

1985/3

A NEUMANN JÁNOS SZÁMÍTÓGÉP-
TUDOMÁNYI TÁRSASÁG LAPJA

MIKROSZÁMÍTÓGÉP
MAGAZIN



Tanuláshoz, munkához, játékhoz pro primo:

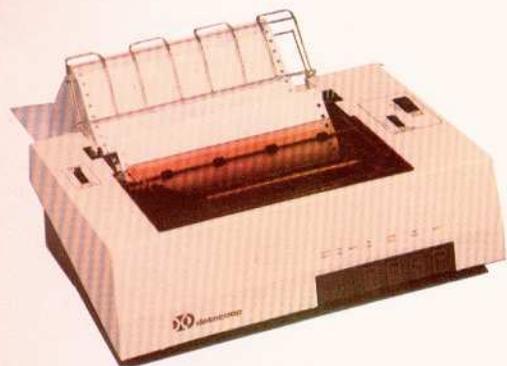


személyi számítógép



- BASIC programozási nyelv
 - billentyűzet: kapacitív elven működő magyar ABC kis- és nagybetűkkel
 - megjelenítés: 16 sor × 42 betűhely (szöveg) 256 × 192 képelem (grafikus)
 - program és adattárolás, kazettás magnetofonnal
 - TV készülékhez csatlakoztatható
 - háromféle változat memóriakapacitás (kbájt) szerint:
- | ROM | RAM | Fogy. ár (Ft) |
|-----|-----|---------------|
| 16 | 16 | 11 500,- |
| 16 | 32 | 16 150,- |
| 16 | 48 | 19 339,- |
- saját tápegység (4 600,- Ft)

DCD · PRT · 80 GRAFIKUS MOZAIKNYOMTATÓ



- Nyomtatható karakterek száma soronként: 80
- Nyomatási sebesség: 80 kar/s
- Karakterkészlet: 95 ASC II/IS
- Kétirányú nyomtatás logikai kereséssel
- Egyidejű másolatok száma: 1 eredeti 3 másolat
- Mikroprocesszoros vezérlés
- Egykártyás felépítés
- Változtatható papírszélesség
- Grafikus szimbólum készlet
- Bit vezérlésű grafika (tű címzés)
- Traktoros vagy gumigörgős papírtovábbítás
- Papírhiány érzékelés
- Dupla széles karakterek
- Automatikus soremelés
- Festékszalag kazetta
- DZM vagy CENTRONICS interfész


datacoop

Kapható:


ELEKTROMOBIL

2. sz. bolt
Budapest XIII.
Jászai Mari tér 5.
T: 530-800

**A kiadvány
a Tudományosrendszervezési
és Informatikai
Intézettel
együttműködve készül**

A szerkesztő bizottság
vezetője:
Kovács Győző

Munkatársak:
Broczkó Péter
(hírek)

Buday György István
(személyi számítógépek)

Jakab Agnes
(ember-gép kapcsolat)

Kovács Győző
(levelezés)

Lindner László
(sakkprogramozás)

Pataki Ernő
(programozástechnika)

Petróczy Judit
(könyvek)

Pogány Csaba
(alkalmazástechnika,
tanfolyam)

Simonyi Endre
(klub)

Takácsy Ildikó
(favágás)

Varga András
(iskola – számítógép)

Vass Nándor
(alkalmazások)

Votisky Zsuzsa
(játékprogramok)

Zárda Sarolta
(piac)

A szerkesztőség munkatársai:

Albert Tibor
Nacsa Sándor

Felelős szerkesztő:
Könyves Tóth Pál
Szerkesztőség:
1027 Budapest II., Fő u. 68.
Telefon: 154-250

Kiadja: a Delta Szaklapkiadó
és Műszaki Szolgáltató
Leányvállalat
Felelős kiadó:
Faklen Pál igazgató
1442 Budapest VII., Garay u. 5.
Telefon: 415-583, 215-440

Terjeszti a Magyar Posta
Előfizethető
bármely postahivatalban,
a kézbesítőknel,
a Posta hírlapüzleteiben
és a Posta

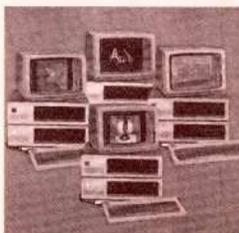
Központi Hírlap Irodában
(Budapest V., József nádor tér 1.
Postacím: 1900 Budapest)
közvetlenül
vagy postautóvalnyon,
valamint átutalással
a PKH 215-96162
pénzforgalmi jelzőszámra.
Előfizetési díj:
egy évre 180,- Ft,
fél évre 90,- Ft.

Szedte:
a Nyomdaipari Fényszedő Üzem
(857482/9)

Nyomás:
Petőfi Nyomda, Kecskemét,
Külső Szegedi út 6.
(85/50063)
Telefon: 28777
Felelős vezető:
Ablaka István igazgató

INDEX: 25629
ISSN 0236-6088

**Címképünk:
Magyarországon is
bejelentették
az IBM PC, PC XT
és PC AT típusú
személyi számítógépeket**



Tartalom

Ütjelentés	2
Számítógép a tanulásban	3
COMPUTER-M	9
„A profizmust hiányolom...”	12
Robottechnika	24
Word processing Japánban	26
µprogramok	34
A technológiák szerepe	47
Tudáspróba	47
Adok – veszek – cserélék	47

ISKOLA – SZÁMÍTÓGÉP

Kísérletek a képernyőn	4
A HT-1080Z számábrázolásáról	5
Gépi kódú kiegészítések BASIC-hez	6
Ahol élnek a lehetőségekkel	7

PROGRAMOZÁSTECHNIKA

Még egyszer az APL-ről	10
Lokális változók BASIC programokban	11
LOGO – a programozástanulás új útja	13

TANFOLYAM

Alapozás X.	16
-------------	----

µKLUB

Építsünk számítógépet! VIII.	29
Mi már építünk!	30
Kazettaszabvány	30
Ki ad magyarázatot?	32
Közérdekű válaszok az olvasóknak	32

SAKKPROGRAMOZÁS

Specializált számítógépet vagy programot?	36
Lépésről lépésre	38

JÁTÉKPROGRAMOK

Repülésszimulációk pilóta szemmel	40
-----------------------------------	----

AZ OLVASÓ ÍRJA

HÍREK, ÉRDEKESSÉGEK	46
---------------------	----

KÖNYVEK	48
---------	----

Útjelentés

*„De nem jó, ugy-e nem jó menni a messzi égen,
A felhők rajzain szenvedni oktan,
És nézni végtelen, némán, boldogtalan
Páris, a legszebb várost, ahol semmim sincs nekem”
(SZÉP ERNŐ: HÁZTETŐ A MONTMARTRE-ON)*

Sokáig gondolkoztam, hogy megírhatom-e franciaországi útjelentésemet szerkesztéségi cikk formájában. Az esetek többségében az útjelentés formális; nem azért készítk, hogy bárki is elolvassa. Az is zavart, hogy a μ M ez évi 1. számában megjelent „nyelvörkődésem” is egy francia–magyar megbeszéléshez kapcsolódott – nem lesz egy kicsit sok a franciákból?

Végül úgy döntöttem, hogy mégis megosztom az élményt az Olvasókkal, tehát elmondom, hogy mit is tettek a franciák a *társadalmi informatizálásért*.

Nem sokkal a látogatás előtt üzleti úton voltam Franciaországban. Szokásom szerint lerokdabtam a szállodában, és ugyanazzal a mozdulattal már be is kapcsolom a televíziót, amelynek képernyőjén rövidesen megjelent két kéz és egy személyi számítógép. Egy barátságos hang éppen azt magyarázta, hogy „ma olyan társadalomunk élünk, amelyben mindenkinek ismernie kell a számítástechnikát, tudnia kell kezelni a számítógépeket, mert nézzék csak, ez egyáltalán nem boszorkányság (mutatja), én is megtanultam, mert ha ezt nyomon meg (nyomja), akkor ez van a képernyőn, ha meg azt, akkor meg az. Nem is olyan hosszú idő múlva körülbelül ennyit kell tudnia mindenkinek, a munkásnak és a miniszternek egyaránt.”

Nagyon tetszett a műsor. Egy kicsit irigyletem is az előadót, olyan jól és pergően, sőt meggyőzően mondta a szöveget, és közben mutatta is a gépen, amiről beszélt. Aztán távolodott a kamera, és kiderült, hogy a francia miniszterelnök, Fabius úr volt az előadó.

Ezek után egyáltalán nem csodálkoztam, amikor a mostani tárgyalások alkalmával kiderült, hogy az egy éve elhatározott társadalmi programot miniszterelnöki utasítás gyorsította fel elsősorban azzal, hogy jelentősen megnövelte az anyagi támogatást, és létrehozta azt az intézményrendszert, amelynek a program végre kell hajtania.

Franciaországban az én ismereteim szerint ez a harmadik olyan központi támogatott akció, amelynek célja a franciák „fájdalommentes” átvezetése (az egyik

tárgyalópartnerem ötletes megfogalmazása) az informatika korába.

Az első volt a hatvanas években a *Plan Calcul*, ami megteremtette Franciaországban a számítástechnikát és az elektronikai ipart. Rengeteg pénzbe került akkor, de később még többre került volna. A második akció 1975-ben a *távbeszélő-hálózat korszerűsítése* és kiterjesztése volt, amelynek eredményeképpen 3 év alatt a 100 főre jutó 5 készülékről 42-re bővült a telefonhálózat, ami azt jelenti, hogy a francia családok 98 százaléka van ma telefonja. Most indult a harmadik akció, amelynek az a célja, hogy minden francia értse az informatikához.

Az előző két vállalkozás alapozta meg ezt a harmadikat, amelyet az első kettő nélkül nem lehetne hatékonyan végrehajtani.

Joggal merül fel a kérdés, hogy szükség van-e egyáltalán a társadalmi informatizálásra. Nem csupán egy gyorsan terjedő divathullámról van szó, amelyre kidobnak egy csomó pénzt, és az eredmény semmi. Annnyit még el kell mondanom – ezt a franciák is többször emlegették –, hogy ami a célkitűzést, a módszereket és az eredményeket illeti, a francia és a magyar társadalmi program nagyon közel áll egymáshoz, majdnem azonosak. Legfeljebb a rendelkezésre álló anyagi eszközökben van vagy két nagyságrendnyi különbség, pedig csak ötször annyian vannak, mint mi. Ezenkívül más volt a célja a program indításának is.

Franciaországban a polgárok megtanítása az informatikára nemcsak gazdasági, hanem egyre inkább politikai kérdés. A francia kormány néhány időket él át, a szocialista kormányra hárul a feladat, hogy gazdaságatlan és ugyanakkor nagyon sok munkát foglalkoztató iparágakat megszüntessen, bevezesse a robotikát, ami ugyancsak a fizikai munkalehetőségeket csökkenti. Ezeknek az intézkedéseknek a következménye, hogy a munkanélküliek száma sohasem volt olyan magas, mint mostanában.

Nemcsak Franciaországban, de az iparilag fejlett országokban mindenütt minden 10 új munka-

helyből csak egy a fizikai. Az elmúlt 10 évben az informatikában alkalmazottak száma ugyanakkor meghatszorozódott. Ezt a franciák világosan látják, és ezért felgyorsították a társadalmi informatizálására indított programot. Arra számítanak, hogy az így kiképzett emberekkel feltölthetik a ma még betöltetlen, egymás után keletkező új informatikai munkahelyeket, és ezzel erőteljesen csökkenthetik a munkanélküliek számát. Az sem lényegtelen, hogy ezek az emberek a monoton fizikai munka helyett alkotó szellemi munkával foglalkozhatnak.

Franciaországban – miként nálunk is – az iskola-számítástechnikát is a társadalmi program szolgálatába állították. A kormány „megszarolta” az iskolákat. Azt mondta, akkor kapnak számítógépeket, ha azokat legalább napi 10 órában használják, elsősorban az iskolai munkában, másodsorban – délután és este – átengedik bárkinek, aki csak bejön az iskolába, hogy dolgozzon rajtuk.

A franciák is ugyanazokkal a problémákkal küzdenek, mint mi, nagyon sokan idegenkednek a gépektől, nem tudják eldönteni, leüljenek-e melléjük vagy sem, vegyenek-e gépet vagy ne. Franciaországban ebbe a kategóriába tartoznak a kisfarmerek, a kereskedők, a magánvállalkozók. Arra biztatják őket, hogy menjenek be a legközelebbi iskolába, üljenek le a gépek mellé és csinálják meg a mérleget, a napi elszámolást, aztán vigyék haza a mágneslemezt, és ezt egészen addig folytassák, amíg nélkülözhetetlenné nem válik számukra a gép. Akkor már bátran vehetnek gépet.

Valahogyan itt hat vissza a társadalmi program a gazdasági életre. Franciaországban ugyanis az amerikai, japán, de még az angol számítógépek ára is jóval alacsonyabb, mint a francia gépeké. Azt hiszem, hogy a társadalmi programra szánt pénzt azért vásárolták végül francia gyártmányú gépeket, mert ezzel nemcsak fel tudták tüntetni a számítógépgyártást a gazdaságos sorozatnagyságra, és így versenyképes árakat tudtak produkálni, hanem azért is, mert akik az iskolai számítóközpontokban a francia gépeket ismerik meg, azok később ezeknek a vásárlóivá válnak. Így a gazdasági vezetés arra számíthat, hogy a gyártási volumen még tovább növekszik, ami újabb munkaalkalmakat jelent.

A rendszeres felmunkatörtétszása és a társadalmi számára hasznos projektek támogatására a kormány önálló intézményrendszert szervezett, az ADI-t (Agence de l'Informatique) és az általa irányi-

tott klubozogalmat, amelyet X 2000-nek neveztek el.

Az ADI feladata, hogy állami pénzezből támogatást adjon olyan projekteknek, amelyek az informatika társadalmi méretű elterjedését segítik. Csak néhány példa. A francia posta létre akarta hozni az elektronizált postahivatalokat. Az ADI azt mondta: ez egy előremutató projekt, támogatom. Megbeszéltek, hogy vásárolnak jó néhány gépet, és a projektben részt vevő postahivataloknak szétosztják, mindenki csináljon vele, amit tud. Volt olyan hivatal, amelyik köszönettel nem kért a gépekből, de volt, amelyik több gépet is kért. Megkapta. Aztán egy-két hónap múlva az ADI megnézte, hogy mit csináltak a gépekkel. Ahol okos dolgokat találtak ki, ott újabb támogatást kaptak, például programozói segítségét vagy újabb gépeket. Egy fél év alatt elkészült vagy félszáz alkalmazói program. Ezeket összefésülték, megigazították, rendszerbe fogták össze, és még azon az őszön bemutatott a SICOB-on az elektronikus postahivatal. A programokat sokszorosították, megszervezték a többi postahivatalban a tisztviselők betanítását, és a probléma el volt intézve. Egyébként elmondták, hogy a szakértőknek az volt a véleményük, hogy egy rendszert nem szabad így létrehozni. Ők a szokásokat javasolták: tanulmány, elfogadják, módosítják, véglegesítik, megtervezik, programozzák, tesztelik, bemutatják, elfogadják, bevezetik – és erre 2–3 évet szántak. A postásoknak azonban jobban tetszett a fél év – szabálytalanul. Ki tudja, miért?

Ugyanígy segítette az ADI a miniszteriumok információs rendszerének kialakítását. Külön felelőse van a fizikailag hátrányos helyzetű gyerekek informatikai oktatásának, támogatják a sport és az informatika közötti együttműködést (például az edzők továbbképzése terminálon), de a turisztikai és utazási adathálózat kialakítását és más, fantasztikusnak tűnő projekteket is.

Az X 2000 és a magyar mikroklubok között megint csak az a különbség, hogy ők sokkal gazdagabbak. Az X 2000 klubokat az X 2000 Fondation hozza létre. Ha valahol egy ilyen klubot akarnak szervezni, akkor elsőrendű követelemény, hogy a helyi szervezeteknek pénzük legyen, mert az X 2000 Fondation az alapításhoz csak annyival járul hozzá, amennyivel a helyi közigazgatás, gyárak és intézmények együttesen. A Fondation gépeket és oktatási technológiát ad, gyakorlatilag minden segítséget – kiképezi például a személyzetet –, hogy a klub jól működjön.

Az anyagi támogatás csak a induláshoz elég, aztán a központi támogatás megszűnik, a klubnak el kell tartania magát. Ezért üzletek kötnek, például a környék gyáraival, amelyeknek munkáit a gyár költségen kiképezik és továbbképzik. Aki munkanélküli, annak nem kell a tanfolyamokon fizetnie, és ingyen használhatja az X 2000 klub valamennyi gépét és a könyvtárat. Ha akar, beülhet melegezni is. Ha egyszer odamegy, akkor ott is marad, erre számítanak. A gép- és könyvtárhálózat a tanfolyami időn kívül mindenki számára ingyenes. Az X 2000 klubok évente nagy összejöveteleket tartanak, amelyek célja a tapasztalatok kicserélése, új oktatási módszerek bemutatása, egyáltalán az évi egyszeri találkozás a személyi kapcsolatok kialakítására.

Megnéztük az ASPASIE klubot Torcyban. Torcy egy kisváros kb. 60 km-re Párizstól, a Val Maubue új városnak a központja. Franciaországban is van ugyanis olyan program, amely a kisebb településeket egyesíti, ezáltal új városok jönnek létre. Val Maubue is így született, hat kis településből, amelyek megtartották önállóságukat, sőt a nevüket is. Az új város elnökét választják. Val Maubue vezetője Torcy polgármestere. Vannak dolgok, amelyeket közösen intéznek, közösen építenek. Ilyen volt az ASPASIE Központ is, ami ha úgy tetszik, akkor információs központ, de kultúrház is, klub is. („Sokta került, de megérté” – mondta az elnök.)

Van ennek az ASPASIE-nek egy különlegessége: saját információs rendszert üzemeltet. A rendszer megértéséhez el kell mondani, hogy a már említett telefonhálózat korszerűsítéséhez a népszerűen elektronikus telefonkönyvnek nevezett rendszer is hozzá tartozik. Franciaországban minden embernek ki telefonkönyvet, amelyről a megbeszélés résztvevői egymást túlléctálva közölték, hogy „boszszantó”, mert a sok változás miatt az év végére már gyakorlatilag használhatatlan, ezért a kinyomtatása kidobott pénz.”

A posta is így gondolta, ezért határozta úgy, hogy a mintegy 22 millió telefonkészülék mellé ugyanannyi Minitel terminált is elhelyeznek – természetesen ingyen – a telefonkönyv helyett. A terminállal egész Franciaország összes telefonszámát le lehet kérdezni. Nem akartam elhinni, de azt mondták, a telefonkönyv költségének, valamint a szerkesztési, oda- és visszazárítási költségnek a megtakarításával a terminál két-három év alatt amortizálódik. A telefon-előfizetők egy része a terminál

inm gényli, ők egyelőre telefonkönyvet kapnak, a többiek előre meghatározott terv szerinti időben a terminált kapják meg. Val Maubue vezetése ezekre a terminálokra építette a tervét és alakította ki az új város információs rendszerét.

A posta hatékony közreműködésével létrehoztak egy kis lokális hálózatot, amelynek használata természetesen ingyenes. A központi gép az ASPASIE-ben van, a csatlakozó adatbázisok pedig az egyes településeken. Minden számítógép kétélféle célt szolgál. Egyrészt ezekben tárolják a mindenkit érdeklő adatokat (a sport- és kulturális eseményeket, a polgármester közleményeit, az országos rendezvények idejét stb.), másrészt postaládaként használják. Ebben bárki elhelyezhet címzett vagy mindenkinek szóló közleményeket, üzeneteket.

Miután minden lakásban van telefon, a postaládától azt várják, hogy közelebb hozza egymáshoz az új város lakóit, például az azonos érdeklődésűket megtalálják egymást, de közelebb kerül egymáshoz a városok vezetése és a polgárok is. A polgármester maga is arra számít, hogy az emberek nem járnak be a városházra ügyeiket elintézni, hanem leadják azokat a postaládába, aztán megkapják a választ, esetleg vitatkoznak is. Úgy számolta, hogy kb. napi két órát fog a terminálnál az üzenetek megválaszolásával eltölteni, és nem 4-5 órát félfogadással.

A rendszert tovább akarják fejleszteni. Ha már mindenki hozzá szokott a terminálhoz, akkor lehetőség lesz például otthoni munkahely kialakítására, hiszen a lokális hálózatokon keresztül elérhető lesz a postai adathálózat is, akkor pedig a munkahelyi számítógépek is lehet hazáról dolgozni.

Lehetne még beszélni az iskolai számítógép-hálózatról, ahol a diák meg tudja tanulni az adatkezelést és megismerheti a telekommunikációs lehetőségeket, a hátrányos helyzetű gyerekek részére készített speciális számítógépekről és perifériákról, az óvodás korú gyerekek játékos programozásitanításáról, de be kell fejeznem.

Amit láttunk, az nálunk sem elérhetetlen; illetve csak azt a részét nem tudjuk megvalósítani – sajnos –, amire az infrastruktúra hiányzik, az olcsó számítógép és a telefonhálózat.

Azt hiszem, ha egy iparilag fejlettségi országban érdemes az informatika társadalmi méretű bevezetését támogatni, akkor ez nálunk sem kidobott pénz – igaz, a haszon nem jelentkezik közvetlenül és azonnal. De jelentkezik!

KOVÁCS GYÖZŐ

Számítógép a tanulásban

A számítógép az oktatásban nagyon mély és radikális változást fog eredményezni. A változás befolyásolásához új adatokra, statisztikákra és elképzelésekre van szükségünk. Elkerülhetetlen a gondolkodás új elméleti megfogalmazása.

A hagyományos oktatási pszichológia bizonyos beavatkozások hatását vizsgálja. Például: mi a hatása a halmazelmélet tanulásának a számolási képességre? A számítógép esetében a gép hatását vizsgálják a gondolkodás fejlődésére, az iskolára, az osztályra. Ez azonban teljesen értelmetlen. Azt feltételezik, hogy a programozás kognitív hatása jelentkezik a megjavult tervezési képességen.

Ha a programozás befolyásolja a gondolkodást, akkor ez közvetlenül nem a programozás elsajátítását jelenti. A hatása inkább indirekt. A programozási jártasság lehetővé teszi, hogy a tanár új módon tanítson, a diákok pedig új módon tanuljanak.

A számítógép olyan radikális változást eredményez a tudáselsajátítás folyamatában, mint amilyen a ceruza használata hozott az oktatásban. Ennek kezdetét már látjuk a számítógép széles körű felhasználásában: egyesek számára már ma is napi használati eszköz.

Mostanában érkezünk el oda, hogy lehetségessé válik a számítógép oktatásban történő alkalmazásának alaposabb megfigyelése. Ennek előfeltétele, hogy viszonylag sok számítógép álljon rendelkezésre. Remélhetőleg közeleg az az idő, amikor minden tanulóknak lesz számítógépe.

Ma még súlyos helyzetet teremt a számítógéphiány. A gépek jelenlegi használata inkább kárt okoz az oktatás szempontjából, mert alacsony a képzélszint és erősíti a téves felfogásokat. A legtöbb fiatal nem kerül a géppel olyan kapcsolatba, hogy ellenőrizni tudná azt. Ez a hiányosság persze nem teszi szükségessé a számítógépek használatát, de fontos a számítógép és ember viszonyának megfogalmazása.

Kísérletek bizonyítják, hogy az iskolában inaktív gyerekek a LOGO használata közben igen aktivákká váltak. A program használatával törekedtek eredményeik javítására, és végül elemi módon kapcsolatba kerültek a matematikával. A LOGO felhasználása három változó: a sebesség, az idő és a távolság megismerését tette lehetővé. Nem valószínű, hogy ezt számítógép nélkül el lehetett volna érni. A példából látszik, hogy a számítógép segítségével olyan személyiségtipusok is bekapcsolhatók az oktatásba, tanulásba, akik a hagyományos oktatásban nem motiváltak. A gép módot nyújt arra, hogy a tanulás tárgyát és módját a személyre szabjuk.

A számítógép alkalmazásával megvalósítható az ún. tanítás nélküli tanulás: kissé eltúlozva, a gép használatával a gyerekek egyedül tudnak tanulni. Ma már vannak tapasztalatok olyan iskolákból, ahol sok a számítógép. Itt a tanár szerepe megváltozott: már nem oktat, hanem figyeli a diákok munkáját, segít, ha valaki elakad.

A számítógépnek megvan az a különleges tulajdonsága, hogy interaktív és kísérleti. Így a tudás elsajátítása lépésenkénti és folyamatos lehet. A LOGO létrehozásának is ez volt a fő gondolata. Jelentőségei akkor érthetik meg igazán, ha szembeállítjuk a tanulás és tanítás eme új felfogását a hagyományossal, ahol az ismeretátadás és elsajátítás mindig egyszerre, nagyobb adagokban történik, passzív befogadás mellett.

Írásunk *Seymour Papert* New Theories for New Learnings című, a School Psychology Review-ben 1984 októberében megjelent publikációja alapján készült.

– illyész –

Kísérletek a képernyőn

Első osztályban és a második osztály első hónapjaiban, amíg a technika tantárgy keretében sorra nem kerülnek a számítástechnikai ismeretek, csak azok a tanulók tudják használni a számítógépet, akik általában iskolában vagy a gimnáziumban részt vettek számítástechnikai szakköri munkában. Kiemelkednek azok a diákok – és egyre többen vannak –, akiknél már otthon is működik egy-egy ZX81, Spectrum vagy Commodore 64.

Az alábbiakban arról lesz szó, hogyan használható a számítógép a kémia- és fizikaórákon a gimnáziumokban.

Mindezenre a számítógép akkor alkalmazható jól az oktatásban, ha a tanulók hozzá szoknak a jelenléthez, és nemcsak kivételes esetekben találkoznak vele. Ezért mindegyikük számára lehetőséget kell teremtenünk az egyéni ismerkedésre a géppel.

A TIJ-től kémiai és fizikai témákat feldolgozó oktatási programokat kaptam kipróbálásra, a pedagógiai gyakorlatban való alkalmazhatóság vizsgálatára. A programok két gépet, a HT-1080Z és az ABC-80 számítógépen futtatottak.

A gimnáziumok első osztályos fizika tananyagához a TIJ programcsomagot forgalmaz, amely a munkafüzet 6., 7. és 8. feladatlapjához kapcsolódik. A programcsomagban az a legnagyobb előnye, hogy a tanulók kísérleteit kiegészíti, illetve gyorsabbá teszi, de az értelmezést a tanóra bízza.

A 6. feladatlap a tankönyv „Miért terjed ki a gáz?” című témájához kapcsolódik. Céлом itt az volt, hogy a tanulókkal elsajátíttassam a termodinamika második fő tételét, mint a statisztikus törvényszerűség eredményét. Felhívtam figyelmüket a modellfogalom lényegére, hogy az bemutassa a tanulmányozott jelenség bizonyos jegeit, szerkezetét és dinamikáját.

Az óra elején összefoglaltuk a gázok addig megismert tulajdonságait és az ezek alapján kialakított modellképünket. Ezután egy játékos modellkísérlettel haladtunk tovább, melyet a tanulók párokban végeztek el. A kísérlet címe „Játék a darazsakkal”. Lényege a következő.

Játék a darazsakkal

Minden pár két egymásba nyíló szobát rajzol. Az egyik szobába 6 darab, 1–6-ig megszámozott papírdarabkát tesznek, a játékoság kedvéért ezeket darazsoknak nevezzük. A darazsak véletlenszerűen röpöknek az egyik szobából a másikba, amit a tanulók először dobókocka segítségével biztosítanak. Amelyik darázs számát dobják, az átrepül a másik szobába. Minden tizedik dobás után fel kell jegeznük a darazsak eloszlását a szobákban.

A kísérlet kiértékelésekor kiderül, hogy a leggyakoribb eloszlás az, amikor három darázs az egyik és három a másik szobában van, és a legkisebb valószínűséggel fordul

elő az, hogy mind a hat darázs az egyik helyiségben lenne.

Sok dobás elvégzéseére nincs idő, ezért a tanulók elkezdik ugyan a kísérletet, de húsz-harminc dobás után abbahagyják, és a programot tekintik meg.

Elmondom, hogy ez egy szimulációs program: a képernyőn figyelemmel követhetjük, hogy melyik darázs mikor megy át. A kockadobás szimulációjára a számítógépbe épített véletlenszám-generátor szolgál, amely ebben az esetben 1 és 6 között állít elő számokat. 60 dobás után a gép kiírja minden darázs átrepülése utáni eloszlást, így a folyamatot dinamizmusában lehet látni.

A program segítségével a kísérlet sokkal gyorsabban tudjuk elvégezni. A darazsak eloszlásának kiírása után a bal illetve a jobb oldalon átlagosan tartózkodó darazsak számát grafikon ábrázolja. Így igen szemléletesen látható az egyenletes eloszlás.

A darazsak játék modellszerűen magyarázza a gázok szétterjedését, és ennek felhasználásával jutunk el a termodinamika második fő tételének egyik megfogalmazásához.

Osztokodás az energián – Az energia egyenletes eloszlása

A 7. és 8. feladatlaphoz kapcsolódó programok is a kockadobások felgyorsítását teszik lehetővé, és így alapvető fizikai törvények statisztikus értelmezéséhez adnak segítséget.

Disszociáció

Ezzel a programmal egy elemi kémiai reakciót figyelhetünk meg, amelyben három anyag vesz részt. Megvizsgálhatjuk az $A + B \rightarrow C$ és $C \rightarrow A + B$ reakciótipusokat, valamint mindkét irányú reakciót egyszerre. Megadható mindkét reakció sebessége, és az egyes anyagok kiindulási koncentrációja. A program kétféle eredményt adhat: vagy az egyes molekulák százalékos megoszlásáról készít grafikon, vagy a reakcióterben zajló eseményeket figyelhetjük meg. A bemenő paraméterektől függően megfigyelhető az egyesülés, a bomlás és a disszociáció.

A periódusos rendszer elemei

Ezt az oktatóprogramot tartom a leghasznosabbnak és legeredményesebbnek. Játékos for-

mában gyakoroltatja az elemek nevének és vegyjelének, a periódusos rendszerben elfoglalt helyüknek és annak az elektronszerkezettel való kapcsolatának ismeretét. A programnak három üzemmódja és három nehézségi fokozata van: 1. a megjelenő vegyjelkezhz tartozó neveket kell beírni; 2. a megjelenő elemek vegyjelét a nyílak segítségével a periódusos rendszerben elfoglalt helyüknek megfelelő négyzetbe kell mozgatni; 3. a véletlenszerűen kiválasztott elem elektronszerkezetét kell beírni.

Minden játék végén, 10 kérdés után a program értékeli, a hibás válaszokat kiemeli, és a periódusos rendszerben villogtatja ezeknek az elemeknek a vegyjelét. A képernyőn megjeleníthető a teljes periódusos rendszer. A kérdésekre adott válaszokat hangeffektusok kísérik, ami motiváló hatású.

ELEK

Segítségével az elemek elektronszerkezetét lehet gyakorolni. Két üzemmódban használható. Az első esetben megjeleníti egy elem elektronszerkezetét. A második üzemmódban egymás után néhány elem elektronszerkezete jelenik meg, esetleg hibásan, és el kell dönteni időre (több sebességi fokozat van), hogy helyesek-e. A játékok a gép pontozza.

OKTE

A molekulák elektronszerkezetének gyakorlását teszi lehetővé. Szintén két üzemmódban van. Az elsőben megjeleníti egy molekula elektronszerkezetét. A másodikban a molekulák hiányosan jelennek meg a képernyőn, és a tanulóknak kell kiegészíteni azokat.

Sajnos a program csak 19 molekula elektronszerkezetét képes ábrázolni, így a választási lehetőség kicsi. A játék végén a számítógép összesítést végez, és értékeli a tanuló munkáját.

Analitikai program

Ugyanúgy, mint az előző kettő, a számítógépnek azt a tulajdonságát használja ki, hogy nagy mennyiségű adat tárolására képes. Ezzel a programmal 15 fajta ionos vegyület reakciói lehet tanulmányozni 9 fajta reagenssel. Minden párosításnál kiírja a számítógép, hogy van-e változás a reakció során, vagy nincs, illetve a keletkezett oldat vagy csapadék milyen színű. A tanuló választhat egy olyan üzemmódot, amelyben egy ismeretlen vegyületet kell meghatározni reakciói alapján. A számítógép 15 vegyület közül választja ki véletlenszerűen az ismeretlent, ezek közül 7 kationja a Fresenius-féle csoportosítás szerint a kationok első osztályába, a maradék 8 a harmadik osztályába tartozik.

MEND

Az elemek tulajdonságainak változását a periódusos rendszer egy kiválasztott oszlopán (VIIIa) belül lehet megfigyelni, gyakorolni ezzel a programmal, amely tulajdonképpen egy „jós-lás-játék”. A program a VIIIa oszlopban különböző fizikai sajátságokat tüntet fel, de egy érték a kiírásból mindig hiányzik. Ezt kell a

A HT-1080Z számábrázolásáról

tanulónak megjósolnia. A program célja, hogy megismertesse a periódusos rendszer egy oszlopán belül a fizikai tulajdonságok változásának tendenciáját.

Kémiai egyensúly

A program a $H_2 + I_2 \rightleftharpoons 2HI$ egyensúlyi folyamat dinamikájának demonstrálására készült. A hidrogén-jodid képződése egyike ama kevés folyamatnak, amelyek mindkét irányban egyszerű mechanizmus szerint, jól mérhető sebességgel mennek végbe. A programban megadhatók a kezdeti koncentrációk, majd a képernyőn egyszerre jelenik meg numerikusan és grafikusán a kiindulási és az egyensúlyi beállta utáni összetétel százalékosan. Az egyensúlyi keverékben változhatók valamelyik anyag mennyisége. Ha az egyensúlyban levő rendszer valamelyik anyagának koncentrációját megváltoztatjuk, akkor átalakulás után új egyensúly alakul ki. Ezt látványosan és gyorsan szimulálja a program.

A Millikan-kísérlet szimulációja

Az első osztályos kémiához, illetve a negyedikés fizikához kapcsolódik. A Millikan-kísérlet célja az elektron töltésének meghatározása. Töltéssel rendelkező olajcseppek a gravitáció következtében két elektróda között. A tanuló feladata, hogy a lemezek között olyan előjelű és akkora feszültséget létesítsen, hogy a csepp lebegjen. A feszültségérték szabályozása az olajcsepp felfelé, illetve lefelé irányuló elmozdulása alapján történik. A képernyőn a feszültség és az olajcsepp sebessége jelenik meg.

Lebegés esetén a „kísérleti” üzemmódból ki lehet lépni, de a szimulált kísérlet többször megismételhető. A tanuló feladata, hogy ha adott csepp tömeg mellett $1,6 \cdot 10^{16}$ V feszültség állítana meg egy -1 coulomb töltésű cseppet, akkor mondja meg, hogy hány coulomb volt az egyes cseppek töltése, és mennyi ezek legnagyobb közös osztója.

A program a választ értékeli, felhívja a figyelmet az esetleges előjeltevesztésre, vagy hibás érték esetén a számítási hibára, arra, hogy a tanuló esetleg kevés kísérletet végzett, végül kiírja az elektron töltésének helyes értékét.

A program az olajcsepp tömegét véletlen szimulálja, töltése csak az elemi töltés egész számú többszöröse lehet. A program ezekhez az értékekhez határozza meg a lebegéshez szükséges feszültséget, melyet a tanulónak az olajcsepp mozgása, illetve lebegése alapján kell meghatározni. A megadott adatot a számítógép összehasonlítja a tárolt, már kiszámított értékkel, egyezés esetén dicsért, egyébként felhívja a figyelmet a hibára.

A jövőben egyre nagyobb „tudású” és memóriájú számítógépek fognak az oktatásba kerülni. A finomgrafikával rendelkező gépek és a több szín használatával, a lemezgyeségek és mérőműszerek csatlakoztatásával megváltozik a számítógépek alkalmazási lehetősége, és oktatásban is emberközelibb programok készülhetnek.

SIEGLER GÁBOR

Osszuk el a 7-et 3-mal a számítógép segítségével, azaz adjuk ki a PRINT 7/3 parancsot! A képernyőn megjelenő eredmény: 2.33333. Ha most a következő parancsot adjuk a gépnek: PRINT 7/3-2.33333, akkor a képernyőn megjelenő

$3.09944E-06$ ($= 3.09944 \cdot 10^{-6} = 0.0000030944$) szám azt mutatja, hogy a számítógép által elvégzett művelet eredménye és a képernyőn megjelenő szám általában nem egyenlő egymással.

Nézzünk még néhány további, meglepőnek tűnő példát! Adjuk a következő parancsokat a számítógépnek:

```
A = 7 : B = 3 : PRINTA/B
      (az eredmény: 2.33333)
A# = 7 : B# = 3 : PRINTA# / B#
      (az eredmény: 2.333333333333333)
C# = 7/3 : PRINTC#
      (az eredmény: 2.33333253860474)
PRINT 100000/0.00001
      (az eredmény: 1E+10)
A = 100000/0.00001 : PRINTA
      (az eredmény: 1E+10)
A = 100000/0.00001 : PRINT A-1E+10
      (az eredmény: 1024)
A# = 100000/0.00001 : PRINTA#
      (az eredmény: 1000001024)
```

Ezek a számítások arra is rámutatnak, hogy a különböző típusú változók használata módosítja a képernyőn megjelenő eredményt. Az így szerzett tapasztalatok megértéséhez meg kell ismerkedni az iskolaszámítógép belső számábrázolásával. Az egyszerűs és a duplatpontos számábrázolást fogjuk vizsgálni.

Ehhez nagy segítséget nyújt a következő program:

```
10 CLS
20 A = 1 : A# = 1
30 I = VARPTR(A) : J = VARPTR(A#)
40 FOR L = 1-3 TO I+3 : PRINT PEEK(L) :
  : NEXTL : PRINT : PRINT
50 FOR L = J-3 TO J+7 : PRINT PEEK(L) :
  : NEXTL
60 GOTO 60
```

A 30-as sorban szereplő VARPTR függvény azt a memóriacímét adja meg, amelyiken az A (illetve az A#) változó értékének az ábrázolása kezdődik. A PEEK függvény segítségével pedig adott című memóriabájtok tartalmát tízes számrendszerbeli alakjukban tudjuk kiírni a képernyőre.

A programot futtassuk le! Az eredmény a következő lesz:

```
1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11.
4 08 65 00 00 129
8 08 65 00 00 00 00 00 129
```

Az A, illetve az A# változónév helyett más változónévvél (például B és B#, X és X#, A1

és A1#, CD és CD#, CD1 és CD1#) futtatva a programot, mindkét sorban csak a 2. és a 3. bájtok értékei változnak.

Ha a változónév egy karakterből áll, akkor a 2. bájtot értéke θ , a 3. bájtot értéke pedig a karakter ASCII-kódja. Ha a változónév két karakterből áll, akkor a 2. bájtot értéke a 2. karakter ASCII-kódja, a 3. bájtot értéke pedig az 1. karakteré. Innen is látható, hogy ha a változónév két karakterrel hosszabb, a számítógép akkor is csak az első két karaktert veszi figyelembe.

Változtassuk most az eredeti programban az A, illetve az A# változók tartalmát, és futtassuk így is a programot. Megfigyelhetjük, hogy az eddigi futtatások során az eredmény 1. oszlopában álló két szám nem változott. Ez azért van így, mert mindkét sorban az 1. bájtot értéke a változó típusát mutatja. Ez az érték egyszerűs pontosságú változó esetén 4, ami azt is jelenti, hogy a változó értékét a nevet követő 4 bájton ábrázolja.

Dupla pontosságú változó esetén az 1. bájtot értéke 8, és a változó értékét 8 bájton ábrázolja a gép. A továbbiakban az utóbb említett 4, illetve 8 bájtot jelentésével fogunk foglalkozni.

Futtassuk a következő programot:

```
10 CLS
20 INPUT A
30 I = VARPTR(A)
40 FOR L = I-3 TO I+3 : PRINT PEEK(L) :
  : NEXTL : PRINT : PRINT
50 GOTO 20
```

Adjuk az A változónak 2 különböző egész kitevőjű hatványértékeit nullától növekvő, majd csökkenő sorrendben (1,2,4,8,16,...,1,0.5,0.25,0.125,0.0625,... stb.). Látható, hogy ha a hatványkitevőt 1-gyel növeljük, akkor a 7. bájton 1-gyel nagyobb, ha a hatványkitevőt 1-gyel csökkentjük, akkor a 7. bájton 1-gyel kisebb szám jelenik meg. Ha most ugyanezeknek a számoknak a -1 -szeresét adjuk meg, akkor az eltérés az előző eredményeknél csupán annyi, hogy a 6. bájton megjelenik a $128 (= 2^7)$ érték.

Adjuk az A változónak a $2^n + 2^{n-1}$ (n egész) alakú értékeket (például: $2^1 + 2^0 = 3$, $2^2 + 2^1 = 6$, $2^3 + 2^2 = 10$, $2^4 + 2^3 = 14$, $2^5 + 2^4 = 18$, $2^6 + 2^5 = 22$, $2^7 + 2^6 = 26$, $2^8 + 2^7 = 30$, $2^9 + 2^8 = 34$, $2^{10} + 2^9 = 38$, $2^{11} + 2^{10} = 42$, $2^{12} + 2^{11} = 46$, $2^{13} + 2^{12} = 50$, $2^{14} + 2^{13} = 54$, $2^{15} + 2^{14} = 58$, $2^{16} + 2^{15} = 62$, $2^{17} + 2^{16} = 66$, $2^{18} + 2^{17} = 70$, $2^{19} + 2^{18} = 74$, $2^{20} + 2^{19} = 78$, $2^{21} + 2^{20} = 82$, $2^{22} + 2^{21} = 86$, $2^{23} + 2^{22} = 90$, $2^{24} + 2^{23} = 94$, $2^{25} + 2^{24} = 98$, $2^{26} + 2^{25} = 102$, $2^{27} + 2^{26} = 106$, $2^{28} + 2^{27} = 110$, $2^{29} + 2^{28} = 114$, $2^{30} + 2^{29} = 118$, $2^{31} + 2^{30} = 122$, $2^{32} + 2^{31} = 126$, $2^{33} + 2^{32} = 130$, $2^{34} + 2^{33} = 134$, $2^{35} + 2^{34} = 138$, $2^{36} + 2^{35} = 142$, $2^{37} + 2^{36} = 146$, $2^{38} + 2^{37} = 150$, $2^{39} + 2^{38} = 154$, $2^{40} + 2^{39} = 158$, $2^{41} + 2^{40} = 162$, $2^{42} + 2^{41} = 166$, $2^{43} + 2^{42} = 170$, $2^{44} + 2^{43} = 174$, $2^{45} + 2^{44} = 178$, $2^{46} + 2^{45} = 182$, $2^{47} + 2^{46} = 186$, $2^{48} + 2^{47} = 190$, $2^{49} + 2^{48} = 194$, $2^{50} + 2^{49} = 198$, $2^{51} + 2^{50} = 202$, $2^{52} + 2^{51} = 206$, $2^{53} + 2^{52} = 210$, $2^{54} + 2^{53} = 214$, $2^{55} + 2^{54} = 218$, $2^{56} + 2^{55} = 222$, $2^{57} + 2^{56} = 226$, $2^{58} + 2^{57} = 230$, $2^{59} + 2^{58} = 234$, $2^{60} + 2^{59} = 238$, $2^{61} + 2^{60} = 242$, $2^{62} + 2^{61} = 246$, $2^{63} + 2^{62} = 250$, $2^{64} + 2^{63} = 254$, $2^{65} + 2^{64} = 258$, $2^{66} + 2^{65} = 262$, $2^{67} + 2^{66} = 266$, $2^{68} + 2^{67} = 270$, $2^{69} + 2^{68} = 274$, $2^{70} + 2^{69} = 278$, $2^{71} + 2^{70} = 282$, $2^{72} + 2^{71} = 286$, $2^{73} + 2^{72} = 290$, $2^{74} + 2^{73} = 294$, $2^{75} + 2^{74} = 298$, $2^{76} + 2^{75} = 302$, $2^{77} + 2^{76} = 306$, $2^{78} + 2^{77} = 310$, $2^{79} + 2^{78} = 314$, $2^{80} + 2^{79} = 318$, $2^{81} + 2^{80} = 322$, $2^{82} + 2^{81} = 326$, $2^{83} + 2^{82} = 330$, $2^{84} + 2^{83} = 334$, $2^{85} + 2^{84} = 338$, $2^{86} + 2^{85} = 342$, $2^{87} + 2^{86} = 346$, $2^{88} + 2^{87} = 350$, $2^{89} + 2^{88} = 354$, $2^{90} + 2^{89} = 358$, $2^{91} + 2^{90} = 362$, $2^{92} + 2^{91} = 366$, $2^{93} + 2^{92} = 370$, $2^{94} + 2^{93} = 374$, $2^{95} + 2^{94} = 378$, $2^{96} + 2^{95} = 382$, $2^{97} + 2^{96} = 386$, $2^{98} + 2^{97} = 390$, $2^{99} + 2^{98} = 394$, $2^{100} + 2^{99} = 398$, $2^{101} + 2^{100} = 402$, $2^{102} + 2^{101} = 406$, $2^{103} + 2^{102} = 410$, $2^{104} + 2^{103} = 414$, $2^{105} + 2^{104} = 418$, $2^{106} + 2^{105} = 422$, $2^{107} + 2^{106} = 426$, $2^{108} + 2^{107} = 430$, $2^{109} + 2^{108} = 434$, $2^{110} + 2^{109} = 438$, $2^{111} + 2^{110} = 442$, $2^{112} + 2^{111} = 446$, $2^{113} + 2^{112} = 450$, $2^{114} + 2^{113} = 454$, $2^{115} + 2^{114} = 458$, $2^{116} + 2^{115} = 462$, $2^{117} + 2^{116} = 466$, $2^{118} + 2^{117} = 470$, $2^{119} + 2^{118} = 474$, $2^{120} + 2^{119} = 478$, $2^{121} + 2^{120} = 482$, $2^{122} + 2^{121} = 486$, $2^{123} + 2^{122} = 490$, $2^{124} + 2^{123} = 494$, $2^{125} + 2^{124} = 498$, $2^{126} + 2^{125} = 502$, $2^{127} + 2^{126} = 506$, $2^{128} + 2^{127} = 510$, $2^{129} + 2^{128} = 514$, $2^{130} + 2^{129} = 518$, $2^{131} + 2^{130} = 522$, $2^{132} + 2^{131} = 526$, $2^{133} + 2^{132} = 530$, $2^{134} + 2^{133} = 534$, $2^{135} + 2^{134} = 538$, $2^{136} + 2^{135} = 542$, $2^{137} + 2^{136} = 546$, $2^{138} + 2^{137} = 550$, $2^{139} + 2^{138} = 554$, $2^{140} + 2^{139} = 558$, $2^{141} + 2^{140} = 562$, $2^{142} + 2^{141} = 566$, $2^{143} + 2^{142} = 570$, $2^{144} + 2^{143} = 574$, $2^{145} + 2^{144} = 578$, $2^{146} + 2^{145} = 582$, $2^{147} + 2^{146} = 586$, $2^{148} + 2^{147} = 590$, $2^{149} + 2^{148} = 594$, $2^{150} + 2^{149} = 598$, $2^{151} + 2^{150} = 602$, $2^{152} + 2^{151} = 606$, $2^{153} + 2^{152} = 610$, $2^{154} + 2^{153} = 614$, $2^{155} + 2^{154} = 618$, $2^{156} + 2^{155} = 622$, $2^{157} + 2^{156} = 626$, $2^{158} + 2^{157} = 630$, $2^{159} + 2^{158} = 634$, $2^{160} + 2^{159} = 638$, $2^{161} + 2^{160} = 642$, $2^{162} + 2^{161} = 646$, $2^{163} + 2^{162} = 650$, $2^{164} + 2^{163} = 654$, $2^{165} + 2^{164} = 658$, $2^{166} + 2^{165} = 662$, $2^{167} + 2^{166} = 666$, $2^{168} + 2^{167} = 670$, $2^{169} + 2^{168} = 674$, $2^{170} + 2^{169} = 678$, $2^{171} + 2^{170} = 682$, $2^{172} + 2^{171} = 686$, $2^{173} + 2^{172} = 690$, $2^{174} + 2^{173} = 694$, $2^{175} + 2^{174} = 698$, $2^{176} + 2^{175} = 702$, $2^{177} + 2^{176} = 706$, $2^{178} + 2^{177} = 710$, $2^{179} + 2^{178} = 714$, $2^{180} + 2^{179} = 718$, $2^{181} + 2^{180} = 722$, $2^{182} + 2^{181} = 726$, $2^{183} + 2^{182} = 730$, $2^{184} + 2^{183} = 734$, $2^{185} + 2^{184} = 738$, $2^{186} + 2^{185} = 742$, $2^{187} + 2^{186} = 746$, $2^{188} + 2^{187} = 750$, $2^{189} + 2^{188} = 754$, $2^{190} + 2^{189} = 758$, $2^{191} + 2^{190} = 762$, $2^{192} + 2^{191} = 766$, $2^{193} + 2^{192} = 770$, $2^{194} + 2^{193} = 774$, $2^{195} + 2^{194} = 778$, $2^{196} + 2^{195} = 782$, $2^{197} + 2^{196} = 786$, $2^{198} + 2^{197} = 790$, $2^{199} + 2^{198} = 794$, $2^{200} + 2^{199} = 798$, $2^{201} + 2^{200} = 802$, $2^{202} + 2^{201} = 806$, $2^{203} + 2^{202} = 810$, $2^{204} + 2^{203} = 814$, $2^{205} + 2^{204} = 818$, $2^{206} + 2^{205} = 822$, $2^{207} + 2^{206} = 826$, $2^{208} + 2^{207} = 830$, $2^{209} + 2^{208} = 834$, $2^{210} + 2^{209} = 838$, $2^{211} + 2^{210} = 842$, $2^{212} + 2^{211} = 846$, $2^{213} + 2^{212} = 850$, $2^{214} + 2^{213} = 854$, $2^{215} + 2^{214} = 858$, $2^{216} + 2^{215} = 862$, $2^{217} + 2^{216} = 866$, $2^{218} + 2^{217} = 870$, $2^{219} + 2^{218} = 874$, $2^{220} + 2^{219} = 878$, $2^{221} + 2^{220} = 882$, $2^{222} + 2^{221} = 886$, $2^{223} + 2^{222} = 890$, $2^{224} + 2^{223} = 894$, $2^{225} + 2^{224} = 898$, $2^{226} + 2^{225} = 902$, $2^{227} + 2^{226} = 906$, $2^{228} + 2^{227} = 910$, $2^{229} + 2^{228} = 914$, $2^{230} + 2^{229} = 918$, $2^{231} + 2^{230} = 922$, $2^{232} + 2^{231} = 926$, $2^{233} + 2^{232} = 930$, $2^{234} + 2^{233} = 934$, $2^{235} + 2^{234} = 938$, $2^{236} + 2^{235} = 942$, $2^{237} + 2^{236} = 946$, $2^{238} + 2^{237} = 950$, $2^{239} + 2^{238} = 954$, $2^{240} + 2^{239} = 958$, $2^{241} + 2^{240} = 962$, $2^{242} + 2^{241} = 966$, $2^{243} + 2^{242} = 970$, $2^{244} + 2^{243} = 974$, $2^{245} + 2^{244} = 978$, $2^{246} + 2^{245} = 982$, $2^{247} + 2^{246} = 986$, $2^{248} + 2^{247} = 990$, $2^{249} + 2^{248} = 994$, $2^{250} + 2^{249} = 998$, $2^{251} + 2^{250} = 1002$, $2^{252} + 2^{251} = 1006$, $2^{253} + 2^{252} = 1010$, $2^{254} + 2^{253} = 1014$, $2^{255} + 2^{254} = 1018$, $2^{256} + 2^{255} = 1022$, $2^{257} + 2^{256} = 1026$, $2^{258} + 2^{257} = 1030$, $2^{259} + 2^{258} = 1034$, $2^{260} + 2^{259} = 1038$, $2^{261} + 2^{260} = 1042$, $2^{262} + 2^{261} = 1046$, $2^{263} + 2^{262} = 1050$, $2^{264} + 2^{263} = 1054$, $2^{265} + 2^{264} = 1058$, $2^{266} + 2^{265} = 1062$, $2^{267} + 2^{266} = 1066$, $2^{268} + 2^{267} = 1070$, $2^{269} + 2^{268} = 1074$, $2^{270} + 2^{269} = 1078$, $2^{271} + 2^{270} = 1082$, $2^{272} + 2^{271} = 1086$, $2^{273} + 2^{272} = 1090$, $2^{274} + 2^{273} = 1094$, $2^{275} + 2^{274} = 1098$, $2^{276} + 2^{275} = 1102$, $2^{277} + 2^{276} = 1106$, $2^{278} + 2^{277} = 1110$, $2^{279} + 2^{278} = 1114$, $2^{280} + 2^{279} = 1118$, $2^{281} + 2^{280} = 1122$, $2^{282} + 2^{281} = 1126$, $2^{283} + 2^{282} = 1130$, $2^{284} + 2^{283} = 1134$, $2^{285} + 2^{284} = 1138$, $2^{286} + 2^{285} = 1142$, $2^{287} + 2^{286} = 1146$, $2^{288} + 2^{287} = 1150$, $2^{289} + 2^{288} = 1154$, $2^{290} + 2^{289} = 1158$, $2^{291} + 2^{290} = 1162$, $2^{292} + 2^{291} = 1166$, $2^{293} + 2^{292} = 1170$, $2^{294} + 2^{293} = 1174$, $2^{295} + 2^{294} = 1178$, $2^{296} + 2^{295} = 1182$, $2^{297} + 2^{296} = 1186$, $2^{298} + 2^{297} = 1190$, $2^{299} + 2^{298} = 1194$, $2^{300} + 2^{299} = 1198$, $2^{301} + 2^{300} = 1202$, $2^{302} + 2^{301} = 1206$, $2^{303} + 2^{302} = 1210$, $2^{304} + 2^{303} = 1214$, $2^{305} + 2^{304} = 1218$, $2^{306} + 2^{305} = 1222$, $2^{307} + 2^{306} = 1226$, $2^{308} + 2^{307} = 1230$, $2^{309} + 2^{308} = 1234$, $2^{310} + 2^{309} = 1238$, $2^{311} + 2^{310} = 1242$, $2^{312} + 2^{311} = 1246$, $2^{313} + 2^{312} = 1250$, $2^{314} + 2^{313} = 1254$, $2^{315} + 2^{314} = 1258$, $2^{316} + 2^{315} = 1262$, $2^{317} + 2^{316} = 1266$, $2^{318} + 2^{317} = 1270$, $2^{319} + 2^{318} = 1274$, $2^{320} + 2^{319} = 1278$, $2^{321} + 2^{320} = 1282$, $2^{322} + 2^{321} = 1286$, $2^{323} + 2^{322} = 1290$, $2^{324} + 2^{323} = 1294$, $2^{325} + 2^{324} = 1298$, $2^{326} + 2^{325} = 1302$, $2^{327} + 2^{326} = 1306$, $2^{328} + 2^{327} = 1310$, $2^{329} + 2^{328} = 1314$, $2^{330} + 2^{329} = 1318$, $2^{331} + 2^{330} = 1322$, $2^{332} + 2^{331} = 1326$, $2^{333} + 2^{332} = 1330$, $2^{334} + 2^{333} = 1334$, $2^{335} + 2^{334} = 1338$, $2^{336} + 2^{335} = 1342$, $2^{337} + 2^{336} = 1346$, $2^{338} + 2^{337} = 1350$, $2^{339} + 2^{338} = 1354$, $2^{340} + 2^{339} = 1358$, $2^{341} + 2^{340} = 1362$, $2^{342} + 2^{341} = 1366$, $2^{343} + 2^{342} = 1370$, $2^{344} + 2^{343} = 1374$, $2^{345} + 2^{344} = 1378$, $2^{346} + 2^{345} = 1382$, $2^{347} + 2^{346} = 1386$, $2^{348} + 2^{347} = 1390$, $2^{349} + 2^{348} = 1394$, $2^{350} + 2^{349} = 1398$, $2^{351} + 2^{350} = 1402$, $2^{352} + 2^{351} = 1406$, $2^{353} + 2^{352} = 1410$, $2^{354} + 2^{353} = 1414$, $2^{355} + 2^{354} = 1418$, $2^{356} + 2^{355} = 1422$, $2^{357} + 2^{356} = 1426$, $2^{358} + 2^{357} = 1430$, $2^{359} + 2^{358} = 1434$, $2^{360} + 2^{359$

értéke az előbbinél 128-cal nagyobb lesz.

Megfigyeléseink magyarázata abban rejlik, hogy a HT-iskolaszámítógép az egyszerűes pontosságú valós típusú változók értékeit *kettes számrendszerben* ábrázolja. Ehhez a 0-tól különböző számokhoz a következő, ún. normálalakot használja:

$m \cdot 2^k$, ahol $1/2 \leq |m| < 1$, a k pedig egész szám. A k az ún. *karakterisztika*, értékét 128-cal megnövelve a 7. bájton láthatjuk. Az m mantissza 3 bájton (6., 5., 4.), (azaz $3 \cdot 8 = 24$ biten), tehát 24 kettes számrendszerbeli számjeggyel ábrázolja.

Az m mantisszára megadott feltétel miatt annak első kettesedje jegye mindig 1. Ezt használja fel a számítógép arra, hogy a szám előjelét jelölje. Pozitív szám esetében ezt a számjegyet 0-ra változtatja, negatív számnál pedig meg hagyja 1-esnek. (Természetesen ha a gép egy pozitív számmal műveletet végez, akkor ezt a bitet helyreállítja.) A θ számot úgy ábrázolja a számítógép, hogy a 7. bájta tartalma θ .

Tanulmányozzuk az elmondottakat a következő példákban:

$$3 = 11_2 = -0.11_2 \cdot 2^3$$

- a karakterisztika: 3

- a megnövelt karakterisztika: $131 = 3 + 128$

- a 7. bájta tartalma: $1000011_2 (= 131)$

- a mantissza: 0.11_2

- a kettesed pont utáni 24 számjegye:

$11000000\ 00000000\ 00000000$

- a szám pozitív, ezért az első kettesedjegy értékét 0-ra változtatjuk, és így

- a 6. bájta tartalma: $01000000_2 (= 64)$

- az 5. bájta tartalma: $00000000_2 (= 0)$

- a 4. bájta tartalma: $00000000_2 (= 0)$

$$-3 = -11_2 = -0.11_2 \cdot 2^3$$

- a karakterisztika ugyanaz, mint az előző példában: 3

- a 7. bájta tartalma nem változott

- a mantissza: -0.11_2

- a kettesed pont utáni 24 számjegye nem változott és a szám negatív, ezért

- a 6. bájta tartalma: $11000000_2 (= 192)$

- az 5. és a 4. bájta tartalma nem változott.

$$0.1 = 0.1100_2 \cdot 2^{-3}$$

- a karakterisztika: -3

- a megnövelt karakterisztika: $125 = -3 + 128$

- a 7. bájta tartalma: $01111100_2 (= 125)$

- a mantissza végtelen szakaszos kettesed tört: 0.1100_2

- a kettesed pont utáni 24 számjegye:

$11001100\ 11001100\ 11001100$

- a szám pozitív, ezért az első kettesedjegy értékét 0-ra állítjuk, továbbá, mivel a kettesed pont utáni 25. számjegy 1, ezért felfelé kerekítünk és a 24. számjegyet -1-re állítjuk, így

- a 6. bájta tartalma: $01001100 (= 76)$

- az 5. bájta tartalma: $11001100_2 (= 204)$

- a 4. bájta tartalma: $11001101_2 (= 205)$

$$-0.1 = -0.1100_2 \cdot 2^{-3}$$

- ez az előző szám -1-szerese, ezért annak ábrázolásától csak a 6. bájton különbözik,

- a 6. bájta tartalma: $11001100_2 (= 204)$

A dupla pontosságú számábrázolás az elmondottaktól csak annyiban különbözik, hogy a mantissza nem 3, hanem 7 bájton ($7 \cdot 8 = 56$ biten) van elhelyezve.

Itt szeretnénk felhívni a figyelmet a lap 1984. 3. számának 17. oldalán található 3. táblázatra.

Módsítsuk most az első program 20-as sorát a következőképpen:

```
20 A = 7/3 : A# = 7/3 : PRINT "A = ";
```

```
A, "A# = "; A# : PRINT
```

és futtassuk le. Az eredmény:

```
A = 2.333333 A# = 2.333333253860474
4 0 65 85 85 21 130
8 0 65 0 0 0 0 85 85 21 130
```

Végezzük el most a következő módosítást a 20-as sorban:

```
20 A = 7/3 : A# = 7#/3 : PRINT "A = "; A, "A# = "; A# : PRINT
```

és futtassuk le így is a programot. Az eredmény:

```
A = 2.333333 A# = 2.3333333333333333
4 0 65 85 85 21 130
8 0 65 85 85 85 85 85 85 21 130
```

Az első példában a számítógép a 7/3-ot mindkét esetben egyszeres pontossággal számolta ki. Ezt az értéket az A# változóba ugyanilyen pontossággal helyezte el, a mantissza további négy bájttal nullákkal töltötte fel. Az így ábrázolt szám tehát a

$$0.10010101\ 01010101\ 01010101_2 \cdot 2^2 = \left(\frac{1}{2} + \frac{1}{2^4} + \frac{1}{2^6} + \frac{1}{2^8} + \dots + \frac{1}{2^{24}}\right) \cdot 2^2 = \frac{9786709}{4194304}$$

Amikor az A és az A# változók értékét kiírtattuk, akkor a számítógép a fenti közelítő tört értékét 6, illetve 16 jegy pontossággal adta meg.

A második példában a 7-es után írt számjel (#) hatására a gép az osztást dupla pontossággal végezte el, és ezt a számítási eredményt helyezte el az A# változóba. Ekkor tehát a következő számot ábrázolta:

$$0.10010101\ 01010101\ 01010101\ 01010101\ 01010101\ 01010101_2 \cdot 2^2 = \left(\frac{1}{2} + \frac{1}{2^4} + \frac{1}{2^6} + \frac{1}{2^8} + \dots + \frac{1}{2^{56}}\right) \cdot 2^2 = \frac{42033596522124629}{18014398509481984}$$

Ez a tört pedig a 7/3-ot már legalább 16 jegy pontossággal közelíti, és ezt a közelítő értéket láthatjuk a képernyőn.

Felsorolunk néhány további példát, ahogy a 7/3 közelítő értékét 16 jegy pontossággal megkapjuk:

```
A# = 7#/3 : PRINT A#
PRINT 7#/3
PRINT 7D0/3
A# = 0.7# : A# = A#/0.3#
A# = 0.7D0 : A# = A#/0.3#
```

Az elmondottak alapján már megmagyarázhatók a bevezetőben említett példák, és további érdekes és értékes tapasztalatok is nyerhetők.

GYAPJAS FERENC-MAJOR ZOLTÁN

HT-1080Z

GÉPI KÓDÚ KIEGÉSZÍTÉSEK BASIC-HEZ

Ismert dolog, hogy a SYSTEM/12288 bővítés meghívása után, ha BASIC program futása közben lenyomunk valamilyen billentyűt, a programfutás kis időre, kb. fél másodpercre megáll, ami például játékprogramok esetén nagyon kellemetlen.

Ez ellen többféleképpen lehet védekezni:

- Nem hívjuk meg a bővítést, vagy ha már megvan, akkor a gépet kikapcsoljuk.
- Ha mégis szükség van a kibetűkre, SYSTEM/12294 begépelésével hívjuk a bővítést. Ekkor csak kibetűk lesznek.
- Mivel nem tarthatjuk számon minden programunkról, hogy bővítéssel jő-e vagy anélkül, az ilyen programok elején meghívjuk azt a gépi rutint, amely a megfelelő rendszerváltozók áttöltésével „megszabadít” minket a bővítéstől.

A gépi rutin assembler nyelven a következő:

```
LD HL,06D2H
LD DE,4000H
LD BC,0036H
LDIR
RET
```

Ez a kis rutin legegyszerűbb egy REM sorban tárolni, ugyanis a program kimenetekkor így magával viszi a gépi rutint is.

A REM sor sorszáma 0 legyen, így kisebb a valószínűsége, hogy véletlenül elérünk más sort. Ez azért fontos, mert a REM sor címének állandónak kell lennie.

A REM után írjunk be 14 db szökőzt, ennek helyére fog kerülni a gépi rutinunk.

Ezután hívjuk meg a monitort SYSTEM/12710 beírásával, majd írjuk: M42EE, azaz a 42EEH címre fogunk beírni. Ez az első BASIC program sor első utasítása utáni első karakter címe. Most a következő adatokat írjuk be (NEW LINE nem kell):

```
21 D2 06 11 01 40 1D 01 36 01 05
ED B0 C9
```

X billentyűvel kilépünk az M üzemmódból, B-vel a monitorból. Írjuk be a következő programot:

```
1 POKE 16526,238:POKE 16527,66:
A = USR(0)
```

Ezzel kész is van. Bármely BASIC program, amely így kezdődik, RUN-nal indítva kikapcsolja a bővítést.

UHERKOVICH PÉTER

Kedves Olvasónk!

Kéziratokat (amiket nem rendeltünk meg) nem örzünk meg és nem küldünk vissza. Levelek, cikkek stb. közlése esetén szerkesztőségünk fenntartja magának a jogot az írásk rövidítésére.

Két év telt el azóta, hogy az iskolaszámítógép-program keretében a középiskolákba megérkeztek az első számítógépek. Nem egyforma fogadtatásban részesültek, és nem is ugyanazt a hatást váltották ki minden iskolában. Nem tölthetik be a számítógépek a nekik szánt szerepet ott, ahol hét lakat alatt elzárva őrzik őket, és csak szigorúan meghatározott időben lehet hozzájuk férni. Annál nagyobb érdeklődést ébreszthetnek ellenben a számítástechnika iránt azokban az iskolákban, ahol tanár és diák bármikor leülhet melléjük, ahol őrztől iránnyítás mellett ismerkedhetnek és dolgozhatnak velük a tanulók. Ilyen körülmények között a számítógép körül általában felpeszűl az élet, nagy a lelkesedés, és sorra születnek a szebbnél szebb eredmények. Mint például abban a két iskolában, amelyről a következő beszámolókat szólnak.

Ahol élnek a lehetőségekkel

BUDAPESTEN

Iskolánkban, az Ihsás Dániel Ipari Szakközépiskolában az 1982/83-as tanév végén jelent meg az első személyi számítógép. A következő tanévben két szakkörben ismerkedtek a tanulók a számítógéppel és a BASIC nyelvvel, és hamarosan ontani kezdték az oktatást segítő kisebb-nagyobb programokat.

A diákok lelkesedésére jellemző, hogy a szakköri foglalkozásokon a második-harmadik alkalom után a játékgépprogramok már elő sem kerültek. Nem az érdeklődés felkeltése, hanem az okozott gondot, hogy a szakkörösök megtalálják a helyes arányt a tanulás és a programozás között. Néhányan legszívesebben még éjjel is az iskolában maradtak volna a számítógép mellett.

A tanárok megnyerésére bemutató szakórát rendeztünk, amely a számítógép tanórái alkalmazhatóságát bizonyította. Két tanulónk nyári számítógépes táborban vett részt, ahol a gépi programozás alapjait ismerték meg.

Az 1984/85-ös tanévben az iskolai számítástechnikai munka kiszélesedett. Bekapcsolódott ugyanis egy BASIC alapanyagot végzett kolléga, és az elmúlt évben legeredményesebben szereplő két harmadik évfolyamos tanuló is szakkört kezdett vezetni. Így az idén már négy szakkör működik. Bebizonyosodott, hogy a lelkesedés nem szalmaláng volt, nem az újdonság varázsának szólt.

Felvettük a kapcsolatot bázisvállalatunk, a Hungarocamion szervezési osztályával, ahol Commodore 64-es számítógépeken tartják nyilván a járműállomány futási-javitási adatait. Munkájukat programokkal kívánjuk segíteni, azzal a nem titkolt vágygal, hogy előbb-utóbb kapunk tőlük egy gépet használatra.

Új színtöltöt jelentett a tanulói újságot megmozgató tanulmányi versenyek sorában a programpályázat, amelyet hazánk felszabadulásának 40. évfordulója tiszteletére írtunk ki.

Novemberben házi alapfokú programozási tanfolyamot indítottunk iskolánk tanárai és a gazdasági osztály dolgozói számára.

Továbbra is legnagyobb problémánk, hogy a diákok és tanárok nagyon szűkre szabott időben dolgozhatnak, mert jelenleg is csak az induláskor kapott, egyetlen számítógép áll rendelkezésünkre.

MÓCZÁR LÁSZLÓ

DEBRECENBEN

A Péchy Mihály Építőipari Szakközépiskolában szerencsésen találkozott a vezetés támogatása, az új iránt fogékony tanári kezdeményező-készség és a tanár-diák jó munkakapcsolata, amikor 1983 májusában megkapták az első számítógépet a TII központi osztásából. Itt már voltak előzmények: az iskolában korábban is foglalkoztak számítástechnikával, igaz, csak programozható zsebszámológéppel, a PTK 1050-nel. Ezek a kis gépek nemcsak szakköri keretek között működtek, hanem a matematikaórák hasznos eszközei voltak, és ma is azok.

A HT-1080Z megjelenése nem jelentett technikai csodát, természetes volt a továbblépés. Az újszülött nagy becsbe került. Külön helyiséget kapott, és tanárok, diákok időkorlátozás nélkül foglalkozhattak vele. A tanév végére több alkalommal bemutatókat szerveztek, lehetett játszani, ismerkedni a programozás rejtelmeivel. Egész nyáron nyitva volt az iskola kapuja.

A PTK-szakkörökből és az új érdeklődőkből alakult meg az első számítástechnikai szakkör. Hetente két órát foglalkoztak elméleti kérdésekkel a gyerekek, és 3-4 fős csoportokban oldották meg a szakköri feladatokat. Nyolc foglalkozás elég volt ahhoz, hogy megtanulják kezelni a gépet, egyszerű grafikai programokat írjanak.

1983 novemberében két részre osztott a szakkör. A jobbkból haladó csoport alakult, ök ma már a gépi kódot tanulják. De a lemaradók

sem veszítették el a kedvüket, ök is lelkes géphasználók.

1984 tavaszán újabb kezdő szakkört indítottak. Megvolt a technikai bázis is, az iskola két új HT-gépet kapott. Nem érdemtelenül, mert itt az átlagossal többet futott a gép, éltek minden lehetőséggel: verseny, pályázat, KISZ játékcsoport, számítógépes bemutató óra fizikából. A tavaszi BNV-n is szerepeltek: önálló programjaitkat mutatták be a tanulók.

A szakkörök vezetője Szálodabágyi Zsigmond tanár. Az ő irányításával jutottak el oda, hogy ma már a haladó szakkör önképzőkörként dolgozik. Valamennyi programjuk, a leggyeszebbtől a bonyolult építészeti tudást igénylőig, mind önálló munka.

A szakkör tagjai közül nyáron hatan a veleneci, ketten a hajdúszoboszlói számítástechnikai táborban voltak.

Esenként előfordult, hogy szülők, kollégák felvetették: nem terheli-e túl a gyerekeket a sok számítógépezés? Ez az aggodalom azonban alaptalan volt. Várakozáson felüli tanulmányi és érettségi eredmények igazolták, hogy az értelmes, komoly szellemi munka kedvezően hat a tanulmányi eredményekre.

Az 1984/85-ös tanév új feladatot és új lehetőséget hozott, ami nem kis örömet jelent. A TII jóvoltából – mintegy elismerésként a figyelemreméltó eredményekért – egy Commodore 64 típusú gépet kapott az iskola.

Okosan élnek a lehetőségekkel. Jól hasznosítják a külső segítséget, érkezzen az a tanügyi igazgatástól vagy a Megyei Pedagógus Továbbképzési Intézetétől. Készülnek a számítástechnika tanterv szerinti oktatásának bevezetésére, felméri diákjaik majdani munkahelyeinek számítástechnikai gyakorlatát, hogy azzal is megismertessék tanulóikat.

GALGÓCZY IMRE

Kedves Olvasónk!

Szives tájékoztatására közöljük a Lapkiadó Vállalat azon budapesti és vidéki üzleteinek címét, ahol lapunkat, a Mikroszámítógép Magazinint megvásárolhatja:

BUDAPEST

- VII., Almásy tér 11.
- VII., Dohány u. 82.
- VII., Dohány u. 84.
- IX., Üllői út 119.
- XII., Bószörményi út 3/a.
- XIII., Rajk László u. 24.
- XIV., Thököly út 21.

VIDÉK

- Bükkfürdő, pavilonsor
- Debrecen, Simonfi u. 1.
- Győr, Arpad út 63.
- Köszeg, Jurisics tér 10.
- Sopron, Orsolya tér, Lábasház
- Szeged, Bajcsy-Zsilinszky u. 19.
- Szombathely, Berzsenyi Dániel tér 3.
- Pécs, Jókai u. 25.

FFGT

FLOPPYMAT FEJLESZTŐ GAZDASÁGI TÁRSASÁG

VÁLLALATOK, SZÖVETKEZETEK, INTÉZMÉNYEK RÉSZÉRE AJANLJA

- alkalmazásban levő, kipróbált mikroszámítógépes adatfeldolgozó rendszereinek széles választékát,
- újabb - a választékban nem szereplő - rendszerek kifejlesztését, bevezetését,
- tanácsadást meglévő ügyvitel korszerűsítésére, a hatékonyság növelésére

NÉHÁNY BEVÁLT RENDSZERÜNK

- SZÖVEGFELDOLGOZÁS
- BOLTI ELSZÁMOLTATÁS
- TERMELÉS ÜGYVITEL
- TÖBBRAKTÁRAS KÉSZLET-NYILVÁNTARTÁS, -ELSZÁMOLÁS SZOLGÁLTATÁSAINK



- ALACSONY ÁRFEKVÉS
- AZONNALI SZÁLLÍTÁS
- BEVEZETÉS, BETANÍTÁS
- SZÉLES KÖRŰ REFERENCIA
- KÍVÁNSÁGRA HELYSZÍNI BE-MUTATÓK TARTÁSA

Tagvállalatok:

ECONORG
SZÖVORG
VILATI

Felvilágosítással szolgál:

Krepler Károly	127-628
Sólyom József	189-162
Szabó József	327-748

AZ ALKALMAZOTT GÉPTÍPUS

- (a Floppymat gépcsalád) a kategóriájában kitűnik:
- kiemelkedő adatfeldolgozási sebességével,
 - igen rugalmasan változtatható konfigurációjával,
 - viszonylagos olcsóságával (célgép már 245 eFt-tól)



COMPUTER-M

Újabb színfolttal gazdagodott a számítástechnikai kiskereskedelem. COMPUTER-M néven megnyitotta első ügyfélszolgálati irodáját a KSH Számítástechnikai és Ügyvitelszervező Vállalat Budapesten, a VI. kerület, Lenin krt. 57-59. szám alatt. Az iroda patinás környezetben kínálja a legmodernebb technika eredményeit – a kisiskolástól kezdve a professzionális alkalmazóig, mindenkinek.

A nyitás előtt néhány nappal beszélgettünk dr. Kondricz Józseffel, a SZÜV vezérigazgatójával.

– Mít írnának képzeletbeli cégtáblájukra, ha az iroda megnyitásának motivációját kellene megfogalmazni?

Az elsődleges cél a számítástechnika-alkalmazás támogatása, fórum teremtése a számítástechnika megfelelő színvonalú terjesztésére, kiskereskedelmi hálózatunk megalapozása. Egyrészt lakossági szolgáltatás, másrészt előretolt helyőrségünk üzleti partnereink számára.

– Mít kínálnak a lakosság, a magánember részére?

Lakossági szolgáltatásaink a mikroszámítógépekhez és a tájékoztatáshoz kötődnek. Az irodában 6 mikrogepes munkahelyet alakítottunk ki, egyrészt professzionális, másrészt személyi számítógépek használatára. 2 db MO8X, 2 db TAP34 és 2 db Commodore 64 számítógéprendszer áll az érdeklődők és fejlesztők rendelkezésére. Az abszolút kezdők szakembereink segítségével ismerkedhetnek meg ezekkel a gépekkel. Akik saját fejlesztése akarják használni, díjazás ellenében vehetik igénybe az iroda gépparkját. A professzionális mikrogepek bérleti díja egy órára 170 forint, a Commodore 64 gépé 100 forint. A diákok kedvezményt élveznek.

– Milyen szolgáltatásokat tartalmaz ez a bérleti díj?

Természetesen a szükséges alapszoftverek használatát, például Commodore gép esetében a CP/M modul is. Ezenkívül kézikönyvek, dokumentáció használatát és tanácsadást. A fejlesztő használhatja saját adathordozóját, vagy a helyszínen vásárolhat. A hajlékonylemezek ára típusolt függ. A legáltalánosabb ár 10 darabos csomagoként 1200 forint körül van.

És ide tartozik egy másik jó hír is a mikrogepes fejlesztőknek. A mikronyomtatóba való leprellő is kapható: 1000 lap ára 470 forint. A leprellőt csak korlátozott mennyiségben kívánjuk itt árulni, kifejezetten az egyéni fejlesztők számára.

A szakkönyvek, kézikönyvek is megvásárolhatók. Mintegy 50féle szakkönyv kapható szervezési, programozási témakörben a Műszaki Könyvkiadó, az LSI, a Statisztikai Kiadó és a SZÁMALK könyvkiadatlából. Kaphatók a számítástechnikával foglalkozó újságok, magazinok is.

– Mít az, amiért a professzionális felhasználónak vagy a számítástechnikát alkalmazni kívánó vállalatnak érdemes ide betérni?

Ingényszeri szaktanácsadást adunk a megvalósítandó feladatokhoz illeszkedő gépkiválasztás kérdésében, amennyiben ehhez az üzletben történő konzultáció elegendő. Ezen túl vállalkozó vállalat átvilágítását is, megfelelő díjazásért. Az iroda mögött áll a SZÜV több száz fős szervezőszakember-kapacitása. Irodánkban MO8X és TAP34 típusú mikrogepek vásárolhatók minta után, bevizsgált alap- és felhasználói szoftverekkel együtt. Géptípusonként 10–10 alkalmasú szoftvert kínálunk.

Nemcsak a SZÜV szellemi kapacitása,

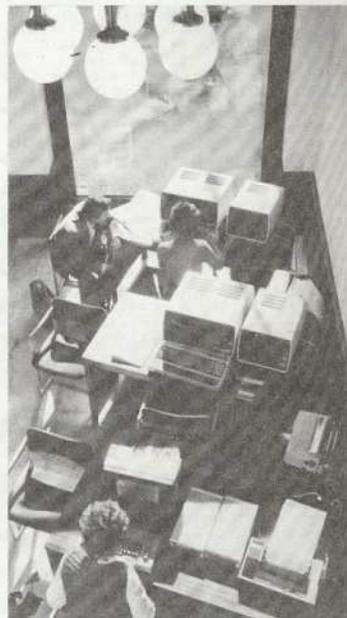
hanem nagyszámítógépes eszközbázisa is a felhasználó rendelkezésére áll új irodánkban keresztül. A helyszínen vállaljuk a Commodore 64-gyel előállított adatállományok alkalmassá tételét nagyszámítógépes rendszerbe történő bevitelre, mágnesszalagra történő konvertálás útján. Terveink között szerepel az iroda adatátviteli vonalon való összekapcsolása a SZÜV budapesti számítógéppontjával, ahol R35, IBM és Siemens gépek fogadják a bemenő adatokat.

– Milyen szolgáltatást várhatnak az irodától a konkurrens számítástechnikai vállalatok?

Mint említettem, alapvető feladatunknak tartjuk magának a számítástechnika-alkalmazásnak fórumot teremteni. Bármely vállalat termékeit kérésre bemutatjuk, szolgáltatásait terjesztjük. Ebből a szempontból irodánk bemutatóterem is.

– Tud-e valami érdekeset mondani az iroda működéséről, terveiről?

Legfőbb erősségünknek tartjuk az iroda 7



fős személyi állományát, amely kizárólag magasan kvalifikált, nagy szakmai tapasztalattal rendelkező mérnökökből, rendszer-szervező közgazdászokból és informatikai szakemberekből áll. Szakmánkhöz kapcsolódik, hogy a kezdeti tapasztalatok után az iroda teljes adminisztrációját számítógépen kívánjuk elvégezni. Erre annál is inkább szükség van, mert még az idén két újabb iroda kezd meg működését Kecskeméten és Szombathelyen, a következő években pedig a SZÜV minden vidéki egységéhez szeretnénk kialakítani ezt a fórumot.

ZÁRDA SÁRA

Még egyszer az APL-ről

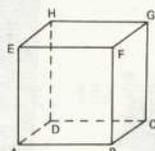
Nagy örömmel olvastam az 1984/6. számban Lovrics László MELEDA az APL-ben című cikkét. Jól mutatja be, hogy az APL mennyire áttekinthető, a részleteket eltakaró, csak a lényegre törő programokat eredményezhet, amelyek megfogalmazása és megírása nem kíván nagy szellemi koncentrációt.

Ugyanakkor fájjalom a szerzőnek azt a megállapítást, hogy „APL-lel könnyű áttekinthetetlen és jöszerve módosíthatatlan programokat írni...”. Ez akkor igaz, ha már a programtervezés során olvashatatlanná (erről is kellene vitáznunk!) és pornográfikus programok írását tűztük ki célul.

Az ellenkező irányzat bemutatására közlöm a Számítástechnika című lap 1984. 1-2. számában megjelent feladatot (Hack Frigyes, ELTE):

23. számú feladvány

Az ábrán látható kocka drótból készült élhálózatán egy hangya mozog egyenletesen. Egy élen egy másodperc alatt megy végig. Az A csúsból indul el, és az átlélessen G csúcsban elhelyezett táplálékot keresi. A kocka-



```
V KOCKA9 N
[C1] T←(I+0),0pU+ 7 3 p 2 4 6 , 1 3 7
    2 4 8 , 1 3 5 , 4 6 8 , 1 5 7 , 2
[C2] X←(N←I+T+R+1)/Z
[C3] Y←(8N2+1R+R,UC2+1R:233)/Y
[C4] 'ABCDHEFG'ERJ
[C5] *%,0pT+T,~1+pR
[C6] Z:A,C0.5J F←+(A+1+2×L0.5+T/T)+.T
[C7] 'ATLAG=' ,r(+/T)+N
[C8] ((pA),3)p 3 0 rA), ' , 'D 'C1+F+.
V
```

1. program

```
V KOCKA N
[C1] a Ez a program a 'Számítástechnika' 23. sz.
[C2] a feladványának (1984. 1-2) megoldása
[C3] a
[C4] a Eloszor megszámoljuk a kocka sarkait az alábbi meg-
[C5] a feleltés szerint, mert így a megoldás hajszalni-
[C6] a val könnyebb!
[C7] a A B C D E F G H
[C8] a 1 2 3 4 6 7 8 5
[C9] a Tehát a celsarok, a 'G' kapja a legnagyobb számot.
[C10] a
[C11] a Most elkészítjük a lehetséges utak 7x3-as matrixát,
[C12] a ahol az n-edik sorban található számok
[C13] a azt mutatják, hogy az n-edik sarokból
[C14] a melyik sarokba lehet eljutni:
[C15] a UT←7 3 p 2 4 6 , 1 3 7 , 2 4 8 , 1 3 5 , 4 6 8 , 1 5 7 , 2 6 8
[C16] a A 'T' idők gyűjtésére szolgáló TGY gyűjtőt
[C17] a urrese tesszük:
[C18] a TGY←I 0
[C19] a IN számolja a már elvégzett kísérleteket:
[C20] a IN←0
[C21] a A program argumentuma, N, adja meg az elvégzendő
[C22] a kísérletek számát:
[C23] a CIKN←+(N←IN+IN+1)/NVEG
[C24] a Mindig az egyes (A) sarokból indulunk,
[C25] a R fogja tartalmazni a megtett utat:
[C26] a R←1
[C27] a Itt vegezzük el az új ut generalasát, és hozzákap-
[C28] a csoljuk az eddig megtett uthoz. A generalasnal
[C29] a az R utolsó eleme által meghatározott sorban
[C30] a lévő számok közül véletlenszerűen választunk egyet:
[C31] a CIKI:R+R,UTC2+1R;23J
[C32] a Ha az utóljára kapott sarok nem a 8-as (G),
[C33] a akkor folytatjuk az 'utkeresést' :
[C34] a +(8N2+1R)/CIKI
[C35] a Kiírattjuk a megtalált utat:
[C36] a 'ABCDHEFG'ERJ
[C37] a A kísérlet során kapott T idő egyvel kevesebb,
[C38] a mint az utban lévő csomópontok száma; T új értéket
[C39] a hozzafuzzuk TGY-hez:
[C40] a TGY←TGY,~1+pR
[C41] a Újabb kísérletet kezdünk el:
[C42] a +CIKN
[C43] a Megkeressük a legnagyobb időt:
[C44] a NVEG:MAX+T/TGY
[C45] a Meghatározzuk a lehetséges mozgási idők
[C46] a gyakoriságát, FT-t (az összes lehetséges időt
[C47] a 'A' tartalmazza):
[C48] a FT←+(A+2×MAX-2)+.TGY
[C49] a Kiírattjuk a gyakorisági táblázatot:
[C50] a A,C0.5J FT
[C51] a Kiírattjuk a mozgáshoz felhasznált idő
[C52] a átlagos értéket:
[C53] a 'ATLAG=' ,r(+/TGY)+N
[C54] a A gyakoriságokról - grafikon helyett többet mutató
[C55] a hisztogramot készítünk, megfelelően formázva:
[C56] a ((pA),3)p 3 0 rA), ' , 'D 'C1+FT+.CI[FT]
[C57] a
[C58] a A program IBM 5110 számítógépen, APL nyelven készült,
[C59] a tervezési,írasi és tesztelési idő: 36 perc
[C60] a Készítette: Kertész Adam, Bp. XI. Vahot u. 6
V
```

2. program

```
KOCKA_30
ADHDADHEHEHG
ADADCG
AEABCG
ADCG
ABDFFEAEHG
AEHG
ADDCBFEFFBFCG
AEADCG
AEHDHEAEFBAEHEFG
ADHG
ABAEHEHDCG
ADCDAEADABFG
ADAHEFEFCBFBFEAEHAEAEHAEHAEHDCG
ABCDHAEHHDHABCG
AEHEHDBAEHG
ADCDHG
AEFRABCFBG
ABADHDCBFBADHABCBFG
ADCDAEAEFBFBFEHAEABEAEFFG
AEADHEHDCG
AEHG
ADDCBFEFFBFCG
ABCBCBAEADHDAEAEABFG
ABAEADCDAEFEABFG
AEFFFEARADCG
ABCDHG
ADADCDABFBFCBDCBDCG
ABCBAEFG
AEFFG
AEHEFG
3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32 33 34 35
5 0 6 0 1 0 4 0 4 0 2
0 2 0 1 0 2 0 1 0 1 0
0 0 0 0 0 0 0 0 0 0 1
ATLAG= 10.733
3 00000
4
5 000000
6
7 0
8
9 0000
10
11 0000
12
13 00
14
15 00
16
17 0
18
19 00
20
21 0
22
23 0
24
25
26
27
28
29
30
31
32
33
34
35 0
```

3. program

LOKÁLIS VÁLTOZÓK BASIC PROGRAMOKBAN

```

KOCKA9 30
ADHDADHEHEHG
ADADCG
AEABCG
ADCG
ABFFFEAHEHG
AEHG
ADDCDFEFBFBCG
AEADCG
AEHDHEARFBAEHEFG
ADHG
ABAEHEHDCG
ADCAEADABFG
ADAEHEFFCBFBFEAEFBAEAEHEARFEHEHDCG
ARCDHDAEHHDABCG
AEHEHDAEAEHG
ADCDHG
AEFHARCBFG
ARADHDCEFBADHDABCFBG
ADCDHDAEAFBFEHEAEAEFEFG
AEADHEHDCG
AEHG
ADDCDFEFBFBCG
ABCBCBARAEADHDAEAEABFG
ABAEAEADCAEAEABFG
AEFRFEABDCG
ABCDCG
ADADCDAEFCDCBCBCDCG
ABCBAEFG
AEFG
AEHEFG
 3 5 7 9 11 13 15 17 19 21 23
25 27 29 31 33 35
 5 6 1 4 4 2 2 1 2 1 1
 0 0 0 0 0 1
ATLAG= 10.733
 3 00000
 5 000000
 7 0
 9 0000
11 0000
13 00
15 00
17 00
19 00
21 0
23 0
25
27
29
31
33
35 0
    
```

A mikrogépes BASIC nyelvek gyakran csak globális változónevek használatát teszik lehetővé, ráadásul ezek szintakszisa (legfeljebb két- vagy háromszóval) nem sok lehetőséget ad a változók olyan megnevezésére, ami jól emlékeztetne funkciójukra. Nagyobb programok írása és még inkább módosítása során ez könnyen arra vezethet, hogy egy-egy változót a program különböző szubrutinjában inkonzisztens módon használjunk. Az alábbi egyszerű konvenció segíthet az ilyen hibák elkerülését.

1. A program minden egyes szubrutinjához egy-egy szintszámot rendelünk a következőképpen:

- 0 szintűek azok a szubrutinok, amelyek nem hívnak más szubrutinokat;
- n szintűek azok a szubrutinok, amelyek legfeljebb n-1 szintű szubrutinokat hívnak.

Tegyük fel például, hogy egy program szubrutinszerkezetét az ábrán látható sémmával lehet ábrázolni. Az ábra szerint SB nem hív szubrutint, így 0 szintű, SA szintje 1, az FPR főprogram pedig 2.

Az így kialakított szintek száma egy-egy programban egy jól meghatározott N érték (valószínűleg 10-nél kisebb). A főprogram szintje a legmagasabb. (Az egyes szubrutinok szintszámait ajánlatos a szubrutint bevezető kommentárban is rögzíteni!)

2. A változónevekre vonatkozó megállapodás legyen a következő: a 0 szintű szubrutinok lokális változói kezdődjenek például I betűvel, a következő szintű J-vel, azután K-vel, ... stb.

3. A szubrutinok hívási-visszatérési paramétereit nem kezdődhetnek az előbbi betűkkel, hiszen éppen ez teszi lehetővé, hogy egy-egy szubrutin egyetlen is rontsa el egy másik szubrutin helyes működését.

Ez a konvenció jól alkalmazható minden olyan nyelvben, amely csak globális változó-

neveket támogat. Természetesen ára is van: csak akkor vezethető be, ha programunk szubrutinszerkezetét alaposan át gondoltuk, megtervezük. Ez viszont – függetlenül a változónevek megválasztásától – mindenképpen célszerű!

Előfordulhat olyan programmódosítás, amelynek során egy S1 szubrutinba újabb szubrutinhívást kell beiktatnunk. Ha az újabban hívott S2 szubrutin n2 szintje a hívó n1 szintjénél nem kisebb, akkor programunk szintszámrendszere megváltozna, hiszen az S1 és az őt közvetlenül vagy közvetve hívó összes szubrutin szintje megemelkedne.

A szintmódosító szubrutinhívást úgy kell végrehajtani, mintha a hívó S1 szubrutin szintje magasabb lenne a hívotténál, ugyanakkor különbözne a már meglévő magasabb szintek mindegyikétől. Nevezük ezt a szintet az $n2+1 = n3$ szint egy alszintjének. Az S1 szubrutin lokális változóit az S2 hívása előtt el kell menteni az n3 szint megfelelő alszintjére. S2-ből visszatérve pedig vissza kell állítani az elmentett értékeket. Az alszinteken futó szubrutinok változóinak neveit – megint csak önkényesen – kezdhetjük például P, Q, R ... betűkkel.

Az ilyenfajta programmódosítás meglehetősen körültekintést igényel, hiszen előfordulhat, hogy az S2 hív olyan n1 szintű S1 szubrutint, amelybe egyszer egy szintmódosító S2' szubrutinhívást kell beiktatnunk, így S1 és S1' elronthatják egymás elmentett lokális változóit.

Nyilvánvaló, hogy a változót névkonvenció nem helyettesíti a valódi lokális-globális szimbólumkezelést, de segítheti könnyebben módosítható, megbízhatóbb programok írását.

DEBRECZENI ISTVÁN

4. program

háló bármelyik csúcsából egyenlő valószínűséggel indul el az adott csúcshoz esatlakozó bármelyik élen, tehát akár visszafelé is, oda, ahonnan érkezett. Kivétel a G csúcs, melyet elérve 0 valószínűséggel megy tovább, tehát megáll. Nyilván az A csúcsból leghamarabb 3 sec alatt éri el a G csúcsot. Lehetséges azonban, hogy nem a legrövidebb utat választja, sőt az esetleges visszafordulások miatt is veszít időt.

Készítsünk programot, mely

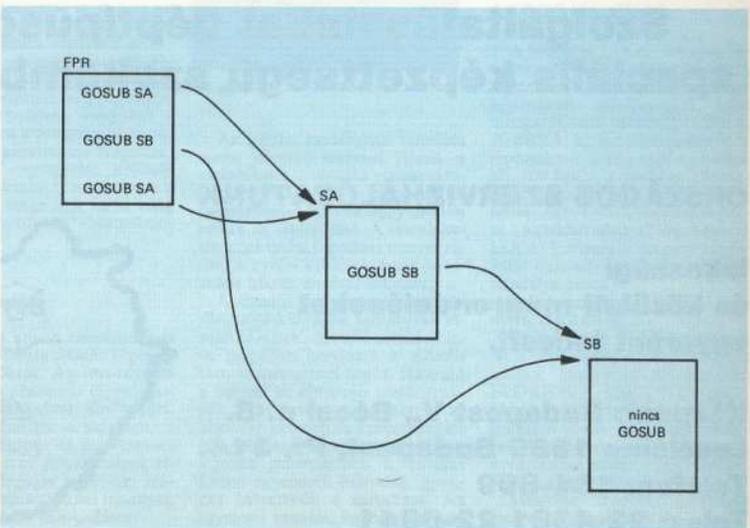
1. szimulálja a hangya bolyongását a kocka élhálózatán, és meghatározza a célbeéréshez szükséges T időt, továbbá szimulálja az A-ból G-be tartó mozgás N számú kísérletét, és meghatározza a lehetséges mozgási idők előfordulási gyakoriságát, $F(T)$ -t, ahol $T=3, 4, \dots$. A program inputja a kísérletek száma;
2. a képernyőre kirajzolja az $F(T)$ gyakoriságok grafikonját, és meghatározza a mozgáshoz felhasznált idő átlagos értékét az N számú kísérlet alapján.

A feladvány részleges, és a feladat kitűzését nem mindenben kielégítő, BASIC nyelvű megoldását a Számítástechnika 1984. áprilisi számában találhatja meg az olvasó.

Bemutatom az én megoldásomat, az olvasható (1. program) és az ebből származó „olvashatatlan” (?) (2. program) programokat, valamint ezek futásának eredményeit (3. és 4. program).

A KOCKA és KOCKA9 programok futásánál a véletlenszám-generátor kezdőértékét azonosra állítottam meg, ezért egyezik meg egymással a két eredmény. A KOCKA9 írásánál már figyelembe vettem azt is, hogy a megvalósult lépésszám csak páratlan lehet.

KERTÉSZ ÁDÁM



A HÓDÍTÓ ELEKTRONIKÁBAN AZ ORSZÁGBAN MINDENÜTT AZ ÖN PARTNERE VAGYUNK!



INFORMÁCIÓTECHNIKAI VÁLLALAT

VÁLLALJUK:

**ELEKTRONIKUS ÍRÓGÉPEK,
SZÁMOLÓGÉPEK,
MÁSOLÓGÉPEK,
SZEMÉLYI SZÁMÍTÓGÉPEK,
KISSZÁMÍTÓGÉPEK
ÉS SZÁMÍTÓGÉPRENDSZEREK**

- üzembe helyezését,
- szerviz kiszolgáltatását,
- javítását,
- műszaki ellátását.

**Szolgáltatásainkat géptípusokra orientált
speciális képzettségű szakembereink látják el.**

ORSZÁGOS SZERVIZHÁLÓZATUNK

**lakossági
és közületi megrendeléseket
egyaránt teljesít.**

**Központ: Budapest V., Bécsi u. 8.
Levélcím: 1369 Budapest, Pf. 314.
Telefon: 184-899
Telex: 22-4381 22-6841**



LOGO - a programozás- tanulás új útja

Közhely, hogy ma mindenkit érdekel a számítástechnika. Nagy felelősség nehezedik arra a mozgalomra, amely az érdeklődés hullámaikat igyekszik megfelelő mederbe terelni. A lényeg a megfelelő szón van. Lehet regiszterekkel és processzorokkal bővíteni a naiv szimpatizánsokat, csak az a kérdés, meddig maradnak meg hitüknél. Lehet BASIC-et és COBOL-t oktatni a tömegnek, ha nem zavarja álmunkat, hogy holnap százszázalékos háttérrel fordítani a számítógép-bákványok.

Az embereket nem fogja izgatni a csoda, ha kiderül, hogy nem sok közül van hozzá, nem tudnak hozzá közel kerülni. Csak az a gép lehet igazán érdekes, amelyet aktívan lehet megismerni, alkotva lehet használni. Legyen figyelmeztető jel számunkra az idegen nyelvek oktatásának gyakran emlegetett krízise.

A programozás oktatásában csak olyan módszert lenne szabad tömegesíteni, amely nagy valószínűséggel sikerre vezet. Lehet, hogy jogosan büszkélkedünk az iskolai-számítógép létezésével és terjesztésével, a személyi számítógépek szabadabb beáramlásával az országba. Ideje lenne azonban komolyan elgondolkodni azon, hogyan kell megtanítani tömegméretben ezeket az eszközöknek a használatát, valahogy úgy, ahogy a goyóslótól tartó kezét megtanítjuk a betű formálására, a szavak leírására, a gondolatok rögzítésére. E feladat megoldásához kívánunk néhány gondolatot felvetni a LOGO nyelv kapcsán.

A programozásoktatás nehézségei

A számítógép világa a tanuló programozó számára marsbéli táj, melynek minden elemét s a közöttük lévő összefüggéseket önmagukban kell elfogadni, anélkül, hogy emlékeztetőnek az ismert „földi” dolgokra.

Egy programozási nyelv megtanulása alapvetően azért nehéz,

mert olyan elvont fogalmak és kapcsolatok megértését igényli, amelyek nehezen asszociálhatók valós fogalmakkal vagy a tudatba már beépült elvont fogalmakkal.

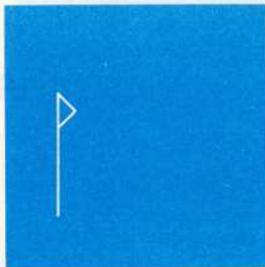
A tanulási folyamat egyik nélkülözhetetlen eleme, az *asszociatív visszacsatolás* csak elvétve kapcsolódik be a programozási nyelv elsajátításába. Kezdetben leginkább a matematika fogalmaival kapcsolódhatnak a programozás fogalmai, gondoljunk például az aritmetikai utasításokra. Már sokkal átteleesebben asszociálható egy tárolási művelet, beírás az operatív tár valamely rekeszébe. Még kevésbé hasonlít valami ismerre a bitmanipuláció, az eljárás-hívás, a megszakítás.

A ciklust például önmagában még értené a tanuló, de a FORT-RAN-ban leírva és a programozó-vegbe ágyazva nehezen tudja fel-fogni értelmét. Önmagában egy utasítás legtöbbször érthető, de a programzöveg sorban következő vagy elágazó utasításai közötti kapcsolat felismerése igényli az algoritmus logikájának, szigorú következettségének ismeretét is.

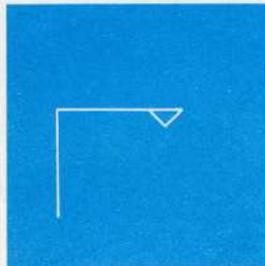
A program algoritmusra egyébként csak feltételesen kapcsolható annak a feladatnak az elvi megoldási algoritmusához, amelynek a kiszolgálására a program hivatott. Többszörös absztrakció keletkezését lehet itt megfigyelni, különösen, ha figyelembe vesszük, hogy a feladat már maga is a valós világ valamilyen vetületét, absztrakcióját jelenti.

A vizuális visszacsatolás

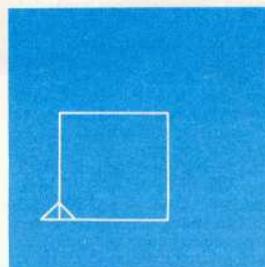
Az iskolai anyag elsajátítása is igényli az elvonatkoztatás képességét a tanulóktól. Az írás-olvasás megtanulása absztrakt elemek sorának elsajátításával jár együtt. Hasonló a helyzet a számtan, a mértan, a földrajz és más tantárgyak esetében is. A nehézségek ellenére a pedagógia mindenki számára biztosítja az iskolai tananyag megfelelő szintű befogadását.



1. ábra



2. ábra



3. ábra

Az iskolai tantárgyak tanítása során jelentős szerepet játszik a szemléltetés, a vizuális visszacsatolás, amely asszociatív folyamatok megindulását biztosítja, így módon erősíti az elsajátítást. A szemléltetés az oktatási folyamat szervezése, a nyelvi közlések erejét a vizuális közlés erejével fokozza.

A vizuális visszacsatolás szerepe változó a különféle tantárgyak elsajátításában. Az írás-olvasás eleve magában hordozza a vizuálitást, természeténél fogva. Hasonló a helyzet az ábrázoló tantárgyaknál, mint a rajz, a geometria, az ábrázoló geometria, a műszaki rajz stb. Magasabb szinten, például a legtöbb művészetben a vizuális forma egyenesen közvetíti, gyakran helyettesíti a tartalmat. Az egyetemi tanulók legtöbbször a ma-

tematika tételek bizonyításának megtanulását papíron végzi, hogy a lépéseket, a bizonyítás algoritmusát könnyebben rögzítse agyában.

A programozás elsajátítása legjobban az írás-olvasás vagy egy idegen nyelv megtanulásához hasonlítható, azzal a különbséggel, hogy az utóbbiak elsajátításának már korai szakaszában a valóság már átél és megismert fogalmainak újraélése, asszociálása folyik.

Bár nincsenek erre vonatkozóan adataink, egyáltalán nem vagyunk biztosak abban, hogy a FORT-RAN vagy a BASIC programozási nyelvek megtanítását sikerülne megvalósítani az általános iskolákban. Ennek legfőbb okát abban látjuk, hogy az említettekhez hasonló, általános angyalteredj magas szintű nyelvek tanulása során nem keletkeznek vagy csak nagyon át-telelesen olyan asszociációk, amelyek támpontot nyújtanak az absztrakt anyag elsajátításában.

Ebben a tekintetben változással kecsegtet a LOGO nyelv. (A nyelv ismertetését lásd a *µM* 1984/6. és 1985/1. számaiban.) A LOGO az első olyan párbeszeds programozási nyelv, amely biztosítja a nyelvi dialógus hatékony grafikai dialógus fejlesztését egy egyszerű grafikai eszköz segítségével. A LOGO didaktikus hatása abban rejlik, hogy a programozás elsajátításának már kezdeti szakaszában megadja a szemléltetés lehetőségét, és ezzel a tanulás általában kritikus első fázisában felfedezéshez, sikerélményhez, intellektuális örömhöz juttatja a tanulót.

A LOGO vizuálitása

A grafikus LOGO bekapcsolásakor megjelenik a háromszöglet-részkesz-köz (teknošbeka), amely közhözött utasításokkal lehet mozgatni, mint például ELŐRE n, JOBBRA k. Értelmszerűen n a lépésszámot (a képernyő szélessége kb. 200 lépés), k pedig az elfordulás szögét jelenti. Ugyanígy lehet hátra, illetve balra haladva rajzolni. (Az okfejtésben az angol nyelvű LOGO kifejezések magyar megfelelőit használjuk, mivel ennek elvi akadálya nincs.)

Tehát kezdjünk el LOGO-t tanulni, 6-tól 60-ig!

Húzzunk egy 80 lépésnyi vonalat a képernyőn, aztán forduljunk jobbra. Begepeljük az utasítást: ELŐRE 80 JOBBRA 90 (lásd az *I. ábrát*). A képernyőn azonnal megjelenik a vonal. Mi történt? Begepelünk két közhözött parancsot, s vizuálisan máris meggyőződhetünk a végeredményesükről.

Menjünk tovább, ismételjünk meg a két utasítást:

ELŐRE 80 JOBBRA 90 (2. ábra). A képernyő válasza nem késik. Ismétéljük ezt meg még kétszer:

ELŐRE 80 JOBBRA 90

ELŐRE 80 JOBBRA 90

Rajzoltunk egy négyzetet (3. ábra).

A programozót öröm tölti el, olyan ez, mint mikor a gyerekeknek először sikerül az ajándék kéréspárral megtenni az utat a sarokig, és visszatérni aggódó szülőjéhez.

Az elvont eszközökkel leírt eljárásnak azonnal szemmel látható eredménye született.

Most összegezzük az eljárásunkat:

ELŐRE 80 JOBBRA 90

ELŐRE 80 JOBBRA 90

ELŐRE 80 JOBBRA 90

ELŐRE 80 JOBBRA 90

Máris motszkálni kezd a gondolat, és az első osztályos tanulók között is sok akadna, aki megkérdezné: miért kell ugyanazt négyyszer leírni, nem lehetne egyszerűbben? Az előadás és a szorzás közötti összefüggés a leírt eljárás és a képernyőn lévő ábrát tekintve nyilvánvalóan megmutatkozik. A tanító a kérdésre megkönnyebbülve feleli: lehet, persze hogy lehet egyszerűbben. Lehet ismételni, imé:

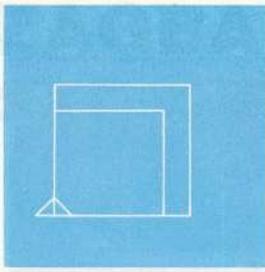
ISMÉTELD 4 ELŐRE 100 JOBBRA 90

Lerajzoltunk egy 100 lépés oldalú négyzetet (4. ábra). A kisiskolások most megilletődve hallgatnak, a tanítón a meglepetés sora. „Lehet ám négyzetet rajzolni még egyszerűbben!” – szól, s legelőli: NÉGYZET, mire a képernyőn megjelenik egy harmadik, ezúttal 120 lépés oldalú négyzet (5. ábra). „Hát a LOGO tudja, mi az a 'négyzet'?” – csodálkoznak a kisdíjak. „Nem tudta.” – válaszolja a tanító, „de én előzőleg megtanítottam neki. Még pedig így:”

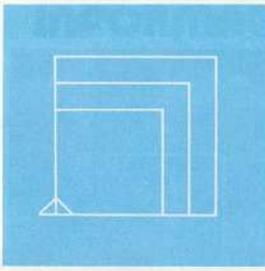
TUDD NÉGYZET

ISMÉTELD 4 ELŐRE 120 JOBBRA 90 KÉSZ

„De hát ezzel a LOGO csak 120 lépés oldalú négyzetet tud rajzolni” – veszi észre az egyik kisdíj. S így észrevétlenül eljutottunk nemcsak az eljáráshoz, hanem annak paraméterezéséhez. Az első



4. ábra



5. ábra

órán! A tanítónak nincs is más dolga, mint bevezetni a változót (hála a korszerű oktatásnak, ez ma már a 2. osztályban természetes), és elkezdhetik a bonyolultabb ábrák rajzolását, a különböző ötletek képernyőre varázsolását.

A BASIC nyelv esetében ugyanezt a fejlődést hosszabb munkával talán elértük volna, de nem biztos, hogy mindenkinél. Jelentős mozzanatnak tartjuk, hogy a LOGO esetében nemcsak az üzemmód párbeszédes, hanem a szemléltetés révén olyan kapcsolat jön létre tanuló és tanító között, hogy az oktatási folyamatban is szinte magától kialakul a párbeszéd. (Folytatása következik.)

DR. HROTKÓ GÁBOR

Cséfalvai Károly, a Magyar Villamos Művek Tröszt Zervezési és Számítástechnika-alkalmazási Főosztályának vezetője a hazai számítástechnika évtizedek óta elismert tekintélye. A ma 55 éves szakember először a NIM-től kapott megbízást 1958-ban egy korszerű számítóközpont létesítésére. Ezt követően több hasonló központ (Csepel, VEIKI, MTA, KSH stb.) felállításában vett részt munkatársaisal. Gyakorlati munkája mellett tevékenyen dolgozott a szakember-utánpótlás oktatásában: tanított egyetemen, mérnöktovábbképzőben, ipari továbbképző központokban.

– Az Ön egyik tanulmányában találkoztam egy különös kifejezéssel: mikro-számítástechnika. Azt kell értenünk ezen, hogy a számítástechnika bizonyos szempontból kettévágott „nagyra” és „kicsire”?

– Nem erről van szó. A mikro-számítástechnika kifejezésre éppúgy átmenetileg lesz csak szükség, mint ahogy annak idején a mikrobarázdás lemezről beszélünk. Hogy mégis ezt használjuk, azokra az új lehetőségekre utalunk, amelyek kihozták a számítástechnikát a géptermekek elefántesontornyából az alkalmazások frontvonalára. A kis és nagy gépek között egyébként érdekes kettőség áll fenn: az egyre intelligensebb mikroszámítógépek egyre nagyobb, óriási adattömeget mozgató számítógépek létét is feltételezik.

– Ebben, az előbb említett átmeneti értelmezésben tapasztalatai alapján miként ítéli meg a hazai számítástechnika helyzetét?

– Abból kell kiindulnunk, hogy néhány éven belül nem lesz eladható a világpiacokon olyan termék, amely valamilyen kapcsolatban ne állna a mikro-számítástechnikával. Nem csupán a gépekre, berendezésekre gondolok, hanem egy a mezőgazdasági termékekre is. Enélkül ma már nincs hatékony minőségbiztosítás.

– A mezőgazdaságról szólva sok minden eszébe jut az embernek, a számítástechnika talán a legkevésbé. Hogy lehet itt felhasználni a számítógépeket?

– Például egy talajmintából meghatározható, hogy adott területen milyen növényi kultúrák termesztethetők a leggazdagosabban és a legjobb minőségben, ehhez milyen műtrágya kell stb. Egyébként nálunk már működnek ilyen eszközökkel felszerelt laboratóriumok a mezőgazdaság megsegítésére. Ezeket kellene népszerűsíteni a

„süllyeszd el a hajómat” típusú játékok helyett.

Visszatérve kiinduló tézisünkre, amelyet a piacokért folytatott, kiélezett verseny nap mint nap meggyőzően bizonyít, sajnos meg kell állapítanunk, hogy a mikro-számítástechnika hazai felhasználásában még saját lehetőségeinkhez képest is elmaradtunk.

– Állandóan új és új számítógépek üzembe állításáról hallunk. Ha ezeket nem használjuk ki megfelelően, miben látja ennek okait?

– A számítástechnika fejlődése három tényező: az emberi erőforrások, a technikai lehetőségek és a társadalmi igények egymásrahatásának eredménye. Ha külön-külön nézzük őket, elmaradásunkat döntő mértékben egyik sem okozhatja. Arra kell tehát gondolnunk, hogy a probléma a tényezők összhangjának hiányában keresendő, összefüggésben belső, strukturális aránytalanságokkal.

– Véleményem szerint egyik tényező fejlődése sem függetleníthető a másik kettőtől. Például egy magasabb színvonalú géppark beállítása elképzelhetetlen, ha erre nincs társadalmi igény, működtetése pedig lehetetlen képzett szakemberek nélkül. Kérem, fejtsé ki részletesebben, milyen aránytalanságok érvényesülnek a hazai számítástechnikában?

– A szakember-háttérrel vizsgálva nincs különösebb okunk panaszra. Ezt bizonyítják sikeres hazai alkalmazások, sőt külföldön dolgozó szakembereink elismerése is. Viszonylag erős a hazai elméleti kutatási háttér is. Mindezzel együtt elmaradásunk egyik okát abban látom, hogy szakembereink egy részéből hiányzik az a fajta profizmus, amely a látványos fő elemek mellett legalább akkora figyelmet fordít a kevésbé „hátlás”, ám használhatóság szem-

VÁLLALKOZÓT KERESÜNK

1500 fős iparvállalat mikroszámítógépes bérelszámolási rendszerének kidolgozására, vagy már máshol bevezetett, hasonló rendszer adaptálására.

Az ajánlatokat az alkalmazásra javasolt számítógép-konfiguráció megjelölésével, „Pontos ügyvitel” jeligére kérjük a szerkesztőségbe.

„A profizmust hiányolom...”

BESZÉLGETÉS CSÉBFALVI KÁROLYAL



pontjából nélkülözhetetlen részletekre. Az alkalmazásnál pedig ez a profizmus azt jelenti, hogy a számítástechnikusnak szem előtt kell tartania az adott terület valamennyi, speciális követelményét, elsősorban az alkalmazási biztonságot. Komoly hiány van nálunk tapasztalt alkalmazástechnikai szakemberekben!

– A műszaki háttérrel szólnak sokan elégtelennek tartják az anyagi forrásokat. Ez volna talán az elmaradás egyik fő oka?

– A közismerten szűkös lehetőségek helyett a műszaki háttérrel kapcsolatban olyan gondokról szeretnék beszélni, amelyek gátolják meglévő lehetőségeink kiaknázását. Nincs megbízható minőségellenőrzés, amely pedig minden egyeből terméknél – a száraztésztától a vasalóig – természetes; megoldatlan a szervizhálózat, a szabványügyi tevékenység, a szoftverellátás megszervezése és még sorolhatnám a „szűk kereszt-

metseteket”. Kirívó gond, hogy a gépek számának gyors növekedését kísérő eufóriában elfeledkezünk a különböző járulékos alkatrészekről, csatlakozó elemekről. Minél messzebb távolodunk a gép központi egységétől, annál szorogatóbb ez a hiányosság.

– A közhangulatból ítélve úgy tűnik, a társadalmi igény, illetve befogadókészség okozhatja a legkisebb fejfájást.

– Igen, az igények nagyobbak, mint amennyit jelenleg ki tudunk elégíteni. Ami a befogadókészséget illeti, azért nem olyan felhőtlen a kép. Sokszor lehet találkozni olyan esettel, hogy ragyogó gépet hozzá méltatlan környezetben állítanak fel. A számítógépek alkalmazását ugyanúgy tervezni kell, mint egy ház építését. Tragikomikus az „adott az Úr báránykát, majd ad hozzá legelőt is” típusú fejlesztés. Ez megint a profizmus kérdését veti fel.

– Mindannak alapján, amit

elmondott, még a számítástechnikában rejlő elsődleges lehetőségeket – termelékenység-, minőségjavulás, gyorsaság stb. – sem tudtuk eddig kellő mértékben kihasználni. Hogyan tudunk akkor másodlagos, sőt további hatásokat figyelembe venni, amelyek egyes fejlettebb országokban már nem kis mértékben jelentkeznek?

– Mégis, ezekre már most fel kell készülnünk. A mikro-számítástechnika terjedése tolvagyűrűző átalakulásokkal jár. Ez érinti az alapanyag-ellátást, a szerszámgépipart és más iparágakat. A munkás nélküli üzemek, a székhely nélküli intézmények lehetősége etológiai, szociológiai, munkajogi kérdéseket vet fel, de megváltozik a közlekedés, a távközlés szerepe is, hogy csak néhány példát említsék. Mindez előbb-utóbb nálunk is óhatatlanul jelentkezik, azért kell már most foglalkozni velük, hogy a strukturális változások ne érjenek készületlenül.

A jelenlegi helyzetet vissza-térve: ha nem tárjuk fel ennek jellemzőit, nincs esélyünk a gyorsabb felzárkózásra. Látni kell továbbá, hogy a tárgyalagos elemzés egyben kijelöli rövid- és hosszabb távú tennivalóinkat. Egy olyan központi stratégiának megfelelően, amelynek fő elve, hogy a mikro-számítástechnika először lehetőleg a közvetlen értéket előállító anyagi természetben jelenjen meg. Ezt a gyors, mérhető „profi” eredményre törést kell a hazai számítástechnikusainknak az eddiginél lényegesen jobban, a szemléletben és a mindennapi gyakorlatban egyaránt magukévá tenniük. Az anyagi és erkölcsi „hozam”-on túl enélkül a felzárkózás és egyáltalán a továbblépés nehezen képzelhető el.

– Szakmai körökben azt hallottam Önről, hogy annak idején sok mindent előre meglátott. Tudna-e most is mondani valamit a számítástechnikával foglalkozók számára, amit jövőbeni munkájukhoz megfontolásra ajánl?

– Nem nehéz olyat jósolni, ami „a levegőben lóg”, csak észre kell venni. Az új technika birtokbavételének lázában elterelődött a figyelem a matematika olyan új eredményeiről, amelyek éppen ehhez az új technikához tartoznak. Ma már egy sereg olyan alkalmazási területet lehet említeni, ahol ezek a jól átgondolt modellezési munkán alapuló, új matematikai módszerek vezetnek eredményre. Így például az ún. prediktorkorrektor módszerek általánosításai még a leggyorsabb nyilvántartási, elszámolási rendszereknél is alkalmazhatók. Ha ugyanis a feldolgozás soron következő lépcsőjét egy megfelelően fejlett modell segítségével előre megmondjuk („prediktáljuk”), akkor az időközben beérkező tényadatokkal csupán a jóslást kell korrigálni.

Az elkövetkezendő években a matematika numerikus és nem numerikus módszereit rendre kézbe kell venni, és az aktuális igényeknek, valamint technikának megfelelően értékelni.

LACZKA MIKLÓS

Alapozás X.

Pneumatikus operátorunk matematikai modelljeivel foglalkoztunk a legutóbbi alkalommal. Ennek révén belekóstoltunk egy kis sé magába a matematikai modellkészítésbe is. Hasznos és tanulságos lesz, ha folytatjuk e tevékenységet, még pontosabb, még egzaktabb matematikai modellek megalkotása érdekében. Emellett természetesen folytatjuk a konkrét munkát is az ún. „logikai hálózatok” építésével. Most ez utóbbi téren lépünk előbbre egyet.

Mik lehetnek a céljaink?

A digitális berendezések néhány fajta operátorból épülnek fel. Áll ez mind az elektronikus, mind a hidraulikus, mind a pneumatikus, mind pedig minden más (például fényvel és más rezgésekkel dolgozó) digitális automatára is. Elméletileg érdekes, a gyakorlat számára pedig hasznos, és ezért kíváncsok, hogy egyfajta operátorból minél többfajta egység legyen felépíthető. Az is kíváncsok természetesen, hogy ezek, az így felépített operátorok ráadásul még különbözőn is minél többféle feladat megoldására legyenek alkalmasak.

Pneumatikus operátorunk (1. ábra) az előbbi szempontból különlegesnek bizonyul. Látni fogjuk majd, hogy – bizonyos értelemben – mindenféle feladatra való számológép-építő egység (összetett operátor) felépíthető belőle.

Fogjunk hozzá tehát bátran, és alkossunk ún. „kapcsolásokat”! Ez azt jelenti, hogy kössük össze az operátorokat a lehető zavartalan áramlást lehetővé tevő csövekkel. Az így kialakuló kapcsolásokat aztán vizsgáljuk meg, hogy hogyan működnek, mire tudjuk azokat használni.

Olyasfélé az a konstrukciós munka, mintha operátoraikat atomok (ionok) lennének, és belőlük ionokat (illetve molekulákat) építenénk fel. A vegytani szerkezeti képletben ún. „karok” jelölik a közvetlen kapcsolatokat, esetükben pedig lévetezések. A vegytanban sokféle atom sokféle kapcsolása lehetséges, itt egyféle „atom” sokféle kapcsolása. „Univerzális”, „atomunk” az 1. ábrán látható pneumatikus operátor. A kapcsolási műveletek révén létrejövő rendszert „hálózatnak” is és „kapcsolásnak” is szokták nevezni. A logikai hálózat és logikai kapcsolás szakkifejezést az indokolja – vagy csak indokolta (?) –, hogy valamikor a logika kitértekül volt (mert csak kitértekül logikát tudunk elképzelni), és e hálózatokban is kitérteknek tekintünk – kezdetleges modellek alapján – egyes állapotjelzőket.

Mielőtt szabadjára engednénk képzelőtehetségünket, és mielőtt e nagyon hasznos és szórakoztató kísérletezés elkezdenénk, célszerű azzal is foglalkoznunk, hogy egyáltalán milyen eredményre juthatunk. Milyen eredmények lennének például az ideálisak számunkra? Hátha kielégíthetők lesznek ezek az igények.

Mire volna szükségünk tehát, hogy ha például legkényelmesebben kívánánk digitális berendezést, például számológépet építeni. E gyűjtést – nagyon hasznos munka – most az Olvasóira hagyjuk. Birkózson vele egy kicsit! Itt csupán egyféle kíváncságot említünk. Nyilvánvalóan jó lenne, ha volna például négyzetreemlést végző, összeadást végző, szorzást végző operátor. Tegyük fel, hogy e kíváncságot kielégíthető, és könnyen elő tudunk állítani ilyen operátoro-

kat. Hogyan folytathatnánk ebben az esetben a tervezést? A válaszadáshoz nincs szükség arra, hogy meg tudjuk mondani, hogy milyen ezeknek az operátoroknak a belseje. (Erre a kérdésre most, ebben a pillanatban nem is tudnánk felelni, később azonban ki fog derülni, hogy ez nem nehéz dolog.) Ha az operátor belsejéről nem is kell mindent tudnunk, környezetéhez való kapcsolódási lehetőségeit azonban mindenképpen ismernünk kell.

Ha tervezni akarunk, ábrázolnunk is kell a szóban forgó operátorokat. Az operátorokat nyilván most is egyetlen téglalappal jelöljük. Az is nyilvánvaló, hogy az operátorokat összekötő vezetékeket is ábrázolni kell valahogyan. Ha ennek a hagyományos, vonalas módját választjuk, akkor azt kell megmondanunk, hogy milyen lenne például egy öszszeképző és közvetlen környezetének vázlata. Erre a kérdésre az öszszeképzőre csatlakozó vezetékek megajrolásával kielégítő választ tudunk adni. Vessünk egy pillantást a 2. ábra vázlataira! Vajon melyik vázlat lehet jó? Talán első pillanatban meglepően hangzik, hogy mindegyik vázlat jó lehet.

A vázlatok annak a módjáról adnak eligazítást, hogy hogyan történhet az operátor informálása és milyen módon informálhatja az operátor a környezetét. Más szavakkal ez a következőt jelenti. Hogyan bocsáthatjuk az operátor rendelkezésére az operandusokat? És hogyan bocsáthatja az operátor rendelkezésünkre az operátumot, azaz az operáció eredményét? A válasz az, hogy *sokféleképpen*. Oly sokféleképpen, hogy nincs is helyünk részletes elemzésükre. Lássunk azonban néhány kiragadott esetet, a 2. ábra elemeként sorrendjében! Szerencsére ezeknek a szemlélyeknek alapján is könnyű elképzelni a lehetőségek gazdag sokaságát, egyúttal teljes áttekintésükre is megadják a lehetőséget. (Az ábrán x az egyik, y a másik öszszeadandó, z pedig az eredmény, számjegysorozat formájában.)

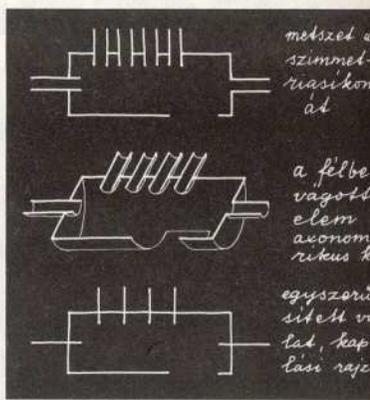
2.(1) ábra. x és y számjegysorozatának minden eleme külön-külön vezetékeken egyidejűleg áll az öszszeképző rendelkezésére (egy darabig). Ha az operátor elkészült az eredmény-számjegysorozattal, akkor annak minden eleme egyidejűleg rendelkezésre áll az előzőktől különböző, külön-külön vezetékeken (egy ideig).

Egy másik lehetséges magyarázat ugyanerre az esetre a következő. x és y számjegysorozatának minden eleme külön-külön vezetékeken áll az öszszeképző rendelkezésére, egy-egy darabig, valamikor, nem szükségképp egymással egyidőben. Az eredmény számjegysorozatának elemei az előzőktől különböző, külön-külön vezetékeken észlelhetők egy ideig, valamikor, nem szükségképp egymással egyidőben.

2.(2) ábra. x és y számjegysorozatának minden eleme külön-külön vezetékeken, egyidejűleg áll az öszszeképzőnek rendelkezésére (egy darabig). Az eredmény számjegysorozatának elemei egy közös, az előzőktől különböző vezetéken észlelhetők, valamikor, valamennyi ideig, egymástán sorrendben.

Egy másik lehetséges magyarázat ugyanerre az esetre a következő. x és y számjegysorozatának minden eleme külön-külön vezetéken áll az öszszeképző rendelkezésére, egy darabig, valamikor, lehet, hogy számjegyenként más-más időben. Az eredmény számjegysorozatának elemei egy közös, az előzőktől különböző vezetéken észlelhetők valamikor, valamennyi ideig, egymástán, valamilyen, nem szükségképp a sorozatban levő sorrendjükben.

E példákbl már megállapíthatjuk, hogy az egyes



1. ábra

információk megjelenési (tartózkodási) helyére és megjelenési (ott-tartózkodási) idejére vonatkozó információkra különösen kell figyelniük. A 2. ábrán szereplő példák esetében 2 információnak például észlelése, feldolgozása lehet

- ugyanabban az időben, ugyanazon a helyen történő;
- ugyanabban az időben, különböző helyen történő;
- különböző időben, ugyanazon a helyen történő;
- különböző időben, különböző helyen történő.

Estünkben a) nem lehetséges, mivel nem lehet egy csőben (pneumatikus vezetékekben) egy ponton ugyanakkor áramlás is és nem áramlás is, tehát a két különböző információ megkülönböztetésére nem volna alkalomunk. De lehetséges például egy információsorozat elemeinek időben sorban, egymás után való bejuttatása valahová úgy, hogy mindegyik elemet külön-külön, egymással (geometriailag) párhuzamos vezetékeken továbbítsunk.

Ha az információk megragadása (megnevezése) csak két jellemzővel (idő, hely) történik, az a) esetben nyilvánvalóan nem változtható meg 2 információ megkülönböztetése. Egyébként azonban nem lehetetlen a megkülönböztetés, mert például a fényinformáció (fényimpulzusnak) hullámhossza is van, és ez alkalmas a hely és idő információ azonoságára esetében 2 információ egymástól való megkülönböztetésére. Ez utóbbi esetben természetesen az azonos helyű, azonos időben jelentkező, azonos hullámhosszú információk megkülönböztetése okoz majd problémát, ami egy újabb megkülönböztetésre alkalmas jellemző szerepeltetésével már megoldható lesz.

Magyarázatra szorul még a 2. ábra (7), a (9) és a (10) esete.

Lássuk például a (10)-et! x és y számjegysorozatát képviselő információk csak egymás után jelenhetnek meg az egyetlen bemeneti vezetéken. Sorrendjük lehet például $x_1, y_1, x_2, y_2, \dots$ vagy $x_1, x_2, \dots, y_1, y_2, \dots$ vagy bármi más sorrend. Az operátornak kell tudnia, hogy mikor mit érzékel.

A (11), (12), (13), (14) és (15) esetek azért érdekesek, mert kiderül belőlük, hogy a gyakorlatban nem mindig lehet egy vezetéket bemeneti vagy kimeneti vezetékeknek nevezni. Ezekben a példákban egyes ve-

metszet a szimmetriánál

a felbe vágott elem azonos ritkus

egyszerű síkelt val tal, kap lási rajc

zetékeken egyszer bemeneti, máskor pedig kimeneti információ van (áramlik). A (15) példa esetében ezek egymás után, például az $x_1, x_2, \dots, y_1, y_2, \dots, z_1, z_2, \dots$ sorrendben észlelhetők ugyanazon a ponton (ugyanabban a vezetékben).

Hangsúlyozzuk, hogy elméleti célokból meg lehet csinálni és sokszor célszerű is megcsinálni a különböző funkciójú és különböző irányú információáramlások külön-külön vezetékeken való lebonyolítását, például matematikai modellekben.

Pneumatikus vezetékek esetében természetesen nem lehet a vezetékek egy pontja egyidőben különböző irányú információk továbbítására igénybe venni. (Fényinformációkra ez természetesen nem áll).

Elektronikában újabban egyre gyakoribb egy-egy pont (vezeték) kétirányú használata (általában nem egyidőben). Ilyenkor ugyanazon a ponton egyszer bemeneti, máskor pedig kimeneti információk érzékelhetők.

Ha egy vonal (vezeték) kétirányú, akkor nincs értelme bemenetnek vagy kimenetnek nevezni, helyesebb a *be-kimeneti vezeték* vagy csak a *csatlóvonalon, csatlóvezeték* név használata.

A 2. ábra (16) példáját csak tévedések elkerülése

céljából szerepeltettük, ez az (1) példa „rövidített” változata, a vezetékek ismételt megrajzolása helyett azok multipllicitását (hányszoroságát) jelöljük a ferde áthúzás mellett.

A gyakorlatban nem ritka egyes be- vagy kimeneti pontok összekapcsolása ugyanannak az operátornak valamelyik másik be- vagy kimeneti pontjával. Ez gyakran jól használható megoldásokat szolgáltat. Azonban matematikai modellek használatánál elzavart gyakran előfordul az implicit kapcsolat az ilyen hálózatok matematikai modelljében. Ebből pedig ritkán szokott jó származni. Az implicit függvények ezgakt matematikai kezelése, modellezésben való helyes alkalmazhatóságuk ismerete terén ugyanis nincs különösebb okunk a dicsévre.

Megjegyezzük, hogy a vegyes (be-ki) pontokat olyankor, amikor bemeneti pontként működnek bemeneteknek, olyankor, amikor kimeneti pontként működnek, kimeneteknek, amikor pedig „lezártak” (nem működtethetők sem bemenetként sem kimenetként), nem léteznek tekinthetjük.

Igy, összefoglalva az eddigieket, jegyezzük meg, hogy a bemeneti információknak a bemeneteken való rendelkezésre állásának (rendelkezésre bocsátásá-

nak), a bemeneti információ feldolgozásának, és a kimeneti információknak a kimeneteken való rendelkezésre bocsátásának a módja *sokféle lehet*.

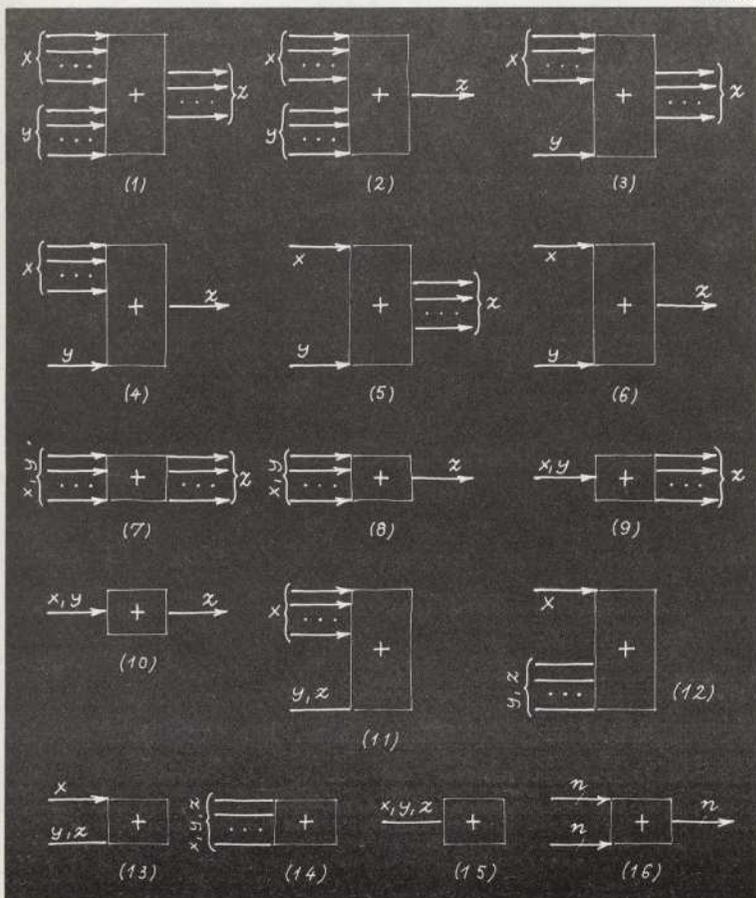
Példáinkból láthatóan, nagyon is sokféle, és mindegyik típus megvalósítható és meg is valósítják valamilyen gyakorlati hasznú berendezésben.

Mivel nekünk most célunk *minden lehetséges módon* digitális berendezést építeni, kiválasztunk egy – talán a leggyakoribb – típust és ezzel foglalkozunk annak érdekében, hogy *egy lehetséges megoldást* elő állítsunk. Típusunk a 2. ábra (1) példa típusa lesz.

Céljaink érdekében főleg – de nem kizárólag – olyan operátorokra lesz szükségünk, amelyek függvény-helyettesítési értéket képeznek. Ezek az operátorok tehát, mivel függvény-helyettesítésiértékkiszámító feladatot látnak el, függvény-helyettesítésiértékkiszámító egységek, függvény-helyettesítésiértékkiszámító gépek. Hogyan működjenek ezek a függvény-helyettesítésiértékképző (kiszámító) operátorok? Ezt most – a sok-sok lehetőség közül – mi határozzuk meg.

Rögtön az elején megmondjuk, hogy matematikai értelemben függvénnyel ritkán találkozunk a számológepben. Ritkán van, vagy talán *soha sincs egyidőben jelen a hozzárendelt azzal, amire őt rendeljük*. Az „ős” (azaz a független változó értéke) időben előbbi szokott lenni a „kép”-nél (azaz a függő változó értékénél). Emiatt – függvények esetében is – mindig tudnunk kell valahonnan, hogy mit mely időpontban vegyünk számításba, mit mikor olvassunk le, figyeljünk meg, érzékeljünk stb. E függvénymodellezés lényegét jobban megértjük, ha transzformáció-végrehajtásról beszélünk, így rögtön világos, hogy az operátor által transzformációval képzett mindig később jelenik meg annál, amit transzformál. (Legfeljebb véletlen lehet a két egymáshoz tartozó értékpár egyidejű jelenléte a bemeneti, illetve kimeneti pontokon.)

2. ábra



Egyszerű kapcsolások, hálózatok

Most már nyugodtan rátérhetünk az operátorok kedvünk szerinti összekötötésére és az így kialakított hálózatok vizsgálatára. A vizsgálatot most főleg egy szempontra korlátozzuk, arra, hogy a hálózat milyen transzformáció megvalósítására, milyen függvény modellezésére használható.

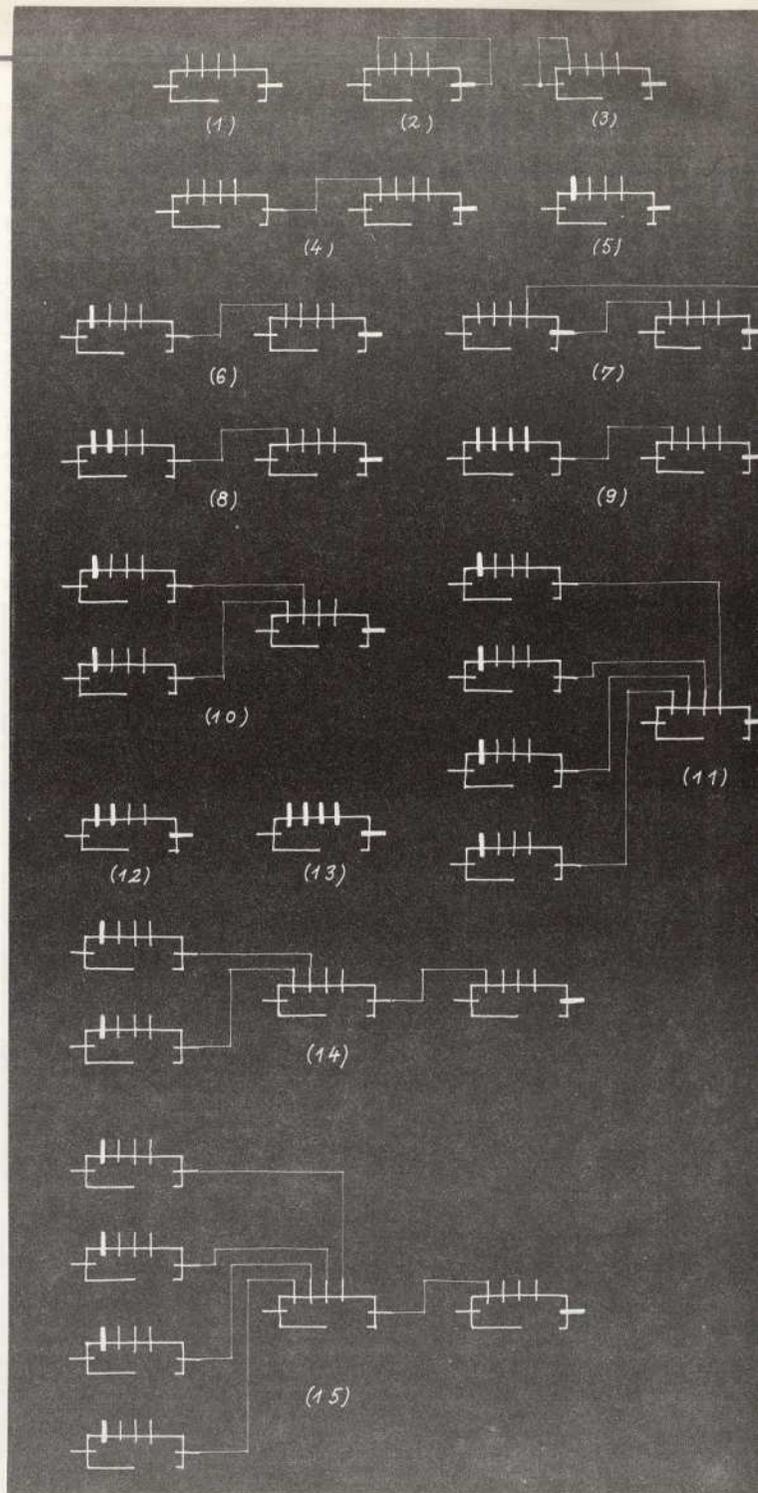
Mit jelent ez konkrétan? Azt, hogy ha a szóban forgó feladatot elvégző operátor bemenetein bizonyos ideje már fennáll egy állapot, akkor ennek az időtartamnak a kezdetét követő bizonyos idő múlva a bemeneteken levő állapotok és a rendszer által egyértelműen meghatározott (minden másról viszon nem függő) állapot áll be, és marad fenn a kimeneteken, bizonyos ideig. (3 darab „bizonyos” idő szerepel a definícióban. Hogy ezek miktől és hogyan függenek pontosan, érdekes, de a gyakorlatban ritkán fontos kérdés, mert ezeknek az időeknek mindentől független felső vagy alsó korlátjukkal szoktak tervezni, és ilyen korlátok a gyakorlati esetekben mindig lenni szoktak.)

Definíciókban hangsúlyoztuk a kimeneti állapotoknak a rendszertől való függését. Ez a függőség nem azonos a rendszer előző állapotaitól való függőséggel. Ez is gyakori és sokszor hasznosított jelenség. Sőt a pontos „bizonyos idők” is függhetnek a rendszer előző állapotaitól és majdnem mindig függhetnek is, főleg más, például elektronikus rendszerek esetében. (Az ezektől is független felső és alsó korlátok azonban ilyenkor is léteznek.)

Példáinkban a rendszer előző állapotaitól való függést – most – mivel ez ezgakt módon megindokolható, el fogjuk hanyagolni.

A kimeneti állapotok és a bemeneti állapotok között csak a kialakítótól a kialakított felé irányuló hatással számolunk. (Példáink többsége olyan, hogy

3. ábra



ez az irány egyértelműen megállapítható.) Az ellenkező irányú hatást elhanyagolhatónak vesszük, tehát feltételezzük a visszahatásmentességet. Természetesen ez is ritkán van így teljes pontossággal. A 3 bizonyos idő pontos meghatározásához visszahatással is számolni kellene, különösen más, például elektronikus operátorok esetében. Szerencsére igaz az is, hogy van olyan felső, illetve alsó korlátja a „bizonyos időknek” (bizonyos időtartamoknak), amely minden esetben igaz, és így a visszahatásmentesség feltételezése sem vezet helytelen eredményre, a felső és az alsó korlátokat használó becslésekkel dolgozó tervezés során.

A teljesség kedvéért említeni kell azt is, hogy a „bizonyos idő” egymástól és sok minden mástól is függhetnek. A gyakorlat által használt esetekben azonban erre is áll, hogy e függés oly kismértékű, hogy a már említett alsó és felső korlátok használatával minden problémát meg tudunk oldani.

Talán felesleges említeni, hogy rendszerünk viselkedése általában nem függ közvetlenül az időtől, csak az egyes jelenségek időbeli viszonyaitól; pneumatikus operátorunk például ugyanúgy működik hétfőn, mint kedden.

Kezdjük a kapcsolásokat, a hálózatépítést a legegyszerűbb esetekkel! A példákkal való ismerkedés során mindig célszerű a „mit csinál?”, a „hogyan működik?” kérdés mellett azt is megvizsgálni, hogy a kapcsolás révén létrejövő rendszer kimenetén tapasztalható állapot, azaz az operátum mikor mit indikál (azaz mit jelez), mikor miről informál, mikor mire lehet belőle következtetni, mikor miről tudósít, hiszen a rendszerépítés egyik fő célja az informálás.

Nem minden kapcsolatban van szerepe minden bemeneti, illetve kimeneti pontnak (csöcsconknak); amelyeket az összetett rendszer be-, illetve kimenetként használunk közülük, azokat vastagabb vonalsszakasz jelöli a 3. ábrán.

Az összes operátor táp-bemenetére természetesen egy táp-levegő vezeték csatlakozik. A rajzok áttekinthetősége érdekében a táplevegő-hálózat megrajzolását mellőztük.

A 3.(1) ábrán látható rendszer kimenetén mindig 1-et tapasztalunk, ez a rendszer állandó függvényi modellje. Kimeneti pontja egyenértékű a táplevegő-hálózat egy szabad csöcsconkjával. A 3.(3) és a 3.(4) rendszer ennek párja, szintén állandó függvényi modellje, melynek értéke azonban 0. E rendszerek kimeneti pontja a szabad légtérből jövő vezetékével egyenértékű. A 3.(5) ábra rendszere 0,5 értékre tükröző operátort modellez; 1-gyel tudósít a bemenet (valamennyi idővel előbbi) 0 állapotáról, és 0-val jelzi a bemenet (valamennyi idővel előbbi) 1 állapotát.

A 3.(6) rendszer az azonos leképezést modellezi, azaz 1-gyel indikálja a bemenet (valamennyi idővel előbbi) 1 állapotát, 0-val pedig (valamennyi idővel előbbi) 0 állapotát. Ez az operátor késleltetésre is használható.

A 3.(2) és a 3.(7) rendszer kemény dió. Most nem foglalkozunk velük. (Az Olvasó azonban szánjon rájuk néhány percet! Próbálja megfejteni, hogy hogyan működnek! Elemzések később sorra fog kerülni.)

A 3.(8) rendszer 1-gyel indikálja bemenetei valamelyikén (valamennyi idővel előbb) fennállt 1 állapotot, 0-val pedig azt az esetet, amikor minden bemenete 0 állapotban volt. Az operandus-operátum vagy a bemeneti állapot-kimeneti állapot táblázat 2 bemenet esetére a következő.

operandus	operátum	
0	0	0
0	1	1
1	0	1
1	1	1

A 3.(9) rendszer ugyanolyan, mint a 3.(8), de 4 bemenete van. E rendszerek maximumfüggvény modellezésére használhatók.

A 3.(10) rendszer 1-gyel indikálja azt, hogy mindegyik bemenete (valamennyi idővel előbb) 0 állapotban volt, 0-val pedig az összes többi esetet. Mini-

munképző operációt végez, alsó burkoló függvény modellezésére használható.

A 3.(11) rendszer ugyanaz, mint a 3.(10) rendszer, de 4 bemenete van. Táblázata 2 bemenet esetére a következő.

operandus		operátum	
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

A 3.(12) rendszer operandus-operátum táblázata 2 bemenet esetére:

operandus		operátum	
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Az operátum 1 voltából arra lehet következtetni, azaz a kimenet azt indikálja 1 értékkel, hogy mindkét bemenet (valamennyi idővel előbb) 0 állapotban volt. Minden más esetet 0-val indikál.

A 3.(13) ugyanaz, mint a 3.(12) de 4 bemenet esetére.

A 3.(14) rendszer azt indikálja 1 értékkel, hogy a bemenetek közül legalább egy (valamennyi idővel előbb) 0 állapotban volt; az egyetlen ettől eltérő esetet, amikor minden bemenet állapota 1 volt, 0-val indikálja. Táblázata a következő.

operandus		operátum	
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

A 3.(15) rendszer ugyanolyan, mint a 3.(14), de 4 bemeneti pontja van.

A 3.(16) rendszer csak bemeneti pontjainak számában tér el a 3.(8) és a 3.(9) rendszertől.

A 3.(17) rendszer 8 bemeneti ponttal rendelkezik. Látható, hogy van hely még 2-szer négy bemeneti pont létesítésére. Ugyanazt a feladatot végzi, mint a 3.(10) vagy a 3.(11) rendszer.

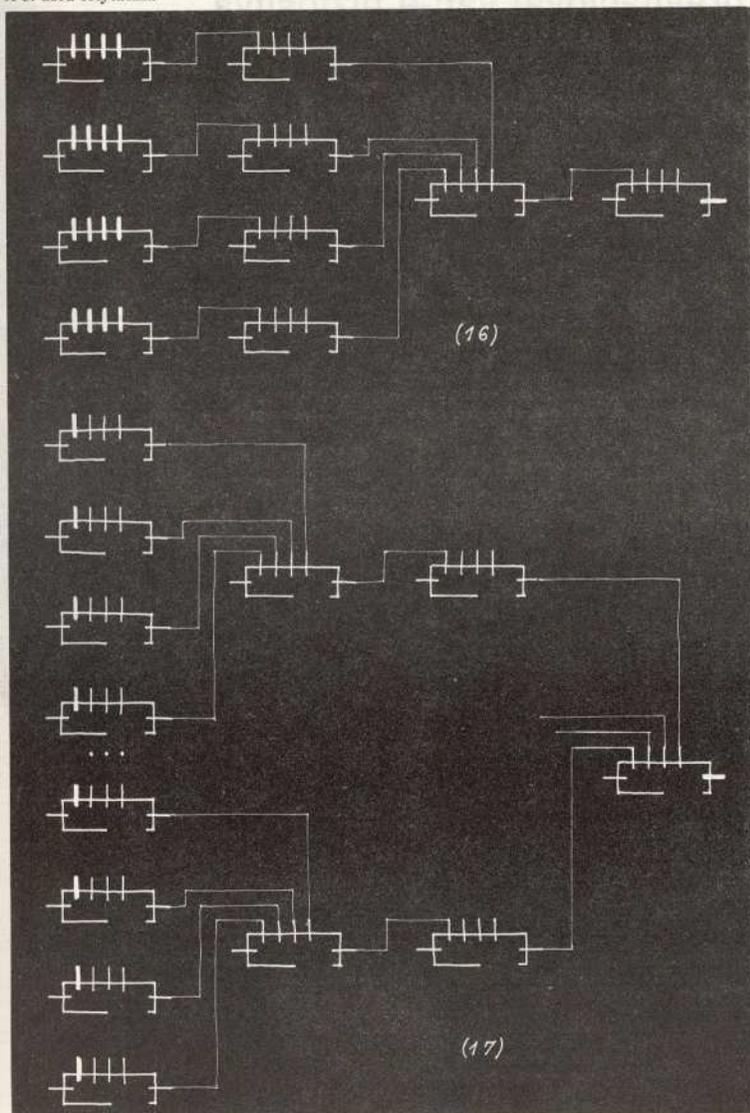
A példák magyarázatánál – helykimélés céljából – nem említettük mind a három „bizonyos időt”, csak egyet. Természetesen a másik kettőnek változatlanul szerepe van, a pontos működéstörvényben szerepelniük kell.

Természetesen vetődik fel a kérdés, hogy minden elképzelhető kapcsolásról meg tudjuk-e mondani, hogy minden elképzelhető bemeneti folyamat esetében hogyan fog működni. A válasz az, hogy nem. A gyakorlati esetekben sem tudunk válaszolni az ilyen kérdésekre, csak bizonyos tulajdonságú kapcsolások és bizonyos tulajdonságú bemeneti folyamatok esetében. Más a helyzet az elméletben. Ha van egy jó matematikai modellünk, azzal többre vagyunk képesek. A kérdés csak az, hogy emögött az (elméleti!) több mögötti van-e valóság, és ha igen, milyen közel van hozzá. Nem tudjuk.

Felhívjuk a figyelmet arra, hogy egy jó matematikai modellnek már a csövezetékek alakját és hosszát is kezelnie kell, elektromos esetekben pedig például a kialakuló és állandóan változó elektromágneses térerőt is számításba kell venni. Egyik követelménye sem tekinthető teljesítettnek. Szerencsére ezek a fajú tehetetlenségek a lehetséges esetek olyan tartományára esnek, amelyeket a gyakorlat ki tud kerülni.

Példáink meggyőzően bizonyítják, hogy pneumatikus operátorokkal kapcsolással nagyon sokféle operátor előállítható. A bemeneti pontok: állapotról nagyon sokféle információ létrehozására képes rendszereket tudunk felépíteni. Joggal vetődik fel a kérdés, hogy van-e ennek a tevékenységnek határa, azaz van-e olyan, csak a bemenetek állapotától függő 2 értékű (helyesebben 2 elemű érték készletű) függvény, aminek helyettesítési értékét kiszámító rendszert már nem tudnánk operátorokból kapcsolással létrehozni. Könnyű bizonyítani, hogy ilyen függvény nincs. Azaz képesek vagyunk bármilyen operandus-operátum táblázathoz egy azt megvalósító rendszert megalkotni, a már megismert keretek között. Ezzel is foglalkozunk majd a következő alkalommal.

A 3. ábra folytatása



Commodore 64 felhasználók figyelmébe ajánljuk az
alábbi programokat:

1. *Kémény áramlástechnikai méretezés*

Alkalmas szén-, olaj-, gáztüzelésű kazánok kéményeinek méretezésére. A kémény anyaga lehet: téglá, beton, hőszigetelt fém.

Méretezhető deflektoros kémények is, figyelembe vehető, hogy a kazánház légpótlása, ill. a kazán huzata mesterségesen, vagy gravitációval van-e biztosítva.

2. *Fűtőtest-kiválasztás*

Alkalmas központi fűtés tervezéséhez, hőszükséglet-számításra. A kiválasztott fűtőtestekről a mennységek összesítésével anyaglistát készít, mely sornyomtatón is lekérhető.

3. *TRACE-64 programozói segédlet*

Segédlet, mellyel az alkalmazói program elkészítési ideje a felére csökkenthető.

Képes önmagát bármely területre áthelyezni, disassembláláskor lehetőség van címkék használatára is. Leghatékonyabb eszköze a NYOMKÖVETÉS, zero-page területet nem bántja, s csak 4 kb-ot foglal el.

4. *SOFTKEEPER titkosító program*

Lehetővé teszi a futó programok, illetve programok által használt adatok tetszőleges kulcs szerinti átkódolását és titkosított formában történő tárolását. Az eredeti tartalomhoz csak a helyes kulcs ismeretében lehet hozzájutni.

A gépi kódú program a kódoló utasításon kívül, még 12 – a felhasználó munkáját megkönnyítő – utasítást ismer.



1. Gépjármű nyilvántartó és járulékszámító
program

Alkalmas az 50/1981. (X. 31.) és a 85/1982. (XII. 4.)
PM sz. rendelet alapján 20 termelőegységre 300
gépjárműnek a járulék számítása szempontjából
szükséges adatainak tárolására, a járulék
negyedévi és évvégi kiszámítására, valamint
kinyomtatására.

2. Nagyüzemi
keresetadó számító
programrendszer

Alkalmas
a mezőgazdasági és
ipari vállalatoknál
alkalmazható ún.
nagyüzemi keresetadó
kiszámítására, max.
10 000 dolgozó esetén.
Sávonként összesíti
a keresetet, az adót és
a hozzá tartozó
létszámot.

3. Költségfelosztó programcsomag
számlaszámra, kettő adatsor
gyűjtésére, illetve tetszőleges
nagyságú összeg felosztására,
a felosztott összegnek a költséghely
(számlaszám) sorrendje szerinti
sorbarendezésére.

A programok megrendelhetők
az alábbi címen:

Alkotó Ifjúság Egyesülés
Számítástechnikai Iroda
1519 Budapest, Pf.: 330.
Telefon: 112-666
113-608



Robottechnika

mes választani, amelyek egybeesnek a robotkarnak mozgatóegységeit irányító adatokkal. Miért? Egyszerű, hogy a robot karjának mozgását is legalább három részre lehet bontani, három, egymástól független mozgást végző részegységre. Ezeknek a részegységeknek külön mozgatórendszerük van.

Korunk technikájában egyre nagyobb szerep jut a számítógépeknek. Sokoldalúságával a mindennapi élet számos területén hódít teret magának. Fontos segítő-társ az adatfeldolgozásban, ügyvitelben, de utjabban a tervezésben, a tudományos munkában is. Mi egy kevésbé ismert felhasználást mutatunk be: a számítógépeknek az iparban a munkafolyamatok irányítására való alkalmazását.

Régi törekvése az embernek, hogy mindennapi, fárasztó, esetleg egyhangú vagy nehéz munkáját könnyebbé tegye. Ezt a törekvést a történelem során az emberrel választól napjainkig végigkövethetjük az egyszerű munkaeszközök készítésétől kezdve a bonyolult ipari berendezésekig. A munkaeszköz fejlődése jelentősen befolyásolja a társadalom alakulását is. Míg az óskorban a szerszámok készítését a létminimum elérése tette szükségessé, napjainkban már a gépek kiszorítják a munkásokat a gyárakból, az emberek a két kezi munkaterületekről a szellemi fejlesztő tevékenységek felé áramlanak. A gépek tehát átveszik fáradságos munkánkat: nekünk csak magukat a gépeket kell megterveznünk, szervezést rendezsük állítanunk, működésüket irányítanunk, ellenőriznünk.

Ez utóbbi tevékenységeket korunkban már igen nehéz lenne emberekkel végezteni, mivel ehhez az emberi képességeket meghaladó pontosság, gyorsaság, összetettség, összeganság kell. Ezeket az irányítási, ellenőrzési, vagyis szabályozási feladatokat oldja meg a szigorú kívánalmaknak megfelelően a számítógép.

Egy ilyen folyamat részletes elemzését nem tüzhetjük ki célunknak. Sokkal inkább azt, hogy nagy vonalakban bepillantást adjunk a számítógép gyakorlati felhasználásába. Először is készíteni kell egy modellt, amit irányítani fogunk. A modellt úgy kell kiválasztani, hogy a valósághoz közel álljon, vagyis majdnem annyit tudjon, mint igazi társa.

Mire legyen jó a modell?

Egy modell általában akkor jó, ha alapvető tulajdonságaiban eredetijével megegyezik. Ez esetben nyilvánvalóan a felhasználás az alapvető tulajdonság, tehát: helyettesítse az embert valamilyen összetett, egyhangú munkában. Építsünk például egy robotot.

Mit tudjon a robot?

Elsősorban olyan műveleteket végezzen, amelyek sokszor ismétlődnek, olyan pontosságot, kitartást követve az embertől, amit nem valószínű, hogy teljesíteni tudna. Könnyen lehessen átállítani egyik munkáról a másikra.

Egy modell általános tulajdonságaihoz természetesen az is hozzátartozik, hogy kiemljen bizonyos részleteket az igazi rendszerből, és ezeket a hangsúlyozott részleteknek alárendelje a többi. Jelen esetben a szabályozóhoz kapcsolódó részletek lesznek a legfontosabbak. Ennek megfelelően az egész konstrukciót úgy kell majd megalkotni, hogy a szabályozhatóságát lehetőleg minden egyes részegység segítse.

Ezek voltak a legfontosabb elméleti kérdések. Most már ismerjük az elvi célokat, következhet a megvalósítás.

Gyakorlati kérdések

Ha az emberi kezét akarjuk géppel helyettesíteni, mennyivel tudjon többet és mennyivel kevesebbet? Az emberi kéz (kar) elsődleges feladata tárgyak megfogása és mozgatása. A tervezés során figyelembe kell venni kezünk hiányosságait: elemezve a testek mozgását, majd megfigyelhetjük, hogy az emberi kéz a mozgatóeszköz csak egy részét tudja elvezetni. Többek között ebben többet tud majd a robotunk. Karunk hiányosságait esetleg egy teljesen más felépítésű robotkarral pótolhatjuk.

A tárgyak megfogásának mechanizmusában viszont sok mindent elhagyhatunk, egyszerűsíthetünk. Nem lesz szükségünk az emberi kéz rendkívül összetett és bonyolult ujjaira, ujjperceire. Lehet, hogy robotkezdünk nem tudja majd a tárgyakat annyifeleképpen megfogni, de erre nem is lesz szükség. Kevesebbet fog tudni a gyorsaság terén is, hiszen kezünk közelítőleg ugyanazt a pontosságot éri el a nagy és kis sebességű mozgások esetén, robotkezdünk pedig a pontosságot, a mozgások finomságát a legtöbb esetben a gyorsasággal fordított arányban tudja biztosítani.

A szabályozhatóságot szem előtt tartva, a kettő közül a gyorsaság rovására a pontosságot kell előtérbe helyezni. Ez viszont újabb előnyökhöz vezet az emberi kézzel szemben, hiszen a kéz pontosságának van egy felső határa, robotkezdünk pedig az egységnyi időre jutó elmozdulást lényegesen korlátlanul sokszorozhatjuk, ami a pontosság hatalmas növekedéséhez vezet.

Milyen anyagok állnak rendelkezésre?

Sajnos ez a tényező minden előzetes számítást és tervezést elronthat. A mi modellünk viszonylag kevés alkatrészt igényel, inkább konstrukciós ötleteket próbálhatunk bonyolult szerkezeti megoldásokat helyettesíteni. Ezenkívül rendelkezésünkre álló kész szerkezeti egységeket is felhasználhatunk.

A tervezés legfontosabb lépései

Az első tennivaló a testek mozgatóval kapcsolatos fogalmak tisztázása. A test mozgása adottnak tekinthető, ha bármely pillanatban bármely pontjának meg tudjuk adni a helyzetét.

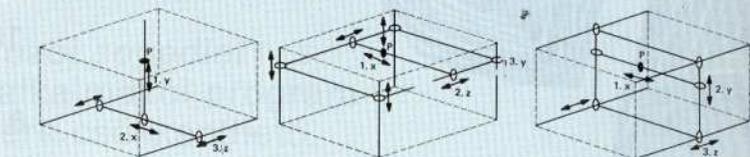
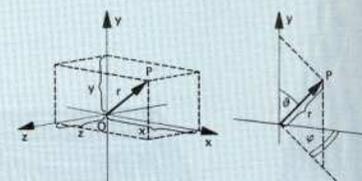
Természetesen a bármely ponton lényeges pontot kell érteni, vagyis a testet is modellezni kell lényeges pontjai alapján. Ilyen testmodell többféle is van, de mi csak kettőt vizsgálunk:

1. a tömegpontot, amelyen olyan idealizált testet kell érteni, amelynek csak tömege van, kiterjedése elhanyagolható;
2. a merev testet, amely a mi szempontunkból a tömegpontnál is fontosabb. Ez olyan speciális pont-

1. ábra

A következőkben bemutatunk néhány példát a lehetséges három adat kiválasztására. Ezek – éppen az előbb leírtakból következően – már döntően befolyásolják a konstrukciót. Ezért a szerkezeti felépítés lehetségeiről is döntően kell megvizsgálnunk e példák jó és rossz tulajdonságait, előnyeit, hátrá-

1. ábra



2. ábra

rendszer, amely tömegpontjainak távolsága állandó.

A mozgások leírását nyugvópont tekintett környezetből az ún. vonatkoztatási rendszer segítségével végezzük. A tömegpont mozgásának leírásához a kinematikai leírásmodot alkalmazzuk. Ennek lényege, hogy a tömegpont helyzetének ismeretéből az idő függvényben következtünk a sebességére és ebből a gyorsulására.

Egy pont helyét egy vonatkoztatási rendszerben a helyvektorral adhatjuk meg. A helyvektor egy kijelölt kezdőpontból a tömegpontba mutató vektor $\vec{OP} = \vec{r}$. Derékszögű koordináta-rendszerben (amely három egymásra merőleges x,y,z tengelyekből áll) P tömegpont koordinátái P(x; y; z), az ún. gömbpólár koordináta-rendszerben pedig P(r; θ ; φ) (1. ábra).

Látható, hogy háromdimenziós térben egy pont meghatározásához három egymástól független adat szükséges. A későbbiekben mutatunk alkalmazási példát az említettek közül másféle adattípusokra is.

A tömegpont pályáján azt a görbét értjük, amelyet a pont mozgása során leír. A pályát a pályaegyenlettel írhatjuk le. Ezt az említett három független változó (f.g.h) esetén a következő paraméteres egyenlet-rendszer adja:

$$\begin{cases} f(t) = \dots \\ g(t) = \dots \\ h(t) = \dots \end{cases}$$

Ez a felismerés fontos tervezési irányvonalakat határoz meg. Tegyük fel ugyanis, hogy csak egy ilyen idealizált tömegpontoszerű test mozgására lesz szükség. Melyik legyen az a három adat, amellyel jellemezzük a pályáját? Nyilvánvalóan azt a hármat érde-

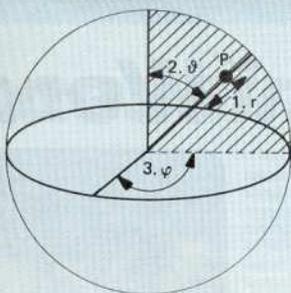
nyaikat. Nem foglalkozunk egyelőre a működtető egységekkel; ezekre a továbbiakban még eljutunk a kinematika részleteit ismertetve.

Fontosabb szempontjaink: a mozgásfüggőség a különböző helyzetű, egyelőre még mindig pontoszerű tekintett testekhez, a megfelelő egyensúlyi probléma ja. Ezen azt értjük, hogy a robot karjának bármilyen helyzetében megőrizhesse egyensúlyát, megadott súlyintenzitétű, egyelőre még mindig pontoszerűnek tekintett testekhez, a megfelelő egyensúlyi probléma ja. Ezen azt értjük, hogy a robot karjának bármilyen helyzetében megőrizhesse egyensúlyát, megadott súlyintenzitétű testek mozgására készben. A mozgató szerkezeteknek esetünkben ugyanis nem az emelés, a súly megtartása az elsődleges feladata, hanem a mozgás. Természetesen a test és a robotkar súlyának bizonyos részét a mozgatószerkezeteknek kell magukra vállalniuk. És most lássuk a példákat:

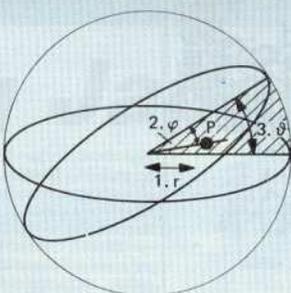
I. típus

A kiválasztott három adat legyen x,y,z (lásd az 1. ábrát). Ez azt jelenti, hogy mindhárom mozgatóegységet egyenes vonalú mozgást kell megvalósítani. Nevezük el ezeket a velük párhuzamos koordinátákról X,Y,Z-nek. A mozgatóegységek nyilvánvalóan egymáshoz kapcsolódnak, megpedig úgy, hogy az első a tárgyat mozgattja, a második az első, vele együtt természetesen a tárgyat, a harmadik pedig az első kettőt.

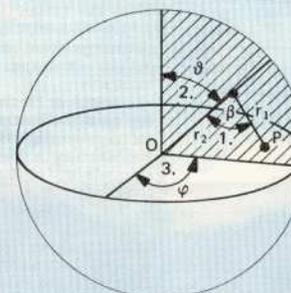
De mi legyen a sorrend? Ez attól függ, hogy milyen jellegű munkára van szükség. A hat lehetséges sorrend közül három különbözőt mutatunk be (2. ábra). Az ábrán bejelöltük a tér azon részét, amelyben aka-



3. ábra



4. ábra



6. ábra

lódó φ vagy θ mozgatóegység forgómozgást végeztet vele. Az egyensúly létrehozására két mód kínálkozik. Az egyik: a forgástengely másik oldalán ellensúly segítségével létrehozni a két forgatónyomaték nagyságának egyenlőségét és irányának ellenkezőségét.

A forgatónyomaték (erőmomentum) $\vec{M} = \vec{r} \times \vec{F}$ melynek nagysága $|\vec{M}| = |\vec{r}| \cdot |\vec{F}| \sin \alpha$ (ez esetünkben θ vagy φ) ahol \vec{r} az \vec{F} támadáspontjába mutató vektor, \vec{F} pedig $\frac{\Delta(m \cdot \vec{v})}{\Delta t}$ -nek definiálható (4. ábra).

A két forgatónyomaték akkor lesz egyenlő, ha az $|\vec{r}| \cdot |\vec{F}|$ szorzat egyenlő lesz az $|\vec{r}_1| \cdot |\vec{F}_1|$ szorzattal. Az egyensúlyra vonatkozó paraméterek 1 indexűek. Mivel $\alpha_1 = 180^\circ - \alpha$, tehát $\sin \alpha = \sin \alpha_1$.

Az ellentétes irány az \vec{r} és \vec{r}_1 ellentétes előjeléből következik. Mivel m és m_1 állandók, csak r -től és r_1 -től függ az egyensúly. Látszik, hogy θ és φ megváltozása nem befolyásolja az egyensúly kérdését.

A szerkezeti megvalósításhál érdemes figyelembe venni, hogy az $\frac{m}{m_1} = \frac{r_1}{r}$ összefüggés felhasználható arra, hogy az ellensúlyt ugyanazzal a meghajtó egységgel (R-rel) mozgassuk, mint a testet.

A másik lehetőség a rugós ellensúlyozás. Ez azonban nagyon bonyolult méretezést és kivitelezést igényel, ezért eltekintünk az ismertetésétől. Meg kell említenünk azonban, hogy kis súlyok esetén a vászercsukat és a θ vagy φ mozgatóegységek megfelelő méretezésével az egyensúlyozó egység elhagyható. A mozgás határa – mivel R-nek bizonyos határon túlnővése esetén nem lehet az egyensúlyt biztosítani – megközelítőleg egy r sugarú gömbre korlátozódik. A „megközelítőleg” a gömb középső részén lévő rész elérhetetlenségéből és az egyes szerkezeti felépítéseknek fellelő, a teljes körbefordulást gátló tényezőkből ered. Meg kell jegyeznünk, hogy a valóságban ilyen II. típusok esetén általában megelégedhetünk egy félgömbnyi térral is.

Vizsgáljuk most a II. típusú θ mozgatóegységet szerint az 5. ábrán látható módon szokták megoldani. Ezzel a következő problémák adódnak. 1. Korlátozott a szögelfordulási érték ($\theta < 180^\circ$), de a legtöbb esetben ennél többre nem is szokott szükség lenni. 2. At kell számítanunk az ábrán jelölt „a” hosszúsági értéket a szögelfordulási értékekre, mivel ezek nincsenek egyenes arányban egymással.

Például az egyenes sebességű mozgásnál a mozgatóegységeknek változó sebességű mozgást kell végezniük. Ez a megoldás azért vált be, mert ebben az esetben olyan mozgatóegységeket is lehet forgómozgás létrehozására használni, amelyek csak egyenesvonalú mozgást tudtak volna megvalósítani.

III. típus

Ezeknek a robotoknak három szögelfordulási adatra van szükségük. Legjellegzetesebb típusuk a 6. ábrán látható. Mozgása az $r_1 + r_2$ sugarú gömbre korlátozódik $r_2 = r_1$ esetén. Ha $r_2 \neq r_1$, akkor egy $|r_2 - r_1|$ sugarú gömbbel kevesebb. Természetesen érvényesek erre a típusra a II. típushoz tartozó mozgáskorlátozónál tett megjegyzések is.

A bemutatott típusoknak sok kivevője ismert, mi azonban csak ezeknek a kivételével fogunk foglalkozni.

Az eddigiekben talán túl sok elmélettel terheltek meg olvasóinkat, ezért most néhány gyakorlati kérdés-felvetéssel foglalkozunk.

Fontos kérdés, hogy robotkarunk mozgásához milyen erőforrásokat használunk. Jó megoldás hidraulikus meghajtóművek alkalmazása, de ilyenek nekünk sajnos nem állnak rendelkezésünkre. Maradnak tehát az elektromotorok.

Ezeknek két, számukra fontos fajtájuk van: a szinkronmotorok és az ún. léptetőmotorok.

A szinkronmotorokkal gyakran találkozhatunk, hiszen ezeket alkalmazzák a kis villanyviszonylatban, motoros modellekben. Előnyük, hogy viszonylag kis méretűek, és működtetésükhöz mindössze két vezetékkel kell elvezetni hozzájuk. Nagy hátrányuk viszont viszonylagos pontatlanságuk. Emiatt és nagy fordulatszám miatt csak nagy áttétel segítségével lehet pontos mozgások megvalósítására használni ezeket.

Léptetőmotorokkal a hétköznapi életben ritkán találkozhatunk, gyakran előfordulnak viszont például nyomtatók, rajzoló meghajtóműveknél. Az a tulajdonságuk, nevezetesen a pontosság, amely miatt jól alkalmazhatók ilyen célokra, alkalmassá teszi őket robotkarok meghajtására is. Ezek a motorok ugyanis egy fordulatot kis lépésekre bontva végeznek el, innen származik a nevük. Az egy lépésre eső szögelfordulási típusonként változik, általában 1 és 3 fok közötti érték.

A léptetőmotorok működési elvükből következően nagyon pontosak, és sebességüket is tetszőlegesen változtathatjuk, mivel ez csak a lépések gyakoriságától függ. Természetesen van azért egy maximális követési időköz. E motorok hátránya viszont, hogy a legkisebb típus is sokszor nagyobb, mint egy normál szinkronmotor, továbbá, hogy vezérlésükhöz típusától függően legalább 6 vezeték szükséges.

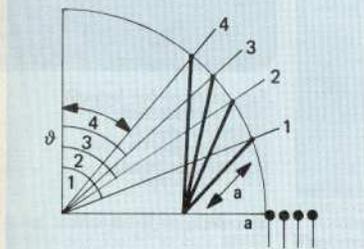
Mint már említettük, a robotok fő alkalmazási területe az ipar. Az ipari robotokra általában jellemző, hogy nagyon rugalmasak, könnyen átprogramozhatók egy munkáról a másikra. Visszacsofolási folyamat nélkül, tünő programjuk szerint végzik munkájukat. Különböző típusaikat ma már nagyon sok cég gyártja, és alkalmazási területük is széles. Többüskük az utvonali egyszerű bejáratásával programozható. Áruk 20–30 ezer dollár között változik.

Az első ilyen robotok gyártóinak egyike, az Animation cég így mutatja be Unimate nevű robotját: „Itt van a fémapír új munkásszáma típusa: nem tart kávészünetet, huszonegy óras munka alatt sem fátad el, nem érdekli a kereset és a nyugdíj. Érzéketlen a porra, füstre, melegre. Ez az ipari robot!”

A másik fejlesztési út a szinkron-manipulátorok kifejlesztése. Ilyen a H. Kleinwächter intézetben kidolgozott Systelmann elnevezésű manipulátor, amely egy irányító személy modulzeitait távolabb, vele egyidőben, nagy pontossággal utánozza.

Ezeknek a fejlesztéseknek távoli célja az ún. szuperintelligens robotegységek kifejlesztése, melyek hűen reprodukálhatók az ember mozgását, és a működésük közben adódó szituációknak megfelelően bizonyos döntésekre is képesek lennének. De létrehozásuk a távoli jövő feladata. Ugyancsak várat meg magára az emberi járást utánozó szerkezetek elkészítése. Létének ugyan már ilyenek, de ezek is csak egy irányító személy mozgását képesek utánozni.

HERENDI MIKLÓS



5. ábra

dály nélkül mozgathatja a testet a robot. Látható, hogy ez valamennyi esetben a fél tér, amit persze csökkentenek a méretezésből eredő hatások.

Például az a) esetben a Z irányú mozgáshoz mindkét oldalán fel kell függeszteni az X irányú mozgatóegységet. Az egyensúly problémája ebben az esetben van a legjobban megoldva: mindenütt a vászercsüket viselik a test súlyát. Mégis az Y irányú mozgásoknak is határt kell szabni. A Z irányú mozgásnál elkerülhető olyan szerkezeti megoldás, amelynél ez a mozgás végtelen hosszúságúvá tehető. Összegezve: a test mozgásának olyan téglai alakú térrész áll rendelkezésére, amelynek Z irányú kiterjedése végtelen.

A másik két esetben hasonló szempontokat figyelembe véve, az ábrán látható a térrész alakulása.

II. típus

A kiválasztott három adat legyen r, θ, φ (1. ábra). Az I. típusal ellentétben itt már csak két lehetőségünk van az R, θ, φ mozgások sorrendjére (3. ábra). A 2. árához hasonlóan itt is megjelöltük a test mozgásának határait. Az R irányú mozgáshoz stabilizálás szükséges, mivel az a mozgáshoz tartozó mozgatóegység csak egy ponton lehet felfüggesztve, hiszen a hozzá kapso-

Word

A számítástechnika alkalmazásának nagyon sok olyan vonatkozása van, ami az emberi kultúra és írásbeliség nemzeti sajátosságaihoz kapcsolódik. Különösen igaz ez olyan ország esetében, mint Japán. Cikkünk szerzője hosszabb időt töltött a szigetországban ösztöndíjas tanulmányúton. Úgy gondoljuk, igen tanulságosak a japán írásrendszer számítógépes vonatkozásaival kapcsolatos tapasztalatai.

„Nekem ez kínaiul van” – szoktuk mondani Európában, ha valami nagyon érthetetlen számunkra. Valószínűleg a Nanzan Egyetem számítóközpontjában tanuló egyetemisták diáktársaim egy része is így kiáltott volna fel a számítógépterminálokon megjelenő sok kínai írásjel láttán, ha ezek a diákok töréntenesen nem a környező délkelet-ázsiai országok valamelyikének kínai kolóniáiból érkeztek volna a japán egyetemre.

Hasonló szövegeket ugyanis gyakran láthattunk a terminálokon, tekintve, hogy az elmúlt években Japánban is rendkívül gyors fejlődésnek indult a számítógépes szövegfeldolgozás, és ezen belül a szövegfeldolgozás. Hasonlóan az USA-hoz, ahol ez a terület a számítógép-alkalmazások legdinamikusabban fejlődő területévé nőtte ki magát néhány évvel ezelőtt.

A Japánban „word processing”-nek nevezett szövegfeldolgozás azonban tartalmilag nem azonos azzal, amit az USA-ban értenek ezen a terminológián. Japánban ugyanis jelenleg háromféle írásrendszert használnak: a kandzsi, a hiragana és a katakana rendszereket.

Az előbbi írásrendszer elemei önálló jelentéstartalommal rendelkező, kínai eredetű írásjelek, az utóbbi kettő elemei pedig a japán nyelvben használt szavak fonetikus írásjelei. A kandzsi jelkészlet elemeinek száma több tízezer, amiből 1850-et tanítanak meg a kötelező kilencéves oktatás során. A hiragana és a katakana rendszer alapjeleinek száma 50.

A normál japán szövegekben mindhárom írásrendszert egyformán gyakorisággal használnak. Az önálló jelentéstartalommal rendelkező szavakat (tehát az igéket, a főneveket, a mellékneveket stb.) kandzsikkal jelölik; a ragokat, jeleket, képzőket hiraganakkal; az idegen szavak fonetikus átírá-

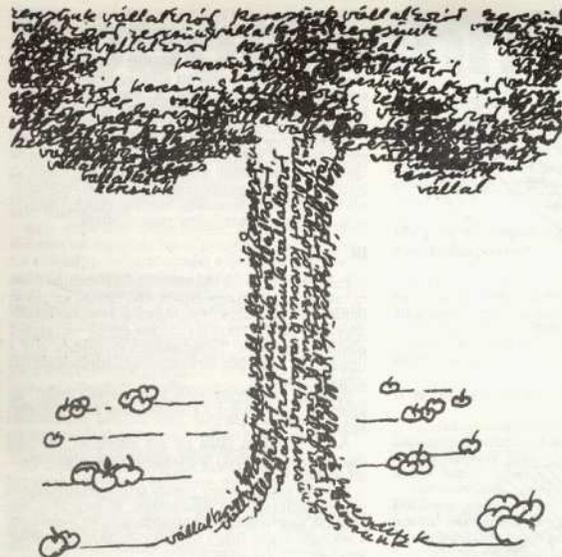
Egy széles körben használt japán wordprocessor, a korszerű gépírás elengedhetetlen számítógépes eszköze

sára pedig a katakanákat használják.

Bár a hiragana nyelvkészlet 50 alapjellel (tehát 50 alapszótaggal) minden japán szöveg leírható, a gyakorlatban ezt az egyszerű írásmódot mégsem használják. A kandzsikkal ugyanis tömörebben és érthetőbben tudják kifejezni a gondolataikat. Igaz ugyan, hogy ennek megértéséhez sokszor a japán – és esetenként a kínai – kultúrtörténet ismeretére is szükség van.

A **n** kandzsi például eredeti formájában kör alakú volt évezredekkel ezelőtt, csak később vált szögletesé. Így érthető, hogy az írásjel jelentése: *nap*. A * kandzsi jelentése *fa*, a # kandzsié *liget*, a n kandzsié pedig *erdő*.

Ha a *fa* alsó részét kiemelem, akkor a * kandzsi kapom, amelynek a jelentése: *gyökér* vagy *eredet*. Ha a *fát* és a *napot* kombinálom, egy új kandzsi kapok, aminek a képe a következő: **☀**. Ennek jelentése: *kelet*, hiszen ke-

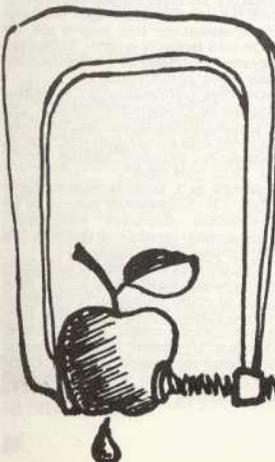


Vállalkozót keresünk SZÁMÍTÁSTECHNIKAI SZOLGÁLTATÓK KATALÓGUSÁ-nak

egyszeri összeállítására
és folyamatos karbantartására

A katalógus nyújtson egyszerű tájékoztatósi lehetőséget a számítástechnikai tevékenységi körökhöz rendelt

- a szolgáltatók által eddig elvégzett tevékenységről,
- a referencialhelyekről,
- az értékesíthető, ill. az adaptálható munkákról,
- a kapacitásadatokról, ill. a vállalkozási készségről.



A katalógus összeállítására vonatkozó ajánlatokat, a tartalmi, formai kivitelre és a vállalkozás körülményeire vonatkozó elképzeléseket kérjük a
1502 Budapest, Pf. 129
címrre beküldeni.

ÉLELMISZERIPARI
ÜGYVITELSZERVEZÉSI
ÉS GÉPI ADATFELDOLGOZÓ
VÁLLALAT

Telex: 22-6035

Processing Japánban



Amit sohasem tud pótolni a számítógép, az a művészi kalligráfia vonalvezetésének sajátos karaktere

Ezek mellett az érthetőséget megkönnyítő előnyök mellett a kandzsi írásrendszernek egyéb előnyei is vannak. A japán nyelv sajátossága, hogy nagyon sok azonos kiejtésű, de különböző jelentéstartalmú szót tartalmaz. Ezeket a jelentésbeli különbségeket tehát csak a kandzsi írásjelekkel lehet szövegekből lehet felismerni.

Az egyik leggyakoribb japán személynévnek, a „Hirosi”-nak a jelentése például akár ötféle is lehet, attól függően, hogy milyen kandzsi használnak a szó leírására (lásd az 1. ábrát). Ha ezt a szót latin betűs átírásban vagy a fonetikus szótagírások egyikével, például a hiraganával íránk (kiejtés: hi) (kiejtés: ro) (kiejtés: si) alakban, akkor ezek a kandzsival kifejezhető jelentésbeli különbségek mind érthetlenné válnának.

裕 浩 宏 弘 寛

gazdag erős nagy széles jövedvű, nyíltszívű

1. ábra

2. ábra

き	ょ	ん	の	く	が	つ	に	に	ほ	ん	へ	い	ま	し	た	。
去	年	の	九	月	に	日	本	へ	行	き	ま	し	た	。		
múlt	év	nek	a	9.	havá	ban	Japán	ba	utaz	tam						

A múlt év szeptemberében Japánba utaztam.



billentyű, amely jelzi, hogy a képernyőn megjelent karakterek (kandzsi) közül melyiket kell kiválasztani.

Befejezésül leírok egy teljes japán mondatot a hiragana jelekkel, ahogy azokat a wordprocessorokba betáplálják. Utána meg adom azt a mondatot, ahogyan az nyomtatásban megjelenik (kandzsi és hiragana speciálisan ragokra), majd alálok írom az egyes jelek magyar megfelelőjét (2. ábra). A mondat fordítása: A múlt év szeptemberében Japánba utaztam.

Ebből a példamondatból látható, hogy a japán nyelvű mondat leírásához csupán 27 leütésre van szükség a japán wordprocessoron, amelynek klaviatúrája alig nagyobb, mint a Magyarországon használt irógépeké, hiszen lényegében csak az 50 alapelv van rajta. A mondat magyar megfelelője csak 42 leütéssel írható le.

A példamondatból azonban más is kiderül. Nevezetesen az, hogy a japán és a magyar mondat szerkezete nagyon hasonló. A szórend teljesen azonos; egyik nyelvben sincsenek prepozíciók, csak jelek, ragok és képzők stb. Persze ez a jelenség nem is igazán meglepő, ha belegondolunk, hogy mindkét nyelv az ural-altaji nyelvek közé tartozik.

A két nyelv szerkezetének összehasonlító elemzése túlműtatta ennek a cikknek a kereteit. Az azonban talán már az eddiekből is kiderült, hogy nekünk, magyaroknak a japán mégsem egészen „kinaiul van”.

leten látható ilyen jelenség, hogy a nap jön fel az égére egy fa mögött.

Ha a 日 és a * kandzsiakat vonom össze egymással, akkor a 日本 jelkombinációt kapom, aminek a jelentése: Japán. Japán ugyanis a „Nap Eredetű Ország”; a legendák szerint az első császár Ameterasz Omikaminak, a Napistennek volt az unokája.

A kandzsi jelentése város, esetleg főváros. Ennek alapján érthető, hogy 東京 Tokiót jelent, szó szerint Keleti Fővárost, mivel Japán három állandó fővárosa közül ez van a legkeletebbre.

Ha már most a „kelet” írásjelét ismerjük, akkor tanuljunk meg még két másik égtájat is: 北 jelentése észak, 南 jelentése pedig dél. Mivel Kinában ugyanezt az írásrendszert használják, ezután már azt is megértjük, hogy Peking írásjelei 北京 Északi Fővárost, Nankingéi pedig 南京 Dél Fővárost jelentenek.

A fentiekből tehát kitűnik, hogy az ideogramma írásmódnak több előnye is van a betűkből felépített írásrendszerekhez képest. Hátránya viszont az, hogy a több ezer jelet 10–15 évi tanulással lehet csak elsajátítani, az írógéppel való írás pedig a több ezer jel miatt hihetetlenül lassú és nehézkes. Ez utóbbi probléma megoldására fejlesztették ki a japán wordprocessorot.

Ennek a szoftverrendszernek a feladatköre azonban – amint erre már utaltam – egyáltalán nem azonos az amerikai értelmezés szerinti wordprocessor feladatkörével. Tekintve, hogy a japán szövegekben sohasem jelölik a szóközöket, hanem folyamatosan írják a jeleket egymás után, a nyomdakész szövegek hasábfelületének kialakítására például egyáltalán nem kell gondot fordítani.

A rendszer lényege abban van, hogy a szöveg írásakor a fonetikus szótagírás-jeleket (tehát a csu-

pán 50 alapeljből álló hiraganákat vagy katakanákat) billentyűzzük be a gépbe, majd ennek alapján a rendszer megkeresi a számítógép tárolójában tárolt több ezer ideogramma közül az adott jelsorozat-hoz tartozó kandzsiakat. Azokat ezután mind kiírja a képernyőre, ahonnan könnyen kiválasztható a megfelelő, és ez ezután bekerül majd a végleges szövegbe. Gondoljunk például a már említett Hirosi esetre.

Bármilyen hihetetlennek is tűnik, egy ilyen wordprocessor alkalmazásával gyorsabban elő lehet állítani a háromféle írásrendszer jeleivel egy nyomdakész japán nyelvű szöveget, mint ha például angol fonetika szerint, latin betűs írógépen gépelnék le. A Hirosi név latin betűs írásmódja esetén például 9 billentyűt kell leütni az írógépben: egy betűváltás a nagy kezdőbetű miatt, hét betű és egy szóköz – a japán wordprocessoros írásmód esetén pedig csak ötöt: három szótagbillentyű: っし, egy karaktermegjelenítő billentyű és egy

M PROGRAMOK

COMMODORE VC-20 Finom felbontású grafika

Annak ellenére, hogy a VC-20 BASIC nyelvjá-
rása semmilyen rajzoló utasítást nem tartalmaz,
mégis lehet vele némi ügyeskedéssel rajzolni.
Két példán mutatom be, hogyan oldható ez
meg BASIC programból.

Rajzolás botkormány segítségével

Az 1. programot betöltés után RUN-nal in-
dítsuk el. A képmező mérete kisebb lesz, meg-
változik a keret színe, megjelenik egy piros pont
a képmező közepén. Ebből kiindulva a botkormá-
ny mozgásával vonalakat rajzolhatunk
nyolcféle irányban. Ha megnyomjuk a lövögombot,
a rajzolópont láthatatlanul elmozgatható.
Így szaggatott vonalat is tudunk rajzolni,
és törölni is tudjuk a már ábrázoltakat.

Ha megújunk a rajzolat, nyomjuk meg a SPACE
billentyűt. Ekkor a képernyő ismét alapállapotba
kerül, és egy kérdőjel jelenik meg rajta. A program
tőlünk vár utasítást arra, hogy mit csináljon.

Válaszolhatunk az 1., 2., 3. billentyűk lenyom-
ásával. Az első esetben törli az előbb rajzolt
ábrát, és előlről kezdhetjük a rajzolatot. A másod-
dik esetben visszakapjuk az előzőleg készített
rajzot, melyet ott folytathatunk, ahol az előbb
abba hagytuk. A harmadik esetben a program-
ból való kilépésre utasíthatjuk gépünket.

Gépi kódú monitorprogram segítségével a raj-
zolt ábrát ki tudjuk menteni mágneses tárolóra, így
nem vessz el a rajz a számítógép kikapcsolásakor.

1. program. Rajzolás botkormány segítségével

```
10 T=5120:0=7663:R=36864:K=37151
20 POKE786,194:POKEK-1,3
30 PRINT"Q":POKER,17:POKER+1,45:POKER+2,144:POKER+3,21:POKER+5,253:POKER+15,39
40 FORA=1TO16:IFORS=1TO10:POKEO+S*16+A,A*10+S-11:NEXTNEXT
50 POKEX+3,127
60 IF Z=2THENG0
70 FORI=TT00+16:POKEI,0:NEXTX=64:Y=00
80 G=PEEK(K):X=X+(GAND16)=0:Y=Y+(GAND4)=0:Y=Y-(GAND8)=0
90 IFPEEK(197)=32THENI80
100 X=X-(PEEK(K+1)AND128)=0:Y=Y-(GAND32)=0
110 IFX<1THENX=1:GOTO100
120 IFX<126THENX=126:GOTO100
130 IFY<1THENY=1:GOTO100
140 IFY<150THENY=150:GOTO100
150 IFQ=1THENI70
160 W=T+INT(X/8)*160+Y:POKEW,PEEK(W)OR2+(7-XAND7)*100TO80
170 W=T+INT(X/8)*160+Y:POKEW,PEEK(W)AND255-(2*(7-XAND7))*100TO80
180 POKER,12:POKER+1,38:POKER+2,150:POKER+3,174:POKER+5,240:POKER+15,27:POKEK+3,
255
190 PRINT"Q":INPUTZ:IFZ=1ORZ=2THEN90
200 IFZ=3THENEND
210 GOTO100
READY.
```

A program készítése

Ilyen program készítéséhez ismernünk kell a VC-20 memóriatérképet. Észrevehetjük, hogy a \$1DFF című lufelé a sztringváltozók és ma-
guk a sztringek részére, továbbá a tömbök ré-
szére fenntartott helyek eredeti funkciójukban
nem kellene a rajzolásához, tehát nyugodtan
igénybe vehetjük ezeket a megszorítással,
míg nem iratunk ki betűket a képernyőre, nem
végzünk semmilyen sztringműveletet és nem
használnuk a programban tömböket.

A képernyőt apró cellákra kell osztanunk.
A cellákból vízszintes 16 darabot, függőle-
sen 10 darabot kell elhelyeznünk ahhoz, hogy
megközelítően négyzet alakot kapjunk, mivel
egy cella maga is egy 8 x 16 pontból álló mátrix.
Így egy 128 x 160 pontból, azaz 160 cellából
álló képmezőt kapunk.

Ehhez a képernyőtár címeinek értékét is át
kell szervezni, mégpedig a kezdő címűl (\$1E00)
egészben a \$1E95 címig.

A fentebb említett sztring- és tömbtárolók címei-
nek tartalmát is törölni kell (\$1400-\$1DFF) indu-
lásnál és akkor, ha új rajzot kívánunk készíteni.

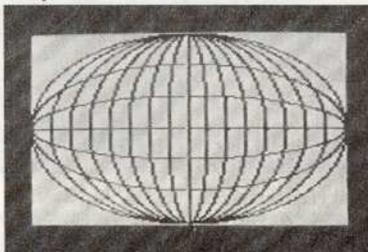
A botkormány értékeinek lekérdezése sokkal
bonyolultabb, mint a Commodore 64-nál. Az
alábbiak szerint kell eljárni:

1. At kell állítani a bemenetet ellenőrző tár-
részből a \$9122 rekesz értékét 127-re, hogy a
botkormány vezérlőutasításait a gép fogadja.

2. Állandóan le kell kérdeznünk a \$911F és
\$9120 tárrékeszek értékeit, ugyanis ezeket állít-
ja át a botkormány. Pontosabban a \$911F tár-
rékesz 2. bitjét a felfelé mozgató, 3. bitjét a
lefelé mozgató, 4. bitjét a balra mozgató, 5.
bitjét a lövögomb, a \$9120 rekesz 7. bitjét a
jobbra mozgató állítja át.

bit	7	6	5	4	3	2	1	0
\$911F	-	-	lövés	balra	le	fel	-	-
\$9120	jobbra	-	-	-	-	-	-	-

1. kép



A billentyűzet lekérdezését a SC5 tárrékesz
végzi. Ezt használjuk fel a SPACE billentyű
figyeléséhez.

A képmezőből nem szabad a botkormányt
vezérelve kilépni, mert ezeket a képmező-
tékeket a program a már többször említett
RAM területen tárolja. Amennyiben ebből ki-
lépnénk, lefelé belesorognánk a változó-, majd a
BASIC program tárolóba, illetve felfelé a képer-
nyőtárba, és ezzel tönkretennénk programunkat.

Ezt a hibát megakadályozhatjuk, ha figyeljük
X és Y értéket, és nem engedjük azokat 0 alá
és 126, illetve 158 fölé emelkedni.

A működés

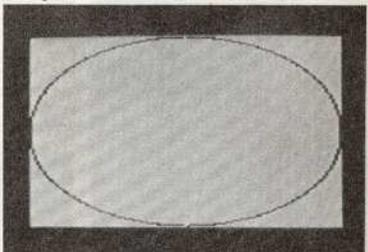
20. sor: Letiltja a RUN-STOP és RESTORE
billentyűk működését, mivel használatuk
szépséghibát okoz

30. sor: Beállítja a képmező méretét és a keret szí-
nét

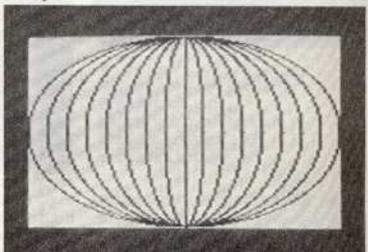
40. sor: Megjeleníti a képernyőtár tartalmát
50. sor: Lehetővé teszi a botkormány állapotá-
nak beadását

70. sor: A lefoglalt tár törlését végzi és a kezdő-
pontot középre állítja

2. kép



3. kép



2. program. Ellipszisek

```
10 T=5120:0=7660:R=36864:K=30720
20 PRINT"Q":POKER,19:POKER+1,501
POKER+2,144:POKER+3,32
30 POKER+5,253:POKER+15,26:POKE56,20
40 FORA=0TO15:FORB=0TO15
50 POKED+S*16+A,A*16+S
60 POKED+S*16+A+K,0
70 NEXTNEXT
80 FORZ=TT07167:POKEZ,0:NEXT
90 FORI=64TO9STEP-8
100 FORZ=0TO2+5STEP4/200
110 C=SIN(G)*I+64:DCOS(G)*63+63
120 X=INT(C):Y=INT(D)
130 W=T+INT(X/8)*128+127-D
140 POKEW,PEEK(W)OR2+(7-CAND7)
150 NEXTNEXT
160 FORP=0TO42STEP21
170 FORG=0TO2+5STEP4/200
180 C=SIN(G)*64+64:DCOS(G)*P+64
190 X=INT(C):Y=INT(D)
200 W=T+INT(X/8)*128+127-D
210 POKEW,PEEK(W)OR2+(7-CAND7)
220 NEXTNEXT
230 GOTO 230
READY.
```




Konverziós tábla sztringben

Az első módszer alapja az, hogy a klaviatúrán 121 olyan jel található, amely sztring változóba téve, a lemeze konverzió nélkül felírható. Ha ezeket a jeleket egy 121-es számszerelemnek tekintjük, az eredeti, 10-es rendszerű számokat programmal átkódolva és a lemeze kiírva, az eredeti helyigénynek törtérzésre lesz csak szükségünk. Például ha az ábrázolni kívánt szám 118, ez decimálisan 6 bájtot (előjel, szám, blank, mezőelválasztó) foglal el; a mi számszerelemünk azonban egyetlen grafikus jel képeben, egy bájtban tárolódik.

További sürítési lehetőség, hogy megvizsgálva a rekordban szereplő mezők helyérték-maximumát, a rekordon belül fix helyre és fix hosszúra deklaráljuk őket, kiírás előtt egyetlen sztringbe láncoljuk (konkaténáljuk) az átalakított változókat, és ezt a sztringet írjuk a lemeze.

Ha egy változó kisebb, mint a számára fenn-

tartott hely, 0-val töltjük fel az üres helytérket. A mezők kezdőpozíciója a rekordon belül egy tömbben tároljuk, és íráskor, olvasáskor ezt használjuk a mező eléréséhez.

A sztring módszer előnye, hogy segítségével megtakarítható a 2 bájtos mezőelválasztó jel és pozitív számokat feltételezve az előjelbájt. A számszerelem konverziós táblájának kialakításakor viszont ügyelnünk kell arra, hogy az ne tartalmazza a pi, F1-F7, kurzormozgató, szín, RVS ON/OFF jeleket, mert ezek a sztring vagy a be-kiviteli műveletek közben másképp viselkednek, mint a többi jel. Mi a táblázat kialakításakor 0-val kezdünk (ez a 121-es rendszerben is 0), majd balról jobbra, felülről lefelé haladva, először normál, majd SHIFT, ezután COMM tabuláció mellett ütogetűk végig a billentyűzetet.

Az ismertetett 1. program fő részében a konverziós tábla, a mezők kezdőpozícióit tartalmazó tömb kialakítása és a be-kiviteli műveletek találhatóak. Az 1000-es címtől kezdődik a két-irányú konverziót végrehajtó szubrutin. FL=0 esetében 121-ből binárisra, FL=1 álláskor binárisból 121-esre kódolja a számot.

A bemenő adatok FL=0 esetén:
 CH\$ a sztring, amit átkódolunk számmá.
 H0 a kódozni kívánt mező kezdőpozíciója CH\$-ban,
 H1 a kódozni kívánt mező utolsó pozíciója CH\$-ban.

Az eredményt az SZ változó tartalmazza.
A bemenő adatok FL=1 esetén:
 SZ az átalakítani kívánt szám,
 H1 az eredményül kapott sztring hossza (0-val kiegészítés).
 CH\$-ban a 121-re átkódolt szám található a rutin végén.

2. program

```

100 REM FOPROGRAM
102 DIM HZ$(4,3,2),HZ$(4,3,2)
104 FOR I=0 TO 4 FOR J=0 TO 3 FOR K=0 TO 2: HZ$(I,J,K)=NEXT I: NEXT J: NEXT K
106 FOR I=0 TO 4 FOR J=0 TO 3 FOR K=0 TO 2: HZ$(I,J,K)=0: NEXT I: NEXT J: NEXT K
108 VP$="" V1$="" HZ$="" V2$=2: GOSUB 9000
110
112 REM TRAC LENEZRE VAGY KAZETTARA VP$=BOL
114
116 REM OLVASAS LEMEPROL VAGY KAZETTAROL VP$=BA
118 WHI="HZ$" V2$=1: GOSUB 9000
120
122 REM FELDOLGOZAS
124
126 END REM FOPROGRAM VEGE
128
130 REM *****
132
9000 REM TOMB-STRING KONVERZIO
9002 REM IRASZIRALT VALTOZOK *** VNE="0" VALOS TIP: TOMB NEVE ***
9004 REM *** VNE="000" EGESZ TIP: TOMB NEVE ***
9006 REM *** VP$ STRING NEVE ***
9008 REM *** V1 ADATIRANY 1=IR 2=OLVAS ***
9010 REM *** V2 HIBAKOD ***
9012 REM *** V1 V2 TOMBOK KEZDO ES VEGTIME ***
9014 REM *** VE,VV ADATOK KEZDO ES VEGTIME ***
9016 REM *** V3=1 TOMBNEV NEH DEFINIALT ***
9018 REM *** V3=2 A TOMB TUL NAGY ***
9020 REM *** V3=3 ADATIRANY HIBA ***
9022
9024 V1=0: V2=0: V3=0: V1=0: V3="" VE=0: VV=0
9026 V3=0: V1=ASC(LEFT$(VNE,1)): V2=ASC(MID$(VNE,2,1))
9028 IF LEN(VNE)=> THEN V1=V1+128: V2=V2+128
9030 FOR V1=PEEK(47)+PEEK(49)*256 TO PEEK(49)+PEEK(50)*256
9032 IF PEEK(V1)=V1 AND PEEK(V1+1)=V2 THEN 9036
9034 NEXT V1 V3=1: RETURN
9036 IF PEEK(V1+3)>0 THEN V3=2: RETURN
9038 VE=PEEK(V1+4)+V2*V1+5: VV=V1+PEEK(V1+2)+1
9040 IF V3=1 THEN 9046
9042 IF V3=2 THEN V3=3: RETURN
9044 VP$="" FOR VI=VE TO VV V3=VP$+CHR$(PEEK(VI)): NEXT VI: RETURN
9046 FOR VI=VE TO VV V3=LEFT$(MID$(VP$,VI-VE+1,1)+CHR$(0),1): POKEVI,ASC(V3)
9048 NEXT VI: RETURN
9050 REM STR VEGE
9052
9054 REM *****

```

Ha a rutint konkaténáció nélkül, csak átkódolás kívánjuk használni, FL=0 esetén H0=1: FL=1 esetén a szubrutin meghívása előtt CH\$=""

A módszer alkalmazásával egy 30 mezőből álló, decimálisan ábrázolva 4 db egyjegyű, 10 db kétjegyű, 4 db háromjegyű, 5 db négyjegyű és 7 db hatjegyű számot, mely eredetileg 188 bájtban fért el, 60 bájtos rekordba tudunk elhelyezni, azaz rekordonként mintegy 70 százalékos helytakarékoságot sikerül elérnünk.

Tömb-sztring konverzió

A második módszer annyiban egyezik az előbbivel, hogy a változókat kiírás előtt egy sztringbe gyűjtjük, és ezt a sztringet írjuk ki a lemeze. Ebben az esetben azonban a kiindulási alapunk az volt, hogy a tárban az egész számok 2 bájton, a valóság pedig 5 bájton tárolódnak. Az alábbi szubrutin segítségével ezt a tárolási módot ideális megoldásként a lemezen is tudjuk alkalmazni.

A szubrutin működésének lényege, hogy egy adott tömb adatait a tárban lévő megjelenési formában bájtönként ASCII karakternek értelmezve, ezekből a karakterekből állít elő egy sztringet. A sztring hossza egész típusú tömb esetén a tömb összes elemszámának kétszerese, valós típusú tömb esetén annak ötszöröse. A hossz természetesen független a tömb egyes elemeinek értékétől. A konverzió visszafelé is igaz, tehát a szubrutin egy sztringet visszar az őt előállító vagy az előállított adatainak dimenziójú tömbbe. A rutin természetesen csak olyan sztringet értelmez helyesen, amelyet maga állított elő korábban. Mivel egy vektor egydimenziós tömbként is felfogható, a konverzió vektor esetén is érvényes. A tömb dimenziójára a BASIC szabályai érvényesek. A szubrutin meghívása előtt a következő változóknak kell értéket adni:

VN\$ a konvertálandó tömb neve és típusa.
 VPS a sztring neve, olvasás esetén értéke="", VD\$ adatrány, a tömbre vonatkoztatva.
 Ezek a változók kötések, illetve csak a rutin módosításával változtathatók. A szubrutin hibavizsgálatot végez, és ha hibát talál, azt a VH% nevű változóba adja vissza. VH%=0 esetén nincs hiba.

A közölt BASIC lista a szubrutint és az azt kipróbáló főprogramot tartalmazza (2. program). A szubrutin gépi kódú változatát természetesen lényegesen hatékonyabban használhatjuk.

Az adatfeldolgozó rendszerekben általában egy-egy több dimenziós tömbbe tárolva, tapasztalataink szerint ez a szubrutin igen egyszerű rekordkezelést tesz lehetővé, a lemeze tárkapacitását pedig - hardverbeavatkozás nélkül - lényegesen megnöveli. A mi rendszerünkben a 95 mezőből álló, mezőnként átlag négyjegyű számot tartalmazó rekord hagyományos tárolási forma esetén $95 \times 8 = 760$ bájttal hosszú lenne. Az ismertetett tárolással ezzel szemben mindössze $95 \times 2 = 190$ bájtra van szükség, ami a lemeze kapacitásának négyszeres növekedését jelent.

Végül néhány gyakorlati megjegyzést fűzünk a szubrutin használatához. A konvertálandó tömböt valós vagy egész típusúként deklarálhatjuk a legelső DIM utasításban, ezáltal a tömb nevét kereső FOR ciklus rövidebb. Tömbből való olvasás előtt a VPS sztringváltozó kezdeti értéke VPS=""

GOSZTOLA FERENC-VÖLGYI CSIK JÓZSEF

Építsünk számítógépet! VIII.

Ebben a részben eredetileg a központegység-kártya ismertetését akartam elkezdni, de engedve a nagyszámú olvasólevélben szereplő kívánságnak, a gép teljesítőképességével foglalkozom. A minimális, egy közepes, egy teljes kiépítésű változatot, valamint amerikai cégek által az SS-50 buszhoz (ami a mi gépünkben is van) elkészített fejlesztéseket foglalom össze.

A **minimális változat** az alapkártyából; az 1 k RAM-ot tartalmazó videokártyából; a 6803-ban levő 128 bájt RAM-on kívül más RAM-ot nem tartalmazó, egy 2716-ban tárolt ROM monitoron kívül más ROM-ot nem tartalmazó központegység-kártyából; a billentyűzetillesztőből; egy tévévevőből és egy közönséges magnetofonból áll.

Ennek a változatnak 64 × 16 karaktert megjelenítő képernyője, 110–2400 Baud átviteli sebességű magnetofon-háttértára (jó minőségű magnetofon és kazetta esetén 4800 Baud) van. A 2 k ROM és 128 bájt RAM (+1 k a képernyőnek) csak kis tárgyényű gépi kódú programok tárolására elég.

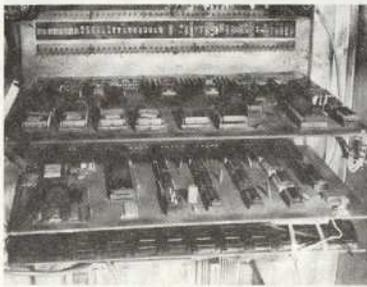
Elősegés a rendszert kiszolgáló SIMON COS (kazettás operációs rendszer) tárolására, melynek parancsai:

- két változatú programfuttatás,
- BASIC hideg indítás,
- memóriavizsgálat, -módosítás,
- karaktersorozat (48 karakterig) kikeresése,
- regisztervizsgálat,
- automatikus programáthelyező, adatáthelyező, címmódosító parancs egyidejűleg legfeljebb 8 tömb módosítására, írásvédelem figyelembevételével,
- tárterület-törlés,
- töréspont be/kiiktatás,
- csatornakiválasztás,
- tárkiírás,
- adatbeolvasás magnetofonról, beállítható sebességgel,
- két tárterület tartalmának összehasonlítása,
- adatkivétel magnetofonra, beállítható sebességgel,
- időkéskeltetés kiadása,
- a 6803 belső RAM-területének be/kikapcsolása,
- videovezérlések (kezdeti feltételek beállítása, fénypont fel-le, jobbra-balra, sor/lap elejére, törlés),
- megszakításkezelés.

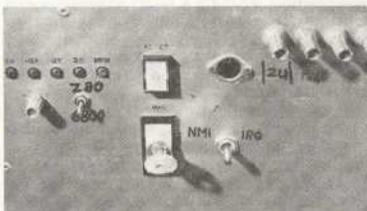
Minden parancs hívható szubrutinként is. Ez a változat alkalmas a rendszer működését vizsgáló programok tárolására is.

Az **átlagos kiépítésű változat** ugyanezeket a kártyákat tartalmazza, de a videokártyán 6 k RAM, a központegység-kártyán 16 k ROM és 16 k RAM (a kártya módosításával 32 k ROM és 64 k RAM is megvalósítható) van.

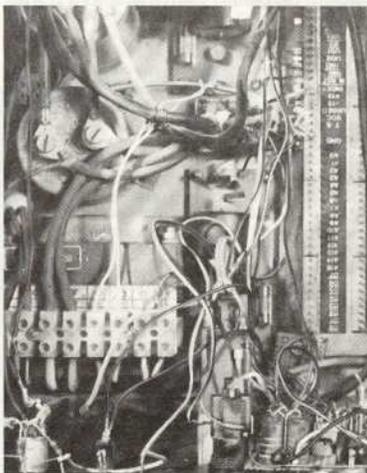
Ennél a változatnál a képernyő grafikus is lehet 256 × 192 pont felbontással, és monitor használata esetén 80 × 24 karakterrel.



1. ábra



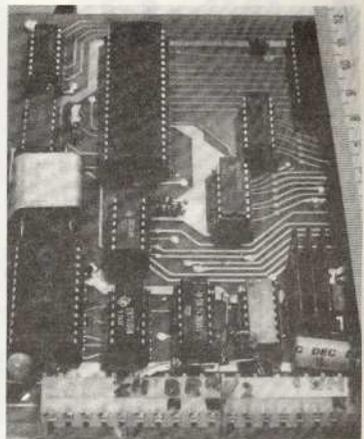
2. ábra



3. ábra

A géphez 4, 8, 11 k helyfoglalású BASIC-fordítók állnak rendelkezésre. Programfejlesztéshez assembler, disassembler, ún. debugger (ez az előző kettőn felül például intelligens szimulátort, töréspontokkal számos paramétert, vizsgálat is tartalmaz) van.

Teljes kiépítés. A gép bővíthető 16 k RAM befogadására tervezett, 16 csatornás A/D átalá-



4. ábra

ító-, soros illesztő-, párhuzamos illesztő-, magnetofonkezelő-, hajlékonytárcsás lemezegység kezelő kártyákkal. A központegység-kártya lehet 6800, 6809, Z80 mikroprocesszorú is. Ezek közül bármelyik kettő egyidejűleg is benne lehet a gépben, mint ahogy az 1977-es gépünkről készült, 1. képen látható.

A két központegység-kártya közül az aktív (a másik letiltása mellett) egy egyszerű kapcsoló választja ki, ahogy az ugyanannak a gépnek két oldaláról készített 2. és 3. képen látható. Ez utóbbin követhető, hogy a két kivezetéses kapcsoló egyik vezetéke az NMI kapcsolóhoz, a másik a busz UDI-vonalához vezet.

Az átkapcsolás a mikroprocesszort a buszról lekapcsoló lábra adott jellel történik (4. kép). Mi ezt az egyszerű megoldást – amelyet utánnuk mintegy másfél évvel kezdett el gyártani a Microsoft cég az Apple II típusú géphez, SOFTCARD néven, és amelynek különböző változataiból már millióan felülüli darabszám készült – szabadalmaztatni kívántuk. **Megvalósíthatatlanság** indokával azonban kérelmünket elutasították.

Az **SS-50 buszhoz kapható egyéb fejlesztések:** a 6800, 6809, 6803 kártyáink lehetővé teszik a FLEX és az OS9 operációs rendszerek használatát, a Z80 kártya pedig, monitorával együtt a CP/M 2.2 változatát képes használni. Ezeket keresztül gyakorlatilag mindenféle szoftver elérhető.

A hardverlehetőségek között megemlítem a 256 k DRAM, a mindenféle lemezegységillesztő, a nagy felbontású (1024 × 512 képpont) grafika és a folyamatmérő/vezérlő kártyákat, és végül, de talán ez a legfontosabb, a 68008 és 68000 típusú mikroprocesszoros kártyákat.

Mi már építünk!

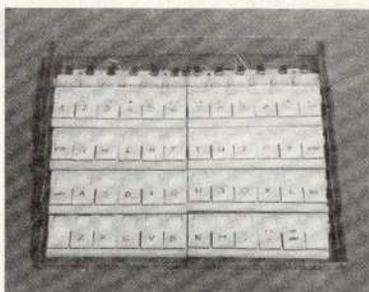
Mint már beszámoltam róla, elkészült épülő számítógémem billentyűzete. De első próbálkozásra ritkán sikerülnek jól a dolgok, és sajnos ebben az esetben is ez történt. Legelőször csak néhány forrasztási hibára derült fény, amelyeket hamar kijavítottam. Nem ez okozta a fő gondot. A gyakorlat ugyanis azt mutatta, hogy bármelyik billentyűt nyomom le, mind a tizenhárom vezetéken folyik áram.

Sokáig kerestem a hibát. Először arra gondoltam, hogy a vezetékek érintkeznek, de más-hol volt a baj: a földelésnél. Kiderült, hogy amikor az egyes vezetékeket leföldeltem, nem akadályoztam meg az áram átfolyását a többi vezetékre. Néhány dióda alkalmazásával megszüntettem a hibát. Előbb ugyan egy másik NYAK-ot kellett készíteni, de ez már nem okozott különösebb nehézséget.

A kész billentyűzetet a képen láthatják. Az érintkezők – vasútmolekulákkal foglalkozó olvasók biztosan felismerik – az ún. váltóirányító pult kapcsolói. Az egyes billentyűk kódolására diódamátrixot alkalmaztam.

Az illesztőt nem számítva, mindössze 700 forintba került a billentyűzet, amiben szerepe van annak, hogy a hat érintkezőt tartalmazó kapcsolópult ára csak 40 forint volt.

Végül az illesztőrésztől néhány szót. Használatát az indokolja, hogy a számítógépeknek csak 8 adatvezetéke van arra a célra, hogy kommunikáljanak vele, míg mi a karaktereket 13 biten kódoltuk. A billentyűzetillesztő tehát azt a célt



szolgálja, hogy a mikroprocesszornak alkalmas időben, számára feldolgozható információt küldjön.

Ez a következőképpen történik. A kódot egy 6 és egy 7 bitből álló részre választja szét a rendszer, és először a kód első 7 bitjét küldi ki, majd pedig a második 6 bitet. A szétválasztást a 74LS157-ek végzik, a 74LS75-ök pedig a felhasználásig tárolják az információt. A rendszer vezérlését a 7472 végzi, a 7404 csak késleltetésre van.

Mivel legfeljebb 7 adatvezetéket használunk, a 8. vezetéknek kúrdhatjuk ki az engedélyező jelet.

Gondolom, sokaknak feltűnt, hogy a billentyűzetilestő elvi kapcsolási rajza hasonlít a dr. Simonyi Endrének a lap 1984/2. számában közölt cikkében szereplő rajzhoz, de sokban eltér tőle. A magyarázat az, hogy az itt közölt rajz nemcsak a rendszer vezérlését tünteti fel, hanem a célnak is jobban megfelel, mivel ennél a megoldásnál egyszerűbb szoftverrel lehet a kezelést végezni.

GATTYÁN ANDRÁS

Kazetta-szabvány

A mikrogepek gyártásának már kezdetén látható volt, hogy a termékek csereszabattósága sokkal kevésbé lesz biztosított, mint a nagybobb gépeknél. Ott sokkal kevesebb cég volt a piacon, és volt egy, az IBM, amely a hardver- és szoftverválasztékot döntő módon meghatározta. Itt azonban sok kis cég versenyzett, eltérő hardver- és szoftvereképzésekkel, eszközökkel.

A hardverpiacon igazán sikeres szabványosítási kísérlet nem volt, mert mire megszületett az egyébként is laza, S-100 szabvány, már a mikrogepek mintegy 80 százalékát a Tandy, a Commodore és az Apple cégek gyártották, amelyeknek egyike sem fogadta ezt el.

Sokkal eredményesebbek voltak a szoftvertermékek egységesítésére irányuló kísérletek. A lemezooperációs rendszereknél a Digital Research CP/M-je ugyan sok változatban készült el, és a „három nagy” egyike sem használta, de valamennyi, tömeggyártásba került géphez elkészült a szoftver- és hardverillesztés. Ez utóbbi a 6502 típusú mikroprocesszorú gépeknél Z80-as központiségű-kártya beépítésével és az aktív központi egység kapcsoló kiválaszthatóságával járt együtt. Az első ilyen, kereskedelmi forgalomba került kártya a Microsoft cég SOFTCARD nevű kártyája volt.

A BASIC-fordítók között a Microsoft cég róla elnevezett fordítóinak különböző változatú gyártották a „nagyok” is. Ezek azonban sok közös tulajdonságuk ellenére egymás programjait nem értik. Még az azonos elnevezésű és jelentésű utasításokat, függvényeket sem, mivel ezekhez különböző kulcsszókódok tartoznak.

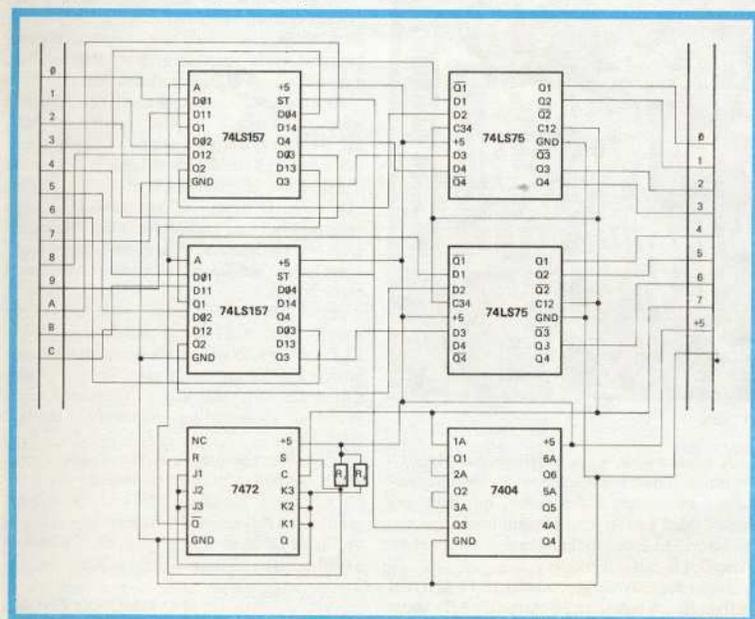
Ez a félig sikerült szoftveregységesítés is csak a 80-as évek elejére jött létre, de a zűrzavar már kezdetől fogva megvolt. Ennek csökkentésére a Byte című szaklap összehívott egy konferenciát, amelyen a gyártók (akkor még a „három nagy” egyike sem gyártott mikrogepet) és a főbb felhasználók megkísérelték az álláspontok közelítését. Az egyetlen, amiben megállapodtak, a kazettajel volt. Az erre vonatkozó szabvány, a konferencia színhelyének nevére, mint Kansas City-szabvány vált ismertté, és bár eleinte elterjedt, a lemezenszerek elterjedésével fokozatosan visszaszorult. Mai jelentősége azonban, adatok átvitelét gépről gépre.

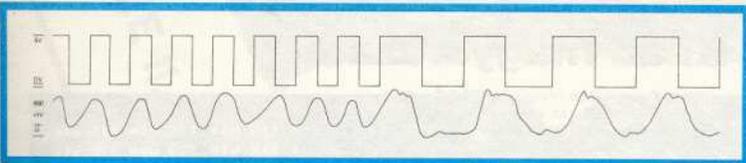
Mit tartalmaz ez a szabvány? Előírja, hogy

- a jel négyzög alakú periodikus, tehát két feszültséginté, azaz bináris,
- az információhordozó egy megadott idő alatti periódusszám, konkrétan 1/300 s alatt négy (ez a 0) vagy nyolc (ez az 1) periódus (1. ábra),
- a megengedett frekvenciaingadozás $\pm 30\%$ vagyis a magnetofon sebességingadozására gyakorlatilag érzéketlen,
- az amplitúdóingadozás érdektelen,
- a jelformázás is érdektelen. Ez utóbbi kettő oka az, hogy az igazi információ csak a feszültségváltozás irányváltozásainak száma. Így például az ábrán látható torzulása a jelsorozatnak 1:5 amplitúdóingadozás mellett sem okozott problémát egy korábbi gépünknek.

A szabvány nagy előnye a könnyű megvaló-

A billentyűzetilestő elvi kapcsolási rajza





```

5 REMI:PROGRAM:KC-TOELTOE
10 CLS: CLEAR500,14335:A=14346:PR
INT"BEHELYEZENDO A KAZETTA! A M
AGNETOFONT OLVASAASRA KELL AALIT
ANI!" :B#="HA KEESZ ADJON BE EGY
SZAAMOT!" :PRINT#: INPUT:POKE&H3
803,0:POKE&H3802,57
20 FORI=0TO107:READA#:POKEA+I,VA
L("H"&A#):NEXTI:MOTORON:EXEC&H3
80A:E=256*PEEK(&H3802)+PEEK(&H38
03):PRINT"VEEGCI M":I
30 E=256*PEEK(&H3802)+PEEK(&H380
3)
40 PRINT"A BEOLVASAAS KEESZ! CSE
REELJEN KAZETTAAT!" :PRINT#: INPU
TU: INPUT "PROGRAMNEEV" :C#:CSAVE
M C#,&H4C00,E,&H4C00:STOP
50 DATA,38,02,8D,0D,A7,80,CE,0
4,08,A7,C4
70 DATAF,38,02,20,F2,39,36
170 DATA04,8D,2D,2F,FC,8D,29,8D,
27,2D,F6,C6,08,F7,3C,08
180 DATAF,8D,11,44,8D,30,2E,02,
9A,80,8D,08,7A,30,08,2E
190 DATAF2,37,04,39,1F,12,8E,01,
04,30,82,26,FC,1F,21,39
200 DATA36,02,4F,4C,F6,FF,20,C4,
01,F1,38,01,27,F5,F7,38
210 DATA1,80,0E,37,02,39,F6,FF,
20,C4,01,F7,38,01,8D,E8
220 DATA8D,2E,2E,04,8D,DA,8D,DB,
39
    
```

1. lista

síthatók olcsó eszközökkel, olcsó illesztő, magnó, szalag. Hátránya a lassúsága: átviteli sebessége mindössze 300 Baud. A kis sebesség és torzításérzékenység lehetővé tette a programok továbbítását telefonvonalakon is magnetofonhoz és vissza.

Az eddigiek alapján a KC-illesztő hardver és szoftver egyszerűen megvalósítható és jól használható eszköz. Az íróhardvernek képesnek kell lennie TTL-szintű, majd ezekből a magnó által venni képes jelek létrehozására. Ehhez egy párhuzamos csatorna egy bite, egy 7407, néhány ellenállás és kondenzátor elég. Az olvasónak a magnó által torzított jelet kell TTL-jellé alakítania, majd azt a mikroprocesszorhoz továbbítani. Ehhez az előző elemek azzal az elbárállító elemekkel, hogy a 7407 helyett egy összehasonlító elem (például LM 339) szerepel.

Az olvasószoftver a kapott logikai 0, 1 sorozatokból a küldött bitsorozatokat kell, hogy azonosítsa és tárolja. Ennél többet egy univerzális olvasóprogramtól nem várhatunk, mert még azt sem sikerült szabványosítani, hogy megjelenjen a bitek. A legelterjedtebb változatnál ún. 8 bites adatot, start-, stopbiteket (esetleg párossággellenőrző biteket) visznek át. A startbit (logikai 1) jelzi az adat elejét, melyet követ az adat, végül a stopbit (logikai 0). Az adatbajt általában hexadecimális (tizenhatos számszendszerben) számként átvitt ASCII-kód. Ez azt jelenti, hogy a bajtót két 4 bites (azaz hexadecimális) száma bontjuk, majd ASCII-karakterre

alakítjuk. Ez utóbbinál, ha a hexadecimális számjegy 0-9, akkor hozzáadunk 48-at, ha pedig A-F, akkor 65-öt. Az olvasónál természetesen a folyamat fordított.

Ilyen típusú adatok beolvasására szolgáló BASIC program látható az 1. listában. A programot DRAGON típusú gépre írtam. A DRAGON-BASIC egy kiterjesztett, szines Micro-soft-BASIC, amelynek különleges tulajdonságait alig használtam ki, azért, hogy a program más Micro-soft-BASIC-kel rendelkező gépeken (például HT-1080Z, Commodore gépek, Apple II) csaknem változtatlanul futtatható legyen.

A módosításokhoz: a CLEAR utasítás első száma a karaktersorozatok részére foglal helyet, míg a második a BASIC programterület felső határát adja meg; a \$H-jel hexadecimális számot jelöl; az EXEC a gépi kódú program behívása; a CSAVEM utasítás gépi kódú programot visz kazettára névvel, kezdő- és végcímmel, valamint a program indító címével.

A gépi kódú rész természetesen mikroprocesszor-függő. Így az csak 6809 típusnál használható változatlanul. Az átírás a 2. listához fűzött megjegyzésekkel azonban egyszerű. A program a 3900H címűtől tölt, függetlenül attól, hogy a beolvasandók milyen címeken futnak. A DRAGON-nál a beolvasható mennyiség kb. 25 k lehet (a továbbfeldolgozás miatt azonban csak ennek a fele).

A program gépi kódú része a DRAGON-kepernyő bal felső sarkát „ablaknak” használja (Commodore 64-nél ez szintén a 0400H címre töltéssel történik), és a töltés végéig állandósuló jel (ez a később említett formátumnál "9") megjelenésnek kell a magnót leállítani. Ezután RUN 30 paranccsal a beolvasottakat TRS-80 formátumban, amit a DRAGON is használ, kazettára viszi. Ehelyett géptől, kiéptéstől független, saját kazettára, illetve lemezre kivitelét kell itt megvalósítani.

Az így kivitt program visszatöltve általában nem futtatható, még akkor sem, ha a mienként azonos típusú gépen készült, és a mi gépünkön is megszokott formátumban. A szalagra vitt adatokat ugyanis általában sorokra-blokkokra bontva viszik ki, melyek mindegyikét azonosítóval-típusjellel (esetleg kezdőcímmel is), valamint ellenőrzőszámmal egészítik ki. Ezeket értelmezni kell, és figyelembevételükkel lehet csak a programot úgy elhelyezni a memóriában, hogy az futtatható legyen.

Ennek megvalósítása szolgáló BASIC program látható a 3. listában. Ez a program a Motorola cég ún. S1-S9 formátumát dolgozza fel. A program a kívánt kezdő- és az előző program által kiírt végcímmel dolgozik. Amennyiben ez nem lehetséges, úgy ezt jelzi, és újat kér. A sorok/blokkok kezdőcímeit kirja, így ha azok nem egyenletesen követik egymást, valami probléma adódott. A hibás részt RUN 200 paranccsal vizsgálhatjuk. A program csak 2800H cím feletti területre töltethető! Ez alatti területre csak úgy tölthetünk, ha először a megengedett területre töltünk, majd utána külön áthelyezzük.

Hogyan dolgozik a program? Először is kere-

Címek: (Kazdóci m helye): 3902-3H
 REF: 3901
 Bit(házam): 3900
 PIA(cím): FF20
 A(bla(cím): 0400

A program:

BE 3802	S LDX	K	Töltés a tároló címetl
RD 8D	BSR	T	Olvas be egy adatot
A7 80	STA	X+	Tárol a megadott címetl
CE 04	LDA	#A	Tedd az „ablakba”
A7 C4	STA	J	
BF 3802	STX	K	Tárol a tároló címetl
29 F2	BRA	S	Újra az elejére
39	RTS		
26 F4	T PSHU	B	Véd meg a B-regisztert
8D 2D	T1 BSR	V	Kapod a startbitet
2D FC	BLT	T1	
8D 29	BSR	V	Jön a bitsorozat...
8D 27	BSR	V	
2D F6	BLT	T1	Ha hibás, vissza
C6 05	LDB	#8	Allítsd be a bitszámot
F7 3800	STB	BI	
4F	CLR	A	Töröld az adattárolót
8D 11	BSR	I	Várj egy bitnyi időt
44	T2 LSR	A	Lejtsd a következő bit-hez
8D 30	BSR	AA	Bitállapot?
2E 02	BGT	T3	Ha „0”, folytatás!
8A 80	ORA	#S80	„1”, változtatás!
8D 08	T3 BSR	I	Várj egy bitnyi időt
7A 3800	DEC	BI	Ha van még bit...
2E 72	BGT	T2	Folytasd!
37 04	PULU	B	Töltés vissza B-t
39	RTS		Térj vissza!
1F 12	I TFR	X, Y	Véd meg az X-regisztert
BE 0104	LDX	#S0104	Baud-sebesség állandó
38 02	I1 LEAX	X	Visszszámolás
26 FC	BNE	T1	
1F 21	TRF	Y,X	Töltés vissza az X-regisztert
39	RTS		Térj vissza!
36 02	V PSHU	A	Véd meg az A-regisztert
4F	CLR	A	Töröld a számlálót
F6 FF20	LDB	PIA	Olvas be a magnóról
C4 01	ANDB	#1	
F1 3801	CMPB	REF	Ha azonos a referenciával...
27 F5	BEQ	V1	várj a változásra!
F7 3801	STB	REF	Tárol az új értéket!
80 0E	SUBA	#S0E	Frekvencia állandó!
37 02	PULU	A	Töltés vissza az A-regisztert
39	RTS		Térj vissza!
F6 FF20	AA LDB	PIA	Olvas be a magnóról
C4 01	ANDB	#1	a referencia tárolóba!
F7 3801	STB	REF	
8D 09	BSR	V	Határozd meg a hullámformát!
8D DE	BSR	V	
2E 04	BGT	AA1	
8D DA	BSR	V	„1”-hez tartozó idő
8D D8	BSR	V	
39	AA1 RTS		Térj vissza!

* Ezek az értékek 894 kHz frekvencián (zártott 6809-re vonatkoznak). Más esetekre az egyes utasítások végrehajtási ideje alapján kell a megfelelő értéket meghatározni.

2. lista

si a megadott ASCII 00 kódsorozatot (legalább 5), majd az "S1" blokk kezdest. A blokk kezdése alapján megvizsgálja, hogy a megkívánt végleges címetület nem fedie-át a betöltési címetületet. Ha igen, módosíthatjuk a megkívánt címet.

Átdolgozása más formátumokra a következő segítségével történhet. A bevezető, blokkfejlesztés-vizsgálat mindig szükséges. Eltérés csak a karakterek kódjában, számban lehet (50, 70, 80, 130-as sor). A tetszés szerinti címre töltendő formátumnál természetesen a blokkból címki-

```

10 REMS1-S9 MOTOROLA FORMATTUM F
ELDDLOGOZO
20 CLS: CLEAR500, &H27FF: INPUT "PRO
GRAMNEVEV, KIWAANT KEZDOECIM, ERE
DETI VEEGCIM": C%, A%, E: CLOADM C%, A
-19456: C=0: E=E+19456-A
30 FORI=A TO E: B=PEEK(I): IFB<>0
HENS0
40 C=I: I=E
50 NEXTI: IFC<>0 THEN70
60 PRINT "NINCS ELEJEI": STOP
70 A1=0: FORI=C TO E: B=PEEK(I): IF
B<>0 THEN100
80 FORJ=0 TO4: B=B+PEEK(J+1): NEXTJ
: IFB<>0 THEN100
90 C=I: I=E
100 NEXTI: IFB<>0 THEN60
105 B1=0
110 D=0: IF C<E THEN130
120 PRINT "VEEGE!": IFB1=0 THENPRIN
T " CSAK 0 -SOROZAT!":
125 STOP
130 FORI=C TO E: B=PEEK(I): A1=PEE
K(I+1): A1=100*B+A1: IFA1<>5331THE
N150
140 C=I+4: I=E: D=1: A#=CHR$(PEEK(I
+2))+CHR$(PEEK(I+3)): A1=VAL("&H
+A#"): B1=B1+1: IFB1<>0 THEN150
145 C1=C-4
150 NEXTI: IFD=0 THEN120
160 A#="" : FORI=0 TO3: A#=#+CHR$(P
EEK(I+C)): NEXTI: B=VAL("&H"+A#): I
FB<A THEN185
170 IFB<E THEN185
180 PRINT "A TALAALT CIM": B: " BE
LEESIK A MEGADOTT TARTOMANYBA "
: A1: "-": E1: "A TALAALT KEZDOECIM":
: C2: GOTO20
185 IFB1<>0 THEN190
186 C2=B: PRINT "A TALAALT KEZDOEC
IM " : B1: " MODOSSITJAK(I/N)?": INPU
TU#: IFU#="N" THEN190
187 INPUT "UJ CIM": B2: IFB2<>0 &H2000
THEN189
188 PRINT "NEM MEGENGEDETT!": GOTO
186
189 B2=B2-C2: C2=C2+B2
190 B=B+B2: PRINTB: " " : FORI=0 TO
2*A1-1: A#=#+CHR$(PEEK(C+I))+CHR$(P
EEK(C+I+1)): I=PEEK, VAL("&H"+A#): C
=C+2: B=B+1: NEXTI: C=C+6: GOTO110
200 INPUT "A HIBA KEZDOE EES VEEG
CIME": X, Y: D1=2*(4+A1)*X-C2/A1:
D2=2*(4+A1)*Y-C2/A1: FORI=D1 TO
D2: PRINTHEX$(PEEK(I+C1)): NEXTI
: STOP
    
```

Ki ad magyarázatot?

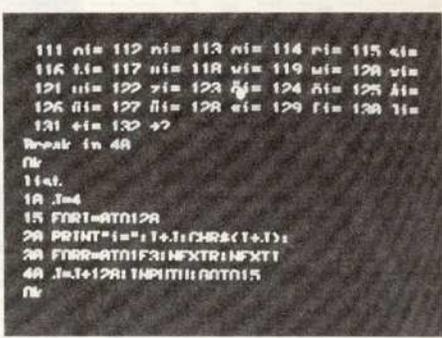
Ebben a sorozatban olyan „furaságokat” közlünk (az amerikai szakirodalomban BUG), amiket egy-egy gép elkövet, és amikre a géphez adott dokumentációkban nines magyarázat.

Reméljük, hogy akadnak olyan olvasók, akik képesek magyarázattal szolgálni, és ezt el is küldik részünkre. Ezeket, ha valóban magyarázatok, közöljük.

Magyarázatot a gyártó is adhat! Várjuk a „furaságokat”!

Elsőként lapunk 1984/6. számában, a Primóval kapcsolatban megjelentek várjuk a válaszokat. Kiegészítésként az eredeti cikk 3. fényképen látható eredményt szolgáló programot is közöljük.

DR. SIMONYI ENDRE



Közérdekű válaszok az olvasóknak

1. A rovat szerkesztő az idővesztés és a levelek továbbküldözgetésének csökkentése céljából felkéri az olvasókat, hogy a cikkekben nem említett témakörrel kapcsolatban keressék meg

● elsősorban a hozzájuk legközelebbi első helyen működő klubot (ezekről tájékoztatást kaphatnak a Neumann János Számítógéptudományi Társaságban),

● ha ez eredménytelen lenne, úgy a HCC budapesti szervezeteinek megfelelő szekcióját (tájékoztassuk ugyanott),

● és csak legvégső esetben a rovat szerkesztőt.

2. Azoknál a cikkeknel, ahol az szerepel, hogy „A Klub kellő számú érdeklődő esetén vállalja” valaminek az elkészítését, vegyék a szöveget komolyan, és ne küldjenek feleslegesen sürgető leveleket. Ettől a feltételtől nem változnak meg!

3. Könyvek, szaklapok, adatlapok, kapcsolások, részegységek, alkatrészek beszerzése ügyében tanácsom azonos az 1. pontban leírtakkal, llyeneket nem tudok küldeni!

4. Aki a HCC tagjai akarnak lenni, írják meg az NJSZT-nek, hogy melyik szekcióbba akarnak belépni, és kérjék, hogy levélüket továbbítsák a szekciónak.

5. Csak olvasható névvel és címmel ellátott levelekre tudok válaszolni, ezért kérem, hogy ezeket az adatokat vagy írógéppel vagy nyomtatott nagybetűvel írják!

6. „Építsünk számítógépet!”

A gép építéséhez szükséges ismeretek. Mindenfajta számítógép megépítéséhez (és ez alól ez a gép sem kivétel) elengedhetetlenül szükséges kellő forrászat, szerelési gyakorlat. Ennek hiányában csak drága alkatrészek tönkretélése árán tanuljuk meg ezeket a műveleteket. Természetesen ezenfelül bizonyos tájékozottságra is szükség van az elektronikus áramkörök és a gépbe beépítésre kerülő alkatrészek terén. Ezeket az ismereteket terjedelműgység miatt ebben a sorozatban nem adhatjuk. (A Microcomputing című szaklapban egy olyan sorozat, amely ezeket az alapsmereteket is nyújtotta, és a végén egy nagyon egyszerű mikro-számítógépet, 24 részből áll, és egy-egy rész terjedelme a miénkénél háromszorosa volt.)

A gép építéséhez szükséges eszközök. Feltételezve, hogy az áramköri kártyák készítése sikerül,

a gépépítőknak forrasztás, ellenőrző mérések végzése, összeszerelés lesz a feladatuk. (Az élesítés ez a felsorolás nem tartalmazza!) Ehhez jó forrasztópáka, statikus és dinamikus (de legalább statikus feszültség, áram, ellenállás), elektronikus jellemzők méréséhez szükséges eszközök nélkülözhetetlenek. Az írógéppel készült eredeti változatok építőire mechanikai megmunkálás is vár, így ilyen irányú ismeretekre és eszközökre is szükség van.

A szerző lehetőségeinek korlátai. Kénytelen vagyok ismételten leírni, hogy nem áll módomban egy világégen, amely esetleg reklám céljából hajlandó lenne az olvasóknak számítógépet, áramköri kártyát, alkatrészeket, filmeket, rajzokat küldeni - ingyen. Ez utóbbiakat és a beültetés nélküli kártyák fényképeit, részletes elvi rajzait, működési leírását a sorozatban ismertettem, illetve folyamatosan ismertetem. Az ezekkel kapcsolatos levelekre ezért külön nem válaszolok.

A gép teljesíthetősége. Eredeti elképzelésem az volt, hogy az ezzel kapcsolatos rész az egyes egységek és összállítás módjuk ismertetése után, a sorozat végén következne majd. A sok kérdés miatt azonban a sorozat mostani száma erről szól.

Az olvasók tapasztalatai. Ha közérdekű, közöljük azokat. Így került sor a módosított billentyűzetillesztő közlésére.

Közreműködés a „gyártásban”. Több olvasó is ajánlkozott arra, hogy díjazás fejében részt venne a gépépítésben, bemérésben, élesítésben. A konkrét ajánlatokat és kéréseket, a segítséggel kapcsolatos tapasztalatokat készek vagyunk közvetíteni és hozzájárulást esetén közzétenni.

A szerző elérés módja. Minden a µM-hoz érkező és nekem címzett levelet, telefononként megkapok, ezért kérem, hogy a továbbiakban is kizárólag a szerkesztőség címen keressenek meg.

Megrendelés. Sokan szeretnék megrendelni az áramköri kártyákat. Az egyik olvasó javasolta, hogy írjanak azok, akik az ár ismeretében megrendelnék. Tehát: írjanak, akik a 3500 forintot meg nem haladó áron megrendelnék az alapkártyából, központiség-kártyából, képernyőhájt kártyából álló készletet!

DR. SIMONYI ENDRE

3. lista

olvasás és címvizsgálat (160, 180, 186-os sor) elmarad. A programban használt kevéssé ismert jelek: a CLOADM utasítás adott nevű gépi kódú programot tölt be a megadott címtől; a HEXSI függvény decimális egész számot alakít át hexadecimálissá.

Az „ablak” használata megkönnyíti ismeretlen formátumok feldolgozását is, hiszen az ismétlődések a „szemünk előtt” játszódnak le. Amennyiben nem adódik szabályosság, és csak egy-egy „értelmes” karaktert látunk, úgy lehet, hogy nem egy stopbita a formátum. Ezen a gépi kódú program megfelelő részének módosításával segíthetünk.

A második program indítása előtt DRAGON 64-nél beadandó: EXEC 48000.

DR. SIMONYI ENDRE

APPLE II. és a vele kompatibilis számítógépekre ajánljuk programjainkat

A programok megrendelhetők az alábbi címen:
Alkotó Irijűség Egyesülés
Számítástechnikai Iroda
1519 Budapest, Pf.: 330.
Telefon: 112-666, 113-608



- Hőcserélő méretezés
 - Kémény méretezés
 - Klíma szimulációs program
 - Épületek hőszükséglet számítása
 - Építészeti rajzolóprogram
 - Mesterséges szivattyús központi rendszer méretezése
 - Kétsöves szivattyús központi rendszer méretezése
 - Deflektoros kéményméretezés
- A programok
más típusú számítógépekre is átdolgozhatók.

A technológiák szerepe

A mikroelektronika oly mértékben a mikroszámítógépes technika megalapozója, hogy túlzottan divattá vált szerepének egyoldalú hangsúlyozása, ami néha már fráziszzerű puffogtatásnak hat. Ennek az a veszélye, hogy a mikroelektronika afféle univerzális „bölcsék kövének” tetszik, ami minden bajra önmagában megadja a megfelelő megoldást. Másként mutatkozik azonban, ha azoknak a mélyebb összefüggéseknek az oldaláról nézzük, amelyek keresztlégyfolyamatosan hatást gyakorolt és gyakorol a modern informatika arculatának kialakulására. És egész másképpen jelentek-ez akkor, ha a társ technológiákkal együtt szemléljük befolyását, és nem utolsósorban az emberi tényezőt sem felejtjük el.

Sok-sok MOS memória

Az ún. fémoxid-felvezető (Metal-Oxide-Semiconductor - MOS) típusú memóriáramkörök kifejlesztése előtt felvezető-technológiával csak bipoláris flip-flop komponenseken alapuló áramköröket gyártottak a felvezetőgyártók. Az utóbbiak nem tudták kiszorítani az operatív táruk felépítéséhez addigra széles körben elterjedt ferritgyűrűs megoldást, és elsősorban a processzoron belüli tárelemek (regiszterek) kialakításában kerültek alkalmazásra. A nagyobb elemstűréség és kisebb áramfelvétel miatt a MOS technológia viszont lehetővé tette az áramkörök integráltsági fokának oly mértékű növelését, aminek eredményeképpen a felvezető memória egyre versenyképesebb lett a nehézkesen tekercselhető, drága és nagy térfogatigényű ferritgyűrűs megoldással szemben.

Ezt a lehetőséget ismerték fel az 1968-ban alapított Intel cég vezetői, és felismerésüket hamarosan siker is koronázta. 1969-ben hozták piacra a 256 bit tárkapacitású, ún. statikus RAM-ot, majd egy évvel később az 1024 bites, dinamikus RAM integrált áramkört. A dinamikus RAM megoldás különösen piacképesnek bizonyult, mivel az elemi információ lehető legegyszerűbb tárolása (tulajdonképpen elektromos töltések formájában) miatt az alkalmazott mikroelektronikai technológia adott színvonalra mellett négyszeres kapacitású chipet tudtak kialakítani. Mindez tükröződött az 1 bitre eső, arányosan alacsonyabb árban, és még piacképesebb tette a MOS memória technológiát.

A 70-es évekre a nagyszámítógépek és az egyre nagyobb teljesítményű mini- (megamini, supermini) számítógépek széles körű vállalati elterjedése volt a jellemző. Mindez gyakorlatilag korlátlan felvevőpályát jelentett a dinamikus RAM-ok gyártóinak, hiszen ezekhez a gépekhez egyre több és egyre nagyobb kapacitású memóriarendszerekre volt szükség. Eppen használt technológiák berendezéseiket így teljes kapacitással tudták üzemeltetni, viszonylag rövid idő, 3-4 év alatt át tudtak térni a következő technológiai generációra, és egyre jobb minőségű, egyre nagyobb termelékenységű berendezéseket tudtak üzembe helyezni. Az Intel 1982-ben bevezetett HMOS-III technológiája például már a hatodik volt a sorban, és a sebesség-teljesítmény szorzat két nagyságrenddel jobb volt, mint a legelső technológiáé.

Az egyedi áramkörök tervezési technológiája is korlátozás nélkül fejlődhetett, lehetővé téve, hogy minden három év után újabb, négyszeres tárkapacitású és gyorsabb elérési idejű RAM generáció kerüljön gyártásba. Így jelentek meg egymás után a 4, 16, 64 és 256 kbites dinamikus RAM-ok.

A számítógépgyártók „memóriaéhsége” még az egyes chippek kapacitásnövelésénél is erősebben növekedett, így az egyre többre kerülő gyártóberendezéseket, valamint az egyre növekvő tervezési költségeket a RAM chippek áraiiban nem lehetett érzékelni.

A 80-as években ez azt jelentette, hogy az éppen legnagyobb tömegben használt dinamikus RAM generáció egyetlen chipjének ára, kapacitásától függetlenül, hozzávetőleg mindig 4 dollár körül alakult. Az egy bitre eső ár állandóan csökkent és a számítógépgyártók 2-4 évenként tudtak áttermi a következő generációra (lásd a táblázatot). Ez az áttérés jelentkezik az állandóan csökkenő memóriáárakban, amint az ábrán látható. A ciklikus piaci tendencia a továbbiakban is folytatódni fog, ehhez azonban a 80-as évek elejétől már a mikroprocesszor alapú számítógépek egyre bővülő termelése biztosítja a hátteret.

A mikroprocesszoros technika kezdetei

Egy japán zsebszámítógépgyártó, a Bizcomp Corp. azzal kezdte meg 1969-ben az Intelt, hogy

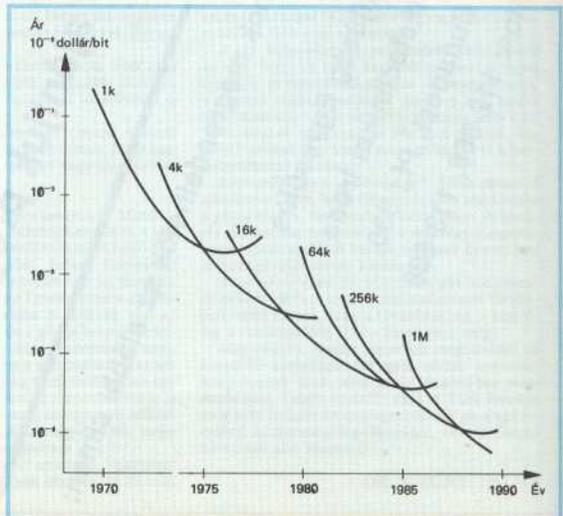
az készítés számára néhány nagy integráltságú céláramkört berendezéséihez. A tervezés már akkor sem került kevésbé, így az első mérlegelésnél az Intel el akarta vetni az üzleti ajánlatot. A későbbi felülvizsgálat során azonban az Intel egyik fiatal mérnökének, Marcian E. Hoffnak mentőötlelete támadt. Azt javasolta, hogy készítsenek egy általános célú processzort, majd azt programozzák be úgy, hogy képes legyen támogatni az igényelt zsebszámítógép-funkciókat. Az így kidolgozott és 1971 novemberében piacra hozott, Intel 4004 típusú mikroprocesszor még véletlenül primitív volt. 4 bites adatokkal tudott műveletet végezni, mindössze 4,5 kbajt memóriát volt képes megcímezni, de már meglejtente benne az azóta is jellemző főbb architektúrális elvek, mint például a belső sinszervezés.

Egy másik megkeresés eredményeképpen született meg 1972-ben az Intel első 8 bites mikroprocesszora, az Intel 8008. A Computer Terminal Corporation (jelenleg Datapoint) gyors, diszkrét elemekből (TTL) akart egy bitsoros processzort kialakítani, olcsónak tervezett képernyős termináljai pro-

gramozható megvalósításához. A tároló verem, ún. stack egységet viszont az Inteltől kívánta beszerezni nagy integráltságú áramkör formájában. Az Intel ellenjavaslatára az egész processzort egyetlen chipen kezdték megvalósítani. A javasolt utasításkészletet további utasításokkal egészítették ki, 16 kbajt címzést alakították ki és a 8 bites adatkezelési képességet egy 8 bites mélységű processzor stack koncepcióval kombinálták. Az alkalmazott p-MOS technológia miatt az így elkészült processzor elég lassú volt (0,5-0,8 MHz ütemfrekvencia), hasonló jellegű alkalmazások, tehát berendezések cél szerinti funkcióinak programozott (emulált) megvalósítására azonban már bátran lehetett ajánlani.

A 8008 átültetések az Intel második generációs, n-MOS technológiájára, a mikroprocesszor chipeket tervező Federico Faggin úgy találta, hogy érdemes lenne az egész processzor-architektúrát áttervezni. Így született meg az 1974 elején piacra került Intel 8080 típusú mikroprocesszor. A 2-3 MHz ütemfrekvenciájú processzor már 64 kbajt memóriát volt képes meg-

A dinamikus RAM áramkör tárkapacitása	1 kbit	4 kbit	16 kbit	64 kbit	256 kbit	1 Mbit
A mintapeldányok megjelenésének éve	1970	1973	1976	1979	1982	1985
A következő generációra való, ár szerinti áttérés éve	1975	1979	1983	1983	1988	-
Ár az áttérés pillanatában (10 ⁻⁵ dollár/bit)	200	30	7	3,5	1	-



mezni, és utasításokészlet em nem néhány hét alatt, hanem sokkal alaposabban átgondolva alakították ki.

Faggin már ekkor úgy látta, hogy a 8080 architektúrális képességeit (utasításkészlet, regiszterek stb.) továbbfejlesztve újabb processzort is ki lehet fejleszteni, ami már számítógép-építési célokra is tökéletesen megfelel. Javaslatát azonban az Intel vezetők nem támogatták, mivel úgy vélték, hogy ennek nincs perspektívája, sőt ez már számítógépgyártó vevőkörük üzleti érdekeit is veszélyeztetné. Faggin eltávozott tehát az Inteltől, és az Exxon olajmonopólium pénzügyi támogatásával megalapította a Zilog céget. Így született meg 1976 márciusában a 8080-nal felülről kompatibilis Z80 processzor.

Az Intellel konkuráló Motorola cégénél, a 6800-as mikroprocesszor fejlesztésében működött közre egy másik neves személyiség, Chuck Peddle. (Nem egy mikro-számítógépes vezéregységiség keresztnéve becézett alakjában fordul elő a mikrogepes sajtóban, így mi is ezt a gyakorlatot vagyunk kénytelenek követni.) A Motorola-nál Peddle néhány eredeti áramkörök megoldás kialakításával tűnt ki. Így például az ő nevéhez fűződik a mikroprocesszoros gépek bemeneti/kimeneti alrendszerének gyors fejleszthetőségét támogató PIA (Peripheral Interface Adapter) koncepció és annak megvalósítása nagy integráltságú áramkörök chip formájában. Számítógépes rendszerszemléleté egyáltalán nem meglepő, hiszen 1959-től kezdve, 11 éven keresztül dolgozott a General Electric számítógépgyártó részlegénél. Számítógépes múltja indítható ott arra, hogy munkatársaival együtt 1974-ben átmenjen a MOS Technology nevű, újonnan alakult céghez, és belefogjon egy igazán olcsó és architektúráisan igen hatékony mikroprocesszor, a 6502 fejlesztésébe.

Az 1975 második felében megjelent 6502-t valóban igen olcsón kinalták. Kiskereskedőnél mindössze 25 dollárba kerül, szemben a többi mikroprocesszor kb. 200 dolláros árával. A dinamikus RAM áramkörök generációját bemutató táblázatunkra pilsantva láthatjuk, hogy éppen ebben az időben lesznek előnyösebb a 4 kibites DRAM áramkörök, vagyis néhányszor tíz dollárért már 4 kibjt, illetve 8 kibjt kapacitású memóriához is be lehetett szerezni a áramköröket. Peddle még más chipeket (például a PIA egyik továbbfejlesztett változatát) is kidolgozott a 6502-höz, így minden kézen állt ahhoz, hogy akár a számítógép-építő hobbiisták készíteni tudjanak maguknak személyes használatú mikroszámítógépet. A hobbiizmus pedig, mint látni fogjuk, elvezetett a személyi számítógéphez.

Hobbiból lett háziipar

Az első, nagyobb teljesítményű mikroprocesszorokat már általános célú géphe is érdemes volt beépíteni, de nem volt könnyű megtalálni sem a megfelelő piacot, sem pedig az ennek megfelelő számítógép-konstrukciót. A francia R2E által kifejlesztett első mikroszámítógép nem ért el azonnali, áttűtő piaci sikert. Az USA-ban is több kísérlet volt 1974 során (például az Intel 8008 alapú Mark-8 készlet), amelyeknek szinte alig volt hatásuk az események további alakulására.

Az igazi mikrogepes piacot a Micro Instrumentation and Telemetry Systems (MITS) nyitotta meg 1975 januárjában, amikor Altair 8800 típusú gépének hirdetése megjelent a Popular Electronics című magazin címlapján. A MITS-et 1970-ben alapították New Mexico állam Albuquerque nevű városában, egy Ed Roberts nevű úriember garázsában. A cég elsősorban olyan kisméretű elektronikus berendezéseket gyártott sikerrel, amelyek játékraktákba és repülőmodellekbe építve a helyzetjelzésre, a kényszerlezállási ejtőernyő működötésére és fényjelzők vezérlésére szolgáltak. Nem volt tehát a cégtől idegen az elektronika hobbi célokra való alkalmazása, és ez a továbbiak szempontjából döntőnek bizonyult.

A Roberts által tervezett Altair 8800 egyáltalán nem hasonlított külső megjelenésében a ma ismert mikroszámítógépekhez. Előlapján az akkori miniszámítógépekhez hasonló kapcsoló- és lámpásor helyezkedett el. Belsejében viszont egy nyitott, jogilag nem levedett, egyszerű sinrendszert alakított ki Roberts, így eleve biztosítva volt a moduláris bővíthetőség nemcsak az eredeti gyártó által szállított kártyákkal, hanem más gyártók adott feladatra kialakított kártyáival és ezekkel csatlakozó részegységekkel is. Roberts persze inkább az vezette erre a megoldásra, hogy a hobbiisták számára ezzel is vonzóvá és minímális kiépítésű alapkészlet (kit) formájában egyúttal elérhetővé is tegye terméket. A vevő által összeszerelendő alapkészletet 395 dollárért hirdette meg a MITS, míg az összeszerelt változatot 621 dollárért árulták.

Összehasonlításként érdemes megemlíteni, hogy az Altair 8800 Advanced Accounting/Engineering Systemnek nevezett teljes kiépítésű rendszert 10 500 dollárért hozták forgalomba. A valóban olcsó Altair 8800 alapkészletből több ezret adott el a MITS 1975 év során, és így valóban megalapozta azt a lavinaszerű folyamatot, ami azóta is jellemzi a mikroszámítógépek széles körű elterjedését.

1975 másik meghatározó eseménye volt az első számítógép-építő klubok megalakulása. Ekkoriban még igencsak szűkebben voltak a hobbiisták a mikroelektromikai elemek alkalmazásával és

az első gépek „továbbépítésével” kapcsolatos információknak, rászorultak egymás szakismeretére és segítségére.

A kaliforniai Menlo Parkban 1975 márciusában megalakult Homebrew Computer Club (HCC) volt talán a legelső a klubok között. Ennek a csoportosulásnak mégsem ez adja meg a hirtét, hanem az a sajátos szerep, amit a mikroszámítógépes ipar kibontakoztatásában betöltött. A Gordon French nevű úr garázsában megrendezett első összejövetelen és az első év további összejövetelein szinte kivétel nélkül olyanok vettek részt, akik vállalkozóként, konstruktorként vagy inkább mindkét minőségben jelentős szerephez jutottak a későbbi mikroszámítógépes ipar kialakulásában.

Roger Melen és Hary Garland, a HCC tagjai és a Cromemco cég alapítói hozták forgalomba az első, Altair sinrendszere dugaszolható kártyát. Dazzlernek (káprázotató, de más értelemben meghökéltető, elémítőt és elképesztőt is jelent) nevezett termékük színes grafika megjelenítését tette lehetővé a tévéképernyőn, ezért az első színes grafikai kártyának is tekinthető.

Mivel akkoriban még meglehetősen kevés számítógépes szakkult volt, a Cromemco elsősorban postai csomagküldő módszerrel forgalmazott. Az első szakküzetek viszont sikerrel használták az érdeklődés felkeltésére. Az egyik nevezetes sztori szerint New York város éjszakai autósait valószínűleg hipnotizálta a Dazzlerhez szállított Kaleidoscope nevű program segítségével generált véletlenszerű mintázat, amit az üzlet kirakataiban elhelyezett tévé képernyőjén láthattak. Az ezért kezeztelt forgalmi dugót csak úgy tudta megmozdítani a rendőrség, hogy éjnek idején megkereste a tulajdonost, és kikapcsoltatta a veszedelmes készüléket.

Melen és Garlandot egyre többen követték, és egymás után jöttek létre a garázsokban és más, a házhoz tartozó helyiségekben működő kisvállalatok. Egy sajátos háziipar (cottage industry) alakult ki tulajdonképpen, ami egyben meghatározta a mikroszámítógépes ipar maig fennálló, széles körű kooperáció alapuló működési jellegét.

Ás ilyen kezdetet persze nemcsak a kiegészítő kártya- és részegységgyártók határozták meg, hanem azok az újonnan alapított vállalatok, amelyek mikroszámítógépek konstrukcióját az Altair 8800 nyitott sinrendszere alapozták. Ilyenek voltak az IMSAI és a Processor Technology. Az utóbbinak az alapítója és főkonstruktoré is HCC-tag volt.

Ezek az újabb mikroszámítógépek még csak kismértékben javítottak az eredeti Altair konstrukción. A minigépszerű előlapot felváltották az állandó (ROM) me-

móriába beírt programból (firmware) való működtetésre, amire irógépszerű billentyűzet állt rendelkezésre. Az Altair sinrendszert azért hagyták meg, mert az első időszak számítógép-építő vevői már ismerték azt, a későbbiekben egyre nagyobb számú többi vevőnek pedig gépe komplettálása szempontjából volt az előnyös, hogy a kiegészítő kártyákat az alapherendezés-gyártótól függetlenül is be tudta szerezni. Valamennyi gyártónak is előnyös volt ez a megoldás, mivel csak egy-egy területre kellett szűkös erőforrásaikkal koncentrálniuk, és mégis számíthatnak arra, hogy széles körű piaca lesz termékeiknek.

Ugyanebben a kezdeti időszakban alakultak ki azok a kereskedelmi és árubemutató helyek, amelyek azóta is a leginkább meghatározóak. Az egyre terjedő kiskereskedelmi számítógépes szakküzetek a lehető leguralmasabb módon tudták közvetíteni a bővülő vevőkör felé az egyre szélesebb áruválasztékot. Megrendeléseikkel egyúttal hatékonyan befolyásolták az újabb és újabb gyártói törekvéseket. A hobbiistáknak találkozóhelyet biztosítottak, és a klubokhoz hasonlóan, az információ és tapasztalatcsere sajátos fórumává lettek. A boltok propagandája révén ezek és ezek lettek az új számítógépes örület megszállottjai. A klubok, a háziipar és a garapodó bálhálózat így további, még gyorsabb mozgásra ösztönözte a megindult mikroszámítógépes lavinát, miközben egymást is erősítették.

Hamarosan szükségessé váltak az országos méretű kiállítások és termékbemutatók, ahol a gyártók a kereskedőknek, valamint érdeklődőknek és egymásnak be tudták mutatni új termékeiket. Először a keleti parton rendeztek ilyen eseményt 1976-ban (East Coast Computer Show), majd a West Coast Computer Show-t a (Realistic Computer Faire). Az utóbbi megszervezésében, talán mondanunk sem kell, a HCC egyik első tagja, Jim Warren járt az élen.

Az első keleti parti bemutató arról is nevezetes, hogy itt kapott új nevet a már több gyártó által is használt Altair sinrendszere. Hogy ne kelljen ezeknek az új gyártóknak ingyen propagandát biztosítani a velük versenben álló Altairnek, azt találták ki, hogy S-100 sinrendszernek nevezzék, mivel 100 érintkezős csatlakozási felületet irt elő kártya szinten. Ekkor még nem beszélhettünk igazi szabványról, mivel hiányzott pontos és mindenre kiterjedő meghatározása, de kb. két évtel később, a második nyugati parti számítógépvásáron már erre is sor került.

A 80-as évek elején országos ágazati szabványként (IEEE-696) is megjelent az első, berendezésgyártótól független ipari szabvány, bizonyítva azt a jelentős

reprezent, amit a mikroszámítógépes iparban betöltött. A lavina továbbra is erősödését más berendezésgyártótól független, de facto szabványos megoldások is elősegítették, elsősorban az alapszoftver területén.

Az olcsó, szabványos alapszoftverek

Két, a középiskolát éppen csak hogy elvégzett fiatalember, Bill Gates és Paul Allen az Altair 8800 megelégedésekor úgy érezték, hogy most jött el az ő idejük. Már korábban is foglalkoztak a gondolatokkal, hogy a 8008 mikroprocesszorra egy BASIC rendszert készítsenek, de akkor úgy találták, hogy annak utatásiskészlete még nem elegendő a feladathoz. Annak ellenére, hogy nem volt Altair gépek, nem riadtak vissza magától a feladattól, mivel már több éves szoftverfejlesztő múlt állt mögöttük üzleti értelemben is. Az Intel 8080 kézikönyv alapján egy 8080-at utánzó (szimulátor) programot írtak a kis cégek számára elérhető PDP-10 névű számítógépre, majd Gates megtervezte a 8080 BASIC értelmező (interpreter) programot, amit a szimulátoron próbálhattak ki először. Ezután hívták fel a MITS-et, ahol mindjárt rájöttek, hogy a két fiú nem a levegőben beszél, és repülőgépen székelyükre vitték őket.

Ilyen előzmények után jelent meg a piacon az első mikroszámítógépes BASIC rendszer. Ez nagyon kompakt változat volt, mivel mindössze 4 kb-át memóriára volt szüksége, amiből 3,2 kb-ot foglalt el maga az interpreter. Igen komoly szoftverfejlesztési teljesítmény, ha meggondoljuk, hogy ez a tömör rendszer még lebegőpontos aritmetikával is rendelkezett.

Hobbisták között tehát villámgyorsan terjedt Gates és Allen 4 k-s BASIC-je, piackepés volt adaptációja más 8080 alapú gépekre is, illetve egyre fokozódó igény jelentkezett az egymással felülről kompatibilis, de önmagában egyre többet tudó BASIC rendszerek sorozatára. Ebből nőtt ki a Microsoft nevű független szoftvergyártó vállalat, élén a még ma is roppant fiatal egyik alapítóval.

Az Altairt követő első gyártók egyike volt az IMSAI. Floppy háttérrel is felszerelhető gépének első példányait még azzal szállította le elsők veivőinek, hogy az operációs rendszert majd valamikor később biztosítja. Az IMSAI egy aprócska kaliforniai céget, a Digital Research-t kérte fel, hogy gépéhez kidolgozzon egy megfelelő operációs rendszert. A cég alapító tulajdonosa, Gary Kildall, főállásban a Naval Postgraduate School tanára volt, és már a 4004 megjelenése óta készített alapszoftvereket az Intel mik-

roprocesszor alapú rendszerekre. Az Intel megbízásából ő fejlesztette ki például a PL/M nevű nyelvi rendszert, és ennek során, mintegy melléktermékként, már kifejlesztett egy lemezes operációs rendszert. Ezt a rendszert tudta igen gyorsan adaptálni Kildall az IMSAI gépekre. Később rájött, hogy rendszerének gépfüggő részeit egy külön részbe, a BIOS-be (Basic Input-Output System) zárva, más 8080 alapú gépekre is könnyen átvihető lesz az operációs rendszer.

Az első nyugati parti számítógépes vásáron a vásár alapítója, Jim Warren arra beszélte rá Kildallt, hogy szoftvercsomagot készítsen rendszeréből. Így a hobbisták maguk tudták adaptálni azt mikrogepek konkrét hardverrendszerére. Ez akkori szemmel nézve óriási piacot jelentett, mivel jellemző volt, hogy a hobbisták maguk kompletálták mikroszámítógépeiket például háttérrendszerrel, így a megfelelő operációs rendszerrel is nekik kellett gondoskodniuk. A CP/M 1.3-nak nevezett első változat mindössze néhány száz dollárba került, így egyre nőtt az érdeklődés és a vevőkör, míg a CP/M de facto ipari szabvány nem lett. Elterjedésében fontos tényező volt a magasabb szintű szoftverelemek (például alkalmazási szoftver) irányában mutatott szabványos csatlakozási felülete is.

Lavinászerű folyamatként jelentkezett az, hogy érdemes volt egyéb rendszer- és alkalmazási szoftvert erre a dorozható operációs rendszerre írni, minék következtében még jobban növekedtek az operációs rendszer eladásai, ezáltal viszont még fokozottabban érdemessé vált a szoftverek ehhez az operációs rendszerhez igazodva fejleszteni. Erre a folyamatra rátalálva válhatott a kibontakozó mikroszámítógépes szoftveripar másik „óriásává” a Digital Research cég.

Az Intel innovatív mikroelektronikai elképzelései hamar elvezettek egy sajátos konstrukciójú számítógépes piachoz, jöhetett a cég eredeti elképzelései között ilyen egyáltalán nem szerepelt. A sajátos konstrukciót az jellemozte, hogy az egyén vált a mikroszámítógépes vásárlóivá, a hardver és a szoftver iparlag szabványos csatlakozási felületi biztosították a lehető legalacsonyabb kompletálási árat, valamint a lehető legkedvezőbb bővítési lehetőséget, és mindez az érdeklődők egyre nagyobb tömegének tette elérhetővé a számítógépet.

Ez tehát a mikroszámítógépes lavina első hullámának lényege, aminek értelemszerűen el kellett vezetnie a még nagyobb hatású személyi számítógépes konstrukció megjelenéséhez.

NACSA SÁNDOR
(Folytatjuk)

Sakkprogramozás

Specializált számítógépet vagy programot?

Milyen számítógépet vásároljunk? sorozatunk keretében közbetevőleg egy olyan kérdéssel foglalkozunk, amely – amint a mind gyakrabban hozzánk intézett kérdésekből is kitudnik – sokakat foglalkoztat, akik számítógép ellen szeretnének sakkozni. A kérdés így hangzik: „Mit vásároljak inkább, sakk-számítógépet, vagy vegyek például egy Sinclairt vagy Commodore 64-et és ahhoz sakkprogramot?”. E kérdés természetesen beleavag témakörünkbe, hiszen amikor az olvasónak vásárlási döntésében kívánunk segítséget nyújtani, éppúgy gondolunk a sakk-számítógépekre, mint a személyi számítógépekre írt sakkprogramokra – ha ez sorozatunk címében eddig kifejezetten nem is szerepelt.

Két piac

Talán meglepi az olvasót, mégis jellemző az elektronikus sakkpartnerek mai piaci helyzetére, hogy az iparlag fejlett országok vagy inkább boltjaiban, áruházaiában másutt kell keresni a sakk-számítógépeket és a személyi programokat.

Érdekes kísérletről olvastunk a Computerschach und -Spiele című NSZK-beli folyóirat egyik legutóbbi számában. Egy szakember végigjárt egy sor szaküzletet, amelyben személyi számítógépeket és ezekre írt programokat, főként játékokat árultanak, és sakkprogramokat kerestek. A legtöbb helyen egyet sem vagy csak nagyon kevés típust talált, több esetben használati utasítás nélkül; sakkszere felvilágosítást pedig jóformán sehol sem kapott. (Itt hívjuk fel a figyelmet: senki, soha ne vásároljon semmilyen számítógépet vagy programot, akár újonnan, akár használaton a kezelési útmutató nélkül!) A magyarázat egyszerű: a személyisámítógépes-kereskedelemlennek a sakkprogram valahol a perifériáján szerepel. A sakk-számítógép viszont önálló, igen elterjedt üzletág; csaknem valamennyi nagy áruházban és számos szaküzletben kaphatók, nagy választékban, és sakkszere felvilágosításban sincs szor hiány.

Itt mutatjuk be olvasóinknak az említett folyóiratból vett jegyzékét, amely a forgalomban levő, különböző személyi számítógépekre írt programok legtöbbjét tartalmazza, sok esetben DM-ben megjelölt árukkal együtt. Ez az összeállítás hasznos információforrás, de számolni kell azfval, hogy aki egy bizonyos programot szeretne megvásárolni, nem könnyen találja meg.

Játékerő-összehasonlítás

Aki kíván személyi számítógéphez káiván sakkprogramot vásárolni, viszonylag szűk választékot talál, hiszen ugyanahhoz a géphez nem áll tucatnyi program rendelkezésre. A választáskor ugyanazokat a szempontokat kell figyelembe venni, mint a sakk-számítógépek esetében: a játékerőt, a lépések megjelenésének módját, a program különleges funkcióit, s bizonyos mértékben a kivitelt, az esztétikai kívánalmakat is, ami itt például a tetszőleges grafikai megoldásban vagy a használati utasítás áttekinthető kivitelezésében, a kazetta vagy diszket és a könyvecské csomagolásának vonzerejében rejlik. A tényezőz azonban más súllyal esnek latba, mint a specializált személyi számítógépek esetében.

A kielégítő játékerő természetesen egyformán fontos, bármilyen elektronikus partnerrel szemben küzdünk. Meg kell mondani, hogy a legjobb programok – például a Sargon III különféle verziói, amelyeknek alkotói a rovatunkban már többször említett Kathe és Dan Spracklen – sem érik el a mai kiemekekő sakk-számítógépekét.

A múlt év szeptemberében Glasgow-ban tartott, 4. mikro-VB résztvevői között a Sargon III programnak három, az Apple IIe, az Apple/Macintosh és az IBM mikroszámítógépekre írt verziója szerepelt, s kielégítő eredményt értek el, de elmaradtak a legjobb specializált számítógépektől. Érdemes megjegyezni, hogy az Apple verziókhöz gyorsító programot alkalmaztak; amely az eredetileg 1 MHz sebességű processzor teljesítményét három-négyesére fokozta.

**Colossus 2,1/Apple II -
Sargon III/Apple/Macintosh**

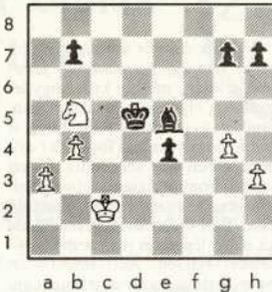
ábrákhoz, igen élvezeti és esztétikusan megtervezett diagramok útján történő játékot. Ehhez persze ismerni kell az algebrai sakkjelölést is, mert a lépéseket betáplálni csak a billentyűzetten lehet, kivéve, ha a készülékhez botkormány is rendelkezésre áll, amellyel igazán élvezetes a bábukat a kurzor útján ide-oda tologatni. A színes képernyő csak fokozza a játék nyújtotta esztétikai hatást.

Ennél fontosabb, hogy a képernyő nagy felülete sokkal jobb lehetőségeket kínál különféle információk közlésére, mint egy sakk-gép, és annyszor lehet egy-egy állásban különböző célokra felhasználni, ahányszor a programozó kívánja. Csaknem minden fejlett program részletes betekintést enged például abba, hogy mely lépést, lépéseket elemzi. A sakk-számítógép kívánságra szintén megmutatja az éppen vizsgált lépést, de egyszerre csak egyet; többre nincs fizikai lehetőség, mivel ezt rendszerint a diódká gyulladásával teszi. A személyi számítógépek sorra ki tudja írni, hogy mely lépéseket, változatokat - akárhányszor! - elemzi. Persze bármikor megadja a hadállás értékszámát is (amelyről a használati utasítás bővebben tájékoztat), és hogy hányadik fellépésnél tart; mindezeket egyszerre, egy képernyőkijelzéssel. Ezt az adottságot a jó sakkzók is nagyon szívesen fogadják.

Összefoglalás

A bevezetőben felvetett kérdésre tehát azt válaszolhatjuk, hogy aki nem arra helyezi a legfőbb súlyt, hogy elektronikus partnere a lehető legfejlettebb sakk tudással bírjon, és főként ha készülékével nemcsak sakkolni akar, vásároljon nyugodtan sakkprogramot személyi számítógépéhez, jól fog szórakozni vele. Hívja segítségül az itt közölt táblázatot, de levelben megkeresésre szívesen adunk további felvilágosítást, tanácsot bárkinek, aki sakk-számítógépet vagy sakkprogramot kíván vásárolni.

DR. LINDNER LÁSZLÓ



a b c d e f g h

Hadállás sötét 49. lépése után

Következett 50. Hc3+, Kd4! (A program „tudja”, hogy a királyával a gyalog elé kell kerülnie.) 51. h4 (A király behatolása ellen nincs védelem, mert 51. Hd1-re e3! után e2 megállíthatatlan.) 51. - Ke3 52. g5? (Típusos számítógéphiha, amelyet azonban egy igazán jó program ma már nem követ el. Nem „látja”, hogy a következő csere után az e gyalogot nem lehet feltartóztatni. Ezért okvetlenül Hd5+ kellett, s csak Kf3 után g5. Sötét akkor is nyer, de még nincs könnyű dolga.) 52. - Fxc3 (A program fejlettségének tanújele, hogy ez a csere, amely két lépéssel előbb még bizonytalan kimenetelű volt, ebben a helyzetben azonnal nyeresre vezet.) 53. Kxc3, Kf2 34. Kc4, e3 és sötét nyer.

A személyi számítógépek előnyei

Természetesnek kell tartanunk, hogy ha egy hardver kifejezetten a sakk céljaira készül, magasabb játékerőt lehet vele elérni; az általános számítógépek tervezői nem gondolhatnak olyan különleges szempontokra, amelyek előmozdítják a sakk tudást. Ugyanakkor azonban a személyi számítógépek rendelkeznek más adottságokkal, amelyeket a sakkra specializáltak nélkülöznek.

Első helyen a képernyőt kell említeni. Aki hozzászokik a sakk-

PROGRAM	SZERZŐ	ÉV	TÁR HORDOZÓ	SZÁMÍTÓGÉP	ÁR (DM)
Microchess 1.0	Jennings	78	1 k	AIM 65	
Microchess 1.5	Jennings	78	4 k	TRS 80	
Microchess 2.0	Jennings	78	8 k kazetta	PET 2001	75,—
Microchess 2.0	Jennings	78	16 k lemez	Apple II	
Sargon I	Spracklen	78	8 k lista	Jupiter 3	50,—
Sargon I	Spracklen	78	lemez	TRS 80	
Sargon I	Spracklen	78	lemez	Apple II	
Sargon II	Spracklen	79	lemez	TRS 80	
Sargon II	Spracklen	79	24 k lemez	Apple II	
Sargon II	Spracklen	83	ROM	VC 20	59,—
Sargon 2.5	Spracklen?		kazetta	Sharp	62,—
Sargon III	Spracklen	83	48 k lemez	Apple II	199,—
Sargon III	Spracklen	84	300 k lemez	Macintosh	
Sargon III	Spracklen	84	lemez	Liza	
Sargon III	Spracklen	84	lemez	IBM PC	
Chess 7.0	Atkin	84	48 k lemez	Apple II	249,—
Chess 7.0	Atkin	84	48 k lemez	Atari	249,—
How about...	Atkin	84	64 k lemez	Apple II	105,—
Mychess A	Kitlinger		lemez	Cromenco	
Mychess A	Kitlinger		lemez	Osborne	
Mychess B	Kitlinger		lemez	Cromenco	
Mychess B	Kitlinger		lemez	Osborne	
Schachmeister	Levy	81	ROM	TI 99/4A	99,—
Boss	Schäfer	82	lemez	VC 20	
Grandmaster	Schäfer	83	kazetta	C64	79,—
Grandmaster	Schäfer	83	lemez	C64	79,—
The Mate		82	48 k lemez	Apple II	798,—
Chess CXL 4009		79	ROM	Atari	119,—
Chess		83	ROM	Atari	139,—
Sfinks	Fink			TRS 80	
Cyrus	Int.Softw.	81	ROM	Dragon 32	129,—
Caissa	Wrench	84	lemez	C64	79,—
Colossus 2.0	Bryant	84	26 k kazetta	C64	65,—
Chess			48 k kazetta	Oric	49,50
Chess		83	kazetta	Acorn B	
White Knight	Bryant		kazetta	Acorn B	
Alybadix (Prb)	Blom		lemez	Apple II	
Mate 4F (Prb)	Korhonen		lemez	Apple II	
Videopac C7010	Rens	82	ROM	G 7400	399,—
Schach		81	16 k kazetta	ZX81	39,50
ZX Schach 2			16 k kazetta	ZX81	65,—
Spectrum Chess		82	48 k kazetta	ZX-Spectrum	39,50
Chess The Turk Hutchby		82	48 k kazetta	ZX-Spectrum	49,—
The Chessplayer/Wren Hilton		82	48 k kazetta	ZX-Spectrum	39,50
Cyrus IS	Int.Softw.	81		ZX-Spectrum	
Petchess				CBM 8032	
Chess Bluebush				IBM PC	
Spoc. The Chessm.			lemez	IBM PC	

Jó alkalmat nyújtott az összehasonlításra az angol Martin Bryant alkotta Colossus program legújabb, ugyancsak Apple-re írt, 2,1 indexszel ellátott verziójának szereplése is. Ennek teljesítménye elmaradt a Sargonokétól. A Colossus 2,1 - amikor e sorokat írjuk - még nincs a piacon, de nálunk is minden bizonnyal sokan ismerik a Commodore 64-re írt, Colossus 2 programot, amely az 1984. év számítógépére készült, ez ideig talán legjobban sakkzó program. Ennél pedig - amint alkotója maga is megerősítette - az Apple verzió, már csak a számítógépek teljesítő-képessége közötti különbség miatt is, feltétlenül jobb.

Sietve hozzátesszük: a Colossus programok egy közepes játékerőjű sakkzó igényeit kielégítik; a Commodore-tulajdonosoknak e program megvételét nyugodtan javasoljuk. Igen érdekes volt Glasgow-ban a személyi számítógépekre írt programok játszmáit figyelemmel kísérni. A sorolás szeszélye két Apple-t is szembeállított egymással, a Sargon III/Macintosh, illetve a Colossus 2,1 programmal. A játszma színvonalas, mindkét fél részéről szinte hibátlan volt. A végjátékban Sargon helyzete némileg előnyösebbnek tűnt, de ennek érvényesítéséhez a gépek végjátékismeretéhez képest alapos tudásra volt szükség.

COMMODORE

- VC 20 RAM bővítő 8-16 k,
- C 64 és VC 20-ra írt szótárprogram,
- C 64 EPROM égető,
- C 64 szoftver.

ELPROTERV GMK
Telefon: 805-869

LÉPÉSRŐL LÉPÉSRE

Tud-e a sakkprogram tanulni?

Az ember folyamatosan tanul a hibáiból: minden elvesztett játszmából leszűr bizonyos következtetéseket. Keresi a vesztes okát, és további játszmáiban igyekszik az elkövetett hibát elkerülni. Jó lenne valahogy ezt a képességet a számítógépbe is beépíteni, hiszen, mint látjuk, pusztán a számítógép lehetőségeinek, gyorsaságának kihasználásával nem tudunk verhetetlen sakkprogramot készíteni.

Az ember, amikor hibából tanul, gazdag fogalomkörben mozog, a sakkról meglevő igen sok ismeretébe építi be azokat a tanulságokat, amik a jövőben a hasonló vereségek elkerülésére vezethetnek. A jelenlegi sakkprogramok azonban, mint az eddigiekből kiderült, rendkívül keveset tudnak magáról a sakkról.

Az ember egy vereségből levonhat például egy olyan következtetést, hogy „ebben a változatban mielőtt a királyszárnyon támadást indítanék, le kell zárni a c vonalat”, de milyen következtetést vonhat le egy sakkprogram?

Az előző mondat egyetlen fogalmát sem ismeri. Ha megpróbáljuk az előző mondatot megfogalmazni azokkal az eszközökkel, amelyekkel a sakkprogram dolgozik, legfeljebb valami olyasmit állapíthatunk meg, hogy „az adott partiban a saját figuráimhoz rendelt mozgékonyaságérték túl magas volt, a király biztonságát kifejező szám pedig túl alacsony, ez volt a hiba oka”.

Az ezt a hibát felismertük, megtehetjük, hogy a megfelelő pontértékeket a kiértékelő függvényben módosítjuk, de itt nem a program tanult, hanem mi, aminek eredményeképpen a programot ártuk. Ha a program tanulta, akkor magától, automatikusan képes lenne a fenti következtetést levonni, és a módosítást végrehajtani.

Az első probléma az, hogy hogyan ismerje fel a program, mikor követte el a hibát. Ez többnyire még az embernek is problémát okoz, gazdag fogalomkincsével együtt; a programnak pedig szűkösen eszköztára segítségével kell megfogalmaznia olyan kritériumokat, amelyek alapján egy lépést utólag hibásnak minősíthet.

Vegyük először a legegyszerűbb esetet. Ha egy program egy játszmát elveszt, mindig kell, hogy legyen egy olyan pont a játszmában, amikor a program már maga is tudja, hogy veszíteni fog, de az előző lépésben még nem tudta. Ez azt jelenti, hogy a vizsgált fában a minimax algoritmus a fa kiinduló csúcsához veszítő értéket rendel, de az előző lépésben még nem veszítő értéket rendelt. Azt, hogy ilyen pont biztosan van a játszmában, az bizonyítja, hogy a kiinduló állásban, a játszma kezdetén, a program még nyilván nem értékeli veszteséget az állást, egy lépéssel a matt előtt pedig már biztosan annak értékeli.

Ezen a ponton első közelítésre megkapaszkodhatunk: minősítsük hibásnak azt a lépést, amit akkor tett, amikor még nem tudta, hogy veszíteni fog, de amire az ellenfél olyan lépéssel tudott válaszolni, hogy azután a program saját értékelése radikálisan megváltozott, ez már ő is tudta, hogy veszíteni fog. Természetesen semmi sem garantálja, hogy tényleg ez a lépés volt a hiba, hiszen lehet, hogy a program értékelései nem voltak pontosak, és tudtán kívül már jó néhány lépés óta veszítő állásban volt.

Az könnyen megakadályozhatjuk, hogy a program a hibásnak minősített lépést máskor is megismételje. Ehhez elegendő, ha fejlegyűzzük a program életében lejárt összes elvesztett játszmákat, és mindegyikben megjelöljük a hibásnak minősített lépést. A továbbiakban ezt a lépést örökre letiltjuk: úgy kezeljük, mintha az adott állásban szabálytalan lépés lenne.

Ennek a megoldásnak van egy távolabbi mutató haszna is: előfordulhat, hogy a program akkor is el fogja kerülni a továbbiakban a hibát, ha azt nem konkrétan a hibásnak minősített lépés okozta, hanem valamelyik korábbi. Elégé valószínű ugyanis, hogy a minimax algoritmus éppen a hibásnak minősített lépés következményeit értékelté hibásan, és már jóval korábban emiatt a változat miatt döntött egy lépés mellett. Ha ez így történt, akkor a program ezután a jóval korábban elkövetett hibát is el fogja kerülni, pusztán a letiltott későbbi lépés miatt.

A megoldást mégsem érezzük kielégítőnek: idővel nagyon elszaporodnak a veszett játszmák, és valamennyit végignézni minden egyes lépéslehetőség megvizsgálásával rendkívül időigényes lenne.

És különben sem elegáns a megoldás.

Egy ötlettel azonban lényegesen javíthatjuk a fenti megoldás hatékonyságát, sőt általánosabbá is tehetjük. Az ötlet a következő: ne magukat a lépéseket figyeljük, hanem azokat a számokat, amiket a program a minimax algoritmus végrehajtásánál végeredményként kapott, tehát amiket a program a lépések kiválasztásánál az éppen adott álláshoz rendelt. Ezek a számok nagyon sokfélek lehetnek: egy játszma szinte biztosan egyértelműen jellemezhető pusztán azaz, ha sorban felsoroljuk ezeket a számokat. (Ez nem azt jelenti, hogy csupán a számok alapján könnyű a játszmát rekonstruálni, hanem csak azt, hogy lehetséges.) Sőt, ha csak az utolsó néhány kiértékelő függvény-értéket adjuk meg, szinte biztos, hogy már az is egyértelműen jellemzi a játszmabeli situációt.

Minél finomabb a kiértékelő függvény, annál biztosabb, hogy ez a helyzet. Ha például a kiértékelő függvény annyira durva, hogy csak az anyagi egyensúlyt számolja, akkor a rekonstrukció nem is végezhető el. A BELLE programnak az előző részben bemutatott kiértékelő függvénye egyszerűsége ellenére is szinte mindig elég a rekonstrukcióhoz.

Megtehetjük, hogy nem a konkrét, hibásnak minősített lépést tiltjuk le mindörökre, hanem azt, hogy az a néhány szám, amely a hibásnak minősített lépés megtételénél az utolsó néhány kiértékelő függvény-értéket jelezte, valaha is újra előálljon. Az eredmény lényegében ugyanaz, mintha a konkrét lépést tiltottuk volna le. Amit kockázatunk, az annyi, hogy esetleg egy egész más helyzetben is előáll ugyanaz a néhány szám, és a letiltás arra az esetre is érvényes lesz, pedig ott esetleg éppen az így letiltottnak tekintett lépés lenne a helyes. Ez a kockázat azonban elenyészően kicsi.

A letiltás viszont ténylegesen általánosabb, mintha csak a konkrét játszma konkrét hibás lépését tiltottuk volna le: például a program ezután akkor sem fogja elkövetni a hibát, ha a kritikus állás előzetes lépéscserékkel áll elő. Sőt szerencsés esetben még akkor sem, ha az állás nem pontosan azonos az elvesztett játszmabelivel, csak nagyon hasonló hozzá.

A bemutatott „tanulási” algoritmus még így is meglehetősen durva. Inkább arra hasonlít, amikor

az ember elhatározza, hogy „ezután nem játszom ezt a változatot”, mint a korábban leírt finomabb fajta tanulságokra.

Erdemes egy momentumot kiemelni, aminek a továbbgondolással sokkal finomabb tanulási eljárásokhoz juthatunk. Hogy mi lehetett a hibás lépés, abból vetjük észre, hogy a kiértékelő függvénynek a minimax algoritmus utáni végső értéke egyik lépésről a másikra ugrásszerűen lecsökkent. Ez így már jóval általánosabb, mint amennyit kihasználtunk belőle: egy ilyen radikális csökkenés mindig azt jelzi, hogy a kiértékelő függvény pontatlan volt. Az ellenfél tudott olyat lépni, ami után a kiértékelő eljárás (persze most már legalább egy lépéssel tovább tekintve előre) lényegesen más eredményt ad, mint előzőleg.

Ilyenkor a program megteheti, hogy utólag, „post festa” megvizsgálja, hogyan kellett volna a kiértékelő függvény egyes komponenseinek a súlyát úgy módosítani, hogy a két érték egymáshoz közelebb kerüljön. Jó esély van arra, hogy egy ilyen módosítás után a kiértékelő függvény pontosabb lesz, vagy legalábbis stabilabb, bár az eredményt elvileg semmi sem garantálja.

Ilyen elven működik például az amerikai A. L. Samuel által készített programja, ami tíz év alatt úgy feljavította saját kiértékelő függvényét, hogy közepes erejű dámajátékosból mestererejűvé vált: legyőzte Michigan állam dámbajnokát, bár az Egyesült Államok bajnokától kikapott.

A most leírt tanulási elveknek mint szinte minden gondolatnak a sakkprogramozás területén, számtalan változata, finomítása létezik. Valamennyi ténylegesen tanuló eljárás, abban az értelemben, hogy a programok saját játékukat változtatják, és szerencsés esetben ettől tényleg erősebbekké is válnak. Ahhoz azonban, amit az ember ért tanuláson, nem sok közik van.

Az ember a tanulás során nemcsak az algoritmusának egyes paramétereit képes megváltoztatni, hanem magát az algoritmust is: gondolkodása, egész sakkszemlélete változik, ahogy tanul és ahogy egyre erősebb sakközvető válik. A következő folytatásokban azt fogjuk vizsgálni, hogy mi az, amit az ember tud, de a sakkprogramok nem tudnak, és mi teszi milderre képessé az embert.

MÉRŐ LÁSZLÓ



a **KERSZI**

bemutatja Önnek
a SZATIR-t



a Személyi Állomány Tervezési és Információs Rendszer

elsődleges feladata az, hogy mérsékelje a személyzeti munkát végzők nagy adminisztratív megterhelését, lehetővé tegye a nyilvántartások kis helyen történő tárolását, gyors visszakeresését, illetve módosítását.

a SZATIR – jelenlegi állapotában –

a programot is tartalmazó lemezen legfeljebb 1000 személy adatait kezelheti. Egy dolgozóról 19 adatot tart nyilván. A dolgozók különböző szempontok szerint csoportosíthatók. A változási szempontok tizenkét félek lehetnek.

A PROGRAM FUTÁSÁHOZ SZÜKSÉGES ESZKÖZÖK:

- 1 DB **COMMODORE 64** SZÁMÍTÓGÉP
- 1 DB **VIC-1541** TIPUSÚ LEMEZEGYSÉG
- 1 DB **MÁTRIX** NYOMTATÓ
- 1 DB **KÉPERNYŐ** /TV KÉSZÜLÉK/

A program ára betanítással együtt 25000 Ft.

KERESKEDELMI SZERVEZÉSI INTÉZET

1134. BP. DÓZSA GYÖRGY ÚT 150.

Telefon: 202-650, 202-670

MARKETING OSZTÁLY



ZX-Spectrumra

REPÜLÉSSZIMULÁCIÓK

PILÓTASZEMMEL

Régóta alkalmaznak szimulátorokat a repülőgépek személyzetének kiképzésére. Ezek életlően megépített vezetőfülkék, amelyek az igazi repülés látványát, hangját és esetleg gyorsulásait is imitálják. Rajtuk „repülve” olyan veszélyzetek élhetők át, amelyeket valódi repülőgépen túl kockázatos lenne gyakorolni.

A személyi számítógépek fejlődésével a szimulációs programok kiterjedtek a kiképző központokból, és szélesben elterjedtek a kisgépekben. ZX-Spectrumon a következők a legismertebbek: Airliner, Delta Wing, Flight Simulation, Fighter Pilot, Nightflight.

A továbbiakban ezek közül a Flight Simulation-nel és a Fighter Pilot-tal foglalkozunk, mert ezek a legjobbak.

Kezdjük a leírások ismertetésével, hiszen sokan bizonyára ezek nélkül kapták a programot, vagy a nyelvi nehézségek akadályozták a tanulmányozást.

Flight Simulation

Kis, egymotoros, 2-4 üléses túragépet vezetünk, ami főleg sétatrepülésre készült, de néhány egyszerűbb műrepülő-figurát is elvisel. A képernyőn egyidőben láthatók a repülőgép műszerei (1. ábra) és az ablakon keresztül a környező táj, de átválthatunk a térképre is, amelyről gépünk helye és hozzátvőleges iránya is leolvasható.

Induláskor döntenünk kell, hogy milyen helyzetből akarjuk megkezdeni a repülést:

1. Valahol a levegőben repülve (in flight)
2. A leszállás végső fázisában (final approach)
3. A kifutópályáról felszállással (take off)

Ezután választhatunk, hogy akarunk-e szelet (Y) vagy sem (N). Végül megjelenik a műszerfal, és a kilátás. Kezdődhet a repülés.

A térképen két repülőtér található. Egy nagy nemzetközi (MAIN), amelynek pályája pontosan kelet-nyugati irányban fekszik és hossza több, mint 2 km, és egy kisebb, sportrepítő, amely 700 m hosszú, észak-déli betonnal. VIGYÁZAT! A térképen a pályák mérete torzított!

A pályák meghosszabbításában azok végétől 1-1 mérföldre a leszállás megkönnyítésére rádió-irányadók vannak. További három adó áll a három tó partján is. Mindig az az adó villog a térképen, amelyet a „B” gombbal kiválasztottunk, és amiről mért távolságot és irányt a navigációs műszerek mutatják.

A jobb felső sarokban 1000 láb magas hegyek vannak, amelyek ugyan a repülőgép ablakán át nem láthatók, de nekik lehet ütközni, ha 1000 lábnál alacsonyabban repülünk. Ha pedig 1000 és 2000 láb közötti magasságon szállunk fölöttük, a rádió-magasságmérő a hegyek tetejéről mért magasságot jelzi, ami persze pont ezer lábbal kevesebb, mint amit a másik magasságmérő mutat.

A biztonságos repüléshez tartsuk be az alábbi szabályokat:

- Ne ütközzünk földnek, hegynek vagy tónak!
- Ne szálljunk le behúzott futóval vagy 1000 láb/percnél nagyobb függőleges sebességgel!
- Ne guruljunk nagy sebességgel (gázzal) a pályán kívül!
- Ne repüljünk kisebb sebességgel, mint 80 csomó behúzott fékszárnyal, 70 csomó félig nyitott fékszárnyal, 60 csomó teljesen nyitott fékszárnyal!

Az utolsó szabály megszegése nem jelent automatikusan katasztrófát, csak „áteséshez” vezet, aminek következtében a gép egyik oldalra lebillen és süllyedni kezd. Ha van még elég magasság, gyors gazadással és kormányzással megmenthetjük a helyzetet.

Fighter Pilot

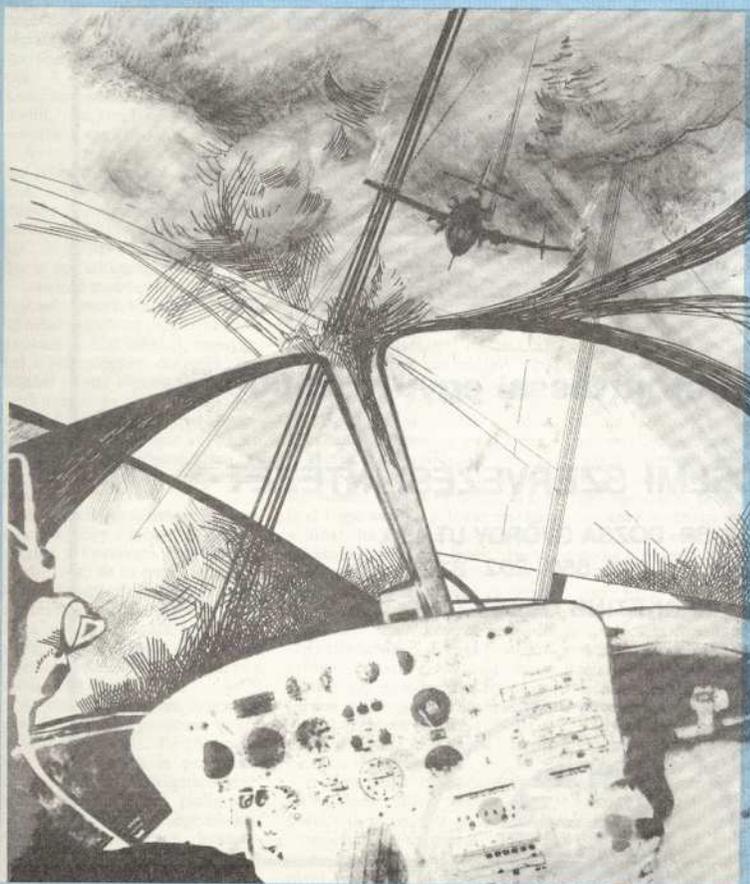
Az F 15-ös repülőgép adatai alapján készült. Teljesen műrepülhető, és a légi harcot is lehet gyakorolni rajta. Műszerfala a 2. ábrán látható.

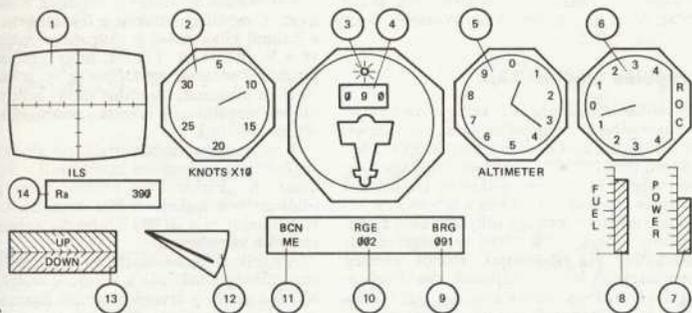
Leszállás (Landing Practice)

A gép 1700 láb magasan 6 mérfölddel a fő (BASE) pálya déli vége előtt repül a pálya irányába. A futómű kiengedve, kész a leszállásra. A leszállító műszer (ILS) vagy a fedélzeti számítógép jelzései alapján a magassági kormány, a fékszárny és a gáz segítségével szabályozható a sebesség és a süllyedés. Sikeres földetérés után a gázt teljesen le kell venni, majd a fékkel megállni.

Gyakorló repülés (Flying Training)

Kezddskor a BASE pálya déli végén áll a gép, orral északnak. Felszálláshoz adjunk gázt. A felszálló sebesség eléréskor húzzuk hátra a botot. Úgy a legrövidebb a nekifutás, ha behúzott fékkel adjuk a gázt, és csak akkor engedjük fel, azaz kezdjük meg a nekifutást, ha elérjük a teljes toloerőt. Felemelkedés után sűrűn húzzuk be a futót, mielőtt elérjük a 300 csomós

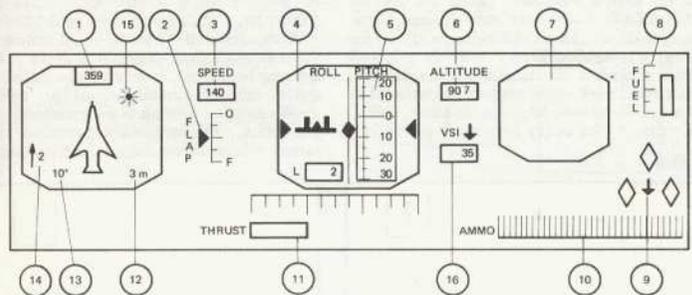




- 1 „ILS” leszálló műszer
- 2 Sebességmérő-beosztás 10 csomókbán (1 csomó = 1,85 km/ó)
- 3 Rádió-irányadó helyzete a géphez képest (villogó pont)
- 4 Repülési irány fokban
- 5 Magasságmérő
Kismutató: 1000 láb
Nagymutató: 100 láb (1 láb = 0,3 m)
- 6 „Varióméter” a függőleges sebesség mérése (beosztása 1000 láb/perc), felfelé: emelkedik, lefelé: süllyed
- 7 Motorteljesítmény
- 8 Üzemanyag-mennyiség
- 9 Iránya fokban
- 10 Távolsága mérőföldben (1 mérőföld = 1,85 km)
- 11 A beállított rádió-irányadó neve
- 12 Fékaszárny ↑ behúzva, ↓ kibocsátva
- 13 Futómű ↑ behúzva, ↓ kibocsátva
- 14 Rádió magasságmérő (csak 1000 láb alatt mér)

1. ábra. A Flight Simulator műszerfala

2. ábra. A Fighter Pilot műszerfala



- 1 Repülési irány fokban
- 2 Fékaszárny ↑ behúzva, ↓ kibocsátva
- 3 Sebesség csomókbán
- 4 Dőlési szög fokban
L: balra (left), R: jobbra (right)
- 5 Bólintási szög fokban
kék (bóltét), felfelé, sárga (világos): lefelé
- 6 Magasság lábban
- 7 „ILS” leszálló műszer vagy fedélzeti számítógép
- 8 Üzemanyag-mennyiség
- 9 Futómű ↓ (zöld) kibocsátva, ↑ (piros) behúzva
- 10 Lőzeremenyiség
- 11 Hajtómű-teljesítmény
- 12 Távolsága mérőföldben
- 13 Rádióadó vagy ellenséges gép iránya fokban
- 14 Beállított rádió-irányadó jele
- 15 Rádióadó vagy ellenséges gép helyzete a repülőgéphez képest (villogó pont)
- 16 Független sebesség (láb/másodperc)

sebességet. A felszálló sebességet csökkenthetjük a fékaszárny kinyitásával is (140 csomóról 130-ra). Gurulás közben az oldalkormányával változtathatjuk a repülőgép irányát.

Légi harc gyakorlása (Air to Air Combat Practice)

A gép az ellenséggel azonos szinten, 2 mérőfölddel mögötte repül. Váltunk át harci üzemmódra, és kapcsoljuk be a fedélzeti számítógépet. Így leolvashatjuk a másik gép irányát, távolságát és repülési magasságát. Az ellenséges gép 550 csomós sebességgel halad és nem viszonozza a tüzet, csak menekül. Ha már látja az ellenséget, próbálja befogni a célkeresztbe, és nyisson tüzet.

Légi harc (Air to Air Combat)

Ebben a fő üzemmódban meg kell oltalmaznia a négy repertert (BASE, TANGO, DELTA és ZULU). Kezddeskor a BASE pályáról felszállva a radar és a fedélzeti számítógép segítségével bemért ellenséget közelítse meg. Akkor láthatja meg, ha 1 mérőföldnél közelebb van és a magasságuk különbsége kisebb 5000 lábnál. A saját gép sérüléseit a radar közepén levő kis repülőgép színváltozása jelzi. A negyedik találat vezgates. Ajánlatos a súlyosan sérült géppel kiválni a harcból. VIGYÁZAT! Ha a gép sérülése vagy a lőszer hiánya miatt menekülünk el,

az ellenség úgy is eltalálhat egyszer, ha túl vagyunk az 1 mérőföldön vagy az 5000 lábán. Az ellenség ilyenkor folytatja útját a célba vett repülőter felé, ezzel időt nyerünk, hogy leszállhassunk valamelyik repülőterre a gépet kijavítani vagy feltölteni.

Vakrepülés (Blind Landing)

Ez az üzemmód ködöt imitál. Egyáltalán nem látjuk a horizontot, és a talaj is csak 50 láb alatti magasságból. A műszerek és a térkép segítségével lehet navigálni.

Szél (Crosswind and Turbulence)

Hatására a gép elsodródik, és véletlenszerű szélfúvások kibillentik addigi helyzetéből. Alaposan megnehezíti a repülést. Csak némi gyakorlat megszerzése után ajánlatos használni.

Az ellenfél ügyessége (Pilot Rating)

Változtatható a zöldfülűtől a szemális „ász”-ig. Az légi harcban nyilvánul meg:
 – Milyen hamar veszi észre a közeledő vadászt?
 – Milyen gyorsan fogja be a célkeresztbe az ön gépét?
 – Milyen messziről lehet lelőni?
 – Az ön gépének tulajdonságát az ellenség ügyességi foka nem befolyásolja. Nem ért vigyázni az „ással”, nagyon kemény ellenfél.

A botkormány fajtájának kiválasztása (Controls)

A biztonságos repülés szabályai:
 – Ne ütközzünk földnek vagy hegnyelnek!
 – Ne szálljunk le behúzott futóval, 10 láb/perc-nél nagyobb függőleges sebességgel vagy 2 foknál nagyobb oldalra dőléssel!
 – Ne guruljunk 250 csomónál nagyobb sebességgel!
 – Ne legyen kibocsátva a futómű 300 csomónál nagyobb sebességnél!
 – Ne változtassuk a fékaszárny állását, ha a sebesség nagyobb 472 csomónál!
 – Ne repüljünk lassabban 130 csomónál csukott fékaszárnyal és 120 csomónál, ha a fékaszárny teljesen nyitva van, mert „átések” a gép!

A navigáció

A repülésben az irányt nem égtájjakkal fejezzük ki, hanem fokkal. A teljes kör 360 fok, a 0°-os irány az észak és jobbra haladva növekszik: észak 0°, kelet 90°, dél 180°, nyugat 270°.
 Ilyen iránymeghatározással találkozunk a repülési irányt kijelző műszereknél (iránytű), a rádió-iránymérőkön és a kifutópályák fekvésének megadásánál.
 A rádióadóik iránya az az égtáj, amerre az adó a repülőgéptől esik (3. ábra). Ha a rádióadó felé akarunk repülni, addig kell fordul-

S = South = dél, E = East = kelet, W = West = nyugat). Ez a műszer csak akkor mutat, ha az adó 6 mérföldnél közelebb van.

A repülés mechanikája

A repüléshez először is le kell győzni a súlyt, azaz levegőbe kell emelni az egész szerkezetet. Ezt repülőgépeknél a szárnyakon keletkező felhajtóerő teszi lehetővé, melynek nagysága függ a sebességtől és a szárny alakjától. Ha például vízszintes repüléskor csökken a sebesség, a felhajtóerő is kisebb lesz, így süllyedni kezd a gép. A szárny alakját a fékszárny segítségével változtathatjuk. Ha kibocsátjuk, azonos sebesség esetén nagyobb lesz a felhajtóerő, kisebb sebességgel is levegőben tudunk maradni. Ezt a tulajdonságát használjuk ki leszállásnál például, ha nem akarunk nagy sebességgel földet érni, és így túl sokat gurulni a megállásig.

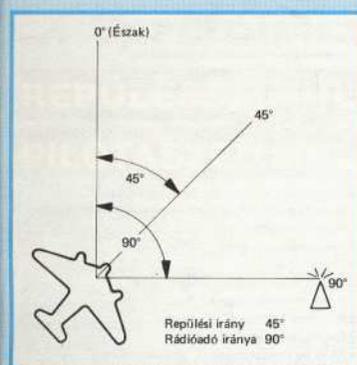
A légellenállás szintén a sebességtől, a repülőgép alakjától és a levegő sűrűségétől függ. Nagy sebesség eléréséhez nagy ellenállást kell leküzdeni, és ehhez persze nagy toleró szükséges, így lehet a sebességet gázzal szabályozni. VIGYÁZAT! A „Fighter”-ben egészen kis sebességeknél, az „átesés” közelében a szárny körül egyre erősebben kezd örvényleni a levegő, amitől megnő a légellenállás, így a sebesség csökkenését csak egyre nagyobb gázadással lehet megállítani, amit felgyorsulás követ, ezért le kell venni a gázt és így tovább. A gázzal való

„fűrészelést” úgy lehet elkerülni, ha lassításkor 3-400 csomós sebességnél kinyitjuk a fékszárnyat. A repülőgép alakját a fékszárnyon kívül a futómű kibocsátása is változtatja, megnövelve a légellenállást. Fontos, hogy az ellenállás megnövekedésével egyidőben gázt is adjunk, mert ha elkésünk, megeshet, hogy nem tudjuk időben megállítani a sebesség vesztés csökkenését, és „átesünk”.

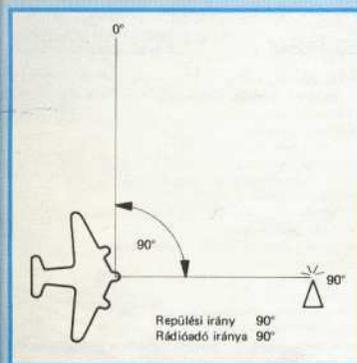
A nagy magasságban uralkodó alacsony levegősűrűség kelemesen lecsökkenti a légellenállást. A „Fighter”-nél például teljes gázzal földközelpen legfeljebb 800 csomóra tudunk felgyorsulni, míg 60 000 láb magasságban 1440 csomó a végsebesség.

A légellenállás leküzdéséhez és az emelkedéshez szükséges tolerót a hajtómű szolgáltatja. Mivel a motor a levegő oxigénjét használja fel az üzemanyag elégetéséhez, magasan a ritkább levegőben lecsökken a teljesítménye. Vadászgépeken általában van utánégető. Ez a berendezés a teljes gáz ráadás után kapcsolódik be, ha továbbtoljuk a gázkart. Nagy mennyiségű üzemanyagot fecskezend a fűvőcsőbe, megnövelve a tolerót, de még jobban a fogyasztást. A „Fighter”-en a toleró műszerének utolsó (piros) harmada az utánégető működését jelzi.

A repülés irányát fordulókkel tudjuk megváltoztatni. Ehhez elegendő a gépet a kívánt irányba bedönteni, és már megkezd az elfordulást, mégpedig annál gyorsabban, minél nagyobb szögben tértünk el a vízszintestől. Nagy bedöntésű, szűk fordulókban azonban tapasztalunk néhány furcsa dolgot. A fellépő centrifu-



3. ábra



4. ábra

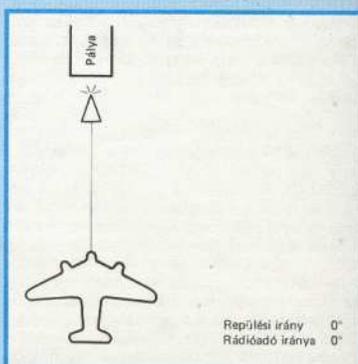
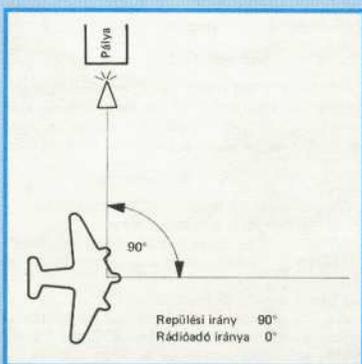
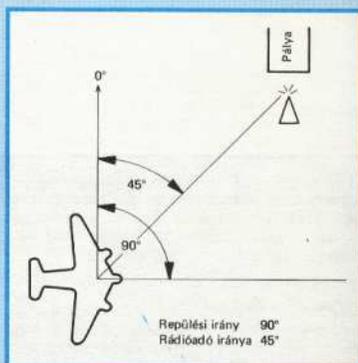
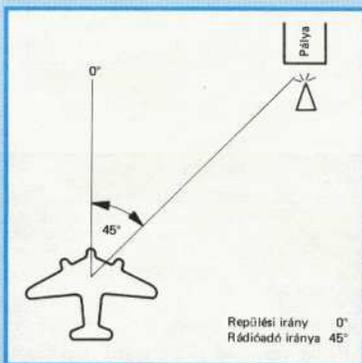
nunk, amíg a repülés iránya meg nem egyezik a rádióadó irányával (4. ábra).

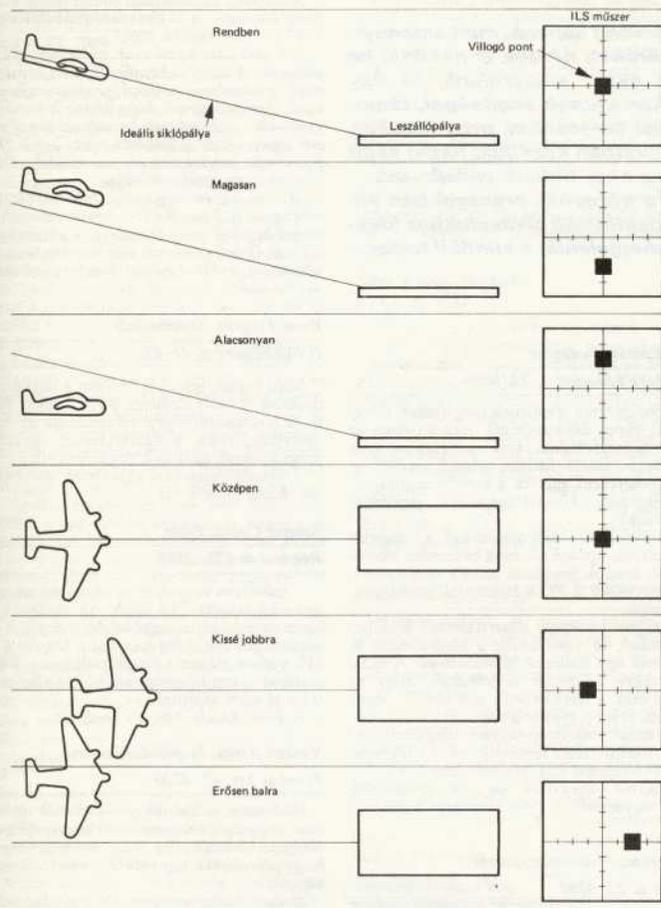
Bonyolultabb a helyzet, ha az sem mindegy, hogy milyen irányból repülünk az adó felé, mert például le akarunk szállni, és azt csak a pálya hosszában lehet. Nézzünk erre egy példát az 5. ábrán. A kiinduló helyzetből nem 45°-os irányra fordulunk, hanem túl azon, az egyszerűség kedvéért 90°-ig. Ebben az irányban haladva azt tapasztaljuk, hogy a rádióadó irányát jelző szám egyre csökken. Amikor eléri a 0°-ot, akkor vagyunk a leszállópályára meghosszabbításában. Most már csak balra kell fordulnunk, hogy megkezdhessük a leszállást. A gyakorlatban persze nem akkor kezdjük a balra fordulást, amikor a pályát már kereszteltük, hanem egy kicsit előbb, hogy ne fussunk túl rajta fordulás közben.

Leszálláshoz a legnagyobb segítséget az ILS leszálló műszer adja. Tájékoztát, hogy milyen a gép eltérése az ideális helyzettől. A 6. ábrán látható, hogy a műszer villogó pontja arrafelé mozdul el, amerre korigálni kell az irányt. VIGYÁZAT! A pályához közeledve a műszer egyre érzékenyebben reagál az eltérésekre. Ne-hogy kis magasságban hirtelen erős süllyedésbe vigyük a gépet, mert pár méterrel feljebb vagyunk a kelleteni.

A fedélzeti számítógép nagyon pontos helymeghatározást tesz lehetővé, mivel a beállított rádióadóhoz viszonyított helyzetet lábban jelzi ki égtájuk szerint (N = North = észak,

5. ábra





gális erő (vektoriálisan) hozzáadódik a repülőgép súlyához, és a gép vízszintes repülésből süllyedésbe kezd.

Igen nagy bedöntéseknél (45° fölött) észrevehetővé válik a kormányok hatásának felcserelődése: az oldalkormányval lehet az emelkedés vagy süllyedés mértékét szabályozni, míg a magassági kormány húzásával inkább a fordulót szűkíthetjük.

A repülőgépvézetés minden apró fogására természetesen már csak helyszűke miatt sem térhettünk ki, de célnk különben is csupán a programok használatának lehetőségé tétele volt. Megfelelő érdeklődés esetén később magasabb szinten visszatérhetünk a témára, megtárgyalva olyan kérdéseket, mint a műrepülés, a légi harc, a navigációs számítások vagy a bonyolult időjárásban végrehajtott leszállás. Addig is mindenkinek sok jó leszállást! HAY GYÖRGY

Program(ön)kritika

Több olvasónk is hiányolta az 1984/4. szám Százalábú rovatában, hogy a ZX-Spectrum fejlec nélküli LOAD SAVE rutinjához nem közzétöltük a kódot. Ime a decimális kód:

```

LOAD
SCF 55
LD(IX),B 221,33,N,N (kezdőcím, előbb az alacsony, aztán a magas bájtt)
LD(DE),C 17,N,N (programhossz hasonlóan)
LD A,255 62,255
CALL 1366 205,86,5 (ROM LD-bájtt rutin)
RET 201
SAVE
LD(IX),NN 221,33,N,N
LD(DE),C 17,NN
LD A,255 62,255
CALL 1218 205,194,4
RET 201
    
```



Igy is lehet javítani az 1984/6. számban megjelent 1 k-s ZX81 Autóverseny programot:
 7 LET A=A+INT(RND*3-1)+
 +(A>0)-(A<18)
 9 GOTO 10*(P=38 OR P=49)
 A javítást Máté Gábor küldte be.

Lapunk 1984/5. számában adtuk közre a játékprogram-pályázat negyedik és ötödik helyezést elért programjait. Am a szerzők neveit sajnálatos módon felcserelettük. Helyesen: a „Jégcsapolyadás” című programot Perczes Zsolt, az „Autózás” című pedig Tőkei Gyula készítette. A szerzőktől és az olvasóktól elnézést kérünk.

FLIGHT SIMULATION

↑ (7) Magassági kormány előre-nyomása (süllyedés)	X	Oldalkormány jobbra (jobbra fordulás)
↓ (6) Magassági kormány hátrahúása (emelkedés)	P	Gázadás (nagyobb teljesítmény)
→ (8) Csűrés jobbra (jobbra bedőlés)	O	Gáz levétele (kisebb teljesítmény)
← (5) Csűrés balra (balra bedőlés)	F	Fékszárny kibocsátása
Z Oldalkormány balra (balra fordulás)	D	Fékszárny behúzása
	G	Futómű kibocsátása-behúzása
	B	Rádió-irányadó kiválasztása
	M	Térkép-átkapcsoló

FIGHTER PILOT

↑ (7) Magassági kormány előre-nyomása (süllyedés)	A	Gázlevétel (kisebb teljesítmény)
↓ (6) Magassági kormány hátrahúása (emelkedés)	S	Fékszárny kibocsátása
→ (8) Csűrés jobbra (jobbra bedőlés)	W	Fékszárny behúzása
← (5) Csűrés balra (balra bedőlés)	U	Futómű kibocsátása-behúzása
	N	Rádió-irányadó kiválasztása
	M	Térkép-átkapcsoló
	B	Fék
CShift Oldalkormány balra (balra fordulás)	SSHift	„ILS” leszálló műszer – fedélzeti számítógép-átkapcsoló
Z Oldalkormány jobbra (jobbra fordulás)	C	Harci üzemmód-átkapcsoló
	O	TÜZ (harci üzemmódban)
Q Gázadás (nagyobb teljesítmény)	H	Megáll a program
	J	Továbbindul a program

KEDVES OLVASÓINK!

Eljött az az idő, amikor már sokkal több levelet kapunk, mint amennyire válaszolni tudunk. Ezért kérem a levélírókat, nézzék el nekünk, ha hasznos tanácsaikra, javaslataikra csak akkor válaszolunk, ha úgy érezzük, hogy azokra választ várnak. Sokan kérnek segítséget, támogatást, például rajzokat, alkatrészt, építési tanácsokat, programokat. Ezeket a leveleket legtöbbször ebben a rovatban közöljük, hátha segít valaki; jelenleg létszámunk mellett másra nem tudunk vállalkozni.

Meg szeretnénk szervezni, hogy a µM a környezet országokban élő magyar anyanyelvű számítástechnikai érdeklődésű olvasóinkhoz hivatatosan is eljusson. Talán mire ez a szám megjelenik, a külföldi terjesztés gondja is megoldódik.

Lássuk a leveleket!

Malinicsák János, Létavértes,

Nap u. 5. 4281

Írják meg, hova kell beküldeni azokat a programokat elbírálásra, amelyek az oktatási programajánlatban szerepelnek.

A programokat a Tudományszervezési és Informatikai Intézetbe (Bp., Egy József u. 1-9. 1111) kell küldeni. Küldhet oktatási programokat a µM-ba is, ha jók, megjelentejtük.

Fekete Sándor,

Románia

Szeretnék az Önök mikroklubjának levelező tagja lenni. Persze lehet, hogy erre nincs mód, de próbáljuk meg a lehetetlent.

Szeretnék magamnak csinálni egy személyi számítógépet, ami persze dokumentáció nélkül lehetetlen. Márpedig nekem semmiféle dokumentáció nincs, tehát Önöktől kérek segítséget.

Levélét átadtam a HCC elnökének, dr. Simonyi Endrénének. Lemásoltam, és elküldtem az Eptisünk számítógépeti sorozat eddig megjelent cikkeit, remélem, sikerül a számítógépet megépítenie.

Németh Róbert, Várpalota,

Táncsics u. 2. 8100

A nyár folyamán külföldi túránkról egy Laser 210 típusú 8 kbájtos számítógépet hoztunk, amely kiegészül egy 16 kbájtos memóriabővítővel. Az annyira áhított gépet tehát megkaptam, de sajnos nem nagyon tudom használni, mert nem tudok hozzájutni magyar fordítású gépkönyvhöz, illetve használatát magyarul utasítóshoz. Sajnos megfelelő fordítót nem találunk a környezetünkben. Mivel a Spectrum géphez több magyar nyelvű anyag is rendelkezésre áll, így próbáltuk ehhez hasonlóan működtetni, de sajnos nem egyezik meg a billentyű jelrendszerében. Ennek ellenére volt már sikerélményem is, a gépkönyvben lévő kisebb programmal. A beütött programot a gép beveszi, működött, magóra felveszi és a magnószalagról visszajátssa. De a gyári programos 16 kbájtos kazetát nem veszi be, és azt sem tudom, hogyan lehetne a grafikáját finomítani.

Példával külföldre utazóknak. Nagyon nagy kockázatot vállal, aki olyan számítógépet vásárol, amelyből kevés vagy egy sincs az országban. Nem tud használható leírást szereztetni, nincs a gépnek szervező és nem tud programot cserélni sem. Sajnos!

Lehel László, Budapest,

Rosenberg házaspár u. 23. 1054

– Rendszeresen jelennek meg cikkek arról, hogy új gépek beszerzésénél mire ügyeljen az esetleg tapasztalatlan, szakismeretekkel nem rendelkező vásárló. Milyen nyugati gépeket érdemes beszerezni, melyek a kompatibilitás, alkalmazhatóság követelményeinek legjobban megfelelők?

– Rendszeresen tájékozottassanak a legújabb termékekről, típusokról, ezek beszerzési lehetőségeiről, áraitól.

– Ismertessék a PC-k behozatali szabályait, tudnivalóit.

– Magyar kisgépek ismertetésénél közöljenek árakat, ez vonatkozik a hirdetésekre is, mivel csak így lennének informatívak. A hazai hirdetésekre különben is jellemző, hogy az árakról csak a legritkábban esik szó. Jó lenne szakítani ezzel a gyakorlattal.

– A hazai mikrógép-helyzet leírásánál nagyobb objektivitásra törekedjenek, s a takargatást, kendőzést hagyják más lapoknak.

Azt hitük, objektívak vagyunk, ezután még jobban vigyázunk. A többi javaslatot hasznosítjuk.

Gál Ferenc, Mosonmagyaróvár,

Tarcsai u. 22. 9200

Lapjukat számról száma nagy odafigyeléssel olvasom, 8. osztályos fiammal együtt. Nagy örömről szolgál a Tévé BASIC is, melyben az Önök érdemei elvitathatatlanok. Jó volna, ha lapjukban legalább néha helyet szorítanának a VC–20-szal kapcsolatos dolgoknak is. Ismerteink közül egynek ZX81, ötnek VC–20 gépe van.

Az első VC–20 „programka” már megjelent az 1985/1. számban, de atig kapunk VC–20 programot. Nem küldene?

Póli Róbert, Foktó,

Duna u. 9. 6331

Ha a lap valóban laikusok, kezdők számára is készült, akkor szerintem a kezdők számára érdekes témákkal kellene foglalkoznia. Gondolok itt például a BASIC vagy bármely más programozási nyelv ismertetésére, több számon keresztül, méghozzá általános szinten is, mivel ilyen szakirodalmat nehéz találni.

A másik dolog, ami igencsak nehezényezek, hogy az 1984/3-as számban kiadott jelent-

kezési lapot a BASIC tanfolyamra kitöltöttem, postára adtam, de visszajelzés nem érkezett.

Szeretném hasznosítani eddigi tudásomat, és programjaimat is ki akarnám próbálni a klubokban. Hol vannak ezek?

Sok javaslatra közül csak személyeiket közhölhetek. A lapot a kezdőknek is és a már komoly gyakorlatot szerzett programozóknak is írjuk. Arra számítunk, hogy a kezdők hamarosan gyakorlati szakemberek lesznek, és akkor szívesen lapozzák át a korábbi számokat is. Ezért beláthatja, nem lenne jó, ha minden cikkünk alapfokú ismereteket közlőne.

Mire ez a szám megjelenik, mindenki biztosan megkapta a Tévé BASIC visszaigazolást és a vizsga helyéről szóló értesítést. A klubok jegyzéke az Őletben jelenik meg folyamatosan. Héttalap lévén, az Őlet gyorsabban jut az olvasóhoz, mint a µM.

Rumi Frigyes, Szobathely,

Petőfi Sándor u. 41–43.

Mint sokan, így én is örültem a játékprogramoknak, de más területek is érdekelnek. Például: a mikroszámítógép alkalmazása az önálló nyelvtanulásban, a háztartásban, egy önálló programozott házi hi-fi stúdióban stb.

A témajavaslatokkal egyetérték, azért közöljük, hátha akad író is.

Szinyei Csaba, Gönc,

Rákóczi út 171. 3895

7. osztályos vagyok és az iskolában számítástechnika fakultációra járok. Az iskolának még nincs számítógépe, legközelebb Encsen, a gimnáziumban van. Meg szeretném tanulni a BASIC nyelvet, de nincs hozzá irodalmam. Kérem, ajánljon olyan könyvet, amiből az alapismereteket el lehet sajátítani.

A Tévé BASIC könyv a legújabb.

Vaszko Attila, Hajdúböszörmény,

Fazekas krt. 47. 4200

Hallottam a Számítógép-kezelők műsorában, hogy a jelentkezési lapok hamar elfogytak. Nagyon örülök, ha meg tetszene engedni, hogy jelentkezési lap nélkül is részt vehessek a vizsgán.

Kedves Attila, remélem, Te is megkaptad már az értesítést, hogy hol tudsz vizsgázni. Várom!

Kovács Gábor, Sellye,

Ifjúság u. 41. 7960

8. osztályos tanuló vagyok. Ebben az évben kezdem foglalkozni a számítógéppel. Annnyira megtetszett, hogy úgy döntöttem, hozzákötöm az életemet. Erről volna két kérdésem: lehet-e gimnáziumba jelentkezni, és onnan olyan főiskolára mennem, ahol számítógépes foglalkozást tanulhatok? És ha igen, melyik ez a főiskola?

Örülök, hogy ilyen szép pályát választottál. Úgy tudom, hogy aki jó tanuló, az biztosan felveszik a gimnáziumba, ezért kívánom, hogy nagyon jó legyen a bizonyítványod. Számítás-technikát ma már minden egyetemen és főiskolán tanítanak. Ha számítógép-tervezéssel szeretnél foglalkozni, akkor például a Kandó Kálmán Villamosipari Műszaki Főiskolát ajánlom, ha programozni, szervezni akarsz, akkor pedig a Pénzügyi és Számviteli Főiskolát. Azt hiszem, nem tévedek, hogy négy év múlva, amikor főiskolát

kell választanod, már bizonyára nagyobb választódnak ajánlani. Készséges levelet küldtem.

Cszmadia Tamás, Komló,

Gorkij u. 13. 7300

A μ M szakembergárdája fejtsen ki nagyobb aktivitást más lapokban is (természetesen ismeretterjesztési célokkal). Azaz lényegesen fokozni kell a nyomást a társadalomra. Ez pedig 30-40 ezres példányszámmal nem megy. Ha tehát a példányszám nem növelhető, akkor meg kell jelenni más, nagyobb lapokban.

A μ M (az NJSZT-vel együtt) fejtsen ki nyomást az oktatásügy irányítóira, hogy a gimnáziumokban ne nagyon lehessen fakultációként pszichológiát, nevelést és hasonló tárgyakat felvenni, hanem igenis legyen több matematika, fizika, kémia, nyelvek, biológia. Ezeket a tárgyakat a továbbtanulásban a tanulók 80-90 százalékra fel tudja használni.

Megszüntették az ún. reál tagozatokat – nagy hiba volt. Most aztán hurrá, minden „elhumánosult”. Tisztelet a kivételnek! Be kellene látnunk, hogy a magyar oktatásügy évszázadok óta erős human beállítottságú volt – ami tévedés volt. Amennyiben ezt a vonalat követjük, alighanem rövidesen sehoh sem lesz Magyarország. Tévedés ne essék, nem a human ismeretek sokkalom, hanem a reál ismeretek keveslem. Amikor én érettségiztem, még nem csökkentették a művelődési anyagot, és mégis meg tudtuk tanulni. Ma is azt mondom – bár akkor nem így láttam –, hogy matematikából, fizikából többet kellett volna kapnunk. Pedig én reál tagozatra jártam.

Az eddigi oktatáspolitikai eredménye az, hogy míg a bölcsészeti károkban a túljelentkezés férfi a tiszterest is, addig a műszaki egyetemeken néha alig másfélszeres, ha nem kevesebb. Be kell látni, hogy a technikai kultúrát hátréberbe szorító oktatás a LEMARADÁS ISKOLAJA!

Lapot nehéz szerkeszteni, egyelőre a legfőbb gondunk, hogy emeljük a μ M színvonalát. Hogy más lapokban is írjunk, arra csak néha van erőnk.

Ami az oktatásügyet illeti, én is azt hiszem (ezt már néhány korábbi írásom is tanúsítja), hogy alapos reformra van szükség. En azért nem szívesen esnék a ló másik oldalára, tehát óvakodnék attól, hogy kizárólag vagy főleg reál tárgyakat tanítsanak. Szerintem nagyobb szabadságot kellene adni a tanulóknak, hogy azt tanulják, ami érdekli őket, reál és human tárgyakat együtt. Ma – véleményem szerint, ami lehet, hogy nem helyes – például a számítástechnikában olyan specialistákra lenne szükség, akik jól ismerik a szakterületet (például állattenyésztés, kórházzervezés, könyvtári rendszerek) és jó számítástechnikusok is, de az sem árt, ha a human tárgyakban is műveltek. Azt hiszem, hogy az ilyen specialisták „előállítására” a mai merev felsőoktatási rendszer nem képes, de olyan javaslatokat kellene adnunk, amelyek ezt a problémát megoldják. Nem hiszem, hogy nem fogadnák szívesen.

Kovács László, Dunajváros,

2408, pf. 24.

A televízióból értesültem arról, hogy az NJSZT figyelemmel kíséri az országban megalakuló mikroklubok működését. Barátaimmal létrehoztunk egy mikroklubot, amelynek az RCC (Rádióamatőr Computer Club) nevet adtuk. A klubhelyiség is „feljavított” alagsori helyiség, amelyet közösen takarítottunk, tapé-

táztunk és rendeztünk be. Jelenleg egy ZX81 típusú mikrogepünk van.

Pillanatnyilag öten vagyunk, de már jöttek érdeklődők, akik be akarnak lépni. Sajnos erre az egy gépre máris sokan vagyunk. Nagy gondunk, hogy a dunajvárosi Munkás Művelődési Központ elzárkózott mindenemű segítségétől. Azt állították, hogy nincs helyiségük, és hogy ott soha nem lesz mikroklub. Nem egészen egy hónappal ezután említette meg egyik ismerősöm, hogy ott is beindították egy mikroklubot! Sajnos már belépi nem tudtam.

En azt hiszem, hogy a Művelődési Központnak sem rossz, ha szakmai segítséget kapnak lelkes rádióamatőröktől. Ha kell, az NJSZT Fejér megyei vezetősége vagy a dunajvárosi csoport biztosan segít.

Zsigó Bálint, Miskolc,

Iván u. 8. 3534

Olvastam a μ M 1984/4. számának vezércikkét. Teljesen egyetértek azzal, hogy növeljük az egy évben megjelenő lapok számát. A példányszámot is emelni kellene, ugyanis néhány helyen nehezen lehet beszerezni. Nem tudtam megvenni az 1983-as számot. Kérem segítségüket, hogy ha van a szerkesztőségben még ebből, akkor küldjenek címre egy példányt. Vagy ha nincs, mondják meg, hol lehet beszerezni?

Mindenkinek üzenem, hogy az 1983-as szám egyáltalán nem kapható. Van néhány példány az 1984/3. és az 1985/1. számból, mással szerencsére nem szolgálhatunk.

Csikszenthy Csaba, Budapest,

Kádár u. 9-13/A. 1132

10 éves vagyok, és nagyon szeretném, ha felvennék a klubba. Mi ennek a feltétele?

Még alig tudok BASIC nyelven, de már tudom kezelni a Honeywell és az Apple-Macintosh gépeket. Hamarosan kapok egy ZX-Spectrum személyi számítógépet. Apukám számítógép-matematikuss, és Honeywell gépen dolgozik. Egy éve érdeklődöm a számítástechnika iránt, és gyakran beamegyek apukám munkahelyére.

A központi klubba (V., Báthori u. 16.) és az NJSZT-be is a helyszínen lehet jelentkezni. Úgy látszik, az alma (Apple) nem esett messze a fájától, gratulálok.

Horváth Attila, Pécs,

Pollack Mihály u. 5. 7633

Sajnos nem vagyok a Magazin rendszeres olvasója, és főleg azért nem, mert eddig nem is

Kabalapályázatunk véget ért!

Pályázatunkra igen sok ötletes rajz, vázlat érkezett, de helyhiány miatt csak egy részüket tudtuk közreadni. Epp ezért mondunk köszönetet mindazon játszóársainknak is, akik bár elküldték alkotásaikat, mégsem láthatták őket viszont lapunk hasábjában.

Kérjük kedves olvasóinkat, hogy nyílt levelezőlapon jelöljék meg azt a három kabalát, totemfigurát, amelyet a legszemélyesebbnek találnak.

És most jöjjön a szavazás! Határidő: augusztus 15.

Címünk: Mikroszámítógép Magazin Szerkesztősége 1027 Budapest II., Fő utca 68. Pf. 451 Budapest 1372

nagyon tudtam a létezéséről. Gimnáziumba járok, és már másfél éve foglalkozom a számítástechnikával. Szeretnék valamilyen klubba belépni, mert sokszor akadnak problémáim, amelyekre elérhető szakkönyvekben nem találok választ, de sajnos nem tudok semmilyen klubról a közelben.

A klubok listája először az Őtlet február 7-i számában jelent meg, a későbbi számokban pedig a folytatás.

Ifj. Fűri Gábor, Szeghalom,

Május 1. út 2/1. 5520

Nagy számítógép-rajongó vagyok. Sikerült vennem egy VC-20-as számítógépet, amelyet igen megszerettem, jó volt rajta dolgozni. Múlt időben mondom, mert a gépem elromlott.

Mint már megadtam, nem helyesen kezeltem. HT-hez voltam szokva, ezért az egyes elemek bekapcsolási sorrendjére nem fordítottam különösebb figyelmet. Ez volt a hiba. Később más géptulajdonosok mondták, hogy ez a géptípus igen gyengén van védve a külső hatások ellen, ezért a helytelen működtetés súlyos következményekkel járhat. Ez történt az én esetemben is. Gépem beboldult. Körtünetei a következők: Első bekapcsoláskor a képernyő elsötétül, de a belső kep és a READY jelzés nem jelenik meg. Többzori kapcsolgatás után – a kötelező szüneteket betartva – megjelenik a READY és a villogó kurzor. Ez azonban nem tart sokáig. A villogás megszűnik, és a gép érzéketlen lesz a billentyűzetre. Szóval beboldult. Többzori próbálkozás után megnyugszik; programozható lesz.

Sajnos már nem merem bekapcsolni a gépet mert félek, hogy ezzel csak romtom a helyzetet. Ezért kérem Önt, hogy segítsen rajtam. Biztosan ismer olyan szakembert, aki meg tudná mondani, mi a gép baja, és hogy lehetne megjavítani. Kérem, juttassa el hozzá leveletem. Itt vidéken nincs olyan ember, akihez fordulhatnék ezzel a számmalra igen súlyos gondnal.

Úgy tudom, hogy VC-20 szerviz nincs náhunk. Így csak a levél közzétételével tudok segíteni.

Tóth Sándor, Keszthely,

Ady Endre u. 22. 8360

A tanácsát szeretném kérni számítógép ügyben. Előre kell bocsátanom, hogy a technikai kedvelő, de ebben az ügyben abszolút laikus vagyok. Szeretnék venni egy személyi számítógépet, amelyet tudnék használni elsősorban munkám során, másodsorban itthoni célokra. Egy 200 fős üdülőben raktározásként dolgozom. Hozzám tartozik kb. 1000 fős anyag, több csoportban. Elsősorban a munkámat szeretném megkönnyíteni a géppel, az itthoni felhasználás csak jéték, szórakozás lenne.

A levélből úgy értem, hogy saját számítógépet akar venni, és ezt szeretné a munkahelyén is használni. Valószínűleg háttérterelés (mágnesleme) gépre lesz szüksége. Ebből pedig csak a Commodore 64 kapható a magánforgalomban. Többet kellene a felhasználásról tudnom, hogy segíteni tudjak.

Egyetlen kérésem maradt, hogy olvasóink ne kiméljenek bennünket. Olvasói kapcsolat nélkül nem boldogulhat egyetlen szerkesztőség sem. A legközelebbi találkozásig megköszöni a leveleket:

KOVÁCS GYÖZŐ

Bio-számítógép

Szilícium és más félvezetők helyett szerves molekulákból épül fel az a hagyományos elven működő számítógép, amelyen a japán Sharp cég szakemberei dolgoznak. A „Bio-morzsza” a molekulák nagyságrendjében működik, a mértsökkenéssel a számítási folyamat is gyorsul. A gyártás során először rendkívül vékony szerves filmet készítenek, amelynek anyagából peptidok segítségével elektrodát alakítanak ki. A ki- és bekapcsolás enzimek segítségével történik. Ebből ered a gép másik nagy előnye, ugyanis az enzimek vegyi reakciói jelentéktelen hőfejtéssel járnak.

Igazgatók az iskolapadban

Az idén ősszel szakköri jelleggel Kecskemét valamennyi általános iskolájában bevezetik a számítástechnikát. Az aktív első lépéseként az iskolaigazgatókat ültették a hazai gyártmányú iskolaszámítógépek mellé, nemcsak azzal a céllal, hogy a gépek kezelését megtanulják, hanem, hogy az alkalmazási lehetőségek megismerésével ösztöndíjazók ők az új technológia alkalmazására.

A nyolchetes tanfolyamot a kecskeméti Gépipari és Automatizálási Műszaki Főiskola szakemberei vezetik. Kecskemét 24 általános iskolájában több, mint 13 ezer diák tanul. Közülük először a hetedikesek és nyolcadikosok vesznek részt számítástechnikai képzésben, a hatodikosok pedig szakköri foglalkozásokon ismerkedhetnek a számítógéppel.

Utastájékoztató

Ferihegyen

Számítógéppel vezérelt lapozótáblák és nagyképernyős monitorok segítségével egyidejűleg 16 induló és 12 érkező járatról tájékozódhatnak majd az utasok a Ferihegyi repülőtér új forgalmi épületében. A rendszer működtetéséhez szükséges VT-20 alapú számítógépet, az azt kiegészítő különleges egységeket, valamint a szükséges programokat Székesfehérváron készítették.

Az új utastájékoztató rendszert március-április folyamán telepítették Ferihegyre, és a rendszertervező és kivitelező feladatokat ellátó VILATI irányításával augusztus végéig helyezik üzembe. A jó áttekinthetőség mellett előnye lesz a

nagyfokú üzembiztonság és a megbízhatóság is. A számítógép ugyanis a menetrend adatait szolgáltató nemzetközi rendszer meghibásodása esetén önállóan is tudja irányítani az információs hálózatot.

Ferihegy korszerű tájékoztatói rendszerének elkészítésében több külföldi és hazai vállalat vesz részt.

Tv-Computer

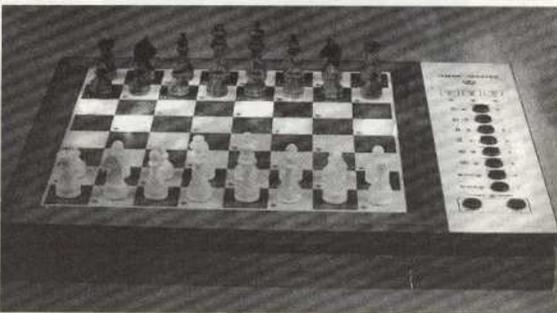
Házi számítógép gyártását kezdte meg ez év második negyedévében a székesfehérvári Videoton Elektronikai Vállalat. A tavalyi tavaszi BNV-n már sikerrel bemutatott gép magyar ékezetes billentyűzettel készül és bármilyen típusú fekete-fehér és színes televízióhoz csatlakoztatható. Az utóbbi lehetőségek már jelzi, hogy ez az első hazai gyártású, színezélesi képességgel is rendelkező házi számítógép. A gép háttértárolója háztartási kazettás magnetofon, de hajlékonylemez tárolót is illesztenek hozzá.

A gyár még az idén kétezer darabot készít a Tv-Computerből, jövőre pedig már az igényeknek megfelelően tudja növelni a gyártást.

A második sakkszámítógép

Az erfurti Mikroelektronikai Kombinátban készült el a szocialista országok első sakkszámítógépe, az SC 2, amely mellett most a közel múltban bemutatott a második típusú is: a Chess Mastert. A gép mikroelektronikai bázisát az első típusal megegyezően a Z80 mikroprocesszornak az NDK-ban gyártott, U880 elnevezésű funkcionális megfelelője képezi. A gép alapfunkciói az emberrel való játékon kívül lehetőséget biztosítanak a gépnek önmagával való játékra, két játékos számára döntőbírói szerep betöltésére. A játékos 8 erősségi fokozat közül választhat, természetesen színválasztási lehetőség is van. A gép járulékos szolgáltatásai: jó lépés tippelése, problémabevitel, a mérkőzés állásának megváltoztatása, feladványmegoldás: matt 1-4 lépésben. A Chess Master másodpercenként 15 lépésváltozást vizsgál meg, így „gondolkodási ideje” a választott erősségi fokozattól és a mérkőzés állásától függően 2 másodperc és 12 óra között változik.

Ennek a gépnek a prototípusa sikerrel szerepelt az 1983-ban Budapesten rendezett Sakkszámítógép Világbajnokságon is.



Elektronikus

útvám

Hongkongban teljesen egyedülálló módszerrel kívánják mérselteni az üzleti negyedek utcáin lassan elviselhetetlenné váló forgalmi dugókat és az ezzel járó fűlsiketítő dudálást. Az elképzelések szerint a városállam fontosabb útvonalain csúcsforgalom idején autózók út-vámot fizetnének. Ennek összegét úgy állapítanák meg, hogy a szóban forgó útvonalakon elektronikus érzékelőket helyeznének el, és a fölöttük áthaladó gépkocsikra vonatkozó információt egy központi számítógépbe továbbítanák. Az autótulajdonosoknak mintegy 50 dollárért meg kellene vásárolniuk azt a szintén elektronikus elven működő kódszám-kijelzőt, amelyet beépítenek majd a gépkocsijukba. Az autók havonta megkapják a számlát, hogy a kijelölt területeken hol és hányszor haladt keresztül, és ezért máll kell fizetniük.

Az ellenzék véleménye szerint a kormány túlságosan nagy hatású eszközt kapna az autók mozgásának figyeléséhez, a megkezdett ki-

sérleti telepítés vezetője azonban hangsúlyozta, hogy a rendőrség csupán akkor tekinthet bele az információkba, ha igen valószínű az autótulajdonos vagy más bűncselekmény.

Az 5,3 milliós Hongkongban a lakosság mindössze 8 százaléka autótulajdonos, de az egy négyzetkilométerre jutó 270 autó feltehetően a világ legnagyobb autósűrűsége. Hivatalos becslések szerint az útvám következtében a csúcsidőszakban mintegy 10 százalékkal csökkenhet az autóforgalom.

Kedves Olvasóink!

A jövőben örömmel adnánk helyet azoknak a híreknek is, amelyeket a vállalatok, szövetkezetek sajtóinformációként bocsátanak rendelkezésünkre. Amennyiben a közérdeklődésre számot tartó híreket folyamatosan beküldik szerkesztőségünkbe, bővebb, színesebb tehetjük hírrovalunkat. Máris élve e lehetőséggel, a Számítástechnika-Alkalmazási Vállalattól kapott információkból közlünk egy csokorral.

MÁTKA

Hamarosan eladó sorba kerül a MÁTKA, a Számítástechnika-Alkalmazási Vállalat Mezőgazdasági Általános Technológiai és Költéséglelmező Alkalmazás elnevezésű számítógépes programcsomagja. A szóba jöhető kerék, illetve bevők a termelőszövetkezetek és a kisebb állami gazdaságok lehetnek. A hazai fejlesztésű, valamint az IBM és vele kompatibilis személyi számítógépeken egyaránt alkalmazható program a számítvelet adta lehetőségeknél részletesebben és naprakészen, a technológiai folyamatokkal összefüggésben elemzi az önköltségeket. A program figyelembe veszi az adott mezőgazdasági üzem sajátosságait. A nyomtatott táblázatos jelentések mellett, a gazdálkodó egység vezetőinek módjában áll a személyi számítógép képernyőjén keresztül bármikor informálódni valamilyen részleg pillanatnyi költséghelyzetéről.

A rendszer első eleme, a Gépi üzem alrendszer forgalmazását rö-

videsen megkezdte a SZÁMALK. Folyamatban van a *Növénytermesztés* és az *Allattartás* alrendszer kidolgozása.

A MÁTKA rendszer bevezetése és üzemeltetése a mezőgazdaságban dolgozó vezetőktől és szervezőktől nem kíván több számítástechnikai ismeretet, mint amennyi a kézikönyvekből és néhány napos tanfolyam útján elsajátítható.

Számítógép-alkalmazók klubja

A közelmúltban újra megkezdte működését a szocialista országokban készült gépeket használó vállalatok, illetve szakemberek társadalmi szervezete, az ESZR/MSZR Alkalmazók Klubja. A Számítástechnika-Alkalmazási Vállalat Marketing Főosztályának támogatásával működő klub célja többek között, hogy lehetőséget nyújtson az információ-és tapasztalatszerésre, konzultációt biztosítson a gyártók képviselőivel, segítse a kezdő szakembereket. A klub a Neumann János Számítógéptudományi Társaság Felhasználói Körének szervezeti egységként Budapestet tevékenykedik.

ÁSZ bolt

Az ÁSZ az ÁPISZ Kereskedelmi Vállalat és a SZÁMALK közelmúltban megnyitott közös számítástechnikai szaküzlete. A Budafoki úti boltban közéleti és magánvásárlók egyaránt megtalálhatják mindazokat a termékeket, amelyek a mikro- és kisszámítógépek napi üzemeltetéséhez szükségesek: mágneslemezek, mágnesszalagok, mágnesszalag-zárógűrűket, festékkendőket, számítástechnikai szakkönyveket, különféle programokat, személyi számítógépes játékprogram-kazetákat. Az üzletben számítógéparúsítás is folyik. Erre a célra külön számítógép-bemutatótermet alakítottak ki. Az ÁSZ bolt árai több tekintetben igen alacsonyak. A tőkés import hajlékony mágneslemezek ára például mintegy 300 forinttal olcsóbb, mint a bizományi kereskedelemben.

1. Melyik a legkisebb szám, amelyet sorozatban gyártott számológépben számábrázolási alapszámként használtak?
2. Van-e olyan szám, amelynél kisebb számmal mint számábrázolási alappal nem lehet digitális számológépet építeni?
3. Mit jelent a plug-kompatibilitás?
4. Mi a bug szó helyes kiejtése?
5. a) bug, b) bag, c) bág, d) badzs, e) bádzs.
5. Mit jelent a bistabil szó?
6. Létezik-e más, például monostabil és tristabil rendszer is?
7. Van-e olyan sajtótermék, amelynek nevében számítástechnikai szakkifejezés van?
8. Ki alkotta a kibernetika kifejezést?
9. a) A. M. Turing, b) N. Wiener, c) J. von Neumann.
9. Helyes-e a következő szöveggyázat? „Akumulátor a számológépek az a része (máséger regisztere), ahová például egy művelet eredménye vagy egy tárkezes tartalma betölthető, át-tölthető, és amelynek a tartalmához egy másik szám hozzáadható, abból kivonható vagy azzal összehasonlítható.”
10. Mi az SBC?
11. Mi köze a programbeli ciklusnak a körhöz?

12. Mi a foolproof?
 13. Az ALGOL vagy a FORTRAN nyelv jelent meg előbb?
 14. Melyik nagyobb szám, a 9 alapú számábrázolási rendszerben a 6, vagy a 8 alapú számábrázolási rendszerben a 7?
 15. Melyik volt az első, sorozatban gyártott modern (elektronikus, digitális, belső programozású stb.) számológép, amit hazánkban üzembe helyeztek?
 16. Létezett-e olyan sorozatban gyártott számológép, amelynek egyetlen írással és olvasással alkalmas központi tára (ma már mindig perifériális egységként használt) dobta (mágnesezhető réteggel bevont író és olvasó fejek előtt forgó henger) volt?
 17. Mi az MTBF?
 18. Igaz-e, hogy a lyukszalagokon kör alakú, a lyukkártyákon pedig téglalap alakú lyukasztást használnak?
 19. Igaz-e, hogy „kétszeres pontossággal” (pontosabban: kétszeres szóhosszúsággal) dolgozva kétszer annyi értékes jegyű lesz a számolás végeredménye?
 20. Mit jelent a CR?
- A kérdésekre a választ a 48. oldalon közöljük.
Összeállította: TAKÁCSY ILDIKÓ

A programpályázat eredménye

Az 1985. március 22-én megrendezett II. Országos Számítástechnikai Verseny döntőjének színhelye ismét a székszárdi Garay János Gimnázium és Övönöképző Szakiskola volt. Az NJSZT, a Garay gimnázium és a µM szerkesztője közöl meghirdetett pályázatra 50 program érkezett. Ezek közül a legjobbnak ítélt 16 került a döntőbe és szerepelt a nagy nyilvánosság előtt lezajlott versenyen.

A vetélkedőt Kovács Győző, az NJSZT főtítkára nyitotta meg, majd a Commodore 64 beszélőprogramja üdvözölte a megjelent vendégeket. A zsüri egyenként beszélgetett a versenyzőkkel a bevezetett programjaikról. Ezután következett az értékelés.

A játékprogram kategóriában 10 program versenyzett. A fődíjat Herendi Miklós ABC-80 gépre írt, Kenya nevű programja kapta. A program elefantvadászatra invitál, amelyet elthüien modellez. Sok humorral kíséri az elefantvadászt – egy olyan világban, ahol minden pénzbe kerül. Ebben a kategóriában a második helyezett Gyuk Zsolt ugyancsak ABC-80-ra készült, Kereskedő című programja lett. Ez egy üzletember munkájába enged betekintést. A játékos feladata sokrétű: számolnia kell a

pénzügyi gondokkal és megfontoltan kell vásárolnia. A harmadik helyet Tihor Miklós programja, a Péter és Pál nyerte el. A program színesen mutatja a két szomszédot, akik kertjeikben fákat ültetnek, és az azokra szálló madarakat gyűjtik.

Az oktatóprogram kategória döntőjébe 6 HT program került. A legjobbnak Sándor Imre Algoritmus nevű programját ítélték, amely a számok sorba rakásának technikáját mutatja be és magyarázza el. A második helyezett egy Otto-motort bemutató program lett. Készítője Zsembery Péter. A program szemléletesen ábrázolja a szerkezet felépítését és működését. A harmadik helyezett, Márka Szabolcs programja a gázrészeszkék elrendeződését, két helyiségben található részeszkék arányát és a különböző arányok valószínűségét is ábrázolja grafikonon.

A Híradástechnika Szövetkezet a legjobb HT gépre készült programért egy számológépet ajánlott fel. A díjat Kovács Zoltán nyerte, gépi küldő játéknak kitűnő programozástechnikájáért.

A közönség díjat Tihor Miklós és Pámer Róbert kapta. A Tolna megyei Tanács különdíjat Sándor Imrénének és Rátai Istvánnak ítélték.

ADOK – VESZEK – CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hírdetéseket közlünk. A díjazás: közületeknek gépi soronként (60 karakter) 100,- Ft, magánszemélyeknek az első 50,- Ft, minden további sor 20,- Ft. Az NJSZT tagjainak az első három sor ingyenes. Hírdetéseiket a szerkesztőség címére várjuk.

- **ÚJ COMMODORE VC-20** 18 ezer forintért eladó. Telefon: 06-80-30-991
- **VC-20 ALAPEGPE** Disassembler és Assembler program, további játékprogramok eladók. Tóth Kornél, Bp. VIII., Leonardo da Vinci u. 29. 62. 8.
- **ZX-SPECTRUMIRA** programok kaphatók: havidíjakból bérelés, cselezés, keresetszint-adózás, gmk-tagok jövedelemadózása. Füller Miklós, Bonyhád, pf. 66. 7151
- **KÖZVETÍTŐJÜK** személyi számítógépek adásvételét. Alacsony áron kínálunk megvételre Commodore VC-20 és C64, valamint ZX81 számítógépekhez tartozókat és programokat. Kérje tájékoztatónkat! INTER GM. Kasposvár, Rákóczi tér 1. 7400
- **TI-99/4A-HOZ** vennek kiegészítőket és modulokat. Vári Dénes, Hódmezővásárhely, Kapitány u. 17.
- **VC-20-HOZ** magyar nyelvű programozási kézikönyv eladó. Ára 500,- Ft. Sági Endre, Ceglédbercel, Cserői u. 10. 2737. Telefon 172-411/16-53

- **SZÍVESEN RÉSZT VENNÉK A CSEREAKCIÓBAN.** Van gépem, botkormányral és magocsatlakozással. Egy cartridge kaptam is van, ALÉN ADDITION nevű, összeadón alapuló ügyességi játékkal. Tóthfaluys Péter, Bp. Klauzál tér 16. 1072
- **COMMODORE 64-RE** készült angol, amerikai, olasz játékok és utility programok cseréire magyar felhasználókkal Roberto Banfi, Via-Filippo Reina 18, 21047 Saronno (Varese), Olaszország. (Levelezés olasz vagy angol nyelven)
- **SZÖVEGFELVITELRE VÁLLALKOZÓKAT KERESÜNK** Commodore 64-re. Gép, lemezegység szükséges, nyomtató elony. Érdeklődni: 66-99-88
- **48 K-S ZX-SPECTRUMHOZ** sakk- és repülésszimulációs programok cserélnék más Spectrum játékokra. Kéméri József, Kecskemét, Botond út 2. C/IV/18. 6000
- **MEGVITELRE AJÁNLOM** VC-20-hoz és C64-hoz kapcsolható Antech számítógép-magnómat és játékkazettáimat (Commodore 64-re). Szabó József, Karcag, Kossuth tér 7-9. 5300
- **TIMEX SINCLAIR** gépetem oszilloszkópra cserélném vagy elfogadhatok áron eladnám. Juhász János, Siofok, 8601 pf. 61.
- **ELCSERÉLNÉM VAGY ELADNÁM** VC-20-as számítógépetem tartozékokkal. Cseréknél ZX Spectrumot kérek. Murin Ákos. Telefon: 649-8885

BASIC bajnokság

(Budapest, 1984.
Számítástechnikai Koordinációs
Intézet,
139 oldal. Ára: 60,- Ft.)

A Számítástechnikai Koordinációs Intézet és leányvállalatai, a Sci-L és a Sctel az utóbbi években a Budapesti Nemzetközi Vásáron számítógépes BASIC-vetélkedőt rendeztek, amelyek sikere arra ösztönözte a szervezőket, hogy a verseny feladatait könyv formájában is kiadják. Így az olvasók az egyes feladatokat hosszabb ideig tanulmányozhatják, és a megoldások kiválasztása közben programozási ismereteiket is bővíthetik. A vetélkedőre eredetileg csak MO8X gépre írt programokat a könyv szerkesztője a Magyarországon jelenleg legelterjedtebb mikroszámítógépekre is átiratta. Ezáltal lehetővé válik a HT-1080Z, a Commodore 64 és a Sinclair Spectrum gépeken egyaránt megoldott feladatok összehasonlítása.

Hice, G. F. - Turner, W. S. - Cashwell, L. F.: Számítógépes rendszerek fejlesztésének módszertana

(Budapest, 1983.
Műszaki Könyvkiadó,
433 oldal. Ára: 78,- Ft.)

Vége megindul a szervezeten és ellenőrzöttén végzett számítástechnikai munka is hazánkban. Végül! Sok-sok éves szónoklás és „fogjuk meg és vigyék!” stílusú konferenciáisi, irányelvi, bizottsági és határozatos utáni.

Egyetlen könyvtől? Hiszen egy fecske nem csinál nyarat. Csinálni ugyan nem csinál, de igaz. De annyit egyetlen fecske is el tud érni, hogy eszünkbe juttatja a nyarat. Ez a könyv azonban sokkal többet ért el annál, hogy eszünkbe juttassa, hogy van gondosan szervezett és végigvezetett számítástechnikai munka is. Azt is elérte, hogy a munka részleteinek és fogásainak – ha nem is minden részletben vitán felül álló, de – nagyon gazdag, a gyakorlatban vizsgázott készlete is összegyűjtve, rendezetten a kezünkbe került.

Ennek következtében viszont elke-rehethetlenné válik, hogy meginduljon a számítástechnikai munka értékelése, ha nem is rendszeresen és intézményesen, de ösztönösen és szubjektíven. Ma nálunk ez is óriási eredménynek számít. Meggyeizzük, hogy komoly értékelésműtől alapunka évtizedek óta nem látott napvilágot Magyarországon.

Egy, a számítástechnikai tevékenység lélektani vonatkozásával foglalkozó, nem érdektelen mű néhány ével

előzött már megjelent hazánkban. (Jelenleg azonban már ez sem kapható.) Taylor 1911-ben kiadott, korszakformáló, világhírű munkaszervezési könyvét 1983-ban (72 év után!) lefordították magyarra. Sokat tanulhat belőle az is, aki a számítástechnika szervezésével foglalkozik.

Ezzel aztán ki is merült a számítástechnika alkalmazás módszertana iránt érdeklődőket segíteni tudó, jelentősebb magyar nyelvű művek listája. Egy kiemelkedő munkát meg meg kell említenünk, Fuori, D'Arco és Orilla „Számítógépek kezelése” című könyvét, amelyben spontán érvényesült a munkaszervezési szemléletmód. (E kiváló munka megjelentése is a Műszaki Könyvkiadó érdeme.

Hice, Turner és Cashwell munkája jellegében rokon Fuori, D'Arco és Orilla könyvével, ugyanis ez is arra törekszik, hogy a tárgyal terület minden lényeges elemét érintse. Aki elkezd olvasni, nagyon hamar azt veszi észre, hogy benne van egy léggömbben. Hogy forr, el kerülőlte a munka, amely nem esik, fejleszt, amelynek sikere nekünk is örömet szerez. És ez a munka abba-hagyhatatlanul érdekes, magunkat és a közösséget is gazdagabbá teszi. De csak ha olyan szellemben végezzük, ahogyan e könyv tárgyalja. Mert a munka lehet pusztá kin is, de lehet felémelő, felszabadító tevékenység is. Hogy mikor melyik, azt a tevékenységnek az igazsághoz, a rendezés, az értékekhez való viszonya dönti el.

Ilyen szempontból minden bizonyos vizualizáló a tevékenység értelmessége. Ehhez viszont elengedhetetlen annak minden – nemcsak pénzügyi, hanem humán – igényt kielégítő szervezősége. Ennek eléréséhez segít hozzá ez a hazai téren úttörő könyv, melynek páratlanul gazdag anyaga ismeretnek ma már nem nélkülözhető egyetlen olyan számítástechnikus sem, aki ad valamit magára és munkájára.

Könyvbírálat lett volna a feladatunk. Most azonban engedtessek meg, hogy ne legyünk únéprontok a részletek elemzése fontos, de a jelen születesnap hangulatba nem illő elvágásával, és csak néhány jótanácsot adjunk arra vonatkozóan, hogy a számítástechnika-módszertan e szerencsésen választott import eseménye harmonikus és egészséges fejlődésben tovább, és kiheverhesse a világra segítés során elszenvedett kisebb-nagyobb torzulásokat.

Módszertani munkáról, mégpedig sokakra ható módszertani munkáról léven szó, sokkal jobban kell ügyelni a szakfelfejlesztés pontosságára és jóságra. (Különbség van például tábla és táblázat között, – hogy csak egyet említsünk, azt is az enyhébbek közül.) Ugyanilyen okokból célszerű a magyarra is lefordított munkáikat a magyar címmel és adatokkal együtt kiadni megemlíteni.

Nemcsak olyan számológépes rendszerek vannak, amilyenekkel ez a nagyon értékes könyv foglalkozik. Van-nak például műszaki-tudományos,

gazdaságmatematikai, számológépes technológiai folyamatirányítási, robottechnikai, NC és CNC rendszerek is. Ezek létesítésének pedig megvannak a jellegzetes módszertani sajátosságai, amelyeknek legalább vázlatos megemléstés – kiegészítés vagy függelék formájában – teljesebbé tenné az ilyen munkát, amelynek a teljességre törekvés az egyik fő célja.

Végül pedig vannak és lesznek egyre nagyobb számban és fontosságban szemrely gépekre készített rendszerek (rendszerezékek) is. Ezek kifejlesztésének, üzemeltetésének módszertanát meg nem írta meg. Ezt a feladatot Hice, Turner és Cashwell könyve után könnyebb is és nehezebb is lesz elvégezni. Könyvebb, mert van már miből tanulni. Nehezebb, mert azt a munkát majd ehhez a könyvhöz fogják hasonlítani.

POGÁNY CSABA

James W. Coffron: Mikroprocesszoros rendszerek gyakorlati hibakeresése

(Budapest, 1984.
Műszaki Könyvkiadó,
245 oldal. Ára: 80,- Ft.)

Az amerikai szerző először a mikroprocesszorok felépítésével foglalkozik, majd a CMS hardverelemző alkalmazását írja le a legismertebb mikroprocesszorok vizsgálatánál. Nagyon ötletes módszereket mutat be, de az egyszerű vizsgálati eljárásokat is ismereti. Hasznos tanácsokat ad a hibakeresés menetéről, a vizsgáló ROM programjára vonatkozóan. Végül a TRS-80 személyi számítógépen konkrét hibakereséseket mutat be.

A könyv elterjedt célját, melyet a szerző így fogalmaz meg: „Célom az volt, hogy olyan könyvet írjak, amely világosan megmutatja, hogyan fogjunk hozzá a nyolcbites mikroprocesszor-rendszerek gyakorlati hibakereséséhez. Gondosan kifejezték, logikusan felépített, érthető és bőségesen illusztrált útmutatót igyekeztem adni olyan személyek számára, akiknek szándéka ilyen jellegű munkát végezni. Ha törekvésem siker koronázta, az olvasó a könyv elolvasása után minden félelem és habozás nélkül képes lesz a nyolcbites mikroprocesszor-rendszerekben a hibát megkeresni.”

Egyéb figyelemreméltó kiadványok

- Commodore 64.
Software alkalmazói segédlet
(Budapest, 1984.
Ipari Informatikai Központ,
416 oldal. Ára: 360,- Ft.)
- A szocialista országok
mikro- és miniszámítógépei.
Összeállította: dr. Broczok Péter
(Budapest, 1984.
Központi Statisztikai Hivatal,
37 oldal. Ára: 48,- Ft.)

VÁLTOZÁS A TUDÁSPRÓBA KÉRDÉSEIRE

1. Tudomásunk szerint a minusz kettő. A lengyel UMC-1 típusú gép minusz kettő alapú számbázisú rendszerrel dolgozott. Ilyen gép hazánkban is működött az 1960-as években.

2. Nincs, mert például bármelyik negatív egyes szám is használható számbázisú rendszer alapalmazatának. Ha pedig kijelöljük az alapműszertanított, akkor pedig kettőnek bármelyik negatív egyes kettőjegyű hatványra lehet alapozni, és 2 negatív egyes kettőjegyű hatványai között végtelen sok olyan van, amely kisebb bármilyen rögzített pozitív számmal.

Két egyes plög-komputertudásunk alapján az a kérdés feladatot, elvágásra alkalmassá, de még más egyetkezővel való csatlakoztatás lehetősége is megkezdte. Tehát például az egyes csatlakoztatás kihívás és beidőzése a másikat, a berendezésjegyes ugyanúgy működőkhöz ezrel az egységgel, mint az előbbivel.

A plög angol szó (ejtsd: plág) azt dugaszoló, csatlakoztató jelent.

4. A) A bogár, peloska (és átvitt értelemben) programhiba jelentésű angol szó helyes kiejtés bóg, rövid á-val.

5. Egy rendszer bástiba, ha 2 stabil állapota van. Ilyen például a kettőállású háztartási elektronikus bilénő kapcsoló vagy a torusz (mentő-őv), vagy a fejű gépkocsiműködő állapon, a nehézségi erőterben.

6. Létezik. Aromatizált például az inga, a kelfejeljen jatkák, a csengőgomb. Bástiba például a lencse, a lapos ellipszoid, a diszkosz. Triaxial például a forgástengelyen átmenő csatlakoztatót folytatódó. Tetrasztiba például a szabályos háromszög alapú egyesű gúla. Pentasztiba például a szabályos négyűszög alapú egyesű gúla és a szabályos háromszög alapú egyesű hasáb. Hexasztiba például a szabályos ötsűszög alapú egyesű gúla és a szabályos négyűszög alapú egyesű hasáb, például a kocka. És a szabályos hatsűszög alapú mind ismert felvételre is.

7. Több is. A BIT, a BYTE, a CHIP szakfelfejlesztés például mind ismert felvételre is.

8. Egyik sem. A szó mindegyikének korábbi eredete.

9. Nem. Félleg azért, mert így a tár minden rekesze akkumulátor lenne.

10. Single Board Computer kezdőbetűből alkotott szó, egyetlen nyomtatott áramkörű kártyán elhelyezett, így gyártott (és így forgalmazott) számológép. (Single = (itt) egyetlen, Board = (itt) áramkörű kártya, Computer = számológép.)

11. A program operátornak kapcsolatára je az ún. „folyamatábrán” grafikusban (írnyelvtől) kör, idegen szóval kibék. Ez a grafikus ciklus és a geometriai kör topológailag homeomorf, azaz kölcsönösen egyértelmű és folytonos transformációval egymásba alakítható.

12. Nagymértékben elronthatatlan, életvesztéketlen, különleges biztonságos, kezelési hibák nélkül. Olyan berendezés, illetve eljárás jelölje, amit szinte még a bolondok sem tudnak elrontani, illetve másképpen végezni, mint ahogyan azt terveztek. (Fool = bolond, proof = (itt) álló, szó szerint bolondlódó, mint például virág, tőzballó, kaspóaláló.)

13. A FORTRAN.

14. Az utóbbi.

15. Az M-3 típusú, szovjet gyártású, később hazai szakemberek által továbbfejlesztett gép.

16. A Mean Time Between Failures (Faults) kezdőbetűiből alkotott szó. A hibátlan rendelkezésre állási időszakok hosszának átlagos értéke. A megbízhatóság, a rendelkezésre állás egyik legjellemzőbbé. Jelentése: számszoros hibák közötti átlagidő.

18. Általában igen, de van (volt) lyukszámos rendszer, amely téglalap alakú és lyukkártályos rendszer, amely kör alakú lyukszámos használt.

19. Nem. Még az sem igaz, hogy minden esetben pontosabb lesz az így kapott eredmény, mint az egyszerű pontosságú” végzett számolás. Sőt pontatlanság is lehet.

20. A Carriage Return kezdőbetűiből alkotott rövidítés. Szó szerinti jelentése „kocsi visszahúzó”. A kocsi elektronikus rögzítésként a szó elejére állást vezetők bilentyű felirata. Majdnem mindig a következő sor elejére állást eredményez, miután a papír egy sorral való továbbmozdítás vezérlés, azaz szóval „sorváltás” eredményező bilentyű az utóbbi időben kihalt. A képernyő megjelenítés esetében az irányítómunkák (a cursornak) a következő sor elejére állást vezérlő (sor)ezelőiről is tudósító bilentyű felirata.



SOFTINVEST

SZOFTVER KERESKEDELMI ÉS FEJLESZTÉSI BETÉTI TÁRSULÁS 1525. BUDAPEST PF. 51. TEL.: 358-530/866.

Alapítók:

Központi Statisztikai Hivatal
Ipari Minisztérium
Művelődési Minisztérium
COMPORGAN
ÉGSZI
FÜTI
KSH SZÜV
OKISZ SZSZV.
SZÁMALK
SYSTEM

Minden piacépes SZOFTVER

termék forgalmazásában, vagy ötlet
megvalósításában partnerek vagyunk

- Közös kockázatú fejlesztések
- forgalmazói tevékenység (késztermék bizományosi értékesítések)
- ügynöki tevékenység (eladó-vevő kapcsolat létrehozása)
- fővállalkozói funkciók ellátása,
 - szakmai tanácsadás
 - információszolgáltatás

Azonnali szállítással ajánljuk a következő szoftvereket:

1. IBM PC-XT és minden kompatibilis mikroszámítógépre, ezen belül kiemelten PROPER 16-ra, VT-16-ra:
 - Főkönyvi és folyószámla könyvelés,
 - Készletgazdálkodás,
 - Bér- és munkaügyi rendszer.

Ezen géptípusra továbbá vállaljuk nagyobb projektek menedzselését, esetleg fővállalkozási formában.

2. MO8X, VT-20/A és IV, TZ-80, MULTICENTER, COMMODORE 64 gépekre ajánljuk a CSB-807 általánosan használható szoftvert, amelynek segítségével *programozói munka nélkül* létrehozhatók, karbantarthatók és a legváltozatosabb szempontok szerint lekérdezhetők egyszerű és bonyolult nyilvánvalóságok.
3. COMMODORE 610, 720 gépre:
 - Főkönyvi és folyószámla-könyvelés.
4. COMMODORE 64 gépre:
 - Széleskörűen használható vendéglátóipari és egészségügyi programcsomagok.

A SOFTINVEST minden általa forgalmazott szoftvert értékesítés előtt szakértőkkel minőségileg bevizsgáltat.

Minden szoftver értékesítésénél szolgáltatást is ajánlunk, nevezetesen tanácsadást, ha szükséges, szervezői kapacitás biztosítását.



SZOFTVER BÖRZE
a MİGÉRT Bemutatótermében!

COMMODORE 64 személyi számítógépre készült szoftverek széles választékával várjuk Kedves Ügyfeleinket.

- Altalános célú felhasználói programok
- Feladat-orientált felhasználói programok
- Oktató és Játékprogramok
- Compilerek és fejlesztést segítő programok
- Rendszer-programok
- Utility segédprogramok
- Adatátviteli programok (emulátorok)

A kiválasztott programok helyben kipróbálhatók és megvásárolhatók!

ÚJDONSÁGI ÚJDONSÁGI ÚJDONSÁGI ÚJDONSÁGI ÚJDONSÁGI!

Commodore 64 felhasználók figyelmébe ajánljuk az alábbi hardver-kiegészítőket:

- TEDIFACE INTELLIGENS TELEFON MANAGER (I.T.M.)
- CE-LUX I. és II. típusú Fényceruza
- ROMOM K 6311. mátrixnyomtató + interface

A berendezések megtekinthetők
és megrendelhetők:



Bp. VIII. Rákóczi út 57/a.

Műszer és Irodagépértékesítő Vállalat
Bemutatótermében