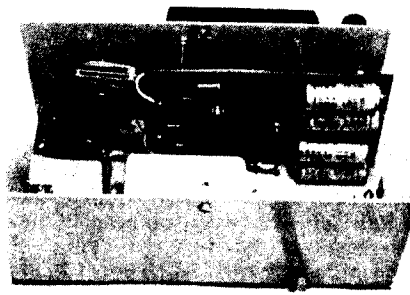


A diódán fellépő maximális — záróirányú — feszültség tehát azonos a transzformátor szekunder feszültségének kétszeres csúcsértékével (u_p ; „p” = peak = csúcs), ami másnéven a feszültség csúcsból-csúcsig mért szintje (u_{pp}).

A kétszeres csúcsfeszültség fellépését tudjuk kihasználni ahhoz, hogy feszültség-kétszerezést, illetve tet-szőleges számú többszörözést hoz-hassunk létre. Az 1c ábra feszült-ség-kétszerező kapcsolást mutat, amelyre jellemző az egymással sor-bakapcsolt két dióda és a 2 db kon-den-zátor. A kapcsolás kimenetén levő kondenzátoron tiszta egyen-feszültség jelenik meg terheletlen eset-ben, míg a bal oldali kondenzátor az egyenirányító negatív pólusához ké-pest váltófeszültségen van. Az 1d ábra az előző kapcsolás kismértékű átrajzolásával keletkezett. Mivel a váltóáramú oldal (felső oldal) kon-den-zátorán megtalálható a transz-formátor szekunder váltófeszültsége, újabb feszültség-kétszerező kapcso-lást alkalmazhatunk, sorbakapcsolva azt az előzővel (1e ábra). Így 4 db diódával és 4 db kondenzátorral fes-zültség-négyszerező kapcsolást kap-hatunk. Az 1. ábrán látható kap-csolások egyutas egyenirányítók, az-az a kimenetükön terhelés alatt fel-lépő brumm-feszültség frekvenciája azonos a tápláló váltófeszültség frekvenciájával (pl. 50 Hz, háló-zati táplálás esetén).

Más típusú feszültség-kétszerezőt mutat a 2a ábra. Páratlan számú feszültség-sokszorozást egy páros számú sokszorozó és egy egyszerű egyenirányító sorbakapcsolásával nyerhetünk (2b ábra). Feszültség-többszöröző kapcsolást kialakítha-tunk kétutas kivitelben is a 3. ábra kapcsolásai szerint, de ezek a meg-oldások ritkán fordulnak elő a gya-korlatban a nagyfokú dióda- és kon-den-zátor-igényük miatt.



9. ábra. A készülék belső elrendezése

A feszültség-sokszorozást bemu-tató gyakorlati készülékként egy Zener-dióda vizsgálat (Zener-Tester) mutatunk be a következőkben (4. ábra). Ezen egyszerű készülékkel maximálisan 40 V-os Zener-diódák üzemi feszültsége mérhető. Segítsé-gével ismeretlen típusok, lekopott felirátú diódák, Zener-diódát helyet-tesítő áramkörök letörési (Zener-) feszültsége határozható meg, illetve ismert típusok jó, vagy rossz álla-potára kaphatunk felvilágosítást.

A készülék kapcsolási rajza az 5. ábrán látható. A mintegy 11 V-ot szolgáltató csengőtranszformátor szekunder oldali feszültségéből 4-szerezés útján kb. -60 V egyenfe-szültséget állítunk elő. Erre szolgál a 4 db dióda és a 4 db 100 μ F-os elektrolitikus kondenzátor együtte-se. A készülék előlapi szorító kap-csaira csatlakoztatott vizsgálandó Zener-diódára egy áramgenerátor fokozaton keresztül kapcsoljuk rá a -60 V-ot. A BFY34 tranzisztor-ból és a 3 db szilícium diódából, va-lamint a 3 db ellenállásból álló áramgenerátor nagy belső ellenállása révén közel állandó áramot bocsát át a vizsgálandó Zener-diódákon, nagyjából függetlenül attól, hogy azokon mekkora feszültség esik.

Kapcsolásunkban az átfolyó áram 5,5...8 mA, amely áramra a Zener-diódák már a letörési karakterisz-tikájuk meredek szakaszán dolgoz-nak. A mérendő diódán átfolyó áram és a mérendő dióda Zener-feszültsége (U_z) közötti összefüggés a 6. ábra görbéjén látható.

A vizsgálandó Zener-dióda feszült-ségét egy 100 μ A-es műszer mutatja, amelynek előtét ellenállásai a K kapcsolóval átkapcsolhatók. A mű-szer végkitérése 8 V, illetve 40 V a kapcsoló állásától függően. Ez a kétáramkörös tárcás kapcsoló szolgál a készülék ki-be kapcsolá-sára is a transzformátor szekunder körében. A bekapcsolt állapotot egy CQY40 LED jelzi az előlapon. A vele antiparalel kapcsolt szilícium dióda a váltóáramú táplálás miatt szükséges.

A mérőműszer tekercsével párhu-zamosan kapcsolt szilícium dióda a műszert védi a 8 V-hoz tartozó előtétellenálláson fellépő nagyobb áramerősségektől, amikor a mérő-pontok üresen állnak (kb. 0,8 mA). A készülékben mindenütt BAY42 típusú diódákat alkalmaztunk. Természetesen más típusok is megfe-lelnek. A feszültség-négyszerező kap-csolásban nagyobb áramú típuso-kat használunk (pl. 1N4005 stb.), míg az összes többi helyre 1N4148, vagy ehhez hasonló kisáramú szilí-cium diódák is megfelelnek.

Az elektromos áramkör egy 130 $\times$ 75 mm méretű, egyoldalt felfro-zott nyák-lapon foglal helyet, amely-nek fóliarajzát a 7. ábra mutatja, beültetése a 8. ábrán látható. A NYÁK-lemezt a Mod. 71 típusú mű-szer két csavarja tartja a készülék előlapjával párhuzamosan. A BFY34 tranzisztorra hűtőcsillag alkalma-zása szükséges. A készülék dobozá-nak mérete: 200 $\times$ 80 $\times$ 105 mm. A belső elrendezés a 9. ábrán látható.

—ÖRR—

Úttörők Mikro-klubja

Dr. Appel György—Lukács János

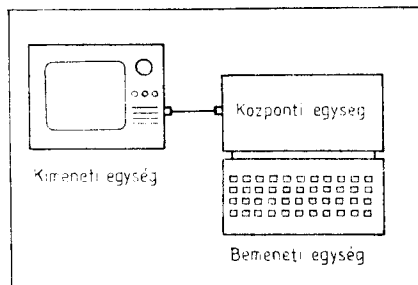
A személyi számítógépek megje-lenése felgyorsította a számítás-technika iránt az érdeklődést. A fia-talok is szeretnének megismerkedni a személyi számítógépekkel, a prog-ramozás alapelemeivel. De hol is-merkedjenek meg? Ezen próbálunk segíteni, amikor itt e cikksorozatot elkezdjük. A cikksorozattal párhuza-mosan megindul az általános isko-lákban is az iskolaszámítógép-prog-ram. S itt lehetőség nyílik, hogy az érdeklődő tanulók megismerkedjen-e témával. Terveink szerint ismertetjük a számítástechnika tör-ténetét, a személyi számítógép fel-építését, a BASIC nyelvet, prog-ra-mozási fogásokat. Természetesen

mindezt úgy tesszük (hisz feltételez-zük ifjú rádióamatőrök is lesznek!), hogy feladatainkat az elektrotech-nika témaköréből választjuk. S ha megismerkedünk az alapokkal, rá-térhetünk a gép felépítésének mé-lyebb ismeretére, ill. a gépi-nyelv (vagy más programozási nyelv) tit-kaira.

Szeretnénk, ha a cikksorozat nem-csak a szerzők egyoldalú közlései lennének. Az olvasótáborból min-denki feltehet a témával kapcsolatos bármilyen kérdést, a szerzők min-den esetben válaszolnak. A levele-ket a szerkesztőségbe lehet küldeni, írjátok rá: „Az úttörők Mikro-klubja”.

Röviden a számítástechnika történetéről

Az emberek mindig számoltak, még ha csak számláltak is. (Már az ősember is összeszámolta a zsák-mányt.) Dönteni, még ha nem is nevezte ezt logikai művelet végzésé-nek, nap mint nap kellett és kell ma is. Hosszú időnek kellett eltelni ahhoz, hogy ezeknek a műveleteknek gépi előkészítésére, megoldására, el-végzésére tömegesen álljanak ren-delkezésre eszközök, hogy megje-lenhessenek, tömegesen hozzáférhe-tők legyenek az e célra ma legal-kalmasabb gépek, a számítógépek.



1. ábra

Tekintsük át ennek az útnak fő állomásait: I. e. 3000-ban a számok és a kínaiak már ismerték számrendszereket és ez időből van már írásos emlék a számjegyről. A X. sz. végén az abacus, a „golyós számológép” az a *Gerbert* szerzetes hozta Európába, akitől mint *II. Szilveszter* pápától *I. István* király a koronát kapta. A XV–XVII. században a nagy földrajzi felfedezések, a hajózás, a térkép készítése és a tájékozódás igen sok és bonyolult számítás igényelt. 1614 óta *Napier* és *Briggs* jóvoltából a logaritmus, a logaritmustábla tette könnyebbé a számolást. A számításokat ekkor még „gyalog” végezték, de hiszen volt idő, mert a vitorláshajó lassan haladt. 1642-ben *Pascal* már mechanikusan működő összeadógépet, 1671-ben *Leibniz* a négy alapművelet elvégzésére alkalmas gépet szerkesztett. 1887-ben a svéd *Odher* állítható fogazású számkerékkel előállították azt a „kurblis” számológépet, amit még ma is használnak. A tűzérési löelemek még ha nem is olyan pontos, de gyors kiszámításához a XIX. sz. végétől használták a logaritmét. A múlt század végén jelentek meg szöveg gépek vezérlésére és adatfeldolgozására a lyukkártyák.

Ennyi előfutár, no meg az elektromosság térhódítása lehetővé tette az elektromos számológépek (pénztárgépek) megjelenését. Az utóbbi tíz évben az elektronika fejlődése révén hozzáférhető közelségbe kerültek a zsebszámológépek, a kalkulátorok. Ezek némelyike már nagyon sokat „tud”, de még mindig csak számológép. A számítógép az már más, az már valamivel több.

Neumann János (1903–1957)

1948-ban a következőkben fogalmazta meg a számítógéppel szembeni követelményeket:

1. A feladat megoldásához szükséges programot a gépben lehessen tárolni és ez a program változtatható legyen.

2. A program, vagy annak részei akárhányszor ismételtetők legyenek.

3. Ne csak számolni tudjon, hanem logikai műveletek végzésére is alkalmas legyen.

Ezeknek a szempontoknak megfelelő első gép 1941-ben készült. 2600 jelfogóból épült fel. A sok mechanikus mozgó alkatrész miatt működése lassú volt, mindössze

30–50 műveletet végezhetett percenként.

1946-ban készült el az első elektronikus számítógép, az ENIAC a pennsylvaniai egyetemen. Külső programvezérlésű számítógép volt és 20 db tízjegyű számot tudott tárolni. Műveleti sebessége: 5000 összeadás vagy 400 szorzás másodpercenként. A gép 18 000 elektroncsövet és 1500 jelfogót tartalmazott, ennek megfelelően helyigénye mintegy 180 m² alapterületű helyiség volt. Teljesítményigénye pedig 100 kW. Az ilyen „csöves” számítógépeket nevezik első generációs gépeknek. A második generációs gépek már tranzistorokból épültek fel. A harmadik generációs gépek aktív eleme a több tízezer tranzistorokból álló áramkör, az integrált áramkör. Az 1970-es évek elején jelent meg a számítógép központi egységének (processzorának) integrált áramköre. Ezeknek a korszerű negyedik generációs gépeknek a segítségével szélesíthetjük ki a számítógépet alkalmazók köre. Ma már mikroprocesszoros vezérlést találhatunk fényképezőgépekhez, mérőműszerekben, gépjárművekben, pénztárgépekben stb.

Maradjunk a számítógépeknél, még hozzá ezek közt is a legkisebbeknél, a személyi, házi, mikrokomputerek. Különböző típusú, nevű, méretű számítógépek közelébe kerülhetünk. HT – 1080Z, ZX–81, SPECTRUM, COMMODORE... de ne is folytassuk, mert nem lenne elég a hely mindegyik megemlézésére. Nézzük inkább azt ami közös ezekben a gépekben.

A gép technikai kialakítása, ami a felhasználó által nem (illetve csak megfelelő elméleti tudás birtokában) változtatható meg, a hardver. Ebben az összeállításban nem a kódolás miatt – hanem mert egyelőre nem célunk és esetleg elrettentő dolgok is előkerülnének – a gépet, a gép „testét”, a hardvert fogadjuk olyanak, amilyen. Ha ez egy ZX–81, akkor megjelenésében is „fekete doboz”. A számítógép – a hardver – fő részei az ábrán láthatóak. Azokat az egységeket, melyek segítségével a számítógéppel a kapcsolatot tarthatjuk („beszélgethetünk” a géppel) perifériáknak nevezzük.

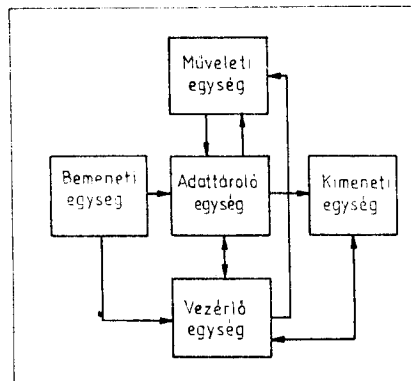
Vannak nélkülözhetetlen perifériák, melyek nélkül még ki sem próbálhatjuk a gép működését. Ezek a bemeneti mint kimeneti egység és a képernyő (TV) mint kimeneti egység. Egy sok célra felhasználható számítógéphez azonban más perifériák is csatlakoztathatók.

A központi egység (CPU) fő részei:

operatív tár,
műveleti rész,
vezérlő rész.

Ennek figyelembevételével a számítógép sémája:

A vezérlő rész gondoskodik az egyes részek működésének összehangolásáról.



2. ábra

A műveleti egységben történik a műveletvégzés (aritmetikai, logikai). A decimális (tíz-es számrendszerben közölt) számok itt bináris (kettes számrendszerű) alakra kódolódnak.

A számítógépeknél kettes számrendszert alkalmaznak. A számok tárolása, a logikai műveletek igen és nem értékének ábrázolása megfelelő feszültség szint (impulzus) jelenlétével vagy hiányával egyszerűen megvalósítható.

Az operatív tárban egy-egy információt tárolunk. Az információ tárolásnál használt alapegység a bit.

8 bitet együtt kezelve 1 byte (bájt)-ról beszélünk. Egy byte-on 256 különböző variáció szerint tárolhatunk biteket. Így mód van arra, hogy egy-egy bájtban belül betűt vagy numerikus jelet (karaktert) egyaránt ábrázolhatunk. 1024 byte = 1 kbyte.

A mikroprocesszoros gépek 1, 4, 8, 16, 48, 64 kbyte kapacitásúak. A rekeszek sorszáma címek nevezük. Ennek segítségével lehet a megfelelő információhoz hozzájutni.

A táruk (memóriák) egyik csoportja csak kiolvasható (ROM). Ezekbe megfelelő technológiával a működéshez szükséges alapinformációkat gyárilag eleve betöltik.

A számítógépben levő (közvetlen hozzáférésű) táruk másik csoportja az amelybe a felhasználó tehát mi is beírhatunk információkat, adatokat, utasításokat, ezek a RAM-ok.

Számítsd ki, hogy egy 16 kbájtos RAM-ban vajon hány bit információ tárolható.

Egy ilyen nagy vonalakban megismert számítógéppel is igen sokféle feladat oldható meg.

Ilyen lehet:

- játék,
- számítás,
- ábrázolás, rajzolás,
- mérés,
- vezérlés.

Ehhez természetesen megfelelő szoftver azaz (a tárba betölthető v. betöltött, és a vezérlő egység által végrehajtandó) program szükséges.

(Folytatjuk)

Úttörők Mikro-klubja 2.

Dr. Árpád György Kulcsár János

Bekapcsoljuk a számítógépet

Bekapcsolás előtt valamennyi perifériát (külső memória, interface, kezelő egység, TV-vevő mint display stb.) rá kell csatlakoztatni a számítógépre, csak ezután szabad feszültséget adni a berendezésre!

Szét szerelésnél, illesztések esetleges igazításánál a berendezést először feszültségmentesíteni kell!

A számítógépet bekapcsolva a gép „bejelentkezik”, kiírja a kurzort, magyarázó szöveget (vagy csak reklámot). (A HT 1080Z READY?-vel jelentkezik.)

Ha a gép kurzorral jelentkezik, akkor máris kész arra, hogy végrehajtsa parancsainkat vagy utasításainkat (lásd később). Ha READY?-vel jelentkezik, nyomjuk meg a NEW LINE gombot. Ezzel tudomására hoztuk a gépnek, hogy készen vagyunk arra, hogy foglalkozzunk vele. (A READY angol szó, azt jelenti KÉSZ).

A NEW LINE beadása után a gép megjelenti a kurzort, jelezve ezzel azt, hogy ő is készen áll. (Az előző mondatban már egy kicsit úgy foglalmaztunk, mint a „nagyok”, azt mondtuk: beadtuk a NEW LINE-t. Ez a fogalmazás jobb annál, mintha azt mondanánk — lenyomtuk a NEW LINE-gombot. Ugyanis általában az utasítások, parancsok szavai nincsenek felírva a billentyűkre, betűnként kell bebillentyűzni őket.)

További nehézséget okoz (nem túl nagyot), hogy a parancsszavakat és utasításokat angolul kell beadni a gépnek.

A számítástechnikával foglalkozó szakemberek többféle programozási nyelvet dolgoztak ki, ezek közül számunkra érdekes a BASIC programozási nyelv. A BASIC betűszó: Beginners All purpose Symbolic Instruction Code. Magyarul: Kezdők általános célú szimbolikus utasítás-kódja.

Ennek a programnyelvnek az angol nyelvű utasításaival tudjuk a gépet utasítani a különböző feladatok végrehajtására.

Tudnunk kell azt, hogy a gép „belső” nyelve, az úgynevezett gépi nyelv egészen más szervezésű és az átlagember számára még kevésbé érthető, mint a BASIC.

Mielőtt tovább mennénk, tisztázni kell mit is nevezünk programnak. A számítógép számára a program egy előírt utasítási sorrend. A számítógép memóriájában tárolja az előírt utasításokat és azok sorrendjét. A program futtatása során a gép az előírt sorrendben végrehajtja az utasításokat. A számítógép ROM-jában levő program (fordítóprogram, interpreter) a gép részére érthető

formájúra fordítja a BASIC utasításainkat. A BASIC-nek vannak „tájszólásai”, de a BASIC alaputasításokat minden számítógép „megérti”.

Beszéltünk utasításokról és parancsokról. A legtöbb BASIC kulcszó használható a programon belül sorszámmal ellátva utasításként és ilyenkor a program futtatása során mindig elvégzi a gép az utasításban foglaltakat.

Ha nincs sorszám a BASIC kulcszó előtt, akkor ezt parancsnak hívjuk. Ilyenkor a gép végrehajtja a parancsra vonatkozó feladatot. De ezt csak egyszer teszi. Minden utasítás beírása, vagy a parancs közlése után, ha egy sort befejeztünk, mindig új sorba kell lépünk a New Line (Return) billentyű segítségével.

A program futásának indítását a RUN parancs beadásával lehet kezdeményezni (RUN: fuss). A futó programot a BREAK parancsral tudjuk leállítani. (BREAK: törés). A LIST parancs hatására a számítógép listázza a programtárolójában elhelyezett programokat. A játékprogramok között igen gyakran találkozhattok olyan programokkal amelyeknek a listázása le van tiltva. A LIST parancs hatástalan. Ezen ne lepődjete meg, nem a számítógép romlott el. A programok védelme miatt van erre szükség.

A programozás fő lépései:

- a feladat megfogalmazása,
- a feladat megértése,
- a megoldási menet (algoritmus) elkészítése,
- a BASIC-utasítás.

Az utasítások növekvő sorszáma szerint helyezkednek el a memóriában. Az utasítások végrehajtása ebben a sorrendben történik. Cél-

szerű a program utasításait tizedesével számolni, főleg akkor ha a gép nem tudja átszámolni a programcímeket. Félő, hogy kezdetben így is „beszorulunk”, mert időközben kiderül hogy még egy (meg még néhány) lépésre szükség van a program zavartalan futásához.

Mielőtt programozni kezdenénk ismerkedjünk a géppel, a billentyűzettel egyszerű parancsok (látványos vagy kevésbé látványos) végrehajtásával. A későbbiekben úgy sem „illik” a számítógépet sem csak gyári játékprogramok futtatására sem csak parancs üzemmódban számológépként használni. Ezekre a célokra szolgálnak a videojátékok, illetve a zbeszámológépek.

Print utasítás.

A „PRINT” (írd) írhatjuk sor számmal és sorszám nélkül (parancsként). Ez utóbbival kezdjük.

Billentyűzzük be a gépbe a következőket:

PRINT 3*5 — nyomjuk meg a New Line billentyűt, ekkor megkapjuk az eredményt: 15. (* a szorzás jele.)

Gyakorlásképpen végezzük el a következő feladatokat:

PRINT 4*(5+3) (New Line)

PRINT (5-7)/4 (New Line) (/ az osztás jele)

PRINT 5↑2 (New Line) (↑ a hatványozás jele, azaz 5²)

PRINT 5- (7+3) /4.2 (New Line) (. tizedespont)

A PRINT utasítással szöveget is nyomtathatunk.

Ha szöveget akarunk nyomtatni, azt idézőjelbe kell tennünk.

Ami idézőjelek között van, azt nem ellenőrzi a gép.

Pl.:

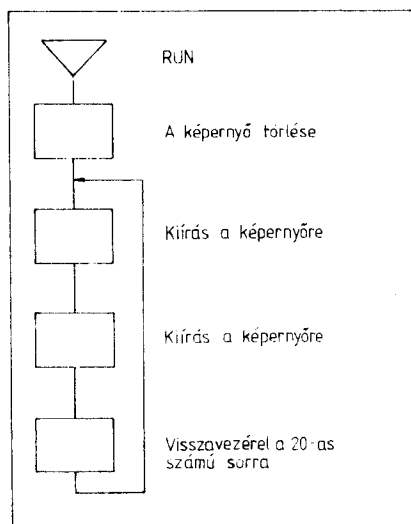
PRINT "ISKOLASZÁMÍTÓ-GÉP" (New Line)

PRINT "ÁLTALÁNOS ISKOLA" (New Line) (használjuk a szóköz-billentyűt!)

PRINT "EZ EDDIG SIKE-RÜLT" (New Line)

Ha eddig minden simán ment, akkor is tele van a képernyő, ha közben hibát vétettünk, akkor a számítógép „visszakérdez”, hibajelzést ad és nem hajtja végre a parancsot.

(Ne csüggedj, kezdetben igen sokszor fogja kiírni a gép a ? SN ERROR szintaktikai [helyesírási] hibára vonatkozó üzenetét. Ilyenkor lehetséges, hogy a 0 [nullát] és az O betűt jelentő billentyűt összekeverted, de ha a tizedespont helyett tizedesvesszőt írsz, akkor külön [Extra Ignored] is üzen a HT 1080.) Ha szeretnél újra tiszta képernyőre dolgozni, a CLS (töröld le a képernyőt) parancsot kell végrehajtani.



Amin láttuk, a gép akkor vesz tudomást az általunk bebillentyűzött parancsról, ha a New Line gombot is lenyomjuk. Mielőtt közölnénk a géppel a feladatot, parancsot, utasítást még egyszer nézzük végig a sort és végezzük el a szükséges javítást.

Géptípustól függ a javítás lehetősége, de a legegyszerűbb (ilyen rövid feladatoknál) ha a kurzor visszafelé mozgásával visszaradírozuk a sort a hibás részig.

A PRINT kulcsszó elé sorszámot is írhatunk.

10 PRINT "RÁDIÓTECHNIKA"
(New Line)

Ennek hatására nem íródik ki a megfelelő szöveg, csak ha a programot a RUN parancssal futtatjuk. Tehát:

10 PRINT "RÁDIÓTECHNIKA"
(New Line)
RUN (New Line)
A képernyőn megjelenik:
RÁDIÓTECHNIKA
READY?

Alkalmazzuk az eddigieket és írjuk úgy meg a programot, hogy

végrehajtás után csak a kívánt szöveg legyen a képernyőn.

10 CLS (New Line)
20 PRINT "NA LÁTOD?" (New Line)
RUN (New Line)

Ha ez a gép tényleg számítógép, akkor a múltkor megbeszélt (Neumann-féle) ismérveknek meg kell felelnie. Hogy is szól az 1. pont?

Ellenőrizzük le a gépet, megvan még a programunk?

Add be a listázz (LIST) parancsot. Ugye előadta a gép a programot? (Ha nem, akkor valamit nem jól csináltál. Kezdd előlről!)

Változtassunk a programon, írjuk be

30 PRINT "EZ A VÁLTOZÁS"
(New Line)
Futtassuk újra a programot
RUN (New Line)

Nézzük meg egy-két változtatást. Listáztatás után írjuk be

20 PRINT "NA LÁTOD?" (New Line)

A képernyőn most két 20-as sor van, de a gép az utoljára beírt 20-as sort jegyezte meg.

Futtassuk le ezt a programot, mit tapasztalsz?

Kísérletezzünk tovább, írjuk át a 20. sort.

20 PRINT "NA LÁTOD?". Most még csak így. Van módszer, amivel a javítást szövegszerkesztést (EDIT-álást) egyszerűbben is el lehet végezni.

RUN (New Line)

A látottak alapján jegyezzük meg

A ; az egymásutáni nyomtatás utasítása. Ha csak vesszőt teszünk, akkor 16 hellyel hátrább írja a gép a szöveget. A számítógép jellemzőinek 2. pontjából a program többszöri futtatását (futtathatóságát) már láttuk. Kíséreljük meg a program egy részének ismétlését.

Fűzzük a program végére a 40 GOTO 20 utasítást.

A GOTO jelentése: menj és folytatd ott.

Mielőtt kiadnánk a RUN parancsot gondoljuk végig mi fog történni. Készítsünk vázlatot a program folyamatáról (ábra).

Futtatás után BREAK gombbal tudjuk megállítani a szöveget a képernyőn.

(Folytatjuk)

APRÓHIRDETÉSEK

Lapunkba apróhirdetést a szerkesztőség vesz fel (1374 Bp., Pf. 603.), előzetes befizetés mellett. Ára szavanként 4 forint. Az első szó, valamint a nagybetűs szavak ára kétszeres. A keretes hirdetések díjtétele másfélszeres. Olvasóink a feladandó apróhirdetésük szövegét szerkesztőségünknek levélben küldhetik el, bélyeggel ellátott, saját névre megcímezett válaszborítékkal együtt. Mi ennek alapján megállapítjuk a hirdetés díját és csekket küldünk. A szerkesztőség készpénzt nem fogad el!

Nem hiányelk! Elhozott, hibás, komplett sorkimenő transzformátorok teljes újrakészítése, különleges impregnálással, beépítéskészre, elismert kítáló minőségben. GARANCIAVAL! Hálózati feszültség szabályozó zárlatvédelmi automatikával. Máté Imre mérnök, 1105 Budapest, Cserkesz u. 19. Telefon 9 óráig 770-410.

Műszaki kereskedésben híradástechnikai alkatrészek, tranzisztorok, IC-k, kondenzátorok, ellenállások kaphatók. Vidékre csomagot küldök 300 Ft felett. Érdeklődni telefonon: 14-861. Nagykanizsa, Petőfi 84. Schmidt Tibor.

COLOR TV-SHOP JAVÍTÁS—VÉTEL—ELADÁS

PAL-rendszerű készülékek SECAM-osztása kétnormásítás és javítása műszeres beállítással. Nyugati URH-vevők áthangolása, kétnormásítása.

UGYANITT

színes televíziók, rádiók, műszerek, személyi számítógépek és egyéb híradástechnikai berendezések, valamint tartozékok és alkatrészek vétele és eladása. Virág István, 1089 Budapest, Baross u. 121. 331-490.

Szovjet és színes TV-k, hangszórók, hangosító berendezések, CB-készülékek javítása, telepítése. Egyedi konstrukciók kialakítása. Generál Ip. Szöv. géblines részlege. Sándor Gábor, SZARVAS, Szabadság u. 28/2. Tel.: 116 (CB: próba 200, 32 USI).

Szolgáltatás levelezés útján! A Rádiótechnika-ban megjelent cikkeimhez új fejlesztési berendezéseimhez egységcsomagot, alkatrészeket, hasznos építési tanácsokat adok: RT 81/9.; képmintagenerátor AY-3—8500-zal, SD727B-modul; 81/1—3.; ritmusgenerátorok, 83/5.; szuperdobgép 2708-as EPROM-mal. Új fejlesztésű közlésre szánt témák: komplex dobgép, új hangszerszám +1 IC-s ritmus-rész egybeépítve. RGB, fekete-fehér képminta-generátor, programozható RAM-kártya, RF-videomodulátor, hangfrekvenciás vobbulátor XR 2206-tal, MINI effektus-szintetizátor SN 78477-tel, oktávemelő és egyéb érdekes zenélő témák, ZX—81 memória-bővítő 16, 32 K-ra. Kérjen részletes tájékoztatást. Címzett válaszborítékot kérek. Levél-cím: 2181 Bag, Puskin u. 99. A hirdetett készülékek mintapéldányai megtekinthetők, megvásárolhatók: TV-szerviz, Tápiószecső (MN Műv. háznál), munkanapokon 14—19-ig.

Gyakorolja otthon az elektronikát. Készítsen JANEL—PANEL elektronikus játékokat. Választható ellenében ismertetőt küld JANTHO ISTVÁN, 3459 IGRICI Áll. Gazdaság.

Eladó frekvenciámérő: 6 digit 50 MHz-es TTL IC-ekkel (elektronikus skála is), 8 digit 300 MHz-es CMOS+95H90, 8 digit 600 MHz-es 7216A+11C90 IC-ekkel, mind-egyik meleg termosztáttal. Cím: Hochrein András, 7623 Pécs, Kolozsvár u. 5/a. Tel.: 21-157.

AMATŐRBOLT

GMK-ki Amatőrök! Elektronikai alkatrészek, olcsó műszerek kaphatók.

Böngéssze!

Budapest IX. ker., Ráday u. 3.
A metró Kálvin téri megállójánál.
Ide mindig érdemes benézni.

VIDEO! SZÍNES TV! HI-FI!

Képmagnók, HI-FI berendezések, SZÍNES TV-KÉSZÜLÉKEK javítása és áthangolása; PAL—SECAM átalakítás; garantált minőség, természetes színviszonyok, 3 év garancia-vall! Színes TV-k, tunerok, URH-készülékek, rádiók, autórádiók (színtézeresek is) O—C kétnormásítás ki-fogástalan minőségben, a készülék megbontása nélkül!!!

CSAK NÁLUNK! Professzionális „O—C” kétnormásító egység HI-FI tunerokhoz autórádiókhoz (színtézeres berendezésekhez is), külön is megvásárolható!!! BEÉPÍTETT DIGITÁLIS ÓRÁK PONTOSÍTÁSA bármely berendezésben. Kivánságra speciális áramkörök megvalósítása! Forduljon hozzánk bizalommal! Zselér Gusztáv, XVIII., Vörös Hadserg 357. Tel.: 480-299.

Ugyanitt híradástechnikai alkatrészek, diódák, tranzisztorok, sorkimenők, elektroncsövek, nagyfeszültségű sokszorozók, hangszórók, modulok, tápegységek kaphatók. Kéri Márton: 480-299. Hétfő: zárva. K., Cs., P.: 9—18-ig. Sz.: 9—13-ig.

MICOLOR—01 színes nagyfelbontású grafika HT—1080Z típusú személyi számítógéphez. Átkapcsolhatóan használható 32 Kbyte memória-bővítésként vagy grafikaként. Felbontás 256×256 pont. Két képsík 2×4 szín-nel vagy 1 képsík 16 színnel, a színek 256 alapszín közül választhatók. Az egyik kép tetszés szerinti irányba mozgatható, a képsíkok tetszőlegesen takarhatják egymást. Az összes választási lehetőség programozható. Beépített tápegység. Fekete-fehér televízióhoz VIDEO, színes televízióhoz RGB kimenet. Ára iskoláknak 14 900 Ft. HT—1080Z számítógép sebességének megduplázása kapcsolóval. Ára: 1050 Ft. Megtekinthető és megrendelhető: Monori István villamosmérnök, elektronikai műszerész kisiparos, 1165 Budapest, Géza u. 15/b. Tel.: 838-225.

Rádió és TV-javításhoz használatos műszereket vennék! Régebbi típusú „roncs” állapotban lévő készülékek is számításhoz jöhetnek. Pl.: TV-miniszkóp, woblerszkóp, tv-tester, képmintagenerátor, szignálgenerátor stb. Ajánlatokat rövid ismertetéssel kérek. Varga Ferenc, 2403 Dunajváros, Pf. 17.

Úttörők Mikro-klubja 3.

Dr. Appel György, Lukács János

Értékkadó utasítások

A számítógép memóriájában adatokat tárolhatunk, amelyekre később — a program végrehajtása során — szükségünk lehet. A különböző feladatok során az adatok változhatnak. A BASIC programnyelvben az adatokat változóknak nevezzük még akkor is, ha azok egy konkrét esetben nem változnak. A változtatás lehetősége azonban megvan. A változókat az angol abc nagybetűs elemeivel jelölhetjük, A—Z-ig. Ilyen módon 26 egybetűs jelölést alkalmazhatunk. Használhatunk azonban kétbetűs jelölést is pl. AX, PE, TE, TA, BA stb. Ha két-től több betűt használunk fel a változó jelölésére, a BASIC akkor is csak az első két betűt jegyzi meg. Tehát a fenti példáknaál maradván a PE helyett írhatnánk PETER, PEST, PETI, PEREG stb. a TE helyett pedig TERA, TERI, TEMES stb.

Például a `PI=3.1415`, `New Line` parancsot a gép megérti és a `PI` nevű változóhoz hozzárendeli a 3.1415-ös értéket. Ha ezután rákérdezzük: `? PI` `New Line`. A gép válasza: 3.1415. Érdekesképpen kérdezzük meg a géptől: `?PI LIS` `New Line`. A gép válasza: 3.1415.

Ez utóbbi két példa kapcsán még azt is megtanultuk, hogy a gép elfogadja a `PRINT` helyett a kérdőjelet. Ez az adottsága megmarad program üzemmódban is. Írjuk be a következő kis példaprogramot:

```
10 PI=3.1415
20 ?"PI="; PI
```

Futtassuk le és listáztassuk ki!

A listáztatás után érdekes dolgot tapasztalunk. A képernyőn a 20 sorban a kérdőjel helyett a `PRINT` szót látjuk.

Az Ohm törvénye: $I = \frac{U}{R}$ alak-

ban felírt összefüggésben adott (de minden feladatnál más) feszültség és ellenállás értékéből akarjuk meghatározni az áramerősség értékét. Amikor papíron ceruzával megoldjuk a feladatot, tulajdonképpen akkor is egy programot hajtunk végre (meghatározott sorrendben elvégezzük a számítást). Próbáljuk meg szabatosan végigvezetni a feladatot a múlt alkalommal megbeszélt „programozás fő lépései” alapján. A feladat: kiszámítani és a végén aláhúzva kiírni az áramerősség értéket az $I = \frac{U}{R}$ összefüggés alapján legyen $U = 220$ V, legyen $R = 50$ Ω (ohm).

A megoldás menete:

Ismerem U értékét
Ismerem R értékét
Behelyettesítem a képletbe
Elvégzem a számítást
Kiírom az eredményt

Meg lehetne oldani máshogy? Lehetne más sorrend szerint dolgozni? Bizony nem.

Mielőtt megírnánk a BASIC programot, ismerkedjünk meg az értékkadó utasításával: `LET A=12`, azaz a továbbiakban legyen az A betűvel jelölt változó értéke 12. (A `LET` szöcskát nem szükséges kiírni.) Ezek után ebben a programban valahány-

szor egy (macskakörmökön kívüli) A betűvel találkozunk a gép, akkor annak értékével dolgozik. Próbáljuk ki. Kapcsoljuk be a gépet és parancs formájában közöljük:

<code>A=12</code>	<code>New Line</code>
<code>PRINT A</code>	<code>New Line</code>
<code>PRINT B</code>	<code>New Line</code>

`Print A`-ra a képernyőre kiírja a 12-es számot. `Print B`-re géptípustól függően kapunk választ. A `HT` ilyen esetben (ha nem adtunk értéket egy változónak) $\emptyset$ -t ír ki.

A Sinclair nem találja (és öntevékenyen nem értelmezi nullának) a változó értékét, ezért hibajelzést ad.

Állítsuk össze a programot, azaz a megoldás menetét fordítsuk le BASIC utasításokra.

```
10 LET U=220
20 LET R=50
30 LET I=U/R
40 PRINT I
```

Ezért kár volt bekapcsolni a számítógépet és még el sem fogadná a megoldást egy szigorúbb fizikatanár. Módosítsuk a 40. sort és az 50. programsort írjuk a program végére. (A mértékegységeket figyelembe véve!)

```
40 PRINT "I="; I; "A"
50 PRINT "-----"
```

Ha több különböző adatpárral akarjuk elvégezni a feladatot, akkor vagy minden futás előtt a kívánt új értékkel újra írjuk a 10. és 20. programsort (vagy eleve tároljuk az összes adatot a programban, de erről később), vagy kihasználjuk a



párbeszéd jellegű programszervezés lehetőségét.

Használjuk az INPUT utasítást: állj meg és várj addig, amíg a billentyűzet segítségével beírom a változó értékét.

Ezzel a programunk

```
10 INPUT U
20 INPUT R
30 LET I=U/R
40 PRINT "I="; "A"
50 PRINT "-----"
```

Szépséghibát most is találunk, hiszen ki emlékszik már arra, hogy a futtatás közben megjelenő kérdőjel után melyik változó értékét kell begépelni?

Mi most még tudjuk, tehát futtasuk a programot.

```
RUN New Line
? 220 New Line
? 50 New Line
I=4.4 A
```

Alakítsuk át a programot.

```
10 INPUT "U="; U
20 INPUT "R="; R
```

.

.

60 GOTO 10

Ennek segítségével sorozatosan oldhatjuk meg az Ohm törvényre vonatkozó feladatokat.

Készítsünk programot, melynek segítségével az egyenáramú hálózatra kapcsolt fogyasztó teljesítménye határozható meg a feszültség és ellenállás ismeretében.

Mi történik, ha számok helyett betűt írunk be amikor a gép számot vár? Kírja: REDO? és várja a számértéket.

Érdekes dolgot tapasztalhatunk, ha a híradástechnikában előforduló kis feszültség- és MΩ nagyságú ellenállás-értékeket írunk be, ilyenkor előfordulhat, hogy a túl kicsi számmal már nem tud a gép mit kezdeni. (Később majd megkeressük ennek a hibának kivédésére alkalmas lehetőséget.)

No és mi történik akkor, ha $R=0$ értékkel akarunk számítást végeztetni? A gép ilyenkor hibajelzéssel figyelmeztet arra, hogy 0-val akarunk osztani, márpedig azt nem lehet.

Gyakorló feladatként írjunk programot, melynek segítségével a vezeték keresztmetszete határozható meg. A bemenő adat — INPUT — a vezeték átmérője, melyet D-vel jelölünk, a kimenő adat — amit ki-nyomtatunk — a keresztmetszet, melyet A-val jelölünk.

Így a BASIC program:

```
10 INPUT D
20 A=D*D*3.14/4
30 ? A
40 GOTO 10
```

(A 40. sorban adjuk azt az utasítást, hogy térjen vissza a program elejére, ahol újabb D értéket adhatunk.)

A programot a RUN (fuss) utasítással futtatjuk. Ekkor a képernyőn egy ? jelenik meg. Ezután a D átmérő konkrét értékét kell beírni, pl.: 5-öt és az új sorban máris ott az eredmény: 19.625. Ha nem akarjuk tovább folytatni a program futtatását, akkor a BREAK billentyűvel leállítjuk a programot.

A programot érthetőbbé (elegánsabbá) tehetjük, ha PRINT utasításokkal magyarázó szöveget írunk ki.

```
10 PRINT "A VEZETEK
KERESZTMETSZETET"
20 PRINT "SZAMITJUK KI"
25 ?
```

```
30 PRINT "KEREM
AZ ATMERO"
40 PRINT "MEROSZAMAT
BEIRNI!"
```

```
50 INPUT D
60 PRINT "AZ ATMERO="; D
70 A=D*D*3.14/4
80 PRINT "A KERESZTMET
SZET="; A
```

```
85 ?
90 GOTO 30.
```

A futtatás (RUN) után a következő jelenik meg a képernyőn:

```
A VEZETEK
KERESZTMETSZETET
SZAMITJUK KI
```

```
KEREM AZ ATMERO
MEROSZAMAT BEIRNI!
? (itt beírunk pl.: 5-öt!)
Ekkor megjelenik
AZ ATMERO = 5
A KERESZTMETSZET = 19.625
Majd a GOTO 30 hatására ismét
megjelenik:
KEREM AZ ATMERO
MEROSZAMAT
```

. stb.

*

Megjegyzés:

25 és 85 sor egy PRINT utasítást tartalmaz csak. Ennek csak az a jelentősége, hogy egy sor üresen maradjon, így áttekinthetőbben olvashatjuk a képernyőre kerülő szöveget.

(Folytatjuk)

APRÓHIRDETÉSEK

Kedves Olvasóink! Feladandó apróhirdetésük szövegét levélben küldhetik el szerkesztőségünknek (1374 Budapest, Pf. 603.), bélyeggel ellátott, megcímzett válaszborítékkal együtt. A hirdetés díja szavanként 4 Ft. Az első szó, valamint a nagybetűs szavak ára kétszeres. A kereset hirdetések díjtétele másfélszeres. A hirdetési díj befizetésére csekket küldünk, amelyet kérünk befizetés után a szerkesztőségnek visszaküldeni.

Nem hlányelkk! Elhozott, hibás, komplett sorkimenő transzformátorok teljes újrakészítése, különleges impregnálással, beépítéskészre, elismert kitérő minőségben, GARANCIÁVAL! Hálózati feszültség szabályozó zárlatvédelmi automatikával. Máté Imre mérnök, 1105 Budapest, Cserkesz u. 19. Telefon 9 óráig 770-410.

Műszaki kereskedésben híradástechnikai alkatrészek, tranzistorok, IC-k, kondenzátorok ellenállások kaphatók. Vidékre csomagot küldök 300 Ft felett. Érdeklődni telefonon: 14-861. Nagykanizsa, Petőfi 84. Schmidt Tibor.

Szovjet és színes TV-k, hangszórók, hangosító berendezések, CB-készülékek javítása, telepítése. Egyedi konstrukciók kialakítása. Generál Ip. Szöv. gabines részlege, Sándor Gábor, SZARVAS, Szabadság u. 28/2. Tel.: 116 (CB: próba 200, 32 USB).

Gyakorolja otthon az elektrotechnikát. Készítsen JANEL-PANEL elektronikus játékokat. Válaszbélyeg ellenében ismertetőt küld JANTHO ISTVÁN, 3459 IGRICI Áll. Gazdaság.

HOBBY SHOP Elektromos barkácsbolt. Híradástechnikai alkatrészek, barkácsárak olcsó áron. VÉTELELADÁS. 1093 Bp. IX., Szamuely u. 39/b. T.: 377-212. 500 Ft felett vidékre utánvét. Nyitvatartás: H., K.: 9—18, Sz., Cs., P.: 10—18.

Antennaerősítők, konverterek készítése a távolsági TV- és URH-vétel kedvelőinek. Gyári antennaerősítők javítása. Sándor Antal, 1137 Bp., Katona József u. 5. Telefon: 313-475.

Rádó- és TV-javításhoz használatos műszereket vennék! Régebbi típusú „roncs” állapotban levő készülékek is számításba jöhetnek. Pl.: TV-miniszkop, woblerek, TV-tester, képmintagenerátor, szignálgenerátor stb. Ajánlatokat rövid ismertetéssel kérek. Varga Ferenc, 2403 Dunaújváros, Pf. 17.

Színes képesővek felújítása Japán regenerátorral — előzetes megbeszélés után. XVIII., Vöröshadsereg út 21. Televízió szervíz. Kedd, csütörtök, péntek 9—17-ig.

VIDEO! SZÍNES TV! HI-FI!

Képmagnók, HI-FI berendezések, SZÍNES TV-KÉSZÜLÉKEK javítása és áthangolása; PAL—SECAM átalakítás; garantált minőség, természetes színvisszaadás, 3 év garanciával!!! Színes TV-k, tunerok, URH-készülékek, rádiók, autórádiók (szintézerek is) O-C kétnormásítása kifogástalan minőségben, a készülék megbontása nélkül is!!!

CSAK NÁLUNK! Professzionális „O-C” kétnormásító egység HI-FI tunerokhoz, autórádiókhoz (szintézerek berendezésekhez is), külön is megvásárolható!!! BEÉPÍTETT DIGITÁLIS ÓRÁK PONTOSÍTÁSA! bármely berendezésben. Kívánságra speciális áramkörök megvalósítása! Forduljon hozzánk bizalommal! Zsellér Gusztáv, XVIII., Vörös Hadsereg 357. Tel.: 480-299.

Ugyanitt híradástechnikai alkatrészek, diódák, tranzistorok, sorkimenők, elektroncsövek, nagyfeszültségű sokszorozók, hangszórók, modulok, tápegységek kaphatók. Kéri Márton: 480-299. Hétfőn zárva, K., Cs., P.: 9—18-ig, Sz.: 9—13-ig.

AMATŐRBOLT! GMK-k! Amatőrök! Elektronikai alkatrészek, olcsó műszerek kaphatók! Bőngészd! Budapest IX. ker., Ráday u. 3. A métró Kálmán téri megállójánál. Ide mindig érdemes benézni!

Úttörők Mikro-klubja 4.

Dr. Appel György, Lukács János

Feltételes elágazás

Egy már majdnem logikai műveletet végző utasítással ismerkedünk meg.

A feladatok megoldása során olyan esetek is vannak, ahol a képlet kiszámításának valamilyen feltétele van. Ilyen például az, amikor az $\frac{1}{x}$ függvényt kell meghatározni. Ebben az esetben az x értéke nem lehet 0, mert amint tanultuk, nullával nem lehet osztani.

Mondhatnád most, hiszen én adom meg milyen számokkal dolgozzon a gép, én tudom, hogy 0-val nem lehet osztani, majd x értékét nem veszem 0-nak. Ebben igazad is van de a nevezőben lehet bonyolultabb kifejezés is, amelynek az alakulását nem láthatod előre pl.:

$\frac{1}{A-B-X}$. Ha ebben az esetben A

és B egy előző számítás eredménye és az INPUT-ban csak X szerepel, már nem tudod kézben tartani a nevező értékének alakulását.

Amikor a nevező értéke nulla és a gép elvégzi az osztást, akkor a képernyőn ezt fogod látni: ?/0 ERROR IN... Vagyis nulla a nevezőben hiba a ... sorban. Az IN szócska után a pontok helyén a megfelelő BASIC-sorszámot látod.

A fenti és a többi más hibát is speciális utasításokkal meg lehet „fogni”. Mi most nem foglalkozunk ezekkel a speciális „hibaelfogó” BASIC-utasításokkal.

Megtanulunk egy másik BASIC-utasítást, amely amellelt, hogy egy hiba felderítésére alkalmas, más

hasznos és fontos feladatokat tudunk vele megoldani: egy feltételnek a teljesülésétől vagy nem teljesülésétől tehetjük függővé, hogy a programunk a továbbiakban merre ágazzon el.

Erre alkalmas az:

IF [feltétel] THEN [utasítás v. sorszám]
utasítás.

(IF jelentése: ha; THEN jelentése: akkor.)

Az IF utáni zárójelben szerepelhet minden olyan feltétel, amely BASIC nyelven megadható. Ha a feltétel teljesül a gép a THEN után következő utasítást végrehajtja, illetve az IF-THEN-t tartalmazó sor utáni legalacsonyabb BASIC-sorszámú sorra lép.

Ha a feltétel nem teljesül, a program az IF-THEN utasítást tartalmazó sort követő legalacsonyabb BASIC-sorszámú sorra lép.

Az IF-THEN utasításnak a teljes formátuma: **IF [feltétel] THEN [utasítás, v. sorszám] ELSE [utasítás v. sorszám]**.

Ebben a változatban az utasítás a feltétel teljesülésekor ugyanúgy működik, mint ahogyan már azt láttuk a csontkitöltött változat (IF-THEN) esetében.

Ha a feltétel nem teljesül, akkor az ELSE után megadott utasítást hajtja végre illetve az itt megadott sorszámra ágazik el a program futása.

Az alább megadott példaprogramban tanulmányozhatod az IF-THEN működését. A program az egyenesvonalú egyenletes mozgást végző test sebességét számítja ki: az út jele S, az idő jele T és a sebes-

séget V-vel jelöljük. Az adatokat a fent megadott nevű változóknak tároljuk. Arra figyelj, hogy a programlistában ? jellel helyettesítettük a PRINT szócskát. Arról már tettünk említést, hogy a gép elfogadja ezt a formát is. Amikor a programot listáztatod, a ? jelek helyett mindenhol PRINT fog szerepelni. Tehát ez a lehetőség a beírást könnyíti meg (nem kell a sok PRINT-et bepötyögtetni).

A programban a nevező nullává válásának vizsgálata (tesztelése), a nullával való osztás elkerülése egyben a kezelőt figyelmeztető szöveg kiíratására történő vezérlés (ugratás, elágaztatás) a 80. sorban valósul meg. Figyeljétek meg, hogy a T (idő) beadása után és meg az osztás elvégzése előtt történik a beavatkozás. Ha a $T=0$, a program futása a 120. sornál folytatódik, tehát átugorja a közbenső sorokat és nem fog osztani nullával (100. sor S/T). A megfelelő szöveg kiíratása után az 50. sorra kerül a program végrehajtása. Ott ismét kéri az idő mérőszámát.

Az út mérőszámát megadtuk már és ezt az értéket az S változó tartalmazza. Hogy tudnánk erről meggyőződni anélkül, hogy a program futását megbolygatnánk? A következő módon: nyomd meg a **BREAK** billentyűt, utána: ?, S, **New Line**. A képernyőn láthatod: **BREAK IN 60**

READY

? S

... (az út mérőszáma)

READY

A program további változatlan futását a **CONT** utasítás beadásával érhetjük el. Írjuk be **CONT New Line**. A képernyőn megjelenik a kérdőjel (?), a program visszatért a 60. sorba.

A példaprogramnál nem kerek egész órák esetében az idő mérőszámát óra, tized óra formában kell beírni. Pl. három és fél óra = 3,5 és nem a megszokott módon: 3 óra 30 perc.

5 CLS

10 ? "SEBESSEG SZAMITASA"

15 ?

20 ? "AZ UT MEROSZAMA

(KM)"

30 INPUTS

40 ? "AZ UT = "; S; "KM"

45 ?

50 ? "AZ IDO MEROSZAMA

(ORA)"

60 INPUTT

70 ?

80 IFT=0THEN 120

100 ? "A SEBESSEG = "; S/T;

"KM/ORA"

102 ? " — — — 27 darab — — "

104 ?

110 GOTO20

120 ? "AZ IDO SZAMERTEKE

NEM LEHET"

122 ? "NULLA! UJ IDO ADATOT

KEREK"

130 GOTO50

140 END

(Folytatjuk)

(Folytatás az 56. oldalról)

Megjegyzések az adótáblázatokhoz

(L. az 56. oldalon)

Komplett — minden hazai műsorszóró adónkra kiterjedő — adótáblázatokat már évek óta nem közzétűnk lapunkban. A hiányt pótolva most olvasóink kívánságának megfelelően az 1—4. táblázatokban nemcsak a TV-adóállomások, hanem az URH- és AM-rádióadók főbb adatait is leközöljük. A lapban megjelent utolsó komplett adótáblázattal — 1979/12., III. 1980/2. szám — összevetve látható, hogy azóta igen sok változás történt: örvendetes mértékben nőtt műsorszóró hálózatunk, egyre több helyen biztosítva a jobb vételi körülményeket.

Néhány megjegyzés a táblázatokhoz. A közölt ERP teljesítményadatok (effektív kisugárzott teljesítmény) névleges értékek, becsült, sok esetben nem pontos adatok. A zárójelben levő számok még nem működő, tervezett adókra vonatkoznak. Az URH-rádióadók nem mindig az 1. táblázatban szereplő műsort sugározzák, ha az adott időpontban éppen regionális műsor van. A miskolci stúdió műsora alatt a miskolci, kékesi és tokaji, egyébként Kossuth műsort sugárzó URH-adók a miskolci műsort, a pécsi és nagykanizsai

Kossuth URH-adók pedig a pécsi helyi stúdió műsorát közvetítik ennek műsoridejében. Hasonló a helyzet a soproni és Kab-hegyi Kossuth URH-adókkal, valamint a 3. műsor győri URH-adójával is: helyi műsoridőben átveszik a győri stúdió műsorának sugárzását.

4. táblázat. Külföldre szülő AM-rádióadások frekvenciái

Frekvencia (MHz)	Névleges hullámhossz (m)
6,025	49,79
6,035	49,71
6,110	49,10
7,220	41,55
7,225	41,52
9,520	31,51
9,585	31,30
9,835	30,50
11,910	25,19
12,000	25,00
15,160	19,79
15,220	19,71
17,710	16,94
21,665	13,85

Úttörők Mikro-klubja 5.

Dr. Appel György, Lukács János

Feltételes elágazás (folytatás)

Térjünk vissza arra a próbálkozásunkra, amikor adatra várva betűvel akartuk a gépet „megetetni”. Ha olyan jellegű feladat adódik, melynél betűk, nevek lehetnek a változók, ott tudatni kell a géppel, hogy INPUT-ként betűket (karaktereket) fogadjon el.

Azokat a változókat, amelyek betűket tartalmaznak, stringváltozónak nevezzük, jelölésük A\$, B\$...Z\$. Készítsünk programot, mely megkérdezi a nevedet, azt közölve eltörli a képernyőt és megszólítással üdvözlő szöveget ír ki.

```
10 PRINT "ÉN A SZÁMÍTÓGÉP  
VAGYOK"  
20 INPUT "A TE NEVED"; N$  
30 CLS  
40 PRINT "SZERVUSZ"; N$  
50 PRINT  
60 GOTO 10
```

Próbáljuk ki azt is, hogy most betűk helyett számokat írunk be. Ezt elfogadja a gép. (Ennek még később hasznát vehetjük.)

A string ugyanis nemcsak betűkből, hanem karakterekből, így számokból és rajzjelekből állhat. A betűkkel, stringekkel összehasonlító műveleteket (<, >, =), sőt összeadást is végezhetünk. Készítsünk programot a nagyobb szám keresési program alapján, mely névsorba írja ki a két egymás után beírt nevet.

```
10 PRINT "NÉVSOR"  
20 PRINT "KÉREM AZ EGYIK  
NEVET"  
30 INPUT A$  
40 PRINT "KÉREM A MÁSIK  
NEVET"  
50 INPUT B$  
60 PRINT  
70 IF A$ > B$ THEN PRINT B$,  
A$ ELSE PRINT A$, B$  
80 PRINT "xxxxxxxxxxxxxxxxx"  
90 GOTO 20
```

Módosítsuk úgy a programot, hogy tudjuk, hányadik feladatot (névsorba rendezést) végzi.

```
10 PRINT "NÉVSOR"  
20 LET N = 1  
30 PRINT "KÉREM AZ EGYIK  
NEVET"  
40 INPUT A$  
50 PRINT "KÉREM A MÁSIK  
NEVET"  
60 INPUT B$  
70 PRINT  
80 PRINT N; "FELADAT"  
90 IF A$ < B$ THEN PRINT B$,  
A$ ELSE PRINT A$, B$  
100 PRINT "xxxxxxxxxxxxxxxxx"  
110 LET N = N + 1  
120 GOTO 30
```



A 110-es lépést eddig tanult matematikai tudásunk szerint nem szabad értelmezni, ez BASIC-ül van. Magyarul így mondhatnánk: legyen a továbbiakban N változó értéke eggyel nagyobb, mint eddig volt.

Hogyan kellene átírni a 120-as sort, ha azt akarjuk, hogy a program futása csak nyolcszor ismétlődjen?

Készítsünk programot, melynek segítségével A, B, C, D, E, F betűk valamelyikének lenyomására a gép kiírja a betűt és mellé annak morse jelét.

```
10 INPUT A$  
20 PRINT A$  
30 IF A$ = "A" THEN PRINT "  
.."  
40 IF A$ = "B" THEN PRINT "  
...."  
50 IF A$ = "C" THEN PRINT "  
...."
```

```
60 IF A$ = "D" THEN PRINT "  
..."  
70 IF A$ = "E" THEN PRINT "  
.."  
80 IF A$ = "F" THEN PRINT "  
...."  
90 GOTO 10
```

Azért ezt a módszert ne folytassuk a teljes ABC-re, mert idő előtt beleunhatunk a programozásba. Amúgy is lesz elegánsabb lehetőségünk.

Ha a kiíratást a képernyő adott helyére kívánjuk elhelyezni, akkor ismerni kell a képernyő beosztását. HT 1080Z 16 sorban 64 oszlopban 1024 karaktert tud elhelyezni. Az adott helytől való kiíratáshoz a PRINT @ utasítást használhatjuk (@-t mi nevezzük „kukacnak”). Az ez után írt szám mutatja a kezdő pozíciót.

Hogyan? A képernyő bal felső sorának első pontja a 0 pont, a második 1, a harmadik 2, s így tovább...

A @ jel közvetlenül a PRINT utasítás után következik, és az utána írt szám értékének 0–1023 között kell lenni. Így feladatunk igen egyszerűen így oldható meg:

```
10 PRINT @ 330, "SZÁMÍTÁS-  
TECHNIKA"
```

Ha egy sorban nem a sor elején szeretnénk kezdeni, akkor a TAB utasítást használhatjuk, pl.

```
10 PRINT TAB (10) "SZÁMÍTÁS-  
TECHNIKA"
```

hatására az első sor 10. helyétől kezdve íródik ki a szó.

Készítsünk programot, melynek segítségével a bal felső sarokból indulva átlósan lefelé kiírhatjuk a RÁDIÓ-TECHNIKA szót.

(Folytatjuk)



Úttörők Mikro-klubja 6.

Dr. Appel György, Dr. Kulcsár János

Ciklusok

A számítógéppel megoldandó feladatok zömében bizonyos számításokat többször el kell végezni. Esetenként — pl. rajzoláskor — pontosan kell kiírni egymás után.

Azoknál a feladatoknál, melyeknél az egymás után elvégzendő számításokat más-más (de számtani sort alkotó) változó értékkel kell elvégezni, érdemes ciklust szervezni.

A ciklus lefutása a FOR-NEXT utasítás hatására történik. A ciklus elején a FOR I = kezdő érték TO végérték alakú utasítással közöljük a programmal, hogy a következőkben egymás után milyen értékei legyenek az (itt most I-vel jelölt) ciklusváltozónak. A ciklus végére NEXT I utasítást kell írunk. Ha a cikluskezdő utasítást pl. FOR I = 1 TO 10 alakúra írjuk, akkor tulajdonképpen azt is közöltük a géppel, hogy hányszor ismételve meg a ciklust.

Írjunk egy rövid programot, mely a ciklusutasítások alkalmazásával egymás után tízszer kiírja a nevedet:

```
10 CLS
20 FOR I=1 TO 10
30 PRINT "SAMU"
40 NEXT I
```

módosítsuk úgy a ciklus magját — azt az utasítást (utasításokat), melyet többször végére kell hajtani — hogy a kiírt szöveg elé írja azt is ki, hogy mennyi az I értéke.

```
30 PRINT I; "SAMU"
```

Tovább alakítva a programot, mindig egy hellyel beljebb írássuk ki a nevet

```
30 PRINT TAB (I) "SAMU"
```

Itt már felhasználtuk az I változó, rendre más, számszerű értékét.

Hasonlóan, de most már kimondottan táblázat készítés érdekében használjuk fel a ciklusról megismerteket.

Írja ki a programunk, 1-től 15 MHz hullámhossz értékek mellé a hullámhosszt m-ben.

```
10 CLS
20 FOR F=1 TO 15
30 PRINT F; "MHz", 200
  F; "m"
40 NEXT F
```

Vannak esetek, amikor a ciklusváltozó értékének egyesével való növelése túl nagy (vagy túl kicsi). Olyankor a (ha az egymás utáni értékek nem eggyel növekszenek) a ciklust kezdő utasítás sorában a lépésközt is meg kell adni.

```
pl. FOR I = 1 TO 10 STEP 0,5
```

```
vagy FOR I = 99 TO 1 STEP -2
```

Készítsünk programot, mely táblázatos alakban kiírja, hogy 220 V-os hálózatra kötött 20, 30, 40...200 Ω -os ellenállást milyen áramerősség terheli, s ez mekkora teljesítményt jelent.

```
10 CLS
20 ? "U (V)", "R (ohm)", "I/A/",
  "P(W)": (Ez a táblázat fejléce.)
30 U = 220
40 FOR R = 20 TO 200 STEP 20
50 I = U/R: P = U*I
60 ? U, R, I, P
70 NEXT R
```

Igen szép — elegáns — megoldásokat tesz lehetővé a ciklus alkalmazása programjainkban. Vannak azonban szabályok, melyeket be kell tartani, mert ellenkező esetben vad dolgokat képes művelni a program.

Lehet, hogy a hibajelzést egész más helyen adja ki a gép, mint ahol azt elkövettük, így azután igen nehéz lesz a hibát okozó sort (sorokat) megtalálni. Nem szabad ciklusba beugrani, kiugrani sem nagyon illik, — azaz a ciklus magjában levő utasításhoz program futása közben csak a FOR utasításon keresztül szabad eljutni. Véletlenül se adjunk a FOR utasításban adott ciklusváltozó értéknek a cikluson belül új értéket.

A ciklus befejezése, a NEXT utasítás után, a ciklusváltozónak a záróértéknél lépésközzel nagyobb értéke lesz!

Nézzük meg

```
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
40 PRINT "ELHAGYTUK A
  CIKlust MOST I="; I
```

Igyekezzél kiismerni a ciklusszervezéssel kapcsolatban mindenféle lehetőséget. Hiszen mint a sorozat elején tisztáztuk, a számítógép (töb-bek között) attól számítógép, hogy ciklusszervezés lehetséges. Az igazi szépségek ott kezdődnek, amikor a cikluson belül újabb (meg még újabb) ciklust szervezünk.

Számítógép-típustól függ, hogy hány ciklust lehet egymásba ágyazni (skatulyázni), jelenleg ezzel nem kell törődnünk. „Kedvenc” időtöltésed volt már az alsó tagozatban is, mikor matematikából fel kellett írnod, hogy a piros, fehér, zöld színeket hányféle módon lehet zászolóvá összeválogatni. A próbálgatás időrabló, a megoldást számszerűen megadó képletet pedig nem ismered, a szabályt, a törvényszerűséget azonban könnyen megfogalmazhatjuk. Írjuk fel az 1-es és 2-es számjegyek összes lehetséges felírási módját, ha mindig mindkét számjegyet felhasználjuk. Itt vegyük sorra a programozás alapszabályát:

1. a feladat megfogalmazása
2. a feladat megértése
3. a megoldás menete
4. programírás



Számítógépes szakköri foglalkozás a Kvassay Jenő Ipari Szakközépiskolában (Fotó: Harsányi)

A feladatot megkaptuk, ha szabatos a megfogalmazás meg is lehet érteni. Mi lesz a megoldás menete?

Először veszem az 1-es számjegyet és mellé írom a második szám lehetséges változatait. Majd veszem az első szám következő (NEXT!!) értékét és mellé is kiíratom a második szám lehetséges értékeit.

```
10 FOR E=1 TO 2 (mert az első helyen álló számnak két értéke lehet)
20 FOR M=1 TO 2
30 PRINT E; M
40 NEXT M
50 NEXT E
```

Vigyázat! Itt a két ciklus egymásba skatulyázása! Több ciklus esetén is be kell tartani ezt a szabályt!

Rajzban talán egyszerűbb megérteni.

```
FOR I =
FOR K =
FOR L =
NEXT L
NEXT K
NEXT I
```

Természetesen egy ciklusba több ciklus is ágyazható.

```
FOR I =
FOR K =
NEXT K
FOR L =
NEXT L
NEXT I
```

Nézzünk meg egy feladatot:

Rajzoljunk 10 sorban egymás alá csillagot úgy, hogy az első sorban

egy, a másodikban kettő és így tovább, legyen.

```
10 CLS
20 FOR F=1 TO 10
30 FOR V=1 TO F
40 PRINT " * ";
50 NEXT V
60 PRINT
70 NEXT F
```

A program végrehajtását jó lenne lelassítani. A gépet ilyenkor arra kell utasítani, amit a hirtelen válaszáadás ellen javasolnak. Számolj magadban tízig! Építsünk be az előző programba egy „üres” ciklust. Ez nem csinál mást csak „magában számol”, de ezzel múlik az idő.

```
45 FOR D=1 TO 50:NEXT
```

Itt és most alkalom kínálkozik arra, hogy válaszoljunk arra a kérdésre, hogy milyen gyorsan dolgozik a gépünk. Hiszen ha nem csinál semmit „csak” magában számol, akkor is nagyon lassúnak tűnik a működése, pedig a sorozat elején másodpercenként sok ezres műveleti sebességről is beszéltünk. Ez igaz, de a programunkat BASIC nyelven írjuk. Az interpreter a kulcsszavakat értelmezi és a gép számára érthetővé teszi, a gép pedig végrehajtja a gépi kódú utasítások sorozatát, majd ilyen, vagy olyan módon kijelzi az eredményt. Közben a gép sok más egyéb mellett figyeli azt is, hogy a billentyűzetről nem érkezik-e például BREAK parancs, állandóan frissíti a TV képet és memóriát is a HT bővített BASIC-ben, még a

program futása közbeni billentyűlenyomást is vizsgálja.

A HT számítógépben a vezérlő kristály 10,644 MHz-es leosztások után a központi vezérlőegység 1,77 MHz-es órajel-frekvenciát ad. Így a processzor egy olyan összeadást, melynek összege kisebb, mint 255 2~2,5 µs alatt végez el.

Azt, hogy milyen gyorsan végzi el az összeadások sorozatát, azt a matematika történetéből ismert feladvány megoldásával vizsgáljuk. mérjük meg. Készítsünk egy programot, mely 1-től 100-ig összeadja a számokat. Ilyen feladatot kapott 1786-ban Gauss, mint kisiskolás. A számítás Gauss percek alatt elvégezte, igaz neki is a számítás menetének megtalálása és nem a tényleges számolás vette el az idejét. Az itt közölt program nem az ő zseniálisnak számító megoldási menetére, hanem a hagyományos elképzelésre épül. (Ez a hosszadalmasabb!)

```
10 SUMMA=0
20 FOR I=1 TO 100
30 SUMMA=SUMMA+I
40 NEXT I
50 PRINT SUMMA
```

Futtasd újra a programot és mérd meg a futás idejét. Gyorsabbá tehetjük a program futását, ha előre kikötjük (és itt most meg is tehetjük), hogy a számokat egész számként kezelje, valamint ha összevonjuk egy programsorrá amit érdemes.

```
10 DEFINT I,S:S=0
20 FOR I=1 TO 100:S=S+I:NEXT
30 PRINTS
```

(Folytatjuk)

Új nemzetközi verseny...

CQ World-Wide VHF WPX

Időpontja 1985-ben: július 20-án 00-tól 21-én 24 UT-ig. (48 óra).

Sávok: valamennyi engedélyezett amatőrsáv 50-től 1296 MHz-ig.

Üzemmodok: bármely engedélyezett üzemmód használható, de az átjátszókon és a műholdakon keresztül létesített összeköttetések a versenyben érvénytelenek.

Egyszerűsített versenyzési mód: a hívójel mellett csak a folyamatosan növekvő sorszámot kell adni.

Pontozás: minden egyes összeköttetés 144 MHz-ig 1—1 pont, 144-től 432 MHz-ig 2-2 pont, 1296 MHz-en 4 pont. Ugyanazon állomással ugyanazon sávon csak 1 QSO érvényes, tekintet nélkül az üzemmódokra.

Szorzók: a sávonként elért prefixek, a WPX-diploma szabályai szerint. Például: HG1, HG2, YU1, YZ1, YT3, 4N4, I1 stb., de pl. 14LCK/2 szorzóként 12-nek vagy OK3CDM/HA5 szorzóként HA5-nek számít, IK2XYZ/6=IK6 stb.

A végeredmény számítása: a pontok összege szorozva a sávonkénti prefixek összegével.

Kategóriák: egykezelősök 1-sávon, egykezelősök össz-sávon, egykezelősök 1-sávon vagy össz-sávon, de QRP-adóval (max. 25 W p. e. p. kimenő teljesítményűvel), többkezelősök egy-, vagy többsávon, ugyan-ezek kitépülve, vagy FM üzemmódban.

Díjazás: tekintettel az első világversenynél alkalmára, valamennyi résztvevő, aki jegyzőkönyvet küld, emléklapot fog kapni a rendezőktől (a CQ Magazin versenyzőbizottságától), de oklevéllel jutalmaznak kategóriánként minden országost is. — F. J. —

Új TV-adó Komádiban

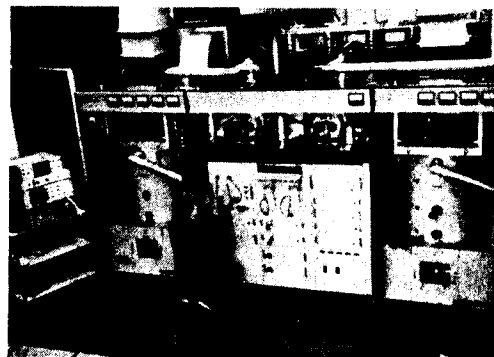
Május 13-án Beez Sándor, a Posta Rádió és Televízióműszaki Igazgatóság vezetője házilapú ünnepség keretében adta át a NEC gyártmányú japán adóberendezést a komádi TV-állomás kollektívájának. Mint arról postal híreinkben is olvashatunk, a legkorszerűbb, léghűtési 10/1 kW-os adóberendezés a hozzá tartozó Rohde-Schwarz adó-antennarendszerrel Hajdú-Bihar, Békés és Szolnok megyék területén javítja a Magyar Televízió második műsorának vételi lehetőségét. Az adó telepítésével egyidőben rekonstrukcióra is sor került Komádiban.

Új mikrohullámú berendezéseket adtak át, közvetlen 20 kV-os távvezetékkel létesítettek, új szükségáramfejlesztők, erősáramú kapcsolótér került beszerelésre. A modulációs ellátás javítása érdekében Földesen új mikrohullámú állomás épült, mely teljesen automatikus üzemi.

A Posta a közeljövőben korszerűsíti az állomás első műsorát sugárzó antennarendszert, illetve URH tartalékadókat telepít. Az 1. műsor 17 éve sugárzó adójának cseréjét a VII. öt éves terv keretében tervezik.

A 32. csatornán vízszintes polarizálással sugárzó kettős adóberendezés három hónapos próbajelzése alatt már sugározza a második műsort.

P. I.



Az új, az országban első léghűtéses japán NEC adó

Úttörők Mikro-klubja 7.

Dr. Appel György, Dr. Lukács János

Rajz

A számítógép segítségével karekteteket (betűket, jeleket, számokat) írathatunk a képernyőre, de alkalmas rajz készítésére is. A HT 1080Z 128×48 grafikus rajzolósi egységből álló felbontóképességgel rendelkezik. A koordinátarendszer kezdőpontja géptípustól függően a bal alsó vagy felső sarokban található (HT-nál a bal felsőben). Egy képernyőpont megjelenítésére a HT gépen a SET (X, Y) utasítás szolgál. Mód van arra, hogy ezt a pontot „eltüntessük”. Erre a RESET (X, Y) utasítás szolgál. A felbontóképesség adta korlátok között különböző összefüggéseket, azok függvényképét ábrázolhatjuk koordinátarendszerben.

Először rajzoljuk meg a koordinátatengelyeket. A program elkészítéséhez a következő gondolatmenetet kövessük:

A tengelyek egyenes szakaszok.

Az egyenes egymásután elhelyezkedő pontokból épül fel (SET).

Az egyenest a képernyő egyik szélétől a másikig (FOR NEXT) akarjuk megrajzolni.

A program tehát:

```
10 CLS
20 FOR X=0 TO 63
30 SET (X, 0)
40 NEXT X
```

Vannak gépek, BASIC nyelv-járások, melyek lehetővé teszik, hogy egyetlen utasítás (kulcsszó) segítségével vonalat, kört rajzolhassunk. A szerényebb grafikus adottságokkal rendelkező gépeken is rajzolhatunk kört, ellipszist, szinuszgörbét. Ezek rajzolásához az általános iskolai matematika anyagnál többre lenne szükség. Külön magyarázat nélkül egy-két összefüggést találhatsz itt, melyek beírásával különböző görbe vonalakkal rajzolt idomot rajzolhatsz.

```
10 REM KOR
20 CLS : SET (32,20)
30 FOR I=0 TO 6.3 STEP 0.4
40 X=32+20 * COS (I)
50 Y=22+20 * SIN (I)
60 SET (X, Y)
70 NEXT I
```

```
10 REM ELLIPSZIS
20 CLS : SET (32,20)
30 FOR I=0 TO 6.3 STEP 0.3
40 X=32+30 * COS (I)
50 Y=22+15 * SIN (I)
60 SET (X, Y)
70 NEXT I
```

```
10 REM SZINUSZGORBE
20 CLS
30 FOR I=0 TO 6.3 STEP 0.2
40 SET (I * 10,22 - 20 * SIN (I))
50 NEXT
```

A szinuszgörbe rajzolásánál láthattuk, hogy a SET utasításban a pont koordinátáit az ábrázolni kívánt függvény összefüggésével (képletével) is megadhatjuk. Készítsük el azt a programot, melynek segítségével ábrázolható a múltkor táblázatos alakban felírt összefüggés. A 220 V-os hálózatra, akkor különböző ellenállásokat kötöttünk és meghatároztuk az átfolyó áram erősségét. Alakítsuk át a programot úgy, hogy ne kiírja az eredményt, hanem az eredménnyel arányos helyre rajzolja be a pontokat.

```
10 CLS
20 U=220
30 FOR R=20 TO 200
40 SET (R/3, 44 - U/R)
50 NEXT R
60 GOTO 60
```

A 60. sorra azért van szükség, hogy a program lefutása végén ne írja ki a READY feliratot, amely elrontaná az ábránkat.

A szép, könnyen áttekinthető képernyő és megoldás kedvéért persze a programot tovább kell alakítani. Így meg kell rajzolni a koordináta tengelyeket:

```
22 FOR X=960 TO 991 : PRINT
  ÉX, "—" : NEXT X
24 FOR Y=64 TO 896 STEP 64 :
  PRINT ÉY, "I" : NEXT Y
26 PRINT É1, "I" : PRINT É
  990, "R" :
28 PRINT É80, "U=" ; U ;
```

Ennyi rajz után már jó lenne radirozni. Erre az RESET utasítás szolgál.

Először az egész képernyőt írjuk tele pontokkal. Ezt érdemes — a ciklusok egymásba ágyazásának gyakorlása érdekében — az alábbi programmal megvalósítani.

```
10 FOR X=0 TO 63
20 FOR Y=0 TO 47
30 SET (X, Y)
40 NEXT Y
50 NEXT X
```

Erre a „negatív” alapra rajzolva semmit sem látnánk, ezért alkalmazzuk a RESET utasítást és húzzunk egy ferde vonalat a képernyőre.

```
60 FOR X=0 TO 45
70 RESET (X, Y)
80 NEXT
100 GOTO 100
```

Ha egy pontot mozgatni akarunk a képernyőn, akkor a pont új helyre való rajzolása előtt az előző koordinátákra rajzolt helyről ki kell azt radirozni. Nézzük ezt programban. Mozogjon a pont ferdén lefelé a képernyőn.

```
10 CLS : X ELŐZŐ = 0 : Y
  ELŐZŐ = 0
20 FOR X = 1 TO 45
30 RESET (X ELŐZŐ, Y ELŐZŐ)
```

```
40 SET (X, Y)
50 X ELŐZŐ = X : Y ELŐZŐ =
  X : (mert 45 fokban mozog.)
60 NEXT
```

Ha túl gyorsnak találjuk a mozgást, akkor — mint már ismerjük ezt a megoldást — írjunk be egy üres ciklust a rajzó ciklusba.

```
60 FOR T = 0 TO 30 : NEXT T
```

Annak érdekében, hogy újabb és újabb golyó (labda) induljon el a felső sarokból küldjük vissza a programot a ciklus elejére.

```
80 GOTO 20
```

Kísérletezz, cseréld fel a 30 és 40-es programsort. Mit tapasztal?

A képernyő grafikus lehetőségeinek megismerése érdekében, a játék kedvéért készítsünk programot, amelynél a labda a felső sarokból indulva a 32. sorig mozogjon lefelé 45° alatt, innen pedig mintha visszazapattant volna a 33. sorra rajzolt falról mozogjon 45° alatt lefelé.

Először a „falat” rajzoljuk meg.

```
10 CLS : XE = 0 : YE = 0
20 FOR X = 20 TO 40
30 SET (X,33)
40 NEXT
```

A pont az előző program szerint mozogjon lefelé.

```
50 FOR X = 0 TO 32
60 RESET (XE, YE)
70 SET (X, X)
80 XE = X : YE = X
90 NEXT
```

A pont visszafelé mozgatása ugyanolyan módon lehetséges, mint a lefelé mozgatás CSAK meg kell találni az összefüggést az X és Y koordináták között.

```
100 FOR X = 32 TO 63
110 RESET (XE, YE)
120 SET (X, 63-X)
130 XE = X : YE = 63-X
140 NEXT
150 GOTO 50
```

(Folytatjuk)



A szegedi MHSZ „Rózsa Ferenc” Rádióklub mély fájdalommal tudatja, hogy tisztelt és szeretett rádióamatőr társunk

Tarnai Titusz János
HABDB

75 éves korában elhunyt.

János OM a rádióklubban több mint húsz éven keresztül tanította, nevelte az amatőröket, képezte a rádiótávirászkokat. Míg egészsége engedte, fáradhatatlanul dolgozott a klubban. 1968-tól, mint adóengedélyes sok ezer összeköttetése révén vált ismertté hívójel.

Emlékét megőriztük.

Úttörők Mikro-klubja 8.

Tömbök

A legutóbb rajzoltunk, most gyakorlatként rajzoljuk meg azt a függvényt, mely megmutatja, hogy különböző feszültségekre kapcsolt ellenálláson időegység alatt mennyi hőenergia szabadul fel. A függvény

$$W = U \cdot I \cdot t, \text{ ahol } I = \frac{U}{R} \text{ és így}$$

$$W = \frac{U \cdot U \cdot t}{R} = \frac{U^2 \cdot t}{R}$$

Rajzoljuk meg a függvényt! A feszültség változzék 0-tól 12 V-ig, az ellenállás a példa kedvéért legyen 3 ohm, az idő 1 s.

```
10 CLS: R = 3: T = 1
20 FOR U = 0 TO 12
30 SET (U, 45 - U * U * T / R)
40 NEXT
```

Szebb lesz az ábra, ha a vízszintes (X irányban) széthúzzuk a pontokat mondjuk három-négyszeresére. Ehhez a program 30-as sorát át kell írunk.

```
30 SET (3 U, 45 - U * U * T / R)
```

Ezután futtassuk a programot úgy, hogy a feszültség 0-tól 24 V-ig változzon.

```
20 FOR U = 0 TO 24
```

A programunk hibajelzéssel leáll, mert nem fér ki a képernyőre az ábrázolni kívánt pont.

Ilyenkor, ha tudjuk a várható maximális értékeket, úgy olyan mértékben kell összenyomni a koordináták mentén a függvényt (olyan léptéket kell választani), hogy ráférjünk a képernyőre, így most

```
30 SET (2 U, 45 - U * U * T / R / 4)
```

Vigyázzunk a számítógép ismeri a műveletek elvégzésének sorrendjét, ezért az $\frac{a}{b \cdot c}$ programozásilag

$a/(b \cdot c)$ vagy $a/b/c$ módon írandó. Ha nem tudjuk előre a várható szélső értékeket, akkor célszerű a géppel kiszámíttatni, — ezeket meg kell őrizni! — és kikerestetni az előforduló maximális (minimális) értéket. Majd ez alapján már kiszámíthatjuk az ábrázoláshoz szükséges értéket. Itt most két új feladattal találkozunk. Az egyik megőrízni — tárolni bizonyos számértékeket. A másik a maximális érték keresése.

Kezdjük az előbbivel.

Amennyiben a programfutás későbbi szakaszában szükségünk lehet X1, Y1, X2, Y2, X3, Y3...Xp, Yp értékeire, akkor ezeket tárolni kell. A változókat a különböző gépek különböző módon, illetve a RAM különböző területein tárolják. Az egy tömbben tárolni kívánt összetartozó számsorokat (később majd betűk, stringek) számoszlopok, vagy táblázatok tárolásához lehetőleg a program elején ki kell jelölni, mintegy le kell foglalni egy memóriaterületet.

Ezt a számítástechnikában tömbdeklarációnak nevezzük. A tömb kijelölésénél (deklarációnál) meg kell adni az azonosítást és a tömb méretét. Ha az előző példát folytatjuk, akkor láthatjuk, hogy az ábrázoláshoz szükséges lépték meghatározásához össze kell gyűjteni az összes kiszámított $W = U \cdot U \cdot T / R$ értéket. A 20-as soron kijelölt ciklusváltozó határértékeiből láthatjuk, hogy 25 feszültségértékkel fogunk számolni, tehát 25 W érték „kelekezik”.

Így a kiszámított értékek tárolásához szükséges tömb deklarálása 10 DIM W (25)

```
20 R = 3: T = 1
30 FOR U = 0 TO 24
40 W = U * U * T / R
```

A most éppen kiszámított W érték a W (1), W (2)...W (24) értéket be is kell tölteni a kijelölt tömbbe. A HT gép értelmezi és tárolja a W (0) jelölésű tömbelemet, a ZX-nél csak a W (1) indextől szabad indulni. A kiszámított értéket tömbbe töltjük:

```
50 W(U) = W
60 NEXTU
```

Most már ugyan betöltöttük a gépbe az adatokat, de így még számunkra hozzáférhetetlenek. (Mielőtt folytatnánk képzeljünk el egy olyan tömböt [tárolórészt], melyben két oszlopban helyezkednek el az adatok, esetleg szavak; az ilyen tömb deklarációja HT-nél DIMA\$ (100,2). Ez akár 100 szó tárolására alkalmas szótárprogram tárolója lehetne. Egyszer talán eljutunk odáig, hogy egy egyszerű szótárprogramot összeállítsunk.) Ne szaladjunk azonban ennyire előre. Feladatunk most a függvény ábrázolásához szükséges lépték meghatározása. Ehhez keressük ki az előforduló legnagyobb értéket.

```
12 WMAX = 44
```

Amikor kiszámíttatjuk a W értékeit, vizsgáltsuk meg nem nagyobb-e az általunk meghatározott (még ábrázolható) értéknél. Ha nagyobb, akkor ez legyen a maximális érték

```
42 IF W > W MAX THEN LET W MAX = W
```

Így miután a ciklust befejezte a program már „tudja” a legnagyobb értéket. A lépték, amivel a függőleges irányú koordináták mérőszámát szorozni kell, $L = 44/WMAX$

A program tehát az alábbiak szerint folytatódik:

```
70 CLS: L = 44/WMAX
80 FOR U = 0 TO 24
90 SET (2 * U, 45 - (U * U * T / R) * L)
100 NEXTU
```

Mivel az $U \cdot U \cdot T / R$ értéket egyszer már meghatároztuk és betöltöttük

a tömbbe, a 90-es sort természetesen írhatnánk úgy is, hogy 90 SET (2 * U, 45 - W(U) * L).

Tömböket kell kijelölni, ha sok változót kívánunk tárolni és a program végrehajtása során azokat különböző szempontok szerint akarjuk felhasználni, egymás mellé rendelni.

Nézzünk egy másik példát! HT-re írjuk, de kisebb „helyesírási” szintaktikai eltérésekkel más fajta gépeken is alkalmazható.

Készítsünk programot, mely — egyelőre csak — 10 név és telefonszám tárolására alkalmas. A program tegye lehetővé a tömbök feltöltését, majd ha mind a 10 adatpárt betöltöttük, akkor akár név, akár telefonszám alapján legyen lehetőség az összetartozó adatok kiírására. (Így ez a program is csak a gyakorlást szolgálja. Nagyobb adattárolásnál már gondoskodni kell az adatok megmentésének lehetőségéről is.) A program elején ki kell jelölni az adattároláshoz a tömböket. A HT-nél a betűk, szövegek tárolása érdekében a szövegváltozók, stringek tárolásához külön memóriaterületet kell szabaddá tenni, ez a CLEAR utasítással lehetséges. A memóriában egy betű, egy karakter tárolása 1 bájt történik, így ha a 10 név és a hozzá tartozó információ összesen 500 betűből áll, akkor

```
10 CLS
20 CLEAR500:DIMNS(10):
DIMHS(10)
30 REM betöltés
40 FOR I = 1 TO 10
50 PRINT I;:INPUT „NEV:”
51 NS(I)
60 INPUT „TELEFONSZAM:”
61 HS(I)
70 NEXTI
80 REM Adatok tarolva
```

A keresésnél be kell írunk a K\$-el jelölt keresett nevet, vagy telefonszámot, a program „lapozza végig” a tömböt és ha megtalálja, írja ki az összetartozó adatokat.

```
90 CLS
100 REM „A keresés”
programrészlete
110 INPUT „A keresett NEV
vagy TELEFONSZAM:”;K$
120 CLS
130 FIGYEL = 0
140 FOR I = 1 TO 10
150 IF K$ = NS(I)
THENPRINTNS(I), HS(I)
160 IF K$ = HS(I)
THENPRINTNS(I), HS(I)
170 IF K$ = NS(I)
THENFIGYEL = 1
180 IF K$ = HS(I)
THENFIGYEL = 1
190 NEXT I
200 IFFIGYEL = 0
THENPRINT „A listan nem
szerepel”
210 PRINT
220 GOTO110
```

Dr. Appel György, dr. Lukács János