

BARÁTUNK A SZÁMÍTÓGÉP



BEVEZETŐ

Számítógépkorszakban élünk.

Az újságokban nap mint nap rövid hírekben, hosszabb tudósításokban olvashatunk a számítógépek elterjedéséről, újabbnál újabb alkalmazási lehetőségeikről. A televízió rendszeresen beszámol a számítógépek térhódításáról, a megjelenő új típusokról, gépcsodákról.

A külföldre utazók gyakran felkeresnek egy-egy számítógépüzletet, ahol végre megveszik a régen óhajtott gépet. A merészebbek és szakmailag jobban felkészültek maguk vágnak neki a számítógépépítésnek és több kevesebb kudarc után azért elkészítik saját kis gépüket. Számítógéppel találkozunk a gyárakban, az irodákban, az üzletekben, a kórházakban és természetesen az iskolákban is.

Valóban ennyire fontos a számítógép?

Igen! A számítógép ma még csak fontos, de hamarosan már nélkülözhetetlen lesz. Nehéz lenne felsorolni, hol és mire használjuk már ma is a számítógépeket, a jövő pedig szinte elképzelhetetlen nélkülük.

Nem is olyan régen még a számítógép tiszteletet parancsoló gépcsoda volt. Fehérköpenyes mérnökök állták körül és működése egy átlagember számára érthetetlen, megismerhetetlen volt. Ma a személyi számítógépekkel 5-6 éves gyerekek is „dolgoznak”. Új fogalmakkal ismerkedünk: szoftver, hardver, bázis és természetesen a különböző gyárakból áramló géptípusokkal: ZX, Spectrum, Commodore, Apple, IBM és ami külön öröm: HT és PRIMO.

E két utóbbi hazai gyártmány. A HT 1080 Z a középiskolások iskola-számítógépe. Minden középiskolában ott találjuk, és a sok ötletes játékprogram mellett egyre inkább előtérbe kerülnek a tanítást-tanulást segítő oktatásprogramok is. Ezek jelentős részét az iskolák tanárai és diákjai írják.

És itt van végre a PRIMO. Ez az első olyan hazai számítógép, amit egyszerűen bárki megvehet az üzletben. Az elmúlt egy év alatt több ezer darab került belőle forgalomba és reméljük, hamarosan minden általános iskolába eljut.

Ez a ti gépetek, igyekezzetek mielőbb megszerezni!

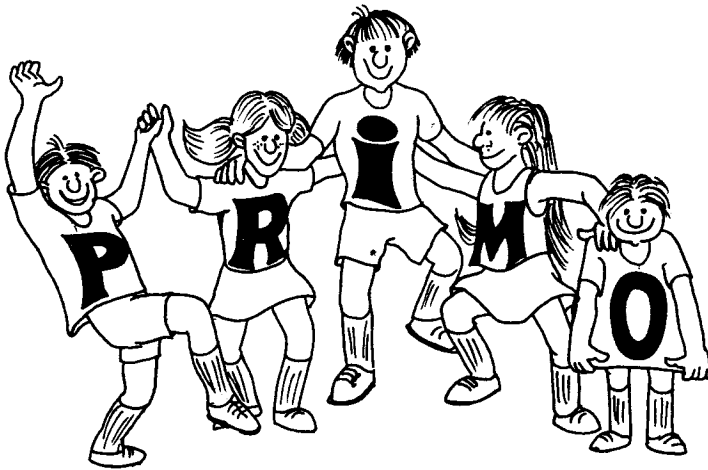
S ha megszereztétek, ismerjétek meg, tanuljátok meg a kezelését. Szeretnénk, ha mielőbb kezdődhetne a barátkozás a számítógéppel. Ehhez a barátkozáshoz kívánunk segítséget nyújtani a Barátunk a számítógép közösségi játékunkkal.

Játékra hívunk benneteket.

Nem számítógép-építő szakkört vagy programozó tanfolyamot indítunk, ezek még koraiak lennének. De segítséget adunk azoknak, akiket érdekel a számítógép, kíváncsiak történetére, a működés alapjaira, az alkalmazás fontosabb területeire és természetesen a programozásra.

A játék segítésére adjuk ezt a játékkönyvet a kezetekbe. Olvassátok figyelmesen, és ha úgy érzitek, egy-egy részhez ti is hozzá tudnátok tenni még valamit, itt a lehetőség. A felsorolt feladatok megoldásait nem kell kötelezően beküldeni apró jutalom reményében. Nagyobb jutalom az a tudás, amit a játék során megszereztek! Ezért nem írunk ki konkrét feladatokat és beküldési határidőket.

Szeretnénk, ha magatok jönnétek rá a számítógép kezelésében rejlő örömökre, és önállóan hátránátok el a felmerülő nehézségeket is. Mert azok is lesznek. Az önállóan megszerzett tudás értékesebb és tartósabb talán azért is, mert nehezebb megszerezni. Segítsetek egymásnak és tapasztalataitokról számoljatok be nekünk is. Mi ezeket a tapasztalatokat folyamatosan közreadjuk a Pajtásban, később a televízióban.



Beküldött anyagaitokat szakemberek nézik át, ha segítséget kértek, megadjuk. A legeredményesebben dolgozókat nyárra szaktáborba hívjuk, ahol értékeljük a játék első évének tapasztalatait és közösen kidolgozzuk a következő évek feladatait. Nem titkoljuk, hogy segítségetekkel szeretnénk

később egy PRIMO Basic programozó tanfolyamot indítani úttörőknek és belevágunk egy „Úttörő–Primo” számítógép-összeépítési akcióba is.

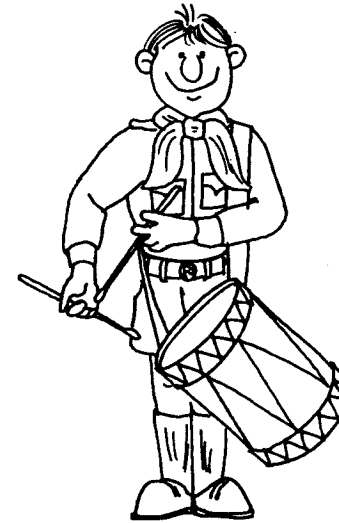
Azért indul tehát most a Barátunk a számítógép játék, hogy nagy ter-
veinkhez segítőtársakat találjunk köztetek.

Címünk:

Barátunk a számítógép
Balatoni Úttörőváros
Zánka
8250

ÚTMUTATÓ

Játékot hirdetünk. Olyan játékot, melyben a feladatokat magatok választjátok. Ki-ki érdeklődésének, tudásának, szabadidejének megfelelően kapcsolódhat be a játékba. Azt is eredménynek tartjuk, ha valaki, akit eddig nem érdekelt a számítógép, egyszerűen csak átlapozza ezt a füzetet, megnézi a rajzokat és a fényképeket, esetleg beleolvass egy-egy fejezetbe. Talán egyszer ő is kedvet kap majd a játékos ismerkedéshez.



Persze ennél többre is számítunk!
Játékkönyvünkben röviden áttekintjük a számítógépek történetét. Az összefoglalás nem teljes, aki újabb adatokat, esetleg rajzokat, képeket talál, küldje el és a következő könyvbe már azt is belevesszük.
Ez is egy feladat.

Áttekintjük a könyvben azt is, milyen részekből áll egy számítógép. Közületek azok, akik már ismerik a számítógépeket, biztosan másként mutatnának be egyes részeket. Várjuk a véleményeket, ti hogyan mutatnátok be a számítógépeket barátaitoknak, akik most kezdik az ismerkedést.

A számítógépek működtetéséhez nem kell nagy matematikai tudás, aki azonban a működést meg is akarja érteni, annak nemcsak a számítógéppel, hanem a matematikával és az elektronikával is jó barátságot kell kötnie.

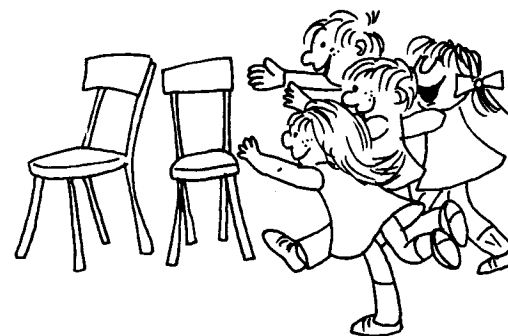
A számrendszerek nagyon érdekes részei a matematikának, de kell-e ezekkel foglalkozni egy játékkönyvben? Erről mi a véleményetek?

De ne vágjunk a dolgok elé, a feladatok majd a játékkönyv második felében teszik próbára felkészültségeteket. Kezdjük most az ismerkedést, a barátkozást — időben jó pár ezer évet visszaugorva!

AMÍG A SZÁMÍTÓGÉP ELKÉSZÜLT

A kezdet

A számolás története évtizedekre nyúlik vissza, kezdetét nem tudjuk meghatározni. Az ősember már számolt, pedig még írni, sőt beszélni sem tudott. Számolási módját azóta is változatlanul használjuk. Hihetetlen? Pedig így van! Természetesen most ne a matematikaórákra gondoljatok, az ősember nem tudott sem szorozni, sem osztani. Tudott viszont összehasonlítani, mert az élet erre kényszerítette.



Emlékezzetek csak vissza, hogyan szoktuk az iskolában óra előtt gyorsan megállapítani, van-e hiányzó? Mindenki leül a helyére, és ha marad üres szék, van hiányzó.

Mit csinálunk ilyenkor? A székek számát és a tanulók számát összehasonlítjuk. „Matematikai nyelven” azt mondjuk: megállapítjuk, hogy egy adott halmaznak (a tanulók halmazának) ugyanannyi, több vagy kevesebb eleme van, mint egy másik halmaznak (a székek halmazának). Ez az összehasonlítás pedig egyszerű párképzéssel történik: egy szék és egy tanuló alkot egy párt. Ha a párképzés végén minden székre jut egy tanuló, azt mondjuk, a két halmaz elemei egyenlő számúak. Ha viszont a párképzés végén még marad üres szék, az egyik halmaz kevesebb elemet tartalmazott, mint a másik.

Ez az „egy szék, egy tanuló” eljárás a számolás alapja. Az ősember tehát így számolt, pontosan el tudta dönteni, ki talált több tojást vagy ki szedett több gyümölcsöt.

De miért kell az ősember számolásáról beszélni, mikor bennünket a modern számítógép érdekel inkább?



Azért, mert a számítógép működése nagyon hasonló az ősember gondolkodásához. A kisebb–nagyobb–egyenlő kérdésének eldöntésével kezdődött az ember számolása és érdekes módon a számítógép is erre alapozza működését.

Véletlen volna?

Azt hiszem, érzitek, hogy ez a hasonlóság nem véletlen. A számítógépet az ember hozta létre, működésében azokat a törvényszerűségeket alkalmazza, amelyeket a fejlődés évezredes útján megismert és amelyekre a jelenlegi technikai fejlettség mellett lehetőség van. Ezért a nagy hasonlóság.

A folytatás, ami újabb kezdet is . . .

Lépjünk gondolatban néhány ezer évet és máris a kőkorszak végén járunk. Az ember életében döntő változást hozott a termelésre való áttérés. Már nemcsak saját szükségletei kielégítésére gyűjtögetett, vadászott, halászott, hanem termelt is, termékeit pedig cserekereskedelemben értékesítette. A cseréhez viszont elengedhetetlen volt a termelt vagy cserére kerülő termék mennyiségének meghatározása, megszámlálása. A kettőnél nagyobb számokat eleinte az egy és a kettő többszöri ismétlésével fejezték ki. Valahogy így: öt helyett kettő meg kettő, meg egyet mondtak. S hogy a számolást el ne tévesszék, segédeszközöket használtak.

A legrégebb számolóeszközt maga a természet bocsátotta az ember rendelkezésére: ezek a kéz ujjai. Ahogy a kisgyermek az ujjain igyekszik kiszámolni, mennyi kettő meg egy, úgy számoltak elődeink 7–10 ezer évvel ezelőtt. Ha kevés volt a saját tíz ujjuk, a gazdagabb kereskedők rabszolgáik ujjával számoltak tovább. (Latinul az ujjat digitusznak mondják, digitális eljárásnak pedig a részenkénti számolási módot, amely a számítógépek egyik típusának elvi alapja lett.)



Később az ujjak helyett kavicsokat használtak. Az ujjak és a kavicsok az első számolási segédeszközök. Az egyiptomiak majd a görögök a tíz ujjat táblába vésték, bemélyedéssel helyettesítették, melyre merőlegesen is húztak néhány vajat. Az így elkészített táblát abakusznak nevezték és a vajatokban csúsztatgatott apró kavicsokkal rendkívül ügyesen és gyorsan tudtak összeadni, kivonni.

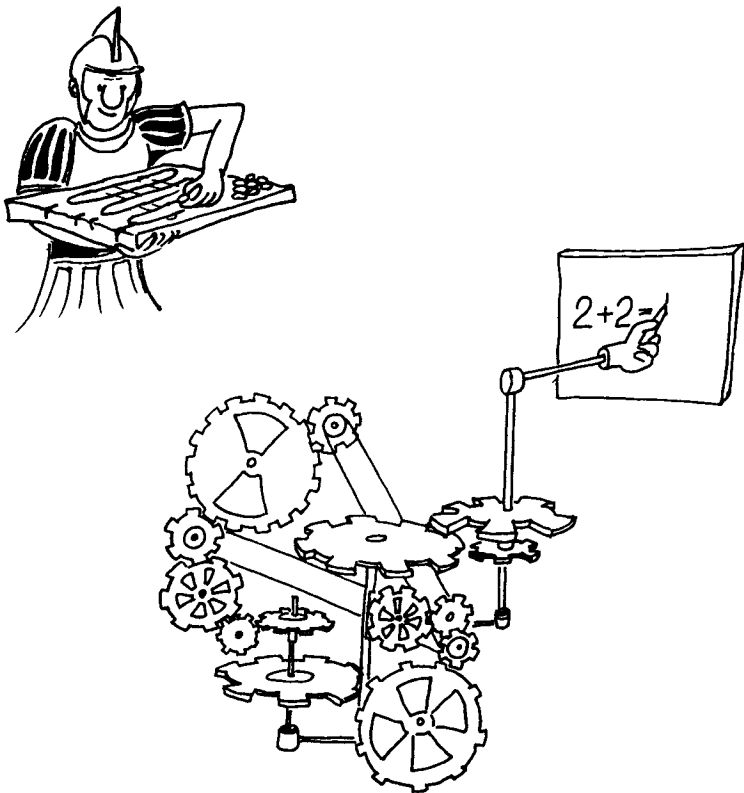
Kiseb változtatásokkal az abakusz vagy számvető tábla csaknem 3000 éven keresztül szolgált számolási segédeszközzel.

A számolás eredményét valamilyen formában rögzíteni kellett, hogy megőrizhető legyen. Kezdetben egyes helyeken rovásokat véstek fára, csontra, kőre, máshol viszont a csomókötözést találták alkalmasnak erre a célra.

Ezek a faragott csontok, rovátkolt fák vagy csomózott kötelek ugyan csak a korai számolás segédeszközei, ma úgy is mondhatnánk: „memóriái”

voltak. Érdekes, hogy a rovások alkalmazása több ezer éven át fennmaradt, és még a múlt század utolsó éveiben is használták állatok nyilvántartására vagy adósságok feljegyzésére.

A rómaiak már törtrészekkel is számoltak; az abakusz segítségével és a táblára kirakott kavics latin nevéből: a „calculus”-ból alakult ki a kalkulálni, számolni ige. Ezért nevezték és nevezik még ma is az alapműveleteket végző számológépeket kalkulátoroknak. Az abakusz azonban még nem volt számológép, mindössze segédeszköz, amely megkönnyítette a számolást.



Az ember már a számolás korai szakaszától kezdve vágyott olyan segítségre, mely megkíméli őt a számhalmazok emlékezetben tartásától és a számolás fárasztó műveleteitől, de a technika fejlődése csak a 17. században tette lehetővé a valóban hathatós segítség megteremtését.

Ismét új fejezet kezdődik . . .

Az 1600-as évek elején a számolások gépesítésére – mert ekkor már ez volt a közvetlen cél – az órásmesterség, az órastechnika gyors fejlődése és nagy-szerű eredményei ígértek megvalósulást. A finoman megmunkált fogaskerekek, apró rudazatok, csapágycsuklók technikailag lehetővé tették számológép elkészítését és Wilhelm Schickard 1623-ban élt is ezzel a lehetőséggel. Számológépe több, egymásba kapcsolódó fogaskerékből állt.

A kerekek kerületén tíz egyenlő részre osztva helyezte el a számokat 0-tól 9-ig. Egy fordulat befejezésekor a kisebb helyiértékű számot mutató kerék tovább fordította egy számjegynek megfelelő értékkel a következő helyiérték kerekét. Ugye ismerős a megoldás!

Naponta találkozunk ilyen számláló művekkel a magnetofonokban, a gáz-, víz-, villanyórában vagy a motorkerékpárok, autók kilométerórájában. Schickard gépe a négy alapművelet: az összeadás, a kivonás, a szorzás és az osztás végzésére volt alkalmas és – mint már az eddigiekből is kiderült – tízes számrendszerben dolgozott.

Tőle függetlenül, 1642-ben Pascal francia tudós szerkesztett hasonló gépet, ez azonban csak összeadni és kivonni tudott, majd 1671-ben a Leibniz által összeállított számológép már Schickard gépéhez hasonlóan szorozni is tudott.

A mechanikus számológépek történetét általában Pascal és Leibniz gépeitől számítjuk, a legtöbb könyvben az ő gépeik leírásával kezdik a számológépek bemutatását. Ennek az az oka, hogy Schickard gépét kevesen ismerték és a gép az idők folyamán elkallódott. Pontos leírása viszont megmaradt, ami alapján 1960-ban újból elkészítették és a számológép valóban tökéletesen működött. Pascal és Leibniz koruk ismertebb tudósai voltak, az általuk készített gépeket nemcsak megcsodálták, de sokan le is másolták, a másolatok ma a technikai múzeumok legértékesebb kincsei közé tartoznak.

A folytatásra 200 évig várni kell

A mechanikus számológépek fejlődésében két évszázados pihenő következett. Tökéletesítették, csinosították a gépeket, de jobbat, pontosabbat, többet tudót nem sikerült készíteni.

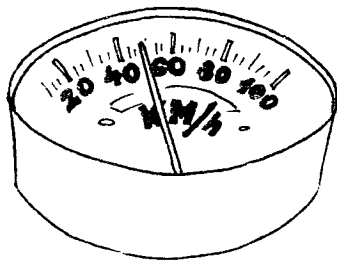
Nem is lehetne ma sem, hiszen a finommechanika mint technikai újdonság megteremtette a pontos órák, számológépek és még sok más mechanikus műszer elkészítésének alapjait, s ez egy bizonyos fokon túl már nem fejleszthető. A fejlődéshez új utakat kellett keresni.

A korábbi számológépek digitális elven működtek, lépésről lépésre számolva jutottak el az eredményhez. A kor technikai színvonala azonban határt szabott az ilyen rendszerek továbbfejlesztésének. Más módot, más kiindulási alapot kellett keresni és ezt az analóg gép megtervezésével Charles Babbage 1825-ben meg is találta.

Mi is ez az analóg eljárás?

A digitálist már ismerjük: a kavicsok rakosgatásától az átbillenő—elforduló mechanikus szerkezetekig mindenszámolási művelet kis részekből, megszámlálható lépésekből áll össze. Az átmenet egyik értékről a másikra darabos, nem fokozatmentes. Ezzel szemben az analóg eljárás összehasonlításra alapul. Két vagy több mennyiséget hasonlít össze folyamatosan. A lényeg az összehasonlításra és a folyamatosságon van. Gyakorlati példaként most sem kell messzire menni, hiszen sok olyan szerkezet, berendezés vesz körül bennünket, melyek analóg elven működnek. Nézzük például a gépjárművek sebességmérőjét. A mutató kitérése *arányos* a jármű sebességével. Ha gyakorlati, tapasztalati alapon a különböző sebességértékekhez tartozó mutatóállásokat megjelöljük, később a mutatóra tekintve a sebesség értékét leolvashatjuk. A jó sebességmérő a legkisebb sebességváltozásra is érzékeny, a mutató a változás irányától függően elmozdul. A sebességet azonban nem tudjuk pontosan megállapítani! Persze ez nem minden esetben hátrány, hiszen gyakorlatilag teljesen mindegy, hogy az autó sebessége 78,1 km/óra vagy 78,3 km/óra.

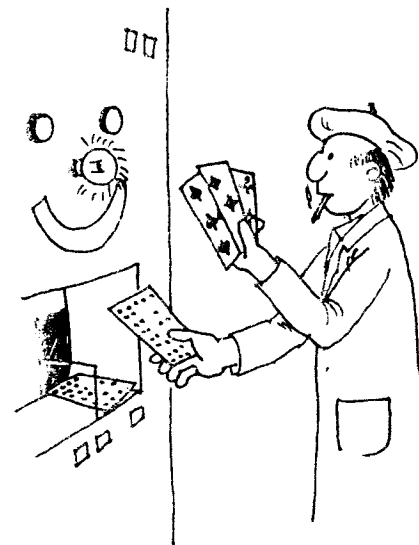
Miért van ez a pontatlanság? Mert nem tudjuk pontosan leolvasni a mutató pillanatnyi állását. A műszer beosztásai között mozgó mutató helyét becsülnünk kell, s ez kinek-kinek a képessége, gyakorlata szerint sikerül. A gyorsan áttekinthető eredményért a pontossággal kell fizetnünk minden analóg elven működő eszköznél.



De térjünk vissza Babbage számológépéhez! Az a felismerése, hogy az akkor egyeduralkodó digitális elven működő gépek mellett más — analóg — eljárás is követhető, korszakalkotónak és a számítógépek történetében alapvetőnek bizonyult. Ne higgyük azonban, hogy az első analóg számológésközt ő „találta fel”. Jó kétszáz évvel korábban elkészült a ma is használatos logarléc, amely analóg számolóeszköz. Babbage ismerte a logarlécet és bizonyára ez is közrejátszott abban, hogy szakított az addigi elképzelésekkel és kidolgozta egy mechanikusan működő, de analóg elvű számológép terveit.

Jó száz év múlva az elektromosság megismerése teremtette meg a technikai feltételt gépe elkészítéséhez. Ettől függetlenül őt tekintjük a

modern számítógépek atyjának, hiszen elvi megállapításai alapozták meg a korszerű számítógépek elkészítését.



Vissza a digitális technikához

1808-ban egy kísérletező kedvű francia takács, Joseph Marie Jacquard automata szövőszéket készített. A mintának megfelelően ezután a szálakat nem kézzel kell kiszámolni, kiválasztani, kiemelni, hanem azt egy lyukasztott papírkártya, a lyukkártya végzi. A működés egyszerű és megbízható. A hosszanti szálakat átfűtik egy függőlegesen álló tűszerű fémpálca felső végén levő lyukon és a pálcák alsó vége a papírkártyára támaszkodik. Ha a kártyán valamelyik pálcá helyén lyuk van, a pálcá beleesik a lyukba, ezzel a hozzá tartozó fonalszálat elmozdítja. A lyukak megfelelő elhelyezésével tetszés szerinti mintát lehetett kialakítani, a korábbi fásasztó és nagy figyelmet igénylő kézi munkát automata gép vette át. A lyukkártya-technika gyorsan terjedt és Herman Hollerith ötlete alapján a gépi adatfeldolgozás máig is használt eszközként vált. Hollerith történelmi jelentőségű felismerése abból állt, hogy az azonos méretűre vágott kartonlapokon a lyukak megfelelő rendszerű elhelyezésével számokat, betűket, jeleket — adatokat — rögzíthetünk.

A lyukasztáshoz lyukasztókészüléket szerkesztett és létrehozott egy kódrendszert, ami alapján az adatokat a kártyára lyukasztotta. Ezután a lyukkártyákat egy osztályozó gépbe tette. A kártyák itt elektromos érintkezők között haladtak végig: ahol lyuk volt a kártyán, záródott egy-egy áramkör. Az érintkezők összekapcsolásával egyszerű módon el lehetett érni, hogy tetszés szerinti lyukkombinációnál adjon jelzést a gép, illetve a jelzéssel egyidőben a kártya alatt kinyíljon egy ablak, amin át a kívánt kódnak megfelelő kártya beleesett egy fiókba. Az érintkezők különböző összekapcsolásával más és más szempontok szerint válogathatták át a kártyákat illetve gyűjthették össze a kártyákon lyukasztással rögzített adatokat. Végül az összegyűjtött kártyákat egy számlálómű megszámlálta és a mai mutató órákhoz hasonló számlapon jelezte folyamatosan az eredményt. A lyukkártyás adatfeldolgozás főpróbájára az 1890. évi népszámlálás adatainak feldolgozásánál került sor az Egyesült Államokban. Az igencsak kezdetleges gép segítségével a korábnál sokkal gyorsabban végeztek az adatok feldolgozásával és bebizonyosodott most már a gyakorlatban is, hogy ezen a módon milyen hatalmas mennyiségű adatot lehet egyszerűen tárolni és gyorsan feldolgozni.



A számológéptől a számítógépig

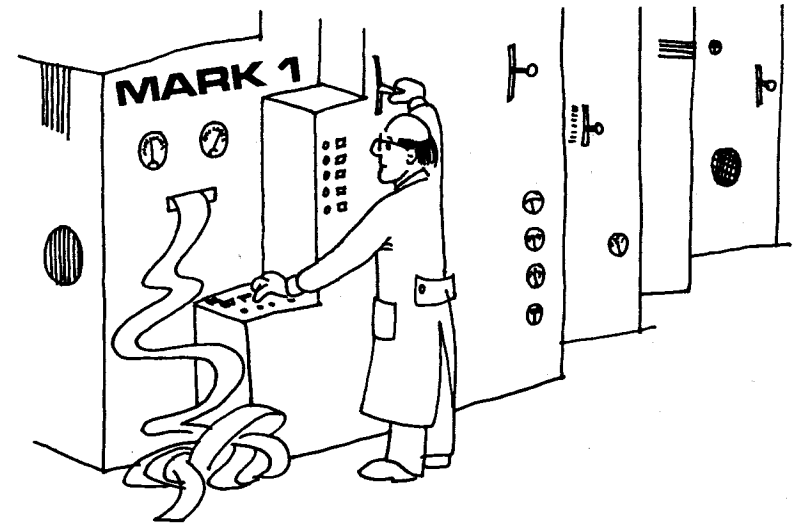
A lyukkártya új útra terelte a számológépekkel kísérletezőket. Nyilvánvalóvá vált, hogy a kártyákon rögzített adatokkal jól lehet dolgozni. Igaz, a kiértékelés, a válogatás még mechanikus eszközökkel történt, azonban ezek elvileg teljesen mások voltak, mint a fogaskerekekből, rudazatokból, hajtókarokból felépített bonyolult korábbi számológépek. Működési sebességében össze sem lehet hasonlítani a két rendszert, a lyukkártyás gép olyan gyorsan dolgozott.

De volt még egy nagy hibája. Amint elvégzett egy adott feladatot, megállt. Ekkor az eredménytől függően a lyukakat letapogató érintkezők más kombinációját kellett beállítani és lehetett tovább dolgozni. Babbage, akiről korábban már szóltunk, azt tervezte, hogy gépe nemcsak egyes számításokat fog végezni, hanem ezeket össze is fogja kapcsolni, tehát ő már a programozásra gondolt. A programot ő is lyukkártyára tervezte, de mint

tudjuk, a gép nem készült el. De ha el is készül, nehezen képzelhető el, hogy az 50 000 számkerékből felépülő gép beváltotta volna a hozzá fűzött – bár elvileg helyes – reményeket.

Az elektrotechnika a 19. század végén gyors fejlődésnek indult és hatása a számítógéptervezők, -építők munkáját is alapjaiban változtatta meg. Nehéz lenne pontosan felsorolni minden kísérletet, melyek többnyire eredményesek is voltak, mert ezek jó része az 1940 utáni időkre esik. Ezek az évek pedig már a II. világháború évei, és a számítógép is – mint olyan sok más technikai vívmány – háborús feladatok ellátásával kezdte pályafutását. A következőkben csak a fejlődés szempontjából érdekes három típus – a tisztán elektromechanikus, a vegyes és a tisztán elektronikus felépítésű gép – legjellegzetesebb darabjairól lesz szó.

1941-ben Zuse elektromechanikus számológépet szerkesztett, ami 2600 jelfogóból állt. Ő kettes számrendszerben működtette gépét, hiszen az elektronika, az elektromos áramkörök működése mellett ez szinte természetes is volt. (A kettes számrendszerben csak két számjegy, a 0 és az 1 szerepel. Nyitott kapcsolónak a 0, zárt érintkezőnek az 1 felel meg, ezért a számok tárolására, műveletek végzésére a jelfogók és kapcsolók jól alkalmazhatók.)



Vele egyidőben az Egyesült Államokban készült a MARK 1 számítógép, ami jelfogókat és elektroncsöveket vegyesen tartalmazott. Érdemes a gép méreteire felfigyelni. Tömege 35 tonna volt és 16 méter hosszú szekrénsorhoz hasonlított. Számolási sebessége is tekintélyes, legalább is a korábbi

gépekhez viszonyítva. Egy összeadást 0,3 másodperc alatt, egy szorzást 6 másodperc alatt végzett el. Ebben az időben már az Egyesült Államokban dolgozott a magyar származású matematikus, Neumann János, aki az egyre gyorsabban dolgozó, de még mindig csak a számológépből mai értelmében is helytálló számítógépet készített. Korábban a lassan dolgozó számológépek-nél nem okozott nagy gondot, hogy egy-egy művelet után a további utasításokat emberek adják a gépnek, ezzel a működés nem lassult észrevehetően. Az új gépeknél, különösen a MARK 1-nél a számolási műveletek már olyan gyorsak voltak, hogy a gépnek sokat kellett várakozni, amíg az új utasítást megkapta. A tisztán elektronikus gép sebessége pedig több százszoros vagy ezerszeres is lehetett volna, ha nem kell idejének túlnyomó részét várakozással tölteni.

Neumann János a következőket javasolta:

- Az új számítógépek teljesen elektronikusak legyenek, azaz ne legyen bennük mechanikusan mozgó alkatrész. Ezzel lehetővé válik az igen nagy sebességű működés.
- A gépek kettes számrendszerben dolgozzanak. Az elektronika szinte kínálta a kettes számrendszer alkalmazását, így ez természetes volt.
- Legyen a gépeknek belső memóriája. (Ez Babbage elképzelése volt.) Segítségével a számítások eredményeit a gép tárolta, újabb számításokor azonnal fel tudta használni.
- Végül a legfontosabb, a program tárolásának kidolgozása volt az, amitől a számológép számítógéppé változott. A tervezett gép tehát tárolja belső memóriájában az összes programlépést, azokat az utasításokat, melyeket eddig lépésenként kellett a géppel közölni. A műveletek gyors elvégzése után ugyanilyen gyorsan megkeresi a következő utasítást és máris végzi tovább a számításokat.

Részben ezeknek az elveknek megvalósításával készült el – még Neumann János nélkül – 1944-ben az első teljesen elektronikus számítógép, az ENIAC. A hatalmas termet betöltő gép 18 000 elektroncsövet használt és energiafelhasználása 174 kW volt. Történeti érdekesség, hogy ezt a gépet nevezték el „computer”-nek és az elnevezésen azóta általában számítógépeket értünk.

Mit tudott az ENIAC?

A mai gépekhez viszonyítva természetesen nagyon keveset. Akkor azonban még a szakemberek is meglepetten figyelték az ember számára már követhetetlenül gyors működést. Az első feladat, amit a gép megoldott, egy atomfizikai probléma volt. Ha hagyományosan, papírral, ceruzával nekikezdett volna egy fizikus, több mint 100 évig kellett volna számolnia. A gép két hét alatt végzett a számítással, de ennek az időnek a nagyobb része is csak a programozással telt el, a tényleges gépidő 2 óra körül volt.

Ez volt az első számítógép.

Vagy talán mégsem?

Utólag nehéz eldönteni, mert a háborúban a kutatásokat titkosan kezelték. Tény viszont, hogy közelítőleg ugyanebben az időben Angliában is elkészült egy hasonló méretű és teljesítményű gép, a COLOSSUS 1, amit viszont hadi célokra, a rejtjelezett üzenetek megfejtésére használtak, ezért létezéséről csak később jelent meg tájékoztatás.

A német szakirodalom természetesen Zuse gépétől számítja a számítógépkorszakot. Ezen azonban ne csodálkozzunk, hiszen mindhárom gép alapvetően eltért elődeitől és kisebb-nagyobb mértékben már a mai elvek szerint működtek. Nem érdemes sokat foglalkozni az elsőbbség kérdésével, mert a lényeg mindhárom esetben azon az igyekezeten van, mellyel a negyvenes évek kutatói megteremtették a számológéptől minőségében, szolgáltatásaiban, működési elvében többet és mást nyújtó gépet, a *számítógépet*.

Számítógép-generációk

A számolás kialakulása kezdetén néhány ezer év pontosságával határoztuk meg a számunkra érdekesebb eseményeket.

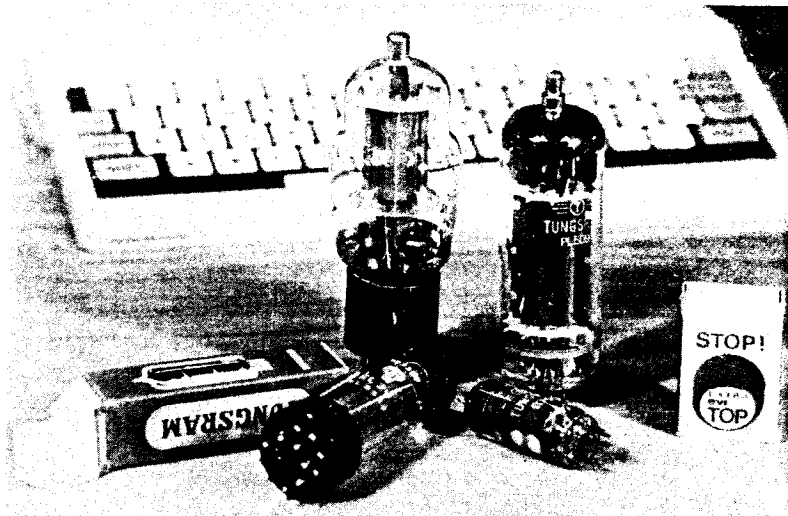
A fejlődés üteme ilyen pontatlanságokat is megengedett akkor. A számológépeknél már gyorsabban követték egymást az események, de egy-két száz év eltérés különösen az első időkben még itt is megengedett. Jól érezhető azonban a fejlődés gyorsulása. A számítógépek első példányai már egy évtizeden belül készültek el. Itt már az éveknek, sőt a hónapoknak is jelentősége van. Ami azóta történik, azt viszont már gépenként és alkotóként időrendbe szedni lehetetlenség. A számítógépek fejlődése eddig még soha nem tapasztalt méreteket öltött. Ha most leírnám a pillanatnyilag legtekintélyesebb számítógép „tudományát”, „képességeit”, mire ez az írás nyomdába – illetve ez a könyv a kezetekbe – kerülne, már megjelenne legalább tucatnyi, annál sokkal többet tudó gép.

Azért valami rendszerezést mégis lehet tenni, ehhez a gépek alkotó elemeiből kell kiindulnunk.

Az első gépek tisztán elektroncsövekkel működtek. (1. kép)

Ezeket *első generációs* gépeknek nevezzük. Nem sok készült belőlük, mert a hihetetlen mennyiségű alkatrész közül mindig meghibásodott valamelyik, s így az üzembiztonság nem volt valami nagyszerű.

A sok elektroncső tetemes energiát fogyasztott és ennek döntő többsége a környezetet fűtötte. A keletkezett hő károsan befolyásolta más alkatrészek működését, ezért gondoskodni kellett a folyamatos hűtésről, az állandó hőmérséklet biztosításáról. A gyakori hibák miatt minden számítást többszörösen ellenőrizni kellett, ami megnövelte a gépek munkaidejét, és ezzel nőtt az újabb hiba valószínűsége. De már működtek és hibáik ellenére az ötvenes évek közepéig használták is őket. A Szovjetunióban BESZM, URAL, az Egyesült Államokban IBM, UNIVAC néven készültek ezek az óriások.

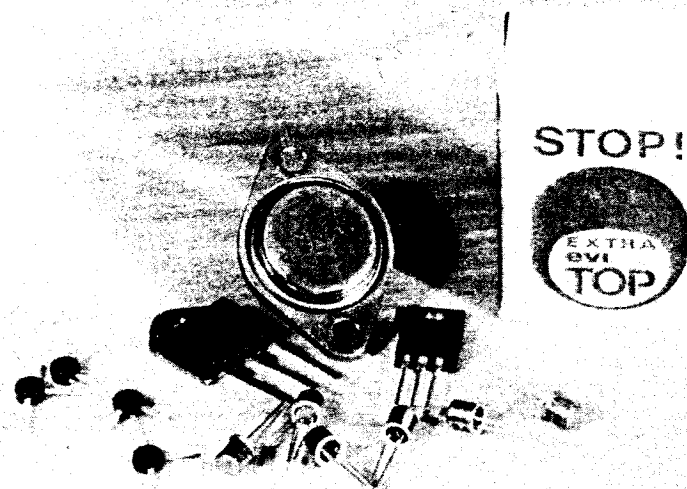


1. kép

1948-ban elkészült az első tranzisztor és néhány év múlva már tömegesen gyártották. Megnyílt az út a számítógépek új generációjának, a félvezető vagy *második generációs* gépek előállításához. A tranzisztor felfedezése újabb fontos határkő nemcsak a számítógépek, hanem az egész elektronikai ipar fejlődésében.

Mi is a tranzisztor? Valószínű, hogy mindannyian találkoztatok már ezzel a piciny háromlábú alkatrészsel, hiszen a zsebrádióktól kezdve csaknem minden elektronikai készülék nélkülözhetetlen alkotórésze volt. Ezek a készülékeken jól olvasható a „Solid-State” kifejezés, ami magyarul szilárd testet jelent. A tranzisztor olyan alkatrésze ezeknek a készülékeknek, mely vezérli, erősíti a rajta átfolyó áramot és ez a folyamat egy kristályos félvezető anyagban, vagyis szilárd testben játszódik le. Az elektronszóben a hasonló folyamatok ritkított gázban, izzószállal fűtött alkatrészek között zajlanak, ezért az elektronszó mindig melegszik. A méretei sem lehetnek nagyon kicsik, hiszen tíz-húsz apró részből, lemezekből, izzószálból, huzalokból és természetesen üvegburából kell előállítani. A tranzisztor ezzel szemben egyetlen piciny kristály, hozzáerősítve a három láb és a védő fémtok. Részletesebb működését később majd tanulni fogjátok. (2. kép)

A félvezetőkből, tranzisztorokból felépített elektronikus eszközök mindazt tudják, amit a régebbi elektronszóes készülékek. Ugyanakkor igen sok előnyük is van. Ezek közül lássunk néhányat:



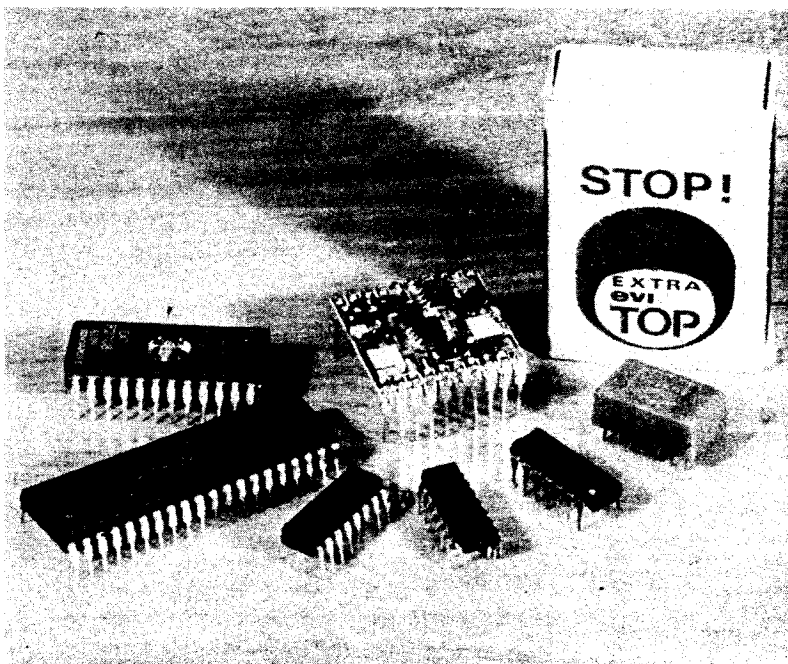
2. kép

- Lényegesen tartósabbak. Az izzószál az elektronszóben előbb utóbb kiég, hasonlóan az izzólámpákhoz és ezzel használhatatlanná válik. Tranzisztorban nincs izzószál, élettartalma tehát olyan nagy, hogy csak becsülni tudjuk.
- Sokkal kisebb energiával működésbe hozható, hiszen nem kell egy izzószálat állandóan fűteni.
- Bekapcsolás után azonnal működik, mert nem kell megvárni, míg az izzószál bemelegszik.
- Előállításuk egyszerűbb, kevesebb alkatrészből készülnek.
- Olcsóbbak.
- Rázkódást, ütést jól bírják, nem úgy, mint az üvegburás elektronszó, ami hamar eltörik.

Ennyi előnyös tulajdonság éppen elég volt ahhoz, hogy rövid időn belül elkészüljenek az első tranzisztoros számítógépek. A méretek csökkentek. A korábbi teremnyi méret már egy íróasztalban elfért és természetesen a felhasznált elektromos energia is csak töredéke volt az előzőnek.

A legnagyobb előny azonban mindenképpen a megbízhatóság lényeges megnövekedésében jelentkezett. Ezek a gépek már alkalmasak voltak hosszabb idejű felügyelet nélküli működésre. Jól bírták a rázkódást, ütődést, tehát megfeleltek az akkor induló úrkutatás szigorú feltételeinek. A műholdak mind vittek magukkal kisebb-nagyobb számítógépet, de a földi irányítóközpontok munkája sem képzelhető el nagyteljesítményű számítógépek nélkül. Az első generációs nagy számítógépekbe az elektroncsöveket foglalatok rögzítették. A bizonytalan vagy elpiszkolódott érintkezés sok hiba forrása volt. A második generációs gépeknél az alkatrészeket forrasztással kapcsolták egymáshoz. A gyakorlat azonban azt mutatta, hogy ez a kötésmód — aminél tökéletesebbet akkor nem ismertek — nem is olyan tökéletes. A nagy megbízhatóságú félvezető eszközöket gyenge forrasztással összeszerelve nem tudják kihasználni az alkatrészek jó minőségéből adódó előnyöket.

A megoldás elvben egyszerű volt: a különböző félvezető alkatrészeket huzalozás nélkül, közvetlenül egymás mellé kell már a gyártáskor helyezni. Elkészültek tehát az első ilyen szoros felépítésű alkatrészek, melyeket integrált áramkörnek neveztek el. (3. kép)



Az eleinte 10–50 tranzisztorból álló egységeket hamarosan követték a bonyolultabb, több alkotórészből álló integrált áramkörök. Ma már nem számít ritkaságnak az olyan integrált áramkör (angol nevének rövidítéséből IC), amely 1 mm² területű kis lapocskán akár több ezer tranzisztort is tartalmaz.

A számítógépek működéséhez — majd később látni fogjuk, miért — azonos típusú áramkörökből kell nagyon sok. Ha ezeket egy-egy IC-ben csoportosítják, kevesebb szereléssel kisebb méretű számítógépeket lehet készíteni. Az IC-eket tartalmazó számítógépek a *harmadik generációs* számítógépek. Az integrált áramkörök gyártása — éppen számítógépek segítségével — teljesen automatizálható, a tömeges termelés viszont csökkenti az előállítás költségét. Ma egy átlagosan bonyolult IC alig kerül többre, mint egy hagyományos tranzisztort, pedig belsejében több ezer tranzisztort is tartalmazhat.

Az integrált áramkörökből felépített modern számítógépek rendkívül gyorsak. A programozást viszont emberek végzik, akik összehasonlíthatatlanul lassabban dolgoznak. Hogyan lehet feloldani ezt az ellentmondást? Nyilvánvaló, hogy nem a gépek lassításával! A megoldást az időosztásos eljárás adja. Egy gyors gép akár több száz programon is dolgozhat egyidőben.

Először elvégzi az első programozó által kért műveleteket, majd átlép a másik programozóra és ott is elvégzi a soron következő feladatot. Így folytatja sorban mindaddig, míg minden programozót végig nem látogat, majd újból kezdi előlíról. Mindez a másodperc tört része alatt megy végbe, így a programozók azt hiszik, a gép csak velük foglalkozik. A gép saját munkáját úgy osztja be, hogy a sok, egymástól független feladat ne keveredjen össze, egyik feladat végzése ne zavarja vagy hátráltassa a másik végzését. A gép látszólag megsokszorozza önmagát.

A kisebb teljesítményű, úgynevezett személyi számítógépek egyszerűbb esetben 2–3 IC-ből állnak csak és méreteik egy könyv méreteivel azonosak, sőt akár kisebbek is lehetnek. A méretek csökkentésében már csak a kezelhetőség szab határt. Túl kis billentyűkkel ugyanis nehéz dolgozni, ezért a gépek méretei a közeljövőben sem fognak ennél jobban lecsökkenni. Legalábbis addig, míg az adatokat, utasításokat, a programot billentyűkön keresztül, kézzel kell a gépbe juttatni. Ma ez a gyakorlat, de már dolgoznak az emberi beszédet megértő számítógép tervein, s akkor a billentyűk is feleslegessé válnak.

Az integrált áramkörök fejlődése sem állt meg. Napjainkban a miniatürizálás olyan határig jutott, hogy a félvezető eszközöket molekuláris vastagságú rétegekből igyekeznek felépíteni. Így kis helyen még bonyolultabb műveletek végzésére alkalmas áramköröket hozhatnak létre. Az új alkatrészből új számítógépeket építenek, az új számítógépekkel pedig újabb alkatrészeket tervezhetnek. A fejlődés megállíthatatlan. Azt viszont, hogy milyen ütemben és milyen irányban folytatódjon a gépek fejlődése, közü-

letek döntik majd el azok, akik ma még csak ismerkednek, barátkoznak a számítógéppel, de pár év múlva esetleg már szakemberként élethivatásukul választják a számítógépek világának tervezését, alakítását.

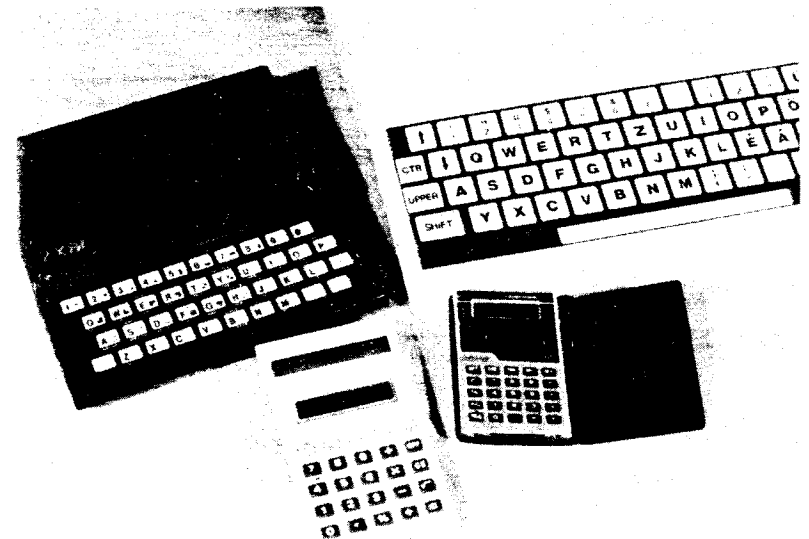


MIBŐL ALL A SZÁMÍTÓGÉP?

Két betű a különbség

Húsz évvel ezelőtt az iskolákban teljesen természetes volt, hogy egy-egy hosszabb matematika- vagy fizikafeladat megoldása több oldalnyi helyet foglalt el a füzetekben. A szorzásokat, osztásokat, de még a magasabb rendű műveleteket, a hatványozást és a gyökvonást is írásban hosszú és gyakran fárasztó munkával kellett elvégezni. Aki tehetett, megtanult logarléccel számolni, vállalva az ezzel járó pontatlanságokat és a tévesztés lehetőségét.

Közületek már kevesen láttak logarléccet és még kevesebben azok száma, akik használni is tudják. Az idő eljárt felette és a zsebszámológépek nemcsak helyettesítik; de lényegesen többet tudó segítő társainkká váltak.



A zsebszámológépek nem számítógépek. A magyar nyelvben azonban a számoló és a számítógép nagyon hasonlóan hangzik, csak két hang vagy – leírva – két betű a különbség. De ez a két betű kicsoda különbséget jelent!

A mikroelektronika csodái ezek a sok nyomógombos kis készülékek és gyakran elbizonytalanodunk egy-egy új típus láttán, mi is ez tulajdonképpen.

Könnyen arra gondolhatunk, hogy a programozható gép a számítógép, a nem programozható pedig a számológép.

Pedig nem ez a különbség!

A meghatározó a gép kapcsolata a külvilággal, környezetével. Amelyik gépekhez kapcsolhatók különféle tartozékok, eszközök, azok *számítógépek*.

A képernyő és az írógépszerű billentyűzet egyértelművé teszi a választást, a kis – zsebben is hordozható – gépek számológépek, még ha programozhatók is. Ezek programozása vagy a billentyűkön keresztül vagy – ami szintén gyakori – kis mágneses lemezzel történik. Ügyeljünk tehát a helyes elnevezésre, szóhasználatra! (4. kép)

Szedjük szét a számítógépet?

Akiben a technikai érdeklődés leghalványabb lángja is lobog, egy-egy új, számára ismeretlen eszköz láttán erős kísértést érez az alapos ismerkedésre. Szemügyre veszi kívülről, majd bekapcsolja és igyekszik mindent megtudni az eszköz „képességeiről”. Az ilyen ismerkedést a gyártók megkönnyítik és minden eszközhöz, géphez részletes használati utasítást mellékelnek.



Csak eddig és ne tovább! Az ismerkedést a szakemberek csak eddig javasolják, pedig milyen jó lenne azt is tudni, mi van a gép belsejében és – esetleg – hogyan működik. A csavarhúzó mindig kéznél van, egy két mozdatat és...

A folytatást az alábbiak közül többnyire a véletlen dönti el:

1. Nem történik semmi. A csavarok nem mozdulnak, mert esetleg nem megfelelő szerszámokkal kezdjük a szerelést. Erőltetés, feszítés – és a drága gép jóvátehetetlenül megrongálódik.
2. A csavarok szépen kijönnek. Egyik innen, másik onnan, de ki tudja azt mind megjegyezni. Összerakáskor azután sehogyan sem akar a dolog sikerülni. Ez még a kisebb baj, mert ezen egy szakember könnyen segít. De mi történik, ha „csakazértis” össze kell rakni. A finomabb készülékek ezt általában nem élik túl.
3. A csavarok kijönnek. Azok is, amelyeket nem kellett volna kicsavarunk. Amikor a dobozt kinyitjuk, meglepődve látjuk, hogy néhány alkatrész ide-oda vándorol és közben üti-veri a helyükön maradtakat. Itt már csak a szakember segíthet, viszont ez nem olcsó megoldás.
4. A doboz szépen kinyílik. Előttünk áll a csodálatos szerkezet. Persze hogy be is kapcsoljuk, hiszen ha eddig eljutottunk, már a működést is meg kell nézni! Egy rossz mozdatat, néhány szikra és csattanás kíséretében áram nélkül maradunk. A lehetséges két rossz közül ez a jobbik. Valószínű, hogy a készülék alaposan tönkrement, de egy életveszélyes áramütést is kaphattunk volna. Irány a biztosítékszekerény majd a szerviz.

A számítógépben több olyan alkatrész is van, amit elég csak megérintenünk, és máris tönkremegy. Persze nem valami gonosz varázserőttől, ami belőlünk árad, hanem attól az elektromos töltéstől, amit műszálas ruhadarabjainknak, műanyag padlóburkolatunknak, szigetelőtalpú cipőnknek köszönhetünk.

Az ilyen elektromos (elektrosztatikus) feltöltődés ellen viszont nagyon nehéz védekezni. A javítóműhelyekben egész sor különleges védőberendezést használnak, hogy elkerüljék az érzékeny alkatrészek ilyen meghibásodását.

Ezzel már válaszoltunk is a címben feltett kérdésre. Semmiképpen sem szabad szétszedni a számítógépet!

Az ismerkedésnek vannak más, sokkal veszélytelenebb és hatékonyabb módjai is. Gondoljuk csak el, ha belenézhattunk volna Babbage egykori gépébe és ott meglátjuk az 50 000 fogaskereket. Vajon megértettük volna-e működését? Vagy az ENIAC több ezer egyforma elektroncsövéből milyen következtetést vonhattunk volna le? Azt hiszem, nagyjából ugyanígy járunk, ha egy mai számítógépet felboncolunk. Néhány „százlábú” integrált áramkörön, néhány apró alkatrészen kívül semmit sem látunk. Ennyitől pedig nemigen leszünk okosabbak, hiszen alkatrészeket külön-külön már többnyire mindannyian láttatok.

Ismerkedünk

Szerte a világon rengetegféle számítógépet gyártanak. Nagyokat, kisebbeket és egészen kicsiket. A legkisebbek a személyi számítógépek. A működési elv minden gépnél ugyanaz, sőt még fontosabb alkatrészeik is többnyire megegyeznek, így elegendő a legegyszerűbb gép működésével megismerkednünk és akkor már a nagy gépek sem lesznek idegenek többé.



Magyarországon is gyártanak néhány éve személyi számítógépeket. A HT 1080 Z típusú iskolaszámítógépet már közületek is bizonyára sokan látták, esetleg néhány érdekesebb játékprogramjával már játszottatok is. Ezt a gépet a középiskolásoknak készítik, és minden középiskolába eljutott már pár darab.

A PRIMO kevésbé ismert még, hiszen alig egy éve került le a tervezők asztaláról. Ez az igazi, nektek való – úttörő gép! Magyar tervek alapján magyar és néhány szocialista országban gyártott alkatrészből készül. Szép formájú, jól kezelhető és érdekessége, hogy billentyűit nem kell (nem is lehet) lenyomni, csak megérinteni. Az éfintésre halk fűttyel válaszol. Bármilyen televíziókészülékhez hozzá lehet kapcsolni, a programokat pedig közösleges magnetofonon lehet tárolni.

Az ismerkedésre mi ezt a gépet választottuk. Lássuk tehát először is, mit ír róla a gyártó vállalat. Részletek a műszaki adatokból:

Megnevezése: PRIMO

Típus: A-32

Mikroprocesszor: 8 bites (U 808)

Operatív tár kapacitása: 16 Kbyte RAM
16 Kbyte ROM

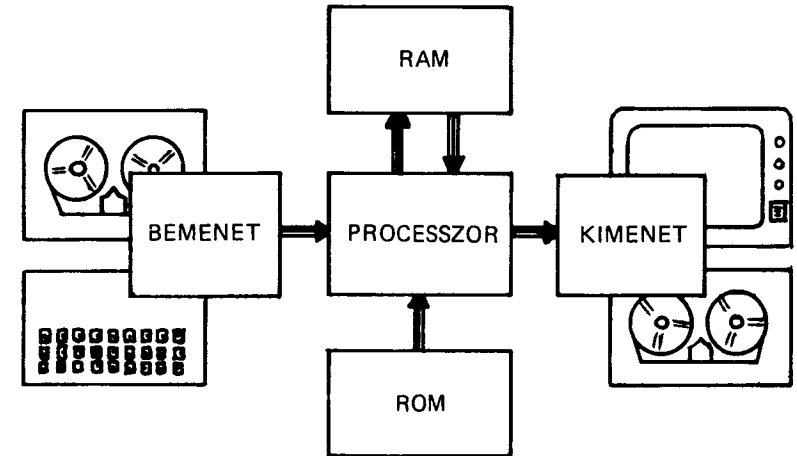
Program: BASIC

Tömeg: 800 g

A PRIMO gép – hasonlóan a többi személyi számítógéphez – a következő fő szerkezeti részekből áll:

1. Bemeneti egység (billentyűzet vagy magnetofon)
2. Adattár (RAM)
3. Programtár (ROM)
4. Kimeneti egység (tv vagy magnetofon)
5. Mikroprocesszor

A szerkezeti rajzból (1. sz. ábra) láthatjuk, hogy a gép lelke a mikroprocesszor, hiszen az minden szerkezeti résszel közvetlen kapcsolatban van. Ezek szerint ez a legfontosabb része a számítógépnek?



1. ábra

Nem! A számítógép részei között helytelen ilyen kérdést feltenni, mert mindegyik rész a maga területén nélkülözhetetlen, egymást nem pótolják, önállóan egyikük sem működik.

Természetesen a mikroprocesszor sem! Az öt rész együtt adja a számítógépet és az egyes részek tulajdonsága a számítógép tulajdonságaiban jelentkezik.

Mit jelent ez?

Ha például az adattár kis befogadóképességű (úgy is mondjuk: kis kapacitású), hiába van nagy programtár és legújabb típusú mikroprocesszor a gépben, alig-alig használhatjuk valamire, hiszen képtelen befogadni egy komolyabb feladat elvégzéséhez szükséges adathalmazt.

A számítógép tervezői nagy igyekezettel gyűjtik össze mind a négy fő részből a legtöbbet nyújtó alkatrészeket, hogy az elkészült gép is a lehető legtöbbet tudja. Így van ez a PRIMO-nál is.

Hogyan mondjam? . . .

Máris itt vagyunk az első problémánál. Hogyan mondjam meg a gépnek, mit csináljon? Sajnos, ez a gép még nem érti közvetlenül az emberi beszédet, de azért könnyen kapcsolatba léphetünk vele. Erre a célra szolgál *bemeneti egysége*.

Ez az a része a számítógépnek, melyen keresztül kapcsolódik kezelőjéhez, tehát olyannak kell lennie, hogy mi, a kezelők könnyen el tudjunk rajta igazodni. Nézzük csak az 1. ábrát: az információ itt egyirányú, az embertől a gép felé mutat. A PRIMO bemeneti egysége két módon is lehetővé teszi, hogy kapcsolatba lépjünk a géppel. Lehetőségünk van a billentyűzeten keresztül „beszélgetni”, de az előre elkészített programokat egyszerűen magnetofonról is „bejátszhatjuk” a számítógépbe. Ez a két mód általánosan elfogadott a személyi számítógépeknél, de a mai gépek „megértik” még a mágneslemezről beolvasott programokat, sőt két vagy több gépet közvetlenül is össze lehet kapcsolni és ekkor egymás között lehet a programokat vagy az adatokat cseréltetni.

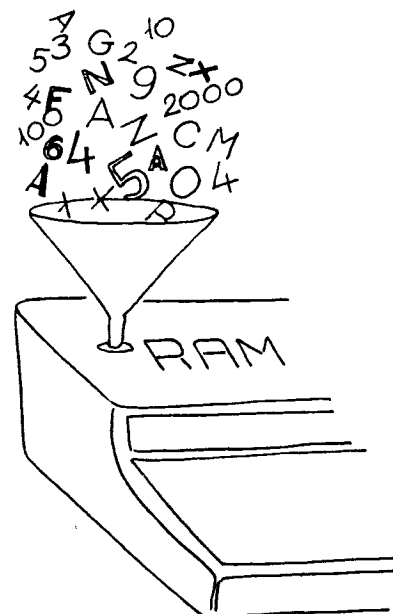
A billentyűzet kicsit szokatlan, hiszen hiába nyomjuk, nem mozdul. Nem is kell nyomni, csak éppen megérinteni. Mégpedig minél nagyobb felületen, hiszen ez a titka működésének. Ujjheggyel vagy körömmel megérintve alig-alig várhatunk jó működést. A billentyű elmozdulását hangjelzés igyekszik pótolni, s ezzel egyidőben már látjuk a képernyőn is a kívánt változást.

A gyakorlat igazolta a tervezőket, a billentyűzet jól működik. De miért jó az ilyen különleges billentyűzet?

Két fontos előnye is van a hagyományos megoldásokkal szemben. Az első és igen lényeges előny az ára. Ez sokkal olcsóbb minden más típusú billentyűnél. Az egyszerű személyi számítógépeknél a hagyományos billentyűzet ára eléri a gép árának felét. Az olcsó gépeknél ezért a lehető legegyszerűbb megoldásokat keresik és alkalmazzák. (Az ugyancsak olcsó ZX 81 típusú gépnél például két rugalmas fóliacsíkot nyomnak össze billentyűként.) A másik előny a megbízhatóság. A hagyományos billentyűzet megnyomáskor elmozdul és az elmozdulás során mechanikus vagy tisztán elektromos úton zárja a kívánt áramköröket. A mozgó billentyűzet könnyebben sérül, meghibásodik, az egyszerű érintkezők elpiszkolódnak, elkopnak, az elektronikus megoldás (pl. Hall generátor) viszont még drága is. Ennyi érv azt hiszem bőségesen elég az érdekes billentyűmegoldás mellett. Természetesen a kezelést meg kell tanulni, meg kell szokni. Kezdőknek ez nem okoz gondot, a gyakorlott gépkezelők pedig – ha néha kritizálják is a megoldást – bizonyosan megértéssel fogadják.

Könnyen tanul, könnyen felejt

A következő fontos rész az adattár (RAM). A gép bekapcsolásakor üres, nem tartalmaz semmi adatot. Mint a neve is mutatja, ez a rész jegyzi meg kívánságainkat, az elvégzendő feladatok adatait mindaddig, míg a gépnek ezekre szüksége van a feladat elvégzése során. A kapott részeredményeket ugyancsak ebben az adattárban tárolja a gép.



Az adattár jellemzője az a kapacitás, befogadóképesség, amit a műszaki adatok között mindig megtalálunk. A PRIMO-nál ez legalább 16 Kbyte. 16 Kbyte (16 kilobájt) olvassuk átlagos memóriakapacitást, a pontos jelentésére később visszatérünk. A „legalább” szó pedig azt jelenti, hogy a PRIMO számítógép többféle memóriakapacitással készül, s közülük a legkisebb a 16 Kbyte-os. Általában a memória nagysága 16 Kbyte vagy ennek többszöröse, a PRIMO gépeknél ez 32 Kbyte-ot, illetve 48 Kbyte-ot jelent. A népszerű és egyben olcsó külföldi személyi számítógépek ennél jóval kisebb kapacitásúak. Például a közismert ZX 81 típusú gép csak 1 Kbyte, a Commodore VC 20-as pedig csak 3,5 Kbyte memóriakapacitással rendelkezik. Ne gondoljátok azonban, hogy az 1 Kbyte-tal sokat lehet kezdeni.

Ezt nagyon jól tudják a gép gyártói is, ezért készítenek hozzá úgynevezett memóriabővítőt, amit csatlakoztatva a ZX 81 is 16 Kbyte kapacitású lesz, azonban ennek az ára legalább annyi, mint magáé az alapgépé. Így azután végeredményben nem is olyan olcsó az olcsó gép! A PRIMO-nál nincs szükség ilyen kereskedői fogásra.

Az adattár — angol rövidítése szerint RAM — tehát a számítógépnek az a része, mely szükség esetén megjegyzi az adatokat, s ha már a további munka során ezekre nincs szükség, feleslegessé válnak, egyszerűen kitörli őket, a helyükre írja az új adatokat.

Nehezebben tanul, nehezebben felejt

A számítógép következő része a programtár — angol nyelvű rövidítése: ROM. Lényegében mindazt el lehetne róla mondani, amit az adattárról. Ugyanúgy adatokat tárol, tárolóképességét — kapacitását — szintén Kbyte egységekkel mérjük.

A PRIMO programtár kapacitása 16 Kbyte, ami egy ilyen kisgépnél meglepően sok. Nyilvánvaló, a tervezők jól, egyszerűen, de sokoldalúan használható gépet akartak készíteni és ehhez elég nagy programtárra van szükség.

A programtár (ROM) és az adattár (RAM) a PRIMO legkisebb változatánál tehát azonos kapacitású. Nagy különbség azonban, hogy az adatokat nem mi közöljük a programtárral, sőt nem is a gép, hanem azt még a gyárban beírják és ezen általában már nem is lehet változtatni. Ezeket az előre rögzített adatokat nevezzük a gép belső programjának.

Mire valók ezek az adatok? Nehéz lenne felsorolni, mi mindent szólnak ide össze a számítógépgyárakban, de azért nézzünk néhányat:

Itt tárolják a számítógép „karakterkészletét”, vagyis azokat a számokat, betűket, jeleket, amit a számítógép működése során felhasznál és mi a képernyőn látunk. Természetesen nem betűket, csillagokat vagy zárójeleket írnak a programtárba, hanem ezek kettes számrendszerbeli megfelelőjét. A gép ugyanis csak ilyen számokkal tud dolgozni, ezért a betűket, számokat (10-es számrendszerűeket), írásjeleket, műveleti jeleket egy nemzetközileg elfogadott rend szerint kettes számrendszerű számokká alakítják, és ebben a formában helyezik el a tárban. Ezzel a problémával egy későbbi fejezetben még részletesebben is foglalkozunk.

Itt tárolják az úgynevezett „fordító programot”. A gép — mint tudjuk — mindig csak számokkal, kettes számrendszerű számokkal dolgozik. Még a legegyszerűbb feladatot is nehéz lenne ilyen formában közölni a géppel, ezért elhelyeznek benne egy fordító programot, ami lefordítja a gép nyelvére

kívánságainkat. Az „A” betű például a fordító program segítségével azonnal számmá alakul, ha megérintjük a megfelelő billentyűt, és ezt a számot tárolja a már megismert adattár mindaddig, míg szükségesnek találjuk.

A PRIMO 120 féle betűt, számot, jelet, összefoglaló nevén karaktert ismer.

Vannak egyszerűbb és bonyolultabb fordító programok. Minél bonyolultabb, többet tud a fordító program, annál könnyebb dolgoznunk a számítógéppel. A PRIMO fordító programja a gép kettes számrendszerű gépi nyelvét úgynevezett BASIC (bázis) nyelvre fordítja le, ami tulajdonképpen angol szavak rövidítéséből áll. Ha tehát a géppel dolgozni akarunk, meg kell ismernünk a BASIC programnyelvet.

Miért nem tud egy magyar számítógép magyarul?

Erre a kérdésre és a BASIC programnyelv alapjaira egy későbbi részben még visszatérünk.

Itt látható a . . .

Azt már tudjuk, hogy a számítógéppel a bemeneti egységen keresztül tudjuk kívánságainkat közölni. De ha van bemeneti egység, akkor kell kimeneti egységnek is lenni, hiszen semmi értelme sem lenne az olyan gépnek, amely tudomásul veszi ugyan a kívánságainkat, el is végzi a feladatokat, de az eredményt megtartja magának.

Természetesen van kimeneti egység a számítógépeken, mégpedig elég változatos megoldásokkal.

A legegyszerűbb gépeknél az eredményt eredeti, kettes számrendszerbeli formában adja vissza a gép egy számkijelzőn, hasonlóan a zsebszámológépekhez. Ez nem valami szerencsés, nem is igen használják. Egy lépéssel jobb, ha ugyanez az eredmény 16-os számrendszerbe átvirva jelenik meg, hiszen így könnyebben kezelhető. (A 16-os vagy hexadecimális számrendszer 0–9-ig azonos a tízes számrendszerrel, a 10 és 15 közötti számokat ABCDEF betűkkel jelöli.)

Még jobb, ha az eredményt a fordítóprogram átfordítja egy magasabb szintű, jobban érthető nyelvre, például a BASIC-ra. Ez az eljárás terjedt el általában a személyi számítógépeknél, így a PRIMO is ezt alkalmazza. Az eredmény a tv-képernyőn jelenik meg, így kapcsolatunk a géppel mindkét irányban biztosított.

Természetesen az eredményt néha rögzíteni is akarjuk, hogy később ismét elővehessük és dolgozhassunk vele. A rögzítés történhet nyomtatással (papírra) vagy a hangfelvételekhez hasonlóan magnetofonra. A PRIMO elvileg mindkét megoldásra képes, gyakorlatilag azonban csak a magneto-

fonfelvételre van lehetőségünk, mert a megfelelő nyomtatógép még elég költséges. Egyszerű kazettás magnetofonnal viszont könnyen rögzíthetjük programjainkat.

A bemeneti és kimeneti egységeket — a billentyűzetet, a képernyőt, a felvevő és lejátszó magnetofont, az esetleges nyomtatót vagy másnevezetű rögzítőt — a számítógép perifériáinak nevezzük. Egyes számítógépeknél ezek az egységek teljesen össze vannak építve, míg másoknál — főként az egyszerűbb gépeknél — olyan csatlakozási helyeket alakítanak csak ki a gépen, hogy a kereskedelemben kapható, más célra is szolgáló eszköz (pl. tv, magnetofon) egyszerűen csatlakoztatható legyen. A drága monitort (képernyőt) például jól helyettesítheti egy közönséges televíziós készülék; adatrögzítőként használhatjuk akár hordozható magnós rádiókat is.

Kimeneti egységként nyilvánvalóan a televízió készülék nélkülözhetetlen.

Milyen tv-t használjunk?

A PRIMO csak fekete-fehér képet ad, tehát színes tv-t használni felesleges. A nagyképernyős tv-n nagyobbak ugyan a betűk, de ha nem akarjuk a szemünket rövid időn belül tönkretenni, távolabbról kell néznünk.

Ekkora hely azonban (2–3 méter) általában nincs, ezért ajánljuk a kisképernyős készülékeket. A VIDEOTON TC 1612 típusú készüléke ideális, igen jó, nyugodt, tiszta képe van és torzítása sem számottevő.

De ugyanilyen jól használhatjuk a szovjet gyártmányú Junoszty televíziókat is, ezek olcsóbban beszerezhetők. Jó, ha ezeket a kisképernyős készülékeket is legalább 1 méter távolságból nézzük.

Számítsunk arra, hogy aki elkezd a számítógéppel az ismerkedést, az naponta egyre több-több időt tölt el a gép mellett. Nem mindegy tehát, milyen környezetben, hogyan dolgozik. Ügyeljünk a munkahely kialakításánál az egészségünk megvédésére. Ülünk megfelelően magas asztalhoz, ahol kényelmesen tudjuk a gépet kezelni és írni is, mert mindkettőre szükség lesz. A tv-készülék legyen szemmagasságban és elegendő távolságra. Kényelmes fotelban elsüllyedve, vagy a szőnyegen hasalva nem lehet komolyan dolgozni!

Végül, de nem utolsó sorban . . .

A számítógép részei közül utolsónak maradt a mikroprocesszor. Mit tud ez a parányi alkatrész, melyről a szakemberek éveken keresztül valóságos dicshimnuszokat zengtek? A mikroelektronika óriásának, a számítógépek lelkének nevezték és beláthatatlan jövőt jósoltak neki.

Valóban ilyen különleges alkatrészrel állunk szemben?

Túlzás nélkül mondhatjuk, hogy a mikroprocesszor a mikroelektronika remeke. A kis méretek sok ezer félvezetőt takarnak, de az igazi érdem nem az alkatrészek mennyiségében, hanem a processzor működésében rejlik.

A mikroprocesszor felelős a számítógép részeiből összeálló rendszer működéséért, ez diktálja a tempót a teljes működés során valamennyi részegységnek.

A számítógép bekapcsolása után a mikroprocesszor végigellenőrzi a felügyeletére bízott egységeket és ha mindent rendben talál, üzenetet küld kezelőjének, kezdhetjük a munkát. Mindezt mi a gyakorlatban úgy érzékeljük, hogy bekapcsolás után a képernyőn megjelenik a felirat:

PRIMO BASIC SYSTEM

OK

Miért éppen ezt írja ki?

Azért, mert a programtár elejére ezt írták be a gyárban. A processzor, miután rendben talált mindent, elkezd a program végrehajtását. Az első programutasítás pedig az, hogy írja ki a képernyőre a fenti sorokat.

Ezután a gép várakozik. Most mi következünk kívánságaink előadásával. Közben azt hihetnénk, a gép áll és mindaddig nem indul, míg hozzánk nem érünk. Valójában ez nem így van. A mikroprocesszor fáradhatatlanul figyel és kérdeztetik a bemeneti egységet — esetünkben a billentyűzetet —, vajon megérintettük-e valamelyiket. Másodpercenként több százszor végigfut minden billentyűn és minden eredményt értékel. Ha nincs változás, az „OK” felirat változatlan marad. Alatta, a következő sor elején kis fekvő vonal villog. Ez a cursor (kurzor), és azt jelenti, hogy az első betűnk ide fogja írni.

Érintsük meg az A billentyűjét! A cursor helyén megjelenik az A betű és a villogó vonal egy hellyel jobbra lép.

Mit csinált ezalatt a mikroprocesszor?

Dolgozott, mégpedig igen sokat. Lássuk csak szépen sorjában. A bemeneti mű jelezte, hogy változás történt, az A betűt megérintették. A processzor nagyon gyorsan végigfut a billentyűkön és megállapítja, melyiket érintettük meg. Ezután fűtlyel jelzi, tudomásul vette akaratunkat. A processzor csak számokkal tud dolgozni, ezért az A betűt a programtárban rögzített szabálynak megfelelően átalakítja számsorrá. Megkeresi az adattár legelső szabad helyét, és oda beírja ezt a számsort. Ezután megnézi, hol van a képernyőn a következő hely. A számsorból a programtárban gyárilag rögzített előírások szerint összeállítja az A betű képét és kiírja a képernyőn oda, ahol azt a cursor eddig jelezte.

Bonyolult, de logikus. A tényleges működés még bonyolultabb, hiszen itt csak rövidítve, a lényegét mutattuk meg. A processzor tehát dolgozik a programtárral, ebből olvassa ki, mi a következő lépésben a teendője. De dolgozik az adattárral is, hiszen ide gyűjti az adatokat. Foglalkoztatja a kimeneti egységet és figyeli a bemeneti egységet.

A sebessége óriási. Általában másodpercenként 1–3 millió elemi lépést tud tenni. Ezért tűnik úgy, hogy a billentyű megérintésével egyidőben már meg is jelent a betű képe a tv képernyőjén.

Bonyolultabb a helyzet, ha valamilyen számítást kérünk a géptől. Ilyenkor begyűjti az adatokat, majd ugyancsak a programtár útmutatása alapján sorban elvégzi a műveleteket. Végül az eredményt kijelzi. A számítás során elvégzett lépések számára következtethetünk a számítás idejéből. Már egyszerűbb esetben is több száz vagy ezer apró lépésre bontja a feladatot. Összetettebb számításnál esetleg már az eredményt is később írja ki, különösen a lassabban dolgozó számítógépek. A processzorba beépítve saját tárolókat is találunk, és itt helyezik el az úgynevezett aritmetikai–logikai egységet, amely a számítások alaplépéseit vezérli. A számítások során ezeket a belső, kis kapacitású tárolóit is használja, ezekben dönti el két számról, hogy kisebbek-e, nagyobbak-e vagy egyenlők-e egymással és itt végzi az elemi alpműveleteket is. A legtöbb processzor sem szorzást, sem osztást nem végez, ezeket visszavezeti az egyszerűbb összeadásra, kivonásra, illetve más műveletekre, például léptetésre. Ezek alaposabb megismeréséhez azonban már jól kell tudni kettes számrendszerben számolni.

A mikroprocesszor bemutatásával végére értünk az első ismerkedésnek. A főbb részek ismeretében nagy vonalakban már el tudjuk képzelni, hogyan működik a számítógép. Az alaposabb megismeréshez két irányban is teszünk egy-egy kis kitérőt.

Az első irány a kettes számrendszer. Az előző fejezetekben gyakran esett szó arról, hogy a modern számítógépek kettes számrendszerben dolgoznak. A digitális elven működők valóban ezt az egyszerű számrendszert használják, ezért vezet első utunk erre.

Második kitérőnk a BASIC nyelv irányába indul, mert az általunk ismert és használt legtöbb személyi számítógép ezen a nyelven tartja a kapcsolatot kezelőivel.

EGY KIS MATEMATIKA

Ha számrendszerekről beszélünk, általában a tízes számrendszerre gondolunk. A gyakorlatban szinte minden számításunkat ebben a rendszerben végezzük el, megszoktuk és természetesnek találjuk, hogy tízféle számjeggyel minden számot le tudunk írni. Ismerjük a helyiértéket és – ki jobban, ki rosszabbul – műveleteket is tudunk végezni a tízes számrendszerben.



Tudjuk, hogy a helyiértékek jobbról balra növekednek és mindig éppen tízszeres értékkel. A helyiértékek nevei is ennek megfelelően: egyesek, tízesek, százaskok, ezresek . . . stb. Az 1985 tehát 5 egyesből, 8 tízesből, 9 századból, 1 ezresből áll. Mindezt valamennyien ismerjük. Tudjuk azt is, hogy ennek megfelelően más számrendszerek is léteznek és éppen elég gondot okoznak amikor például percek, másodpercek kell számlálnunk, esetleg összeadnunk vagy kivonnunk. Az időmérés sajnos nem illeszkedik a tízes számrendszerhez és még hosszú ideig emlékeztetni fog bennünket arra, hogy nemcsak tízes számrendszer van a világon.

Mi a helyzet a számítógépeknél?

A mechanikusan, fogaskerekekkel, számkerekekkel, rudazatokkal működő számológépek is tízes számrendszerben dolgoztak. Megtehették, hiszen csak elhatározás kérdése volt, hogy egy számkeréken hány számjegyet helyeznek el, azaz hány egyenlő részre osztják a kört. Az elektronikus gépek-

nél viszont már nagyon nehéz lenne a tíz számjegynek tízféle értéket megfeleltetni, hiszen az elektromosság egyik fontos tulajdonsága, hogy vagy van áram vagy nincs.



Ez a két állapot biztosan megkülönböztethető, ezért az elektromos számoló- majd számítógépek erre a két állapotra épülő számrendszerben, a kettes számrendszerben dolgoznak.

A tízes számrendszerhez hasonlóan kettes számrendszerben is jobbról balra növekednek a helyiértékek, de nem tízszeresére, hanem csak kétszeresére, és a számok írásához is csak két számjegyet, a 0-t és az 1-t használjuk.

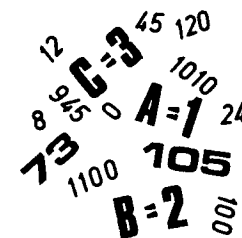
tízenhatosok	nyolcasok	négyesek	kettesek	egyesek
1	0	1	0	1

Az 10101 szám tehát tízes számrendszerbe átszámítva: egyszer tizenhat + nullaszer nyolc + egyszer négy + nullaszer kettő + egyszer egy. Ez összesen 21. Már is látható a kettes számrendszer egyetlen hibája — helyesebben hátránya —: sokkal több számjeggyel fejezi ki a számokat, mint a tízes számrendszer. Még jobban látszik ez a különbség például a 95 leírásakor. Tízes számrendszerben két számjegy, kettes számrendszerben pedig — 101111 — hét számjegy kell a leírásához. Ezt a hátrányt a számítógépek könnyen vállalják, hiszen számolási sebességük olyan nagy, hogy hosszabb számok sem okoznak lényeges idővesztést.

Adjunk össze két számot a kettes számrendszerben:

$$\begin{array}{r} 10010 \\ 1011 \\ \hline 11101 \end{array}$$

Tízes számrendszerben ez: $18 + 11 = 29$.



A kivonás hasonlóan történik. A szorzást és az osztást illetve a magasabb fokú műveleteket összeadásra és kivonásra vezetik vissza. Biztosan gyorsabb lenne a számolás, ha a gépek tudnának közvetlenül szorozni vagy osztani, de így egyszerűbb és biztonságosabb módon végzik a műveleteket, igaz több lépésben, de még így is nagyon gyorsan. Ha a feladat megoldása lehetővé teszi, azért néhány egyszerű „trükköt” még kihasználunk a számítógépek. Például ha egy számot kettővel kell megszorozni, egyszerűen egy értékkel balra léptetik a helyiértéktáblázatban és máris kész a szorzás. Próbáljátok ki!

Mindenhol csak számok

A kettes számrendszer tehát elfogadott, a gépek ezt ismerik. De hogyan lesz a betűkből vagy írásjelekből szám? Elég egyszerűen. Ha felírjuk az ABC betűt sorban és megszámozzuk őket, máris előtünk áll egy átalakítási lehetőség. Tehát az A = 1, B = 2, C = 3, D = 4 . . . stb. Természetesen a számo-

kat átírjuk a kettes számrendszerbe és már akár dolgozhatunk is vele. Vagy gondoljunk csak a morse-jelekre. Pontokból és vonalakkól állnak. Ha a pontnak 0-t, a vonásnak 1-et feleltetünk meg, azonnal kettes számrendszerbeli alakot kapunk. Az A betű morse-jele . - , tehát 01, a B betűé - . . , tehát 100, a C betűé - . - . , tehát 1010 és így tovább.

Mindkét bemutatott lehetőségnek van egy elég nagy hibája: a különböző betűknek vagy jeleknek különböző hosszúságú számsor felel meg.

Ez nagyon megnehezítené a gép munkáját, hiszen nem tudná pontosan megállapítani, mikor ér véget egy betűnek megfelelő számsor, ezért hosszabb szüneteket kellene tartani a jelek között, ami lassítaná a műveleteket.

Megoldást az egységesített, azonos hosszúságú számsorokból álló átalakítás ad. A számítógépek egységesen minden betűt, számot, írásjelet, műveleti jelet nyolc számjegy hosszúságú kettes számrendszerű számban fogadnak el. Ezt az átalakítási kódot ASCII kódnak nevezzük. A nyolc számjegy hosszúságú jelsorozat 256 féle jelnek felelhet meg, tehát bőven elférnek benne a számok, a kis- és nagybetűk és az egyéb jelek is.

Még egy számrendszer

A számítógépek tehát a nyolc számjegyű számokat szeretik – természetesen kettes számrendszerben. Nekünk viszont elég nehéz kezelni, leírni, kiolvasni ezeket a hosszú számokat.

Hogyan könnyíthetjük meg a dolgunkat?

A számrendszerekhez kell fordulni segítségért. A nyolc számjegy két 4–4 jegyű csoportra osztható. Négy számjeggyel kettes számrendszerben 0-tól 15-ig minden számot le tudunk írni. Ha lenne 16 féle különböző jelünk, máris csak egy jelet kellene írni a négy számjegy helyett. Aki egy kicsit még emlékszik a számrendszerekről írottakra, máris felismeri, hogy 16-os számrendszerben szeretnénk dolgozni.

De ehhez tényleg 16 különféle jel kell. A megoldást nem volt nehéz megtalálni. 0-tól 9-ig használjuk a tízes számrendszerben megszokott számjegyeinket, utána pedig az ABC első betűit:

Így:

10 -- A
11 -- B
12 -- C
13 -- D
14 -- E
15 -- F

Ezzel együtt áll a 16 különféle jel, azaz a tizenhatos számrendszer. Az így felírt számokat hexadecimális számoknak nevezzük, és megkülönböztetésül a számok után egy H betűt írunk: BH 11-et jelent.

Két hexadecimális számjeggyel pedig $16 \times 16 = 256$ -ig írhatunk fel számokat. A számítógéphez szükséges, 8 jegyű kettes számrendszerbeli számot tehát egyszerűen két számjeggyel is le tudjuk írni:

tizenhatosok	egyesek
2	F

Például 2 F, azaz 2 FH azt jelenti, hogy a szám hexadecimális formában van írva, és F-szer egyet + 2-szer 16-ot ér: $F = 15$, tehát $2 \times 16 + 15 = 47$.

A 47-nek, azaz a 2 F-nek az ASCII kód szerint éppen a törtvonal felel meg, ami a számítógépeknél egyben az osztás jele is. A számítógép-programozók általában ezt a hexadecimális alakot szeretik és úgy alakítják ki a gép belső programtárolóját, hogy segítségével a mikroprocesszor a beadott hexadecimális számot azonnal át tudja alakítani a további műveletek végzésére alkalmas kettes számrendszerbeli vagy más néven bináris számmá.

A PRIMO műszaki adatainál a mikroprocesszor után azt írták: 8 bites. Ez éppen azt jelenti, hogy nyolc számjegy hosszúságú kettes számrendszerű számmal tud dolgozni. Nem véletlen ez az egybeesés. A mai személyi számítógépek ilyen hosszúságú egységekkel dolgoznak.

Végül egy táblázat

Az ASCII kódokkal a számítógéppel foglalkozók gyakran találkoznak. Ismerkedjünk meg mi is ezzel a kódrendszerrel!

Természetesen ez az ismerkedés nem azt jelenti, hogy egész táblázatot meg kell tanulni. Sőt! A táblázatok azért vannak, hogy használjuk őket. Kár volna ilyen nagy adathalmazal fárasztani magunkat. Azt viszont tudni kell, hogy egy-egy fontos adat hol található meg. A táblázatok helye – különösen ha azok nagyobb terjedelműek – a szakkönyvek végiglapozni. Ez a játékkönyv ugyan nem szakkönyvnek készült, de azért jobbnak tartottuk, ha a többi melléklettel együtt az ASCII kódok táblázata is az utolsó oldalakra kerül. A táblázat első oszlopában a bináris kódokat találjuk. Ezek a kettes számrendszerben kifejezett számok. A következő oszlopban a decimális kódokat, ezek a tízes számrendszerben kifejezett számok. A harmadik oszlop a hexadecimális kódokat tartalmazza, ezek a tizenhatos számrendszerbeli alakok. Végül a negyedik oszlop azokat a karaktereket mutatja, melyeket a számítógépek általában a képernyőre kiírnak. Nézzük meg a táblázatot a 78. oldalon!

Aki figyelmesen olvasta a kettes számrendszerről írottakat, tudja, hogy a nyolc számjegyből 256 különböző számjegy írható fel. Itt viszont csak ennek a fele, 128 szerepel. Miért? Egyszerűen azért, mert a 128 féle karakter már éppen elegendő a nagy- és kisbetűk mellett a számok, a műveleti és egyéb jelek megjelenítésére. Ez igaz is az angol ABC szerint. De a PRIMO magyar gép, magyar ékezetes betűket is ismer, tehát az eredeti kódoktól néhány helyen el kellett térni. Az eltérés a PRIMO gép CHRŠ kódtáblázatból a 77. oldalon olvasható.

A számítógépeken általában nem találunk 128 féle billentyűt, egy billentyű több féle karaktert is képes leírni. A PRIMO-nál is minden billentyű két jelentésű, sőt a gép még olyan jeleket is le tud írni, amik nem is szerepelnek a billentyűin. Erről bővebben majd a programozásnál tudhattok meg.

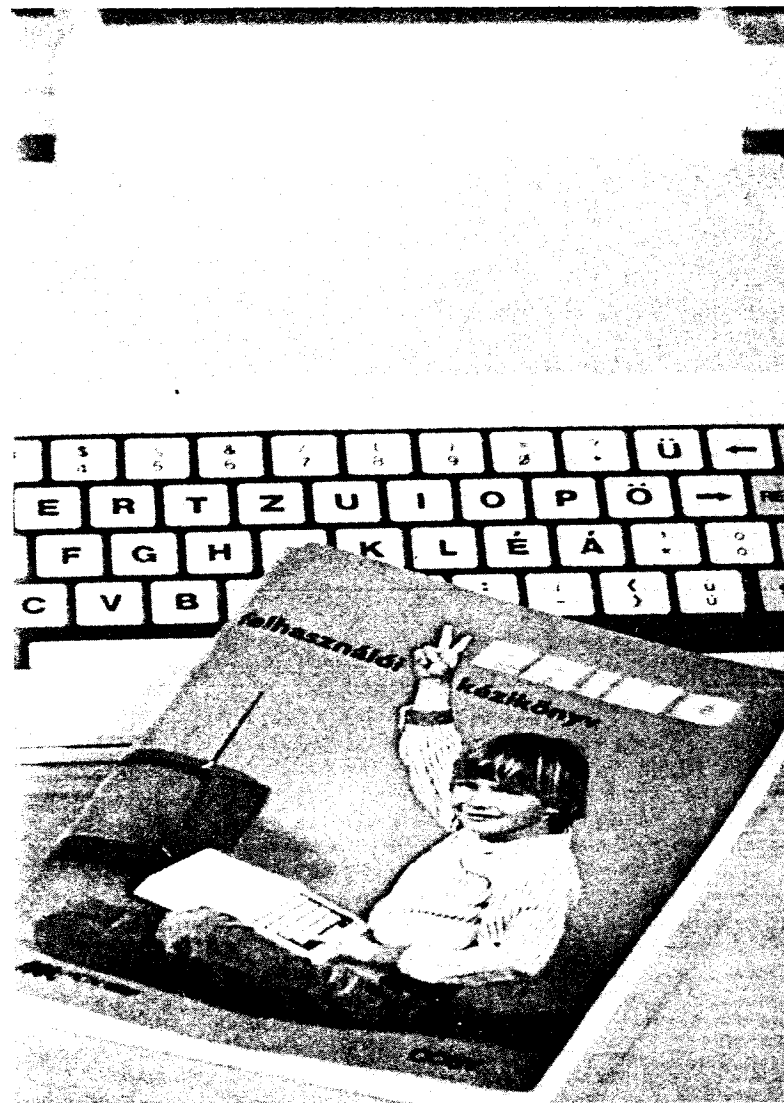
Tanuljunk „bézikül”

A számítógép a logika nyelvén ért. A gyakorlott szakemberek olyan jól ismerik már a gépeket, hogy ők közvetlenül számokban fogalmazzák meg a feladatokat, amiket a géppel elvégeztetni szeretnének. Ez az eljárás azonban nagyon bonyolult és megtanulása egy speciális szakma elsajátításával egyenlő. Nyilvánvaló, hogy ez az út csak kevesek számára járható. A másik megoldás a gép „megtanítása” egy általunk is könnyen megtanulható írott nyelvre.



Miért nem egyből az élő beszéd megértésére?

Ennek is eljön az ideje! Nem is olyan soká kell már várni az olyan számítógép megjelenésére, amely megérti beszédünket és választ sem olvasni, hanem hallgatni kell. Addig azonban tanuljunk meg egy programnyelvet, lehetőleg azt, melyet a mi gépeink, így a PRIMO is „megértenek”.



5. kép

Ez a nyelv a BASIC. Kifejezetten kezdők számára készítették, tehát azoknak, akik szinte semmit sem tudnak a számítógépről, de szeretnék használni. Könnyen és gyorsan megtanulható, különösen, ha ott a számítógép is. Ha nincs gép, a tanulás is nehezebb, mert a gép segíti a tanulást! Ez első hallásra különösnek tűnik, de amikor elkezditek a programírást, a gyakorlatban is tapasztalhatjátok, hogy a gép milyen gyakran küld „hiba-üzenetet”, azaz figyelmeztet az elkövetett hibára.

A tanuláshoz tehát számítógép mellett kezdjétek!

Egy kis technika

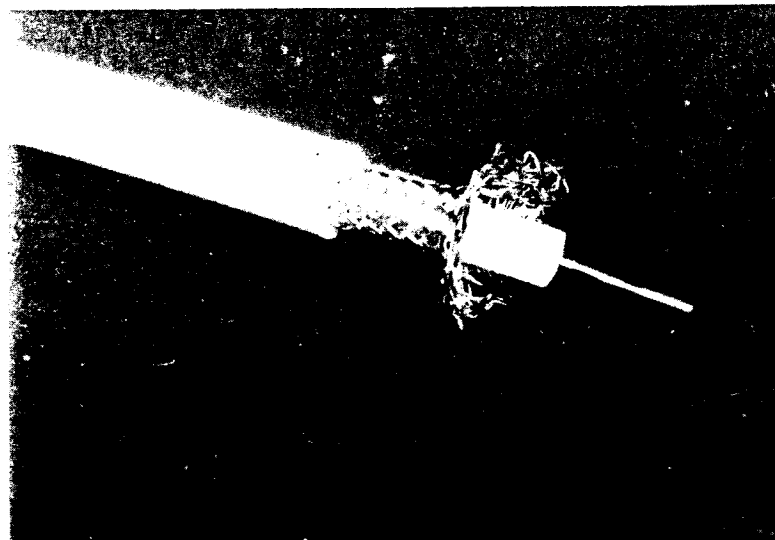
Vegyük elő a PRIMO-t és kapcsoljuk be. Ez így leírva elég egyszerűnek tűnik, a gyakorlatban sem sokkal bonyolultabb. A PRIMO-hoz a gyártók jól használható felhasználói kézikönyvet adnak, ebben mindazt megtaláljuk, ami a beinduláshoz szükséges.

Amire feltétlenül ügyeljünk:

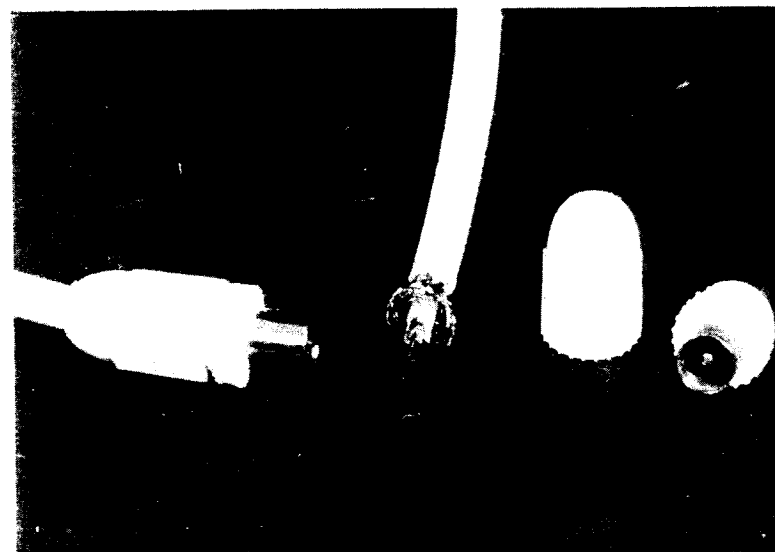
- A számítógéphez televíziót is csatlakoztatnotok kell. Mindkét készülék 220 voltos hálózati árammal működik és feltételezhetően egy elosztóba csatlakoztatjátok. Vigyázzatok az elosztó és a kábelek épiségére, mert törött elosztó vagy sérült szigetelésű kábel súlyos balesetet okozhat. Gondoljatok arra is, hogy a számítógép a kisebb gyerekeket is vonzza és ők még esetleg nem ismerik az áram hatását. Vigyázzatok rájuk is!
- Ha úgy érzitek, hogy akár a tv, akár a számítógép „ráz”, azonnal kapcsoljátok ki mindkét készüléket és szóljatok szakembereknek. Ezek a szabályok minden elektromos készülékre vonatkoznak.
- A számítógépet és a tv-t összekapcsoló kábel elég rövid és nagyon kemény, könnyen megfeszül. Célszerű kicserélni egy hosszabb és hajlékonyabb kábelre. Ilyen kábelt magatok is készíthettek, de jobb, ha szüleitek, tanáraitok vagy MHSZ rádiós szakkörbe járó barátaitok segítségét kéritek az elkészítéshez. Fontos! Csak koaxiális kábelt szabad használni, amit arról lehet felismerni (a neve is ezt jelzi), hogy van egy belső vezetéke, amit szigetelő anyag vesz körül, majd a másik vezeték erre a szigetelőre szövik rá. Ugy néz ki, mintha harisnyát húztak volna a belső vezeték köré. A „harisnyán” kívül újabb szigetelő réteg van. (6. kép)

A kábelvégek megtisztításához és a koaxiális dugó szét- és összeszereléséhez kis ügyesség is kell. A 7. kép segítséget ad azoknak, akik maguk vállalkoznak az elkészítésére.

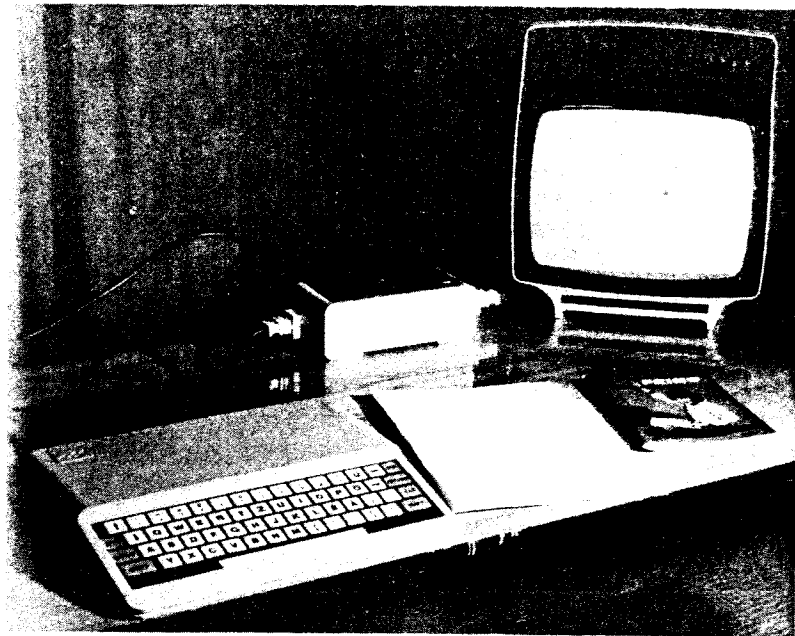
Ezek után már valóban be is kapcsolhatjuk számítógépünket és a televíziót. A számítógép hangját magától a géptől halljuk, a tv hangját ezért halkítsuk le, hogy ne zavarjon a zúgása. A PRIMO számítógép két részből áll: magából a számítógépből és az úgynevezett tápegységből. (8. kép)



6. kép



7. kép



8. kép

A tápegység alakítja át a hálózati áramot olyan értékűre, melyre a gépnek szüksége van. Itt találjuk a KI-BE kapcsolót is. Ha a kapcsolót benyomjuk, világítani fog és ezzel jelzi: a számítógép be van kapcsolva. A számítógép „képét” a televízió a 36-os csatorna közelében találjuk meg.

Ha a készülékeket jól csatlakoztattuk, a televízió megjelenik a számítógép bemutatkozó szövege: PRIMO BASIC SYSTEM.

De mit csináljunk, ha mégsem történik ez, azaz minden igyekezetünk ellenére számítógépes „rendszerünk” nem akar működni?

Semmiképpen se ütögessük, kopogtassuk a készüléket vagy feszítgessük a csatlakozókat, mert ezzel esetleg újabb hibákat okozunk. A PRIMO felhasználói kézikönyvében találunk hibakereső táblázatot, ez alapján keressük meg a hibát.

Ismerkedünk a billentyűkkel

Számítógépünk tehát működik, kezdjük a munkát! A billentyűzetről már volt szó, tudjuk, hogy nem kell nyomni, csak megérinteni. Érintsük meg valamelyik betű billentyűjét. Máris halljuk a gép jelzését és látjuk a képernyőn a megjelenő betűt. Ha folyamatosan rajta hagyjuk ujjunkat a billentyűn, a gép megismétli a betűt, mégpedig elég gyorsan és mindaddig, míg fel nem emeljük az ujjunkat. Kezdetben csak egy-egy betűt írunk, a gyors ismétlésre ráérünk később, amikor már nagyobb gyakorlatunk lesz a kezelésben. Ha a betűt eltévesztettük, vagy más ok miatt javítani akarunk, nyomjuk meg a balra mutató nyilat: erre ugyanaz érvényes, mint a betűkre, egy érintésre egyet töröl visszafelé, de ismétlésre is képes, ha tovább nyomjuk. A nagy- és kisbetűk között az UPPER feliratú billentyűvel választhatunk. Nem kell folyamatosan nyomni, mint az írógépeknél, ha nagybetűt akarunk írni, elég egy érintés és a továbbiakban már nagybetűket ír. Újabb érintésre visszatér a kisbetűs írásra.

Vannak olyan billentyűk, melyeken egymás fölött két jel is található. Ezek közül az alsót írja a gép, ha megérintjük a billentyűt, a felső jelet úgy tudjuk leírni, hogy megérintjük a SHIFT feliratú billentyűt és vele egyidőben a kívánt jelet tartalmazó billentyűt. Most tehát egyszerre két billentyűt kell megérinteni!

Ha az egész képernyőt le akarjuk törölni, érintsük meg a CLS billentyűt. A betűk közti szüneteket az alsó hosszú billentyűvel írhatjuk.

Először számolunk

A számítógépek számológépként is működnek. Nagyon bonyolult számításokat végezhetünk el gyorsan és egyszerűen.

Kezdjük a munkához!

Töröljük le a képernyőt (CLS) és írjuk be: $2 + 3 =$ (= jelnél nyomni kell a SHIFT-et is!).

Zsebszámológépeink erre azonnal kiírják az eredményt, hiszen azok csak erre a célra, számolásra készülnek, a PRIMO azonban nem, mert nem „tudja”, hogy most csak egyszerűen számolni akarunk.

Akaratunkat egy *parancs* segítségével juttatjuk el a géphez. A Basic nyelvet ismerő gépeknél ez a parancs a PRINT. Kezdjük újra az írást (CLS) és most először betűnként a PRINT szót, majd közvetlenül utána a feladatot írjuk bel

PRINT 2 + 3

Így már felismeri a gép, hogy számolni akarunk, és már csak azt várja, mikor jelezzük a feladat felírásának befejezését.

Ezt a jelzést a RETURN billentyű megérintésével adjuk. Ekkor azonnal elvégzi a gép az addig leírt feladatot és a következő sorba kiírja az eredményt.

PRINT 2 + 3

5

OK

Vigyázzunk, a gép most csak a számokat és a műveleti jeleket várja tőlünk, tehát = jelet ne írjunk!

PRINT 2 + 3 = felírásra a RETURN billentyű érintése után MO Error hibaüzenetet kapunk eredmény helyett.

Miért?

Mert a gép az egyenlőség jel után még vár egy számot, a számítás eredményét. Ha beírjuk az általunk jónak vélt eredményt, a gép „kiértékeli” számításunkat és ha helyes az eredmény 1-et, ha helytelen, 0-t ír ki.

Próbáljuk ki ezt is!

PRINT 3 + 2 = 5

—1

OK

Tehát jól számoltunk, az eredményt helyesnek tartja a gép.

PRINT 3 + 2 = 4

∅

OK

Az eredmény nem jó. Az OK felirat azt jelenti, hogy a feladat megoldása befejeződött.

Máris látjuk, hogy mennyivel többet tud ez a gép, mint az egyszerű számológép!

Számoljunk tovább!

A kivonás hasonlóan történik, mint az összeadás.

PRINT 12—8

4

A gép természetesen ismeri a negatív számokat is, tehát a 8—12 kivonást is elvégzi

PRINT 8—12

—4

A feladatok előtt a géppel mindig közölni kell a PRINT parancsot, mert csak ezt követően fog számolni. Egyes gépeknél azonban — a gyorsabb kezelés érdekében — a PRINT szó helyettesíthető a kérdőjellel. Tehát elég csak egy kérdő jelet (?) beírni (SHIFT billentyűre ügyeljünk!) és máris írhatjuk a feladatot.

? 12 + 8

20

Így egyszerűbb a beírás. Előfordul azonban, hogy ? helyett + jelet ír a gép, mert nem tartottuk érintve a SHIFT billentyűt. Ilyenkor a ← érintésével javítjuk a beírást.

A szorzás jele a *

? 5 * 8

40

Az osztás jele a törtvonal

? 12/3

4

A törtvonal beírásakor is ügyeljünk a SHIFT billentyű kezelésére. Nullával a gép sem tud osztani, csakúgy mint mi. Ha mégis erre kényszerítitek, hibaüzenettel figyelmeztet a képtelenségre.

A hatványozást — mint műveletet — a gép ismeri, de ti ismeritek-e már?

2 · 2 · 2 = 2³

A gép a hatványozást felfelé mutató nyíl hatására végzi.

? 2 ↑ 3

8

Aki ismeri a hatványozást, az lényegében a gyökvonást is ismeri, hiszen a gyökvonás azonos egy törtkitevőjű hatványozással.

$\sqrt[3]{8} = 8^{\frac{1}{3}}$

A törteket mindig tegyük zárójelbe! Zárójelnek csak a kerek zárójelet használhatjuk.

A zárójeleket a 8 és 9 számok felett találjuk, az alsó sorban levő két jel a kisebb—nagyobb jel!

? 8 ↑ (1/3)

2

Ezt már bonyolultabb leírni, ezért figyeljük jobban a képernyőt. Összetettebb számításoknál is használjunk bátran zárójeleket, mert a gép egyébként a műveleteket a következő sorrendben végzi el:

— először elvégzi a hatványozást,

— majd a szorzást vagy osztást,

— végül az összeadást vagy kivonást.

A rosszul felírt feladatot a gép csak a rossz felírásnak megfelelően tudja elvégezni! Lássunk erre is egy példát.

? 48/4 + 2

14

Tehát először az osztást végezte el, majd az eredményhez hozzáadta a 2-t.

Írjuk fel a (4 + 2)-t zárójelben

? 48/(4 + 2)

8

Így az osztást már a zárójelben levő összeggel végzi.

Természetesen a gép nemcsak egész számokkal tud számolni. A tizedes törtek felírásánál *tizedesvessző helyett* a gép a *pontot* fogadja el.

Az összeadás ismertetésekor bemutattuk, hogy a gép képes egy megoldott feladat helyességét is ellenőrizni. Ott csak az = jelet használtuk.

A gép azonban el tudja dönteni két számról azt is, hogy melyik a kisebb vagy nagyobb.

? 3 < 4

—1

A -1 az eredmény helyességét jelenti, azaz a 3 valóban kisebb mint 4.
? 3 > 4

0

Tehát az eredmény nem helyes.

Két érdekességgel zárjuk a számolásról szóló részt.

Ha kör kerületét, vagy területét számoljuk, a π (pi) értéke helyett magát a két petűt (PI) kell beírni és a gép így is pontosan számol.

A 3 egység sugarú kör kerülete tehát:

? 2 * 3 * PI

18.8496

A négyzetgyökvonást pedig a SQR parancsra is elvégzi, mint a legtöbb BASIC nyelvű gép.

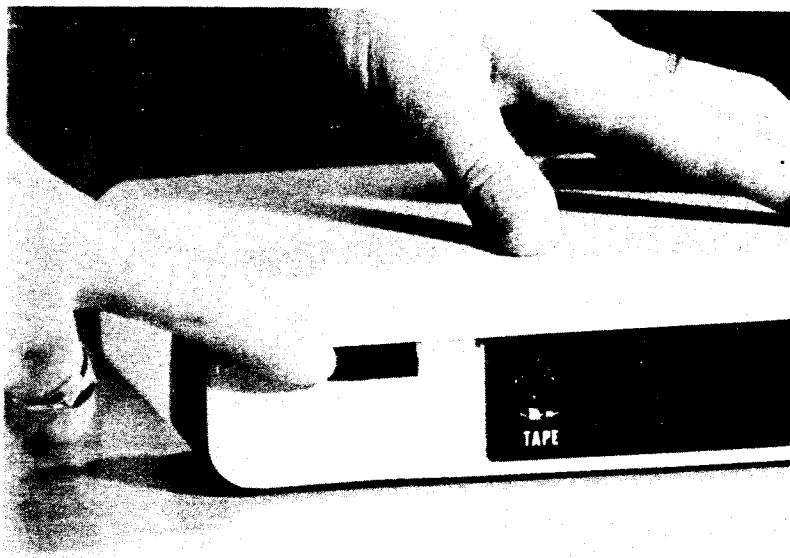
? SQR (25)

5

Itt a zárójelről ne feledkezzünk meg. Negatív számból természetesen nem fog négyzetgyököt vonni a gép sem.

Ezek után már csak szorgalom és türelem kell ahhoz, hogy megtanuljatok a PRIMO-val jól és gyorsan számolni.

Szólni kell még a PRIMO néhány jó tulajdonságáról. Ha nagyon kicsi képernyős tv-vel dolgoztok vagy a kép más okból nem látható jól, a betűk méreteit egyszerű módon megnövelhetitek és ezáltal jobban olvasható,



9. kép

nagyobb betűket kaptok. Érintsetek a CTR és a B billentyűket egyszerre. A felhangzó fűtty és a hosszabbra nyúlt villogó cursor már a nyújtott karakterek írását jelenti.

Érdekes hatást érünk el, ha ezután a CTR és a D billentyűket érintjük. A betűk most fehér mezőben feketén jelennek meg. A CTR és C érintésére az egész képernyő „negatívba” fordul át, tehát ami eddig fehér volt, az ezután fekete lesz. Visszafordítás a CTR és az S billentyűk egyidejű érintésével lehetséges.

A CTR és D visszaállítása a CTR és a T billentyűkkel, a CTR és a B (nyújtás) visszaállítása pedig a CTR és az R billentyűkkel történik. Alapállásba egyszerűbben is vissza lehet térni, csak a számítógép hátoldalán levő RESET gombot kell megnyomni. (9. kép)

Most már rajtatok a sor, hogy a megfelelő képet kiválasztva gyakoroljátok a számolást!

Mi is a program?

A számítógéppel különböző feladatokat oldhatunk meg. A feladatok apró lépések sorozatából állnak. Ezeket a lépéseket sorszámokkal látjuk el, a gép pedig a sorszámok növekvő sorrendjében elvégzi a műveleteket.

A BASIC program tehát sorokból áll és minden sornak van egy sorszáma.

Készítsünk egy egyszerű programot két tetszés szerinti szám összeadására!

Mit kell tennünk?

Először is közöljük a géppel az egyik számot, majd a másikat, ezután utasítjuk, hogy a két számot adja össze, végül pedig írja ki az eredményt. Rajzoljuk le ezt a tervet. (2. ábra)

A sorok számozását általában 10-zel kezdik és tízesével folytatják. Ez nem előírás, inkább a célszerűség kívánja meg, így később a program bővítésekor lehetőségünk van újabb sorok beírására.

10 INPUT A

Ez lesz az első sorunk. Az INPUT azt jelenti, adatot fogunk közölni a géppel. Az A jelenti majd az első összeadandót. Ide tulajdonképpen bármilyen más betűt is írhatnánk, tehát X-et vagy Y-t is.

A másik sor hasonló:

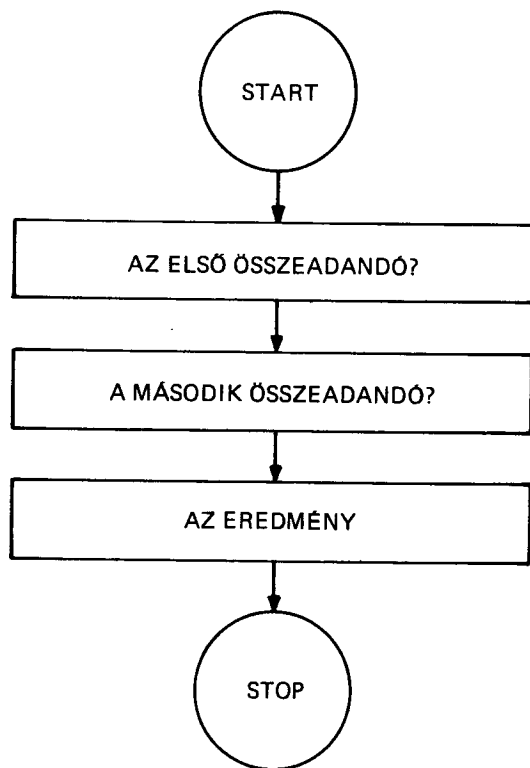
20 INPUT B

Itt B jelenti a második számot.

30 C = A + B

A harmadik sorban végezzük el az összeadást. Az összeg értéke lesz a C. Végül kiíratjuk az eredményt.

40 PRINT C



2. ábra

Ezzel összeadó programunk elkészült. Az INPUT és PRINT szavakat *utasításoknak* nevezzük. A 30-as sorban is van egy utasítás, de ezt a PRIMO gépen nem kell kiírni. (Egyes gépeken ezt a sort így kellene írni: 30 LET C = A + B.)

A BASIC programban a sorok mindig sorszámmal kezdődnek, ezt mindig utasítás követi.

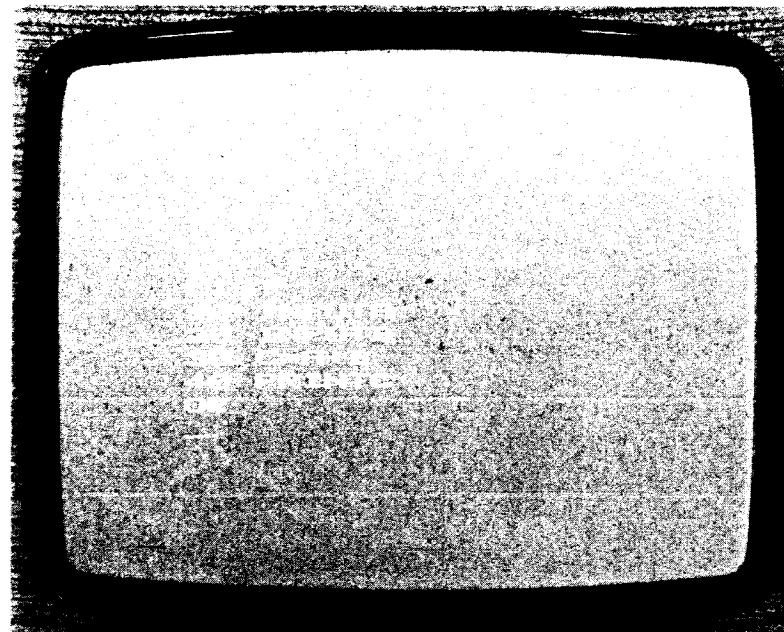
A sorszám nélküli utasítást *parancsnak* nevezzük, ezeket a gép azonnal végrehajtja. Ilyen parancs volt például a korábban megismert PRINT is.

De térjünk vissza első programunkhoz, mely lényegében készen várja a kipróbálást. A programokat a BASIC nyelvben RUN paranccsal indíthatjuk el, tehát a következő sorba – természetesen sorszám nélkül – írjuk be a RUN szót. A RETURN billentyű megnyomása után máris látunk egy kérdőjelet. Ezzel jelzi a gép, hogy adatot vár, azaz várja az első összeadandó értéket, amit a programunk 10. sorában A-val jelöltünk.

Írjunk be egy számot: 128 (ismét RETURN).

A gép megjegyzi a beírt számot. A képernyőre is kiírja és megjelenik egy újabb kérdőjel. Ez a következő adat kérését jelenti, amit a program 20. sorában B-vel jelöltünk. Írjunk ide 248-at (ismét RETURN).

A gép továbblép a programban, elvégzi a 30. sorban leírt összeadást, majd a 40. sornak megfelelően az eredményt kiírja: 376. A program ezzel végére ért. (10. kép)



10. kép

Újabb RUN paranccsal újból elindíthatjuk. Ekkor A és B értékét ismét tetszés szerint választhatjuk, mert a gép az előzően beírt adatokat „felülírja”.

Ez a program jól működik, de elég kényelmetlen, mert minden összeadás előtt újból el kell indítani a programot a RUN paranccsal. Ezen egyszerűen segíthetünk, ha bővítjük a programunkat újabb sorral és ebben arra utasítjuk a gépet, térjen vissza a program elejére, a 10. sorba:

50 GOTO 10

Ha most elindítjuk programunkat, minden elvégzett összeadás után várja az újabb számokat, és ez a folyamat a gép kikapcsolásáig folytatódna.

A program zárt hurok alakú lett. Az ilyen program futását természetesen meg lehet szakítani. Erre alkalmas a gép hátoldalán elhelyezett RESET kapcsoló, amelynek megnyomásával a mikroprocesszor alapállapotba kerül, de a memóriába beírt program változatlanul megmarad. (9. kép)

A program végrehajtását a BRK billentyű érintésével is megszakíthatjuk.

Ezek után valamelyik módon állítsuk meg a programunkat.

Írjuk be a LIST parancsot! Hatására a képernyőn megjelenik korábban beírt programunk sorról sorra.

Most tehát előttünk van az egyszerű program és ismét változtathatunk rajta.

Szüntessük meg az 50. sorban levő visszacsatoló utasítást, azaz töröljük az 50. sort.

A BASIC programnyelv tulajdonsága, hogy ha egy programban két vagy több azonos sorszámú sort talál, mindig az utolsónak beírtat tekintti érvényesnek. Írjunk tehát egy újabb 50-es sort.

```
50 BEEP 10, 100
```

A BEEP utasítására a gép hangjelzést ad, tehát programunkban az eredmény után halljuk a jelzést. Egy ilyen rövid programról azonnal látjuk, milyen célra készült. Nem így a hosszú és bonyolult programoknál. Célszerű ezért a program elejére címet írni:

```
5 PRINT „Összeadás” (RETURN)
```

A PRINT utasítás újabb alkalmazását láthatjuk.

Az idézőjelek közé írt karaktereket változatlanul kiírja a képernyőre ez az utasítás. Használatával áttekinthetőbb a program, könnyebben érthető mikor mit kell tennünk.

```
7 PRINT „Írd be az első számot!”
```

```
15 PRINT „Írd be a második számot!”
```

```
35 PRINT „A két szám összege!”
```

A 7. és 15. sor a két INPUT utasítás elé kerül, így már érthetőbbé válik, mit kell tennünk a kérdőjel megjelenésekor.

Programunk sorai most eléggé összekeveredtek, azonban ettől függetlenül a gép növekvő sorrendben hajtja végre az utasításokat.

Adjunk LIST parancsot! Már is növekvő sorrendben láthatjuk program-sorainkat. Szépítsük a programot!

A számolás kezdete előtt töröljük le a képernyőt CLS utasítással.

```
2 CLS
```

A legtöbb program első soraiban megtaláljuk a CLS utasítást, mert így a korábbi programrészek vagy eredmények nem zavarják, áttekinthetőbb a kép.

Ezzel első programunk eljutott a végleges formájáig.

Vagy mégsem?

A programokon – legyenek azok bármilyen egyszerűek – mindig találunk valami javítani valót. Nem kivétel ez alól mostani programunk sem!

Válasszuk el a címet az első adattól egy üres sorral.

```
6 PRINT
```

Ha a PRINT után semmit sem írunk, üres sort kapunk a képernyőn.

Az összeg elé is tegyünk egy üres sort!

```
32 PRINT
```

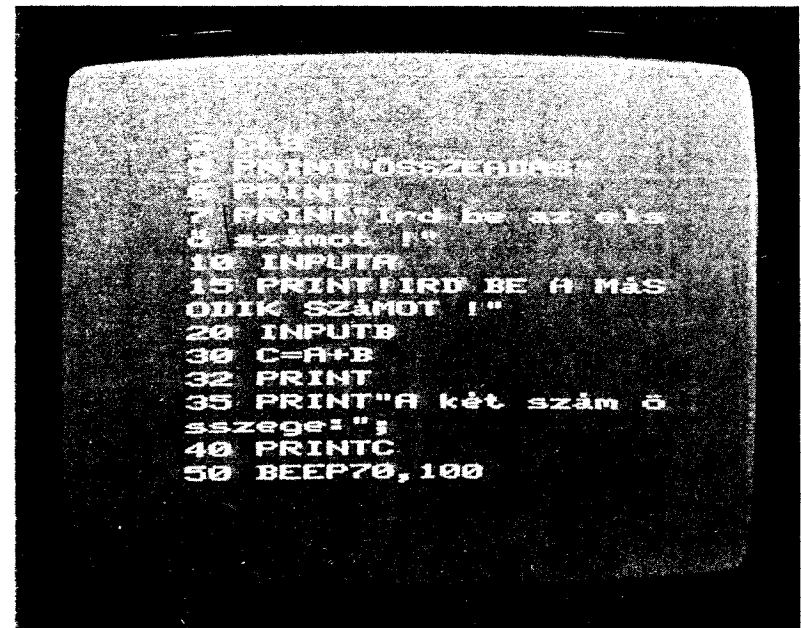
Újabb változtatás: írassuk az eredményt az „A két szám összege:” után, azaz a 35. sor utasításának végrehajtása után ne kezdjen a gép a 40. sor utasításának végrehajtásakor új sort, hanem folytassa a megkezdettet. Ehhez a 35. sort javítani kell.

Javításra az EDIT parancs szolgál. Írjuk utána a javítandó sor számát.
EDIT 35 (RETURN)

```
35 PRINT „A két szám összege:”
```

Tehát megjelenik a javítandó sor az EDIT parancsra.

A jobbra mutató nyíl segítségével karakterenként kiíratjuk újból a sort és elvégezhetjük rajta a szükséges javításokat. Most nem kell sokat javítanunk, mindössze a sor végére egy pontosvesszőt írni. Írjuk be! (Majd RETURN)



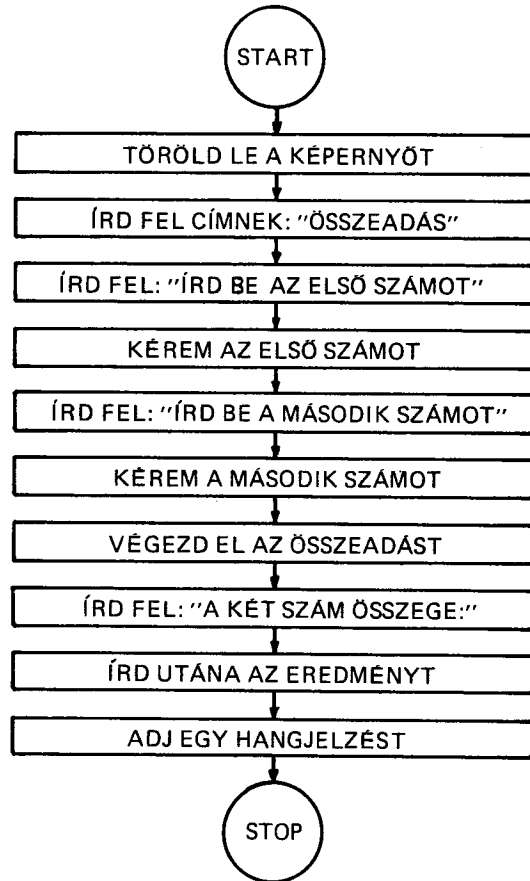
35 PRINT „A két szám összege:”;

A ; (pontosvessző) azt jelenti BASIC nyelven, hogy a következő karaktereket (ami nálunk éppen az eredmény lesz) az előzőek után folyamatosan írja gép. Lehetne természetesen még további változtatásokat is kipróbálni, most azonban elégedjünk meg ennyivel.

LIST paranccsal nézzük végig a programot. (11. kép)

Ha valamit nem értetek a programban, lapozzatok vissza oda, ahol elkezdttük a tervezését.

Rajzoljuk le összeadó programunk folyamatábráját! (3. ábra)



3. ábra

Már az egyszerűbb programok készítését is megkönnyíti, ha először megtervezzük azt a folyamatot, amit a gépnek majd el kell végeznie. Papíron könnyebb a javítás, módosítás és a folyamatábra alapján már könnyen elkészíthetjük végleges programunkat.

Ha összehasonlítjuk a feladat elején rajzolt folyamatábrát az utóbbival, láthatjuk, hogy a lényeges elemek nem változtak, a bővítés csak kényelmi szempontokat szolgált. A folyamatábra egyenesen, elágazások nélkül vezet a kezdettől a végéig.

Nagyon ritka az ilyen program. A következőkben egy bonyolultabb programot tervezünk, melyben újabb BASIC utasításokat ismerhetünk meg.

Írassuk ki a nevünket nyolcszor egymás alá a géppel! Nyilván nem ez lesz a megoldás

```
10 PRINT „NÉV”
20 PRINT „NÉV”
30 PRINT „NÉV”
40 PRINT „NÉV”
50 PRINT „NÉV”
60 PRINT „NÉV”
70 PRINT „NÉV”
80 PRINT „NÉV”
```

Azt már látjuk viszont, hogy mindig ugyanazt a nevet kell kiírni, tehát valamilyen módon ismételtelen ugyanazt az utasítást kell végrehajtatnunk a géppel és a végrehajtást számolnunk kell. Ha elértünk a nyolcadik kiírásig, a programot meg kell állítani.

Írjuk be az alábbi programot:

```
10 PRINT „KATI”
20 GOTO 10
```

Ha elindítjuk a programot (RUN) a képernyőn egymás alatt megjelennek a KATI feliratok, mert

- a 10. sor azt jelenti: írd ki a KATI szót,
- a 20. sor pedig azt jelenti: menj vissza a 10. sorra.

Így tehát a kiírás a végtelenségig folytatódik.

A GOTO utasítást feltétlen ugró utasításnak is szokták nevezni, mert azt jelenti: ugorj az adott számú (nálunk éppen a 10-es) utasításra és ott folytasd a programot.

A kialakult végtelen ciklus a BRK billentyűvel szakítható meg. Állítsuk meg a programot, mert nekünk csak nyolcszor kellene kiírni a nevet.

Számláljuk meg a kiírásokat! Jelölje A a kiírások számát. Kezdetben $A = 1$, mert ez lesz az első kiírás. Ezután növeljük meg A értékét eggyel. Ezt a BASIC-ban így írják $A = A + 1$. Szokatlan a forma, de ez most nem matematikai összefüggés, hanem A értékének növelése. Ezután vizsgáljuk meg A értékét: elérte-e már a nyolcat. Ha nem, küldjük vissza a programot a név kiírására, majd növeljük az A értékét újból eggyel és ismét vizsgáljuk meg, elértük-e a feltételül szabott értéket. A visszaugrások tehát most feltételesek, csak $A \leq 8$ esetben kell ismételni a kiírást.

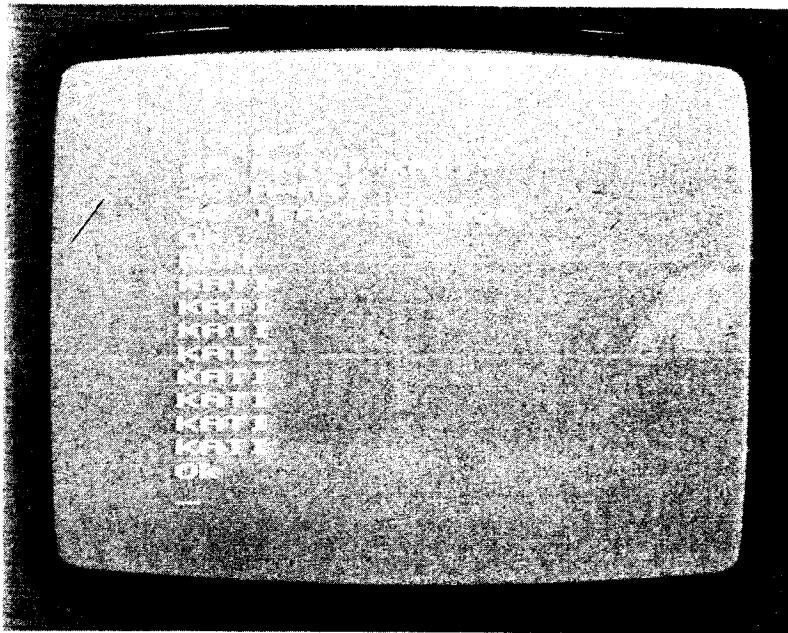
Az ilyen feltételes ugrást az IF . . . THEN GOTO . . . utasítás hozza létre.

Az IF után írjuk a feltételt (≤ 8), a GOTO után pedig annak a sornak a sorszámát, ahová vissza akarjuk ugratni a programunkat.

A program tehát ilyen lesz:

```
10 A = 1
20 PRINT „KATI”
30 A = A + 1
40 IF A <= 8 THEN GOTO 20
RUN
```

A 40. sorból a GOTO utasítást elhagyhatjuk, a program akkor is működik. (12. kép)



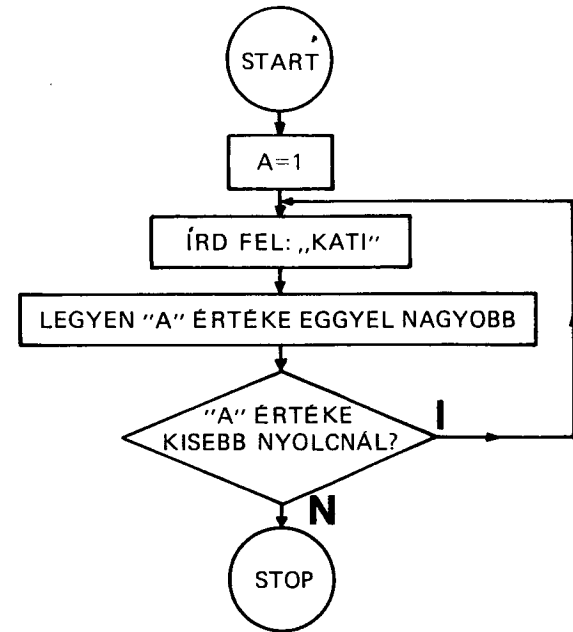
12. kép

```
40 IF A = 8 THEN 20
```

Amikor tehát A értéke 8-nál nagyobb lesz, a program nem tér vissza a 20. sorba, hanem megáll, mert nincs nagyobb számú utasítás.

Rajzoljuk le a folyamatábrát! (4. ábra)

Az ábrán jól látjuk a feltétel teljesülését.



4. ábra

Hasonló eredményt érhetünk el a FOR . . . TO . . . NEXT utasításokkal is. Lássuk így programunkat!

```
10 FOR A = 1 TO 8
20 PRINT „KATI”
30 NEXT A
```

A FOR A = 1 TO 8 azt jelenti: növekedjen A értéke 1-től 8-ig egyesével, és minden esetben hajtsa végre a következő utasításokat a NEXT A utasításig. Ez a megoldás az előbbinél elegánsabb, rövidebb, és gyakoribb is.

A FOR–NEXT ciklusképző utasítás nagyon gyakori a programokban, érdemes jól megismerni. Segítségével feltérképezhetjük a PRIMO teljes karakterkészletét, amelyeket a billentyűkön nem láthatunk.

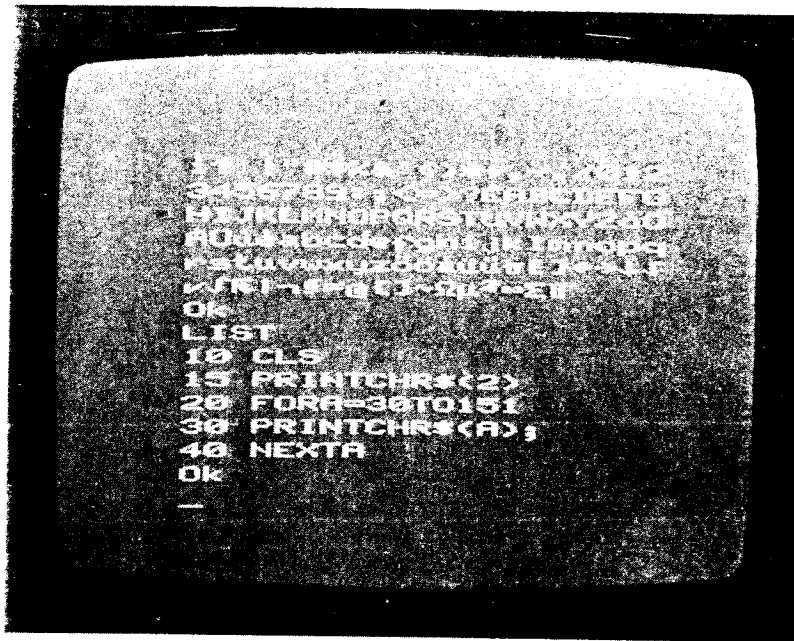
Az ASCII kódról korábban már szóltunk, a BASIC nyelvben ezek a kódok CHR\$ kódként szerepelnek. A PRIMO CHR\$ kódjai 1-től 30-ig vezérlő kódok, ezeknek nincs képük a képernyőn. Közülük érdekes a CHR\$ (7) kód, ami hangjelzést ad.

A látható karakterek a 30-tól 151-ig terjedő tartományban vannak. Írjunk tehát egy FOR . . . NEXT ciklust, amely 30-tól 151-ig fut és írassuk ki minden közbelső érték CHR\$ karakterét. A jobb láthatóság kedvéért a karaktereket megnyújtva írjuk (15. sor).

```

10 CLS
15 PRINT CHR$ (2)
20 FOR A = 30 TO 151
30 PRINT CHR$ (A)
40 NEXT A
RUN

```



13. kép

Ha olyan jelet akarunk írni, melynek nincs a billentyűkön képe, táblázatból kikeressük a CHR\$ kódját és ezt írjuk a helyére. Például: T²

```
PRINT CHR$ (128)::PRINT CHR$ (148)
```

Már tudjuk, hogy a számítógép minden számítást apró elemekre bont és úgy végzi el. Hasonlóképpen az általunk készített programok is apró részekből, sorokból állnak és az ezekben leírt utasításokat végzi el a gép sorról sorra haladva.

Gyakori a programban, hogy egyes részeket többször is meg kell ismételnünk változtatás nélkül. Most nem az előzőekben bemutatott egysoros kírásra gondolunk, hanem olyan programrészeire, amely több tíz vagy száz sorból is állhat. Ilyenkor nem célszerű ismételtetni a hosszú részeket, hanem

egy utasítással akármikor elővehetjük egyszer megírt programrészünket, melyet felhasználásig a számítógép memóriájának egy részében tárolunk. Ezt az önálló programrészt szubrutinnak nevezzük, és a GOTO utasításhoz hasonló GOBUB utasítással keressük meg. Legyen egy egyszerű szubrutin a következő:

```

1000 PRINT
1010 PRINT „ *** KATI *** ”
1020 BEEP 70, 300
1030 RETURN

```

Az első három sorhoz nem kell sok magyarázat, ezek az utasítások már ismertek. Az ezres sorszám jó messze van a programtól, ide már nyugodtan írhatjuk a kis szubrutint.

Ez persze nem előírás, a szubrutin állhat a programban bárhol, de kezdetben helyesebb, ha jól elkülönítjük, nehogy belekeveredjen a főprogramba.

Az 1030 -as sorban levő RETURN utasítás zárja a szubrutint és egyben visszatéríti a program végzését az eredeti programba. Éppen oda, ahonnan elugrattunk a szubrutint elvégezni. (A RETURN utasítást betűnként kell beírni, tehát nem a jobb oldali zöld billentyűt jelenti!) Legyen a programunk a következő:

```

10 CLS
20 PRINT „KIPRÓBÁLJUK A SZUBRUTINT”
30 PRINT
40 GOSUB 1000
50 GOSUB 1000
60 GOSUB 1000
70 END

```

Az első három sor itt is ismerős: az üres képernyőre felírtuk a program címét. A 40. sor GOSUB 1000 utasítása hatására a program az 1000. sorban folytatódik. A szubrutin végén a RETURN hatására a program a 40. sor után folytatódik tovább. Az 50. sorban újból a szubrutin kerül elő, és ugyanez ismétlődik a 60. sorban is. Ezzel a viszonylag egyszerű eljárással jelentősen megkönnyíthetjük a programírást.

A szubrutinok használata nagyon gyakori a programokban, érdemes megismerni, sőt megtanulni a használatát.

Futtassuk le a mintaprogramot (RUN) és látjuk, a szubrutin segítségével egyszerűbben oldhattuk meg egy kírás ismétlését, sőt még minden kírás után hangjelzést is adhatunk. Szubrutin alkalmazására több példát is találunk a mellékletben leírt BARKOCHBA programban.

Röviden még szólnunk kell a PRIMO rajztudásáról is, ami a hasonló kategóriájú számítógépekhez viszonyítva nagyon jó. A képmezőt 192 vízszintes sorra, minden sort 256 képpontra osztja. Ezeket a piciny képpontokat egyenként fel lehet villantani a SET utasítással. A pontok helyét a matematikában már megismert koordináta-rendszer segítségével határozhatjuk meg. Az x-tengelyt a képernyő alján, az y-tengelyt a képernyő bal szélén rajzolhatjuk ki. Lássuk, hogyan?

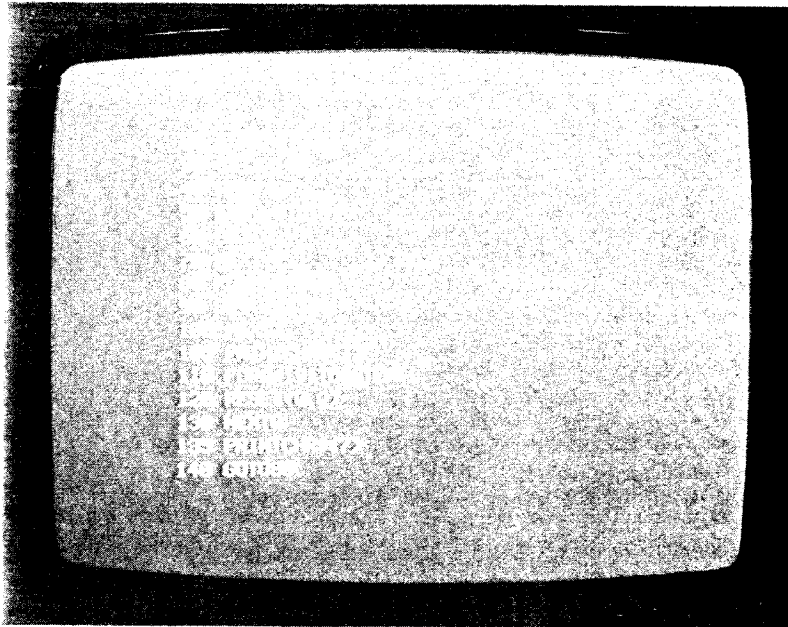
Természetesen a már megismert FOR . . . NEXT utasításokkal.

```
10 CLS
20 FOR X=0 TO 255
30 SET (X,0)
40 NEXT X
Ez idáig az X-tengely.
50 FOR Y=0 TO 191
60 SET (0,Y)
70 NEXT Y
```

Ez pedig már az y-tengely. Hasonló FOR . . . NEXT ciklusokkal tetszés szerinti egyeneseket, grafikonokat, ábrákat rajzolhatunk. Ha pedig teljesen szabálytalan volanat akarnak rajzolni, akkor pontonként kell megtervezni és bizony türelmesen beírogatni a sok SET utasítást. Legalábbis addig, míg alaposabban meg nem ismeritek a PRIMO-t és a programozást.

Végül egy mozgó vonal programját nézzük meg.

A RESET utasítás a SET utasítás ellentettje, a megadott koordinátájú pontokat kioltja, tehát törölhetjük vele a vonalakat is. (A RESET utasítást is betűnként kell beírni!)



14. kép

Folytassuk az előző programot:

```
80 FOR Z = 0 TO 191
90 SET (Z, Z)
100 NEXT Z
105 PRINT CHR$(7)
110 FOR Q = 191 TO 0 STEP-1
120 RESET (Q, Q)
130 NEXT Q
135 PRINT CHR$(7)
140 GOTO 80
```

Hasonló mozgó ábrákat könnyen rajzolhattok Ti is. (14. kép)

A CHR\$(7) kódnál és a BEEP utasításnál már említettük, hogy a PRIMO nemcsak a billentyűk érintésére ad hangot, hanem a hangjelzés beépíthető a programokba is.

A PRIMO nem hangszer, de azért egy kis ügyeskedéssel egészen jó dallamokat lehet kicsalogatni belőle. A hangadó utasítás a BEEP. Utána két számot kell írni. Az első adja a hang magasságát, a második a megszólalás hosszát.

Próbáljuk ki!

```
BEEP 200, 500
```

Ez egy viszonylag alacsony hang. Egy oktávval magasabban szól, ha az első számjegyet felére csökkentjük.

```
BEEP 100, 300
```

A két érték között kiszámíthatjuk bármely hanghoz tartozó számértéket. Ezek a következők: 200; 188.7; 178.2; 168.8; 158.7; 149.8; 141.4; 133.5; 126; 118.9; 112.2; 105.9; 100

Az egész hangokat dőlt írással jelöltük. Ezek alapján egy dūr skálát muzsikáló program a következő:

```
10 PRINT „DUR SKALA”
```

```
20 BEEP 200, 300: BEEP 178.2, 300: BEEP 158.7, 300:
   BEEP 149.8, 300: BEEP 133.5, 300: BEEP 118.9, 300:
   BEEP 105.9, 300: BEEP 100, 300
```

A program beírásakor ügyeljünk arra, hogy a számítógép a tizedesvessző helyett pontot kér és az egyes BEEP utasításokat kettősponttal válasszuk el.

Egy kis ügyességgel bármilyen programba beépíthetjük a „zenei” részeket. Érdemes kísérletezni!

Folytathatnánk az ismerkedést a BASIC nyelv további utasításaival, de nem volt célunk BASIC tankönyvet írni, inkább csak a kezdéshez szükséges legfontosabb ismereteket szeretnénk volna bemutatni.

Lássuk csak, mivel is ismerkedtünk meg eddig?

```
Parancsok:  RUN   PRINT
             LIST
             NEW
             EDIT
```



```
Utasítások: PRINT IF ... THEN RETURN
            INPUT FOR ... NEXT GOSUB
            GOTO
            BEEP SET
            CLS RESET
```

Nem is olyan sok, és mégis mennyi programot írhatunk már a segítségükkel.

Aki eddig eljutott, biztosan nem sajnálja a fáradságot és megismeri a teljes BASIC nyelvet. A szakkönyvek ehhez minden segítséget megadnak, de a játékban résztvevőknek mi is mindenben segítünk.

FELADATOK

Akinek már ez is sok

Valójában aki idáig figyelmesen olvasta játékkönyvünket, feladatok egész sorát oldotta meg. Megismerte nagy vonalakban a számítógépek történetét, már azt is tudja, milyen részekből áll egy számítógép. „Ismeri” a PRIMO gépet és tud egyszerű BASIC programot is írni. Nem kis feladatok voltak ezek és bizonyára sok időt áldoztatok rá.

Aki úgy érzi, számítógépből ennyi elég, máris megállhat! Reméljük csak pihenni.

És pihenés alatt érdemes azon elgondolkodni, hogy az elolvasott részekből melyek tetszettek, melyek voltak nehezen érthetők és melyeket nem lehetett egyáltalán megérteni. Célunk az volt, hogy jól olvasható és viszonylag könnyen érthető, a számítógéphez kedvet csináló könyvet adjunk a kezetekbe. Írjátok meg véleményeteket, mennyire sikerült célunkat elérni.

A Barátunk a számítógép játéknak indul. A játéknak — mint minden játéknak — vannak játékszabályai. A számítógépes játék játékszabályait nem lehet néhány mondatban elmondani, ez a játék több mint játék. Inkább a megismerkedés módjára illik a játék szó, de valójában a számítógépekkel való ismerkedés komoly és kitartó munkát igényel. Mi nem akarunk számítógépes szakembereket — programozókat, számítógéptervezőket — képezni, de az érdeklődőknek minden lehető segítséget megadunk az ismerkedésre — kinek-kinek igényei szerint.

Akinek a számítógépek története kötötte le a figyelmét és úgy érzi, olvasmányai között sok érdekes és számunkra értékes adattal találkozott, írja meg ezeket. Később egy másik könyvben segítségükkel már teljesebb képet tudunk adni. Aki úgy gondolja, hogy az eddig leírtakhoz egyszerű, de jó rajzokat tudna készíteni, kezdjen hozzá. A jó rajzok érthetőbbé, hangsúlyosabbá teszik a játékkönyvet. Várjuk tehát a rajzokat is.

Aki nem akar elmélyülni a programozás rejtelseibe — mert esetleg még nincs számítógépe — az is bekapcsolódhat az előző feladatok megoldásával játékunkba.

Ha pedig a pihenőt megunva az ismerkedés folytatása mellett döntötetek, következnek az igazi feladatok.

Akinek ez is elég

A következő feladatok a számítógépek alkalmazásával kapcsolatosak. Most lesz szükség igazán alkotó fantáziátokra! És persze egy kis tudásra is, hiszen néhány feladatnál konkrét megoldásokat is várunk, ezek pedig tudás nélkül . . .

A feladatok között találtok könnyűeket, nehezeket, érdekeseket, még érdekesebbeket és unalmasakat is. Válasszatok közülük és oldjátok meg eredményesen. Nem sűrges az idő, több oldalról is körüljárhatjátok a témákat, de igyekezzetek alapos munkát végezni. Ne feledjétek, a feladatok megoldásából a legnagyobb hasznot ti húzzátok, amikor részletesen és főként önállóan ismerkedtek egy-egy témakörrel.

Az önálló munkát nem könnyű megszokni, ezek a feladatok viszont igénylik az önállóságot. Tervezzétek meg a megoldást! Gondoljátok végig, kitől milyen segítséget kell kérnetek, hogyan kezdtek neki a megoldásnak, hol gyűjtitek össze a szükséges adatokat.



A megoldásokban szívesen olvasnánk segítőtjeiről is. Ezek után lássuk a feladatokat!

1. Számítógép az iskolában

Van az iskolátokban számítógép? Reméljük, a kérdésre többségetek már igennel felel. Ők vannak könnyebb helyzetben, hiszen ha van gép, biztosan használják is valamire. De mire? A számítógép helyettesítheti például a táblát. Gyorsan lehet a képernyőre írni, könnyen és tisztán lehet letörölni.

De érdemes-e ezért használni a számítógépet?

Nyilvánvalóan nem.

Akkor mire használjuk?

A számítógép helyettesítheti-e például a tanárt?

Tud feladatokat adni, a megoldásokat értékelni, sőt esetleg osztályozni is. Nem részrehajló, tárgyilagos.

De nem tud beszélgetni, nem tud összevont szemöldökkel csúnyán nézni egy-egy buta válasz után és nem tud kedvesen mosolyogni, biztatóan bölintani a bizonytalankodóra, sőt simogatni sem tud, pedig az is jólesik néha.

A számítógép tehát nem helyettesítheti a tanárt mindenben.

A számítógép segítő eszköz. Segíti a tanárt a munkájában, mert átvállalhat olyan feladatokat, melyekhez nem kell gondolkodni. Mert a számítógép bármilyen nagy és bonyolult is lehet, gondolkodni nem tud. Ezt fontos tudnunk. Segíthet bennünket a tanulásban, mert fáradhatatlanul tudja adni például a számtan feladatokat és tévedés nélkül azonnal értékeli az eredményeket. De csak addig tart a tudása, ameddig a programja engedi. Az a program, amit a tanárotok vagy esetleg ti írtatok.

Tanulni sokféle módon lehet. Minden esetben más és más módja van a segítésnek is. Kevés olyan programot ismerünk még, amit az iskolában jól lehetne használni. Írjatok ilyeneket. Kinek melyik tantárgy a kedvence, ahhoz kapcsolódó programot írjon. Persze, tudjuk jól, hogy többségetek még csak most ismerkedik a számítógéppel, a programíráshoz viszont elég nagy tudás és gyakorlat kell. Ez azonban ne kedvetlenül senki benneteket. Írjátok le részletesen, mit kellene tudni a programnak és mi leírások alapján elkészítjük azt gyakorlott szakemberekkel. Általában az ötlet születik meg nehezebben, s ha az megvan, a többi már lényegesen egyszerűbb.

Várjuk tehát elképzeléseiteket, mire használhatjuk a számítógépet az oktatásban. Írhattok kész programokat is, de lehetőleg PRIMO gépre, és beküldhettek programterveket, ötleteket, javaslatokat, ha azok megvalósítható az oktatás szolgálatába állíthatók.

A játékkönyv mellékletei között találtok egy földrajzi témához kapcsolódó egyszerű programot, amit beírva PRIMO gépetekbe, játszva tanulhattok meg néhány dolgot a szomszédos országokról.

2. Számítógépes játékok

A személyi számítógépek gyors elterjedését a jobbnál jobb játékprogramok segítették. Játszani mindenki szeret és sokan csak azért vásároltak számítógépet, hogy játszhassanak. A gyárak is elsősorban játékprogramjaikon keresztül reklámozzák számítógépeiket.

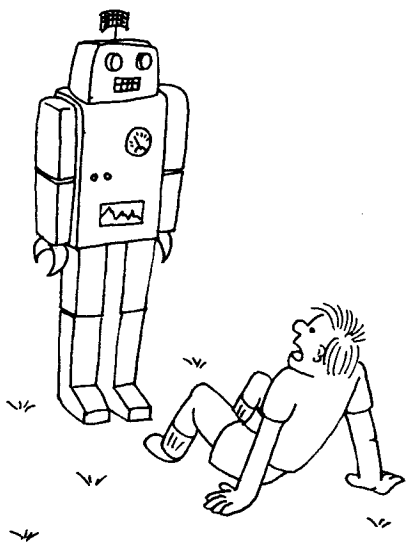
Mi is szeretnénk ötletes, izgalmas játékot játszani PRIMO gépünkön, de a gép annyira új még, hogy alig írtak rá játékprogramokat. Írjátok ti vagy írjunk közösen.

Itt is érvényes, amit az oktatóprogramoknál leírtunk, elég a játék ötletét, elképzelését leírni, mi majd szakemberekkel megírjuk a programot. De nagyon valószínű, hogy már vannak önállóan írt játékprogramjaitok, esetleg olyanok, melyeket más gépeken láttatok. Ezek leírásait is várjuk, hogy mielőbb közreadhassuk őket.

Jó lenne már sakkozni vagy labirintusjátékot játszani a PRIMO-val! De az egyszerűbb számkitalalós vagy reflexjátékokat esetleg ti is elkészíthetitek.

3. Számítógépek és robotok

Az 1985-ös vilákiállítás Japánban, Tokiótól nem messze, egy Cukuba nevű kisváros mellett rendezték meg. A kiállítás a jövőbe kalauzol, a számítógépek jövőbeni alkalmazását igyekeznek bemutatni úgy, ahogy a szakemberek azt ma elképzelik. A kiállított eszközök között ül egy orgonista és fáradhatatlanul játssza az eléje tett, közönséges kottapapírra írt dallamokat. Egy robotzenész. A feje tv-kamera, a teste emberéhez hasonló, csak kicsit szögletesebb, keze, lába úgy mozog, mint az emberé, hiszen az orgonajátékhoz nagyon pontos és finom, szinte emberi mozgásra van szükség. A kamera által „látott” képet egy számítógép dolgozza fel és ad utasítást a karokat, ujjakat, lábakat mozgató motoroknak, gépelemeknek. Ilyenek lesznek a robotok? Nem valószínű!



Ma még, ha robotokról hallunk, önkéntelenül is valami emberhez hasonló — gépemberre gondolunk, amely fáradhatatlanul járkal, dolgozik, és természetesen egyenlőre csak a fantázia világában él.

Nem véletlen ez, hiszen a gépemberek története századokra nyúlik vissza. Az ügyesen megszerkesztett fró, rajzoló, zenélő vagy sakkozó gépemberek azonban nagyon is valóságosak voltak. Igaz, hogy csak azt tudták, amire készítették őket, de már ez a tehetségük is csodálatra méltó volt.

Az elektronikus robotok már tanulékonyabbak. Ez nem csoda, hiszen működésüket számítógép vezérli, a számítógép pedig programjától függően sokféle feladat elvégzésére is alkalmas.

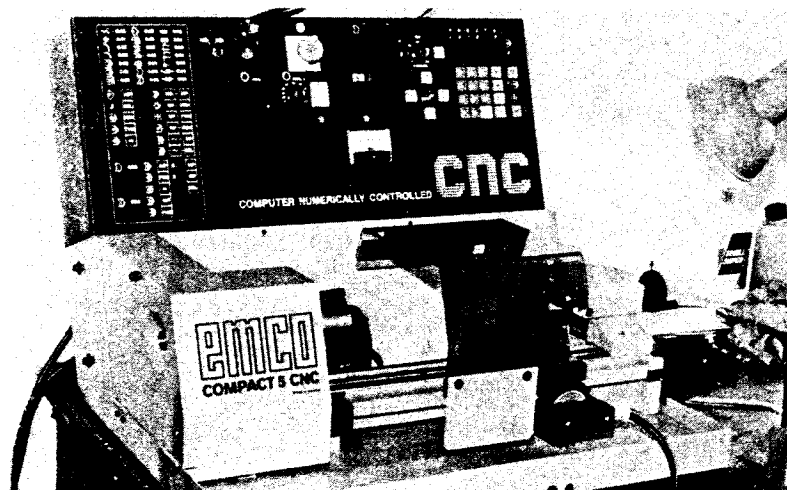
Az emberi forma az elektronikus robotoknál már háttérbe szorul. Az ipari robotoknak — legalább is a többségüknek — egyáltalán nincs emberi alakjuk.

Kényes, nagy figyelmet, pontosságot igénylő feladatokat tudnak végezni fáradhatatlanul. A pontosság még érthető, elképzelhető, de a figyelem? A robot a számítógépbe táplált programot hajtja végre. Ha ez a program olyan, hogy az emberi figyelmet, körültekintést utánozza és rugalmasan változik az adott körülményekhez igazodva, akkor a robot emberi mércével mérve is figyelmes.

Számtalan változatát ismerjük az ipari robotoknak. A bevezetőben említett vilákiállításon láthatók azok a japán autók, melyeket robotok szereltek össze.

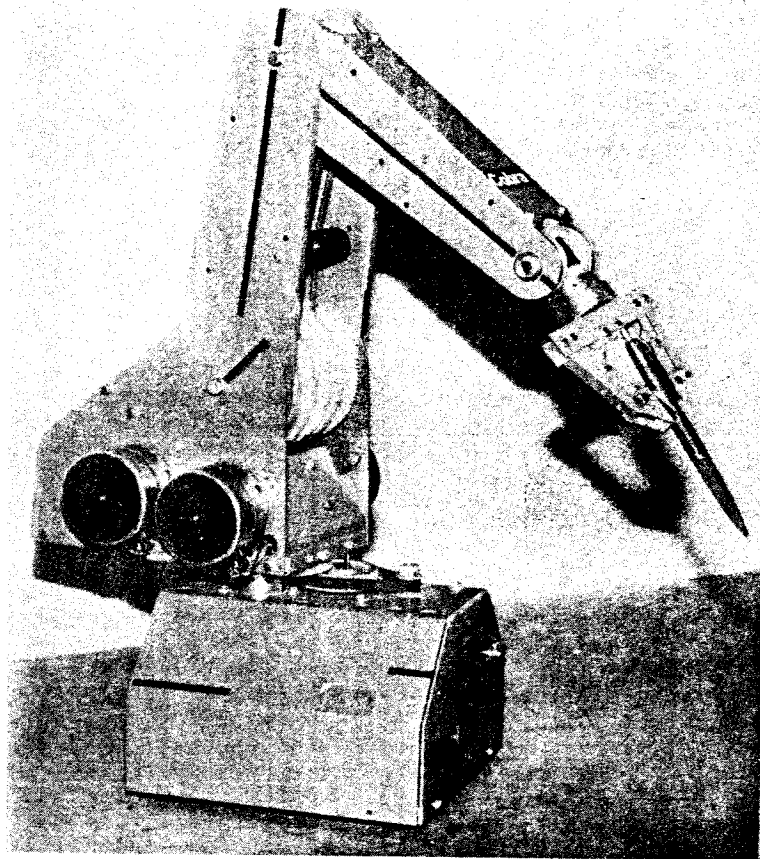
Napjaink visszatérő témája a gazdaságosabb, olcsóbb termelés. A robotok pedig olcsóbban dolgoznak az embernél, ezért indult rohamos fejlődésnek a robottechnika.

Ha elfogadjuk, hogy nemcsak az emberformájú gépeket nevezzük robotoknak, meglepődünk, milyen sok robot dolgozik már nálunk is.



15. kép

Itt vannak például a számítógép vezérlésű esztergagépek. (15. kép)
Az ember elkészíti egy bonyolult alkatrész előállítás programját, be-
táplálja a számítógépbe és már nincs is más dolga, a gép pontosan elvégzi a
munkát és dolgozik folyamatosan, ha kell napokon-heteken át megállás
nélkül. A képen bemutatott gépen a jövő technikatánárai tanulmányozzák
a számítógépek ipari alkalmazását. De ugyanitt – az ELTE Általános Tech-
nika Tanszékén – működik egy iskolarobot, amely apró tárgyakat tud
megfogni, felemelni és a kívánt helyre letenni. Mindezt számítógépvezérlés-
sel. (16. kép)



16. kép

Az űrkutatás el sem képzelhető robotok nélkül, hiszen az ember számá-
ra veszélyes vagy elviselhetetlen környezetben is kiválóan dolgozhatnak a
robotok. Bizonyára olvastatok ti is arról, hogyan hoztak talajmintákat a
Holdról a szovjet űrhajók, vagy hogyan végeztek talajvizsgálatokat a Vénusz-
ra leszállt automata űrállomások robotjai.

Robotpilótákról is hallottatok már. A korszerű utasszállító repülő-
gépeket csaknem egész útjuk során számítógépek vezérlik. A számítógép
tehát a robotokon keresztül tud átvállalni egyre több munkát az embertől.
Egyet azonban nem: gondolkodni sem a számítógép, sem a robotok nem
tudnak. A gondolkodás, a gondolatok csak az emberre jellemzők.

Amikor gondolkodó gépekről, gondolkodó robotokról beszélnek,
soha ne feledjük, ezek a gépek csak azt teszik, amire az ember megtanította
őket. Végzik kitartóan a program diktálta feladatokat, számolnak, vezérel-
nek, szabályoznak, hamarosan beszélnek is, de mindvégig gépek maradnak.
A PRIMO számítógépet elsősorban nem robotok vezérlésére készítették,
de mint a legtöbb számítógép, ez is alkalmassá tehető robotok vezérlésére.
Ehhez azonban szét kellene szedni, amit most még nem tanácsolunk.

De összegyűjthetitek, rendszerezhetitek, feldolgozhatjátok a robotok
történetét, a környezetekben látott robotok használatát, a robottechnika
fejlődését. Ez a gyűjtőmunka is része a játékunknak, és örömmel olvasnánk
arról, hogy gyűjteményetekkel mások is megismerkedhetnek. Akik pedig
már gyakorlottak más számítógépek kezelésében, azoktól természetesen
olyan megoldásokat várunk, melyben gépüket (a HT 1080Z-t, a ZX 81-et)
vezérlési feladatokra is megtanították. Segítséget ehhez az Ötlet című lapból
vehetek, ötleteiteket pedig elkülditek nekünk, s mi tesszük közzé – talán
éppen az Ötletben.

4. Számítógépek a termelésben

Számítógépes játékunkba a nevezés feltétele volt, hogy környezetekben
kutassátok fel a számítógépek alkalmazását. Ezt nyilvánvalóan meg is tetté-
tek, így ez a feladat nem jelent különösebb nehézséget.

Vagy mégis? Hiszen most nem általában a számítógépekről érdeklő-
dünk, hanem konkrétan a termelés irányításában, szervezésében elfoglalt
helyüket kell felkutatni. Aligha találtok már olyan gyárat vagy üzemet,
ahol ne alkalmaznának számítógépet sőt számítógépeket. Ha mégis rábukkan-
tok ilyen helyre és úgy látjátok, ott is szükség lenne a számítógépre, erről
is írhattok. Ez is lehet egy feladat.

Valószínűbb azonban, hogy a felkeresett üzemben ott a számítógép és
alaposan ki is veszi részét a termelésből. Talán még robotokkal is találkoztok!

Ha úgy találjátok, hogy ott még sok feladatot átvállalhatnának a számí-
tógépek, erről is írjatok esetleg rajzzal, programokkal, folyamatábrákkal
kiegészítve.

Ugye, már nem is olyan egyszerű ez a feladat, sőt talán ez a legnehe-
zebb és legkomolyabb.

5. Számítógépek az ember életében

- A számítógépek a jövő gépei, a mai ember életében még nincs jelentőségük.
- Számítógépek nélkül már el sem tudnám képzelni az életemet.

Kinek higgyünk? Érzitek ugye, az igazság valahol a két kijelentés között van. Az igaz, hogy a számítógépek a jövő gépei, de már ma is nélkülözhetetlenek sok esetben. Legfeljebb nem tudjuk, milyen sok helyen segítenek bennünket napról napra. Elolvassuk az újságot és eszünkbe se jut, hogy az újságcikkeket már nem a régi ólombetűs szedőgépeken szedik, hanem számítógépvézérelt fényszedőgépeken. Nézzük a televízióban a képiújságot, de miért is gondolnánk arra, hogy most egy számítógép által összeállított képet látunk. A játéktermekről már nem is beszélünk, pedig a video játékgépeket is természetesen számítógépek vezérlik. De számítógép vezérli a forgalomirányító lámpákat, a pályaudvarok utastájékoztató tábláit, a stadionok eredményjelzőit is.

Soroljátok ti tovább! Hol találkozunk számítógéppel úgy, hogy észre sem vesszük? A gyűjtőmunkához három területet javasunk: a háztartást, a közlekedést és a szórakozást.

6. A jövő számítógépei

Eddig többnyire azzal foglalkoztunk, milyenek a számítógépek, mire és hol használják őket. Nézzünk most **egy** kicsit a jövőbe. Milyenek lesznek a jövő számítógépei és milyen új **feladatok** elvégzésére válnak alkalmassá. Illetve fogalmazzunk pontosabban: azt, hogy milyenek lesznek, nem tudhatjuk pontosan, de azt már igen, milyenek szeretnénk, ha lennének. Beszélnek, írnak, olvasnak, esetleg tanulnak is helyettünk?

De mit csinálunk akkor mi, emberek? Erre is gondoljatok, amikor fantáziátok segítségével felvázoljátok a számítógépek jövőjének körvonalait. A számítógépekét és az embereket, hiszen a számítógépeknek csak az emberek mellett lehet jövőjük, és az emberek jövője is nagymértékben függ a számítógépektől.

Hogyan élünk 50, 100, 500 vagy 1000 év múlva?

Lesznek-e még akkor számítógépek vagy már csak azok lesznek? Mi erről a véleményetek?

Elképzeléseiteket rajzokkal, illusztrációkkal tegyétek színesebbé, teljesebbé.

Akinek még ez is kevés

Vannak — és reméljük lesznek is egyre többen — közöttetek, akik az előző feladatokra csak legyintenek és azt kérdezik: — Ezek is feladatok? Mi ennél már sokkal többet tudunk!

Ez nagyszerű! Itt az alkalom, hogy bennünket, a kezdőket vagy szerény haladókat beavassatok titkaitokba. Ne sajnáljátok a fáradságot és mutassátok meg, mi mindenre képesek a ti számítógépeitek. Azt ugyanis tudjuk, hogy amit a számítógépek tud, azt tőletek tanulta, tehát ti elég sok mindent ismerhettek már a gépekről és a programokról.

Előző feladataink nem fogták át a számítógépek alkalmazásának valamennyi területét. Nem is tehették volna, hiszen ki tudja ma már, mi mindenre taníthatók meg a számítógépek. Így azután esetleg érdekes területek maradtak ki a felsorolásból. Most még kimaradtak, de segítségetekkel legközelebb már teljesebb lesz a feladatsor, bővebbek a programmelléletek. Játékkönyvet ígértünk a címben, nem játékgyűjteményt. A játékszabályokat most megismerhették, következhetnek a jobbnál jobb programok, amiket elsősorban tőletek várunk (írottan vagy kazettán). Várunk számítógépépítő, -bővítő, -kiegészítő leírásokat, terveket, elképzeléseket. Nem szabjuk meg a felhasználható számítógéptípust, de emlékeztetünk rá, hogy a PRIMO-t szeretnénk minél sokoldalúbban használni.

IRODALOM

MELLÉKLETEK

- D. Alcock: Ismerd meg a BASIC nyelvet!
Billigné Szőnyi K.—Csébfalvi K.: Mire képes a számítógép?
Bradbeer—Bono—Laurie: Műsor a számítógép
C. Brown: Beszélgessünk a számítógépről
R. Coles: Kulcs a mikroszámítógéphez
Filep—Bereznai: A számírás története
O. Jursa: Kibernetika
J. G. Kemeny: Az ember és a számítógép
F. Lehberg—T. Lutz: Hogyan gondolkodik az elektronikus számítógép?
V. Pekelisz: Kibernetikai kis enciklopédia
Primo felhasználói kézikönyv
Szűcs E.: Beszélgessünk a technikáról
Szűcs E.: Technika és számítógép
Tóth I.—Stábel O.: A számítógép és az ember
F. G. Withington: A számítógép a valóságban

CHR\$ KÓDOK

CHR\$ kód	karakter	CHR\$ kód	karakter	CHR\$ kód	karakter
1	Normál üzemmód	52	4	103	g
2	Nyújtott karakter	53	5	104	h
3	Negatív kép	54	6	105	i
4	Negatív alap	55	7	106	j
5	Aláhúzott karakter	56	8	107	k
6	Előtörlés	57	9	108	l
7	Hangjelzés	58	:	109	m
8	Egy karakter vissza	59	.	110	n
9	Horizontális tabulátor	60	<	111	o
10	—	61	=	112	p
11	—	62	>	113	q
12	CLS	63	?	114	r
13	RETURN	64	É	115	s
14	Csak kocsivissza	65	A	116	t
15	—	66	B	117	u
16	Alsóindex	67	C	118	v
17	Felsőindex	68	D	119	w
18	2. kód inverze	69	E	120	x
19	3. kód inverze	70	F	121	y
20	4. kód inverze	71	G	122	z
21	Aláhúzás kikapcsoló	72	H	123	ö
22	6. kód inverze	73	I	124	ó
23	—	74	J	125	á
24	—	75	K	126	ü
25	—	76	L	127	ű
26	—	77	M	128	π
27	—	78	N	129	l
28	—	79	O	130	l
29	—	80	P	131	l
30	—	81	Q	132	l
31	↑	82	R	133	l
32	Space	83	S	134	l
33	!	84	T	135	l
34	"	85	U	136	l
35	#	86	V	137	l
36	\$	87	W	138	l
37	%	88	X	139	l
38	&	89	Y	140	l
39	.	90	Z	141	l
40	(91	ó	142	l
41)	92	Ö	143	l
42	*	93	Á	144	l
43	+	94	Ü	145	l
44	,	95	ű	146	l
45	—	96	é	147	l
46	.	97	a	148	l
47	/	98	b	149	l
48	0	99	c	150	l
49	1	100	d	151	l
50	2	101	e		
51	3	102	f		

ASCII KÓDTÁBLÁZAT

A kód			A karakter
bináris	Dec.	Hex.	
bit 76543210			
00000000	0	00H	
00000001	1	01H	
00000010	2	02H	
00000011	3	03H	
00000100	4	04H	
00000101	5	05H	
00000110	6	06H	
00000111	7	07H	
00001000	8	08H	
00001001	9	09H	
00001010	10	0AH	
00001011	11	0BH	
00001100	12	0CH	
00001101	13	0DH	
00001110	14	0EH	
00001111	15	0FH	
00010000	16	10H	
00010001	17	11H	
00010010	18	12H	
00010011	19	13H	
00010100	20	14H	
00010101	21	15H	
00010110	22	16H	
00010111	23	17H	
00011000	24	18H	
00011001	25	19H	
00011010	26	1AH	
00011011	27	1BH	
00011100	28	1CH	
00011101	29	1DH	
00011110	30	1EH	
00011111	31	1FH	
00100000	32	20H	
00100001	33	21H	
00100010	34	22H	
00100011	35	23H	
00100100	36	24H	
00100101	37	25H	
00100110	38	26H	
00100111	39	27H	
00101000	40	28H	
00101001	41	29H	
00101010	42	2AH	
00101011	43	2BH	
00101100	44	2CH	
00101101	45	2DH	
00101110	46	2EH	
00101111	47	2FH	
00110000	48	30H	
00110001	49	31H	
00110010	50	32H	
00110011	51	33H	
00110100	52	34H	
00110101	53	35H	
00110110	54	36H	
00110111	55	37H	
00111000	56	38H	
00111001	57	39H	
00111010	58	3AH	
00111011	59	3BH	
00111100	60	3CH	
00111101	61	3DH	
00111110	62	3EH	
00111111	63	3FH	

A kód			A karakter
bináris	Dec.	Hex.	
bit 76543210			
01000000	64	40H	ð
01000001	65	41H	A
01000010	66	42H	B
01000011	67	43H	C
01000100	68	44H	D
01000101	69	45H	E
01000110	70	46H	F
01000111	71	47H	G
01001000	72	48H	H
01001001	73	49H	I
01001010	74	4AH	J
01001011	75	4BH	K
01001100	76	4CH	L
01001101	77	4DH	M
01001110	78	4EH	N
01001111	79	4FH	O
01010000	80	50H	P
01010001	81	51H	Q
01010010	82	52H	R
01010011	83	53H	S
01010100	84	54H	T
01010101	85	55H	U
01010110	86	56H	V
01010111	87	57H	W
01011000	88	58H	X
01011001	89	59H	Y
01011010	90	5AH	Z
01011011	91	5BH	[
01011100	92	5CH	\
01011101	93	5DH]
01011110	94	5EH	^
01011111	95	5FH	_
01100000	96	60H	a
01100001	97	61H	b
01100010	98	62H	c
01100011	99	63H	d
01100100	100	64H	e
01100101	101	65H	f
01100110	102	66H	g
01100111	103	67H	h
01101000	104	68H	i
01101001	105	69H	j
01101010	106	6AH	k
01101011	107	6BH	l
01101100	108	6CH	m
01101101	109	6DH	n
01101110	110	6EH	o
01101111	111	6FH	p
01110000	112	70H	q
01110001	113	71H	r
01110010	114	72H	s
01110011	115	73H	t
01110100	116	74H	u
01110101	117	75H	v
01110110	118	76H	w
01110111	119	77H	x
01111000	120	78H	y
01111001	121	79H	z
01111010	122	7AH	{
01111011	123	7BH	
01111100	124	7CH	~
01111101	125	7DH)
01111110	126	7EH)
01111111	127	7FH)

BARKOCHBA

```

1  GOSUB 5:GOSUB 5:GOSUB 5: FOR I=50 TO 10 STEP -5: BEEP I,200:
   NEXT I: GOTO 10
5  FOR I=60 TO 100 STEP5 :BEEP I,100 :NEXT I
6  RETURN
10 CLS
20 ? "EGY ÉRDEKES JÁTÉKRA HIVLAK. A TE FELADATOD LESZ, HOGY
   MAGYARORSZÁG ÉS A SZOMSZÉDOS ORSZÁGOK KÖZÜL VÁLASSZ
   EGYET."
30 ? "KÉRDÉSEKET TESZEK FÖL, AMELYEKRE TE IGENNEL (I-GOMB)?
   VAGY NEMMEL (N-GOMB) VÁLASZOLHATSZ. A VÁLASZAID ALAPJÁN
   MEGPRÓBÁLOM KITALÁLNI, MELYIK ORSZÁGRA GONDOLTÁL."
35 S=-1
40 ? "KIGONDOLTAD AZ ORSZÁGOT? " :GOTO 2000
50 ? "AKKOR IGYEKEZZ!" :GOTO 35
60 ? "A DUNA ÁTFOLYIK RAJTA? " :GOTO 2000
70 ? "SZOVJETUNIÓRA GONDOLTÁL? " :GOTO 2900
80 ? "SZOCIALISTA ORSZÁG? " :GOTO 2000
90 ? "AUSZTRIÁRA GONDOLTÁL? " :GOTO 2900
100 ? "EBBEN AZ ORSZÁGBAN VAN AZ A CSATAMEZŐ, AHOL PETŐFI
    SÁNDOR ELESETT? " :GOTO 2000
110 ? "ROMÁNIÁRA GONDOLTÁL? " :GOTO 2900
120 ? "PÉNZEME KORONA? " :GOTO 2000
130 ? "CSEHSZLOVÁKIÁRA GONDOLTÁL? " :GOTO 2900
140 ? "CSODÁLATOS ÜDÜLŐHELYE AZ ADRIAI-TENGER? " :GOTO 2000
150 ? "JUGOSZLÁVIÁRA GONDOLTÁL? " :GOTO 2900
160 ? "MAGYARORSZÁGRA GONDOLTÁL? " :GOTO 2900
170 ? "HÁT VANNAK MÉG PÓTOLNIVALÓID. LÉGYSZIVES NE PRÓBÁLJ
    FÉLREVEZETNI!" :GOTO 190
180 ? "ÜGYES ÉS TÁJÉKOZOTT VAGY!" :GOTO 190
190 ? "AKARSZ MÉG JÁTSZANI? " :GOTO 3000
200 END
2000 AS=INKEY$
2010 IF AS="" OR AS <> "I" AND AS <> "N" THEN 2000
2015 S=S+1
2020 IF AS="I" THEN GOSUB 4000: ON S+1 GOTO 60, 80, 100, 110, 130, 150
2030 IF AS="N" THEN GOSUB 4000: ON S-1 GOTO 50, 70, 90, 120, 140, 160
2900 Z=-1
3000 AS=INKEY$
3010 IF AS="" OR AS <> "I" AND AS <> "N" THEN 3000
3015 Z=Z+1
3020 IF AS="I" THEN GOSUB 4000: ON Z+1 GOTO 180,35
3030 IF AS="N" THEN GOSUB 4000: ON Z+1 GOTO 170,200
4000 ? CHR$(7) : FOR I=1 TO 500 : NEXT I: CLS: BEEP 50,200
4010 RETURN

```