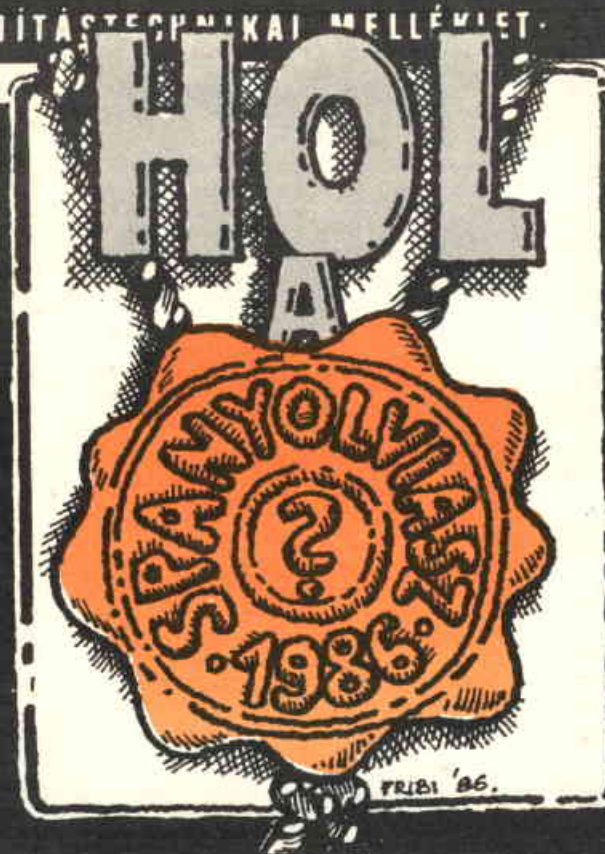


Mostanában volt szerencsém látni néhány hazai tervezésű és kivitelezésű számítógépet. Nincs ebben semmi különös, hiszen azért akadnak már magyar mikro- és személyi számítógépek, meg nagyobbcskák is, minik, meg manók. Hovatovább már a kinti áresés is begyűrűzik nálunk. Kivételesen nem a nyugati gyártmányú gépek ára, nem is a néhány ezer forintos Commodore-ra célok. Arra csak, hogy olykor már a hazai gépek ára is kezd a realishoz, a megfizethetőhöz közeledni. Látom például mostanában, hogy már a tévében is reklámozzák a számítógépeket, szövegszerkesztőket. Különböző újságok címlapjáról is gyakran köszönnek rám dúskeblű hölgyek egy-egy ismert számítógép mellől. Mondom, nagy öröm ez – nemcsak a dús kebelre gondolkok –, hiszen ha már reklámozni kell, akkor lassan van esélyünk rá, hogy eljutunk ahhoz a kánaáni állapothoz, hogy végre ne az eladó, hanem a vásárló diktálja a piacnak. Hiszen ha hirdetnek egy terméket, akkor már több van belőle, mint amennyit el lehet adni, márpedig ha többet van, akkor ebből áresés következik, meg új termékek bedobása. Összateszem a két kezem, s azt mondom, bár adná az ég, hogy ne csak álmodozunk!

Ha pedig beindul a hazai gépgyártás, ha lassanként már a vevő igényeire is hajlandók odafigyelni, nos akkor néhány alapvető változásnak kell végbemennie a gépek tervezési folyamatában. Amint az elején írtam, láttam néhány magyar gépet mostanában. Sajnos ahol kellett volna, hogy látsszon, ott nem látszott, hogy magyar. Ahol meg nem kellett volna, ott leritt róla.

Ideje lenne már a gép tervezésekor odafigyelni a felhasználói szempontokra. Sajnos, az elmúlt húsz év ipari gyakorlata nálunk nem támogatja azt a szemléletet, hogy a vásárlónak jogai vannak. A vásárló örüljön ha kap végre színes tévét, autót, meg efféléket. Hogy még igényei is legyenek? Nőde ilyet! Mintha ezt a még mindig jelenlévő gondolkodásmódot erősítenék bizonyos számítástechnikával foglalkozó cégek is.



Gyökeresen átváriálják a számítógép billentyűzetét, szinte mindenben eltérnek a hazai írógépszabványtól! Hogy a vásárlók egyike-másika néhány éve már megtanult gépelni, netán vakon gépelni, az nem számít. Most tanuljon újból! Ez nem írógép, számítógép!

Hogy a képernyőn megjelenő karakterek egyike-másika összekeverhető! Sebj, előbb-utóbb úgyis megszokja a kedves vásárló, addig legfeljebb lassabban dolgozik rajta! Hogy a gombok elhelyezése mindennek mondható, csak ideálisnak nem, leginkább a lassú munkát teszi lehetővé? Igazán semmiség.

Nem sorolom tovább a példákat. Úgysem a konkrét esetek az érdekesek, hanem maga a szemlélet. Maga az a munka-mechanizmus, amelyben úgy terveznek számítógépet, hogy a munkába be sem vonják azokat a szakembereket, akik segíthetnének. Persze, nem kell mindenhez érteni a gépek konstruktöreinek, hardver-, szoftvertervezőinek. De ha már nagyképpen verik a cégek a mellüket, hogy ilyen-olyan korszerű menedzsment technikával dolgoznak, talán bizony a menedzsmentben eljuthatnának arra a szintre, hogy egy-egy feladatra olyan teameket állítsanak össze, amelyekben helyet kapnának a legkülönfélébb szempontokat képviselő szakemberek, netán még a felhasználói szempontokat is ismerők, sőt olyanok is, akik a gépek mellett szoktak ülni, s naphosszat gyötörni azokat.

Bizonyos dolgokra – hál'istennek – nincsenek szabványok. Jó szakemberek ezt mindig valami nagyszerű megalkotására használják ki. Nálunk egyelőre ezek a nagyszerű dolgok hiányoznak. Helyette szenvedünk, s várjuk, hogy mikor jut eszébe egy magyar számítógépgyártó cégnek, hogy föl találja a spanyolviaszt. Addig meg játszunk a sikító titkárnőt. (Régi vicc, amely csak az újszülötteknek új: – Tudod milyen a sikító, titkárnő? – ??? – Mindig sikít örömeiben ha megtalál egy Angyalosi László

BELÜLRŐL

- 26 **Hiroldal** – amelyben bemutatjuk 1985 személyi számítógépei közül az Atari 520 ST-t
- 28 **Programajánlat** – egy igazán komoly felhasználói program, mégpedig egy rendező ZX Spectrumra
- 32 **Sorvezető** – amelyben a HT-használóknak megmondjuk, hogy ha új gépükbe akarják tölteni és használni az EDTASM nevű fordítót, akkor mit kell tenniük
- 33 **A következő generáció** – BIT-LET-ünk történetében ritka esemény, hogy külföldi lapból veszünk át cikket. Ezt mégis érdemesnek tartottuk!
- 34 **Mi a szoftverlopás?** – elmondja nekünk a Szerzői Jogvédő Hivatal illetékese
- 35 **Szoftverötletek** – nyomtató rutin a HT-hez: egy apró hibaigazítás a legutóbbi szoftverötletekkel kapcsolatban
- 36 **Programajánlat** – karaktertervező ezúttal a C 16-ra
- 38 **Posta** – amelyben egyértelműen közöljük sokak kérdésére, hogy márpedig a C 64 játékprogramok nem futnak a C 16-on
- 39 **Kétségnyerő** – egy pályázat, amely nagy díjai ellenére érdektelenségbe „fúlt”, mi mégis értékeliük, s a díjakat is kiadjuk!
- 40 **C 16 nyerő** – második feladata, amelyben utazunk!

PROGRAM AJÁNLAT



1. lista

```

$ 1 IF LEN K$=N OR LEN K$>4 THEN LET w=n: RETU
RN
  4 FOR i=e TO LEN K$: LET k=CODE K$(i): IF K<4
  8 OR K>57 THEN PRINT #n:"Szamot irj be!": PAUSE
  p: LET w=n: RETURN
  6 NEXT i: LET w=e: LET k=VAL K$: RETURN
  10 CLS : PRINT "Kod Tevekenyseg:" " 0. Uj al
  lomany létrehozasa" " 1. Elsodleges mezo megadasa
  a" " 2. Masodlagos mezo megadasa" " 3. Harmadlag
  os mezo megadasa" " 4. Adott rekord modositasa"
  " 5. Adott rekord torlese"
  20 PRINT " 6. Nev,adat Keresese(valogatas)"
  7. Alaphelyzet (valog=letrah)" " 8. Listazas val
  ogatott rek.-bol"
  25 PRINT " 9. Teljes progam kimentese"
  27 IF (PEEK 23642)-(PEEK 23628)<e THEN GO TO
  100
  29 PRINT "Letrehozott rekordok szama :";a(e)
  30 PRINT "Valogatott rekordok szama :";a(6)
  31 PRINT "Rekordok szama (max.) :";a(7)
  32 PRINT "Rekordok hossza (fix) :";a(t)-e
  33 PRINT "Elsodleges mezo kezdoo cime :";a(2)
  34 PRINT "Elsodleges mezo hossza :";a(3);" ("
  a(9);")"
  38 PRINT "Masodlagos mezo kezdoo cime :";a(4)
  39 PRINT "Masodlagos mezo hossza :";a(5);" ("
  a(10);")"
  40 PRINT "Novo=(0) ill. fogyo=(1) sorrend"
  41 PRINT "Harmadlagos mezo kezdoo cime :";a(11)
  42 PRINT "Harmadlagos mezo hossza :";a(12);" ("
  ";a(13);")"
  44 PLOT n,16: DRAW n,159: DRAW 255,0: DRAW n,-
  159: DRAW -255,n
  47 PLOT n,88: DRAW 255,n
  50 INPUT #n;"Tevekenyseg Kodja (0-9):"; LINE K
  $: GO SUB e: GO TO w+50
  60 IF K>9 THEN GO TO 50
  63 GO TO (K+e)*100
  70 IF K$="*" THEN GO TO t
  75 GO SUB e: GO TO a+w*t
  100 INPUT #n;"Uj allomanyt hozunk létre ?" "(ig
  en=1,nem=mas gomb):"; LINE K$: IF K$<"1" THEN
  GO TO t
  110 CLEAR 64899: LET t=8: LET n=0: LET e=1: DIM
  a(13): LET a(2)=e: LET a(4)=e: LET p=70
  115 INPUT #n;"Letrehozható rekordok maximalis"
  "szama:"; LINE K$: LET a=115: GO TO p
  125 LET a(7)=K: IF K>3000 THEN GO TO a
  130 INPUT #n;"Rekordok (fix) hosszúsága:"; LINE
  K$: LET a=130: GO TO p
  140 LET a(t)=e+K: IF a(7)*a(t)>37300 THEN PRIN
  T #n;"A memoria ehhez keves!": PAUSE p: GO TO 11
  5
  150 DIM A$(a(7),a(t)): RANDOMIZE USR 65040: GO
  TO t
  200 INPUT #n;"Elsodleges kulcs kezdoo cime:"; LIN
  E K$: LET a=200: GO TO p
  210 LET aa=K
  220 INPUT #n;"Elsodleges mezo hossza:"; LINE K$
  : LET a=220: GO TO p
  230 LET a(2)=aa: LET a(3)=K: IF K>255 THEN GO
  TO a
  240 IF aa+K>a(t) THEN PRINT #n;"Ujbol add meg
  a mezo leirasat!": PAUSE p: GO TO 200
  250 INPUT #n;"Rendezes novekvo(0) vagy""csokke
  no(mas gomb) sorrendben:"; LINE K$: IF K$="0" TH
  
```

```

EN LET a(9)=n: GO TO t
  260 LET a(9)=e: GO TO t
  300 INPUT #n;"Masodlagos kulcs kezdoo cime:"; LIN
  E K$: LET a=300: GO TO p
  310 LET aa=K
  320 INPUT #n;"Masodlagos mezo hossza:"; LINE K$
  : LET a=320: GO TO p
  330 LET a(4)=aa: LET a(5)=K: IF K>255 THEN GO
  TO a
  340 IF aa+K>a(t) THEN PRINT #n;"Ujbol add meg
  a mezo leirasat!": PAUSE p: GO TO 300
  350 INPUT #n;"Rendezes novekvo(0) vagy""csokke
  no(mas gomb) sorrendben:"; LINE K$: IF K$="0" TH
  EN LET a(10)=n: GO TO t
  360 LET a(10)=e: GO TO t
  400 INPUT #n;"Harmadlagos kulcs kezdoo cime:"; LI
  NE K$: LET a=400: GO TO p
  410 LET aa=K
  420 INPUT #n;"Harmadlagos mezo hossza:"; LINE K
  $: LET a=420: GO TO p
  430 LET a(11)=aa: LET a(12)=K: IF K>255 THEN G
  O TO a
  440 IF aa+K>a(t) THEN PRINT #n;"Ujbol add meg
  a mezo leirasat!": PAUSE p: GO TO 400
  450 INPUT #n;"Rendezes novekvo(0) vagy""csokke
  no(mas gomb) sorrendben:"; LINE K$: IF K$="0" TH
  EN LET a(10)=n: GO TO t
  500 INPUT #n;"Mely sorszamu rekordot mod.:"; LI
  NE K$: LET a=500: GO TO p
  510 LET KK=K: IF K=n OR K>a(e) THEN GO TO a
  520 INPUT #n;"Mely oszloptol:"; LINE K$: LET a=
  520: GO TO p
  530 INPUT #n;"Uj adat:"; LINE K$: IF K=n OR LEN
  K$>a(t)-K THEN GO TO a
  550 INPUT #n;"Rendezes novekvo(0) vagy""csokke
  no(mas gomb) sorrendben:"; LINE K$: IF K$="0" TH
  EN LET a(13)=n: GO TO t
  600 INPUT #n;"Mely sorszamu rek. toroljuk:"; LI
  NE K$: LET a=600: GO TO p
  610 IF K>a(6) THEN GO TO a
  620 LET a$(K,20)=" ": REM graphics 8, azaz CODE
  a$(K,20)=128
  630 LET a(e)=a(e)-e: LET a(6)=a(e): RANDOMIZE U
  SR 65233: GO TO t
  700 INPUT #n;"Az elsodleges mezoben milyen"" s
  tringet keres:"; LINE K$: IF K$="*" THEN GO TO
  t
  720 IF LEN K$>a(3) THEN GO TO 700
  730 LET a(6)=USR 65109: CLS
  740 INPUT #n;"Kimaradtakat valasztja ?" "(igen
  =*, nem=mas gomb):"; LINE K$: IF K$<"*" THEN G
  O TO t
  750 RANDOMIZE USR 65073: LET a(6)=a(e)-a(6): GO
  TO t
  800 RANDOMIZE USR 65069: LET a(6)=a(e): GO TO t
  900 INPUT #n;"Mely sorszamtól listazzunk:"; LIN
  E K$: LET a=900: GO TO p
  910 IF K>a(6) THEN GO TO a
  930 CLS : FOR i=K TO a(6): PRINT i;" ";a$(i,2 T
  O ): NEXT i: PAUSE n: GO TO t
  1000 INPUT #n;"Milyen neven mentünk:"; LINE K$:
  IF K$="*" THEN GO TO t
  1020 IF LEN K$>K THEN GO TO 1000
  1030 SAVE "Gorg-sort" LINE 1040: SAVE "Rutin"COD
  E 64899,700: GO TO t
  1040 LOAD "Rutin"CODE : GO TO t
  
```

Gyakori számítástechnikai feladat egy adatállomány rendezése különböző szempontok szerint. Pl. a könyvtárban szerzőnként és témacsoportonként is nyilvántartják az állományt. A következő BASIC program egy egyszerűsített megoldása ezen könyvtári feladatnak, és egyben kerete az ezt követő assembly nyelvű programnak.

A mintaállomány felépítése legyen a következő.

A könyvek (rekordok) száma legyen 1434, és a rekordok hossza pl. 25 karakter. Az egyes rekordokat mezőkre bontjuk, amelyek között kijelölhetünk egy elsődleges és egy másodlagos mezőt tetszős szerint menet közben is. A mindenkori rendezés alapja a kitüntetett mezőkben szereplő ASCII kódok növekvő ill. csökkenő sorrendje. Először az elsődleges mezők tartalmát hasonlítjuk össze, és ha azok azonosak, akkor a másodlagos mezők tartalma dönt a 2 rekord egymáshoz viszonyított helyzetéről. Megoldható, hogy pl. az elsődleges mező szerint csökkenő, míg a másodlagos mező alapján növekvő rendezést érjünk el.

Az egyes könyvekről pl. 5 különböző adatot (mezőt) tároljunk a következőkben

- 1-8. pozíciókon a szerző neve
 - 10-16. pozíciókon a mű címe
 - 18. pozíción minősítés pl. A = bent, B = kikölcsönzött
 - 20. pozíción a témacsoport jele pl. A = Magyar irodalom
 - 22-25. pozíciókon az olvasó száma
- A rekordok száma és hossza „kötetlen” 3 megszorítás figyelembe vételével.
- Az egyes mezők hossza legfeljebb 255 karakter;
 - a teljes adatállomány min. 500 és max. 37 300 karakter méretű;
 - a rekordok száma max. 3000 lehet.

A fenti adatállományon az 1. ábrán látható műveleteket végezhetjük el.

Az egyes tevékenységekből egyetlen „*” karakter megadásával térhetünk vissza a menühöz változtatás nélkül.

Az ábra 23-24. sorában egy új (3.) rekord felvitelének befejezése (ENTER) előtti pillanata látható.

A példában a témacsoport jelenti az elsődleges mezőt egy betű hosszan, míg a szerző neve a másodlagos mező 8 betű terjedelemben. Felmerülhet, hogy pl. az olvasók sorszáma szerint rendezve szeretnénk látni a kint levő műveket. Ebben az esetben elsődleges mezőt a 18. pozíciótól 1 hosszan jelöljük ki az 1. kód alatt. A 2. kód alatt az olvasók mezőjét adjuk meg a 22. pozíciótól 4 betű szélességben növekvő rendezési iránnyal. Ha a kint levő könyvek jele a „B”, akkor a 6. tev. kóddal kikeressük a „B”-ket, majd a 8. kód segítségével megkapjuk a kívánt listát. Az ily módon rendezett állományból további válogatással a lista szűkíthető.

Az utolsó rekord kilistázása után bármely gomb lenyomására visszatér a menü.

A BASIC felhasználók számára elegendő az 1. és a 2. lista azzal a kiegészítéssel, hogy a begépelés után GO TO 110 szükséges, majd üzemszerű használat előtt CLEAR 64956: LOAD "" az indítás. Beírásakor a 620-sz sorba ne szökőzt, hanem grafikus szökőzt írjunk!

Egy rekordot felvitelkor kiegészítjük egy státusz byte-tal, amely a rekord legelső byte-ja lesz. Ennek 7. bitje 1-es

Kód	Tevékenység:
0.	Új állomány létrehozása
1.	Elsődleges mező megadása
2.	Másodlagos mező megadása
3.	Új rekord felvitel
4.	Adott rekord módosítása
5.	Adott rekord törlése
6.	Név, adat keresése (valogatás)
7.	Alaphelyzet (valog=letrah)
8.	Listázás valogatott rek.-ból
9.	Teljes program kimentése

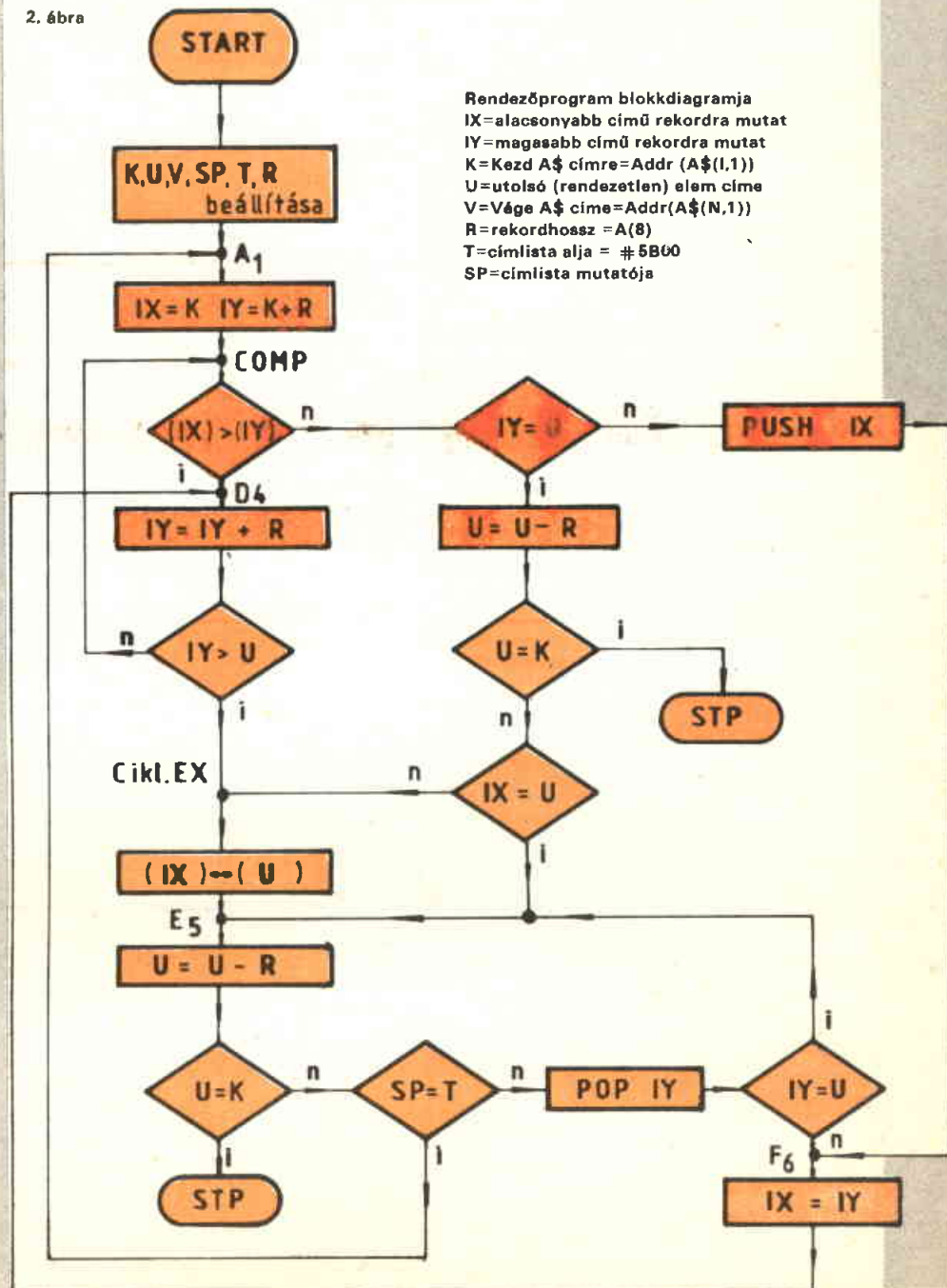
```

Letrehozott rekordok száma : 2
Valogatott rekordok száma : 2
Rekordok száma (max.) : 1434
Rekordok hossza (fix) : 25
Elsődleges mező kezdő címe : 20
Elsődleges mező hossza : 1 (0)
Másodlagos mező kezdő címe : 1
Másodlagos mező hossza : 8 (0)
Novo=(0) ill. foq40=(1) sorrend
    
```

```

3 Szerzo... Cime... S T Olv.
Ird->Anonymus Memoar A A K123
    
```

2. ábra



Rendezőprogram blokkdiagramja
 IX=alacsonyabb című rekordra mutat
 IY=magasabb című rekordra mutat
 K=Kezd A\$ címre=Addr(A\$(1,1))
 U=utolsó (rendezettlen) elem címe
 V=Vége A\$ címe=Addr(A\$(N,1))
 R=rekordhossz =A(8)
 T=címlista alja = #5B60
 SP=címlista mutatója

PROGRAM AJÁNLAT



2 lista

lesz, ha a rekordot töröljük, és 0 egyébként. A 6. bitje 1-es, ha a rekord még „felhasználatlan” és 0-s különben. A 0. bitje 0, ha a rekord „válogatott” és 1-es, ha „lemarad”. Látható, hogy ha utólag szeretnénk visszanyerni egy-egy már törölt rekordot, akkor ez a BASIC programból kilépve (!) talán elérhető, mivel a „törölt” rekordok a „felhasználatlanok” mögé kerülnek rendezéskor, míg az „élő” rekordok elől állnak.

Az assembly nyelvű program részai:

- képernyő „törlése” (PAPER 0: INK 0)
- BASIC változók keresése (A(10): A# (...)) és k#)
- új állomány kezdeti feltöltése
- string keresése mezőn belül
- a válogatásnál lemaradtak újra elérése
- kiválogatottak és lemaradtak szerepcseréje
- rendezés 3 mező szerint

Csak ez utóbbival foglalkozunk, mert e cikk fő célja ennek ismertetése.

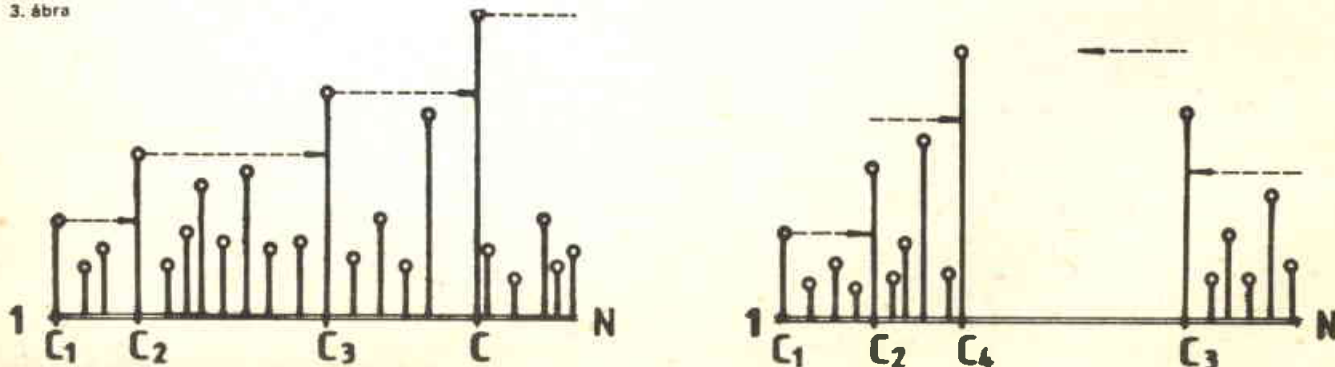
A rendezőprogram leírásánál a legegyszerűbb esetből indulunk ki. Legyen adott N db 1 b hosszúságú rekord, amelyeket növekvő sorrendbe rendezünk át. A rendezést egyoldalas maximumkereső eljárásnak tekintjük, amelyben mindig a legnagyobb elemet keressük alulról felfelé. Az állomány a „saját” helyén rendeződik egy átmeneti tároló segítségével. A 2. ábrán az egyes elemek (rekordok) egymással való kapcsolatát a vonalak hosszával jellemeztük. (A leg-rövidebb vonalhoz tartozó elemet szénjuk legelsőnek.) A szaggatott vonal azt jelzi, hogy egy adott elemtől kezdve milyen magasságú elemet keresünk, és egy ilyen hol találtunk.

1. Legyen c1, c2, c3, stb. azon rekordok címei, amelyeket az első kettő, az első három, az első N db elem közül a legnagyobbnak találtunk. Egy címet csak egyszer írunk fel és 2 azonos rekord közül a magasabb címűt „nagyobb” te-

A gépi kodu program listája, amelyet CLEAR 64956 után kell POKE utasításokkal beírni

64957	1	255	2	17	1	88	33	0	88	62	63	119	237	176	62	129	50	37
64975	64	205	14	254	17	248	63	1	56	0	237	176	62	193	50	37	64	205
64993	14	254	35	78	35	70	17	6	0	25	34	7	64	9	237	82	35	237
65011	91	35	64	237	82	34	17	64	62	75	50	37	64	205	14	254	35	126
65029	35	50	27	64	35	34	22	64	201	42	75	92	17	6	0	58	37	64
65047	190	200	126	254	96	48	8	35	78	35	70	3	9	24	238	254	128	40
65065	4	48	4	24	15	207	1	254	160	56	234	254	192	48	8	35	203	126
65083	40	251	25	24	212	254	224	56	218	35	25	25	24	244	205	189	253	62
65101	64	42	17	64	237	91	7	64	237	82	68	77	98	107	119	19	237	176
65119	201	62	55	24	1	175	50	120	254	205	189	253	42	7	64	237	91	35
65137	64	237	75	30	64	203	30	55	63	203	22	25	11	120	177	32	244	24
65155	114	205	189	253	33	0	0	34	25	64	237	91	35	64	42	7	64	34
65173	2	64	42	2	64	126	230	193	32	87	237	75	26	64	58	10	64	144
65191	60	237	75	5	64	9	71	221	42	22	64	126	35	221	190	0	40	21
65209	16	247	42	2	64	203	198	25	34	2	64	237	75	17	64	237	66	56
65227	203	24	40	229	197	237	75	26	64	24	9	221	35	126	35	221	190	0
65245	32	16	16	245	42	25	64	35	34	25	64	193	225	42	2	64	24	209
65263	193	225	24	187	205	189	253	243	253	229	237	115	32	64	49	0	88	42
65281	17	64	34	12	64	221	42	7	64	253	42	7	64	237	75	35	64	253
65299	9	1	0	0	62	1	205	195	255	58	40	64	31	58	10	64	237	75
65317	5	64	56	5	205	195	255	24	3	205	201	255	58	45	64	31	58	20
65335	64	237	75	15	64	56	5	205	195	255	24	96	205	201	255	24	91	225
65353	56	88	237	75	35	64	253	9	42	12	64	253	229	209	237	82	56	2
65371	24	183	17	48	64	221	229	225	237	75	35	64	237	176	221	229	209	42
65389	12	64	237	75	35	64	237	176	237	91	12	64	33	48	64	237	75	35
65407	64	237	176	205	221	255	33	0	0	57	17	0	88	237	82	210	6	255
65425	225	229	253	225	237	91	12	64	237	82	48	229	253	229	221	225	24	168
65443	253	229	225	237	91	12	64	175	237	82	48	4	221	229	24	234	205	221
65461	255	221	229	225	237	91	12	64	237	82	56	156	24	191	221	229	253	229
65479	24	4	253	229	221	229	209	225	9	235	9	71	26	190	194	72	255	35
65497	19	16	247	201	42	12	64	237	75	35	64	175	237	66	34	12	64	237
65515	91	7	64	235	237	82	216	237	123	32	64	253	225	237	75	25	64	251
65533	201	60	0															

3. ábra



3. lista

HISOFI GENESZ ASSEMBLER
 Copyright © HISOFT 1983
 All rights reserved

Pass 1 errors: 00

```

1 *D+
2 *C-
3 ;RENDEZO PROG. 85.07.18.
64957 4 ORG 64957
16394 5 VARS EQU 64800 ;VARIAB
16394 6 LETEZ EQU VARS ;A(1)
16399 7 KULCS1 EQU VARS+5 ;A(2)
16399 8 HOSSZ1 EQU VARS+10
16399 9 KULCS2 EQU VARS+15
16404 10 HOSSZ2 EQU VARS+20
16409 11 SELECT EQU VARS+25
16414 12 REKSH EQU VARS+30
16419 13 REKHOS EQU VARS+35
16424 14 IRANY1 EQU VARS+40
16429 15 IRANY2 EQU VARS+45
16386 16 IDEIG EQU VARS+2
16391 17 KezdA EQU VARS+7 ;A*(1)
18 ;Rendezetlenek kozt utolso
16396 19 Utolso EQU VARS+12
16401 20 VegeA EQU VARS+17 ;A*(N)
16406 21 STRING EQU VARS+22 ;K*
16411 22 STR*HO EQU VARS+27 ;K*
16416 23 Reg:SP EQU VARS+32
16421 24 NEVE EQU VARS+37
16432 25 PUFFER EQU VARS+48
26 ;Cimlista "alja"
22528 27 TETO EQU 65000
28 ;Attrib torlese
64957 29 KERESD LD BC,22FF
64960 30 LD DE,65001
64963 31 LD HL,65000
64966 32 LD A,20011111
64968 33 LD (HL),A
64969 34 LDIR
35 ;3 BASIC változo erdekes
64971 36 TombA LD A,E81 ;A(10)
64973 37 LD (NEVE),A
64976 38 CALL TALALD
64979 39 LD DE,VARS-8
64982 40 LD BC,56
64985 41 LDIR
64987 42 TombA LD A,E01 ;A*(N)
64989 43 LD (NEVE),A
64992 44 CALL TALALD
64995 45 INC HL
64996 46 LD C,(HL)
64997 47 INC HL
64998 48 LD B,(HL)
64999 49 LD DE,6
65002 50 ADD HL,DE
65003 51 LD (KezdA*),HL
65006 52 ADD HL,BC
65007 53 SBC HL,DE
65009 54 INC HL
65010 55 LD DE,(REKHOS)
65014 56 SBC HL,DE
65016 57 LD (VegeA*),HL
65019 58 K*STR LD A,E48 ;K*(1)
65021 59 LD (NEVE),A
65024 60 CALL TALALD
65027 61 INC HL
65029 63 INC HL
65030 64 LD (STR*HO),A
65033 65 INC HL
65034 66 LD (STRING),HL
65037 67 RET
68 ;Basic változok keresese
65038 69 TALALD LD HL,(23627)
65041 70 LD DE,6
65044 71 ISMET LD A,(NEVE)
65047 72 CP (HL)
65048 73 RET Z
65049 74 LD A,(HL)
65050 75 CP E00
65052 76 JR NC,TOVA1
65054 77 TOMBOS INC HL
65055 78 LD C,(HL)
65056 79 INC HL
    
```

```

65057 80 LD B,(HL)
65058 81 INC BC
65059 82 ADD HL,BC
65060 83 JR ISMET
65062 84 TOVA1 CP E00
65064 85 JF Z,HIANY
65066 86 JR NC,TOVA2
65068 87 JR PLUSZ
65070 88 HIANY RST B
65071 89 DEFB 1
65072 90 TOVA2 CP LAB
65074 91 JR C,TOMBOS
65076 92 CP E00
65078 93 JR NC,TOVA2
65080 94 NUMabc INC HL
65081 95 BIT 7,(HL)
65083 96 JR Z,NUMabc
65085 97 PLUSZ ADD HL,DE
65086 98 JR ISMET
65088 99 TOVA3 CP XE0
65090 100 JR C,TOMBOS
65092 101 INC HL
65093 102 ADD HL,DE
65094 103 ADD HL,DE
65095 104 JR PLUSZ
105 ;DIM A* utani init
65097 106 FILL CALL KERESD
65100 107 LD A,E40
65102 108 LD HL,(VegeA*)
65105 109 LD DE,(KezdA*)
65109 110 SBC HL,DE
65111 111 LD B,H
65112 112 LD C,L
65113 113 LD H,D
65114 114 LD L,E
65115 115 LD (HL),A
65116 116 INC DE
65117 117 LDIR
65119 118 RET
119 ;Lemaradtak ujra elerese
65120 120 RESET LD A,E27 ;SCF
65122 121 JR BEIR
122 ;Valogatottak es lema-
123 ;radtak szerepcserje
65124 124 INVERZ XOR A ;NOP
65125 125 BEIR LD (POKE),A
65128 126 CALL KERESD
65131 127 LD HL,(KezdA*)
65134 128 LD DE,(REKHOS)
65138 129 LD BC,(REKSH)
65142 130 BITCH6 RR (HL)
65144 131 POKE NOP ;NOP v SCF
65145 132 CCF
65146 133 RL (HL)
65148 134 ADD HL,DE
65149 135 DEC BC
65150 136 LD A,B
65151 137 OR C
65152 138 JR NZ,BITCH6
65154 139 JR SORT
140 ;k* szerinti valogatás
65156 141 VALOB CALL KERESD
65159 142 LD HL,0
65162 143 LD (SELECT),HL
65165 144 LD DE,(REKHOS)
65169 145 LD HL,(KezdA*)
65172 146 LD (IDEIG),HL
65175 147 E1 LD HL,(IDEIG)
65178 148 LD A,(HL)
65179 149 AND E01
65181 150 JR NZ,SORT
65182 151 LD BC,(STR*HO-1)
65187 152 LD A,(HOSSZ1)
65190 153 SUB B
65191 154 INC A
65192 155 LD BC,(KULCS1)
65196 156 ADD HL,BC
65197 157 LD B,A
65198 158 K2 LD IX,(STRING)
65202 159 H3 LD A,(HL)
65203 160 INC HL
65204 161 CP (IX+0)
65207 162 JR Z,ILLIK
65209 163 DJNZ H3
65211 164 LD HL,(IDEIG)
    
```

```

65214 165 SET 0,(HL)
65216 166 H4 ADD HL,DE
65217 167 LD (IDEIG),HL
65220 168 LD BC,(VegeA*)
65224 169 SRC HL,BC
65226 170 JR C,E1
65228 171 JR SORT
65230 172 ILLIK PUSH HL
65231 173 PUSH BC
65232 174 LD BC,(STR*HO-1)
65236 175 JR H6
65238 176 D5 INC IX
65240 177 LD A,(HL)
65241 178 INC HL
65242 179 CP (IX+0)
65245 180 JR NZ,VISSZA
65247 181 H6 DJNZ D5
65249 182 LD HL,(SELECT)
65252 183 INC HL
65253 184 LD (SELECT),HL
65256 185 POP BC
65257 186 POP HL
65258 187 LD HL,(IDEIG)
65261 188 JR N4
65263 189 VISSZA POP BC
65264 190 POP HL
65265 191 JR Y2
65267 192 RENDEZ CALL KERESD
193 ;Rendezes 2 mezo szerint
65270 194 SORT D1
65271 195 PUSH IY
65273 196 LD (Reg:SP),SP
65277 197 LD SP,TETO
65280 198 LD HL,(VegeA*)
65283 199 LD (Utolso),HL
200 ;IY=isebb IY=nagyobb cim
65286 201 A1 LD IX,(KezdA*)
65290 202 LD IY,(KezdA*)
65294 203 LD BC,(REKHOS)
65298 204 ADD IY,BC
205 ;Stat.byte nel. kezdete=0
206 ;stat.byte hossza=1
65300 207 B2 LD BC,0
65301 208 LD A,1
65305 209 CALL CMP_EL
210 ;Eloszites mezo megadasa
65308 211 LD A,(IRANY1)
65311 212 RR6
65312 213 LD A,(HOSSZ1)
65315 214 LD BC,(KULCS1)
65319 215 JR C,hatra
65321 216 CALL CMP_EL
65324 217 JR masod
65326 218 hatra CALL CMP_HA
219 ;Masodlagos mezo megadasa
65329 220 masod LD A,(IRANY2)
65332 221 RRA
65333 222 LD A,(HOSSZ2)
65336 223 LD BC,(KULCS2)
65340 224 JR C,Hatrab
65342 225 CALL CMP_EL
65345 226 JR B7
65347 227 Hatrab CALL CMP_HA
65350 228 JR B7
229 ;"Call comp"-RET torlese
65352 230 C3 POP HL
65353 231 JR C,B7
232 ; Vegigneztuk ?
65355 233 D4 LD BC,(REKHOS)
65359 234 ADD IY,BC
65361 235 LD HL,(Utolso)
65364 236 PUSH IY
65366 237 POP DE
65367 238 SBC HL,DE
65369 239 JR C,CiklEX
65371 240 JR B2
241 ;2 elem cserje
65373 242 CiklEX LD DE,PUFFER
65376 243 PUSH IY
65378 244 POP HL
65379 245 LD BC,(REKHOS)
65383 246 LDIR
65385 247 PUSH IY
65387 248 POP DE
65388 249 LD HL,(Utolso)
    
```

```

65391 250 LD BC,(REKHOS)
65395 251 LDIR
65397 252 LD DE,(Utolso)
65401 253 LD HL,PUFFER
65404 254 LD BC,(REKHOS)
65408 255 LDIR
256 ;Ures-e a cimlista ?
65410 257 E5 CALL Ueqk
65413 258 LD HL,0
65416 259 ADD HL,SP
65417 260 LD DE,TETO
65420 261 SBC HL,DE
262 ;Ha a cimlista ures JP A1
65422 263 JP NC,A1
65425 264 POP HL
65426 265 PUSH HL
266 ;Leemelunk egy cimet
65427 267 POP IY
65429 268 LD DE,(Utolso)
65433 269 SBC HL,DE
270 ;Ha max.,akkor JP E5
65435 271 JR NC,E5
65437 272 F6 PUSH IY
65439 273 POP IY
65441 274 JR D4
275 ;Ha utolso,akkor JP H8
65443 276 B7 PUSH IY
65445 277 POP HL
65446 278 LD DE,(Utolso)
65450 279 XOR A
65451 280 SBC HL,DE
65453 281 JR NC,H8
65455 282 PUSH IY
65457 283 JR F6
284 ;Utolsó azonos a kezdovel?
65459 285 H8 CALL Ueqk
65462 286 PUSH IY
65464 287 POP HL
65465 288 LD DE,(Utolso)
65469 289 SBC HL,DE
65471 290 JR C,CiklEX
65473 291 JR E5
65475 292 CMP_EL PUSH IY ;NOVEKVO
65477 293 PUSH IY
65479 294 JR I9
65481 295 CMP_HA PUSH IY ;CSOKKEND
65483 296 PUSH IY
297 ;Cimok tenyleges kiszam.
65485 298 I9 POP DE
65486 299 POP HL
65487 300 ADD HL,BC
65488 301 EX DE,HL
65489 302 ADD HL,BC
65490 303 LD B,A
304 ;2 elem osszehasonlitas
65491 305 COMP LD A,(DE)
65492 306 CP (HL)
65493 307 JP NZ,C3
65496 308 INC HL
65497 309 INC DE
65498 310 DJNZ COMP
65500 311 RET
312 ;Utolsó = első ?
65501 313 Ueqk LD HL,(Utolso)
65504 314 LD BC,(REKHOS)
65508 315 XOR A
65509 316 SBC HL,BC
65511 317 LD (Utolso),HL
65514 318 LD DE,(KezdA*)
65518 319 EX DE,HL
65519 320 SBC HL,DE
65521 321 RET C
322 ;Rendezett az allowany
65522 323 UTOSZO LD SP,(Reg:SP)
65526 324 POP IY
65528 325 LD BC,(SELECT)
65532 326 EI
65533 327 RET
65534 328 END
    
```

Pass 2 errors: 00
 Table used: 019 frsra 1000

kintjük. Ezeket a címeket feljegyezzük találati sorrendben egy címlistára, amelyen (többnyire) c1 a legelső.
 2. Végignézve az összes rekordot a legnagyobb elemet megtaláltuk, de ennek címe (c) már nem kerül címlistánk tetjére. Most ha kell, akkor a legnagyobb elemet felcseréljük a legutolsó elemmel. Így a „rendezetlen” elemek száma csökken. Most ezek között keressük legnagyobbat.
 3. A legnagyobb elem vagy a címlista tetjén álló (ha a címlistánk nem üres) vagy a c címtől kezdődően fordulhat elő. (Ha a lista üres, akkor az 1. ponttól folytatódik a rendezés.) A keresést egy pontból (c3) és egy intervallumból (c-től (N-1)-ig terjedően) álló halmazon folytatjuk tovább a teljes rendezésig. A c3-at a további keresés előtt levesszük a listáról.

Megjegyzések:

- Ha BASIC programunkat megszakítjuk akkor GO TO 10-zel, „melegen” indíthatjuk.
- A mezők hossza legalább 1 legyen.
- Ha nincs szükség ténylegesen másodlagos mezőre, akkor átmeneti megoldásként válasszuk másodlagos mezőnek az elsődleges mező első karakterét.
- Ha a címlista üres, akkor az 1-től „kezdődhet” a keresés.
- Ha a címlista „tele” van, akkor az állomány már rendezett.
- Két elemet legfeljebb egyszer hasonlítunk össze.
- Mozgatásra (2 elem felcserélésére) akkor kerülhet sor, ha a magasabb elem a helyére kerül.
- A kétoldalas maximumkeresés lefolyása a 3. ábrán látható.
- A maximum keresése alulról indul felfelé.
- Ha a c1 felkerült a címlistára (az esetleges c2 cím találati miatt), úgy a keresés az állomány felső végétől kezdve lefelé folytatódik a c3 címig.
- Ezután a c2-től felfelé keressük a c4-es címet.
- Ezzel az eljárással az esetek nagy részében a második legnagyobb elem címét kevesebb összehasonlítással kapjuk meg.

A rendező program próbafutásakor 5 ill. 10 byte hosszú rekordokat készítettünk a ROM 0-s címétől kezdődő átlagos rendezetlenségű halmazából. A rekordhossztól függetlenül „tekinthető” a rendezési idő e próbánál.

A próbaállomány kijelölése a menü 0-ban szerepel, míg annak feltöltéséhez az alábbiakat írjuk a programhoz:

```
145 POKE 65114,33: POKE 65115,n: POKE 65116,n
150 DIM a$(a(7),a(t)): RANDOMIZE USR 85097
160 POKE 65114,107: POKE 65115, 119:POKE 65116,19
170 LET a(e)=a(7): LET a(6)=a(e)
180 FOR i=1 TO a(e): LET a$(i,e)
    = " ": NEXT i: GOTO t
```

Rekordszám	átl. idő	„legrosszabb”
500 db	15 sec	20 sec
1000 db	59 sec	76 sec
2000 db	227 sec	305 sec

A „legjobb” idő mind a három esetben néhány másodperc.
 Csendes István, 1022 Budapest, Herman O. u. 2.



EDTASM program a HT-1080Z/64-re

A HT-1080Z iskolaszámítógépre – tudomásom szerint – két ASSEMBLER fordító van forgalomban. Az egyik EDTASM (TRS-80), a másik az EDI. Sajnos már találkoztam az EDTASM-re átkerestelt EDI-vel is. Ez az átnevezés rossz ötlet volt! Megtetszött, mert egy program két elnevezést kapott, ráadásul olyat, amiből az egyik már forgalomban volt. Én sokkal jobban szeretem az – eredeti – EDTASM fordítót használni.

Nemrégiben megjelentek a 48 K RAM-mal rendelkező, 64 K-nak mondott HT-1080Z/64 számítógépek. Amikor az EDTASM-ot betöltöttem ebbe a gépbe, furcsa dolgot tapasztaltam: a kiírt szöveg olvashatatlanná lett. Az újabb HT nem egészen kompatibilis az előzőekkel! Ezekben a 0-31-es POKE-kódok esetében a képernyőre az ékezetes betűk és egyéb speciális jelek jelennek meg. (Az első típusú gépeknél ezen kódok esetén is az ABC betűi találhatók!) Miután az EDTASM-ot az új gépeken is szerettem volna használni, hozzáálltam a fordítóprogram kisebb módosításához. Az eredményt ezúton adom közre.

Az ASSEMBLER fordító sajnos nem használja a videóra vonatkozó DCB-t, ezért ennek a módosítása nem vezet eredményre. Meg kellett keresnem azt az utasítást, amelyik a képernyőre viszi az információt. Két ilyen is találtam, azonban elég csak az egyikbe beleavatkozni. A 455EH címen található a következő utasítás:

```
LD (HL), A
```

A HL regiszterpárban van a képernyő aktuális címe, az A regiszterben a kiviendő karakter. A következő byte-tól pedig beírtam egy ugrást a saját rutinomra, melyet olyan helyre helyeztem el, amelyet a fordító nem használ.

```
JP 416DH
```

(Ezt a címet a LEVEL III használja!)

A rutinnak két feladatot kell ellátnia:

1. Amennyiben a kiírt jel POKE-kódja kisebb, mint 27, akkor ehhez hozzá kell adni 64-et (ezek a betűk).

2. A fordítóprogramból történő kiugrás 3 byte-ot igényel, az ezen a helyen eredetileg található utasításokat az új helyen pótolni kell.

1. A karakter módosítása:

```
CP 27
JR NC,ELO
SET 6,(HL)
```

A programrészlet csak a betűk esetén módosítja a kódot. A SET utasítással a „64 hozzáadása” történik meg. Ezt azért lehet így elvégezni, mert a HL címen található byte-ban a 64 értékkel rendelkező bit biztosan 0 (hiszen kisebb, mint 27).

2. Két utasítást kell pótolni:

```
INC HL
LD A,(16424)
```

Ezután vissza lehet térni az ASSEMBLER fordítóba. Az ugrás helyett lehetne használni a CALL-RET utasításokat, de jelen esetben ezt a megoldást jobbnak tartom. (Nem kell vizsgálni a stack telítettségét).

A teljes rutin a következőképpen néz ki:

```
NAME EDTA48
00100 EDTASM - 48 K RAMMAL RENDEL-
    KEZŐ GÉPRE
00110 ORG 416DH
00130 CP 27
00140 JR NC,ELO
00150 SET 6,(HL)
00160 ELO INC HL
00170 LD A,(16424)
00180 JP 4563H
00190 ORG 455FH
00200 JP 416DH
00210 END 468AH
```

Ez a rutin a fordítóval kivihető kazettára úgy, hogy később SYSTEM-mel betölthető legyen, vagy a gépi kódot monitorban be lehet gépelni. Használatkor mind a két esetben a fordítót kell előbb betölteni!

A közölt utasítások gépi kódja:

```
FE1B, 3002: CBF6, 23: 3A2840: C36345
```

Ezt a 416DH címtől kell beírni, a következőt pedig a 455FH-től: C36D41

Ezután a fordítóprogram indítható. Az indítási címe: 468AH. Amennyiben READY?-re kerül sor és még szükséges a fordító, szintén ugyanezen a címen indítható. Ebben az esetben újraindítás előtt az előzőekben leírt rutint újból be kell vinni, mert a felhasznált RAM-területet az inicializáló rutinok felülírják.

Mikes Gábor

KERAVILL MEV
 ELEKTRONIKAI
 MÁRKABOLT
 BP.V. MŰZEUM kft. 11.
 MIKROELEKTRONIKA:
 A JÖVŐ A JELENBEN.

 FÉLVEZETŐK,
 INTEGRÁLT ÁRAMKÖRÖK,
 MIKROPROCESSZOROK
 ÉS CSATLAKOZÓK.
 SZAKTANÁCSADÁS CSOMAGKÜLDŐ SZOLGÁLAT.

Cikkünk a Personal Computing egyik tavalyi számában megjelent írás kivonata.

A professzionális személyi számítógépek világa érdekes átalakuláson megy át napjainkban. Az Intel 8088 mikroprocesszor, ami lehetővé tette az IBM PC és MS-DOS rendszerű társainak megalkotását, feladta vezető pozícióját fiatalabb testvére az Intel 80286 javára. Az új, sokoldalú processzor jóvoltából új számítógép generáció van születőben, melyet a nagyobb sebesség (legalább 6 MHz-es ütemjel), a kibővített RAM terület és nagykapacitású háttértárak jellemeznek.

Úgy tűnik, hogy a sokat módosított MS-DOS rendszer végre összerendeződött, a kompatibilitás ismét előtérbe került. Kinek sikerült mindent létrehozni? A jelok az IBM-re utalnak, melynek csúcstechnológiája összehozta az 180286-ot a továbbfejlesztett memóriákkal és a hatásosan kibővített MS-DOS szabvánnyal. Így jött létre az új IBM AT. Azután jöttek a követők – és a nevek a PC kategóriájából ismerősen csengenek: Compaq, Texas Instruments és Kaypro.

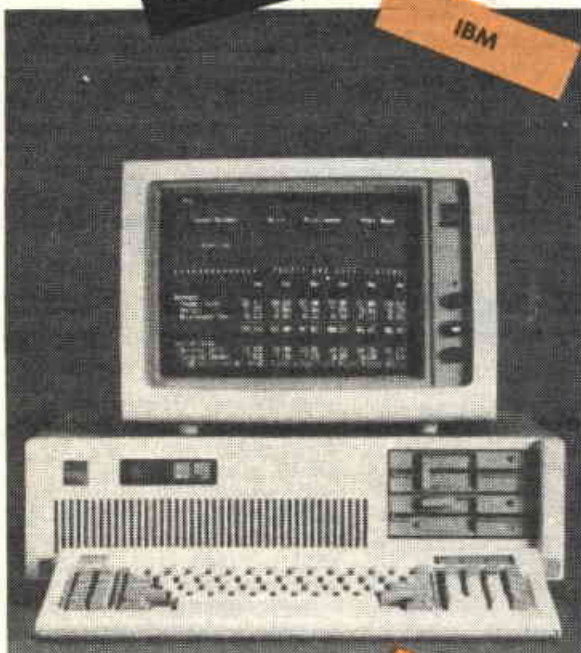
A Kaypro 286i az AT kompatibilis gépek családjának legegyszerűbb és legolcsóbb tagja. Jellemzői: 6 MHz-es órajel, 512 K memória, mely 15 Mbyte-ig bővíthető, valamint két darab floppy meghajtó, egyenként 1,2 Mbyte. Ára: 4550 dollár, ebben az operációs rendszer ára nem foglaltatik benne. Ezek a jellemzők szinte tökéletesen megegyeznek az AT tulajdonságaival.

A Compaq és a TI viszont – bár az AT árát nagyjából tartják – a gép tulajdonságait kissé felülről közelítik. A Compaq Deskpro 286 gépe 6, illetve 8 MHz-es órajellel működik, 512 K memóriával és egy 1,2 Mbyte-os floppyval, illetve egy 30 Mbyte-os lemezegységgel rendelkezik. A készülék ára MS-DOS rendszerrel 6000 dollár. A memória a bővítősatlakozók igénybevétele nélkül 2,2 Mbyte-ig bővíthető, a csatlakozók segítségével 8,2 Mbyte-ig. A Compaq Portable 286-ot csaknem ugyanilyen adatok jellemzik, de ez a gép – mint neve is mutatja – egy kizárólagos helyet kapott.

A TI cég Business-Pro-ja 6 MHz-es órajellel fut, 512 K memória áll rendelkezésre, mely 15 Mbyte-ig bővíthető, valamint 1,2 Mbyte floppy és 21 Mbyte lemezkapacitás. Hogy ezeket az igen drága gépeket el lehet adni, az a szoftver fejlesztésnek köszönhető. Az első előrelépés a rendszerprogramok területén következett be az MS-DOS rendszer továbbfejlesztésével. A nagyobb memóriák felhasználásának látványos lehetőségét hozhatja meg a szoftverigények napjainkban érezhető növekedése. A mesterséges intelligencia fogalmának kiterjesztése és az egységek ún. szakértői rendszerekbe vagy felhasználás-specifikált munkahelyekre történő szervezése valószínűleg az AT osztályú gépekre vár. De ennél sokkal hétköznapiabb síkon is jelentkezik az igény felhasználóhoz közeli „barátságos” és hatásos programok írására, melyet az új gépek memóriamérete és sebessége jelentős mértékben megkönnyít.

A 80286-os generáció gépeinek ropant kapacitása viszont előbb-utóbb fájdalmas döntésre kényszeríti a szoftverfejlesztőket. A kis sebességre, kevesebb memóriára és háttértárra írt programok ugyanis soha nem fogják egy olyan Deskpro 286i képességeit kidomborítani, mely 8MHz ütemfrekvenciával, 5 Mbyte RAM területtel és 70 Mbyte lemezterülettel működik. Ezek az új „mega-gépek” nemcsak arra születtek, hogy a ma feladatait hatósabban hajtsák végre, de olyan feladatokat is kikényszerítenek, melyeket az előző generáció meg sem próbálhatott.

A következő generáció

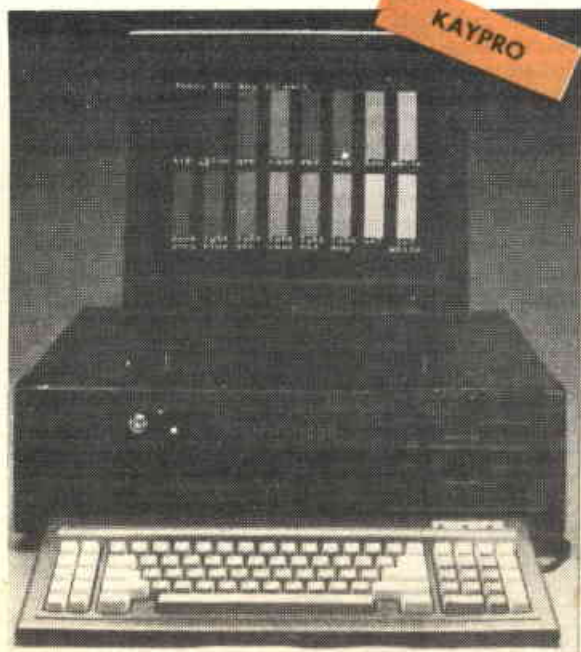


IBM

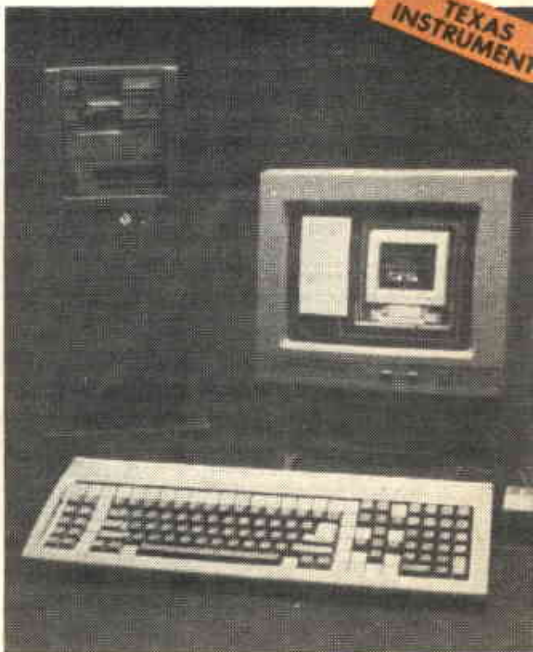


PERSONAL COMPUTING

COMPAQ



KAYPRO



TEXAS INSTRUMENT