

..... Show must go on !

TV-Computer

TVC-sek havonta megjelenő, belső terjesztésű kiadványa. 92.július
Alapította: TVC-Központ Szerkesztő: Vindics István 18.szám

Kicsit nehéz helyzetben vagyok most.

Ezt a kis cikket akkor írom, amikor már az árság többi része teljesen elkészült, s így utólagosan értékelhetem a helyzetet.

A júniusi szám bejelentése - az árság esetleges megszűnéséről - nagyon váratlanul ért. Talán mondhatom azt is, hogy legtöbbünket kis szorongással töltött el, hogy mi lesz ezután. Többen jelezték, hogy kellene az árság, hiszen gyakorlatilag ez az egyetlen hely, amire a TVC-sek számíthatnak. Az országos lapok közül egyedül a Mikrovilágnak volt néhány próbálkozása, de most hónapokig szó sem volt benne a gépünkről. (Igaz, most, hogy elhatároztam: nem érdemes megvennem, megjelentettek ismét egy oldalt, de csak ennyit érdemlünk?)

Elhatároztam, valamit tenni kell. Néhány levélváltás után kiderült, nem sok "pályázó" van a szerkesztői tisztre, de végleges megoldás még nem született. Egyelőre annyiban állapodtunk meg, hogy a 18. és 19. számot megcsinálom, aztán majd meglátjuk. Ha van érdeklődés a lapra, és más, előnyösebb megoldás nincs, akkor folytatom ezt a munkát.

Sajnos nincsenek kiterjedt kapcsolataim, ez rányomja bélyegét erre a néhány oldalra is. Így utólag áttekintve számomra nem túl változatosak a témák - azért remélem sokan hasznát veszik a leirtaknak. Őszintén remélem, hogy lesz némi vállalkozó kedv bennetek, és elkülditek a véleményeket, problémás kérdéseket, közlésre szánt anyagokat, hogy a következő szám ne a szerkesztő, hanem az olvasó "képére" formálódjon.

Talán elnézést kell kérnem, hogy gyakorlatilag a fél árság a saját játékaimról szól. Mentő körülmény csak az lehet, hogy ebben az évben (a WAR-on kívül) nem is hallottam új játékról. Ha ez csak az én tájékozatlanságomat mutatja, vagyis mégis létezik új játék, szívesen leközlöm a leírását. Hasonló a helyzet a gépi kóddal, most kicsit sokra sikeredett (vagy a többi dolog kevés?). No mindegy.

Mindezek ellenére - jeleztem, hogy ez a rész készül utoljára - lenne egy fontos dolog, ami kimaradt a leírásokból. Ez a hiány a játékok irányításával kapcsolatos. Általában a programok elején megkapod a megfelelő információt - legalábbis a billentyűzet használatáról. Megnyugtatóan közölhetem, hogy az összes új játékom játszható billentyűkkel, de joystickal is. Elkerülendő a belső botkormány "nyúvására" kísértését, legtöbbször a B botkormánnyal mennek a programok. Egyetlen program van, amelynek az irányító billentyűt "eltitkoltam". Ez az OSZKAR, irányítása: f-fel, @-le, [-bal,]-jobb, RETURN-tárgy felvétele.

Kérem, olvassátok el a mellékelt tájékoztatót is. Ebben megtalálható az árság ára is. A játékok eladásából származott eddigi bevételem csak azt teszi lehetővé, hogy egy számot előlegezzek meg mindenkinek. A 19. számot csak akkor tudom küldeni, ha az árát rendeztéték.

TARTALOM:

Levellezési rovat (helyett)	2. oldal
/néhány apróság, ami lehet, hogy Téged is érdekel.	
Gépi kód - abszolút kezdőknek	2. oldal
/assembler nélkül könnyebb?	
UPM lemezmeghajtó (2)	3. oldal
/közvetlen töltés a képernyőre - VT-DOS-ban is!	
Hardver: Eprom-törölő	3. oldal
/avagy "vak vezet világtalant" ?	
Walltris leírás	4. oldal
/új variáció egy ismerős témára.	
SN6-ASSEMBLER	4. oldal
/izelítőül az utasítások listája.	
STV 92/2	5. oldal
/a csomagban szereplő 3 Primo játék leírása.	
Gépi kódú programozás [TVCM]	6-7. oldal
/avagy egy "doboz" a képernyőre.	
ATOMIX	8. oldal
/egy tanulságos programhiba.	
A nyomtatásról	8. oldal
/amely kimeríthetetlen téma.	

Mi lesz a következő számban (19)?

- UPM sorozat befejezése (röviden)
- Gépi kód kezdőknek (sokkal hosszabban)
- Egy színezési algoritmus (basiscben és gépi kódban, avagy hogy lehet 4-szintű üzemmódban 7 szint - vagy többet - használni?)
- GRIFF (egy grafikai programnyelv) ismertetése)
- Jótanácsok a gépi kódú programozáshoz (TVCM-től)
- ... és még ki tudja (, hogy mit küldtök) ?

A 19. szám megrendelhető a mellékelt tájékoztató szerint.

Az észrevételeket, közlésre javasolt anyagokat, hirdetéseket a következő címre kérem:

VINDICS ISTVAN, MECSEKNADASD, LISZT F. U. 39. 7695.

A lapzárta időpontja augusztus 25, így az árság megjelenése szeptember első napjaira várható.

Továbbra is írhattok a TVC-Központnak: Szentés, Pf 143. 6601.

Levellezési rovat (helyett)

Az ismert körülmények miatt - lásd előző oldal - nem kaphattam eddig az újság "szerkesztőjeként" levelet, ezért a "magánlevellezésemből" egy-két érdekesebb dolgot szeretnék itt közzétenni.

Elsőként a nyomtatásról. Ezzel elég sok gond akad, mert a Videoton nyomtatók mellett előfordulnak IBM-kompatibilis printerek is TVC-s használatban. Mivel a grafikus nyomtatás nagyon eltér a két esetben, a VT-21200-ra irt rutin nem működik másik nyomtatókon - többek között a VT-21500-on sem. Akiknek ilyen problémájuk volt, azoknak most jó hírrel szolgálhatok: elkészítettem a Varázsecset egy olyan változatát, amely IBM-kompatibilis printerre dolgozik. A használata: a megszokott módon kiválasztható a nyomtatás üzemmód. Figyelj arra, hogy a nyomtató fogadja a jeleket, mert a rajzprogram csak akkor fogadja el a funkcióválasztást, ha ki tud küldeni egy <CR> karaktert! Ezután az ikon villogni kezd, és a 0-3 billentyők lenyomásával indítható a nyomtatás. A kép nagysága (a nagyméretű kép 90 fokkal elforgatva), illetve a színek száma választható. A kétszínű üzemmód azt jelenti, hogy az alapszín fehér, a másik 3 fekete lesz. Háromszínű módban az alap fehér, a 3-as szín fekete, az 1-2 pedig szürke (pontosabban csikos, lásd az 5. oldal WALLTRIS képét). A kódok: 0-kicsi, két 1-nagy, két 2-kicsi, három 3-nagy, három.

Az érdeklődők megkaphatják a programot tőlem, ingyen. Ha más programot nem vásárol, akkor egy válaszborítékot kérnek, és persze az adathordozót is, amire felveszem.

Más téma. A SID kártya tulajdonosok problémája, hogy nem tudnak egyszerűen áttenni zeneadatokat C64-ről. Mostantól tudok rajtuk segíteni. Készítettem egy illesztőáramkört és programot, amellyel a TVC kezelni tudja a Commodore 1541-es floppyt illetve a nyomtatókat. Tehát ha valakinek arra van szüksége, hogy adatokat tegyen át C64 lemezzel TVC lemezre, keressen fel (mondjuk levélben), és az előző bekezdésben lévő feltételekkel szívesen állok rendelkezésére. Persze ez nem azt jelenti, hogy működőképes zeneprogramot kap, csak az adatok bepötyögésétől kímélheti meg magát. Az illesztőegységgel kapcsolatban esetleg érdekes lehet, hogy bár egyelőre csak próbapanel formájában létezik, de szeretném "gyárthatóvá" tenni. Becsléseim szerint az legkedvezőtlenebb esetben is 1500 Ft-on belül tud maradni az előállítás, de előfordulhat, hogy 1000 forintot sem éri el. Ha valakit esetleg érdekel a dolog, kérem jelezze.

Persze nem csak zeneadatokat átvitelére lehet használni. Például a C64-es Newsroom grafikáit - mint pl. az ATOMIX előképén szereplő figura, ami eredetiben fekete-fehér - is át lehet tenni TVC-re. Nekem most kb. 100 darab képem van áttéve, vagyis könnyen meg tudok tölteni egy 60 perces kazettát vagy egy lemezt ilyen képekkel. Ez is megrendelhető nálam, lásd a mellékelt tájékoztatót.

Végül egy felhívás. Sajnos nagyon kevés program és irodalom van a TVC-hez (persze mondhatnánk: mihez képest?), ezért jó lenne, ha ehhez a néhány dologhoz hozzá lehetne jutni. Például hirdették a TVC ZENE, TVC GRAFIKA című könyveket, de nem találkoztam velük a könyvesboltokban. Ha tudtok róluk valami biztosat (egyáltalán megjelent-e?), kérem írjátok meg, szeretnék a bennük lévő érdeklődő információkat közkinccsá tenni. Hasonló lehet a helyzet több felhasználói programmal, érdekesebb leírásokat is szívesen vennék. Kaptam pl. egy leírást a GRIFF programról, a következő számba szeretnék róla egy ismertető cikket megjelentetni.

Gépi kód - abszolút kezdőknek

A most következő kis cikkben a gépi kód programozásról lesz szó, mégpedig a lehető "legprimitívebb" módon, assembler használata nélkül. Sokan ugyanis nem mernek elindulni a gépi kód felé, mert az assembler első látásra túl nagy falatnak tűnik. A cikk megértéséhez csak némi basic ismeret kell.

Először át kell ismételni a számítógép felépítését, persze csak nagy vonalakban. A gép lelke a mikroprocesszor, ami a gép logikai működését teljesen felügyeli. Ez végzi az adatok bevitelét, feldolgozását, tárolását, vagyis mindent, amit általában a számítógépnek tulajdonítunk. Hogyan működik? Egyfelől a szekezetéből kifolyólag alkalmas erre a feladatra, de ami bennünket érint az az, hogy a memóriában található adatok (gépi kód program) alapján vezérelhetjük a működését. A memória technikai szempontból egy elektromos alapokon működő szerkezet, logikailag viszont úgy tekinthetjük, hogy számokat tárolunk benne. (Pl. Az "öt" számot jelölheti ez a két betű, de ugyanezt jelenti az arab (5) vagy római (V) számjegy, a dobókocka pötytyözése, itt pedig elektronkupacok segítségével jelöljük.). A mikroprocesszornak vannak ún. regiszterei, amelyek működésükben kicsit hasonlítanak a basic változóihoz. Ezek közül egy (a PC nevezetű) mindig azt a helyet mutatja, ahol a következő utasítást jelentő adat van a memóriában. A regiszterek részletesebb leírását a következő cikkekre hagyjuk, most - a példaprogramok miatt - a HL regisztert említem csak meg. (Aki ért már kicsit is a dologhoz, az tudja, hogy "csalok": a H és az L két különálló regiszter, de sok esetben együtt is kezelhető.). Mielőtt erre a példára rátérnék, még egy mondat a gép szerkezetéről. A számítógéphez sok "külső egység" csatlakozik, ilyen a képelőállító áramkör, a hang, a billentyűzet, a magnó, floppy stb. illesztő áramkörei. Ezeket a számítógép a memória tartalmának megváltoztatásával (kép), illetve OUT-IN utasításokkal tudja vezérelni.

Tehát ha gépi kód programot szeretnénk módosítani, először el kell helyezni a memóriában POKE utasításokkal, majd elindítani az USR függvény alkalmazásával. A biztonság kedvéért a paraméterezések: POKE cím, adat ill. A=USR(cím, szám). A cím mindig a memóriára utal, a memóriában levő adatok helyét mindig egy 0-65535 közötti szám jelzi. Ebből nekünk (basicben) nem mindet szabad használni, általában 10000-30000 közötti értékeket érdemes választani. Az USR függvény alkalmazása esetén a PC a megadott címre áll, vagyis a processzor a mi utasításainkat hajtja végre, és indulás előtt a HL értéke a megadott szám lesz.

A rutinokat mindig a következő basic programba helyezzük:

```
10 CIM=25600:INPUT HL
20 READ ADAT:POKE CIM+I,ADAT:I=I+1:IF ADAT<>201 THEN 20
30 A=USR(CIM,HL):PRINT A:END
40 DATA !: ide kerülnek az adatok
Írjuk be most a következő adatsort, és futtassuk le!
40 DATA 35,201
```

A 35-ös szám a processzornak azt jelenti, hogy növelje meg 1-el a HL tartalmát, a 201 pedig azt, hogy térjen vissza, jelen esetben a basicbe. Az USR függvényt úgy építették fel, hogy eredményként HL tartalmát adja, tehát a kapott szám a beírtnál eggyel nagyobb lesz. Annak az utasításnak a kódja, amely HL-hez saját magát adja (basicben: HL=HL+HL), 41. Tehát a 40 DATA 41,41,41,201 adatsor a beírt szám nyolcszorosát adja, bizonyos megszorításokkal.

Ezekről a következő számban lesz szó, addig is lehet próbálkozni ezen kódok variálásával (persze körültekintően!)

UPM lemez meghajtó (2)

A funkcióhívások egyszerűsége mellett időnként kényelmetlenségeket is tapasztalunk. Ilyen például az, hogy a játék előképét nem lehet azonnal a képernyőre tölteni, mivel a memórialapozás a funkcióhívás során kötött: az első három lapon RAM van, legfelül pedig ROM (U0-U1-U2-SYS vagy U0-U1-U2-EXT).

Ez azonban csak akkor igaz, ha az RST 30 hívással akarjuk elérni a kívánt funkciót. A csatoló ROM-jának ismeretében azonban a probléma megoldhatóvá válik. Nézzük meg mit kell tudnia egy kazettakompatibilitást biztosító bővítőegységnek!

A csatoló ROM az EXT-el együtt lapozható a 3. lapra, és \$C000-tól érhető el. Az elején azonosítás céljából néhány karakter áll. (Jelen esetben a MOPS után a DISK szót találjuk). \$C00B-n találjuk az inicializáló rutin címét, amelyet reset esetén hajt végre a számítógép. A \$C00D cím tartalmazza a funkcióhívások számát (itt 9), majd a funkcióhívások ugrótábláját, amit azt hiszem érdemes részletezni:

(\$C00E) : 0. funkció = megszakításkezelés
 (\$C010) : 1. funkció = egy bájt mozgatása
 (\$C012) : 2. funkció = blokkmozgatás
 (\$C014) : 3. funkció = fájl nyitás
 (\$C016) : 4. funkció = fájl zárás
 (\$C018) : 5. funkció = fájl ellenőrzés
 (\$C01A) : 6. funkció = UPM funkciók
 (\$C01C) : 7. funkció = fizikai szektorkezelés
 (\$C01E) : 8. funkció = paramétertábla beállítás

Ez utóbbiról még nem esett szó. A paramétertábla átírásával lehetővé válik, hogy ne csak 720K-s lemezeket kezelhessünk, hanem pl. 360K-s 1 oldalas lemezt is. Így lehet olyan programot készíteni, amely egy UPM duálfloppyn lehetővé teszi fájlok másolását, törlését, akár UPM, akár VT-DOS, akár IBM 360K lemezen (A GATO BT "UPMXTVT" programja).

Most pedig a példaprogram következik - TVC-ASSEMBLER forrásszövegben -, amely a képernyőre töltést valósítja meg:

```
10 ORG $2000
12 LOAD $2000
14 LD A,$D0 ; memórialapozás:
16 LD (3),A ; : U0-U1-VID-EXT
18 OUT (2),A
20 LD DE,NEV
```

```
22 LD A,$D3 ; fájl nyitás kódja
24 LD HL,($C014) ; ugrótábla
26 CALL UGRAS ; végrehajtja a funkciót
28 LD DE,$7FF0 ; fejléc miatt
30 LD BC,$3C15 ; Varázsecset képhossz
32 LD A,$D2 ; blokk betöltése
34 LD HL,($C012)
36 CALL UGRAS
38 LD A,$D4 ; fájl lezárás
40 LD HL,($C016)
42 CALL UGRAS
44 LD A,$B0 ; visszatérés TVC-ASSEMBLER-be
46 LD (3),A ; ( a szükséges memórialap.)
48 OUT (2),A
50 RET
52 UGRAS: OR A ; jelzőbitek beállítása
54 JP (HL) ; irány a megfelelő rutin!
56 NEV: DB 3,"KEP" ; név hossza és a karakterek
58 END
```

Ha már ROM bővítésekről beszélünk, érdemes megemlíteni a programmodul (oldalt!) kezelését. Az inicializálás során a beépített ROM a SYS-U1-U2-CART lapozást állítja be, tehát a modul a \$C000 címre kerül. Ezután megvizsgálja az első négy bájtot. Ha itt a MOPS szó található, a \$C004 címre ugrik, tehát a modulnak kell átvenni az irányítást. Ha nincs itt a MOPS, akkor zavartalanul folyik tovább a számítógép inicializálása.

Az UPM kompatibilis funkcióhívások felsorolása a következő cikkekre marad. Ezek használatára általában ritkán kerül sor, pedig így lényegesen gyorsabb a lemezkezelés, mint a kazettakompatibilis hívások esetében. (A VT-DOS-tól azonban alaposan elmarad.) Fejlesztés alatt áll egy TVC-ASSEMBLER-re épülő, de lényegesen többet tudó program, amelyben a funkcióhívásokra épül a lemezkezelés. Ennek utasításkészletét találod a 4. oldalon. A program UPM verziója gyakorlatilag készen van, részletes leírással megrendelhető (100 Ft), a "szerkesztőség" címén (1. oldal). Hamarosan elkészül a VT-DOS verzió, és tervezzük a tetszőleges memóriakezelés beépítését (vagyis pl. diszassemlálhatjuk a SYS ROM-ot úgy, mintha a 0. lapon lenne.)

HARDVER: Eprom-törölő

Elsőként le kell szögeznem: míg programozásban elég jó szinten érzem magam, hardver tekintetében leginkább a "lelkes amatőr" minősítést érdemelném. Ennek ellenére - hogy ez a szám se maradjon hardver cikk nélkül -, elmondanám az epromtörölő berendezésem elkészültének történetét.

Ha komolyabb dolgokat akarunk csinálni, kell egy epromégető. Ezzel az ember elkezd dolgozni, és az "égetett", de már nem használatos memóriák lassan elszaporodnak. Hogy újra felhasználhassuk, ki kell törölni a tartalmát. Ez elméletben nagyon egyszerű: UV fényrel kell megvilágítani. De honnan vegyünk UV fényt? Legkézenfekvőbb a Nap, de nem nagyon hatékony. Előbb megsül az eprom, mint hogy egy bitet is felejtene. Allitóltag a kvarclámpa is jó, de szintén hőgútát okozhat szegény memóriáknak. Szóval legjobb egy kifejezett UV lámpát szerezni, de hogyan?

Felkerestem a legközelebbi lámpaszaküzletet (Pécs), de

itt csak olyan égőt tudtak ajánlani, amelyet a "disco"-kban használnak. Végül elkieseredésembe - mivel már sokat kinlődtam ezzel a témával -, kiadtam érte az 1200 Ft-ot. Persze ez sem vált be elsőre. A búróján lévő anyag erősen megszőrte a fény UV tartalmát, és az eprom megint csak melegedett, de nem felejtett. Ezután következett a drasztikus átalakítás: eltávolítottam róla az üveget (óvatosan!) és az izzószálat, hiszen ez azonnal leégett volna. Az izzószál helyett egy 7.5 kilohomos 2 W-os ellenállást tettem. (Ez lehet, hogy nem szakszerű, de így is működik.) Az egészet rögzítettem egy műanyag dobozban, hogy a szigetelés rendben legyen. Esztétikusan meg kellett oldani a 220 V bejutását a dobozba, és "máris" készen volt a "nagy mű".

Most egész közel tehetem az epromot a lámpához, a hó nagyon kicsi. Általában 5 perc elég a törléshez, de előfordul, hogy néhány "megátalkodott" bit miatt még egy picit kezelni kell.

WALLTRIS

Gondolom sokak kedvence a TETRIS játék, ezt mutatja az is, hogy minden géptípuson megtalálható és több változat elkészült TVC-re is. Az alapötletet többirányba fejlesztették: a Coloris-ban például a színek vették át a formák szerepét, létezik a térbeli idomokkal játszható játék is - egyelőre még csak más gépeken.

A Walltris egy kicsit átmenet a térbeli változat felé, de még nem egészen az. Sikidomok csúsznak le egy térbeli gödör falán, ezekből a gödör alján kell teljes sorokat vagy oszlopokat megtölteni. A játékos jobbra-balra tudja mozgatni a falakon az aktuális idomot, a harmadik lehetőség, hogy leejti; forgatni nem lehet! Fontos tudni az irányításról, hogy a gödör alját elérve automatikusan bekapcsolódik az ejtés üzemmód, ezután nem tudunk beleszólni a mozgás menetébe.

Az indulás után lassan elkezd megtelni a gödör alja, és előállhat olyan helyzet, hogy a sikidomunk egy része a gödör falán marad. Ekkor a kérdéses fal pirosra színeződik. Ez annyiban kellemetlen, hogy nem tudunk rajta mozogni a következő 3 sikidommal. Ha ez a büntetésünk letelt, a piros szín eltűnik. A játéknak akkor van vége, ha már nem tudunk mozdulni, vagyis ha mind a négy fal "elvörösödött". Ekkor a pontszámunk felkerül a listára - persze csak akkor, ha elég nagy. A csúcslistán kívül láthatjuk még a képen az aktuális pontszámot, a kirakott sorok számát, a mozgás sebességét és a nehézségi szintet.

A játék négy nehézségi fokozaton játszható (0-3), ezek a potyogó tárgyak alakjában térnek el. A legkönnyebb fokozaton leginkább 2-3 négyzetből állnak az idomok, a 3-as szinten csupa 4 négyzetből álló alakzat váltakozik véletlenszerűen. A sebesség 0 és 9 között állítható. Hogy mivel? Lehet például botkormányal (belső vagy 1-es), de használható a pont, vessző, kivonás, ALT, szóköz billentyűket is.

Két dolgot érdemes még tudni. Az oldalfalak sorrendje: észak, kelet, dél, nyugat. Ha a felső fal szabad, ott fog elindulni a következő idom, és a sorok vizsgálatánál is a gödör aljának északi részén levő vízszintes sorokat keresi. Ha ilyet talál, a tőle északabbra levő darabokat eggyel délre húzza. Ezután hasonlóan folytatódik az eljárás a keleti falnál. Potyoghat egy idom úgyis, hogy egy része az egyik, más része a másik falon van. Ilyenkor a gödör aljára érve meglepő dolgokat tapasztalunk. Ennek magyarázata, hogy az idom darabjai egyszerre, de egymástól függetlenül, sőt egymásról tudomást sem véve mozognak. Így fordulhat elő, hogy a gödör alján az alakzat szétszakad, vagy éppen fedésbe kerülnek a darabjai. Ez először valószínűleg kellemetlenül fog érinteni, de később ügyesen fel lehet használni a játék e sajátosságát.

Ennyit elegendő tudni a program használatához, de a programírók számára akad néhány érdekesség e játék kapcsán. Ezeket a "Egy színezési algoritmus" című cikkben találhatod meg. Ha pedig csak játszani akarsz, jó szórakozást!

ENG-ASSEMBLER LEÍRAS

6'N'6 SOFTWARE, 1992.07.02

A PARANCSONK ÖSSZEFOGLALASA

A paraméterek meghatározása:

cim,h, : legfeljebb 4 jegyű hexadecimális számok
 sor : decimális sorszámok
 szám helyett használható : címke is
 int : intervallum
 megadása: cím1 cím2 cím Lh
 forrásszövegnél: sor1-sor2
 bsor : bájt sorozat
 benn: bájt, két bájt, karaktersorozat, ?
 fnév : fájl név
 megadása: MU:név.kiterjesztés

A PARANCSONK ISMERTETÉSE (ABC SORRENDEN)

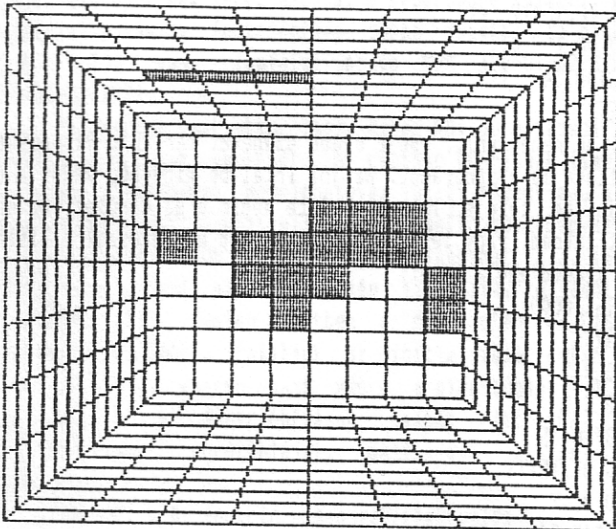
A : fordítás (AD, AP).
 B n1 m n2 : átváltás más alapú számrendszerbe.
 C int cím : memória másolása.
 D (cím) : disassemblálás (DP).
 E int bsor : bájt sorozat keresése.
 lehet még: <bájt >bájt [bájt1 bájt2]
 F karaktersorozat : keresés forrásszövegben.
 G (cím) : gépi kódú program indítása.
 H : a forrásprogram határait írja ki.
 I int bsor : memória feltöltése.
 J : kilépés BASIC-be.
 K (int) : forrásszöveg törlése.
 L (int) : forrásszöveg listázása.

M (cím) : memóriatartalom aktuális formában (MP).
 N (sor){,d}{,int} : forrásszöveg átsorszámozása.
 O (sor){,d} : automata sorszámozást kapcsol be (OK).
 P (int) : forrásprogram nyomtatóra listázza.
 Q (cím) : töréspontok beírás ill. lista.
 QT (cím) : törli a töréspontot.
 R"fnév" cím : fájl olvasása. (alap: .CAS)
 RT"fnév" : forrásszöveg beolvasása. (alap: .ASM)
 S (maszk) : szimbólumtábla elemei és értékei (SP).
 T (cím) : nyomkövetés (alap: PC).
 U fnév : batch fájl végrehajtása.
 V int cím : memória összehasonlítás.
 W"fnév" int : memóriaterület mentése. (alap: .CAS)
 WT"fnév" : forrásszöveg mentése. (alap: .ASM)
 X (reg=érték) : regiszterértékek kiírása, értékadás.
 Y p : p alapján beállítja az M formátumát.
 (O: alap 1: csak szám 2: csak szöveg)
 YD, YE, YT, Y:, Y\ : formátumkapcsolók.
 Z bsor : a bájt sorozatot nyomtatóra küldi.
 . cím bsor : a c címre elhelyezi a bájt sorozatot.
 \ (maszk) : directory.
 @ (maszk) : a megadott fájlok törlése.
 >"fnév1"="fnév2" : fájl átnevezése. (USER-t nem változtat!)
 [{\$}port]{,bájt} : IO port kezelés.
 ? szl mj szZ : számolási művelet végrehajtása.
 A műveleti jelek (mj): + - * / & v x
 * cím (/o/c) (/s) : forrásszöveg áthelyezés.
 : cím utasítás : utasítást memóriába fordítja.

Használható direktívák: ORG, LOAD, EQU, DB, DW, DS, INCLUDE, SYS, END.
 (INCLUDE"fnév": lemezről befordítja a megadott fájlt)

WALLTRIS

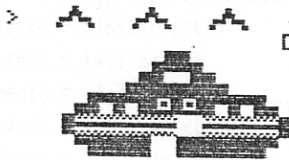
STV 92/2



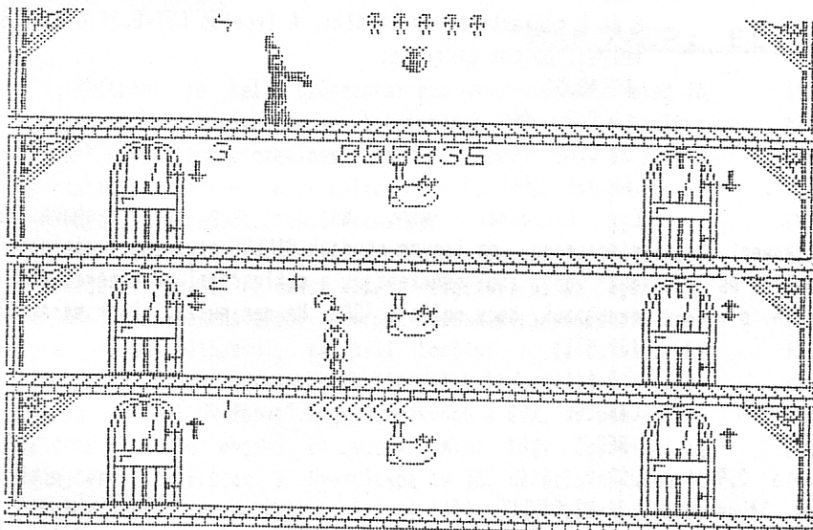
SZINT
 0
 SOROK
 0
 SOR
 00004
 PONT
 000000

LISTA
 000000
 000000
 000000
 000000
 000000

IDŐ
 07110



CSÚCS : 05000
 PONT : 00400



Sokan biztosan az első programcsomagot sem láthatták még, de már itt a második. Most négy gépi kódú programot ajánlok fel a játszani kedvelő géptársaknak. A négy játék közül a WALLTRIS-ről külön leírást olvashattok, a három Primo Átiratot itt szeretném néhány sorban ismertetni.

UFO: a feladatunk egy "repülő csészéalj" megsemmisítése. Ebben akadályoz a mozgó védőpajzs, melyen lyukakat lehetünk. Mivel egy ilyen találatnál potyognak a szétlőtt pajzs darabjai, igyekeznünk kell időben félreállni. Ezután a hasonló veszélyeket elkerülendő, érdemes a lyukon át tüzelni, bár ez nem mindig sikerül. Sőt, ha balszerencsénk a pajzs egy vastagabb részét találjuk el, a sérülései begyógyulnak. A következő probléma, hogy az idegenek idomított madarai felfogják a lövedékeinket, ha rossz ütemben lövünk. Ha itt is szerencsénk volt, a golyó az ufo testébe hatolhat. Addig kell roncsolnunk, amíg el nem találjuk a vezérlőtermet, ekkor átjutunk a következő "járműhöz". A játék erénye, hogy részletes ismertetést, és sok animációt tartalmaz.

LINA: Egyszerű, de nagyszerű (bocs!) kis játék két játékos számára. Az alapötlet például a Tron című filmből ismert, úgy kell irányítanunk a vonalunkat, hogy se a falnak, se az ellenfél vonalának, se a sajátunknak ne menjünk neki. Aki tovább bírja, az kapja a pontot.

OSZKAR: "hivatalosan" terjesztett program volt annak idején a Primo mellett Spectrumon is, ezért meglehetősen finom grafikával, részletes animációval készült. Egy várkastély termeiben járkalhatunk, amelyek inkább labirintus-, mint lakóhelyszerűen épülnek fel. Feladatunk megkeresni a gyönyörű Mirandolinát, akit a várőr fogva tart. Az utunkat az ajtók kellemetlen elhelyezkedésén kívül a kőbor szellemek is akadályozzák. Figyelnünk kell az élelem és a pénz gyűjtésére, és sorolhatnám. A továbbiakat olvasd el a játék elején lévő leírásban. Egyetlen dologra hívnám fel még a figyelmet: az ajtón akkor tudsz átmenni, ha a lovag orra az ajtó közepénél van; és persze figyeld a nyilakat!

Ha szeretnéd megkapni a játékokat, a júniusi számban leirtak szerint járj el. Vagyis egy-egy programcsomag ára 200 forint, ehhez jön az adathordozó ára, 50 Ft (5 peces kazetta programcsomagonként, illetve lemez, az általad kért formátumban). Lehetőség van a régi programjaim megvásárlására is, ez a programcsomag mellett plusz 50 Ft-ba kerül. Külön kategóriát csak az ATOMIX képvisel, amelyért szintén 50 Ft-ot kérek. Felhívom a figyelmet, hogy az árak nem azért ilyen alacsonyak, mert rosszak a programok. Annyit szeretnék elérni, hogy nálam vegyétek meg a saját készítésű programjaimat, és ne illetéktelen másolóktól (akik ráadásul lehet, hogy többet kérnek!). Jó lenne tudnom, hogy csak egy-két embernek írom a programokat, vagy többen is érdeklődnek a TVC iránt.

Szóval várom az érdeklődő leveleket, illetve a megrendeléseket a következő címen:

Vindics István, Mecseknádasd, Liszt F u. 39. 7695

Gépi kódú programozás [TVCM]

Az alábbi sorokat azoknak szánom, akik tisztában vannak az INC, DEC, POP, PUSH, RST, CP, JR, LD, RET, DJNZ utasításokkal. Akiknek valami nem világos, azoknak ajánlom az újság régebbi számait, illetve a gépkönyveket.

Feladat: Egy 7*8-as "doboz" kirakása a 2,2-es pozícióba.

Első lépés minden programnál a kezdőcíme (ORG), a vége (END) és egy RET letétele, na meg a START:

```
ORG 10000
```

```
START:
```

```
RET
```

```
END
```

Ha már ennyivel készen is vagyunk, akkor el is lehetne kezdeni. Első az, hogy a kérdéses helyre pozicionálni a későbbi kiírásokhoz, 35-ös funkció:

```
LD B,2
```

```
LD C,2
```

```
RST48,35
```

Es ha már ott vagyunk, akkor ki kell írni a "doboztetőt", amit előbb le kell tenni pl. TEXT1 néven a RET és az END közé, hogy ne zavarják a programot:

```
RET
```

```
TEXT1: DB "r-----," <<<
```

```
END
```

Most pedig a pozicionálás után tenni a kiíratórutint, a 34-es funkcióhívást:

```
RST48,35
```

```
LD DE,TEXT1 <<<
```

```
LD BC,7 <<<
```

```
RST48,34 <<<
```

```
RET
```

Most egy kicsit kifújhatod magad, és ki is lehet próbálni az eddigi munkát (mert működik!). Ha kirajzolja a doboztetőt a megfelelő helyre, akkor minden rendben. Ha nem, akkor hasonlítsd össze ezzel a listával, hátha elgépeztél valamit:

```
ORG 10000
```

```
START:
```

```
LD B,2
```

```
LD C,2
```

```
RST48,35
```

```
LD DE,TEXT1
```

```
LD BC,7
```

```
RST48,34
```

```
RET
```

```
TEXT1: DB "r-----,"
```

```
END
```

Ha eddig rendben, akkor most jön a vége (hogy cifrább legyen), mivel az is csak ennyiből áll, vagyis egy pozicionálás és egy kiírás. Első a doboz "aljának" lerakása a TEXT1 után, pl. TEXT2 néven:

```
TEXT1: DB "r-----,"
```

```
TEXT2: DB "l-----" <<<
```

```
END
```

Most pedig jön a pozicionálás 35-ös funkcióval, a 2,9 pozícióba (itt lesz a vége), majd a blokkírás 34-es funkcióval, mint az előbb. Az előző RST48,34 és a RET közé kell tenni:

```
RST48,34
```

```
LD B,2 <<<
```

```
LD C,9 <<<
```

```
RST48,35 <<<
```

```
LD DE,TEXT2 <<<
```

```
LD BC,7 <<<
```

```
RST48,34 <<<
```

```
RET
```

Hű, de nehéz volt, ideje újabb szünetet tartani. Pihenésképpen futtasd le a programot. Ha jól irtál be mindent, akkor a doboz alja és teteje látszik, közte pedig üres terület. Ha mégsem ezt látnád, vagy lefagyott, akkor vedd össze ezzel a listával, és keresd meg a hibát:

```
ORG 10000
```

```
START:
```

```
LD B,2
```

```
LD C,2
```

```
RST48,35
```

```
LD DE,TEXT1
```

```
LD BC,7
```

```
RST48,34
```

```
LD B,2
```

```
LD C,9
```

```
RST48,35
```

```
LD DE,TEXT2
```

```
LD BC,7
```

```
RST48,34
```

```
RET
```

```
TEXT1: DB "r-----,"
```

```
TEXT2: DB "l-----" <<<
```

```
END
```

Most pedig jön a munka legjava! Ehhez először le kell tenni a doboz "törzsét" alkotó karaktereket, pl. TEXT3 néven, az END elé:

```
TEXT2: DB "l-----" <<<
```

```
TEXT3: DB "l-----" <<<
```

```
END
```

Ezt a bizonyos karaktersort nem csak egyszer kell kirajzolni, hanem 6-szor, ráadásul egymás alá! Egy ciklust kell szervezni, amihez mondjuk az A regisztert használjuk. A koordináták is változnak, ezeknek is van egy kezdeti értéke, amit a ciklus elején meg kell határozni, a 2,3 pozíció, amit a legkönnyebb a B és C regiszterekben tárolni. A legelső RST48,34 után csinálj helyet, és oda szórd be:

```
RST48,34
```

```
LD B,2 <<<
```

```
LD C,3 <<<
```

```
LD A,6 <<<
```

Egy ciklusnak -mint tudjuk- van eleje, ahová mindig visszamegy, ez legyen mondjuk CIKLUS nevű, és a ciklusnak van egy vége, ahol csökkentjük a ciklusszámláló értékét (DEC), és megnézzük, hogy nulla-e (CP). Ha nem nulla, akkor ugrás vissza (JR NZ)!

```
LD A,6
```

```
CIKLUS: <<<
```

```
DEC A <<<
```

```
CP 0 <<<
```

```
JR NZ,CIKLUS <<<
```

A regiszterekre szükségünk lesz! Ezért még a ciklus elején MINDEN lényeges regisztert el kell menteni, mint pl. a ciklusszámláló és a pozíció. A ciklus végén (a DEC előtt) vissza kell hozni őket, hogy tudjunk velük dolgozni.

Elmentendő (PUSH) és visszahozandó (POP) a BC és az AF regiszter:

```

CIKLUS:
PUSH BC <<<
PUSH AF <<<
POP AF <<<
POP BC <<<
DEC A
CP 0

```

JR NZ,CIKLUS

A PUSH és a POP között nyugodtan dolgozhatunk a regiszterekkel, nem okoz felfordulást. Mivel a BC regiszterpárban az aktuális koordináta van, célszerű egy pozicionálással kezdeni (35), majd a szokott módon a doboz "hasát", a TEXT3-at kell kiírni (34):

```

PUSH AF
RST48,35 <<<
LD DE,TEXT3 <<<
LD BC,7 <<<
RST48,34 <<<
POP AF

```

Ezzel tulajdonképpen kész is. Várjunk csak! Kimaradt az, hogy a ciklus minden egyes lefutásakor növelje a sorszámolót! A C regiszter értékét kell növelni akkor, amikor a BC regiszter tényleg a pozíciót tartalmazza. Célszerű a ciklus végén, a POP-ok után:

```

POP BC
INC C <<<
DEC A

```

Es ezennel kész is program! A babérokon csúcsulva el lehet indítani. Elméletileg most jó, de ha mégsem azt csinálja, ami kellene, akkor hasonlísd össze ezzel:

```

ORG 10000
START:
LD B,2
LD C,2
RST48,35
LD DE,TEXT1
LD BC,7
RST48,34
LD B,2
LD C,3
LD A,6
CIKLUS:
PUSH BC
PUSH AF
RST48,35
LD DE,TEXT3
LD BC,7
RST48,34
POP AF
POP BC
INC C
DEC A
CP 0
JR NZ,CIKLUS
LD B,2
LD C,9
RST48,35
LD DE,TEXT2
LD BC,7
RST48,34

```

```

RET
TEXT1: DB "-----"
TEXT2: DB "-----"
TEXT3: DB "|   |"
END

```

Ákinek ez is sor volt, az most tegye félre az órságot, és gondolja át még egyszer, hogy is működik a program, mit csinál, hogyan csinál, és futtasd le többször is.

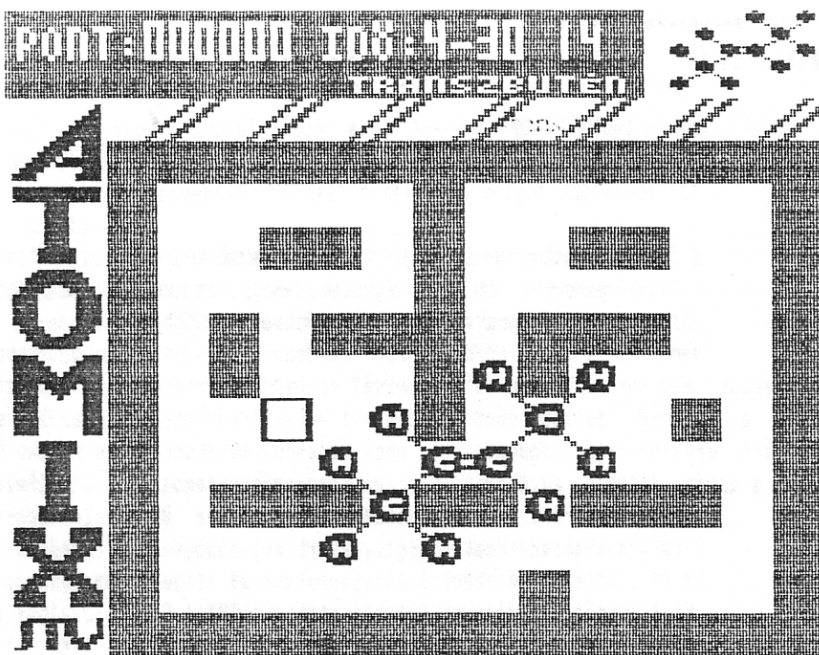
Ha úgy érzed, hogy minden világos és érthető, akkor néhány trükköt mutatok, amivel le lehet egyszerűsíteni a programot! A jelzőbiteket nemcsak a CP állítja be, hanem az INC és a DEC is, mégpedig úgy, hogy ha a növelés/csökkentés értéke 0, akkor Z, egyébként NZ. Vagyis a programban a CP 0 felesleges. Másrészt a ciklusból kilépéskor a BC a 2,9 poz-ra mutat (kövesd csak végig!), ezért felesleges az értékadás. A ciklus elején a koordináta egy sorral lejjebb van, mint a legelső pozicionáláskor. Tehát ha a legelső RST48,35 előtt elmented PUSH-al a pozíciót, a ciklus előtt POP-pal visszaolvasod és INC C-vel beállítod, akkor rövidebb és egyszerűbb a program. A ciklus is egyszerűsíthető! Igaz, hogy a B regiszter a pozíciót jelöli, de értéke mindig 2! Ezért használható ciklusképzésre, és mivel a C-vel együtt menthető, megtakarítjuk a PUSH/POP AF-t. Tehát: LD A,6 helyett LD B,6; a fentebb említett PUSH/POP törölendő, a DEC A is, a JR NZ helyett DJNZ, és a ciklusban lévő RST48,35 elé kell egy LD B,2. Ebben az esetben a B,C ciklusból kilépéskor nem a 2,9 pozíciót adja, ezért utána kell írni egy LD B,2-t!

Íme egyszerűsítve a program:

```

ORG 10000
START:
LD B,2
LD C,2
PUSH BC
RST48,35
LD DE,TEXT1
LD BC,7
RST48,34
POP BC
INC C
LD B,6
CIKLUS:
PUSH BC
LD B,2
RST48,35
LD DE,TEXT3
LD BC,7
RST48,34
POP BC
INC C
DJNZ CIKLUS
LD B,2
RST48,35
LD DE,TEXT2
LD BC,7
RST48,34
RET
TEXT1: DB "-----"
TEXT2: DB "-----"
TEXT3: DB "|   |"
END

```



A nyomtatásról

Többször tapasztalhattatok már, hogy némely kép kinyomtatás közben meglehetősen átalakul, tekintve, hogy a színes képernyő helyett a fekete-fehér printerre kerül. Létezik ugyan színes nyomtató - és nem is elérhetetlen -, de a legtöbb esetben mégiscsak fekete festékszalaggal kell számolnunk.

Az egyik módszer a nyomtatott kép szebbé tételére, ha megpróbáljuk a szürke különböző árnyalatait kihozni belőle. Kisebb felbontású képeknél erre jobb lehetőségünk van, mert a nyomtató felbontása általában viszonylag nagy. Az a STAR printer, amivel ez a pár oldal készült, az A/4-es papírszélességen 960 pontot tud ábrázolni. Ha tehát ezzel egy C64 tulajdonos multicolor képet szeretne nyomtatni, minden képpontra 6 jutna a printeren, amivel már lehet alakítani.

A TVC esetében általában négyszíndű képeket használunk - ehhez igazodott a nyomtatórutinok többsége is -, így ideális programmal 3 pontot feleltethetnénk meg egy képpontnak. Mivel ez a szám "nem szép" a számítógépek világában, ezért gyakorta megelégszünk 2-vel is. A fenti művecske is négyszíndű képről lett nyomtatva, de egy kicsit másképp.

Nem kell meglepődni, a rajz eredetileg tényleg 16 színdű, mint ahogy a 14-es számban olvasható, nyomtatás előtt színeztem át négyes grafikára. Persze ez nem megy simán. Először két színnel - feketével és fehérrel - színeztem át mindent a Pictures-ben, majd kis gépi kódú rutinnal áttettem négyesbe. A nyomtatórutin 3 árnyalatot tud, ezért a "szürke" kialakítása következett. Még egy konvertálás, és kettes grafikába került a kép, ahonnan már egyszerűen kinyomtattam a biteintát.

Mindezt azért meséltem el, mert nem tudom, hogy érdekel-e mást is néhány ilyen konvertáló rutinocská. Ha igen, néhány cikkben alaposan körüljárhatnánk a témakört, és mindenki "nyomtatási szakemberré" válhatna.

Ha pedig nem érdekel a nyomtatás, azért ennek a képnek egyébként is hasznát veheted: így néz ki az Atomix 14. pálya megfejtése. (Ez a molekula már nagyon sok játékos elakasztott.)

Ismét kell néhány szót ejtenünk erről a játékról. Mivel semmi nem tökéletes, ebben is találtam ismét egy apró, de bosszantó hibát. Az örök időt utólag - többek kérésére - tettem bele a programba, ezért nem törődtem sokat a tesztelésével. A júniusi számban megjelent, hogy az örök idő mellett a kezdőpálya számát is be lehet állítani némi ügyeskedéssel. A nyolcasig még csak működik a dolog, de utána működésbe lép a hiba, ami a TVC billentyűzetfelépítésének a következménye.

Nézzük, mit is kell csinálni a kívánt pálya eléréséhez. Az ESC mellett lenyomni egy billentyűkombinációt, meghozza a 0. sorból (bal felső rész). A kód a kettes számrendszer segítségével alakul ki. Az egyes billentyűk értéke: [5]=1 [3]=2 [2]=4 [0]=8 [6]=16. A kód a pálya számánál 1-gyel kisebb érték, tehát ha a képen látható 14-es pályán szeretnénk kezdeni, akkor a 13=8+4+1-et kell létrehoznunk, vagyis lenyomva kell tartani a [0],[2],[5] gombokat. Mivel az elindulás után ezekkel különböző menüpontok végrehajtása kezdődik ezért érdemes az [1] gombot is lenyomva tartani, és őt felengedni utoljára.

Nézzük a hiba kijavítását. A program elején van egy tömörítést kibontó programrészlet, ennek kezdetétől számítva a 65. bajt 254 és utána 247. Ez egy CP #F7 utasítás, ami ezt vizsgálja, hogy az ESC billentyű sorában ő le van nyomva, de más nem. E helyett egy AND #08 utasítást kell betenni (230,8), ami megengedi, hogy más is lenyomva legyen.

De miért van erre szükség, hiszen mást nem akarunk lenyomni? Mivel a TVC billentyű mátrixszerden vannak kötve, ha három alkalmasan választott gombot lenyomunk, a gép egy negyediket is érzékel. Például: ha a [0],[6],[ESC] gombokat lenyomjuk, a gép a [CTRL] lenyomását is érzékeli. Ez megállapítható abból, hogy az alábbi táblázatban a négy billentyű egy téglalap csúcsait alkotja.

Ez a magyarázat arra, hogy miért lép fel olyan billentyű, amit nem is nyomtunk le.

sor	bitek							
	7	6	5	4	3	2	1	0
0	[4]	[1]	[7]	[6]	[0]	[2]	[3]	[5]
1	[7]	[0]	[6]	[*]	[0]	[9]	[8]	[^]
2	[R]	[0]	[@]	[Z]	[;]	[W]	[E]	[T]
3	[U]	[P]	[@]	[I]	[0]	[0]	[I]	[I]
4	[F]	[A]	[<]	[H]	[\]	[S]	[0]	[6]
5	[J]	[e]	[0]	RET	[A]	[L]	[K]	DEL
6	[V]	[Y]	LOC	[N]	SFT	[X]	[C]	[B]
7	[M]	[~]	SPC	CTR	ESC	[.]	[,]	ALT
8		RJL	RJR	RJA	RJF	RJD	RJU	INS

(a 9-es sorban a bal botkormány olvasható)