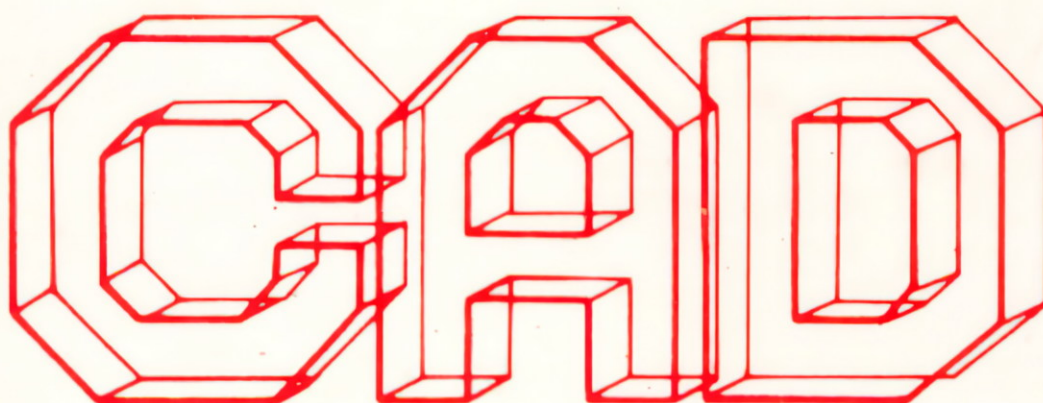


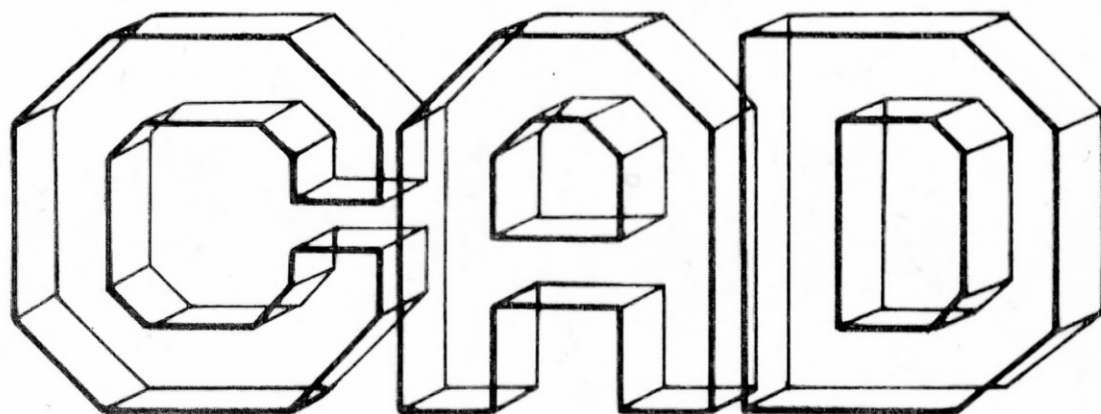
Heift



***Bevezetés
a számítógéppel segített
műszaki tervezésbe
Commodore 64-es
számítógéppel***

DATA BECKER – NOVOTRADE

Heft



***Bevezetés
a számítógéppel segített
műszaki tervezésbe
Commodore 64-es
számítógéppel***

DATA BECKER – NOVOTRADE

A könyv eredeti címe: Einführung in CAD mit dem COMMODORE 64 (1985)

Fordította: DR. BAINNER LÁSZLÓ

Lektorálták: DR. PEREDIJ JÓZSEF és MÁTÉ LAJOS

A programok kidolgozásában részt vett: BUDAI ZOLTÁN



A kiadásért felel RÉNYI GÁBOR, a NOVOTRADE RT. igazgatója Budapest, 1987.

Szerkesztette: DR. SZÉCHY GÁBOR

Műszaki szerkesztő: DÉVÉNYI ERIKA

Szedte a Neotyp Nyomdaipari Szolgáltató Kiszövetkezet, Budapest

Készült a Somogy Megyei Nyomdaipari Vállalat kaposvári üzemében – 171647

Felelős vezető: MIKE FERENC igazgató

ISBN 963 02 4830 1

Hungarian translation © Dr. Baintner László

Copyright © 1985 DATA BECKER GmbH — Merowingerstr. 30. 4000 Düsseldorf

Minden jog fenntartva. A DATA BECKER cég írásbeli hozzájárulása nélkül tilos a könyvet vagy annak részeit bármilyen eljárással (nyomtatás, fotókópia vagy egyéb technika), elektronikus rendszerek felhasználásával másolni, sokszorosítani, terjeszteni.

Előszó

1. BEJÉZET

1.1. Alkalmazások

1.1.1. A CAD

1.1.2. A CAD alapjai

1.1.3. A CAD

1.1.4. A CAD

1.1.5. A CAD

1.2. A CAD a C 64-es CAD

1.3. A C 64-es CAD

1.4. A C 64-es CAD

1.5. A CAD

1.6. A CAD

2. BEJÉZET

2.1. CAD-programok C 64-re

2.1.1. A C 64-es CAD

2.1.2. A CAD

2.1.3. A CAD

2.1.4. A CAD

2.1.5. A CAD

2.1.6. A CAD

2.1.7. A CAD

2.1.8. A CAD

2.2. A CAD

FONTOS TUDNIVALÓ

2.2.1. A CAD

A jelen könyv keretén belül ismertetett kapcsolások, eljárások és programok nem tekinthetők szabadalmi oltalom alá eső ipari termékeknek. Ezek elsősorban amatőr és oktatási célokra szolgálnak. A szerzők rendkívül nagy gondot fordítottak a kapcsolások, műszaki adatok és programok helyességére, a részletek kidolgozása során többszöri ellenőrzést végeztek. Mindez azonban nem zárja ki az esetleges hibalehetőségeket.

Az előforduló hibákért és az ebből adódó következményekért a DATA BECKER cég sem szavatosságot, sem jogi felelősséget nem vállal. Az esetlegesen előforduló hibák közlését a szerzők hálással fogadják.

TARTALOMJEGYZÉK

Bevezetés	9
1. FEJEZET	
Alapgondolatok	11
1.1. Mi a CAD?	11
1.2. A CAD elemei	12
1.2.1. Az ember	12
1.2.2. A hardver	12
1.2.3. A szoftver	13
1.3. Biztosít-e a C 64-es CAD-lehetőségeket?	14
1.4. A C 64-es CAD hardvere	15
1.5. A C 64-es CAD szoftvere	16
1.6. A számítógép belső ábrázolása	17
1.7. Összefoglalás	18
2. FEJEZET	
CAD-program a C 64-esre	19
2.1. A C 64-es CAD legkisebb építőelemei	19
2.1.1. Pont, vonal, téglalap, kör, ívek	19
2.1.2. Különböző vonalvastagságok	19
2.1.3. Párhuzamos vonalak	20
2.1.4. Szaggatott vonalak	21
2.1.5. Tengelyvonalak	24
2.1.6. Méretevonalak és méretvonalak	26
2.1.7. Sraffozások	29
2.2. A műszaki rajz: kis építőelemek összessége	37
2.2.1. Az elemektől a rajzig	37
2.2.2. A rétegtechnika alkalmazása	38
2.2.2.1. Segédraszter a rajzi koordináták meghatározására	39
2.2.2.2. A rajz	41
2.2.2.3. A méretezés (kótázás)	50
2.2.2.4. Feliratozás	60
2.2.2.5. Összefoglalás	63
2.2.3. Léptékek	64
2.2.3.1. Nagyítás, kicsinyítés (és torzítás)	64

2.2.3.2. Zoomolás	69
2.2.4. Részek törlése és kiegészítése	73
2.2.4.1. Törlés nagy felületen	73
2.2.4.2. Törlés finom struktúrában	75
2.3. Összetett elemek (makrók)	81
2.3.1. Mi a makró? Mire használjuk a makrókat?	81
2.3.2. A C 64-es CAD különböző makrói	81
2.3.2.1. Téglatest és kocka	82
2.3.2.2. Gúla	84
2.3.2.3. Háromszög alapú hasáb	86
2.3.2.4. Henger	88
2.3.2.5. Kúp	90
2.3.2.6. Csonkakúp	92
2.3.2.7. Gömb	94
2.4. Makrókból összeállított háromdimenziós rajzok	96
2.4.1. Drótvázmodell	96
2.4.2. Felületmodell	100
2.4.2.1. Makrók a felületmodellekhez	100
2.4.2.2. Felületmodellek makrókból	104
2.4.3. Testmodell	110
2.4.4. Összefoglalás	113
2.5. Mit tehetünk még a rajzokkal?	115
2.5.1. Kettőzés	115
2.5.2. Tükrözés	118
2.5.2.1. Ábrák tükrözése	118
2.5.2.2. Felületek tükrözése	120
2.5.3. Forgatás	123
2.5.3.1. Ábrák forgatása	123
2.5.3.2. Felületek forgatása	125
2.5.3.3. Térbeli forgatás	127
2.5.4. Robbantott rajzok	132
2.5.5. Mozgások	135
2.5.6. Árnyékolás és színezés	137
2.5.7. Metszetek	137
2.5.8. A finomfelbontású grafika tárolása mágneslemezen	141
2.5.9. A finomfelbontású grafika betöltése mágneslemezről	144

3. FEJEZET

Mire használhatjuk a CAD lehetőségeit?

3.1. Számítások és műszaki leírások	147
3.2. Elektronikai rajzok és áramkörtervek	152
3.3. Szervizutasítások és alkatrészlisták	154

4. FEJEZET

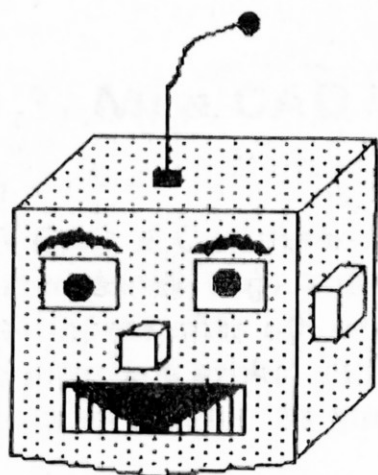
Útmutató egy CAD-rendszer felépítéséhez

Végszó

155

165

BEVEZETÉS



Bemutakozhatok?

Caddy vagyok. Ott jelenek meg ebben a könyvben, ahol nagyon érdekes vagy nagyon fontos témáról van szó. Az egyes fejezetek végén az ikertestvéremmel együtt összefoglaljuk a legfontosabbakat.

Most mesélek valamit a könyv felosztásáról:

Az 1. Fejezetben a számítógéppel segített műszaki tervezés, a CAD, különösen a C 64-es CAD alapelveiről van szó.

A 2. Fejezetben térünk a tárgyra. Ebben a fejezetben mutatjuk be azokat a legfontosabb programokat, amelyekre a műszaki tervezési feladatok megoldásához szükség van. Reméljük, hogy ezeket a programokat azonnal ki is fogják próbálni.

A 3. Fejezetben bemutatjuk, milyen lehetőségei vannak a számítógéppel segített műszaki tervezésnek.

A 4. Fejezetben egy CAD-rendszert mutatunk be, amit CADDYMAT-nak nevezünk el. Míg a 2. Fejezetben csak programelemeket mutatunk be, amelyek feltétlenül szükségesek a műszaki tervek ábrázolásához, a 4. Fejezetben a programelemeket egy jól kezelhető rendszerbe foglaljuk. A CADDYMAT egyes funkcióit menü segítségével választhatjuk ki. Ha ügyesen kihasználjuk rendszerünk lehetőségeit, könnyedén oldhatunk meg műszaki tervezési feladatokat.

Ne várjanak kész rendszert! Ennek terjedelme e könyv lehetőségeit meghaladná.

Sokkal inkább azt szeretnénk bemutatni, hogyan lehet különböző programokat egy menü segítségével párbeszédés üzemmódban alkalmazni. Minden programot úgy építettünk fel, hogy egymással összekapcsolhatók legyenek. A könyvben gyakran találkozunk olyan példákkal, ahol az egyes programelemeket együttesen használjuk.

Ugyancsak találunk példát arra, hogyan kell összerendezni az egyes építőelemeket, hogy kérdés-felelet módban dolgozhassunk velük.

És még volna egy ajánlatunk:

Csodákat ne várjanak! Egy fiókból is csak azt lehet kivenni, amit beletettünk. Ez vonatkozik a számítógépre és Önmagunkra is. Csak apránként haladjunk!



1. FEJEZET

ALAPGONDOLATOK

1.1. Mi a CAD?

A CAD a Computer-Aided-Design rövidítése, és számítógéppel segített műszaki tervezésnek vagy automatizált műszaki tervezésnek (AMT) lehet fordítani. Ezek a fordítások már utalnak arra, hogy mit is jelent ez a fogalom. Érthetőbb lesz, ha pontosítjuk azokat a tevékenységeket, amelyeket egy konstrukció során a mérnöknek el kell végeznie, és amihez a számítógépet is alkalmazhatjuk:

- tervezés,
- számítás-méretezés,
- műszaki rajz előállítás,
- dokumentálás,
- szimuláció és teszt.

Az a konstruktőr, aki CAD-programokat használ, a fiókba rakhatja ceruzáját, vonalzóját és körzőjét, ezek helyett a számítógép billentyűzetét vagy valamilyen grafikus bemeneti eszközt használhat, a rajztáblája legjobb esetben alátét lehet. Minden a képernyőn és a rajzgépen valósul meg, és ez évről évre csodálatosabb, a kívülálló számára egyre félelmetesebb.

A CAD mellett találkozhatunk még a CAM és a CAE rövidítéssel is.

A CAM a Computer-Aided-Manufacturing, ami a számítógéppel segített gyártást jelenti.

CAE a Computer-Aided-Engineering, a CAD és CAM együttesének foghatjuk fel.

Ebben a könyvben csak a CAD-dal foglalkozunk.

Még egy megjegyzés:

Ne tévesszük össze a CAD-ot a grafikával. A CAD sokkal többet jelent. Emlékezzünk csak az előbbieken leírt funkciókra! Természetesen a grafika a CAD fontos alkotó része, de még sok más elem is tartozik hozzá.

1.2. A CAD elemei

1.2.1. AZ EMBER

Már biztosan gyanítják, hogy azt fogjuk írni: minden CAD-rendszer legfontosabb eleme az ember. Ez valóban így van. Egyrészt a számítógépek és a programok csak annyira lehetnek hasznosak és jók, amennyire azok az emberek értenek hozzá, akik működtetik, és használják. Másrészt az ember a legtöbb feladatot meg tudja oldani egy ilyen rendszerrel, még akkor is, ha nagy nehézségek árán kompromisszumokat kell kötnie.

A legnagyobb probléma egy CAD-rendszer befogadása. Az újjal szemben mindenki elővigyázatos, sokszor fél tőle. Ugyanakkor a hatékonyság növelését és a mérnökök teljesítményének ellenőrzését egy CAD-rendszer megkönnyíti.

Ön kedves Olvasó, a legkülönbözőbb okokból érdeklődhet a CAD iránt. De egy dolog Önre is vonatkozik: Ha csalódott abban az eredményben, amit a számítógép produkál, az okot először Önmagában keresse. Az Ön rendszere csak annyit tudhat, mint Ön!

1.2.2. A HARDVER

Egyszerűsítve azt mondhatjuk, hogy a hardver maga a számítógép, és mindaz, ami körös-körül hozzá kapcsolódik, azaz a perifériák. A perifériák köre, a géphez csatlakoztatható perifériák mennyisége a számítógéptől és a csatlakozásainak minőségétől függ. A C 64-eshez rendelkezünk megfelelő perifériákkal. Tulajdonképpen mindent hozzákapcsolhatunk, ami csak előfordul.

Egy jó teljesítményű hardvercsomaghoz, amivel automatizált műszaki tervezést végezhetünk, a következők tartoznak:

- billentyűzet,
- számítógép,
- kölső tároló,
- képernyő,
- grafikus nyomtató.

A kereskedelemben kapható rendszerekben még sok másfajta periféria is előfordul, mint pl. rajzgép, fényceruza, menütablet, és hálózatban működő számítógépek is dolgoznak.

Kedves Olvasó! Gondolom, már megszokta, hogy a teljesen személyes véleményemet is mindig közlöm. Így itt is:

Általában teljesen mindegy, hogy milyen hardvert használunk. Kérdezze meg a barátait! Az egyik az x cég gépére esküszik, a másik a y cégére. Ez, elnézést a kifejezésért, de badarság. A hardvernek természetesen bizonyos feltételeket teljesítenie kell, de ezt mind a két cég hasonlóképpen biztosítja. A manapság rendelkezésünkre álló számítógépek olyan sok lehetőséget nyújtanak, hogy mindezek kifürkészésére egy emberélet sem elegendő. Hogy mit lehet és mit nem lehet, az elsősorban a felhasználótól, és másodsorban a szoftvertől függ. Tulajdonképpen mi mindnyájan zsenik vagyunk, így egy CAD-rendszer teljesítőképessége messzemenőleg a szoftvertől függ. Ezért ebben a könyvben a továbbiakban már kizárólag a szoftverrel foglalkozunk.

1.2.3. A SZOFTVER

A szoftver azoknak a programoknak az összessége, amire egy számítógépnek szüksége van egy kívánt eredmény eléréséhez.

Először is szükség van egy rendszerszoftverre, ami azokat peremfeltételeket határozza meg, amiknek a keretében a feladatot meg lehet oldani.

Szükség van egy programnyelvre, aminek segítségével közölhetjük a géppel, hogy mit kívánunk tőle.

Szükség van egy szoftverre, amellyel működtetni lehet a perifériákat.

Szükség van egy feladatorientált szoftverre, ami egy szakterület speciális követelményeit a számítógép által érthető szabályokba foglalja.

Ebben a könyvben egy feladatorientált szoftverrel foglalkozunk, amelynek segítségével számítógéppel segített műszaki tervezést lehet végezni. Megköveteljük a C 64-esünktől, hogy számunkra konstruáljon, de természetesen meg kell mondanunk, hogy ezt hogyan kell elvégeznie. A 2. Fejezet programjai arra adnak útmutatást, hogy az előzőeket hogyan lehet közölni a géppel.

1.3. Biztosít-e a C 64-es CAD-lehetőségeket?

Természetesen igen, különben nem született volna meg ez a könyv. De pontosítsuk ezeket a lehetőségeket, amelyekkel most foglalkozni szeretnénk. Különösen az a kérdés: Mit várhatunk a C 64-es CAD-tól?

A koordináták megadását és a konstrukció manipulálását a billentyűzet segítségével végezzük el. Bizonyára látták már, hogy CAD-munkahelyeken a billentyűzet helyett fényceruzát használnak. De kinek volt már közülünk fényceruzája? Ami fontosabb: Egy fényceruza segítségével nem lehet pontos koordinátákat megadni. Pl. egy 10-es méret csak véletlenül lesz 10. Sokkal inkább 9,95 vagy 10,72 lesz belőle. Ez azt jelenti, hogy a fényceruza, valahová a kívánt pont mellé talál. Ha pontosan 1-et akarunk megadni, akkor hosszasan el kell pepecselni, amíg eltaláljuk a megfelelő pontot. Ezalatt már a billentyűzetről is megadhattuk volna a koordinátákat. Éppen ezért az újabb rendszerek menütablet-et használnak, vagy adatleíró nyelvvel próbálják meg a számítógépet vezérelni. Mindez a kezelés kényelmét és gyorsaságát szolgálja. Ezzel most mi nem foglalkozunk, de ezen a ponton sincs alapvető hátrányunk a kereskedelemben kapható rendszerekkel szemben.

A számítógépet a belső tárkapacitásával és a számítási sebességével szokták jellemezni. Ebből a szempontból is felveheti a versenyt a C 64-es a műszaki gyakorlatban alkalmazott kisebb rendszerekkel szemben. A C 64-es belső tárja viszonylag kicsi, de erre figyelemmel lehetnek a programjaink. Fel lehet úgy darabolni egy programot, hogy mindig csak egy része van a belső tárban a többi része valamilyen külső tárban marad.

A külső tár, mágneslemez-meghajtó egység végtelenül nagy. Ugyanis annyi lemezt tehetünk be, amennyit csak meg tudunk fizetni. Ez legfeljebb kényelmetlen és lassú, de elvi hátrányt nem jelent a nagy rendszerekkel szemben.

A megkonstruált ábráinkat a képernyőn jelenítjük meg, és grafikus nyomtatón nyomtatjuk ki. Itt jelentkezik egy valódi hátrány, amit türelemmel nem lehet áthidalni. A jó képminőség feltétele a képernyőn és a nyomtatón a felbontás; ez a képet alkotó pontok számával jellemezhető. John Wayne vagy Brigitte Bardot $625 \times 832 = 520\,000$ ponton ábrázolható fekete-fehér tv-képernyőn, és 1,08 millió ponton színesben (ahol a felbontás csak ennek egyharmada, mert minden ponthoz 3 szint rendelünk). A professzionális CAD-rendszerek $1280 \times 1024 = 1\,310\,720$ ponttal dolgoznak. A C 64-es ezzel szemben csak $320 \times 200 = 64\,000$ pont áll rendelkezésre egy képhez. A viszonylag kis felbontás közvetlen következménye: ferde vonalaknál, köröknél és íveknél lépcsőzések keletkeznek. A kép jellemzően digitálisnak tűnik. De ezzel meg lehet békélni, és csodálatos, hogy mi mindent lehet 64 000 képponttal elővarázsolni.

1.4. A C 64-es CAD hardvere

A 2. Fejezet programjai a következő konfiguráción működnek:

Commodore 64-es,

1 db 1541-es hajlékony mágneslemez meghajtó egység (floppy),

MPS 801-es mátrixnyomtató,

színes monitor vagy tv-készülék.

Pontosabban szólva, ezen a konfiguráción fejlesztettük ki a programokat. Ha más nyomtatónk vagy más mágneslemez-meghajtó egységünk van, a programok akkor is működnek. Természetesen feltétel a Commodore 64-es.

Még néhány szó a tárolásról. A belső szabad tárterületet munkatárnak használjuk. Itt dolgozzuk fel a programokat, megváltoztatjuk az adatokat, megadjuk a koordinátákat. Röviden: A konstrukció manipulációja a munkatárban történik. Ha kikapcsoljuk a számítógépet, minden elvész.

Azokat a programokat, adatokat, eredményeket, amelyeket hosszabb időre tárolni kívánunk, mágneslemezekre kell menteni. Ezeket a lemezeket csak a munka kezdetekor és befejezésekor használjuk. A meghajtóegységbe csak rövid időre tesszük be, és a megfelelő művelet után azonnal ki is vehetjük.

Emellett használnunk kell olyan lemezeket is, amelyek a konstruálás során állandóan a meghajtóegységben vannak, mert az éppen szükséges programokat innen töltjük be. Különlegességként használunk még a közbenső eredmények tárolására is lemezt. A konstruálás sokszor nagyon bonyolult, hosszantartó feladatot jelent. Azért, hogy egy hiba miatt ne vesszen el az egész addigi munkánk, bizonyos időközönként a közbenső eredményeket célszerű kimenteni. Ehhez a programlemez ki kell venni, és a közbenső eredményeknek szánt lemezt kell behelyezni. Ha két meghajtóegységünk van, akkor megtakaríthatjuk a sokszor körülményes cserélgetéseket. Az 1-es meghajtó a programok számára állandóan rendelkezésre áll, a 2-es meghajtóval olyan gyakran vehetjük fel a közbenső eredményeket, ahogy akarjuk. Különösen akkor, amikor a 4. Fejezet CADDYMAT-jával kívánunk dolgozni, érdemes egy második meghajtóegységet is beszerezni.

1.5. A C 64-es CAD szoftvere

A következő szoftverek állnak rendelkezésre:

- a számítógépben levő rendszerszoftver,
- a Commodore BASIC
- a SIMON'S BASIC, BASIC-titerjesztés

és azok a programok, amelyeket ennek a könyvnek a keretében szeretnénk kifejleszteni.

Talán csodálkoztak az 1.3. alfejezetben olvasott adaton, hogy a Commodornak $320 \times 200 = 64\,000$ képpontja van. Eddig csak a 40 oszlophoz és a 24 sorhoz voltunk hozzá szokva. Ez azért van, mert még nem dolgoztunk a SIMON'S BASIC-kel. Ez egy olyan BASIC-kiterjesztés, amelyre mindeképpen szükségünk van a CAD-rendszerünk kifejlesztéséhez. SIMON'S BASIC nélkül nem működik a C 64-es CAD! Mielőtt a számítógépünkkel konstruálhatnánk a SIMON'S BASIC-et mágneslemezről be kell tölteni. Ezután sok olyan parancs is rendelkezésünkre áll, amelyeket a számítógépes műszaki tervezéshez készítettek. Ezek között főleg grafikus parancsok vannak, de vannak olyan parancsok is, amelyek a kényelmes programszerkesztést, a szövegfeldolgozást támogatják. Nagyon fontos az a parancs is, amivel a képernyő tartalmát a mátrixnyomtatón könnyedén megjeleníthetjük.

Pl.: A HIREs parancs bekapcsolja a finomfelbontású grafikát. Így azonnal vízszintesen 320 pont, függőlegesen pedig 200 pont, összesen 64 000 pont áll a rendelkezésünkre. A COPY parancs ezt a 64 000 pontot képernyőről a mátrixnyomtatóra viszi át.

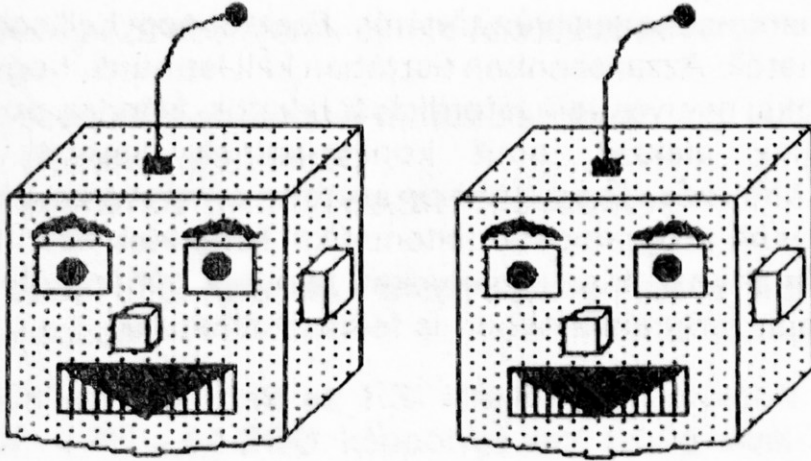
Egyetlen hátránya ennek a bővítésnek az, hogy a SIMON'S BASIC a C 64-es szabad belső tárjának egy részét elfoglalja. Kb. 39 kbyte állt eddig rendelkezésünkre, a bővítés betöltése után csak kereken 31 kbyte marad.

De ne féljünk, egy program, amely 31 kbyte-ot használ, az már egy hatalmas program. Ilyen itt nem fordul elő. Mindent szépen, tisztán kis részekre felbontottunk, és a 4. Fejezet CAD-rendszere kinyúlik a külső tárolóra is.

1.6. A számítógép belső ábrázolása

A számítógépen belül minden a matematika nyelvén történik. Ezekről nem kell sokat tudni, mégis látványosan használhatók. Azzal azonban tisztában kell lennünk, hogy a programok a gondolatok matematikai nyelvre való lefordítását jelentik. Minden pont, minden vonal, minden komplex alakzat, amit konstrukciónk használ, a számítógépben egy matematikai kifejezés. Ezért néhány esetben elmagyarázzuk a programok matematikai alapelvét is. A programokat matematikai ismeretek nélkül is lehet használni. Ha mégis áttanulmányozzuk az alapelveket, találunk néhány olyan megoldási módszert, amelyet a saját programjainkban is felhasználhatunk.

1.7. Összefoglalás



1.1.: CAD = Computer-Aided-Design

Tervezés
Számítás-méretezés
Rajzok előállítása
Dokumentálás
Szimuláció és teszt.

1.2.: CAD-rendszer = Ember + Hardver + Szoftver

A minőséget először az ember utána a szoftver és csak ezután a hardver dönti el.

1.3.: A C 64-es CAD teljesítőképessége

Egyetlen valódi hátránya a drága és nagy rendszerekkel szemben: a megjelenítőegységek gyengébb felbontása.

1.4.: A C 64-es CAD-hoz szükséges:

Commodore 64-es
1 mágneslemez-meghajtó egység
mátrixnyomtató
színes monitor vagy színes tv.

1.5.: A C 64-es CAD-hoz szükséges szoftver

A SIMON'S BASIC bővítés.

1.6.: Nem árt, ha van kedvünk a matematikához!

2. FEJEZET

CAD-PROGRAM A C 64-ESRE

2.1. A C 64-es CAD legkisebb építőelemei

2.1.1. PONT, VONAL, TÉGLALAP, KÖR, ÍVEK

Ezekhez az alapelemekhez a SIMON'S BASIC utasításait használjuk fel. Célszerű, a kézikönyvet a kezünk ügyében tartani, mert gyakran szükségünk lesz rá. Amit ott leírtak, azt itt már nem ismételjük meg, különben ez a könyv még egyszer ilyen vastag lenne.

Vége, elkezdhetjük a feladatorientált programunk készítését.

2.1.2. KÜLÖNBÖZŐ VONALVASTAGSÁGOK

A műszaki rajzokon látható éleket vastag folyamatos vonallal, a méretvonalakat és a segédvonalakat vékony folyamatos vonallal ábrázoljuk. A vonalvastagság az ábrázolás mindenkori nagyságától függ. Szükségünk van tehát egy programra, amely különböző vastagságú vonalakat tud húzni.

```
10000 REM VASTAGVONAL
10001 GOTO 10020
10002 REM VASTAGVONAL ALPROGRAM
10003 IF AB=AD THEN 10012
10004 FOR AJ=1 TO AE
10005 AF=AA+AJ:AG=AB
10007 AH=AC+AJ:AI=AD
10009 :LINE AF,AG,AH,AI,1
10010 NEXT AJ
10011 GOTO 10019
10012 FOR AK=1 TO AE
10013 AG=AB+AK:AF=AA
10015 AI=AD+AK:AH=AC
10017 :LINE AF,AG,AH,AI,1
10018 NEXT AK
10019 RETURN
10020 HIRES 0,7
10021 AA= 50 :REM A KEZDŐPONT X-KOORDINÁTÁJA
10022 AB= 20 :REM A KEZDŐPONT Y-KOORDINÁTÁJA
```

```

10023 AC=150 :REM A VÉGPONT X-KOORDINÁTÁJA
10024 AD= 20 :REM A VÉGPONT Y-KOORDINÁTÁJA
10025 AE= 2 :REM A VONAL VASTAGSÁGA
10026 GOSUB 10002
10027 GOTO 10027

```

Ez a program tetszőleges hosszúságú, irányú és vastagságú vonalakat tud húzni.

Az AA—AD változók a vonal koordinátáit, az AE változó a vonalvastagságot tartalmazza.

Vegyünk egy milliméter beosztású papírt, és jelöljük ki rajta egy területet, amely vízszintesen 320 mm, függőlegesen 200 mm. Ezen a területen belül minden metszéspont egy képpontnak felel meg. Így könnyedén meg tudjuk határozni a koordinátákat.

Az 10027. sor badarságnak tűnik. Mégis szükségünk van rá, mert e nélkül a számítógép kikapcsolná a finomfelbontású grafikát. Általában ezen a helyen egy másik program kapcsolódik. Futtassuk le egyszer a programunkat a 10027. sor nélkül. Ha az eredeti programból ki akarunk lépni, akkor a RUN/STOP gombot kell megnyomnunk. Ha egyszerre több vonalat akarunk a képernyőre rajzolni, akkor a 10027. sortól kezdődően értéket kell adni az AA—AD változóknak, esetleg az AE-nek is, és be kell írni a GOSUB 10002 utasítást is, pontosan úgy, ahog az a 10021—10026. sorban látható. Ezt annyiszor ismételhetjük, ahányszor csak akarjuk. Ne feledkezzünk meg az utolsó „badar” sorról! Az alprogram a program elején áll. Át kell ugrani egy GOTO utasítással, mert különben a RUN utasítás hatására a program alprogramhívás nélkül rászalad, ami elkerülhetetlenül hibajelzéshez vezet. Ha ki akarjuk nyomtatni a vonalakat, az utolsó sorba írjunk egy COPY parancsot. Természetesen előbb a szükséges csatornát egy OPEN utasítással meg kell nyitni.

2.1.3. PÁRHUZAMOS VONALAK

Gyakran előfordul, hogy egy terven párhuzamos vonalakat kell rajzolni.

A következő program tetszőleges hosszúságú, helyzetű, és tetszőleges távolságú vonalakat rajzol.

A BA—BD változók a kiindulási vonal koordinátái, BE a párhuzamosok távolsága.

Különböen a 2.1.2. pontban leírtak itt is értelemszerűen érvényesek.

```

10050 REM PÁRHUZAMOS
10051 GOTO 10056
10052 REM PÁRHUZAMOS ALPROGRAM
10053 IF BB=BD THEN 10059
10054 BF=BE:BG=BB
10055 BH=BC+BE:BI=BD
10056 :LINE AF,AG,AH,AI,1
10059 BF=BA:BG=BB+BE

```

```

10061 BH=BC:BI=BD+BE
10063 :LINE BA,BB,BC,BD,1
10064 :LINE BF,BG,BH,BI,1
10065 RETURN
10066 HIRES 0,7
10067 BA= 50 :REM A KEZDŐPONT X-KOORDINÁTÁJA
10068 BB= 20 :REM A KEZDŐPONT Y-KOORDINÁTÁJA
10069 BC=150 :REM A VÉGPONT X-KOORDINÁTÁJA
10070 BD= 20 :REM A VÉGPONT Y-KOORDINÁTÁJA
10071 BE= 15 :REM A PÁRHUZAMOSOK TÁVOLSÁGA
10072 GOSUB 10052
10073 GOTO 10073

```

2.1.4. SZAGGATOTT VONALAK

A műszaki rajzokban a takart éleket szaggatott vonalakkal jelzik.

Ez a program tetszőleges hosszúságú és helyzetű szaggatott vonalakat rajzol.

A CA—CD változók tartalmazzák a szaggatott vonal koordinátáit.

Egy egyszerű szaggatott vonal rajzolásához ilyen terjedelmes program szükséges. Ha minden lehetőséget figyelembe veszünk — vízszintes, függőleges, balra dőlő, jobbra dőlő vonal — a programunk tekintélyesen megnőhet.

```

10100 REM SZAGGATOTTVONAL
10101 GOTO 10107
10102 REM SZAGGATOTTVONAL ALPROGRAM
10103 CE=ABS<CC-CA>
10104 CF=ABS<CB-CD>
10105 IF CA=CC THEN 10137
10106 IF CB=CD THEN 10158
10107 CG=INT<SQR<<16*CE↑2>/<CF↑2+CE↑2>>>
10108 CH=INT<<CG*CF>/CE>
10109 CI=INT<CE/CG>
10110 CJ=-1
10111 IF CA<CC AND CD<CB THEN 10115
10112 IF CA<CC AND CB<CD THEN 10117
10113 IF CC<CA AND CD<CB THEN 10120
10114 IF CC<CA AND CB<CD THEN 10123
10115 CG=-CG
10116 GOTO 10125
10117 CG=-CG:CH=-CH
10119 GOTO 10125
10120 CG= CG:CH= CH
10122 GOTO 10125
10123 CH=-CH
10124 GOTO 10125
10125 CK=CA+CG:CL=CB+CH

```

```

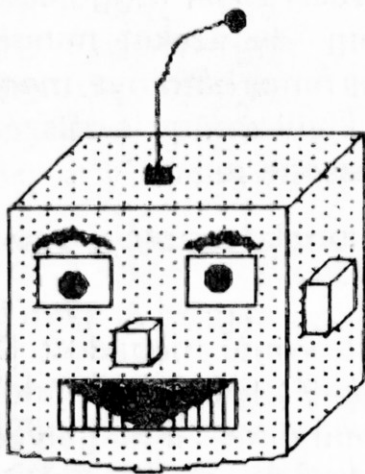
70127 REM HIRES
70128 FOR CM=1 TO CI
70129 GOSUB 10178
70130 CK=CK-CG
70131 CL=CL-CH
70132 CN=CK-(2*CG)
70133 CD=CL-(2*CH)
70134 :LINE CK,CL,CN,CD,CP
70135 NEXT CM
70136 GOTO 10186
70137 REM FÜGGÖLEGES
70138 CI=INT(CF/4)
70139 CJ=-1
70140 CH= 4
70141 IF CB<CD THEN 10143
70142 IF CD<CB THEN 10145
70143 CH=-CH
70144 GOTO 10147
70145 CH= CH
70146 GOTO 10147
70147 CL=CB+CH
70148 FOR CM=1 TO CI
70149 GOSUB 10178
70150 CK=CA
70151 CL=CL-CH
70152 CN=CA
70153 CD=CL-(2*CH)
70154 :LINE CK,CL,CN,CD,CP
70155 NEXT CM
70156 GOTO 10186
70157 REM VÍZSZINTES
70158 CI=INT(CE/4)
70159 CJ=-1
70160 CG= 4
70161 IF CA<CC THEN 10164
70162 IF CC<CA THEN 10166
70163 CG=-CG
70164 GOTO 10167
70165 CG= CG
70166 REM HIRES
70167 CK=CA+CG
70168 FOR CM=1 TO CI
70169 GOSUB 10178
70170 CK=CK-CG
70171 CL=CB
70172 CN=CK-(2*CG)
70173 CD=CB
70174 :LINE CK,CL,CN,CD,CP
70175 NEXT CM

```

```

10177 GOTO 10186
10178 REM CIKLUS
10179 CJ=CJ#(-1)
10180 IF CJ= 1 THEN 10182
10181 IF CJ=-1 THEN 10184
10182 CP=1
10183 GOTO 10185
10184 CP=0
10185 RETURN
10186 HIRES 0,7
10187 CA= 50 :REM A KEZDŐPONT X-KOORDINÁTÁJA
10188 CB=100 :REM A KEZDŐPONT Y-KOORDINÁTÁJA
10189 CC= 50 :REM A VÉGPONT X-KOORDINÁTÁJA
10190 CD=199 :REM A VÉGPONT Y-KOORDINÁTÁJA
10191 GOSUB 10102
10192 GOTO 10192

```



Bizonyára feltűnt már, hogy

- az első program sorszámai 10000-rel kezdődnek, és a további programok sorszámai ennél mind nagyobbak;
- a sorszámok közötti lépésköz minimális, de az egyik program vége és a következő eleje között bőségesen áll hely a rendelkezésünkre;
- az első program változói mind A-val kezdődnek, a másodiké B-vel, a harmadiké C-vel.

Az egésznek az a célja, hogy tetszőleges programokat kombinálhassunk egymással anélkül, hogy a sorszámok és a változók összekeverednének. Minden programot alprogramként be lehet illeszteni egy új főprogramba.

Bizonyára azt is észrevették, hogy nem használunk „valódi” parancsokat. Megváltoztatjuk a programsorokat, ha az eredményt meg akarjuk változtatni. Ha minden programnak önállóan kell működnie, akkor ez a legegyszerűbb módszer. Ha a programokat rendszerbe kívánjuk szervezni, akkor természetesen más a helyzet. A CADDYMAT-nál (4. Fejezet) nem kell programsorokat változtatni, ott párbeszéd során közölhetjük a géppel kívánságainkat.

Nem lesz nehéz a programokat saját kívánságunk szerint rendszerbe foglalni. Pl. a programok végét, ott ahol a változókat módosíthatjuk, levághatjuk (ezért állnak a bemenő paraméterek mindig a program végén), és INPUT utasításokkal kifejleszt-
hetjük saját párbeszédünket.

A független programelemeknek az az előnye, hogy láthatjuk elkészítésük módját. Hátrányuk, hogy mindig csak egy problémát tudunk velük megoldani. Egy teljes rendszernél az egyes részletek megoldásai már nem ismerhetők fel, de összetett feladatokat is meg lehet velük oldani.

2.1.5. TENGELYVONALAK

A műszaki rajzokon a szimmetriatengelyeket pontvonallal, eredményvonallal jelölik. Ez a program is hasonlóan nézhetne ki, mint a 2.1.4. pont szaggatott vonalat rajzoló programja, csak még bonyolultabb lenne a különböző vonalhosszúságok miatt.

Legyünk ravaszak, és vezessünk be néhány ésszerű korlátozást! Csak függőleges, vízszintes és 45°-os ferde tengelyvonalakat engedünk meg, de ezeket minden irányban. Ez a feltételezés jelentősen egyszerűsíti a feladatot, és nincs hátránya, mert a legtöbb esetben a tengelyvonalak ilyen irányban futnak. Ezen kívül amúgy is célszerű a nem 45°-os ferde vonalakat elkerülni a vonalak lépcsőzetessége miatt.

Ebben a fejezetben a függőleges, vízszintes és 45°-os tengelyvonalat rajzoló program két változatát is bemutatjuk.

Az első változatban még egy egyszerűsítést bevezettünk: a vonal megadott DE hosszúságát csak hozzávetőlegesen tartjuk be. A vonal néhány képponttal mindig rövidebb. Ezt a műszaki rajzokon egyáltalán nem lehet észrevenni, mert a tengelyvonalak mindig hosszabbak, mint azok az elemek, amelyeket elvágnak, és valahol abbamaradnak. Így a vonal vége nem találkozik másik vonallal.

```
10200 REM TENGELYVONAL V1
10201 GOTO 10209
10202 REM TENGELYVONAL V1 ALPROGRAM
10203 DH$="666666666666116611"
10204 DI=INT<DE/16>
10205 DJ$=DUP<DH$,DI>
10206 :ROT DC,DD
10207 :DRAW DJ$,DA,DB,1
10208 RETURN
10209 HIRES 0,7
10210 DA= 50 :REM A KEZDŐPONT X-KOORDINÁTÁJA
10211 DB= 50 :REM A KEZDŐPONT Y-KOORDINÁTÁJA
10212 DC= 3 :REM A TENGELYVONAL IRÁNYA
10213 DD= 2 :REM A NAGYÍTÁSI TÉNYEZŐ
10214 DE= 99 :REM A TENGELYVONAL HOSSZA
10215 GOSUB 10202
10216 GOTO 10216
```

A második változatban a hozzávetőleges vonalhosszúságra vonatkozó egyszerűsítést megszüntettük. Itt a vonal pontosan olyan hosszú, mint azt a DE változóban megadtuk. Már ez a kis részlet a programot 5 utasítással megnövelte. Ha a 2.1.4. és 2.1.5. pont programjait összehasonlítjuk, szemléletes példát látunk arra, hogy néhány értelmes korlátozás a programot jelentősen egyszerűsítheti.

Állandóan ellenőrizzük, hogy csak a követelményeknek eleget tevő programot készítsünk. A szükségtelen általánosság drága belső tárhelyeket foglal, és meghosszabbítja a számítási időt. Ezen kívül jelentős szellemi energiát is igényel.

Most még valamit a DA—DE változókról: DA és DB a kezdőpont koordinátái. Innen indul a tengelyvonal abba az irányba, amit a DC paraméter meghatároz.

DC megengedett értékei:

DC = 0 ... 0°

DC = 1 ... 45°

DC = 2 ... 90°

DC = 3 ... 135°

DC = 4 ... 180°

DC = 5 ... 225°

DC = 6 ... 270°

DC = 7 ... 315°

Így pl. a DC = 0 függőleges tengelyvonalnak a DC = 1 45°-os tengelyvonalnak felel meg. A szögek az óramutató járásával megegyező irányban növekednek.

A DD változó nagyítási tényező. Ezzel a vonalkák hosszát a tengelyvonalon belül, valamint a teljes tengelyvonal hosszát szabályozhatjuk. A megengedett értékek 0-tól 255-ig terjednek, de egy átlagos hosszúságú tengelyvonal esetén is már a 4-szeres szorzóval a képernyő szélén túlra kerülünk.

DE a tengelyvonal hosszát adja meg. Elméletileg a maximális hossz DE = 256. A nagyítási tényezővel ezt azonban túlléphetjük. Meghosszabbíthatjuk a vonalat úgyis, hogy a végpontjába a következő tengelyvonal kezdőpontját helyezzük.

Az előbbi megállapítások csak elméleti jellegűek, hiszen egy függőleges tengelyvonal legfeljebb 200 egység (ezalatt mindig képpontot értünk) hosszú lehet.

```
10220 REM TENGE LYVONAL V2
10221 GOTO 10232
10222 REM TENGE LYVONAL V2 ALPROGRAM
10223 DH$="BBBBBBBBBBB116611"
10224 DI=INT<DE/16>
10225 DK=MOD<DE,16>
10226 DL$=LEFT$(DH$,DK)
10227 DM$=DUP<DH$,DI>
10228 DJ$=DM$+DL$
10229 :ROT DC,DD
10230 :DRAW DJ$,DA,DB,1
```

```

10231 RETURN
10232 HIRES 0,7
10233 DA= 50 :REM A KEZDŐPONT X-KOORDINÁTÁJA
10234 DB= 50 :REM A KEZDŐPONT Y-KOORDINÁTÁJA
10235 DC=  2 :REM A TENGELYVONAL IRÁNYA
10236 DD=  1 :REM A NAGYÍTÁSI TÉNYEZŐ
10237 DE=150 :REM A TENGELYVONAL HOSSZA
10238 GOSUB 10222
10239 GOTO  10239

```

2.1.6. MÉRETNYILAK ÉS MÉRETVONALAK

A műszaki rajzokon a méreteket méretnyilak, méretvonalak és segédvonalak segítségével jelöljük meg. A CAD-rendszerünkben használt méretnyilak eltérnek a szabványostól (mind a DIN, mind az MSZ szabványtól). Ezeket a szabványokat akkor rögzítették, amikor még nem volt számítógép, és ezért nem is lehettek a CAD-rendszereknek megfelelőek. Mostanában a professzionális felhasználók saját szabványt alkotnak. Miért nem tesszük mi is ezt? A méretvonalakat nem szakítjuk meg, és nem írjuk közéjük a méretszámokat. A CAD-rajzokon a méretvonalak nyíltól nyílig haladnak, és a méretszámok a vonal felett állnak.

Ebben a fejezetben 3 programot találunk, amelyek egyre kényelmesebben használhatók.

A MÉRETNYÍL program egy méretnyilat rajzol a kapcsolódó méretvonallal tetszőleges hosszúságban, vízszintesen, függőlegesen vagy 45°-os irányban.

Az EA és EB változók a nyíl hegyének koordinátáit adják meg. Az EC paraméterrel azt lehet meghatározni, hogy a nyíl és a méretvonal milyen irányba mutasson.

EC megengedett értékei azonosak a DC változó megengedett értékeivel, amit a 2.1.5. pontban ismertettünk.

ED ismét a nagyítási tényező, és EE a méretvonal hosszát adja meg.

```

10250 REM MÉRETNYÍL
10251 GOTO 10260
10252 REM MÉRETNYÍL ALPROGRAM
10253 EF$="606000365555506000000000365555555553333"
10254 EG$="6"
10255 EH$=DUP(EG$,EE)
10256 EI$=EF$+EH$
10257 :ROT EC,ED
10258 :DRAW EI$,EA,EB,1
10259 RETURN
10260 HIRES 0,7
10261 EA=100 :REM A MÉRETNYÍL HEGYÉNEK X-KOORDINÁTÁJA
10262 EB=100 :REM A MÉRETNYÍL HEGYÉNEK Y-KOORDINÁTÁJA
10263 EC=  0 :REM A MÉRETNYÍL IRÁNYA

```



```

10264 ED= 1 :REM A NAGYÍTÁSI TÉNYEZŐ
10265 EE=100 :REM A MÉRETVONAL HOSSZA
10266 GOSUB 10252
10267 GOTO 10267

```

A KETTŐS MÉRETNYÍL program a méretvonal másik végére is rajzol egy méretnyilat, de ez ellentétesen áll.

Az EA és EB változók a kiindulási nyíl koordinátáit adják meg. Ha EC = 0, akkor az első méretnyíl lefelé mutat. A méretvonal felfelé indul ettől a nyíltól kezdve, és a másik méretnyíl felfelé mutat.

Ha EC = 4, akkor az első nyíl felfelé mutat, a méretvonal függőlegesen lefelé halad, és a második méretnyíl lefelé mutat. Ennek megfelelően rajzoljuk a többi irányban is.

Ez egy kicsit bonyolultnak tűnik, de nem az. Meglepő módon a második program csak egyetlen utasítással hosszabb az elsőnél.

```

10280 REM KETTŐS MÉRETNYÍL
10281 GOTO 10291
10282 REM KETTŐS MÉRETNYÍL ALPROGRAM
10283 EF$="606888365555068888888836555555553333"
10284 EJ$="375507888888375555550788888888"
10285 EG$="6"
10286 EH$=DUP<EG$,EE>
10287 EI$=EF$+EH$+EJ$
10288 :ROT EC,ED
10289 :DRAW EI$,EA,EB,1
10290 RETURN
10291 HIRIS 0,7
10292 EA=100 :REM A KEZDŐNYÍL HEGYÉNEK X-KOORDINÁTÁJA
10293 EB=100 :REM A KEZDŐNYÍL HEGYÉNEK Y-KOORDINÁTÁJA
10294 EC= 0 :REM A MÉRETVONAL IRÁNYA
10295 ED= 1 :REM A NAGYÍTÁSI TÉNYEZŐ
10296 EE= 80 :REM A MÉRETVONAL HOSSZA
10297 GOSUB 10282
10298 GOTO 10298

```

A KÓTÁZÁS program még egy lépéssel tovább jut. A méretvonalak és nyilak mellett még segédvonalakat is rajzol. Ezek azok a vonalak, amik a méretezendő szélektől indulnak ki, és meghatározzák, hogy mire vonatkozik a megadott méret. A méretnyilak éppen érintik a segédvonalakat.

Az EA—EE változók azonosak mindkét előző program változóival azzal a különbséggel, hogy EA és EB a segédvonalak kezdőpontjának koordinátái a méretezendő idom konturján.

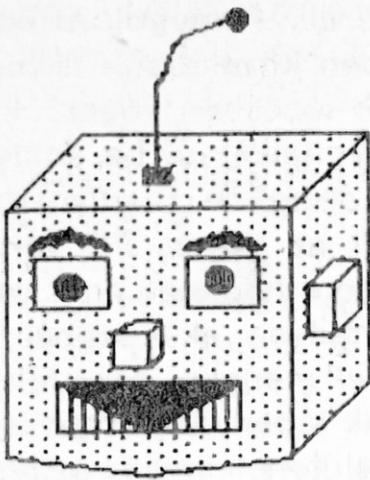
További változók az EM és EN. EM a méretvonal távolságát adja meg a méretezendő test szélétől vagy egy másik méretvonalától, amely vele párhuzamosan fut. A maximális távolság EM = 56 lehet. Ilyen nagy távolságok (ilyen hosszú segédvonalak) nem fordulhatnak elő. Ha mégis, akkor helytelenül méreteztük a rajzot.

EN a méretvonalnak a méretezendő test széléhez viszonyított helyzetét adja meg. Ha EN = 1, akkor a méretvonal a test szélétől jobbra, ha EN = 2, akkor balra halad a méretvonal.

```

10310 REM KÓTA
10311 GOTO 10334
10312 REM KÓTA ALPROGRAM
10313 EF$="60688826555506888888836555555553333"
10314 EJ$="3755078888375555550788888888555511116"
10315 EG$="6"
10316 EH$=DUP<EG$,EE>
10317 EK$="5"
10318 EL$=DUP<EK$,EM>
10319 IF EN#2 THEN 10326
10320 FO$="555333"
10321 FP$=EL$+FO$
10322 FQ$="6"
10323 ER$=DUP<EQ$,EM>
10324 ES$=FP$+ER$+EH$+EJ$+EO$+ER$
10325 GOTO 10332
10326 ET$="6"
10327 EU$=DUP<ET$,EM>
10328 EV$="888000"
10329 EW$=EU$+EV$
10330 ES$=EW$+EF$+EH$+EJ$+EV$+EL$
10331 :ROT EC,ED
10332 :DRAW ES$,ER,EB,1
10333 RETURN
10334 HIRIS 0,7
10335 EA=100 :REM A SEGÉDVONAL KEZDŐPONTJÁNAK X-KOORD.
10336 EB=100 :REM A SEGÉDVONAL KEZDŐPONTJÁNAK Y-KOORD.
10337 EC= 2 :REM A MÉRETVONAL IRÁNYA
10338 ED= 1 :REM A NAGYÍTÁSI TÉNYEZŐ
10339 EF= 60 :REM A MÉRETVONAL HOSSZA
10340 EM= 40 :REM A TÁVOLSÁG A TEST SZÉLÉTŐL
10341 EN= 2 :REM A TEST MELYIK OLDALÁN HELYEZKEDIK EL?
10342 GOSUB 10312
10343 GOTO 10343

```



Bizonyára elcsodálkoztak, hogy nem adtunk semmilyen magyarázatot a programok felépítéséről, az egyes utasításokról. Annak nincs sok értelme, hogy elmagyarázzuk a programokat. Önmagunknak kell pontosan végiggondolni, ha tudni akarjuk miért és hogyan működnek.

A kézikönyvekben utána lehet nézni az egyes parancsoknak. Így könnyebben megérthetjük a programok működését, mintha azt elmagyaráznánk, és akit ez nem érdekel, azt nem untatjuk.

2.1.7. SRAFFOZÁSOK

A műszaki rajzokon a metszett részeket sraffozásokkal jelzik. A sraffozások 45° -os szög alatt futnak balról jobbra, azaz jobbra emelkedően, jobbról balra, azaz balra emelkedően, és mindkét irányban, azaz keresztezve.

A SRAFFOZÁS program tetszőleges felületeket jobbra vagy balra emelkedően besraffoz tetszőleges távolságú vonalakkal.

A KERESZTSRAFFOZÁS program tetszőleges felületeket jobbra és balra emelkedően, azaz keresztezve sraffoz be tetszőleges távolságú vonalakkal. Ez nagyon egyszerűnek hangzik, és használata is valóban egyszerű. A megoldás megtalálása jelent nehéz feladatot.

Mivel a feladatokból és azok megoldásából sokat lehet tanulni, ezért most ezt egy kissé részletezzük.

A LINE paranccsal minden egyes vonalat külön-külön megrajzolhatunk. Ez azonban már egy egyszerű négyszögnél is hatalmas programozási munkát jelentene.

A 2.1.3. pont PÁRHUZAMOSOK programját alkalmazhatnánk, de ez sem csökkentené a munkánkat. Ha tetszőleges felületeket akarunk sraffozni, gyorsan juthatunk olyan sarkokhoz, zsákutcákhoz, amelyeket igen nehéz bevonalkázni úgy, hogy az egyenesek távolsága továbbra is egyenletes legyen. Egy lehetőség, hogy a határoló egyeneseken végighaladunk úgy, hogy minden egyes sarokpontot elérjünk. Ennek a módszernek a hátránya az, hogy a számítási idő elviselhetetlenül hosszú lesz.

Szükségünk van egy olyan gondolatra, amely az egész feladatot alapvetően

leegyszerűsíti, ez az a szoftver, ami itt rendelkezésünkre áll. A megoldás egy új gondolkodásmódból áll, amit majd a következő fejezetben közelebbről is megismerünk: a több rétegben való gondolkodás. Ehhez több lépésben végezzük el a feladatunkat. Az első lépésben az egész képernyőt olyanra vonalkázzuk be, amilyenre a későbbiekben szükségünk lesz. A vonalak a megfelelő távolságban haladjanak vagy jobbra, vagy balra emelkedően. Ezeket a paramétereket az FA és FB változók segítségével adjuk meg. A második lépésben megadjuk azt a felületet, amit be kell sraffozni. A mintaprogramban ez egy téglalap, de ez bármilyen, akár szabálytalan felület is lehet. Az a feltétel csupán, hogy egy zárt vonallal kell megadni. Ezt a felületet a számítógép berajzolja a sraffozott képernyőbe. Harmadik lépésként a számítógép megjelöli azokat a vonalakat, amelyek a sraffozandó felületen kívülre esnek. A negyedik lépésben törli a sraffozandó felületen kívül levő vonalakat, így megmaradnak az éppen kívánt sraffozások.

```

10360 REM SRAFFOZÁS
10361 GOTO 10415
10362 REM SRAFFOZÁS ALPROGRAM
10363 FE=INT(318/FA)
10364 FF=INT(198/FA)
10365 IF FB=2 THEN 10390
10366 FOR FD=0 TO FE
10367 FX=318-FD*FA
10368 FY= 2
10369 FU=122-FD*FA
10370 FV=198
10371 IF FX<198 THEN 10374
10372 :LINE FX,FY,FU,FV,1
10373 NEXT FD
10374 FOR FD=0 TO FF
10375 FX=318
10376 FY= 2+FD*FA
10377 FU=122+FD*FA
10378 FV=198
10379 IF FU>318 THEN 10382
10380 :LINE FX,FY,FU,FV,1
10381 NEXT FD
10382 FOR FD=0 TO FF
10383 FX=198-FD*FA
10384 FY= 2
10385 FU= 2
10386 FV=198-FD*FA
10387 IF FV<2 THEN 10414
10388 :LINE FX,FY,FU,FV,1
10389 NEXT FD :GOTO 10414
10390 FOR FD=0 TO FE
10391 FX= 2+FD*FA
10392 FY= 2
10393 FU=198+FD*FA
10394 FV=198

```

```

10395 IF FU>318 THEN 10398
10396 :LINE FX,FY,FU,FU,1
10397 NEXT FD
10398 FOR FD=0 TO FF
10399 FX=122+FD*FA
10400 FY= 2
10401 FU=318
10402 FV=198-FD*FA
10403 IF FX>318 THEN 10406
10404 :LINE FX,FY,FU,FV,1
10405 NEXT FD
10406 FOR FD=0 TO FF
10407 FX= 2
10408 FY= 2+FD*FA
10409 FU=198-FD*FA
10410 FV=198
10411 IF FU< 2 THEN 10414
10412 :LINE FX,FY,FU,FV,1
10413 NEXT FD
10414 RETURN
10415 HIRES 0,7
10416 FA= 20 :REM A SRAFFOZÓVONALAK TÁVOLSÁGA
10417 FB= 1 :REM JOBBRA(1) VAGY BALRA(2) EMELKEDIK
10418 GOSUB 10362
10419 :REC 50,50,100,100,1
10420 :PRINT 0,0,1
10421 :PRINT 0,0,0
10422 :REC 50,50,100,100,1
10423 GOTO 10423

```

Ezt a munkamódszert a több rétegen való rajzolás és a PAINT parancs többféle rajzolási módban való használata teszi lehetővé.

A sraffozott felületen belül is ki lehet jelölni nem sraffozandó felületeket. Ez a felület is tartalmazzon egy kiindulási pontot úgy, ahogy azt a PAINT parancs megkívánja.

Némi gyakorlás után, különösen a KERESZTSRAFFOZÁS programmal, nagyon érdekes hatásokat lehet elérni, pedig csak egy műszaki rajz igényeit akartuk kielégíteni.

A KERESZTSRAFFOZÁS program még egy másik szempontból is figyelmet érdemel. Bemutatja, hogy hogyan kell az ebben a könyvben bemutatott programelemeket úgy módosítani, hogy közvetlenül párbeszédés üzemmódban dolgozhassunk a számítógéppel.

A programokat általában az után a sor után, ahol a RETURN áll, el lehet vágni, és meg lehet változtatni. Ezt mi is így csináltuk. A SRAFFOZÁSOK programba az INPUT és a TEXT parancsokat ily módon építettük be. Az eredmény: a számítógép megkérdezi, hogy milyen távolságra és milyen irányban kívánjuk a sraffozást, vagy be akarjuk-e fejezni a programot. Az 1,2 vagy 3 megadásával lehet válaszolni. Ha másodszorra a másik irányú sraffozást választjuk, akkor kereszt-sraffozást is lehet rajzoltatni, sőt eltérő távolságú vonalakat is kaphatunk.

```
10360 REM KERESZTSRAFFOZÁS
10361 GOTO 10415
10362 REM KERESZTSRAFFOZÁS ALPROGRAM
10363 FE=INT(318/FA)
10364 FF=INT(198/FA)
10365 IF FB=2 THEN 10390
10366 FOR FD=0 TO FE
10367 FX=318-FD*FA
10368 FY= 2
10369 FU=122-FD*FA
10370 FV=198
10371 IF FX<198 THEN 10374
10372 :LINE FX,FY,FU,FV,1
10373 NEXT FD
10374 FOR FD=0 TO FF
10375 FX=318
10376 FY= 2+FD*FA
10377 FU=122+FD*FA
10378 FV=198
10379 IF FU>318 THEN 10382
10380 :LINE FX,FY,FU,FV,1
10381 NEXT FD
10382 FOR FD=0 TO FF
10383 FX=198-FD*FA
10384 FY= 2
10385 FU= 2
10386 FV=198-FD*FA
10387 IF FV<2 THEN 10414
10388 :LINE FX,FY,FU,FV,1
10389 NEXT FD :GOTO 10414
10390 FOR FD=0 TO FE
10391 FX= 2+FD*FA
10392 FY= 2
10393 FU=198+FD*FA
10394 FV=198
10395 IF FU>318 THEN 10398
```

```

10396 :LINE FX,FY,FU,FV,1
10397 NEXT FD
10398 FOR FD=0 TO FF
10399 FX=122+FD*FR
10400 FY= 2
10401 FU=318
10402 FV=198-FD*FR
10403 IF FX>318 THEN 10406
10404 :LINE FX,FY,FU,FV,1
10405 NEXT FD
10406 FOR FD=0 TO FF
10407 FX= 2
10408 FY= 2+FD*FR
10409 FU=198-FD*FR
10410 FV=198
10411 IF FU< 2 THEN 10414
10412 :LINE FX,FY,FU,FV,1
10413 NEXT FD
10414 RETURN
10415 HIRES 0,7
10416 :TEXT 10,190, "A VONALAK TÁVOLSÁGA ?",1,1,8
10417 INPUT FR
10418 :TEXT 10,190, "A VONALAK TÁVOLSÁGA ?",0,1,8
10419 :TEXT 10,190,"JOBBRA=1 BALRA=2 VÉGE=3",1,1,8
10420 INPUT FB
10421 :TEXT 10,190,"JOBBRA=1 BALRA=2 VÉGE=3",0,1,8
10422 IF FB=3 THEN 10429
10423 GOSUB 10362
10424 :REC 10,10,300,180,1
10425 :PRINT 0,0,1
10426 :PRINT 0,0,0
10427 :REC 10,10,300,180,1
10428 GOTO 10416
10429 END

```

Annyiszor futtathatjuk le a programot, ahányszor csak akarjuk, és természetesen a változóknak mindig új és új értékeket adhatunk meg. Itt ezt úgy oldottuk meg, hogy az új változók mindig újabb vonalakat jelentenek, amelyek a régi vonalak között jelennek meg. De olyan programot is írhattunk volna, amely minden futás előtt a régi vonalakat letörli. Most az volt a cél, hogy egy párbeszédés program alapelvét bemutassuk, és példát adjunk arra, hogyan lehet a programelemeket a szükségleteknek megfelelően bővíteni, módosítani.

Figyeljünk arra, hogy a HIRES utasítást nem szabad kétszer lefuttatni, mert a második futásra a finomfelbontású grafikát kikapcsolná. Ezért a KERESZTSRAFFOZÁS programnál a HIRES utasítást a 10415. sorból a 10361. sorba helyeztük át.

Láthattuk, hogyan lehet egy kis trükkel megkönnyíteni az életünket, és érdekes megoldásokhoz jutni. De, mint mindig az életben, majdnem minden trükknek

hátránya is van. Esetünkben a hátrány az, hogy bonyolultabb rajzoknál nem használhatjuk minden korlátozás nélkül a programunkat, mert minden zárt vonalakkal körbe kerített felületet besraffozna. De rögtön a következő trükköt is fel lehet használni: az első rétegen megrajzoljuk a sraffozást, és a második rétegre, ahol a bonyolult rajz található, csak átmásoljuk. Itt megint a rétegtechnikával találkozunk. Ez nagyon fontos a CAD alkalmazásában, ezért a következő fejezetben ezzel még részletesen foglalkozunk.

Az előbbieket miatt a SRAFFOZÁS és a KERESZTSRAFFOZÁS program csak korlátozottan használható a gyakorlatban. Ezekből sokat tanulhatunk, de egy másik programot fogunk használni, a SZUPERSRAFFOZÁS programot.

Kedves Olvasó! Ha sokat foglalkozott a két sraffozóprogrammal, akkor bizonyára olyan felületekre is akadt, amelyeket ugyan be tudott sraffozni, de helyenként a sraffozás a felületen kívül is folytatódott.

Előbb át kellett volna olvasnia az egész fejezetet! Ugyanis most következik az a program, amellyel valóban minden felületet be lehet sraffozni, ill. olyan rajz esetében is használható, amelyen nem kell minden zárt felületet besraffozni. Ehhez a számítógép automatikusan egy külön rétegben megjegyzi a besraffozandó felületet. A sraffozást a rajztól függetlenül is előállíthatjuk, és utána a két képet egymásra lehet helyezni. Természetesen több sraffozást is elő lehet állítani, és ezeknek nem kell egyforma sűrűségűnek lenni.

Ezt a programot SZUPERSRAFFOZÁS-nak nevezzük.

FA-val adjuk meg a sraffozás vonalainak távolságát.

FB-vel adjuk meg a sraffozás irányát:

1 balról jobbra, 2 jobbról balra, 3 keresztben.

Az FC és FD koordinátákkal a sraffozandó felület egy pontját adjuk meg.

Azt, hogy hogyan kell több réteget egymásra rajzolni, azt a 2.2. alfejezetben fogjuk ismertetni.

Ha minden paramétert megadtunk, és elindítottuk a programot, a számítógép megjelöli a sraffozandó felületet. Ezután a program végigtapogatja azokat a vonalakat, amelyek a leendő sraffozásnak megfelelnek. Azért, hogy látványosabb legyen ez a letapogatás, minden letapogatott pontot egy pillanatra kigyújtunk. A gép megjegyzi azokat a pontokat, amelyek a végleges sraffozásnak is pontjai lesznek. Következő lépésben a gép törli a megjelölt felületet, és megrajzolja a kívánt sraffozást. Ezzel készen is vagyunk. A gép tulajdonképpen minden egyes világító pontot megjegyyez, amely az elvi sraffozási vonalon helyezkedik el. Ezeket a pontokat is újra felrajzoljuk, de mivel ezek a pontok már világítottak, így nem történik semmi. Ezt fontos tudni, mert előfordulhat, hogy letöröljük a sraffozást, módosítjuk a rajzot, és ugyanazt a sraffozást újra berajzoljuk. Így ott is megjelenhetnek pontok, ahol arra nem lenne szükség. Ez külön rétegben elhelyezett sraffozásnál nem fordulhat elő. Követlensége miatt célszerű mindig a rétegtechnikát alkalmazni.

```
10360 REM SZUPERSRAFFOZÓ
```

```
10361 GOTO 10415
```



```

10362 REM SZUPERSRAFFOZÓ ALPROGRAM
10363 FT=1:FW=0:DIM FJ(2000),FK(2000):IF FB=2 THEN10411
10364 :PRINT FC,FD,1
10365 FE=INT(320/FA)
10366 FF=INT(200/FA)
10370 FOR FG=1 TO FE
10371 FY=0
10372 FX=FG*FA
10373 LOOP
10374 FX=FX-1*FT
10375 FY=FY+1
10376 EXIT IF FX=0 OR FY=200
10377 FH=TEST(FX,FY):PLOT FX,FY,2:PLOT FX,FY,2
10378 IF FH=1 THEN 10380
10379 GOTO 10383
10380 FI=FI+1
10381 FJ(FI)=FX
10382 FK(FI)=FY
10383 END LOOP
10384 NEXT FG
10385 FOR FV=0 TO FF-1
10386 IF FB=2 OR FW=1 THEN 10389
10387 FX=320
10388 GOTO 10390
10389 FX=0
10390 FY=FV*FA
10391 LOOP
10392 FX=FX-1*FT
10393 FY=FY+1
10394 EXIT IF FY=200 OR FX=0
10395 FM=TEST(FX,FY):PLOT FX,FY,2:PLOT FX,FY,2
10396 IF FM=1 THEN 10398
10397 GOTO 10401
10398 FI=FI+1
10399 FJ(FI)=FX
10400 FK(FI)=FY
10401 END LOOP
10402 NEXT FV
10403 IF FB=3 THEN 10409
10404 :PRINT FC,FD,0
10405 FOR FO=1 TO FI
10406 :PLOT FJ(FO),FK(FO),1
10407 NEXT FO:FI=0
10408 GOTO 10414
10409 FW=FW+1
10410 IF FW=2 THEN 10404
10411 FT=-1
10413 GOTO 10364
10414 RETURN
10415 HIRES 0,7

```

```

10416 FA= 30 :REM A SRAFFOZÓVONALAK TÁVOLSÁGA
10417 FB= 3 :REM 1:JOBBS-, 2:BAL-, 3:KERESZTSRAFFOZÁS
10418 FC=100 :REM A JELZŐPONT X-KOORDINÁTÁJA
10419 FD= 50 :REM A JELZŐPONT Y-KOORDINÁTÁJA
10420 :REC 80,20,100,90,1
10421 GOSUB 10362
10422 GOTO 10422

```

A matematikabarátoknak egy másik megoldási módszert is megemlítünk, aminek a programját azonban nem mutatjuk be, mert BASIC-ben túl sok feladatot ad a gépnek, és így kivárhatatlanul sokáig dolgozik. De a megoldás elve nagyon érdekes, és más esetekben esetleg hasznos lehet.

Aki gépi kódban programozik, kipróbálhatja az itt közölt algoritmust. Ebben a könyvben mindent BASIC-ben szeretnénk megoldani, ezért csak a program elvét közöljük.

A képernyőre nem egy látható, hanem egy képzeletbeli vonalhálót helyezünk. Egy vonalat az

$$Y = n * A + X$$

képlettel jellemezhetünk, ahol A a vonalak távolsága; n megadja, hogy a sok vonal közül melyikről van szó; X és Y azon pontoknak a koordinátái, amelyeken a vonalsereg áthalad.

A sraffozandó felület minden egyes pontját lekérdezzük, és összehasonlítjuk a képzeletbeli hálóval. Ha a lekérdezett pont koordinátái megegyeznek egy képzeletbeli egyenes pontjával, akkor ezt a pontot a rajzban kigyújtjuk.

Ezzel általános eszközt kapunk, amelynek segítségével tetszőleges felületet kívánt módon besraffozhatunk. Egy ilyen program valóban kitűnően működik, azonban BASIC-ben túl sokáig tart.

2.2. A műszaki rajz: kis építőelemek összessége

Most már rendelkezésünkre állnak a legfontosabb segédeszközök, amelyekkel műszaki rajzokat készíthetünk. Ezért kezdjük most hozzá egy összetett műszaki rajz készítéséhez.

Keressünk egy egyszerű testet, ami építőelemeinkből összerakható. Megpróbálunk mindent úgy ábrázolni, ahogyan azt a szabványok előírják, de eltérünk ezektől, ha számítógéppel nehezen megvalósíthatók. A rajzot kiegészítjük méretvonalakkal és feliratokkal, és beszélünk egy keveset a léptékről és az ablakozásról is. Emellett tapasztalatokat szerzünk a programkészletünk használatáról.

A 2.1. alfejezetben láthattuk, hogyan néz ki egy programkészlet. Ebben a 2.1. alfejezetben megtanulunk ezzel élni. Ennek eredménye: egy műszaki rajz. Ne feledjük el, hogy eddig csak alapelemekkel, alapvető segédeszközökkel foglalkoztunk. Ebben a fejezetben a segédeszközeinket finomítjuk, és újakkal is bővítjük.

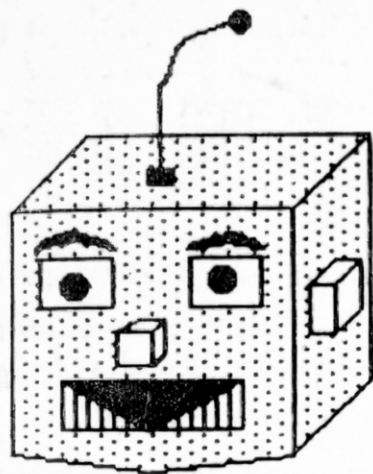
2.2.1. AZ ELEMECTŐL A RAJZIG

Találjunk ki egy olyan ábrát, amely eddigi programjainkból sokat felhasznál. Egy ilyen ábrán el tudjuk magyarázni programjaink alkalmazását. Nem a rajz gyakorlati alkalmazhatóságát tartottuk szem előtt, hanem azt, hogy minél több hasznos tapasztalatot szerezzünk. Nagy hangsúlyt szeretnénk fektetni a rajz teljességére.

Egy tárgyának az ábrázolása a tervezési folyamatnak csak egy lépése. Azt szeretnénk bemutatni, milyen lépéseket kell megtenni, hogy egy műszaki rajzhoz jussunk. Így érthető lesz, hogy a CAD-többet jelent a grafikánál.

Azokra a szabályokra, hogyan kell egy műszaki rajzot felépíteni, nem térünk ki, feltételezzük a szükséges ismereteket. Megmutatjuk, hogy egy műszaki rajzot hogyan lehet a számítógéppel elkészíteni.

2.2.2. A RÉTEGTECHNIKA ALKALMAZÁSA



Egy rajzot több lépésben lehet elkészíteni, és ami lényeges, ezt több rétegre való rajzolással valósítjuk meg. Már megint találkozunk a réteg fogalmával. Ezt most részletesen elmagyarázzuk.

Képzeljük el, hogy a műszaki rajz egyes elemeit külön-külön egy-egy átlátszó fóliára rajzoljuk fel, pl. az ábrázolandó tárgyat, vagy annak kis részleteit egy-egy fóliára. A méretezés (kótázás) egy újabb fóliára kerül. A szövegeket megint egy újabb fóliára írjuk fel, a sraffozásokat pedig egy következőre. A felhasznált rétegek száma kizárólag a rajzzal szemben támasztott követelményektől függ. Erről később még szólnunk.

Képzeljük el, hogy néhány, vagy az összes fóliát egymásra rakjuk. Így teljesen különböző rajzokat kaphatunk különböző információtartalommal. Ezzel a technikával egy egyszerű eszközt kaptunk arra, hogy ugyanabból a néhány rétegből újabb és újabb információkat állítsunk elő. Egy réteget megváltoztathatunk anélkül, hogy ez más rétegeket érintene. Ez a CAD egyik nagy előnye: a rajzok módosítása nagyon egyszerű. Ezt a hagyományos technikánál egyáltalán nem állíthatjuk. A különböző nyelvű feliratokat különböző rétegeken tárolhatjuk. Így különösebb fáradtság nélkül előállíthatunk különböző nyelvű rajzokat, akár magyar, akár német nyelvűeket. Ugyanarról a részről különböző változatokat rajzolhatunk, különbözően méretezhetjük be a rajzot az egyes rétegeken, vagy a rajz alá rajzolhatunk egy rasztert, ami megkönnyíti a tervezést. Ha készen vagyunk, egyszerűen ezt a réteget elhagyhatjuk.

Láthatjuk, hogy milyen előnyei vannak a rétegtechnikának. Azt szeretnénk, ha a számítógépes tervezést azonnal helyesen kezdenék el, és alkalmaznák a rétegtechnikát. Az előbbiek szerint, annyi réteget használhatunk, amennyit csak akarunk. Ez igaz, de némi magyarázatot igényel.

A műszaki gyakorlatban alkalmazott rendszerek általában korlátozott rétegszámmal dolgoznak. Ez a határ általában 100 felett van. Hát akkor hogyan érhetünk el korlátlan számú réteget a mi kis számítógépünkkel? Ennek az a magyarázata, hogy más módon használjuk fel a rétegeket. A professzionális rendszerek belső tárrétegekkel dolgoznak. Ez azt jelenti, hogy a számítógép belső tárában valóban létezik minden egyes réteg matematikailag ábrázolt képe, egy külön erre a célra fenntartott helyen. A belső tár, bármilyen nagy is legyen, egyszer csak betelik, ezért ezek a rendszerek csak meghatározott számú réteggel tudnak dolgozni.

Ennek a technikának a fő előnye, hogy az egyes rétegek megjelenítése nagyon gyors. Mi programrétegeket használunk, mert a számítógépünk csak egyetlen finomfelbontású grafikát tud tárolni. (Ennek a feloldására is készítünk egy eszközt egyik későbbi fejezetben.)

Úgy alakítsuk a programunkat, hogy különálló rétegek keletkezzenek. Ennek csak az a hátránya, hogy több fejtörésre és több türelemre van szükségünk, viszont nem vagyunk néhány rétegre korlátozva.

2.2.2.1. Segédraszter a rajzi koordináták meghatározására

Eddig a koordináták meghatározásához segédeszközként milliméterpapírt és ceruzát használtunk. Ez így egész jól ment, de nem CAD-szerű. Minden segédeszköztől szeretnénk megszabadulni.

A RASZTER program a képernyőnkre helyez egy segédrasztert, amely pontokból és vonalakkól áll. Ennek segítségével közvetlenül a képernyőn határozhatjuk meg a pontok koordinátáit. Tulajdonképpen egy elektronikus milliméterpapírról van szó.

A rajzolás megkezdésekor kirajzoltatjuk a képernyőre ezt a segédrasztert, ezután minden kívánt vonalat és elemet a segédraszter segítségével rajzolhatunk fel. Észrevehetjük, hogy a segédraszterünk nem más, mint egy különállóan kezelhető réteg. Megmutatja, hogy a képernyő melyik részén helyezkednek el az egyes koordináták. A rasztert választhatjuk finomra vagy durvára. A raszterpontok távolsága 1-től 320-ig, ill. 200-ig változtatható, de nem minden érték előnyös. Célszerű a raszterpontok távolságát 5 vagy 10 képpontra választani. Ezen értékeknél az 50, 100, 150 stb. koordinátájú helyeket vonalakkal emeljük ki, és felírjuk a megfelelő számokat is.

A raszter finomságát a GA változóval határozhatjuk meg.

	50	100	150	200	250	300
50						
100						
150						
200						

1. ábra. 5 képpontonkénti segédraszter

GZ a rajzolósi mód változója. Ha 1, akkor rajzoljuk a rasztert, ha 0, akkor töröljük.

A változó majd az INPUT utasítással kapcsolatban kapja meg valódi értelmét. Csak azért használjuk a programunkban, hogy a párbeszédés üzemmódra való átalakítás később könnyebb legyen.

	50	100	150	200	250	300
50						
100						
150						
200						

2. ábra. 10 képpontonkénti segédraszter

```

10430 REM RASZTER
10431 GOTO 10463
10432 REM RASZTER ALPROGRAM
10433 GB=INT(320/GA):GE=0
10434 GC=INT(200/GA)
10435 FOR GD=1 TO GB
10436 GE=GE+GA:GK=0
10437 GF$=STR$(GE)
10438 FOR GJ=1 TO GC
10439 GK=GK+GA
10440 :PLOT GE,GK,GZ
10441 NEXT GJ
10442 IF GE=50 OR GE=100 THEN 10446
10443 IF GE=150 OR GE=200 THEN 10446
10444 IF GE=250 OR GE=300 THEN 10446
10445 GOTO 10448
10446 :LINE GE,0,GE,200,GZ
10447 :TEXT GE-B,0,GF$,1,1,B
10448 NEXT GD
10449 FOR GF=1 TO GC

```

```

10450 GH=GH+GA
10451 GI$=STR$(GH)
10456 IF GH= 50 OR GH=100 THEN 10459
10457 IF GH=150 OR GH=200 THEN 10459
10458 GOTO 10461
10459 :LINE 0,GH,320,GH,GZ
10460 :TEXT 0,GH-B,GI$,1,1,B
10461 NEXT GF
10462 RETURN
10463 HIRES 0,7
10464 GA= 5 :REM A RASZTERVONALAK TÁVOLSÁGA
10465 GZ= 1 :REM RAJZMÓD-JEL
10466 GOSUB 10432
10467 GOTO 10467

```

Természetesen minden egyes programot a RETURN után a saját kívánságainknak megfelelően módosíthatunk.

2.2.2.2. A rajz

A rajz alatt az ábrázolandó test megjelenítését értjük.

A RÉTEG 1 + 2 nevű program egy munkadarabot ábrázol, ami az első két rétegünket jeleníti meg. A rajz ebben az esetben merev, ez azt jelenti, hogy a program segítségével állítjuk elő, nem párbeszéd segítségével.

Először egy alaprétegre van szükségünk, amelyre a többi réteget ráhelyezhetjük.

A 10000-tól 10462-ig terjedő sorokban a segédeszközök állnak rendelkezésre: azok a programelemek, amelyek a rajz elkészítéséhez szükségesek.

Látható, hogy egy kis változtatást vezettünk be. GOSUB helyett EXEC-et, RETURN helyett END PROC-ot használunk. Az alprogramok helyett a procedúrákat (eljárásokat) vezettük be.

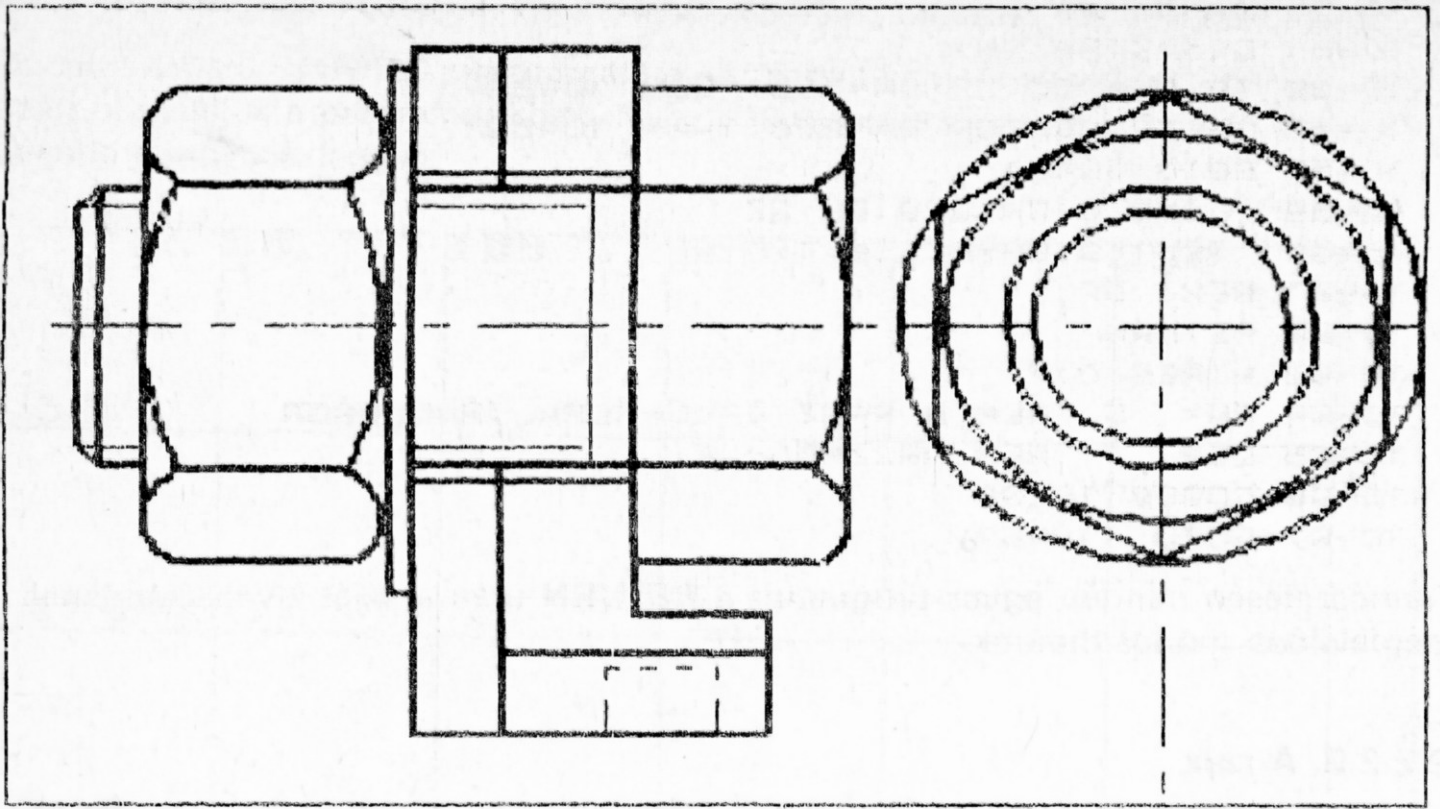
A KOMBI1 procedúra a 10600-tól a 10867-ig terjed. Itt állítjuk elő a programelemekből az első réteget.

Az adatokat párbeszédéses üzemmódban szokták megadni. Itt azért adtuk meg a programban, hogy a rétegtechnika elvét jobban megérthessük.

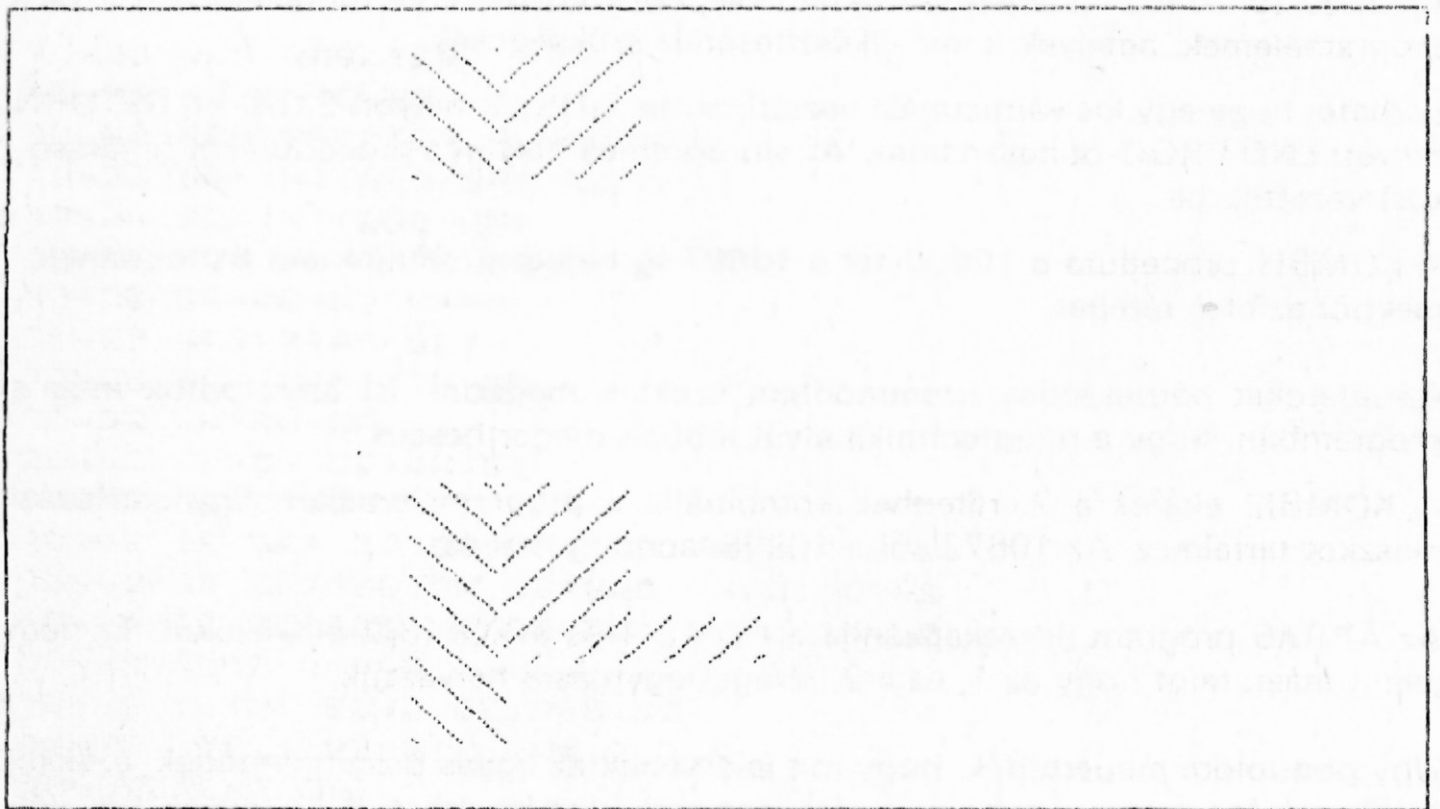
A KOMBI2 eljárás a 2. réteghez kombinálja a programelemeket. Egy sraffozási maszkot tartalmaz. Az 10872-től a 10896. sorokig terjed.

Az ÁBRA5 program összekapcsolja a KOMBI1 és a KOMBI2 eljárásokat. Ez nem jelent mást, mint hogy az 1. és a 2. réteget egymásra helyezzük.

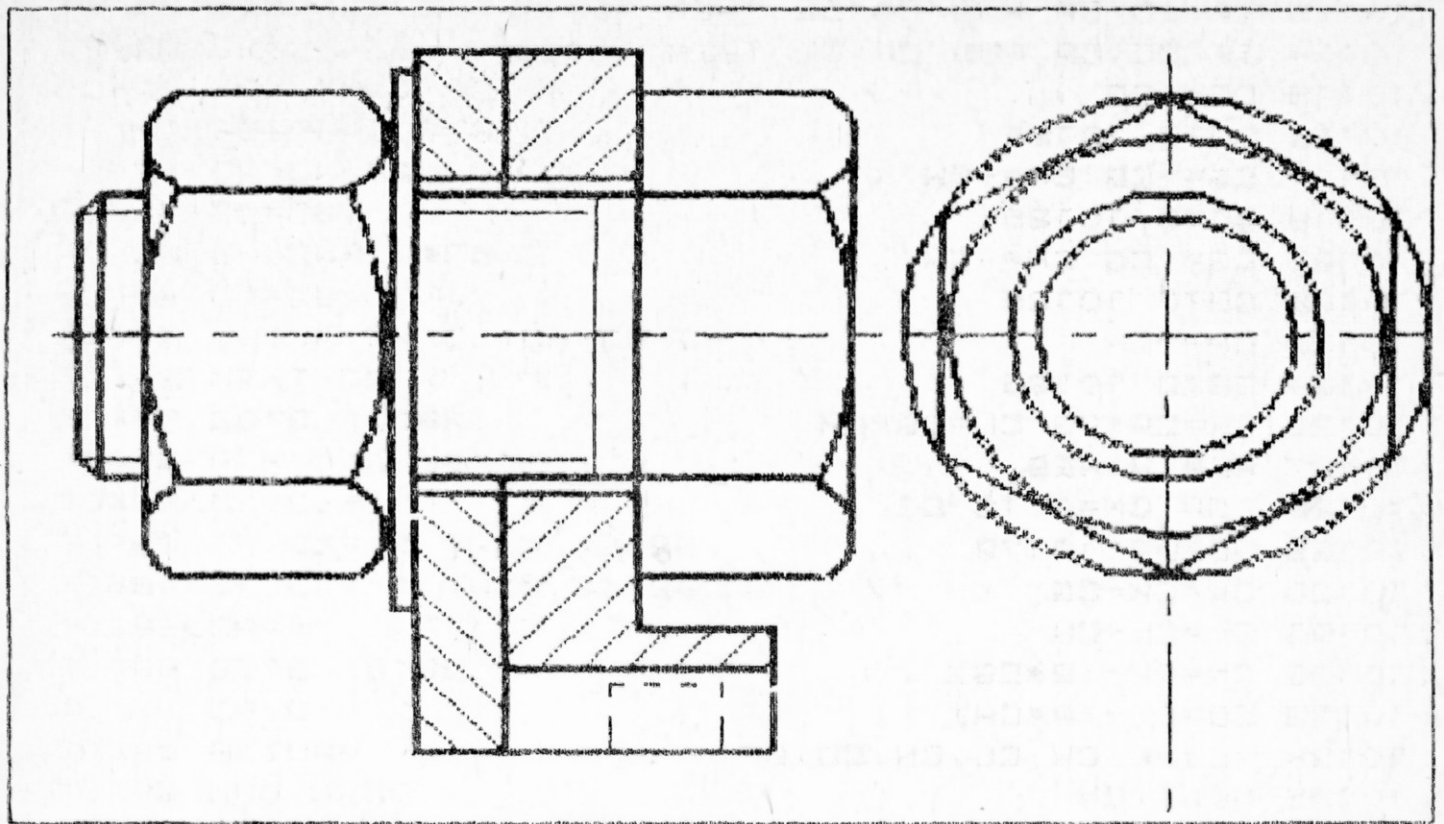
Úgy gondolom megértették, hogy mit jelentenek az egyes programrétegek, ezekkel hogyan dolgozhatunk, és hogy így teljesen szabadon keverhetjük az egyes rétegeket.



3. ábra. Az 1. réteg



4. ábra. A 2. réteg



5. ábra. Az 1. és a 2. réteg

```

10000 REM RÉTEG1+2
10001 GOTO 10900
10002 PROC VASTAGVONAL
10003 IF AB=AD THEN 10012
10004 FOR AJ=1 TO AE
10005 AF=RA+AJ:AG=AE
10007 AH=AC+AJ:AI=AD
10009 :LINE AF,AG,AH,AI,1
10010 NEXT AJ
10011 GOTO 10019
10012 FOR AK=1 TO AE
10013 AG=AB+AK:AF=RA
10015 AI=AD+AK:AH=AC
10017 :LINE AF,AG,AH,AI,1
10018 NEXT AK
10019 END PROC
10102 PROC SZAGGATOTTUDNAL
10103 CE=ABS(CC-CA)
10104 CF=ABS(CB-CD)
10105 IF CA=CC THEN 10137
10106 IF CB=CD THEN 10158
10107 CG=INT(SQR((16*CE^2)/(CF^2+CE^2)))
10108 CH=INT((CG*CF)/CE)
10109 CI=INT(CE/CG)
10110 CJ=-1
10111 IF CA<CC AND CD<CB THEN 10115
10112 IF CA<CC AND CB<CD THEN 10117

```

```

10113 IF CC<CA AND CD<CB THEN 10120
10114 IF CC<CA AND CB<CD THEN 10123
10115 CG=-CG
10116 GOTO 10125
10117 CG=-CG:CH=-CH
10119 GOTO 10125
10120 CG= CG:CH= CH
10122 GOTO 10125
10123 CH=-CH
10124 GOTO 10125
10125 CK=CA+CG:CL=CB+CH
10127 REM HIRES
10128 FOR CM=1 TO CI
10129 GOSUB 10178
10130 CK=CK-CG
10131 CL=CL-CH
10132 CN=CK-(2*CG)
10133 CO=CL-(2*CH)
10134 :LINE CK,CL,CN,CO,CP
10135 NEXT CM
10136 GOTO 10186
10137 REM FÜGGÖLEGES
10138 CI=INT(CF/4)
10139 CJ=-1
10140 CH= 4
10141 IF CB<CD THEN 10143
10142 IF CD<CB THEN 10145
10143 CH=-CH
10144 GOTO 10147
10145 CH= CH
10146 GOTO 10147
10147 CL=CB+CH
10148 FOR CM=1 TO CI
10150 GOSUB 10178
10151 CK=CA
10152 CL=CL-CH
10153 CN=CA
10154 CO=CL-(2*CH)
10155 :LINE CK,CL,CN,CO,CP
10156 NEXT CM
10157 GOTO 10186
10158 REM VÍZSZINTES
10159 CI=INT(CE/4)
10160 CJ=-1
10161 CG= 4
10162 IF CA<CC THEN 10164
10163 IF CC<CA THEN 10166
10164 CG=-CG
10165 GOTO 10167
10166 CG= CG

```

```

10167 REM HIRES
10168 CK=CR+CG
10169 FOR CM=1 TO CI
10170 GOSUB 10178
10171 CK=CK-CG
10172 CL=CB
10173 CN=CK-(2*CG)
10174 CO=CB
10175 LINE CK,CL,CN,CO,CP
10176 NEXT CM
10177 GOTO 10186
10178 REM CIRKUS
10179 CJ=CJ*(-1)
10180 IF CJ= 1 THEN 10182
10181 IF CJ=-1 THEN 10184
10182 CP=1
10183 GOTO 10185
10184 CP=0
10185 RETURN
10186 END PROC
10202 PROC TENGELYVONRL U1
10203 DH$="666666666666116611"
10204 DI=INT(DE/16)
10205 DJ$=DUP(DH$,DI)
10206 :ROT DC,DD
10207 :DRAW DJ$,DR,DB,1
10208 END PROC
10282 PROC KETTŐS MÉRETNYÍL
10283 EF$="6068836555506888888836555555553333"
10284 EJ$="375507888883755555507888888888"
10285 EG$="6"
10286 EH$=DUP(EG$,EE)
10287 EI$=EF$+EH$+EJ$
10289 :ROT EC,ED
10290 :DRAW EI$,ER,EB,1
10291 END PROC
10312 PROC KÓTA
10313 EF$="6068836555506888888836555555553333"
10314 EJ$="375507888883755555507888888888555511116"
10315 EG$="6"
10316 EH$=DUP(EG$,EE)
10317 EK$="5"
10318 EL$=DUP(EK$,EM)
10319 IF EN=2 THEN 10326
10320 EO$="555333"
10321 EP$=EL$+EO$
10322 EQ$="8"
10323 ER$=DUP(EQ$,EM)
10324 ES$=EP$+EF$+EK$+EJ$+EO$+ER$
10325 GOTO 10332

```

```

10326 ET$="B"
10327 EU$=DUP(ET$,EM)
10328 EV$="888000"
10329 EW$=EU$+EV$
10330 ES$=EW$+EF$+EH$+EJ$+EU$+EL$
10332 :ROT EC,ED
10333 :DRAW ES$,EA,EB,1
10334 END PROC
10360 PROC SRAFFOZAS
10361 FQ=FQ+1:IF FQ>1 THEN 10363
10362 DIM FJ(2000):DIM FK(2000)
10363 FT=1:FW=0:IF FB=2 THEN 10411
10364 :PRINT FC,FD,1
10365 FE=INT(320/FA)
10366 FF=INT(200/FA)
10368 FY=2
10369 FU=122-FD*FA
10370 FOR FG=1 TO FE
10371 FY=0
10372 FX=FG*FA
10373 LOOP
10374 FX=FX-1*FT
10375 FY=FY+1
10376 EXIT IF FX=0 OR FY=200
10377 FH=TEST(FX,FY):PLOT FX,FY,2:PLOT FX,FY,2
10378 IF FH=1 THEN 10380
10379 GOTO 10383
10380 FI=FI+1
10381 FJ(FI)=FX
10382 FK(FI)=FY
10383 END LOOP
10384 NEXT FG
10385 FOR FV=0 TO FF-1
10386 IF FB=2 OR FW=1 THEN 10389
10387 FX=320
10388 GOTO 10390
10389 FX=0
10390 FY=FV*FA
10391 LOOP
10392 FX=FX-1*FT
10393 FY=FY+1
10394 EXIT IF FY=200 OR FX=0
10395 FM=TEST(FX,FY):PLOT FX,FY,2:PLOT FX,FY,2
10396 IF FM=1 THEN 10398
10397 GOTO 10401
10398 FI=FI+1
10399 FJ(FI)=FX
10400 FK(FI)=FY
10401 END LOOP
10402 NEXT FV

```

```

10403 IF FB=3 THEN 10409
10404 :PRINT FC,FD,0
10405 FOR FD=1 TO FI
10406 :PLOT FJ(FD),FK(FD),1
10407 NEXT FD:FI=0
10408 GOTO 10414
10409 FW=FW+1
10410 IF FW=2 THEN 10404
10411 FT=-1
10413 GOTO 10364
10414 END PROC
10432 PROC RASZTER
10433 GB=INT(320/GA):GE=0
10434 GC=INT(200/GA)
10435 FOR GD=1 TO GB
10436 GE=GE+GA:GK=0
10437 GF$=STR$(GE)
10438 FOR GJ=1 TO GC
10439 GK=GK+GA
10440 :PLOT GE,GK,GZ
10441 NEXT GJ
10442 IF GE=50 OR GE=100 THEN 10446
10443 IF GE=150 OR GE=200 THEN 10446
10444 IF GE=250 OR GE=300 THEN 10446
10445 GOTO 10448
10446 :LINE GE,0,GE,200,GZ
10447 :TEXT GE-B,0,GF$,1,1,B
10448 NEXT GD
10449 FOR GF=1 TO GC
10450 GH=GH+GA:GU=0
10451 GI$=STR$(GH)
10456 IF GH=50 OR GH=100 THEN 10459
10457 IF GH=150 OR GH=200 THEN 10459
10458 GOTO 10461
10459 :LINE 0,GH,320,GH,GZ
10460 :TEXT 0,GH-B,GI$,1,1,B
10461 NEXT GF
10462 END PROC
10597 REM *****
10598 REM RÉTEG1
10599 REM *****
10600 PROC KOMBI1
10601 DR=10:DB=80:DC=2:DD=3:DE=240
10607 EXEC TENGELYVONAL U1
10608 DR=260:DB=10:DC=4:DD=3:DE=80
10613 EXEC TENGELYVONAL U1
10614 AR=15:AB=50:AC=15:AD=110:AE=2
10619 EXEC VASTAGVONAL
10620 AR=20:AB=45:AC=20:AD=115:AE=2
10625 EXEC VASTAGVONAL

```

10626 AA= 15:AB= 50:AC= 20:AD= 45:AE=2
 10631 EXEC VASTAGVONAL
 10632 AA= 15:AB=110:AC= 20:AD=115:AE=2
 10637 EXEC VASTAGVONAL
 10638 AA= 21:AB= 44:AC= 30:AD= 44:AE=2
 10643 EXEC VASTAGVONAL
 10644 AA= 21:AB=113:AC= 30:AD=113:AE=2
 10649 EXEC VASTAGVONAL
 10650 AA= 16:AB= 49:AC= 30:AD= 49:AE=1
 10655 EXEC VASTAGVONAL
 10656 AA= 16:AB=109:AC= 30:AD=109:AE=1
 10661 EXEC VASTAGVONAL
 10662 AA= 30:AB= 32:AC= 30:AD=128:AE=2
 10667 EXEC VASTAGVONAL
 10668 AA= 85:AB= 15:AC= 85:AD=145:AE=2
 10673 EXEC VASTAGVONAL
 10674 AA= 38:AB= 19:AC= 78:AD= 19:AE=2
 10679 EXEC VASTAGVONAL
 10680 AA= 38:AB= 43:AC= 78:AD= 43:AE=2
 10685 EXEC VASTAGVONAL
 10686 AA= 38:AB=114:AC= 78:AD=114:AE=2
 10691 EXEC VASTAGVONAL
 10692 AA= 38:AB=137:AC= 78:AD=137:AE=2
 10697 EXEC VASTAGVONAL
 10698 :ARC 115, 80,245,295,3,85,85,1
 10699 :ARC 115, 80,245,295,3,84,84,1
 10700 :ARC 0, 80, 65,115,3,85,85,1
 10701 :ARC 0, 80, 65,115,3,86,86,1
 10702 :ARC 43, 32,205,335,3,13,13,1
 10703 :ARC 43, 32,205,335,3,12,12,1
 10704 :ARC 72, 32, 25,155,3,13,13,1
 10705 :ARC 72, 32, 25,155,3,14,14,1
 10706 :ARC 43,128,205,335,3,13,13,1
 10707 :ARC 43,128,205,335,3,12,12,1
 10708 :ARC 72,128, 25,155,3,13,13,1
 10709 :ARC 72,128, 25,145,3,14,14,1
 10710 AA= 86:AB= 14:AC= 90:AD= 14:AE=2
 10715 EXEC VASTAGVONAL
 10716 AA= 86:AB=145:AC= 90:AD=145:AE=2
 10721 EXEC VASTAGVONAL
 10722 AA= 90:AB= 10:AC= 90:AD=180:AE=2
 10727 EXEC VASTAGVONAL
 10728 AA= 91:AB= 9:AC=140:AD= 9:AE=2
 10733 EXEC VASTAGVONAL
 10734 AA=140:AB= 10:AC=140:AD=150:AE=2
 10739 EXEC VASTAGVONAL
 10740 AA=141:AB=150:AC=170:AD=150:AE=2
 10745 EXEC VASTAGVONAL
 10746 AA=170:AB=151:AC=170:AD=181:AE=2
 10751 EXEC VASTAGVONAL

10752 AA= 91:AB=180:AC=171:AD=180:AE=2
10757 EXEC VASTAGUONAL
10758 AA= 91:AB= 40:AC=141:AD= 40:AE=2
10763 EXEC VASTAGUONAL
10764 AA= 91:AB= 44:AC=182:AD= 44:AE=2
10769 EXEC VASTAGUONAL
10770 AA= 91:AB= 49:AC=130:AD= 49:AE=1
10775 EXEC VASTAGUONAL
10776 AA= 91:AB=109:AC=130:AD=109:AE=1
10781 EXEC VASTAGUONAL
10782 AA= 91:AB=113:AC=182:AD=113:AE=2
10787 EXEC VASTAGUONAL
10788 AA= 91:AB=117:AC=141:AD=117:AE=2
10793 EXEC VASTAGUONAL
10794 AA=110:AB= 10:AC=110:AD= 44:AE=2
10799 EXEC VASTAGUONAL
10800 AA=110:AB=114:AC=110:AD=180:AE=2
10805 EXEC VASTAGUONAL
10806 AA=131:AB= 45:AC=131:AD=113:AE=1
10811 EXEC VASTAGUONAL
10812 AA=141:AB= 19:AC=182:AD= 19:AE=2
10817 EXEC VASTAGUONAL
10818 AA=141:AB=137:AC=182:AD=137:AE=2
10823 EXEC VASTAGUONAL
10824 AA=188:AB= 32:AC=188:AD=128:AE=2
10829 EXEC VASTAGUONAL
10830 AA=113:AB=160:AC=170:AD=160:AE=2
10835 EXEC VASTAGUONAL
10836 CA=135:CB=165:CC=160:CD=165
10840 EXEC SZAGGATOTTUONAL
10841 CA=135:CB=165:CC=135:CD=180
10845 EXEC SZAGGATOTTUONAL
10846 CA=160:CB=165:CC=160:CD=180
10850 EXEC SZAGGATOTTUONAL
10851 :ARC 105, 80, 65,115,1,85,85,1
10852 :ARC 105, 80, 65,115,1,86,86,1
10853 :ARC 177, 32, 25,155,1,13,13,1
10854 :ARC 177, 32, 25,155,1,14,14,1
10855 :ARC 177,128, 25,155,1,13,13,1
10856 :ARC 177,128, 25,145,1,14,14,1
10857 :CIRCLE 260,80,30,30,1
10858 :CIRCLE 260,80,29,29,1
10859 :CIRCLE 260,80,35,35,1
10860 :CIRCLE 260,80,36,36,1
10861 :CIRCLE 260,80,60,60,1
10862 :CIRCLE 260,80,59,59,1
10863 :ARC 260, 80, 0,360,60,59,59,1
10864 :ARC 260, 80, 0,360,60,58,58,1
10865 :CIRCLE 260,80,50,50,1
10866 :CIRCLE 260,80,49,49,1

```

10867 END PROC
10868 REM *****
10870 REM RÉTEG2
10871 REM *****
10872 PROC KOMBI2
10874 :REC 91,10,22,32,1
10875 FR= 10:FB= 2:FC=100:FD= 20
10879 EXEC SRAFFOZÁS
10880 :REC 91,118,21,64,1
10881 FR= 10:FB= 2:FC=100:FD=150
10882 EXEC SRAFFOZÁS
10883 :REC 112, 10,30,32,1
10884 FR= 10:FB= 1:FC=120:FD= 20
10887 EXEC SRAFFOZÁS
10888 :LINE 112,118,141,118,1
10889 :LINE 112,118,112,160,1
10890 :LINE 112,162,170,162,1
10891 :LINE 170,162,170,151,1
10892 :LINE 170,151,140,151,1
10893 :LINE 140,151,140,117,1
10894 FR= 10:FB= 1:FC=120:FD=150
10895 EXEC SRAFFOZÁS
10896 END PROC
10897 REM *****
10898 REM *****
10900 REM ÁBRAS
10901 HIRÉS 0,7
10902 EXEC KOMBI2
10903 EXEC KOMBI1
10904 GOTO 11006

```

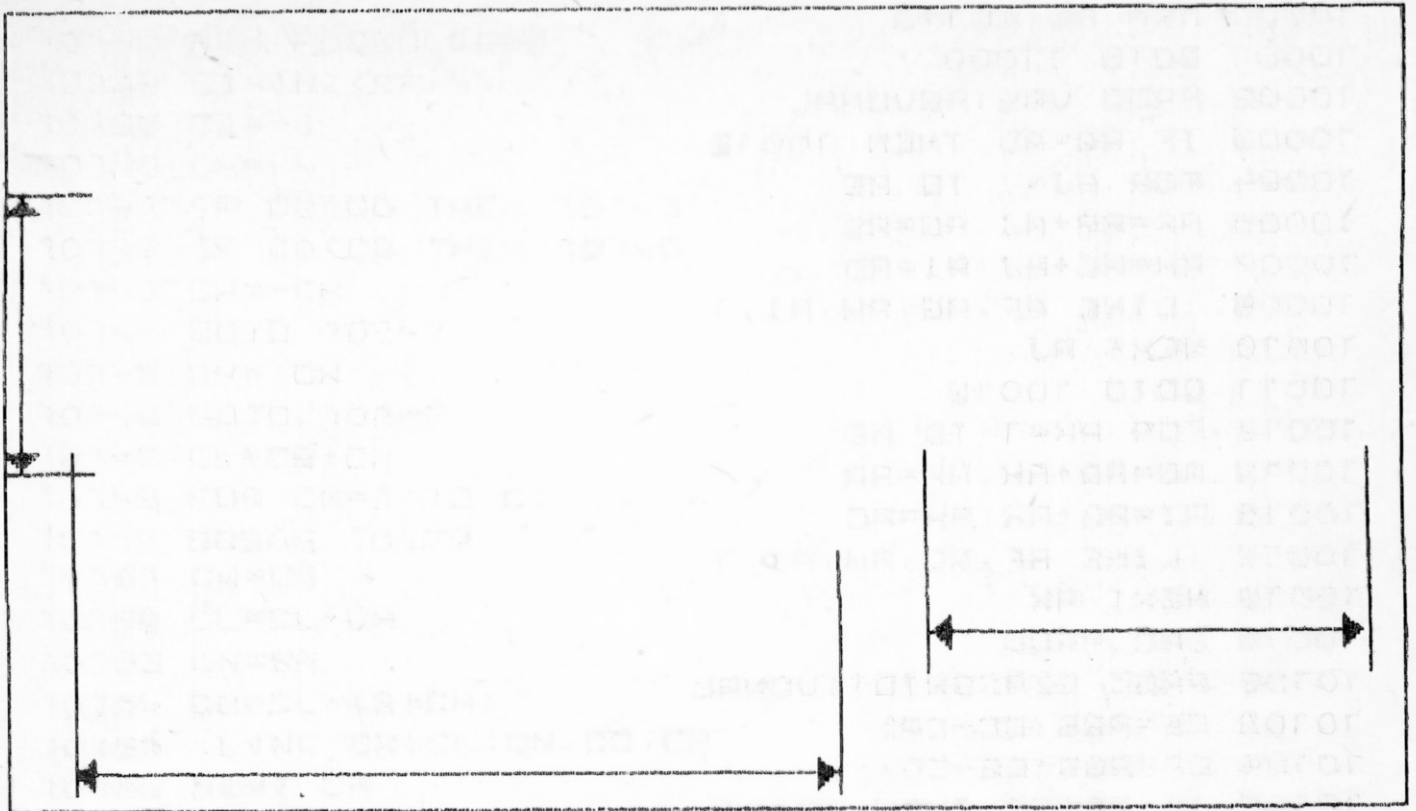
2.2.2.3. Méretezés (kótázás)

A méretezéshez egy további réteget, a 3. réteget vezetjük be. Ezt a KOMBI3 eljárás segítségével állítjuk elő.

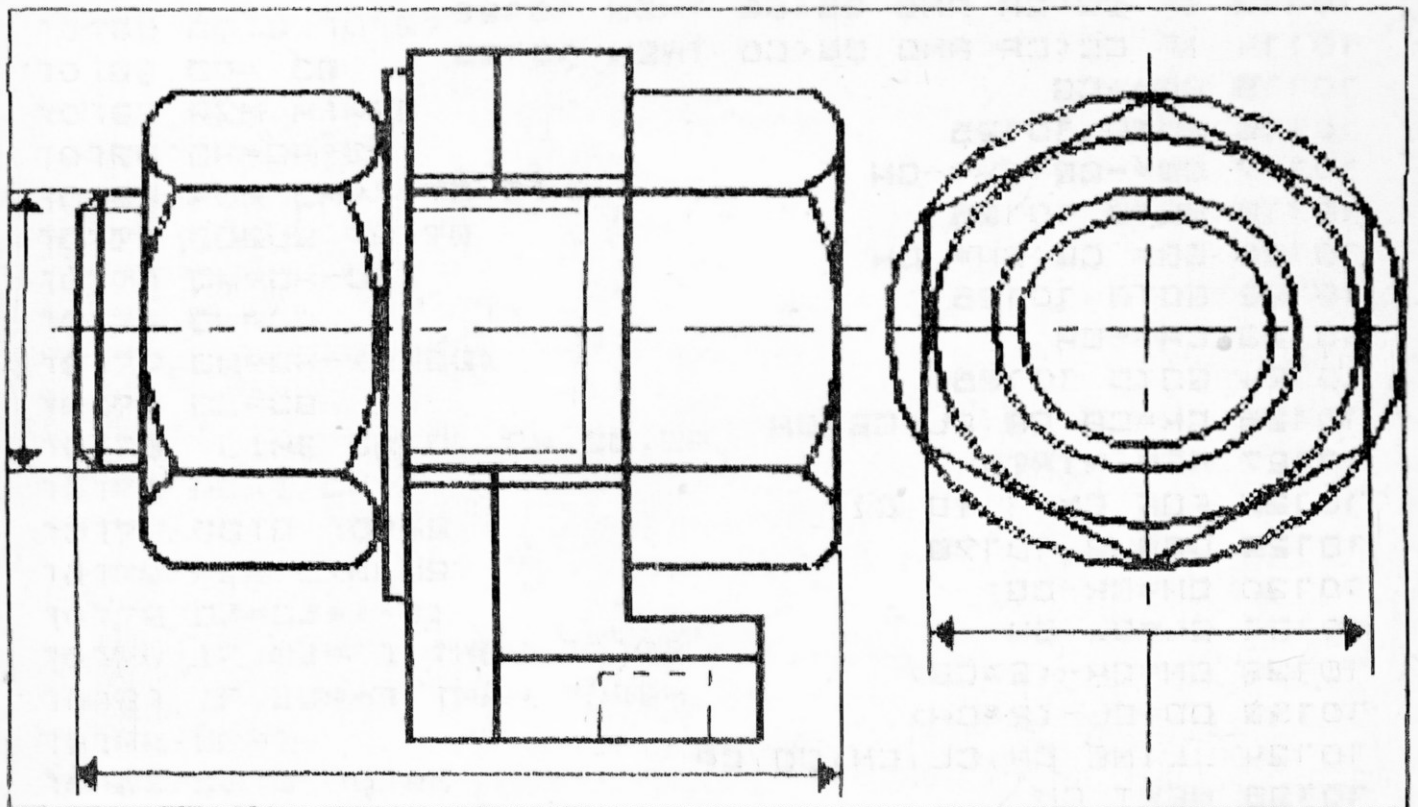
Ha megnézzük a RÉTEG 1 + 3 programot, megfigyelhetjük, hogy a 10001. sor kivételével a 10896. sorig azonos a RÉTEG 1 + 2 programmal.

Hozzáírtuk a programhoz a KOMBI3 eljárást, és az ÁBRA7 program az 1. és a 2. réteg helyett az 1. és a 3. réteget keveri.

Ha akartuk volna, a 2. réteget is felrajzolhattuk volna. Mivel csak az elvet kívánjuk bemutatni, ettől most eltekintettünk. Láthatjuk, hogy nagyon jól fel tudjuk használni és nagyon flexibilisen tudjuk alkalmazni programkészletünket. Természetesen a 10001. sorban egy másik ugrási cím szerepel. Ez azért kell, mert az ÁBRA7 program a KOMBI3 eljárás beírásával magasabb címekre tolódott.



6. ábra. A 3. réteg



7. ábra. Az 1. és a 3. réteg

```

10000 REM RETEG1+3
10001 GOTO 11000
10002 PROC VASTAGUONAL
10003 IF AB=AD THEN 10012
10004 FOR AJ=1 TO AE
10005 AF=AR+AJ:RG=AB
10007 AH=AC+AJ:AI=AD
10009 :LINE AF,RG,AH,AI,1
10010 NEXT AJ
10011 GOTO 10019
10012 FOR AK=1 TO AE
10013 RG=AB+AK:AF=AR
10015 AI=AD+AK:AH=AC
10017 :LINE AF,RG,AH,AI,1
10018 NEXT AK
10019 END PROC
10102 PROC SZAGGATOTTUONAL
10103 CE=ABS<CC-CA>
10104 CF=ABS<CB-CD>
10105 IF CA=CC THEN 10137
10106 IF CB=CD THEN 1015B
10107 CG=INT<SQRT<(16*CE^2)/<CF^2+CE^2>>>
10108 CH=INT<<CG*CF>/CE>
10109 CI=INT<CE/CG>
10110 CJ=-1
10111 IF CA<CC AND CD<CB THEN 10115
10112 IF CA<CC AND CB<CD THEN 10117
10113 IF CC<CA AND CD<CB THEN 10120
10114 IF CC<CA AND CB<CD THEN 10123
10115 CG=-CG
10116 GOTO 10125
10117 CG=-CG:CH=-CH
10119 GOTO 10125
10120 CG=CG:CH=CH
10122 GOTO 10125
10123 CH=-CH
10124 GOTO 10125
10125 CK=CA+CG:CL=CB+CH
10127 REM HIRES
10128 FOR CM=1 TO CI
10129 GOSUB 10178
10130 CK=CK-CG
10131 CL=CL-CH
10132 CN=CK-<2*CG>
10133 CO=CL-<2*CH>
10134 :LINE CK,CL,CN,CO,CP
10135 NEXT CM
10136 GOTO 10186

```

```

10137 REM FÜGGŐLEGES
10138 CI=INT(CF/4)
10139 CJ=-1
10140 CH= 4
10141 IF CB<CD THEN 10143
10142 IF CD<CB THEN 10145
10143 CH=-CH
10144 GOTO 10147
10145 CH= CH
10146 GOTO 10147
10147 CL=CB+CH
10149 FOR CM=1 TO CI
10150 GOSUB 10178
10151 CK=CA
10152 CL=CL-CH
10153 CN=CA
10154 CD=CL-(2*CH)
10155 :LINE CK,CL,CN,CD,CP
10156 NEXT CM
10157 GOTO 10186
10158 REM VÍZSZINTES
10159 CI=INT(CE/4)
10160 CJ=-1
10161 CG= 4
10162 IF CA<CC THEN 10164
10163 IF CC<CA THEN 10166
10164 CG=-CG
10165 GOTO 10167
10166 CG= CG
10167 REM HIRES
10168 CK=CA+CG
10169 FOR CM=1 TO CI
10170 GOSUB 10178
10171 CK=CK-CG
10172 CL=CB
10173 CN=CK-(2*CG)
10174 CD=CB
10175 :LINE CK,CL,CN,CD,CP
10176 NEXT CM
10177 GOTO 10186
10178 REM CIKLUS
10179 CJ=CJ*(-1)
10180 IF CJ= 1 THEN 10182
10181 IF CJ=-1 THEN 10184
10182 CP=1
10183 GOTO 10185
10184 CP=0
10185 RETURN

```

```

10186 END PROC
10202 PROC TENGELYVONAL V1
10203 DH$="666666666666116611"
10204 DI=INT<DE/16>
10205 DJ$=DUP<DH$,DI>
10206 :ROT DC,DD
10207 :DRAW DJ$,DA,DB,1
10208 END PROC
10282 PROC KETTŐS MÉRETNYIL
10283 EF$="60688365555068888888836555555553333"
10284 EJ$="37550788883755555550788888888"
10285 EG$="6"
10286 EH$=DUP<EG$,EE>
10287 EI$=EF$+EH$+EJ$
10288 :ROT EC,ED
10289 :DRAW EI$,EA,EB,1
10291 END PROC
10312 PROC KÓTA
10313 EF$="60688365555068888888836555555553333"
10314 EJ$="37550788883755555550788888888555511116"
10315 EG$="6"
10316 EH$=DUP<EG$,EE>
10317 EK$="5"
10318 EL$=DUP<EK$,EM>
10319 IF EN=2 THEN 10326
10320 EQ$="555333"
10321 EP$=EL$+EQ$
10322 ER$="8"
10323 ER$=DUP<ER$,EM>
10324 ES$=EP$+EF$+EH$+EJ$+EQ$+ER$
10325 GOTO 10332
10326 ET$="8"
10327 EU$=DUP<ET$,EM>
10328 EV$="888000"
10329 EW$=EU$+EV$
10330 ES$=EW$+EF$+EH$+EJ$+EV$+EL$
10332 :ROT EC,ED
10333 :DRAW ES$,EA,EB,1
10334 END PROC
10360 PROC SRAFFOZÁS
10361 FG=FG+1:IF FG>1 THEN 10363
10362 DIM FJ<2000>:DIM FK<2000>
10363 FT=1:FW=0:IF FB=2 THEN 10411
10364 :PRINT FC,FD,1
10365 FE=INT<320/FA>
10366 FF=INT<200/FA>
10368 FY=2
10369 FU=122-FD*FA
10370 FOR FG=1 TO FE

```

```

10371 FY=0
10372 FX=FG*FR
10373 LOOP
10374 FX=FX-1*FT
10375 FY=FY+1
10376 EXIT IF FX=0 OR FY=200
10377 FH=TEST<FX,FY>:PLOT FX,FY,2:PLOT FX,FY,2
10378 IF FH=1 THEN 10380
10379 GOTO 10383
10380 FI=FI+1
10381 FJ<FI>=FX
10382 FK<FI>=FY
10383 END LOOP
10384 NEXT FG
10385 FOR FV=0 TO FF-1
10386 IF FB=2 OR FW=1 THEN 10389
10387 FX=320
10388 GOTO 10390
10389 FX=0
10390 FY=FV*FR
10391 LOOP
10392 FX=FX-1*FT
10393 FY=FY+1
10394 EXIT IF FY=200 OR FX=0
10395 FM=TEST<FX,FY>:PLOT FX,FY,2:PLOT FX,FY,2
10396 IF FM=1 THEN 10398
10397 GOTO 10401
10398 FI=FI+1
10399 FJ<FI>=FX
10400 FK<FI>=FY
10401 END LOOP
10402 NEXT FV
10403 IF FB=3 THEN 10409
10404 :PRINT FD,FD,0
10405 FOR FD=1 TO FI
10406 :PLOT FJ<FD>,FK<FD>,1
10407 NEXT FD:FI=0
10408 GOTO 10414
10409 FW=FW+1
10410 IF FW=2 THEN 10404
10411 FT=-1
10413 GOTO 10364
10414 END PROC
10432 PROC RASZTER
10433 GB=INT<320/GA>:GE=0
10434 GC=INT<200/GA>
10435 FOR GD=1 TO GB
10436 GE=GE+GA:GK=0
10437 GF$=STR$(GE)
10438 FOR GJ=1 TO GC

```

```

10439 GK=GK+GA
10440 :PLOT GE,GK,GZ
10441 NEXT GJ
10442 IF GE=50 OR GE=100 THEN 10446
10443 IF GE=150 OR GE=200 THEN 10446
10444 IF GE=250 OR GE=300 THEN 10446
10445 GOTO 10448
10446 :LINE GE,0,GE,200,GZ
10447 :TEXT GE-8,0,GF$,1,1,8
10448 NEXT GD
10449 FOR GF=1 TO GC
10450 GH=GH+GA:GU=0
10451 GI$=STR$(GH)
10456 IF GH=50 OR GH=100 THEN 10459
10457 IF GH=150 OR GH=200 THEN 10459
10458 GOTO 10461
10459 :LINE 0,GH,320,GH,GZ
10460 :TEXT 0,GH-8,GI$,1,1,8
10461 NEXT GF
10462 END PROC
10597 REM *****
10598 REM RETEG1
10599 REM *****
10600 PROC KOMBIT
10601 DA=10:DB=80:DC=2:DD=3:DE=240
10607 EXEC TENGELYONAL U1
10608 DA=280:DB=10:DC=4:DD=3:DE=80
10613 EXEC TENGELYONAL U1
10614 AA=15:AB=50:AC=15:AD=110:AE=2
10619 EXEC VASTAGVONAL
10620 AA=20:AB=45:AC=20:AD=115:AE=2
10625 EXEC VASTAGVONAL
10626 AA=15:AB=50:AC=20:AD=45:AE=2
10631 EXEC VASTAGVONAL
10632 AA=15:AB=110:AC=20:AD=115:AE=2
10637 EXEC VASTAGVONAL
10638 AA=21:AB=44:AC=30:AD=44:AE=2
10643 EXEC VASTAGVONAL
10644 AA=21:AB=113:AC=30:AD=113:AE=2
10649 EXEC VASTAGVONAL
10650 AA=16:AB=49:AC=30:AD=49:AE=1
10655 EXEC VASTAGVONAL
10656 AA=16:AB=109:AC=30:AD=109:AE=1
10661 EXEC VASTAGVONAL
10662 AA=30:AB=32:AC=30:AD=128:AE=2
10667 EXEC VASTAGVONAL
10668 AA=85:AB=15:AC=85:AD=145:AE=2
10673 EXEC VASTAGVONAL

```

10674 AA= 38:AB= 19:AC= 78:AD= 19:AE=2
 10679 EXEC VASTAGVONAL
 10680 AA= 38:AB= 43:AC= 78:AD= 43:AE=2
 10685 EXEC VASTAGVONAL
 10686 AA= 38:AB=114:AC= 78:AD=114:AE=2
 10691 EXEC VASTAGVONAL
 10692 AA= 38:AB=137:AC= 78:AD=137:AE=2
 10697 EXEC VASTAGVONAL
 10698 :ARC 115, 80,245,295,3,85,85,1
 10699 :ARC 115, 80,245,295,3,84,84,1
 10700 :ARC 0, 80, 65,115,3,85,85,1
 10701 :ARC 0, 80, 65,115,3,86,86,1
 10702 :ARC 42, 32,205,335,3,13,13,1
 10703 :ARC 43, 32,205,335,3,12,12,1
 10704 :ARC 72, 32, 25,155,3,13,13,1
 10705 :ARC 72, 32, 25,155,3,14,14,1
 10706 :ARC 43,128,205,335,3,13,13,1
 10707 :ARC 43,128,205,335,3,12,12,1
 10708 :ARC 72,128, 25,155,3,13,13,1
 10709 :ARC 72,128, 25,145,3,14,14,1
 10710 AA= 86:AB= 14:AC= 90:AD= 14:AE=2
 10715 EXEC VASTAGVONAL
 10716 AA= 86:AB=145:AC= 90:AD=145:AE=2
 10721 EXEC VASTAGVONAL
 10722 AA= 90:AB= 10:AC= 90:AD=180:AE=2
 10727 EXEC VASTAGVONAL
 10728 AA= 91:AB= 9:AC=140:AD= 9:AE=2
 10733 EXEC VASTAGVONAL
 10734 AA=140:AB= 10:AC=140:AD=150:AE=2
 10739 EXEC VASTAGVONAL
 10740 AA=141:AB=150:AC=170:AD=150:AE=2
 10745 EXEC VASTAGVONAL
 10746 AA=170:AB=151:AC=170:AD=181:AE=2
 10751 EXEC VASTAGVONAL
 10752 AA= 91:AB=180:AC=171:AD=180:AE=2
 10757 EXEC VASTAGVONAL
 10758 AA= 91:AB= 40:AC=141:AD= 40:AE=2
 10763 EXEC VASTAGVONAL
 10764 AA= 91:AB= 44:AC=182:AD= 44:AE=2
 10769 EXEC VASTAGVONAL
 10770 AA= 91:AB= 49:AC=130:AD= 49:AE=1
 10775 EXEC VASTAGVONAL
 10776 AA= 91:AB=109:AC=130:AD=109:AE=1
 10781 EXEC VASTAGVONAL
 10782 AA= 91:AB=113:AC=182:AD=113:AE=2
 10787 EXEC VASTAGVONAL
 10788 AA= 91:AB=117:AC=141:AD=117:AE=2
 10793 EXEC VASTAGVONAL

```

10794 AA=110:AB= 10:AC=110:AD= 44:AE=2
10799 EXEC VASTAGVONAL
10800 AA=110:AB=114:AC=110:AD=180:AE=2
10805 EXEC VASTAGVONAL
10806 AA=131:AB= 45:AC=131:AD=113:AE=1
10811 EXEC VASTAGVONAL
10812 AA=141:AB= 19:AC=182:AD= 19:AE=2
10817 EXEC VASTAGVONAL
10818 AA=141:AB=137:AC=182:AD=137:AE=2
10823 EXEC VASTAGVONAL
10824 AA=188:AB= 32:AC=188:AD=128:AE=2
10829 EXEC VASTAGVONAL
10830 AA=113:AB=160:AC=170:AD=160:AE=2
10835 EXEC VASTAGVONAL
10836 CA=135:CB=165:CC=160:CD=165
10840 EXEC SZAGGATOTTVONAL
10841 CA=135:CB=165:CC=135:CD=180
10845 EXEC SZAGGATOTTVONAL
10846 CA=160:CB=165:CC=160:CD=180
10850 EXEC SZAGGATOTTVONAL
10851 :ARC 105, 80, 65,115,1,85,85,1
10852 :ARC 105, 80, 65,115,1,86,86,1
10853 :ARC 177, 32, 25,155,1,13,13,1
10854 :ARC 177, 32, 25,155,1,14,14,1
10855 :ARC 177,128, 25,155,1,13,13,1
10856 :ARC 177,128, 25,145,1,14,14,1
10857 :CIRCLE 260,80,30,30,1
10858 :CIRCLE 260,80,29,29,1
10859 :CIRCLE 260,80,35,35,1
10860 :CIRCLE 260,80,36,36,1
10861 :CIRCLE 260,80,60,60,1
10862 :CIRCLE 260,80,59,59,1
10863 :ARC 260, 80, 0,360,60,59,59,1
10864 :ARC 260, 80, 0,360,60,58,58,1
10865 :CIRCLE 260,80,50,50,1
10866 :CIRCLE 260,80,49,49,1
10867 END PROC
10868 REM *****
10870 REM RÉTEG2
10871 REM *****
10872 PROC KOMBI2
10874 :REC 91,10,22,32,1
10875 FA= 10:FB= 2:FC=100:FD= 20
10879 EXEC SRAFFOZÁS
10880 :REC 91,118,21,64,1
10881 FA= 10:FB= 2:FC=100:FD=150
10882 EXEC SRAFFOZÁS
10883 :REC 112, 10,30,32,1
10884 FA= 10:FB= 1:FC=120:FD= 20

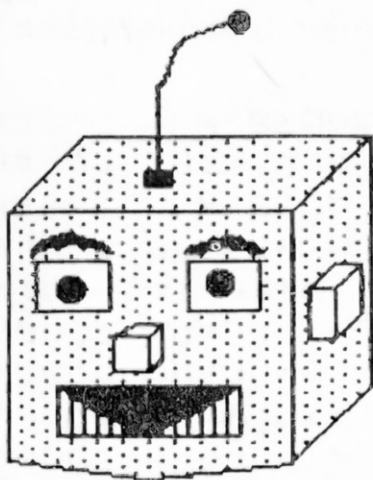
```



```

10887 EXEC SRAFFOZÁS
10888 :LINE 112,118,141,118,1
10889 :LINE 112,118,112,160,1
10890 :LINE 112,162,170,162,1
10891 :LINE 170,162,170,151,1
10892 :LINE 170,151,140,151,1
10893 :LINE 140,151,140,117,1
10894 FR= 10:FB= 1:FC=120:FD=150
10895 EXEC SRAFFOZÁS
10896 END PROC
10898 REM *****
10900 REM RÉTEG3
10901 REM *****
10902 PROC KOMBI3
10903 ER= 21:EB= 45:EC=4:ED=1:EE=64:EM=17:EN=1
10910 EXEC KÓTA
10911 :LINE 16,110, 16,195,1
10912 :LINE 190,195,190,195,1
10913 ER=190:EB=190:EC=6:ED=1:EE=169
10918 EXEC KETTŐS MÉRETNYÍL
10919 :LINE 311,110,311,165,1
10920 :LINE 210,110,210,165,1
10921 ER=210:EB=165:EC=2:ED=1:EE= 94
10926 EXEC KETTŐS MÉRETNYÍL
10927 END PROC
11000 REM *****
11001 REM *****
11002 REM ÁBRÁ7
11003 HIRES 0,7
11004 EXEC KOMBI1
11005 EXEC KOMBI3
11006 GOTO 11006

```



Miért kezdjük a programot előlről, és rögtön utána miért ugratunk a végére? Azért, mert ilyen programszerkezet esetén mindig a RUN paranccsal indíthatjuk a programunkat, és azt tetszőlegesen meghosszabbíthatjuk. Csak azzal kell törődnünk, hogy a 10001. sorban levő ugrási cím mindig helyes legyen.

Még valami fontosat el kell mondanunk: Bizonyára már régóta felmerült a kérdés, hogy miért takarékoskodunk annyira a REM utasításokkal? A tárhely olyan drága, hogy jobb, ha ezektől a REM-ektől eltekintünk. Ha valaki tudni akarja, hogyan struktúráltuk a programot, az megjegyzések nélkül is megtudhatja. Aki megjegyzések nélkül nem tudja ezt megállapítani, annak a megjegyzések sem sokat használnak. A jó tanácsok szerint, a jól struktúrált programok bőséges megjegyzésekkel kezdődnek. Ezt a tanácsot általában csak azok a programozók fogadják meg, akik másképpen nem tudják megtölteni programsorokkal a számítógép belső tárhelyét.

2.2.2.4. Feliratozás

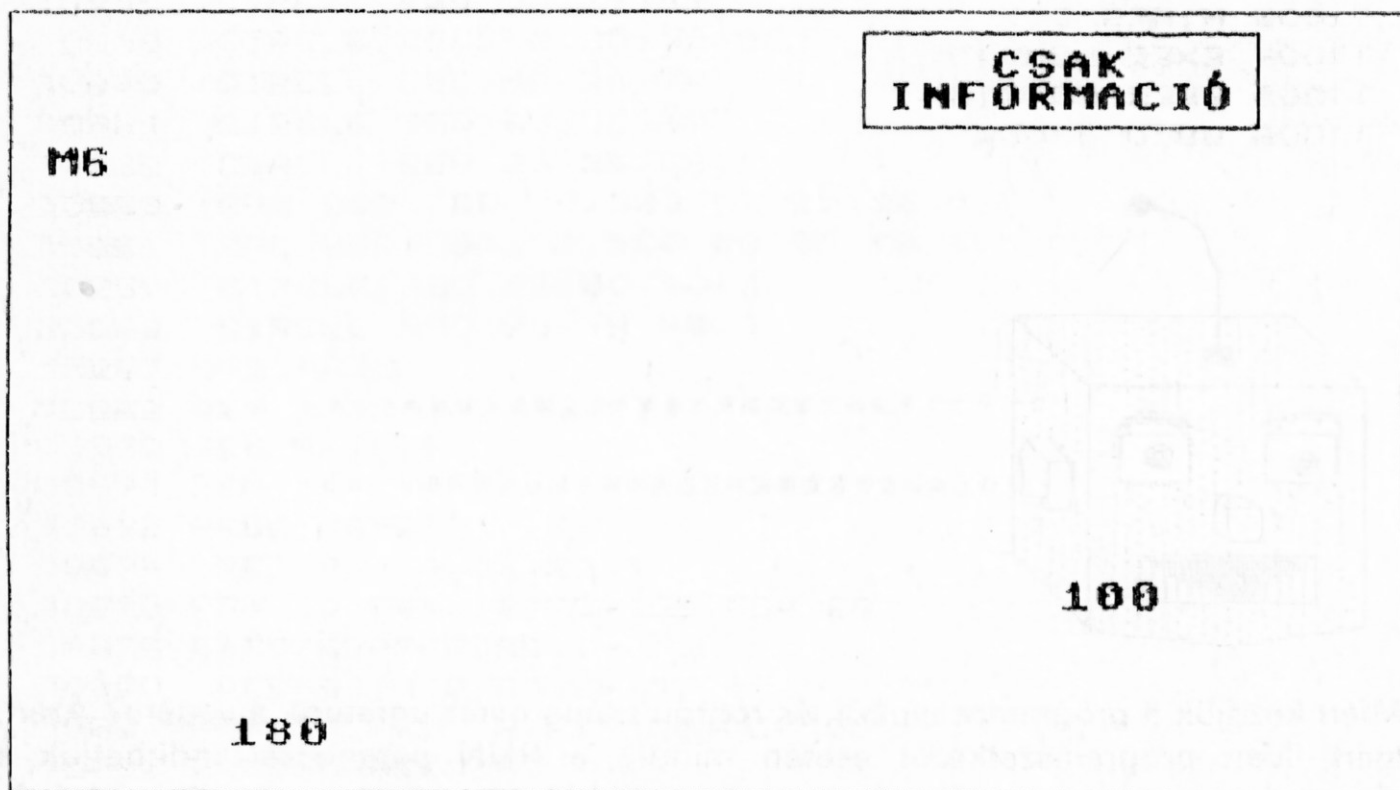
A rajzpéldánkban a feliratokat a 4. rétegre helyezük. Erre a KOMBI4 eljárás való. Most csak az új programrészeket listáztuk ki. Az elvet bizonyára nem kell részletezni. Biztosan észrevették, hogy egy új alapprogramot is írtunk.

Ezt SZÖVEGMEZŐ-nek nevezzük. Ennek segítségével írhatunk szöveget a rajzba. A szövegek írásánál csak arra kell ügyelnünk, hogy egyszerre ne írjunk többet 255 karakternél.

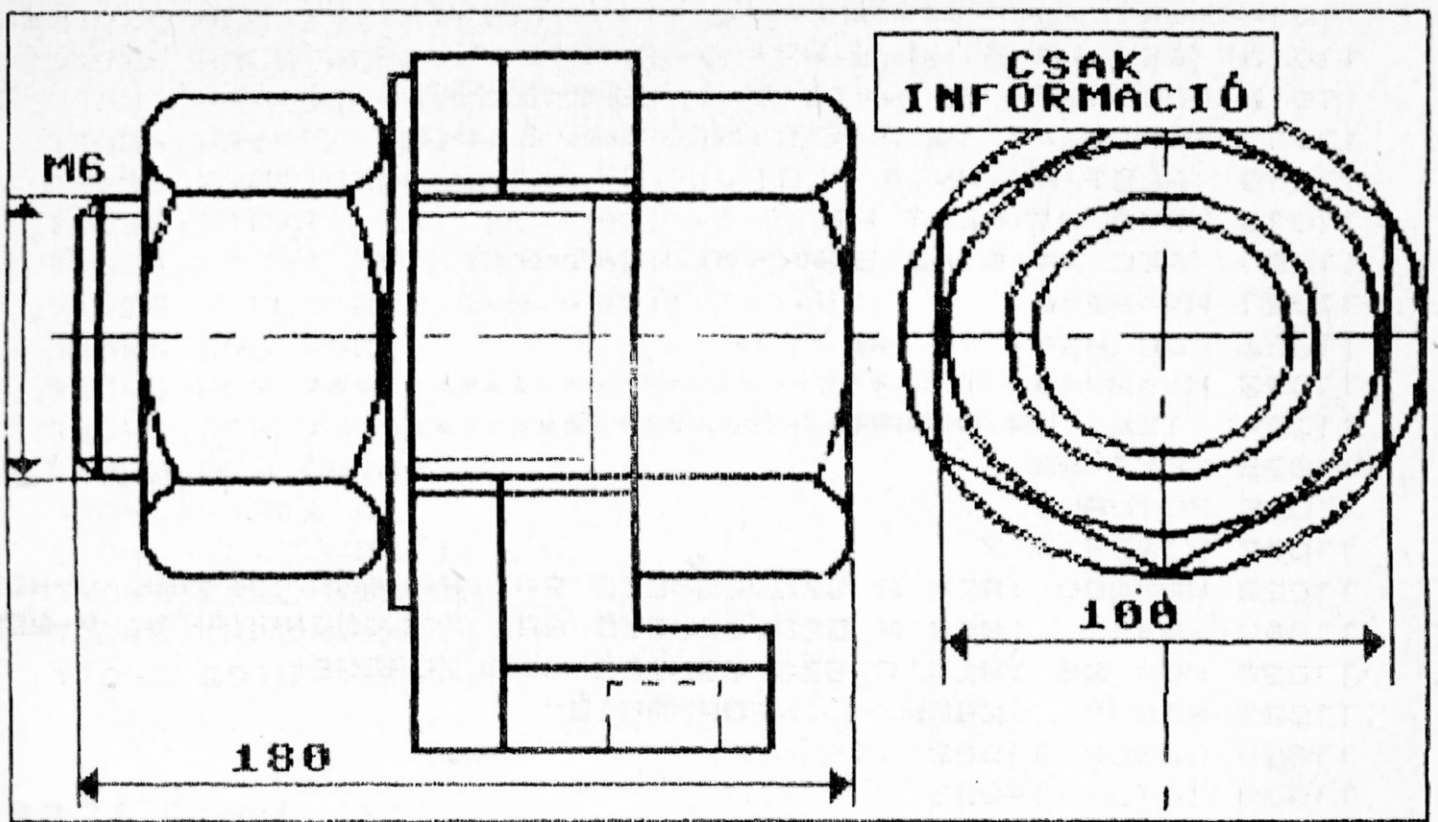
A HA és HB változókkal adjuk meg a szövegmező bal felső pontjának a koordinátáit.

A HC változóval adjuk meg a szövegmező szélességét.

A program a szükséges helyet törli a képernyőn, és feldarabolja a szöveget megfelelő számú sorra.



8. ábra. A 4. réteg



9. ábra. Az 1., a 3. és a 4. réteg

Ez a program megkönnyíti összefüggő szövegeknek egy adott helyre való kiírását. Ez természetesen nem elégít ki minden igényt, pl. nem választja el a szavakat a nyelvtani szabályoknak megfelelően. Ennek megoldása igen munkaigényes lenne, és szükségtelenül nagy belső tárterületet foglalna el egy ilyen alprogram beépítése.

A TEXT parancsot is használhatjuk egyes soroknak a képernyőre való kiírására.

Még egy megjegyzést a méretvonalokról. A régi rajzszabványok szerint a méretvonalat meg kell szakítani, és oda kell írni a méreteket. Ez természetesen számítógéppel nagyon nehezen megvalósítható. Ezért a szabványokat is módosították, és megengedik a számoknak a méretvonalak alá vagy fölé írását.

```

11000 REM SZÖVEGMEZŐ
11001 GOTO 11027
11002 REM SZÖVEGMEZŐ ALPROGRAM
11003 HH=LEN(HE$)
11004 HI=HH*B :HQ=10
11005 HJ=DIV(HI,B)
11006 HK=HJ*B
11007 HL=DIV(HI,HK)
11008 HM=MOD(HI,HK)
11009 HN=HL:IF HM THEN HN=HN+1
11010 HP=-HJ+1
11011 FOR HQ=1 TO HN
11012 HP=HP+HJ
11013 HA$(HQ)=MID$(HE$,HP,HJ)

```

```

11014 NEXT HO
11015 :REC HA-S,HB-S,HK+10,HN*B+B,1
11016 FOR HS=1 TO HN*B+B:HY=(HB-S)+HS
11017 FOR HT=1 TO HK+10 :HX=(HA-S)+HT
11018 :PLOT HX,HY,1:PLOT HX,HY,2
11019 NEXT HT:NEXT HS
11020 :REC HA-S,HB-S,HK+10,HN*B+B,1
11021 HY=HB-B
11022 FOR HR=1 TO HN
11023 HY=HY+B
11024 :TEXT HA,HY,HA$(HR),1,1,B
11025 NEXT HR
11026 RETURN
11027 HIRES 0,7
11028 HA=200 :REM A SZÖVEGMEZŐ BAL FELSŐSARKÁNAK X-KOORD.
11029 HB=10 :REM A SZÖVEGMEZŐ BAL FELSŐSARKÁNAK Y-KOORD.
11030 HC= 85 :REM A SZÖVEGMEZŐ SZÉLESSÉGE
11033 HE$=" CSAK , INFORMÁCIÓ"
11034 GOSUB 11002
11035 GOTO 11035

```

```

10997 REM *****
11001 REM *****
11002 PROC SZÖVEGMEZŐ
11003 HH=LEN(HE$)
11004 HI=HH*B :HQ=10
11005 HJ=DIV(HC,B)
11006 HK=HJ*B
11007 HL=DIV(HI,HK)
11008 HM=MOD(HI,HK)
11009 HN=HL:IF HM THEN HN=HN+1
11010 HP=-HJ+1
11011 FOR HO=1 TO HN
11012 HP=HP+HJ
11013 HA$(HO)=MID$(HE$,HP,HJ)
11014 NEXT HO
11015 :REC HA-S,HB-S,HK+10,HN*B+B,1
11016 FOR HS=1 TO HN*B+B:HY=(HB-S)+HS
11017 FOR HT=1 TO HK+10 :HX=(HA-S)+HT
11018 :PLOT HX,HY,1:PLOT HX,HY,2
11019 NEXT HT:NEXT HS
11020 :REC HA-S,HB-S,HK+10,HN*B+B,1
11021 HY=HB-B
11022 FOR HR=1 TO HN
11023 HY=HY+B
11024 :TEXT HA,HY,HA$(HR),1,1,B
11025 NEXT HR
11026 END PROC
11027 REM *****
11028 REM RÉTEG4

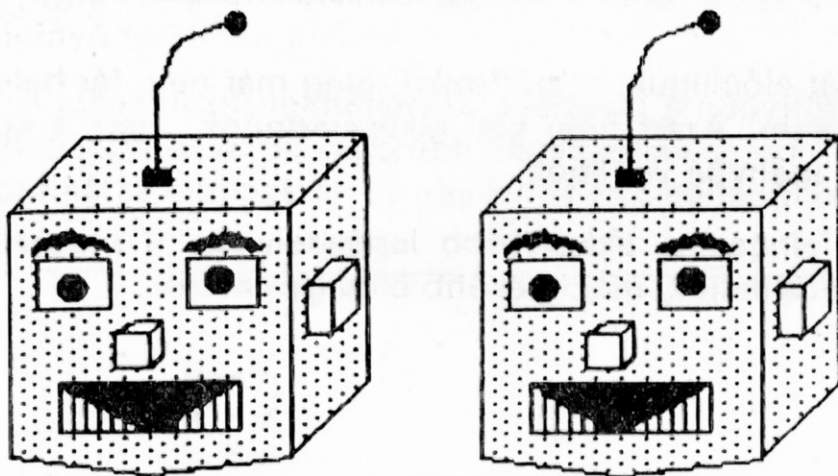
```

```

11029 REM *****
11030 PROC KOMBI4
11031 HA=200:HB=10:HC=85
11034 HE$="    CSAK    ,INFORMÁCIÓ"
11035 EXEC SZÖVEGMEZŐ
11036 :TEXT  8, 35, "MB",1,1,8
11037 :TEXT  50,180,"180",1,1,8
11038 :TEXT  240,145,"100",1,1,8
11039 END PROC
11040 REM *****
11041 REM *****
11042 REM ÁBRAS
11043 HIRIS 0,7
11044 EXEC KOMBI1
11045 EXEC KOMBI3
11046 EXEC KOMBI4
11047 GOTO 11047

```

2.2.2.5. Összefoglalás



Az alapelemek, mint a vonal, tengelyvonal, sraffozások, körök vagy mint alprogramok, vagy mint közvetlen SIMON'S BASIC utasítások állnak rendelkezésünkre.

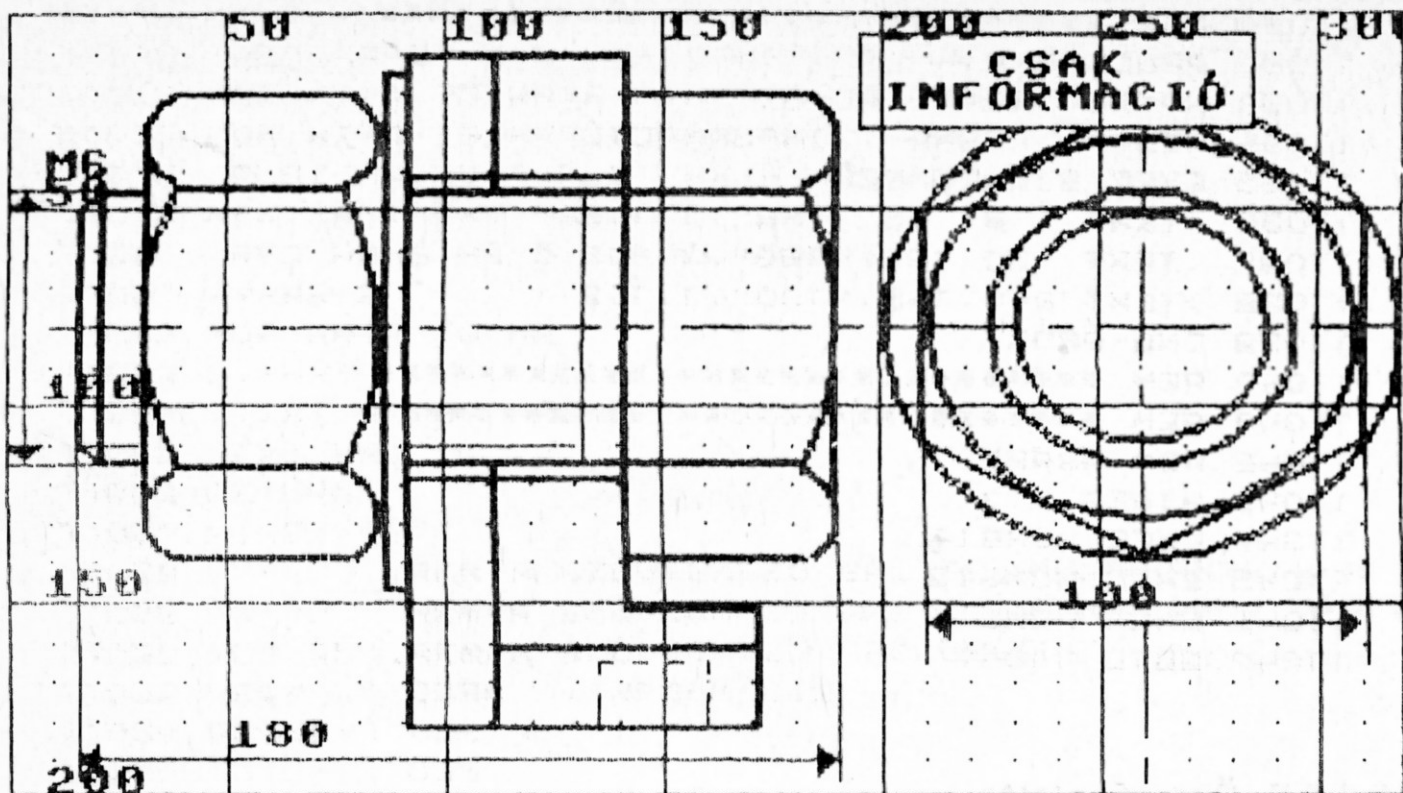
A különböző KOMBI programokkal, amelyek szintén alprogramok, építjük fel az egyes rajzolási rétegeket.

Az ÁBRA programokkal keverjük össze a rétegeket egységes képpé.

Ezt a hierarchiát ha kell még több lépcsővel is ki lehet egészíteni. Így például a menü programot, mint legfelső ÁBRA programot valósíthatjuk meg.

És még valami fontos:

Ha jól megnézzük a 10. ábrát, hiányolni fogjuk a sraffozást. Ennek mélyebb oka van. Minden réteg belső tárterületet igényel. Az a réteg (esetünkben a 2. réteg), amelyikben a sraffozást állítjuk elő, különösen sok helyet igényel. A tár betelik, amikor



10. ábra. Az összes réteg és a segédraszter (sraffozások nélkül)

a 10. ábra itt megjelenő formáját előállítjuk. A sraffozási réteg már nem fér bele. A belső tárba nem fér több program. Azért nem kell elkeserednünk, mert a külső tárunkat, a mágneslemezeket is igénybe vehetjük.

Ha a sraffozást is szeretnénk felrajzolni, akkor több lépésben kell a programot betölteni. Azt, hogy ezt hogyan csináljuk, majd később elmagyarázzuk.

2.2.3. LÉPTÉKEK

A lépték bennünket egészen más értelemben érdekel, mint a nagy CAD-rendszerek felhasználóit. A rajzgépeiken nagyméretű rajzokat tudnak előállítani, amelyeknél érdemes a léptéket mindig az adott rajzméretnek megfelelően beállítani. Mi mindig ugyanabban a méretben kapjuk meg a rajzokat akár képernyőre, akár nyomtatóra dolgozunk. Így a különböző léptékeknek nincs sok értelmük. Eleve kihasználjuk a teljes képernyőméretet a rajzainkhoz. Azonban szükség lehet arra, hogy az egyes képrészleteket más léptékben kiemeljük.

A következő fejezetben láthatjuk, milyen érdekes hatásokat érhetünk el ezzel.

2.2.3.1. Nagyítás, kicsinyítés (és torzítás)

A programtárunk újabb elemmel gyarapodik: azokat az ábrákat, amelyeket koordinátái segítségével adjuk meg, nagyíthatjuk, kicsinyíthetjük, torzíthatjuk, eltolhatjuk, vagy ezek közül egyszerre többet is elvégezhetünk.

LÉPTÉKEZÉS-nek nevezzük azt a programot, amely mindezt elvégzi. A 11059—11086. sorban határozzuk meg a kívánt ábrát. Ezeket a változókat tetszés szerint megadhatjuk, vagy egy másik programból is átvehetjük. A lényeg az, hogy minden egyes változó egy szorzótényezőt kap, amely a léptéket meghatározza. Az X és az Y tengelyhez különböző tényezők tartoznak, M és N. Így az eredeti ábrát torzíthatjuk is, ha N és M nem azonos. Ezen kívül minden egyes változó egy eltolási tagot is kap (V és W az X és az Y tengelyhez), így a nagyított vagy a kicsinyített képeket tetszőlegesen helyezhetjük el a képernyőn.

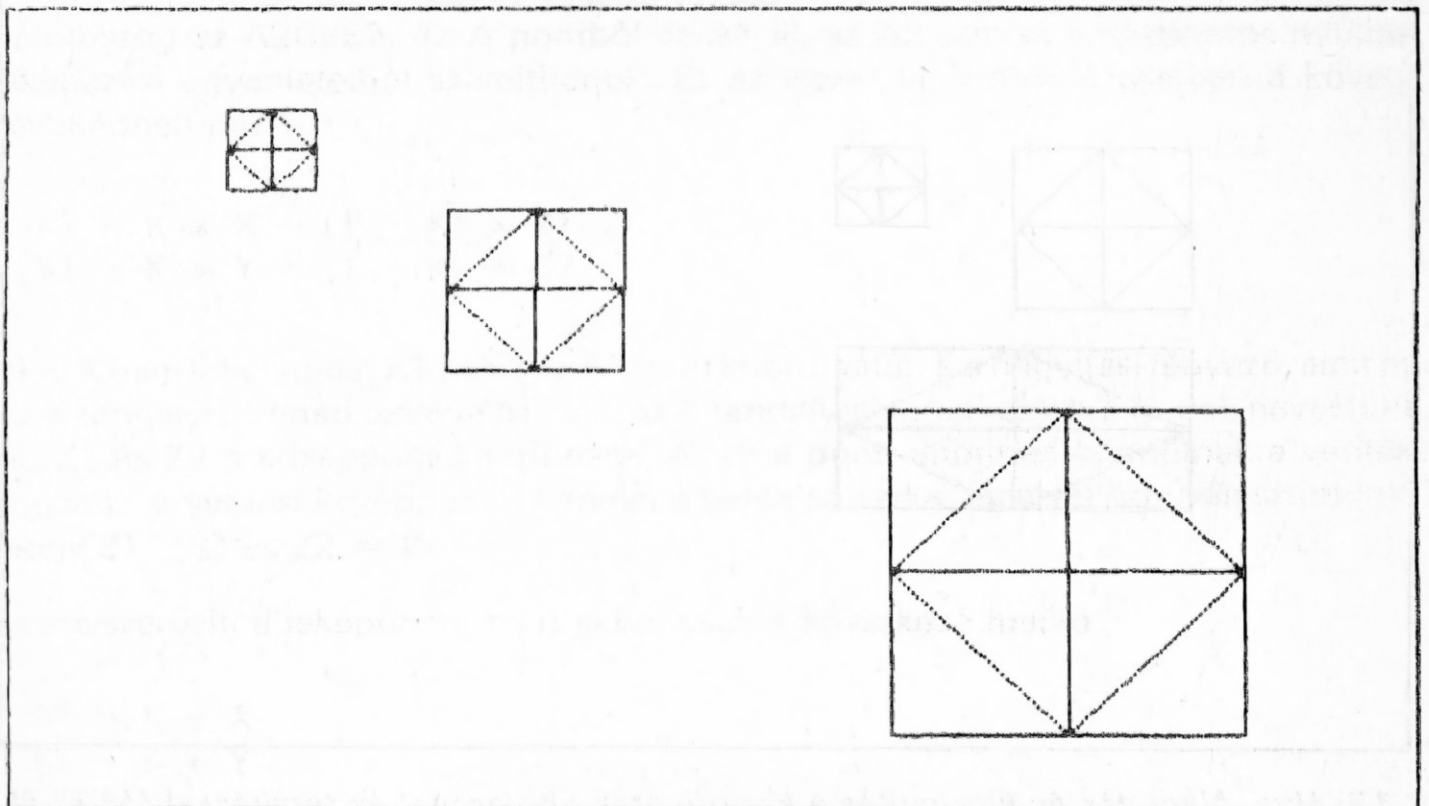
A 11. ábra az eredményeket mutatja be torzítás és eltolás nélkül. Az eredeti ábra a középső, hogy melyik a nagyított és melyik a kicsinyített, az magától értetődő. Nagyítás akkor keletkezik, ha az M és N tényező értéke 1-nél nagyobb; kicsinyítés akkor keletkezik, ha az M és N tényező értéke 1-nél kisebb.

A 12. ábrán ugyanazokat a nagyítási tényezőket használtuk, mint az előző képen, csak olyan eltolásokat alkalmaztunk, hogy az ábrák bal felső sarokpontjai egybeessenek.

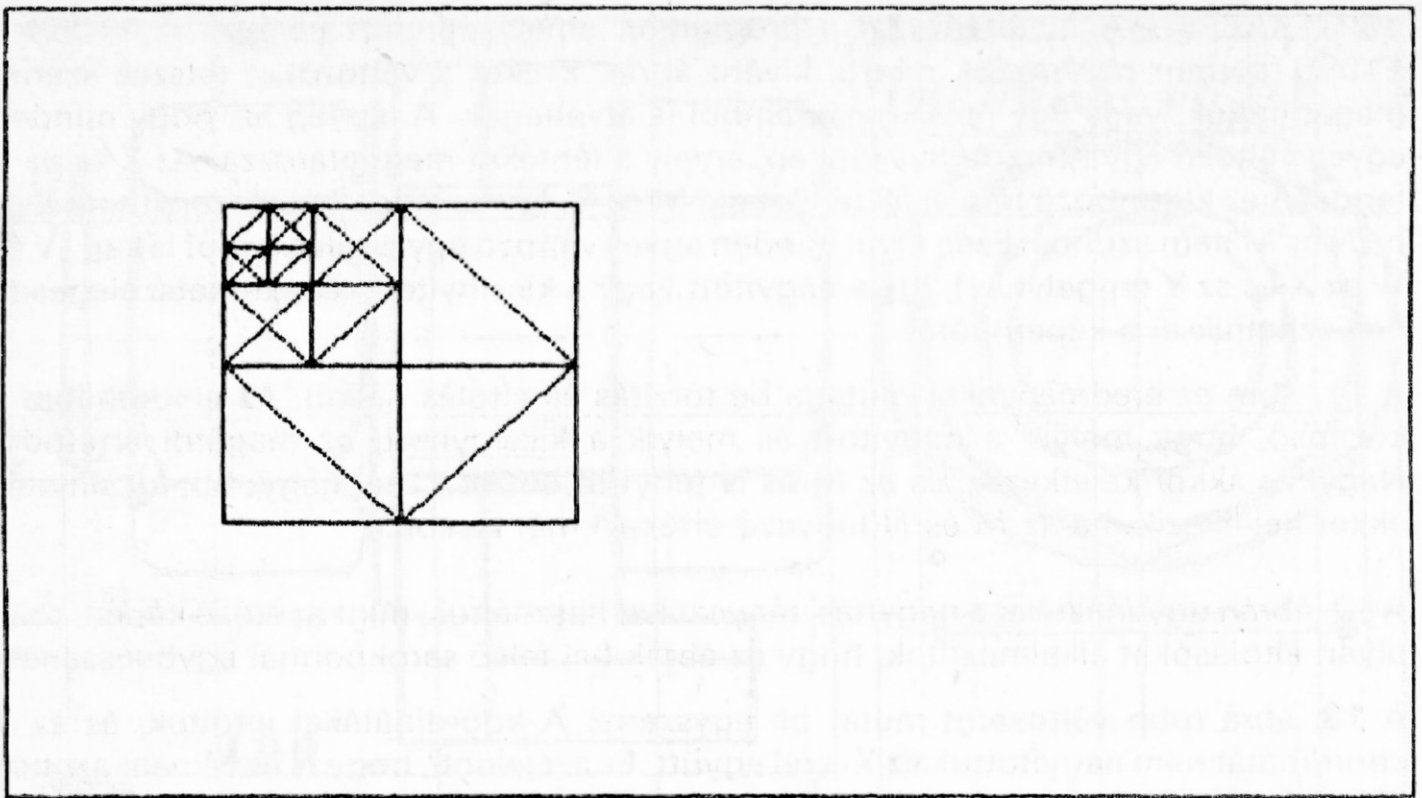
A 13. ábra több változatot mutat be egyszerre. A koordinátákat eltoltuk, és az Y koordinátát nem nagyítottuk az X-szel együtt. Ez azt jelenti, hogy N és M nem azonos ($M = 2$; $N = 1$).

A program kipróbálásánál ügyeljünk arra, hogy a nagy léptékeknél ne jussunk a képernyő területén kívülre.

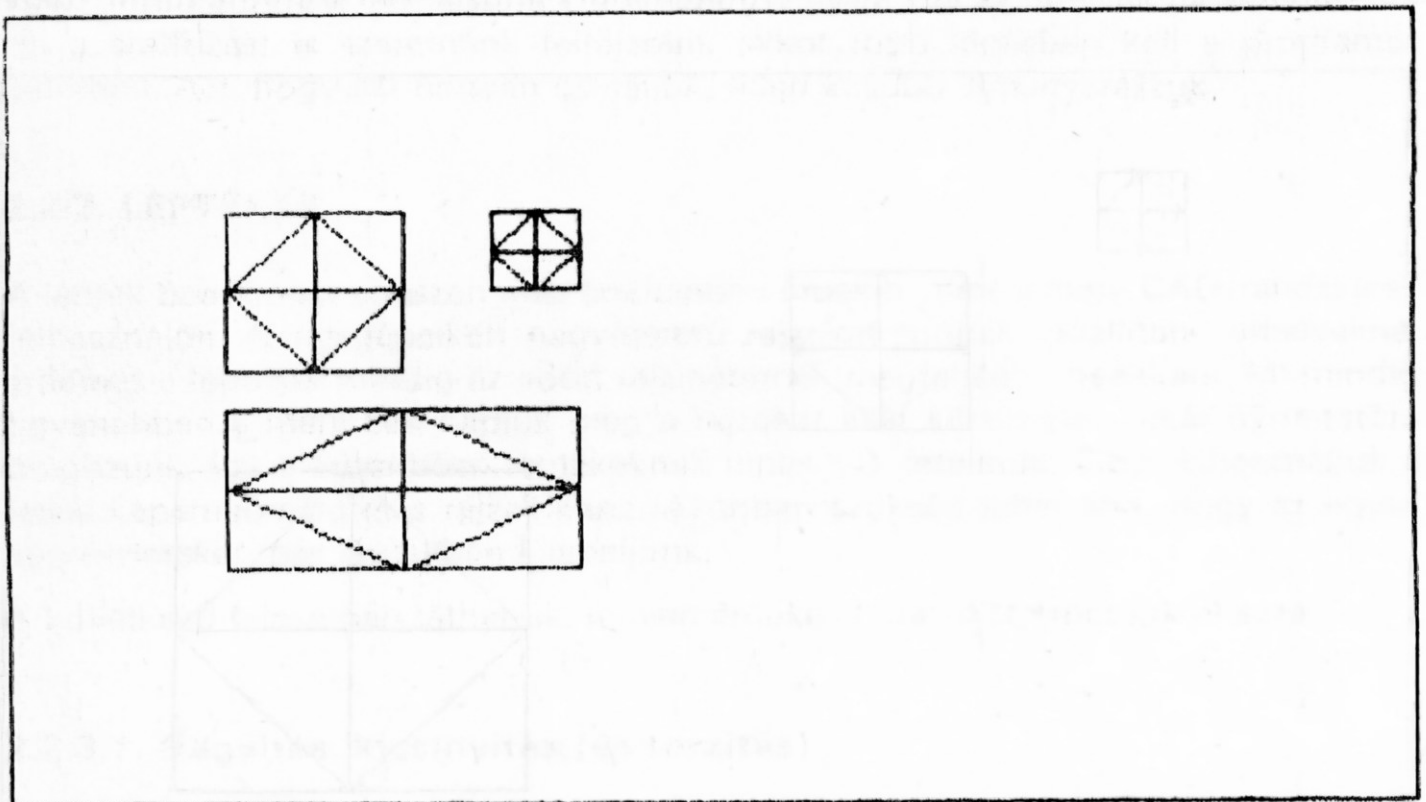
Az itt alkalmazott matematikai eljárással a nagyított ábrát annál jobban eltoljuk az eredeti ábrától, minél nagyobb nagyítást alkalmazunk, hacsak nem gondoskodunk a koordináták eltolásáról. Ez egy kicsit érthetetlenül hangzik. Aki szeretné tudni, hogy



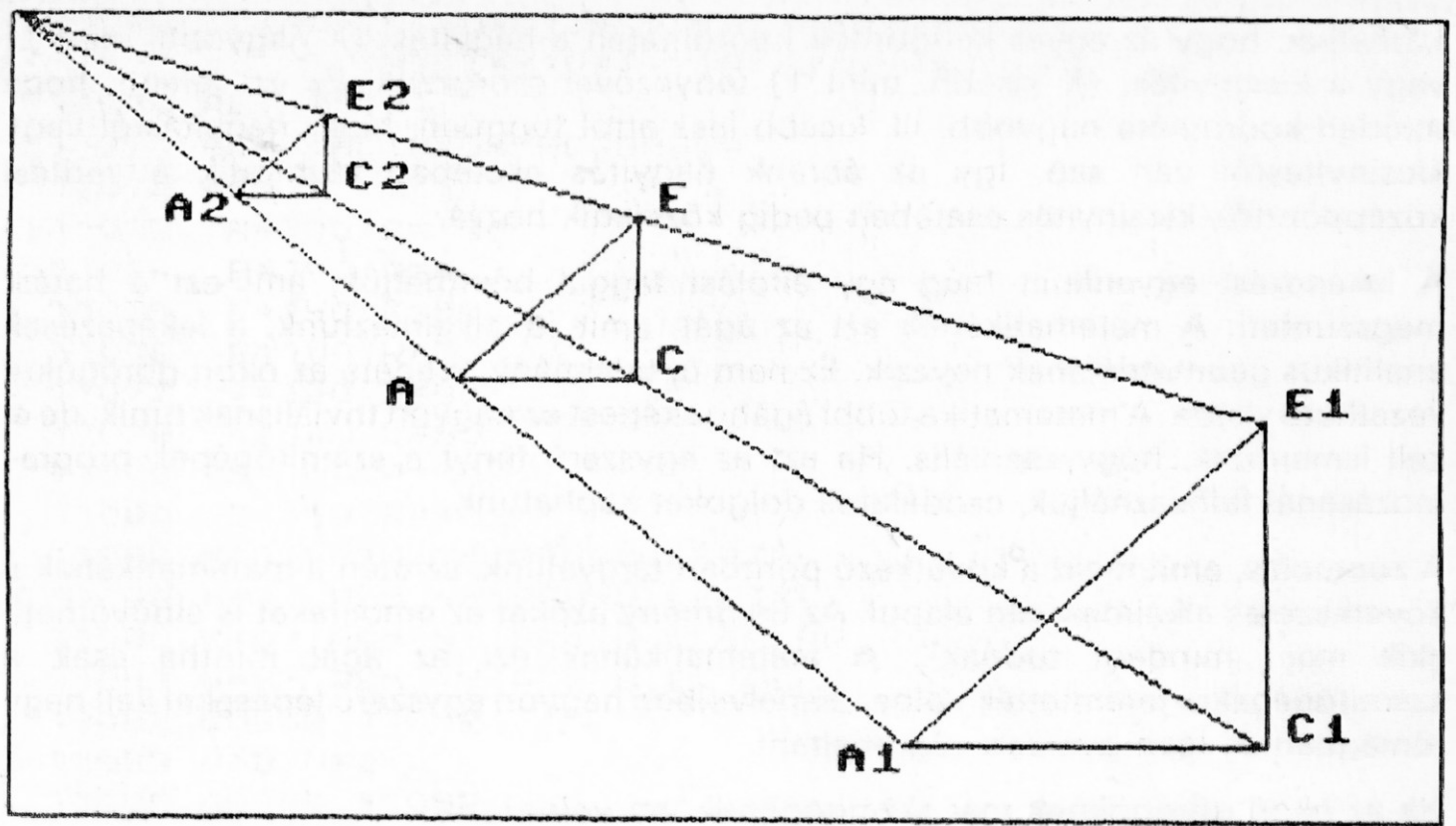
11. ábra. Nagyítás és kicsinyítés



12. ábra. Nagyítás és kicsinyítés a koordináták eltolásával



13. ábra. Nagyítás és kicsinyítés a koordináták eltolásával és torzitással (M és N nem azonos)



14. ábra. A nagyítás és kicsinyítés matematikai alapelve

mi áll e mögött, az olvassa tovább ezt a fejezetet. Aki nem akar törődni a matematikai háttérrel, az egyszerűen lapozzon a következő fejezethez.

A matematikai elvet a 14. ábrán mutatjuk be. Az ACE háromszög a nagyítandó vagy a kicsinyítendő alakzat. A nagyított háromszöget A1 C1 E1-gyel jelöltük. A kicsinyített háromszög az A2C2E2. Az A pontból az A1 ill. az A2 pontot a központos nyújtás leképezési egyenleteiből számíthatjuk. Ez az egyenlet általános esetben a következőképpen néz ki:

$$X1 = K * X + (1 - K) * Z1$$

$$Y1 = K * Y + (1 - K) * Z2$$

ahol K nem lehet nulla; X1 és Y1 az A1 pont koordinátái; K a nagyítási tényező, amit mi az X tengelyre vonatkoztatva M-nek, az Y tengelyre vonatkoztatva N-nek neveztünk el; Z1 és Z2 a középpont koordinátái, ez az a pont, ahonnan kiindulnak a vetítési sugarak. A vetítési középpont a képmező bal felső sarka, amelyet úgy választottunk, hogy Z1 = 0 és Z2 = 0.

Ez egyszerűsíti a leképezést, mert ekkor csak a következő marad:

$$X1 = K * X$$

$$Y1 = K * Y$$

ahol K nem nulla.

Láthatjuk, hogy az egyes képpontok koordinátáit a nagyítási (K nagyobb, mint 1) vagy a kicsinyítési (K kisebb, mint 1) tényezővel szorozzuk. Ez azt jelenti, hogy minden koordináta nagyobb, ill. kisebb lesz attól függően, hogy nagyításról vagy kicsinyítésről van szó. Így az ábránk nagyítás esetében távolodik a vetítési középponttól, kicsinyítés esetében pedig közeledik hozzá.

A leképezési egyenletet még egy eltolási taggal bővíthetjük, ami ezt a hatást megszünteti. A matematikának azt az ágát, amit itt alkalmaztunk, a leképezések analitikus geometriájának nevezik. Ez nem új találmány, eredete az ókori görögökre vezethető vissza. A matematika többi ágához képest ez nagyon triviálisnak tűnik, de el kell ismernünk, hogy zseniális. Ha ezt az egyszerű ténytet a számítógépek programozásánál felhasználjuk, csodálatos dolgokat kaphatunk.

A zoomolás, amit majd a következő pontban tárgyalunk, szintén a matematikának a következetes alkalmazásán alapul. Az eredmény azokat az embereket is elbűvölheti, akik már „mindent tudnak”. A matematikának ezt az ágát mintha csak a számítógépek teremtették volna: alapelveiben nagyon egyszerű lépéseket kell nagy tömegben és igen gyorsan végrehajtani.

Ha az ókori görögöknek már számítógépük lett volna!

```
11050 REM LÉPTÉKEZÉS
11051 HIRES 0,7:GOTO 11071
11052 REM LÉPTÉKEZÉS ALPROGRAM
11053 T=1:U=1
11054 IF M>1 THEN 11057
11055 IF N>1 THEN 11059
11056 GOTO 11060
11057 T=-1
11058 GOTO 11055
11059 U=-1
11060 :LINE INT(A*M+U*T), INT(B*N+W*U),
      INT(C*M+U*T), INT(D*N+W*U), 1
11061 :LINE INT(C*M+U*T), INT(D*N+W*U),
      INT(E*M+U*T), INT(F*N+W*U), 1
11062 :LINE INT(E*M+U*T), INT(F*N+W*U),
      INT(G*M+U*T), INT(H*N+W*U), 1
11063 :LINE INT(G*M+U*T), INT(H*N+W*U),
      INT(A*M+U*T), INT(B*N+W*U), 1
11064 :LINE INT(I*M+U*T), INT(J*N+W*U),
      INT(K*M+U*T), INT(L*N+W*U), 1
11065 :LINE INT(D*M+U*T), INT(P*N+W*U),
      INT(Q*M+U*T), INT(R*N+W*U), 1
11066 :LINE INT(I*M+U*T), INT(J*N+W*U),
      INT(Q*M+U*T), INT(R*N+W*U), 1
11067 :LINE INT(Q*M+U*T), INT(R*N+W*U),
      INT(K*M+U*T), INT(L*N+W*U), 1
11068 :LINE INT(K*M+U*T), INT(L*N+W*U),
      INT(D*M+U*T), INT(P*N+W*U), 1
```

```

11069 :LINE INT(O*M+U*T), INT(P*N+W*U),
        INT(I*M+U*T), INT(J*N+W*U), 1
11070 RETURN
11071 REM VÁLTOZÓK MEGADÁSA
11072 REM X : Y
11073 :A=100:B=90
11074 :C=140:D=90
11075 :E=140:F=50
11076 :G=100:H=50
11077 :I=120:J=90
11078 :K=120:L=50
11079 :O=100:P=70
11080 :Q=140:R=70
11088 M=1 :REM X-IRÁNYÚ LÉPTÉK
11089 N=1 :REM Y-IRÁNYÚ LÉPTÉK
11090 U=0 :REM X-IRÁNYÚ ELTOLÁS
11091 W=0 :REM Y-IRÁNYÚ ELTOLÁS
11092 GOSUB 11052
11093 M=2:N=2
11095 GOSUB 11052
11096 M=1/2:N=1/2
11098 GOSUB 11052
11099 GOTO 11099

```

2.2.3.2. Zoomolás

Ahhoz, hogy az előző pont LÉPTÉKEZÉS programja működhessen, a számítógépnek ismernie kell a nagyítandó alakzat pontjainak koordinátáit.

Egy rajzon belül előfordulnak olyan ábrák vagy ábrarészek, amelyeknek koordinátáit nem ismerjük, mert vonalak vagy felületek metszése során kaptuk. Néha ilyen részleteket is tudni kell nagyítani. Azt a programot, ami ezt végrehajtja, ZOOM-nak nevezzük.

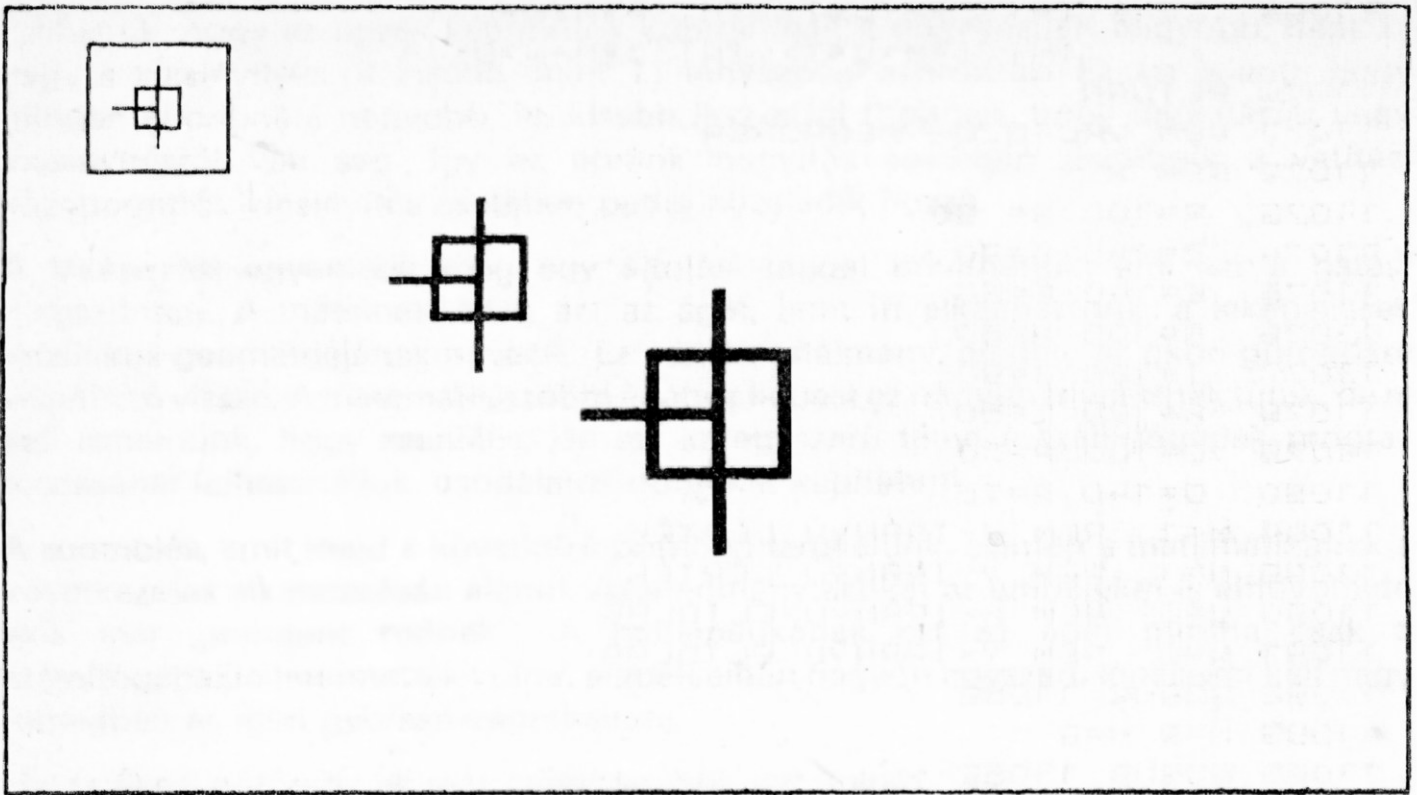
E program segítségével egy előzőleg meghatározott képrész tetszőlegesen (amikorára a képernyő mérete megengedi) nagyíthatjuk. Ez a képrész téglalap alakú.

A képrész pontról pontra „felfűjjük”. Kicsinyítéseket ilyen módon nem végezhetünk, mert a zoomolandó képrész a lehető legkisebb méretű, a pontok egymás mellett vannak. A zoomolásnál nemcsak a távolságokat növeljük meg, hanem magukat a képpontokat is. Ez azt jelenti, hogy a vonalvastagságok és a kitöltött felületek is megnőnek.

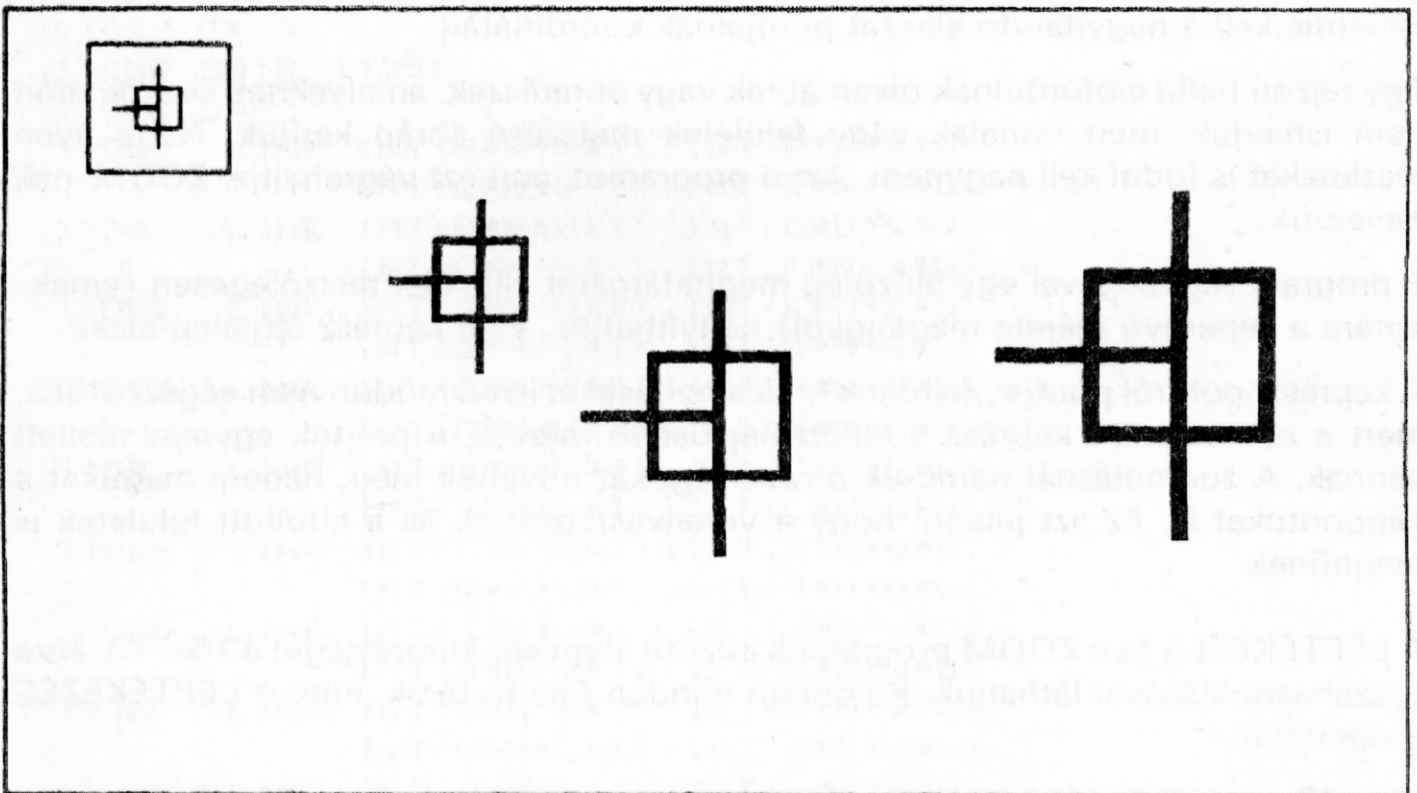
A LÉPTÉKEZÉS és a ZOOM programok közötti alapvető különbséget a 15—18. ábra összehasonlításával láthatjuk. Különbözik minden úgy történik, mint a LÉPTÉKEZÉS programnál.

IA és IB a zoomolandó képrész bal felső sarokpontjának koordinátáit határozza meg.

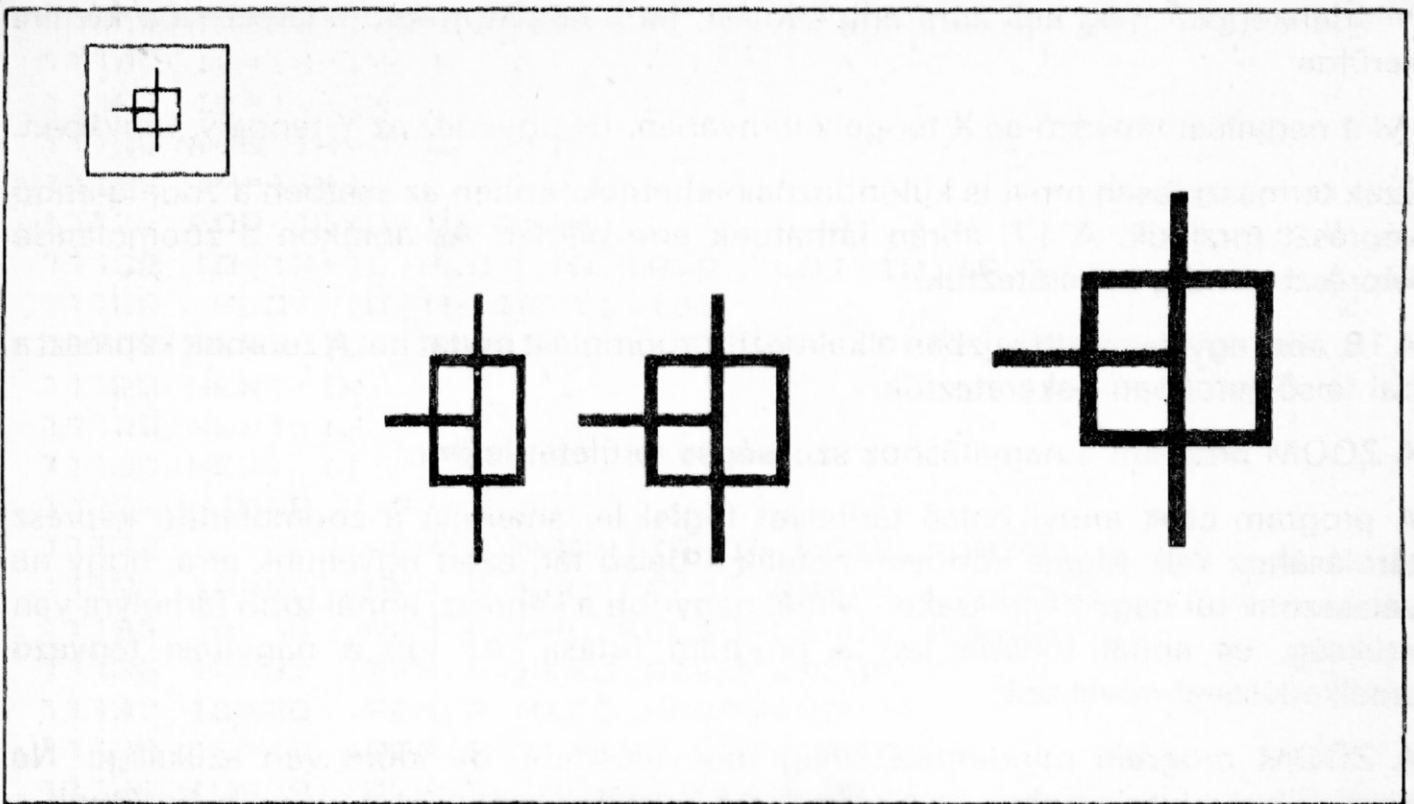
IC a zoomolandó képrész szélességét, ID a magasságát adja meg.



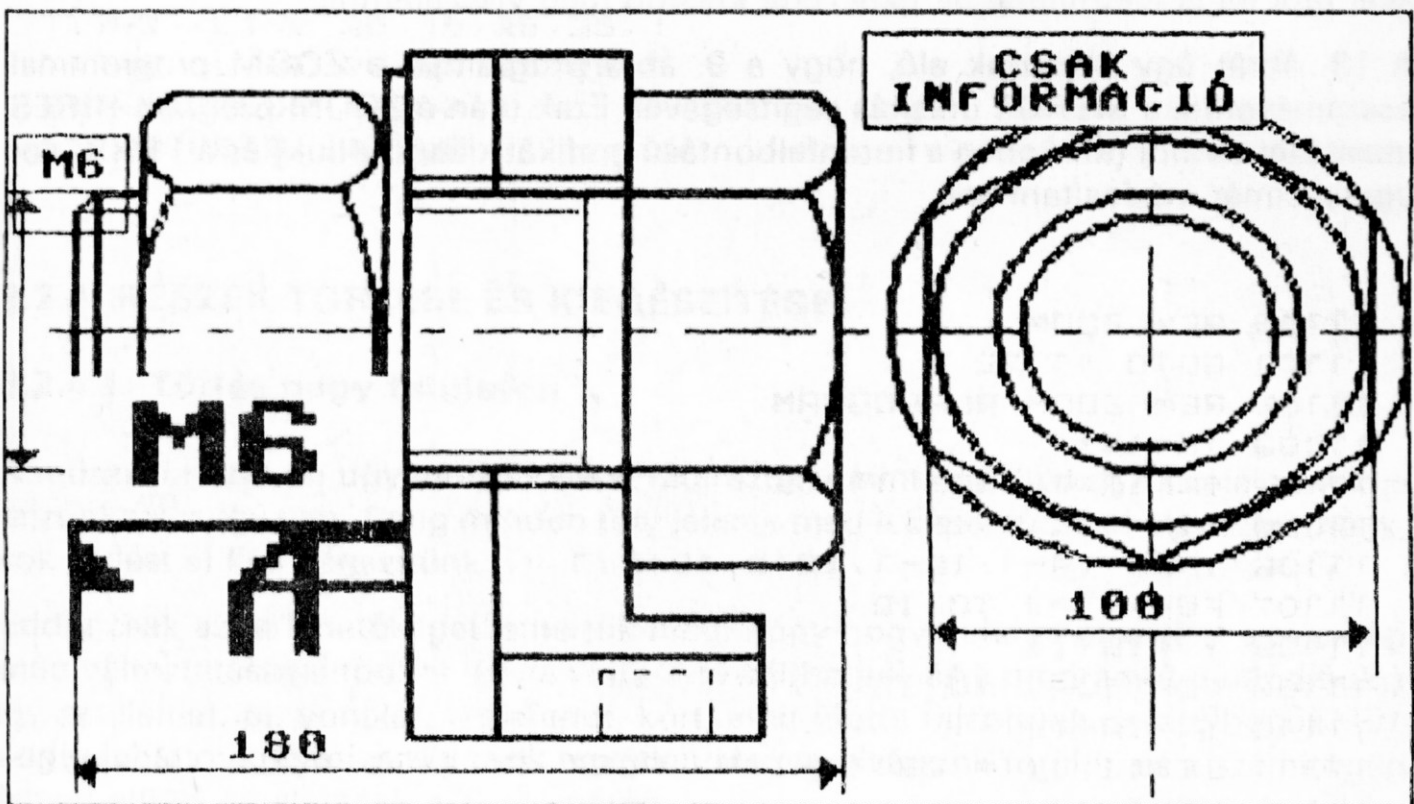
15. ábra. Zoomolás koordinátaeltolás nélkül



16. ábra. Zoomolás koordinátaeltolással



17. ábra. Zoomolás koordinátaeltolással és torzitással



18. ábra. Zoomolás egy összetett ábrában

IE-vel a nagyított képrész X irányú eltolását adhatjuk meg. Ha jobbra akarjuk eltolni, akkor pozitív, ha balra akarjuk eltolni, akkor negatív IE értéket adjunk meg.

IU-val ugyanezt adhatjuk meg, csak Y irányban.

Mindenképpen meg kell adni egy eltolást, ha a nagyított kép a képernyőn kívülre kerülne.

IM a nagyítási tényező az X tengely irányában, IN ugyanez az Y tengely irányában.

Ezek természetesen most is különbözőek lehetnek. Ebben az esetben a zoomolandó képrész torzítjuk. A 17. ábrán láthatunk erre példát. Az ábrákon a zoomolandó képrész mindig bekereteztük.

A 18. ábra egy összetett rajzban alkalmazott zoomolást mutat be. A zoomolt képrész a bal felső sarokban bekereteztük.

A ZOOM program a nagyításhoz szükséges területet letörli.

A program csak annyi belső tárhelyet foglal le, amennyi a zoomolandó képrész tárolásához kell. Mégis könnyen betelik a belső tár, ezért ügyeljünk arra, hogy ne válasszunk túl nagy képréseket. Minél nagyobb a képrész, annál több tárhelyre van szükség, és annál tovább tart a program futása. Az idő a nagyítási tényező emelkedésével növekszik.

A ZOOM program mindenfélét meg tud valósítani, de időre van szüksége. Ne veszítsük el a türelmünket, ha a képernyőn látszólag semmi sem történik. Barátunk, a C 64-es ekkor azon gondolkodik, hogyan folytassa a munkát. A számítógép a megjelölt képrész minden egyes pontját tárolja. Ezek benne maradnak a gépben, és akár újra fel is használhatjuk őket. Erre később még visszatérünk.

A 18. ábrát úgy állítottuk elő, hogy a 9. ábra programját a ZOOM programmal összemásoltuk a MERGE utasítás segítségével. Ezek után a ZOOM program HIRE utasítását törölni (különben a finomfelbontású grafikát kikapcsoljuk) és a 11101. sor ugrási címét módosítani kell.

```
11100 REM ZOOM
11101 GOTO 11132
11102 REM ZOOM ALPROGRAM
11103 IG=IG+1
11104 IF IG>1 THEN 11106
11105 DIM IZ<ID,IC>
11106 :REC IA-1,IB-1,IC+2,ID+2,1
11107 FOR II=1 TO ID
11108 IY=IB+II
11109 FOR IJ=1 TO IC
11110 IX=IA+IJ
11111 IZ<II,IJ>=TEST<IX,IY>:PLOT IX,IY,2:PLOT IX,IY,2
11112 NEXT IJ
11113 NEXT II
11114 IU=IA*IM+IE
11115 IW=IB*IN+IU
11116 FOR II=1 TO ID
11117 IY=IW+<IB+II>*IN
11118 FOR IJ=1 TO IC
```

```

71119 IX=IV+(IR+IJ)*IM
71120 IR=IX-IM
71121 IS=IY-IN
71122 FOR IK=0 TO 2*IN
71123 IP=IS+IK
71124 FOR IL=0 TO 2*IM
71125 IO=IR+IL: PLOT IO,IP,2: PLOT IO,IP,2
71126 : PLOT IO,IP,IZ<II,IJ>
71127 NEXT IL
71128 NEXT IK
71129 NEXT IJ
71130 NEXT II
71132 HIRES 0,7
71133 : : REM A NAGYÍTANDÓ MEZŐ ADATAI :
71134 IA=20 : REM A BAL FELSŐ SAROK X-KOORD.
71135 IB=10 : REM A BAL FELSŐ SAROK Y-KOORD.
71136 IC=30 : REM A MEZŐ SZÉLESSÉGE
71137 ID=30 : REM A MEZŐ MAGASSÁGA
71138 IE= 0 : REM X-IRÁNYÚ ELTOLÁS
71139 IU= 0 : REM Y-IRÁNYÚ ELTOLÁS
71140 IM= 2 : REM X-IRÁNYÚ NAGYÍTÁS
71141 IN= 2 : REM Y-IRÁNYÚ NAGYÍTÁS
71142 : REC 30,20,10,10,1
71143 : LINE 35,15,35,35,1
71144 : LINE 25,25,35,25,1
71145 GOSUB 11102
71146 GOTO 11146

```

2.2.4. RÉSZEK TÖRLÉSE ÉS KIEGÉSZÍTÉSE

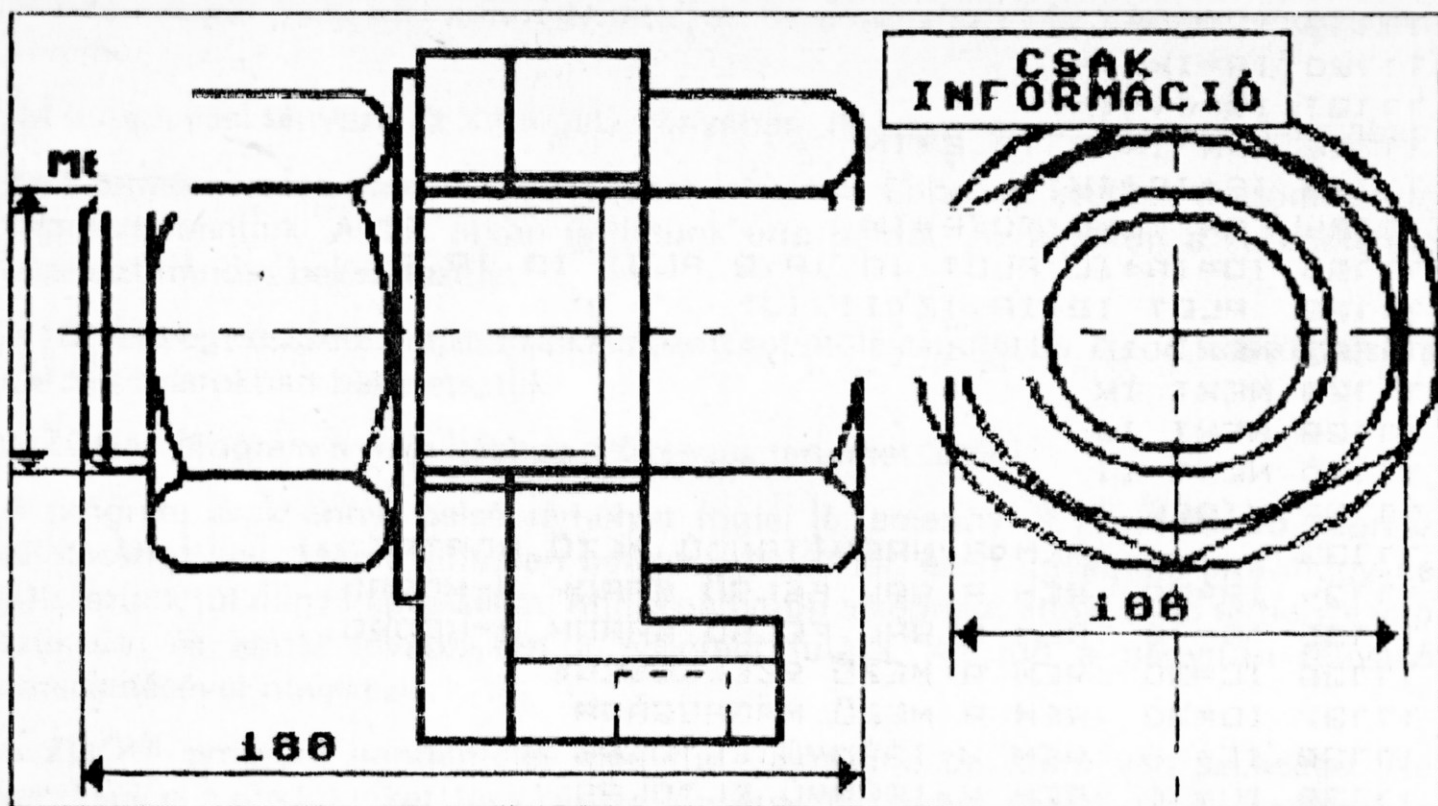
2.2.4.1. Törlés nagy felületen

A műszaki rajzokon ugyanolyan sokat radíroznak, mint rajzolnak. Ez az elektronikus rajzunknál is így van. Amíg minden úgy jelenik meg a képernyőn, ahogy szeretnénk, sok törlést el kell végeznünk.

Eddig csak azt a lehetőséget ismertük meg, hogy hogyan lehet elemeket a rajzolás mód változtatásával törölni. (0-ra vagy 2-re állíthatjuk, és a programot újraindítjuk.) Így az elemet, pl. vonalat, téglalapot, kört, amit előtte rajzoltunk, letörölhetjük. Ezt addig lehet megtenni, amíg csak egyetlen elemet kívánunk törölni, és a számítógép ismeri ennek az elemnek a koordinátáit.

Sokkal gyakrabban fordul elő, hogy egyes területeket szeretnénk letörölni. Egy ilyen mezőben sokféle elem szerepelhet, amelyet a számítógép egyenként már nem is ismer. Ilyen esetekre újabb segédeszközre van szükségünk, amellyel egyszerűen, mint egy radírgumival törölhetünk.

Az elektronikus radírunkat MEZŐTÖRLÉS-nek nevezzük. Tetszőleges helyzetű és



19. ábra. Két képrészt utólag töröltünk

téglalap alakú képrészeket törölhetünk segítségével. A 19. ábra erre mutat be példákat.

KA és KB a törlendő képrész bal felső sarokpontjának koordinátáit adja meg.

KV-vel adjuk meg a mező szélességét, KD-vel a magasságát.

A számítógép megjelöli a törlendő képrészt egy téglalappal, és elkezd törölni. Közben a téglalapot is letörli. Ezután egy üres terület marad, amibe újra rajzolhatunk.

```

11150 REM MEZŐTÖRLÉS
11151 GOTO 11165
11152 REM MEZŐTÖRLÉS ALPROGRAM
11153 :REC KA,KB,KC,KD,1
11154 KX=KA
11155 KY=KB
11156 FOR KE=0 TO KD
11157 KY=KB+KE
11158 FOR KF=0 TO KC
11159 KX=KA+KF
11160 :PLOT KX,KY,1
11161 :PLOT KX,KY,2
11162 NEXT KF
11163 NEXT KE
11164 RETURN
11165 HIRES 0,7
11166 : :REM A TÖRLENDŐ MEZŐ ADATAI:

```



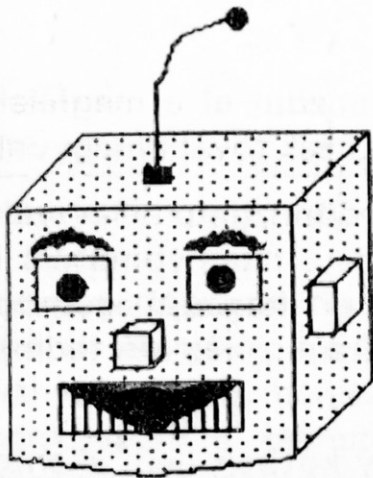
```

11167 KA= 20:REM A BAL FELSŐ SAROK X-KOORDINÁTÁJA
11168 KB= 10:REM A BAL FELSŐ SAROK Y-KOORDINÁTÁJA
11169 KC= 30:REM A MEZŐ SZÉLESSÉGE
11170 KD= 30:REM A MEZŐ MAGASSÁGA
11171 GOSUB 11152
11172 GOTO 11172

```

2.2.4.2. Törlés finom struktúrában

Egy mérnök azonban nemcsak radírral dolgozik, amivel durván körberadírozza a rajzát, hanem egész finom és hegyes ceruzaradírral is. Ezzel igen kis részeket is törölhet, amit ugyanolyan hegyes ceruzával újrarajzolhat. A számítógép a rajz változtatásait különösen könnyen el tudja végezni.



Szükségünk van tehát egy elektronikus ceruzaradírra és egy elektronikus ceruzára, amivel közvetlenül a képernyőn finom változtatásokat is végre tudunk hajtani úgy, hogy a szemünk a képernyőn, kezünk a billentyűzeten van.

Ez az új építőelemünk a CERUZA.

Az elektronikus ceruzával végigpásztázhatjuk a képernyőt, és akárhol meggyújthatunk és elolthatunk egy képpontot. Ez egyben ceruza és radír is.

Ahhoz, hogy az egész menetet ismertessük, legjobban, ha az elején kezdjük.

A program indítása után a finomfelbontású képernyő alsó szélén jelentkezik a program a START X? kérdéssel, hogy melyik X koordinátájú pontba szeretnénk a ceruzát helyezni. Adjuk meg a kívánt koordinátát és nyomjuk meg a RETURN gombot!

3 másodperc múlva a program felteszi a következő kérdést: START Y? Most a kezdőpont Y koordinátáját várja. Ha megadtuk a kívánt értéket, nyomjuk meg újra a RETURN gombot!

3 másodperc múlva újra törli a program a feliratot, és a kívánt kezdőpontban megjelenik egy nyílszerű kurzor. Ezt HIREZ kurzornak fogjuk nevezni. Ez a kurzor előlről nyitott. Egy képponttal a nyílás előtt van az éles zóna. Ha az R gombot (rajzol)

nyomjuk meg, akkor a számítógép kigyújt ezen a helyen egy pontot. Ha a T gombot (töröl) nyomjuk meg, akkor a gép letöröl onnan egy pontot feltéve, hogy volt ott pont. Tehát:

pont kigyújtása	R gomb,
pont törlése	T gomb,

mindig RETURN nélkül.

Ha a HIRES kurzort mozgatni akarjuk, a következő gombokat nyomjuk meg (mindig RETURN nélkül):

kurzor jobbra	J gomb,
kurzor balra	B gomb,
kurzor fel	F gomb,
kurzor le	L gomb.

A HIRES kurzor minden gombnyomásra egy-egy pontot mozdul el a megfelelő irányba. Ezzel vagy kigyújtunk egy pontot, vagy törölünk, vagy csak továbbmegyünk.

Kipróbáltuk, hogyan lehet a HIRES kurzort a közönséges kurzormozgató billentyűkkel mozgatni. Ezt nem találtuk jónak, mert két gomb egyidejű megnyomására is szükség van, így az mindkét kezünket lefoglalja. Az előzőekben megadott gombok használatával két ujjal — az egyik a kurzor mozgatására, a másik a rajzolásra és törlésre — működtethetjük gépünket.

Ha nagyobb távolságokat is át szeretnénk ugrani úgy, hogy közben nem akarunk pontot rajzolni vagy törölni, akkor nyomjuk meg az S gombot. Ekkor a program újra megkérdezi a kezdőpont koordinátáit, és a folyamat előről kezdődik. Közben természetesen a régi helyen a HIRES kurzort töröljük.

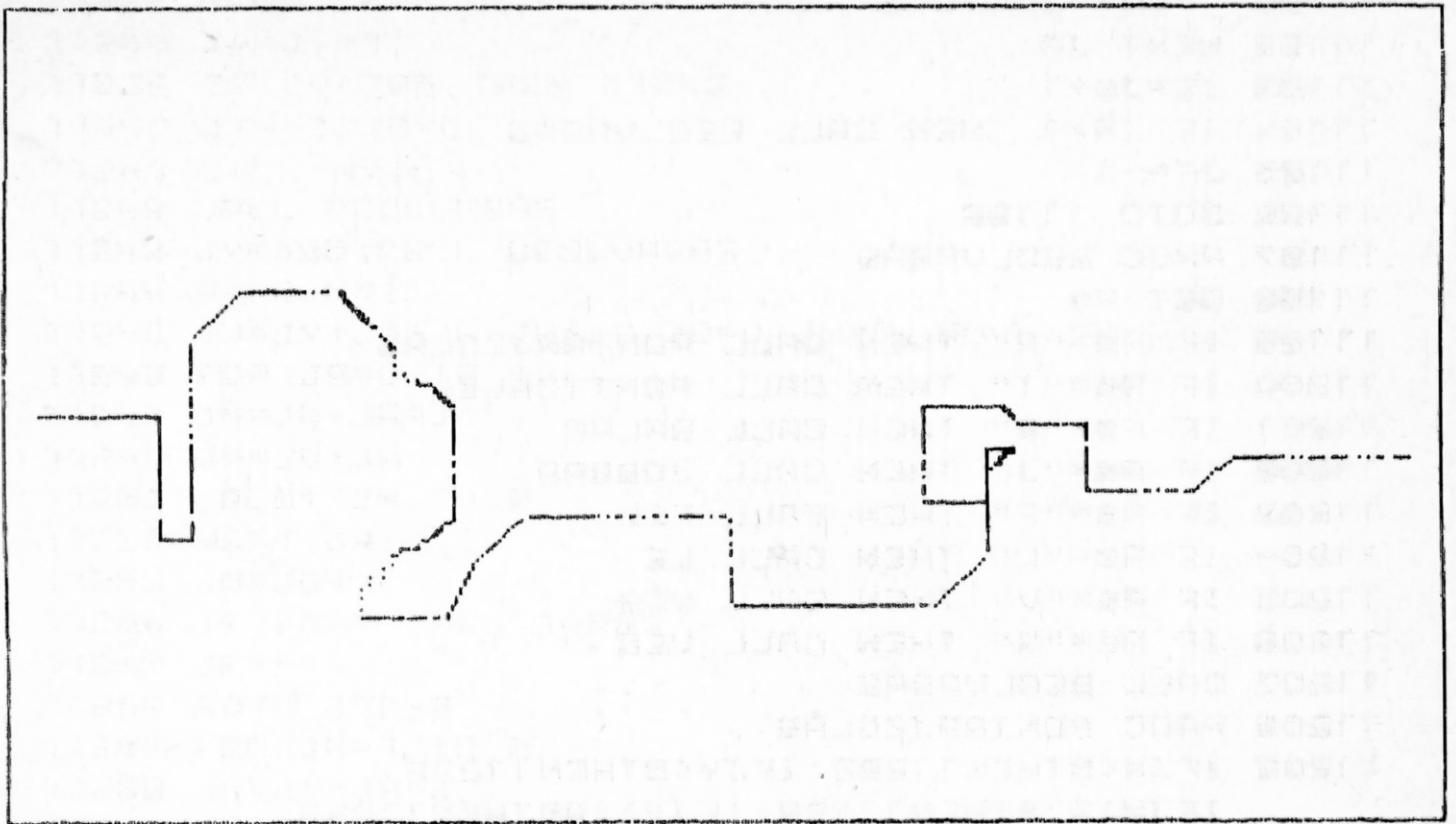
Ha be akarjuk fejezni a program működését, akkor nyomjuk meg a V (mint vége) gombot. Ekkor eltűnik a HIRES kurzor, és kapcsolódhat a következő programhoz.

A kurzort nyugodtan végigvezethetjük a rajzolt felületeken, ezeket a kurzor nem törli. Csak a kurzor előtti pontot tudjuk törölni, ha a T gombot megnyomjuk. Ha rajzolt pontokon halad át a kurzor, akkor ott inverz színben jelenik meg. Ha továbblépünk, akkor pontosan ugyanúgy jelennek meg a pontok, mint annak előtte. Ha töröljük a kurzort, akkor sem törlődnek a kurzor alatti pontok, csak egyszerűen eltűnnek.

A 20. ábra egy tetszőleges vonalat mutat, amelyet a CERUZA programunk segítségével könnyen előállíthatunk.

Még valamit a programról:

Ismét eljárásokból állítottuk össze. Ez most nagyon kényelmes, mert az alprogramok egymásba vannak skatulyázva, és a PROC utasítás szimbolikus ugrási címeivel a program áttekinthető marad.



20. ábra. A CERUZA programmal rajzolt vonal

Az egyes eljárások hívásához nem az EXEC, hanem a CALL utasítást használjuk. Ez fontos! Ha jól megnézzük a programot, láthatjuk, hogy az END PROC utasítás hiányzik. Ennek mélyebb oka van. Ha egy eljárást END PROC-cal zártunk le, akkor megköveteli a számítógép, hogy mindig ennél az utasításnál fejezzük be az alprogramot. Nem érti, ha egy EXEC-kel hívott eljárásból többször kiugrunk anélkül, hogy END PROC utasítást hajtottunk volna végre. Ilyenkor a STACK TOO LARGE hibaüzenettel jelentkezik a program. Erről az üzenetről azonban egyik kézikönyv sem ad tájékoztatást. Abban biztosak lehetünk, hogy a számítógép ismeri ezt a hibaüzenetet, és felismeri a hibát is. Most nem szeretnénk ezt a hibát mélyebben vizsgálni. Elég, ha megállapítjuk, hogy ha különböző eljárások között ugrálunk jobb, ha az END PROC és EXEC utasításoktól eltekintünk, és az alprogramokat a CALL utasításokkal hívjuk.

Ezt a programunkat is úgy építettük fel, mint az összes többi, és tetszőlegesen kombinálható az eddigiekkel.

Ezen kívül arra is találunk példát, hogyan lehet egy programelemből a párbeszédés program egyik elemét kialakítani.

```

11180 REM CERUZA
11181 CALL VÉGE
11182 JL=0
11184 PROC KEZDÉS
11185 JF=1:JS=0
11186 FOR JA=0 TO 2
11187 JW=JY-JA*JF
11189 JU=JX+JA

```

```

11190 :PLOT JV,JW,2
11192 NEXT JA
11193 JS=JS+1
11194 IF JS>1 THEN CALL BEOLVASÁS
11195 JF=-1
11196 GOTO 11186
11197 PROC BEOLVASÁS
11198 GET AS
11199 IF AS="R" THEN CALL PONTRAJZOLÁS
11200 IF AS="T" THEN CALL PONTTÖRLÉS
11201 IF AS="B" THEN CALL BALRA
11202 IF AS="J" THEN CALL JOBBRA
11203 IF AS="F" THEN CALL FEL
11204 IF AS="L" THEN CALL LE
11205 IF AS="V" THEN CALL VÉG
11206 IF AS="S" THEN CALL VÉG
11207 CALL BEOLVASÁS
11208 PROC PONTRAJZOLÁS
11209 IF JX<5 THEN 11222 : IF JY<5 THEN 11236 :
      IF JX>315 THEN 11229 : IF JY>195 THEN 11243
11210 :PLOT JX-2,JY,1:CALL BEOLVASÁS
11211 REM PONTRAJZOLÁS VÉGE
11212 PROC PONTTÖRLÉS
11213 IF JX<5 THEN 11222 : IF JY<5 THEN 11236 :
      IF JX>315 THEN 11229 : IF JY>195 THEN 11243
11214 :PLOT JX-2,JY,0
11215 CALL BEOLVASÁS
11216 PROC BALRA
11217 JX=JX-1
11218 IF JX<5 THEN 11222
11219 JT=0:JS=1
11220 CALL NYÍL
11221 CALL BEOLVASÁS
11222 JX=5:CALL BEOLVASÁS
11223 PROC JOBBRA
11224 JX=JX+1
11225 IF JX>315 THEN 11229
11226 JT=0:JS=-1
11227 CALL NYÍL
11228 CALL BEOLVASÁS
11229 JX=315:CALL BEOLVASÁS
11230 PROC FEL
11231 JY=JY-1
11232 IF JY<5 THEN 11236
11233 JT=1:JS=0
11234 CALL NYÍL
11235 CALL BEOLVASÁS
11236 JY= 5:CALL BEOLVASÁS

```

```

11237 PROC LE
11238 JY=JY+1
11239 IF JY>195 THEN 11243
11240 JT=-1:JS=0
11241 CALL NYfL
11242 CALL BEOLVASÁS
11243 JY=195:CALL BEOLVASÁS
11244 PROC NYfL
11245 JW=JY+JT:JU=JX+JS:JQ=0:JU=0:JF=1:JR=1
11246 FOR JA=0 TO 2
11247 JN=JW-JA*JF
11248 JM=JU+JA
11250 :PLOT JM,JN,2
11252 NEXT JA
11253 JQ=JQ+1
11254 IF JQ>1 THEN 11257
11255 JF=-1
11256 GOTO 11246
11257 FOR JA=1 TO 2
11258 JN=JY-JA*JF
11260 JM=JX+JA
11261 :PLOT JM,JN,2
11263 NEXT JA
11264 JU=JU+1
11265 IF JU>1 THEN 11268
11266 JR=-1
11267 GOTO 11257
11268 GOTO 11198
11269 PROC VÉG
11270 JF=1:JS=0
11271 FOR JA=0 TO 2
11272 JN=JY-JA*JF
11274 JM=JX+JA
11275 :PLOT JM,JN,2
11277 NEXT JA
11278 JS=JS+1
11279 IF JS>1 THEN CALL LEZÁRÁS
11280 JF=-1
11281 GOTO 11271
11282 PROC START
11283 :TEXT 10,180,"START X ?",1,1,B
11284 INPUT B$
11285 :TEXT 90,180,B$,1,1,B
11286 JX=VAL(B$)
11287 :PAUSE 1
11288 :TEXT 10,180,"START X ?",0,1,B
11289 :TEXT 90,180,B$,0,1,B
11290 :TEXT 10,180,"START Y ?",1,1,B
11291 INPUT C$
11292 :TEXT 90,180,C$,1,1,B

```

```
11293 JY=VAL(C%)
11294 :PAUSE 1
11295 :TEXT 10,180,"START Y ?",0,1,8
11296 :TEXT 90,180,C%,0,1,8
11297 CALL KEZDÉS
11303 PROC LEZÁRÁS
11304 IF A$="S" THEN CALL START
11305 IF A$="U" THEN CALL VÁRAKOZÁS
11306 PROC VÉGE
11307 HIRES 0,7
11308 CALL START
11309 PROC VÁRAKOZÁS
11310 GOTO11310
```

2.3. Összetett elemek (makrók)

2.3.1. MI A MAKRÓ? MIRE HASZNÁLJUK A MAKRÓKAT?

Ha gyakran készítünk 3-dimenziós rajzokat, perspektívákat, fáradságos az alapalakzatok megjelenítését mindig a legkisebb elemekből, vonalakból, körökből stb. összeállítani. Hasznos lehet, ha a gyakran használt elemeket egyetlen hívással megjeleníthetjük. Az egyszerű elemekből összetett bonyolultabb elemeket nevezük makróknak. Ezek alapelemként használhatók a még bonyolultabb rajzok elkészítésénél. Az, hogy milyen makrókat állítunk össze magunknak, teljesen a felhasználás területétől függ. Ez azt jelenti, hogy a makrók feladatorientáltak. A következő oldalakon olyan makrókat állítunk elő, amelyek a műszaki rajzokon gyakran előfordulnak. Ezek a szabályos geometriai alakzatok, mint a téglatest, kocka, henger stb. A makróknak az alakja változatlan, de a változók különböző nagyságával és előjelével egyetlen makróval is nagyon sokféle változatot tudunk rajzolni. Ez a programkészletünket jelentősen kibővíti.

2.3.2. A C 64-ES CAD KÜLÖNBÖZŐ MAKRÓI

A számítógépünk az ún. Kavalier-féle axonometriában rajzolja fel az egyes ábrákat. Ennél a vetítési módnál a testek éleinek aránya a következő — szélesség: magasság: mélység = 1 : 1 : 0.5. A mélységi élek (z tengely) 45°-os szögben haladnak. Ez a megjelenítési mód nagyon megfelel a megjelenítő eszközeinknek, a képernyőnek és a nyomtatónak. Az arányszámok a pontok koordinátáinak egyszerű számítását teszik lehetővé, és a 45°-os vonalaknál a képernyőn nem képződnek lépcsők.

Kedvező az is, hogy a kétdimenziós ábrázolás x és y koordinátáit változatlanul megtarthatjuk. Ez megkönnyíti a makrók kombinálását.

Az egyes makrók a testeknek csak stilizált ábráját rajzolják fel. Ez különösen olyan elemekre vonatkozik, amelyeket görbült felületek határolnak, így pl. a kúpra és a gömbre. A makróinkkal drótváz modelleket rajzoltunk. Hogy ez pontosan mit jelent, a következő fejezetben láthatjuk.

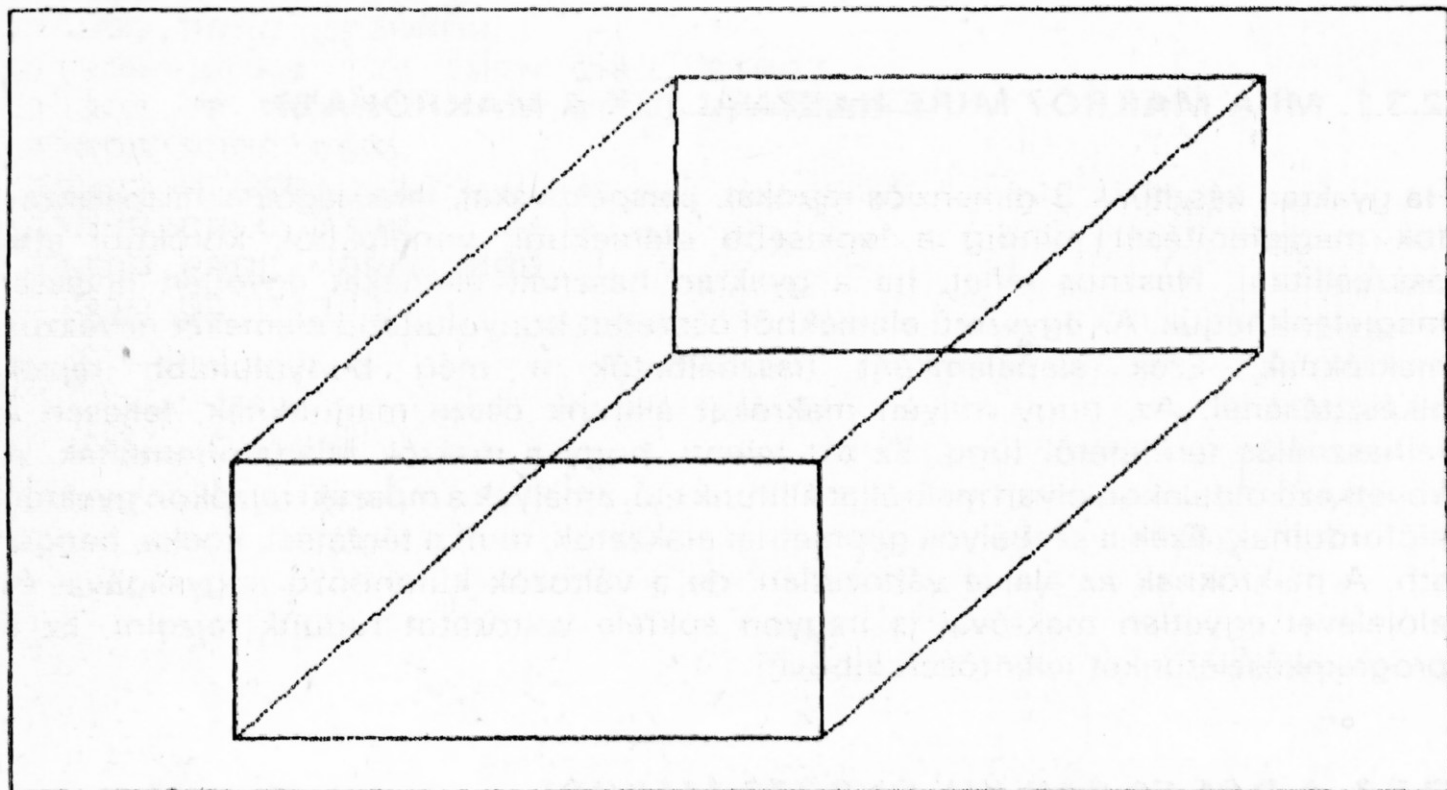
Még valami fontosat:

A testek mélységi méretét nem kell 0.5-tel szorozni, hanem a tényleges méretet kell megadni. A számítógép automatikusan rövidíti a méretet.

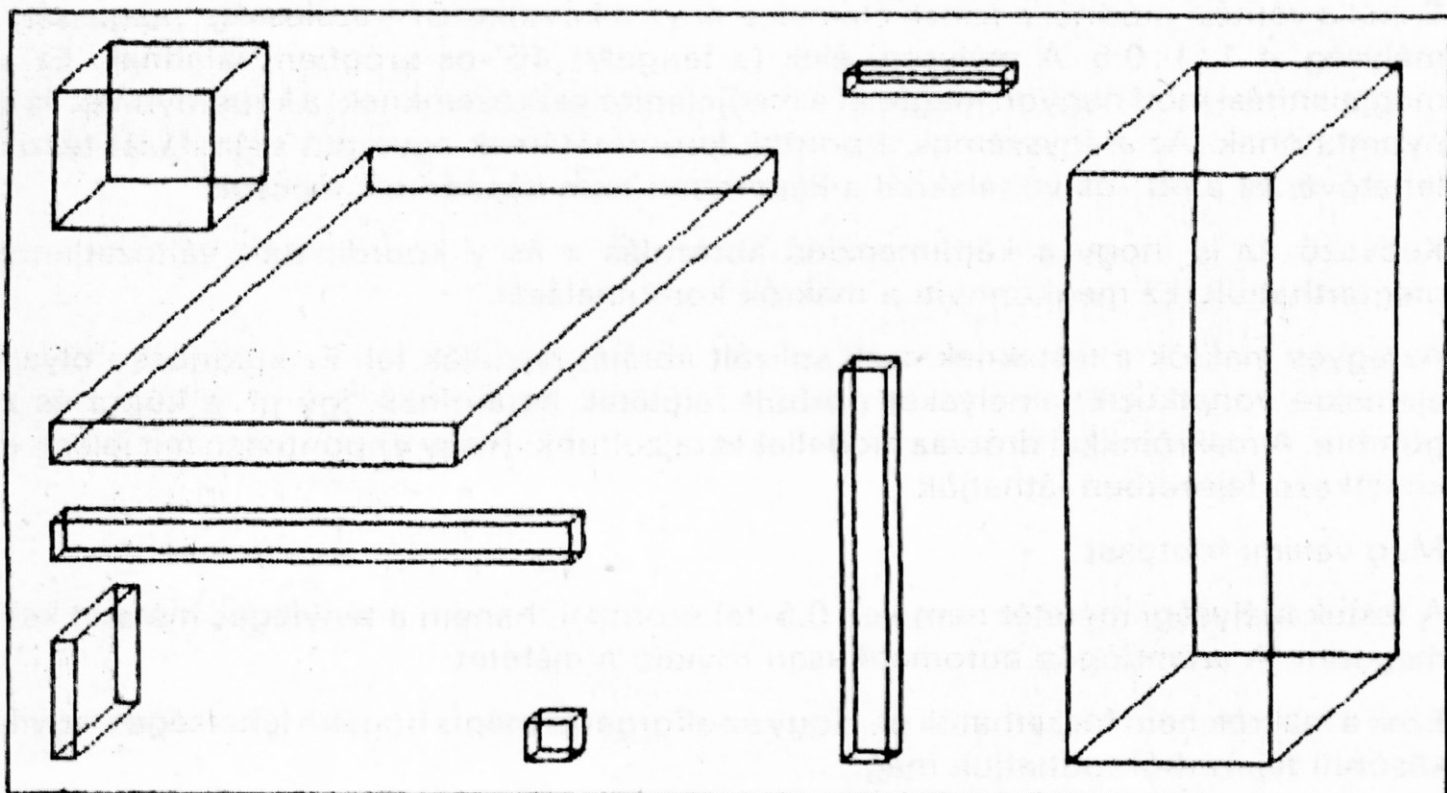
Ezek a makrók nem forgathatók el. Hogy az elforgatás mégis hogyan lehetséges, egyik későbbi fejezetből tudhatjuk meg.

2.3.2.1. Téglatest és kocka

A TÉGLA program tetszőleges téglatesteket, ha minden él azonos hosszúságú, akkor kockát rajzol.



21. ábra. A téglatest



22. ábra. A téglatest változatai

A 21. és 22. ábra néhány rajzolási lehetőséget mutat be.

Az X1 és Y1 változókkal a téglatest előlapjának bal felső sarokponti koordinátáit adhatjuk meg.

Az A1 változó adja meg a téglatest szélességét.

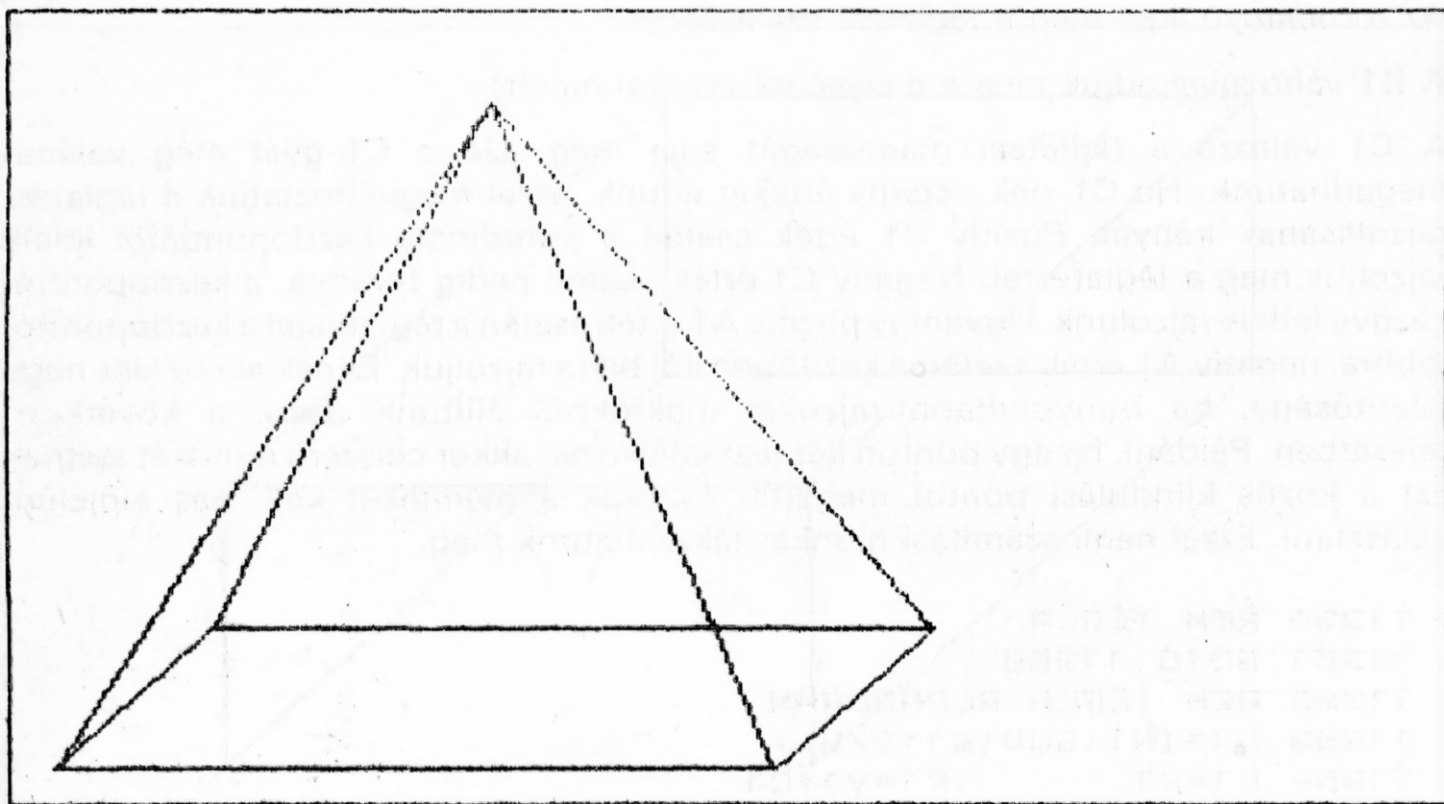
A B1 változóval adjuk meg a mélységet (valódi méret).

A C1 változó a téglatest magasságát adja meg. De a C1-gyel még valamit megadhatunk: Ha C1-nek negatív értéket adunk, ezzel megváltoztatjuk a téglatest rajzolásának irányát. Pozitív C1 érték esetén a koordináta kezdőpontjától lefelé rajzoljuk meg a téglatestet. Negatív C1 érték esetén pedig fordítva, a kezdőponttól kezdve felfelé rajzolunk. Ugyanígy pozitív A1 érték esetén a téglatestet a kezdőponttól jobbra, negatív A1 érték esetén a kezdőponttól balra rajzoljuk. Ennek akkor lesz nagy jelentősége, ha bonyolultabb rajzokat makrókból állítunk össze a következő fejezetben. Például, ha egy ponton két test találkozik, akkor célszerű mindkét testnek ezt a közös kiindulási pontot megadni és csak a méreteket kell más előjelre választani. Ezzel némi számítási munkát takaríthatunk meg.

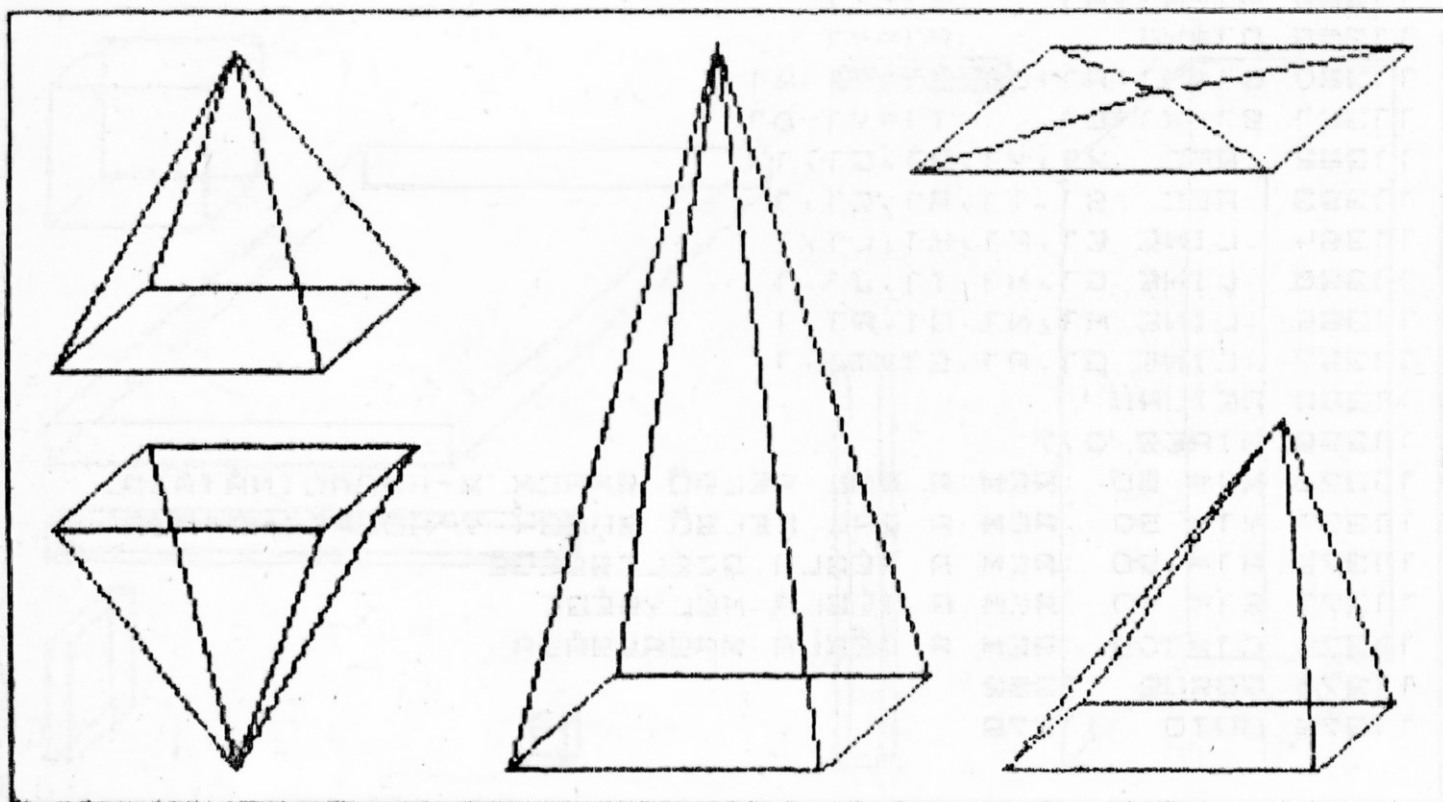
```
11350 REM TÉGLA
11351 GOTO 11369
11352 REM TÉGLA ALPROGRAM
11353 D1=INT(SQR(B1^2/B))
11354 E1=X1      :F1=Y1+C1
11355 G1=X1+A1  :H1=Y1+C1
11356 I1=X1+A1+D1:J1=Y1+C1-D1
11357 K1=X1+D1  :L1=Y1+C1-D1
11358 M1=X1+A1  :N1=Y1
11359 O1=X1     :P1=Y1
11360 Q1=X1+A1+D1:R1=Y1-D1
11361 S1=X1+D1  :T1=Y1-D1
11362 :REC X1,Y1,A1,C1,1
11363 :REC S1,T1,A1,C1,1
11364 :LINE E1,F1,K1,L1,1
11365 :LINE G1,H1,I1,J1,1
11366 :LINE M1,N1,O1,R1,1
11367 :LINE O1,P1,S1,T1,1
11368 RETURN
11369 HIRES 0,7
11370 X1= 50 :REM A BAL FELSŐ SAROK X-KOORDINÁTÁJA
11371 Y1= 50 :REM A BAL FELSŐ SAROK Y-KOORDINÁTÁJA
11372 A1= 80 :REM A TÉGLA SZÉLESSÉGE
11373 B1= 40 :REM A TÉGLA MÉLYSÉGE
11374 C1=100 :REM A TÉGLA MAGASSÁGA
11375 GOSUB 11352
11376 GOTO 11376
```

2.3.2.2. Gúla

A GÚLA program tetszőleges, téglalap alaprajzú gúlát tud felrajzolni.



23. ábra. A gúla



24. ábra. A gúla változatai

A 23. és 24. ábra mutatja be azokat a változatokat, amelyeket ezzel a programmal elő lehet állítani.

Az X2 és Y2 változókkal adhatjuk meg a gúla alaplapján a bal alsó sarokpont koordinátáit.

A2-vel az alaplap szélességét határozhatjuk meg. Pozitív A2 esetén a gúlát a kiindulási ponttól jobbra rajzoljuk; negatív A2 esetén balra rajzoljuk.

B2 a gúla mélységét adja meg (valódi méret).

C2-vel adhatjuk meg, hogy a gúla milyen magas legyen. Pozitív C2 esetén a gúla csúcsát felfelé, negatív C2 esetén lefelé rajzoljuk.

Ebbe a programba még egy további változót is bevezettünk, a Z2 a torzítási tényezőt. Ez matematikai értelemben nem tényező, hanem tag. De a szóhasználatban nem ragaszkodunk mindig a matematikai szakkifejezésekhez. A Z2 segítségével a gúla csúcsát X irányba eltolhatjuk. Ha Z2 pozitív, akkor a gúla csúcsát a megadott értékkel jobbra, ha negatív, akkor balra toljuk. Így ferde gúlákat is kaphatunk. Ez természetesen akkor is működik, ha egy negatív C2 értékkel a gúlát a feje tetejére állítottuk.

```
11380 REM GÚLA
11381 GOTO 11400
11382 REM GÚLA ALPROGRAM
11383 D2=INT(SQR(B2↑2/B))
11384 U2=INT(D2/2)
11385 V2=INT(R2/2)
11386 E2=X2 : F2=Y2
11387 G2=X2+A2 : H2=Y2
11388 I2=X2+A2+D2 : J2=Y2-D2
11389 K2=X2+D2 : L2=Y2-D2
11390 M2=X2+V2+U2+Z2 : N2=Y2-U2-C2
11391 :LINE E2,F2,G2,H2,1
11392 :LINE G2,H2,I2,J2,1
11393 :LINE I2,J2,K2,L2,1
11394 :LINE K2,L2,E2,F2,1
11395 :LINE E2,F2,M2,N2,1
11396 :LINE G2,H2,M2,N2,1
11397 :LINE I2,J2,M2,N2,1
11398 :LINE K2,L2,M2,N2,1
11399 RETURN
11400 HIRIS 0,7
11401 X2=100 :REM A BAL ALSÓ SAROK X-KOORDINÁTÁJA
11402 Y2=100 :REM A BAL ALSÓ SAROK Y-KOORDINÁTÁJA
11403 Z2= 0 :REM FERDESÉG
11404 A2= 50 :REM A GÚLA SZÉLESSÉGE
11405 B2= 60 :REM A GÚLA MÉLYSÉGE
11406 C2= 80 :REM A GÚLA MAGASSÁGA
11407 GOSUB 11382
11408 GOTO 11408
```

2.3.2.3. Háromszög alapú hasáb

A PRIZMA program tetszőleges háromszög alapú hasábot, éket vagy tetőidomot rajzol.

Az X3 és Y3 változókkal adjuk meg a kezdőpontot. Ez általában pozitív A3 érték esetén az alaplap bal alsó sarka.

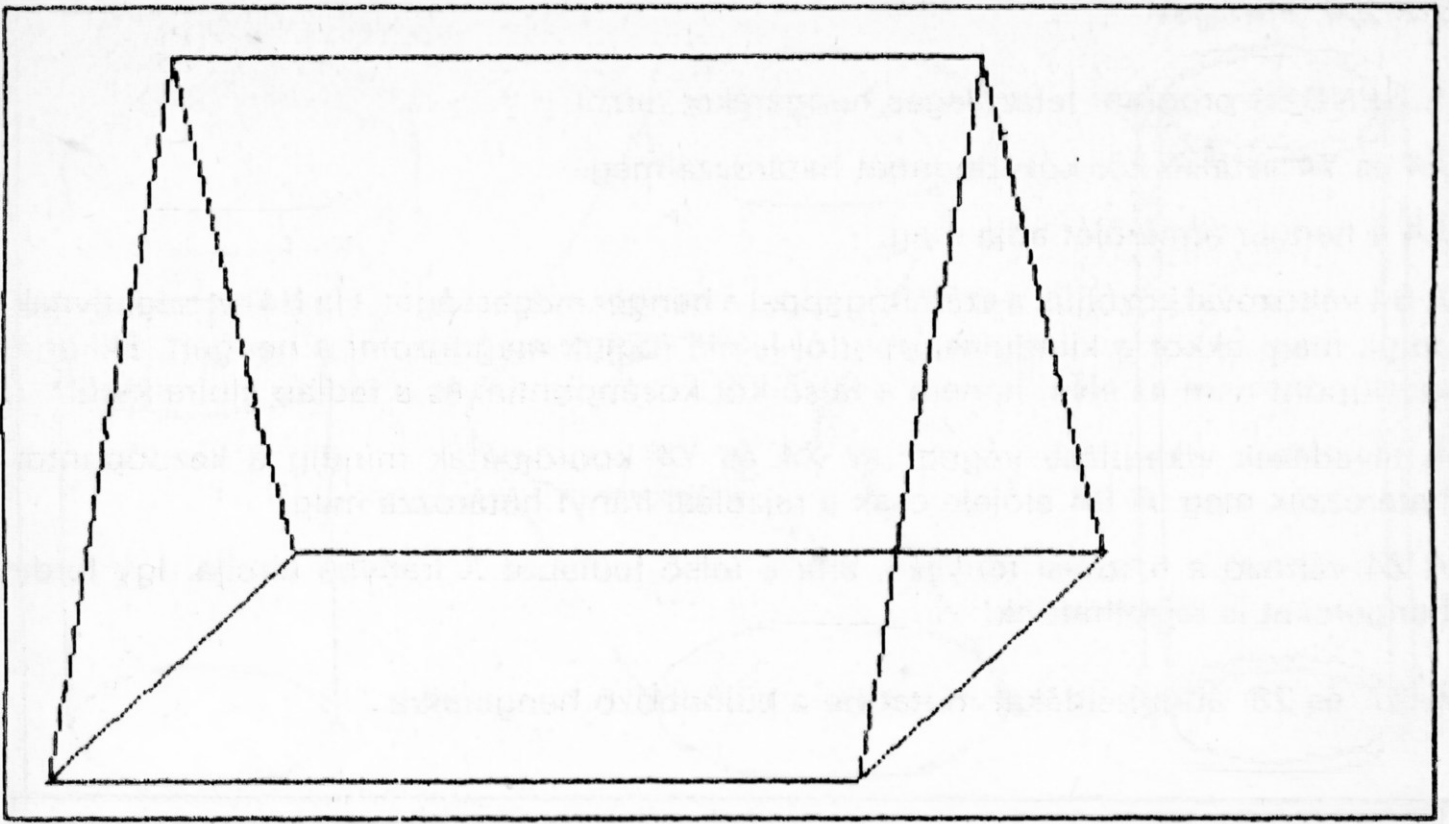
Az A3 értékkel határozzuk meg a szélességet. Az A3 változó előjele határozza meg, hogy melyik irányba rajzoljuk fel az éktestet.

B3 a mélységi méret (valódi méret).

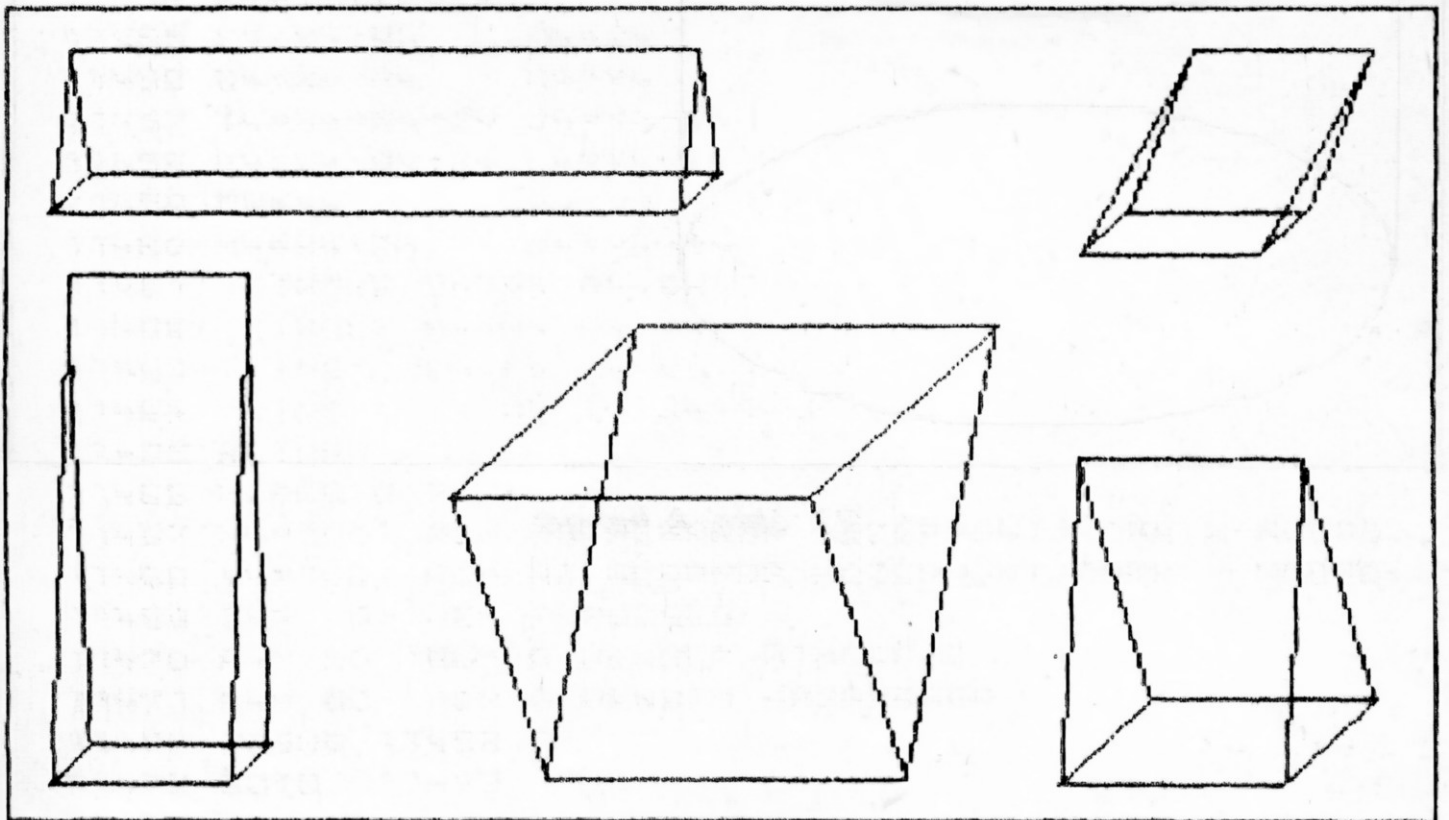
C3-mal a magasságot határozzuk meg. Az előjel határozza meg ismét, hogy felfelé vagy lefelé rajzoljuk-e a hasábot.

Z3 a torzítási tényező, ami a tető gerincvonalának X irányú eltolását határozza meg, és lehetővé teszi ferde vonalú tetők rajzolását. Z3 előjele határozza meg az eltolás irányát. A 25. és 26. ábra az ezzel a programmal rajzolható testek néhány változatát mutatja be.

```
11413 REM PRIZMA
11414 GOTO 11435
11415 REM PRIZMA ALPROGRAM
11416 D3=INT(SQR(B3^2/B))
11417 U3=INT(D3/2)
11418 E3=X3          :F3=Y3
11419 G3=X3+A3      :H3=Y3
11420 I3=X3+A3+D3  :J3=Y3-D3
11421 K3=X3+D3     :L3=Y3-D3
11422 O3=X3+U3+Z3 :P3=Y3-U3-C3
11423 M3=X3+A3+U3+Z3:N3=Y3-U3-C3
11425 :LINE E3,F3,G3,H3,1
11426 :LINE G3,H3,I3,J3,1
11427 :LINE I3,J3,K3,L3,1
11428 :LINE K3,L3,E3,F3,1
11429 :LINE E3,F3,O3,P3,1
11430 :LINE K3,L3,O3,P3,1
11431 :LINE G3,H3,M3,N3,1
11432 :LINE I3,J3,M3,N3,1
11433 :LINE O3,P3,M3,N3,1
11434 RETURN
11435 HIRES 0,7
11436 X3= 50 :REM A BAL ALSÓ SAROK X-KOORDINÁTÁJA
11437 Y3=100 :REM A BAL ALSÓ SAROK Y-KOORDINÁTÁJA
11438 Z3= 0 :REM FERDESÉG
11439 A3=100 :REM A PRIZMA SZÉLESSÉGE
11440 B3= 60 :REM A PRIZMA MÉLYSÉGE
11441 C3= 50 :REM A PRIZMA MAGASSÁGA
11442 GOSUB 11415
11443 GOTO 11443
```



25. ábra. A háromszög alapú hasáb



26. ábra. A háromszög alapú hasáb változatai

2.3.2.4. Henger

A HENGER program tetszőleges hengereket rajzol.

X4 és Y4 az alsó kör középpontját határozza meg.

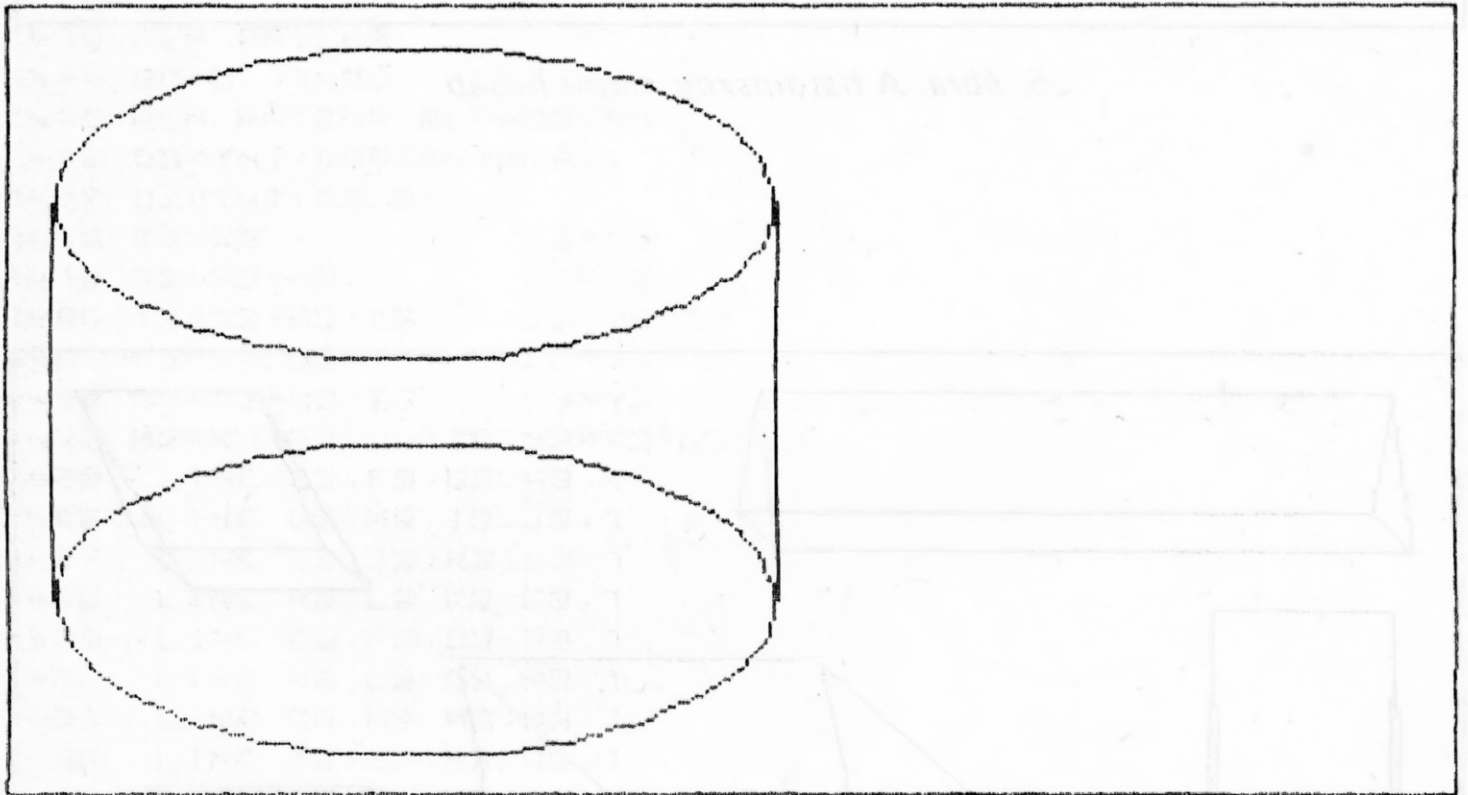
A4 a henger átmérőjét adja meg.

A B4 változóval közöljük a számítógéppel a henger magasságát. Ha B4-et negatívnak adjuk meg, akkor a kiindulási ponttól lefelé fogjuk megrajzolni a hengert. Ekkor a kezdőpont nem az alsó, hanem a felső kör középpontja, és a fedlap alulra kerül.

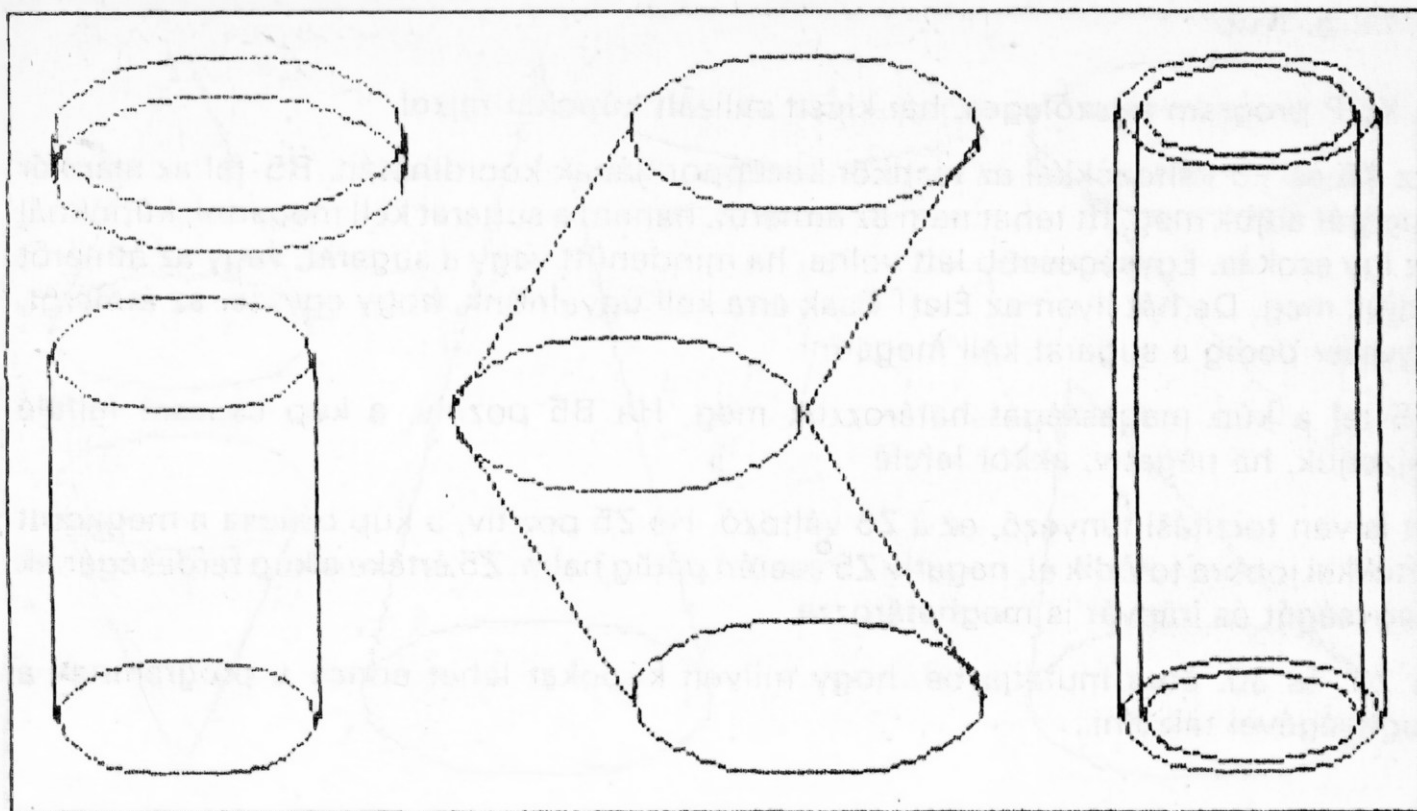
A tévedések elkerülése végett: az X4 és Y4 koordináták mindig a kezdőpontot határozzák meg. A B4 előjele csak a rajzolási irányt határozza meg.

A Z4 változó a torzítási tényező, ami a felső fedlapot X irányba eltolja. Így ferde hengereket is rajzolhatunk.

A 27. és 28. ábra példákat mutat be a különböző hengerekre.



27. ábra. A henger



28. ábra. A henger változatai

```

11450 REM HENGER
11451 GOTO 11466
11452 REM HENGER ALPROGRAM
11453 R4=INT(R4/2)
11454 S4=INT(R4/2)
11455 E4=X4-R4 : F4=Y4
11456 G4=X4+R4 : H4=Y4
11457 I4=X4+R4+Z4 : J4=Y4-B4
11458 K4=X4-R4+Z4 : L4=Y4-B4
11459 O4=X4 : P4=Y4
11460 M4=X4+Z4 : N4=Y4-B4
11461 :CIRCLE O4,P4,R4,S4,1
11462 :CIRCLE M4,N4,R4,S4,1
11463 :LINE E4,F4,K4,L4,1
11464 :LINE G4,H4,I4,J4,1
11465 RETURN
11466 HIRES 0,7
11467 X4=100 :REM AZ ALAPKÖR KÖZÉPPONTJÁNAK X-KOORD.
11468 Y4=100 :REM AZ ALAPKÖR KÖZÉPPONTJÁNAK Y-KOORD.
11469 Z4= 0 :REM FERDESÉG
11470 R4= 50 :REM A HENGER ÁTMÉRŐJE
11471 B4= 80 :REM A HENGER MAGASSÁGA
11472 GOSUB 11452
11473 GOTO 11473

```

2.3.2.5. Kúp

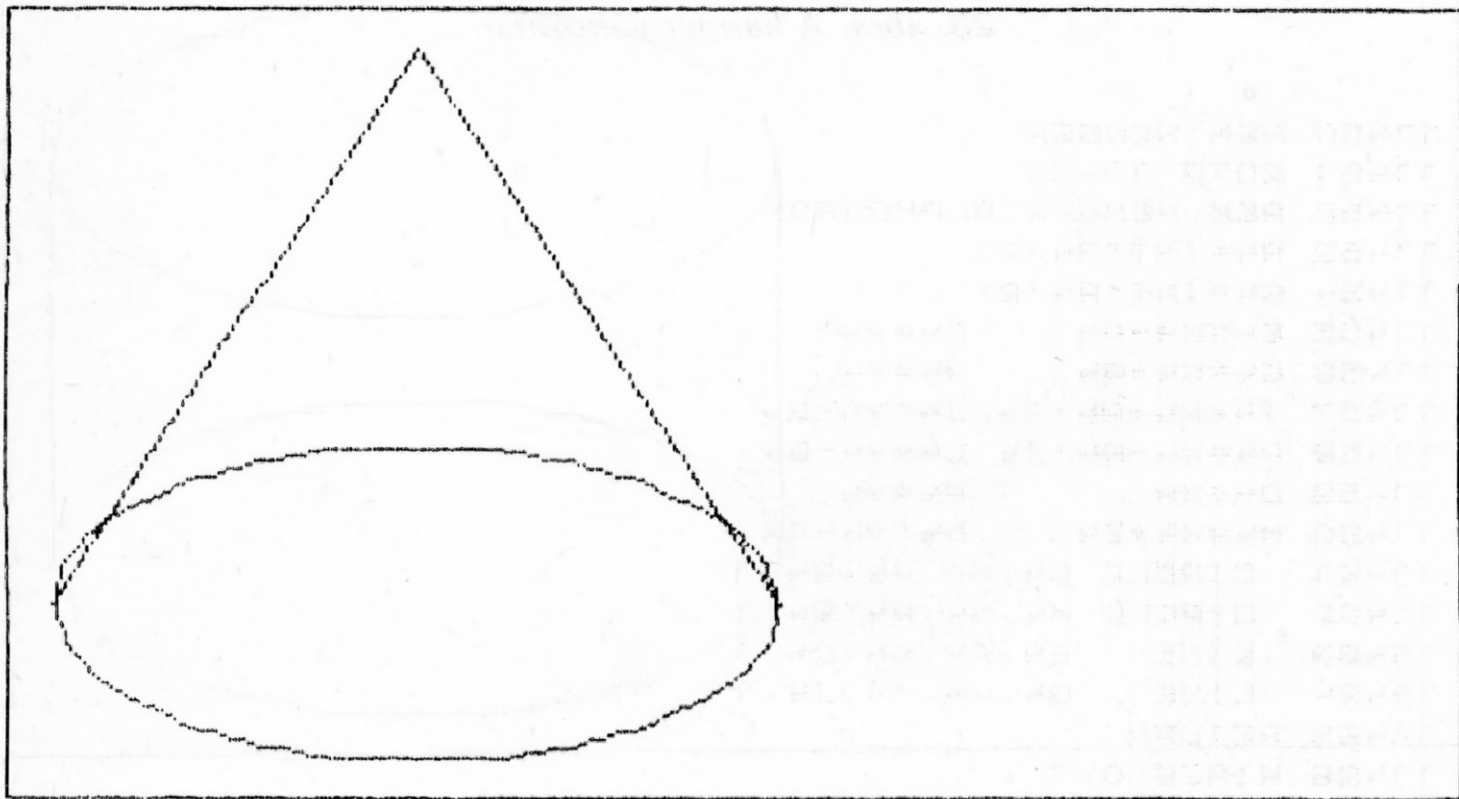
A KÚP program tetszőleges, bár kicsit stilizált kúpokat rajzol.

Az X5 és Y5 változókkal az alapkör középpontjának koordinátáit, R5-tel az alapkör sugarát adjuk meg. Itt tehát nem az átmérőt, hanem a sugarat kell megadni, kúpoknál ez így szokás. Egységesebb lett volna, ha mindenütt vagy a sugarat, vagy az átmérőt adjuk meg. De hát ilyen az Élet! Csak arra kell ügyelnünk, hogy egyszer az átmérőt, egyszer pedig a sugarat kell megadni.

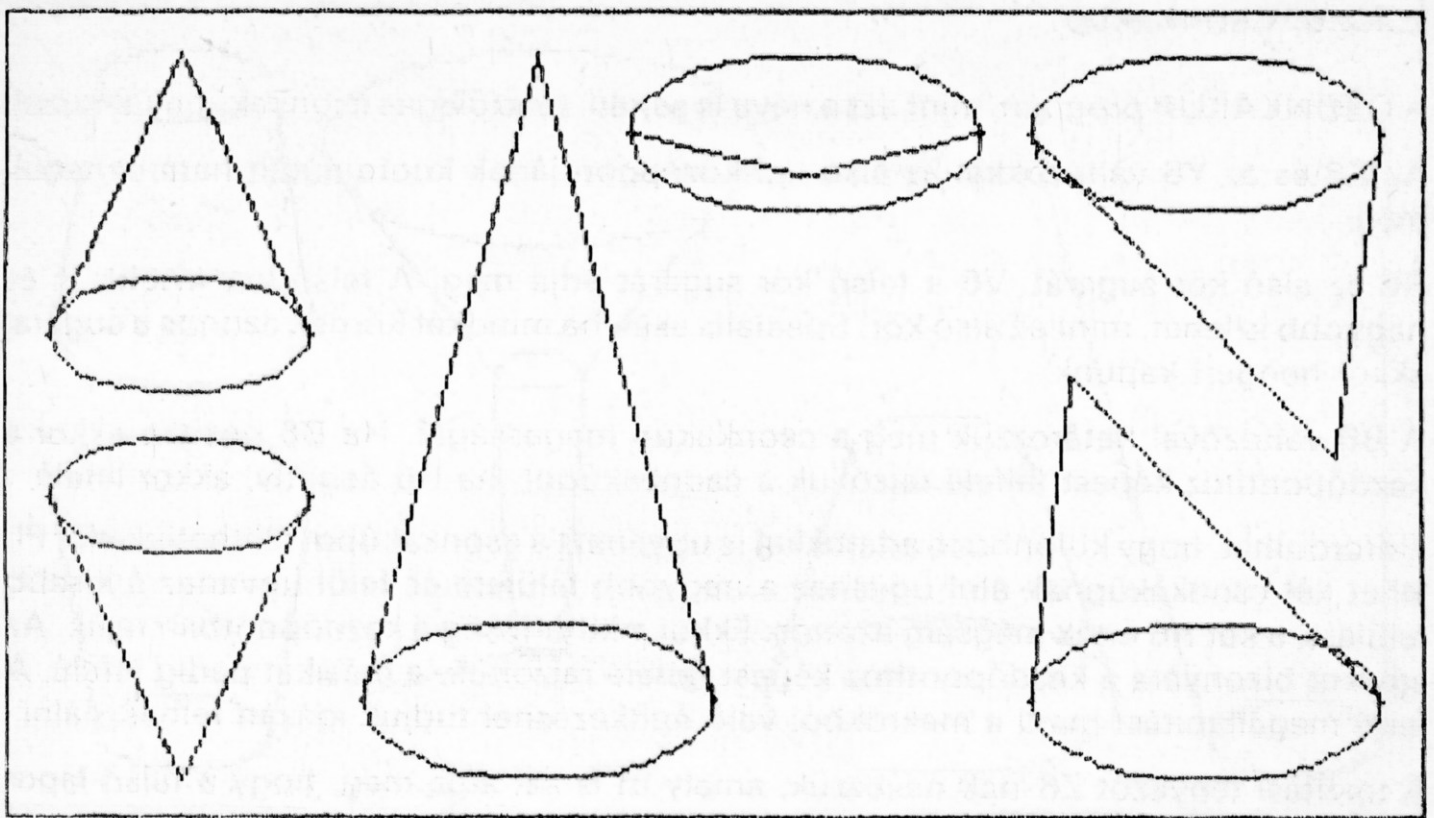
B5-tel a kúp magasságát határozzuk meg. Ha B5 pozitív, a kúp csúcsát felfelé rajzoljuk, ha negatív, akkor lefelé.

Itt is van torzítási tényező, ez a Z5 változó. Ha Z5 pozitív, a kúp csúcsa a megadott értékkel jobbra tolódik el, negatív Z5 esetén pedig balra. Z5 értéke a kúp ferdeségének nagyságát és irányát is meghatározza.

A 29. és 30. ábra mutatja be, hogy milyen kúpokat lehet ennek a programnak a segítségével rajzolni.



29. ábra. A kúp



30. ábra. A kúp változatai

```

11480 REM KÚP
11481 GOTO 11492
11482 REM KÚP ALPROGRAM
11483 S5=INT(R5/2)
11484 E5=X5-R5:F5=Y5
11485 G5=X5+R5:H5=Y5
11486 I5=X5+Z5:J5=Y5-B5
11487 K5=X5      :L5=Y5
11488 :CIRCLE K5,L5,R5,S5,1
11489 :LINE E5,F5,I5,J5,1
11490 :LINE G5,H5,I5,J5,1
11491 RETURN
11492 HIRES 0,7
11493 X5=100 :REM AZ ALAPKÖR KÖZÉPPONTJÁNAK X-KOORD.
11494 Y5=100 :REM AZ ALAPKÖR KÖZÉPPONTJÁNAK Y-KOORD.
11495 Z5= 0 :REM FERDESÉG
11496 R5= 50 :REM AZ ALAPKÖR SUGARA
11497 B5= 80 :REM A KÚP MAGASSÁGA
11498 GOSUB 11482
11499 GOTO 11499

```

2.3.2.6. Csonkakúp

A CSONKAKÚP program, mint azt a neve is sejteti, tetszőleges csonkakúpokat rajzol. Az X6 és az Y6 változókkal az alsó lap középpontjának koordinátáit határozhatjuk meg.

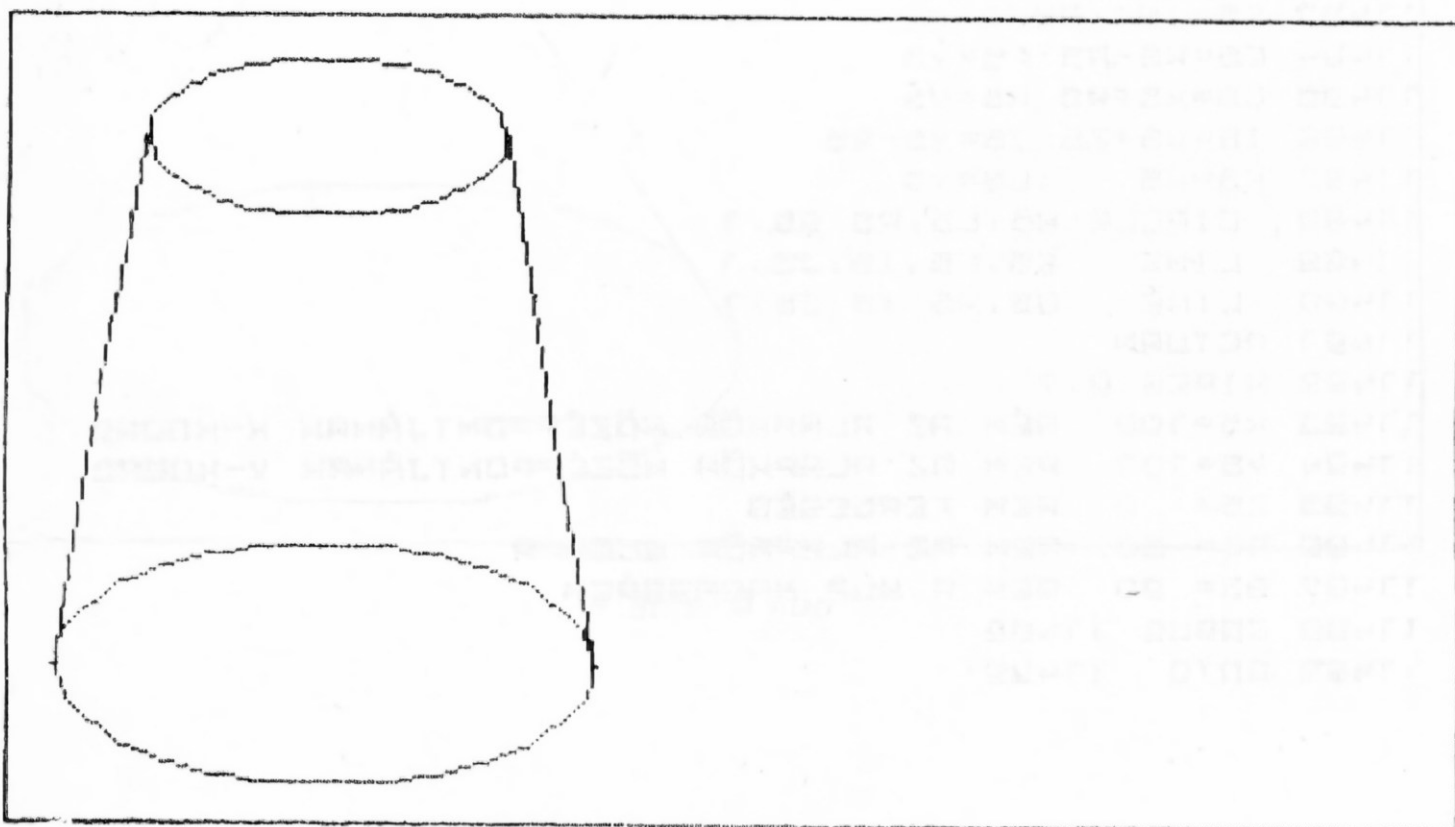
R6 az alsó kör sugarát, V6 a felső kör sugarát adja meg. A felső kör kisebb is és nagyobb is lehet, mint az alsó kör. Speciális eset, ha mindkét körnek azonos a sugara, akkor hengert kapunk.

A B6 változóval határozzuk meg a csonkakúp magasságát. Ha B6 pozitív, akkor a kezdőponthoz képest felfelé rajzoljuk a csonkakúpot, ha B6 negatív, akkor lefelé.

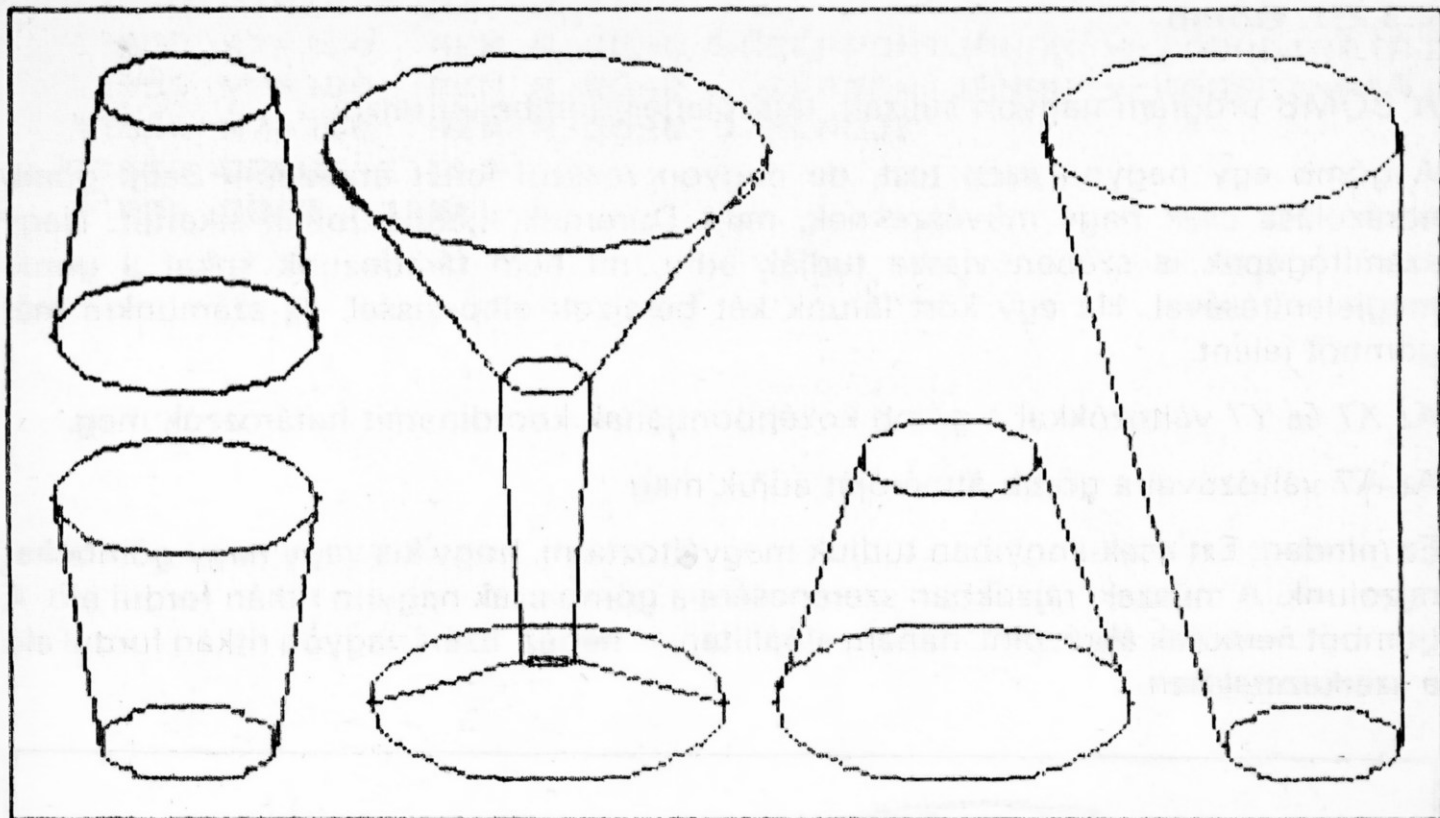
Előfordulhat, hogy különböző adatokkal is ugyanazt a csonkakúpot állíthatjuk elő. Pl. lehet két csonkakúpunk alul ugyanaz a nagyobb felülete és felül ugyanaz a kisebb felülete, a két R6 érték mégsem azonos. Ekkor a különbség a kezdőpontban rajlik. Az egyiket bizonyára a kezdőponthoz képest felfelé rajzoltuk, a másikat pedig lefelé. A fenti megállapítást majd a makrókból való építkezésnél tudjuk igazán felhasználni.

A torzítási tényezőt Z6-nak nevezzük, amely itt is azt adja meg, hogy a felső lapot mennyivel, és az X tengelyen milyen irányba kell eltolni, hogy a megfelelő ferde csonkakúpot megkapjuk.

A 31. és 32. ábra a programmal előállítható csonkakúpok néhány változatát mutatja be.



31. ábra. A csonkakúp



32. ábra. A csonkakúp változatai

```

11508 REM CSONKAKÚP
11509 GOTO 11524
11510 REM CSONKAKÚP ALPROGRAM
11511 SB=INT(RB/2)
11512 WB=INT(UB/2)
11513 EB=XB-RB      :FB=YB
11514 GB=XB+RB      :HB=YB
11515 IB=XB+UB+ZB :JB=YB-BB
11516 KB=XB-UB+ZB :LB=YB-BB
11517 DB=XB+ZB     :PB=YB-BB
11518 MB=XB        :NB=YB
11519 :CIRCLE MB,NB,RB,SB,1
11520 :CIRCLE DB,PB,UB,WB,1
11521 :LINE   EB,FB,KB,LB,1
11522 :LINE   GB,HB,IB,JB,1
11523 RETURN
11524 HIRÉS 0,7
11525 XB=100 :REM AZ ALPKÖR KÖZÉPPONTJÁNAK X-KOORD.
11526 YB=100 :REM AZ ALPKÖR KÖZÉPPONTJÁNAK Y-KOORD.
11527 ZB= 0  :REM FERDESÉG
11528 RB= 50 :REM AZ ALPKÖR SUGARA
11529 UB= 20 :REM A FELSŐ KÖR SUGARA
11530 BB= 80 :REM A CSONKAKÚP MAGASSÁGA
11531 GOSUB 11510
11532 GOTO 11532

```

2.3.2.7. Gömb

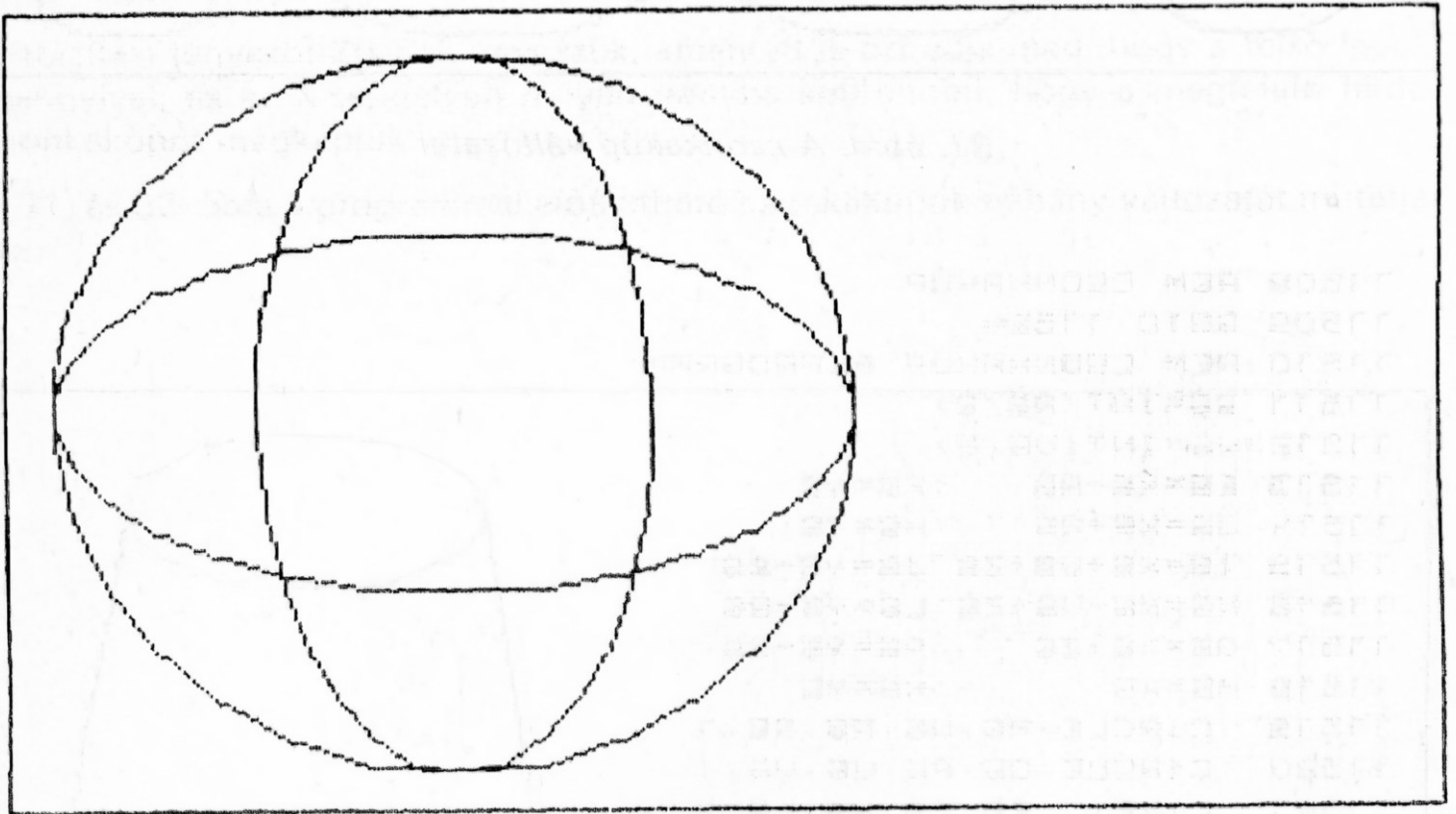
A GÖMB program nagyon stilizált, tetszőleges gömböket rajzol.

A gömb egy nagyon szép test, de nagyon rosszul lehet ábrázolni. Szép gömb ábrázolása csak nagy művészeknek, mint Dürernek, Leonardonak sikerült. Nagy számítógépek is szépen vissza tudják adni, mi nem fáradozunk sokat a gömb megjelenítésével. Ha egy kört látunk két berajzolt ellipszissel, ez számunkra már gömböt jelent.

Az X7 és Y7 változókkal a gömb középpontjának koordinátáit határozzuk meg.

Az A7 változóval a gömb átmérőjét adjuk meg.

Ez minden. Ezt csak annyiban tudjuk megváltoztatni, hogy kis vagy nagy gömböket rajzolunk. A műszaki rajzokban szerencsére a gömb csak nagyon ritkán fordul elő. A gömböt nemcsak ábrázolni, hanem előállítani is nehéz. Ezért nagyon ritkán fordul elő a szerkezetekben.



33. ábra. A gömb

```
11540 REM GÖMB
11541 GOTO 11549
11542 REM GÖMB ALPROGRAM
11543 B7=INT(A7/2)
11544 C7=INT(A7/4)
11545 :CIRCLE X7,Y7,B7,B7,1
11546 :CIRCLE X7,Y7,B7,C7,1
11547 :CIRCLE X7,Y7,C7,B7,1
11548 RETURN
```

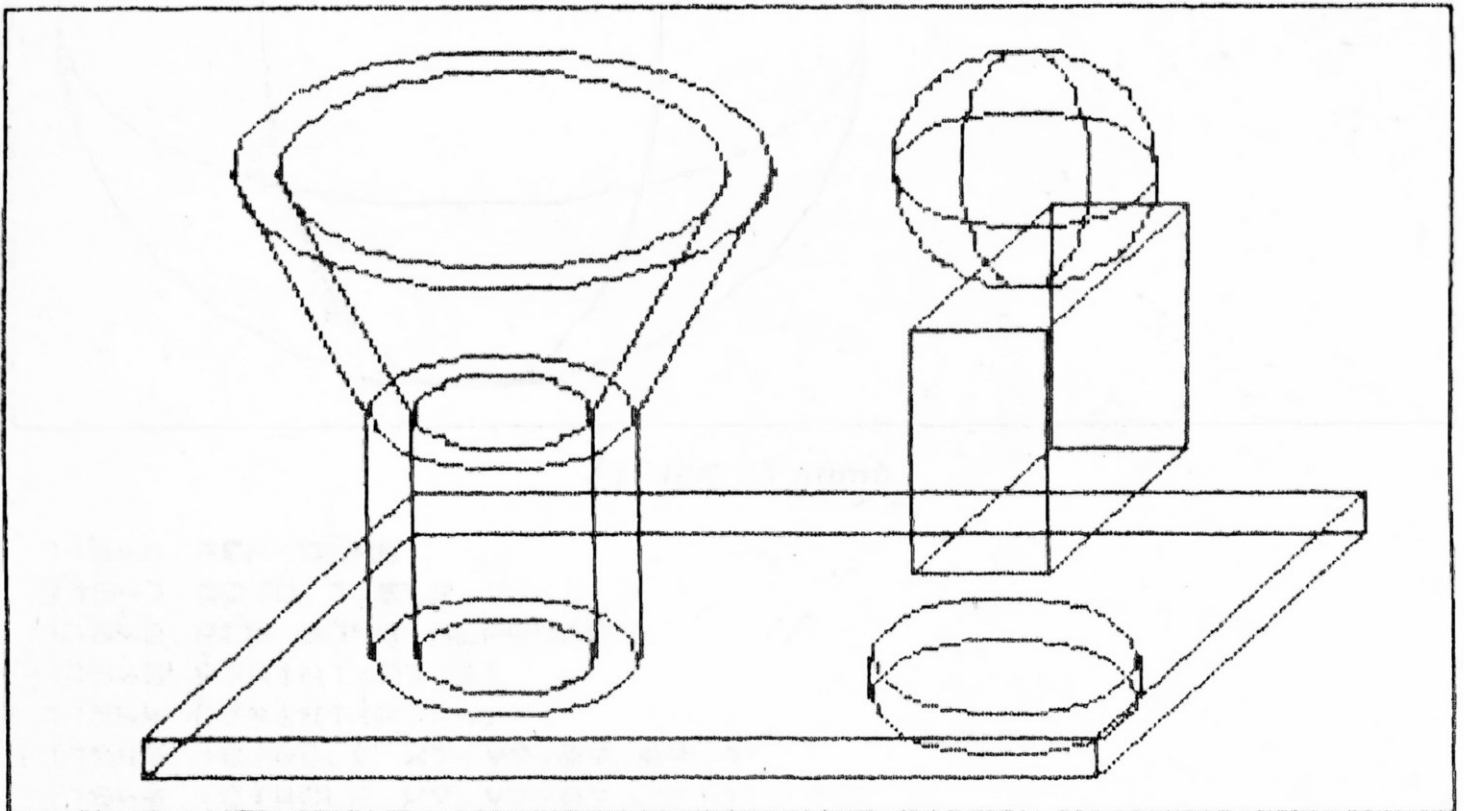

2.4. Makrókból összeállított háromdimenziós rajzok

A különböző makrókat tartalmazó programokat nem öncélúan készítettük. Önmagukban is már jó rajzokat készíthetünk velük, de összetettebb ábrákat is szeretnénk segítségükkel készíteni. Ezért foglaljuk egybe a makrókat úgy, mint azt az alapelemeket rajzoló programoknál is tettük. Az áttekinthetőség kedvéért változtassuk át a makró programokat eljárásokká, és helyezzük el az eljárásokat egymás után.

Egy ÁBRA program bonyolult ábrákat egyszerűen állít elő makrók segítségével. Természetesen újból kijelölhetünk rétegeket, és a makró eljárásokat összekapcsolhatjuk az alapelemeket rajzoló eljárásokkal is. Az így előállított rajzokat háromféle módon jeleníthetjük meg. Három ábrázolási modelltől beszélhetünk. Most ezeket az ábrázolási modelleket szeretnénk bemutatni.

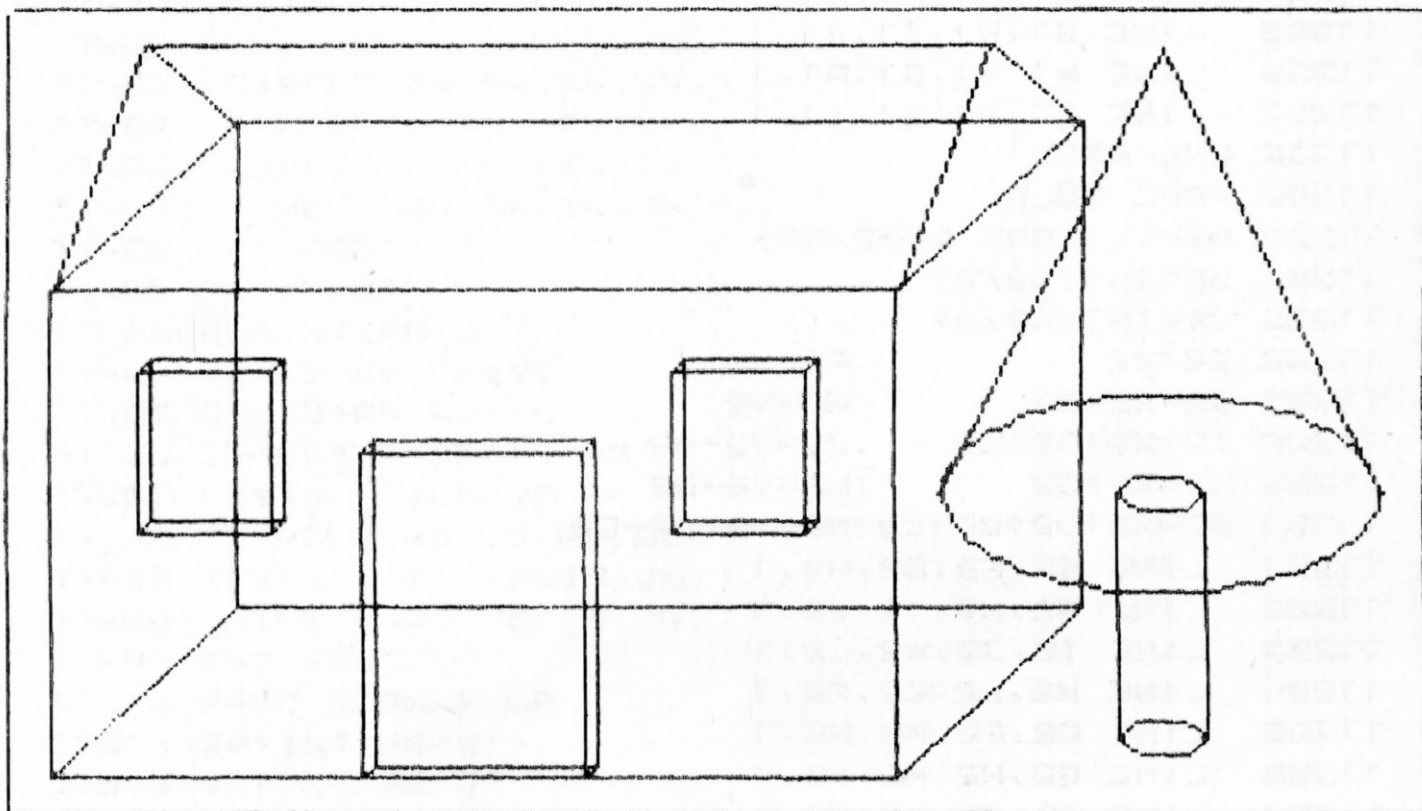
2.4.1. DRÓTVÁZMODELL

A drótvázmodell egy vagy több testnek háromdimenziós ábrázolása, ahol minden egyes él látható. Az ábra olyan, mintha drótból lennének a testek, ezért a különös



34. ábra. Egy makrókból összeállított drótvázmodell rajz

elnevezés. Ennek az ábrázolási módnak az az előnye, hogy a számítógép munkáját jelentősen megkönnyíti. A programnak nem kell eldöntenie, hogy melyek a látható és a láthatatlan élek. Ha bárhol egy élet le kell rajzolni, akkor úgy rajzolja, mintha minden esetben látható lenne. A programozás, az adatmegadás, a számítás, a tárolókapacitás és a számításiidő-szükséglet a lehető legkisebb. A hátránya egy nem egészen világos ábrázolás. Sokszor nem tudjuk, milyen is valójában az ábrázolt test. A szemlélő fantáziája és térlátása függvényében igen különböző hatásokat válthat ki egy ilyen ábra. Összetett rajzok esetében nagyon kellemetlen lehet ez a hatás.



35. ábra. Egy makrókból összeállított drótvázmodell rajz

A 34. és 35. ábrán mutatjuk be, hogyan néznek ki a drótvázmodell rajzok.

A DRÓT1 program mutatja be, mekkora munkát igényel egy ilyen rajznak az előállítás. Ne feledjük el, hogy a legtöbb munkát adja a már megírt eljárások összeválogatása, amit a 11350—11548. sorban mutatunk be.

Ezután kezdődik az ÁBRA program, ami a 34. ábrát rajzolja ki. Ha a 35. ábrát akarjuk kirajzolni, csak ezt az ÁBRA programot kell újabb adatokkal ellátni. Ehhez nem kell sokat tennünk. Néhány sor megváltoztatásával teljesen más képet kaphatunk.

Bizonyára felismerték a makrók használatának nagy előnyét.

```

11350 REM DRÓT1
11351 GOTO 11550
11352 PROC TÉGLA
11353 D1=INT(SQR(B1↑2/B))
11354 E1=X1          :F1=Y1+C1

```

```

71355 G1=X1+A1      :H1=Y1+C1
71356 I1=X1+A1+D1 :J1=Y1+C1-D1
71357 K1=X1+D1     :L1=Y1+C1-D1
71358 M1=X1+A1     :N1=Y1
71359 O1=X1        :P1=Y1
71360 Q1=X1+A1+D1 :R1=Y1-D1
71361 S1=X1+D1     :T1=Y1-D1
71362 :REC  X1,Y1,A1,C1,1
71363 :REC  S1,T1,A1,C1,1
71364 :LINE E1,F1,K1,L1,1
71365 :LINE G1,H1,I1,J1,1
71366 :LINE M1,N1,Q1,R1,1
71367 :LINE O1,P1,S1,T1,1
71368 END PROC
71382 PROC GÚLA
71383 D2=INT(SQR(B2↑2/B))
71384 U2=INT(D2/2)
71385 V2=INT(A2/2)
71386 E2=X2          :F2=Y2
71387 G2=X2+A2      :H2=Y2
71388 I2=X2+A2+D2  :J2=Y2-D2
71389 K2=X2+D2     :L2=Y2-D2
71390 M2=X2+V2+U2+Z2 :N2=Y2-U2-C2
71391 :LINE  E2,F2,G2,H2,1
71392 :LINE  G2,H2,I2,J2,1
71393 :LINE  I2,J2,K2,L2,1
71394 :LINE  K2,L2,E2,F2,1
71395 :LINE  E2,F2,M2,N2,1
71396 :LINE  G2,H2,M2,N2,1
71397 :LINE  I2,J2,M2,N2,1
71398 :LINE  K2,L2,M2,N2,1
71399 END PROC
71415 PROC PRIZMA
71416 D3=INT(SQR(B3↑2/B))
71417 U3=INT(D3/2)
71418 E3=X3          :F3=Y3
71419 G3=X3+A3      :H3=Y3
71420 I3=X3+A3+D3  :J3=Y3-D3
71421 K3=X3+D3     :L3=Y3-D3
71422 O3=X3+U3+Z3  :P3=Y3-U3-C3
71423 M3=X3+A3+U3+Z3 :N3=Y3-U3-C3
71425 :LINE  E3,F3,G3,H3,1
71426 :LINE  G3,H3,I3,J3,1
71427 :LINE  I3,J3,K3,L3,1
71428 :LINE  K3,L3,E3,F3,1
71429 :LINE  E3,F3,O3,P3,1
71430 :LINE  K3,L3,O3,P3,1
71431 :LINE  G3,H3,M3,N3,1
71432 :LINE  I3,J3,M3,N3,1

```



```

11433 :LINE 03,P3,M3,N3,1
11434 END PROC
11452 PROC HENGER
11453 R4=INT<R4/2>
11454 S4=INT<R4/2>
11455 E4=X4-R4 :F4=Y4
11456 G4=X4+R4 :H4=Y4
11457 I4=X4+R4+Z4 :J4=Y4-B4
11458 K4=X4-R4+Z4 :L4=Y4-B4
11459 O4=X4 :P4=Y4
11460 M4=X4+Z4 :N4=Y4-B4
11461 :CIRCLE O4,P4,R4,S4,1
11462 :CIRCLE M4,N4,R4,S4,1
11463 :LINE E4,F4,K4,L4,1
11464 :LINE G4,H4,I4,J4,1
11465 END PROC
11482 PROC KÚP
11483 S5=INT<R5/2>
11484 E5=X5-R5 :F5=Y5
11485 G5=X5+R5 :H5=Y5
11486 I5=X5+Z5 :J5=Y5-B5
11487 K5=X5 :L5=Y5
11488 :CIRCLE K5,L5,R5,S5,1
11489 :LINE E5,F5,I5,J5,1
11490 :LINE G5,H5,I5,J5,1
11491 END PROC
11510 PROC CSONKAKÚP
11511 S6=INT<R6/2>
11512 W6=INT<V6/2>
11513 E6=X6-R6 :F6=Y6
11514 G6=X6+R6 :H6=Y6
11515 I6=X6+V6+Z6 :J6=Y6-B6
11516 K6=X6-V6+Z6 :L6=Y6-B6
11517 O6=X6+Z6 :P6=Y6-B6
11518 M6=X6 :N6=Y6
11519 :CIRCLE M6,N6,R6,S6,1
11520 :CIRCLE O6,P6,V6,W6,1
11521 :LINE E6,F6,K6,L6,1
11522 :LINE G6,H6,I6,J6,1
11523 END PROC
11542 PROC GÖMB
11543 B7=INT<R7/2>
11544 C7=INT<R7/4>
11545 :CIRCLE X7,Y7,B7,B7,1
11546 :CIRCLE X7,Y7,B7,C7,1
11547 :CIRCLE X7,Y7,C7,B7,1
11548 END PROC
11550 REM ÁBRA34
11551 HIRES 0,7

```

```

11552 X1=20 : Y1=190
11553 A1=210 : B1=170 : C1=10
11554 EXEC TÉGLA
11555 X4=110 : Y4=160
11556 A4=60 : B4=60
11557 EXEC HENGER
11558 X4=110 : Y4=160
11559 A4=40 : B4=60
11560 EXEC HENGER
11561 X6=110 : Y6=100
11562 A6=30 : U6=60 : B6=70
11563 EXEC CSONKAKÚP
11565 X6=110 : Y6=100
11566 A6=20 : U6=50 : B6=70
11567 EXEC CSONKAKÚP
11569 X4=220 : Y4=170
11570 A4=60 : B4=10
11571 EXEC HENGER
11572 X1=200 : Y1=80
11573 A1=30 : B1=90 : C1=60
11574 EXEC TÉGLA
11575 X7=235 : Y7=30
11576 A7=60
11577 EXEC GÖMB
11578 GOTO 11578

```

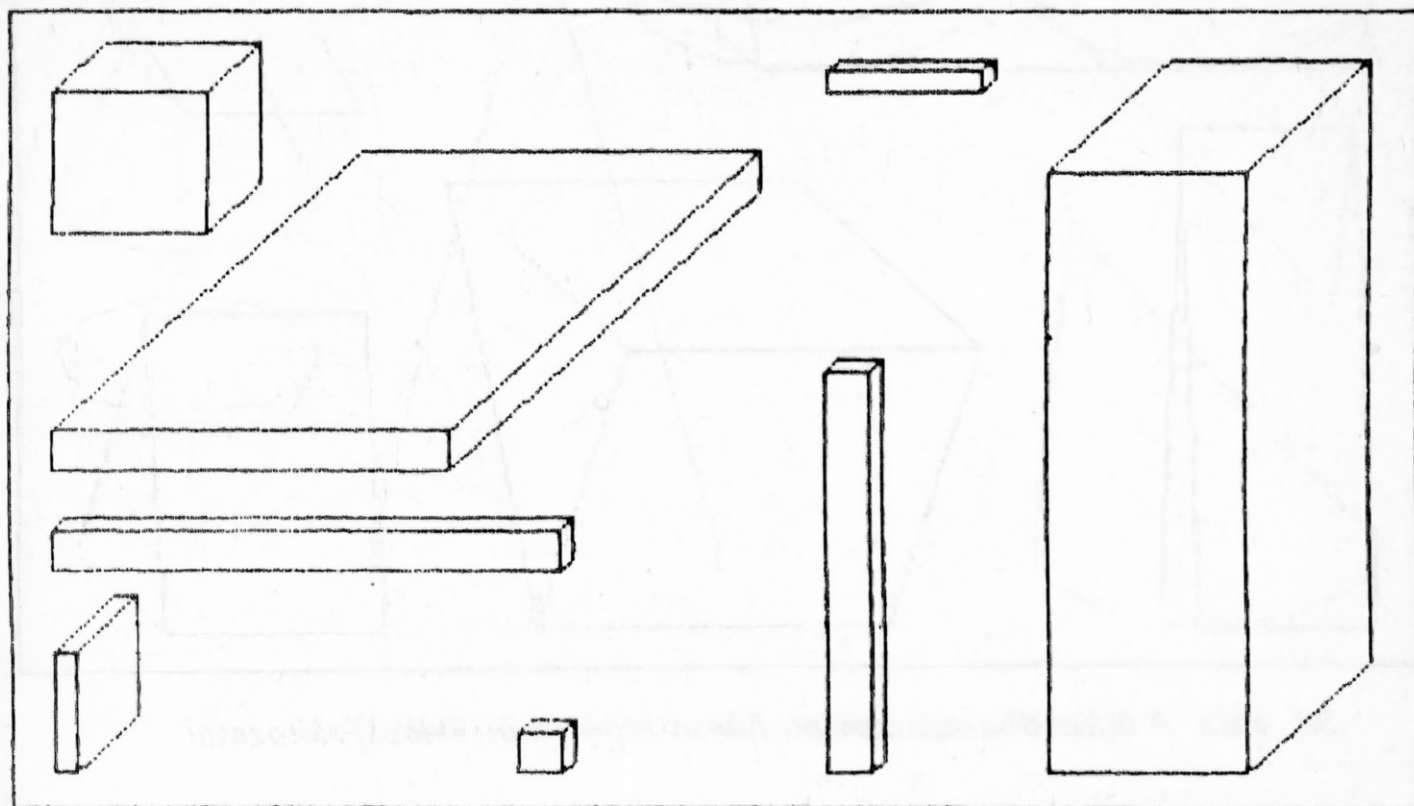
2.4.2. FELÜLETMODELL

Ha a drótvázmodell hátrányát — az esetleges áttekinthetlenséget — meg akarjuk szüntetni, az ábrázoláshoz felületmodellt kell választanunk. A felületmodellnél csak a látható éleket rajzoljuk fel. Ekkor úgy látjuk az egyes tárgyakat, mintha átláthatatlanok lennének, csak a látható felületeket rajzoljuk meg. Ez természetesen nagyobb feladatot jelent. Meg kell értetnünk ugyanis a számítógéppel, hogy mely élek láthatóak, és melyek nem. Egyszerű alakzatoknál a ráfordítás még elviselhető, bonyolultabb testeknél vagy összetett ábráknál a ráfordítás elviselhetetlen mértéket is ölthet. Elviselhetetlen a számítógép számára, mert a tár az információk súlya alatt „összeroppan”. A programozást lényegesen csökkentheti, ha egy automatizmust építünk be, aminek alapján a gép felismeri, hogy mely éleket kell felrajzolni, és melyeket nem. Ez számunkra megkönnyíti, a gép számára viszont megnehezíti a munkát. A professzionális rendszerekbe építenek ilyen automatizmusokat.

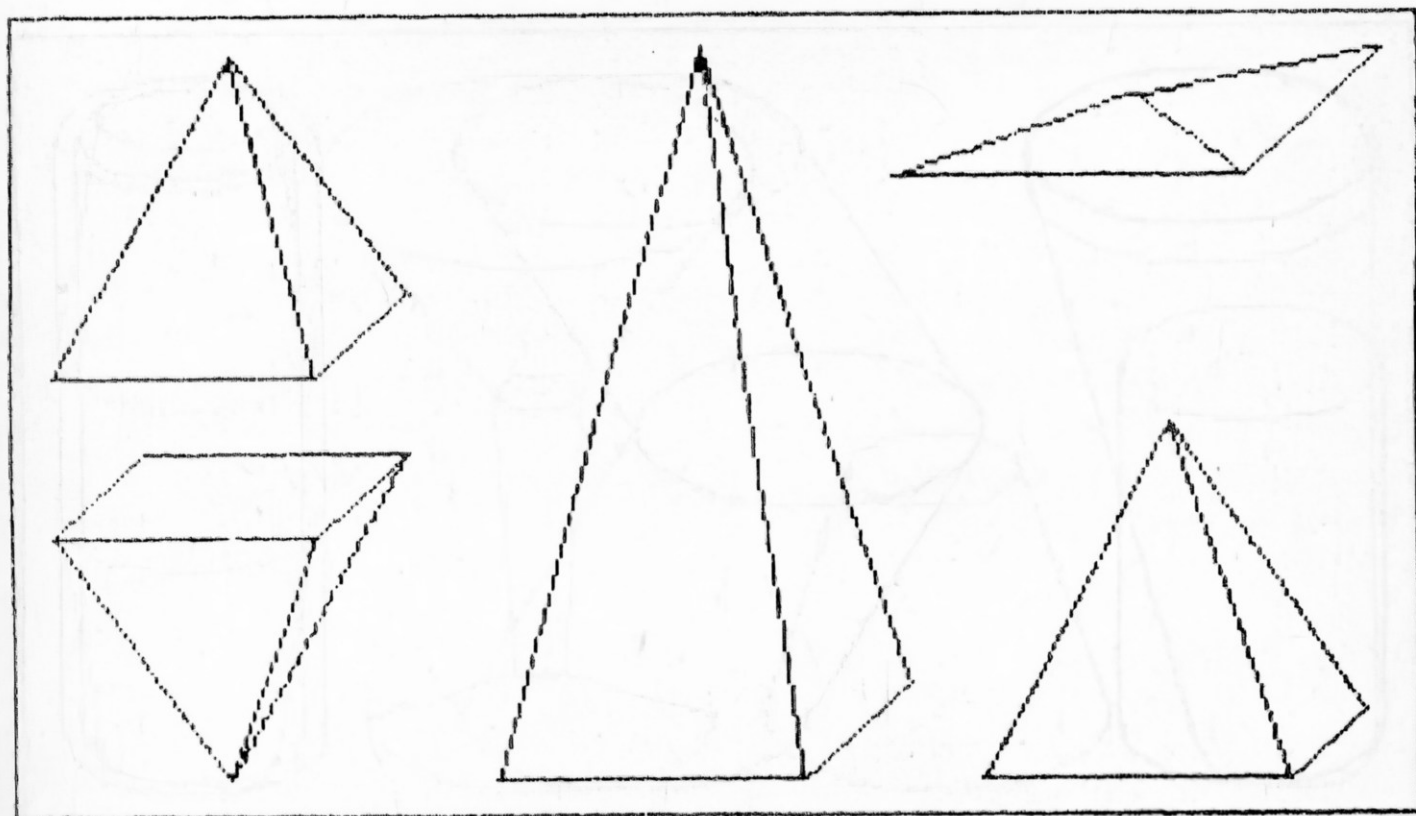
2.4.2.1. Makrók a felületmodellekhez

A drótvázmodellhez létrehozott makrókat alakítjuk át a felületmodell szerinti megjelenítéshez.

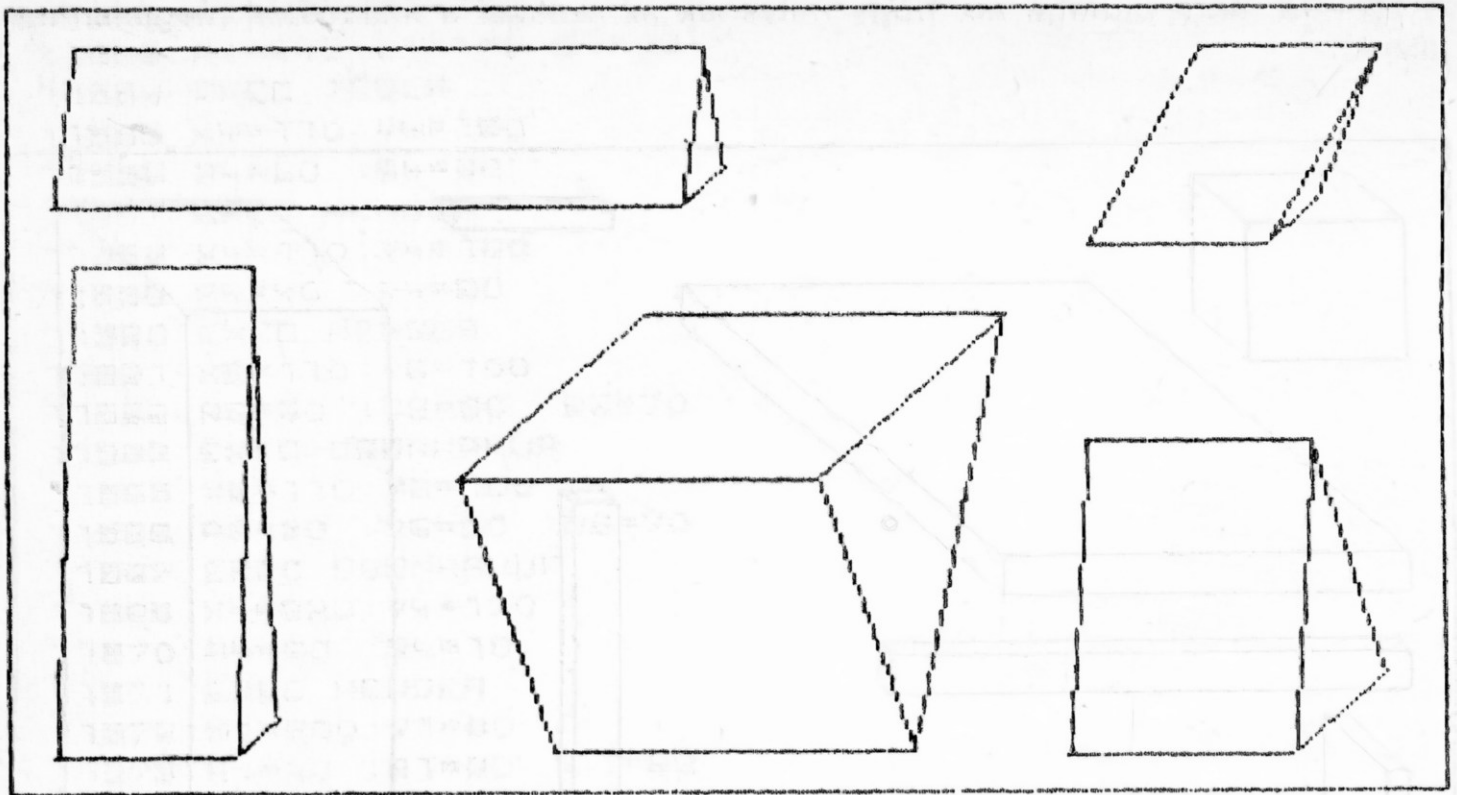
A 36—42. ábra mutatja be, hogy milyenek az ezekkel a makrókkal megjelenített rajzok.



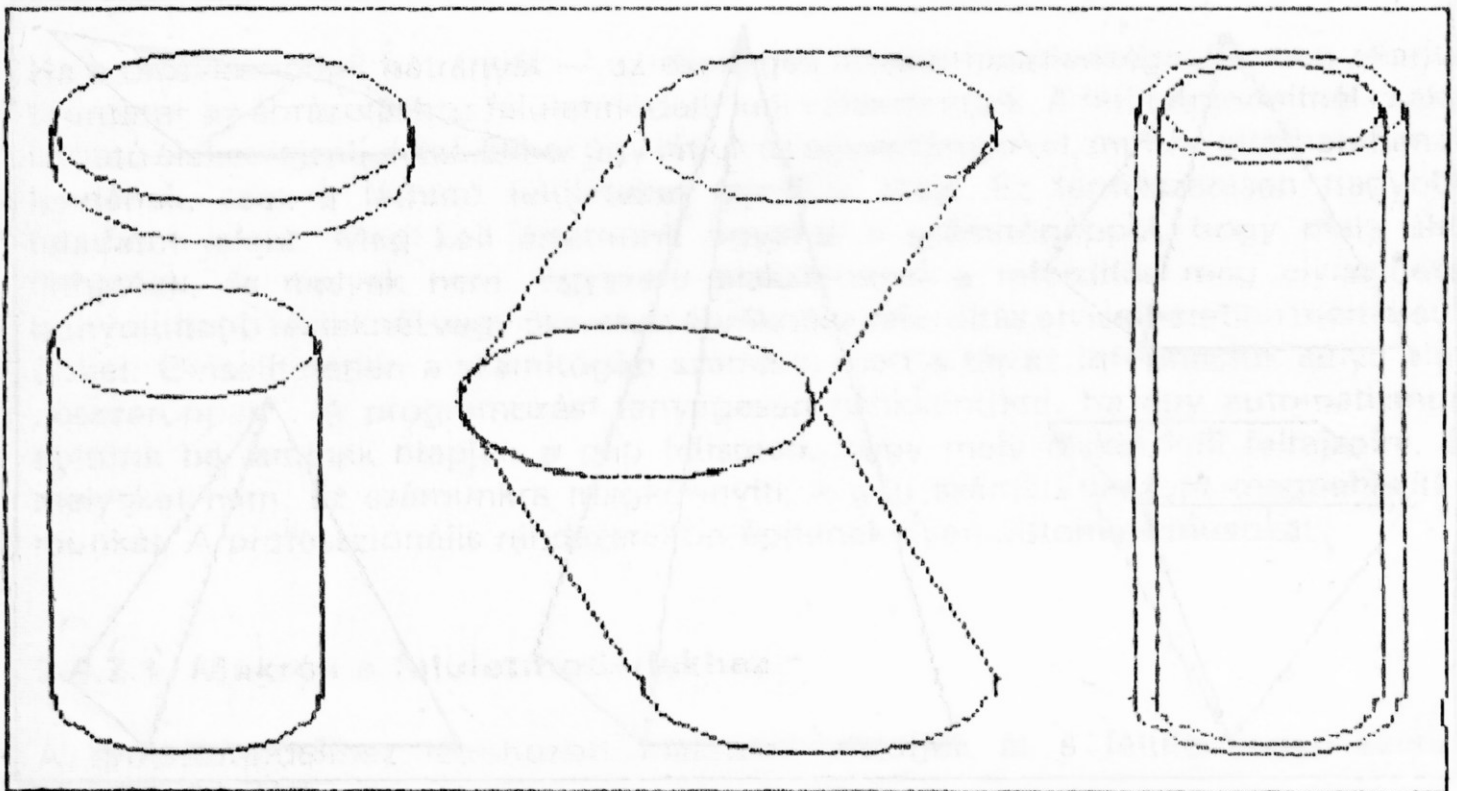
36. ábra. A felületmodell szerinti téglatest változatai



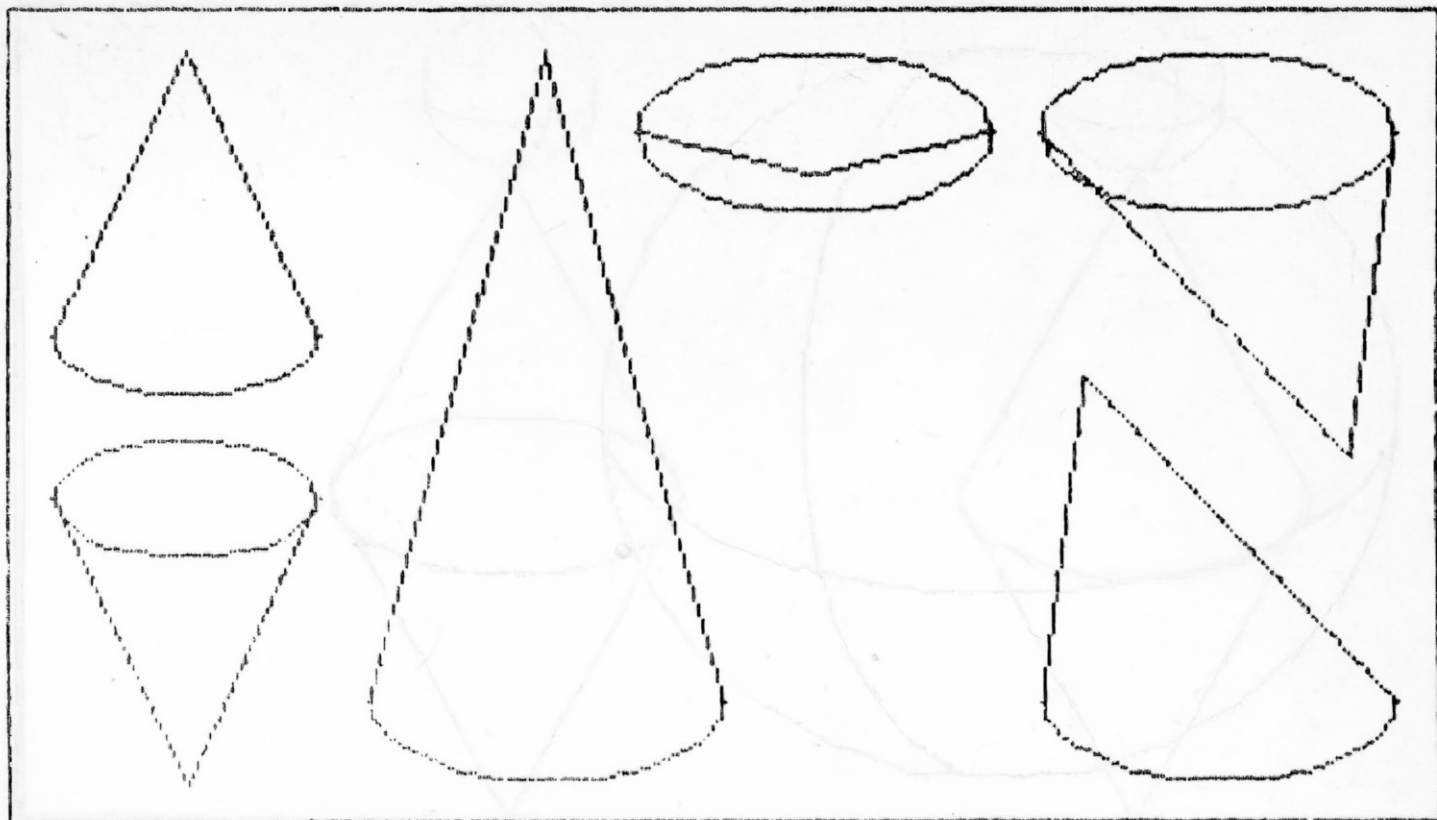
37. ábra. A felületmodell szerinti gúla változatai



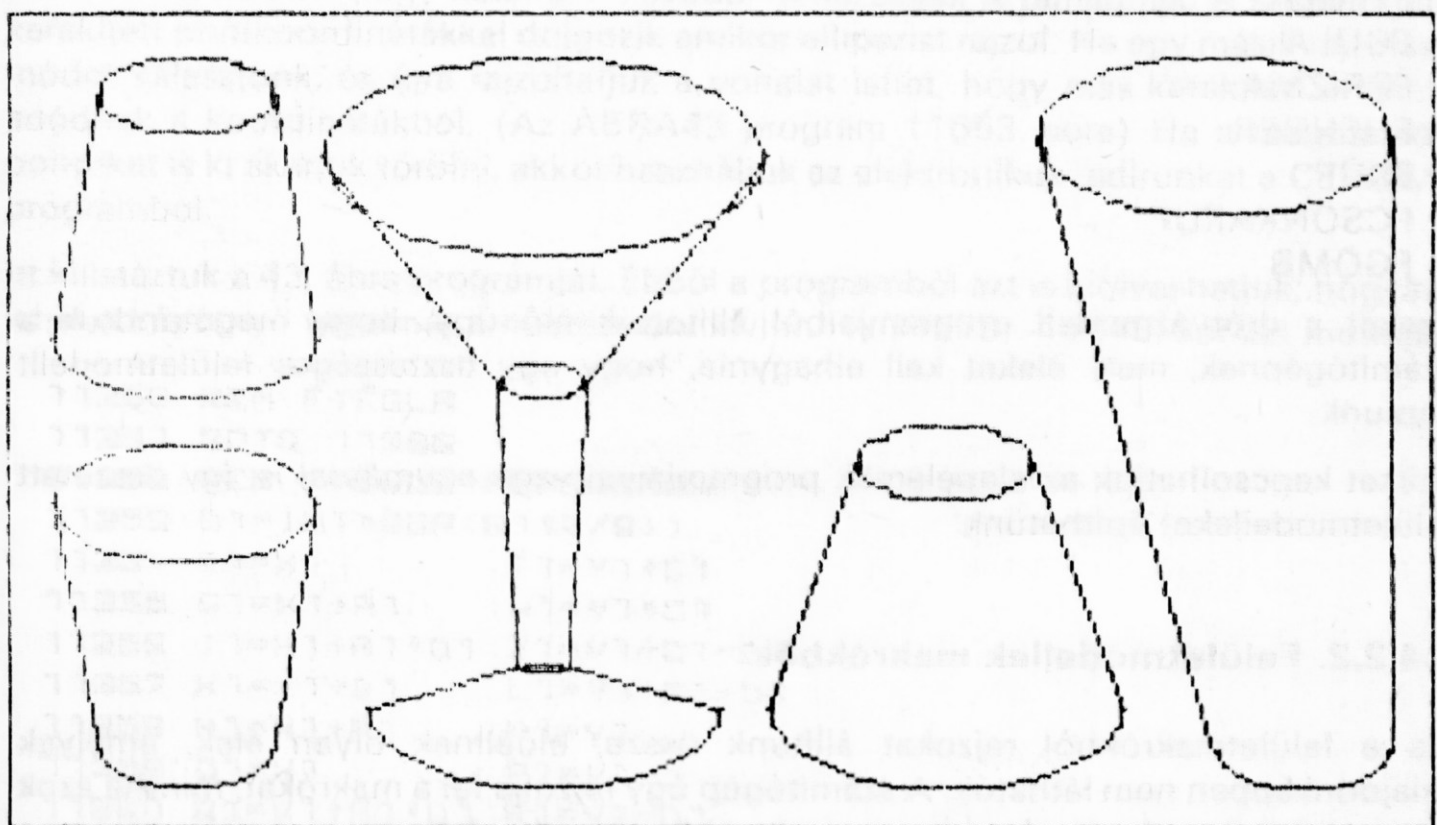
38. ábra. A felületmodell szerinti háromszög alapú hasáb változatai



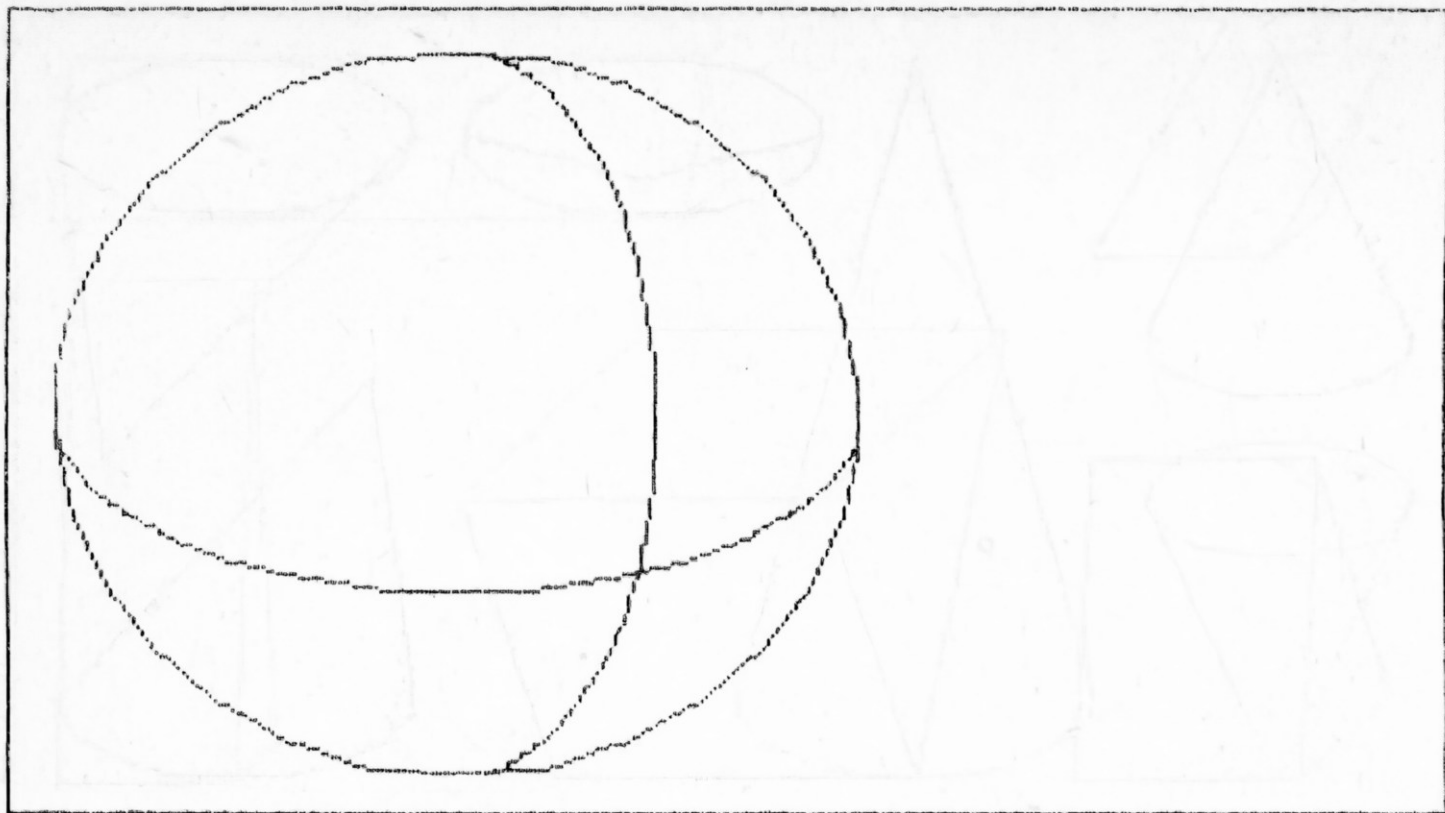
39. ábra. A felületmodell szerinti henger változatai



40. ábra. A felületmodell szerinti kúp változatai



41. ábra. A felületmodell szerinti csonkakúp változatai



42. ábra. A felületmodell szerinti gömb

A programnevek egy F kezdőbetűt kapnak, hogy megkülönböztethessük a drótvázmodell makróitól.

Így a következő programokat kapjuk:

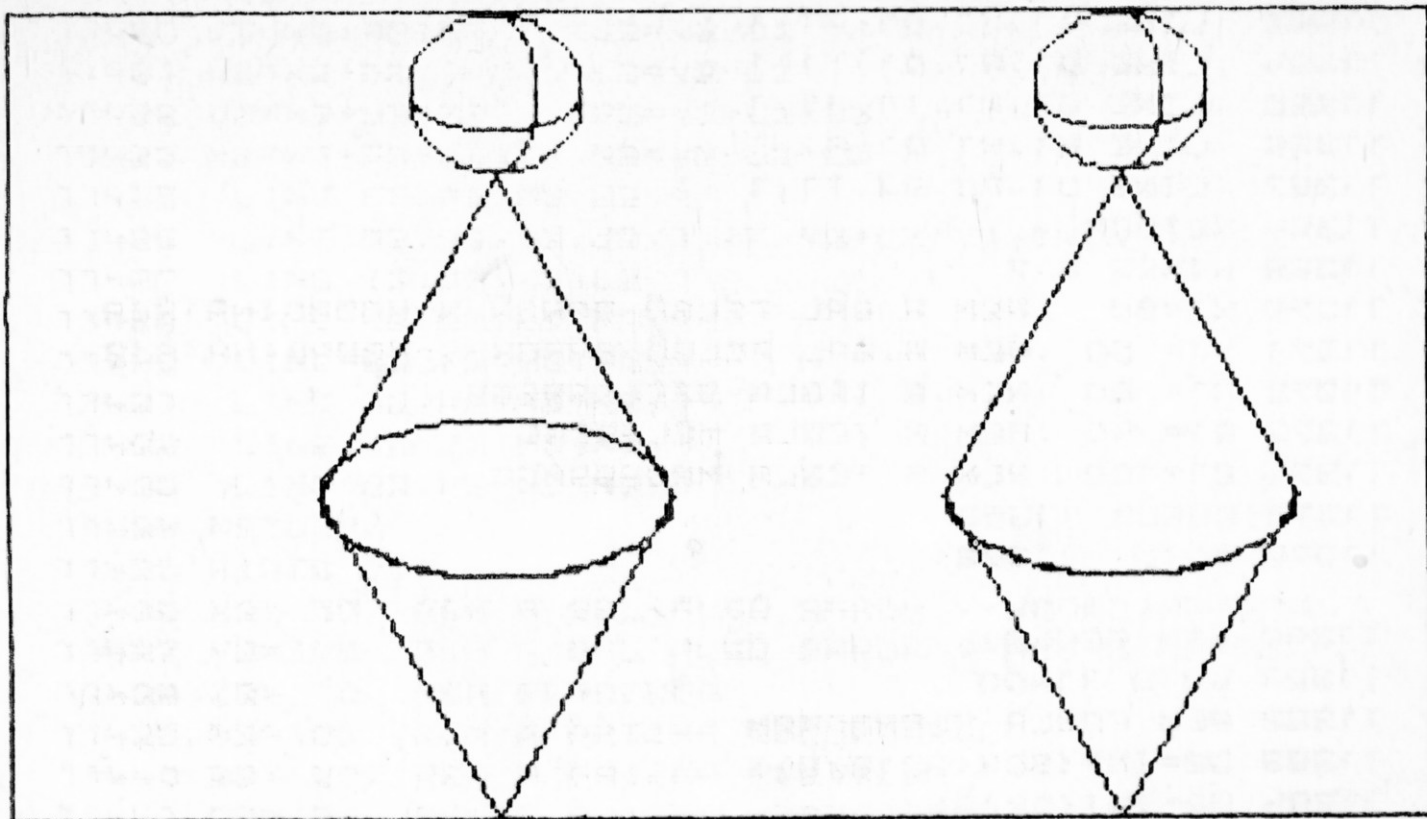
FTÉGLA
FGÜLA
FPRIZMA
FHENGER
FKÚP
FCSONKAKÚP
FGÖMB

Ezeket a drótvázmodell programjaiból állítottuk elő úgy, hogy megmondtuk a számítógépnek, mely éleket kell elhagynia, hogy egy tisztességes felületmodellt kapjunk.

Ezeket kapcsolhatjuk az alapelemek programjaival vagy egymással is, így összetett felületmodelleket építhetünk.

2.4.2.2. Felületmodellek makrókból

Ha a felületmakrókból rajzokat állítunk össze, előállnak olyan élek, amelyek tulajdonképpen nem láthatók. A számítógép úgy rajzolja fel a makrókat, mintha azok különállóak lennének. Azt, hogy mely élek tűnnek el az összeépítés során, a számítógép nem tudja. Az igények és a ráfordítások ördögi köre előről kezdődik.



43. ábra. Egy makrókból összeállított felületmodell rajz Bal oldalt korrigálatlan, jobb oldalt részben korrigáltuk

A 43. ábra bemutat egy összetett felületmodell rajzot. A bal oldali rajzon látható ellipszis fele nem része az ábrának, pedig a különálló makrók esetén látható lenne. A jobb oldali rajzon ezt a hibát korrigáltuk, de csak részben, mert az ellipszis néhány pontja megmaradt. Hogy lehet ez? További nehézségek is támadnak. A számítógép kerekített pontkoordinátákkal dolgozik amikor ellipszist rajzol. Ha egy másik rajzolósi módot választunk, és újra rajzoltatjuk a vonalat lehet, hogy más kerekített értékek adódnak a koordinátákból. (Az ÁBRA43 program 11563. sora) Ha a megmaradt pontokat is ki akarjuk törölni, akkor használjuk az elektronikus radírunkat a CERUZA programból.

Itt kilistáztuk a 43. ábra programját. Ebből a programból azt is kiolvashatjuk, hogyan lehet a képernyő tartalmát a nyomtatóra kivinni. A 11570—11578. sor ezt mutatja.

```

11350 REM FTÉGLA
11351 GOTO 11369
11352 REM FTÉGLA ALPROGRAM
11353 D1=INT(SQR(B1↑2/B))
11354 E1=X1          :F1=Y1+C1
11355 G1=X1+A1      :H1=Y1+C1
11356 I1=X1+A1+D1  :J1=Y1+C1-D1
11357 K1=X1+D1     :L1=Y1+C1-D1
11358 M1=X1+A1     :N1=Y1
11359 O1=X1        :P1=Y1
11360 Q1=X1+A1+D1  :R1=Y1-D1
11361 S1=X1+D1     :T1=Y1-D1
11362 :REC  X1,Y1,A1,C1,1

```

```

11363 :LINE I1,J1,Q1,R1,1
11364 :LINE Q1,R1,S1,T1,1
11365 :LINE G1,H1,I1,J1,1
11366 :LINE M1,N1,Q1,R1,1
11367 :LINE O1,P1,S1,T1,1
11368 RETURN
11369 HIRES 0,7
11370 X1=50 :REM A BAL FELSŐ SAROK X-KOORDINÁTÁJA
11371 Y1= 50 :REM A BAL FELSŐ SAROK Y-KOORDINÁTÁJA
11372 A1= 80 :REM A TÉGLA SZÉLESSÉGE
11373 B1= 40 :REM A TÉGLA MÉLYSÉGE
11374 C1=100 :REM A TÉGLA MAGASSÁGA
11375 GOSUB 11352
11376 GOTO 11376

```

```

11380 REM FGÚLA
11381 GOTO 11400
11382 REM FGÚLA ALPROGRAM
11383 D2=INT(SQR(B2↑2/B))
11384 U2=INT(D2/2)
11385 V2=INT(A2/2)
11386 E2=X2 : F2=Y2
11387 G2=X2+A2 : H2=Y2
11388 I2=X2+A2+D2 : J2=Y2-D2
11389 K2=X2+D2 : L2=Y2-D2
11390 M2=X2+V2+U2 : N2=Y2-U2-C2
11391 :LINE E2,F2,G2,H2,1
11392 :LINE G2,H2,I2,J2,1:IF Y2+C2>Y2 THEN 11395
11393 :LINE I2,J2,K2,L2,1
11394 :LINE K2,L2,E2,F2,1
11395 :LINE E2,F2,M2,N2,1
11396 :LINE G2,H2,M2,N2,1
11397 :LINE I2,J2,M2,N2,1
11399 RETURN

```

```

11400 HIRES 0,7
11401 X2=100 :REM A BAL ALSÓ SAROK X-KOORDINÁTÁJA
11402 Y2=100 :REM A BAL ALSÓ SAROK Y-KOORDINÁTÁJA
11404 A2= 50 :REM A GÚLA SZÉLESSÉGE
11405 B2= 60 :REM A GÚLA MÉLYSÉGE
11406 C2= 80 :REM A GÚLA MAGASSÁGA
11407 GOSUB 11382
11408 GOTO 11408

```

```

11413 REM FPRIZMA
11414 GOTO 11435
11415 REM FPRIZMA ALPROGRAM
11416 D3=INT(SQR(B3↑2/B))
11417 U3=INT(D3/2)
11418 E3=X3 : F3=Y3
11419 G3=X3+A3 : H3=Y3

```



```

11420 I3=X3+A3+D3      :J3=Y3-D3
11421 K3=X3+D3         :L3=Y3-D3
11422 O3=X3+U3+Z3     :P3=Y3-U3-C3
11423 M3=X3+A3+U3+Z3 :N3=Y3-U3-C3
11425 :LINE E3,F3,G3,H3,1
11426 :LINE G3,H3,I3,J3,1:IF Y3+C3>Y3 THEN 11429
11427 :LINE I3,J3,K3,L3,1
11428 :LINE K3,L3,E3,F3,1
11429 :LINE E3,F3,O3,P3,1
11431 :LINE G3,H3,M3,N3,1
11432 :LINE I3,J3,M3,N3,1
11433 :LINE O3,P3,M3,N3,1
11434 RETURN
11435 HIRES 0,7
11436 X3= 50 :REM A BAL ALSÓ SAROK X-KOORDINÁTÁJA
11437 Y3=100 :REM A BAL ALSÓ SAROK Y-KOORDINÁTÁJA
11438 Z3= 0 :REM FERDESÉG
11439 A3=100 :REM A PRIZMA SZÉLESSÉGE
11440 B3= 60 :REM A PRIZMA MÉLYSÉGE
11441 C3= 50 :REM A PRIZMA MAGASSÁGA
11442 GOSUB 11415
11443 GOTO 11443

```

```

11450 REM FHENGER
11451 GOTO 11470
11452 REM FHENGER ALPROGRAM
11453 R4=INT(A4/2)
11454 S4=INT(R4/2)
11455 E4=X4-R4      :F4=Y4
11456 G4=X4+R4     :H4=Y4
11457 I4=X4+R4+Z4 :J4=Y4-B4
11458 K4=X4-R4+Z4 :L4=Y4-B4
11459 O4=X4       :P4=Y4
11460 M4=X4+Z4    :N4=Y4-B4
11461 IF Y4+B4>Y4 THEN 11464
11462 :CIRCLE O4,P4,R4,S4,1
11463 :ARC      M4,N4,90,270,4,R4,S4,1:GOTO 11466
11464 :ARC      O4,P4,90,270,4,R4,S4,1
11465 :CIRCLE M4,N4,R4,S4,1
11466 :LINE     E4,F4,K4,L4,1
11467 :LINE     G4,H4,I4,J4,1
11468 RETURN
11470 HIRES 0,7
11471 X4=150 :REM AZ ALAPKÖR KÖZÉPPONTJÁNAK X-KOORD.
11472 Y4=155 :REM AZ ALAPKÖR KÖZÉPPONTJÁNAK Y-KOORD.
11473 Z4= 0 :REM FERDESÉG
11474 A4=160 :REM A HENGER ÁTMÉRŐJE
11475 B4=105 :REM A HENGER MAGASSÁGA
11476 GOSUB 11452
11477 GOTO 11477

```

```

11480 REM FKÚP
11481 GOTO 11493
11482 REM FKÚP ALPROGRAM
11483 SS=INT(R5/2)
11484 ES=X5-R5:FS=Y5
11485 GS=X5+R5:H5=Y5
11486 IS=X5+Z5:JS=Y5-B5
11487 K5=X5 :LS=Y5 :IF Y5+B5>Y5 THEN 11489
11488 :CIRCLE K5,LS,R5,SS,1:GOTO 11490
11489 :ARC K5,LS,90,270,4,R5,SS,1
11490 :LINE GS,H5,IS,JS,1
11491 :LINE ES,FS,IS,JS,1
11492 RETURN
11493 HIRES 0,7
11494 X5=100 :REM AZ ALAPKÖR KÖZÉPPONTJÁNAK X-KOORD.
11495 Y5=100 :REM AZ ALAPKÖR KÖZÉPPONTJÁNAK Y-KOORD.
11496 Z5= 0 :REM FERDESÉG
11497 R5= 50 :REM AZ ALAPKÖR SUGARA
11498 B5= 80 :REM A KÚP MAGASSÁGA
11499 GOSUB 11482
11500 GOTO 11500

11508 REM FCSONKAKÚP
11509 GOTO 11528
11510 REM FCSONKAKÚP ALPROGRAM
11511 SB=INT(R6/2)
11512 WB=INT(U6/2)
11513 EB=X6-R6 :FB=Y6
11514 GB=X6+R6 :HB=Y6
11515 IB=X6+V6+Z6:JB=Y6-B6
11516 KB=X6-V6+Z6:LB=Y6-B6
11517 OB=X6+Z6 :PB=Y6-B6
11518 M6=X6 :N6=Y6
11519 IF Y6+B6>Y6 THEN 11523
11520 :CIRCLE M6,N6,R6,SB,1
11521 :ARC OB,PB,90,270,4,U6,WB,1
11522 GOTO 11525
11523 :ARC M6,N6,90,270,4,R6,SB,1
11524 :CIRCLE OB,PB,U6,WB,1
11525 :LINE EB,FB,KB,LB,1
11526 :LINE GB,HB,IB,JB,1
11527 RETURN
11528 HIRES 0,7
11529 X6=100 :REM AZ ALAPKÖR KÖZÉPPONTJÁNAK X-KOORD.
11530 Y6=100 :REM AZ ALAPKÖR KÖZÉPPONTJÁNAK Y-KOORD.
11531 Z6= 0 :REM FERDESÉG
11532 R6= 50 :REM AZ ALAPKÖR SUGARA
11533 V6= 20 :REM A FELSŐ KÖR SUGARA
11534 B6= 80 :REM A CSONKAKÚP MAGASSÁGA
11535 GOSUB 11510
11536 GOTO 11536

```

```

11540 REM FGÖMB
11541 GOTO 11549
11542 REM FGÖMB ALPROGRAM
11543 B7=INT(A7/2)
11544 C7=INT(A7/4)
11545 :CIRCLE X7,Y7,B7,B7,1
11546 :ARC X7,Y7,90,270,4,B7,C7,1
11547 :ARC X7,Y7, 0,180,4,C7,B7,1
11548 RETURN
11549 HIRES 0,7
11550 X7=100 :REM A GÖMB KÖZÉPPONTJÁNAK X-KOORDINÁTÁJA
11551 Y7=100 :REM A GÖMB KÖZÉPPONTJÁNAK Y-KOORDINÁTÁJA
11552 A7=100 :REM A GÖMB ÁTMÉRŐJE
11553 GOSUB 11542
11554 GOTO 11554

```

```

11480 REM 43.ÁBRA
11481 GOTO 11550
11482 PROC FKÚP
11483 S5=INT(R5/2)
11484 E5=X5-R5:F5=Y5
11485 G5=X5+R5:H5=Y5
11486 I5=X5+Z5:J5=Y5-B5
11487 K5=X5 :L5=Y5 :IF Y5+B5>Y5 THEN 11489
11488 :CIRCLE K5,L5,R5,S5,1:GOTO 11490
11489 :ARC K5,L5,90,270,4,R5,S5,1
11490 :LINE G5,H5,I5,J5,1:LINE E5,F5,I5,J5,1
11491 END PROC

```

```

11542 PROC FGÖMB
11543 B7=INT(A7/2)
11544 C7=INT(A7/4)
11545 :CIRCLE X7,Y7,B7,B7,1
11546 :ARC X7,Y7,90,270,4,B7,C7,1
11547 :ARC X7,Y7, 0,180,4,C7,B7,1

```

```

11548 END PROC
11550 REM ÁBRA43
11551 HIRES 0,7
11552 X5=110:Y5=120
11553 R5= 40:B5= 80
11554 EXEC FKÚP
11555 X5=110:Y5=120
11556 R5= 40:B5=-80
11557 EXEC FKÚP
11558 X7=110:Y7= 20:A7= 40
11559 EXEC FGÖMB
11560 X5=250:Y5=120
11561 R5= 40:B5=-80
11562 EXEC FKÚP
11563 :ARC X5,Y5,270,90,1,R5,S5,0
11564 X5=250:Y5=120

```

```

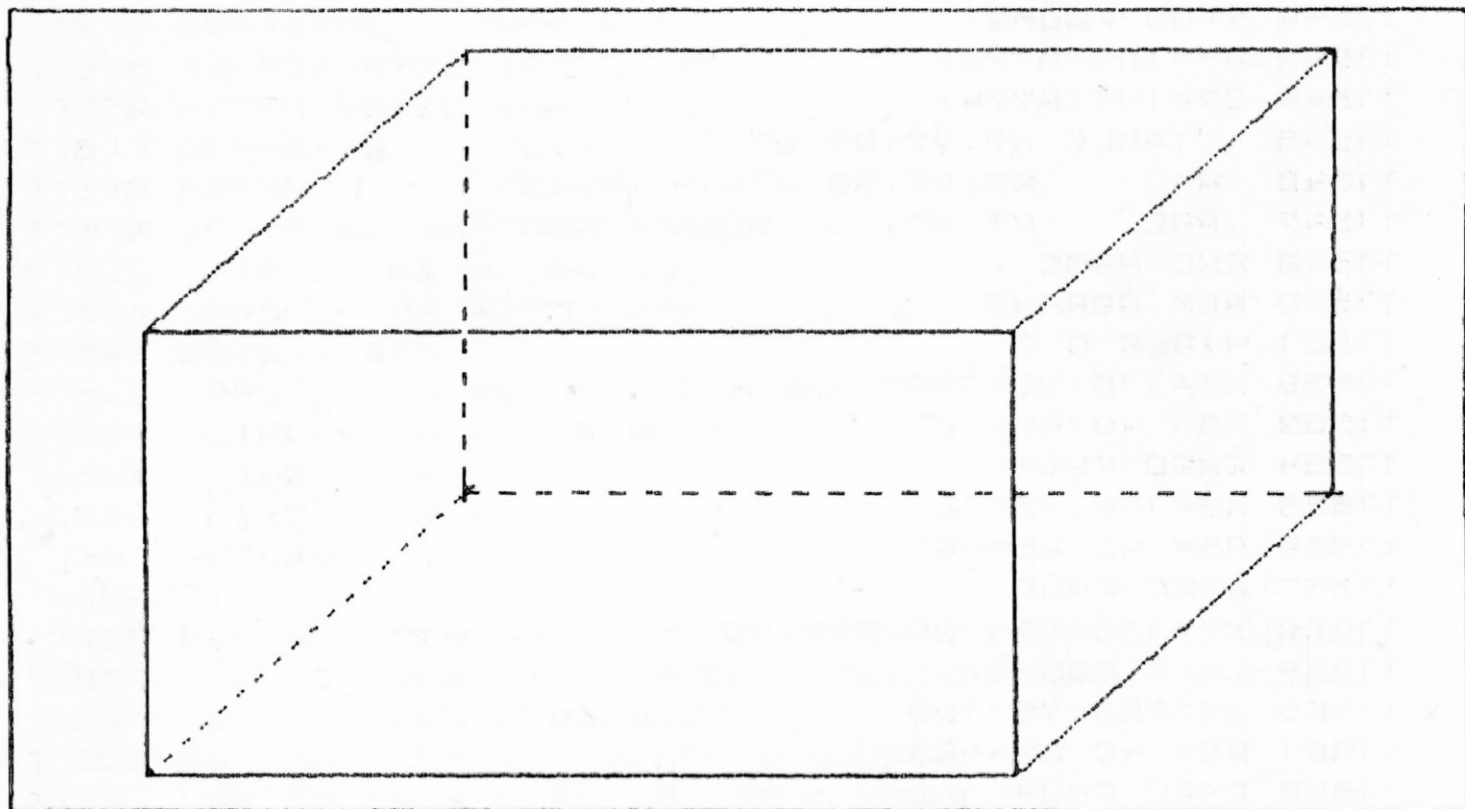
11565 R5= 40:R5= 80
11566 EXEC FKUP
11567 X7=250:Y7= 20:A7= 40
11568 EXEC FGOMB
11570 OPEN 1,4
11571 COPY
11572 OPEN 1,4
11573 PRINT#1,"43.ABRA: MAKRÓKBÓL ÖSSZEÁLLÍTOTT"
11574 PRINT#1,"                                FELÜLETMODELL"
11575 PRINT#1,"                                A BALOLDALI KORRIGÁLATLAN,"
11576 PRINT#1,"                                A JOBBOLDALI RÉSZBEN KORRIGÁLT"
11577 CLOSE 1
11578 GOTO 11578

```

2.4.3. TESTMODELL

A testmodell esetében a nem látható éleket szaggatott vonallal ábrázoljuk, tehát a ráfordítás még több, mint a felületmodellnél. Egy ilyen 3D modellnek az előállítása annyi munkával jár, hogy még a nagy cégek is meggondolják, hogy megengedhetik-e maguknak. Sokszor csak 2 1/2D modelleket készítenek. Ezek olyan modellek, mintha 3D modellek lennének, de mivel kisebb ráfordítással készülnek, csak szűkebb határok között működhetnek.

A teljesség kedvéért meg kell említeni, hogy ilyen professzionális rendszereknek a szokásos 3 vetületben való ábrázolási módot is automatikusan kell előállítania. Ennek



44. ábra. Makrókból és alapprogramból összeállított korrigálatlan testmodell

a hardverét és szoftverét pedig már csak jó nevű cégek biztosítják, akik teljességre törekednek. Mi nem törekszünk erre, vagy ezen törekvésünket gyorsan háttérbe szorítjuk. Testmodellünket kézzel barkácsoljuk össze.

Erre a 44. ábra mutat be példát.

A programot TESTMODELL-nek nevezzük. Láthatjuk, hogy egy felületmakrót és egy alapprogramot, az ún. szaggatott vonal programot kombináltuk.

A 44. ábra azt mutatja, hogy megint kisebb javításokra van szükség, amit a CERUZA programmal végezhetünk el. Mivel a program a ferde éleknél egész számokkal számol, némi eltérések lehetnek a makróban kiszámított és az alapprogramban használt pontkoordináták között.

```
70100 REM TESTMODELL
70101 GOTO 71370
70102 PROC SZAGGATOTT VONAL
70103 CE=ABS<CC-CA>
70104 CF=ABS<CB-CD>
70105 IF CA=CC THEN 70137
70106 IF CB=CD THEN 7015B
70107 CG=INT<SQR<<16*CE↑2>/<CF↑2+CE↑2>>>
70108 CH=INT<<CG*CF>/CE>
70109 CI=INT<CE/CG>
70110 CJ=-1
70111 IF CA<CC AND CD<CB THEN 70115
70112 IF CA<CC AND CB<CD THEN 70117
70113 IF CC<CA AND CD<CB THEN 70120
70114 IF CC<CA AND CB<CD THEN 70123
70115 CG=-CG
70116 GOTO 70125
70117 CG=-CG:CH=-CH
70118 GOTO 70125
70120 CG= CG:CH= CH
70121 GOTO 70125
70123 CH=-CH
70124 GOTO 70125
70125 CK=CA+CG:CL=CB+CH
70127 REM HIRES
70128 FOR CM=1 TO CI
70129 GOSUB 70178
70130 CK=CK-CG
70131 CL=CL-CH
70132 CN=CK-<2*CG>
70133 CO=CL-<2*CH>
70134 :LINE CK,CL,CN,CO,CP
70135 NEXT CM
70136 GOTO 70186
70137 REM FÜGGŐLEGES
70138 CI=INT<CF/4>
```

```

10139 CJ=-1
10140 CH= 4
10141 IF CB<CD THEN 10143
10142 IF CD<CB THEN 10145
10143 CH=-CH
10144 GOTO 10147
10145 CH= CH
10146 GOTO 10147
10147 CL=CB+CH
10149 FOR CM=1 TO CI
10150 GOSUB 10178
10151 CK=CA
10152 CL=CL-CH
10153 CN=CA
10154 CO=CL-(2*CH)
10155 :LINE CK,CL,CN,CO,CP
10156 NEXT CM
10157 GOTO 10186
10158 REM VIZSZINTES
10159 CI=INT(CE/4)
10160 CJ=-1
10161 CG= 4
10162 IF CA<CC THEN 10164
10163 IF CC<CA THEN 10166
10164 CG=-CG
10165 GOTO 10167
10166 CG= CG
10167 REM HIRES
10168 CK=CA+CG
10169 FOR CM=1 TO CI
10170 GOSUB 10178
10171 CK=CK-CG
10172 CL=CB
10173 CN=CK-(2*CG)
10174 CO=CB
10175 :LINE CK,CL,CN,CO,CP
10176 NEXT CM
10177 GOTO 10186
10178 REM CIKLUS
10179 CJ=CJ*(-1)
10180 IF CJ= 1 THEN 10182
10181 IF CJ=-1 THEN 10184
10182 CP=1
10183 GOTO 10185
10184 CP=0
10185 RETURN
10186 END PROC
11352 PROC FTÉGLA
11353 D1=INT(SQR(B1+2/B))
11354 E1=X1 : F1=Y1+C1

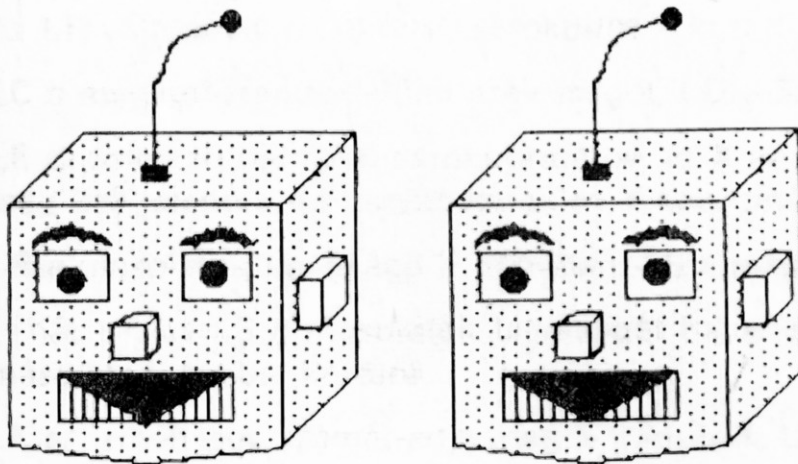
```

```

11355 G1=X1+A1      :H1=Y1+C1
11356 I1=X1+A1+D1 :J1=Y1+C1-D1
11357 K1=X1+D1     :L1=Y1+C1-D1
11358 M1=X1+A1     :N1=Y1
11359 O1=X1        :P1=Y1
11360 Q1=X1+A1+D1 :R1=Y1-D1
11361 S1=X1+D1     :T1=Y1-D1
11362 :REC  X1,Y1,A1,C1,1
11363 :LINE I1,J1,Q1,R1,1
11364 :LINE Q1,R1,S1,T1,1
11365 :LINE G1,H1,I1,J1,1
11366 :LINE M1,N1,O1,P1,1
11367 :LINE O1,P1,S1,T1,1
11368 END PROC
11370 REM ÁBRA44
11371 HIRES 0,7
11372 X1= 30:Y1=80
11373 A1=190:B1=200:C1=110
11374 EXEC FTÉGLA
11375 CA= 30:CB=190:CC=100:CD=120
11376 EXEC SZAGGATOTT VONAL
11377 CA=100:CB= 10:CC=100:CD=120
11378 EXEC SZAGGATOTT VONAL
11379 CA=100:CB=120:CC=290:CD=120
11380 EXEC SZAGGATOTT VONAL
11381 GOTO 11381

```

2.4.4. ÖSSZEFOGLALÁS



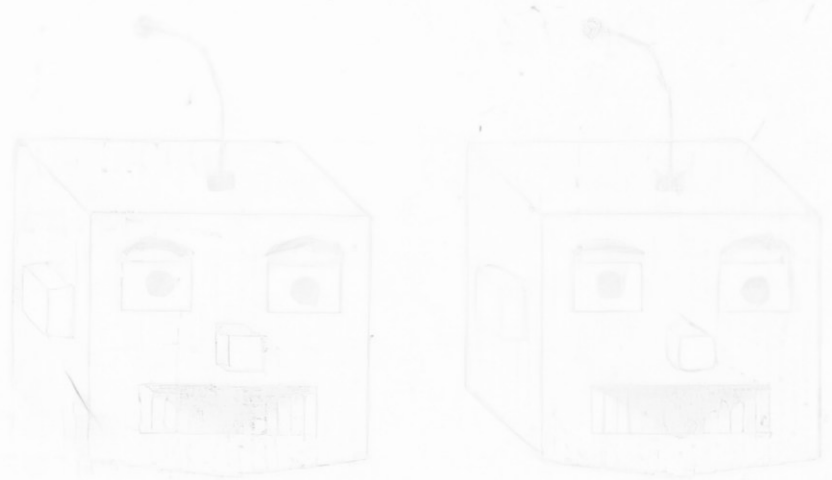
A makrók alapelemekből álló építőelemek, amelyekkel összetett rajzokat lehet készíteni.

A makrókkal drótvázmodell, felületmodell vagy testmodell készíthetünk.

A drótvázmodellnél minden él látható, ezért a rajz áttekinthetetlenné válhat. A drótvázmodelleket könnyen kombinálhatjuk anélkül, hogy később javításokra lenne szükség.

A felületmodellek csak a testek felületét ábrázolják, a láthatatlan éleket nem jelenítik meg. Az ábrázolás áttekinthetőbb, de a ráfordítás növekszik, továbbá, ha nincsen automatizmus beépítve, korrekcióra kell.

A testmodellnél a nem látható éleket szaggatott vonallal ábrázolják. A ráfordítás tovább nő. A nagy cégek is meggondolják, hogy készítsenek-e valódi testmodellt ábrázoló rendszereket.



2.5. Mit tehetünk még a rajzokkal?

Az eddig tanultakkal már nagyon sok mindent elérhetünk. Ezzel azonban még nem elégszünk meg, hanem bizonyos mértékig szeretnénk az CAD magaskolájába^o bepillantani. Ehhez a tanultakat fogjuk meghatározott módon alkalmazni, részben pedig újabb építőelemekkel ismerkedünk meg. Mindig csak az éppen szóban forgó elveket magyarázzuk, és mutatjuk meg. Az ebben rejlő lehetőségeket mindenkinek magának kell kibányásznia.

2.5.1. KETTŐZÉS

Sokszor kívánatos a rajz egy részletét ugyanazon a rajzon belül, de egy másik helyen megismételni vagyis megkettőzni. Egy olyan részt egyszerű megismételni, aminek a koordinátái ismertek. Csak a kiindulási koordinátát kell megváltoztatni. Most erről nem kívánunk beszélni. Képréseket szeretnénk megkettőzni, amin belül tetszőleges, esetleg véletlenül előállt pontok is vannak. A MÁSOLÓ program megsokszorozza ezeket a képterületeket, mégpedig X és Y irányban is annyiszor, ahányszor akarjuk, és olyan távolságokra, ahogy akarjuk. Arra viszont ügyelnünk kell, hogy a megismélt ábrákkal ne jussunk a képernyő területén kívülre.

A megkettőzendő felület bal felső sarokpontjának X koordinátáját az LA változóban helyezzük el.

Az LB változóval a bal felső sarokpont Y koordinátáját határozzuk meg.

LC a megkettőzendő felület szélességét, LD a felület magasságát határozza meg.

LE azon mező bal felső sarokpontjának az X koordinátája, ahol az első másolatnak meg kell jelennie, LF ugyanennek az Y koordinátája.

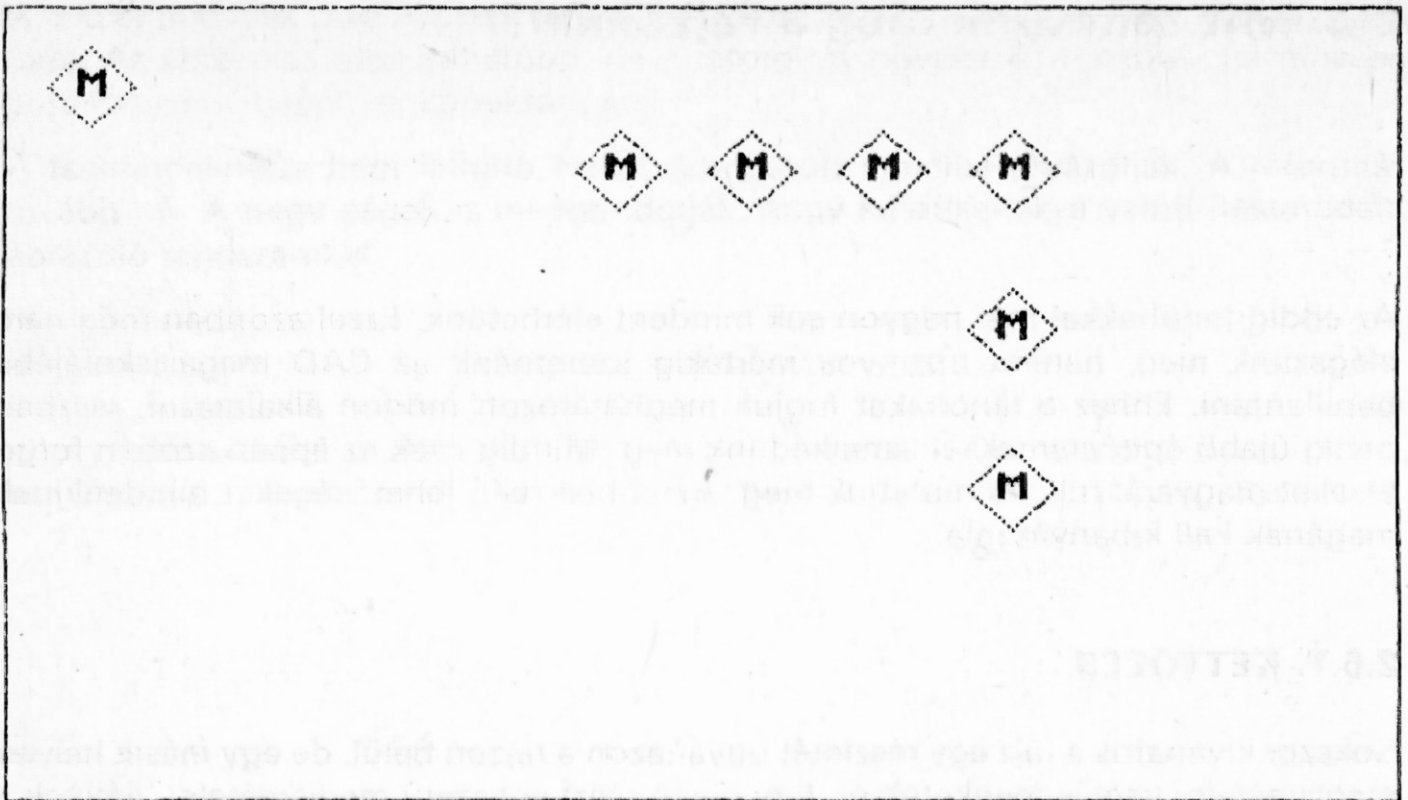
LR a másolatok távolsága X irányban, LS a másolatok távolsága Y irányban.

Tehát LR és LS a másolatok távolságát határozza meg abban az esetben, ha egy másolatnál többet kértünk.

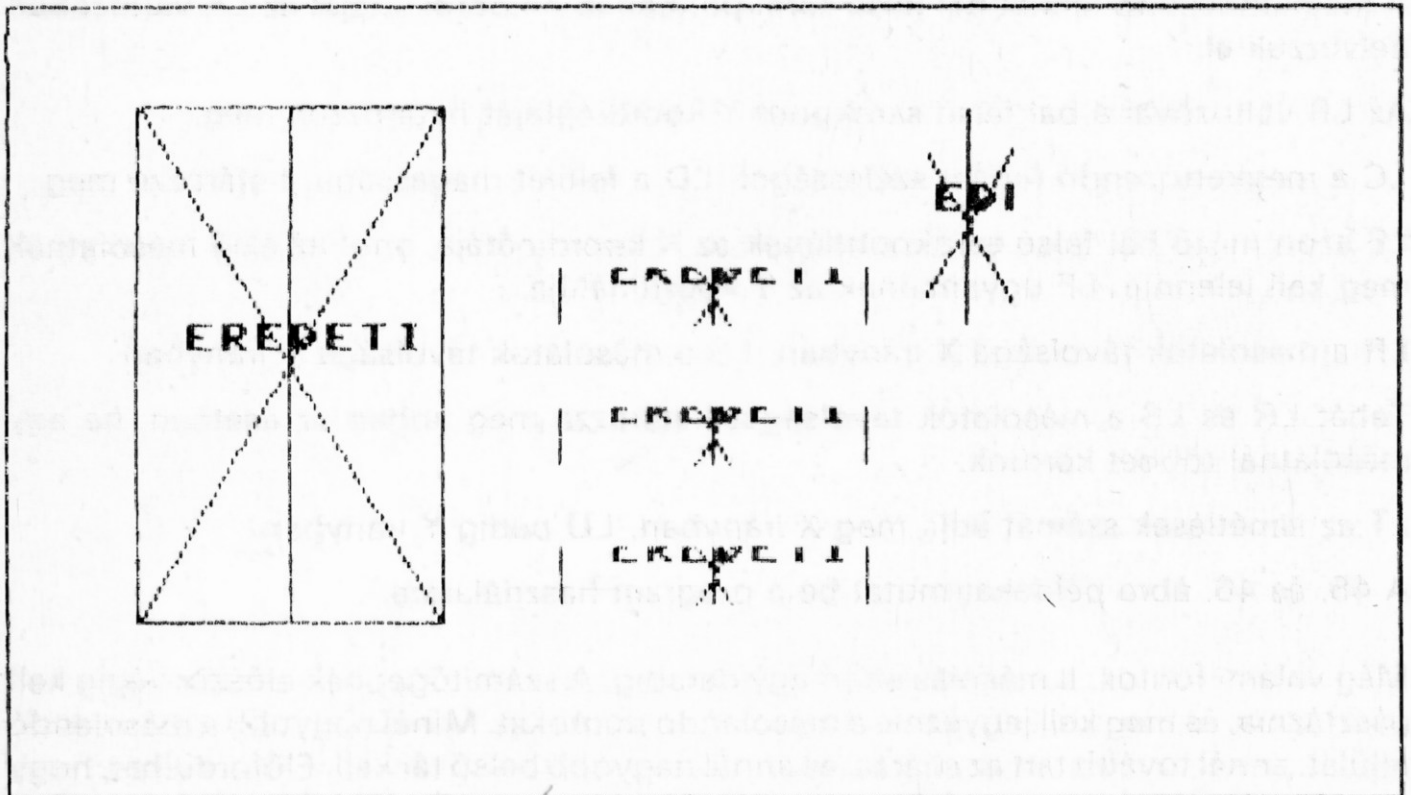
LT az ismétlések számát adja meg X irányban, LU pedig Y irányban.

A 45. és 46. ábra példákat mutat be a program használatára.

Még valami fontos: a másolás eltart egy darabig. A számítógépnek először végig kell pásztáznia, és meg kell jegyeznie a másolandó pontokat. Minél nagyobb a másolandó felület, annál tovább tart az eljárás, és annál nagyobb belső tár kell. Előfordulhat, hogy a belső tár már nem is elég. Ekkor vagy csökkenteni kell a másolandó felületet, vagy le kell tennünk a dinamikus tömbdeklarációról (12004. sor). A számítógép csak a



45. ábra. Képrész kettőzése, sokszorozása. Az eredeti a bal felső



46. ábra. Egy rajz tetszőleges részének megkettőzése

kigyújtott pontokat jegyzi meg. A dinamikus tömbdeklaráció esetén annyi pontnak tart fenn helyet, amennyi a felületen elhelyezkedik, attól függetlenül, hogy hány világító pont van. Ha nem elegendő a belső tár, akkor becsüljük meg a világító pontok számát, és ezt a számot írjuk be a 12004. sorba. Ez minden olyan programra vonatkozik, ahol felületekkel végzünk műveleteket.

Ezek a programok a következők:

MÁSOLÓ

MEZÖTÜKÖR

MEZŐFORGATÁS

KÉPTÁROLÁS

KÉPBETÖLTÉS

```
12000 REM MÁSOLÓ
12001 GOTO 12034
12002 REM MÁSOLÓ ALPROGRAM
12003 LG=LC*LD
12004 DIM LH(LG):DIM LM(LG)
12005 LD=0:LP=0:LQ=0:LZ=0:LK=0
12006 LY=LB
12007 FOR LI=1 TO LD
12008 LX=LA
12009 LY=LY+1
12010 FOR LJ=1 TO LC
12011 LX=LX+1
12012 LL=TEST(LX,LY):PLOT LX,LY,2:PLOT LX,LY,2
12013 IF LL=1 THEN 12015
12014 GOTO 12018
12015 LK=LK+1
12016 LH(LK)=LX
12017 LM(LK)=LY
12018 NEXT LJ
12019 NEXT LI
12020 FOR LN=1 TO LK
12021 LV=LH(LN)+(LE-LA)+LD
12022 LW=LM(LN)+(LF-LB)+LP
12023 :PLOT LV,LW,1
12024 NEXT LN
12025 LQ=LQ+1
12026 IF LQ>LT THEN 12029
12027 LD=LD+LR
12028 GOTO 12020
12029 LZ=LZ+1
12030 IF LZ>LU THEN 12033
12031 LP=LP+LS
12032 GOTO 12020
12033 RETURN
12034 HIRES 0,7
12035 LA= 55 :REM A MEZŐ BAL FELSŐ SARKÁNAK X-KOORD.
```

```

12036 LB= 60 :REM A MEZŐ BAL FELSŐ SARKÁNAK Y-KOORD.
12037 LC= 20 :REM A MÁ SOLANDÓ MEZŐ SZÉLESSÉGE
12038 LD= 55 :REM A MÁ SOLANDÓ MEZŐ MAGASSÁGA
12039 LE=210 :REM AZ ÚJ X-KOORDINÁTA
12040 LF= 25 :REM AZ ÚJ Y-KOORDINÁTA
12041 LR= 0 :REM A MÁ SOLATOK TÁVOLSÁGA X-IRÁNYBAN
12042 LS= 0 :REM A MÁ SOLATOK TÁVOLSÁGA Y-IRÁNYBAN
12043 LT= 0 :REM AZ ISMÉTLÉSEK SZÁMA X-IRÁNYBAN
12044 LU= 0 :REM AZ ISMÉTLÉSEK SZÁMA Y-IRÁNYBAN
12045 :REC 30, 25, 70, 130, 1
12046 :LINE 30, 155, 100, 25, 1
12047 :LINE 30, 25, 100, 155, 1
12048 :LINE 65, 25, 65, 155, 1
12049 :TEXT 42, 80, "EREDETI", 1, 1, 8
12050 GOSUB 12002
12051 CLR
12052 LA= 28 :LB=82 :LC=75 :LD=15
12053 LE=125 :LF=65 :LR= 0 :LS=35
12054 LT= 0 :LU= 2
12062 GOSUB 12002
12063 GOTO 12063

```

2.5.2. TÜKRÖZÉS

2.5.2.1. Ábrák tükrözése

Az ÁBRATÜKÖR program segítségével olyan alakzatokat tükrözhetünk, amelyeknek a koordinátái ismertek.

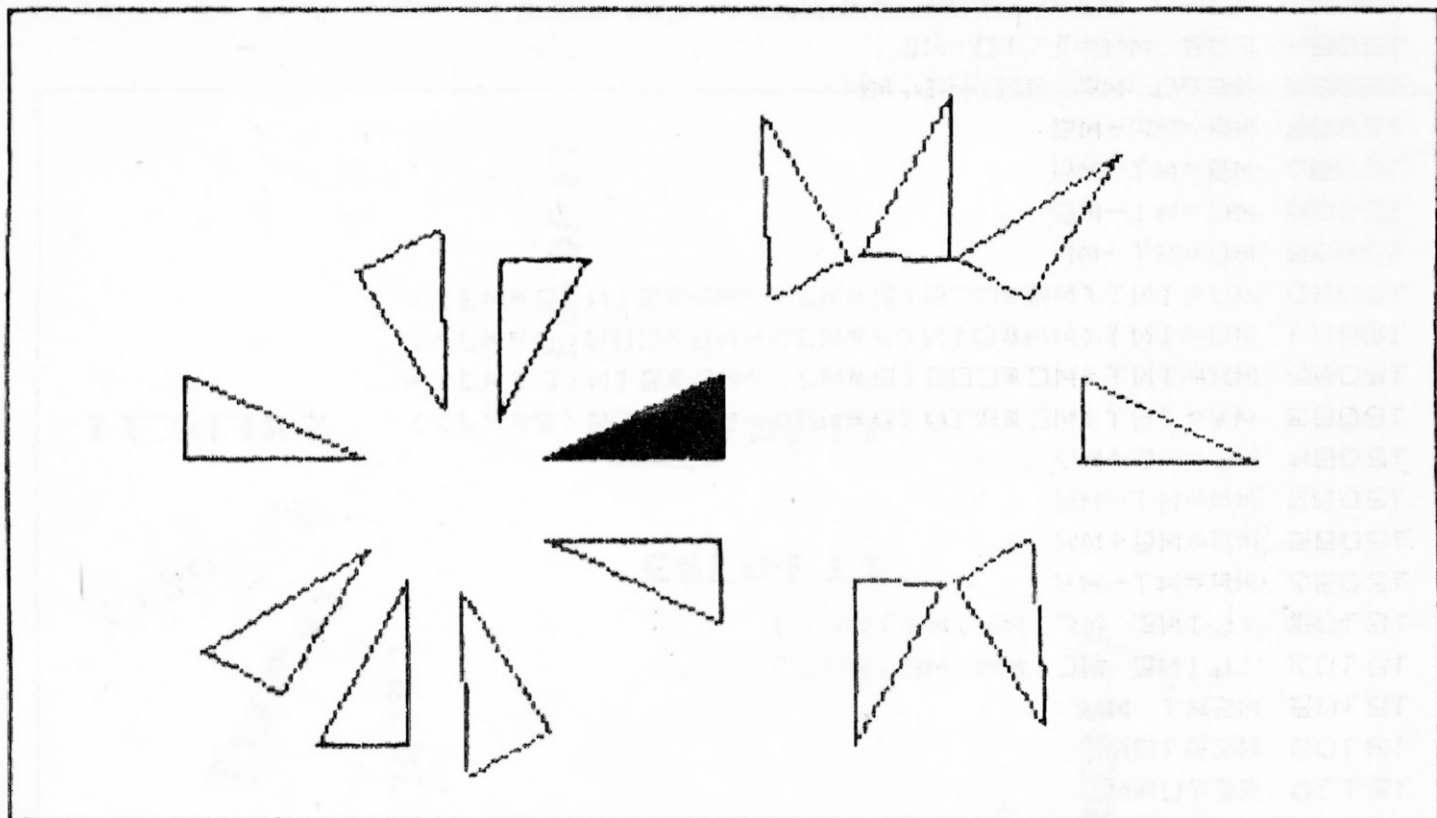
MW-vel adjuk meg a tükrözés tengelyének a hajlásszögét. A szöget 0° és 360° között adjuk meg, matematikai értelmezésben. Ez azt jelenti, hogy a szöget az óramutató járásával ellentétes irányban mérjük, ahol a pozitív X tengely a 0° (360°), a pozitív Y tengely a 90° , a negatív X tengely a 180° és a negatív Y tengely a 270° .

ME a DATA sorok számát adja meg. A DATA sorok segítségével adhatjuk meg a tükrözendő alakzatot. Tetszőleges, egyenesekből álló alakzatoknál a DATA sorokat úgy kell felépíteni, ahogy azt a 12116. sor mutatja. Az értékek mindig párban vannak: az első pont X és Y koordinátája (1-es indexszel jelölve) és a második pont X és Y koordinátája. (Ez az a pont, amivel az első pontot összekötjük.) Így az ábrát további DATA sorokkal ki lehet egészíteni.

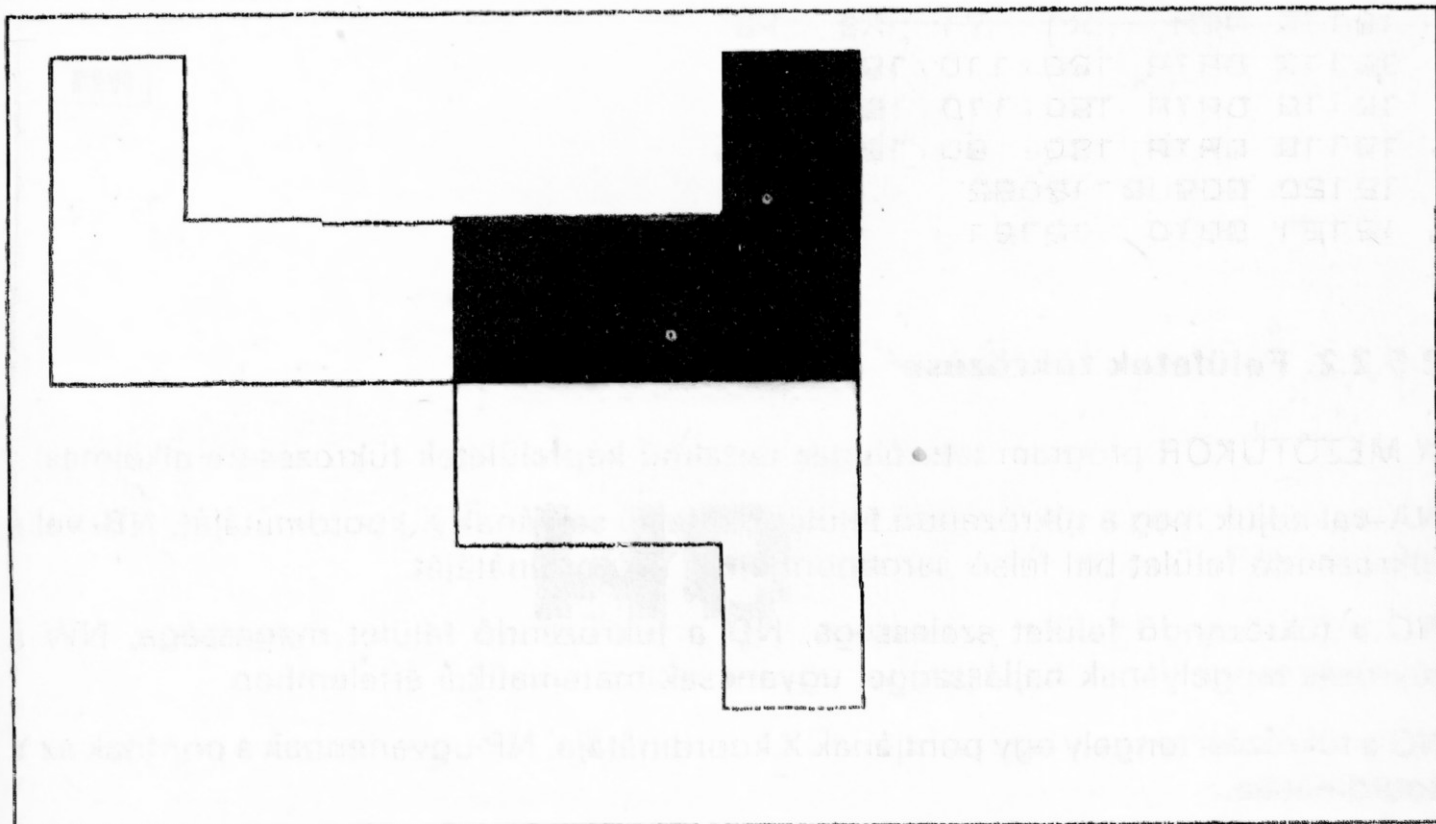
MS a tükrözési tengely egy pontjának X koordinátája, MT a tengely egy pontjának Y koordinátája.

Azt, hogy hova kerül a tükrözött alakzat, azt a tükrözési tengely hajlásszöge és egy pontja határozza meg. Különösen ügyelni kell arra, hogy a tükrözött alakzat ne kerüljön a képernyőn kívülre. Ez sokszor fejtörést okoz, némi gyakorlattal azonban mesterei lehetünk a tükrözésnek.

A 47. és 48. ábra néhány olyan tükrözést mutat be, amit az ÁBRATÜKÖR programmal hajtottunk végre.



47. ábra. Különböző szimmetriatengelyekre való tükrözés. A kitöltött háromszög az eredeti



48. ábra. Tükrözéssel előállított ábra. A kitöltött alakzat az eredeti

```

12080 REM ÁBRATÜKÖR
12081 GOTO 12111
12082 REM ÁBRATÜKÖR ALPROGRAM
12083 MJ=0.01745*MW
12084 FOR MN=1 TO ME
12085 READ MF,MH,MI,MK
12086 MA=MF-MS
12087 MB=MT-MH
12088 MC=MI-MS
12089 MD=MT-MK
12090 MV=INT<MA*COS<2*MJ>+MB*SIN<2*MJ>>
12091 MW=INT<MA*SIN<2*MJ>-MB*COS<2*MJ>>
12092 MX=INT<MC*COS<2*MJ>+MD*SIN<2*MJ>>
12093 MY=INT<MC*SIN<2*MJ>-MD*COS<2*MJ>>
12094 ML=MS+MV
12095 MM=MT-MW
12096 MO=MS+MX
12097 MP=MT-MY
12106 :LINE MF,MH,MI,MK,1
12107 :LINE ML,MM,MO,MP,1
12108 NEXT MN
12109 RESTORE
12110 RETURN
12111 HIRES 0,7
12112 MW= 0 :REM A TÜKÖR SZÖGE
12113 ME= 3 :REM A DATA-SOROK SZÁMA
12114 MS=100 :REM A TÜKÖR EGY PONTJÁNAK X-KOORDINÁTÁJA
12115 MT=120 :REM A TÜKÖR EGY PONTJÁNAK Y-KOORDINÁTÁJA
12116 REM X1 Y1 X2 Y2
12117 DATA 120,110,160,110
12118 DATA 160,110,160, 90
12119 DATA 160, 90,120,110
12120 GOSUB 12082
12121 GOTO 12121

```

2.5.2.2. Felületek tükrözése

A MEZÖTÜKÖR program tetszőleges tartalmú képfelületek tükrözésére alkalmas.

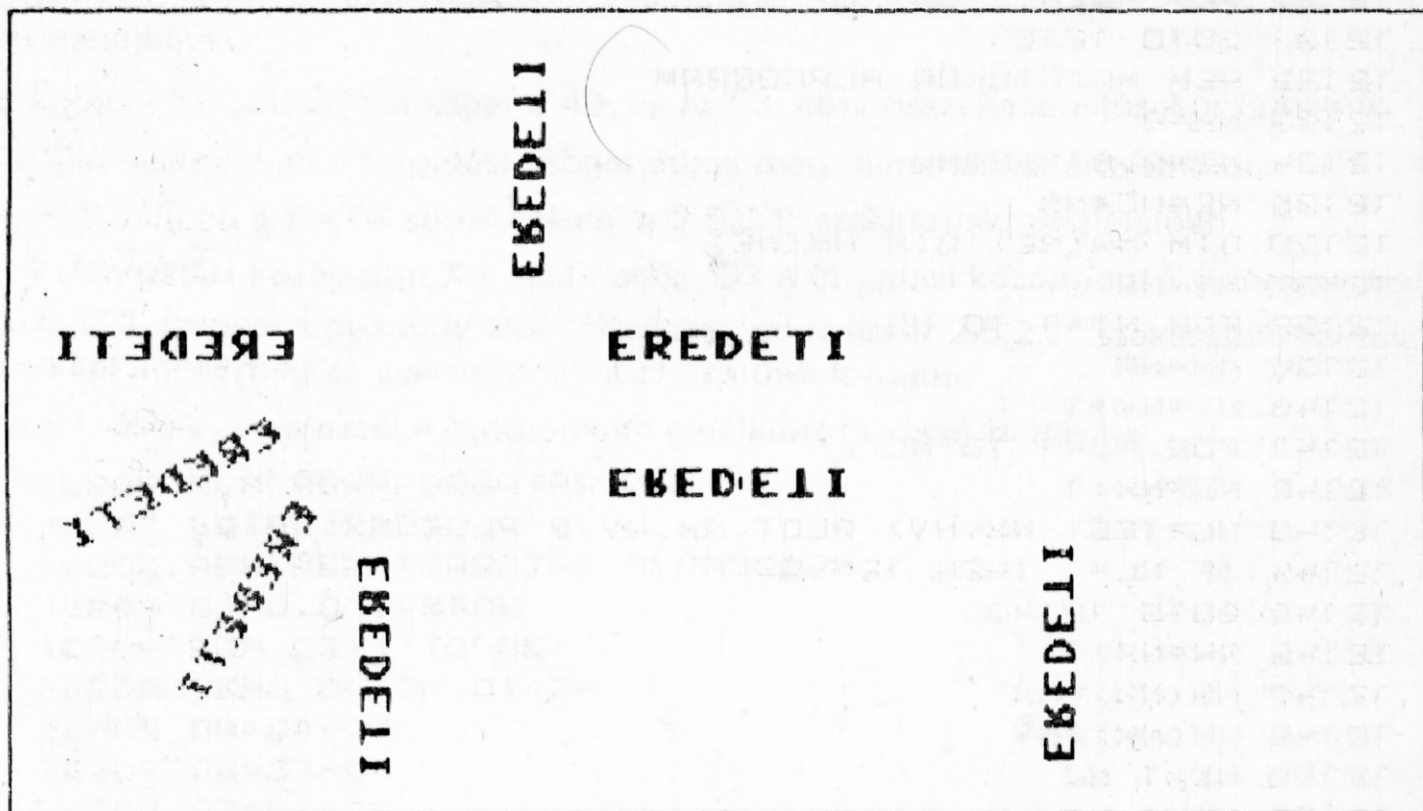
NA-val adjuk meg a tükrözendő felület bal felső sarkának X koordinátáját, NB-vel a tükrözendő felület bal felső sarokpontjának Y koordinátáját.

NC a tükrözendő felület szélessége, ND a tükrözendő felület magassága, NW a tükrözés tengelyének hajlásszöge, ugyancsak matematikai értelemben.

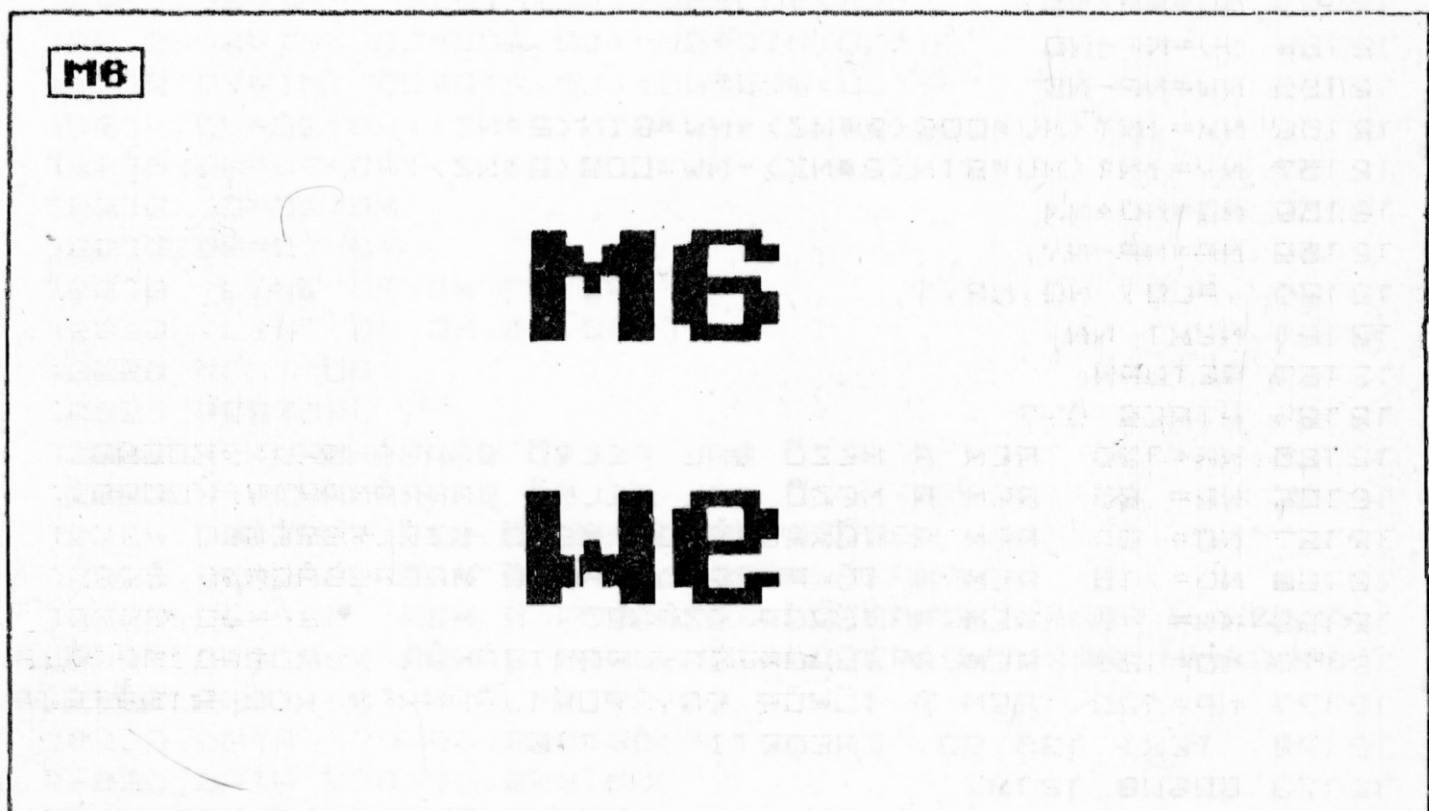
NO a tükrözési tengely egy pontjának X koordinátája, NP ugyanennek a pontnak az Y koordinátája.

Az összes többi az előző pontban mondottaknak megfelelően megy végbe. Különösen ügyeljen NC és ND választására!

A 49. és 50. ábra mutat be példákat a felülettükrözésre.



49. ábra. Egy képrész különböző tengelyekre való tükrözése



50. ábra. Először zoomolt, utána tükrözött képrész. Az eredeti bal oldalt fent

Érdeemes ezeket a képeket pontosabban tanulmányozni. A szabálytalanságok, amiket biztosan már észrevettek, az egész számokra való kerekítésekből adódnak. Ez digitális kép, ezért ennek nincs ellenszere.

```
12130 REM MEZŐTÜKÖR
12131 GOTO 12164
12132 REM MEZŐTÜKÖR ALPROGRAM
12133 NK=0
12134 NZ=0.01745*NW
12135 NE=NC*ND
12136 DIM NH<NE>:DIM NM<NE>
12137 NY=NB
12138 FOR NI=1 TO ND
12139 NX=NA
12140 NY=NY+1
12141 FOR NJ=1 TO NC
12142 NX=NX+1
12143 NL=TEST<NX,NY>:PLOT NX,NY,2:PLOT NX,NY,2
12144 IF NL=1 THEN 12146
12145 GOTO 12149
12146 NK=NK+1
12147 NH<NK>=NX
12148 NM<NK>=NY
12149 NEXT NJ
12150 NEXT NI
12151 FOR NN=1 TO NK
12152 NF=NH<NN>
12153 NG=NM<NN>
12154 NV=NF-ND
12155 NW=NP-NG
12156 NX=INT<NV*COS<2*NZ>+NW*SIN<2*NZ>>
12157 NY=INT<NV*SIN<2*NZ>-NW*COS<2*NZ>>
12158 NQ=NO+NX
12159 NR=NP-NY
12160 :PLOT NQ,NR,1
12161 NEXT NN
12162 RETURN
12163 HIRES 0,7
12164 NA=130 :REM A MEZŐ BAL FELSŐ SARKÁNAK X-KOORD.
12165 NB= 85 :REM A MEZŐ BAL FELSŐ SARKÁNAK Y-KOORD.
12166 NC= 60 :REM A TÜKRÖZENDŐ MEZŐ SZÉLESSÉGE
12167 ND= 15 :REM A TÜKRÖZENDŐ MEZŐ MAGASSÁGA
12168 NW= 0 :REM A TÜKÖR SZÖGE
12169 NO=120 :REM A TÜKÖR EGY PONTJÁNAK X-KOORDINÁTÁJA
12170 NP=120 :REM A TÜKÖR EGY PONTJÁNAK Y-KOORDINÁTÁJA
12171 :TEXT 135,90,"EREDETI",1,1,8
12172 GOSUB 12132
12173 GOTO 12174
```


2.5.3. FORGATÁS

2.5.3.1. Ábrák forgatása

Ugyanazt, amit a tükrözésekkel végeztünk, természetesen a forgatásokkal is elvégezhetjük.

A legmarkánsabb különbséget a 49. és az 53. ábra összehasonlításakor láthatjuk.

Az OW változóban a forgatási szöveget adjuk meg matematikai értelemben.

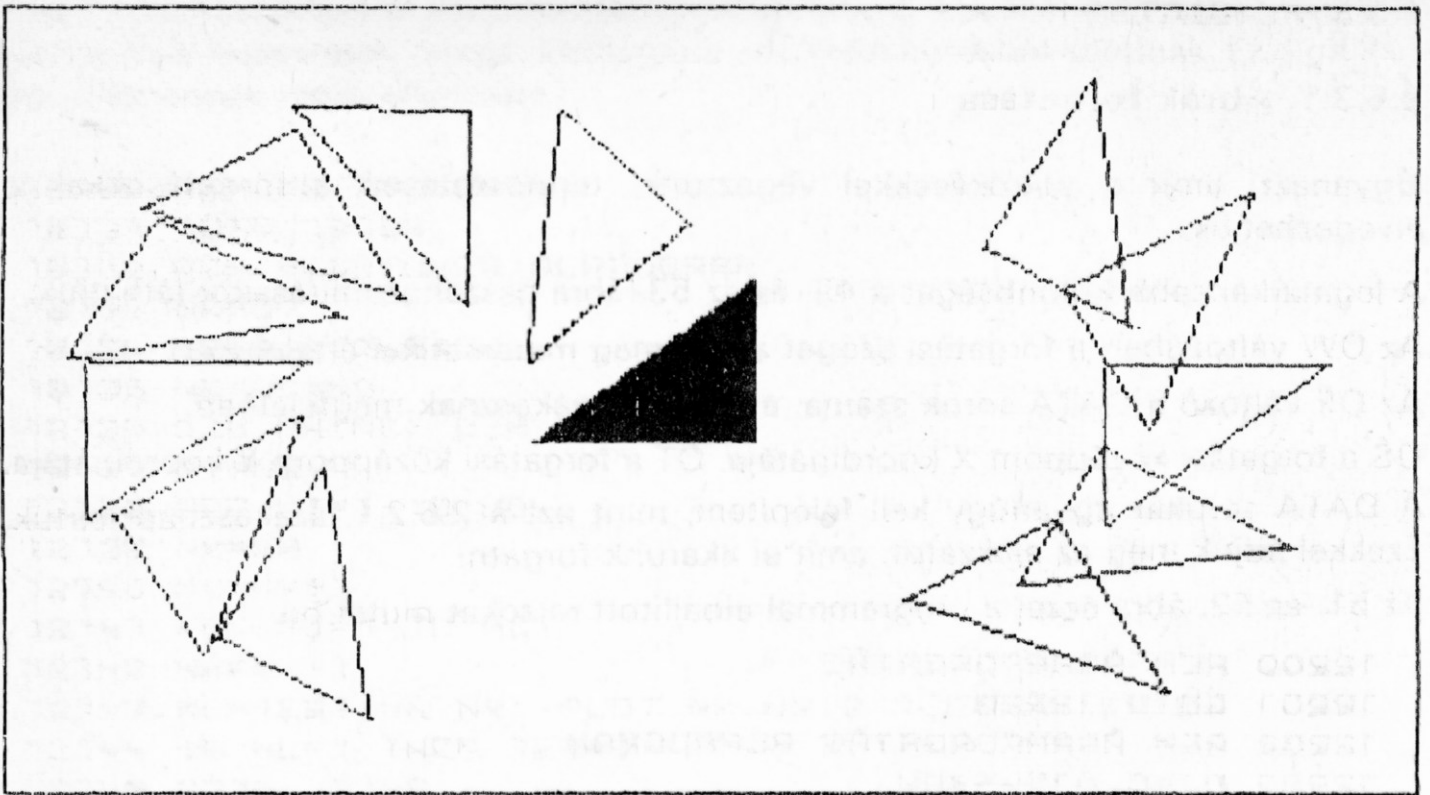
Az OE változó a DATA sorok száma, a 2.5.2.1. szakasznak megfelelően.

OS a forgatási középpont X koordinátája, OT a forgatási középpont Y koordinátája.

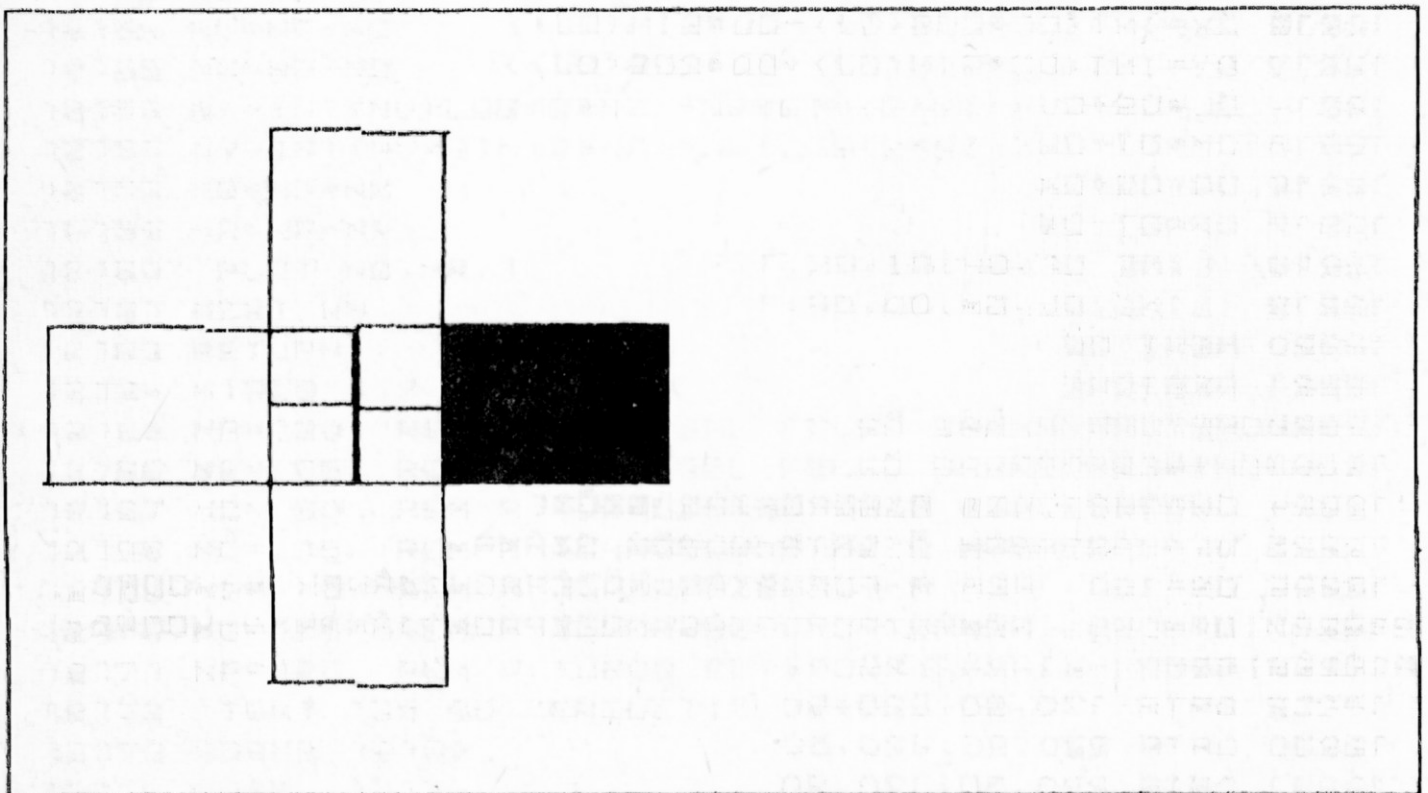
A DATA sorokat ugyanúgy kell felépíteni, mint azt a 2.5.2.1. szakaszban leírtuk. Ezekkel adjuk meg az alakzatot, amit el akarunk forgatni.

Az 51. és 52. ábra ezzel a programmal előállított rajzokat mutat be.

```
12200 REM ÁBRAFORGATÁS
12201 GOTO 12223
12202 REM ÁBRAFORGATÁS ALPROGRAM
12203 OJ=0.01745*OW
12204 FOR OQ=1 TO OE
12205 READ OF,OH,OI,OK
12206 OA=OF-OS
12207 OB=OT-OH
12208 OC=OI-OS
12209 OD=OT-OK
12210 OV=INT<OA*COS<OJ>-OB*SIN<OJ>>
12211 OW=INT<OA*SIN<OJ>+OB*COS<OJ>>
12212 OX=INT<OC*COS<OJ>-OD*SIN<OJ>>
12213 OY=INT<OC*SIN<OJ>+OD*COS<OJ>>
12214 OL=OS+OV
12215 OM=OT-OW
12216 OO=OS+OX
12217 OP=OT-OY
12218 :LINE OF,OH,OI,OK,1
12219 :LINE OL,OM,OO,OP,1
12220 NEXT OQ
12221 RESTORE
12222 RETURN
12223 HIRES 0,7
12224 OW=180 :REM A FORGATÁS SZÖGE
12225 OE= 3 :REM A DATA-SOROK SZÁMA
12226 OS=160 :REM A FORGATÁS KÖZÉPPONTJÁNAK X-KOORD.
12227 OT= 90 :REM A FORGATÁS KÖZÉPPONTJÁNAK Y-KOORD.
12228 REM X1 Y1 X2 Y2
12229 DATA 170,90,220,90
12230 DATA 220,90,220,50
12231 DATA 220,50,170,90
12232 GOSUB 12202
12233 GOTO 12233
```



51. ábra. Különböző középpontok körüli forgatások. A kitöltött háromszög az eredeti



52. ábra. Forgatással előállított ábra. A kitöltött négyszög az eredeti

2.5.3.2. Felületek forgatása

A MEZŐFORGATÁS programmal olyan képernyőterületeket lehet elforgatni, amelyek tetszőleges pontokat tartalmaznak.

PA-val megadjuk az elforgatandó felület bal felső sarokpontjának X koordinátáját, PB a felület sarokpontjának Y koordinátája, PC a felület szélességét határozza meg, PD a forgatandó felület magassága, PW az elforgatási szög, PO az elforgatási középpont X koordinátája, PP az elforgatási középpont Y koordinátája.

A program listájában láthatjuk, hogy ha több forgatást akarunk egymás után elvégezni, akkor CLR utasítást kell kiadni az egyes forgatások között. Erre azért van szükség, hogy a számítógép ne álljon le a REDIMENSION-ERROR hibajelzéssel.

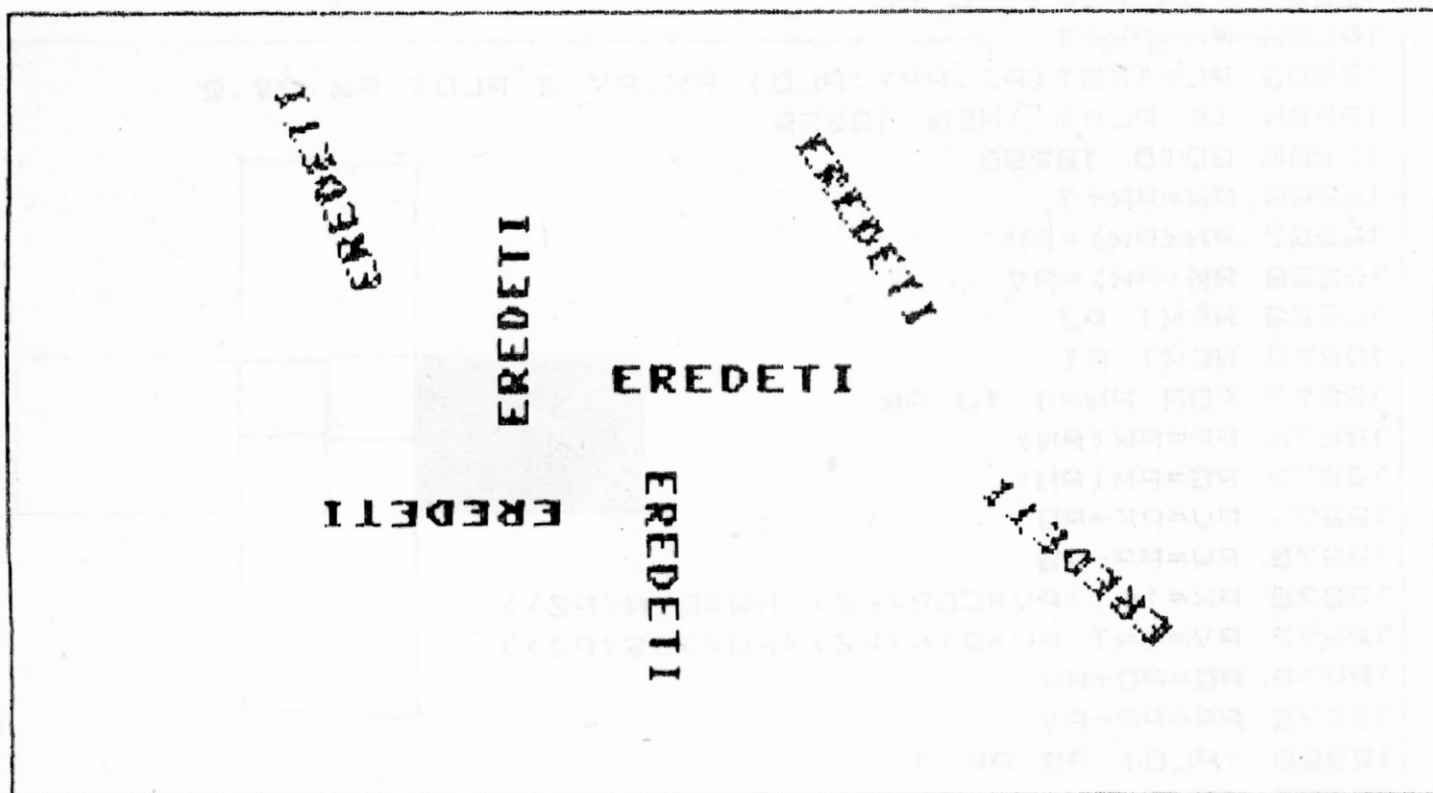
Az 53. és 54. ábrán láthatjuk a program eredményét. E program segítségével tetszőleges helyzetben rajzolhatunk ellipsziseket és más görbéket.

```
12250 REM MEZŐFORGATÁS
12251 GOTO 12283
12252 REM MEZŐFORGATÁS ALPROGRAM
12253 PK=0
12254 PZ=0.01745*PW
12255 PE=PC*PD
12256 DIM PH<PE>:DIM PM<PE>
12257 PY=PB
12258 FOR PI=1 TO PD
12259 PX=PA
12260 PY=PY+1
12261 FOR PJ=1 TO PC
12262 PX=PX+1
12263 PL=TEST<PX,PY>:PLOT PX,PY,2:PLOT PX,PY,2
12264 IF PL=1 THEN 12266
12265 GOTO 12269
12266 PK=PK+1
12267 PH<PK>=PX
12268 PM<PK>=PY
12269 NEXT PJ
12270 NEXT PI
12271 FOR PN=1 TO PK
12272 PF=PH<PN>
12273 PG=PM<PN>
12274 PV=PF-PO
12275 PW=PP-PG
12276 PX=INT<PV*COS<PZ>-PW*SIN<PZ>>
12277 PY=INT<PV*SIN<PZ>+PW*COS<PZ>>
12278 PQ=PO+PX
12279 PR=PP-PY
12280 :PLOT PQ,PR,1
12281 NEXT PN
12282 RETURN
12283 HIRES 0,7
```

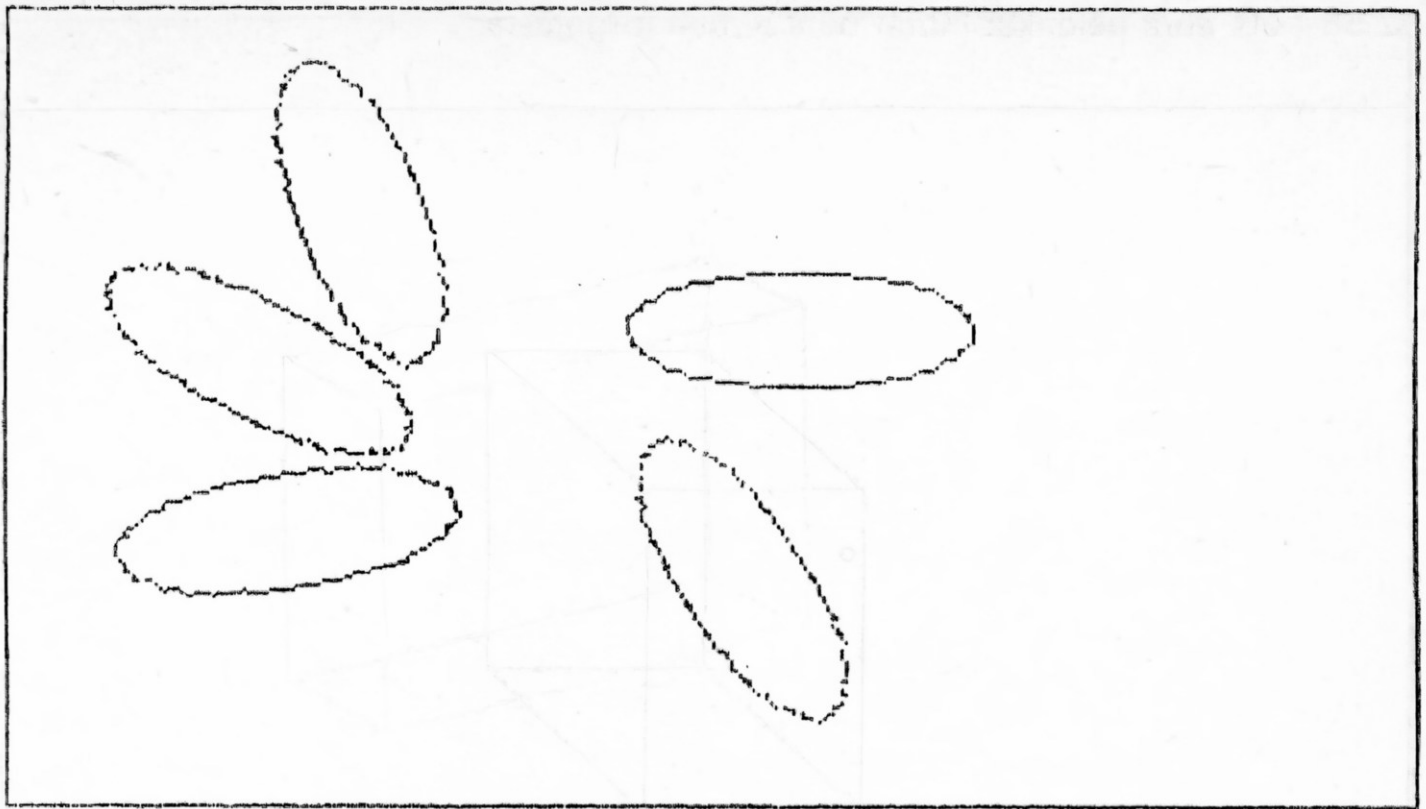
```

12284 PA=130 :REM A MEZŐ BAL FELSŐ SARKÁNAK X-KOORD.
12285 PB= 85 :REM A MEZŐ BAL FELSŐ SARKÁNAK Y-KOORD.
12286 PC= 60 :REM A FORGATANDÓ MEZŐ SZÉLESSÉGE
12287 PD= 15 :REM A FORGATANDÓ MEZŐ MAGASSÁGA
12288 PW= 90 :REM A FORGATÁS SZÖGE
12289 PO=130 :REM A FORGATÁS KÖZÉPPONTJÁNAK X-KOORD.
12290 PP=110 :REM A FORGATÁS KÖZÉPPONTJÁNAK Y-KOORD.
12291 :TEXT 135,90,"EREDETI",1,1,8
12292 GOSUB 12252
12293 CLR
12294 PA=130:PB=85:PC=60:PD=15:PW=180:PO=130:PP=110
12295 GOSUB 12252
12296 CLR
12297 PA=130:PB=85:PC=60:PD=15:PW=270:PO=130:PP=110
12298 GOSUB 12252
12299 CLR
12300 PA=130:PB=85:PC=60:PD=15:PW=110:PO=100:PP=100
12301 GOSUB 12252
12302 CLR
12303 PA=130:PB=85:PC=60:PD=15:PW=300:PO=210:PP=100
12304 GOSUB 12252
12305 CLR
12306 PA=130:PB=85:PC=60:PD=15:PW=135:PO=210:PP=100
12307 GOSUB 12252
12308 GOTD 12308

```



53. ábra. Egy képrész forgatása



54. ábra. Egy ellipszist tartalmazó képrész forgatása

2.5.3.3. Térbeli forgatás

A TÉRBELI FORGATÁS program tetszőleges, egyenes élekkel határolt testeket a térben három tengely körül forgat.

WX az X tengellyel bezárt elforgatási szög. A negatív WX azt jelenti, hogy az Y tengelyt a \emptyset pont körül balra forgatjuk, vagyis az X tengely irányába futó éleket felfelé forgatjuk. Pozitív WX-nél mindez fordítva van.

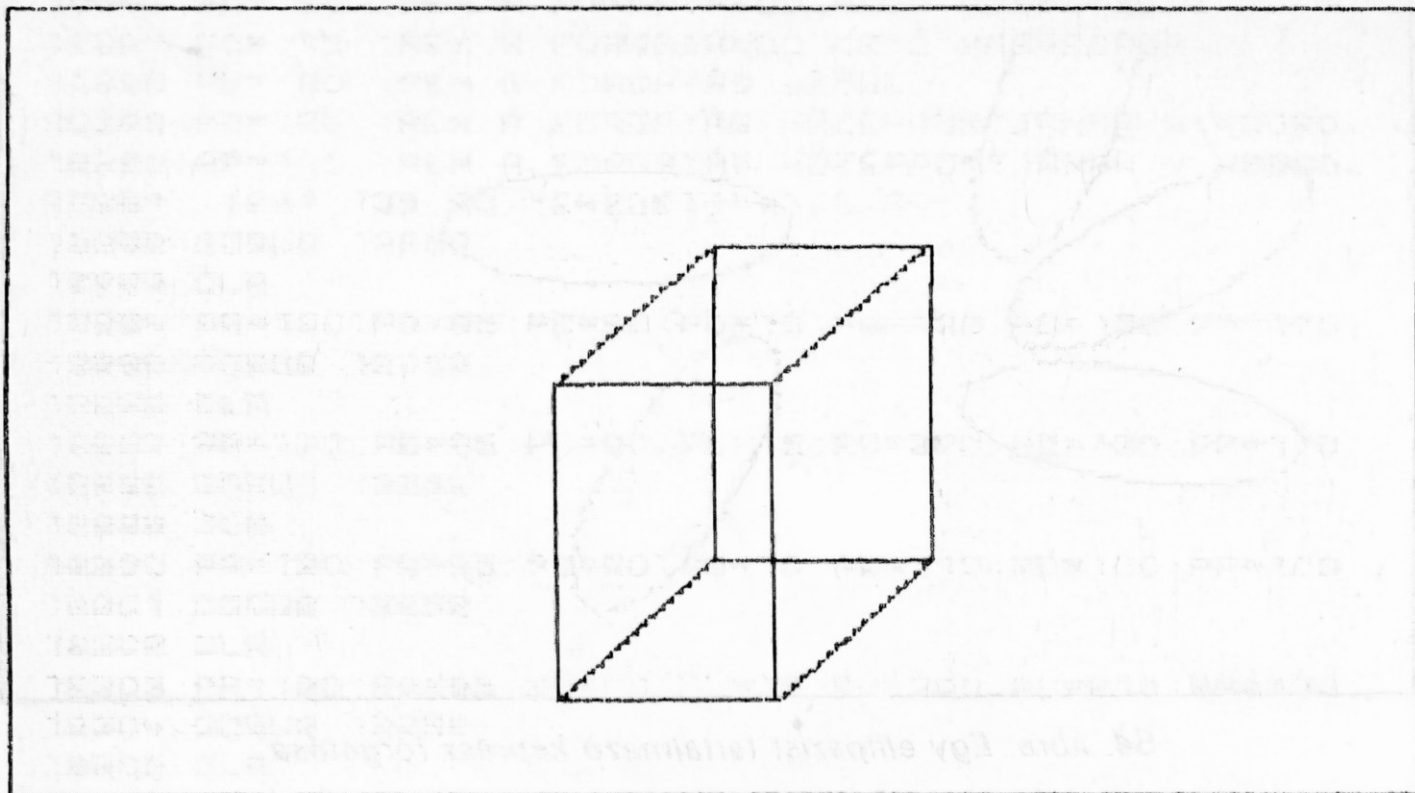
WY az Y tengellyel bezárt forgatási szög. Pozitív WY azt jelenti, hogy az X tengelyt a \emptyset pont körül előre forgatjuk, vagyis az Y tengely irányába futó éleket előre forgatjuk. Negatív WY ennek ellentettjét jelenti.

WZ a Z tengellyel bezárt forgatási szög. Ez a tengely vagy előre, vagy hátra mutat (a valóságban mind az X, mind az Y tengellyel 90° -os szöget zár be, a rajzban 45° -os a hajlásszöge). Pozitív WZ azt jelenti, hogy az Y tengelyt a \emptyset pont körül hátrafelé forgatjuk, vagyis a Z irányban futó éleket felfelé forgatjuk el. Negatív WZ esetén ez fordítva van.

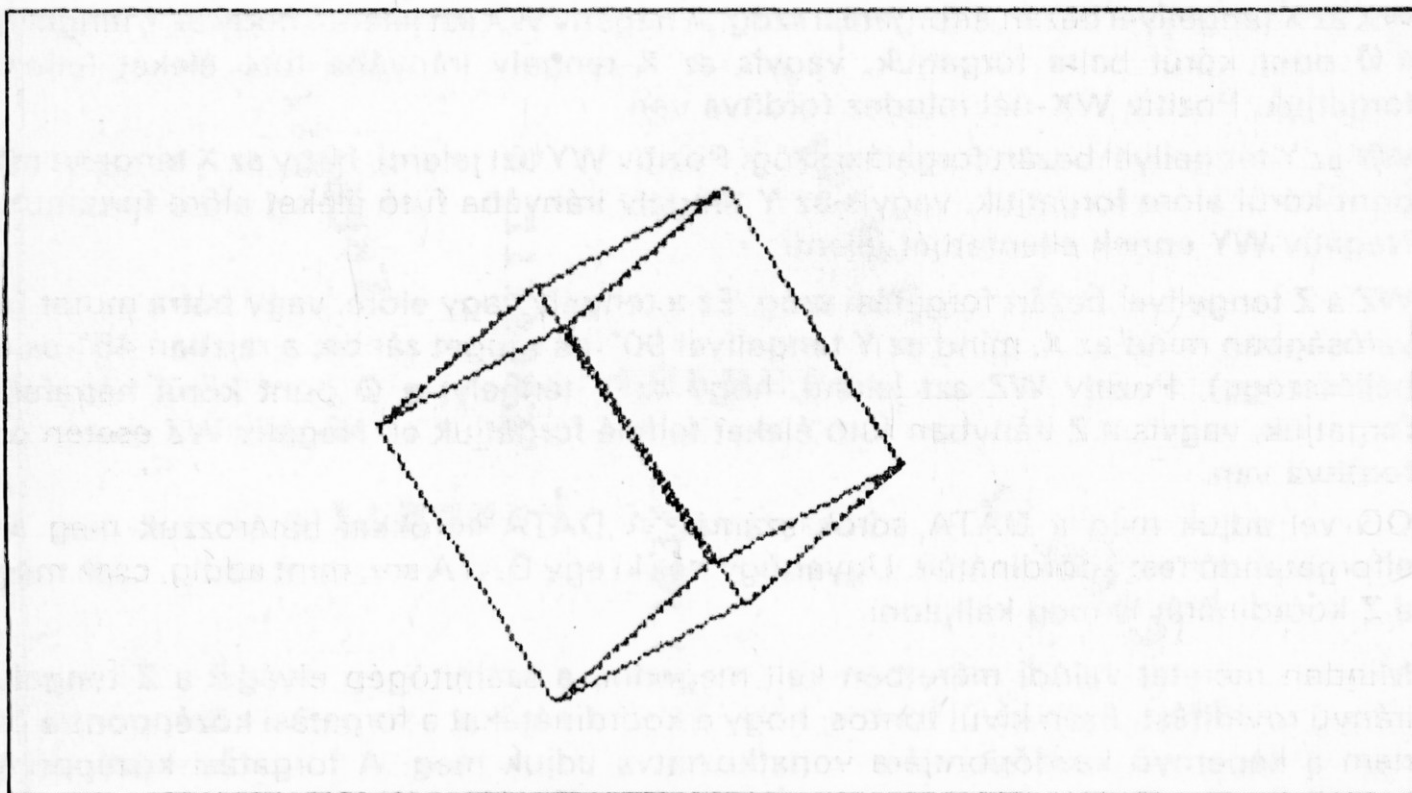
OG-vel adjuk meg a DATA sorok számát. A DATA sorokkal határozzuk meg az elforgatandó test koordinátáit. Ugyanúgy néz ki egy DATA sor, mint eddig, csak még a Z koordinátát is meg kell adni.

Minden méretet valódi méretben kell megadni, a számítógép elvégzi a Z tengely irányú rövidítést. Ezen kívül fontos, hogy a koordinátákat a forgatási középpontra és nem a képernyő kezdőpontjára vonatkoztatva adjuk meg. A forgatási középpont koordinátái: $x = \emptyset$, $y = \emptyset$, $z = \emptyset$. Így egy egyenesekkel határolt testet minden oldalról megnézhetünk. A szögek helyes megadása nem könnyű, és némi gyakorlatot és térlátást igényel.

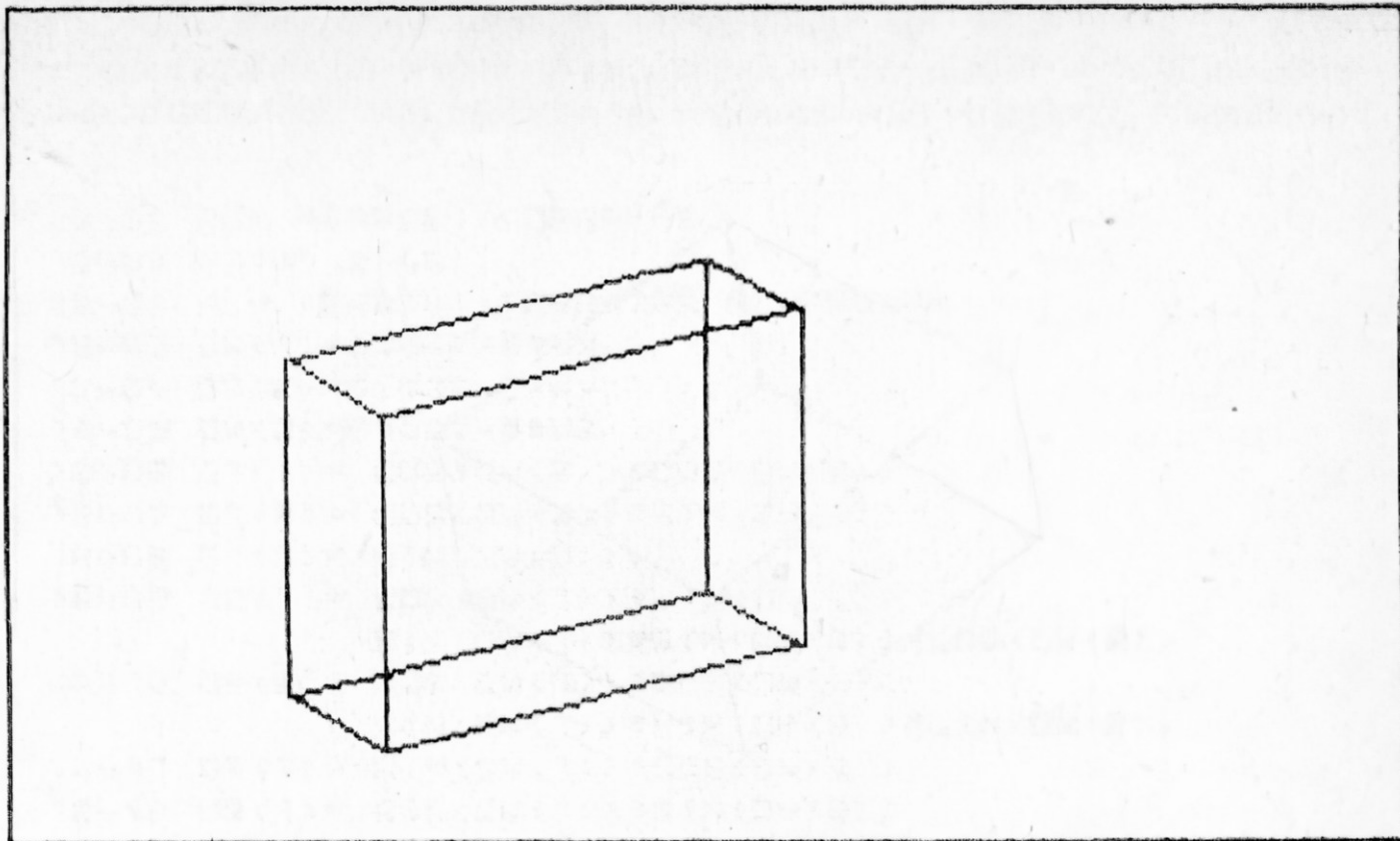
Az 55—60. ábra példákat mutat be a térbeli forgatásra.



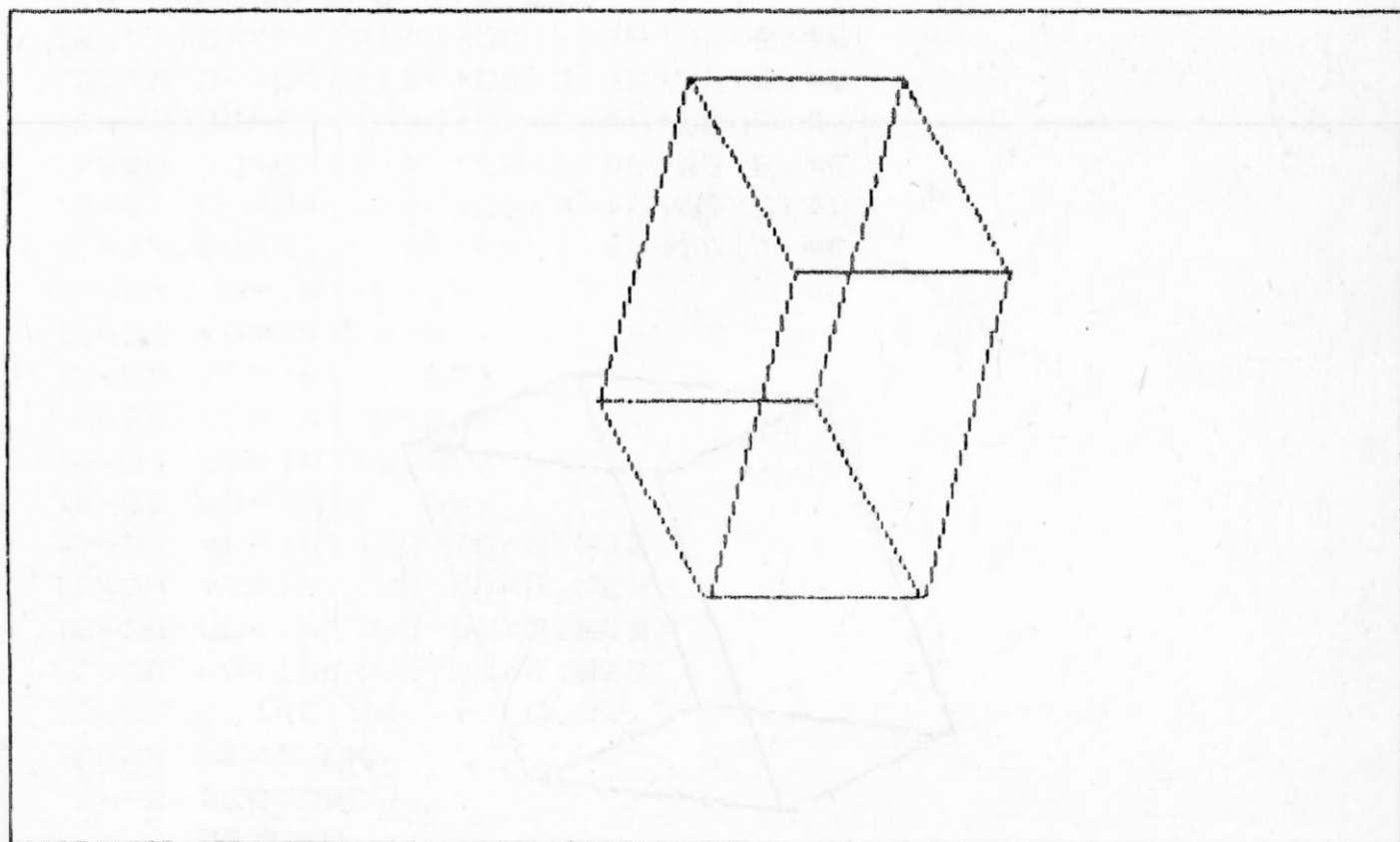
55. ábra. A téglatest eredeti helyzetében



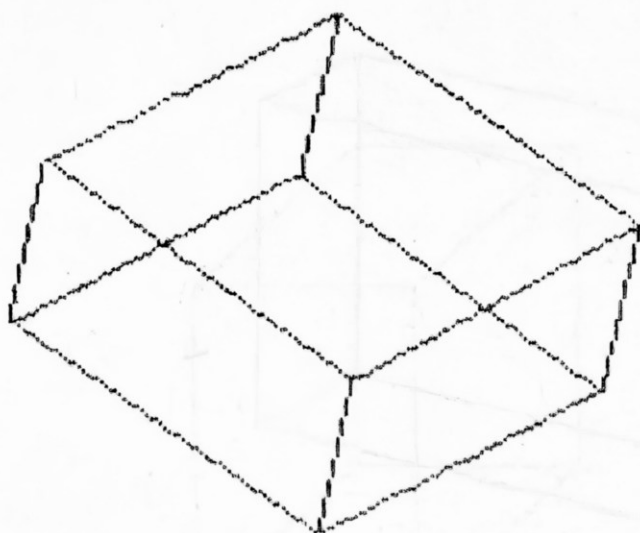
56. ábra. A téglatest $WX = -30^\circ$ -kal elforgatva



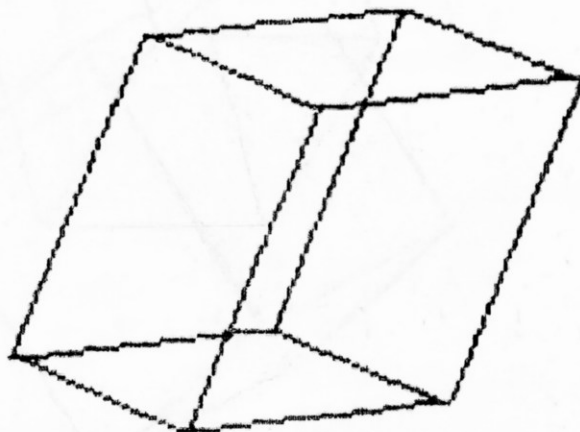
57. ábra. A téglatest $WY = 45^\circ$ -kal elforgatva



58. ábra. A téglatest $WZ = 45^\circ$ -kal elforgatva



59. ábra. A téglatest mind a 3 tengely körül elforgatva



60. ábra. A téglatest $WX = WY = WZ = 15^\circ$ -kal elforgatva

Meg tudják határozni a forgatási középpontot? Egy kis segítség: A forgatási középpont nem változik a képen. A koordinátái, amit V-vel és W-vel jelöltünk, ebben a példában állandóak. Más példában természetesen más értékeket is megadhatunk.

```

12400 REM TÉRBELI FORGATÁS
12401 GOTO 12442
12402 REM TÉRBELI FORGATÁS ALPROGRAM
12403 OW<1>=0.01745*WX
12404 OW<2>=0.01745*WY
12405 OW<3>=0.01745*WZ
12406 O1<1>= COS<OW<3>>*COS<OW<2>>
12407 O1<2>=-COS<OW<3>>*SIN<OW<2>>
12408 O1<3>= SIN<OW<3>>
12409 O2<1>= COS<OW<1>>*SIN<OW<2>>
          +SIN<OW<1>>*SIN<OW<3>>*COS<OW<2>>
12410 O2<2>= COS<OW<1>>*COS<OW<2>>
          -SIN<OW<1>>*SIN<OW<3>>*SIN<OW<2>>
12411 O2<3>=-SIN<OW<1>>*COS<OW<3>>
12412 O3<1>= SIN<OW<1>>*SIN<OW<2>>
          -COS<OW<1>>*SIN<OW<3>>*COS<OW<2>>
12413 O3<2>= SIN<OW<1>>*COS<OW<2>>
          +COS<OW<1>>*SIN<OW<3>>*SIN<OW<2>>
12414 O3<3>= COS<OW<1>>*COS<OW<3>>
12415 FOR OM=1 TO OG
12416 READ A,B,C,D,E,F
12417 OX=O1<2>*C+O2<2>*A+O3<2>*B
12418 OY=O1<3>*C+O2<3>*A+O3<3>*B
12419 OZ=O1<1>*C+O2<1>*A+O3<1>*B
12420 OU=O1<2>*F+O2<2>*D+O3<2>*E
12421 OV=O1<3>*F+O2<3>*D+O3<3>*E
12422 OW=O1<1>*F+O2<1>*D+O3<1>*E
12427 XO=INT<U+OX>
12428 YO=INT<W-OY>
12429 ZO=INT<  OZ>
12430 UO=INT<U+OU>
12431 VO=INT<W-OV>
12432 WO=INT<  OW>
12433 XX=INT<XO-ZO/2.828>
12434 YY=INT<YO+ZO/2.828>
12435 UU=INT<UO-WO/2.828>
12436 VV=INT<VO+WO/2.828>
12438 :LINE XX,YY,UU,VV,1
12439 NEXT OM
12440 RESTORE
12441 RETURN
12442 HIRES 0,7
12443 WX= 15:REM AZ X-TENGELLYEL BEZÁRT SZÖG
12444 WY= 15:REM AZ Y-TENGELLYEL BEZÁRT SZÖG
12445 WZ= 15:REM A Z-TENGELLYEL BEZÁRT SZÖG

```

```

12446 DG= 12:REM   A DATA-SOROK SZÁMA
12449 U =160:REM   A FORGATÁSI KÖZÉPPONT X-KOORD.
12450 W =140:REM   A FORGATÁSI KÖZÉPPONT Y-KOORD.
12451 REM   X1 Y1 Z1   X2 Y2 Z2
12452 DATA  0, 0,  0,50, 0,  0
12453 DATA  0, 0,  0, 0, 0,100
12454 DATA  0, 0,100,50, 0,100
12455 DATA 50, 0,100,50, 0,  0
12456 DATA 50, 0,  0,50,80,  0
12457 DATA 50, 0,100,50,80,100
12458 DATA  0, 0,100,  0,80,100
12459 DATA  0, 0,  0, 0,80,  0
12460 DATA  0,80,  0,50,80,  0
12461 DATA 50,80,  0,50,80,100
12462 DATA 50,80,100,  0,80,100
12463 DATA  0,80,100,  0,80,  0
12464 GOSUB 12402
12465 GOTO  12465

```

2.5.4. ROBBANTOTT RAJZOK

Sok esetben szeretnénk olyan összeállítású rajzot készíteni, mintha az egyes alkatrészeket egy robbanás széttolta volna. Ilyen robbantott rajzokat a ROBBANTÁS program segítségével készíthetünk.

Ezt a programot a TÉGLA programból készítettük. Az ott X1, Y1, X2, Y2-vel jelzett koordinátákat $QX(1) =$, $QY(1) =$, $QX(2) =$, $QY(2) =$ formában adjuk meg.

A QF-fel adjuk meg, hogy akarunk-e robbantani vagy sem:

QF = 1 azt jelenti, hogy nincs robbantás, QF = 2 jelenti a robbantást.

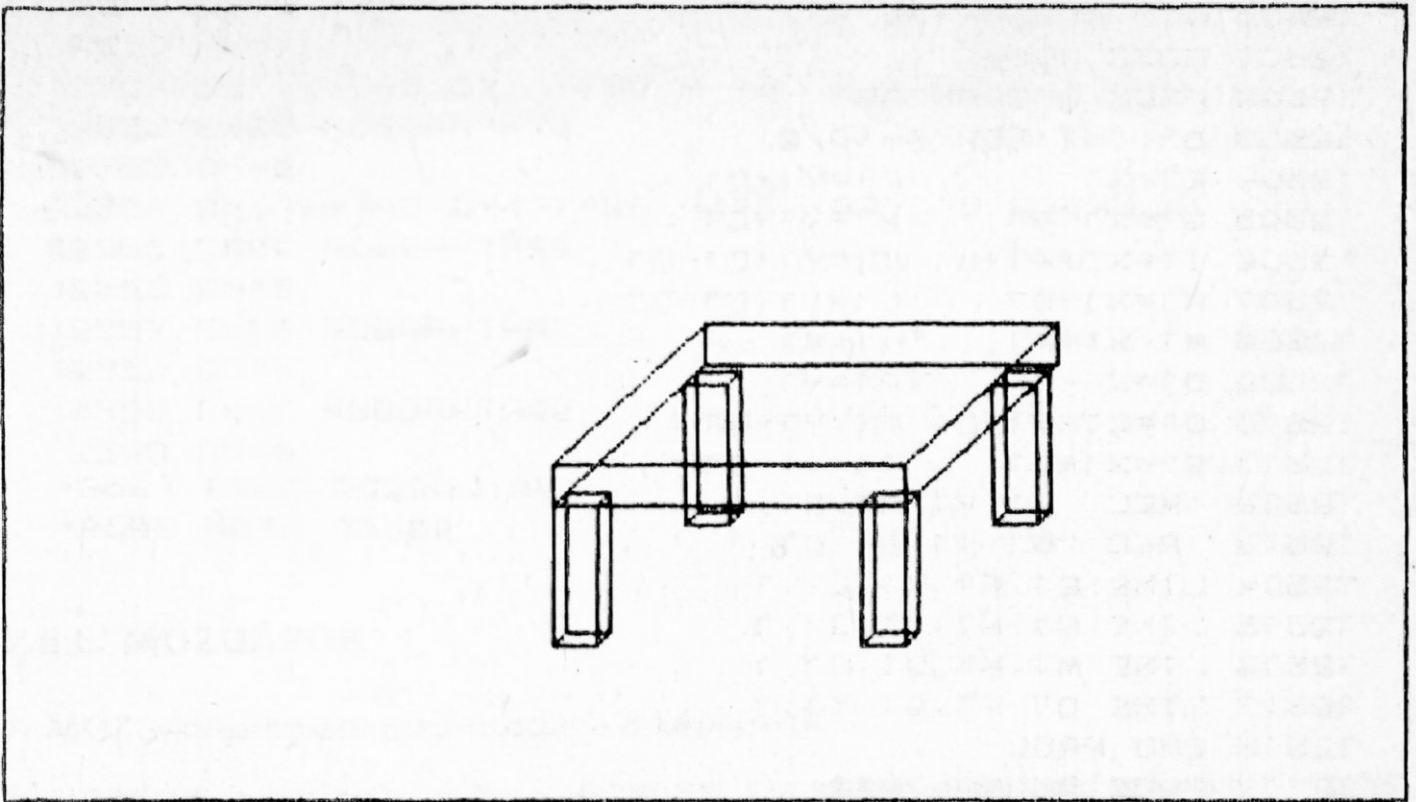
QE-vel adjuk meg a robbantási tényezőt X irányban. Ez azt jelzi, hogy mennyire kell az egyes részeket X irányban széthúzni.

Ugyanez érvényes a QQ-ra is, csak Y irányban.

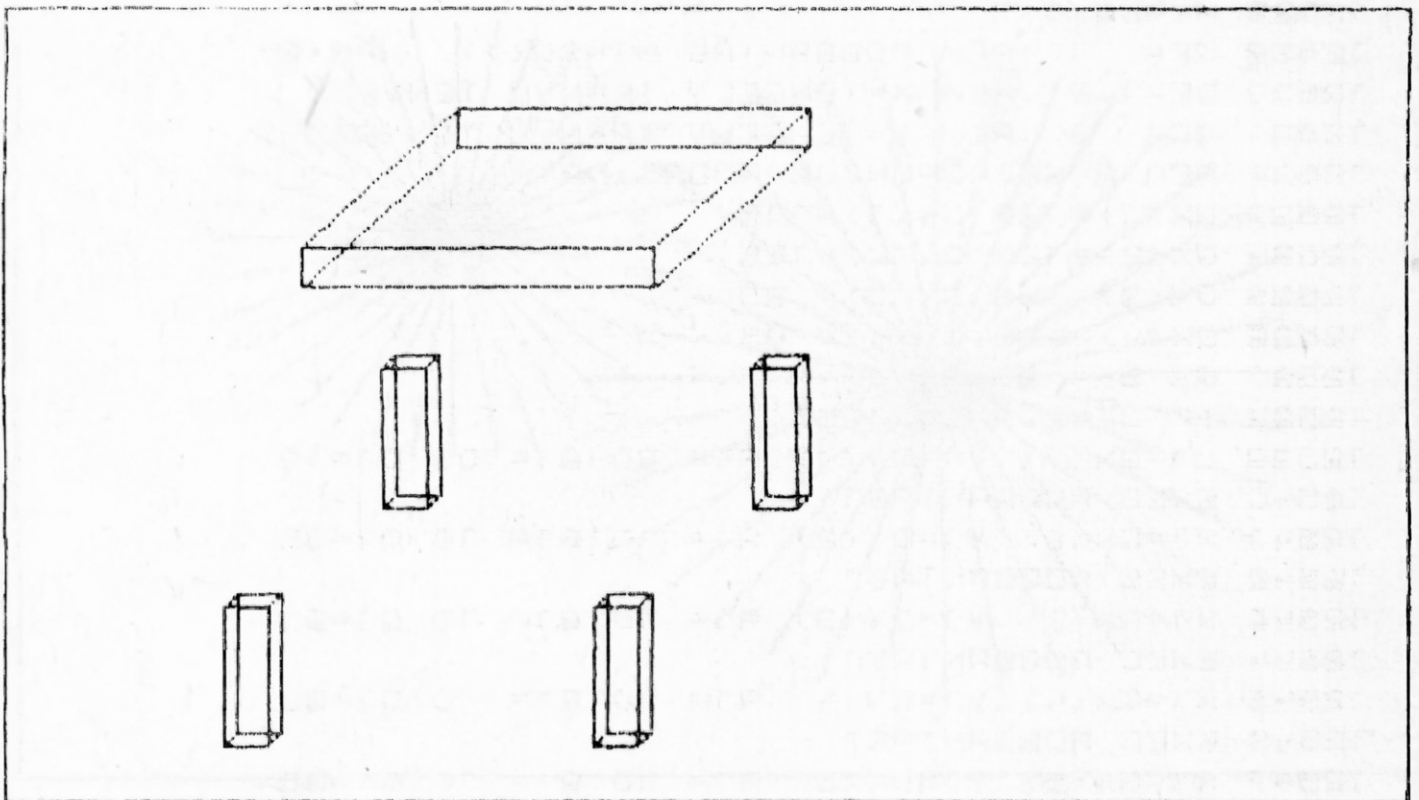
A1, B1 és C1 az egyes téglatestek méretét határozza meg. QN adja meg, hogy a QX-QY táblázatból melyik elemet kell felhasználni.

A 61. és 62. ábra bemutatja, hogyan lehet egy ábrát felrobbantani.

Bizonyára észrevették, hogy ez a program csak a téglatestekre igaz. Csak az elvet akartuk itt bemutatni, hogyan kell illet csinálni. Ha más ábrákat is fel akarunk robbantani, akkor újabb programot kell készíteni, ahol az egyik makrót úgy, ahogy itt bemutattuk, meg kell változtatni.



61. ábra. Összeállított ábrázolás



62. ábra. Robbantott ábrázolás

```

12500 REM ROBBANTÁS
12501 GOTO 12528
12502 PROC ROBBANTÁS1
12503 D1=INT(SQR(B1↑2/B))
12504 E1=X1      :F1=Y1+C1
12505 G1=X1+A1   :H1=Y1+C1
12506 I1=X1+A1+D1:J1=Y1+C1-D1
12507 K1=X1+D1   :L1=Y1+C1-D1
12508 M1=X1+A1   :N1=Y1
12509 O1=X1      :P1=Y1
12510 Q1=X1+A1+D1:R1=Y1-D1
12511 S1=X1+D1   :T1=Y1-D1
12512 :REC  X1,Y1,A1,C1,1
12513 :REC  S1,T1,A1,C1,1
12514 LINE E1,F1,K1,L1,1
12515 LINE G1,H1,I1,J1,1
12516 LINE M1,N1,O1,R1,1
12517 LINE D1,P1,S1,T1,1
12518 END PROC
12519 PROC ROBBANTÁS2
12520 QA=QX(QN)
12521 QB=QY(QN)
12522 QC=QE*QA-100
12523 QD=QQ*QB-100
12524 X1=INT(QC)
12525 Y1=INT(QD)
12526 EXEC ROBBANTÁS1
12527 END PROC
12528 HIRES 0,7
12529 QF= 1 :REM ROBBANTÁS NINCS<1>, VAN<2>
12530 QE=1.2 :REM X-TENGELY IRÁNYÚ TÉNYEZŐ
12531 QQ= 2 :REM Y-TENGELY IRÁNYÚ TÉNYEZŐ
12532 REM A KEZDŐPONTOK KOORDINÁTÁI
12533 QX<1>=125:QY<1>=115
12534 QX<2>=125:QY<2>=125
12535 QX<3>=155:QY<3>= 95
12536 QX<4>=225:QY<4>= 95
12537 QX<5>=195:QY<5>=125
12538 IF QF=2 THEN 12550
12539 X1=QX<1>:Y1=QY<1>:A1= 80:B1=100:C1=10
12540 EXEC ROBBANTÁS1
12541 X1=QX<2>:Y1=QY<2>:A1= 10:B1= 10:C1=35
12542 EXEC ROBBANTÁS1
12543 X1=QX<3>:Y1=QY<3>:A1= 10:B1= 10:C1=35
12544 EXEC ROBBANTÁS1
12545 X1=QX<4>:Y1=QY<4>:A1= 10:B1= 10:C1=35
12546 EXEC ROBBANTÁS1
12547 X1=QX<5>:Y1=QY<5>:A1= 10:B1= 10:C1=35
12548 EXEC ROBBANTÁS1

```

```

12549 GOTO 12562
12550 QN=1
12551 QX<1>=140:QY<1>=80:A1=80:B1=100:C1=10
12552 EXEC ROBBANTÁS2
12553 QN=2
12554 QX<1>=140:QY<1>=80:A1=10:B1= 10:C1=35
12555 EXEC ROBBANTÁS2
12556 QN=3
12557 EXEC ROBBANTÁS2
12558 QN=4
12559 EXEC ROBBANTÁS2
12560 QN=5
12561 EXEC ROBBANTÁS2
12562 GOTO 12562

```

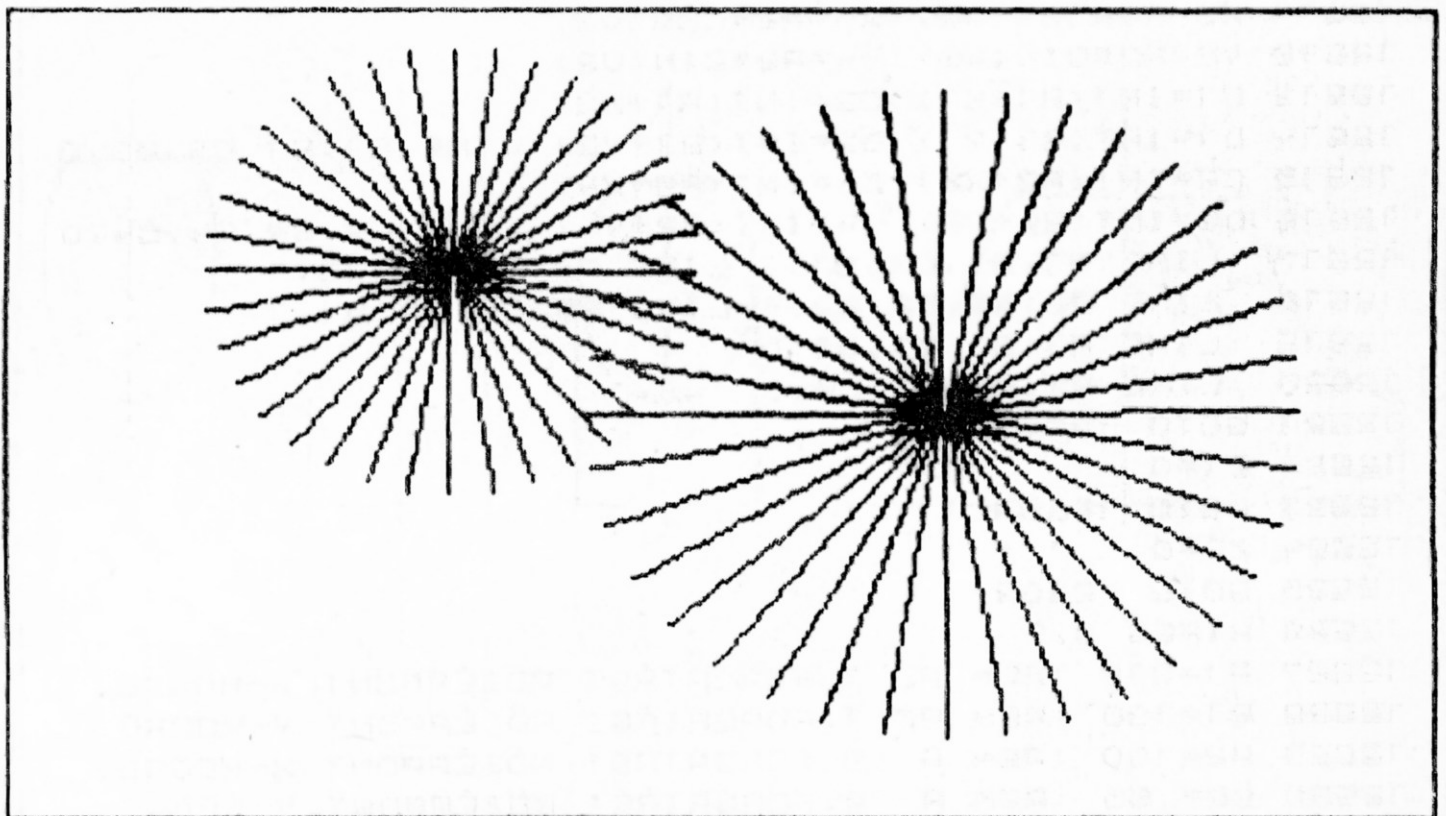
2.5.5. MOZGÁSOK

A MOZGÁS program megmozgatja a képernyőt.

Mozgásokat a sprite-okkal is csinálhatunk. De az nem mindenki számára hozzáférhető, és a mi területünkön nem igazán alkalmas. Ezért azt szeretnénk bemutatni, hogyan lehet másképpen dinamikus folyamatokat előállítani.

A1-gyel az első forgatási középpont X koordinátáját adjuk meg, B1 az első forgatási középpont Y koordinátája.

A2 és B2 a második forgatási középpont koordinátái.



63. ábra. Két forgási középpont. A képernyőn két sugár mozgását láthatjuk.

R1 az első forgó kar hossza, R2 a második forgó kar hossza.

F1-gyel a lépésközt adjuk meg, és F1 előjelével a forgatási irányt. F2-vel ugyanezt adjuk meg a második forgási középpontra.

A 63. ábra mutatja be, hogy mi történik a képernyőn mozgáskor. Egy nyomtatott képen nem lehet a mozgást ábrázolni, de el lehet képzelni a mozgás elvét: A dinamikus folyamat úgy megy végbe a monitoron, hogy először egy vonalat rajzolunk. Valamivel mellette rajzolunk egy újabb vonalat. Ezután az első vonalat egy kellemesnek ítélt idő után letöröljük. Mielőtt a második vonalat letöröljük, valamivel odébb újra rajzolunk egy harmadik vonalat, és csak ezután töröljük le a másodikat. Ez úgy történik, mint a filmen. Ha nem hagynánk állandóan legalább egy vonalat a képernyőn, akkor kellemetlen villogás állna elő. Ennek a hatásnak az eléréséhez fontos azoknak a programsoroknak az elhelyezése, amelyek a vonal rajzolásáért és törléséért felelősek. Nézzék meg pontosan, hol állnak a \emptyset , ill. az 1 rajzolási modu LINE parancsok!

```
12600 REM MOZGÁS
12601 GOTO 12626
12602 REM MOZGÁS ALPROGRAM
12603 Z1=Z1+F1
12604 Z2=Z2+F2
12605 IF Z1>360 THEN 12622
12606 IF Z2>360 THEN 12624
12607 W1=Z1*0.01745:U1=(Z1+F1)*0.01745
12608 W2=Z2*0.01745:U2=(Z2+F2)*0.01745
12609 X1=R1*COS(W1):X3=R1*COS(U1)
12610 Y1=R1*SIN(W1):Y3=R1*SIN(U1)
12611 X2=R2*COS(W2):X4=R2*COS(U2)
12612 Y2=R2*SIN(W2):Y4=R2*SIN(U2)
12613 C1=INT(A1+X1):C3=INT(A1+X3)
12614 D1=INT(B1+Y1):D3=INT(B1+Y3):LINE A1,B1,C3,D3,0
12615 C2=INT(A2+X2):C4=INT(A2+X4)
12616 D2=INT(B2+Y2):D4=INT(B2+Y4):LINE A2,B2,C4,D4,0
12617 :LINE A1,B1,C1,D1,1:LINE A1,B1,C3,D3,1
12618 :LINE A2,B2,C2,D2,1:LINE A2,B2,C4,D4,1
12619 :LINE A1,B1,C1,D1,0
12620 :LINE A2,B2,C2,D2,0
12621 GOTO 12603
12622 Z1=0
12623 GOTO 12603
12624 Z2=0
12625 GOTO 12603
12626 HIRRES 0,7
12627 A1=210 :REM AZ 1.FORGATÁSI KÖZÉPPONT X-KOORD.
12628 B1=100 :REM AZ 1.FORGATÁSI KÖZÉPPONT Y-KOORD.
12629 A2=100 :REM A 2.FORGATÁSI KÖZÉPPONT X-KOORD.
12630 B2= 65 :REM A 2.FORGATÁSI KÖZÉPPONT Y-KOORD.
12631 R1= 80 :REM AZ 1.FORGATÁS SUGARA
12632 R2= 55 :REM A 2.FORGATÁS SUGARA
```

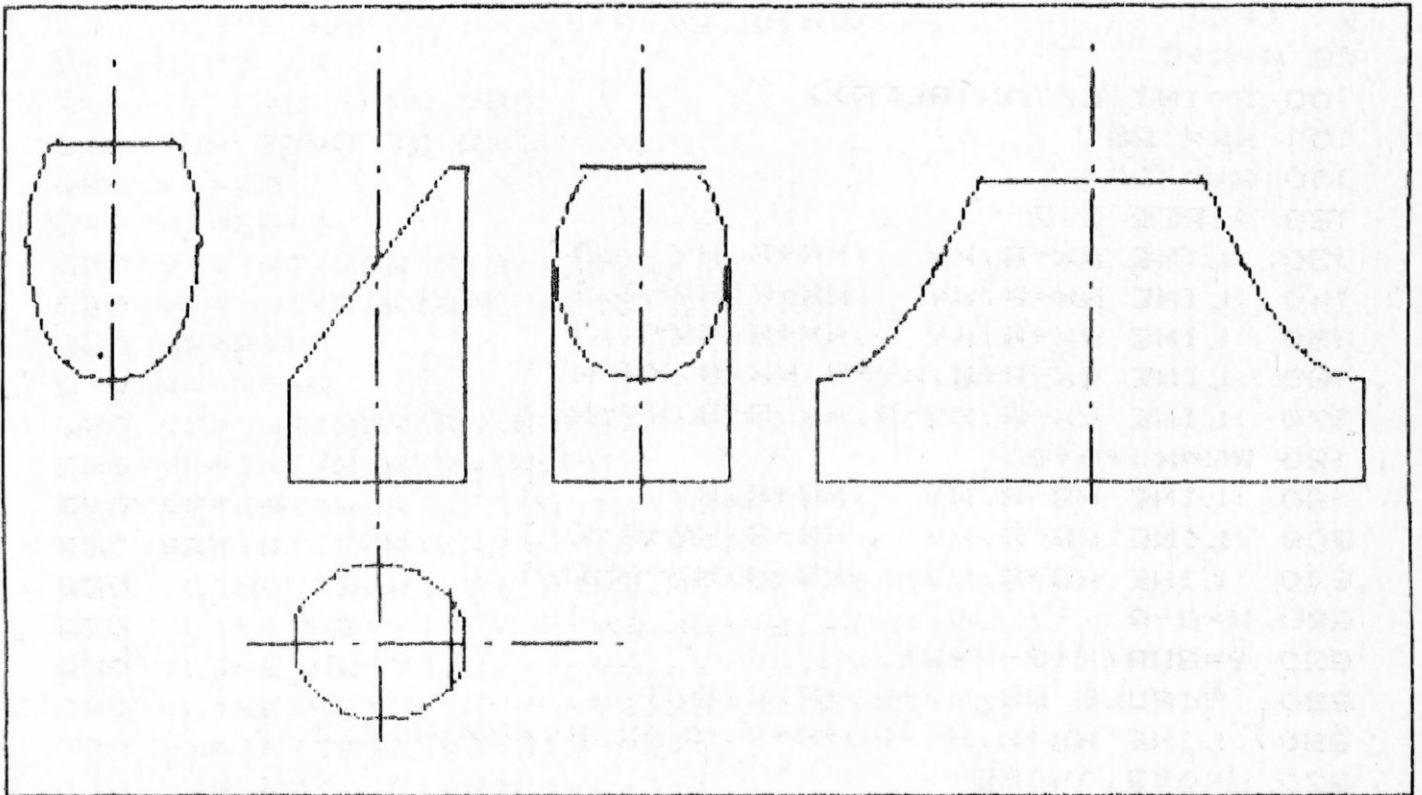
```
12633 F1= 10 :REM AZ 1.FORGATÁS LÉPÉSKÖZE
12634 F2=-10 :REM A 2.FORGATÁS LÉPÉSKÖZE
12635 GOTO 12602
```

2.5.6. ÁRNYÉKOLÁS ÉS SZÍNEZÉS

Szükségünk lehet rá, hogy egyes rajzokat árnyékoljunk vagy bizonyos felületeket beszínezzünk, vagy mint egy mozaikot színes pontokkal jelöljük meg. Ez a technika inkább a grafikához, és nem a műszaki rajzhoz tartozik, ezért csak érintjük ezt a témát. Nagyon korlátozottak a lehetőségeink az árnyékolásokra, mert a nyomtatókon csak nagyon durva raszter áll rendelkezésünkre. A nyomdatechnikában egy tónust kaphatunk, ha bizonyos távolságokra pontokat kigyújtunk, ill. nyomtatunk. Minél sötétebb felületet akarunk elérni, annál több pontot kell felületegységenként kigyújtani. Színesben még a színárnyalatokkal is lehet variálni. Olyan automatizmus-sal, ami a pontokat kigyújtja, nem rendelkezünk, bár itt is fel lehet használni a matematika segítségét. Ez nagyon sok munkát igényel, és a ráfordítás nem térül meg. Pl. a CERUZA program segítségével egyes pontokat kigyújthatunk, de ez nagyon fáradtságos munka.

2.5.7. METSZETEK

A METSZÉS program a forgástengelyhez képest ferdén metszett hengert rajzol fel, mindhárom nézetben. A metszett felületet és az elmetszett henger kiterített palástját is valódi nagyságban ábrázolja.



64. ábra. A tengelyéhez képest ferdén elmetszett henger, a metszett felület valódi méretével és a kiterített palásttal.

Ez a program nem tartozik az alapprogramjainkhoz, hanem ez az alapprogramok felhasználása. Azért mutatjuk be, hogy láthassuk programjainkkal ilyen is lehet csinálni. Az ilyen feladatokat nem érdemes általánosan programozni. Ez a CAD egy speciális ága lenne, ami az ábrázoló geometria összes szabályát figyelembe venné. Ha ilyen ambícióink vannak, a programkönyvtárunk ehhez is hasznos segítséget nyújthat. Ezen az úton azonban egyedül kell továbbmennünk. Ugyanez igaz, ha a különböző testek áthatásaira van szükség. Pontosan szólva a metszetek is az áthatások speciális esetét jelentik, ahol nem rajzoljuk fel a testeket, csak az érintkezési felületüket.

```

10 REM METSZÉS
20 D=40
25 R=D/2
30 H=90
40 A=26
50 SZOG=45
60 ALFA=(3.14159/180)*SZOG
70 C=H-A
80 B=INT(C/TAN(ALFA))
90 E=INT(R*TAN(ALFA))
91 IF E/C<R/D THEN 93
92 GOTO 96
93 B=D
94 C=INT(B*TAN(ALFA))
95 H=A+C
96 IF E=>C THEN 98
97 GOTO 101
98 E=C
99 H=A+C
100 B=INT(E/TAN(ALFA))
101 KX= 85
110 KY=120
120 HIRIS 0,7
130 :LINE KX-R,KY ,KX+R,KY ,1
140 :LINE KX-R,KY ,KX-R,KY-A,1
150 :LINE KX+R,KY ,KX+R,KY-H,1
160 :LINE KX-R+B,KY-H,KX+R,KY-H,1
170 :LINE KX-R,KY-A,KX-R+B,KY-H,1
180 KS=KX+D+20
190 :LINE KS-R,KY ,KS+R,KY ,1
200 :LINE KS-R,KY ,KS-R,KY-A-E,1
210 :LINE KS+R,KY ,KS+R,KY-A-E,1
220 X=B-R
230 Y=SQR(R^2-X^2)
250 :CIRCLE KX,KY+20+R,R,R,1
260 :LINE KX+X,KY+20+R+Y,KX+X,KY+20+R-Y,1
270 U=D*3.14159
280 UH=INT((D*3.14159)/2)
290 UV=INT(U/4)

```



```

300 KM=KX+R+20+D+20+UH
310 :LINE KM-UH,KY,KM-UH,KY-A,1
320 :LINE KM+UH,KY,KM+UH,KY-A,1
330 KW=KX-D-20
331 IF E=C THEN 333
332 GOTO 350
333 FOR Z1=R TO ABS(X)+1 STEP -1
334 :LINE KS-A,KY-H,KS+A,KY-H,1
335 GOTO 360
350 FOR Z1=R TO 1 STEP -1
360 X1=Z1
370 X2=Z1-1
380 Y1=INT(SQR(R↑2-X1↑2))
390 Y2=INT(SQR(R↑2-X2↑2))
400 X3=R-X1
410 X4=R-X2
420 Y3=INT(X3*TAN(ALFA))
430 Y4=INT(X4*TAN(ALFA))
440 Y5=INT(SQR(X3↑2+Y3↑2))
450 Y6=INT(SQR(X4↑2+Y4↑2))
460 :LINE KS-Y1,KY-A-Y3,KS-Y2,KY-A-Y4,1
470 :LINE KS+Y1,KY-A-Y3,KS+Y2,KY-A-Y4,1
480 :LINE KW-Y1,KY-A-Y5,KW-Y2,KY-A-Y6,1
490 :LINE KW+Y1,KY-A-Y5,KW+Y2,KY-A-Y6,1
500 MX=INT(SQR(X3↑2+Y1↑2))
510 NX=INT(SQR(X4↑2+Y2↑2))
520 :LINE KM-UH+MX,KY-A-Y3,KM-UH+NX,KY-A-Y4,1
525 :LINE KM-UH,KY,KM+UH,KY,1
530 :LINE KM+UH-MX,KY-A-Y3,KM+UH-NX,KY-A-Y4,1
540 NEXT Z1
541 IF E=C THEN 760
550 FOR Z2=0 TO X-1
560 X1=Z2
570 X2=Z2-1
580 Y1=INT(SQR(R↑2-X1↑2))
590 Y2=INT(SQR(R↑2-X2↑2))
600 X3=R+X1
610 X4=R+X2
620 Y3=INT(X3*TAN(ALFA))
630 Y4=INT(X4*TAN(ALFA))
640 Y5=INT(SQR((X1+R)↑2+Y3↑2))
650 Y6=INT(SQR((X2+R)↑2+Y4↑2))
660 :LINE KS-Y1,KY-A-Y3,KS-Y2,KY-A-Y4,1
670 :LINE KS+Y1,KY-A-Y3,KS+Y2,KY-A-Y4,1
680 :LINE KW-Y1,KY-A-Y5,KW-Y2,KY-A-Y6,1
690 :LINE KW+Y1,KY-A-Y5,KW+Y2,KY-A-Y6,1
700 MX=INT(SQR(X3↑2+Y1↑2))
710 NX=INT(SQR(X4↑2+Y2↑2))
720 :LINE KM-UH+MX,KY-A-Y3,KM-UH+NX,KY-A-Y4,1
730 :LINE KM+UH-MX,KY-A-Y3,KM+UH-NX,KY-A-Y4,1

```

```

740 IF X2=X THEN 760
750 NEXT Z2
760 :LINE KM-UH+NX,KY-R-Y4,KM+UH-NX,KY-R-Y4,1
761 :LINE KS-Y2,KY-H,KS+Y2,KY-H,1
762 :LINE KW-Y2,KY-R-Y6,KW+Y2,KY-R-Y6,1
770 BX=KW
780 BY=KY-R+5
790 FIN=60
800 GOSUB 980
810 BX=KX
820 BY=KY+20+D+5
830 FIN=150
840 GOSUB 980
850 BX=KS
860 BY=KY+5
870 FIN=90
880 GOSUB 980
890 BX=KM
900 BY=KY+5
910 FIN=90
920 GOSUB 980
930 BX=KX-R-10
940 BY=KY+20+R
950 FIN=60
960 GOSUB 1060
970 GOTO 1140
980 REM FÜGGŐLEGES TENGELYVONAL
990 FOR Z3=0 TO FIN STEP 30
1000 :LINE BX,BY-Z3,BX,BY-Z3-20,1
1010 :LINE BX,BY-Z3-20,BX,BY-Z3-24,0
1020 :LINE BX,BY-Z3-24,BX,BY-Z3-26,1
1030 :LINE BX,BY-Z3-26,BX,BY-Z3-30,0
1040 NEXT Z3
1050 RETURN
1060 REM VÍZSZINTES TENGELYVONAL
1070 FOR Z4=0 TO FIN STEP 30
1080 :LINE BX+Z4,BY,BX+Z4+20,BY,1
1090 :LINE BX+Z4+20,BY,BX+Z4+24,BY,0
1100 :LINE BX+Z4+24,BY,BX+Z4+26,BY,1
1110 :LINE BX+Z4+26,BY,BX+Z4+30,BY,0
1120 NEXT Z4
1130 RETURN
1140 GOTO 1140
1150 OPEN 1,4
1160 PRINT#1,"FERDÉN ELMETSZETT HENGER"
1170 PRINT#1
1180 PRINT#1
1190 PRINT#1,"VALÓDI NAGYSÁGÚ METSZETFELÜLETTEL"
1200 PRINT#1,"ÉS KITERÍTETT PALÁSTTAL"
1210 PRINT#1

```

```

1220 PRINT#1
1230 PRINT#1
1240 PRINT#1
1250 PRINT#1,"AZ ÖSSZES VÁZLATOT A SZÁMÍTÓGÉP SZÁMÍTJA"
1260 PRINT#1
1270 PRINT#1
1280 PRINT#1
1290 PRINT#1
1300 PRINT#1,"ADATBEVITEL : "
1310 PRINT#1,SPC(15),"ÁTMÉRŐ"
1320 PRINT#1,SPC(15),"MAGASSÁG"
1330 PRINT#1,SPC(15),"METSZŐSÍK SZÖGE"

```

2.5.8. A FINOMFELBONTÁSÚ GRAFIKA TÁROLÁSA MÁGNESLEMEZEN

Eddig nem voltunk abban a helyzetben, hogy a finomfelbontású képernyő képét, vagyis a HIREs képet vagy ennek részeit külső tárolóra, mágneslemezre rögzíthettük volna.

A KÉPTÁROLÁS program lehetővé teszi, hogy a képernyő tetszőleges területét mágneslemezre tároljuk. Ha elegendő időt szánunk rá, akkor részletekben az egész képernyő tartalmát is elmenthetjük.

A program közvetlen párbeszédés formában működik. A HIREs programnál a képernyő alsó szélén kérdések jelennek meg, amelyekre válaszolni kell. A válaszok nem jelennek meg a finomfelbontású képernyőn. Ha ezt is meg szeretnénk oldani, akkor még néhány TEXT parancsot el kell helyezni a programban. Ha a programot elindítjuk, a következő kérdés jelenik meg:

TÁROLÁS? I/N.

Válaszoljunk I-vel (igen) vagy N-nel (nem), és nyomjuk meg a RETURN gombot.

Ezután a szöveget letöröljük, de a rajz esetleg felülírt részét nem. Ezután a program a következő kérdést teszi fel: N válasz esetén megkérdezi, hogy akarjuk-e ismételni a folyamatot. Ha itt is nemmel válaszolunk, akkor a program befejeződik. A VÉGE felirat 10s után eltűnik, és folytathatjuk a munkát egy másik programmal.

A program futásának érdekesebb ága az, amikor I-vel válaszolunk. Ezután az ÁLLOMÁNYNÉV? kérdés jelenik meg, annak az adatállománynak (file-nak) a nevét kell megadni, amelyben tárolni kívánjuk a képmező tartalmát. A választott név megadása után nyomjuk meg a RETURN gombot.

A SAROK X?, Y? kérdés következik. Válaszképpen a tárolandó mező bal felső sarokpontjának X és Y koordinátáit kell megadni. Figyeljük meg, hogy az X és Y koordinátákat egyszerre kérdezi, de mind az X, mind az Y koordináta megadása után meg kell nyomni a RETURN gombot.

Ha válaszoltunk, újabb kérdés jelenik meg:

SZÉLESSÉG?, MAGASSÁG?

Adjuk meg a tárolandó felület szélességét és magasságát! (RETURN).

Ezután a képernyőn megjelenik a

LETAPOGATÁS felirat.

Ez azt jelenti, hogy a számítógép a tárolandó felületet pontról-pontra letapogatja, és tárolja a világító pontokat. Ez annál tovább tart, minél nagyobb felületet választottunk.

Emlékezzünk arra, amit a 2.5.1. pontban mondtunk a felületek nagyságáról!

A talált képpontokat a számítógép akkor írja fel, a mágneslemezre, ha megjelenik a TÁROLÁS felirat. Ezt arról is felismerjük, hogy a mágneslemez meghajtó egység működésbe lép.

Ha készen van a számítógép, akkor az

ISMÉTLÉS? I/N

kérdéssel jelentkezik. Ha I-vel válaszoltunk, az egész folyamat előlről kezdődik. N esetén a program befejeződik és a VÉGE felirat jelenik meg. 10 s-mal később folytathatjuk a munkát egy másik programmal.

A FILE-számot és az ID-számot a program 2-re állítja be. Ez természetesen az igényeknek megfelelően változtatható. Az is lehetséges, hogy újabb kérdéseket iktathatunk be, amelyekben ezeket az értékeket kérdezzük meg.

A KÉPTÁROLÁS programmal tehát olyan felületrészeket tárolhatunk, amelyekre később szükségünk lesz. Ezzel nagyon fontos eszközt kaptunk, amelynek egyetlen hibája, hogy egy kicsit lassú.

```
12700 REM KÉPTÁROLÁS
12701 GOTO 12757
12702 REM KÉPTÁROLÁS ALPROGRAM
12703 K=0
12704 :TEXT 10,190,"TÁROLÁS? I/N",2,1,B
12705 INPUT A$
12706 :TEXT 10,190,"TÁROLÁS? I/N",2,1,B
12707 IF A$="N" THEN 12742
12708 :TEXT 10,190,"ÁLLOMÁNYNÉV?",2,1,B
12709 INPUT A$
12710 :TEXT 10,190,"ÁLLOMÁNYNÉV?",2,1,B
12711 :TEXT 10,190,"SAROK X?, Y?",2,1,B
12712 INPUT A,B
12713 :TEXT 10,190,"SAROK X?, Y?",2,1,B
```

```

12714 :TEXT 10,190,"SZÉLESSÉG?, MAGASSÁG?",2,1,B
12715 INPUT C,D
12716 :TEXT 10,190,"SZÉLESSÉG?, MAGASSÁG?",2,1,B
12717 :TEXT 10,190,"LETAPOGATÁS" ,2,1,B
12718 G=C*D:Y=B:K=0
12719 DIM H<G>:DIM M<G>
12720 FOR I=1 TO D
12721 X=A
12722 Y=Y+1
12723 FOR J=1 TO C
12724 X=X+1
12725 L=TEST<X,Y>:PLOT X,Y,2:PLOT X,Y,2
12726 IF L=1 THEN 12728
12727 GOTO 12731
12728 K=K+1
12729 H<K>=X
12730 M<K>=Y
12731 NEXT J
12732 NEXT I
12733 :TEXT 10,190,"LETAPOGATÁS" ,2,1,B
12734 :TEXT 10,190,"TÁROLÁS",2,1,B
12735 :OPEN 2,B,2,A$+"S,W"
12736 PRINT#2,A;CHR$(13)B;CHR$(13)C;CHR$(13)D;CHR$(13)K
12737 FOR N=1 TO K
12738 PRINT#2,H<N>;CHR$(13)M<N>
12739 NEXT N
12740 CLOSE 2
12741 :TEXT 10,190,"TÁROLÁS",2,1,B
12742 REM ISMÉTLÉS
12743 :TEXT 10,190,"ISMÉTLÉS? I/N",2,1,B
12744 INPUT B$
12745 :TEXT 10,190,"ISMÉTLÉS? I/N",2,1,B
12746 IF B$="I" THEN 12751
12747 :TEXT 10,190,"VÉGE",2,1,B
12748 PAUSE 10
12749 :TEXT 10,190,"VÉGE",2,1,B
12750 GOTO 12755
12751 CLR
12752 GOSUB 12702
12753 GOTO 12760
12754 CLR
12755 RETURN
12756 CLR
12757 HIRES 0,7
12758 :TEXT 148,113,"S",1,1,B
12759 GOSUB 12702
12760 END

```

2.5.9. A FINOMFELBONTÁSÚ GRAFIKA BETÖLTÉSE MÁGNESLEMEZRŐL

Természetesen olyan programra is szükségünk van, ami a mágneslemezen tárolt HIRES képeket visszatölti. Ez a KÉPBETÖLTÉS program feladata, amely szintén egy párbeszédés program.

A képernyőn a BETÖLTÉS? I/N és az ÁLLOMÁNYNÉV? kérdés jelenik meg. A kérdéseknek megfelelően válaszoljunk. Ehhez ismerni kell azt a file-nevet, amiben képet tárolunk. Ha a katalógusban megnézzük a file-neveket láthatjuk, hogy a megadott file-név után egy titokzatos S betű került. Ez csak azt jelzi, hogy szekvenciális file-ról van szó. Ezt az S betűt nem kell beírni amikor a file-nevet megadjuk. Különben minden a tároláshoz hasonlóan történik.

A RAJZOLÁS? I/N kérdésre ha N-nel válaszolunk, akkor az ISMÉTLÉS kérdés következik, ha I-vel válaszolunk, akkor a tárolt kép a képernyőn megjelenik úgy, hogy a kigyújtott pontok hozzátevéődnek a képernyőn levő képhez. Így rárajzolhatunk a rajzra egy másikat, csupán a programok megfelelő láncolásáról kell gondoskodnunk.

Erről a módszerről a 4. Fejezetben lesz szó.

```
12800 REM KÉPBETÖLTÉS
12801 GOTO 12847
12802 REM KÉPBETÖLTÉS ALPROGRAM
12803 :TEXT 10,190,"BETÖLTÉS? I/N",2,1,8
12804 INPUT A$
12805 :TEXT 10,190,"BETÖLTÉS? I/N",2,1,8
12806 IF A$="N" THEN 12827
12807 :TEXT 10,190,"ÁLLOMÁNYNÉV?",2,1,8
12808 INPUT A$
12809 :TEXT 10,190,"ÁLLOMÁNYNÉV?",2,1,8
12810 :TEXT 10,190,"BEOLVASÁS",2,1,8
12811 :OPEN 2,8,2,A$+",S,R"
12812 INPUT#2,A,B,C,D,K
12813 G=C*D
12814 DIM H(G):DIM M(G)
12815 FOR N=1 TO K
12816 INPUT#2,H(N),M(N)
12817 NEXT N
12818 CLOSE 2
12819 :TEXT 10,190,"BEOLVASÁS",2,1,8
12820 :TEXT 10,190,"RAJZOLÁS? I/N",2,1,8
12821 INPUT B$
12822 :TEXT 10,190,"RAJZOLÁS? I/N",2,1,8
12823 IF B$="N" THEN 12827
12824 FOR N=1 TO K
12825 :PLOT H(N),M(N),1
12826 NEXT N
12827 REM ISMÉTLÉS
12828 :TEXT 10,190,"ISMÉTLÉS? I/N",2,1,8
```

```
12829 INPUT B$
12830 :TEXT 10,190,"ISMÉTLÉS? I/N",2,1,8
12831 IF B$="I" THEN 12836
12832 :TEXT 10,190,"VÉGE",2,1,8
12833 PAUSE 10
12834 :TEXT 10,190,"VÉGE",2,1,8
12835 GOTO 12839
12836 CLR
12837 GOSUB 12802
12838 GOTO 12843
12839 RETURN
12840 CLR
12841 HIRES 0,7
12842 GOSUB 12802
12843 END
```

3. FEJEZET

MIRE HASZNÁLHATJUK A CAD LEHETŐSÉGEIT?

Ezt a kérdést nem lehet egyszerűen megválaszolni, mert annyi felhasználási lehetőség van, ahány számítógép-használó.

Némi fantáziával feladataink megoldásához legalább a CAD egy részét segítségül hívhatjuk. Ezért, inkább csak gondolatébresztőül, szeretnénk néhány felhasználási lehetőséget példaként bemutatni.

A CAD legfontosabb felhasználási területei talán a következők:

- számítások,
- műszaki leírások,
- tervváltozatok kidolgozása,
- ajánlatok mellékletei,
- szervizutasítások,
- alkatrészlisták,
- formatervezés,
- elektronika.

3.1. Számítások és műszaki leírások

Már egészen a könyv elején írtuk, hogy a számítások is a CAD körébe tartoznak. Ezt a feladatot ezidáig nagyon elhanyagoltuk, pedig a számítógépek mindennapos használatában nagyon fontosak.

Azért, hogy bemutassuk ezen a területen mire képes a számítógép, példaként egy szilárdságtani számítást ismertetünk.

Aki ismeri ezt a szakterületet, az tudja miről van szó. Ezt most nem szeretnénk részletezni, hanem inkább arra szeretnénk felhívni a figyelmet, hogy melyek a számítógépek erősségei a számítás, a rajzolás és a szövegekombináció területén. A számítógép kiszámítja amit megrajzolt, megrajzolja amit kiszámított, és még magyarázó szöveget is ír hozzá. A bonyolult számításokat annyira automatizálja, hogy még egy

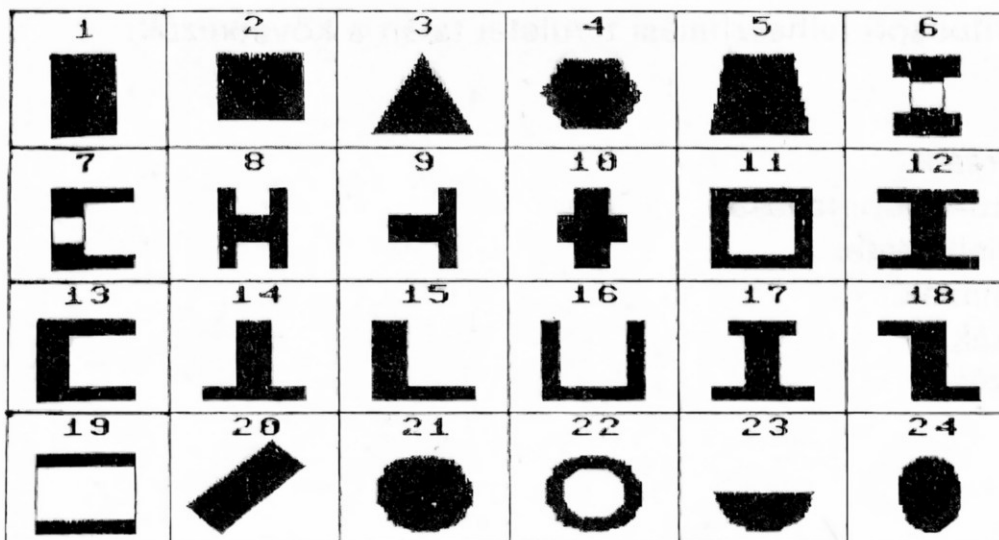
gyakorlatlan is el tudja vele végeztetni. Általánosan érvényes képletet használ, ez alapján gyorsan elvégzi a számításokat, így sok változat ellenőrzését lehetővé teszi.

Egy ilyen számítás programját összeállítani sok munkát jelent. De ha egyszer elkészítettük, akkor nagyon sok munkát takaríthatunk meg, és hihetetlenül rugalmasak és gyorsak leszünk.

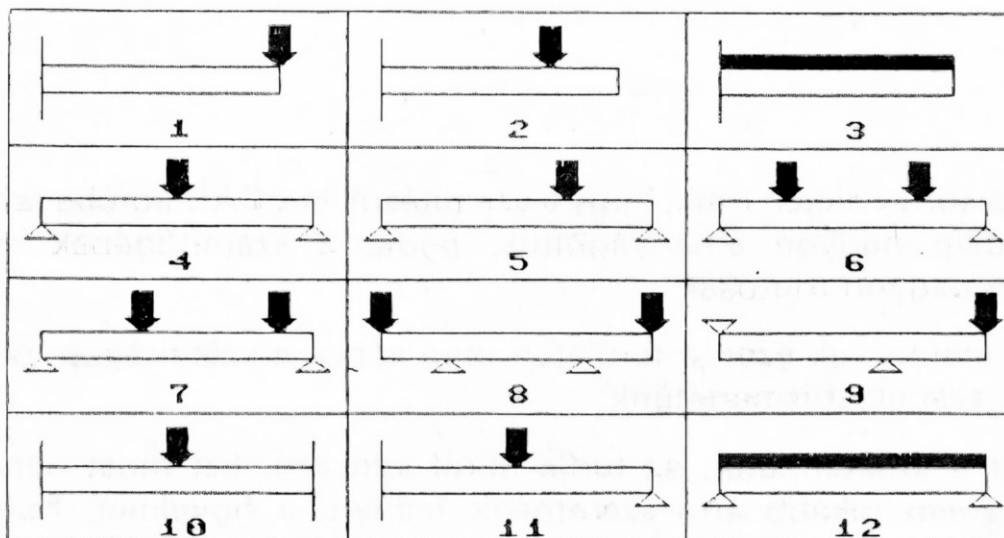
Az előbbieket mindenféle számítógép-alkalmazásra igazak.

Talán még egy különlegességre felhívhatjuk a figyelmet: A műszaki számításokban és leírásokban gyakran előfordulnak különleges jelek. Az 1/C példánkban a szigma és gamma görög betűk. A nyomtatónk lehetővé teszi ilyen különleges jeleknek az összeállítását is.

1/A példa. Egyszerű keresztmetszetek katalógusa



1/B példa. Egyszerű terhelési esetek katalógusa

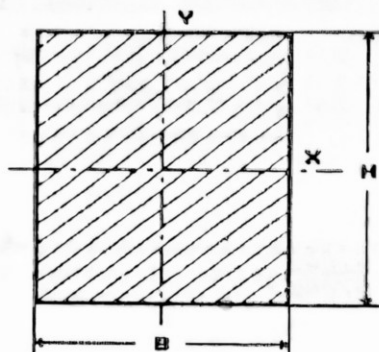


1/C példa. Szilárdsági számítás

 1/C PÉLDA: A KERESZTMETSZET KIVÁLASZTÁSA

 1. LÉPÉS

1. KERESZTMETSZET



INERCIANYOMATÉK

$$I_X = B \cdot H^3 / 12$$

$$I_Y = H \cdot B^3 / 12$$

KERESZTMETSZETI TÉNYEZŐ

$$W_X = B \cdot H^2 / 6$$

$$W_Y = H \cdot B^2 / 6$$

 A KERESZTMETSZET MÉRETEINEK MEGADÁSA:

 2. LÉPÉS

$$B = 4.20 \text{ CM}$$

$$H = 5.60 \text{ CM}$$

 A KERESZTMETSZET ADATAI:

 3. LÉPÉS

INERCIANYOMATÉK	KERESZTMETSZETI
[CM ⁴]	TÉNYEZŐ
	[CM ³]

$$I_X = 87.4656$$

$$I_Y = 34.5744$$

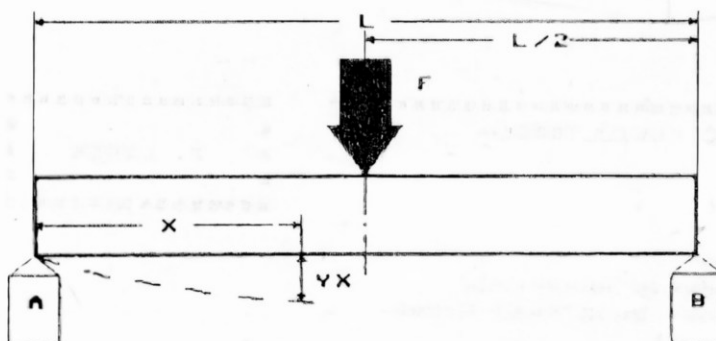
$$W_X = 27.852$$

$$W_Y = 19.464$$

 A TERHELÉSI ESET KIVÁLASZTÁSA:

 4. LÉPÉS

4. TERHELÉSI ESET



ÁLLANDÓ KERESZTMETSZETŰ KÉTTÁMASZÚ TARTÓ.

A KONCENTRÁLT ERŐ HELYE: $x=L/2$

A TÁMASZERŐK: $A=B=F/2$

A NYOMATÉK X TÁVOLSÁGRA: $M=(F \cdot L/2) \cdot (x/L)$

A MAXIMÁLIS NYOMATÉK: $M=F \cdot L/4$

A LEHAJLÁSFÜGGVÉNY:

$y_x = ((F \cdot L^3) / (16 \cdot E \cdot I)) \cdot (x/L) \cdot (1 - (4 \cdot x^2 / (3 \cdot L^2)))$; $x=L/2$

A MAXIMÁLIS LEHAJLÁS: $y_x = (F \cdot L^3) / (4 \cdot 8 \cdot E \cdot I)$

A SZERKEZET PARAMÉTEREINEK MEGADÁSA:

5. LÉPÉS

TÁMASZKÖZ
ERŐ

L = 50 CM
F = 2000 N

RUGALMASSÁGI MODULUS
HAJLÍTÓ HATÁRFESZÜLTÉS

E = 21000000 N/CM²
σ_H = 4800 N/CM²

BIZTONSÁGI TÉNYEZŐ

K = 3

FAJSÚLV

γ = 7.85E-03 N/CM³

A SZÁMÍTOTT ÉRTÉKEK:

6. LÉPÉS

A HAJLÍTÁS TENGELE X
LEHAJLÁSÁBRA Y_X MAX=4.03503542E-03 CM



NYOMATÉKÁBRA M MAX=25000 NCM



A HAJLÍTÁS TENGELE Y
LEHAJLÁSÁBRA Y_X MAX=7.17339831E-03 CM



NYOMATÉKÁBRA M MAX=25000 NCM



A HAJLÍTÁSÓL SZÁRMAZÓ FESZÜLTÉSEK:

7. LÉPÉS

A HAJLÍTÁS TENGELE: X

A KÉPLET: $\sigma_x = (M \cdot MAX \cdot K) / W_x$
SZÁMÍTOTT ÉRTÉK: $\sigma_x = 3476.5451 \text{ N/CM}^2$

A HAJLÍTÁS TENGELE: Y

A KÉPLET: $\sigma_y = (M \cdot MAX \cdot K) / W_y$
SZÁMÍTOTT ÉRTÉK: $\sigma_y = 4555.3935 \text{ N/CM}^2$

A SZILÁRSÁGI FELTÉTEL ELLENÖRZÉSE:

8. LÉPÉS

A SZILÁRSÁGI FELTÉTELT KIELÉGÍTJÜK, HA
 $\sigma < \sigma_H$

A HAJLÍTÁS TENGELE: X
 $\sigma_X = 2476.5457 < \sigma_H = 4500 \text{ N/CM}^2$
SZILÁRSÁGI SZEMPONTBÓL MEGFELEL !

A HAJLÍTÁS TENGELE: Y
 $\sigma_Y = 4555.3935 > \sigma_H = 4500 \text{ N/CM}^2$
SZILÁRSÁGI SZEMPONTBÓL NEM FELEL MEG !

AZ ADATOK ÉS EREDMÉNYEK ÖSSZEZEGÉSE:

9. LÉPÉS

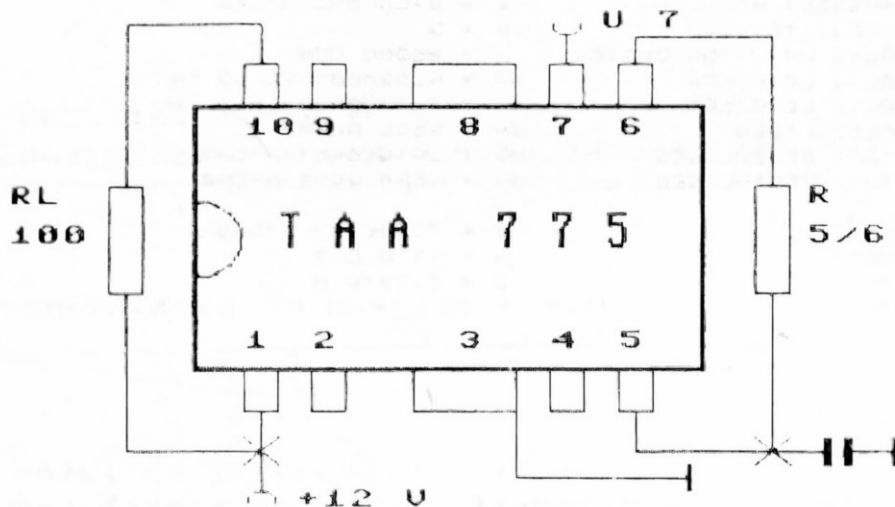
SZÉLESSÉG	B = 4.2 CM
MAGASSÁG	H = 3.8 CM
TÁMASZKÖZ	L = 50 CM
INERCIANYOMATÉK	IX = 67.4656 CM ⁴
INERCIANYOMATÉK	IY = 34.5744 CM ⁴
KERESZTMETSZETI TÉNVEZŐ	WX = 21.952 CM ³
KERESZTMETSZETI TÉNVEZŐ	WY = 16.464 CM ³
ERŐ	F = 2000 N
RUGALMASSÁGI MODULUS	E = 21000000 N/CM ²
BIZTONSÁGI TÉNVEZŐ	K = 3
MAXIMÁLIS HAJLÍTÓNYOMATÉK	M = 25000 NCM
MAXIMÁLIS LEHAJLÁS	VX = 4.03503542E-03 CM
MAXIMÁLIS LEHAJLÁS	VY = 7.17338831E-03 CM
HATÁRFESZÜLTÉG	$\sigma_H = 4500 \text{ N/CM}^2$
SZÁMÍTOTT FESZÜLTÉG	$\sigma_X = 2476.5457 \text{ N/CM}^2$
SZÁMÍTOTT FESZÜLTÉG	$\sigma_Y = 4555.3935 \text{ N/CM}^2$
FAJSÚLV	$\gamma = 7.93E-03 \text{ N/CM}^3$
TÉRFOGAT	U = 1176 CM ³
SÚLV	G = 9.2316 N

3.2. Elektronikai rajzok és áramkörtervek

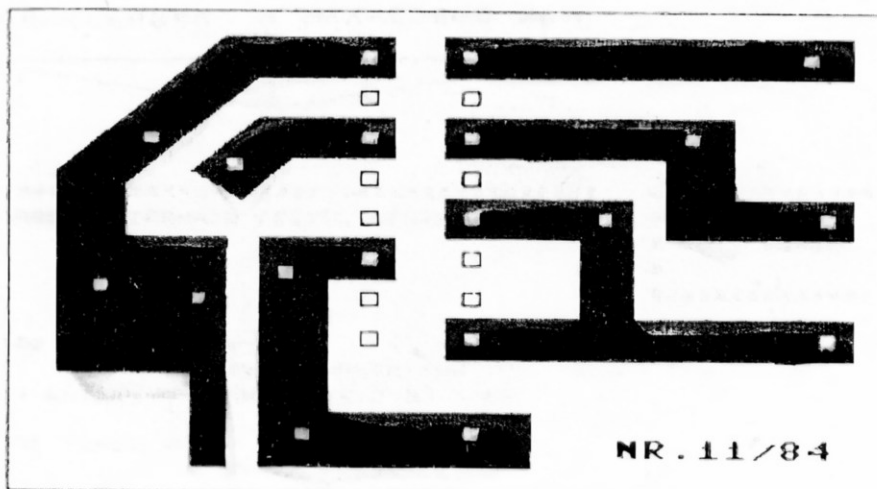
Eddigi példáinkat főleg a mechanika területéről hoztuk. A makróink is a mechanika területén alkalmazhatók a legjobban.

Az alapprogramokat azonban jól használhatjuk pl. a villamosságtan és az elektronika területén alkalmazható makrók összeállításához is. Kisebb makrókat készíthetünk pl. a tranzisztorok, ellenállások, kondenzátorok rajzolásához. Nagyobb makrókkal IC-áramköröket rajzoltathatunk. A módszer nem különbözik attól, amit a mechanika területén bemutatunk. Csak a megjelenésük más, mert az a feladat, amit ábrázolni akarunk szintén különbözik.

2. példa. A makrókat mindig az adott feladathoz alakíthatjuk, itt például az elektronikához



3. példa. Nyomatott áramkör terve



Egy elektronikai rajzot mutat be a 2. példa. A 3. példa a számítógépnek az elektronikában való felhasználására egy másik lehetőséget mutat. Ha a vezetősávokat nem választjuk túl vékonyra, akkor áramkörterveket is készíthetünk. Ennek az az előnye, hogy a változtatásokat nagyon gyorsan át lehet vezetni, és azonnal láthatjuk, hogyan fog a tervezett áramkörünk kinézni.



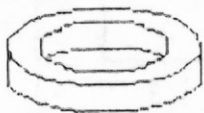
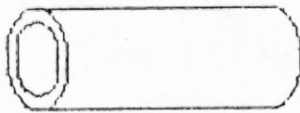
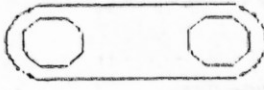
3.3. Szervizutasítások és alkatrészlisták

A számítógép különösen alkalmas listák, táblázatok, számokból és szövegekből álló kimutatások készítésére.

A CAD segítségével ezeket a táblázatokat és listákat, amelyek különösen gyakran fordulnak elő szervizutasításokban, szerelési utasításokban, alkatrészjegyzékekben, vázlatokkal és képekkel is kiegészíthetjük. Így az egész sokkal áttekinthetőbb és érthetőbb lesz. Az egyszerű vázlatok sok kiegészítő információt tartalmazhatnak, így elkerülhető, hogy egyes alkatrészeket összecseréljünk. A vevő pl. az alkatrészlistából felismerheti a szükséges alkatrészt. Ezek a listák naprakészek lehetnek. A 4. példa egy kis alkatrészjegyzéket mutat be, amit nagyon könnyen előállíthatunk a CAD segítségével.

Még sok példát mutathatnánk be, de egyszer csak be kell fejezni. A CAD-ban az a legszebb, hogy tág teret nyit a fantáziának, a felismerőkészségnek, és azonnal ki is próbálhatjuk, mire vagyunk képesek.

4. példa. Alkatrészlista. A számítógép segítségével naprakész állapotban tarthatjuk

ALKATRÉSZLISTA EL 107-5		
ÁBRA	NÉV	JEL
	ALÁTÉT	107-5-1
	HÜVELY	107-5-2
	EMELŐKAR	107-5-3

4. FEJEZET

ÚTMUTATÓ EGY CAD-RENDSZER FELÉPÍTÉSÉHEZ

Két lényeges pont jellemez egy CAD-rendszert:

Az egyik, hogy a programok feldolgozása során és a rajzoláskor minden a finomfelbontású képernyőn megy végbe. A konstruálás kezdetekor bekapcsoljuk a HIREs grafikát, és addig bekapcsolva hagyjuk, amíg a számítógéppel dolgozunk. Ez a programelemeinkre vonatkozóan azt jelenti, hogy a HIREs parancsot nem szabad használnunk. Addig, amíg egyenként futtattuk az egyes programokat, minden egyes programnál szükségünk volt a HIREs parancsra. Most azonban már zavarnak. A HIREs utasításokat könnyen hatástalaníthatjuk: egyszerűen megjegyzés sort csinálunk belőlük, pl.

```
REM HIREs Ø,7
```

Ha gondosan megnézzük azokat a programokat, amelyeket több programból szerkesztettünk, ott is találkozunk ilyen sorokkal.

A másik, hogy a számítógéppel párbeszédés üzemmódban dolgozunk. Ez azt jelenti, hogy a program sorra kéri azokat az információkat, amire szüksége van. Ilyen kérdések a finomfelbontású képernyő alsó sorában jelennek meg, és csak akkor tűnnek el, ha a billentyűzetről megadtuk azokat az információkat, amire a programnak szüksége volt, és ezekkel tovább dolgozhat. Ily módon kérdezi meg a számítógép, hogy egyáltalán milyen program alapján dolgozzon. Ezt a programot betölti a mágneslemezeiről. Előbb helyet csinál magának a tárban, ez azt jelenti, hogy az előzőleg használt programot törli. De ha egy programot a következő programmal kombinálni akarunk (MERGE), akkor ügyeljünk a programsorok megfelelő sorrendjére. Nyilvánvaló, hogy mindig csak a nagyobb sorszámú programsorokat lehet a kisebb sorszámú programsorok mögé rendezni.

A számítógép rendelkezésére kell állnia a különböző segédprogramoknak (törlés, tárolás stb.).

Ha a számítógép egy kérdésére válaszoltunk, akkor — helyes válasz esetén — a gép végrehajtja a kívánt műveleteket, majd felteszi a következő kérdést. A kérdések olyan ügyesen jelennek meg, hogy a válaszhoz szükséges legfontosabb információkat mindig közlik is. Legtöbbször csak I (igen) vagy N (nem) -mel kell válaszolni. Nem kell tehát ismerni egy számítás lefolyását vagy egy program működését. Semmit sem kell megjegyezni, csak azt kell tudni, hogy mit akarunk a számítógéppel megoldani. Az alprogramjaink szempontjából ez azt jelenti, hogy egyenként el kell tudni őket indítani, egymással kombinálhatóaknak kell lenniük, egyiknek sem szabad a másikat zavarnia, és mindegyiknek tartalmaznia kell a saját kérdéssorát.

Az első feltételeket már kielégítettük, de a kérdéssorokat még nem írtuk meg.

Ez sem okoz gondot, a programjainkat könnyen módosíthatjuk az utolsó sorok átírásával és kiegészítésével.

A RETURN utasítás után minden programelemet át kell alakítani. Oda, ahol a bemenő adatok állnak, azokat a kérdéseket kell beépíteni, amiket a programnak kell feltennie, hogy a párbeszédés program működhessen.

Ha egy programot a nevével elindíthatunk, akkor az a programra jellemző kérdésekkel jelentkezik. Ennek a programszerkezetnek az az előnye, hogy a vezérlőmenü nagyon egyszerű, kevés tárhelyet igényel, és egy program csak annyi ideig foglal helyet a belső tárban, amíg szükség van rá. Addig, amíg egy program a lemezen behívásra várakozik, nem foglal el a számítógép belső tárában helyet.

Már ismerjük, milyen feltételeket kell egy CAD-rendszernek kielégítenie. Ezek a kívánságaink, de minden kívánságért a számítógépeknél dupla árral kell megfizetnünk. Előfordulhat, hogy a menüprogramunk kivitelezésénél egyes kívánságainkról lemondunk. Csak egy kis példát akarunk bemutatni, hogyan nézhet ki egy menüprogram, ami három alapprogramot és két segédprogramot foglal rendszerbe. Egy nagyobb program ugyanígy nézne ki elveiben, csak több programból állna.

A menüprogramot CADDYMAT-nak hívjuk, és a következő utasításokkal lehet betölteni:

```
LOAD „CADDYMAT”,8
```

és RUN-nal indíthatjuk el. Ezek után a menü jelentkezik be a nevével, és a legalsó sorban megkérdezi annak a programnak a nevét, amelyet be kívánunk tölteni.

3 alapprogramot és 2 segédprogramot fogunk össze rendszerbe. A következő programnevek közül választhatunk:

```
TÉGLA  
HENGER  
KÚP  
TAROL  
OLVAS
```

Ezeket az alapprogramokat SY-nal megtoldva tároltuk lemezen pl. SYTÉGLA. Így jeleztük, hogy itt rendszerprogramokról van szó. Hogy hogyan alakítottuk át rendszerprogramokká őket, azt még később elmagyarázzuk. Ha ezek közül a programok közül választottunk, akkor azt a RETURN-nel jelezzük. Ezután jön a következő kérdés

```
LOAD? I/N
```

Ha I-t adunk meg, akkor RETURN után a kért programot mágneslemezezői a belső tárba tölti. Előzőleg azonban megjelenik:

KÉREM MÉG EGYSZER A PROGRAM NEVÉT

Bizonyára csodálkoznak, hogy meg kell adni még egyszer a program nevét. Később elmondjuk, miért is van erre szükség.

Ha beírjuk ugyanazt a nevet, a számítógép valóban betölti a programot. Ezután a betöltött program jelentkezik saját kérdéseivel. Ezt most nem kell részletesen megtárgyalnunk. Azokat a változókat kérdezi meg, amelyek a működéséhez kellenek. Csak az a fontos, hogy minden adatot RETURN-nel zárjunk le, még akkor is, ha egy sorban több változót adunk meg. Nem követhetünk el hibát, minden előre meghatározott úton halad végig. Csak akkor hibázhatunk, ha olyan értékeket adunk meg, amellyel a képernyő területén kívülre jutunk. Ekkor leáll a program.

Érdekesebb számunkra a menüprogram és a rendszerprogramoknak a kapcsolata. Ha a LOAD? kérdésre N-nel válaszolunk, akkor a gép megkérdezi:

MERGE? I/N

Ha egy programot betöltünk a belső tárba, akkor az általában minden más programot töröl, de erre még visszatérünk. A MERGE parancsnál ez másképpen van. Ebben az esetben megtartjuk a programokat, az új részt hozzákapcsoljuk a régihez, azzal a feltétellel, hogy az új program sorszámai magasabbak, mint a számítógépben levő program sorszámai. (Itt nem építettünk be semmiféle automatizmust.) A programok kezelésénél figyelni kell arra, hogy mindig csak olyan programot szabad hozzászerezteni, amelynek utasítássorszáma magasabb, mint a belső tárban levő programé. Egy automatikus utasítássorszám-felismerés túl nagy munkát igényelne.

Ha a MERGE? kérdésre I-vel válaszolunk, akkor a számítógép betölti a belső tárba a megfelelő programot (anélkül, hogy újból megkérdezte volna a nevét).

Ezután READY jelenik meg a normál képernyőn. Csak akkor kapcsol át újból a finomfelbontású grafikára, ha RUN parancsot adunk (ellentétben a LOAD parancssal). Most a program előlről indul, és tudnia kell, hogy mely programok találhatóak a tárban, és melyeket kell feldolgozni.

Ha a MERGE? kérdésre N-nel válaszolunk, akkor a következő kérdés olvasható:

BELSŐ? I/N

Ily módon a rendszeren belül tetszőlegesen ugrálhatunk anélkül, hogy újabb programot töltenénk be. Ennek csak akkor van értelme, ha már több program van a belső tárban.

Ha a MERGE? kérdés és a READY megjelenése után RUN-nal indítjuk a programot, és a BELSŐ? kérdésre igennel válaszolunk, akkor indíthatjuk az előzőleg betöltött programot. A BELSŐ? kérdéshez csak akkor jutunk el, ha mind a LOAD, mind a

MERGE parancsra N-nel válaszolunk. Ha a BELSŐ? kérdésre is N-nel válaszolunk, akkor a főprogram újból a menü elejére ugrik, és újból a PROGRAMNÉV? kérdést teszi fel.

És most elérkeztünk a legérdekesebbhez. Az a kérdés, hogy mi történik azután, amikor a LOAD? I/N kérdésre I-vel válaszoltunk, és megadtuk másodszor is a program nevét. Általában minden LOAD parancs törli a belső tárt, így ha betöltjük a kívánt programunkat, törölné a menüprogramot. Szeretnénk azonban a menüprogramunkat megvédeni. A 410, 420 és 430-as sor éppen ezt a védelmet adja. Ezekkel a POKE utasításokkal a BASIC tárterületét feljebb helyezük. A menüprogram így a BASIC terület alatt marad egy dobozba zárva, a számítógép egyáltalán nem vesz tudomást róla. Az új programot a most már feljebb kezdődő BASIC tárterületre írjuk. Az új programot tehát egy új, üres dobozba helyezük el. Mivel a számítógép egyáltalán nem tud a régi dobozról, így azt sem tudja, milyen programnevet adtunk meg előzőleg. Ezért kell másodszor is megadni a program nevét úgy, ahogy ezt a 431—433. sorokban programoztuk.

Miután betöltöttük a programot, szeretnénk ha automatikusan el is indulna. Ezért egy olyan, a programban elől levő sorra kell ugratni, amiről biztosan tudjuk, hogy az újonnan betöltött program kezdete előtt van. Válasszuk a 210-es sort. A menüprogramunk 210-es sora nem indulhat el, mert az a régi dobozunkba be van zárva, és ezért nem ismeri fel a számítógép. A számítógép így a 210. sortól kezdődően az új dobozban keres egy programsorszámot, ahonnan elindíthatja a programot. Ekkor az új programunk felteszi azokat a kérdéseket, amelyek a működéshez kellene. Az egész folyamat az új dobozban megy végbe.

A programelemeket is módosítanunk kellett, hogy rendszerprogramok (SY-programok) legyenek belőlük. Először is a RETURN sor után beépítettük a speciális kérdéseket, és POKE utasításokat írtunk a program végére. Ha befejeztük a program futtatását, és vissza akarunk térni újból a menüprogramba, fel kell nyitni a régi dobozunk fedelét, hogy a számítógépnek újból rendelkezésére álljon a menüprogram. A BASIC tárterületét vissza kell állítani az eredeti címre. A SY-programoknak ezt automatikusan el kell végezniük, különben nem tudnának visszatérni a menübe. A SY-programunk egyelőre benn marad a tárban, csak akkor törlődik, ha újabb SY-programot töltünk be. Előtte a menüprogramot újból vissza kell zárni a dobozába.

Talán már ismerték a POKE és a PEEK utasítások hatalmát. (A PEEK utasítás a POKE utasítás ellentettje, ezzel a belső tár adott helyéről olvashatunk ki.)

A végére értünk. Ezen elvek alapján már felépíthetünk egy saját CAD-rendszert, vagy az itt bemutatott rendszert bővíthetjük. (A bővítéshez csak a 250. sortól kezdődő IF utasításokat kell folytatni, és új SY-programokat kell írni, amiket egyszerűen az alapprogramokból továbbfejleszthetünk.) Javíthatjuk is ezt a rendszert, hisz nagyon messze van még attól, hogy kifinomított rendszernek nevezhessük.

Nem jelennek meg pl. a beadott értékek a képernyőn. Vagy pl. ha egy hibás programnevet adunk meg, akkor a program hibajelzéssel (FILE NOT FOUND) leáll. Ezt is meg lehet változtatni. Vagy pl. az állandó RETURN nyomogatást is meg lehet változtatni. Ezeket azért nem építettük be a programba, hogy a lényeg ne takarják el.

Főleg azt akartuk bemutatni, hogyan lehet egy bonyolult rendszert egyszerű programelemekből összeállítani. Egy ilyen rendszer az igényeknek és a kívánságoknak megfelelően átalakítható, de minden újabb igény további ráfordítást jelent. Tudni kell, hogy mit akarunk, és mi az, amit lehetőség szerint kevés fáradtsággal megoldhatunk. A fáradtság esetünkben a számítógép belső tárjának elfoglalását is jelenti. Ezért szerintem a MERGE-lehetőség teljesen felesleges. Csak feleslegesen foglalják a gép belső tárját. Éppen olyan jól lehet csak LOAD parancsokkal dolgozni, így is könnyedén elkészíthetjük rajzainkat. Így mindig csak annyi tárhelyet foglalunk el, amennyi feltétlenül szükséges. Az a tanácsunk: felejtsek el a CADDYMAT-rendszerben a MERGE parancsot. Ha eltekintünk a MERGE-től, akkor a menüprogramunk is csak az 1—60. és 410—490. sorból áll, és a program nevének kétszeri megkérdezése is szükségtelenné válik.

```

1 REM *****
2 REM *****
3 REM * MENÜPROGRAM *
4 REM * CADDYMAT *
5 REM *****
6 REM *****
7 HIRES 0,7
10 REM START
20 :TEXT 40,80,"CADDYMAT",2,3,32
30 :TEXT 1,190,"PROGRAMNÉV?",2,1,8
40 INPUT A$
50 :TEXT 40,80,"CADDYMAT",2,3,32
60 :TEXT 1,190,"PROGRAMNÉV?",2,1,8
70 :TEXT 1,190,"LOAD? I/N",2,1,8
80 INPUT B$
90 :TEXT 1,190,"LOAD? I/N",2,1,8
100 IF B$="I" THEN 400
110 TEXT 1,190,"MERGE? I/N",2,1,8
120 INPUT B$
130 TEXT 1,190,"MERGE? I/N",2,1,8
140 IF B$="I" THEN 380
150 REM UGRÁS A PROGRAMON BELÜL
160 TEXT 1,190,"BELSŐ? I/N",2,1,8
170 INPUT B$
180 TEXT 1,190,"BELSŐ? I/N",2,1,8
190 IF B$="I" THEN 360
200 GOTO 10
210 IF A$="TÉGLA" THEN 11350
220 IF A$="HENGER" THEN 11450
230 IF A$="KÚP" THEN 11480
240 IF A$="KÉPTÁROLÁS" THEN 12700
250 IF A$="KÉPBETÖLTÉS" THEN 12800
260 :TEXT 1,175,"A PROGRAM NEM LÉTEZIK",2,1,8
270 :TEXT 1,190,"CSAK MÁSIK NÉV? I/N",2,1,8
280 :TEXT 1,175,"A PROGRAM NEM LÉTEZIK",2,1,8
300 :TEXT 1,190,"CSAK MÁSIK NÉV? I/N",2,1,8

```

```

310 IF C$="N" THEN 10
320 :TEXT 1,190,"ÚJ NÉV?",2,1,8
330 INPUT C$
340 :TEXT 1,190,"ÚJ NÉV?",2,1,8
350 GOTO 210
360 REM BELSŐ
370 GOTO 210
380 REM MERGE
390 MERGE A$+" SYS",B:GOTO 210
400 REM LOAD
410 POKE 43,(9000+1) AND 255
420 POKE 44,(9000+1)/256
430 POKE 9000,0:CLR
431 :TEXT 1,190,"KÉREM ÚJRA A PROGRAMNEVET",2,1,8
432 INPUT A$
433 :TEXT 1,190,"KÉREM ÚJRA A PROGRAMNEVET",2,1,8
440 LOAD A$+" SYS",B:GOTO 210
450 REM *****
460 REM *****
470 REM MENÜ VÉGE
480 REM *****
490 REM *****

11350 REM STÉGLA
11351 GOTO 11369
11352 REM TÉGLA ALPROGRAM
11353 D1=INT(SQR(B1+2/B))
11354 E1=X1 :F1=Y1+D1
11355 G1=X1+A1 :H1=Y1+D1
11356 I1=X1+A1+D1 :J1=Y1+D1-D1
11357 K1=X1+D1 :L1=Y1+D1-D1
11358 M1=X1+A1 :N1=Y1
11359 O1=X1 :P1=Y1
11360 Q1=X1+A1+D1 :R1=Y1-D1
11361 S1=X1+D1 :T1=Y1-D1
11362 :REC X1,Y1,A1,D1,1
11363 :REC S1,T1,A1,D1,1
11364 :LINE E1,F1,K1,L1,1
11365 :LINE G1,H1,I1,J1,1
11366 :LINE M1,N1,O1,R1,1
11367 :LINE Q1,P1,S1,T1,1
11368 RETURN
11369 REM HIRES
11370 :TEXT 5,190,"TÉGLA: SAROK X?, Y?",2,1,8
11371 INPUT X1,Y1
11372 :TEXT 5,190,"TÉGLA: SAROK X?, Y?",2,1,8
11373 :TEXT 5,190,"SZÉLESSÉG,MÉLYSÉG,MAGASSÁG?",2,1,8
11374 INPUT A1,B1,C1
11375 :TEXT 5,190,"SZÉLESSÉG,MÉLYSÉG,MAGASSÁG?",2,1,8
11376 GOSUB 11352

```

```

11377 :TEXT 5,190,"ISMÉTLÉS? I/N",2,1,B
11378 INPUT F$
11379 :TEXT 5,190,"ISMÉTLÉS? I/N",2,1,B
11380 IF F$="I" THEN 11386
11381 :TEXT 5,190,"KÉPNYOMTATÁS? I/N",2,1,B
11382 INPUT F$
11383 :TEXT 5,190,"KÉPNYOMTATÁS? I/N",2,1,B
11384 IF F$="N" THEN 11386
11385 OPEN 1,4:COPY
11386 POKE 43,(2048+1)AND255:POKE 44,(2048+1)/256
11387 POKE 2048,0
11388 GOTO 10

11450 REM SHENGER
11451 GOTO 11466
11452 REM HENGER ALPROGRAM
11453 R4=INT(R4/2)
11454 S4=INT(R4/2)
11455 E4=X4-R4 :F4=Y4
11456 G4=X4+R4 :H4=Y4
11457 I4=X4+R4+Z4 :J4=Y4-B4
11458 K4=X4-R4+Z4 :L4=Y4-B4
11459 O4=X4 :P4=Y4
11460 M4=X4+Z4 :N4=Y4-B4
11461 :CIRCLE O4,P4,R4,S4,1
11462 :CIRCLE M4,N4,R4,S4,1
11463 :LINE E4,F4,K4,L4,1
11464 :LINE G4,H4,I4,J4,1
11465 RETURN
11466 REM HIRES
11467 :TEXT 5,190,"ALAPKÖR-KÖZÉPPONT X?, Y?",2,1,B:
INPUT X4,Y4
11468 :TEXT 5,190,"ALAPKÖR-KÖZÉPPONT X?, Y?",2,1,B
11469 :TEXT 5,190,"ÁTMÉRŐ, MAGASSÁG, FERDESÉG?",2,1,B:
INPUT R4,B4,Z4
11470 :TEXT 5,190,"ÁTMÉRŐ, MAGASSÁG, FERDESÉG?",2,1,B:
GOSUB 11452
11471 :TEXT 5,190,"ISMÉTLÉS? I/N",2,1,B:
INPUT F$
11472 :TEXT 5,190,"ISMÉTLÉS? I/N",2,1,B
11473 IF F$="I" THEN 11478
11474 :TEXT 5,190,"KÉPNYOMTATÁS? I/N",2,1,B:
INPUT F$
11475 :TEXT 5,190,"KÉPNYOMTATÁS? I/N",2,1,B
11476 IF F$="N" THEN 11478
11477 OPEN 1,4:COPY
11478 POKE 43,(2048+1)AND255:POKE 44,(2048+1)/256
11479 POKE 2048,0:GOTO 10

```

```

11480 REM SKÚP
11481 GOTO 11492
11482 REM KÚP ALPROGRAM
11483 SS=INT(R5/2)
11484 ES=X5-R5:FS=Y5
11485 GS=X5+R5:HS=Y5
11486 IS=X5+Z5:JS=Y5-B5
11487 KS=X5      :LS=Y5
11488 :CIRCLE KS,LS,R5,SS,1
11489 :LINE   ES,FS,IS,JS,1
11490 :LINE   GS,HS,IS,JS,1
11491 RETURN
11492 REM HIRES
11493 :TEXT 5,190,"ALAPKÖR-KÖZÉPPONT X?, Y?",2,1,B
11494 INPUT X5,Y5
11495 :TEXT 5,190,"ALAPKÖR-KÖZÉPPONT X?, Y?",2,1,B
11496 :TEXT 5,190,"ÁTMÉRŐ, MAGASSÁG, FERDESÉG?",2,1,B
11497 INPUT R5,B5,Z5
11498 :TEXT 5,190,"ÁTMÉRŐ, MAGASSÁG, FERDESÉG?",2,1,B
11499 GOSUB 11482
11500 :TEXT 5,190,"ISMÉTLÉS? I/N",2,1,B
11501 INPUT F$
11502 :TEXT 5,190,"ISMÉTLÉS? I/N",2,1,B
11503 IF F$="I" THEN 11509
11504 :TEXT 5,190,"KÉPNYOMTATÁS? I/N",2,1,B
11505 INPUT F$
11506 :TEXT 5,190,"KÉPNYOMTATÁS? I/N",2,1,B
11507 IF F$="N" THEN 11509
11508 OPEN 1,4:COPY
11509 POKE 43,(2048+1)AND255:POKE 44,(2048+1)/256
11510 POKE 2048,0
11511 GOTO 10

12700 REM SKÉPTÁROLÁS
12701 GOTO 12757
12702 REM KÉPTÁROLÁS ALPROGRAM
12703 K=0
12704 :TEXT 10,190,"TÁROLÁS? I/N",2,1,B
12705 INPUT A$
12706 :TEXT 10,190,"TÁROLÁS? I/N",2,1,B
12707 IF A$="N" THEN 12742
12708 :TEXT 10,190,"ÁLLOMÁNYNÉV?",2,1,B
12709 INPUT A$
12710 :TEXT 10,190,"ÁLLOMÁNYNÉV?",2,1,B
12711 :TEXT 10,190,"SAROK X?, Y?",2,1,B
12712 INPUT A,B
12713 :TEXT 10,190,"SAROK X?, Y?",2,1,B
12714 :TEXT 10,190,"SZÉLESSÉG?, MAGASSÁG?",2,1,B
12715 INPUT C,D

```

```

12716 :TEXT 10,190,"SZÉLESSÉG?, MAGASSÁG?",2,1,8
12717 :TEXT 10,190,"LETAPOGATÁS" ,2,1,8
12718 G=C*D:Y=B:K=0
12719 DIM H(K):DIM M(G)
12720 FOR I=1 TO D
12721 X=A
12722 Y=Y+1
12723 FOR J=1 TO C
12724 X=X+1
12725 L=TEST(X,Y):PLOT X,Y,2:PLOT X,Y,2
12726 IF L=1 THEN 12728
12727 GOTO 12731
12728 K=K+1
12729 H(K)=X
12730 M(K)=Y
12731 NEXT J
12732 NEXT I
12733 :TEXT 10,190,"LETAPOGATÁS" ,2,1,8
12734 :TEXT 10,190,"TÁROLÁS",2,1,8
12735 :OPEN 2,8,2,AS+"S,W"
12736 PRINT#2,A;CHR$(13)B;CHR$(13)C;CHR$(13)D;CHR$(13)K
12737 FOR N=1 TO K
12738 PRINT#2,H(N);CHR$(13)M(N)
12739 NEXT N
12740 CLOSE 2
12741 :TEXT 10,190,"TÁROLÁS",2,1,8
12742 REM ISMÉTLÉS
12743 :TEXT 10,190,"ISMÉTLÉS? I/N",2,1,8
12744 INPUT B$
12745 :TEXT 10,190,"ISMÉTLÉS? I/N",2,1,8
12746 IF B$="I" THEN 12751
12747 :TEXT 10,190,"VÉGE",2,1,8
12748 PAUSE 10
12749 :TEXT 10,190,"VÉGE",2,1,8
12750 GOTO 12755
12751 CLR
12752 GOSUB 12702
12753 GOTO 12760
12754 CLR
12755 RETURN
12756 CLR
12757 REM HIRES
12758 :TEXT 5,190,"KÉPTÁROLÁS",2,1,8
12759 PAUSE 5
12760 :TEXT 5,190,"KÉPTÁROLÁS",2,1,8
12761 GOSUB 12702
12762 POKE 43,(2048+1)AND255:POKE 44,(2048+1)/255
12763 POKE 2048,0
12764 GOTO 10

```

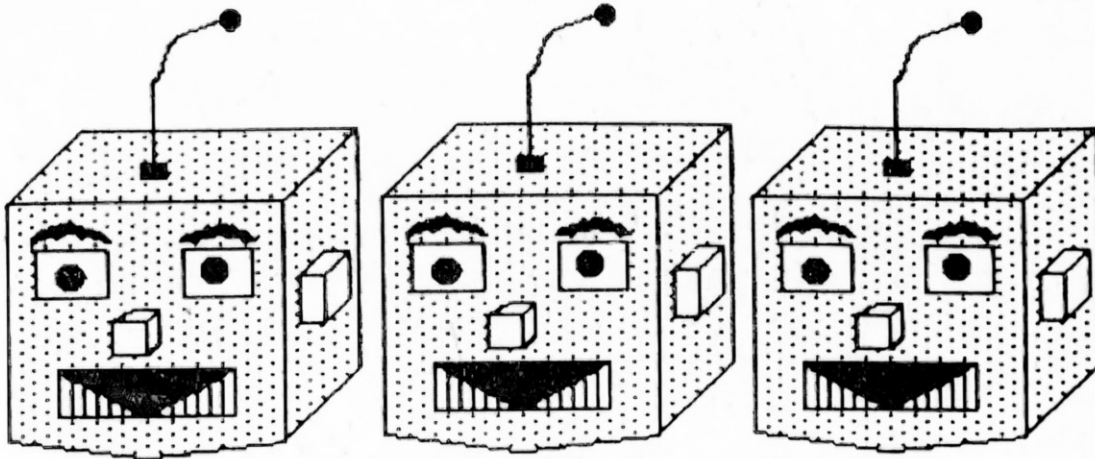


```

12800 REM SKÉPBETÖLTÉS
12801 GOTO 12841
12802 REM KÉPBETÖLTÉS ALPROGRAM
12803 :TEXT 10,190,"BETÖLTÉS? I/N",2,1,B
12804 INPUT A$
12805 :TEXT 10,190,"BETÖLTÉS? I/N",2,1,B
12806 IF A$="N" THEN 12827
12807 :TEXT 10,190,"ÁLLDMÁNYNÉV?",2,1,B
12808 INPUT A$
12809 :TEXT 10,190,"ÁLI DMÁNYNÉV?",2,1,B
12810 :TEXT 10,190,"BEOLVASÁS",2,1,B
12811 :OPEN 2,B,2,A$+",S,R"
12812 INPUT#2,A,B,C,D,K
12813 G=C*D
12814 DIM H(G):DIM M(G)
12815 FOR N=1 TO K
12816 INPUT#2,H(N),M(N)
12817 NEXT N
12818 CLOSE 2
12819 :TEXT 10,190,"BEOLVASÁS",2,1,B
12820 :TEXT 10,190,"RAJZOLÁS? I/N",2,1,B
12821 INPUT B$
12822 :TEXT 10,190,"RAJZOLÁS? I/N",2,1,B
12823 IF B$="N" THEN 12827
12824 FOR N=1 TO K
12825 :PLOT H(N),M(N),1
12826 NEXT N
12827 REM ISMÉTLÉS
12828 :TEXT 10,190,"ISMÉTLÉS? I/N",2,1,B
12829 INPUT B$
12830 :TEXT 10,190,"ISMÉTLÉS? I/N",2,1,B
12831 IF B$="I" THEN 12836
12832 :TEXT 10,190,"VÉGE",2,1,B
12833 PAUSE 10
12834 :TEXT 10,190,"VÉGE",2,1,B
12835 GOTO 12839
12836 CLR
12837 GOSUB 12802
12838 GOTO 12843
12839 RETURN
12840 CLR
12841 REM HIRES
12842 GOSUB 12802
12843 POKE 43,(2048+1)AND255:POKE 44,(2048+1)/256
12844 POKE 2048,0
12845 GOTO 10

```

VÉGSZÓ



Ez a könyv csak ízelítőt adhatott.

A témakör olyan komplex, hogy az itt bemutatott programok nem térhettek ki minden területre. Egyetlen programmal nem lehet minden felhasználási területet lefedni.

Ha továbbfejlesztjük a programot, bizonyára lesznek olyan esetek, amikor a program nem működik. Ekkor valószínűleg valamely peremfeltételt megsértettük. Némi utánanézésel megtalálhatjuk a hibát, és ezzel sok mindent megtudhatunk a számítógépről és annak nyelvéről. Tág tér marad még a kísérletezőkedvünknek.

Köszönöm a figyelmüket.

Ára: 290,— Ft

SZÁMÍTÁSTECHNIKA A KÖNYVESBOLTOKBAN



NOVOTRADE – 2 C ÁRUHÁZ
1136 Bp., Balzac u. 35. Tel.: 402-954

ÁLLAMI KÖNYVTERJESZTŐ V. – NOVOTRADE 2C

BUDAPEST

Táncsics Könyvesbolt
1073 Lenin krt. 17.
Telefon: 422-178

BUDAPEST

Műszaki Könyvárúház
1061 Liszt Ferenc tér 9.
Telefon: 420-353

MŰVELT NÉP KÖNYVTERJESZTŐ V. – NOVOTRADE 2C

PÉCS

Zrínyi Miklós Könyvesbolt
7621 Jókai u. 25.
Telefon: 72-12835

VESZPRÉM

Kölcsey Ferenc
Könyvesbolt
8200 Cserhát út 7.

SZEGED

Tömörkény. Könyvesbolt
6720 Lenin krt. 48.
Telefon: 62-21453

DEBRECEN

Szak- és ismeretterjesztő
Könyvárúház
4024 Hunyadi u. 8.
Telefon: 52-23237

BÉKÉSCSABA

Radnóti M. könyvesbolt
5600 Tanácsköztársaság
út 2.
Telefon: 25-207

SZOLNOK

Szigligeti Könyvesbolt
5000 Ságvári krt. 35.
Telefon: 56-11133

SZOMBATHELY

Savaria Könyvesbolt
9700 Mártírok tere 1.
Telefon: 94-12341

GYŐR

Pattantyús Á. Géza Szak-
könyvesbolt
9021 Molnár Ferenc u. 9.

MISKOLC

Chip-kuckó
3530 Tanácsház tér 14.

Minden érdeklődőt szeretettel vár
az ÁKV, a Művelt Nép és a NOVOTRADE RT!