehajtásához meg kell adni a névkonvenciónak megfelelő adatcsoport(fájl)-nevet.

A véletlen felírási hiba ellen a programba részleges védelmet építettünk be.

2.2.2 Az oktató program

Az oktatóprogram működtetése feltételezi, hogy a névsor fájl és a kurzushoz tartozó tananyagok információközlő és kérdés fájljai a megfelelő (az INST illetve a TAFEL által kialakított) struktúrában, mágneslemezen vannak. Ugyanúgy feltételezi azt is, hogy az INFO fájlban rendelkezésre állnak az eredményfelvételi bejegyzésekhez szükséges rekordok.

Indításkor a program vagy hallgatói vagy tanári üzemmódba kerül. Hallgatói üzemmódban a program először is beolvassa a névsor fájlt az operatív memóriába.

a.) Ezen a ponton várja a hallgatói bejelentkezést. A bejelentkezés a hallgató azonosítójával (kódjával) történik. Ha a program felismerte a hallgató azonosítóját, akkor közli például, hogy

b.) "KOVÁCS LAJOS számára a 6. lecke következik" majd beolvassa a megfelelő (jelen esetben a 6.) leckéhez tartozó ismeretközlő részt és kérdéscsoportot.

megkezdí az ismeretközlést, majd a kérdések feladását. Ha a tudásszint nem volt kielégítı, akkor újra ismeretközlés, majd megint kérdések következnek. Ha meghatározott (beállítható) lépésszámon belül nem sikerül a kívánt szinten megválaszolni a kérdéseket, a program "forduljon humán oktatóhoz" üzenet után újabb hallgató bejelentkezését várja, majd az a.)-tól folytatódik. Ha a kérdésekre adott válaszok alapján a tanuló elérí a kívánt tudásszintet, áttérhet a következı leckére. A "Kéri a következı leckét" kérdésre adott igenlő válasz esetén újra a b.)-vel jelölt résztől folytatódik a program, de most már a következı - jelen esetben a hetedik - leckével. Ha a hallgató nem kívánja azonnal folytatni a tanulást és nemleges választ ad, akkor a "Van még vállalkozó" kérdésre adott igenlő válasz után új hallgató jelentkezheth be, Kovács Lajos pedig majd a következı bejelentkezésekor kapja meg a hetedik leckét. Ha nincs több vállalkozó, akkor a megfelelı információs állapotok rögzítése után a program befejezi működését.

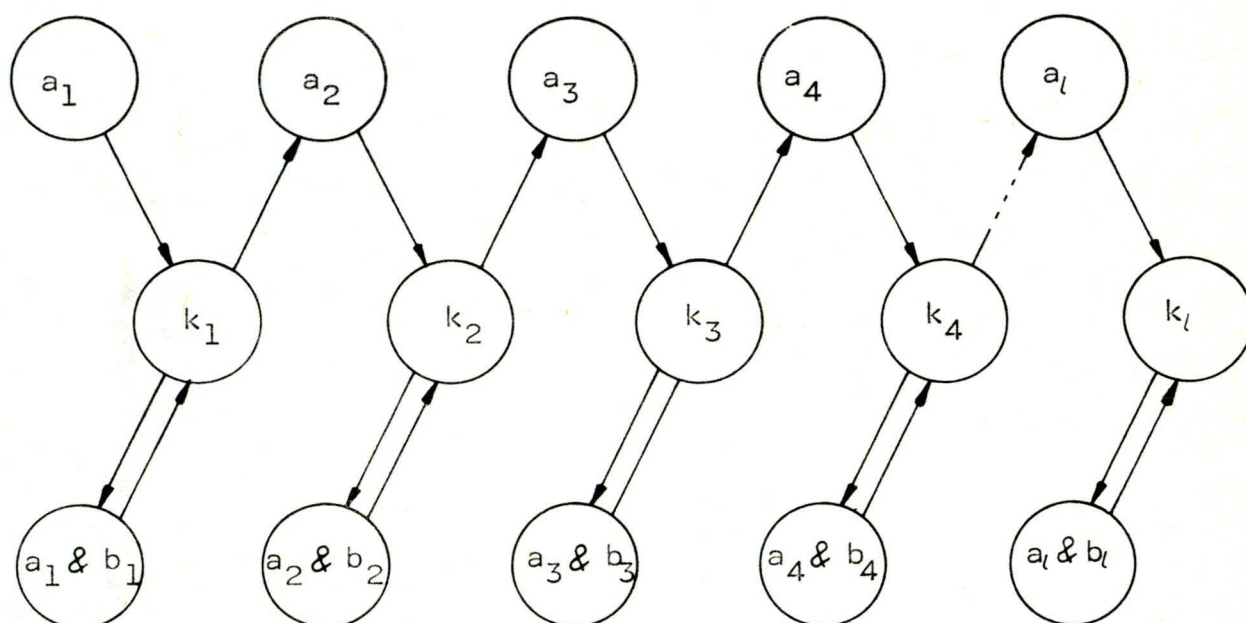
A tanári üzemmódban az OKTATO program a következı tevékenységek elvégzését biztosítja.

- Névsor fájl tartalmának kiírása a képernyőre.
- Az információs fájl tartalmának megtekintése.
- Az információs fájl tartalmának kimentése (ezzel párhuzamosan új eredményfelvételi bejegyzések biztosítása).

Az első két funkció informatív az oktató számára, míg a harmadikra akkor lehet szükség, ha egy hosszabb hallgatói használat során az információs fájl betel-
léssel fenyeget. A program ismeretközlő része bizto-
sítja az úgynevezett többszintű ismeretközlés lehető-
ségét is. Természetesen ehhez a tananyagokat eleve en-
nek megfelelően kell kidolgozni. Többszintű ismeretköz-
lés tervezése esetén a tananyagnak több "részletezett-
ségi szintjét" kell meghatározni. A legszűkebb, mini-
mális ismeretanyag az "A" típusú lapokra, a bővítő is-
meretkör a "B" típusú lapokra kerül, stb.

Az első menetben (amikor a hallgatót először "tanítja"
a gép a megfelelő leckére) csak a minimális ismereta-
nyagot közli vele (csak az A típusú lapok tartalmát ír-
ja ki a képernyőre). Ha az ismeretellenőrzési fázison
ennek birtokában a hallgató nem jutott túl, akkor a
második ismeretközlési menetben már az "A" és "B" tí-
pusú lapok egyaránt közlésre kerülnek. Ha még ez sem
elegendő, akkor az "A", "B" és "C" típusú lapok megje-
lenítése következik a következő menetben, egészen ad-
dig, amíg a hallgató a megfelelő tudásszintet el nem éri
vagy az összes részletezési szintet kimerítette. Ez az
általános, többszintű ismeretközlést biztosító algorit-
mus változtatás nélkül alkalmazható abban az esetben
is, ha a kidolgozott tananyag egyszintű. Ekkor ugyanis
(mivel nincs "B", "C" stb. típusú lap) a meghatározott

menetszámszor mindig ugyanaz az ismeretanyag kerül közlésre. A 17. ábrán egy kétszintű ismeretközlő lapokból álló kurzus feldolgozásának irányítási stratégiája látható.



17. ábra

Itt a_i az i -ik lecke ismeretközlő részének szűkebb változatát (A típusú lapjait);
 k_i az i -ik leckéhez tartozó megértést segítő és ellenőrző kérdések sorozatát;
 $a_i \& b_i$ pedig az i -ik lecke ismeretközlő részének bővebb változatát (A és B típusú lapjait) jelöli.

Az ábrán a_i -vel jelölt részeket egyszer járja be a hallgató, az a_i & b_i részeket pedig elvileg annyiszor, ahányszor szükséges a kívánt szintű elsajátítás érdekében. Ha az a_i & b_i részben a hallgató n -szer járt ($n > 0$), akkor a k_i ellenőrző részben $(n+1)$ -szer halad át.

Egy k_i kérdéscsoporton való áthaladásnak kétféle kimenetele lehet. Ha a hallgató teljesítménye eléri a tananyagkészítő által előírt szintet, akkor rátérhet a kurzus következő leckéjére (azonnal vagy egy későbbi időpontban). Ha viszont nem éri el a kívánt szintet, újra az ismeretközlő funkció lép működésbe, majd a hallgatónak újra kérdésekre kell válaszolnia.

A hallgató a kurzust akkor teljesítette, ha a kurzus utolsó (17. ábrán k_1 -lel jelölt) kérdéssorozatát is sikeresen megválaszolta.

A program kérdező részét úgy alakítottuk ki, hogy elkerüljük a "formális megtanulást", vagyis azt, hogy a tanuló a válaszra próbáljon emlékezni, ahelyett, hogy a kérdését akarná megválaszolni. Ennek megfelelően egy ellenőrzési ponton a program a rendelkezésre álló kérdéskörből véletlenszerűen választja ki a felteendőket. A választás visszatevés nélküli mintavétellel történik, így csak akkor kerülhet elő egy korábbi kérdés, ha már a csoport minden eleme felhasználásra került.

Ez a megoldás feltételezi, hogy a leckékhez viszonylag nagyszámú kérdés tartozik, legalább négyszer-öttször

annyi, mint amennyit egy ismeretellenőrzés alkalmával felhasználunk. Az ismeretellenőrzés menete a következő. Minden leckéhez (kérdéscsoporthoz) hozzá van rendelve egy számérték, amely előírja, hogy hány pontnyit kell kérdezni egy-egy ismeretellenőrzési alkalommal. Az egyes kérdésekhez is tartozik pontérték (amennyi a helyes megválaszolásért adandó); ami természetesen széles skálán mozoghat a kérdések nehézségi fokától függően. Egy véletlenszerűen kiválasztott kérdés megválaszolása után a gép eldönti, hogy kérdezett-e már annyi érték-pontnyit, mint amennyi a leckéhez elő van írva. Ha még nem, újabb kérdés kiválasztás történik, ha már igen, akkor a feltett kérdések potenciális pontértéke és a tényleges elért pontértékekből egy teljesítményszázalék kerül meghatározásra. A számított teljesítmény és a tananyagkészítő által előírt "minimális teljesítményszint" összehasonlításából adódik az ellenőrzés eredményessége, s ennek megfelelően a hallgató továbbhaladási iránya.

Az egyes kérdésekre adott válaszok pontban kifejezett minősítésére többféle stratégia alkalmazható. A program algoritmusa lehetőséget biztosít a választásra, a konkrét értékelési stratégia az installációs paraméterek értékének beállításával rögzíthető. E paraméterek az alábbiak (zárójelben az alapértelmezés látható):

- hibás válasznál a pontlevonás mértéke (0.5)
- neutrális (nem tudom) válasz van-e a

kérdésekhez

(1)

Az alapértelmezés szerinti paraméterállásnál a program:

- helyes válasznál megadja a kérdéshez tartozó pontértéket (ennyivel növeli az aktuális pontszámot);
- neutrális válasznál változatlan marad az aktuális pontszám (ezért a válaszáért 0 pontot ad);
- hibás válasznál csökkenti az eddig elért aktuális pontszámot a jó válaszáért adandó pontszám felével.

Ha például az első paramétert 0-ra állítjuk, akkor hibás válaszáért nincs pontlevonás ($0 \times X = 0$ pontot ad érte a program). Ebben az esetben a második paraméter állása közömbös (hiszen minden olyan válasz, ami nem jó, neutrális válaszként kerül "elszámolásra").

Másik példának tekinthetjük a "dupla vagy semmi" jellegű értékelési stratégiát, ahol a paraméterek értékei 1 és 0. Ebben az esetben nincs lehetőség a biztonsági jellegű "nem tudom" válaszra, mindig meg kell kísérelni a helyes választ. Ha sikerül, akkor az érte járó pontot pluszban, ha nem, mínuszban kapja meg a tanuló. Az installációs paraméterek alkalmazásánál az volt a célunk, hogy ne szűkítsük le a program alkalmazhatósági körét. A konkrét kiértékelési stratégiák ugyanis csak konkrét oktatási célhoz tartozhatnak; általánosan nem megválaszolható, hogy a fenti lehetőségek közül melyik az "igazi" válaszáértékelési stratégia.



A kérdésekre adott válaszok - a hallgató pontszámának megállapítása mellett - információt tartalmazhatnak a tananyagra vonatkozóan. Minden hallgatói válasz önmagában csak egy apró részlet, de e részletek nagyobb tömegben történő összegyűjtése már következtetések, statisztikai elemzések forrásaként alkalmazható.

Az apró momentumokat tehát célszerű megőrizni, ezt a funkciót az OKTATO program (az INFO fájl folyamatos vezetésével) el is látja.

2.2.3 Elemző programok

E programok célja az, hogy az oktatási folyamat során összegyűlő elemi információ darabkákat az általánosabb következtetések megállapításához alkalmas formába rendezzék; azaz az oktatási folyamat hatékonyságának javításához szükséges, a folyamatból visszacsatolható információkat az oktató rendelkezésére bocsássák. A fenti célt elsősorban matematikai statisztikai elemzések alkalmazásával érhetjük el.

Oktatórendszerünk ZX Spektrum számítógépre kidolgozott változatában az elemző funkciókat ellátó programokat is elkészítettük a rendszer részeként. Figyelembe véve, hogy az oktatók különféle egyéni érdeklődési szempontjait egyetlen program sem tudja egészében kielégíteni, ezért olyan programot készítettünk, amely ellátta az alapfunkciókat és strukturájában biztosította újabb, speciális funkciók beépítésének lehetőségét főként a BASIC programozásban jártas oktatók számára.

A COMMODORE 64-es változat kidolgozásakor figyelembe vettük, hogy erre a számítógéptípusra már vannak olyan programcsomagok, amelyek a matematikai statisztikai elemzések széles skáláját nyújtják. Ennek megfelelően munkánk nem arra irányult, hogy újabb speciális matematikai statisztikai alkalmazói programcsomagot készítsünk, hanem arra, hogy az oktatási folyamat során

rögzített, speciális társtruktúrában tárolt adatokat átalakítsuk egyszerű, általános, a meglevő alkalmazói programcsomagokkal kezelhető formára. Az oktatórendszer szervízprogramját (INST) készítettük fel e funkció ellátására. Így egy "nyitott architektúrájú" programrendszer jött létre, amelyhez illeszthető bármely olyan matematikai statisztikai vagy egyéb célú elemző program, amely a szekvenciálisan tárolt adatokat fogadni tudja. A 3.2.3.-ban részletezett elemzéseket a TTSP programcsomaggal végeztük el.

2.2.4 A szervíz program

Az oktatórendszer üzembe helyezéséhez, karbantartásához szükséges néhány funkció, amelynek a tényleges oktatási folyamatban nincs megfelelője, a gépi programrendszer zavartalan működése azonban feltételezi ezen funkciók végrehajtását is (pl. másolatok készítése, NEVSOR fájl kialakítása, illetve módosítása az információs fájl tartalmának kimentése és új terület biztosítása a további feljegyzések számára, stb). E speciális, itt nem részletezett funkciókat rendszerünkben az INST nevű program biztosítja.

2.3 A tananyagok kidolgozása

Az Élelmiszeripari Főiskolán a Számítástechnika oktatásához készült kurzus tananyagában 13 leckében foglaltuk össze a BASIC nyelv alapjaira vonatkozó ismereteket.

Minden lecke ismeretközlő és ismeretellenőrző részből áll.

A tananyagot a hallgatók ismereti szintjének, az oktatási célnak és körülményeknek megfelelően állítottuk össze.

Természetes igény, hogy az ismeretkör tárgyalási szintje, módja, részletessége különböző legyen, nem lenne szerencsés azonos anyaggal tanítani pl. a teljesen kezdőket és a számítástechnikában jártas, de a C64 BASIC-et konkrétan nem ismerő populációt.

Az Élelmiszeripari Főiskolán alkalmazott tananyagot az alábbi szempontoknak megfelelően dolgoztuk ki.

A leckék ismeretközlő részei viszonylag rövidek (5-10 képernyőlap), összefoglaló jellegűek.

Figyelembe véve, hogy a tervezett oktatási formában nem ez a kizárólagos tanulási forrás, inkább az előadásokon elhangzottak felelevenítésére és a könyvekből szerzett ismeretek rendszerezésére, a terminológia tisztázására szolgál.

A kérdések szorosan kapcsolódnak az ismeretközlő részekhez, nem engedtünk meg olyan kérdést, ami az ismeretközlő részben nem tisztázott információt, tudást igényelne.

A számítástechnika jellegének megfelelően a kérdések helyes megválaszolása "precíz" tudást igényel; az OKTATO program kérdésválasztási stratégiájával összhangban maximálisan törekedtünk a formális megtanulás elkerülésére. A cél érdekében alkalmazott eszközök közül néhányat kiemelünk:

- a kérdések sorszám nélkül jelennek meg a képernyőn, így a helyes választ nem lehet számhoz, mint könnyen megjegyezhető azonosítóhoz kapcsolni.
- Minden csoportban több "vizuálisan hasonló" kérdés van, amelyekhez különböző helyes válasz tartozik (18. ábra). A BASIC kurzus során nem törekszünk a teljes BASIC nyelvi eszköztár megtanítására. Bizonyos speciális, a kezdőktől idegen területeket elhagytunk az alapkursus anyagából (pl. DATA és RESTORE utasítás, nyomtatókezelés). Az alapvető részeket viszont a gyakorlati programozáshoz szükséges mélységben dolgoztuk fel.

A kurzushoz tartozó ismeretközlő részek kiíratott formáját a 2. számú melléklet tartalmazza. Az igen bő kérdéskörből csak néhányat emeltünk ki, amelyek a 3. számú mellékletben találhatók.

BAS09K/3

Helyes-e az alábbi Program-
részlet?

```
10 DIM A(41),C%(22)
20 A(10)=3:C%(0)=19
30 A(C%(0))=6
40 DIM D$(11)
50 D$(A(C%(0)))="ABLAK"
```

- A) Igen.
- B) Nem, mert későn szerepel a DIM D\$(11).
- C) Nem, mert indexként nem használhatunk tömbelemet.
- D) Nem, mert egy DIM utasításban csak egy tömböt deklarálhatunk.
- E) Nem tudom.

BAS09K/4

Helyes-e az alábbi Program-
részlet?

```
10 DIM A(41),C%(22)
20 A(C%(0))=11
30 A(A(0)*4)=14
40 DIM D$(11)
50 D$(A(C%(0)))="ABLAK"
```

- A) Igen.
- B) Nem, mert későn szerepel a DIM D\$(11).
- C) Nem, mert a 20-as sorban az indexnek nincs értéke.
- D) Nem, mert a 30-as sorban indexhatár-túllépés történik.
- E) Nem tudom.

3. AZ OKTATÓRENDSZER ALKALMAZÁSI MÓDJA, TAPASZTALATAI

A bevezetésben részletezett, intenzív oktatási igény megoldására az 1983-ban ZX Spektrum számítógépre kidolgozott oktatórendszert kezdtük el kísérleti jelleggel alkalmazni. Már ez a kisebb populációval folyó kísérleti szakasz is igazolta rendszerünk adaptivitását; az elvégzett elemzések kimutatták a gyenge pontokat mind a tananyagokat, mind az oktatóprogram stratégiai hiányosságait tekintve. Az 1984/85-ös tanévre a "hangolásokon" túlmenően nagyarányú átdolgozás is történt, mivel lehetőség nyílt COMMODORE 64 személyi számítógépek alkalmazására. E gépek mind elterjedést, mind terhelhetőséget, üzembiztonságot, mind kényelmet tekintve az adott felhasználási szempontból jobb paraméterűek a ZX Spektrumnál, így elvégeztük a programrendszer adaptációját a COMMODORE 64 számítógépre.

3.1 Az adaptáció tapasztalatai

A COMMODORE 64 és a ZX Spektrum "közös" alapnyelve a BASIC. Ennek ellenére a két BASIC implementáció elég lényeges tulajdonságaiban eltér egymástól, ami a gyakorlatban azt jelenti, hogy a két gép közötti program

konverzió általában meglehetősen nagy feladat, sokszor egyszerűbb a másik gépre új programot írni, mint a meglévőt átalakítani.

Esetünkben a programok átalakítása egyszerű feladatnak bizonyult, amit a rendszer kidolgozott tervezésének és az előrelátó programozásnak tudunk tulajdonítani. A ZX Spektrumra készített programokban mindenütt az általános BASIC tulajdonságokra építve és nem a specifikumok, mellékhatások (az ún. side effect-ek) kihasználásával készítettük a programokat. Az eltérő tulajdonságú háttér-tár lehetőség (mikrokazetta illetve mágneslemez) sem okozott lényeges, a programstruktúrákat érintő változtatásokat. Természetesen a kísérleti szakasz megfelelő tapasztalatait (pl. az oktató algoritmus általánosítása, kérdés-kiértékelési stratégiák választhatósága) fejlesztésként beépítettük a COMMODORE 64-re készített programokba.

Nagyobb feladatot jelentett a tananyagok és főként a kérdések átdolgozása. A két BASIC eltérése - bár az implementáció során ez még nem mutatkozott - a kérdéseknél komolyabb eltéréseket okozott. Gyakori eset volt, hogy a ZX Spektrumon egyértelmű válaszlehetőségek közül a COMMODORE 64-en ugyanarra a kérdésre több jó válasz lett volna, vagy éppen a lehetséges válaszok egyike sem volt helyes (pl. azonosítók, sztringkezelés, PRINT utasítás).

Ez a jelenség olyan szempontból hátráltatta munkánkat, hogy a COMMODORE 64-re kidolgozott kérdéskört nem lehetett azonosnak tekinteni a ZX Spektrumon már kipróbált, elemzett és az elemzések eredményeként többszörösen javított kérdéskörrel.

Az 1984/85-ös tanévben mintegy 80 hallgató használta az oktatórendszert, jelen tanévben pedig egy teljes évfolyam (150 fő) oktatásában alkalmazzuk.

3.2 Oktatási tapasztalatok

Oktatórendszerünk előnyei több szempontból közölíthetők meg:

- a konkrét kitűzött oktatási cél elérése,
- az oktatórendszer (mint szoftver) általános használhatósága,
- a rendszer felhasználása az oktatási részfolyamatok "bevizsgálására", az oktatás hatékonyabbá tételére.

3.2.1 A kitűzött oktatási cél teljesülése

A gyakorlat igazolta - kisebb majd nagyobb populációkon - az elképzelés helyességét. Az oktatórendszer kidolgozásával a hallgatók döntő többségénél sikerült elérni, hogy a számítástechnikai ismeretek ha-

tékonyabb tárgyalásához szükséges mértékben rövid idő alatt megtanulják a BASIC programozás alapjait.

A 4-5 hetes "átfutási" időszakban egyéni ütemezés szerint, önálló géphasználattal hallgatóink egy része teljesítette a kurzust.

Másik alkalmazási formaként az órarend szerinti gyakorlatokon is felhasználtuk az oktatórendszert.

A tananyagok az előadások ismeretanyagát sűrítik, így a gyakorlatokon végzett ismétlés és ellenőrzés segíti a tananyag gyorsabb feldolgozását és ösztönöz a folyamatos tanulásra. A BASIC gyakorlatok a megfelelő előadásokhoz tartozó leckék feldolgozásával kezdődnek. A 8 C-64 számítógépen a hallgatók egyenként 3-25 percnyi időben - az ismeretközlő részek megtekintése után - egyénileg válaszolnak a tananyaghoz tartozó, a gép által feltett kérdésekre. Ez idő alatt a csoport többi tagja a házi feladatok megoldását ellenőrzi a ZX Spektrum gépeken. Az oktató mindkét részcsoporthoz munkáját irányítja.

Ez a felhasználási mód megoldotta azt a problémát is, amit korábban a hiányzásokból adódó pótlások jelentettek. A kitűzött konkrét célon túlmenően is kedvezően befolyásolta oktatórendszerünk a hallgatók viszonyát a számítástechnikai eszközökhöz.

A számítógép természetes eszközként jelenik meg a hallgatók számára már tanulmányaik első szakaszában,

ami igen pozitívan befolyásolta a továbbiakban a számítógépnek feladatmegoldó eszközként való figyelembe vételét.

A hallgatók nagy érdeklődéssel fordulnak a számítástechnika felé; egyre intenzívebben igénylik a tananyagon túlmenő számítástechnikai foglalkozási formákat (pl. számítástechnikai szakmai klub, speciális kurzusok elkészítése az oktatórendszerhez), és egyre több TDK dolgozat tárgya számítástechnikai alkalmazás.

3.2.2 Az oktatórendszer általános használhatósága

Az oktatórendszert az Élelmiszeripari Főiskola több oktatója kipróbálta. A pozitív visszajelzéseket erősítették a József Attila Tudományegyetemen folyó "Számítógép az iskolában" speciál kollégiumon szerzett tapasztalatok. Ezek alapján állíthatjuk, hogy teljesült az a célkitűzés is, hogy olyan tananyag-kezelő eszközt hozzunk létre, amelyet számítástechnikai ismeretek, előképzettség nélkül is eredményesen lehet alkalmazni; beleértve az önálló tananyag kidolgozást is.

A tananyagok soronkénti felvitele és javítása kézenfekvő a számítástechnikai ismerettel nem rendelkező alkalmazók számára. Ezzel szemben a számítógép-használatban járatosabb alkalmazók hasonló munkáját gyorsabbá tenné a teljes képernyőn történő szövegszerkesztési

lehetőség (full-screen). Ennek megoldását a rendszer továbbfejlesztéseként tervezzük, annak ellenére, hogy ebben a változatban a géptől való maximális függetlenség elvét a program strukturájában már nem lehet betartani.

3.2.3 Tananyag optimalizáló alkalmazások

Az oktatóprogram a lehető legrészletesebben rögzíti az egyes hallgatók tanulási folyamatainak mozzanatait, így lehetőség nyílik annak elemzésére is, hogy mennyire "jó" a tananyag; hol vannak a gyenge pontjai, ahol javításra szorul; hogyan lehetne ugyanezt az ismeretkört hatékonyabban oktatni.

Bár a tananyagok "bemérésére" a tényleges kurzusok is lehetőséget nyújtanak, a teljesebb körű elemzés kedvéért ezt a tevékenységet célszerű külön, speciális feltételek között (de "átlagos" populációval) elvégezni. A speciális feltételek elsősorban arra vonatkoznak, hogy minden résztvevő megkapjon minden kérdést, mert így az összefüggésvizsgálatok sokkal pontosabb, értékelhetőbb eredményeket adnak.

Az Élelmiszeripari Főiskolán alkalmazott tananyagokon - több részleges elemzés után - egy 51 fős populációval végeztük a bemérést, felhasználva e populáció más figyelembe vehető eredményeit is.

A bemérés folyamatából az alábbi elemzéseket emeljük ki.

1.) Elkészítettük a "menetszám-hisztogramokat", amelyek azt mutatják, hogy az egyes leckéknél a hallgatóknak hány menetre volt szükségük ahhoz, hogy a kívánt mértékben (előírt teljesítményszázalék szinten) tudják megválaszolni a kérdéseket.

Ha a menetszám-hisztogramokat nem tartja megfelelőnek az oktató, akkor több irányban változtathat.

a.) Ha az átlagos menetszámot túl alacsonynak tartja, akkor

- nehezíti a kérdéseket,
- csökkenti a kérdésekhez tartozó jó válasz pontértékét (így ugyanazon menetben több kérdést kell megválaszolni a hallgatónak azonos teljesítmény eléréséhez),
- emelheti a kérdéscsoporthoz tartozó "megfelelő teljesítmény %" szintet,
- esetleg megváltoztathatja az értékelési stratégiát (büntetőpontok).

b.) Ha az átlagos menetszám túl magas, akkor

- szükség lehet az ismeretközlő rész felülvizsgálatára, kiegészítésére, további "B" és "C" tí-

pusú lapokkal való bővítésére,

- ha az ismeretközlő rész megfelel a célnak, akkor
 - csökkenthető a kérdéscsoporthoz tartozó "megfelelő teljesítményszázalék" szint,
 - könnyíthetők a kérdések,
 - esetleg megváltoztathatja az értékelési stratégiát (büntetőpontok eltörlése).

c.) ha az átlagos menetszám megfelelő, de a hisztogram "kétpupu" (a hallgatók zöme a túl kicsi, illetve túl nagy menetszámnál összpontosul) akkor mindenképp további részletesebb elemzés indokolt.

2.) Elkészítettük kérdésenként a jó és a rossz válaszok gyakorisági eloszlását. "Gyanúsak" azok a kérdések, amelyekre nagyon kevesen tudnak válaszolni (pl. 10 % alatt). Lehet, hogy a kérdés megfogalmazása nem világos, vagy olyan szakszó fordul elő benne, amit a hallgatók nem, vagy más értelmezésben ismernek. Előfordulhat, hogy nem a kérdéssel van baj, hanem a hozzá kapcsolódó ismeretközléssel, ekkor természetesen azon kell változtatni.

Ha a kérdéseknek van szelekciós funkciójuk, akkor azok a kérdések is "gyanúsak", amelyeket magas arányban válaszoltak meg. Esetünkben ezeket a kérdéseket meghagytuk, egyrészt elég magasan (50-80 %

között) húztuk meg a megfelelő szint határát, másrészt figyelembe vettük, hogy könnyebben alakul ki a pozitív attitűd a számítástechnika iránt, ha sikerélménye van a hallgatónak.

- 3.) Kiszámítottuk a kérdésekre kapott pontszámok között a korrelációs együtthatókat. "Gyanúsak" azok a kérdések, amelyek alacsony korrelációs együtthatóval kapcsolódnak szinte minden más kérdéshez. E kérdéseknél különösen fontos a jó válaszok gyakorisági eloszlásának vizsgálata, hiszen eltérő okok lehetnek, ha e kérdésekre zömében jó, rossz vagy neutrális válasz érkezett.

Ha a hallgatók döntően jól válaszolták meg, akkor valószínű, hogy a kérdés túl könnyű, nem szelektív. Ha döntően rossz válaszok érkeztek, akkor félreérthető a kérdés (önmagában vagy az ismeretközlő részekkel egybevetve), míg ha a jó, a hibás, esetleg a neutrális válaszok száma közel azonos, akkor a kérdés nincs megfelelő kapcsolatban az ismeretközlő részekkel, a válaszokban a találgatás dominált.

- 4.) További információkat nyújthat a válasz-eloszlások vizsgálata, ahol nem csak a jó, rossz és közömbös

válaszok számát tekintjük, hanem azt is, hogy a lehetséges rossz válaszok mely variánsai milyen gyakoriságúak voltak.

Ez az elemzés természetesen csak akkor ad többlet információt, ha háromnál több válasz-lehetőség; ezen belül több hibás válasz van. Ezt az elemzést csak speciális esetekben (nem minden kérdésre) a helytelen orientáció feltárása érdekében végeztük el.

5.) A kurzus anyagából dolgozatot is íratunk a kérdésekben résztvevő populáció tagjaival, s a dolgozatok eredményét összehasonlítottuk a kurzus folyamán nyújtott teljesítményeikkel. A korreláció 98 %-os szinten szignifikánsnak bizonyult.

6.) Az azonos csoportba tartozó kérdéseket osztályokba soroltuk (klasztereztük).

Ennek eredményét a későbbiekben tervezzük felhasználni, az ellenőrzési stratégia módosítására. Ekkor a kérdések kiválasztása nem csak a véletlenszám-generátoron múlna, mert az egyes csoportokból kapna egy-egy kérdést minden hallgató. Így a hallgató tudását több oldalról lehetne vizsgálni, ami valószínűleg szignifikánsabb eredményt adna.

Az elemzésekhez futtatott statisztikai programok

eredményeinek egy részét mintaként a 4. sz. melléklet tartalmazza.

3.3 További célkitűzések

A 3.1 és 3.2 részekben részletezett tapasztalatok birtokában joggal állapíthatjuk meg, hogy a tervezett célt oktatórendszerünk elérte, sőt azon túlmutató lehetőségeket is nyújtott az oktatás hatékonyabbá tételére. Mind a konkrét oktatási tapasztalatok, mind az eredeti rendszer C64-re történő adaptációja a rendszer koncepció helyességét és életképességét igazolta.

A munka ezzel nem zárulhat le; a korábbiakban jelzett fejlesztési tervek (full-screen szerkesztés, kérdésosztályok kezelése) mellett alapvetőnek tartjuk az oktatórendszer szélesebbkörű, más alkalmazási területekre kiterjedő gyakorlati felhasználását.

IRODALOMJEGYZÉK

- [1] Ágoston, Gy: Neveléselmélet
Tankönyvkiadó, Budapest, 1973.
- [2] Ágoston, Gy.; Nagy, J.; Orosz S.: Mérések mód-
szerek a pedagógiában
Tankönyvkiadó, Budapest, 1978.
- [3] Aiken, R. M.; Braun, L.: Into the 80's with
microcomputer based learning
Computer, 13. k. 7. sz. 1980. p. 11-16.
- [4] Aiken, R.M.: Teacher Training, for CAL
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [5] Alty, J. L.: Path Algebras: A Usfull CAI/CAL
Analysis Technique
Computer and Education, 8. k. 1. sz. 1984.
p. 5-13.
- [6] Bakos, E.; Bölcsföldi, J.; Koltai, M.; Láng, Cs.;
Olti F.: CHEMISYS felhasználói kézikönyv és út-
mutató a szubrutinok készítői számára
- [7] BASIC Schnellkurs
McGraw - Hill Book Company, GMBH, 1985.
- [8] Bell, T. E.: My computer, my Teacher
Personal Computing, 7. k. 6. sz. 1983.
p. 120-127.

- [9] Billings, K.: Microcomputers in Education:
Now and in the Future
Microcomputing, 1980. jun. p. 100-102.
- [10] Bloom, B. S.: Human characteristics and school
learning
McGraw - Hill, New-York, 1976.
- [11] Bloom, B. S.: Taxonomy of Educational Objectives
David Mc Kay Co. Inc. 1956.
- [12] Bloom, B. S.: Taxonomie von Lernzeilen in kogniti-
ven Bereich
Beltz, Weinheim, Basel, 1972.
- [13] Boecker, H.: Functional Programming in BASIC - PLUS
WCCE 81, Lausanne, Nort Holland Publishing
Company, 1981.
- [14] Bork, A.: Computers and the Future: Education,
Computer and Education 8. k. 1. sz. p. 1-4.
1984.
- [15] Brahan, J. W.; Godfrey, D.: A marriage of Com-
vimience: videotex and computer assisted
learning
Computers and Education. 1982. 6.k. 3.sz.
p. 33-38.
- [16] Bright, G. W.: Explaining, the Efficiency of
Computer Assisted Instruction
AEDS Journal, 16. k. 1983. Spring p.144-152.



- [17] Brückner, H.: Számítógépek az oktatásban - számítógépes oktatás
Statisztikai Kiadó, Budapest, 1978.
- [18] Buerkler, H.; Zehnder, C. A.: The training of
University Students in Computer - Project
Management
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [19] Chastenet de Gery; Hocquenghem, S.: Collective
Use of a Micro - Computer with Graphics to
illustrate the Mathematics Lesson
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [20] Csákány, A.; Tarnay, K.: TPA-I TEASYS általános
leírás
MTA KFKI Budapest, 1982.
- [21] David, B. T.: Computer Aided Design Education
An Overview
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [22] Dean, C.; Whitlock, Q.: A handbook of computer
based training Kogan Page, London
Nichols Publishing Company, New York, 1983.
- [23] Doerr, C.: Microcomputers and the 3 R's. A
guide for teachers Hayden microcomputers
series, Rochelle Park, USA, 1979. p. 177.

- [24] Dowsey, I.; Computer Based Training in IBM
edukation
Hamburg, 1981.
- [25] Dringó, L.: Számítógépek alkalmazása az oktatásban
Doktori értekezés Budapest, ELTE TTK, 1974.
- [26] Du Chang Zheng: CAI System for BASIC
Microprocessing and Microprogramming
12. k. 3/4. sz. 1983. okt./nov. p. 227-230.
- [27] Fekete, G.; Györgyi, A.: Számítógép a tanteremben
Egyetemi Számítóközpont, Budapest, 1980.
- [28] Fekete, G.: Számítógépes oktatás
Doktori értekezés Budapest, Egyetemi Számító-
központ, 1976.
- [29] Forcheri, P.; Lemut, E.; Molfino, M. T.:
The GRAF system: an interactive graphic
system for teaching mathematics
Computers and Education. 7. k. 3. sz. 1983.
p. 177-182.
- [30] Frizot, D.: Teaching English with Computer
Assisted Learning
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [31] Fuchs, W. R.: Az új tanulási módszerek
Közgazdasági és Jogi Könyvkiadó, Budapest, 1973.
- [32] Gleason, G. T.: Microcomputers in Education: The
state of the art.
Educational Technology, 1981.márc. p.7-18.

- [33] Hack, F.: Számítógéppel támogatott probléma-
megoldás
ELTE ABC-s füzetek
Budapest, 1982.
- [34] Hack, F.: Számítógépes oktatóprogramok terve-
zése
ELTE ABC-s füzetek
Budapest, 1982.
- [35] Hajtman, B.: Bevezetés a matematikai statisztikába
Akadémiai Kiadó, Budapest, 1968.
- [36] Hamblen, J. W.: Status of Computing and Computer
Science Education in USA Higher Education:
1978-79
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [37] Hamann, G.O.; Lerne BASIC mit dem Volkscomputer
VC 20. Programierte Unterweisung
DB Verlag GMBH, Gernsbach, 1983.
- [38] Hámori, M.: Kísérletek a tanulási - tanítási
folyamat számítógépes támogatására
Pedagógiai Technológia, 1980/2.
- [39] Hámori, M.; Szalay, M.: MISY: TPA kisszámítógépre
alapozott általános célú oktatás vezérlő
rendszer
KFKI, Budapest, 1978.

- [40] Hámori, M.: Tanulás és tanítás számítógéppel
Tankönyvkiadó, Budapest, 1984.
- [41] Hausen, H.R.: Microcomputer in der US-amerika-
nischen Hochschulansbildung
Angewandte Informatik, 26. k. 11. sz.
p. 441-451. 1984.
- [42] Hofstelter, F. T.: A Model for Administering
Computer Based Education
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [43] Horváth, L.; Szlávi, P.; Zsakó, L.:
Modellezés és szimuláció
ELTE ABC-s füzetek
Budapest, 1982.
- [44] Howe J.: Teaching Handicapped Children to Read:
A Computer Based Approach
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [45] Howarth, T.: Taking a Stand for Computers in
Education
Personal Computing, 1983. máj. p. 121-129.
- [46] Ibrahim, R.L.: Computer Assisted Instruction in
Kuwait: A Pilot Study
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.

- [47] Itelszon, L. V.: Matematikai és Kibernetikai
módszerek a pedagógiában
Tankönyvkiadó, Budapest, 1972.
- [48] Jiqing, W.; Qinztné, Z.; Huahna, Z.: A micro-
computer - Based Instruction System - Basic
programming Instruction
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [49] Keith, A. H.: Content Structuring and Question
asking for computer - based education
Journal of Computer Based Instruction
Summer 1983, 10. k. 1. sz. p. 1-7.
- [50] Kelemen, L.: Pedagógiai pszichológia
Tankönyvkiadó, Budapest, 1981.
- [51] Kemény, J. G.; Snell, J. L.; Thomson, G. L.:
A modern matematika alapjai
Műszaki Könyvkiadó, Budapest, 1971.
- [52] Kenealy, P.: Personal computers: a new generation
emerges
Design News, 39. k. 3. sz. 1983. p. 74-89.
- [53] Kovács, T; Temesszeghy, G.; Fay, G.; Nováky, E.:
Collective Enviromental Forecasting by
Microcomputers: A paradigm for Enviromental
Education
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.

- [54] Kőhegyi, J.: Ismerd meg a BASIC nyelvjárásait
Műszaki Könyvkiadó, Budapest, 1985.
- [55] Kuhn, W.; Lorton, P.: Computer Assisted Instruction
in Music-Ten years of Evolution
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [56] Lalov, B.: Methods of Perfection and Optimization
of Computer Aided Learning
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [57] Lewis, R.; Tagg, E. D.: Computer Assisted Learning
North Holland Publishing Company,
Amsterdam New-York Oxford, 1980.
- [58] Lantz, B. S.; Bregar, W. S.; Farley, A. M.:
An Intelligent CAI System for Teaching
Equation Solving
Journal of Computer - Based Training Summer,
1983, 10. k. 1. sz. p. 35-42.
- [59] Lapuse, R.: BASIC Schulungs - Handbuch Control
Data Institut, Frankfurt/M, 1976.
- [60] Lay, R. W.: Basic Techniques for Teaching "BASIC"
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [61] Lewis, R.: CAL Design
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.

- [62] Lewis, R.; Tagg, E. D.: Computer Assisted Learning
North Holland Publishing Company,
Amsterdam. New-York. Oxford, 1980.
- [63] Lovász, L.; Gács, P.: Algoritmusok
Műszaki Könyvkiadó, Budapest, 1978.
- [64] Lőcs, Gy.; Sarkadi Nagy, J.; Szlanko, J.:
A BASIC programozási nyelv
Műszaki Könyvkiadó, Budapest, 1976.
- [65] Lyster, M.; Dalgaard, L.; Belhage, G.: DISK:
A Research Project in Computer Aided
Learning in Danish Schools
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [66] Márkus T.-né: Fizikai kísérletek mikroprocesszoros
szimulációja
Egyetemi Számítóközpont Tájékoztatója, 25.
Budapest, 1979.
- [67] Marlin, I. W.; Niss, J. F.: Microcomputer management
of Instruction the advanced learning system
Microprocessing and Microprogramming,
11. k. 2. sz. 1983. p. 117-126.
- [68] Mast, Ch. van der: A portable program to present
courseware on microcomputers Computers and
Education, 6. k. 1. sz. 1982. p. 39-44.
- [69] Matavovszky, T.: Oktatórendszerek strukturájának
és működésének leírása
Doktori értekezés, Budapest, ELTE TTK, 1977.

- [70] McIntyre, J.: A concept for the Application of
Computer Assisted Instruction to Education
in Anaesthesiology
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [71] Montrigaund, A.: Automatic Generation of Exercises
on Italian Grammar
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [72] Nagaoka, K.; Fujita, H.: Computerised Training
of Discussion
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [73] Nagy E.-né: Az alkalmazói számítástechnikai képzés
néhány problémája az élelmiszeripari üzem-
mérnökképzésben
A műszaki főiskolák matematika- fizika és
számítástechnika oktatóinak 6. országos
tanácskozása
Kazincbarcika, 1982. január 28-29-30.
- [74] Nagy E.-né: Az eredmények számítógépes kiértékelé-
sének tapasztalatai
Megtanító stratégiák az agrártudományi fel-
sőoktatási intézmények alapozó tárgyaiban;
Tanulmány
Budapest, 1982.

- [75] Nagy, E.-né: FOODSYS - nutrajindustria instrucela programaro
INTERKOMPUTO Nemzetközi Számítástudományi Szinposium
Budapest, 1982.
- [76] Nagy, E.-né: Számítástechnikai képzés az Élelmiszeripari Főiskolán
Élelmezési Ipar 1982. 10.
- [77] Nagy, E.-né: Számítástechnika jegyzet Élelmiszer-technológia szakos hallgatók részére
Élelmiszeripari Főiskola, 1980.
- [78] Nagy, J.: Az információhordozók rendszerré szervezése
(Pedagógiai programcsomag)
Pedagógiai Technológia, 1981. b. 1. 1-9.
- [79] Nagy, J.: A megtanítás stratégiája
JATE Pedagógiai Tanszék, Szeged, 1981. a.
- [80] Nagy, J.: A megtanítás stratégiája
Köznevelés, 1981. c. 33. p. 3-6.
- [81] Nagy, J.: Az OOK és a pedagógiai technológia
OOK, Veszprém, 1979.
- [82] Nagy J.: A témazáró tudásszintmérés gyakorlati kérdései
Tankönyvkiadó, Budapest, 1972.
- [83] Nagy, J.: Oktatástechnológiai kutatásaink néhány alaptendenciája
Pedagógiai Technológia, Veszprém, 1983/84.

- [84] Nagy, J.: Témazáró tesztek reliabilitása és validitása
Szeged, 1975.
- [85] Nagy, J.: Tudástechnológia 4. Fogalom
Kézirat
Szeged, 1985.
- [86] Nagy, S.: Az oktatáselmélet alapkérdései
Tankönyvkiadó, Budapest, 1981.
- [87] Neuhauser, J.J.: A Necessary Redirection for
Certain Educational Technologies
Computers & Education, 1977, p. 187-192.
- [88] O' NEIL, D.: Physics Teacher
Microcomputing, 1980. jun. p. 138-148.
- [89] Otsuki, S; Takeuchi, A.: A Unified C. A. L.
System for Authoring, Learning and Managing Aids
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [90] Pálvölgyi, L.: A modellezés lehetőségeiről a
pedagógiában
Akadémiai Kiadó, Budapest, 1981.
- [91] Perczel, T.: Tesztszerkesztés és tesztanalízis
Kézirat
Felsőoktatási Pedagógiai Kutatóközpont, 1974.

- [92] Petrov, N.; Medvedev, V.; Zykov, M.: The use
of Educational Games in Computer Assisted
Learning
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [93] Pukkila, T.; Puntamen, S.: The computer as an Aid
in Teaching Basic Statistics Courses at
University Level
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [94] Ross, P.; Howe, J.: Teaching Mathamatics through
Programining Ten Years On
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [95] Schumacher, B.: Mikroelektronik in der Schule
Management Heute, Harzburger Hefte, 24. k.
9. sz. 1982. p. I-IV.
- [96] Seymour, C. H.: Basic programming self taught
Reston Publishing Company, 1981.
- [97] Sharples, M.: A Computer Based Teaching Scheme
for Creative Writing
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [98] Shaw, K.A.: Some Uses of Computers in Teaching
Chemistry in UK Secondary Schools
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.

- [99] Skinner, B. F.: A tanítás technológiája
Gondolat, Budapest, 1973.
- [100] Skyrme, D. J.: The evolution of Graphics in CAL
Computers and Education 6. k. 1982.
p. 13-24.
- [101] Solowary, E.; Johnson, W. L.: MENO-II: An AI
based programming tutor
Journal of Computer - Based Instruction
Summer 1983, 10. k. 1. sz. p. 20-34.
- [102] Szlávi, P.; Zsakó, L.: A programozás ABC-je =
Az ABC programozása ELTE ABC-s füzetek
Budapest, 1983.
- [103] Szűcs, P.: The Use of Microcomputers in Hungarian
Education
Országos Oktatástechnikai Központ, Veszprém
1983.
- [104] Tagg, E. D.: Microcomputers in Secondary Education
Proceedings of the IFIP TC3 Working
Conference on Microcomputers in Secondary
Education
Sèvres, Paris, France 14-18. April. 1980.
- [105] Thomas, P. G.: QASBASIC - a structured BASIC-like
language for teaching computing
Computers in Education, 7. k. 3. sz. 1983.
p. 141-148.

- [106] Tibor, E.; Hámori, M.; Göndöcs, K.: Tananyag
elrendezés számítógéppel
Egyetemi Számítóközpont, Budapest, 1980.
- [107] Vágner, Gy.: Számítógépes oktatás
Egyetemi Számítóközpont, Budapest, 1974.
- [108] Want, D.L.: Support for Educational Software on
Microcomputers
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [109] Watson, D. M.: A model for the production of
CAL material
Computer and Education, 7. k. 3. sz. 1983.
p. 167-176.
- [110] Watson, D. M.: Computer Assisted Learning in
The Humanities
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.
- [111] Williams, D. L.: Computer minds school corpo-
ration
Microcomputing, 1980. jun. p. 44-46.
- [112] Williamson, R. A.; Monrol, J. G.: Getting your
kid into college
Personal Computing, 1983. okt. p. 140-147.
- [113] Wong, D.G.: Computer Based Aids for Teaching
Digitál Systems Design
WCCE 81, Lausanne, North Holland Publishing
Company, 1981.

1. SZ. MELLÉKLET

**A PROGRAMOK
FORRASNYELVÜ SZÖVEGE**

```

EM OKTATO/1985.5.12./1985.7.21.
DIM S$(100),K$(100)
DIM HK(50),NV$(50),LOC(50)
DIM SL$(99),ST$(300,2)
DIM CL$(8),DP$(100),DM$(100),DJ$(100)
DIM DT(100),DV$(100)
REM ***** VARIABLES *****
REM LI(MK):TANITO KOR LIMIT
REM LH :TANARI FILE MERET LIMIT
REM FS$ :SZOVEGFILE NEVE
REM FK$ :KERDESFIL NEVE
REM SH :S HOSSZA BYTE-BAN
REM NL :LAPOK SZAMA S-BEN
REM NT :SORTABLA S-HEZ
REM PP 1 KOR POT PONTJA
REM TP 1 KOR TENY PONTJA
REM TZ 1 KOR TELJ.%-A
REM
OPEN15,8,15
LI=2:LH=50
INPUT "HALLGATO";W$
IF W$="*" THEN GOSUB 9900
PRINT "DATUM=":INPUT "EV=";EV:INPUT "HONAP=";HO
INPUT "NAP=";NA
GOSUB 7000:REM DATELL(EV,HO,NA,HD)
IF HD=1 THEN GOTO 210
OPEN5,8,5,"LINFO":GOSUB 610
T1=1:GOSUB 8900:INPUT#5,RV$:CLOSE5
GOSUB 8000:REM TANFIL BE
GOSUB 8100:REM HALLG.BEJ.(AH,AK)
HA=LO(AK)+1
GOSUB 8200:REM CONV1(HA,C2$)
PRINT NV$(AK);" SZAMARA A BAS";C2$:PRINT " TANANYAG KOVETKEZIK"
FS$="BAS"+C2$+"S":FK$="BAS"+C2$+"K"
GOSUB 8300:REM SBE
GOSUB 8400:REM KBE
K0=0
K0=K0+1:HB=0:PP=0:TP=0
MU$(1)="A":MU$(2)="B"
GOSUB 1400:REM TANIT
IF HB>0 THEN GOTO 480
GOSUB 2000:REM KERDEZ(TZ)
REM PRINT "TZ="TZ
GOSUB 3000:REM ERTEKEL(TZ,TUD)
REM PRINT "TUD="TUD
IF TUD=0 AND K0=LI THEN GOTO 550:REM CANCEL
IF TUD=0 AND K0<LI THEN GOTO 350
REM IF TUD=1
LO(AK)=LO(AK)+1
GOSUB 8600:REM TANFIL URIT
PRINT NV$(AK):CS$="KERI A KOVETKEZO TANANYAGOT?":GOSUB 1310
IF T1=1 THEN GOTO 250
CS$="VAN MAS VALLALKOZO":GOSUB 1310
IF T1=1 THEN GOTO 240
REM LEZARAS
GOSUB 8600:REM TANFIL URIT

```

ADY.



```

REM RV% VISSZIRAS LINFO-RA
OPEN5,8,5,"LINFO":GOSUB610
T1=1:GOSUB8900
PRINT#5,RV%CHR$(13)
CLOSE5
CLOSE15
END
PRINT NV$(AKTHAL)"!!!!":PRINT" A FELFOGOKEPESSEGE SZAMOMRA GYENGE"
PRINT "FORDULJON HUMAN OKTATOHOZ!"
GOTO 460
REM DISCTL
INPUT#15,X1,X2$,X3,X4
IF X1>20 THEN PRINTX1,X2$,X3,X4
RETURN
0 REM ICTL
2 I1%=0:H1=0:S1=0
4 IF T1<>1 THEN GOTO 1030
6 IF LEN(W$)>5 THEN H1=1:RETURN
8 FOR J=1TO LEN(W$):IF MID$(W$,J,1)>"9"ORMID$(W$,J,1)<"0"THEN H1=H1+1
0 NEXTJ
2 S1=VAL(W$)
4 IF S1>=32*1024 THEN H1=1:S1=0:RETURN
6 I1%=VAL(W$)
8 RETURN
0 IF T1<>2 THEN H1=1:RETURN
2 STOP
8 RETURN
0 REM KERDEZ
0 PRINTCS$+"(Y/N)":W$="":INPUT W$
0 IF W$="Y" OR W$="I" THEN T1=1:RETURN
0 IF W$="N" THEN T1=0:RETURN
0 GOTO1320
0 REM TANIT
0 FOR SS=1 TO NL
0 IF SLX(SS)<0 THEN GOTO 1590
0 IN=STX(SLX(SS),2):GOSUB 6100
0 FOR J=1 TO LI
0 IF MU$(J)=CH$ THEN GOTO 1480
0 NEXT J
0 PRINT"HIBAS TANANYAG VAN BENT!!!":HB=1:RETURN
0 IF K0<J THEN GOTO 1510
0 GOSUB 1800: REM LAPDISP
0 PRINT"HA ELOLVASTA, NYOMJA MEG A RETURN BILLENTYUT ":INPUT W$
0 NEXT SS
0 RETURN
0 REM LAPDISP (SS,LAP)
5 FOR I=1TO6:PRINT:NEXTI
0 TA=SLX(SS):NO=1:IL=0
0 GOSUB1900:REM SORDISP
5 IL=IL+1
0 TA=STX(TA,1):NO=0
0 IF STX(TA,1)>0 THEN GOTO 1820
5 FORI=IL+1TO20:PRINT:NEXTI
0 RETURN
0 REM SORDISP (TA )
0 SA=STX(TA,2):G$="":IG=0
0 FOR J=SA+NO TO SA+100
0 IF IG>32 THEN STOP
0 IN=J :GOSUB 6100
0 IF CH$="#" THEN PRINTG$:GOTO1990
0 IG=IG+1:G$=G$+CH$
0 NEXT J
0 RETURN

```



```

0 REM KERDEZ
2 TP=0:PP=0
5 IF CL%(1)<1 THEN GOTO 2190
0 GOSUB 7100:REM VELSZ(RS)
0 IF DV%(RS)=0 THEN GOTO 2050
0 IF DV%(0)<1 THEN GOSUB 8400:GOTO 2005
5 GOTO 2010
0 RS=INT(RS):KP=RS: GOSUB 7200:REM KERD KIIR(KP)
0 W$="":INPUT "A HELYES VALASZ JELE:";W$
5 VV$=W$
0 IF LEN(W$)<>1 THEN GOTO 2060
0 PP=PP+DP%(RS):JD$=LEFT$(DJ$(RS),1)
0 IF W$=JD$ THEN PRINT "A VALASZ JO":EP=DP%(RS):TE$="2":GOTO 2110
5 IF W$=RIGHT$(DJ$(RS),1) THEN EP=0:TE$="1":GOTO 2110
0 PRINT " A VALASZ HIBAS. A JO VALASZ A "CHR$(34)JD$CHR$(34)" JELU"
5 EP=-1*INT(DP%(RS)/2):TE$="0"
0 REM ACCT REC(HKOD,LECKE,KOR,RS,/TUDE(0,1,2),MITVAL(W$)/TELJ%
1 GOSUB 3200:REM ACR ELEJE(HK(AK),HA,K0,UR$)
3 UR$=UR$+RIGHT$(" "+STR$(RS),3)+TE$+VV$+"0"
5 GOSUB 3300:REM LINFRA IR(UR$,RV%)
0 TP=TP+EP
5 DV%(0)=DV%(0)-1:DV%(RS)=1
7 REM PRINTEP,TP,PP
0 IF PP<CL%(2) THEN GOTO 2010
0 TZ=(TP*100)/PP
0 RETURN
0 REM ERTEKEL(TZ,TUD)
0 IF TZ<CL%(3) THEN TUD=0:GOTO 3050
5 TUD=1:CS$="IGEN JO"
0 IFTZ<CL%(4) THEN CS$="MEGFELELO"
0 IFTZ<CL%(5) THEN CS$=" J O "
0 PRINT "TUDASSZINT:",CS$
0 REM ACCT.REC(HKOD,LECKE,KOR,000,TELJ.%(TZ))
1 GOSUB 3200
5 UR$=UR$+"000"+RIGHT$(" "+STR$(INT(TZ)),3)
0 GOSUB 3300
0 RETURN
0 REM ACR ELEJE
0 UR$=RIGHT$(" "+STR$(HK(AK)),3)+C2$
0 UR$=UR$+RIGHT$(STR$(K0),1)
0 RETURN
0 REM LINFRAIR(UR$,RV%)
0 OPEN 5,8,5,"LINFO":GOSUB 610
5 RV%=RV%+1
0 T1=RV%:GOSUB 8900
0 PRINT#5,UR$CHR$(13)
0 CLOSE 5
0 RETURN

```

IDY.

```

0 REM LINFO KIIR
0 OPEN#5,8,5,"LINFO":GOSUB610
0 T1=1:GOSUB8900
0 INPUT#5,RV%:PRINT"AZ INFO FILE"RV%-1"BEJEGYZEST TARTALMAZ"
0 IFRV%<2THEN GOTO 3685
0 FOR II=2TORV%:T1=II:GOSUB8900
5 INPUT#5,UR$
7 IFII=INT(II/20)*20<>2THEN GOTO3665
8 INPUT"MEHET";Q$
9 FOR IL=1TO5:PRINT:NEXTIL
0 PRINT"HK. LECKE KOR K.SZ. VAL ERT TELJZ"
5 PRINTMID$(UR$,1,3);" "MID$(UR$,4,2)" "MID$(UR$,6,1);
0 IFMID$(UR$,7,3)="000"THEN GOTO3676
2 PRINT" "MID$(UR$,7,3)" "MID$(UR$,11,1)" "VAL(MID$(UR$,10,1))-1
4 GOTO 3680
6 PRINT" "MID$(UR$,10,3)
0 NEXTII
5 CLOSE5
7 INPUT"MEHET";Q$
0 RETURN
0 REM SBE(INDEX,CH$)
0 IF INDEX<1 OR INDEX>5000 THEN PRINT"HIBA 6!!":RETURN
0 IF LEN(CH$)<>1 THEN PRINT"HIBA 7!!":RETURN
5 JJ=INT((IN-1)/50):JK=IN-50*JJ
8 IF JK<1 THENPRINT"6028":STOP
0 IF JK=1 THEN S$(JJ)=CH$:GOTO 6090
0 S$(JJ)=LEFT$(S$(JJ),JK-1)+CH$
0 RETURN
0 REM SBOL(INDEX,CH$)
0 IF INDEX<1 OR INDEX>5000 THEN PRINT"HIBA 6!!":RETURN
5 JJ=INT((IN-1)/50):JK=IN-50*JJ
8 IF JK<1 THENPRINT"6118":STOP
0 CH$=MID$(S$(JJ),JK,1)
0 RETURN
0 REM DATELL
0 HD=0
0 IF EV<1984 OR EV>1999 THEN GOTO 7090
0 IF HO<1 OR HO>12 THEN GOTO 7090
0 IF NAP<1 OR NAP>31 THEN GOTO 7090
0 RETURN
0 HD=1:RETURN
0 REM VELSZ(RS)
0 RS=RND(1)*(CLZ(1)+1)
0 IF RS<1THENRS=1: GOTO 7190
0 IF RS>CLZ(1)THENRS=CLZ(1):GOTO 7190
0 REM
09 RETURN
0 REM KERD KIIR(KP)
0 IX=DMZ(KP):IL=0
5 FORI=1TO6:PRINT:NEXTI
0 REM SORKEZ
0 GOSUB 7300:REM SORKER(IX,G$,PU )
0 PRINTG$
50 IX=PU:IL=IL+1
50 IF PU>0THEN GOTO 7230
70 FORI=IL+1TO20:PRINT:NEXTI
00 RETURN

```



```

0 REM SORKER(IX,G$,PU)
5 PU=-2:G$="@@@@@"
0 IF IX<10RIX>5000 THEN STOP
0 T1=INT((IX-1)/50):T2=IX-T1*50
5 IF T2<1 THEN RETURN
0 CW$=MID$(K$(T1),T2,51-T2)+K$(T1+1)
0 HS=VAL(LEFT$(CW$,2))
0 G$=MID$(CW$,3,HS)
0 PU=VAL(MID$(CW$,HS+4,4))
0 PZ=IX+LEN(G$)+3
0 RETURN
0 REM TANFILBE
0 OPEN 2,8,2,"TANFIL,S,R"
0 INPUT#2,NHALG,NLEC
0 FOR I=1 TO NHALG
0 INPUT#2,HK(I),NV$(I),LOC(I)
0 NEXT I
0 CLOSE2
0 RETURN
0 REM HALLG.BEJ.
0 INPUT"HALLGATO KODJA(3 SZAMJEGGYEL)=";AH
0 AK=0
0 FOR I=1 TO NH
0 IF HK(I)=AH THEN AK=I
0 NEXT I
0 IF AK=0 THEN PRINT"ISMERETLEN KOD":GOTO 8100
5 PRINT"NEV=";NV$(AK)
0 CS$="RENDBEN":GOSUB 1310
5 IF T1=0 THEN GOTO 8110
0 RETURN
0 REM CONV1
0 IF HA<0 OR HA>99 THEN PRINT"STOP8210":STOP
0 JX=INT(HA/10):JY=HANY-10*JX
0 C2$=CHR$(JX+48)+CHR$(JY+48)
0 RETURN
0 REM S BE
0 OPEN 2,8,2,FS$+",S,R"
0 INPUT#2,SH,NL
2 FOR I=1TO99:INPUT#2,SL$(I):NEXTI
4 INPUT#2,NT
6 FOR I=1TO 300:INPUT#2,ST$(I,1),ST$(I,2):NEXT I
0 FOR I=0 TO 100
0 INPUT#2,S$(I)
0 NEXT I
0 CLOSE2
0 RETURN
0 REM K BE
0 OPEN2,8,2,FK$+",S,R":GOSUB610
5 FORI=1TO8:INPUT#2,CL$(I):NEXT I
0 FORI=1TO100:INPUT#2,DP$(I),DJ$(I),DM$(I),DT(I)
2 NEXTI
0 FOR I=0 TO 100
0 INPUT#2,K$(I)
0 NEXT I
0 CLOSE2
5 FORI=1TO CL$(1):DV$(I)=0:NEXT I
0 DV$(0)=CL$(1)
0 IF CL$(1)<1 THEN PRINT"ITT NINCS KERDES!!!":RETURN
0 RETURN

```



```

0 REM UJINFIR(T1,UR$)
5 J1=INT(T1/256):J2=T1-J1*256
0 PRINT#15,"P"CHR$(5)CHR$(J2)CHR$(J1)CHR$(1)
5 PRINT#5,UR$
0 GOSUB610
0 RETURN
0 REM INFJEJIR(T1,RV%)
5 J1=INT(T1/256):J2=T1-J1*256
0 PRINT#15,"P"CHR$(5)CHR$(J2)CHR$(J1)CHR$(1)
5 PRINT#5,RV%CHR$(13)
0 GOSUB610
0 RETURN
0 REM TANURIT
0 OPEN 2,8,2,"@:TANFIL,S,W"
0 PRINT#2,NHALG,"NLEC
0 FOR I=1 TO NHALG
0 PRINT#2,HK(I),"NV$(I)","LOC(I)CHR$(13)
0 NEXT I
0 CLOSE2
0 RETURN
0 REM INF.ALLOC
5 CS$="VAN MAR INFO FILE A LEMEZEN"
0 GOSUB1310
5 IFT1=1 THEN GOTO 8770
0 W$="":INPUT"HANY BEJEGYZES";W$
5 IFW$="" THEN AU=400:GOTO8740
0 T1=1:GOSUB1010
5 IFH1>0ORS1<2ORS1>2000 THEN GOTO8720
0 AU=S1:UR$="@@@@@@"
0 OPEN5,8,5,"LINFO,L,"+CHR$(15):GOSUB610
5 T1=AU:GOSUB8500
0 FOR II=1TOAU-1:T1=II:GOSUB8500
5 GOSUB610:NEXTII
0 CLOSE5
0 OPEN5,8,5,"LINFO":GOSUB610
5 T1=1:GOSUB8900
0 RV%=1:PRINT#5,RV%CHR$(13)
0 CLOSE5
0 RETURN
0 REM INF.URIT
5 W$="":INPUT"KEREM A MENTO FILE NEVET";W$
0 IFLEN(W$)<10ORLEN(W$)>8 THEN GOTO8805
5 MF$=W$
0 OPEN2,8,2,MF$+",S,W":GOSUB610
0 OPEN5,8,5,"LINFO":GOSUB610
35 T1=1
40 GOSUB8900
50 INPUT#5,RV%
55 IFRV%<2 THEN GOTO 8880
50 FORII=2TORV%:T1=II:GOSUB8900
55 INPUT#5,UR$
70 PRINT#2,UR$CHR$(13)
75 NEXTII
80 CLOSE2
85 CLOSE5
90 RETURN
0 REM POS(T1)
02 J1=INT(T1/256):J2=T1-J1*256
05 PRINT#15,"P"CHR$(5)CHR$(J2)CHR$(J1)CHR$(1)
10 GOSUB610
19 RETURN

```

```

0 REM TANFILGEN
5 LH=50
0 NH=LH :NLEC=99
0 FORI=1TOLH
5 HK(I)=0:NV$(I)="" :LOC(I)=0
0 NEXTI
0 GOSUB8600
0 RETURN
0 REM TANDISP
2 XR$="*****":XR$=XR$+XR$
3 XR$=XR$+XR$
5 CS$="LEMEZROL HOZZAM":GOSUB1310
0 IF T1=1 THEN GOSUB 8000
2 PRINT"J":PRINT" HALLG.SZAM="NH;" LECKE SZAM="NL
5 PRINTXR$
7 PRINT" KOD N E E V";TAB(31)"LECKE"
9 PRINTXR$
0 FOR I=1 TO NH
5 IF INT(I/18)*18=I THEN INPUT "MEHET";W$
0 PRINTHK(I);NV$(I);TAB(32)LOC(I)
5 NEXTI
7 PRINT:PRINT:INPUT "MEHET";W$
0 PRINT"J"
0 RETURN
0 REM TANMOD
5 IF NH=0 THEN PRINT"NINCS MIT MODOSITANI!!":RETURN
0 PRINT"HALLGATOK SZAMA=";NH;:W$="": INPUT W$
5 IF W$="" THEN GOTO 9230
0 T1=1:T2=3:GOSUB 1010:PRINTT1,H1,I1%
5 IF H1<>0 OR I1%<0 OR I1%>999 THEN GOTO9210
0 NH=I1%
0 PRINT"TANANYAGOK SZAMA=";NL;:W$="":INPUTW$
5 IF W$="" THEN GOTO9260
0 T1=1:T2=3:GOSUB 1010
5 IF H1<>0 OR I1%<0 OR I1%>999 THEN GOTO9230
0 NL=I1%
0 FOR I=1TONH
5 PRINTI;".HALLGATO"
0 PRINT"KODJA="HK(I);:W$="":INPUTW$
5 IF W$="*" THEN GOTO 9380
0 IF W$="" THEN GOTO 9300
5 T1=1:T2=3:GOSUB 1010
0 IF H1<>0 OR I1%<0 OR I1%>999 THEN GOTO9270
5 HK(I)=I1%
0 PRINT"NEVE="NV$(I);:W$="":INPUTW$
5 IF W$="" THEN GOTO 9320
0 NV$(I)=W$
0 PRINT"TELJESITETT TANANYAG="LOC(I);:W$="":INPUTW$
5 IF W$="" THEN GOTO 9350
0 T1=1:T2=3:GOSUB 1010
5 IF H1<>0 OR I1%<0 OR I1%>99 THEN GOTO9320
0 LOC(I)=I1%
0 REM
0 NEXTI
0 REM
0 RETURN

```

```
2 PRINT1,"TANARI FILE INIT"
3 PRINT2,"TANARI FILE DISPLAY"
2 PRINT3,"TANARI FILE MODOSITAS"
5 PRINT4,"TANARI FILE KIIRASA DISCRE"
7 PRINT5,"INFO FILE ALLOC"
2 PRINT6,"INFO FILE MENTESE"
2 PRINT7,"INFO FILE MEGTEKINTESE"
4 PRINT8,"VISSZATERES HALLGATOI SZINTRE"
5 PRINT9,"VEGE"
0 W$="":INPUT"A VALASZTOTT TEVEKENYSEG SZAMA=";W$
5 IF W$="9" THEN GOTO9998
7 IF W$="8"THEN RETURN
0 IFW$<"1"ORW$>"7"THEN GOTO 9900
0 ON VAL(W$) GOSUB9000,9100,9200,8600,8700,8800,3600
0 GOTO 9900
8 CLOSE15
9 END
```

DY.


```

REM TAFEL/7.21.
DIM S$(100),U$(100)
DIM L(99),T(300,2)
FOR K=1TO99:L(K)=0:NEXTK
FOR K=1TO300:T(K,1)=0:T(K,2)=0:NEXTK
HOS=0:NT=0:NL=0
FOR K=0TO 100:S$(K)="000":U$(K)="":NEXT K
OPEN15,8,15
SP$="          "
REM CLS
PRINT1,"UJ TANANYAG FELVITEL"
PRINT2,"TANANYAG BETOLTES LEMEZROL"
PRINT3,"DISPLAY"
PRINT4,"TANANYAG MODOSITAS"
PRINT5,"TANANYAG LEMEZRE VITELE"
PRINT9,"VEGE"
INPUT "A VALASZTOTT TEVEKENYSEG SZAMA=";W$
IF W$="9"THEN GOTO390
IF W$<"1"ORW$>"5"THEN GOTO145
ITEV=VAL(W$)
REM PRINTFRE(0)
ON ITEV GOSUB 2500,500,800,950,600
GOTO145
CLOSE15:      END
REM BETOLT
INPUT"KEREM A FILE NEVET";FIN$
OPEN2,8,2,FIN$+",S,R"
INPUT#2,HOS,NL
FOR K=1 TO 99:INPUT#2,L(K)
REM IF K<=NL THEN PRINTL(K);
NEXT K
INPUT#2,NT
FOR K=1 TO 300:INPUT#2,T(K,1),T(K,2)
REM IF K<=NT THEN PRINTT(K,1);T(K,2);
NEXT K
FOR K=0 TO100
INPUT#2,S$(K)
REM PRINTK,S$(K)
NEXT K
CLOSE2
UJS=HOS+1
RETURN
REM FELVITEL
IF NL<1 THEN PRINT"NINCS BENT TANANYAG !!!": RETURN
INPUT"KEREM A FILE NEVET";FIN$
CS$="UJ FILE":GOSUB1510
IF T1%=0THEN PRINT#15,"S0:"+FIN$
OPEN2,8,2,FIN$+",S,W"
HOS=UJS-1
PRINT#2,HOS", "NLCHR$(13)
FOR K=1 TO 99:PRINT#2,L(K)", "
NEXT K
PRINT#2,NTCHR$(13)
FOR K=1 TO 300:PRINT#2,T(K,1)", "T(K,2)", "
NEXT K:LL$=CHR$(34)
FOR K=0 TO100
REM PRINTK,S$(K)
PRINT#2,LL$S$(K)LL$CHR$(13)
NEXT K
CLOSE2
RETURN

```

```

REM DISPLAY
IF NL<1THENPRINT"NINCS BENT TANANYAG!!!" :RETURN
HCX=0:CS$="NYOMYTATORA IS":GOSUB1510
IF T1Z=1 THEN HCX=1:OPEN4,4
FOR I=1 TO NL
IFL(I)<0 AND I=NL THEN GOTO 880
IF L(I)<0 THEN ZUR=2:GOSUB1500
PRINTI;".LAP KOVETKEZIK.  LAPTIPUS=";:INDEX=T(L(I),2):GOSUB6100
PRINTCH$
INPUT"MEHET";W$
GOSUB 1360
INPUT"MEHET TOVABB  ";W$
NEXT I
PRINT"A TANANYAG HELYIGENYE:"HOS"BYTE"
IF HCX=1THEN CLOSE4
RETURN
REM MODOSITAS
IF NL<1THENPRINT"NINCS BENT TANANYAG!!!" :RETURN
FOR I=1 TO NL:REM CLS
IFL(I)<0 AND I<>NL THEN ZUR=2:GOSUB1500
IFL(I)<0 THEN GOTO 1090
PRINTI;".LAP KOVETKEZIK.  LAPTIPUS=";:INDEX=T(L(I),2):GOSUB6100
PRINTCH$
INPUT "MEHET";W$
GOSUB 1360
CS$="AKARJA A LAP TARTALMAT MODOSITANI":GOSUB 1510
IF T1Z=0 THEN GOTO1085
GOSUB 1600
NEXT I
REM GOSUB 3000 REORG
CS$="FOLYTATO LAP FELVITELE:  ":GOSUB 1510
IF T1Z=0 THEN GOTO1190
NL=NL-1
GOSUB 2610
RETURN
REM TERKEP
REM DIM L(99),T(300,2)
L(1)=1:T(1,2)=1:NT=1
FOR I=1 TO 5000:
5 INDEX=I:GOSUB6100
IF CH$="#" THEN NT=NT+1:T(NT,2)=I+1:GOTO1300
IF CH$="@" THEN T(NT,1)=-1:NL=NL+1:L(NL)=NT+1:GOTO1300
REM
IF CH$="&" THENL(NL)=-1:HOS=I:GOTO1320
NEXT I
GOSUB 5000
RETURN
REM LAPDISP
2 FOR IXX=1 TO 5:PRINT :NEXT IXX
5 TAKT=L(I):NOV=1
7 SZA=0
GOSUB 1400: REM SORDISP
2 SZA=SZA+1
5 TAKT=T(TAKT,1): NOV=0
IF T(TAKT,1)>0 THEN GOTO 1370
5 FOR IXX=SZA TO 20:PRINT :NEXT IXX
0 RETURN

```



```

0 REM SORDISP
0 SA=T(TAKT,2)
1 JL=SA+NOV:GOSUB 4000
3 IF LEN(G$)>32 THEN ZUR=1:GOSUB1500
4 PRINT G$
5 IF HCX=1 THEN PRINT#4,RIGHT$(" "+STR$(SZA+1),4)". "G$
9 RETURN
0 G$=""
5 IG=1:ZUR=3
0 FOR J=SA+NOV TO SA+100
5 IF IG>32 THEN ZUR=1:GOSUB1500
0 INDEX=J:GOSUB 6100
5 IF CH$="#" THEN PRINTG$: RETURN
0 IG=IG+1
0 G$=G$+CH$
0 NEXT J
0 REM ZUER RUTIN
5 PRINT"ZUER=",ZUR: STOP
0 REM KERDEZ
0 PRINTCS$+"(I/N)";:INPUTW$
0 IFW$="I"ORW$="Y"THEN T1%=1:RETURN
0 IFW$="N"THEN T1%=0:RETURN
5 GOTO1520
9 RETURN
0 REM TALLOC
0 H1=0
0 IF NT+1>300 THENPRINT"TUL HOSSZU LESZ AZ ANYAG":H1=1: RETURN
0 RETURN
0 REM LAPMOD
0 IAKT=I
0 CS$="BESZURAS":GOSUB1510
0 IF T1%=0 THEN GOTO 2000
0 GOSUB 1550:IF H1=1 THEN GOTO 1650
0 INPUT "HANYADIK SOR UTAN";SU
0 IFSUC1THEN GOTO1680
0 REM IF SU=0 THEN NT=NT+1:T(NT,1)=L(IAKT):L(IAKT)=NT:T(NT,2)=UJS:GOTO1770
0 SS=1:TA=L(IAKT)
0 IF SU<SS THEN GOTO1680
0 IF SU=SS THEN NT=NT+1:T(NT,1)=T(TA,1):T(TA,1)=NT:T(NT,2)=UJS:GOTO1770
0 IF T(TA,1)>0 THEN SS=SS+1:TA=T(TA,1):GOTO1720
0 REM UTOLSO MOGE
0 NT=NT+1:T(NT,1)=T(TA,1):T(NT,2)=T(TA,1):T(TA,1)=NT:T(TA,2)=UJS+1:GOTO1770
0 REM SBE
0 GOSUB 1850
0 GOSUB 1360
5 REM GOSUB 5000
0 CS$="TOVABBI MODOSITAS": GOSUB 1510
0 IF T1%=1 THEN GOTO 1650
0 RETURN
0 REM SORKEZ
5 W$="":PRINT" A SOR UJ TARTALMA=":INPUT W$
0 H1=0:IF LEN(W$)>32 THEN H1=2:GOTO 1890
0 IF LEN(W$)=0 THEN W$=" "
5 FOR J=1 TO LEN(W$)
0 CX$=MID$(W$,J,1)
0 IF CX$="#" OR CX$="@" OR CX$="&" ORCX$=CHR$(34) THEN H1=1
0 NEXT J
5 IF H1=0 THEN GOTO 1900
0 IF H1=1 THEN PRINT "A SOR ILLEGALIS KARAKTERT TARTALMAZ !!!"
5 IF H1=2 THEN PRINT "A SOR TUL HOSSZU !!!"
0 GOTO 1850

```

```

2 FOR J=1 TO LEN(W$)
3 INDEX=UJS:CH$=MID$(W$,J,1):GOSUB6000
5 UJS=UJS+1
7 NEXT J
5 INDEX=UJS:CH$="#":GOSUB6000
7 UJS=UJS+1
9 RETURN
0 REM
5 PRINT" HIBAS SORSZAM":RETURN
0 REM MOD,STB
0 LET CS$="TORLES":GOSUB1510
0 IF T1X=0 THEN GOTO2200
0 INPUT"HANYADIK SORT ";SU
0 IF SU<=1 THEN GOSUB 1980:GOTO 2010
0 SS=1:TA=L(IAKT)
0 IF SU<SS THEN GOSUB 1980:GOTO2010
0 IF SU>SS THEN :GOTO2160
0 REM MEGVAN
0 IF SU>1 THEN GOTO2130
0 REM 1. SOR
0 IF T(L(IAKT),1)>0 THEN L(IAKT)=T(L(IAKT),1):GOTO2150
0 T(SE,1)=T(TA,1)
0 GOTO1790
0 REM SU>SS
0 SE=TA
0 IF T(TA,1)>0 THEN SS=SS+1:TA=T(TA,1):GOTO2060
0 GOSUB 1980 : GOTO2010
0 CS$="MODOSITAS": GOSUB 1510
0 IF T1X=0 THEN GOTO 2270
0 INPUT "HANYADIK SORT";SU
0 SS=1:TA=L(IAKT)
0 IF SU=SS THEN T(TA,2)=UJS:GOSUB 1850: GOTO 1790
0 IF T(TA,1)>0 THEN SS=SS+1:TA=T(TA,1): GOTO 2240
0 GOSUB 1980: GOTO2200
0 GOTO1790
0 REM TANANYAG FELV
0 IF HOS=0 THEN GOTO2600
0 CS$="ELOZO TANANYAG TOROLHETO":GOSUB 1510
0 IF T1X=0 THEN RETURN
0 HOS=0:NL=0:NT=0:UJS=0
0 NL=NL+1:L(NL)=NT+1
0 PRINTNL;".LAP KOVETKEZIK"
5 D$="":INPUT "LAPTIPUS=";D$
0 IF LEN(D$)<>1 THEN GOTO2620
0 IF D$="*" THEN GOTO2920
5 IFD$<>"A"ANDD$<>"B"THEN GOTO2625
0 IS=1
0 NT=NT+1: T(NT,1)=NT+1:T(NT,2)=HOS+1
0 IF IS=1 THEN HOS=HOS+1:INDEX=HOS:CH$=D$:GOSUB 6000
0 UJS=HOS+1:GOSUB 1850: HOS=UJS-1
5 IF IS>=20 THEN GOTO 2710
0 IS=IS+1: PRINT IS;".SOR ";:CS$="LESZ": GOSUB 1510:PRINT
0 IF T1X=1 THEN GOTO 2660
0 NT=NT+1:T(NT,1)=-1:HOS=HOS+1:INDEX=HOS:CH$="@" :GOSUB 6000:T(NT,2)=HOS
5 I=NL:GOSUB 1360
0 GOTO 2610
0 L(NL)=-1:HOS=HOS+1:INDEX=HOS:CH$="&": GOSUB 6000
0 REM OSUB 5000
0 RETURN

```



```

0 REM REORG
0 GOSUB 5000
0 INDEX=0
0 AHA=HOS: IF UJS>AHA THEN AHA=UJS
0 FOR K=0 TO 100
0 U$(K)=S$(K)
5 NEXT K
0 FOR I=1 TO NL
0 IF L(I)<0 THEN GOTO3200
0 IA=L(I): IF T(IA,1)<0 THEN GOTO3160
0 UX=T(IA,2): GOSUB6200
0 INDEX=INDEX+1: GOSUB 6000
0 IF CH$<>"#" THEN UX=UX+1: GOTO3140
5 IA=T(IA,1): IF IA>0 THEN GOTO3120
0 UX=T(IA,2)
0 GOSUB 6200: IF CH$<>"@" THEN STOP
0 INDEX=INDEX+1: GOSUB6000
0 NEXTI
0 INDEX=INDEX+1: CH$="&": GOSUB 6000
0 HOS=INDEX
5 GOSUB 1200
0 GOSUB 5000
0 RETURN
0 REM GYORSKER
0 REM PAR: JL,G$
0 IF JL<1 THEN PRINT"STOP 4020": STOP
0 JJ=INT((JL-1)/50): JK=JL-50*JJ
0 IF JK<1 THEN PRINT"STOP 4040": STOP
0 GG$=MID$(S$(JJ),JK,51-JK)+S$(JJ+1)
0 FOR IC=1 TO 70
0 IF MID$(GG$,IC,1)="#" THEN GOTO 4100
0 NEXT IC
0 G$=LEFT$(GG$,IC-1)
0 RETURN
0 REM KUKUCS
0 PRINT"NL,NT,HOS,UJS=",NL,NT,HOS,UJS
5 PRINT"L TAB="
0 FOR K=1 TO NL: PRINT" ";L(K);: NEXT K: PRINT
5 PRINT"T TAB="
0 FOR K=1 TO NT: PRINT T(K,1),T(K,2): NEXT K
5 INPUT"MEHET";X$
0 FOR IC=0 TO 99
0 PRINT S$(IC)
5 IF INT(IC/20)*20=IC THEN INPUT"MEHET";X$
7 IF IC*100>HOS THEN GOTO 5290
0 NEXT IC
0 RETURN
0 REM SBE (INDEX,CH$)
0 IF INDEX<10 OR INDEX>5000 THEN PRINT"HIBA6": RETURN
0 IF LEN(CH$)<>1 THEN PRINT"HIBA7": RETURN
5 JJ=INT((INDEX-1)/50): JK=INDEX-50*JJ
8 IF JK<1 THEN PRINT"STOP 6028": STOP
0 IF JK=1 THEN S$(JJ)=CH$: GOTO6090
0 S$(JJ)=LEFT$(S$(JJ),JK-1)+CH$
0 RETURN

```


TAFEL

6.LAP

```
0 REM SBOL(INDEX,CH$)
0 IF INDEX<10RINDEX>5000 THEN PRINT"HIBA6":RETURN
5 JJ=INT((INDEX-1)/50):JK=INDEX-50*JJ
8 IF JK<1THENPRINT"STOP 6118":STOP
0 CH$=MID$(S$(JJ),JK,1)
0 RETURN
0 REM UBOL( UX,CH$)
0 IF UX<10R UX>5000 THEN PRINT"HIBA6":RETURN
5 JJ=INT(( UX-1)/50):JK= UX-50*JJ
8 IF JK<1THENPRINT"STOP 6218":STOP
0 CH$=MID$(U$(JJ),JK,1)
0 RETURN
```

DY.

```

REM TAFEL/KERD/7.21.
DIM K$(100),DT(100),TL%(20)
DIM DP$(100),DJ$(100),DM$(100)
SZ$(1)="HELYES VALASZ"
SZ$(2)="PONTERTEK"
SZ$(3)="NEUTRALIS VALASZ"
SZ$(4)="EGY KORBEN KERDEZENDO PONTERTEK"
SZ$(5)="MINIMALIS SZINT(%)"
SZ$(6)=" J O SZINT(%)"
SZ$(7)="IGEN JO SZINT(%)"
OPEN15,8,15:AL%=0
PRINT1,"UJ KERDESCSOPORT NYITAS"
PRINT2,"KERDESCSOPORT TOLTES LEMEZROL"
PRINT3,"KERDESCSOPORT DISPLAY"
PRINT4,"KERDESCSOPORT MODOSITAS"
PRINT5,"KERDESCSOPORT LEMEZRE VITELE"
REM PRINT8,"KUKUCS"
PRINT9,"VEGE"
W$="":INPUT "A VALASZTOTT TEVEKENYSEG SZAMA=";W$
IF W$="9"THEN GOTO 280
IF W$="8"THEN GOTO 250
IF W$<"1"ORW$>"5"THEN GOTO200
ON VAL(W$)GOSUB 2500,500,800,5100,600,3000,3100,4500
GOTO 200
REM VEGE
CLOSE15
END
REM DISCTL
INPUT#15,X1,X2$,X3,X4
IFX1<20 THEN RETURN
PRINTX1,X2$,X3,X4
RETURN
REM BETOLT
INPUT"KEREM A FILE NEVET";FI$
OPEN2,8,2,FI$+","S,R":GOSUB310
FORI=1TO8:INPUT#2,CL%(I):NEXTI
FORI=1TO100:INPUT#2,DP$(I),DJ$(I),DM$(I),DT(I)
NEXTI
FORI=0TO100:INPUT#2,K$(I)
NEXTI
AL%=1
CLOSE2
RETURN
REM KIVISZ
IFAL%=0THEN GOSUB 8000:RETURN
INPUT"KEREM A FILE NEVET";FI$
CS$="UJ FILE":GOSUB 1310
IF T1=0 THEN PRINT#15,"S0:"+FI$
OPEN2,8,2,FI$+","S,W":GOSUB310
FORI=1TO8:PRINT#2,CL%(I),"":NEXTI
FORI=1TO100:PRINT#2,DP$(I),"DJ$(I)","DM$(I)","DT(I)",""
NEXTI
L$="":LL$=CHR$(34)
FORI=0TO100:IFI=100 THENL$=CHR$(13)
PRINT#2,LL$K$(I)LL$L$
NEXTI
CLOSE2
RETURN

```

```

REM DISPLAY
IF ALZ=0 THEN GOSUB 8000:RETURN
PRINT"KERDESEK SZAMA="CLZ(1)
PRINT"EGY KORBEN"CLZ(2) "PONTNYIT KERDEZUNK"
PRINT"MINIMALIS SZINT(%)="CLZ(3)
PRINT"  J O      SZINT(%)="CLZ(4)
PRINT"IGEN JO    SZINT(%)="CLZ(5)
REM PRINT"FREEK      ="CLZ(6)
INPUT "HANYADIK KERDESTOL KERI";CS$
GOSUB 1010:IF H1>0ORS1<1ORS1>CLZ(1) THEN GOTO850
KP=S1
IF KP>CLZ(1) THEN GOTO 890
GOSUB 8100:REM KERDDISP
CS$="KERI A KOVETKEZOT":GOSUB 1310
IF T1=1 THEN KP=KP+1:GOTO 860
RETURN
0 REM ICTL(CS$,H1,S1)
5 H1=0:S1=0
0 IF LEN(CS$)>9 THEN H1=1:RETURN
5 FOR J=1 TO LEN(CS$)
0 IF MID$(CS$,J,1)<"0"OR MID$(CS$,J,1)>"9" THEN H1=H1+1:RETURN
5 S1=VAL(CS$)
0 RETURN
0 REM KERD(CS$,T1)
5 PRINTCS$+"(Y/N)=";:W$="":INPUTW$
0 IFW$="N" THEN T1=0:RETURN
5 IFW$="I"ORW$="Y" THEN T1=1:RETURN
0 GOTO1310
0 REM UJCSOP
0 FORI=1TO8:CLZ(I)=0:NEXTI
5 CLZ(6)=1
0 FOR I=1TO100
0 DPZ(I)=0:DMZ(I)=0:DJ$(I)="
0 K$(I)="":DT(I)=0
5 NEXTI
0 K$(0)="
0 PRINT"EGY KORBEN HANY PONTNYIT KERDEZUNK":CS$="":INPUT CS$
0 GOSUB 1010
0 IF H1>0ORS1<1 THEN GOTO 2570
0 CLZ(2)=S1
0 PRINT"MINIMALIS TUDASSZINT(%-BAN)":CS$="":INPUT CS$
0 GOSUB 1010
0 IF H1>0ORS1<1ORS1>99 THEN GOTO 2610
0 CLZ(3)=S1
0 PRINT"  J O      TUDASSZINT(%-BAN)":CS$="":INPUT CS$
0 GOSUB 1010
0 IF H1>0ORS1<1ORS1>99 THEN GOTO 2650
5 IF S1<CLZ(3) THEN GOTO 2650
0 CLZ(4)=S1
0 PRINT"IGEN JO    TUDASSZINT(%-BAN)":CS$="":INPUT CS$
0 GOSUB 1010
0 IF H1>0ORS1<1ORS1>100 THEN GOTO2690
5 IF S1<CLZ(4) THEN GOTO 2690
0 CLZ(5)=S1
0 CS$=STR$(CLZ(1)+1)+".KERDES VAN"
5 GOSUB1310
0 IF T1=0 THEN GOTO2980
0 KP=CLZ(1)+1:GOSUB 8600:REM KERDFELY.
0 CLZ(1)=CLZ(1)+1:GOTO2750
0 ALZ=1:RETURN

```



```

3 REM TEV6
3 RETURN
3 REM TEV7
3 RETURN
3 REM MEGNEZ
7 GOTO 4515
0 IF ALX=0 THEN GOSUB 8000
2 PRINT "CLX="
5 FOR I=1 TO 8: PRINT CLX(I);: NEXT I: PRINT
7 PRINT "FEJTAB="
0 FOR I=1 TO 100:
5 PRINT DPX(I), DJ$(I), DMX(I), DT(I)
0 IF INT(I/20)*20=I THEN INPUT "MEHET"; W$
5 NEXT I
7 PRINT "K$TAB="
0 FOR I=0 TO 100
5 PRINT K$(I)
0 IF INT(I/10)*10=I THEN INPUT "MEHET"; W$
0 NEXT I
0 RETURN
0 REM FRE
0 REM PRINT "FRE=" FRE(0)
0 RETURN
0 REM MOD
5 REM PRINT "MOD"
8 IF ALX=0 THEN GOSUB 8000: RETURN
0 PRINT 1, "GLOBAL ADATOK MODOSITAS"
5 PRINT 2, "UJ KERDES FELVETEL"
0 PRINT 3, "KERDES TORLESE"
5 PRINT 4, "KERDES MODOSITAS"
0 PRINT 9, "MODOSITASOK VEGE "
5 INPUT "A VALASZTOTT TEVEKENYSEG SZAMA="; W$
0 IF W$="9" THEN GOTO 5190
5 IF W$<"1" OR W$>"4" THEN GOTO 5110
0 ON VAL(W$) GOSUB 5300, 5400, 5500, 5600
5 GOTO 5100
0 RETURN
0 REM GLOBAL MOD
0 FOR I=2 TO 5
5 PRINT SZ$(I+2)="CLX(I)
0 CS$="MODOSUL": GOSUB 1310
02 IF T1=0 THEN GOTO 5380
05 CS$="": PRINT "UJ ERTEK=": INPUT CS$
05 IF LEN(CS$)<1 THEN GOTO 5315
00 GOSUB 1010: IF H1>0 THEN GOTO 5315
05 IF I=2 AND S1<1 THEN GOTO 5315
00 IF I=3 AND S1>100 THEN GOTO 5315
05 IF I=4 AND S1>100 THEN GOTO 5315
00 IF I=4 AND S1<CLX(3) THEN GOTO 5315
05 IF I=5 AND S1<CLX(4) THEN GOTO 5315
00 IF I=5 AND S1>100 THEN GOTO 5315
05 CLX(I)=S1
00 NEXT I
00 RETURN

```

ADY.

```

0 REM UJ KERD FELV.
0 FOR I=1 TO CLZ(1)
5 IF DMZ(I)<1 THEN KP=I:GOSUB 8600: GOTO 5490
0 NEXT I
0 KP=CLZ(1)+1:GOSUB8600
0 CLZ(1)=CLZ(1)+1
0 RETURN
0 REM TORLES
0 W$="":INPUT"HANYADIK KERDEST";W$
5 IF LEN(W$)<1 THEN GOTO 5510
0 CS$=W$:GOSUB1010
5 IF H1>0 THEN GOTO 5510
0 IF S1<10RS1>99 THEN GOTO 5510
5 KP=S1: GOSUB8100
7 IF DMZ(KP)<1 THEN GOTO5590
0 CS$="TORLES MEHET":GOSUB1310
0 IF T1=1 THEN DMZ(KP)=0
0 RETURN
0 REM SZOVEGMOD
0 CS$="":INPUT"HANYADIK KERDES MODOSUL";CS$
5 IF LEN(CS$)<1 THEN GOTO 5610
0 GOSUB1010:IFH1>0ORS1<10RS1>99 THEN GOTO5610
5 KP=S1
0 GOSUB8100:REM KERG.DISP.
5 IF DMZ(KP)<1 THEN GOTO5790
0 CS$="ERTEKELES RESZ MODOSUL":GOSUB 1310:IF T1=0 THEN GOTO 5710
2 JJ$=LEFT$(DJ$(KP),1)
4 JN$=" ":IFLEN(DJ$(KP))>1 THENJN$=MID$(DJ$(KP),2,1)
0 PRINTSZ$(1)="$JJ$
5 CS$="MODOSUL":GOSUB 1310:IF T1=0 THEN GOTO 5665
7 W$="":INPUT"UJ ERTEK=";W$
0 IF LEN(W$)<1 THEN GOTO 5650
2 JJ$=W$
5 PRINTSZ$(3)="$JN$
7 CS$="MODOSUL":GOSUB 1310:IF T1=0 THEN GOTO 5674
0 W$="":INPUT"UJ ERTEK=";W$:JN$=W$
2 IF LEN(W$)<1 THEN GOTO 5665
4 DJ$(KP)=JJ$+JN$
5 PRINTSZ$(2)="$DPZ(KP)
0 CS$="MODOSUL":GOSUB 1310:IF T1=0 THEN GOTO 5710
5 CS$="":INPUT"UJ ERTEK=";CS$
0 IF LEN(CS$)<1 THEN GOTO 5675
5 GOSUB1010:IFH1>0ORS1<10RS1>100 THENGOTO5675
0 DPZ(KP)=S1
0 CS$="KERDES SZOVEGE MODOSUL":GOSUB 1310:IF T1=0 THEN GOTO 5790
5 GOSUB 7500:REM TERKEP(KP,TLZ,SV)
0 CS$="":INPUT"HANYADIK SOR";CS$
5 IF LEN(CS$)<1 THEN GOTO 5720
0 GOSUB1010:IFH1>0ORS1<10RS1>20 THENGOTO5720
5 SS=S1:GOSUB 7600:REM SORMOD(****)
8 GOSUB 8100 :REM KERG.DISP
0 CS$="VAN MEG MODOSITANDO SOR":GOSUB 1310:IF T1=1 THEN GOTO 5715
0 RETURN

```

DY.




```

0 REM SORKER(IX,G$,PU,PP)
5 PU=-2:PP=-2:G$="@@@@@@"
0 IF IX<1 OR IX>5000 THEN STOP
0 T1=INT((IX-1)/50):T2=IX-50*T1
5 IF T2<1 THEN STOP
0 CW$=MID$(K$(T1),T2,51-T2)+K$(T1+1)
0 HS=VAL(LEFT$(CW$,2))
0 G$=MID$(CW$,3,HS)
0 PU=VAL(MID$(CW$,HS+4,4))
0 PP=IX+LEN(G$)+3
0 RETURN
0 REM SORRAK(IX,G$,PU,IY)
0 CW$=RIGHT$(STR$(LEN(G$)),2)+G$+"#"
5 CW$=CW$+RIGHT$(" "+STR$(PU),4)
0 IF IX<1 OR IX>5000 THEN STOP
0 T1=INT((IX-1)/50):T2=IX-50*T1
5 IF T2<1 THEN STOP
0 K$(T1)=K$(T1)+CW$
0 HS=LEN(K$(T1))
0 IF HS<51 THEN GOTO 7285
0 K$(T1+1)=RIGHT$(K$(T1),HS-50)
0 K$(T1)=LEFT$(K$(T1),50)
5 IY=IX+LEN(CW$)
0 RETURN
0 REM PTRUP1(PV,PC)
5 REM PRINT"PTRUP1",PV,PC
0 CW$=RIGHT$(" "+STR$(PV),4)
2 IF PC<1 OR PC>5000 THEN STOP
5 T1=INT((PC-1)/50):T2=PC-T1*50
0 IF T2=0 THEN STOP
5 HS=LEN(K$(T1))
0 CS$="+K$(T1)+K$(T1+1)+" "
5 CS$=LEFT$(CS$,T2)+CW$+RIGHT$(CS$,LEN(CS$)-T2-4)
0 K$(T1)=MID$(CS$,2,HS)
5 IF LEN(K$(T1+1))>0 THEN K$(T1+1)=MID$(CS$,HS+2,LEN(K$(T1+1)))
0 RETURN
0 REM ELRAK(KP,SS,Q$,FE)
0 IF SS=1 THEN DM$(KP)=FE:GOTO 7460
5 IX=DM$(KP):S1=1
0 GOSUB 7000:REM SORKER(IX,G$,PU,PP)
5 IF PU=-2 THEN STOP
0 IF S1+1<SS THEN S1=S1+1:IX=PU:GOTO 7420
5 IF S1+1=SSANDPU<>0 THEN STOP
0 IF S1+1=SS THEN PV=FE:PC=PP:GOSUB 7300:REM PTRUP1(FE,PP)
0 IX=FE:G$=Q$:PU=0
0 GOSUB 7200:REM SORRAK(IX,G$,0,IY)
5 FE=IY
0 RETURN
0 REM TERKEP(KP,TL%,SV)
0 FOR I=1 TO 20:TL$(I)=0:NEXT I
5 SV=0
0 IF DM$(KP)<0 THEN GOTO 7590
5 TL$(1)=DM$(KP):SV=1
0 FOR I=2 TO 20
5 IX=TL$(I-1)
0 GOSUB 7000:REM SORKER(IX,G$,PU,PP)
3 TL$(I)=PU
5 IF PU<1 THEN SV=I-1:GOTO 7590
0 NEXT I
5 SV=20:STOP
0 REM
0 RETURN

```

```

3 REM SORMOD(*****)
3 IF SS>SV+1 THEN PRINT"NINCS ILYEN SOR!!": GOTO 7690
2 IF SS=SV+1 THEN PRINT"UJ SORT VESZEK FEL.": GOTO 7700
5 IX=TL%(SS):IF IX<1 THEN STOP
3 GOSUB 7000:REM SORKER(IX,G$,PU,PP)
5 PA=PU:PRINTG$
3 IF SS=1 THEN DM%(KP)=CL%(6):GOTO 7660
5 IX=TL%(SS-1)
3 GOSUB 7000:REM SORKER(IX,G$,PU,PP)
5 PV=CL%(6):PC=PP
3 GOSUB 7300:REM PRTUP(PV,PC)
0 W$=""
5 PRINT"A SOR UJ TARTALLMA=":INPUT W$
3 GOSUB 8800:REM SORELL(W$,H1)
5 IF H1>0 THEN GOTO 7660
0 IX=CL%(6):PU=PA:G$=W$
3 GOSUB 7200:REM SORRAK(IX,G$,PU,IY)
5 CL%(6)=IY
0 RETURN
0 IX=TL%(SV):IF IX<1 THEN STOP
3 GOSUB 7000:REM SORKER(IX,G$,PU,PP)
0 PA=PU:IF PA<>0 THEN STOP
0 GOTO 7645
0 REM
0 PRINT"NINCS BENT KERDESCSOPORT!!!"
0 RETURN
0 REM KERDISP(KP)
5 PRINT:PRINT:PRINT:PRINTTAB(12)KP".KERDES"
7 IF DM%(KP)<1 THEN PRINT"TOROLVE": GOTO 8190
3 GOSUB 8200:REM KERTORDISP
5 PRINTSZ$(1)="DJ$(KP),SZ$(2)":"DP%(KP)
0 RETURN
0 REM KERTORDISP
0 IX=DM%(KP):IL=0
0 REM SORKEZ
3 GOSUB 7000:REM SORKER(IX,G$,PU)
0 PRINTG$:IX=PU:IL=IL+1
5 IF PU>0 THEN GOTO 8230
0 FORI=IL+1 TO 20:PRINT:NEXTI
0 RETURN
0 REM VESZ1(IX+IG,CH$)
0 X3=IX+IG
5 IF X3<10 OR X3>5000 THEN GOSUB 8400
0 T1=INT((X3-1)/50):T2=X3-50*T1
5 IF T2<1 THEN HB=8:GOSUB 8400
0 CH$=MID$(K$(T1),T2,1)
0 RETURN
0 REM CANCEL
0 PRINT"HIBA VAN A KRETA KORUL"
0 STOP
0 REM VESZ2(IY,X1)
0 X2$=""
5 FORJ=0 TO 3
0 X3=IY+J:GOSUB 8315
5 X2$=X2$+CH$
0 NEXTJ
5 X1=VAL(X2$)
0 RETURN

```



```
0 REM KERDFELV.(KP)
0 PRINT"A SZOVEGET KEREM"
0 FOR SS=1 TO20
5 CS$=STR$(SS)+".SOR VAN":GOSUB1310
0 IF T1=0THEN GOTO 8680
5 PRINT"A SOR TARTALMA=":W$="":INPUTW$
0 IF LEN(W$)>32THEN PRINT"A SOR TUL HOSSZU!!":GOTO 8635
5 IF LEN(W$)=0 THEN W$=" "
0 H1=0:FOR J=1TO LEN(W$):CS$=MID$(W$,J,1)
5 IF CS$="#"ORCS$="@"ORCS$="&"ORCS$=CHR$(34)THEN H1=1
0 NEXTJ
5 IF H1>0 THENPRINT"A SOR ILLEGALIS JELET TARTALMAZ!":GOTO8635
0 Q$=W$:FE=CLZ(6):GOSUB 7400:REM ELRAK(KP,SS,Q$,FE)
5 CLZ(6)=FE
7 NEXT SS
0 PRINTSZ$(1);:W$="":INPUTW$
2 IF LEN(W$)<>1THEN GOTO 8680
4 JJ$=W$
6 PRINTSZ$(3);:W$="":INPUTW$
8 IF LEN(W$)<>1THEN GOTO 8686
0 DJ$(KP)=JJ$+W$
5 PRINTSZ$(2);:W$="":INPUTW$
0 CS$=W$:GOSUB1010:IFH1>0ORS1<1ORS1>100 THEN GOTO8695
0 DP$(KP)=S1
0 REM
0 REM
0 RETURN
0 REM SOREL(W$,H1)
5 H1=0
0 IF LEN(W$)>32THEN PRINT"A SOR TUL HOSSZU!!":H1=2:GOTO8890
0 IF LEN(W$)=0 THEN W$=" "
0 FOR J=1TO LEN(W$):CS$=MID$(W$,J,1)
0 IF CS$="#"ORCS$="@"ORCS$="&"ORCS$=CHR$(34)THEN H1=1
0 NEXTJ
5 IF H1>0 THENPRINT"A SOR ILLEGALIS JELET TARTALMAZ!":GOTO8890
0 RETURN
```

DY.

2. SZ. MELLÉKLET

TANANYAGOK
ISMERETKÖZLŐ RÉSZEK

1. LECKE

ALTALANOS ISMERETEK A BASIC NYELVRŐL

Ebben a leckében a kurzus során megtanulandó BASIC nyelv általános, szerkezeti tulajdonságaival foglalkozunk.

A BASIC egy egyszerű, viszonylag könnyen megtanulható, elterjedt

PROGRAMOZÁSI NYELV;

szinte minden személyi számítógép programozható BASIC nyelven.

A különböző személyi számítógéptípusokon használható BASIC nyelvek (un. IMPLEMENTÁCIÓK) nem azonosak, kisebb-nagyobb mértékben eltérnek egymástól.

A kurzus során a továbbiakban BASIC nyelv alatt a

COMMODORE 64

személyi számítógéphez tartozó BASIC nyelvet értjük.

BAS01.3

A BASIC Programok
BASIC SOROKBÓL
állnak.

Egy BASIC sor
- SORSZÁMBÓL és
egy vagy több
- BASIC UTASÍTÁSBÓL
áll.

Ha egy sorban több utasítás áll,
akkor azokat KETTŐSPONT (:)
választja el egymástól.

P1.

```
10 LET B=3  
20 INPUT A:PRINT A
```

A fenti Programrészlet két
BASIC sorból áll.

Az első sor sorszáma 10; a másodike 20.

Az első sor egy BASIC utasítást tartalmaz (LET B=3);
a második kettőt
(INPUT A és PRINT A).

A BASIC utasításai közül kímél-
 juk a REM kulcsszóval kezdődő,
 ún. MEGJEGYZÉS típusú utasításo-
 kat az alábbiak miatt.

- A sorozám után REM-mel kezdő-
 dó sorok végig egyetlen uta-
 sítást alkotnak, függetlenül
 attól, hogy tartalmaznak-e
 kétszponot.

- Ezen utasítások tartalmaznak nem
 befolyásolja a program végre-
 hajtasát, csak a programot
 olvasó SZEMÉLVÉKNEK szólnak,
 segítve a program érthetősé-
 gét, követhetőséget.

A további leckeekben megismer-
 kedünk a BASIC nyelv alapvető
 utasításaitípusaival és az
 utasítások programmá való
 összerpitésének szabályaival.

BASIC SORSZÁMOK
UTASÍTÁS KULCSSZÓ
UTASÍTÁS TÍPUSOK

A BASIC sorok sorszáma 0 és 9999 közé eső egész szám lehet. Különböző soroknak nem lehet azonos sorszáma, azaz egy Programon belül nem lehetnek azonos sorszámok. A sorokat általában 10-től kezdve tízesével sorszámozzuk, ami azért célszerű, mert így a későbbiekben új utasításokat a meglevők átsorszámozása nélkül szűrhatunk be.

BAS02.3

A sorszámok meghatározzák a BASIC sor helyét a Programlistán.

A Programlista a sorokat
SORSZÁMNÖVEKVŐ RENDBEN
tartalmazza.

A sorszámoknak fontos szerepe van az utasítások végrehajtási sorrendjének kialakításában.

Példák helyes sorszámozásra:

9...	0...	896...
23...	2...	1031...
91...	4...	1456...
92...	5...	1514...

Példák hibás sorszámozásra:

1910...	116...	-1...
1920...	121...	0...
1930...	121.5...	1...
1930...	122...	2...
azonos sorszámú sorok	a sorszám nem egész	a sorszám negatív

BAS02.5

A BASIC utasításokat az
UTASITAS KULCSSZÓ azonosítja.

Az utasítások az utasítás
kulcsszóval kezdődnek.

A megjegyzés utasítások
kulcsszava: REM
(Az angol "remark" szó rövidi-
tése.)

A nem végrehajtható utasítások
kulcsszava: DIM
(Az angol "dimension"=méret,
kiterjedés jelentésű szó rö-
vidítése.)
A többi kulcsszó végrehajtható
utasításokat vezet be.

Példaként a leggyakoribb
kulcsszavak:

LET, PRINT, IF, PAUSE, STOP,
END, INPUT, GO TO, FOR, NEXT

A BASIC utasításokat UTASÍTÁSTÍPUSONKÉNT

tekintjük át.

Egy típusba az azonos kulcsszó-
szavú utasítások tartoznak.

Egy típusba tartozik tehát
az összes megjegyzés utasítás,
mivel a kulcsszó minden eset-
ben REM.

Egy típusba tartozik tehát az
összes nem végrehajtható uta-
sítás, mivel a kulcsszó min-
dig a DIM.

A végrehajtható utasítások
viszont több utasítás típusba
tartoznak

P1. : értékadó /LET/;
feltételes /IF/;
vezérlésátadó /GOTO/.

BAS03.1

3. LECKE

BASIC Programok betöltése és indítása

A COMMODORE 64 személyi számítógép felhasználható:

- meglévő Programok végrehajtására (futtatásra)
- új Programok készítésére.

A meglévő Programok normál (magnó) kazettán vagy mágneslemezen helyezhetők el.

A Programokat mindkét esetben az adathordozóról a számítógép központi (operatív) memóriájába kell tölteni ahhoz, hogy futtathassuk.

A betöltött Program végrehajtása a RUN Parancs hatására kezdődik meg.

BETÖLTÉSKOR AZ ADATHORDOZÓ TARTALMA VALTOZATLAN MARAD.

A központi memóriában egyszerre csak egy Program lehet, azaz betöltéskor törlődik a memóriából a korábban bekerült Program.

Az adathordozókon minden Programnak egyedi neve (azonosítója) van. A továbbiakban ezt a nevet a "Prnev" jelöléssel jelezzük.

BAS03.3

A COMMODORE 64-en alkalmazható betöltési lehetőségek közül csak a legegyszerűbb esetekkel foglalkozunk.

I. Betöltés magnókazettáról.

1. Lépés: a kazetta behelyezése és pozicionálása.
2. Lépés: betöltési Parancs:
LOAD"Prnev",8
A PRESS PLAY ON TAPE üzenet megjelenésekor a 'Play' billentyűt kell lenyomni a magnón.
3. Lépés: futtatási Parancs:
RUN

II. Betöltés mágneslemezről

1. Lépés: a mágneslemez behelyezése.
2. Lépés: a tartalomjegyzék (katalógus) megtekintése.
Parancs: LOAD"\$",8
LIST
Válasz : -a lemez neve
 -a lemezen lévő
 = programok neve
 = szabad terület
 nagysága.
3. Lépés: betöltési Parancs:
LOAD"Prnev",8
4. Lépés: futtatási Parancs:
RUN

BAS03.5

Ha olyan Programot készítünk, amelyre máskor is szükségünk lehet, akkor

MAGNESES ADATHORDOZÓRA kell elhelyezni, hiszen a gép kikapcsolásakor az operatív memória tartalma törlődik.

A legegyszerűbb mentési esetek az alábbiak.

I. Mentés magnókazettára.

1. Lépés: a kazetta behelyezése és Pozicionálása.

2. Lépés: mentési Parancs:
SAVE"Prnev",8

A PRESS PLAY AND REC ON

TAPE Üzenet megjelenésekor a magnón a 'Play' és 'rec' billentyűket kell lenyomni.

3. Lépés: a felírt tartalom ellenőrzése.

A kazetta Pozicionálása.

Parancs: VERIFY "Prnev"

A PRESS PLAY ON TAPE Üzenet megjelenésekor a magnón le kell nyomni a 'Play' billentyűt.

Ha a felvitel sikeres volt, a READY. Üzenetet kapjuk.

BAS03.7

II. Mentés mágneslemezre.

1.Lépés: a mágneslemez betétele

2.Lépés: mentési Parancs:
SAVE"Prnev",8

A READY. üzenet megérkezése után:

3.Lépés: a felírt tartalom ellenőrzése.
Parancs: VERIFY"Prnev",8
Ha a felvitel sikeres volt, akkor a READY. üzenetet kapjuk.

A programokat a SAVE Parancs menti ki ténylegesen az adathordozóra, a VERIFY Parancs csak összehasonlítja a memória-ban lévő és az adathordozón található, "Prnev" nevű programokat.

A VERIFY Parancs nem változtatja meg az adathordozók tartalmát!

4. LECKE ADATTIPUSOK

A Programozás lényege az, hogy az ún. változókhoz értékeket rendelünk, amelyekből feltételesen vagy feltétlenül újabb és újabb értékeket származtatunk, s a változók értékeit kiíratjuk.

A BASIC-ben a változóknak neve és meghatározott TIPUSA van. A C-64 BASIC-ben az alábbi adattípusok használhatók:

- valós egyszerű változó
- valós tömb
- egész típusú egyszerű változó
- egész típusú tömb
- jelsorozatos egyszerű változó
- jelsorozat típusú tömb.

Az egész és a valós típusú változókat együtt, közös néven ARITMETIKAI VÁLTOZÓKNAK nevezzük.

A TÖMBÖK az egyszerű (skaláris) változók csoportjai, amelyekre a közös néven hivatkozhatunk.

Az egyszerű változókhoz és a tömbelemekhez a Program végrehajtása során értékeket rendelhetünk, amit a változó

AKTUALIS ÉRTÉKÉNEK nevezünk.

Amíg egy egyszerű változóhoz vagy tömbelemhez nem rendelünk értéket, addig ALAPÉRTELMEZÉS SZERINTI értéke van, ami aritmetikai változók esetén nulla, jelsorozat típusúaknál az üres jelsorozat ("").

BAS04.3

Ha egy egyszerű változónak vagy tömbelemnek már van értéke és újabb értéket rendelünk hozzá, akkor az utóbb hozzárendelt érték lesz az aktuális értéke, miközben a korábbi értéke elvész.

A változókhoz :

- adatbeviteli
- értékadó
- adat-átvevő

utasítástípusokkal rendelhetünk értékeket.

A VALÓS egyszerű változók értékei pozitív, negatív egészek és törtek, valamint a nulla érték lehet.

A valós számok értéktartománya
-38 és 38 között van.

A valós számok pontossága 9 értékes jegy.

A legkisebb ábrázolható pozitív
-39

szám kb $4 \cdot 10^{-39}$. Ez azt jelenti, hogy az ennél kisebb pozitív számokat a gép nullaként kezeli; ha pedig két szám eltérése ennél az értéknél kisebb, akkor a két számot azonosnak tekinti.

BAS04.5

A valós típusú tömbök elemei valós számok.
Ennek megfelelően ugyanazok a szabályok érvényesek a valós típusú tömbök elemeinek lehetőséges értékeire, mint a valós egyszerű változókra.

Egy EGÉSZ típusú egyszerű változó értéke pozitív vagy negatív egész szám, illetve a nulla érték lehet.

Az egész számok értéktartománya -32768 és +32768 között van.

Az egész típusú tömbök elemei egész számok, tehát az egész típusú tömbelemekre ugyanaz érvényes, mint az egyszerű egyszerű változók értékeire.

Egy JELSOROZAT típusú egyszerű változó értéke jelek (betűk, számok, különleges jelek) sorozata lehet. (Pl. ASDF!787*%)

A sorozat maximális hossza 255 jel.

A jelsorozat típusú tömbök elemei jelsorozatok, tehát az egyes tömbelemek értékeire ugyanaz érvényes, mint az egyszerű jelsorozat típusú változók lehetőséges értékeire.



BAS04.7

A változók nevére (elnevezésére) az alábbi szabályok érvényesek a COMMODORE 64-en:

- A valós (egyszerű vagy tömb) változók neveként ún. EGYSZERŰ NÉV használható.
- Az egész típusú (egyszerű vagy tömb) változók neve egy egyszerű névből és a típusjelző százalék(%) jelből áll.
- A jelsorozat típusú (egyszerű vagy tömb) változók neve egy egyszerű névből és a típusjelző dollár(\$) jelből áll.

- Az egyszerű név tetszőleges, maximum 6 alfanumerikus jelből álló, betűvel kezdődő jelsorozat; amely nem tartalmazhat BASIC kulcsszót (PI, REM, LET). Nem használható három, ún. fenntartott változónév, a TI, ST és TI\$ sem.

- A változók megkülönböztetése a COMMODORE 64-en csak a típusjel és a név első két jele alapján történik; azaz két azonos típusú változót nem különböztet meg a C-64 BASIC, ha nevük első két jele azonos.

BAS04.9

A szabályokat példákkal is szemléltetjük.

ABLAK T1 H4 H2SO4 szabályos egyszerű nevek, alkalmazhatók valós egyszerű változók vagy tömbök nevéként.

B.14 AB?LAK 5LET 613 szabálytalan (hibás) egyszerű nevek.

TIZ% Z4% EZER% szabályos;
K3.3% T!% L%% szabálytalan (hibás) nevek egész típusú változók jelölésére.

A\$ USA\$ KET100\$ szabályos;
BE.6\$ L\$\$ 3\$ T%\$ szabálytalan (hibás) nevek jelsorozatos típusú változók jelölésére.

Az A A\$ A% három különböző változót jelent (amelyeknek lehet egyidejűleg három különböző aktuális értéke).

Az ALMA ALFA ALBERT ugyanazt a változót jelenti (mivel első két jelük azonos).

A BELEP és a BETON% különböző változók, mert különböző típusúak.

BAS04.11

ADATTIPUSOK ÉRTÉKÉNEK ÍRÁSMÓDJA A PROGRAMOKBAN

A Programokban a VALÓS számokat

- egész
- tizedestört
- exponens

formában írhatjuk, általában
tetszés szerint.

P1. 1 és 1.000 ugyanazt az
értéket jelöli.

A számok EGÉSZ formája azonos
a matematikában megszokottal;
azaz számjegyek sorozata,
amelyet megelőzhet egy előjel.
P1. 5 -13 299651 0 +89

A számok tizedestört formája
annyiban tér el a matematikában
megszokottól, hogy a tizedes-
vessző helyett tizedespontot
kell írni.

P1: 6.43 -7.25 -9.00 +0.04

A számok exponens formája egy
tizedestörtből és egy ún.
exponens részből áll, és a
matematikában a számok normál
alakjaként megismert formához
hasonló.

A tizedestört részben itt is
tizedespontot kell alkalmazni.
Az exponens rész egy E betűből
és egy egész számból (karakte-
risztika) áll. A karakteriszi-
ka -38 és 38 közötti érték le-
het.

BAS04.13

P1: 1.96E-3 2.4E9 136.8E-18
helyes valós számok;
1.84E4 2.93E41 66.6E-92
hibásak.

A Programokban az EGÉSZ számokat
(egész típusú változók értékeit)
számjegyek sorozataként adjuk
meg, amelyet megelőzhet egy elő-
jel (+, -).

P1: 546 +789 -9870 -88888888

A Programokban a jelsorozat
típusú változók értékeit idéző-
jelbe zárt jelsorozattal lehet
megadni.

P1: "6.9" "2.4" "ABRAKADABRA"

Jelsorozat elemeként idézőjel
nem szerepelhet!

5. LECKE

ARITMETIKAI ÉRTÉKADÁS, ARITMETIKAI KIFEJEZÉSEK KÉPZÉSE

Az értékadó utasítástípus alkalmazása az egyik lehetőség arra, hogy egyszerű változókhoz vagy tömbelemekhez aktuális értéket rendeljünk.

Az értékadó utasítástípus kulcsszava a LET, az utasítás-típus írási formája az alábbiak egyike lehet.

LET val.egysz.vált=arit.kif.

LETjels.tip.egysz.vált=jels.kif.

LET val.tömbelem=arit.kif.

LETjels.tömbelem=jels.kif.

(Rövidítések: val.=valós
egysz.=egyszerű
arit.=aritmetikai
vált.=változó
kif.=kifejezés
jels.=jelsorozat)

A valós egyszerű változók nevére és lehetséges értékeire (a valós számokra) vonatkozó szabályokat a 4. leckében megismertük.

BAS05.3

Az aritmetikai kifejezés legegyszerűbb esetben valós szám vagy valós egyszerű változó lehet.

```
P1:      LET QTYA=HUND
          LET BASIC=1E16
```

Altalános esetben az
ARITMETIKAI KIFEJEZÉS

- valós egyszerű változókból
- valós számokból
- valós tömbelemekből
- aritmetikai műveleti jelekből
- valós értékű aritmetikai függvényekből

felepített, ZARÓJELEZHETŐ
kifejezés.

A LET kulcsszó és az = jel között álló részt az értékadás BALOLDALANAK; az = jeltől jobbra álló részt az értékadás JOBBOLDALANAK nevezzük. Bár több számítógépen elhagyható az értékadó utasítás kulcsszava (LET), a bal- és a jobboldal megkülönböztetése lényeges, hiszen a baloldali változó kapja meg a jobboldali kifejezés értékét, az aktuális értékre tanult szabályok szerint.

Ebben a leckeében csak az első altípussal, a valós egyszerű változókra vonatkozó értékadással foglalkozunk.

BAS05.5

A valós egyszerű változók és a valós számok fogalmát a 4. leckében megismertük.

Az aritmetikai műveleti jelek az alábbiak:

+	összeadás
-	kivonás
*	szorzás
/	osztás
↑	hatványozás

A fenti műveletek eredménye a számítógép SZÁMABRÁZOLÁSI és PONTOSSÁGI határain belül azonosnak tekinthető a matematikai eredménnyel.

P1: $2E33+3E34$ értéke $3.2E34$
 $3*8.1$ értéke 24.3 lesz.

A számítógép NEM TUDJA KISZAMITANI a művelet eredményét, ha az a szám-ábrázolási tartományon kívülre esne.

P1: $2E33*3E34$ értéke nem kiszámítható.

Ugyanakkor az ábrázolási Pontosság miatt az igen kis Pozitív számok értéke nulla lesz.

P1: $0.001/1E36$ értékeként nullát kapunk.

BAS05.7

Ha egy aritmetikai kifejezésben több művelet szerepel, akkor - zárójeljezés híján - a matematikában megszokott MŰVELETI SORREND szerint történik meg a kifejezés értékének kiszámítása.

P1: $6+4/2$ értéke 8,

$7/4*2$ értéke 3.5 lesz.

$3E20*3E20/1E19$ értéke
NEM KISZAMITHATÓ

$3E20/1E19*3E20$ értéke
9E21 lesz

A hatványkitevő értéke lehet tört illetve negatív érték is.

P1: $4↑0.5$ értéke 2,
 $4↑1$ értéke 0.25

A matematikában megszokottól eltérően a BASIC-ben a szorzás jele nem hagyható el!

P1: 2A helyett 2*A
3AB helyett 3*A*B vagy 3*AB
a BASIC-ben írható forma.

BAS05.9

A zárójellezéssel – a matematikában megszokott módon – megváltoztathatjuk a műveletek elvégzési sorrendjét.

P1: 3E20*3E20/1E19 értéke
NEM KISZAMITHATÓ

de 3E20*(3E20/1E19) értéke
9E21 lesz.

Az aritmetikai kifejezésekben minden olyan helyen, ahol valós típusú szám vagy változó szerepelhet, állhat ARITMETIKAI FÜGGVÉNYHIVATKOZÁS is. Egy függvényhivatkozás a függvénynévből és az értéklistából (un. ARGUMENTUMOKBÓL) áll.

P1: LET X4=COS(0)

A fenti értékadás jobboldala egyetlen függvényhivatkozás. A függvény neve COS, amelynek egyetlen független változója (argumentuma) van, amelyhez ebben a hívásban a 0 értéket rendeltük.

BAS05.11

A függvényeket két csoportra osztjuk.

- Programban definiálható;
- a BASIC nyelvbe "beépített" függvényekre.

Az előbbi függvénycsoport megértése további információkat igényel, ezért ezekkel csak később foglalkozunk.

A beépített függvényekre vonatkozó legfontosabb ismereteket a 6. leckeiben találjuk.

6. LECKE

ARITMETIKAI BEÉPÍTETT FÜGGVÉNYEK

Ebben a leckében azokkal a beépített függvényekkel foglalkozunk, amelyeknek mind az
-értelmezési tartománya,
mind az
-értékkészlete

ARITMETIKAI (valós vagy egész) érték.

E függvényeknek közös tulajdonsága, hogy egy független változójuk (argumentumuk) van.

Ebbe a csoportba tartoznak a matematikából ismert

SIN	szinusz
COS	koszinusz
TAN	tangens
ATN	árkusz tangens
trigonometrikus függvények,	
LOG	logaritmus
EXP	exponenciális
ABS	abszolút érték
INT	egészrész
SGN	előjel-függvény
SQR	a nemnegatív számok négyzetgyökét adó
függvények.	

BAS06.3

Részletesebben csak az INT függvénnyel foglalkozunk.

Ha a függvény argumentuma egész szám, akkor értéke megegyezik az argumentum értékével.

Ha az argumentum nem egész szám, akkor csonkítás történik, azaz értéke az a LEGNAGYOBB EGÉSZ szám lesz, amely nem nagyobb az argumentum értékénél.

```
P1:    INT(2.6) értéke: 2
        INT(0.8) értéke: 0
        INT(-3.3) értéke: -4
```

A végeves (jelsorozat és aritmetikai) függvényekkel a 7. lecke-ben ismerkedünk meg.

Más számítógéptípusokon a BASIC további beépített függvényeket is tartalmazhat.

7. LECKE.

RZ ERTEKADÓ UTASÍTÁSTIPUS II.

RZ értékkadó utasítástípus 5.
leckeiben megismert nagy formájú
közöl itt a másodikkal foglalko-
zunk.

LET JELSOROZAT TÍPUSÚ EGYSZERŰ
VÁLTOZÓ=SZTRINNYKIFEJESZÉS

R LET KULCSSZÓ – AZ ARITMETIKAI
ESZETHEZ HASONLÓAN TÁBB SZÁMITÓ-
GÉPTÍPUSON ELHÁGYHATÓ – 199
COMMODORE 64 GÉPEN IS.

R JELSOROZAT TÍPUSÚ EGYSZERŰ
VÁLTOZÓK NEVE – MINT A 4. LECKÉ-
BEN MEGTANULTUK – EGY EGYSZERŰ
NEVBŐL ÉS A DOLLAJ JELBŐL ÁLL.

R SZTRINNYKIFEJESZÉS LEVEGYZE-
RŐBB ESTBEN
– JELSOROZAT TÍPUSÚ EGYSZERŰ
VÁLTOZÓ, VAGY
– JELSOROZAT ÉRTÉK LEHET.

P1: LET A\$=B\$
LET S\$="ABRÁKADABRA"

BAS07.3

Általános esetben a sztringki-
fejezés:

- Jelsorozat típusú egyszerű
változókából
- Jelsorozat értékekből
- Jelsorozat típusú tömbelemekből
- sztring műveletekkel
- sztring értékű függvényekkel
felépített, zárójellelezhető kife-
jezés.

A jelsorozat típusú egyszerű
változókkal és a jelsorozat ér-
tékekkel a 4. leckeiben megismer-
kedtünk.

Az alábbi sztringműveletek al-
kalmazhatók.

Jelsorozatok ÖSSZEFÜZÉSE (kon-
katenáció), amelyet az operan-
duszok (jelsorozat értékek
vagy jelsorozat értékű kifeje-
zések) közé írt + jellel jelöl-
ünk.

P1.: LET A\$="EGER"+"SZEG" ha-
tására az A\$ értéke "EGERSZEG"
lesz.

Két beépített fogóványt emlí-
tunk meg, amelyek igen haszno-
sok a jelsozokat értékek keze-
lésénél. Mindkettőnek sztring-
kifejezés az argumentuma, a
fogóványértéke viszont aritme-
tikus (azaz VEGYES típusúak).

A LEN fogóvány megadja az ar-
gumentumként szereplő sztring-
kifejezés hosszát (azaz, hogy
hány jelből áll).
Értéke nemnegatív egész szám
lehet és csak az 0-es jelsozo-
zatra ("") ad nulla értéket.

P1.:

LEN("CSE") értéke: 3

LEN("KISKAPU") értéke: 7

BAS07.7

A VAL függvény az argumentum jelsorozat számértékét adja, feltéve, hogy az számként értelmezhető (az idézőjelektől eltekintve).

P1.:

```
VAL "5"           értéke: 5
VAL "6.193"       értéke: 6.193
VAL "2E-2"        értéke: 0.02
```

Nem kiértékelhető (hibás) P1.:

```
VAL "2,9"
VAL "4†2"
```

Az STR\$ függvény a VAL fordítottja; a számérték argumentumot alakítja át jelsorozattá.

```
P1.    STR$(5.93)      " 5.93"-at;
        STR$(-4.22)    "-4.22"-t
ad értéket.
```

Nem kiértékelhető (hibás) eset az, ha a STR\$ argumentuma nem numerikus érték vagy numerikus értékű változó.

```
P1.    STR$("ABC")    vagy
        STR$("5.95")
hibás esetek.
```

8. LECKE

A PRINT utasítástípus

A PRINT utasítástípus rendeltetés az, hogy az utasítások végrehajtása során a számitógépben keletkező eredményeket kitírássuk a képernyőre. Ha a program végrehajtása során PRINT utasítás következik, akkor az utasításban megadott kifejezések értékeit a gép kiszámítja, majd kitírja.

Egy PRINT utasításban a kitírandó (aritmetikai vagy jelsozogat ti-
pusú) kifejezésekben kívül

FORMATUMVEZÉRLŐ TÍTLEK

is szerepelnek. Ezek rendeltetés az, hogy a képernyőre átte-
kinthető, esztétikus elrendezés-
ben írassuk ki az adatokat. A formátumvezérlő títteleket két csoportra osztjuk:

-elválasztó jelek(, ;) ;

-csak a PRINT utasításban al-
kalmazható beépített függvé-
nyek (TAB, SPC).

BAS06.3

P1: az alábbi

```
PRINT 6+49*5.613,INT(COS(630));  
"ABLAK"
```

utasításban 3 kifejezés szerepel; az első kettő aritmetikai, a harmadik jelsorozat típusú. Az első két kifejezés között vessző (,) a hátsó kettő között pontosvessző (;) az elválasztó jel.

A formátumvezérlő tételek hatása a következő:

Pontosvessző esetén a kiírás folyamatos. (A kiírás az "AKTUALIS POZÍCIÓ"-ra történik.)

Vessző esetén az aktuális pozíció arra a következő képernyőoszlopra kerül, amelynek a sorszáma osztható 10-zel. (A képernyőoszlopok számozása 0-tól 39-ig történik.)

BAS08.5

P1.: PRINT "AB", "LAK" eredménye:
AB LAK

PRINT "AB"; "LAK" eredménye:
ABLAK lesz, de

PRINT "PONTTIZJEL", "VALAMI"
eredménye:
PONTTIZJELVALAMI

Az "aktuális" pozícióra hatással lehet a korábbi PRINT utasítás.

P1.: PRINT,,:PRINT "MAXI"

esetén a MAXI kiírás a 20. pozíciótól kezdődik.

Ha egy PRINT utasítás kifejezéssel zárul, vagy nincs egyetlen tétel sem az utasításban (un. üres PRINT), akkor a következő sor elejét tekinti "aktuális" pozíciónak.

P1.:PRINT A,B:PRINT "AKARMI"
vagy PRINT:PRINT "AKARMI"

esetén AKARMI a sor elején kezdődik.

De:
PRINT "HAROM",:PRINT "AKARMI"
vagy:
PRINT "VALAMI";:PRINT "AKARMI"

esetén nem (hanem a 10. illetve a 6. pozíción).

A számértékek és a jelsozozatok
Pontosvessző(,) elhatárolással
történő kírásá nemleges eltérő.
A jelsozozatok ugyanis közvetlen-
en a számok mellé (egybeol-
vasztva) jelennek meg, a számok
között viszont (az elöl helyén
kívül) egy szóköz is lesz.

Pl.:
PRINT "3"; "5" esetén: 35
PRINT 3/5 esetén: 3
PRINT -3/-5 esetén: -3 -5

lesz a képernyőn.

A TAB(X) függvényel az argumen-
tumaként megadott pozícióra való
állítás (tabulálást) írhatunk
elő.

Pl.: PRINT TAB(3) 4 hatá-
sára a kírás a 3. pozíciótól
kezdődik:
4

Ha az adott sorban túlléptek a
TAB argumentumával előirt érte-
ket, akkor a tabulálás hatásta-
lan (nem változik az "aktuális"
pozíció).
Pl.: PRINT TAB(3)-4
esetén a -4-es a 10. pozíció-
tól jelenik meg.
A TAB argumentumaként szereplő
X értéke 0 és 255 között lehet.

```

RZ SPC(X) függvényrel megvaltoztatjuk a kintars "aktuális" pozícióját, meghatározott számú címszót, meghatartunk ki: amennyit, amennyit az argumentumként szent-
repió szám (kifejezés) értéke.
P1.:PRINT:PRINT -3;SPC(1);-9
Hataáára -3 -9 kintars,
PRINT:PRINT"LODA";SPC(1)"RZS"
Hataáára LODR RZS kintars
kapunk.
RZ X értéke 0 és 255 között lehet.

```

```

R: elhatárolo jel a C-64 BASIC-
ben bizonyos esetekben elhaa-
ható, ezzel a lehetőséggel azon-
ban ne éljünk addig, amíg ma-
biztonsággal és gyakorlati nem
vendelkezőnk a BASIC programo-
zásban.

```


9. LECKE

TÖMBÖK, TÖMBELEMÉK

A tömb a BASIC-ben azonos típusú értékek egy csoportját jelenti, amelyekre közös néven hivatkozhatunk.

Mindhárom alaptípus (valós, egész, jelsorozat) esetén alkalmazhatunk tömbösítést, így lehet valós, egész és jelsorozat típusú tömb a BASIC-ben.

A tömbök kezelése szempontjából megkülönböztetjük a tömb DEKLARÁCIÓJÁT és a tömbelemek-re való HIVATKOZÁST.

Azokat a neveket, amelyeket tömbnévként akarunk használni a Programban, DIM utasításban kell deklarálni.

Pl.: DIM A(33), B\$(2,5)

Itt A egy (un. egyszimenziós) valós tömb (vektor) lesz, amely egyidejűleg 34 különböző valós számérték tárolására alkalmas.

A B\$ egy (un. kétszimenziós) jelsorozat típusú tömb (mátrix) lesz, amelynek 3 "sora" és 6 "oszlopa" lesz, s így 18 különböző jelsorozat érték egyidejű tárolására alkalmas.

BAS09.3

A Programban mindazon helyeken, ahol egyszerű változó szerepelhet (Pl. értékadó utasítás bal oldala, kifejezés összetevője, függvény argumentuma) szerepelhet tömbelem (un. tömbelem hivatkozás is).

A hivatkozás helyén a tömbelemet a tömbnévvel és az indexekkel jelezzük.

```
Pl: A(4)=6: A(0)=19+A(4)/SGN(A(4))  
B$(1,2)="NAPOS"  
B$(0,0)=LEFT$(B$(1,2),3)+"LOPO"
```

Lényeges tulajdonsága a C-64 BASIC-nek, hogy műveleteket csak tömbelemekkel végezhetünk, tömbökkel nem.

```
Pl. : 10 DIM A(3)  
      20 A=6  
      30 A(0)=10: A(1)=11: A(2)=12:  
      35 A(3)=13  
      40 PRINT A; A(1); A(2); A(3);
```

A fenti Program végrehajtásakor a 6, 11, 12, 13 értékek kerülnek kiírásra.
A 20-as sorban lévő A=6 utasítás tehát nem az A tömbre vonatkozik, hanem egy A nevű, egyszerű valós változóra.

BAS09.5

A hivatkozás helyén a tömbelemeket mindig ugyanannyi indexszel kell szerepeltetni, mint amennyivel a deklarációban (DIM utasításban) megadtuk.

```
P1.:DIM B$(2,5) esetén
      B$(1)="QQRIQ"
      hibás utasítás.
```

A hivatkozásban szereplő indexek értékeinek is összhangban kell lenni a deklarációban megadott határokkal.

```
P1.:DIM B$(2,5) esetén
      a B$(3,1) vagy
      a B$(0,6)
      hibás hivatkozás.
```

Hibás hivatkozás az is, ha egy index negatív.
Az indexek aritmetikai kifejezések is lehetnek.
A DIM A(33),B\$(2,5) deklaráció esetén szabályos hivatkozások
P1.: A(2*C%),B\$(A(1),2+1)
feltéve, hogy a kifejezések értéke az indexhatárokon (0,33), illetve (0,2) és (0,5) belül vannak.

A dimenziók számára a C64 BASIC csak azt a megszorítást teszi, hogy a programban szereplő tömböknek el kell férnie a memóriában.

```
P1.:DIM Q(2,2,3,2,2,3) megengedett.
```

Egy névnek többnévűvé való mi-
 nősítésére akkor kerül sor,
 amikor a megfélelő DIM utasít-
 ás végrehajtására kerül. Ennek
 megfélelőben hibás az alábbi
 programkészlet:
 $10 \text{ DIM } A(3,4)=10$
 $20 \text{ DIM } A(5,5)$
 Egy nevet csak egyszer szabad
 dimenzionálni.
 Hibás az
 alábbi programkészlet is:
 $10 \text{ DIM } A(5,5)$
 $20 \text{ DIM } B(10,2), A(5,5)$
 Foggetlenül attól, hogy R-t
 egyformán dimenzionálna 10 és
 20 utasításokban.

A dimenzionálás végrehajtása-
 kor minden többnévű megkapja
 az alapértelmezés szerinti ér-
 tékét; aritmetikai típusnál 0-
 t, jelsozogat típusnál ones
 jelsozogatot ("").
 Az egydimenziós maximum 10
 elemű tömbökre ún. automatikus
 dimenzionálást végez a C-64
 BASIC. Ezzel a lehetőséggel
 azonban ne éljünk addig, amíg
 nem rendelkezünk megfelelő
 gyakorlati és BASIC programo-
 zásban.

10. lecke

A feltételes utasítás-
típus

A feltételes utasítás típus
rendeltetés az, hogy a prog-
ramban az utasítások ne csak
egyszerű hajtódjárnak végre-
essenek lehetőség nyílik arra
is, hogy az utasítások végre-
hajtást sorrendjét befolyásol-
juk, így utasításokat, prog-
ramrészeket többször is végre-
hajtassunk a számítógéppel.

A feltételes utasítás alakja:

IF log.kif. THEN feltetlen
utasítások

IF: az utasítás típus kulcsszáva
log.kif.: logikai kifejezés

THEN: elválasztó kulcsszó

MS BASIC 8000000000000000:09 Vagy
több, kétszemponttal elválasztó
tot, nem feltételes utasít-
tás.

Feltetlen utasítás az addig
megismert LET, PRINT, DIM.

P1. :
IF COS(3)>0.5 THEN Z=4:S=9

RZ IF UTASITÁS ÉRTÉLMENŰSÉ:
HA A LOGIKAI KIFEJEZÉS IGAZ,
AKKOR VÉGREHAJUTÁSNA KERÜLT
A THEN UTÁNI UTASÍTÁSOK,
EGYÉBKÉNT (HA A KIFEJEZÉS HÁ-
MIS) PEDIG NEM.
A FENTI PÉLDÁBAN A LOGIKAI
KIFEJEZÉS ÉRTÉKE HAMIS, IGY A
Z ÉS S MEGÜRZT KÖRÁBRÓI ÉRTÉ-
KÉNT, NEM KAPJÁK MEG A 4 ÉS 9
ÉRTÉKEK.

LOGIKAI KIFEJEZÉSEKEN ÁLTALÁ-
NOS ESETBEN
-RELÁCIÓKAL,
-LOGIKAI MŰVELETTI JELEKKEL
KÉPZETT,
-ZÁRÓJELEZHETŐ
KIFEJEZÉSEK ÉRTÜNK.
Egy reláció relációjelből és
két összehasonlító kifeje-
zésből áll.

Relációjelek:
<= Kiseb
>= Kiseb
< Kiseb
> Kiseb
= Kiseb
< Kiseb
> Kiseb
= Kiseb

BAS10.5

Összehasonlítható két kifejezés, ha

- mindkettő kiszámítható aritmetikai kifejezés; vagy
- mindkettő jelsorozatos kifejezés.

```
P1.: 3=3
      4>0%
      5<>INT(A<1>/L%<6>)
      COS(Z)*BIGA<=VAL(D$(9))
      B$(2,9)="BATKA"
```

Jelsorozat értékeknél a < és > összehasonlítás a jelsorozatok ábrázolási kódja szerint történik. Erről elegendő annyit tudni, hogy betűknél megfelel az ABC-beli sorrendnek; a számjegyek (0-9) is sorban vannak; továbbá minden számjegy "kisebb" bármelyik betűnél.

Egy reláció értéke IGAZ vagy HAMIS lehet.

A fenti példákból a $3=3$ mindig igaz, a többi reláció a benne szereplő változók aktuális értékétől függően igaz vagy hamis értékű, esetleg kiszámíthatatlan lehet.

Ha egy reláció formailag hibás vagy kiszámíthatatlan, akkor a Program végrehajtása megszakad.



11. LECKE

VEZÉRLÉSÁRADAS,
CIKLUSSZERVEZÉS

Vezérlésáradás több utasítás-
típussal érhető el BASIC-ben.

Itt a legegyszerűbb esetékeket,
a ~~VEZÉRLÉSÁRADÁS~~ ~~VEZÉRLÉSÁRADÁS~~ ~~VEZÉRLÉSÁRADÁS~~
(GOTO) utasításátípussal és a
~~VEZÉRLÉSÁRADÁS~~ ~~VEZÉRLÉSÁRADÁS~~ ~~VEZÉRLÉSÁRADÁS~~
típussal foglalkozunk.

A további vezérlésáradó utasít-
tásokkal (GOSUB, RETURN, ON)
később találkozzunk.

A GOTO utasításnál megváltoz-
tatjuk a BASIC utasítások
automatikus (utasításorszám-
táblázat) sorrendben történő
végrehajtását.

Pl.: 60 GOTO 500

hatására a következő végrehaj-
tando utasítás az 500-as sor-
számot viseli lez, a 60 és
500 közötti utasítások ki-
márak a végrehajtást sorrend-
ből. Ezért a GOTO-t "UGRO"
utasításnak is nevezzük.

BAS11.3

Megszakad a Program futása (Undefined Statement Error hibaüzenettel) abban az esetben, ha nincs olyan utasítás-sorszám a Programban, amire ugrani akarunk.

Az ugrás történhet "előre" (magasabb); vagy "hátra" (alacsonyabb utasítás-sorszámra).

Az ugrás igazi értelmét a felteteles utasítással együtt láthatjuk.

```
10 X=0.4
20 PRINT 3*X*X+4*X-8
30 IF X<1.8 THEN X=X+0.1 : GOTO 10
40 REM
```

A fenti Program elvégzi a $3X^2+4X-8$ függvény értékeinek kiszámítását a $[0.4-1.8]$ zárt intervallumon, 0.1 -es lépésközzel.

Ugyanezt a feladatot fogalmazza meg a másik Programrészlet is:

```
10 X=0.4
20 PRINT 3*X*X+4*X-8
30 IF X>1.8 THEN GOTO 60
40 X=X+0.1
50 GOTO 10
60 REM ...
```

BAS11.5

A programkészítéskor egyszer leírt utasítások sokszori végrehajtását hátraugrásokkal, (un. hurkos programozással) érhetjük el.

Vigyázni kell arra, hogy a hurkból mindig legyen "kiút", mert egyébként a programunk VÉGTELEN CIKLUSBAN fog kerülni valamelyik sorszám-tartományban.

A "kiút" feltételes, amit úgy érünk el, hogy vagy a hátra, vagy az előreugrást IF utasításban írjuk elő, amelynek a logikai kifejezése a kívánt ponton értéket változtat.

Az IF és GOTO utasításoknál biztonságosabb, egyszerűbb eszköz ciklusszervezésre a FOR, NEXT utasításpár.

A FOR utasítás alakja a következő:

```
FOR cv=ké TO vé STEP lk
```

(cv=ciklusváltozó
ké=kezdőérték
vé=végérték
lk=lépésköz)

FOR: utasítástípus kulcsszó
TO, STEP: határoló kulcsszavak

valós egyszerű változó

aritmetikai kifejezés;
ennek értéket kapja kezdett-
ben a ciklusváltozó

aritmetikai kifejezés;
annál is a ciklusváltozó
értéke a hurokban lévő utasításonként két végrehajtás között

aritmetikai kifejezés;
ha a ciklusváltozó értéke
átlépi ezt az értéket, befejeződik ("lejár") a ciklus

A NEXT utasítás alakja:

NEXT ciklusváltozó

Ez az utasítás határoli a hurokban lévő utasításokat (um. "ciklusmagot"), azaz a NEXT elérésekor (végeértéskor) törli a ciklusváltozó értékeit módosítás, és ezzel fogórn vagy lejár a ciklus, vagy újra kezdődik a magban lévő utasítások végeértéke.

BAS11.9

A korábbi Példa FOR, NEXT utasításpárral megfogalmazva:

```
10 FOR X=0.4 TO 1.8 STEP 0.1
20 PRINT 3*X*X+4*X-8
30 NEXT X
```


12. LECKE

Az INPUT utasítástípus

Az INPUT utasítástípus rendeltetése az, hogy a Program futása közben a változókhoz "kívülről" (a klaviatúráról) megadható értékeket rendeljünk aktuális értékként.

Az INPUT utasítástípus alakjai

(1):

INPUT változólista

vagy

(2):

INPUT szöveg;változólista

Az utasítás végrehajtásakor

- (2) esetén a képernyőn megjelenik a "szöveg"-ként megadott jelsorozat,

- mindkét forma esetén egy kérdőjel(?)

- a Program várakozik, hogy megadjuk a billentyűzetről a változólistán szereplő változók értékeit.

P1:

```
90 INPUT "KEREM AZ UT HOSSZAT";S
    esetén a 90-es utasítás végrehajtásakor
```

```
KEREM AZ UT HOSSZAT?
```

Jelenik meg a képernyőn.

A válasz megadása a megfelelő érték bebillentyűzésével és a RETURN megnyomásával történik.

```
KEREM AZ UT HOSSZAT?
```

```
620 /RETURN/
```

```

EKKOR AZ S VÁLTÓZÓ AKTUALIS
ÉRTÉKE 620 LEZ, EGYEZ ADDIG,
AMIG UJABB ÉRTÉKET NEM RENDE-
LŐNK AZ S-HÉZ (ÉRTÉKADÓ VÁGY
INPUT UTASÍTÁSTIPUSSAL).
HAZ INPUT UTASÍTÁSSAL EGYEZ,
VALÓS ÉS JELSOROZATOS EGYEZŐ
VÁLTÓZÓKHOZ VÁGY TÖMBELEMKEHEZ
RENDELHETŐNK ÉRTÉKET
(AKÁR EGY UTASÍTÁSON BELÜL IS)
P1.:
INPUT A,B%,C%(5),D%

```

```

HA AZ INPUT UTASÍTÁSBAN TÖBB
VÁLTÓZÓ (VÁLTÓZÓK VESSZŐVEL EL-
VÁLASZTOTT SOROZATA) SZEREPEL,
AKKOR A FUTÁSKORI VÁLÁSBAN
TÖBB ÉRTÉKET KELL MEGADNI,
VESSZŐVEL ELVÁLÁSZTVA.

```

```

P1.:AZ
INPUT A,B%,C%(5),D(3)
VEGREHATÁSOKON A KÉPERNYÓN
MEGJELENŐ KÉRDŐJEL UTÁN
3,6,4,2,2,8 VÁLÁSZT ADUNK;
EKKOR:
A      3-6-05;
B%     4      -05;
C%(8)  2      -05;
D(3)   2-8-05;
ÉRTÉKET KAPJA.

```

Ha kevesebb értéket adunk, mint ahány az INPUT változó-
listáján szerepel, akkor a
hiányzó értékeket két kérdőjel
(??) kíséreléssel kéri a prog-
ram.

Ha több értéket adunk, mint
ahány az INPUT változójában
szerepel, akkor "EXTRA INFO-
RED" üzeneteket figyelmeztet
a program arra, hogy a több-
letet figyelmen kívül hagyta.

Az egész és a valós értékeket
ugyanabban a formában adjuk
meg, mint amit a programba
írható konstansokkal megis-
mértünk.

Az INPUT utasítás végrehajtá-
sakor megadott jeliszorozat ér-
teket
-vagy a programban használt-
ható formában (idézőjelek
között) adjátjuk meg,
-vagy idézőjelek nélkül,
feltéve, hogy a jeliszorozat
nem tartalmaz INPUT vezérlő-
jelet (,).
Ha a jeliszorozat értéket idéző-
jelek nélkül adjuk meg, akkor
a konstans elején lévő, "beve-
zető szövegek"-et a számítottép
figyelmen kívül hagyja.

13. LECKE

ALPROGRAMOK KEZELÉSE GOSUB és RETURN utasítástípusok

A GOSUB és RETURN utasításpár egy viszonylag egyszerű programozástechnikai segédeszközt nyújt a program különböző pontjain, többször végrehajtandó utasítás-sorozatok kezeléséhez.

A GOSUB utasítástípus hasonló a GOTO-hoz, abban az értelemben, hogy hatására vezérlésátadás történik az operandusként megadott utasítássorszámra.

Eltérések abban áll, hogy míg a GOTO-nál a vezérlésátadás "végsőleges", addig a GOSUB esetén az első végrehajtott RETURN utasítás hatására a program a megfelelő GOSUB utasítást követő utasítástól folytatódik. A GOSUB utasítástípust ennek megfelelően "alprogram-hívó" utasítástípusnak is nevezik.

BAS13.3

```
P1.: 10    A=3
      20    GOSUB 200
      30    END
      200   A=5
      210   B=9
      220   RETURN
```

A fenti Program esetén az utasítások végrehajtása a

10,20,200,210,220,30

sorszám-sorrendben történik.

Az alprogram-hívás többszörös mélységben is végrehajtható; azaz a GOSUB-bal meghívott alprogramban újabb GOSUB utasítások lehetnek.


```

P1. : 10
      I=3
      GOSUB 500
      GOSUB 500
      GOSUB 500:GOSUB 500
      PRINT I
      END
      I=I+1:GOSUB 500
      J=I*2
      RETURN
      I=I+2
      RETURN
      610 RETURN

```

A fenti program esetén a vég-
rehajtás melyre az alábbi lesz:

10, 20, 500, 600, 610, 510, 520,
30, 600, 610, 30(2. utasítás),
500, 600, 610, 510, 520, 40, 50.
AZ I értéke kimutatáskor 11
lesz.

AZ alprogramok a bontással
(a GOSUB és RETURN utasítások
megfelelő használatával) a
nagyobb méretű programokat ki-
sebb, viszonylag önálló 1091-
es 610-es számú bontásokkal
amelyek egymást elférnek a
képernyőn, így könnyen átte-
kinthetővé válik a programunk.

3. SZ. MELLÉKLET

K É R D É S E K

BAS04K/8

Mikor tekint a COMMODORE 64 személyi számítógép azonos értékűnek két valós számot?

- A. Mindig.
- B. Ha a két változó neve azonos.
- C. ha eltérésük kisebb, mint 4×10^{-39} .
- D. Sohasem.
- E. Nem tudom.

BAS04K/9

Melyik állítás igaz az alábbiak közül?

- A. Egy valós tömböt nevezhetünk ALFAZ-nak.
- B. Egy valós egyszerű változót nevezhetünk QQCS-nak.
- C. Az EREGY és az R1 ugyanazt a változót jelölik.
- D. A NYOLC nevű változó értéke csak 8 lehet.
- E. Nem tudom.

BR504K/10

Milyen típusú lehet egy L\$
nevű változó?

A. Valós tömb.

B. Jelsorozatot típusú egyszer-
re változó vagy tömb.

C. Valós típusú egyszerre
változó vagy tömb.

D. Semmilyen, a változónev
hibás.

E. Nem tudom.

BR504K/11

Milyen típusú lehet egy LX
nevű változó?

A. Egész értékeket tartal-
mazó tömb.

B. Jelsorozatot típusú egy-
szerre változó vagy tömb.

C. Valós típusú egyszerre
változó vagy tömb.

D. Semmilyen, a változónev
hibás.

E. Nem tudom.

Meltek állítás 1982 az aláb-
biak közül?

A. A jelzőkódok típusú változók
értékei csak betűkből és
számokból álló jelzőkódok
lehetnek.

B. A jelzőkódok típusú változók
38 38
értéktartományai -10 és 10
között vannak.

C. Az A2 nem lehet jelzőkódok
típusú változó neve.

D. A jelzőkódok típusú változók
értéke legfeljebb 32 jel
hosszúságú sorozat lehet.

E. Nem tudom.

Meltek állítás 1982 az aláb-
biak közül?

A. Egy programban legfeljebb 26
jelzőkódok típusú sorozat
változó vagy tömb lehet.

B. Egy programban nem lehet
tömb tömb, mint ahogy
sorozat változó van.

C. Minden % jelre végezhető jel-
zőkódok típusú
változó neve.

D. Az A11 és A12 nevek ugyan-
azt a változót jelölik.

E. Nem tudom.

BAS05K/3

Melyik aritmetikai kifejezés helytelen az alábbiak közül?

- A. $A * 23E14 - 2 * -0.5$
- B. $4 - AB * 63 + 19$
- C. $4 - A * B * 63 + 19$
- D. $1/2A - 4$
- E. Nem tudom.

BAS05K/4

Melyik értékadás helytelen az alábbiak közül?

- A. LET B=3*9E16/4
- B. LET BAL=1/2+E16
- C. LET K="6"
- D. LET BUXA=TELI*2E1
- E. Nem tudom.

BAS05/5

Melyik értékadás helytelen az alábbiak közül?

- A. LET XY=93
- B. LET X*Y=93
- C. LET K3L5=FEL+Y
- D. LET KIGYO=VI/A
- E. Nem tudom.

BAS05/6

Melyik értékadás helytelen az alábbiak közül?

- A. LET VACSI=KEX/EC
- B. LET IDEZET="AKARMI"
- C. LET LO=NYI-HA-HA
- D. LET Z=Z+1
- E. Nem tudom.

BAS05K/7

Az alábbi értékadó utasítások közül melyik helytelen?

- A. LET B=B
- B. LET B=C
- C. LET B=B+1
- D. LET B=B\$
- E. Nem tudom.

BAS05K/8

Melyik kifejezés nem számítható ki tetszőleges X érték esetén?

- A. $\text{SQR}(\text{SGN}(X)*X)$
- B. $\text{SQR}(\text{ABS}(X))$
- C. $\text{SQR}(\text{COS}(X))$
- D. $\text{SIN}(\text{COS}(X))$
- E. Nem tudom.

BAS05K/9

Az alábbi kifejezések közül melyiknek az értéke nem független X -től?

- A. $\text{SGN}(X) * X - \text{ABS}(X)$
- B. $X - \text{SGN}(X) * \text{ABS}(\text{INT}(X))$
- C. $\text{SGN}(X) - \text{INT}(\text{SGN}(X))$
- D. $\text{SGN}(X) - \text{SGN}(\text{SGN}(X))$
- E. Nem tudom.

BAS05K/10

Az alábbi kifejezések közül melyiknek az értéke független X -től?

- A. $\text{SGN}(\text{ABS}(X) + 1)$
- B. $\text{SGN}(\cos(X))$
- C. $\text{ABS}(\sin(X))$
- D. $\text{INT}(\sin(X))$
- E. Nem tudom.

BAS07K/1

Az alábbi értékadó utasítások közül melyik helyes?

- A. LET C%="QQCS"
- B. LET C\$=QQCS
- C. LET C="QQCS"
- D. LET C\$=""
- E. Nem tudom.

BAS07K/2

Melyik hamis az alábbiak közül?

- A. LEN(MID\$("REBARBARA",3,5)) értéke: 3.
- B. VAL(MID\$("A23",2,3)+".1E2") értéke: 2310.
- C. SGN(LEN("TYUK")) értéke 1.
- D. VAL(MID\$("23X",1,2)+".3E2") értéke: 323.
- E. Nem tudom.

BAS07K/3

Melyik igaz az alábbi állítások közül?

- A. A VAL függvény argumentuma valós szám.
- B. A LEN függvény értéke egész szám.
- C. A VAL függvényhívás az értékadó utasítás mindkét (bal és jobb) oldalán szerepelhet.
- D. Az üres jelsorozatok hossza nem meghatározható.
- E. Nem tudom.

BAS07K/4

Melyik hibás az alábbi kifejezések közül?

- A. VAL(MID\$("23E16",2,4))
- B. VAL(MID\$("23A16",2,4))
- C. "23A4"+"13B6"
- D. "23A4+13B6"
- E. Nem tudom.

BAS07K/5

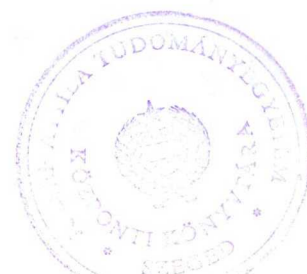
Melyik helyes az alábbi kifejezések közül?

- A. `LEN(VAL("33"))`
- B. `"1463"+"16K4"`
- C. `"5311"-"4423"`
- D. Nem tudom.

BAS07K/6

Melyik hibás az alábbi kifejezések közül?

- A. `LEN(MID$("ABCD",2,4))+VAL("2")`
- B. `VAL("33")+("44")`
- C. `VAL("33")+VAL("44")`
- D. `VAL("33"+"44")`
- E. Nem tudom.



BAS07K/7

Mennyi lesz a következő
kifejezés értéke?

VAL(MID\$("A5B4C3D2E1",8,10))

- A. Nem kiértékelhető, mert
a kifejezés hibás.
- B. 20
- C. 54321
- D. 21
- E. Nem tudom.

4. SZ. MELLÉKLET

E L E M Z É S E K

BAS03 MENETSZAM ELOSZLAS

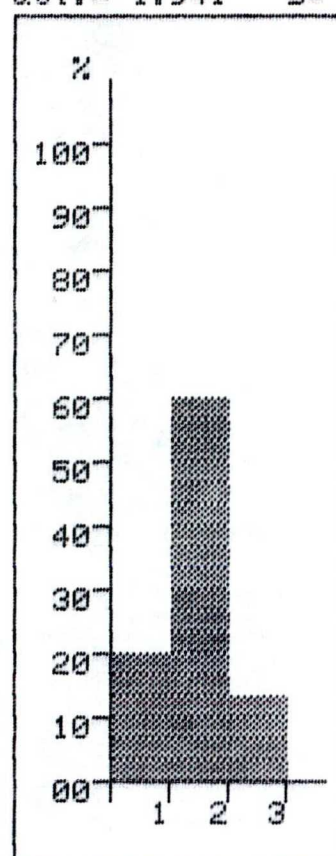
GYAKORISAGI TABLAZAT

atl.= 1.941 s= .608 CV= 31.3 % N= 51

osztaly	gyako- risa9	relativ gyak. %	kumulativ gyak.	relativ kum. %
1	11	21.6	11	21.5
2	32	62.7	43	84.3
3	8	15.7	51	100.0

BAS03 MENETSZAM ELOSZLAS

atl.= 1.941 s= .608 CV= 31.3 % N= 51



BAS07 MENETSZAM ELOSZLAS

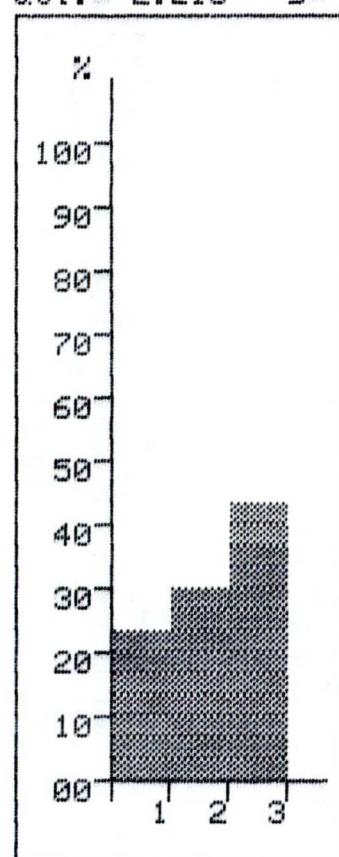
GYAKORISAGI TABLAZAT

atl.= 2.216 s= .8 CV= 36.1 % N= 51

osztaly	gyako- risa9	relativ gyak. %	kumulativ gyak.	relativ kum. %
1	12	23.5	12	23.5
2	16	31.4	28	54.9
3	23	45.1	51	100.0

BAS07 MENETSZAM ELOSZLAS

atl.= 2.216 s= .8 CV= 36.1 % N= 51



AS02K ELEMZESE

ATLAGOK ES SZORASOK

S.SZ.	VALTOZO	ATLAG	SZORAS	CV. %
1.		0.431	0.869	201.6
2.		0.157	0.957	609.6
3.		0.784	0.571	72.8
4.		0.314	0.918	292.4
5.		0.137	0.767	559.9
6.		0.784	0.571	72.8
7.		0.569	0.773	135.9

KORRELACIOS MATRIX

S.SZ	1	2	3	4	5	6
2	.25					
3	.03	-.08				
4	.49	.32	-.13			
5	-.32	.05	.02	-.28		
6	.07	.17	.16	-.13	-.25	
7	.13	.41	-.03	.11	.03	.06

AS04K ELEMZESI

ATLAGOK ES SZORASOK

S.SZ.	VALTOZO	ATLAG	SZORAS	CV. %
1.		0.216	0.775	358.8
2.		0.529	0.825	156.0
3.		0.373	0.862	231.1
4.		0.216	0.935	432.9
5.		0.216	0.935	432.9
6.		0.529	0.776	146.7
7.		0.765	0.580	75.8
8.		0.529	0.750	141.8
9.		0.216	0.695	321.8
10.		0.510	0.776	152.2
11.		0.843	0.500	59.3
12.		0.647	0.736	113.8
13.		0.196	0.971	495.4
14.		6.314	2.914	46.2

CORRELACIOS MATRIX

SZ	1	2	3	4	5	6	7	8	9	10	11	12	13
2	-.02												
3	.11	.58											
4	.13	.08	-.07										
5	-.09	.13	.29	-.03									
6	.1	.48	.53	-.02	.44								
7	.03	.3	.37	.06	.27	.1							
8	-.06	.4	.39	.01	.15	.33	-.12						
9	.1	.07	.16	-.13	.2	.33	.17	.08					
10	.11	.34	.42	.09	.2	.43	-.17	.48	.01				
11	-.01	-.13	0	-.05	-.01	-.19	.21	.07	-.24	-.2			
12	-.07	.28	.18	-.03	.11	-.08	.31	.05	0	.04	.22		
13	.2	.31	-.09	.06	-.02	-.01	-.02	-.03	.05	.1	-.06	.12	
14	.36	.58	.65	.13	.35	.68	.36	.36	.51	.43	-.06	.18	.17

BAS05/1

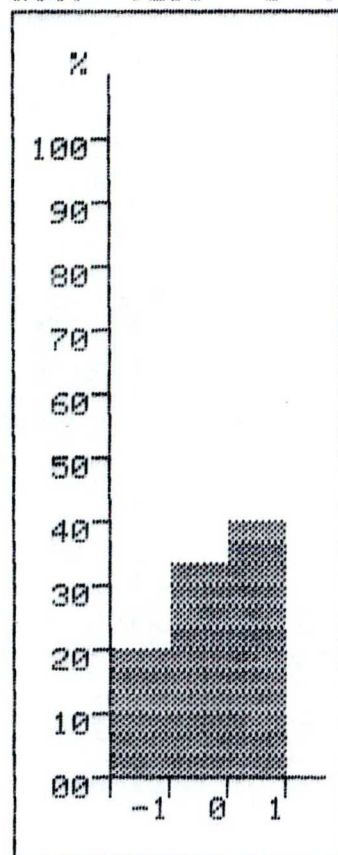
GYAKORISÁGI TÁBLAZAT

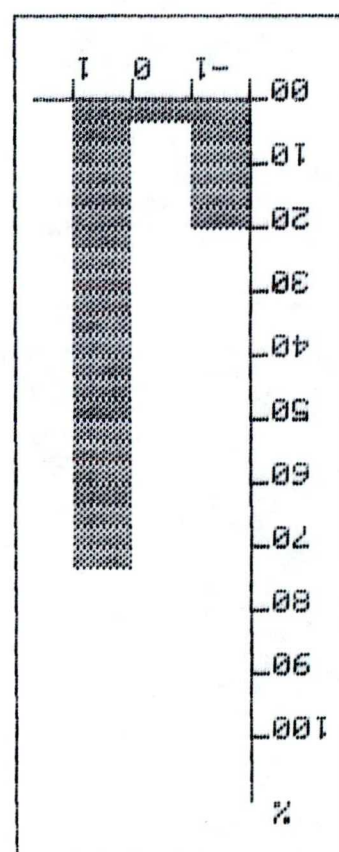
atl.= .216 s= .775 CV= 358.7 % N= 51

osztály	gyako- riság	relatív gyak. %	kumulatív gyak.	relatív kum. %
-1	11	21.6	11	21.5
0	18	35.3	29	56.8
1	22	43.1	51	100.0

BAS05/1

atl.= .216 s= .775 CV= 358.7 % N= 51





atl. = .529 s = .825 CV = 155.9 % N = 51

BRS05/2

osztály	gyak.-	relatív	gyak. %	kumulatív	relatív
-1	11	21.6	11	21.5	21.5
0	2	3.9	13	25.4	25.4
1	38	74.5	51	100.0	100.0

atl. = .529 s = .825 CV = 155.9 % N = 51

GYAKORISÁGI TÁBLAZAT

BRS05/2

BAS05/3

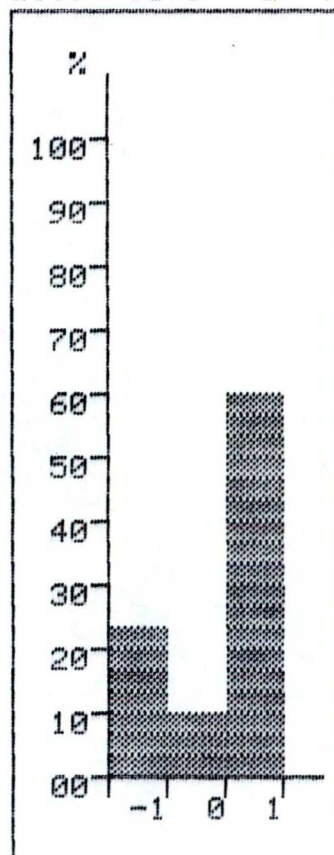
GYAKORISÁGI TÁBLAZAT

atl. = .373 s = .862 CV = 231 % N = 51

osztály	gyako- riság	relatív gyak. %	kumulatív gyak.	relatív kum. %
-1	13	25.5	13	25.4
0	6	11.8	19	37.2
1	32	62.7	51	100.0

BAS05/3

atl. = .373 s = .862 CV = 231 % N = 51



BAS05/4

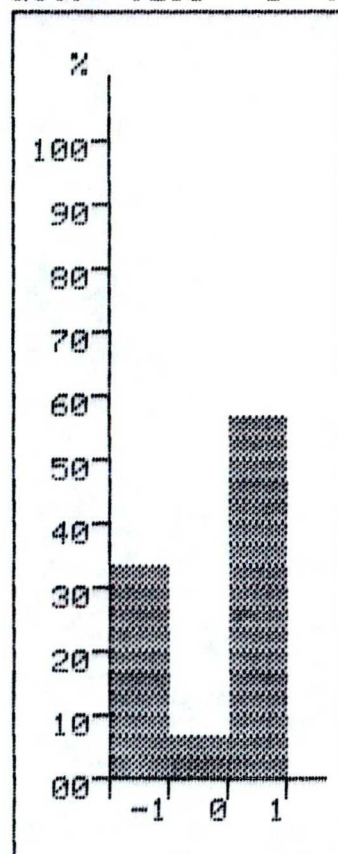
GYAKORISÁGI TÁBLAZAT

atl.= .216 s= .935 CV= 432.8 % N= 51

osztály	gyako- riság	relatív gyak. %	kumulatív gyak.	relatív kum. %
-1	18	35.3	18	35.2
0	4	7.8	22	43.1
1	29	56.9	51	100.0

BAS05/4

atl.= .216 s= .935 CV= 432.8 % N= 51



BAS05/5

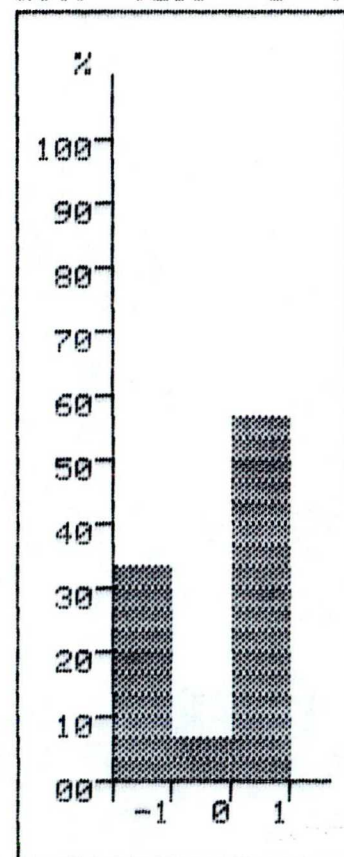
GYAKORISÁGI TÁBLAZAT

atl.= .216 s= .935 CV= 432.8 % N= 51

osztály	gyakor- sítás	relatív gyak. %	kumulatív gyak.	relatív kum. %
-1	18	35.3	18	35.2
0	4	7.8	22	43.1
1	29	56.9	51	100.0

BAS05/5

atl.= .216 s= .935 CV= 432.8 % N= 51



BAS05/6

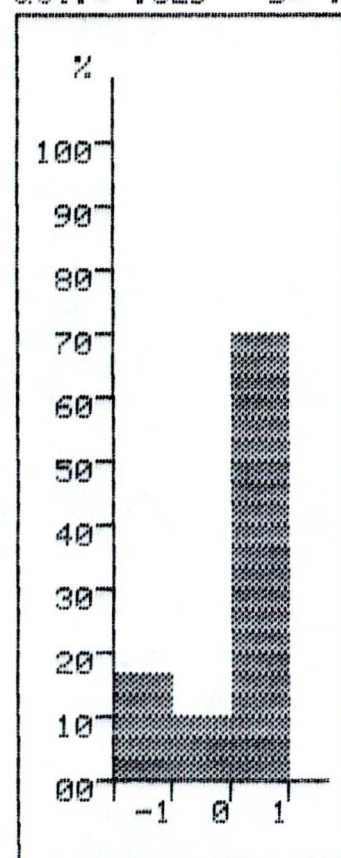
GYAKORISÁGI TÁBLAZAT

atl.= .529 s= .776 CV= 146.6 % N= 51

osztály	gyako- riság	relatív gyak. %	kumulatív gyak.	relatív kum. %
-1	9	17.6	9	17.6
0	6	11.8	15	29.4
1	36	70.6	51	100.0

BAS05/6

atl.= .529 s= .776 CV= 146.6 % N= 51



BAS05/7

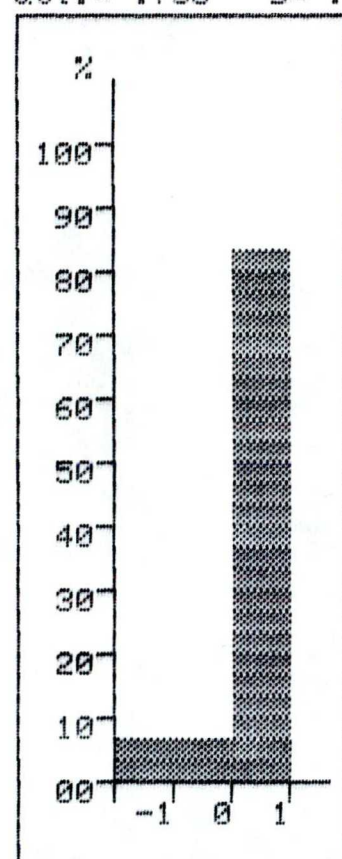
GYAKORISÁGI TÁBLAZAT

atl.= .765 s= .58 CV= 75.8 % N= 51

osztály	gyako- riság	relatív gyak. %	kumulatív gyak.	relatív kum. %
-1	4	7.8	4	7.8
0	4	7.8	8	15.6
1	43	84.3	51	100.0

BAS05/7

atl.= .765 s= .58 CV= 75.8 % N= 51



BAS05/8

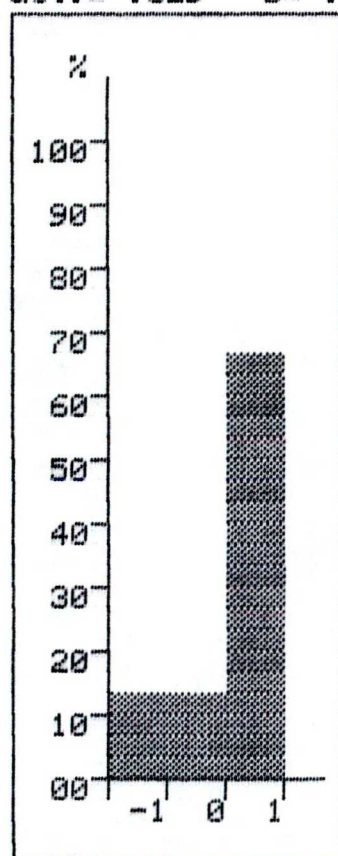
GYAKORISÁGI TÁBLAZAT

atl.= .529 s= .75 CV= 141.7 % N= 51

osztály	gyakor- iság	relatív gyak. %	kumulatív gyak.	relatív kum. %
-1	8	15.7	8	15.6
0	8	15.7	16	31.3
1	35	68.6	51	100.0

BAS05/8

atl.= .529 s= .75 CV= 141.7 % N= 51



BAS05/9

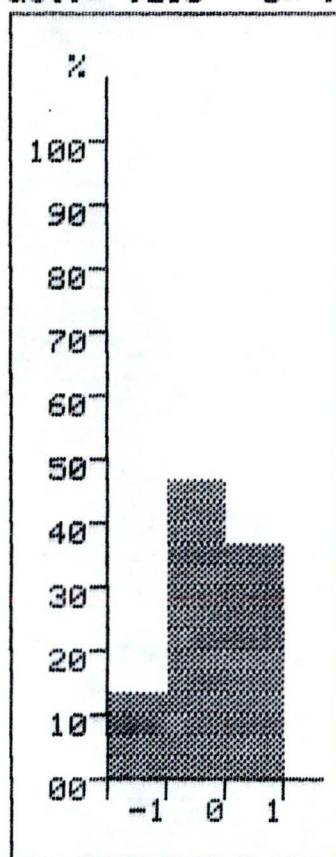
GYAKORISÁGI TÁBLAZAT

atl.= .216 s= .695 CV= 321.7 % N= 51

osztály	gyakor- sági	relatív gyak. %	kumulatív gyak.	relatív kum. %
-1	8	15.7	8	15.6
0	24	47.1	32	62.7
1	19	37.3	51	100.0

BAS05/9

atl.= .216 s= .695 CV= 321.7 % N= 51



BAS05/10

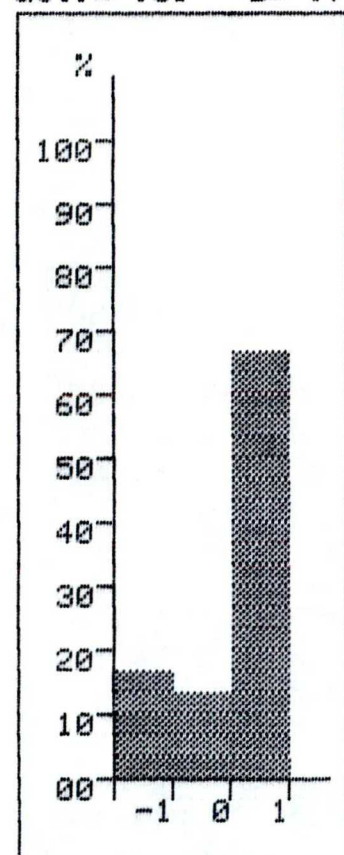
GYAKORISÁGI TÁBLAZAT

atl. = .51 s = .776 CV = 152.1 % N = 51

osztály	gyakoriság	relatív gyak. %	kumulatív gyak.	relatív kum. %
-1	9	17.6	9	17.6
0	7	13.7	16	31.3
1	35	68.6	51	100.0

BAS05/10

atl. = .51 s = .776 CV = 152.1 % N = 51



BAS05/11

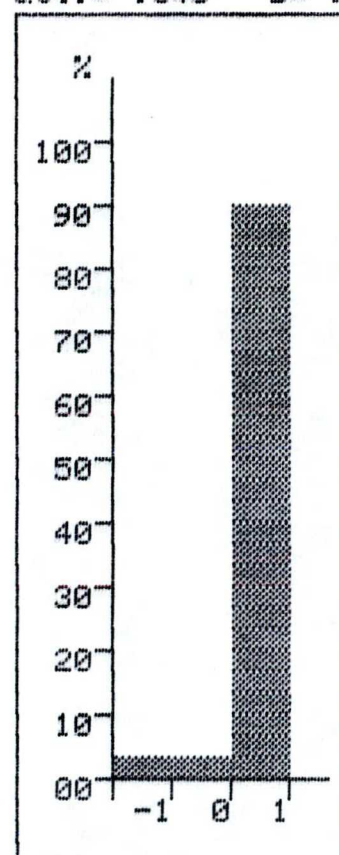
GYAKORISÁGI TÁBLAZAT

atl. = .843 s = .5 CV = 59.3 % N = 51

osztály	gyakoriság	relatív gyak. %	kumulatív gyak.	relatív kum. %
-1	3	5.9	3	5.8
0	2	3.9	5	9.8
1	46	90.2	51	100.0

BAS05/11

atl. = .843 s = .5 CV = 59.3 % N = 51



BAS05/12

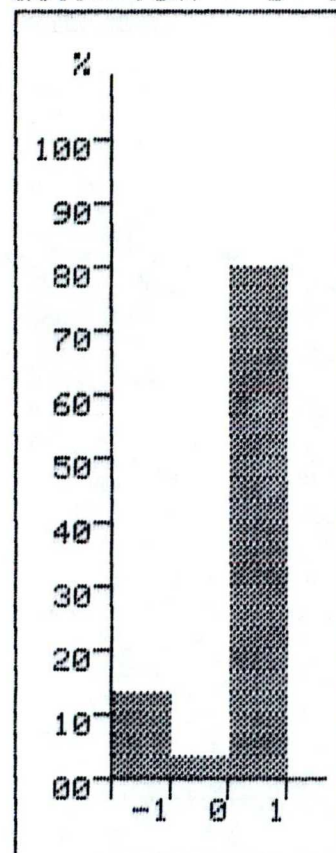
GYAKORISÁGI TABLAZAT

atl. = .647 s = .736 CV = 113.7 % N = 51

osztály	gyakor- iság	relatív gyak. %	kumulatív gyak.	relatív kum. %
-1	8	15.7	8	15.6
0	2	3.9	10	19.6
1	41	80.4	51	100.0

BAS05/12

atl. = .647 s = .736 CV = 113.7 % N = 51



BAS05/13

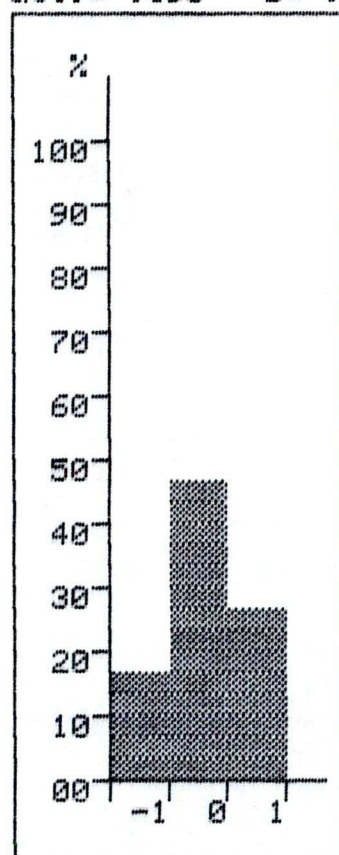
GYAKORISÁGI TABLAZAT

atl. = .196 s = .971 CV = 495.4 % N = 51

osztály	gyako- riság	relatív gyak. %	kumulatív gyak.	relatív kum. %
-1	10	19.6	10	19.6
0	25	49.0	35	68.6
1	15	29.4	50	98.0

BAS05/13

atl. = .196 s = .971 CV = 495.4 % N = 51



KLASZTERANALIZIS

BAS02 KERDESEK

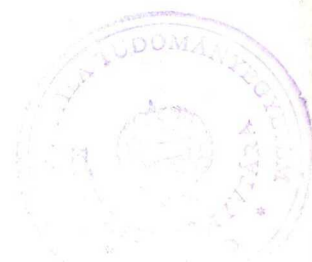
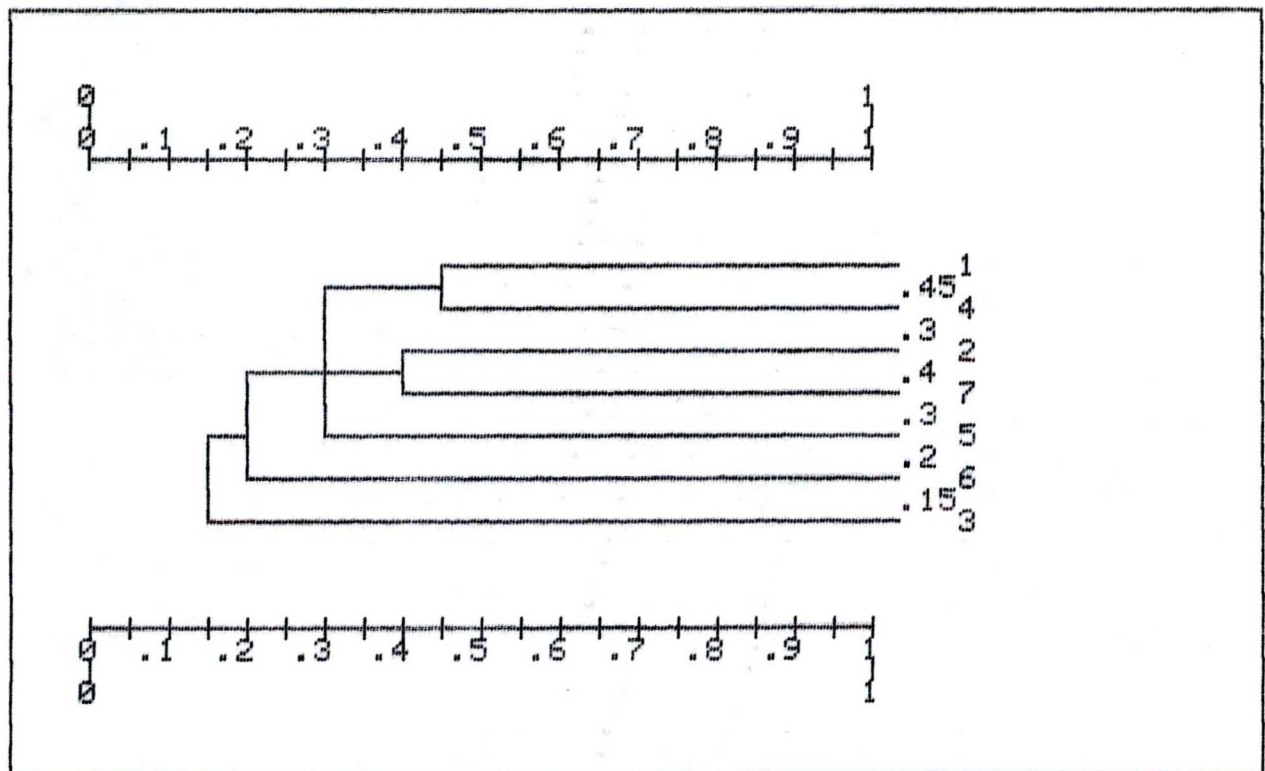
1985.x.23

K elemzes

A változók osztályozása
Single linkage
Korrelációs matrix
A változók száma = 7
A változók :

1
4
2
7
5
6
3

KLASZTERANALIZIS



KLASZTERANALIZIS

BAS02 KERDESEK

1985.x.23

K elemzes

A változók osztályozása

Complete linkage

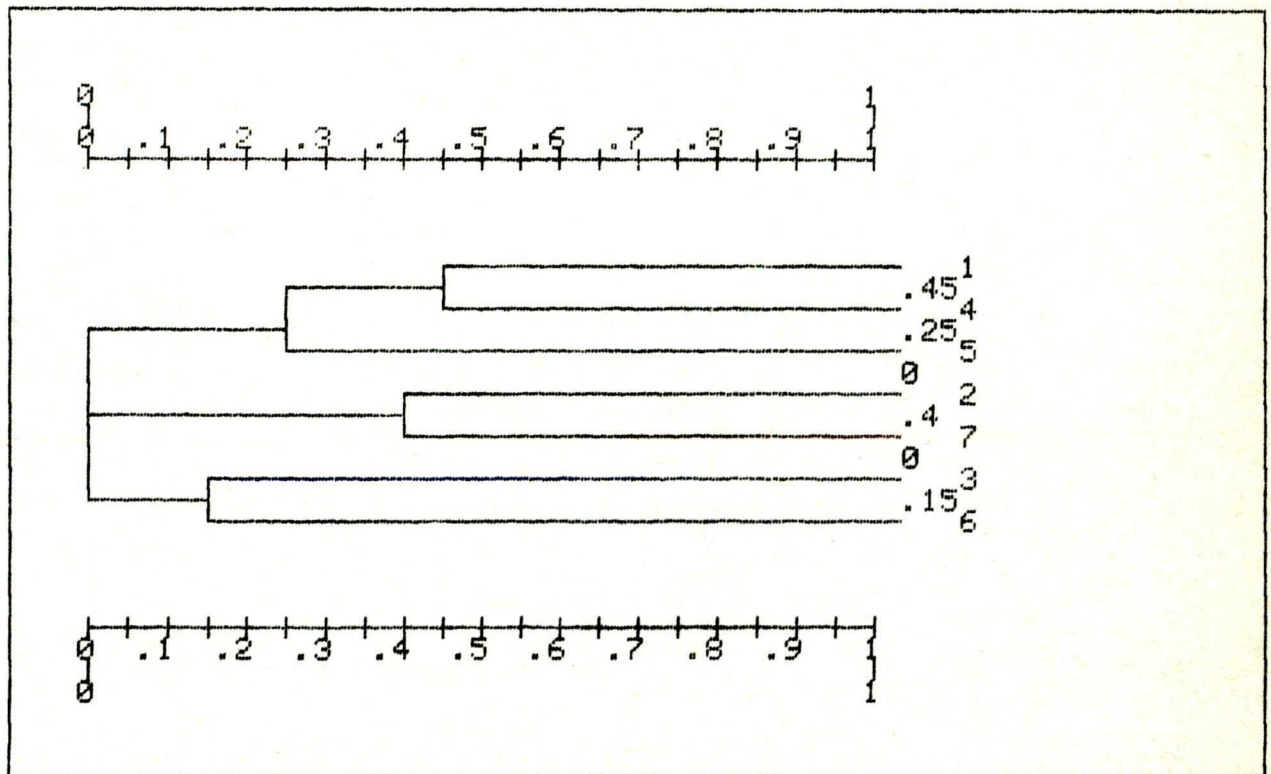
Korrelációs matrix

A változók száma = 7

A változók :

1
4
5
2
7
3
6

KLASZTERANALIZIS



KLASZTERANALIZIS

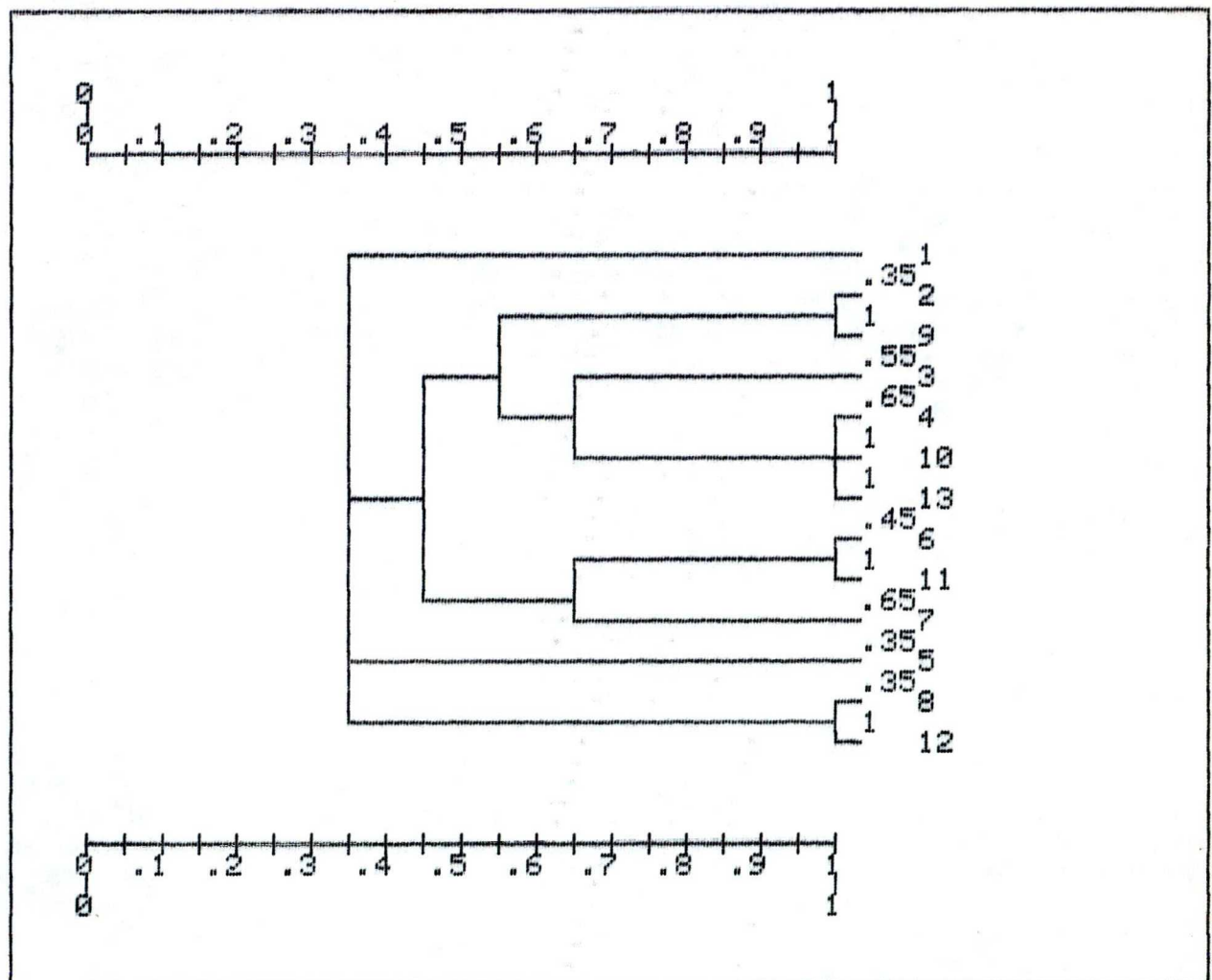
1985.x.23

BAS05 KERDESEK

K elemzes

A változók osztályozása
Single linkage
Korrelációs matrix
A változók száma = 13
A változók :

KLASZTERANALIZIS



KLASZTERANALIZIS

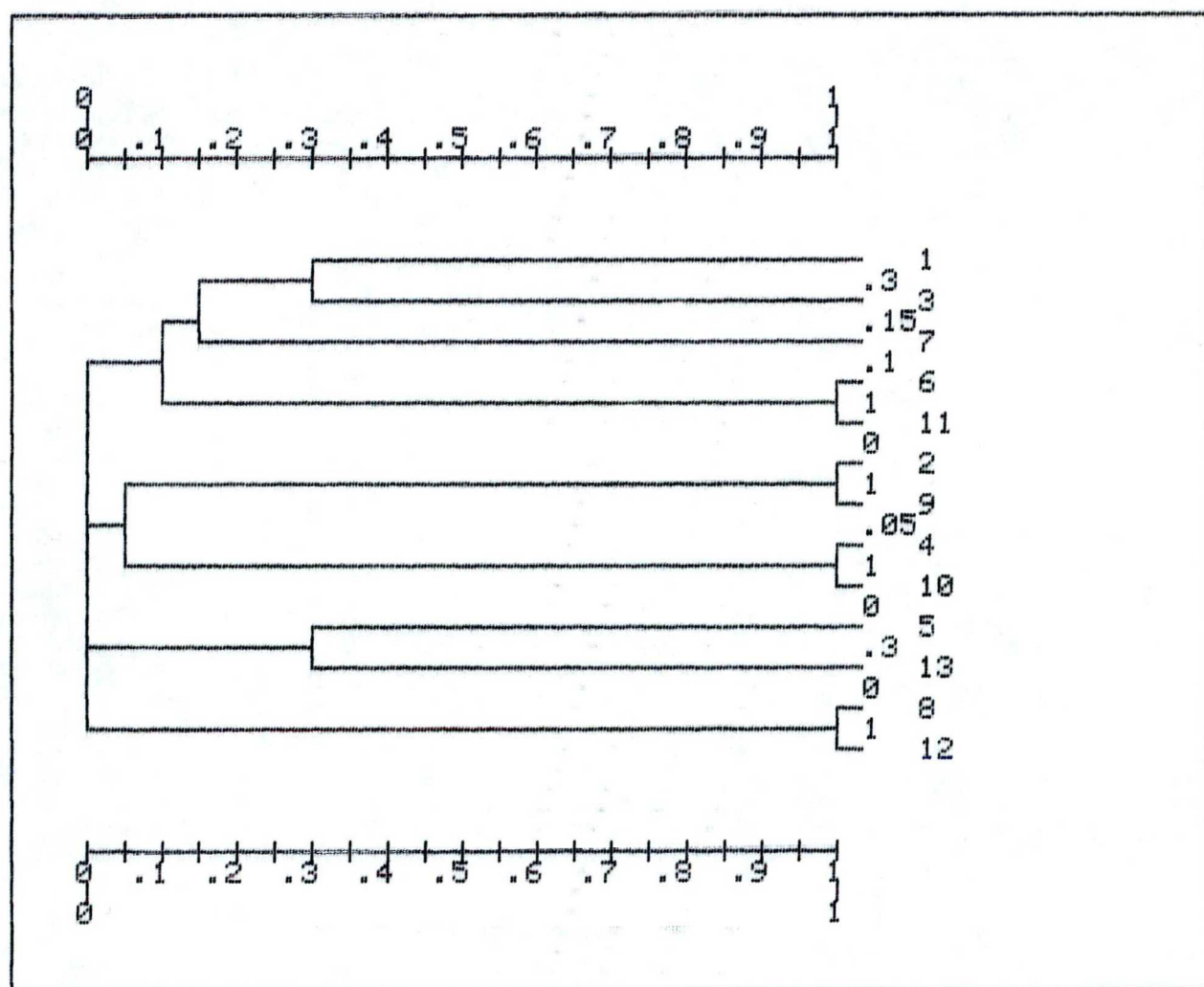
1985.x.23

BAS05 KERDESEK

K elemzes

A változók osztályozása
Complete linkage
Korrelációs matrix
A változók száma = 13

KLASZTERANALIZIS



Köszönetemet fejezem ki dr. Nagy József
egyetemi tanárnak, dr. Kispéter József főisko-
lai tanárnak, Vidákovich Tibor ösztöndíjas gyakor-
noknak és Muhi Piroska szakoktatónak, akik értékes
segítséget nyújtottak munkámhoz.