



BASIC KÄSIKIRJA



BASIC KÄSIKIRJA

JOHDANTO

Tämä kirjanen on tarkoitettu niille, jotka haluavat oppia ohjelmoimaan Dragon 32-tietokoneella Basic-ohjelmointikieltä käyttäen.

Basic-kieli on erittäin tehokas ohjelmointikieli, mutta se on myös hyvin helppo oppia. Se koostuu alle 100 käskystä, mikä on vähemmän kuin 1 % ihmisen normaalista sanavarastosta.

Ohjelmointi saattaa aluksi tuntua vaikealta kuten minkä tahansa uuden asian oppiminen, mutta itse asiassa se perustuu jonkin ongelman ratkaisemiseen loogisessa järjestyksessä, vaihe vaiheelta. Ongelmiin on paneuduttava rauhallisesti ja edellinen vaihe on pyrittävä ymmärtämään täysin ennen kuin siirrytään seuraavaan vaiheeseen. Virheistä ei pidä huolestua, ne ovat osa oppimistapahtumaa. Ohjelmointivirheet eivät voi rikkoa tietokonetta; kun virhe on löydetty, se korjataan ja jatketaan eteenpäin. Kokeile omia ideoitasi. Omat kokeilut voivat olla hyvinkin nopea tapa saada selville, mihin kone pystyy. Perehdy jokaiseen esimerkkiin ja kokeile sitä, ei vain nähdäksesi, mitä tapahtuu vaan myös miksi niin tapahtuu.

SISÄLLYSLUETTELO

Käyttöohjeet	1
Pakkauksen sisältö	1
Opas	2
Dragon 32-tietokoneen toimintakuntoon saattaminen	3
Ohjelmamodulien käyttö	4
Ohjausvivun käyttö	4
Kasettinauhurin käyttö	4
Dragon 32-tietokoneen huolto	5
1. Käynnistäminen	6
Näppäimistö	6
Dragon laskukoneena	7
Laskenta	9
Tekstin kirjoittaminen	14
Tehtäviä	15
2. Käsitteitä	17
Vakiot	17
Muuttujat	17
Muuttujien nimeäminen	18
Muuttujien arvojen määrittäminen	19
Merkkijonoja ja lukuja ei voi yhdistellä	22
Komennot	23
3. Ohjelman laatiminen	25
Ohjelman kirjoittaminen	26
Vaihe vaiheelta	27
Muutosten tekeminen	37
Ohjelman rakenne	38
Ohjelmointiesimerkki	48
4. Hyvä järjestys	51
Nauhurin kytkeminen	51
Ohjelman tallentaminen kasetille	53
Ohjelman lataaminen muistiin	54
Useampien ohjelmien tallentaminen	55
Vihjeitä luotettavaa tallentamista varten	55
Toimitusohjelma (EDITOR)	56

Kursorin siirtyminen rivillä	57
Rivin muuttaminen	57
Lisää systeemikomentoja	60
5. Ohjelman haarautuminen	69
Haarautumiskäskyt	69
Ehdollinen haarautuminen	70
Toisto	81
Sisäkkäiset ohjelmat	85
6. Uusia ulottuvuuksia	91
Luettelot ja taulukot	91
Mikä on funktio	96
Omat funktiosi	103
Muita syöttökäskyjä	105
7. Kuvat ja liikkuvat kuviot	112
Piste on perusosa	112
Kuvien laatiminen	112
Liikkuvat kuvat	116
Uusi grafiikkataso	119
8. Hienografiikka	127
Grafiikkatason valinta	128
Pisteet	129
Janan piirtäminen	130
Ympyrät	138
Sivujen selaaminen	140
9. Ääni	147
Äänen tuottaminen	147
Soittaminen	148
10. Lisää grafiikkaa	157
Piirtäminen	157
Kuvien siirtely	162
11. Loppusilaus	171
Lisää tulostuksesta	171
Kasettisyöttö ja -tulostus	175
Konekieli	181

Liite A	185
ASCII-koodit	185
Graafiset merkit	187
Liite B	188
Ruudukko Print@-lausetta varten	188
Puoligrafiikkaruudukko	189
Hienografiikkaruudukko	190
Liite C	191
Virhekoodit	191
Liite D	193
Trigonometriset funktiot	193
Liite E	194
Kkirjoittimen käyttö	194
Liite F	194
Muistikartta (memory map)	194

KÄYTTÖOHJEET

Täydellinen Dragon 32-paketti sisältää:

1. Dragon 32 mikrotietokone
2. Muuntaja
3. TV-Liitännäjohto
4. Kasettisoittimen liitännäjohto
5. Basic-ohjekirja

Näiden lisäksi tarvitset myös tavallisen TV-vastaanottimen. Dragon-mikrotietokone voidaan liittää sekä väri- että mustavalkotelevisioon. Graafisiin esityksiin saadaan kuitenkin täydelliset värit vain väri vastaanottimella. Dragon on nyt käyttökunnossa.

Laitteen toimintavalmiuksia voidaan kuitenkin laajentaa seuraavien lisälaitteiden avulla:

1. kasettinauhuri ohjelmien ja tiedon tallentamista varten
2. kirjoitin
3. ohjausvipu pelejä varten
4. levyasemat ohjelmien ja tietoineksen laajamittaista tallentamista varten.

Nämä lisälaitteet eivät ole välttämättömiä, mutta kasettinauhuria käyttäessäsi säästyt moninkertaiselta kirjoitustyöltä.

OPAS

1. TV-liitântä (TV) kytketään television antennipistokkeeseen mukana seuraavalla TV-liitântäjohdolla.
2. RESET-painike keskeyttää ohjelman suorituksen tai syötö/tulostus-toiminnot heti. Muistissa oleva ohjelma säilyy muistissa, vaikka reset-näppäintä painetaan.
3. Vasen ohjausvivun liitântä (L.J.STK). Lisävarusteena saatavan ohjausvivun (joystick) 5-napaisen DIN-pistokkeen liitântä.
4. Oikean ohjausvivun liitântä (R.J.STK). Lisävarusteena saatavan ohjausvivun (joystick) 5-napaisen DIN-pistokkeen liitântä.
5. Kasettinauhurin (TAPE) liitântä. 5-napainen DIN-liitântä kasettinauhuria varten. Liitântäjohto mukana vakiovarusteena.
6. Kirjoittimen rinnakkaiskytkentä (P.I/O) standardia "centronics"-tyyppistä johtoa varten.
7. Mm. peliohjelmissä käytettävän ohjelmamodulin paikka. Kone ei saa olla kytkettynä päälle modulia paikoilleen asetettaessa tai poistettaessa.
8. Muuntajaliitântä (TRANS. SUPPLY) koneen mukana toimitettavan muuntajan liitântää varten
9. Monitoriliitântä (MONITOR))
10. Virtakytkin (ON/OFF)

DRAGON 32 -TIETOKONEEN TOIMINTAKUNTOON SAATTAMINEN

1. Kytke TV-liitäntäjohto televisiosi antennipistokkeeseen ja Dragon 32:n TV-pistokkeeseen (1).
2. Kytke muuntaja Dragon 32:n liitäntään (8). Toinen muuntajasta lähtevä virtajohto liitetään seinäpistorasiaan.
3. Kytke TV ja tietokone päälle.
4. Tietokone voidaan liittää sekä väri- että mustavalkovastaanottimeen, jossa on UHF-kanava. Joistakin vanhemmista vastaanottimista tämä puuttuu, eikä Dragonia siis voida liittää niihin. Liitäntäjohto toimitetaan laitteen mukana. Johdon toinen pää kytketään Dragonin TV-liitäntään (merkitty TV) ja toinen pää television antennikoskettimeen. Kun tietokone ja televisio on kytketty päälle, televisiosta etsitään vapaa kanava ja kanavanvalitsia käännetään UHF-alueelle (merkitty yleensä UHF tai U). Kanava säädetään siten, että ruudussa näkyy mustan reunuksen rajaama vihreä neliö (tai mustan reunuksen rajaama vaaleanharmaa neliö, jos vastaanotin on mustavalkoinen).

Vihreässä neliössä on seuraava teksti:

(C) 1982 DRAGONDATA LTD

16K Basic INTERPRETER 1.0

(C) 1982 BY MICROSOFT

OK

Tietokoneesi on nyt käyttövalmis.

Jos kytkennässä esiintyy ongelmia lue television käyttöohjeista asemanvalitsimen (UHF) käyttöä koskevat ohjeet.

OHJELMAMODULIEN KÄYTTÖ

Kytke Dragon TV-vastaanottimeen kuten edellä. Ennen kuin kytke virran tietokoneeseen aseta ohjelmamoduli paikoilleen (7) siten, että etiketti on ylöspäin. Tarkista aina, että virta on kytketty pois päältä ennen kuin asetat ohjelmamodulin paikoilleen tai poistat sen. Ellei virtaa ole kytketty pois, sekä moduli että tietokone saattavat vaurioitua.

OHJAUSVIVUN KÄYTTÖ

Dragon 32:en onsaatavana lisälaitteina erilaisia ohjausvipuja. Ne ovat potentiometrityyppisiä. Ne kytketään ohjausvipuliitännöihin 3 ja 4.

KASETTINAUHURIN KÄYTTÖ

Kaikkia hyvätasoisia kasettinauhureita voidaan käyttää ohjelmien ja tiedon tallentamiseen Dragon 32-tietokoneesta. Nauhurissa pitää olla liitännät kaukosäätöä, kuulokkeita ja mikrofonia varten. Liitännäjohtoja on saatavana useimpia tavallisia kasettinauhureita varten. Tietokoneessa nauhuri kytketään liitännään (5). Nauhuriin tuleva liitäntä riippuu nauhurityypistä. Kasettinauhurin liitännää käsitellään yksityiskohtaisemmin kappaleessa 4.

DRAGON 32 -TIETOKONEEN HUOLTO

1. Pidä kaikki nestemäiset aineet kaukana tietokoneesta.
2. Huolehdi siitä, että kaikki irtonaiset rautalangan pätkät ym. ovat turvallisen välimatkan päässä. Niiden joutuminen koneeseen voi käydä kalliiksi.
3. Tarkista, että kaikki pistotulpat on painettu kunnollisesti paikoilleen ennen kuin kytket virran päälle.
4. Kytke virta pois kaikista laitteista ja ota virtajohto irti, kun laite ei ole käytössä.
5. Irrota laite aina kokonaan virtalähteestä ennen kuin puhdistat kuorta tai näppäimistöä. Pyyhi kuori ja näppäimistö vain hiukan kostealla kankaalla. Älä käytä alkoholi-pitoisia puhdistusaineita.

1. KÄYNNISTÄMINEN

NÄPPÄIMISTÖ

Tietokone on nyt kytketty ohjeiden mukaisesti, ja olet valmis aloittamaan. Kytke virta päälle. Televisiovastaanotuksessa näkyy VIHREÄ neliö ja siinä teksti. (Elleivät nämä ilmesty ruutuun, kytke virta pois ja tarkista kaikki liitännät). Teksti riippuu konetyypistä ja siitä, onko laitteeseen kytketty levyasemia jne. - se ei siis ole tässä vaiheessa tärkeää. Viimeinen rivi on kuitenkin aina sama.

OK

OK ilmoittaa, että tietokone on valmis ottamaan vastaan käskyjä. Tätä ilmoitusta on aina odotettava ennen kuin koneeseen näppäillään mitään. OK-merkinnän alla on vilkkuva neliö, jota kutsutaan kursoriksi. Se osoittaa, missä kohtaa rivillä kulloinkin ollaan.

Tutustu sitten näppäimistöön: Se muistuttaa kirjoituskoneen näppäimistöä, jossa on joitakin ylimääräisiä näppäimiä, ja se myös toimii kuten kirjoituskone. Voit painaa joitakin näppäimiä ja näet kirjaimien ilmestyvän ruutuun. Kursori siirtyy eteenpäin jokaisen kirjaimen mukana ja osoittaa, kuinka pitkälle on edetty. Jos painat samanaikaisesti vaihtonäppäintä (SHIFT) ja nollaa (Ø) ja jatkat kirjoittamista, huomaat, että kuvaruudulle tulevat kirjaimet muuttuvat vihreiksi ja tausta mustaksi. Kirjoita nyt pienillä kirjaimilla

(siis pieni a, b, c, d), mutta ne esiintyvät pieninä kirjaimina vain KIRJOITTIMELLA kirjoitettavassa tekstissä. Tässä kirjassa nolla on aina merkitty Ø erotukseksi kirjaimesta O. Tämä ero on hyvin tärkeä. KAIKKI tietokoneelle annettavat käskyt ja ohjeet on aina kirjoitettava ISOILLA KIRJAIMILLA. Paina uudelleen yhtäikaa (SHIFT) ja Ø ja kirjoita hiukan lisää. Nyt tekstin pitäisi jälleen olla mustaa vihreällä taustalla.

Kokeile seuraavaksi (+)-näppäintä. Tämä on askelpalautin. Sitä painettaessa kursori siirtyy samalla rivillä taaksepäin. ja aina sen vasemmalla puolella oleva kirjain pyyhkiytyy pois. Näin voidaan helposti korjata virheet: mennään rivillä taaksepäin virhekohtaan asti ja kirjoitetaan teksti uudelleen.

Koko ruutu tyhjennetään painamalla CLEAR-näppäintä. Kaikki ruudulla oleva teksti pyyhkiytyy tällöin pois ja kursori siirtyy ruudun vasempaan yläkulmaan. CLEAR tyhjentää vain ruudun, koneen muistiin tallennettu tieto ei häviä.

Harjoittele jonkin aikaa näppäimistön käyttöä tottuaksesi näppäinten sijaintiin ja oppiaksesi korjaamaan virheet. Puhdista sitten ruutu ja tarkista, että kirjoitat isoilla kirjaimilla (mustat kirjaimet vihreällä taustalla).

DRAGON LASKUKONEENA

Tietokoneesi toimii kahdella tavalla. Välittömästi - se toimii heti annetun käskyn mukaan; ja viivästetysti - se tallentaa muistiinsa useiden käskyjen sarjan ja käyttää niitä

ohjelmanä. Välittömissä toiminnoissa tietokone toimii lasku-
koneen tavoin.

Kone ymmärtää Basic-kieltä. Basic koostuu useista erikois-
sanoista, joilla tietokoneelle annetaan toimintaohjeita. Esi-
merkiksi PRINT merkitsee, että koneen on kirjoitettava ruu-
dulle käskyssä määritellyt merkit.

Kokeile. Kirjoita PRINT 12 + 7 ja paina ENTER-näppäintä.

Ruudulle pitäisi ilmestyä:

19

OK

Kokeile jotakin muuta. Kirjoita PRINT 12 + 8/2 ja paina
ENTER-näppäintä:

16

OK

Jos teit kirjoitusvirheen, ruudulle saattoi ilmestyä:

?SN ERROR

Tämä tarkoittaa 'syntaksivirhe', mikä tarkoittaa, ettei kone
ymmärrä jotakin käskyn osaa. Useimmiten kyseessä on kirjoj-
itusvirhe. Kone antaa virheilmoituksen ellei se ymmärrä käs-
kyä tai jos se ymmärtää käskyn, mutta tehtävä on epälooginen
tai mahdoton.

Kirjoita PRINT 3/0 ja paina ENTER-näppäintä.

Saat vastaukseksi

?/0ERROR

Tämä tarkoittaa, että on yritetty jakaa nollalla - mikä on
mahdotonta.

Virheilmoitukset ovat pelkistettyjä tilan säästämiseksi, mutta yksityiskohtaisempi luettelo mahdollisista virhelähteistä on esitetty liitteessä C.

Palatkaamme vielä kirjoitusvirheisiin. Tietokone on erittäin tarkka Basic-kielisten sanojen kirjoitusasusta. Jos huomaat tehneesi virheen ennen kuin painat ENTER-näppäintä, voit käyttää askelpalautinta (+) ja korjata virheen. Muutoin ei ole muuta vaihtoehtoa kuin kirjoittaa koko rivi uudelleen oikeassa asussa.

LASKENTA

Tähän mennessä olemme käyttäneet tietokonetta vain kahteen laskutoimitukseen, yhteenlaskuun (+) ja jakolaskuun (/). Tietokonetta voidaan kuitenkin käyttää kuuteen yksinkertaiseen laskutoimitukseen, ja näitä toimintoja varten on tarkat säännöt. Edellä esitettyssä esimerkissä

PRINT 12 + 8/2

ei ole aivan selvää, onko oikea vastaus 16 vai 10.

12 plus 8 jaettuna 2:lla on 10 vai 8 jaettuna 2:lla on 4 plus 12 on 16.

Tietokoneen antama vastaus on 16 koneen valitseman laskujärjestyksen vuoksi. Laskutoimitukset ja niiden tärkeysjärjestys ovat seuraavat:

1. Potenssiin korottaminen

Esimerkiksi luku 5 korotettuna potenssiin 4 (5^4) on $5 \times 5 \times 5 \times 5$.

PRINT 4 + 3 ↑ 2

lasketaan siten, että kolme korotetaan ensin potenssiin 2 eli lasketaan sen neliö ($3 \times 3 = 9$) ja sen jälkeen lisätään 4, jolloin tulokseksi saadaan 13. Jos potenssiin korottamisia on useampia kuin yksi, ne suoritetaan vasemmalta oikealle. Ko-
keilleesimerkkiä

PRINT 2 ↑ 3 ↑ 2 ↑ 3 (potenssiin korotettaessa käytetään ↑-näp-
päintä näppäimistön vasemmassa reunassa).

Tämä lasketaan siten, että 2 korotetaan ensin itsellään kol-
me kertaa ($2 \times 2 \times 2 = 8$), tulo kerrotaan sitten itsellään
($8 \times 8 = 64$) ja tämä tulo kerrotaan itsellään kolme kertaa
($64 \times 64 \times 64 = 262144$).

2. Negatiivinen etumerkki

Negatiivinen etumerkki on miinusmerkki, joka osoittaa negatiivisen luvun

PRINT -3 + 2

Tietokone liittää ensin miinusmerkin lukuun. Siten $-3 + 2 = -1$. Jos yhteenlasku laskettaisiin ensin, tulokseksi saataisiin -5 , mutta kone ei tee näin.

3. Kertolasku

Kertomerkkinä käytetään tähteä (*), jotta ei tulisi sekaannuksia x-kirjaimen kanssa. Tähti saadaan painamalla SHIFT-näppäin alas.

PRINT 5 * 2 + 3

tulos on 13 ($5 * 2 = 10$ plus $3 = 13$).

4. Jakolasku

Jakolaskussa käytetään merkkiä /.

PRINT 5/2 + 3

tulos on 5.5 ($5:2 = 2,5$ plus $3 = 5,5$)

Jako- ja kertolasku ovat samanarvoisia. Samanarvoiset laskutoimitukset suoritetaan vasemmalta oikealle.

PRINT 5 + 2 * 3 + 4/2

tulos on 13 ($2 * 3 = 6$, $4/2 = 2$, $5 + 6 + 2 = 13$).

5. Yhteenlasku

Yhteenlaskumerkki on +.

6. Vähennyslasku

Vähennyslaskumerkki on -

Yhteen- ja vähennyslasku ovat myös samanarvoisia, joten ne suoritetaan järjestyksessä vasemmalta oikealle, kun kaikki muut laskutoimitukset on ensin tehty.

Tietokoneessa laskutoimitusten tärkeysjärjestys on siis lyhyesti seuraava:

1. ↑ (potenssiin korottaminen vasemmalta oikealle)
2. - (negatiivista lukua ilmoittava etumerkki)
3. * / (kerto- ja jakolasku vasemmalta oikealle)
4. + - (yhteen- ja vähennyslasku vasemmalta oikealle)

Seuraavassa on muutamia laskutoimituksia. Laske tulos ensin valmiiksi päässäsi tai kynän ja paperin avulla ja ko- keile sitä sitten tietokoneella. Jos tietokone antaa eri vastauksen kuin olet itse saanut, yritä saada selville miksi. Ellei sinulla ole aikaisempaa kokemusta siitä, miten tietokone suorittaa laskutoimitukset, sinun on syytä tehdä nämä laskuesimerkit. Useimmat nk. 'tietokoneen te- kemät virheet' johtuvat siitä, että ohjelmoija on noudattanut eri sääntöjä kuin tietokone. Vastauksia ei anneta. Jos tehtävät näppäillään täsmälleen näiden esimerkkien mukaisesti, tietokone antaa oikeat vas- taukset.

PRINT 3 + 2

PRINT 4 + 6 - 2 + 1

PRINT 8 * 4

PRINT 4 ↑ 2 + 1

PRINT 5/4 - 1

PRINT 5 - 4/2

PRINT 6 * - 2 + 6/3 + 8

PRINT 4 + - 2

PRINT 2 * 2 + 3 * 4

PRINT 8/2/2/4

PRINT 20/2 * 5

PRINT 8 * 2/2 + 5 * 3 * 2 ↑ 2

PRINT-sanaa ei aina tarvitse kirjoittaa. Samaa tarkoittava merkki on ?. Jos kirjoitat

? 3 + 2 se on sama kuin PRINT 3 + 2.

Olet varmaan jo tottunut painamaan ENTER-näppäintä jokaisen rivin lopussa.

OK ilmoittaa, että tietokone on valmis ottamaan vastaan uuden tehtävän.

ENTER ilmoittaa tietokoneelle, että käyttäjä on valmis.

Olemme nyt siis selvittäneet laskutoimitusten järjestyksen. mutta tätä järjestystä voidaan myös muuttaa käyttämällä sulkumerkkejä. Oletetaan, että luku 14 halutaan jakaa lukujen 3 ja 4 summalla. Jos kirjoitetaan $14/4 + 3$, saadaan tulokseksi 6.5, koska kone jakaa 14 ensin 4:llä ja lisää sitten tulokseen 3. Tätä ei kuitenkaan haluttu. Jotta laskutoimitus suoritettaisiin halutussa järjestyksessä, kirjoitetaan $14/(4 + 3)$ ja vastaus on 2.

Sulkumerkit muuttavat järjestyksen. Sääntö on yksinkertainen: suluissa olevat laskutoimitukset suoritetaan ensin. Jos sulkuja on useita sisäkkäin, aloitetaan sisimmistä:

$12/(3 + (1 + 2) \uparrow 2)$ lasketaan seuraavasti:

1. $(1 + 2)$ lasketaan ensin: $12/(3 + 3 \uparrow 2)$
2. seuraavaksi $3 \uparrow 2$: $12/(3 + 9)$
3. sitten $(3 + 9)$: $12/12$
4. jakolasku viimeiseksi: 1

Seuraavassa on lisää esimerkkejä. Ellet ole aikaisemmin perehtynyt tietokoneen laskutapaan, sinun kannattaa uhrata muutama minuutti näiden esimerkkien läpikäymiseen, jotta voisit käyttää tietokonettasi tehokkaammin.

? $44/(2 + 2)$
? $(44/2 + 2)$
? $4 + (-5 \times 2)$
? $100(200/(2 \times (9-5)))$
? $42/((9/3) + 1.75 + (5/4))$

TEKSTIN KIRJOITTAMINEN

Toistaiseksi olemme käyttäneet PRINT-komennossa vain numeroita. Maallikot pitävät tietokonetta usein pelkkänä "numeromurskaimena" mutta laskutoimitukset eivät ole tietokoneen ainoita toimintoja. Tietokoneella voidaan käsitellä myös muita merkkejä. Merkeillä tarkoitetaan tässä kirjaimia, numeroita, välimerkkejä ja muita erikoismerkkejä, joihin palaamme myöhemmin. Basic-kielen avulla voidaan käsitellä merkkiryhmiä, joita kutsutaan merkkijonoiksi.

Merkkijono voi olla erilaisten merkkien yhdistelmä - myös tyhjä välilyönti (merkitty ∇) voi olla tärkeä merkki jonoissa. Jono erotetaan kummastakin päästä lainausmerkeillä (" "), jotta kone tietää, että kyseessä on jono eikä pelkkä luku. Esimerkkejä merkkijonoista:

"OTSAKE","Z+*?!","EWERT KUPIAINEN"

"AXY 479","90-6543627"

Kaksi viimeistä merkkijonoa voivat olla esimerkiksi auton rekisterinumero ja puhelinnumero, ja vaikka niihin sisältyykin numeroita, näitä numerosarjoja ei käytettäisi mihinkään varsinaiseen laskutoimitukseen. Basic-kielessä lainausmerkkien

sisällällä oleva numerosarja ei ole luku, esimerkiksi jono "12345" on aivan eri asia kuin luku 12345. Jonot ja luvut tallennetaan aivan eri paikkoihin koneen muistissa. Kokeile:

```
PRINT "2 + 5 = "; 2 + 5
```

Ensimmäinen osa PRINT-lauseesta ilmestyy ruudulle juuri siinä muodossa kuin se kirjoitettiin jonoon. Lainausmerkit sulkevat merkkijonon, ne eivät ole sen osa. Lauseen toinen osa käsitetään numerolausekkeeksi, joten ruudulle ilmestyy

```
2 + 5 = 7
```

Kuten edellä todettiin, tyhjä välilyönti voi myös olla tärkeä merkki merkkijonossa. Basic-lauseessa välilyönneillä ei ole merkitystä, ne vain helpottavat rivin lukemista. Merkkijonoissa niillä kuitenkin on merkitystä, sillä jono kopioidaan täsmälleen samanlaisena lainausmerkkien erottamana merkkiryhmänä kuin se kirjoitetaan ruudulle.

Nyt osaat käyttää tietokonettasi yksinkertaisten ongelmien ratkaisemiseen. Näistä on esitetty esimerkkejä tämän kappaleen lopussa. Ellet ole aikaisemmin perehtynyt tietokoneisiin, kokeile näitä esimerkkejä.

Tehtäviä

1. Lauta on 168 cm pitkä. Mikä on sen pituus tuumina?
(1 tuuma = 2,54 cm).
2. Kakkuun tarvitaan 0,75 kg jauhoja. Kuinka monta desilitraa jauhoja tarvitaan?
(1 kg = 16,5 dl)

3. Auto kuluttaa 43,8 l bensiiniä matkalla. Kilometrimitarin lukema oli matkalle lähdettäessä 22798 ja palatessa 23411. Kuinka monta litraa bensiiniä auto kulutti sadalla kilometrillä?
4. Talletat 200 mk säästötilille, josta maksetaan 11 %:n korko vuodessa. Kuinka paljon talletettu summa on kasvanut 5 vuoden kuluttua?
 $(K = B(1 + R/100)^N)$, jossa B on talletettu summa, R on korkoprosentti, N on vuosien lukumäärä ja K on pääoma N vuoden kuluttua).
5. Polkupyörän pyörän halkaisija on 68 cm. Kuinka monta kierrosta pyörä tekee 1 km:n matkalla?
 (ympärysmitta on $3.14159 \times$ halkaisija).

Vastaukset

1. 66,14 tuumaa
2. 12,375 dl
3. 0,715 litraa/100 km
4. 337,01 mk
5. 468,1 kierrosta.

2. KÄSITTEITÄ

VAKIOT

Kaikissa tähän asti käytetyissä esimerkeissä on ollut ai-noastaan vakioita. Vakio on nimensä mukainen - se ei muu-tu. 3.145 on vakio. Jos se muutetaan luvuksi 3.145, saa-daan vain toinen vakio. Vakiot ovat hyödyllisiä tietokone-ohjelmissa, mutta eivät yhtä käyttökelpoisia kuin muuttujat.

MUUTTUJAT

Muuttuja on sellainen, jonka arvo voi muuttua. Kaavassa

$$X + 5 = Y$$

X ja Y ovat muuttujia, koska niillä on molemmilla useita ar-voja, jotka toteuttavat yhtälön. Tietokoneessa muuttujat ovat alueita tietokoneen muistissa, ja niitä voidaan verra-ta suuren varaston hyllyihin. Jotta ne voidaan tunnistaa ne on merkittävä "nimilapulla" (niille on siis annettava ni-met, kuten X tai Y). Tässä tietokoneessa muuttujia on kah-ta tyyppiä: Numeromuuttujat ja merkkijonomuuttujat. Niitä on myös kahta "kokoa" yksinkertaiset ja taulukot. Tunne-mme jo numero- ja merkkijonovakioiden välisen eron - numeromuut-tujissa on siis numeeriset vakiot (numerot) ja merkkijono-muuttujissa jonovakiot (merkit). Käsittelemme aluksi vain yksinkertaisia muuttujia, taulukkomuuttujia käsitellään myö-hemmin.

MUUTTUJIEN NIMEÄMINEN

Numeromuuttujan nimessä voidaan käyttää mitä tahansa kahden kirjaimen tai kirjaimen ja numeron yhdistelmää.

N, AA, X, TI, Y, Z9, L5, BZ, PQ, K9

ovat esimerkkejä mahdollisista numeromuuttujanimistä.

Dragon hyväksyy kyllä minkä pituisen muuttujan nimen tahansa, mutta se tunnistaa vain kaksi ensimmäistä kirjainta. Se hyväksyy siis esimerkiksi nimet

LUKEMA, LUETTELO, LUPA

mutta nämä käsitetään samaksi muuttujaksi LU. Sama pätee rivimuuttujiin.

NUMEROMUUTTUJIEN NIMET

Numeromuuttujan arvoksi voidaan tallentaa vain numeroita.

Numeromuuttujan nimi voi olla mikä tahansa kirjainten ja numeroiden yhdistelmä, mutta sen pitää alkaa kirjaimella.

Koska tietokone tunnistaa vain nimen kaksi ensimmäistä kirjainta, sellaiset nimet kuin

SARJA, SAATAVA, SAAPUVA

katsotaan samaksi numeromuuttujaksi (SA).

Pitemmät nimet auttavat kuitenkin muistamaan sisällön,

LUKU, KERROIN, SUMMA

ovat hyväksyttäviä, mutta ne vievät muistista enemmän tilaa kuin LU, KE, SU.

Muuhun käyttöön varattuja sanoja, esim. IF, ei hyväksytä muuttujanimeksi.

RETU\$ REIPAS\$ REKISTERI\$

käsitetään kaikki muuttujaksi RE\$.

Merkkijonomuuttujissa voidaan käyttää samoja yhdistelmiä, mutta nimen loppuun on lisättävä \$-merkki.

A\$, P7\$, MN\$, Z0\$, FP\$

sopivat kaikki jonomuuttujien nimiksi.

Muuttujaniminä ei voida käyttää Basic-kielen sanoja.

MUUTTUJIEN ARVOJEN MÄÄRITTÄMINEN

Miten näitä muuttujia sitten käytetään. Kirjoita seuraava esimerkki:

A = 5 (muista painaa ENTER-näppäintä jokaisen
B = 2 rivin lopussa)

C = A + B

D = D + 3

PRINT A,B,C,D (kirjoita huolellisesti myös pilkut)

Ruudulle ilmestyy nyt:

5 2

7 3

Ensimmäinen rivi merkitsee, että muuttuja A saa arvon 5. Toinen rivi ilmoittaa, että muuttujan B arvoksi tallennetaan 2. Kone itse määrää niiden sijainnin muistissaan. Käyttäjän on annettava vain nimi. Kolmas rivi käskää hakemaan muuttujien A ja B arvoiksi tallennetut luvut, laskemaan ne yhteen ja tallentamaan summan muuttujan C arvoksi. Kun tämä on tehty, muuttujien A ja B alkuperäiset arvot ovat pysyneet muuttumattomina (tässä tapauksessa 5 ja 2) ja C on näiden kahden summa (7). Neljäs rivi saattaa tuntua algebraa osaavista hiukan hämmäntävältä. Tämä johtuu siitä, että Basic-kielessä yhtäläisyysmerkillä (=) on eri merkitys kuin matemaatiikassa. Basicissä = merkitsee: käsittele yhtäläisyysmerkin oikean puolen lause (laske se tarvittaessa) ja sijoita yhtäläisyysmerkin vasemmalla puolella ilmoitettujen muuttujan arvoksi.

Tällaista riviä kutsutaan sijoituslauseeksi, ja siinä vasemman puolen on aina oltava muuttuja. Esimerkiksi lauseke $2 = B + C$ voi olla järkevä algebrassa, mutta Basic-kielessä se on mahdoton.

Käsky $D = D + 3$ merkitsee siis: etsi muuttujan D tämänhetkinen arvo (joka on tässä tapauksessa 0, koska sille ei ole annettu arvoa), lisää siihen 3 ja tallenna se uudelleen muuttujaksi D (lisää siis muuttujan D arvoa kolmella). Tämä saattaa tuntua monimutkaiselta, mutta se on erittäin

käyttökelpoinen (ja hyvin yleinen) käsky tietokoneohjelmissä. Samalla tulee esiin eräs toinen muuttujien ominaisuus - niillä voi olla samanaikaisesti vain yksi arvo. Jos jonkin muuttujan arvo määritetään (jos se siis on esiintynyt sijoituslauseessa vasemmalla puolella), annettu arvo kumoaa aikaisemman arvon ja aikaisempi arvo häviää. Jonkin muuttujan arvo voidaan kuitenkin kopioida (joko toiseen muuttujaan arvoksi tai käyttämällä sitä sellaisissa lausekkeissa kuin $C = A + B$) miten usein tahansa sitä muuttamatta.

Jos kirjoitat:

```
A = B                (muista painaa ENTER
B = 17               jokaisen rivin jälkeen)
D = D + 2
PRINT A,B,C,D
```

Saat vastaukseksi:

```
2          17
7          5
```

Muuttujalla A on nyt edellisen esimerkin muuttujan B arvo, muuttujalla B on uusi arvo (17), ja sekä A:n että B:n aikaisemmat arvot ovat hävinneet. Muuttuja C on pysynyt muuttumattomana. Muuttuja D on nyt 5, koska sen arvo oli aikaisemmassa esimerkissä 3.

Merkkijonomuuttajat toimivat aivan samalla tavoin (on vain muistettava, että nimien pitää aina päättyä merkkiin \$).

Kokeile seuraavaa esimerkkiä:

```
A$ = "TÄMÄ ON "           (muista painaa ENTER
B$ = "HYVIN "             jokaisen rivin jälkeen)
C$ = "PITKÄ JONO"
D$ = A$ + B$ + B$ + B$ + B$ + B$ + B$
D$ = D$ + C$
PRINT D$
```

Ruudulle ilmestyy teksti

```
TÄMÄ ON HYVIN HYVIN HYVIN HYVIN
HYVIN HYVIN PITKÄ JONO
```

Riveillä 1, 2 ja 3 olemme määrittäneet merkkijonomuuttujien A\$, B\$ ja C\$ arvon. Rivillä 4 lisäämme muuttujan A\$ kuusi muuttujaa B\$: Jonomuuttujissa plus-merkki (+) ei merkitse samaa kuin numeromuuttujissa. Se tarkoittaa tekstijonon lisäämistä toisen jonon perään. Tästä käytetään termiä konkatenatio. Rivillä 5 lisäämme muuttujan C\$ juuri määritetyn muuttujan D\$ perään. Tämä on yksi tapa muodostaa lauseita tietokoneen avulla.

MERKKIJONOJA JA LUKUJA EI VOI YHDISTELLÄ

Numeromuuttujat ja jonomuuttujat on muistettava pitää erillään. Numeromuuttujan arvoksi voidaan tallentaa vain lukuja ja jonomuuttujan arvoksi vain merkkijonoja. Seuraavanlaiset lauseet:

D = "JONO"

A\$ = 6

B = A\$ * 2

aiheuttavat virheilmoituksen, ?TM ERROR (type mismatch error).

Plusmerkki (+) on ainoa aritmeettinen merkki, jota voidaan käyttää myös jonoissa ja jonomuuttujissa; kaikki muut (-, *, /, †) antavat virheilmoituksen.

KOMENNOT

Tässä kirjassa komennot on esitetty muusta tekstistä erotetuissa lokeroissa, joissa ne esitetään erikseen ja yksityiskohtaisesti. Useimpien lokeroiden lopussa on pieni esimerkki komennon käytöstä. Tutustu niihin huolellisesti, kokeile niitä ja pyri selvittämään, mitä tapahtuu ennen kuin käsket koneen suorittaa ne. Näillä ohjelmilla pyritään osoittamaan, miten kukin yksittäinen lause toimii, mutta niissä on usein myös hyödyllisiä vihjeitä, joita voit ehkä myöhemmin käyttää omissa ohjelmissasi.

MERKKIJONOMUUTTUJIEN NIMET

Jonomuuttujan arvoksi voidaan tallentaa vain merkkijonoja.

Jonomuuttujan nimi voi olla mikä tahansa kirjainten ja numeroiden yhdistelmä, mutta sen pitää alkaa kirjaimella ja päättyä merkkiin \$.

Kuten numeromuuttujissakin, kone tunnistaa vain nimen kaksi ensimmäistä merkkiä, joten

VASTAUKSIA\$, VA1\$, VA2\$

tulkitaan samaksi muuttujaksi (VA\$).

3. OHJELMAN LAATIMINEN

Tähän mennessä tietokoneesi on oikeastaan vain toistanut siihen juuri kirjoittamasi tiedon. Nyt alamme laatia tietokoneohjelmaa.

Ohjelma on käskykokoelma, jolla koneelle annetaan toimintaohjeet. Basic-ohjelma muodostuu useista riveistä. Rivissä on kaksi osaa: ensiksi rivinumero ja toiseksi yksi tai useampia lauseita, lauseet on erotettava toisistaan kaksoispisteellä (:).

```
PRINT A$:A = 47
```

Tässä on Basic-ohjelma:

```
10 CLS
20 PRINT "mikä sinun nimesi on"
30 INPUT NIMI$
40 I=RND(255):J=RND(9)-1
50 CLSJ
60 PRINT@200+J,NIMI$;
70 SOUND I,2
80 GOTO 40
```

Tässä esimerkissä havaitaan joitakin uusia Basic-komentoja. Niistä ei nyt tarvitse välittää, ne selostetaan myöhemmin. Huomaa Basic-ohjelman muoto - se koostuu useista riveistä, joista jokaisella on oma rivinumeronsa ja vähintään yksi lause (rivillä 40 on kaksi lausetta).

OHJELMAN KIRJOITTAMINEN

Ennen kuin ohjelma voidaan syöttää koneen muistiin, muisti on tyhjennettävä kaikesta, mitä siellä saattaa ennestään olla. Tämä tapahtuu siten, että ruudulle kirjoitetaan NEW ja painetaan ENTER-näppäintä. Kirjoita tämän jälkeen ylläolevan esimerkin jokainen rivi juuri siinä muodossa kuin se on edellä ja paina jokaisen rivin jälkeen ENTER-näppäintä. Huomaat, ettei ENTER-näppäimen painamisen jälkeen tapahdu mitään. Numerolla alkavaa riviä ei suoriteta heti, se vain tallennetaan muistiin. Kun ohjelma suoritetaan, suoritus alkaa rivistä, jolla on pienin rivinumero, josta siirrytään seuraavaksi suurempaan rivinumeroon jne.

Koska ohjelman suoritusjärjestys riippuu rivinumerosta, rivit voidaan kirjoittaa missä järjestyksessä tahansa ja kone järjestää ne oikeaan järjestykseen.

Yritä näppäillä ohjelma koneeseen. Jos huomaat tehneesi virheen ennen kuin painat ENTER-näppäintä, käytä nuolella merkittyä askelpalautinta (←) ja korjaa virhe. Jos havaitset virheen vasta ENTER-näppäintä painettuasi, kirjoita rivi uudelleen. Kone tallentaa vain viimeisimmän version rivistä.

Näppäiltyäsi koko ohjelman kirjoita LIST ja paina ENTER-näppäintä. Näin saat näkyviin juuri näppäilemäsi ohjelmarivit siinä muodossa kuin kone on ne tallentanut. Huomaa, että LIST-käskyn edessä ei ole numeroa. Kone toteuttaa heti komennon, jossa ei ole numeroa. Tarkista, että ohjelma on oikein kirjoitettu, ja korjaa mahdolliset virheet kirjoittamalla koko virheellinen rivi uudelleen.

Nyt voimme vihdoin suorittaa ohjelman. Kirjoita RUN ja paina ENTER-näppäintä.

Ruutu tyhjenee, ja ruudun ylälaitaan ilmestyy teksti, jossa kysytään nimeäsi. Kirjoita nimesi ja paina ENTER-näppäintä.

Kone ryhtyy heti työhön:

Ruudulla välähtelee eri värejä, ja nimi näyttää hyppivän edestakaisin ruudun keskellä. Samanaikaisesti kuuluu epämääräisiä ääniä (mikäli olet muistanut vääntää äänen päälle televisiovastaanottimestasi). Tämä jatkuu vaikka kuinka kauan, ellei sitä keskeytetä. Yksi tapa keskeyttää ohjelma on katkaista virta, mutta se ei ole kovin hyvä ratkaisu, sillä samalla ohjelma häviää. Tämä ohjelma voidaan parhaiten keskeyttää painamalla BREAK-näppäintä. Toinen mahdollisuus on käyttää RESET-näppäintä. Kummassakin tapauksessa ohjelma säilyy koneen muistissa. Kokeile molempia tapoja käynnistettyäsi ohjelman RUN-käskyllä (ja painamalla ENTER).

VAIHE VAIHEELTA

Olemme nyt nähneet, mitä ohjelma tekee. Nyt selvitämme, miten tämä tapahtuu, ja käsittelemme jokaisen rivin erikseen.

10CLS

Koska tämän rivin rivinumero on kaikkein pienin, kone suorittaa sen ensimmäiseksi. CLS-komento käskää puhdistamaan ruudun ja säätämään oikean taustavärin (tavallisesti vihreä).

```
20 PRINT "MIKÄ SUNUN NIMESI ON"
```

Tämä on PRINT-lause, johon on tutustuttu jo aikaisemmin. Tällä rivillä teksti saadaan ilmestymään ruudun ylälaitaan.

```
30 INPUT NIMI$
```

INPUT-komennolla kone jää odottamaan, kunnes pyydetty tieto onsyötetty. Nyt se sijoittaa antamasi tiedon komentoa seuranneen muuttujan arvoksi. Tässä tapauksessa kone käsittää muuttujaksi NI\$, ja muut kirjaimet jäävät huomioimatta.

Yleensä on kuitenkin hyvä käyttää pitempiä muuttujan nimiä, jotta ohjelma on helpompi ymmärtää. Pitempi nimi auttaa myös muistamaan, mihin muuttuja viittaa

```
40 I=RND(255):J=RND(9)-1
```

Rivi 40 osoittaa, miten samalle riville voidaan kirjoittaa useampia lauseita. Huomaa, että lauseet on erotettu toisistaan kaksoispisteellä (:). Tällä rivillä esiintyy myös uusi komento, RND. Tällä komennolla kone antaa satunnaisluvun. Satunnaisluku on ikään kuin hatusta nostettu satunnaisesti valittu luku. RND-komennon jälkeen annettu suluisa oleva luku kertoo koneelle, miltä alueelta satunnaisluku valitaan. Ensimmäisessä lauseessa I=RND(255), RND(255) merkitsee, että koneen on valittava satunnaisesti jokin kokonaisluku alueelta 1-255 ja tallennettava se muuttujan I arvoksi. Toisessa lauseessa satunnaisluvun raja-arvoiksi annetaan 1-9, mutta ennen kuin luku tallennetaan muuttujan J arvoksi, siitä vähennetään 1. Tämä tarkoittaa, että J on jokin luku nollan ja 8:n välillä:

50 CLS(J)

Tällä rivillä tyhjennetään ruutu. Taustaväri riippuu nyt kuitenkin muuttujan J arvosta. Käytettävissä on yhdeksän väriä, jotka on numeroitu 0-8. Tämä rivi saa ruudun värin vaihtumaan.

60 PRINT @200+J,NIMI\$;

Tämä on PRINT-lauseen kehittyneempi muoto. Se käskää koneen kirjoittamaan muuttujan NIMI\$ arvon (tässä tapauksessa siis sinun nimesi) alkaen määrätystä paikasta ruudulla. Tässä tapauksessa paikka on 200+J, joka siis sijoittuu paikkojen 200 ja 208 välille. Paikka 200 on rivillä 7, 8 merkin päässä vasemmasta reunasta. (PRINT @-komentoa käsittelevässä loke-rossa on esitetty merkkipaikkojen sijainnit ruudulla). Koska J:n arvo muuttuu, nimesi ikäänkuin hyppii edestakaisin rivillä.

70 SOUND I,2

Tällä rivillä saadaan aikaan epämääräiset äänet. SOUND-lause käskää tietokoneen käyttää äänigeneraattoriaan äänen tuottamiseksi. Komentoa seuraavat kaksi lukua määräävät äänen laadun ja keston.

80 GOTO 40

GOTO-lause käskää hyppäämään annetulle riville (tässä ohjelmassa siis 40).

LIST

LIST-komennolla luetaan ruudulle tällä hetkellä muistissa oleva ohjelma. LIST ei ole ohjelmakäskey, eikä sen eteen siis merkitä rivinumeroa.

Jos ohjelma on niin pitkä, ettei se sovi ruudulle, listaa-
minen voidaan keskeyttää painamalla samanaikaisesti SHIFT- ja
@ -näppäimiä, mutta se on tehtävä nopeasti. Listausta voi-
daan jatkaa painamalla mitä tahansa näppäimistöön näppäin-
tä (ei kuitenkaan SHIFT-, BREAK- eikä CLEAR-näppäintä).

Halutessasi näkyviin vain jonkin ohjelman osan, voit kirjoit-
taa:

LIST n_1 - n_2

jolloin n_1 ja n_2 ovat kaksi eri rivinumeroa (n_2 :n on oltava suurempi kuin n_1).

LIST 40-100

tällä käskyllä saat näkyviin kaikki ohjelman rivit riviltä
40 riville 100

LIST -80

listaa kaikki ohjelman rivit alusta riville 80.

LIST 120-

listaa ohjelman alkaen riviltä 120 ohjelman loppuun.

RUN

RUN-komennolla aloitetaan ohjelman suoritus.

Sen edessä ei ole rivinumeroa.

Jos haluat aloittaa ohjelman jostakin muusta kohdasta kuin alusta, voit tehdä sen näppäilemällä:

RUN rivinumero

jolloin rivinumero on sen rivin numero, josta haluat aloittaa.

RUN 250.

NEW

NEW-komento tyhjentää muistin ja nolaa kaikki muuttujat.

Tällä komennolla ei ole rivinumeroa.

NEW-komennolla on hyvä aloittaa lähettäessä laatimaan uutta ohjelmaa, jotta muistissa mahdollisesti oleva vanha ohjelma ei pääsisi sekoittamaan uutta ohjelmaa.

Sijoituslause

Sijoituslauseella annetaan muuttujalle arvo.

Sijoituslauseen muoto on seuraava:

LET muuttuja=lauseke

Lauseen LET-osa on standardi Basic-käskey, mutta tässä tietokoneessa se ei ole välttämätön, joten sitä ei käytetä tämän kirjan ohjelmalistaissa.

Muuttuja voi tässä olla mikä tahansa muuttujan nimi.

Lauseke voi olla vakio, muuttuja tai molempien yhdistelmä, jolloin ne on liitetty toisiinsa operaattoreilla (+, -, , jne.). Koska merkkijonoja ja numeromuuttujia ei voida yhdistää toisiinsa, sekä muuttujien, että lausekkeen on oltava samaa tyyppiä.

Yhtäläisyysmerkki (=) ei merkitse samaa kuin algebrassa. Basic-kielessä sen nimeksi sopii paremmin 'saa arvon'. Tämä tarkoittaa, että sama muuttuja voi esiintyä sijoituslauseen kummallakin puolella, esimerkiksi:

40 X=X+1

mikä tarkoittaa, että tämänhetkiseen X:n arvoon on lisättävä 1 ja tallennettava saatu arvo jälleen muuttujan X arvoksi. Tällä lauseella X:n arvoa siis lisätään yhdellä.


```
10 S=0:N=0:CLS5
20 PRINT@70,"ANNA LUKU";
30 INPUT X:CLS5
40 S=S+X:N=N+1
50 PRINT @198,"OLET ANTANUT";N;"LUKUA";
60 PRINT@262,"KESKIARVO ON";S/N;
70 GOTO 20
```

PRINT

PRINT-lausetta käytetään, kun halutaan tulostaa ruudulle. Sillä voidaan tulostaa vakioita, muuttujan arvo, merkkijonoja ja myös laskea matemaattisia lausekkeita.

Jos PRINT-lauseeseen sisältyy useampia tietoalkioita, ne on erotettava toisistaan joko pilkulla (,) tai puolipisteellä(;).

Pilkulla erotetut alkiot tulostetaan kahtena sarakkeena, jotka molemmat ovat 15 merkin pituisia. Jos ensimmäinen tietoalkio on pitempi kuin 15 merkkiä, se kirjoitetaan täydellisenä, ja seuraava alkio tulostetaan seuraavalle riville.

Puolipistettä käytettäessä tulostus on tiivistetty. Merkkijonot kirjoitetaan aivan peräkkäin ja numeroalkioiden molemmilla puolilla on välilyönti. Puolipistettä käytettäessä kursori pysähtyy viimeisen kirjoitetun merkin kohdalle ja on valmiina jatkamaan kirjoittamista ohjelman siirtyessä seuraavaan PRINT-lauseeseen.

Ellei PRINT-lause sisällä lainkaan alkioita, kirjoitetaan tyhjä rivi.

10 CLS

20 PRINT

30 PRINT "VVVVVVVVPRINT-lause"

40 PRINT "VVVVVVVVVESIMERKKI"

50 PRINT

60 PRINT "SARAKE YKSI","SARAKE KAKSI"

70 PRINT 14.2,13.7

80 PRINT 1,2,7,11

90 PRINT

100 A\$="TIIVISTETTY":B=3

110 PRINT , "TOINEN RIVI"

130 PRINT A\$;"TULOSTUS RIVILLÄ";B

140 PRINT

150 PRINT "TÄMÄ TULEE NÄKYVIIN";

160 PRINT "VYHTENÄ RIVINÄ"

170 PRINT "VVESIMERKKI",

180 PRINT "VVLOPPU"

INPUT

INPUT-lauseeseen päästyään ohjelma pysähtyy ja odottaa, että näppäimistöltä syötetään pyydetty tieto. INPUT-komentoa pitää seurata vähintään yksi muuttujanimi tai useampia nimiä, jotka on erotettu toisistaan pilkuilla.

```
25 INPUT A,B,F$,H7
```

Ylläoleva lause pyytää näppäilemään 4 tietoalkiota. Ne voidaan näppäillä yksi kerrallaan painamalla jokaisen jälkeen ENTER-näppäintä tai luettelona, jolloin ne erotetaan toisistaan pilkuilla, esim.

```
146.2,78.1,JONO,3 [ENTER]
```

Ennen INPUT-lausetta on aina hyvä kirjoittaa jokin viesti, joka ilmoittaa, mitä tietoja on annettava. Tämä voidaan tehdä joko PRINT-lauseella tai sisällyttämällä se INPUT-lauseeseen seuraavasti:

```
35 INPUT "ANNA KAKSI LUKUA";N1,N2
```

Huomaa, että merkkijono on erotettu puolipisteellä syötettävien muuttujien luettelosta.

Syötettävien tietojen on oltava oikeaa tyyppiä (merkkijonomuuttujan arvoksi jono ja numeromuuttujan arvoksi luku).

Muussa tapauksessa ohjelma keskeytyy, ruutuun ilmestyy virheilmoitus ?REDO ja arvot on syötettävä uudelleen.

Merkkijonomuuttujia syötettäessä ei tarvitse käyttää lainausmerkkejä; sekä "JONO" että JONO ovat hyväksyttäviä muotoja. Kiinnitä esimerkkiohjelmassa huomiota seuraaviin asioihin:

1. riveillä 100 ja 170 INPUT-komento pysäyttää ohjelman kunnes olet valmis. A\$ ei sisällä mitään.
2. jonomuuttujia C\$,P\$ ja T\$ on käytetty, jotta samaa ilmoitusta ei tarvitse kirjoittaa useita kertoja.

```
10 CLS:T$="TÄMÄ ON INPUT-ESIMERKKI"
20 P$="PAINA ENTER KUN HALUAT JATKAA"
30 C$="ANNA 4 LUKUA,"
40 PRINT:PRINT T$:PRINT
50 PRINT C$;"PAINA"
60 PRINT "ENTER JOKAISEN LUVUN JÄLKEEN"
70 INPUT A,B,C,D
80 PRINT:PRINT "ANNOIT ARVOT:"
90 PRINT A;B;C;D:PRINT
100 PRINT P$:INPUT A$:CLS
110 PRINT:PRINT T$:PRINT
120 PRINT "NYTV";C$;"VEROTELTUINA"
130 PRINT "PILKUILLA JA PAINA ENTER"
140 INPUT A,B,C,D
150 PRINT:PRINT "NYT LUVUT OLIVAT:"
160 PRINT A;B;C;D
170 PRINT:PRINT P$:INPUT A$:CLS
180 PRINT:PRINT T$:PRINT
190 INPUT "ANNA MERKKIJONO JA LUKU";B$,N
```

```
200 PRINT:PRINT "ANNOIT MERKKIJONON: ";PRINT
210 PRINT B$
220 PRINT:PRINT "JA LUVUN: ";N
230 PRINT:PRINT "ESIMERKKI PÄÄTTYY"
```

Ohjelma palaa siis riville 40, ja valitsee jälleen uuden satunnaisluvun sekä I:n että, J:n arvoksi. Koska J määrää taustan värin, väri muuttuu, kun ohjelma jälleen etenee riville 50, ja koska myös I on muuttunut, ääni muuttuu rivillä 70. Riviltä 80 ohjelma palaa taas riville 40 ja valitsee uuden satunnaisluvun ja niin edelleen (vaikka kuinka kauan, ellei paineta BREAK- tai RESET-näppäintä).

MUUTOSTEN TEKEMINEN

Tämä oli ensimmäinen ohjelma. Sillä ei saatu mitään kovin hyödyllistä aikaan, mutta sen avulla pääsimme alkuun ja opimme muutamia uusia käskyjä. Käskytyypit on kuvattu yksityiskohtaisemmin tekstistä erotetuissa lokeroissa. (Jokaisessa lokerossa on myös esimerkkiohjelma, jota on hyvä kokeilla). Ensimmäinen ohjelma ei ehkä tyydyttänyt sinua ja haluat tehdä siihen joitakin muutoksia. Koneeseen kirjoitettua ohjelmaa on helppo muuttaa. Koska Basic-ohjelman suoritusjärjestys määräytyy rivinumeron mukaan, rivi voidaan

muuttaa kirjoittamalla samalla rivinumerolla uusi rivi. Myöhemmin kirjoitettu rivi syrjäyttää aikaisemman. Koeta muuttaa riviä 70.

70 SOUND I,K (Muista painaa ENTER-näppäintä)

Ohjelmaan voidaan lisätä rivi antamalla sille numero, joka sijoittaa sen oikeaan kohtaan. Kirjoita:

45 K = RND(20)

Koska tämän rivin numero on 45, se tulee rivien 40 ja 50 väliin.

(Kun laadit omia ohjelmiasi, voit valita rivinnumeroiksi mitä numeroita tahansa 0 ja 63999 väliltä. Yleensä rivit on hyvä numeroida 10 välein. Siis 10, 20, 30 jne., jotta jäisi tilaa lisätä ylimääräisiä rivejä tarvittaessa).

Yritä ajaa muutettu ohjelma (kirjoita RUN). Uusi rivi 45 valitsee toisen satunnaisluvun, nyt väliltä 1-20. Muutettu rivi 70 käyttää tätä satunnaislukua (muuttujan K arvo) äänen keston määrittämisessä.

OHJELMAN RAKENNE

Ohjelmarivejä on helppo kirjoittaa sisään näppäimistöltä. Vaikeudet ilmaantuvat yleensä vasta jälkeenpäin, etenkin ohjelmien pidetessä. Kirjoittamisen pitäisikin olla ohjelman laatimisen viimeinen vaihe.

Kirjoita ensin kynällä paperille mitä haluat tehdä. Jaa tehtävä erillisiin osakokonaisuuksiin. Monet tietokoneella ratkaistavat tehtävät jakautuvat helposti ainakin kolmeen osaan:

1. Valmistelu, otsikot ja ohjeet tiedon syöttöä varten
2. Laskutoimitukset
3. Tulostus

Käsittele nyt jokainen osakokonaisuus erikseen ja jaa se edelleen pienempiin osiin, kunnes olet päätenyt yksinkertaiseen toimintoon, esimerkiksi 'lisää laskuriin 1'. Järjestä nämä toiminnot järkevään järjestykseen ja kirjoita ne muistiin. Siirry sitten seuraavaan osaan. Kun olet käsitellyt kaikki osakokonaisuudet, sinulla on jokaisessa yksittäisessä osassa kokoelma erilaisia ohjelma-askeleita (joista osa voi olla hyvinkin alkeellisia). Kenen tahansa, joka lukee näin syntyneen tuloksen, pitäisi nyt pystyä ratkaisemaan tehtävä -olipa se mikä tahansa- yksinkertaisin laskutoimituksin. Tällaista toimintasuunnitelmaa kutsutaan ATK-terminologiassa algoritmiksi eli askelkaavioksi. Nyt askelkaavio on vain käännettävä kielelle, jota kone ymmärtää. Ihannetapauksessa jokainen algoritmin askel pitäisi voida kääntää yhdeksi riviksi Basic-ohjelmassa.

Kääntämisen jälkeen jokainen osa voidaan testata erikseen ja tarkistaa, ehtä se tekee, mitä sen pitäisikin tehdä, ennen kuin osat kootaan ohjelmakokonaisuudeksi. Kun ohjelma

vihdoin toimii niinkuin sen pitäisikin, sitä ei pidä juhlia polttamalla suunnittelussa käytetyt paperit. Lopullinen suunnitelma, algoritmi, on syytä säilyttää. Jotkin sen osat voivat olla käyttökelpoisia muita ohjelmia laadittaessa, ja virheitäkin saattaa ilmaantua vasta sitten, kun olet jo aikaa sitten unohtanut, miten olet toiminut. Myös itse ohjelmaan kannattaa kirjoittaa kommentteja, jotka auttavat muistamaan, mitä tapahtuu, Basic-ohjelmassa tämä voidaan tehdä REM-lauseella. REM-lause ei tee varsinaisesti mitään. Kone jättää huomioimatta kaiken, mikä seuraa REM-lausetta. Jos siis rivillä on useampia kuin yksi lause, REM-lauseen on oltava viimeisenä. REM-lauseen lyhenne on yksinkertainen lainausmerkki ('). Tämä merkki on näppäimistöllä.

```
10 REM KESKIARVON LASKEVA OHJELMA
140 A=BxC:GOTO 15:REM TAKAISIN ALKUUN
48 K=K+1:'KIERROSLASKURI
```

Vaikka REM-lauseet vievätkin jonkin verran tilaa muistissa, niitä ei ole viisasta jättää kokonaan pois. Ellei ohjelmaan lisätä mitään kommentteja, sen ymmärtäminen esimerkiksi vuoden kuluttua voi olla hyvinkin vaikeaa.

Nämä ohjeet saattavat tuntua liiankin pikkutarkoilta, mutta yleisesti tiedetään, että ohjelmoijat kuluttavat enemmän aikaa ohjelmointivirheiden etsimiseen kuin itse ohjelman laatimiseen. Jos toimit systemaattisesti annettujen ohjeiden mukaisesti, virheiden todennäköisyys on pienempi, tai ne on ainakin helpompi löytää.

SOUND

SOUND-lauseella saadaan aikaan määrätynkorkuinen ja -kestoinen ääni. Siinä tarvitaan kaksi argumenttia:

SOUND korkeus,kesto

Korkeus on jokin luku 1 ja 255 välillä. Matalin ääni on 1 ja korkein 255. Pianon keskiC:tä vastaava ääni saadaan antamalla korkeudeksi 89.

Kesto voi olla mikä tahansa luku välillä 1 ja 255. Kesto=16 antaa äänen, jonka kesto on noin 1 sekunti.

```
10 CLS
```

```
20 PRINT@7,"ÄÄNIESIMERKKI"
```

```
30 PRINT@64,"ANNA LUKU VÄLILTÄ 1-255";
```

```
40 PRINT@96,"JOKA ILMOITTAÄ SÄVELKORKEUDEN";:INPUT H
```

```
50 PRINT@192,"ANNA ÄÄNEN KESTON ILMOITTAVA LUKU";
```

```
60 INPUT V:CLS (RND(9)-1)
```

```
70 SOUND H,V
```

```
80 GOTO 10
```

PRINT@

PRINT@-lauseetta käytetään haluttaessa sijoittaa tulostus johonkin määrättyyn kohtaan ruudulla. Ruudulla on tätä varten 16 x 32 eli yhteensä 512 numeroitua merkkipaikkaa.

Ks. liitteenä olevasta ruudukosta, miten nämä paikat on numeroitu.

PRINT@-lauseen muoto on

PRINT@lauseke, tulostuslista

Lauseke voi olla luku, muuttuja tai matemaattinen lauseke, jonka arvon pituus on 0-511 merkkiä.

Tulostuslista on sama kuin tavallisessa PRINT-lauseessa.

Se voi olla lukuja, muuttujia, merkkijonoja tai lausekkeita, jotka on erotettu toisistaan pilkuilla tai puolipisteillä.

Jos ajatellaan, että ruutuun on ikäänkuin vedetty viivotimella 16 riviä, lauseke

PRINT@32*(RIVI-1),A

kirjoittaa A:n arvon jonkin rivin alkuun ruudulla. Rivi riippuu muuttujan rivi arvosta, joka voi olla jokin luku välillä 1-16.

Seuraavassa esimerkissä on käytetty tarkoituksellisen monimutkaista menettelyä suhteellisen yksinkertaiseen tulokseen pääsemiseksi. Yritä selvittää, mitä esimerkkiohjelma tekee, aja ohjelma ja kirjoita sitten yksinkertaisempi versio saman tuloksen aikaansaamiseksi.

Huomaa puolipisteen käyttö lauseiden lopussa, jotta rivin loppu ei häviäisi. (ellet usko, koeta: kirjoita rivi 110 ilman puolipistettä).

```
10 CLS:P$="PRINT@-KÄSKYÄ":N=1
20 RIVI=12 :A$="RUUDULLE"
30 PRINT@32*(RIVI-1)+15,"SIJOITETTAESSA";
40 PRINT@447 + 12,"MINNE TAHANSA";
50 PRINT@259,"TÄSTÄ";:PRINT@265,"NÄHDÄÄN";
60 PRINT@32*(RIVI-1)+8,"KÄYTTÄÄ";:PRINT@13,"SIVU";
70 PRINT@447 + 20, A$;
80 PRINT@271,"KUINKAV";P$
90 PRINT@32*(RIVI-1)+4,"VOIDAAN";
100 PRINT@134,P$;"VESIMERKKI";
110 PRINT@450,"ALKIOITA";
120 PRINT@18,N
130 GOTO 130
```

Viimeinen rivi (130) saa koneen kiertämään "loputonta silmukkaa", jossa ei tapahdu mitään. Tämä estää OK-merkin ilmes-
tymisen ruudulle. Paina BREAK-näppäintä ohjelman lopetta-
miseksi.

CLS

CLS-käskyä käytetään ruudun tyhjentämiseen ja taustavärin säätämiseen. Normaali taustaväri on vihreä. Jos käytät CLS-käskyä yksinään, ruudun väri muuttuu vihreäksi.

Jos haluat muuttaa ruudun taustaväriä, lisää CLS-käskyn perään jokin numero nollan ja kahdeksan väliltä (CLS2).

Värit ovat:

0-musta	1-vihreä	2-keltainen
3-sininen	4-punainen	5-valkoinen
6-turkoosi	7-sinipunainen	8-oranssi

Värien sävyt riippuvat televisiovastaanottimesta.

Taustaväristä riippumatta, kone kuitenkin kirjoittaa tekstin aina joko mustana tai vihreänä.

10 CLS

20 PRINT@0,"TAUSTAVÄRIESIMERKKI";

30 PRINT@192,"ANNA LUKU 0 ja 8 VÄLILTÄ";

40 INPUT C

50 CLS C

60 PRINT@288,"TÄMÄ ON TAUSTAVÄRI: ";C;

70 GOTO 20

REM

REM-lausetta käytetään lisättäessä kommentteja ohjelmaan.
Tietokone jättää huomioimatta kaiken, mikä seuraa REM-
komentoa (tai sen lyhennystä, ') samalla rivillä.

10 REM TÄMÄ ON KOMMENTTIRIVI

35 0 = 2 * PI * R: 'LASKE KEHÄN PITUUS

GOTO

GOTO-lauseen muoto on seuraava:

GOTO rivinumero

Rivinumero on aina luku (ei muuttuja), ja sen pitää esiintyä myös jossakin muualla samassa ohjelmassa. Ellei rivinumeroa löydy, ohjelma keskeytyy ja näkyviin tulee virhemerkintä ?UL ERROR (määrittelemätön rivi).

GOTO-lause suoritetaan heti. Samalle riville ei siis kannata kirjoittaa mitään muuta lausetta GOTO-lauseen jälkeen, sillä ohjelma ei ehdi koskaan niin pitkälle.

20 CLS

30 GOTO 60

40 PRINT "RIVILLÄ 40"

50 GOTO 80

60 PRINT "RIVILLÄ 60"

70 GOTO 40

80 PRINT "LOPPU (RIVI 80)"

RND

RND-toiminnolla muodostetaan satunnaislukuja.

RND on toiminto eli funktio. Basicissä funktio on sellainen, joka yhden tai useamman luvun avulla suorittaa laskutoimituksen, josta saadaan tulokseksi yksinkertainen arvo. Funktiossa käytettyjä lukuja kutsutaan argumenteiksi ja ne merkitään aina sulkumerkkien sisään funktionimen jälkeen. Funktion tulos palautetaan ohjelmaan.

RND-funktio antaa satunnaisluvun, joka riippuu funktion argumentin arvosta.

Jos argumentin arvo on 0 (RND(0)), funktio antaa arvon, joka on nollan ja yhden välillä.

Jos argumentin arvo on suurempi kuin 0 (RND(6)), funktio antaa kokonaisluvun luvun 1 ja argumentin arvon väliltä. (RND(6)) antaa tulokseksi 1, 2, 3, 4, 5 tai 6. Tarkkaa lukua ei voida tietää, koska luku on satunnaisluku:

```
10 CLS
20 PRINT@8,"ANNA LUKU";:INPUT N
30 CLS:PRINT@194,"3 SATUNNAISLUKUA ALUEELTA 1-";N
40 PRINT@270,RND(N)
50 PRINT@302,RND(N)
60 PRINT@334,RND(N)
70 GOTO 20
```

OHJELMOINTIESIMERKKI

Tehtävä: Käytä tietokonetta ikäänkuin heittäisit kahta noppaa.

OSA 1 - Otsikoiden ja ohjeiden tulostaminen

1. tyhjennä ruutu
2. kirjoita otsake
3. kirjoita ohjeet
4. kirjoita otsakkeet: 1. noppa, 2. noppa.

OSA 2 - Kahden nopan silmälukujen etsiminen

(Noppaa heitettäessä saadaan satunnaisesti mikä tahansa kuudesta silmäluvusta)

1. ensimmäinen noppa on satunnaisluku lukujen 1 ja 6 välillä.
2. toinen noppa on satunnaisluku lukujen 1 ja 6 välillä.

OSA 3 - Tuloksen kirjoittaminen

1. kirjoita 1. nopan ja 2. nopan silmäluku

OSA 4 - Ohjelman lopetus ja toisto haluttaessa

1. pysäytä ohjelman suoritus
2. uusi kierros?
3. toista osat 1, 2 ja 3 tarvittaessa

Käännämme nyt saman Basic-ohjelmaksi (johon on lisätty kommentit(:

```
10 'NOPANHEITON SIMULointi
20 '
30 'OSA 1
40 CLS:REM TYHJENTÄÄ RUUDUN
```



```

50 PRINT "NOPANHEITTO":'OTSAKE
60 PRINT
70 PRINT "PAINA BREAK JOS HALUAT LOPETTA":'OHJE
80 PRINT
90 PRINT "1. NOPPA","2.NOPPA":'OTSAKKEET
100 PRINT
110 REM OSAN 1 LOPPU
120 REM
130 REM OSA 2
140 T1=RND(6):' 1. NOPAN PYÖRITYS
150 T2=RND(6):' 2. NOPAN PYÖRITYS
160 'OSAN 2 LOPPU
170 '
180 'OSA 3
190 PRINT T1,T2:'TULOKSEN NÄYTTÖ
200 'OSAN 3 LOPPU
210 '
220 'OSA 4
230 INPUT "HEITÄ NOPPAA PAINAMALLA ENTER";A$:
240 GOTO 40:'PALUU ALKUUN
250 'OSAN 4 LOPPU

```

Tässä ohjelmassa ei tarvita kaikkia kommentteja. Kirjoita oma versiosi ohjelmasta.

'Oikeaa' ohjelmaversiota ei ole olemassa: Oikea versio on ohjelma, joka toimii. Sama ongelma voidaan ehkä ratkaista tyylikkäämmin tai tehokkaammin, ja yleensä lyhyempi ohjelma on nopeampi ja vaatii vähemmän muistitilaa.

Päätämme tämän kappaleen esittämällä tiivistetyimmän version samasta ohjelmasta. Huomaa, että ohjelma on tässä koottu eri järjestykseen, mutta tulos on sama.

```
10 'NOPANHEITTOA SIMULOIVA
20 'OHJELMA
30 'OSA 1
40 CLS:PRINT@4,"NOPANHEITON SIMULOINTI";
50 PRINT@166,"1.:";PRINT@177,"2.";
60 PRINT@198,"NOPPA";:PRINT@209,"NOPPA";
70 PRINT@450,"PAINA BREAK JOS HALUAT LOPETTAA:";
80 'OSA 4
90 PRINT@355,"HEITÄ NOPPAA PAINAMALLA ENTER";
100 INPUT A$
110 'OSA 2
120 T1=RND(6):T2=RND(6)
130 'OSA 3
140 PRINT@264,T1;:PRINT@275,T2;
150 GOTO 90
```

4. HYVÄ JÄRJESTYS

NAUHURIN KYTKEMINEN

Jotkut ohjelmat ovat jo olleet melko pitkiä, ja on aika työstä kirjoittaa ohjelma aina uudelleen, kun sitä halutaan käyttää. Ohjelmia on suhteellisen helppo tallentaa kasettinauhalle ja kutsua tarvittaessa takaisin koneen muistiin.

Tätä varten tarvitaan kasettinauhuri ja liitännäjohto.

Mitä tahansa hyvälaatuista kasettinauhuria voidaan käyttää, jos siinä on seuraavat toiminnot:

1. mahdollisuus nauhoittaa ulkopuolisesta lähteestä (liitännä, joka on tavallisesti merkitty MIC, AUX tai LINE IN),
2. liitännä kovaäänistä tai kuulokkeita varten (merkitty EAR, MONIT, L/S tai SPKR)
3. kaukosäätöinen käynnistyskytkin (pieni liitännä, joka on yleensä merkitty REM ja sijaitsee mikrofoni-liitännän vieressä)
4. verkkovirtaliitännä. Tämä ei ole välttämätön, mutta heikot paristot voivat aiheuttaa häiriöitä ohjelmien tallentamisessa ja lataamisessa.

Nauhuri kytketään tietokoneeseen liittämällä DIN-pistoke (pyöreä, viisinaipainen pistoke) tietokoneen vasemmalla puolella olevaan liitännään, joka on merkitty TAPE.

Johdon toisessa päässä olevat kolme pistoketta kytketään nauhuriin seuraavasti:

1. pienin pistoke sopii liitántään, joka on merkitty REM (kaukosäätöinen käynnistyskytkin)
2. suuri pistoke, jossa on harmaa tai punainen johto sopii liitántään, joka on merkitty MIC, AUX tai LINE IN. Käytä aina AUX-liitántää, jos nauhurissa on sekä AUX- että MIC-liitántä.
3. toinen suuri pistoke, jossa on musta tai valkoinen johto sopii liitántään, joka on merkitty EAR (kuulokkeet), MONIT, L/S tai SPKR.

Pane nauhuri päälle, aseta nauha sisään ja kelaa se alkuun. Käännä äänenvoimakkuudensäädin kohtaan 6 (tai hiukan yli keskikohdan). Nyt voit tallentaa ohjelmasi.

Jos haluat kelata nauhaa eteen tai taaksepäin nauhurissa, sinun on ensin irrotettava REM-pistoke. Muista asettaa pistoke takaisin paikoilleen lopetettuasi kelaamisen. REM-pistoke voi myös olla paikoillaan kelauksen aikana, mutta tällöin on näppäiltävä MOTOR ON ja painettava ENTER-näppäintä ennen kuin aletaan kelata eteen- tai taaksepäin. Kun kelaaminen on lopetettu, näppäillään MOTOR OFF ja painetaan ENTER-näppäintä, jolloin nauhuri on jälleen yhteydessä tietokoneeseen. Jos haluat käyttää kasettinauhan uudelleen, vanha materiaali on ensin pyyhittävä pois.

OHJELMAN TALLENTAMINEN KASETILLE

Näppäile ohjelma koneeseen ja suorita se tarkistaaksesi, että se toimii oikein. Jatka sitten seuraavasti:

1. tarkista, että nauhurissa on tyhjä kasetti ja että se on paikoillaan vähintään 10 sekuntia,
2. paina PLAY- ja RECORD-näppäimet yhtäaikaa alas siten, että ne lukkiutuvat,
3. näppäile käsky:

CSAVE "OHJELMA1" (paina ENTER-näppäintä)

Nimi "OHJELMA1" voidaan korvata millä tahansa haluamallasi nimellä. Nimen pitää alkaa kirjaimella ja se voi olla enintään 8 merkin pituinen. Kun painat ENTER nauhurin moottori käynnistyy ja ohjelma nauhoitetaan. Jonkin ajan kuluttua ruudulle ilmestyy OK ja nauhurin moottori pysähtyy.

Ohjelma on edelleen myös tietokoneen muistissa, ja vain sen kopio on tallennettu nauhalle. Olet nyt tallentanut nauhalle ohjelman, jonka nimi on OHJELMA1 (tai minkä nimen sille sitten annoitkin).

Voit nyt kokeilla, onko ohjelma tallentunut kunnollisesti myöhempää käyttö varten. Kelaa takaisin ohjelman alkuun, näppäile SKIPF ja paina ENTER-näppäintä. Ellei ohjelma ole tallentunut, kirjain S jää ruudulle (muista painaa kasettinauhurin PLAY-näppäintä). Jos tallentamisessa on tapahtunut jokin virhe, ruudulle tulee virheilmoitus I/O ERROR. SKIPF-käskyn jälkeen vanha ohjelma on edelleen tietokoneenmuistissa, ja se voidaan tallentaa sieltä uudelleen, ellei se onnistunut ensimmäisellä kerralla:

OHJELMAN LATAAMINEN MUISTIIN

Kun tallennettu ohjelma ladataan uudelleen tietokoneeseen, kirjoitetaan ensin NEW, jotta muistissa mahdollisesti oleva ohjelma pyyhkiytyisi pois. Tämän jälkeen toimitaan seuraavasti:

1. nauha kelataan alkuun
2. painetaan PLAY-näppäintä niin, että se lukkiutuu alas
3. kirjoitetaan käsky

CLOAD "OHJELMA1" (paina ENTER-näppäintä)

Nauhurin moottori käynnistyy ja kirjain S ilmestyy tyhjän ruudun vasempaan yläkulmaan. Se osoittaa, että tietokone etsii ohjelmaa. Kun ohjelma on löytynyt, S häviää ja tilalle ilmestyy F OHJELMA1

Kun ruudulle tulee OK ja nauhurin moottori pysähtyy, ohjelma on ladattu tietokoneen muistiin. Voit tarkistaa sen näppäilemällä LIST.

Ellei ohjelma ole muistissa, tai jos ruudulle ilmestyy I/O ERROR, kasettinauhurisi saattaa vaatia erilaisia säätöjä. Tarkista, että kytkennät on tehty oikein ja toista sitten lataus- ja listaustoimenpiteet käyttäen erilaista (suurempaa) äänenvoimakkuutta, kunnes lataaminen onnistuu.

Ellei lataaminen onnistu, voit kuunnella kasettia, kun olet ensin irrottanut nauhuriin tulevat johdot ja heikentänyt hiukan äänenvoimakkuutta. Ensin pitäisi kuulua kaksi piipitystä ja tämän jälkeen tiheää 'surinaa'. Se on tallennettu ohjelma. Ellei mitään kuulu, ohjelma ei ole tallentunut nauhalle. Yritä uudelleen tarkistettuasi, että kaikki kytkennät on tehty oikein.

USEAMPIEN OHJELMIEN TALLENTAMINEN

Jos samalle nauhalle tallennetaan useampia kuin yksi ohjelma, on kyettävä estämään ohjelman tallentuminen jo nauhalla olevien ohjelmien päälle. Nauha pitää siis kelata viimeksi nauhoitetun ohjelman ohi. Tämä tapahtuu seuraavasti:

1. kelaa nauha ensin alkuun,
2. paina PLAY-katkaisinta niin, että se lukkiutuu alas.
3. kirjoita käsky:

```
SKIPF "OHJELMA1"
```

Moottori käynnistyy, tietokone etsii (S) ohjelman, löytää sen (F), lukee sen ohi, pysäyttää moottorin ja ilmoittaa ruudulla OK.

4. kelaa vielä eteenpäin pari sekuntia, jotta ohjelmien väliin jää tauko,
5. paina STOP-näppäintä,
6. paina samanaikaisesti PLAY ja RECORD-näppäimiä, anna uudelle ohjelmalle nimi ja tallenna se CSAVE-käskyllä.

Kun ohjelma on tallennettu, nauhuri on ohittanut sen loppukohdan, joten uusia ohjelmia voidaan nyt näppäillä ja tallentaa edellisen ohjelman perään.

VIHJEITÄ LUOTETTAVAA TALLENTAMISTA VARTEN

1. Käytä nauhuria verkkovirralla tasaisen nopeuden varmistamiseksi.
2. Käytä uusia, korkealuokkaisia kasetteja. Lyhyet nauhat (C30 tai C20) ovat parhaita. Pitempien nauhojen käyttöä ei suositella.

3. Aloita ohjelman etsiminen aina nauhan alusta. Älä luota nauhurin laskijaan.
4. Älä jätä PLAY- tai RECORD-näppäintä alas. Paina aina STOP-näppäintä lopetettuasi tallentamisen tai lataamisen.
5. Kelaa kasetit alkuun ennen kuin poistat ne nauhurista.
6. Kiinnitä kasetteihin nimilappu heti tallentamisen jälkeen ja poista tärkeitä ohjelmia sisältävien kasettien takareunasta pieni muovipala, joka estää kasetin uudelleenäänittämisen.
7. Älä säilytä kasetteja television äläkä kaiutinkaappien päällä tai läheisyydessä, sillä niiden magneettikenttä voi vahingoittaa ohjelmia.

Vahinkoja voi sattua, vaikka oletkin varovainen. Joidenkin ohjelmien laatiminen on saattanut vaatia paljonkin aikaa ja vaivaa, joten niistä on syytä ottaa toinen kopio toiselle kasetille.

TOIMITUSOHJELMA (EDITOR)

Ohjelmarivien pidentyessä kirjoitusvirheiden todennäköisyys suurenee. Tähän asti virheet on korjattu vain kirjoittamalla koko rivi uudelleen. Esittelemme nyt toimitusohjelman eli editorin, jonka ansiosta tämä ei enää ole välttämätöntä.

EDITORin avulla kursoria voidaan siirrellä eteen- ja taaksepäin rivillä, muuttaa, poistaa tai lisätä uusia merkkejä.

Toimitusohjelmaa kutsutaan näppäilemällä

EDIT rivinumero (paina ENTER-näppäintä)

jolloin rivinumero on se ohjelman rivi, jota halutaan korjata.

Rivi tulee kokonaisuudessaan näkyviin ruudulle. Sen alapuolelle tulee näkyviin rivinumero, jonka vieressä kursori välkky.

KURSORIN SIIRTYMINEN RIVILLÄ

Kursori on nyt rivin alussa: Välilyöntinäppäimellä kursori siirretään rivillä eteenpäin. Kursori siirtyy ja kirjaimet, jotka se on ohittanut tulevat näkyviin. Askelpalautusnäppäimellä (+) päästään rivillä taaksepäin. Voit nopeuttaa kursorin siirtymistä näppäilemällä numeron ennen kuin painat haluamaasi näppäintä (esim. 5 VÄLILYÖNTI siirtää kursoria 5 merkkiä eteenpäin ja 3+ siirtää sitä 3 merkkiä taaksepäin). Näitä kahta näppäintä käyttäen voit siirrellä kursoria minkä merkin kohdalle tahansa rivillä. Merkki ei näy, koska kursori välkkyi sen päällä. Kahdella muulla käskyllä pääset suoraan johonkin määrättyyn kohtaan rivillä. Halutessasi lisätä jotakin rivin loppuun, näppäile X. Kursori siirtyy tällöin suoraan rivin loppuun ja voit lisätä haluamasi merkit. Hakutoimintoa käyttäen voit siirtyä suoraan haluamasi merkin päälle. Näppäile S ja heti sen perään merkki, jonka kohdalle haluat siirtyä. SA siirtää kursorin ensimmäisen A-kirjaimen kohdalle rivillä. Jos rivillä on useampia kuin yksi A, ja haluat siirtyä kolmannen A:n kohdalle, voit näppäillä 3 SA, joka tarkoittaa, että riviltä on etsittävä kolmas A-kirjain.

RIVIN MUUTTAMINEN

Kun kursori on siirretty haluttuun kohtaan välilyöntinäppäintä, askelpalautinta tai S-näppäintä käyttäen, voit

1. Poistaa merkin näppäilemällä D. Tämä poistaa kursorin

alla olevan merkin. Halutessasi poistaa useampia kuin yhden kirjaimen, näppäille luku D:n eteen. (5D poistaa seuraavat viisi kirjainta kursorin sijaintikohdasta alkaen).

2. Muuttaa merkin näppäilemällä C, jonka jälkeen kirjoitetaan uusi merkki. (CF muuttaa kursorin alla olevan merkin kirjaimeksi F. Samoin kuin kohdassa 1., 3C muuttaa seuraavat kolme merkkiä kolmeksi merkiksi, jotka annetaan näppäimistöltä).
3. Lisätä merkkejä näppäilemällä I, jonka jälkeen näppäilään lisättävätmerkit. I-näppäintä painettaessa toimitusohjelma ottaa käyttöön lisäystilan. Tämä tarkoittaa sitä, että kaikki kirjoitetut merkit lisätään riville. Lisäystilasta päästään takaisin normaaliin toimitustilaan painamalla samanaikaisesti SHIFT ja ↑.
4. Listata rivin senhetkisen asun näppäilemällä L. Rivillä näkyvät tällöin ne muutokset, jotka tähän mennessä on tehty. Kun rivi on näytetty, kursori palaa rivin alkuun.
5. Toimitustilasta poistutaan painamalla ENTER-näppäintä. Kone sijoittaa tällöin muutetun rivin takaisin muistissa olevaan ohjelmaan, ja ruutuun tulee OK. Toimitusohjelmasta voidaan poistua missä vaiheessa tai tilassa tahansa painamalla ENTER-näppäintä.
6. Peruuttaa kaikki riville tehdyt muutokset näppäilemällä A. Alkuperäinen rivi tulee näkyviin ruudulle. Sitä voidaan nyt tarvittaessa muuttaa uudelleen.

Seuraavassa on esimerkki englanninkielisen lauseen korjaamisesta. Voimme samalla oppia hieman englanninkielen oikeinkirjoitusta. Aluksi korjaaminen voi tuntua hiukan hankalalta, mutta korjaustoiminnot voi oppia hämmästyttävän lyhyen harjoittelun jälkeen. Painettavat näppäimet on merkitty hakasuluilla.

[E] [D] [I] [T] [Ø] [ENTER]

1Ø PRINT "THEIR ARE MANY MISTOOK IN LINE

1Ø ■ ■ ilmoittaa kursorin sijainnin

[1] [Ø] [VÄLILYÖNTI] kursori siirtyy kymmenen merkkiä eteenpäin; voidaan myös näppäillä

[2] [S] [1] siirtää kursorin toisen I:n kohdalle

[C] [R] [C] [E] muuttaa I:n R:ksi ja R:n E:ksi

[S] [M] siirtää kursorin MANY-sanan alkuun

[2] [D] poistaa M:n ja A:n

[VÄLILYÖNTI] [C] [C] hyppää N:n yli ja muuttaa Y:n O:ksi

[S] [O] siirtää kursorin MISTOOK-sanan ensimmäisen O:n kohdalle

[D] [C] [A] poistaa ensimmäisen O:n ja muuttaa toisen O:n A:ksi

[VÄLILYÖNTI] [I] [E] [S] siirtyy K:n yli ja lisää ES

[SHIFT] [↑] poistuu lisäystilasta

[S] [I] siirtyy L-kirjaimen kohdalle sanassa LINE

[I] [T] [H] [I] [S] [VÄLILYÖNTI] lisää sanan THIS ja yhden välilyönnin ennen LINE-sanaa

[SHIFT] [↑] poistuu lisäystilasta

[X] siirtää kursorin rivin loppuun ja siirtyy lisäystilaan

[VÄLILYÖNTI] [N] [O] [W] ["] lisää sanan NOW ja lainausmerkit, jotka sulkevat merkkijonon

[VÄLILYÖNTI] [↑] poistuu lisäystilasta

[L] listaa rivin tämänhetkisen asun

```
10 PRINT "THERE ARE NO MISTAKES IN THIS LINE NOW"
```

```
10 ■
```

[ENTER] tallentaa rivin muistiin ja poistuu toimitusohjelmasta.

Käytettävissä on myös kaksi muuta toimituskäskyä, joita on käytettävä varoen.

1. Merkki K (kill) poistaa koko loppurivin cursorin sijaintikohdasta lähtien. K ja merkki poistaa kaiken cursorista alkaen siihen asti, kun annettu merkki ensimmäisen kerran esiintyy. (3KA poistaa rivin kolmanteen A-kirjaimen asti).
2. Merkki H (hack) poistaa rivin lopun cursorista lähtien ja siirtyy sitten lisäystilaan. Sitä on hyvä käyttää, kun rivin loppu halutaan kirjoittaa uudelleen.

Toimitusohjelma (editor) helpottaa ohjelman korjaamista ja muuttamista. Voit kirjoittaa uudelleen jonkin aikaisemmista esimerkkiohjelmista ja muuttaa sitä sitten toimitusohjelman avulla.

LISÄÄ SYSTEEMIKOMENTOJA

Esimerkiksi LIST ja RUN ovat systeemikomentoja. Ne eivät ole ohjelman osa, vaan niillä annetaan tietokoneelle suora käsky toimia heti. Seuraavassa on lisää vastaavanlaisia, ohjelmointia helpottavia komentoja.

DEL - poistaa ohjelmarivin

Tähän mennessä olemme oppineet poistamaan ohjelmarivin näppäilemällä rivinumeron ja painamalla sen jälkeen ENTER-näppäintä. Tämä on hyvä keino yhden tai kahden rivin poistamiseksi, mutta jos halutaan poistaa 30 tai 40 riviä, voidaan käyttää komentoa:

DEL rivinumero-rivinumero

joka poistaa kaikki rivit ensimmäisestä annetusta rivinumerosta toiseen annettuun numeroon asti tämä rivi mukaanluettuna.

DEL 100-250

poistaa tällä hetkellä koneen muistissa olevasta ohjelmasta kaikki rivit rivistä 100 riviin 250, joka on viimeinen poistettava rivi.

DEL-käskyä voidaan käyttää myös toisessa muodossa:

DEL 20 poistaa vain rivin 20

DEL 30- poistaa kaikki rivit riviltä 30 ohjelman loppuun

DEL -200 poistaa kaikki rivit ohjelman alusta riviin
200 ja myös tämän rivin

DEL - poistaa koko ohjelman

RENUM - ohjelmarivien uudelleennumerointi

RENUM-komennolla voidaan muuttaa kaikki ohjelman rivinumerot tai osa niistä. RENUM-käsky muuttaa myös GOTO, GOSUB, IF THEN, ON GOTO ja ON GOSUB-lauseisiin sisältyvät rivinumerot ja huo-

lehtii siitä, että ohjelma haarautuu edelleen oikeille riveille. Näitä lauseita käsitellään myöhemmin.

RENUM uusi_rivi, aloitusrivi, väli

Uusi_rivi on ensimmäisen uudelleen numeroitavan rivin uusi rivinnumero. Aloitusrivi on se rivi, jolta uudelleennumeroinnin halutaan alkavan, ja väli ilmaisee, millaisin välein uudelleennumeroitavat rivit halutaan numeroida. Kaikki tai mikä tahansa näistä määritteistä voidaan jättää pois. Jos uusi_rivi jätetään pois, uudeksi riviksi oletetaan 10. Jos aloitusrivi jätetään pois, koko ohjelma numeroidaan uudelleen, ja jos väli jätetään pois, rivit numeroidaan 10 numeron välein.

RENUM numeroi koko ohjelman 10, 20, 30, ...

RENUM 100,50,5 numeroi rivit uudelleen alkaen riviltä 50
 100, 105, 110, ... (kaikki rivinumerot ennen riviä 50 jäävät ennalleen)

RENUM 110,,2 numeroi koko ohjelman uudelleen 110, 112,
 114, ...

RENUM ,,5 numeroi koko ohjelman uudelleen 10, 15, 20, ...

Huomaa, että kaikkien pilkkujen on oltava paikoillaan silloinkin, kun määrite on jätetty pois. RENUM-käskyillä ei voida muuttaa rivien järjestystä.

CSAVE

CLOAD

SKIPF

Komento CSAVE tallentaa ohjelman kasettinauhalle. Ohjelman nimi saa olla enintään kahdeksan merkin pituinen.

CSAVE "OHJELMA"

Tallennettaessa tietoa kasetille käytetään ylimääräistä parametriä A, jolloin tieto tallentuu ASCII-muodossa. Se voidaan tällöin lukea myöhemmin komennolla INPUT*-1.

CSAVE "DATA",A

CLOAD-käskey lataa annetun ohjelmatiedoston kasetilta koneen muistiin.

CLOAD "OHJELMA"

SKIPF-komennolla hypätään annetun ohjelman yli seuraavaan ohjelmaan tai annetun ohjelman loppuun.

SKIPF "OHJELMA"

EDIT

EDIT-komentoa käytetään jonkin määrittyn rivin muuttamiseen.

EDIT rivinumero

EDIT-tilassa voidaan käyttää kaikkia seuraavia komentoja:

L	Listaa rivin tämänhetkisen asun
C <u>m</u> <u>e</u> <u>r</u> <u>k</u> <u>k</u> <u>i</u>	Muuttaa (change) annetun merkin
<u>n</u> C <u>m</u> <u>e</u> <u>r</u> <u>k</u> <u>k</u> <u>i</u>	Muuttaa seuraavat <u>n</u> merkkiä uusiksi merkeiksi
I	Lisää merkkejä
D	Poistaa (delete) merkin
<u>n</u> D	Poistaa seuraavat <u>n</u> merkkiä
H	Poistaa rivin lopun kursorista lähtien ja siirtyy lisäystilaan (hack)
X	Pidentää riviä. Siirtää kursorin rivin loppuun ja siirtyy lisäystilaan (extend)
S <u>m</u> <u>e</u> <u>r</u> <u>k</u> <u>k</u> <u>i</u>	Etsii kohdan, missä annettu merkki esiintyy ensimmäisen kerran ja siirtää kursorin sen päälle
<u>n</u> S <u>m</u> <u>e</u> <u>r</u> <u>k</u> <u>k</u> <u>i</u>	Etsii kohdan, missä annettu merkki esiintyy <u>n</u> . kerran ja siirtää kursorin sen päälle
K	Poistaa rivin lopun kursorista lähtien (kill)
<u>n</u> K <u>m</u> <u>e</u> <u>r</u> <u>k</u> <u>k</u> <u>i</u>	Poistaa rivin siihen asti, missä annettu merkki esiintyy <u>n</u> . kerran
<u>n</u> VÄLILYÖNTI	Siirtää kursoria eteenpäin <u>n</u> merkkiä. Ellei <u>n</u> -arvoa anneta, sen arvoksi oletetaan 1.

n ←	Siirtää kursoria <u>n</u> merkkiä taaksepäin. Ellei n-arvoa anneta, sen arvoksi oletetaan 1.
SHIFT ↑	Poistuu lisäystilasta ja palaa toimitustilaan
(ENTER)	Poistuu toimitusohjelmasta (editor), tallentaa rivin ja palauttaa ohjauksen näppäimistölle

DEL

DEL-komentoa käytetään annettujen ohjelmarivien poistamiseen tällä hetkellä koneen muistissa olevasta ohjelmasta.

DEL rivinumero 1 - rivinumero 2

poistaa kaikki ohjelmarivit rivinumerosta 1 rivinumeroon 2 nämä rivit mukaanluettuina. Molemmat rivinumerot voidaan jättää pois, ja komentoa voidaan käyttää seuraavissa muodoissa:

DEL - Poistaa koko ohjelman (sama kuin NEW).

DEL -100 Poistaa kaikki rivit alusta riville 100 ja myös rivin 100.

DEL 300- Poistaa rivit alkaen rivistä 300 ohjelman loppuun asti.

DEL 100-200 Poistaa kaikki rivit alkaen rivistä 100 riville 200 ja myös rivin 200.

RENUM

RENUM-komennolla voidaan muuttaa kaikki ohjelman rivinumerot tai osa niistä. RENUM-komento muuttaa myös haaraumiskäskyjen (GOTO jne.) kohderivien numerot, jotta ohjelma etenisi oikealla tavalla.

RENUM uusi_rivi, aloitusrivi, väli

numeroi uudelleen kaikki rivit alkaen aloitusriviltä uuden rivin ilmoittamalla rivinumerolla ja niin monen numeron välein kuin väli osoittaa. Mikä tahansa määrite voidaan jättää pois, ja komentoa voidaan käyttää seuraavasti:

RENUM numeroi uudelleen koko ohjelma. Rivit numeroidaan 10, 20, 30, ...

RENUM100 numeroi uudelleen koko ohjelma: Rivit numeroidaan 100, 110, 120, ...

RENUM100,50,5 alkaa uudelleennumeroinnin riviltä 50. Rivit numeroidaan 100, 105, 110, ...

RENUM,,20 numeroi uudelleen koko ohjelman. Rivit numeroidaan 10, 30, 50, ...

Huomaa, että jos jokin määrite jätetään pois mutta seuraava merkitään, pilkkujen on oltava paikoillaan. RENUM-käskyllä ei voida muuttaa rivien järjestystä.

TR - ohjelman kulun seuraaminen

Ohjelman kulkua on joskus hyvä seurata jos ohjelmassa ilmenee ongelmia. Tämä on mahdollista TR-toiminnon (trace) avulla. Seuranta otetaan käyttöön näppäilemällä TRON ennen kuin ohjelmaa aletaan suorittaa. Kone kirjoittaa nyt rivinumeron ruudulle sitä mukaa kuin ohjelma etenee. Näin voit nähdä, haarautuuko ohjelma oikeassa kohdassa. TR-kytketään pois päältä näppäilemällä TROFF. TRON ja TROFF voidaan lisätä myös ohjelmaan, jolloin niiden eteen merkitään rivinnumero. Näin voidaan seurata vain jotakin ohjelman osaa.

STOP - pysäytys ja aloitus

Ohjelma voidaan pysäyttää ohjelman suorituksen aikana kirjoittamalla ohjelman rivi, johon sisältyy STOP-komento.

185 STOP

Tämä rivi pysäyttää ohjelman riville 185. Ruudulle tulee teksti, joka ilmoittaa, että ohjelma on pysähtynyt ja rivin, jolle se on pysähtynyt. Nyt voit tarkistaa minkä tahansa muuttujan arvon näppäilemällä PRINT tai ? ja sen jälkeen muuttujan nimen. Ohjelma käynnistetään uudelleen näppäilemällä CONT (continue = jatka), ja suoritus jatkuu STOP-komentoa seuraavalta riviltä.

Osaat nyt numeroida ohjelman uudelleen, poistaa haluamasi rivit, muuttaa minkä tahansa rivin sisältöä ja tallentaa valmiin ohjelman kasettinauhalle. Koska pystymme nyt säilyttämään ohjelmamme, alamme laatia käyttökelpoisempia ohjelmia, joilla voidaan tehdä muutakin kuin väläytellä eri värejä ruudulla ja tuottaa ääniä.

TR

Ohjelman kulkua voidaan seurata käyttämällä TR-toimintoa (trace). Rivinumero tulee näkyviin ruudulle sitä mukaa kuin ohjelma etenee riviltä toiselle. Seuranta on kytkettävä päälle ennen kuin ohjelmaa aletaan suorittaa.

TRON kytkee seurannan päälle

TROFF kytkee seurannan pois päältä

Nämä ovat molemmat suoria komentoja, eivätkä vaadi rivinumeroa.

Jos halutaan seurata jonkin ohjelman osan suoritusta, TRON ja TROFF voidaan lisätä ohjelmaan.

END

STOP

CONT

Ohjelman suoritus päättyy END-lauseeseen. STOP keskeyttää ohjelman suorituksen riville, jolla STOP-komento on annettu. Ruudulle ilmestyy BREAK AT n, joka ilmoittaa, että ohjelma on keskeytynyt rivillä n. Ohjelman suoritusta jatketaan näppäilemällä CONT. Tähän komentoon ei liitetä rivinumeroa. Ohjelma jatkuu STOP-komentoa seuraavalta riviltä. END-lauseen jälkeen ohjelma ei jatku, vaan se on ajettava kokonaan uudelleen.

5. OHJELMAN HAARAUTUMINEN

Kappaleen 3 lopussa opettelimme laatimaan ohjelman jakamalla sen erillisiin osa-alueisiin. Seuraavassa tarkastelemme, miten nämä osat sidotaan toisiinsa ohjailemalla ohjelman kulkua. Tätä kutsutaan haarautumiseksi. Olemme jo tutustuneet haarautumis- eli hyppykäskyyn GOTO. Tämä on ehdoton hyppykäsky. Kun ohjelma kohtaa GOTO-käskyn, se haarautuu heti käskyn määrittämälle riville ja jatkaa ohjelman suoritusta siitä. Tässä tapauksessa sillä ei ole valinnanvaraa. GOTO tarkoittaa, että annetulle riville on siirryttävä heti. Toistaiseksi olemme käyttäneet GOTO-käskyä etupäässä, palataksemme ohjelman alkuun. Vaikka usein onkin tarpeen toistaa jotakin ohjelmakohtaa yhä uudelleen, sen halutaan tuskin jatkuvan ikuisesti. Ei ole myöskään hyvän tavan mukaista käyttää BREAK-näppäintä ohjelman pysäyttämiseen. Onneksi Basic-kielessä on useita sellaisia lauseita, joiden avulla ohjelman kulkua voidaan ohjailla.

HAARAUTUMISKÄSKYT

Ensimmäinen näistä on GOTO-lauseen pitempi muoto. ON...GOTO-lauseetta käytetään seuraavasti:

ON numerolauseke GOTO rivinumeroluettelo

Numerolausekkeen arvo lasketaan ja katkaistaan tarvittaessa kokonaisluvuksi (desimaalipilkua seuraavat numerot jätetään pois). Ohjelman suoritus jatkuu tämän jälkeen joltakin luet-

telossa mainitulta riviltä. Jos lausekkeen arvoksi on saatu 1, siirrytään ensimmäiseen rivinumeroon, jos se on 2, toiseen rivinumeroon jne.. Jos lausekkeen arvo on alle 1 tai suurempi kuin luettelossa annettujen rivien lukumäärä, lausetta ei huomioida lainkaan ja suoritus jatkuu seuraavalta riviltä. Jos lausekkeen arvoksi saadaan negatiivinen luku, ohjelma keskeytyy ja antaa virheilmoituksen. Yleensä on hyvä tarkistaa, että arvo on aiotulla alueella ennen kuin ohjelma on edennyt ON...GOTO-lauseeseen. Seuraavassa on joitakin esimerkkejä tämän lauseen käytöstä:

```
140 ON P GOTO 200,300,400
210 ON X-4 GOTO 20,40,700,10,690
185 ON B*C/D-E GOTO 115,285,900,40
```

ON...GOTO-lause on käyttökelpoinen tapa valita useiden vaihtoehtojen välillä, ja se voidaan luokitella ehdolliseksi hypy-käskyksi. Käytämme sitä tällä tavoin jäljempänä tässä kappaleessa.

EHDOLLINEN HAARAUTUMINEN

Huomattavasti käyttökelpoisempi ehdollinen hypykäsky on IF...THEN-lause. Se on ehkä käyttökelpoisin yksittäinen Basic-kielen lause. Se voi olla hyvin yksinkertainen mutta myös erittäin laaja ja monimuotoinen. Sen yksinkertaisin muoto on:

```
IF ehto THEN rivinumero
```

Tämä lause merkitsee: Jos ehto pätee, siirry annetulle riville, jatka muussa tapauksessa suoraan seuraavalle riville.

```
120 IF D>9 THEN 250
```

```
180 IF A$="KYLLÄ" THEN 600
```

Edelläolevassa esimerkissä ohjelma haarautuu riviltä 120 riville 250 jos, ja vain jos, D:n arvo on suurempi kuin 9. Jos D on pienempi tai yhtä suuri kuin 9, ohjelma jatkaa seuraavalle riville. Rivillä 180 haarautuminen riville 600 tapahtuu vain, jos merkkijonomuuttujan A\$ arvoon sisältyvät merkit KYLLÄ. Jonon on tällöin oltava täsmälleen samanlainen. Jos se on esimerkiksi K, JOO, YEA, JEES, JEP, OK (tai vaikka "kyllä") ohjelma hylkää haarautumiskäskyn. Täydellisessä muodossaan IF...THEN-lause on seuraava:

```
IF ehto THEN toiminto_1 ELSE toiminto_2
```

joka tarkoittaa: Jos ehto pätee, suoritetaan toiminto_1, muussa tapauksessa suoritetaan toiminto_2.

```
210 IF P=3 THEN PRINT "TOSI" ELSE PRINT "EPÄTOSI"
```

Tässä esimerkissä TOSI tulostetaan vain, jos P on 3. Jos P:llä on jokin muu arvo, tulostetaan EPÄTOSI. Molemmissa tapauksissa ohjelman suoritus jatkuu seuraavalta riviltä. Sekä toiminto_1 että toiminto_2 voivat sisältää useampiakin lauseita.

```
210 IF P=3 THEN PRINT "TOSI":R=R+1:GOTO 560
```

```
ELSE PRINT "EPÄTOSI":L=L+1
```

Jos P on 3, kone kirjoittaa TOSI, lisää muuttujan R arvoa

yhdellä ja jatkaa ohjelman suoritusta riviltä 560. Jos arvo on jokin muu, se kirjoittaa EPÄTOSI, lisää muuttujan L arvoa yhdellä ja jatkaa seuraavalta riviltä.

Toiminto_1 ja toiminto_2 voivat olla mitä tahansa käyviä Basic-kielen lauseita, ja niihin voi sisältyä myös muita IF...THEN-lauseita. Ainoa rajoitus on, että IF...THEN-lauseen pitää mahtua täydellisenä yhdelle käskyriville. (Yksi käsky rivi saa sisältää enintään 256 merkkiä rivinumero mukaan luettuna).

Toistaiseksi olemme tarkastelleet vain, mitä tapahtuu, kun ehtolausekkeen arvo on laskettu. Varsinainen valinta tapahtuu siinä vaiheessa, kun ehtoa testataan. Ehto voi olla TOSI tai EPÄTOSI; muita vaihtoehtoja ei ole. (Tietokone antaa vaihtoehdolle "tosi" arvon 1 ja vaihtoehdolle "epätosi" arvon 0). Yksinkertaisen ehdon muoto on:

Lausekkeen_1_suhde_lausekkeeseen_2

lauseke_1 ja lauseke_2 ovat tavallisia Basic-lausekkeita, siinä muodossa kuin ne esiintyvät sijoituslauseissa. Molempien lausekkeiden on oltava samaa tyyppiä (molemmat numeerisia tai molemmat merkkijonoja). Suhde voi olla mikä tahansa seuraavista:

<u>Merkitys</u>	<u>Symboli</u>	<u>Esimerkki</u>
Sama kuin	=	60 IF X=Y+2 THEN 100
Pienempi kuin	<	110 IF A*B+2<C/2 THEN 20
Suurempi kuin	>	185 IF A\$>B\$ THEN PRINT A\$
Pienempi tai yhtä suuri kuin	<=	220 IF ?*W9<=B/Z9 THEN A=A-1
Suurempi tai yhtä suuri kuin	>=	415 IF B7>=0 THEN X=0
Erisuuri kuin	<>	80 IF A\$<>"KYLLÄ" THEN 999

Huomaa, että suhde voidaan liittää sekä merkkijonoihin, että numeerisiin lausekkeisiin. Merkkijonoja verrattaessa jokainen merkki tarkistetaan vuorotellen yksi kerrallaan. IF... THEN-lauseetta voidaan siis käyttää aakkostamiseen, esim.: Ehto "A"<"B" on tosi, koska kirjain A on aakkosissa ennen B:tä. Ehto "AAA"<="AA" on epätosi, koska esimerkiksi sanakirjassa AAA olisi AA:n jälkeen (tyhjä on ennen A:ta).

Ehtoja voidaan laajentaa yhdistämällä kaksi tai useampia ehtoja käyttäen operaattoreita AND ja OR.

```
270 IF A=4 AND B=7 THEN 500
```

AND-operaattori merkitsee, että molempien ehtojen on oltava tosia, jotta koko ehtolauseke olisi tosi. Jos siis tässä esimerkissä A on 4, mutta B:n arvo on jokin muu kuin 7, koko lauseke on epätosi ja ohjelma jatkaa seuraavalle riville.

OR-operaattori merkitsee, että jos jokin tai jotkut mainituista ehdoista pätee, koko lauseke on tosi.

```
320 IF D<3 OR F+G>25 THEN 100
```

Ohjelma haarautuu riville 100, jos D on pienempi kuin 3 riippumatta lausekkeen F+G arvosta. Se haarautuu vastaavasti myös, jos F+G on suurempi kuin 25, riippumatta D:n arvosta.

Seuraava ohjelma on yksinkertaistettu opetusohjelma. Ohjelmaan lisätyt kommentit kertovat, mitä kussakin osassa tapahtuu. Kiinnitä huomiota ON...GOTO-käskyn käyttöön option eli vaihtoehdon valitsemiseksi ja IF...THEN-lauseen käyttöön laskutoimitusten tarkistamiseksi. Rivillä 800 on uusi komento INKEY\$. Tällä komennolla tarkistetaan, onko näp-

pöintä painettu. Mahdollinen näppäily merkki tallennetaan muuttujan A\$ arvoksi (ks. rivi 800). Tätä komentoa käytettäessä ei tarvitse painaa ENTER-näppäintä syötetyn tiedon jälkeen. Ks. erillistä lokeroa, jossa kuvataan INKEY\$-komentoa.

Ohjelma näyttää paljon pitemmältä kuin se itse asiassa on, sillä kommenttirivejä on yli puolet. Myöhemmin emme enää käytä yhtä runsaasti kommentteja tilan säästämiseksi. Väilyönti on merkitty kärjellään seisovalla kolmiolla (V).

```
10 REM ARITMEETTINEN HARJOITUSOHJELMA
20 REM
30 REM LASKURIEN NOLLAUS JA VAIKEUSASTEEN MÄÄRITTELY
40 REM
50 R=0:W=0:CLS
60 T$="ARITMEETTINEN HARJOITUS"
70 PRINT @2,T$
80 PRINT @64, "VVVALITSE VAIKEUSASTE"
90 PRINT @96,"VVKOKONAISLUKU 1-10":INPUT L:L1=10*L-1
100 REM
110 REM LASKUTAPAVAIHTOEHDOT
120 REM
130 CLS: PRINT @6,T$
140 PRINT @71,"1.VYHTEENLASKU"
150 PRINT @103,"2.VVÄHENNYSLASKU"
160 PRINT @135,"3.VKERTOLASKU"
170 PRINT @167,"4.VJAKOLASKU"
180 REM
190 REM KIRJOITUSKOH DAN MÄÄRITTELY
```

```

200 REM
210 P=224:Q=352
220 REM
230 REM LASKUTAVAN VALINTA
240 REM
250 PRINT@P,"VVMITÄ NÄISTÄ HALUAT KOKEILLA";:INPUT A
260 REM
270 REM KAHDEN LUVUN VALINTA TEHTÄVÄÄ VARTEN
280 REM
290 N1=RND(L1):N2=RND(L1)
300 REM
310 REM HAARAUTUMINEN LASKUTAPAAN
320 REM
330 ON A GOTO 390,460,530,600
340 REM
350 REM TUNTEMATON LASKUTAPA - UUSI YRITYS
360 REM
370 CLS:SOUND 160,3:GOTO 130
380 REM
390 REM YHTEENLASKUOSA
400 REM
410 PRINT@P,"VVVVVVVVVVVVYHTEENLASKU"
420 PRINT@Q,"VVPALJONKO ON";N1;"PLUS";N2;:INPUT N4
430 N3=N1+N2
440 IF N4=N3 THEN 690 ELSE 730
450 REM
460 REM VÄHENNYSLASKUOSA

```

```

470 REM
480 PRINT@P,"VVVVVVVVVÄHENNYSLASKU"
490 PRINT@Q,"PALJONKO ON";N1;"MIINUS";N2;:INPUT N4
500 N3=N1-N2
510 IF N4=N3 THEN 690 ELSE 730
520 REM
530 REM KERTOLASKUOSA
540 REM
550 PRINT@P,"VVVVVKERTOLASKU"
560 PRINT@Q,"PALJONKO ON";N1;"KERTAA";N2;:INPUT N4
570 N3=N1*N2
580IF N4=N3 THEN 690 ELSE 730
590 REM
600 REM JAKOLASKUOSA
610 REM
620 PRINT@P,"VVVVVVVVVJAKOLASKU"
630 PRINT@Q,"PALJONKO ON";N1;"JAETTUNA";N2;:INPUT N4
640 N3=N1/N2
650 IF N3=N4 THEN 690 ELSE 730
660 REM
670 REM OIKEA VASTAUS
680 REM
690 R=R+1:PRINT@P."VVVVVVVVVOIKEIN":GOTO 770
700 REM
710 REM VÄÄRÄ VASTAUS
720 REM

```

```
730 G=G+1:PRINT@P,"VVVVVVVVVVVVVÄÄRIN":  
    PRINT@P+32,"VVVVVVVASTAUS ONV";N3:GOTO 770  
740 REM  
750 REM TOISTON TARKISTUS JA RIVIEN TYHJENNYS  
760 REM  
770 FOR D=1 TO 1600:NEXT D  
780 PRINT@P,"V":PRINT@P+32,"V":PRINT@Q,"V"  
790 PRINT@P,"UUSI TEHTÄVÄ?(K/E)"  
800 A$=INKEY$:IF A$=""THEN 800  
810 IF A$="K" THEN 130  
820 REM  
830 REM TULOKSEN NÄYTTÖ JA LOPETUS  
840 REM  
850 CLS:PRINT@128,"SAIT";R;"OIKEIN JA";G;"VÄÄRIN"  
860 END
```

ON...GOTO

ON GOTO-lause antaa mahdollisuuden haarauttaa ohjelma vaihtoehdoisesti eri riveille.

ON lauseke GOTO rivinumeroluettelo

Lausekkeen arvo lasketaan ja katkaistaan desimaalipilkusta, ellei saatu luku ole kokonaisluku. Ohjelma haarautuu sitten siihen luettelossa mainittuun rivinumeroon, jonka sijainti vastaa lausekkeesta saatua tulosta. Jos lausekkeen arvo on 4, ON GOTO-käskey valitsee rivinumeroluettelosta neljännen rivinumeron ja haarautuu tälle riville. Jos lausekkeen arvoksi saadaan negatiivinen luku, kone antaa virheilmoituksen. Jos lausekkeen arvo on nolla tai suurempi kuin luettelossa olevien rivinumeroiden lukumäärä, lause jätetään huomioimatta ja ohjelma jatkaa seuraavalle riville.

```
10 CLS:PRINT "TOISEN ASTEEN YHTÄLÖN RATKAISEMINEN"
```

```
20 INPUT "A,B,C";A,B,C:IF A=0 THEN 20
```

```
30 R=-B/(2*A):D=B*B-4*A*C:S=SGN(D)
```

```
40 P=SQR(D*S)/(2*A)
```

```
50 ON S+2 GOTO 80,60,70
```

```
60 PRINT "KAKSOISJUURI":PRINT R,R:END
```

```
70 PRINT "REAALIJUURET":PRINT R+P,R-P:END
```

```
80 PRINT "KOMPLEKSIJUURET":PRINT R,R:PRINT P,-P:END
```

IF...THEN...ELSE

IF-lauseen täydellinen muoto on:

IF ehto THEN toiminto_1 ELSE toiminto_2

Lause tutkii ehdon, joka on joko tosi tai epätosi. Jos se on tosi, suoritetaan toiminto_1, jos se on epätosi, toiminto_2.

Ehtoon sisältyy lausekkeen suhde lausekkeeseen. Lausekkeet voivat olla mitä tahansa Basic-lausekkeita, jotka ovat samaa tyyppiä (molemmat numeerisia tai molemmat merkkijonoja). Suhde on jokin seuraavista 'operaattoreista'

= Sama kuin	<> Erisuuri kuin
> Suurempi kuin	< Pienempi kuin
>=Suurempi tai sama kuin	<= Pienempi tai sama kuin

Jos ehtoja on useita ne yhdistetään toisiinsa loogisilla operaattoreilla AND, OR, NOT:

Ehto AND ehto on tosi vain jos molemmat ehdot ovat tosia

Ehto OR ehto on tosi, jos yksi ehdoista on tosi

NOT ehto on tosi, jos ehto on epätosi

Toiminto_1 ja toiminto_2 voi olla ehdollinen Basic-lause tai myös toinen IF-lause. Niihin voi sisältyä myös useita lauseita, jotka on erotettu toisistaan kaksoispisteella(:).

Käskyn ELSE-osa voidaan jättää pois. Tällöin ohjelma jatkaa seuraavalta riviltä, jos ehto on epätosi.

```

10 CLS:PRINT@ 11,"ARVAUSLEIKKI":N=RND(100):T=0
20 INPUT "ARVAA LUKU";G:T=T+1
30 IF G=N THEN PRINT "ARVASIT OIKEIN";T;"YRITYKSELLÄ":END
40 IF G>N THEN PRINT "VÄÄRIN, LIIAN SUURI" ELSE PRINT
    "VÄÄRIN, LIIAN PIENI"
50 GOTO 20

```

INKEY\$

INKEY\$ on funktio. Se tutkii, onko jotakin näppäintä painettu, ja lukee näppäilyllän merkin näppäimistöltä.

Sitä voidaan käyttää haluttaessa sijoittaa yksi merkki merkkijonomuuttujaan eikä merkin jälkeen tarvitse painaa ENTER-näppäintä.

```

10 CLS 0:PRINT@ 5,"PAINA JOTAKIN NÄPPÄINTÄ JA MINÄ";
20 PRINT@ 38,"KERRON SINULLE, MIKÄ SE OLI:V";
30 B$="ET PAINA MITÄÄN NÄPPÄINTÄVV"
40 C$="PAINAMASI NÄPPÄIN OLI:V"
50 A$=INKEY$
60 IF A$="" THEN PRINT@ 193,B$;ELSE PRINT@ 193,C$;A$;
70 FOR D=1 TO 600:NEXT D:GOTO 50

```


TOISTO

Ohjelmassa halutaan usein toistaa samoja rivejä useita kertoja. Tämä voidaan tehdä seuraavanlaisen haarautumiskäskyn avulla. Näppäile tämä pieni ohjelma ja suorita se.

```
10 CLS
20 FOR I=1 TO 50
30 PRINT@198,"LASKURI I="
40 PRINT@214,I
50 NEXT I
60 PRINT@369,"SILMUKKA PÄÄTTYI"
```

Ohjelma on liian nopea, jotta voitaisiin nähdä, mitä tapahtuu. Lisää siis rivi:

```
45 FOR J=1 TO 100:NEXT J
```

Kuten näet, ohjelma laskee yhdestä 50:en. Toistuvat rivit ovat 30, 40, ja 45. Tällaista toistosarjaa kutsutaan sil-mukaksi. Ohjelma hyppää tässä tapauksessa riviltä 50 takaisin riville 20. Silmukkaa ohjaavat lauseet ovat rivillä 20 (silmukan alkupää) ja rivillä 50 (silmukan loppupää). Rivi 20 tulkitaan:'kaikilla I:n arvoilla 1-50 yhden numeron välein suoritetaan kaikki seuraavat lauseet NEXT-lauseeseen asti'. Muuttuja I toimii laskurina ja sen arvot ovat 1, 2, 3,...50. Lisäsimme rivin 45, koska ohjelma laskee liian nopeasti. Tämä silmukka ei suorita mitään lausetta, joten laskuri (tässä tapauksessa muuttuja J) vain laskee yhdestä sataan. Tauko (laskeminen) kestää niin kauan, että se

hidastaa toista silmukkaa siinä määrin, että voimme lukea numerot ruudulta. Jos nyt muutamme rivin 20:

```
20 FOR I=1 TO 50 STEP 2
```

laskuri laskee nyt yhdestä alkaen kahden numeron välein (1, 3, 5, 7,...49). STEP määrittelee laskurin askelvälin. Jos STEP jätetään pois, kone olettaa askelväliksi 1. Muuta esimerkkiä kirjoittamalla seuraavat rivit:

```
15 INPUT "ALKU, LOPPU, ASKELVÄLI";A,B,C
```

```
20 FOR I=A TO B STEP C
```

```
70 PRINT@428,"JA I=";I
```

Aja ohjelma antamalla eri arvoja alulle, lopulle ja askelvälille. Kokeile esimerkiksi seuraavia arvoja

ALKU	LOPPU	ASKELVÄLI
50	1	1
-50	50	5
1	10	0.5
2.6	3.9	0.1
1	-2	1

Huomaat, että voit laskea eteenpäin tai taaksepäin millaisin välein tahansa. Ainoa sääntö on, että jos askelväli on positiivinen, alkuarvon on oltava pienempi kuin loppuarvon tai jos askelväli on negatiivinen alkuarvon on oltava suurempi kuin loppuarvon. Edellä olevista esimerkeistä viimeinen (1, -2, 1) osoittaa, että silloinkin, kun tätä sääntöä rikotaan (ei ole mahdollista laskea luvusta 1 lukuun -2 +1:n välein), silmukka suoritetaan kerran.

Silmukat voivat olla myös sisäkkäin (rivillä 45 oleva silmukka on riviltä 20 alkavan silmukan sisällä). Sisäkkäisten silmukoiden on sulkeuduttava oikeassa järjestyksessä.

```
20 FOR I=1 TO 10
30 FOR J=-2 TO 4STEP 0.6
40 FOR K=1 TO 0 STEP -0.1
.
.
.
100 NEXT K
110 NEXT J
120 NEXT I
```

Jokaisella FOR-lauseella on oltava vastaava NEXT-lause, jolloin NEXT-lauseeseen sisältyvä muuttuja osoittaa, mihin FOR-lauseeseen se kuuluu. Silmukat on suljettava päinvastaisessa järjestyksessä kuin ne avatiin. Jos kolme viimeistä riviä olisivat

```
100 NEXT J
110 NEXT K
120 NEXT I
```

Ohjelma pysähtyisi ja antaisi virheilmoituksen, koska silmukat J ja K menevät ristiin. Jos kaikki silmukat päättyvät samaan kohtaan, NEXT-lauseet voidaan yhdistää seuraavasti

```
100 NEXT K,J,I
```

mutta muuttujien järjestyksen on tällöinkin oltava oikea kuten edellä.

Silmukan muodostamiseen käytettyjä muuttujia (tässä esimerkiksi I, A, B ja C) voidaan käyttää silmukan sisällä. Alun, lopun tai askelvälin arvoa ei kuitenkaan saa muuttaa. Las-kuria voidaan muuttaa sijoittamalla se sijoituslauseen vasemmalle puolelle, mutta tämä ei ole suositeltavaa.

Muita haarautumiskäskyjä, kuten GOTO tai IF...THEN, voidaan käyttää silmukan sisällä silmukasta poistumiseen ennen sen päättymistä. Silmukkaan ei kuitenkaan voida mennä sisään keskeltä. Silmukka on aina aloitettava FOR-lauseella.

Seuraavassa ohjelmassa on esimerkki sisäkkäisten silmukoiden käytöstä. Ohjelma jäljittelee digitaalikelloa. Saadaksesi sen todella tarkaksi, sinun on ehkä säädettävä hidastussilmukkaa rivillä 220. Huomaa INKEY\$-komennon käyttö kellon käynnistämiseksi ja pysäyttämiseksi riveillä 90, 150 ja 170.

```
10 CLS 0:PRINT@10,"DIGITAALIKELLO";
20 'KIRJOITUSKOHDAT
30 P=261:Q=196
40 'OHJETEKSTI
50 PRINT@386,"PYSÄYTÄ JA KÄYNNISTÄ VÄLILYÖNTINÄPPÄIMELLÄ";
60 PRINT@421,"JA KÄYNNISTÄ UUELLEEN 'R'-NÄPPÄIMELLÄ";
70 'TEKSTIN KIRJOITUS JA KÄYNNISTYKSEN ODOTUS
80 PRINT@Q,"TUNTIA";:PRINT@Q+10,"MIN";:PRINT@Q+22,"SEK";
90 A$=INKEY$:IF A$<>"V" THEN 90
100 'ALOITTA A SILMUKAN TUNNIT, MIN JA SEK
110 FOR H=0 TO 23:FOR M=0 TO 59:FOR S=0 TO 59
```

```

120 SOUND 220,1
130 'SEKUNNIN KYMMENESOSASILMUKKA
140 FOR T=0 TO 9
150 A$=INKEY$:IF A$<>"V" THEN 200
160 'PAINETUN NÄPPÄIMEN TARKISTUS
170 A$=INKEY$:IF A$="R" THEN 110
180 IF A$<>"V" THEN 170
190 'AJAN NÄYTTÖ
200 PRINT@P,H;:PRINT@P+10,M;:PRINT@P+19,S;". ";T;
210 'KELLON SÄÄTÖSILMUKKA
220 FOR D=1 TO 13:NEXT D
230 NEXT T
240 'SEKUNNIN KYMMENESOSASILMUKAN LOPPU
250 NEXT S,M,H
260 'SEK, MIN JA TUNTISILMUKAN LOPPU
270 PRINT@448,"KELLO PYSÄHTYI"
280 END

```

SISÄKKÄISET OHJELMAT

Ohjelmalla on perusrakenne, joka koostuu yleensä useista pienistä osaohjelmista. Joitakin näistä osista saatetaan tarvita useita kertoja pääohjelman eri vaiheissa. Ohjelman rakennetta voidaan usein yksinkertaistaa käsittelemällä näitä tilanteita aliohjelmina (subroutines). Aliohjelma on osa suurempaa ohjelmakokonaisuutta, ikäänkuin ohjelma ohjelmassa. Aliohjelman tärkein piirre on, että sitä kutsuttaessa ohjel-

ma suorittaa aliohjelmaan kuuluvat ohjelmarivit ja palaa sitten kohtaan, jossa aliohjelmaa kutsuttiin. Aliohjelmaa kutsutaan seuraavalla lauseella:

```
GOSUB rivinumero
```

jolloin rivinumero on se ohjelmarivin numero, josta aliohjelma alkaa. Koska aliohjelman on palattava takaisinkin, se päättyy aina käskyyn

```
RETURN
```

GOSUB-lause toimii samalla tavoin kuin GOTO-lause. Erona on se, että GOTO-käskyllä ohjelma haarautuu jollekin riville ja jatkaa siitä. Se ei palaa takaisin muutoin kuin toisella GOTO-käskyllä. Näppäile seuraava esimerkki ja aja se:

```
10 CLS:PRINT "PÄÄOHJELMA"  
20 GOSUB 50  
30 PRINT "TAKAISIN PÄÄOHJELMASSA"  
40 END  
50 PRINT "1. ALIOHJELMA"  
60 GOSUB 90  
70 PRINT "TAKAISIN 1. ALIOHJELMASSA"  
80 RETURN  
90 PRINT "2. ALIOHJELMA"  
100 RETURN
```

Ohjelmaa suoritettaessa rivit ajetaan seuraavassa järjestyksessä:

```
10,20,50,60,90,100,70,80,30,40
```

Huomaa, että ensimmäinen aliohjelma (rivit 50-80) kutsuu toista aliohjelmaa (rivit 90,100). END-lause rivillä 40 merkitsee loppua. Se osoittaa, missä ohjelma varsinaisesti päättyy. Yritä jättää pois rivi 40 ja suorita ohjelma uudelleen. Tulokseksi pitäisi tällöin saada ?RG ERROR IN 80. Tämä johtuu siitä, että ohjelmaan sisältyi RETURN-lause, vaikka ei käynyt selvästi ilmi, että oli tultu aliohjelmaan. Ohjelma ikäänkuin putosi pääohjelmasta ensimmäiseen aliohjelmaan. Aliohjelmat on aina suojattava eli varmistettava, että niihin päästään vain GOSUB-käskyllä ja niistä poistutaan vain RETURN-käskyllä. Aliohjelman sisällä voidaan käyttää GOTO, IF...THEN ja muita vastaavia hyppykäskyjä, mutta ne eivät saa aiheuttaa haarautumista riville, joka on aliohjelman ulkopuolella.

ON...GOTO-lauseetta vastaavasti myös aliohjelmiin voidaan haarautua ehdollisesti, jolloin lauseen muoto on seuraava:

ON lauseke GOSUB rivinumeroluettelo

Useimmat kokeneet ohjelmoijat pitävät kirjastoa aliohjelmista, ja monet uudet ohjelmat voidaan laatia niiden pohjalta. Aliohjelmat on yleensä hyvä numeroida suurilla rivinumeroilla, 10000-, jotta ne voidaan sijoittaa suoraan ohjelmaan numeroimatta niitä uudelleen.

Seuraavilla sivuilla on useita esimerkkejä aliohjelmista, joten emme tässä anna erikseen esimerkkejä.

FOR...NEXT

FOR...NEXT koostuu kahdesta lauseesta.

```
FOR numeromuuttuja = lauseke TO lauseke STEP lauseke
```

ja

```
NEXT numeromuuttuja
```

FOR...NEXT-lauseet toimivat yhdessä ja määräävät, miten monta kertaa jokin ohjelman osa suoritetaan. Tätä kutsutaan silmukaksi.

Lausekkeiden arvo lasketaan ja silmukka laskee ensimmäisen lausekkeen arvosta toisen lausekkeen arvoon niin monen numeron välein kuin kolmas lauseke, askelväli (step), ilmoittaa. Laskurin kunkinhetkinen arvo on talletettuna numeromuuttujassa. Silmukka suoritetaan ainakin kerran, vaikkeivät ääriarvot ja askelväli olisikaan mahdollisia. Jos lauseesta jätetään STEP-osa pois, kone olettaa arvoksi +1.

Silmukat voivat olla myös sisäkkäin, mutta niiden on tällöin oltava oikeassa järjestyksessä.

FOR...NEXT-rakenteesta voidaan hypätä pois GOTO, IF...THEN tai muilla samankaltaisilla lauseilla, mutta silmukkaan ei voida mennä sisään keskeltä.

```
10 CLS: CLEAR: DIM L(1000)
```

```
20 INPUT "ANNA LUKU"; N: IF N > 1000 OR N < 2 THEN 20
```

```
30 CLS: PRINT "ALKULUVUT VÄLILLÄ 2 JA"; N
```



```
40 FOR I=2 TO N:IF L(I)<0 THEN 80
50 PRINT I;
60 IF I>SQR (N) THEN 80
70 FOR J=I TO N STEP I:L(J)=-1:NEXT J
80 NEXT I
```

GOSUB

RETURN

ON...GOSUB

GOSUB-käskyllä ohjelman suoritus jatkuu aliohjelman alusta. RETURN-käskyllä palataan takaisin GOSUB-lauseetta seuraavalle riville.

GOSUB-käskyä seuraa rivinumero, joka on aliohjelman ensimmäinen rivi.

GOSUB 1600

Aliohjelmassa pitää olla ainakin yksi RETURN-käsky.

ON...GOSUB-lauseella voidaan haarautua mahdollisesti aliohjelmiaan, ja se toimii vastaavalla tavalla kuin ON...GOTO-lause.

ON lauseke GOSUB rivinumeroluettelo

Jos lausekkeen arvo on negatiivinen, ohjelma pysähtyy ja antaa virheilmoituksen. Jos lauseke on nolla tai suurempi

kun luettelon rivinumeroiden lukumäärä, lause jätetään huomioimatta ja ohjelma jatkaa seuraavalle riville.

```
10 CLS:INPUT "ANNA KAKSI LUKUA";A,B
20 INPUT "ANNA NYT JOKIN LUKU VÄLILTÄ 1-4";C
30 ON C GOSUB 100,200,300,400
40 IF C>=1 AND C<=4 THEN END
50 PRINT C;"EI OLE 1 JA 4 VÄLILLÄ":GOTO 20
100 PRINT "YHTEENLASKU";A;"PLUS";B;"ON";A+B
110 RETURN
200 PRINT "VÄHENNYSLASKU";A;"MIINUS";B;"ON";A-B
210 RETURN
300 PRINT "KERTOLASKU";A;"KERTAA";B;"ON";A*B
310 RETURN
400 PRINT "JAKOLASKU";A;"JAETTUNA";B;"ON";A/B
410 RETURN
```

6. UUSIA ULOTTUVUUKSIA

Kappaleessa 2 esiteltiin muuttujatyypit. Todettiin, että muuttujia on kahta "kokoa": yksinkertaisia ja taulukkomuuttujia. Toistaiseksi olemme käsitelleet vain yksinkertaisia muuttujia.

Jos haluaisit käyttää tietokonettasi esimerkiksi kaikkien kirjojesi tai äänilevyjesi luetteloimiseen, sinulla ei pian enää olisi muuttujia, joilla voisit tallentaa ne. Tällainen ohjelma olisi myös erittäin vaikea kirjoittaa, kun jouduttaisiin pitämään järjestyksessä kaikki erilaiset nimet. Tässä tietokone tulee avuksi antamalla käyttöösi taulukko-muuttujat.

LUETTELOT JA TAULUKOT

Taulukkumuuttujat ovat erittäin käyttökelpoisia käsiteltäessä luetteloita. Esimerkiksi kirjaluettelo voitaisiin laatia seuraavasti:

1. Otsake 1
2. Otsake 2
3. Otsake 3

.

.

.

Voit nyt viitata johonkin kirjoistasi esimerkiksi pyytämällä numeron 8 listalta. Ohjelmassamme annetaan nimi tälle muuttujajonolle (otsikoille) ja johonkin yksittäiseen luet-

telon arvoon voidaan viitata antamalla sen indeksi (alkion järjestysnumero). Ensin siis on annettava luettelolle nimi.

Taulukkomuuttujien nimeämisessä noudatetaan samoja sääntö- ja kuin yksinkertaisissa muuttujissa. Tietokone erottaa nämä toisistaan, koska taulukkomuuttujan perässä on aina sulussa indeksinumero.

A(5) viittaa numeerisen taulukon 5. alkioon (luettelo A)

D7\$(28) viittaa merkkijonotaulukon D7\$ alkioon n:o 28.

Taulukkoa laadittaessa tietokoneelle on kerrottava taulukon nimi ja koko. Tässä käytetään DIM-lausetta:

DIM taulukkonimi (luku), taulukkonimi (luku, luku)

DIM tarkoittaa dimensiota, ulottuvuutta, sillä se ei ainoastaan nimeä taulukkoa, vaan määrää myös sen maksimikoon. Luku voi olla mikä tahansa positiivinen kokonaisluku tai yksinkertainen muuttuja, mikäli yksinkertaiselle muuttujalle on jo annettu arvo. Älä määrittele taulukoita suuremmiksi kuin on välttämätöntä, sillä laajat taulukot vievät paljon muistitilaa.

10 DIM A(22),NA\$(40)

Tällä rivillä konetta käsketään muodostaa taulukko, jonka nimi on A ja pituus 22 alkiota sekä merkkijonotaulukko, jonka nimi on NA\$ ja pituus 40 alkiota. (Alkioiden todelliset lukumäärät ovat kuitenkin 23 ja 41, sillä indeksinumero alkavat nollostasta). Kun jossakin lausekkeessa halutaan viitata taulukkoalkioon, indeksinumero on lisättävä sulussa nimen jälkeen.

25 A(4)=7.0

32 A(M)=BxC/A

Rivi 25 antaa taulukon A alkioille 4 arvon 7.0. Rivi 32 laskee lausekkeen arvon ja sijoittaa tuloksen taulukon A alkion M arvoksi. Huomaa, että rivillä 32 lausekkeen oikealla puolella oleva A on yksinkertainen muuttuja, jonka nimi on A, eikä sillä ole mitään tekemistä vasemmalla puolella olevan taulukon A kanssa. Taulukot voivat olla myös kaksikulotteisia, joten rivi:

10 DIM T(10,5),TB\$(12,4)

muodostaa numeerisen taulukon T, jossa on 10 riviä ja 5 saraketta. Merkkijonotaulukossa TB\$ on 12 riviä ja 4 saraketta. Esimerkiksi opettaja voi haluta tallentaa 25 oppilaan koetulokset 6 eri aineessa. Tässä tapauksessa voitaisiin muodostaa taulukko KOKEET(25,6). Kun nyt halutaan viitata johonkin taulukon alkioon, on käytettävä kahta indeksinumeroa.

25 PRINT KOKEET(10,3)

antaa 10. oppilaan koetulokset aineessa 3. Indeksinumeroiden on tietenkin tällöin viitattava alkioon, joka on olemassa. Lause KOKEET(30,7) aiheuttaa virheilmoituksen, koska näin suurta taulukkoa ei ole määritelty näin suureksi.

Seuraavassa esimerkissä käytetään taulukoita. Ohjelma on lyhyt, mutta vaatii hiukan päänvaivaa ennenkuin saadaan selville, miten se toimii. Sitä käytetään tavallisen

korttipakan sekoittamiseen. Jokaiselle kortille annetaan numero 1-52 taulukossa X. Alkiot poimitaan nyt satunnaisessa järjestyksessä taulukosta X ja sijoitetaan taulukoon Y. Kun ohjelma päättyy, taulukossa Y on numerot 1-52, mutta satunnaisessa järjestyksessä (sekaisin). Pakan sekoittamisessa ei voida käyttää satunnaislukugeneraattoria, joka antaisi suoraan satunnaisluvun, koska kukin kortti voidaan poimia vain kerran. RND-toiminto voi antaa saman numeron useita kertoja. Rivi 50 kirjoittaa sekoitetun taulukon. Huomaa, että se käyttää lauseketta taulukon indeksinumerona.

```
10 DIM X(52),Y(52):CLS
20 FOR I=1 TO 52:X(I)=I:NEXT I
30 FOR I=52 TO 1 STEP-1
40 J=RND(I):Y(I)=X(J):X(J)=X(I):NEXT I
50 FOR I=1 TO 13:FOR J=1 TO 4:PRINT Y(4*(I-1)+J);:NEXT
    J,I:END
```

Merkkijonotaulukon alkioita voidaan siirrellä samalla tavoin. Yleisimpiä käyttötapoja on luettelon aakkostaminen. Seuraavassa esimerkissä riviltä 200 alkava aliohjelma lajittelee sanaluettelon. Tietokoneella tehtävästä lajittelusta on kirjoitettu useita kirjoja. Tässä käytetty menetelmä on yksinkertainen. Ellet ennestään tunne tätä menetelmää, käy se läpi käsin käyttämällä yksinkertaista luetteloa D,B,A,E,C.

```

10 CLS:DIM W$(50)
20 INPUT "KUINKA MONTA SANAA";N
30 CLS:PRINT "ALKUPERÄINEN LUETTELO"
40 FOR I=1 TO N:PRINT I;".V V";
50 INPUT W$(I):NEXT I
60 GOSUB 200
70 PRINT@ 18,"LAJITELTU LUETTELO"
80 FOR I=1 TO N
90 PRINT@ 18+32*I,I;".V V";W$(I)
100 NEXT I:END
200 M=N
210 F=0:FOR I=1 TO M-1
220 IF W$(I)<=W$(I+1) THEN 240
230 T$=W$(I):W$(I)=W$(I+1):W$(I+1)=T$:F=1
240 NEXT I:IF F=1 THEN M=M-1:GOTO 210 ELSE RETURN

```

DIM

DIM-lausetta käytetään taulukon laajuuden määrittämisessä. Taulukot voivat olla yksi- tai kaksiulotteisia, ja ne voivat sisältää joko lukuja tai merkkijonoja (tekstiä). Taulukoiden nimet noudattavat samoja sääntöjä kuin yksinkertaisten muuttujien nimet.

DIM taulukkonimi (n), taulukkonimi (n, n)

Ellei taulukon enimmäiskoko ylitä kymmentä, DIM-lause ei ole välttämätön.

Kun taulukkoalkioon viitataan esim. lausekkeessa, nimen jälkeen on lisättävä suluissa indeksinumero.

A(14,K),N9(B),L\$(14,4)

MIKÄ ON FUNKTIO

Tutustuessamme RND-käskyyn, kutsuimme sitä funktioksi. Se ei ole ainoa funktio. Atk:ssa funktio on erityinen aliohjelma, joka saa tietyn arvon, kun sille annetaan sarja argumentteja. Basic-kielessä funktion muoto on seuraava:

Funktion nimi (argumentit)

ja sitä voidaan käyttää lausekkeissa samalla tavoin kuin muitakin aritmeettisiä operaattoreita (\uparrow , \ast , $/$, $+$, $-$). Funktiot sijoittuvat kuitenkin tärkeysjärjestyksessä ensimmäisiksi ennen kaikkia muita operaattoreita sulkumerkkejä lukuunottamatta.

Funktion argumentit ovat arvoja, jotka funktiolle annetaan. Funktio siis saa tuloksen arvokseen. Argumentit voivat olla vakioita, muuttujia tai lausekkeita.

RND (10),RND(X) tai RND(A*2+F)

ovat kaikki hyväksyttäviä argumentteja. Huomaa, että argumentti on aina sulkumerkkien sisällä.

Tietokoneessa on useita standardifunktioita, rakenteeseen sisältyviä funktioita, jotka kuuluvat Basic-kieleen. Rakenteeseen sisältyvät funktiot voidaan jakaa viiteen ryhmään. Esittelemme jokaisen ryhmän erikseen, luettelemme funktiot, selitämme ne lyhyesti ja annamme esimerkkirivin. Niitä käytetään tämän jälkeen ohjelmissa, joten jokaisesta ei tässä anneta esimerkkiohjelmaa.

1. ryhmän funktiot

Nämä ovat numeerisia funktioita, pääasiassa matemaattista käyttöä varten. Niiden argumentit ovat numeerisia ja ne saavat numeerisen arvon. Ensimmäisen ryhmän funktioita voidaan käyttää vain numerolausekkeissa. Ellet tunne ennestään trigonometrisia funktioita, tutustu liitteeseen D kirjan lopussa.

Funktion nimi	Toimitus	Esimerkki
ABS(X)	X:N itseisarvo	100 A=ABS(D*2-C)
ATN(X)	X:n arkustangentti <u>radiaaneina</u> . TAN(X):n käänteisfunktio	110 PRINT "KULMA-"; ATN(R3)
COS(X)	X:n kosini, jossa X on kulma <u>radiaaneina</u>	510 F7=COS(X+4)
EXP(X)	Korota e (luonnollisen logaritmi-järjestelmän kantaluku) potenssiin x (e^x). LOG(X):n käänteisfunktio	215 Q=EXP(-A*A)

Funktion nimi	Toimitus	Esimerkki
FIX(X)	Antaa X:n kokonaisosan (katkaisee luvun desimaalipilkusta)	172 N=FIX(Z x .05)
INT(X)	Jos X on positiivinen, katkaisee luvun kuten FIX. Jos X on nega- tiivinen, pyöristää alaspäin (esim. INT(-12.001) on -13)	280 P=INT(100*x)
JOYSTK(X)	Antaa vasemman tai oikean pelin- ohjausvivun vaaka- tai pystysuun- taisen aseman seuraavasti: X=0, vaaka-asema, vasen ohjausvipu X=1, pystyasema, vasen ohjausvipu X=2, vaaka-asema, oikea ohjausvipu X=3, pystyasema, oikea ohjausvipu	1040 A=JOYSTK(0): B=JOYSTK(1)
LOG(X)	X:n luonnollinen logaritmi. Argu- mentin arvon on oltava suurempi kuin nolla. EXP(X):n käänteisfunktio	617 L1:5.2*LOG(W4)
PEEK(X)	Antaa muistiosoite X:n sisällön.	55 P=PEEK(65280)
POINT(X,Y)	Tarkistaa, onko mosaiikkigrafiikan kenno kytketty päälle vai pois. X:n arvon on oltava alueella 0-63 (vaakatasossa), Y:n alueella 0-31 (pystysuunnassa). Antaa arvon -1 tekstitilassa tai 0-8, joka ilmoit- taa värikoodin, jos se on kytket- ty päälle.	50 IF POINTS(5,A)=C THEN 210

Funktion nimi	Toimitus	Esimerkki
POS(X)	Antaa tämänhetkisen tulostuspaikan. Ainoat argumentin X arvot ovat: Ø ruudulle ja 1 kirjoittimelle	168 IF POS(Ø)<3Ø THEN PRINT A\$
PPOINT(X,Y)	Tarkistaa hienografiikkakennon. Jos kenno on pois päältä, antaa ar- von Ø, muussa tapauksessa antaa kennon värikoodin (1-8). X:n arvo voi olla alueella Ø-255 ja Y:n Ø-191.	115 C=PPOINT(A1,A2)
RND(X)	Antaa satunnaisluvun (kokonaisluku) väliltä 1-X. Jos X=Ø, antaa satun- naisluvun väliltä Ø-1.	22Ø PRINT @RND(51Ø);
SGN(X)	Antaa argumentin etumerkin X negatiivinen antaa -1 X nolla antaa Ø X positiivinen antaa +1	412 Y=RND(ABS (N))*SGN(N)
SIN(X)	X:n sini, jossa X on kulma <u>radi-</u> <u>aaneina</u> .	2Ø5 S=SIN(K*PI/18Ø)
SQR(X)	Argumentin neliöjuuri; X ei saa olla negatiivinen. Jos X on nega- tiivinen, funktio antaa ABS(X):n neliöjuuren, jolla on negatiivinen etumerkki	33Ø C=SQR(A A+B B)
TAB(X)	Taulukon sijoittaminen ruudulle	52Ø PRINT TAB(7); "OTSAKE"

Funktion nimi	Toimitus	Esimerkki
TAN(X)	X:n tangentti, jossa X on kulma <u>radiaaneina</u> . ATN(X):n käänteis- funktio.	840 R5=B/TAN(EQ-5)

2. ryhmän funktiot

2. ryhmän funktion argumentti on numeerinen, mutta se saa arvoksi merkkijonomuuttujan. Niitä voidaan käyttää vain merkkijonotoiminnoissa.

Funktion nimi	Toimitus	Esimerkki
CHR\$(X)	Antaa merkin, jonka koodiarvo kymmenjärjestelmässä on X. X:n on oltava alueella 0-255. Ks. liite A koodiluettelo.	20 M\$=CHR\$(143)+CHR(128)
HEX\$(X)	Laskee desimaaliluvun X heksadesimaaliarvon.	42 PRINT HEX\$(30)
STR\$(X)	Muuttaa numerolausekkeen vastavaksi merkkijonoksi. VAL(X\$)-funktion vastakohta.	175 A\$=STR\$(12.49)

3. ryhmän funktiot

3. ryhmän funktiot ovat merkkijonofunktioita. Argumentit (yleensä vähintään kaksi) ovat merkkijono ja luku. Ne saavat kaikki arvokseen merkkijonon ja niitä voidaan siis käyttää vain merkkijonolausekkeissa.

Funktion nimi	Toimitus	Esimerkki
LEFT\$(X\$,N)	Antaa N ensimmäistä merkkiä merkkijonossa X\$.	114 A\$=LEFT\$(B\$,7)
MID\$(X\$,M,N)	Antaa N merkkiä merkkijonossa X\$ alkaen merkkipaikasta M. Jos N jätetään pois, saadaan koko jono merkistä M oikealle. M:n on oltava suurempi kuin 0.	760 K\$=MID\$(W\$,I,4)
RIGHT\$(X\$,N)	Antaa merkkijonon X\$ N viimeistä merkkiä.	340 T\$=RIGHT\$(Q\$,B+7)
STRING\$(N,C)	Antaa N:n pituisen merkkijonon, joka muodostuu C:n määrittelemästä merkistä. Argumentti_C voi olla joko luku (merkin ASCII-koodi) tai itse merkki lainausmerkkien sisällä.	400 A\$=STRING\$(5,67) 410 PRINT STRING\$(32,"x")

4. ryhmän funktiot

Nämä ovat 2. ryhmän funktioiden kaltaisia sekafunktioita. Niiden argumentti on merkkijono ja ne saavat numeerisen arvon, ja ne voivat siis esiintyä vain numeerisissa lausekkeissa.

Funktio nimi	Toimitus	Esimerkki
ASC(X\$)	Antaa merkkijonoargumentin ensimmäisen merkin ASCII-koodin desimaaliarvon. Ks. koodit liitteestä A.	715 P=ASC(F\$)-64
INSTR (P,S\$,T\$)	Etsii merkkijonosta S\$ merkkijonoa T\$ alkaen jonon S\$ kohdasta P. Antaa 0, ellei jonoa löydy, muussa tapauksessa ilmoittaa kohdejonoa T\$ alkukohdan.	212 F=INSTR(N,X\$,"AB")
LEN(X\$)	Antaa merkkijonon X\$ pituuden. Kaikki merkit, myös välilyönnit, lasketaan. Jos merkkijono on tyhjä, antaa 0.	845 N=LEN(N\$)
VAL(X\$)	Muuttaa numeroista koostuvan merkkijonon luvuksi. Jos merkkijono alkaa kirjaimella, antaa 0. STR\$(X):n vastakohta.	92 Z=VAL(AB\$)

5. ryhmän funktiot

5. ryhmän funktiot ovat systeemifunktioita. Niillä ei ole argumentteja.

Funktion nimi	Toimitus	Esimerkki
---------------	----------	-----------

INKEY\$	Tarkistaa näppäimistön ja antaa näppäilymerkin (jos näppäintä on painettu). Antaa merkkijonon, joten INKEY\$-funktiota on käytettävä merkkijonolausekkeessa.	146 P3\$=INKEY\$
MEM	Antaa käytettävissä olevan vapaan muistitilan.	PRINT MEM
TIMER	Antaa kellopulsstilaskurin arvon, joka on alueella 0-65535. Nollataan käskyllä TIMER=0. Timer-väli on noin 22 millisekuntia.	62 T1=TIMER-T 63 TIMER=0

OMAT FUNKTIOSI

Koneen antamien funktioiden lisäksi voit itse luoda noin 26 omaa numeerista funktiota. Nämä funktiot määritellään seuraavalla lauseella:

```
DEF FN kirjain (valemuuttuja) = laskukaava
```

Kirjain on mikä tahansa aakkosten 26 kirjaimesta välillä A-Z. Valemuuttuja on kirjain, joka korvataan funktion argumentilla, kun funktiota kutsutaan. Laskukaava on Basic-lauseke, joka koostuu valemuuttujasta ja/tai muista muuttujista. Lausekkeessa voi olla myös muita funktioita, joko rakenteeseen sisältyviä tai määriteltyjä, mutta funktio ei voi kutsua itse itseään.

Yhtälö $y = ((x-3)^2 + (x-4)^4) / x^3$ voidaan kääntää suoraan määrittelyksi funktioksi:

```
25 DEF FNY(X)=((X-3)^2+(X-4)^4)/X^3
```

Tässä X on valemuuttuja eikä muuttujanimi. Kun funktiota kutsutaan ohjelman myöhemmässä vaiheessa, se korvataan argumentilla. Funktiota käytetään sisällyttämällä se lausekkeeseen samalla tavoin kuin rakenteeseen sisältyvä funktio.

```
150 Y(I)=FNY(X)+FNY(W)
```

Funktiota voidaan siis käyttää myös apuohjelmina (yleisesti käytettyjä toimintoja). Olet ehkä huomannut, että koneen ruudulle tulostamat numerot ovat hieman rönsyileviä. Tietokone pyrkii mahdollisimman suureen tarkkuuteen lukuja kirjoittaessaan. Aina tällainen tarkkuus ei kuitenkaan ole tarpeen ja toisinaan se on suorastaan haitallista, kuten rahassummaa tulostettaessa. Seuraavalla funktiolla tulokseen saadaan haluttu määrä desimaaleja (D).

```
10 DEF FND(X)=INT(X*10^D+0.5)/10^D
```


Huomaa, että X on valemuuttuja mutta D ei ole, vaan D:n arvo on annettava funktion ulkopuolelta, esimerkiksi INPUT-lauseella. Esimerkki funktion käytöstä:

```
205 PRINT FND(A)
```

Koska kaikki trigonometriset funktiot vaativat, että argumentit ilmoitetaan radiaaneina, tarvittaisiin funktio, joka muuttaa asteet radiaaneiksi:

```
10 DEF FNR(X)=X/57.295779
```

Tarkempi tulos saadaan lauseella:

```
10 DEF FNR(X)=X*ATN(1.0)/45
```

Vakio pii (π) voidaan muodostaa seuraavasti:

```
20 DEF FNP(X)=4.0*ATN(1.0)
```

Huomaa, että tässä tapauksessa valemuuttujalla ei ole mitään vaikutusta. Se on mukana vain, koska argumentti vaaditaan.

Funktiota ei voida kutsua, ellei sitä ole aikaisemmin määritelty, joten funktiomääritelmät on aina hyvä sijoittaa ohjelman alkuun.

MUITA SYÖTTÖKÄSKYJÄ

Toistaiseksi olemme antaneet muuttujille arvoja vain INPUT-lauseella. Tämä ei ole erityisen käytännöllistä. Olet ehkä jo huomannut, että INPUT-lause ei hyväsky kaikkia merkkejä. Jos merkkijono aloitetaan välilyönneillä, ne häviävät, ja jos käytetään pilkkua, kaikki mikä tulee pilkun jälkeen, häviää. Vaihtoehto on LINE INPUT-lause:

```
LINE INPUT "ilmoitus"; merkkijonomuuttuja
```

LINE INPUT toimii samalla tavoin kuin INPUT, mutta se hyväksyy kaikki merkit, myös välilyönnit ja pilkut. Ilmoitus on sama kuin INPUT-lauseessa, ja merkkijonomuuttuja voi olla mikä tahansa jonomuuttuja. Yhdessä LINE INPUT^l-lauseessa voi olla vain yksi muuttuja.

```
25 LINE INPUT "NÄPPÄILE TEKSTIRIVI";L$
```

Usein on välttämätöntä muodostaa määrätty määrä vakioita ennen kuin ohjelma alkaa todella toimia. Nämä voidaan tietenkin antaa erikseen INPUT-lauseella joka kerta, kun ohjelmaa ajetaan. Käytännöllisempää on kuitenkin käyttää READ ja DATA-lauseita. Niitä käytetään aina yhdessä ja muoto on:

```
READ muuttujaluettelo
```

```
DATA arvoluettelo
```

READ-lause toimii samalla tavoin kuin INPUT-lause, mutta se ei pysäytä ohjelmaa ja odota arvon antamista, vaan lukee arvon suoraan DATA-lauseesta, joka kuuluu osana ohjelmaan.

DATA-lause voidaan sijoittaa mihin tahansa ohjelmassa. Jos DATA-lauseita on useampia kuin yksi, READ aloittaa lauseesta, jonka rivinumero on pienin ja jatkaa siitä järjestyksessä.

```
10 DATA 1,2,3,4,5
```

```
20 FOR I=1 TO 3
```

```
30 READ A:PRINT A:NEXT I
```

```
40 READ D,G:PRINT D,G
```

Ylläoleva esimerkki lukee DATA-luettelon ensimmäisen alkion (1), tulostaa sen, lukee toisen (2) jne. Kun kaikki alkiot on luettu, osoitin siirtyy seuraavan alkion kohdalle. Rivi 40 lukee kaksi viimeistä alkiota. Jos lisäät nyt rivin

```
50 READ X:PRINT X
```

ja ajat ohjelman uudelleen, saat vastaukseksi virheilmoituksen ?OD ERROR IN 50. Tämä tarkoittaa, ettei DATA-luettelossa ole enää jäljellä alkioita (OD=out of data). Lisää vielä rivi:

```
45 RESTORE
```

Nyt ohjelma taas toimii ja X saa arvon 1. RESTORE-lause palauttaa osoittimen jälleen ensimmäisen DATA-lauseen alkuun. DATA-lauseisiin voidaan sisällyttää myös merkkijonoja. Jos READ-lauseessa on sekalaisia muuttujia, on huolehdittava siitä, että arvot ovat samassa järjestyksessä DATA-luettelossa. Jos olet kirjoittanut ohjelman, jossa on runsaasti merkkijonoja, olet jo ehkä saanut virheilmoituksen ?OS ERROR. Tämä tarkoittaa, että jonomuuttujille varattu muistitila on loppunut. CLEAR-lauseella merkkijonoille voidaan varata lisää muistitilaa.

```
10 CLEAR 1000
```

Tämä lause varaa muistista 1000 merkin (bitin) tilan merkkijonomuuttujien tallentamista varten. Koska CLEAR samalla nolaa kaikki muuttujat, sitä voidaan käyttää vain ohjelman alussa.

KERTAUSTA

Tässä kappaleessa ja kappaleissa 1, 2, 3 ja 5 esitetyt tiedot ovat Basic-kielen ydin. Myöhemmin esitellään vielä lisää kommentoja, mutta ne ovat enemmänkin täydentäviä.

Seuraavissa kappaleissa käytetään hyväksi kaikkea tähän mennessä esiteltyjä tietoja, sillä ne ovat oleellisena osana useimmissa ohjelmissa. Ennen kuin siirrytään piirtämään kuvia tietokoneella, on hyvä kuitenkin kerrata tähän mennessä opittuja tietoja ja varmistua siitä, että kaikki on ymmärretty. Tämä helpottaa grafiikkaohjelmien laatimista.

Käy esimerkit uudelleen läpi ja yritä muuttaa niitä itsellesi sopiviksi. Lopetamme tämän kappaleen pariin uuteen esimerkkiin. Ensimmäisessä on laajennettu edellisen kappaleen korttipakan sekoitusohjelmaa. Ohjelma "jakaa" kortit pelaajille. Kiinnitä huomiota seuraaviin seikkoihin:

1. Pakan sekoittaminen tapahtuu nyt riviltä 190 alkavassa ali-ohjelmassa.
2. READ- ja DATA-lauseita käytetään aloitustaulukoiden luomiseen.
3. Rivit 130 ja 140 etsivät maan ja määräävät, mikä kortti asianomaisesta maasta jaetaan.

```
10 DIM X(52),PAKKA(52),KORTTI$(13),MAA$(3)
20 FOR I=0 TO 3:READ MAA$(I):NEXT I
30 DATA PATA,RUUTU,RISTI,HERTTA
40 FOR I=1 TO 13:READ KORTTI$(I):NEXT I
50 DATA ÄSSÄ,KAKKONEN,KOLMONEN,NELONEN,VIITONEN,KUUTONEN,SEISKA
60 DATA KAHDESIKKO,YHDESIKKÖ,KYMPPI,SOTAMIES,KUNINGATAR,
KUNINGAS
```

```

70 CLS:INPUT "KUINKA MONTA KORTTIA JAETAAN";N
80 GOSUB 190
90 ALKU=1
100 LOPPU=ALKU+N-1:IF LOPPU>52 THEN GOTO 80
110 CLS:PRINT @ 10,"KORTTISI OVAT":PRINT:PRINT
120 FOR I=ALKU TO LOPPU
130 F=INT((PAKKA(I)-1)/13)
140 K=PAKKA(I)-F 13
150 PRINT TAB(8);MAA$(F);"-";KORTTI$(K)
160 NEXT I:ALKU=ALKU+N
170 PRINT @ 448,"UUSI JAKO. KYLLÄ TAI EI";:INPUT A$
180 IF A$="KYLLÄ" THEN 100 ELSE END
190 FOR I9=1 TO 52:X(I9):NEXT I9
200 FOR I9=52 TO 1 STEP -1
210 J9=RND(I9):PAKKA(I9)=X(J9):X(J9)=X(I9)
220 NEXT I9:RETURN

```

Toisessa esimerkissä käytetään melkein kaikkia mahdollisia merkkijonofunktioita. Ohjelma tarkistaa annetun tekstin ja ilmoittaa jokaisen kirjaimen esiintymistiheyden. Tällaista ohjelmaa käytetään usein koodikielisiä sanomia tulkittaessa. Näkemällä hiukan vaivaa tätä ohjelmaa voidaan muuttaa siten, että se etsii jonkin määrätyn sanan tai merkkijohdistelmän.

```

10 CLEAR 1000:CLS:READ A$
20 DATA ABCDEFGHIJKLMNOPQRSTUVWXYZ
30 PRINT "ANNA TEKSTIRIVI":PRINT
40 LINE INPUT L$
50 FOR I=1 TO LEN(A$):CLS

```

```

60 T$=MID$(A$),I,1):C=0:P=1:P$=L$
70 F=INSTR(P,L$,T$)
80 IF F>0 THEN C=C+1 ELSE 140
90 P$=LEFT$(P$,F-1)+STRING$(LEN(T$),CHR$(128))
100 IF F>LEN(L$)THEN 120
110 P$=P$+RIGHT$(L$,LEN(L$)-F)
120 P=F+LEN(T$)
130 IF P<=LEN(L$)-LEN(T$)+1 THEN 70
140 PRINT P$
150 PRINT@ 354, "LÖYTNYT";C;"KERTAA KIRJAIN";T$
160 PRINT@ 416, "ANNA VÄLILYÖNTI JOS HALUAT JATKAA JA N JOS
HALUAT LOPETTA"
170 Z$=INSKEY$:IF Z$=""THEN 170
180 IF Z$="N"THEN 200
190 NEXT i
200 CLS:END

```

TÄMÄ SIVU ON TARKOITUKSELLA
JÄTETTY TYHJÄKSI.

DEF FN

DEF FN-lausetta käytetään numeerisen funktion määrittämiseen.

DEF FN nimi (valemuuttuja)= laskukaava

Nimi voi olla mikä tahansa kirjain välillä A-Z.

Valemuuttuja voi olla mikä tahansa kirjain, ja se korvataan argumentilla, kun funktiota käytetään. Vain yhtä valemuetujaa voidaan käyttää.

Laskukaava kuvaa toiminto valemuetujan ja/tai muiden muuttujien avulla.

Määritellyt funktiot on kirjoitettava yhdelle ohjelmariville.

Määritellyssä funktiossa voidaan käyttää laskukaavassa muita funktioita, joko määriteltyjä tai rakenteeseen sisältyviä.

Ne eivät kuitenkaan voi kutsua (käyttää) itseään.

Funktio on määriteltävä ennen kuin sitä käytetään. Funktiot onkin syytä määritellä ohjelman alussa.

Muita matemaattisia funktioita voidaan määritellä käyttäjäkohtaisiksi funktioiksi seuraavasti:

10 DEF FNS(X)=1/COS(X):'SEKANTTI

20 DEF FNI(X)=-ATN(X/SQR(-X*X+1))+1.5708

30 DEF FNH(X)=-EXP(X)/(EXP(X)+EXP(-X))*2+1

40 DEF FNM(A)=INT((A/B-INT(A/B))*B+0.5)*SGN(A/B)

50 B=8:PRINT FNM(13)

LINE INPUT

LINE INPUT-lause sijoittaa koko rivin merkkijonomuuttujaan, myös pilkut ja alussa olevat välilyönnit, joita INPUT-lause ei hyväsku.

LINE INPUT "ilmoitus"; merkkijonomuuttuja

Ilmoitus on mikä tahansa lainausmerkeissä oleva ilmoitus. Se voidaan jättää pois, jos sitä käytetään, se on erotettava merkkijonomuuttujasta puolipisteellä(;). Merkkijonomuuttuja voi olla mikä tahansa jonomuuttuja. LINE INPUT-lauseessa voi olla vain yksi muuttuja. LINE INPUT-komennolla tallennettavan rivin enimmäispituus on 255 merkkiä.

10 CLEAR 500:CLS

20 LINE INPUT "ANNA KOKO NIMESI";N\$

30 LINE INPUT "JA OSOITTEESI";O\$

READ

DATA

RESTORE

READ-lause lukee seuraavan alkion DATA-riviltä ja antaa alkion arvon määrätyn muuttujan arvoksi muuttujaluettelossa.

READ muuttujanimien_luettelo

DATA-lause tallentaa tietoa ohjelman sisällä ja se voi olla numeroitu rivi missä kohdassa ohjelmaa tahansa.

DATA arvoluettelo

READ- ja DATA-lauseissa voidaan käyttää sekä merkkijono- että numeromuuttujia, jos ne ovat oikeassa järjestyksessä. Jono-
muuttujan arvoksi on annettava merkkijono jne. Luettelon al-
kiot erotetaan toisistaan pilkulla(,).

RESTORE-lause siirtää osoittimen takaisin pienimmällä rivi-
numerolla merkityn DATA-rivin ensimmäiseen alkioon. Seuraava
suoritettava READ-lause lukee siten arvot ohjelman ensimmäi-
sestä DATA-lauseesta.

RESTORE

Pieni ohjelmaesimerkki valaisee käyttöä:

```
10 CLS:PRINT:PRINT:PRINT
20 READ A,B:IF A=-9999 THEN RESTORE:GOTO 20
30 PRINT A;"V+V5VONV";:INPUT C
40 IF C=B THEN PRINT "OIKEIN" ELSE PRINT "VÄÄRIN"
50 FOR D=1 TO 600:NEXT D:GOTO 10
60 DATA 8,13,12,17,5,10,27,32,14,19,3,8
70 DATA 7,12,6,11,1,6,-9999,-9999
```

CLEAR

CLEAR-lause nolaa kaikki muuttajat ja varaa muistista tilaa merkkijonomuuttujille.

CLEAR 500

varaa 500 merkkiä (bittiiä) muistitilaa merkkijonomuuttujia varten. CLEAR-lauseella voidaan määrätä myös suurin Basic-muistiosoite ja varata tilaa konekielisiä ohjelmia varten.

CLEAR 200,14000

varaa 200 merkkiä (bittiiä) merkkijonomuuttujia varten ja asettaa suurimmaksi Basic-osoitteeksi 14000. Konekieliset ohjelmat voidaan nyt tallentaa 14001:stä eteenpäin.

Ellei CLEAR-käskyä käytetä, kone varaa automaattisesti 200 merkkiä merkkijonotilaa.

7. KUVAT JA LIIKKUVAT KUVIOT

PISTE ON PERUSOSA

Kaikki, minkä kone tulostaa ruudulle, muodostuu pisteistä. Kuvia muodostaessaan se kytkee määrättyt pisteet päälle, jolloin ne näkyvät ruudulla värillisinä, ja muut ovat mustia. Kaikki näppäilemämme kirjaimet muodostuvat näistä valopisteistä. Pisteiden koon määrää kulloinkin käytössä oleva grafiikkatila. Jos pisteet ovat suuria, käytössä on puoli-grafiikka (low resolution), ja jos pisteet ovat pieniä, hienografiikka (high resolution) (mitä pienempiä pisteet ovat, sitä useampia pisteitä ruudulle mahtuu).

Tietokone kykenee työskentelemään neljässä eri grafiikkatilassa, alkaen 512 pisteestä ja päättyen 49152 pisteeseen ruudulla. Valinnanvaraa on siis runsaasti päätettäessä, miten paljon yksityiskohtia piirrettäviin kuviin halutaan. Aloitamme kuvien piirtämisen karkeimmasta grafiikkatilasta ja etenemme hienompaan suuntaan. Piirroksia ja liikkuvia kuvia laadittaessa käytetään hyvin samankaltaisia menetelmiä grafiikkatilasta riippumatta.

KUVIEN LAATIMINEN

Kappaleessa 3 PRINT@-lauseen yhteydessä kuvattiin näyttöruudun 16x32 merkkipaikkaa. Voimme siis sijoittaa merkin mihin kohtaan tahansa ruudulla ilmoittamalla halutun merkkipaikan.

CHR\$-funktion avulla (ks. kappale 6, funktiot) voimme piirtää graafisia erikoismerkkejä. Seuraava ohjelma näyttää kaikki merkit, joita CHR\$-funktion avulla voidaan käyttää.

```
10 FOR I=1 TO 255:CLS 0
20 PRINT@100,"CHR$(";I;")";
30 PRINT@120,CHR$(I);
40 FOR D=1 TO 600:NEXT D,I:CLS
```

Lukuja 1-31 käytetään kontrollimerkkeinä, joten ne eivät näy ruudulla. 32-127 ovat näppäimistön merkit. Koodit 128-255 ovat graafiset erikoismerkit. Täydellinen luettelo käytettävissä olevista merkeistä on esitetty liitteessä A. Graafiset merkit ovat värikuvioita, joista voidaan koota yksinkertaisia muotoja. Yksinkertaisin kuvio on värillinen suorakulmio. Esimerkiksi CHR\$(143) on vihreä suorakulmio (vihreä on väri n:o 1). Kun lisätään 16, saadaan CHR\$(159), joka on keltainen (väri 2) suorakulmio. Voit jatkaa samalla tavoin. Kuvioita on 16, CHR\$(128)-CHR\$(143), joiden värit ovat vihreä ja musta. Jos haluat saman kuvion erivärisenä, sinun tarvitsee vain lisätä koodiin tarvittava 16 jaollinen luku.

+16 keltainen	+32 sininen	+48 punainen
+64 valkoinen	+80 turkoosi	+96 sinipunainen
+112 oranssi		

Seuraava ohjelma osoittaa, mitä tapahtuu, kun koodiin lisätään 16. Ohjelmaa voidaan käyttää myös televisiovastaanottimen värisävyä säätämiseen. Käytä tällöin V=143.

```
10 CLS0:INPUT"ANNA KOODI VÄLILTÄ 128-143";V
```

```

20 FOR I=1 TO 14:FOR J=V TO 255 STEP 16
30 FOR K=1 TO 4:PRINT CHR$(J);:NEXT K,J,I
40 GOTO 40

```

Kuten ylläoleva osoittaa, CHR\$-merkit voidaan kirjoittaa suoraan ruudulle. Ne ovat kuitenkin merkkejä, joten ne voidaan sijoittaa myös merkkijonomuuttujiin. Niitä on silloin huomattavasti helpompi käsitellä.

Alamme nyt piirtää kuvaa. Piirrämme linnan, koska sen muoto on yksinkertainen ja koska se osoittaa, miten yksinkertaiseen kuvioon voidaan vähitellen lisätä yhä enemmän yksityiskohtia. Näppäile ja suorita jokainen osa yksitellen sitä mukaa kuin ne annetaan, niin voit nähdä piirroksen eri vaiheet.

Ensiksi rakennamme muurin ruudun poikki. Se on siis 32 lohkoa leveä ja teemme siitä 6 lohkoa korkean.

```

10 CLEAR 500:CLS 0
20 FOR I=1 TO 6:FOR J=1 TO 32
30 MUURI$=MUURI$+CHR$(207):NEXT J,I
40 PRINT@256,MUURI$;
200 GOTO 200

```

Ensimmäinen rivi varaa tilaa tarvitsemillemme merkkijonoille. Rivit 20 ja 30 'rakentavat' muurin valkoisista lohkoista, CHR\$(207), ja tallentavat sen merkkijonon MUURI\$ arvoksi. Rivi 40 tulostaa muuttujan MUURI\$, joka tulee näkyviin ruudulle suurena värillisenä suorakulmiona. Viimeinen rivi (200) vain pitää kuvan ruudulla.

Seuraavaksi liitämme muuriin sakarat. Teemme sen käyttämällä vuorotellen valkoisia ja mustia lohkoja. Tähän tarvitaan vain yksi rivi.

```
50 FOR I=1 TO 16:B$=B$+CHR$(128)+CHR$(207):NEXT I
60 PRINT@224,B$;
```

Rivi 50 määrittelee sakarat ja 60 piirtää ne muurin harjalle. Seuraavaksi tarvitaan torni. Torni rakennetaan samasta materiaalista kuin muuri; otetaan siis muutama tiili MUURI\$:sta.

```
70 P=11
```

```
80 FOR I=1 TO 3:PRINT@128+32*I+P,LEFT$(MUURI$,10);
:NEXT I
```

Rivi 80 ottaa 10 tiiltä MUURI\$:sta ja rakentaa 3 kerrosta muurin keskelle. P:n arvo sijoittaa tornin keskelle. Jos haluat sijoittaa tornin johonkin muuhun kohtaan, voit muuttaa P:n arvoa. Torni on 10 asemaa leveä, joten P voi olla mikä tahansa luku välillä 0-22. Nyt voimme lisätä sakarat myös torniin.

```
90 PRINT@128+P,LEFT$(B$,10);
```

Linnan perusrakenteet ovat nyt valmiit. Ampuma-aukkoja voidaan lisätä käyttämällä eri merkkiä ja päällekirjoitusta.

```
100 FOR I=2 TO 32 STEP 4:PRINT@288+I,CHR$(206);:NEXT I
```

Tarvitaan myös portti. Sitä varten piirretään mustia lohkoja sopivaan paikkaan.

```
110 C$=CHR$(128)+CHR$(128)+CHR$(128)
```

```
120 FOR I=353 TO 417 STEP 32:PRINT@I+P+5,G$;:NEXT I
```

Portti tallennetaan muuttujan G\$ arvoksi, ja sen tulostaa rivi 120. PRINT@-lausekkeen P sijoittaa sen samaan linjaan tornin kanssa.

Linnasta, jonka portti on aina avoin, ei ole suurtakaan hyötyä, joten tarvitaan myös laskuristikko. Käyttämällä toista merkkiä, (142+112), voimme muodostaa eräänlaisen laskuristikon.

```
130 P$=CHR$(254)+CHR$(254)+CHR$(254)
```

```
140 FOR I=353 TO 417 STEP 32:PRINT I+P+5,P$;
```

```
150 FOR K=1 TO 300:NEXT K,I
```

Rivi 130 piirtää oranssin laskuristikon ylhäältä alaspäin.

Hidastussilmukka K laskee ristikon hitaasti alas.

Linna on nyt valmis, mutta halutessasi voit lisätä siihen vielä esimerkiksi sinisen vallihaudan, lipun tornin huippuun tai muita yksityiskohtia.

LIIKKUVAT KUVAT

Linnaesimerkissä rivit 140 ja 150 osoittivat, miten kuvaan saadaan liikettä suorittamalla osat peräkkäin. Tällainen liike rajoittuu kuitenkin pääasiassa esimerkiksi oven aukeamiseen ja sulkeutumiseen. Huomattavasti parempi menetelmä on piirtää kuva ensin kokonaan, pyyhkiä se sitten pois ja piirtää uudelleen hieman erilaisessa asennossa. Koska kuva piirretään useita kertoja uudelleen, piirtämisosan pitäisi olla aliohjelma. Ensin laadimme kuvan 3x4 ruudun kokoiseksi. Ylin rivi on pää, seuraava on vartalo ja viimeinen jalat.


```

10 CLEAR 500:CLS 0
20 M1$=CHR$(128)+CHR$(193)+CHR$(194)+CHR$(128)
30 M2$=CHR$(196)+CHR$(207)+CHR$(207)+CHR$(200)
40 M3$=CHR$(128)+CHR$(202)+CHR$(197)+CHR$(128)

```

Kuva on nyt tallennettu kolmeen merkkijonomuuttujaan M1\$, M2\$ ja M3\$. Seuraava aliohjelma tulostaa merkkijonot oikeassa järjestyksessä.

```

500 P=32*Y+X
510 PRINT@ P,M1$;:PRINT@ P+32,M2$;
520 PRINT@ P+64,M3$;:RETURN

```

Tämä ohjelma piirtää kuvan kolmena alekkaisena rivinä alkaen kohdasta, jonka määrittävät X:n ja Y:n arvot. Kuten muistetaan, X ja Y ovat PRINT@-lokerossa kuvatun ruudukon koordinaatit. X on 0-31 vaaka-akselilla ja Y on 0-15 pystyakselilla. Sinun on näppäiltävä kaikki allaolevat rivit riville 160 asti ennen kuin voit suorittaa ohjelman.

Jotta saisimme hahmon liikkumaan, meidän on nyt siis muutettava piirroksen sijaintia ruudulla. X:n ja Y:n arvoa on siis muutettava. Tämä voidaan tehdä näppäimistöltä. Käytämme INKEY\$-lausetta näppäimistön lukemiseen, ja luonnollisinta on käyttää nuolinäppäimiä. Myös näillä näppäimillä on koodit, kuten kirjaimillakin:

```

[+] CHR$(8)
[+] CHR$(9)
[+] CHR$(10)
[↑] CHR$(94)

```

X:n arvolla säilytetään kuvan vaaka-asento ja Y:n avulla kuvan pystyasento. Jos painetaan + -näppäintä, kuvaa halutaan siirtää vasemmalla, joten X:n arvosta vähennetään 1. On kuitenkin varottava, ettei kuviota siirretä pois ruudulta.

```
90 GOSUB 500
```

```
100 A$=INKEY$:IF A$=""THEN 100
```

```
120 IF A$=CHR$(8) THEN X=X-1:IF X<0 THEN X=0
```

Rivi 100 lukee näppäimistöä, kunnes jotakin näppäintä painetaan. Jos painetaan + -näppäintä, rivi 120 vähentää yhden X:n arvosta, tarkistaa siirtyykö kuva pois ruudulta, ja jos se siirtyy, pysäyttää kuvan vasempaan reunaan. Aivan samalla tavoin kuvaa voidaan siirtää oikealle sekä ylös ja alas.

```
130 IF A$=CHR$(9) THEN X=X+1:IF X>28 THEN X=28
```

```
140 IF A$=CHR$(94) THEN Y=Y-1:IF Y<0 THEN Y=0
```

```
150 IF A$=CHR$(10) THEN Y=Y+1:IF Y>13 THEN Y=13
```

```
160 GOSUB 500:GOTO 100
```

Rivillä 130 X:n suurimmaksi sallituksi arvoksi asetetaan 28. Se ei missään tapauksessa voi olla suurempi kuin 31, ja on muistettava, että kuva on neljän lohkon levyinen. Sama pätee Y:n arvoon: on muistettava jättää tilaa kolmelle riville. Rivi 160 kutsuu aliohjelman, joka piirtää kuvan; sen jälkeen palataan jälleen riville 100 ja tarkistetaan, onko jotakin näppäintä painettu. Jos ohjelma suoritetaan nyt, kuvaa voidaan kyllä siirrellä edestakaisin ruudulla, mutta siitä syntyy melkoinen kaaos, koska kuvaa ei pyyhitä pois edellisistä sijaintikohdista. Tätä varten tarvitaan pois-

pyyhkivä merkkijono ja aliohjelma, joka tulostaa sen.

```
50 BL$=CHR$(128)+CHR$(128)+CHR$(128)+CHR$(128)
```

```
600 P=32*X+X:PRINT@ P,BL$;
```

```
610 PRINT@ P+32,BL$;:PRINT@ P+64,BL$;
```

```
620 RETURN
```

Uusi aliohjelma (600) tekee täsmälleen saman kuin ensimmäisenkin aliohjelma, mutta tällä kertaa se tulostaa mustia lohkoja. Nyt kuva vain pyyhittää pois juuri ennen kuin sitä siirretään.

```
110 GOSUB 600
```

Kuvaa pitäisi nyt voida liikutella mihin tahansa ruudulla. Tällaisessa muodossa ohjelma on vain esimerkki, mutta tämäläntyyppisiä liikkuvia kuvia voidaan käyttää peleissä ja yksinkertaisissa opetusohjelmissa.

UUSI GRAFIIKKATASO

Siirrymme nyt seuraavaan grafiikkatasoon. Tässä ruudukko on 32x64 eli ruudulla on 2048 pistettä. Tämä ja edellinen taso, 16x32, ovat puoligrafiikkatasoja ja niitä voidaan haluttaessa käyttää yhdessä. Tämän ruudun pisteet voidaan sytyttää ja sammuttaa kahdella käskyllä:

```
SET(X,Y,V) ja RESET(X,Y)
```

SET-komennolla piste X,Y sytytyy ja V määrää sen värin. X:n arvo on 0-63 ja Y:n arvo 0-31 ja ne ovat vaaka- ja pystyakseli kuten aikaisemminkin. V on sen värin koodi, jonka haluat pisteen väriksi, ja se on siis jokin luku välillä 0-8.

RESET-komento sammuttaa pisteen X,Y. Näillä komennoilla kuva saadaan liikkumaan, kun pisteitä sammutetaan ja sytytetään määrättyssä järjestyksessä. Kokeile seuraavaa ohjelmaa:

```
10 CLS 0:X1=0:Y1=0:XI=2:YI=2
20 X2=X1+XI:IF X2>63 OR X2<0 THEN XI=-XI:SOUND 180,1:GOTO 20
30 Y2=Y1+YI:IF Y2>31 OR Y2<0 THEN YI=-YI:SOUND 180,1:GOTO 30
40 SET (X2,Y2,8):RESET (X1,Y1):X1=X2:Y1=Y2:GOTO 20
```

Voit nähdä, mitä ohjelma tekee, mutta kuinka se tapahtuu? Ohjelma lisää pisteestä X1,Y1 alkaen X1:n arvoa vähitellen pienellä arvolla XI ja Y1:n arvoa YI:llä ja luo näin uuden pisteen X2,Y2. Rivi 40 sytyttää uuden pisteen ja sammuttaa vanhan (X1,Y1). X2,Y2 pisteestä tulee nyt vanha piste (X1, Y1) ja ohjelma haarautuu takaisin riville 20 ja muodostaa jälleen uuden pisteen X2,Y2. Näin palloa siirretään ruudulla. Kun pallo tulee ruudun reunaan, lisäyksen etumerkki muuttuu. Esimerkiksi X:n arvo alkaa tällöin pienentyä, ja liikkeen suunta muuttuu. Näin pallo saadaan pomppaamaan reunoista takaisin. Lisäyksen (XI ja YI rivillä 10) arvoa muutamalla pallo saadaan liikkumaan eri nopeuksilla. Tämän tyyppinen ohjelma on pohjana useimmissa tietokoneella pelattavissa pallopeleissä, mutta ne on yleensä kirjoitettu konekielillä eikä Basic-kielellä.

Liikkuvia pisteitä voidaan käyttää myös "ammuntapeleissä". Tällaisissa peleissä pisteen on liikuttava ruudun poikki, ja lisäksi tarvitaan ase, joka voidaan laukaista. Tässäkin voitaisiin käyttää nuolinäppäimiä, kuten aiemminkin, mutta

ohjausvivut ovat huomattavasti kätevämpiä. Ne kytketään tietokoneen sivuseinämässä olevaan liitoskohtaan, ja liikettä voidaan ohjaillla niiden avulla paljon sujuvammin kuin nuolinäppäimillä. Ohjausvivun asento luetaan funktiolla JOYSTK. JOYSTK(0) antaa vasemman ohjausvivun vaaka-asennon ja JOYSTK(1) pystyasennon. JOYSTK(2) ja JOYSTK(3) antavat oikean ohjausvivun vastaavat koordinaatit. Koska funktion antama arvo on aina alueella 0-63, se on syytä määrittää siten, että se sopii ruudulla käytössä olevaan grafiikkatilaan.

```
10 CLS 0:FOR I=0 TO 3
20 PRINT@ 74+32*I,"OHJAUSVIPU(";I;)"VV";JOYSTK(I);
30 NEXT I:FOR D=1 TO 400:NEXT D:GOTO 10
```

Suorita ylläoleva ohjelma ja käytä ohjausvipuja. Näet, että arvot muuttuvat, kun ohjausvivun asento muuttuu. Voit käyttää myös ohjausvivussa olevaa painiketta. Lisää rivi:

```
25 P=PEEK(65280):PRINT@ 202,"PAINIKEARVOVV";P;
```

PEEK-funktio käskee konetta tarkistamaan määrätyn kohdan muistissa. Muistiosoitteessa 65280 on ohjausvivun painikkeen lukemisen tulos. Se on joko 127 tai 255. Jos painetaan alas vasen painike, se muuttuu 125:ksi tai 253:ksi. Jos painetaan oikeaa painiketta, se muuttuu 126:ksi tai 254:ksi. Jos molemmat painikkeet on painettu alas samanaikaisesti, luku on 124 tai 252.

Alamme nyt laatia peliohjelmaa - taistelua kahden avaruusaluksen välillä. Voimme käyttää ohjausvipuja alusten liikuttelamiseen ja painikkeita ammusten laukaisemiseen.

Piirrämme ensin avaruusaluksen käyttäen vastaavaa menetelmää kuin aikaisemminkin. Kumpikin avaruusalus on 2x3 lohkon kokoinen, toinen on keltainen ja toinen sininen, ja ne tallennetaan merkkijonotaulukoihin S\$ ja S2\$.

```
10 CLEAR 500:FOR I=0 TO 5:READ S(I):NEXT I
20 DATA 128,131,128,134,140,137
30 FOR Y=0 TO 1:V=(Y+1)*16
40 S$(Y)=CHR$(S(0)+V)+CHR$(S(1)+V)+CHR$(S(2)+V)
50 S2$(Y)=CHR$(S(3)+V)+CHR$(S(4)+V)+CHR$(S(5)+V)
60 NEXT Y
```

Ellet pidä avaruusalusten muodosta, voit tehdä niistä sellaiset kuin haluat muuttamalla rivin 20 tietoja.

Seuraavaksi on luettava ohjausvipujen asento, tarkistettava, että pysytään ruudulla, ja päätettävä alusten sijainnista.

```
70 FOR Y=0 TO 1:A(Y)=JOYSTK(Y*2):
80 B(Y)=INT(JOYSTK(1+Y*2)/2)
85 IF A(Y)>58 THEN A(Y)=58
90 IF A(Y)<2 THEN A(Y)=2
100 IF B(Y)>27 THEN B(Y)=27
110 L(Y)=INT(B(Y)/2)*32+INT(A(Y)/2):NEXT Y
```

Ohjausvipujen asennot luetaan vuorotellen riveillä 70 ja 80. Rajat asetetaan riveillä 85-100 (muista alusten koko). Lopputulos muutetaan sitten arvoksi PRINT@-komentoon. Tämä on esimerkki siitä, miten kahta puoligrafiikkatilaa voidaan käyttää sekaisin - ohjausvipu toimii toisella ja PRINT@-komento toisella tasolla.

Seuraavaksi on piirrettävä alukset ja palattava tarkistamaan, onko ohjausvipuja liikuteltu.

```
120 CLS 0:FOR Y=0 TO 1:PRINT@ 0,Z(0);:PRINT@ 26,Z(1);
130 PRINT@ L(Y)+S$(Y);:PRINT@ L(Y)+32,S2$(Y);:NEXT Y
170 A$=INKEY$:IF A$=""THEN 70
180 CLS:END
```

Rivi 120 kirjoittaa myös tuloksen, mutta sitä osaa ei ole vielä laadittu valmiiksi. Pelaaminen voidaan lopettaa painamalla mitä tahansa näppäimistön näppäintä, muutoin ohjelma haarautuu jälleen riville 70 ja lukee uudelleen ohjausvipujen asennon. Aja ohjelma niin pitkälle ja tarkista, että avaruus-alukset liikkuvat kaikkialle ruudulla.

Seuraavassa vaiheessa laukaistaan ammuksiset ja näytetään ammuksen lentorata. Tämä on hankala vaihe, sillä on kyettävä lukemaan ohjausvipujen painikkeet ja määrittelemään, kuka on ampunut. Koska alukset voivat liikkua kaikkialla ruudulla, on myös kyettävä näyttämään ammuksen suunta. Yksinkertaistaaksemme tilannetta annamme ammuksen lähteä aluksesta vain vaakatasossa suoraan kohdealusta kohti. Se siis liikkuu samassa tasossa kuin tulittava alus.

```
140 P=PEEK(65280)
150 IF P=125 OR P=253 THEN F=1:T=0:GOSUB 200
160 IF P=126 OR P=254 THEN F=0:T=1:GOSUB 200
```

Nämä rivit lukevat ohjausvipujen painikkeet ja määrittelevät, kumpi alus tulittaa. Aliohjelma 200 näyttää ammuksen.


```

200 V1=B(F):H1=A(F):H2=A(T):ST=1
210 IF H1>H2 THEN ST=-1
220 FOR H=H1+ST*5 TO H2+2 STEP ST
240 SET (H,V1,4):SOUND 200,1:RESET (H-2*ST,V1)
250 NEXT H:RETURN

```

Huomaa vaiheen vaihto rivillä 210, jos vasen ja oikea asento vaihtuvat. Ammuksen liikkumisen saa aikaan rivi 240.

Nyt on enää tarkistettava, onko saatu osuma, ja jos on, on tuotettava asianmukainen ääni ja merkittävä muistiin pistetilanne.

Osuma tarkistetaan POINT-funktiolla. Sen muoto on POINT(X,Y), jolloin X,Y on piste, joka halutaan tarkistaa. Funktio antaa 0, ellei piste ole päällä, ja värikoodin numeron, jos pisteessä palaa valo.

Koska ruutu on musta ja ammuimme oikeaan suuntaan, meidän tarvitsee tietää vain, onko ammuksen kulkureitillä yhtään pistettä, jossa palaisi valo. Jos lisäämme aliohjelmaan seuraavan rivin:

```

230 IF POINT(H,V1)>0 THEN GOSUB 300:RETURN

```

Ohjelma kutsuu aliohjelmaa 300, mikäli alus saa osuman.

Tässä aliohjelmassa pidetään pistetilastoa ym. Tämän jälkeen ei ammusta kannata enää laukaista uudelleen, joten myös tästä aliohjelmasta poistutaan ja aloitetaan jälleen alusta.

```

300 Z(F)=Z(F)+1
310 FOR K=1 TO 15:I=RND(5)-2:J=RND(4)-2

```



```
320 SET(H+I*ST,V1+J,8):SOUND(RND(95)),1
```

```
330 NEXT K:RETURN
```

Tämä aliohjelma pitää pistetilastoa, piirtää ja antaa 'räjähdysäänen'.

Vaikka tämä ohjelma on vain 28 rivin pituinen, sillä saadaan aikaan peli, jossa on runsaasti liikettä. Harjoitustentävänä voit kehittää ja parantella sitä edelleen.

Tämä on ollut pitkä ja yksityiskohtainen kappale, mutta siihen sisältyy suurin osa niistä tekijöistä, joita tarvitaan laadittaessa graafisia esityksiä tietokoneella käytettiinpä mitä grafiikkatilaa tahansa.

SET

SET-komennolla sytytetään jokin piste ruudulla ja annetaan sille jokin määrätty väri puoligrafiikkatilassa.

SET(x,y,v)

x,y ovat ruudulla olevan pisteen koordinaatit. x:n arvon on oltava alueelta 0-63 ja y:n 0-31.

v on halutun värin värikoodi. Sen on oltava jokin luku välillä 0-8.

```
10 CLS 0:SET(5,27,8):SET(6,27,8)
```

```
20 FOR X=0 TO 6:FOR Y=28 TO 30
```

```
30 SET(X,Y,8):NEXT Y,X
```

```
40 FOR X=7 TO 63:FOR D=1 TO 200:NEXT D
50 FOR Y=27 TO 30:IF Y=27 THEN RESET(X-2,Y)
60 SET(X,Y,8):RESET(X-7,Y):NEXT Y,X
70 GOTO 70
```

RESET

RESET-komennolla sammutetaan jokin SET-komennolla sytytetty piste. Sitä käytetään puoligrafiikkatilassa.

RESET(x,y)

x,y ovat sammutettavan pisteen koordinaatit. x:n arvon on oltava alueella 0-63 ja y:n 0-31.

Piste muutetaan taustan väriseksi eli 'sammutetaan'. Ks. esimerkki SET-komennon kohdalta.

8. HIENOGRAFIIKKA

Siirrymme nyt käsittelemään hienografiikkatasoja, jotka ovat puoligrafiikkatasoista täysin erillisiä. Kahta puoligrafiikkatasoa voidaan käyttää yhdessä, ja ne tulostetaan ruudulle, jota kutsutaan 'tekstiruuduksi'.

Hienografiikkatasoja ei voida sekoittaa tekstiruudun kanssa. Toisesta voidaan siirtyä toiseen, mutta tekstiä ei voida kirjoittaa hienografiikkaruudulle eikä hienografiikkaa voida piirtää tekstiruudulle.

Kun jotakin piirretään hienografiikalla, tietokone kirjoittaa ohjeet tietojen tulostamisesta määrättyyn kohtaan muistissa. Tätä muistikohtaa kutsutaan nimellä 'video RAM'. Video RAM luetaan sitten televisioon ja muutetaan kuviksi. Video RAMissa on varattu tätä tarkoitusta varten määrätty määrä 'sivuja', yleensä neljä. Käytettyjen yksityiskohtien lisääntyessä myös tulostusta varten tarvittavien ohjeiden määrä lisääntyy. Ne vaativat tällöin myös enemmän tilaa, joten sivuja on varattava enemmän. Tämä tapahtuu PCLEAR-komennolla, jota seuraa haluttu varattavien sivujen lukumäärä. Sivujen enimmäismäärä on 8.

PCLEAR8

Koska jokainen sivu vie 1536 (1,5 k) muistipaikkaa, kannattaa varata vain todellista tarvetta vastaava sivumäärä. Käytettävissä oleva muistitila on vakio, joten mitä enemmän tilaa varataan graafisia sivuja varten sitä vähemmän jää tilaa oh-

jelmalle. PCLEAR toimii samalla tavoin kuin CLEAR ja sitä käytetään vain ohjelman alussa.

GRAFIKKATASON VALINTA

Varattavan muistitilan määrä riippuu siitä grafiikkatasosta, jota halutaan käyttää. Hienompien grafiikkatasojen haitta on, ettei niissä voida käyttää koko sitä väriasteikkoa, jota puoli-grafiikkatasoissa käytetään. Käytettävissä olevat värit ja grafiikkataso määräytyvät sen mukaan, missä tilassa työskennellään. Tila määritetään PMODE-komennolla.

PMODE tila, aloitussivu

Tila on jokin luku välillä 0-4, ja aloitussivu on se sivunumero video RAMissa, josta kirjoittaminen halutaan aloittaa. Kuten aiemminkin ruutu jakautuu ruudukosi. Tällä kertaa tarvitsee muistaa kuitenkin vain yksi koko (256x192). Vaikka grafiikkataso muuttuu tilasta riippuen, ruudun pisteet määritellään kuitenkin aina 256x192-ruudukon mukaan. Ero on piirrettävän pisteen koossa. Valittu tila määrää myös, mitä värejä voidaan käyttää. Jokaisessa tilassa on kaksi väri-
valikoimaa. Väri-
valikoima valitaan SCREEN-komennolla, joka valitsee myös ruututyyppin (puoli-/hienografiikka).

SCREEN tyyppi, väri-
valikoima

Tyyppi on 0, jos valitaan tekstiruutu, ja 1, jos valitaan hienografiikkaruutu. Väri-
valikoima on samoin 0 tai 1. Normaaliarvot, joita tähän asti olemme käyttäneet ovat SCREEN0,0.

Ne antavat tekstiruudun, jossa on musta teksti vihreällä pohjalla. (Myös SCREEN0,1 on mahdollinen. Se antaa mustan tekstin oranssilla pohjalla, mutta aina kun kone tulostaa, se vaihtaa tekstin takaisin mustaksi vihreällä pohjalla). Hienografiikkaa käytettäessä tyypin on oltava 1. SCREEN-komentoa esittelevässä lokerossa on taulukko, jossa on annettu kaikki mahdollisuudet ja värivalikoimat.

Kuten taulukosta nähdään, grafiikkataso ja värivalikoima ovat kiinteästi yhteydessä toisiinsa. Lisäksi voidaan havaita, että tilanumeron noustessa nolasta neljään myös tarvittava sivumäärä suurenee.

Yhden grafiikkaruudun tulostaminen vaatii siis PMODE0-tilassa ainoastaan yhden sivun muistitilaa mutta PMODE3- ja PMODE4-tilassa neljä sivua. Kun värivalikoima on määritetty, kone valitsee taustaväriksi sen värin, jolla on värivalikoiman pienin koodinumero. Väri, jonka numero on suurin, valitaan kuvion väriksi. Esimerkiksi PMODE3 ja SCREEN1,0 piirtää punaisen kuvion vihreälle pohjalle. Kuvion ja taustan väriä voidaan muuttaa COLOR-lauseella.

COLOR kuvio, tausta

jossa kuvio ja tausta ovat halutut värikoodit siitä värivalikoimasta, jota asianomaisessa tilassa voidaan käyttää.

PISTEET

Käskyt CLS, SET, RESET ja POINT ovat tuttuja puoligrfiikkatasoista. Niiden vastineet hienografiikkatasossa ovat PCLS,

PSET, PRESET ja PPOINT, erotukseksi puoligrafiikkatilasta.

Niiden tehtävät ovat samat kuin aiemminkin:

PCLS tyhjentää hienografiikkaruudun, ja jos käskyä seuraa värikoodi, se säätää taustan halutunväriseksi. PSET sytyttää pisteen ja PRESET sammuttaa pisteen. PPOINT tarkistaa, onko piste valaistu vai sammuksissa. Seuraava esimerkki käy vuoro-tellen läpi kaikki käytettävissä olevat tilat ja värivalikoimat. Se sijoittaa mielivaltaisen värisiä pisteitä ruudulle. Pisteet ovat suorakulmaisella ruudukolla. Tyhjät kohdat ruudukossa johtuvat siitä, että satunnaisesti valittu väri on sama kuin taustaväri tai sitä ei ole värivalikoimassa.

```
10 FOR P=0 TO 4:Pmode P,1
20 FOR S=0 TO 1:SCREEN 1,S
30 PCLS:FOR I=50 TO 150 STEP 20
40 FOR J=50 TO 150 STEP 20
50 F=RND(8):PSET(I,J,F):NEXT J,I
60 FOR D=1 TO 1000:NEXT D,S,P
```

Katso tarkkaan pisteiden kokoa. Ne osoittavat kussakin tilassa käytettävissä olevan hienografiikkatason.

JANAN PIIRTÄMINEN

Nyt osaamme siis sijoittaa pisteitä ruudulle. LINE-lauseella voimme piirtää janan kahden pisteen välille. Poista edellisestä esimerkistä rivit 40 ja 50 ja muuta riviä 30 seuraavasti:

```
30 PCLS:LINE(10,180)-(245,10),PSET
```

Suorita ohjelma. Ruudun vasemmasta alakulmasta oikeaan yläkulmaan ilmestyy viiva. Lause käskää koneen piirtää viivan alkupisteestä (10,180) loppupisteeseen (245,10) kuvion värillä (PSET). Jos PSET-komento muutetaan PRESET-komennoksi, viiva piirretään taustavärillä. Taustavärillä piirrettyä viivaa ei voi nähdä. Sillä tavoin voidaan myös pyyhkiä pois aikaisemmin piirretty viiva. PSET ja PRESET kuuluvat olennaisesti LINE-lauseeseen eikä niillä ole tässä mitään tekemistä pisteiden syyttämisen ja samuttamisen kanssa.

LINE-lauseessa ei aina tarvitse määritellä viivan alkupistettä. Ellei alkukohtaa ole määritetty, viiva alkaa edellisen viivan loppupisteestä. (Ellei LINE-käskyä ole aikaisemmin käytetty ohjelmassa, edelliseksi pisteeksi valitaan 128,96, eli ruudun keskikohta). Lisää vielä yksi rivi esimerkkiin.

```
40 LINE-(130,180),PSET
```

Nyt piirretään viiva edellisestä loppupisteestä (245,10) ruudun alareunassa olevaan pisteeseen (130,180). Neliö tai suorakulmio voidaan piirtää neljällä viivalla, mutta LINE-komentoa voidaan myös laajentaa siten, että se huolehtii tästä. Käytä EDIT-tilaa ja lisää B riville 30, jonka muodon pitäisi nyt olla seuraava:

```
30 PCLS:LINE(10,180)-(245,10),PSET,B
```

Vinoviivan sijasta saadaan nyt suorakulmio. Kun halutaan piirtää suorakulmio, on vain määritettävä kahden vastakkaisen kulman sijainti ja lisättävä LINE-lauseeseen B. Palaa nyt EDIT-tilaan ja lisää F rvin 30 loppuun.

```
30 PCLS:LINE(10,180)-(245,10),PSET,BF
```

Rivin loppuun lisätty F tarkoittaa, että suorakulmio on väritettävä kuvion värillä. Käytämme tätä joustavaa komentoa nyt talon piirtämiseen.

Suorita ohjelma pienissä erissä, jotta voit nähdä, miten talo rakentuu. Ensiksi annetaan grafiikkataso ja piirretään seinä.

```
10 PMODE3,1:SCREEN1,0:PCLS
```

```
20 LINE(60,48)-(200,144),PSET,B
```

```
260 GOTO 260
```

Seuraavaksi lisäämme katon

```
40 LINE (60,48)-(130,20),PSET
```

```
50 LINE-(200,48),PSET
```

ja autotallin, jossa on ovi.

```
70 LINE (200,144)-(255,94),PSET,B
```

```
90 LINE (210,144)-(245,104),PSET,BF
```

Voimme käyttää samaa menetelmää ja piirtää talon oven.

```
100 LINE (160,144)-(188,105),PSET,BF
```

Ikkunaksi piirretään suorakulmio ja kaksi ristikkäistä viivaa.

```
110 LINE (85,132)-(135,108),PSET,B
```

```
120 LINE (110,108)-(110,132),PSET
```

```
130 LINE (85,120)-(135,120),PSET
```

Toisen kerroksen ikkunat piirretään samalla menetelmällä.

```
140 LINE (90,84)-(125,64),PSET,B
```


150 LINE (90,74)-(125,74),PSET
160 LINE (110,84)-(110,64),PSET
170 LINE (155,64)-(175,84),PSET,B
180 LINE (165,84)-(165,64),PSET
190 LINE (155,74)-(175,74),PSET

Talo viimeistellään piirtämällä vielä piippu.

200 LINE (150,40)-(160,15),PSET,BF

Tämä pieni ohjelma osoittaa, miten nopeasti voidaan piirtää kuva vain yhtä lausetyyppiä käyttäen. Tämä edellyttää tietenkin, että tiedetään, mihin viivat on piirrettävä. Pisteet löytyvät helpoimmin, jos otetaan kopio liitteessä B esitetystä graafisesta ruudukosta, Piirretään kuva siihen ja luetaan pisteet suoraan piirroksesta.

PCLEAR

PCLEAR-lausetta käytetään varattaessa graafisia sivuja hienografiikkatiloissa.

PCLEAR n

n on jokin luku yhden ja kahdeksan välillä. Ellei PCLEAR-lausetta anneta, oletusarvo on PCLEAR4.

Koska jokainen graafinen sivu vaatii 1536 merkkiä (1,5 kilotavua) muistitilaa, varataan vain tarvittava määrä sivuja.

PMODE

SCREEN

SCREEN-lauseella vaihdetaan graafinen ruutu tekstiruuduksi ja päinvastoin.

SCREEN tyyppi, väriasteikko

Tyyppi on joko 0, joka tarkoittaa tekstiruutua ja puoli-grafiikkaa, tai 1 hienografiikkaa käytettäessä.

Väriasteikko on joko 0 tai 1. Tekstiruudun väriasteikko on 0, musta vihreällä pohjalla, tai 1, musta oranssilla pohjalla. Grafiikassa väriasteikko riippuu käytettävästä työtilasta seuraavasti:

PMODE luku	Ruudukon koko	Pisteen koko	Käytettävä sivumäärä	Käytettävissä oleva väriasteikko	
				SCREEN 1,0	SCREEN 1,1
0	128x96	■ ■ ■ ■	1	Musta(0) Vihreä(1)	Musta(0) Valkoinen(5)
1	128x96	■ ■ ■ ■	2	Vihreä(1) Keltainen(2) Sininen(3) Punainen(4)	Valkoinen(5) Turkoosi(6) Violetti(7) Oranssi(8)
2	192x128	■ ■	2	Musta(0) Vihreä(1)	Musta(0) Valkoinen(5)
3	192x128	■ ■	4	Vihreä(1) Keltainen(2)	Valkoinen(5) Turkoosi(6)

PMODE	Ruudukon	Pisteen	Käytettävä	Käytettävissä oleva väriasteikko	
luku	koko	koko	sivumäärä	SCREEN 1,0	SCREEN 1,1
				Sininen(3)	Violetti(7)
				Punainen(4)	Oranssi(8)
4	256x192	■	4	Musta(0)	Musta(0)
				Vihreä(1)	Valkoinen(5)

COLOR

COLOR-lauseella muutetaan kuvion ja taustan standardiväriä hienografiikkatilassa.

COLOR kuvio, tausta

Sekä kuvio että tausta on jokin numero välillä 0-8, joka ilmoittaa halutun värin värikoodin. Molempien värien on kuuluttava käytössä olevan tilan väriasteikkoon.

PCLS

PCLS-lause on hienografiikkaveriso CLS-lauseesta. Sillä tyhjennetään ruutu ja annetaan ruudun taustaväri hienografiikkatilassa.

PCLS v

v on halutun taustavärin värikoodi. Sen on kuuluttava käytössä olevan työtilan väriasteikkoon. Ellei väri kuulu asteikkoon tai v on jätetty pois, käytetään standardiväriä. Ks. värikoodit lokerosta, jossa esiteltä CLS-komento.

PSET

Hienografiikkaversio SET-komennosta.

PSET (x,y,v)

valaisee pisteen (x,y) värillä v. x:n arvon on oltava alueella 0-255 ja y:n alueella 0-191. v on värikoodi 0-8 ja sen on kuuluttava väriasteikon väreihin.

PRESET

PRESET-lauseen hienografiikkamuoto.

PRESET (x,y)

sammuttaa pisteen (x,y). Tämä tarkoittaa, että piste muutetaan jälleen taustan väriksi. x:n on oltava alueella 0-255 ja y:n alueella 0-191.

VÄRITTÄMINEN

PAINT-käskyillä voidaan värittää minkä muotoinen kuvio tahansa millä värillä tahansa. On vain ilmoitettava mistä värittäminen aloitetaan ja mitä väriä käytetään. Lisäksi on ilmoitettava sen reunan väri, mihin värittäminen päättyy.

PAINT (x,y),a,b

Jossa x,y ovat aloituspisteen koordinaatit ja a ja b ovat väritykseen käytettävän värin ja rajoittavan värin koodit.

Lisää seuraava rivi taloesimerkkiin:

```
30 PAINT (90,90),2,4
```

Tämä tarkoittaa, että värittäminen aloitetaan pisteestä (90,90) ja väritetään keltaisella värillä (väri 2), kunnes vastaan tulee punainen (väri 4) reuna. Suorita ohjelma nähdäksesi, mitä tapahtuu. Poista nyt rivi 30 ja kirjoita sama rivi uudelleen riviksi 195 ja suorita ohjelma uudelleen.

```
195 PAINT (90,90),2,4
```

Huomaa, miten väritys nyt pysähtyy ikkunan reunoidin, joita ei aikaisemmin ollut kuvassa. Maalaa autotalli samalla tavalla.

```
80 PAINT (210,140),2,4
```

Seuraavaksi maalaamme katon. Jos PAINT-lauseesta jätetään pois joko väritykseen käytettävä väri tai reunan väri, kuvion nykyinen väri valitaan poisjätetyn värin tilalle.

Lopuksi väritämme taivaan.

```
210 LINE (0,64)-(60,64),PSET
```

220 LINE (200,64)-(255,64),PSET

230 PAINT (0,54),3,4

Nyt talo on hiukan iloisemman näköinen. Haluat ehkä vielä parannella sitä. Voit lisätä esimerkiksi polun ja aidan. Voit myös yrittää piirtää omia kuvia, värittää niitä ja keilla, mitä tapahtuu.

YMPYRÄT

Olemme piirtäneet viivoja, neliöitä, suorakulmioita, ja nyt piirrämme ympyrän. CIRCLE-lauseella voidaan piirtää ympyröitä, soikioita ja kaaria.

CIRCLE (x,y), säde, väri, kl_suhde, alkupiste, loppupiste
x,y on ympyrän keskipiste. Säde on ympyrän säde, joka on mitattu ruudun pisteinä. Väri on yksi käytössä olevan työtilan väriasteikon väreistä (jos se jätetään pois, väriksi valitaan kuvion väri). Muut parametrit liittyvät soikioiden ja kaarien piirtämiseen, ja käsittelemme niitä jäljempänä. Piirrämme ensin ympyröitä.

10 FOR P=0 TO 4:PMODE P,1

20 SCREEN 1,1:PCLS

30 FOR R=120 TO 10 STEP-10

40 CIRCLE (128,96),R:NEXT R

50 FOR D=1 TO 500:NEXT D,P

Tämä ohjelma piirtää ympyröitä, joiden keskipiste on ruudun keskellä. Ympyröitä on vaikea piirtää, ja jotta saataisiin hyvin tarkka ympyrä, on ehkä käytettävä PMODE4-tilaa.

Vaikka ympyrä meneekin ruudun ulkopuolelle, se ei aiheuta ongelmia. Jos yritetään piirtää viiva pisteeseen, joka ei ole ruudulla, viivaa ei piirretä lainkaan, varsinkaan hienografiikkatiloissa. Yritä lisätä ohjelmaan rivi:

```
42 LINE (128,96)-(300,40),PSET
```

ja katso, minkä tuloksen saat. PAINT-komentoa voidaan käyttää ympyrän värittämiseen:

```
45 PAINT (127,96)
```

Käyttämällä parametria kl_suhde voit muuttaa ympyrän ellipsiksi, soikioksi. Kl_suhde tarkoittaa korkeus-levyysuhdetta. Ympyräkomennossa leveys on aina sama: kaksi kertaa säde. Korkeutta voidaan säädellä kl-suhteella. Jos se on suurempi kuin 1, soikion korkeus on suurempi kuin leveys. Jos arvo on pienempi kuin 1, ympyrä litistyy toiseen suuntaan ja nyt leveys on suurempi kuin korkeus. Leveys x-akselilla (vaakasuorassa) on aina sama, ja vain korkeus y-akselilla (pystysuorassa) muuttuu. Kun kl-suhde on 0, ympyrästä tulee vaakasuora viiva. Jos suhde on kovin suuri ympyrä lähenee pystysuoraa viivaa (itse asiassa pitkää, kapeaa suorakulmiota). Suurin sallittu arvo on 255. Tee muutoksia edellisen esimerkin riveihin 30 ja 40 seuraavasti:

```
30 FOR H=0.5 TO 3 STEP 0.5
```

```
40 CIRCLE (128,96),40,,H:NEXT H
```

Huomaa ylimääräiset pilkut rivillä 40. Ne on merkitty, koska väri on jätetty pois.

CIRCLE-komennolla voidaan piirtää myös kaaria (ympyrän osia). Jotta tätä mahdollisuutta voitaisiin käyttää, on määritettävä kaaren alku- ja loppu. Sekä alku- että loppukohtaan on oltava jokin luku 0 ja 1 välillä. Ympyrän alkukohta on kello 3 kohdalla ajateltaessa kellotaulua. Piirtäminen tapahtuu myötäpäivään alkupisteestä lähtien. Jos alkukohta on esimerkiksi 0.25 ja loppukohta 0.75, piirretään ympyrän vasen puoli klo 6:sta klo 12:en. Jos alkukohta on 0.5 ja loppukohta 1.0, piirretään ympyrän yläpuolisko. Seuraavassa ohjelmassa käytetään kaaria kuvion piirtämiseen.

```
10 PMODE4,1:SCREEN1,1:COLOR0,5:PCLS
20 FOR R=15 TO 60 STEP 5
30 CIRCLE (128,96+R),R,,1,.5,1
40 CIRCLE (128,96-R),R,,1,0,.5
50 CIRCLE (128-R,96),R,,1,.75,.25
60 CIRCLE (128+R,96),R,,1,.25,.75
70 FOR D=1 TO 500:NEXT D
80 NEXT R
90 GOTO 90
```

SIVUJEN SELAAMINEN

Kuviin voidaan saada liikettä myös piirtämällä hiukan erilainen kuva jokaiselle sivulle ja "selaamalla" sitten sivuja nopeasti. Sivujen lukumäärä määritettiin PCLEAR-komennolla ja PMODE-lauseen toinen parametri määrittelee sivun, johon kirjoitetaan. Tietenkin on myös muistettava, mikä työtila on käytössä. PMODE3- ja PMODE4-tiloja käytettäessä yksi

graafinen ruutu vaatii 4 sivua, joten itse asiassa voidaan selata vain sivuja 1 ja 5. PMODE1- ja PMODE2-tiloissa, jotka tarvitsevat 2 sivua, selattaisiin sivuja 1, 3, 5 ja 7. Seuraava esimerkki osoittaa, miten tämä tapahtuu. Kokeile ohjelmaa kaikilla PMODE-arvoilla.

```
10 PCLEAR 8:PCLS
20 INPUT "MODE";M:ON M GOTO 40,40,50,50
30 S=1:GOTO 60
40 S=2:GOTO 60
50 S=4
60 FOR P=1 TO 8 STEP S:P:MODE M,P:PCLS
70 LINE (128,0)-(128,(P-1)*15),PSET
80 SCREEN1,1:FOR I=1 TO 1000:NEXT I,P
90 FOR P=1 TO 8 STEP S:GOSUB 150:NEXT P
100 IF M>2 THEN D=4:S1=3 ELSE D=7:S1=S
110 FOR P=D TO 1 STEP -S1:GOSUB 150:NEXT P
120 GOTO 90
150 P:MODE M,P:SCREEN 1,1
160 FOR T=1 TO 20:NEXT T:RETURN
```

Rivit 60 ja 70 piirtävät muuttuvan kuvan eri sivuille. Kaikki piirtäminen tapahtuu näkymättömissä, koska toistaiseksi ei ole vielä annettu yhtään SCREEN-komentoa. Ohjelman loppuosa näyttää jokaisen sivun vuorotellen selaten niitä ensin eteen- ja sitten taaksepäin, mikä antaa vaikutelman liikkeestä. Voit havaita, että mitä useampia sivuja käytetään, sitä tasaisempi liike on.

Kuvia voidaan piirtää myös PCOPY-komennolla.

PCOPY lähtösivu TO tulosivu

Voit kopioida minkä tahansa sivun sisällön mille tahansa toiselle sivulle, mikäli sivu on aikaisemmin varattu PCLEAR-käskyllä. PCOPY-komentoa voidaan käyttää myös kaksoiskappaleiden tekemiseen PMODE3- ja PMODE4-sivuista. Seuraava ohjelma osoittaa, miten PCOPY-komentoa käytetään tähän tarkoitukseen. Huomaa, että kuvan sijoittamisessa on noudatettava erityistä huolellisuutta:

```
10 PCLEAR 8:PMODE3,4:PCLS
20 LINE (100,20)-(140,40),PSET ,BF
30 CIRCLE (50,25),20
40 CIRCLE (200,50),20
50 FOR D=3 TO 1 STEP-1
60 PCOPY 4 TO D:NEXT D
70 FOR P=4 TO 1 STEP-1:PMODE3,P
80 SCREEN1,1:FOR I=1 TO 1000:NEXT I,P
90 GOTO 90
```

PMODE3- ja 4-tiloissa ruutu muodostuu neljästä sivusta. Sivun 1 on ruudun ylin neljännes, sivun 2 seuraava neljännes jne. Kopioimalla sivun 4 sisällön sivulle 1 olet kopioinut näyttöruudun ylimmän neljänneksen alimpaan neljännekseen. PMODE1- ja PMODE2-tiloissa voidaan saavuttaa sama vaikutus, mutta tässä tapauksessa ruutu on jaettu kahteen eikä neljään osaan.

Lisää grafiikkaa esitellään kappaleessa 10.

LINE

LINE-lausetta käytetään piirrettäessä viivoja ja suorakulmioita hienografiikkatiloissa.

LINE (x1,y1)-(x2,y2),a,b

x1,y1 ovat viivan alkupisteen koordinaatit.

x2,y2 ovat viivan loppupisteen koordinaatit.

a on joko PSET tai PRESET. Jos käytetään PSET-käskyä, viiva piirretään asianomaisella kuvion värillä, jos käytetään PRESET-käskyä, viiva piirretään taustan värillä.

b on vapaavalintainen parametri. Jos sitä käytetään, se on joko B tai BF. Jos annetaan B, viivan sijasta piirretään suorakulmio, jonka yläkulma on pisteessä x1,y1 ja oikea alakulma pisteessä x2,y2. Jos annetaan BF, suorakulmio piirretään ja väritetään kuvion värillä.

```
10 PMODE4,1:SCREEN1,1:PCLS 5:COLOR 0,5
```

```
20 FOR I=1 TO 1000
```

```
30 X=X+L*SIN(R):Y=Y+L*COS(R)
```

```
40 IF X<-128 OR X>128 THEN 90
```

```
50 IF Y<-96 OR Y>95 THEN 90
```

```
60 LINE- (X+128,Y+96),PSET
```

```
70 R1=R1+60:R=R1/57.29578:L=L+0.5
```

```
80 NEXT I
```

```
90 GOTO 90
```

PAINT

PAINT-lausetta käytetään hienografiikkatilassa haluttaessa värittää jokin kuvio jollakin määrättyllä värillä.

PAINT(x,y),v,r

x,y ovat sen pisteen koordinaatit, josta värittäminen aloitetaan.

v on värittämiseen käytettävän värin koodinnumero. Värikoodin on oltava jokin luku välillä 0-8, ja sen on kuuluttava käytössä olevan tilan väriasteikkoon. Jos se jätetään pois, värittämiseen käytetään kuvion väriä.

r on reunavärin värikoodi, joka osoittaa, mihin värittäminen lopetetaan. Myös sen on oltava värikoodi välillä 0-8. On huomattava, että värittäminen jatkuu sellaisen reunan yli, joka on jotakin muuta väriä kuin annettu reunaväri r. Jos r jätetään pois, käytetään kuvion väriä.

Ks. CIRCLE-lausetta esittelevästä lokerosta esimerkki PAINT-lauseen käytöstä.

CIRCLE

CIRCLE-lauseella piirretään ympyröitä, soikioita ja kaaria. Sitä voidaan käyttää vain hienografiikkatilassa.

CIRCLE (x,y), säde, väri, kl-suhde, alkupiste, loppupiste

x on ympyrän keskipisteen x-koordinaatti (0-255)

y on ympyrän keskipisteen y-koordinaatti (0-191)

säde on ympyrän säde mitattuna ruudun pisteinä.

väri on värikoodi (0-8). Sen on kuuluttava käytettävissä olevaan väriasteikkoon. Ellei sitä mainita, käytetään kuvion väriä.

kl-suhde on korkeus-leveyssuhde (0-255). Sitä käytetään piirrettäessä soikioita. Jos kl-suhde jätetään pois, käytetään arvoa 1.

alkupiste on kaaren aloituskohta(0-1). 0-kohta on kello 3 kohdalla. Ellei parametriä anneta, oletusarvo on 0.

loppupiste on kaaren loppupiste(0-1). Piirtäminen tapahtuu myötäpäivään alkupisteestä lähtien. 0.5-kohta on kello 9 kohdalla. Jos paramteri jätetään pois, oletusarvona käytetään arvoa 1.

10 PMODE3,1:SCREEN1,0:PCLS

20 CIRCLE (180,156),28,3:PAINT(180,156),3,3

30 CIRCLE (110,156),28,3:PAINT (110,156),3,3

40 CIRCLE (144,80),68,4,1,0,0.5

50 LINE (212,80)-(76,80),PSET:LINE-(48,32),PSET

60 PAINT (144,82):CIRCLE (144,80),70,4,-8,-.79,1

70 LINE (160,80)-(160,28),PSET:PAINT(210,75)

80 GOTO 80

PCOPY

PCOPY on hienografiikkalause, jota käytetään kopioitaessa jonkin grafiikkasivun sisältö toiselle grafiikkasivulle.

PCOPY lähtösivu TO tulosivu

Lähtösivun ja tulosivun ovat molemmat jokin luku välillä 1-8, ja ne viittaavat sivuihin, jotka on aikaisemmin varattu PCLEAR-komennolla. Yhden näyttöruudun alan vaatima muistitila vaihtelee hienografiikkatilasta riippuen. Tämä on syytä ottaa huomioon PCOPY-lausetta käytettäessä.

PCOPY 3 TO 5

9. ÄÄNI

ÄÄNEN AIKAANSAAMINEN

Graafiset ja myös muut tietokoneohjelmat voidaan usein tehdä mielenkiintoisemmiksi lisäämällä niihin ääni. Olemme jo käyttäneet SOUND-komentoa, mm. kappaleen 7 esimerkissä. Ääni voidaan lisätä ohjelmaan yksinkertaisemmallakin tavalla. Tietokone käyttää kasettinauhuria ohjelmien tallettamiseen, ja se voi myös pyörittää kasettinauhaa tarvittaessa. Komento MOTOR ON ja MOTOR OFF ohjaavat moottoria. AUDIO ON ja AUDIO OFF-komennot kytkevät kasetilta tulevan äänen television kovaääniseen ja pois. Lisäämällä nämä käskyt ohjelmaan voit siis esimerkiksi liittää taustamusiikkia piirrosohjelmiisi. Tai voit käyttää opetusohjelmakasettia, jossa annetaan ohjeita ja esitetään kysymyksiä. Seuraava esimerkki osoittaa, miten helppoa tämä on. Ellei sinulla ole sopivaa kasettia, käytä jotakin ohjelmakasettiasi. Kuulemasi oudot äänet ovat tietokoneiden 'keskustelua'.

```
10 CLS:PRINT@ 135,"PAINA VÄLILYÖNTINÄPPÄINTÄ"  
20 PRINT@ 197,"HALUTESSASI KÄYNNISTÄÄ TAI PYSÄYTTÄÄ"  
25 PRINT@ 229,"NAUHURIN"  
30 A$=INKEY:IF A$<>"V" THEN 30  
40 IF F=0 THEN MOTOR ON:AUDIO ON:F=1 ELSE MOTOR OFF:  
    AUDIO OFF:F=0  
50 GOTO 30
```

Kelaa kasettinauha alkuun, paina PLAY-näppäintä ja suorita ohjelma. Voit keskeyttää kasetin kelaamisen painamalla välilyöntinäppäintä.

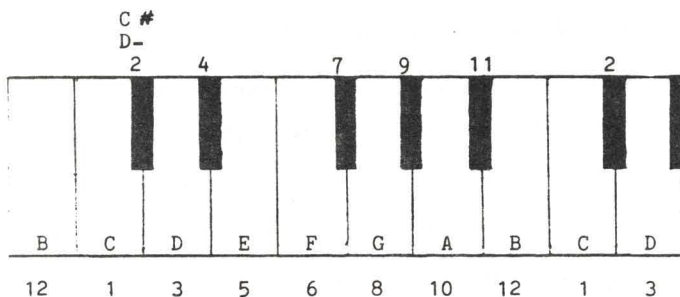
Tätä menetelmää voidaan käyttää laadittaessa opetusohjelmia, joihin liittyy kysymyksiä ja vastauksia. Voit myös liittää omiin 'animaatioelokuviisi' musiikki- ja äänitehosteita.

SOITTAMINEN

Tietokone voidaan myös ohjelmoida soittamaan melodioita. PLAY-komento muuttaa merkkijonon sisällön ääniksi.

PLAY musiikkimerkkijono

Musiikkimerkkijono voi olla merkkijonovakio, merkkijonolauseke tai merkkijonomuuttuja. Se ei ole mikä tahansa vanha merkkijono, vaan musiikkimerkkijono, johon sisältyy nuotti, oktaavi, nuotin_pituus, tempo ja tauot. Nuotti on luonnollisesti se nuotti, joka halutaan soittaa. Helpoimmin tämä voidaan tehdä antamalla kirjain, joka vastaa tavallisia englantilaisia musiikkinuotteja A, B, C, D, E, F, G. Suomalainen nuotti H on siis englannissa B. On muistettava käyttää B:tä, sillä Basic ei hyväksy H-kirjainta nuotiksi. Puoli sävelaskelta korkeampi nuotti merkitään# -merkillä tai +-merkillä. Puoli sävelaskelta matalampi nuotti merkitään-. Huomaa, ettei tietokone tunnista B# eikä C-, koska ne eivät ole 12 sävelen asteikolla. Nuotti voidaan antaa myös käyttämällä numeroa, joka vastaa sen paikkaa 12 sävelen asteikolla.



Nuotit ja niitä vastaavat numerot on merkitty kuvassa pianon koskettimistolle.

PLAY-komentoa voidaan käyttää suorana komentona, joten musiikkijonot voidaan tarkistaa ennenkuin ne lisätään ohjelmaan. Aloitamme harjoittelemalla asteikolla.

PLAY "CDEFGABCCBAGFEDC" (Asteikko C)

PLAY "GABCDEF#GGF#EDCBAG" (Asteikko G)

Ylläoleva asteikko C-duurissa on melkein oikein, mutta G-duurissa oleva asteikko on pelkkä kaaos. Tämä johtuu siitä, että asteikot liikkuvat eri oktaaveissa ja tietokoneelle on kerrottava se. Oktaavi valitaan kirjoittamalla 0 ja sen perään jokin luku 1 ja 5 välillä. 03, jossa on keski-C, valikoituu automaattisesti, kun tietokone kytketään päälle. Tämä oktaavin arvo on käytössä, kunnes uusi arvo annetaan. Tästä syystä haluttu oktaavi on parasta määritellä jo alussa. Kokeillaanpa asteikkoa uudelleen.

PLAY "03CDEFGAB04CC03BAGFEDC"

PLAY "03GAB04CDEF#GG#EDCO3BAG"

C-asteikko voidaan soittaa myös käyttämällä numeroita kirjaimien sijasta:

```
PLAY "03;1;3;5;6;8;10;12;04;1;1;03;12;10;8;6;5;3;1"
```

Huomaa merkkijonossa käytetty välimerkki (;). Puolipistettä voidaan käyttää aina haluttaessa, mutta numeroiden yhteydessä se on välttämätön, jotta ei syntyisi sekaannuksia.

Koska musiikkijono on merkkijono, siihen voidaan soveltaa kaikkia merkkijonotoimituksia. Seuraava esimerkki soittaa C-asteikon PLAY-komennon koko alueella.

```
10 A$="CDEFGAB":FOR I=1 TO 5
20 B$="O"+STR$(I)+A$
30 PRINT B$:PLAY B$:NEXT I
```

Käyttämällä samaa menetelmää ja numeroita kirjaimien sijasta voimme soittaa koko kromaattisen asteikon PLAY-komennolla.

```
10 FOR I=1 TO 5:A$="O"+STR$(I)+";"
20 FOR J=1 TO 12:PLAY A$+STR$(J):NEXT J,I
```

Nuotit ovat sävelmissä harvoin samanpituisia, joten meidän on annettava myös nuottien pituudet. Ne annetaan musiikki-merkkijonon parametrilla nuotin_pituus (L). L-kirjaimen jälkeen annetaan jokin luku alueelta 1-255. Luvun suuretessa nuotti lyhenee. L1 on siten kokonuotti, L2 on puolinuotti, L4 neljäsosanuotti jne. 1/255 nuotti on mahdollinen, mutta harvat säveltäjät käyttävät sitä. Piste tarkoittaa, että nuotin kestoja pidennetään puolella sen normaalikestosta. PLAY-komennolla tämä saavutetaan lisäämällä piste (tai niin monta pistettä kuin halutaan) L-parametriin liittyvän luvun perään.

L4.=1/4+1/8=3/8-nuotti.

Nyt osaamme jo niin paljon, että voimme soittaa yksinkertaisen melodian:

```
5 CLEAR 500
```

```
10 A$="O2L4GG;L2GDL4BB;L2BGL4GB;
```

```
    O3L2DDL4CO2B;L1AL4AB;
```

```
    O3L2CCO2L4BA;L2BGL4GB;
```

```
    L2ADL4F#A;L1G;"
```

```
20 B$=A$+A$:PLAY B$
```

Välimerkkejä käytetään tässä (;) tahtimerkkeinä eivätkä ne ole aivan välttämättömiä. Saatoit tunnistaa melodian Clementine, mutta se soitettiin aivan liian hitaasti. Tämä vika korjataan tempo-parametrillä. Annetaan kirjain T ja sen jälkeen jokin luku alueelta 1-255. Mitä korkeampi luku on sitä nopeammin melodia soitetaan. Muuta riviä 20 seuraavasti:

```
20 B$=A$+A$:PLAY "T6" +B$
```

Muuttele T:n arvoa ja yritä löytää mielestäsi sävelmään sopiva tempo.

Useimpiin sävelmiin on lisättävä taukoja säkeiden välille, ja myös äänen voimakkuutta on voitava muuttaa joissakin juoksu-
sutuksissa. Tauko-parametri merkitään kirjaimella P, jonka jälkeen annetaan luku. Se noudattaa samaa kaavaa kuin parametri nuotin pituus (L), mutta luvun perässä ei voida käyttää pistettä. Jos halutaan lisätä tauko, jonka pituus vastaa nuotin pituutta L4, on annettava P4P8. Parametrillä ään
en voimakkuus voidaan säädellä äänen voimakkuutta anta-

malla kirjain V ja sen jälkeen jokin luku alueelta 0-31.
Mitä suurempi luku annetaan, sitä kovempaa sävel soitetaan.
Seuraavassa esimerkissä äänen voimakkuudeksi määrätään
crescendo.

```
10 A$="V1002L4GG;L1GP4V14L4GGG;  
    L1GP4V18L4GGG;L2BL4BBBV22L2BL4BBB;  
    V2603L2DL4DDDL2DL4DDD;  
    V30L1GL2.F#L4C#;L2EDCO2A;  
    L1GL2AL4.DL8A;L2B"
```

```
20 PLAY "T5"+A$
```

Sävelmissä on usein sävelkulkuja, jotka toistuvat eri kohdissa. Sen sijaan, että sama sävelkulku näppäiltäisiin ohjelmaan useita kertoja, se voidaan tallentaa merkkijonomuuttujaan. Antamalla kirjain X ja sen jälkeen merkkijonomuuttujan nimi tällaista alimerkkijonoa voidaan käyttää osana tavallisen musiikkikomennon osaa. X:n jälkeen on annettava merkkijonon nimi ja puolipiste seuraavasti:

```
10 X$="03L2GB04C;DL4C03BAG;"  
20 Y$="L2ADD;L1.A"  
30 Z$="L2ADD;L1.C"  
40 A$="XX$;XY$;XX$;XZ$;L2BGG;04C03L4  
    BAGF#;XY$;XX$;XZ$;"  
50 PLAY "T8"+A$
```

Olisimme voineet käyttää alimerkkijonoa Clementine-esimerkissä kirjoittamalla rivin 20 muotoon:

```
20 PLAY"T6XA$;XA$;"
```

Dollarimerkin (\$) jälkeen on aina lisättävä puolipiste (;). Puolipiste on välttämätön ainoastaan käytettäessä alimerkkijonoja tai lukuja nuottien merkitsemiseen (kirjaimien sijasta).

Äänenvoimakkuuden (V), oktaavin (O), tempon (T) ja nuotin pituuden (L) kanssa voidaan käyttää myös eräitä muita merkkejä. Kirjaimen perään liitettävien lukujen sijasta voidaan käyttää seuraavia "päätteitä" (perään liitettäviä merkkejä):

- + Lisää tämänhetkistä arvoa yhdellä
- Vähentää tämänhetkisestä arvosta yhden
- > Kertoo tämänhetkisen arvon kahdella
- < Jakaa tämänhetkisen arvon kahdella

Ylläoleva asteikkoesimerkki voidaan nyt kirjoittaa uudelleen näiden uusien merkkien avulla seuraavasti:

```
10 PLAY "O1C":FOR I=1 TO 4:PLAY "DEFGABO+C":NEXT I
```

Melodioita voidaan löytää yksinkertaisille instrumenteille, kuten huilulle ja trumpetillem, kirjoitetuista nuoteista.

Vaikka et haluaisikaan käyttää tietokonetta sävelmien soittamiseen, tutustu silti PLAY-komentoon. Sen avulla voidaan liittää ääniefektejä peleihin. Kokeile jotakin tämän kappaleen esimerkkiä antamalla tempo-parametriksi T255.

Viimeisessä esimerkissä käytetään PLAY-komennon avulla modernia teknologiaa 400 vuotta vanhan sävelmän soittamiseen.

```
10 A$="O3L2E;L1GL2AL2.BL4O+C#L2.O-B;  
L1AL2F#L2.DL4EL2F#;L1GL2EL2.EL4DL2E;
```

L1F# V10L2DV8L10-BV6L20+E;L1GL2AL2.B
L40+C# L20-B;L1AL2F# L2.DL4EL2F# ;
L2.GL4F# L2EV8L2.D# V10L4C# V15L2D# ;
L1.EL1EP1;"
20 B\$="04L1.DL2DL4C# O-L2B;L1AL2F# L2.D
L4EL2F# ;L1GL2EL2.EL4DL2E;L1F# L2D
O-L1B0+L2B;O+L1DL2DL2.DL4C# O-L2B;
L1AL2F# L2.DL4EL2F# ;L2.GV10L4F# L2
EV6L2.D# L4C# V4L2D# ;V15L1.EL2EP1;"
30 PLAY "T10XA\$;XB\$;XA\$;XB\$;"

AUDIO

AUDIO-lauseella kytketään yhteys kasettisoittimesta tulevien äänisignaalien ja television kovaäänisen välille. AUDIO ON ohjaa kasettisoittimesta lähtevät signaalit televisiovastaanottimeen. AUDIO OFF katkaisee yhteyden.

MOTOR

MOTOR-lauseella ohjailaan kasettisoittimen moottoria. MOTOR ON käynnistää moottorin ja MOTOR OFF pysäyttää sen.

Tämä komento toimii vain, jos kasettisoittimen PLAY-näppäin on painettu alas.

PLAY

PLAY-lausetta käytetään musiikin tuottamiseen. Argumentti on merkkijonolauseke, merkkijonovakio tai merkkijonomuuttuja. Lauseen muoto on:

PLAY musiikkimerkkijono

jossa musiikkimerkkijono on parametriluettelo, johon kuuluvat seuraavat alkiot:

<u>nuotti</u>	kirjain alueelta 'A'-'G' tai luku alueelta 1-12. Suomalainen nuotti H on B, koska tämä on englanninkielisissä maissa yleinen käytäntö.
<u>oktaavi</u>	'0' ja jokin luku alueelta 1-5. Standardiarvo on 03. Kone käyttää automaattisesti standardiarvoa ellei muuta ilmoiteta.
<u>nuotin_pituus</u>	'L' ja sen jälkeen jokin luku alueelta 1-255. Standardiarvo on L4.
<u>tempo</u>	'T' ja jokin luku alueelta 1-255. Standardiarvo on T2.
<u>äänen_voimakkuus</u>	'V' ja jokin luku alueelta 1-31. Standardiarvo on V15.
<u>tauon_pituus</u>	'P' ja jokin luku alueelta 1-255.
<u>alimerkkijonojen_suuritus</u>	'X' ja merkkijonomuuttuja sekä puolipiste.

Puoli sävelaskelta korkeampi nuotti merkitään '+' tai '#'.
Puoli sävelaskelta matalampi nuotti merkitään '-'. Molemmat merkit on liitettävä sävelen perään.

Parametriä nuotin pituus voidaan muuttaa lisäämällä luvun perään piste (.). (Esim. L2.) on pisteellä merkitty nuotti.

Oktaavia, äänen voimakkuutta, tempoa ja nuotin pituutta voidaan kaikkia muuttaa liittämällä niiden perään seuraavia päätteitä:

- + lisää tämänhetkistä arvoa yhdellä
- vähentää tämänhetkisestä arvosta yhden
- > kertoo tämänhetkisen arvon kahdella
- < jakaa tämänhetkisen arvon kahdella

10 X\$="O3L4EF #L4.EL8AAG #ABL4O+C #O-B"

20 A\$="XX\$;O4C #O-AF# O+DC#O-BL2AXX\$;

O+C #DEL8DO-BL<AG #L<AL4.BL8O+C #L4

DO-BL4.O+C #L8DL<EC #L4.EL8EEEEEL1

EL4.EL8DC #EDO-BL<AG #L<A"

30 PLAY "T2V20"+X\$+A\$

10. LISÄÄ GRAFIIKKAA

Kappaleessa 8 osoitimme, miten LINE- ja CIRCLE-lauseiden avulla voidaan saada aikaan säännöllisiä muotoja, kuten suorakulmioita, ympyröitä, soikioita ja kaaria. Vaikka nämä käskyt ovatkin erittäin käytännöllisiä, saattaisi vaatia huomattavaa kekseliäisyyttä, jotta niiden avulla onnistuttaisiin tuottamaan epäsäännöllisiä muotoja. Tällaisia muotoja on helpoin käsitellä piirtämällä ne.

PIIRTÄMINEN

Paperille piirrettäessä aloitetaan aina jostakin määrätystä kohdasta ja kynää siirretään tietty matka johonkin suuntaan. DRAW-lauseen avulla sama voidaan tehdä ruudulla. Muoto on:

DRAW merkkijono

jolloin merkkijono on joko merkkijonovakio tai merkkijonomuuttuja, johon sisältyy piirtämisohteita. Menettely on hyvin samankaltainen kuin edellisen kappaleen PLAY-lauseessa.

Piirtäminen aloitetaan siirtymällä aloituskohtaan.

Mx,y käskee siirtymään x- ja y-koordinaattien ilmoittamaan pisteeseen. Esimerkiksi piirtämisohte M128,96 käskee siirtymään ruudun keskelle. Siirtyminen johonkin pisteeseen on yleensä parasta tehdä tyhjäsirrtona, jolloin piirtämisen jälki ei tule näkyviin, siis ikäänkuin kynä nostettaisiin irti paperista. Ellei näin tehdä, piirroksen voi jäädä ylimääräisiä viivoja. Tyhjäsirto saadaan aikaan B-kirjaimella. B-kirjainta seuraavat piirtämisohteet piirtävät aina tyhjän

viivan. BM128,96 käskee siirtymään ruudun keskelle viivaa piirtämättä.

Kun aloituskohta on näin määritetty, voidaan liikkua ylöspäin (U=up), alaspäin (D=down), oikealle (R=right) tai vasemmalle (L=left) niin monen pisteen matka kuin halutaan. Sarja U20R20D20L20 piirtää ensin 20 pisteen pituisen viivan ylöspäin, sitten 20 pistettä oikealle, 20 pistettä alaspäin ja 20 pistettä vasemmalle. Se piirtää siis neliön. Seuraavassa on esimerkki:

```
10 PMODE3,1:PCLS:SCREEN1,1
20 DRAW "BM120,96;U26;R13;D26;L13"
80 GOTO 80
```

Merkkijonossa käytetään puolipistettä välimerkkinä. Se ei ole aivan välttämätön, mutta se helpottaa merkkijonojen lukemista. Esimerkkiohjelma piirtää suorakulmion lähelle ruudun keskustaa.

Pysty- ja vaakasuorien viivojen lisäksi voidaan piirtää myös vinoviivoja. Niitä piirrettäessä käytetään alikomentoja E, F, G ja H siten, että esimerkiksi E12 piirtää vinoviivan, joka on 12 pisteen pituinen ja 45 asteen kulmassa pystyakseliin nähden. Kaikki kulmat mitataan suhteessa pystyakseliin seuraavasti:

E 45 astetta.	F 135 astetta
G 225 astetta	H 315 astetta

Näin voidaan piirtää vinoviivoja kaikkiin 4 suuntaan. Lisää esimerkkiin rivi:

```
40 DRAW "L6;U6;E6;BR13;F6;D6;L6;BU26;H6;G6"
```

ja suorakulmiosta tulee raketti. Kone muistaa lopetuskohdan, joten rivi 40 jatkaa piirtämistä tästä viimeisestä pisteestä. Tutki rivin 40 merkkijonoa ja pyri selvittämään, miten tämä tapahtuu. Piirtäminen lopetettiin suorakulmion vasempaan alakulmaan, ja BR13 käskee siirtymään 13 pistettä oikealle piirtämättä.

Raketti piirretään standardilla kuvion värillä, mutta väriä voidaan haluttaessa muuttaa käyttämällä kirjainta C ja sen jälkeen värikoodia 0-8. Käytä edit-tilaa ja muuta rivi 40 seuraavasti:

```
40 DRAW "C7;L6;U6;E6;BR13;F6;D6;L6;BU26;H6;G6"
```

Raketti on nyt kaksivärinen. Piirros voidaan värittää samalla tavoin kuin muutkin kuviot, mutta C-parametri muuttaa standardin kuvion värin, joten on varottava värittämästä kaikkea.

Kuva on aika pieni, joten suurennamme sitä S-parametriä käyttäen. Sen avulla piirrosta tai piirroksen osia voidaan suurentaa tai pienentää 1/4-asteikolla. S1 pienentää siten kuvan 1/4 suuruiseksi, S2 2/4 suuruiseksi (puoleen), S8 muuttaa sen 8/4 suuruiseksi (kaksinkertaiseksi) jne. Standardimittakaava on 4/4, siis kuvion alkuperäinen koko. S-parametrin perään voidaan liittää mikä tahansa luku väliltä 1-62.

Lisää esimerkkiin rivi

```
15 DRAW "S12"
```

ja raketti suurenee kolminkertaiseksi alkuperäiseen kokoonsa verrattuna..

Lisäksi voidaan käyttää kulmaparametriä A. Sen avulla voidaan pyörittää koko piirrosta tai sen osia, sillä kaikki A:ta seuraavat rivit piirretään siten, että niissä on An-parametrin ilmoittama siirtymä. Tällöin n on jokin luku väliltä 0-3 seuraavasti:

0 0 astetta	1 90 astetta
2 180 astetta	3 270 astetta

Laajenna ohjelmaa lisäämällä seuraavat rivit:

```
18 FOR I=0 TO 3:DRAW "A"+STR$(I):PCLS
50 FOR D=1 TO 100:NEXT D,I
```

Nyt raketti kääntyy, mutta se muuttaa myös väriä. Tämä johtuu siitä, etää tietokone muistaa viimeisen piirtämänsä kohdan lisäksi myös viimeiset C:n ja A:n arvot. Tämä ongelma voidaan ratkaista antamalla C8 rivin 20 merkkijonon alussa.

Rivi 20 osoittaa, että kuten PLAY-lauseessakin myös DRAW-lauseessa käytettäviä merkkijonoja voidaan käyttää merkkijonofunktioiden yhteydessä. Vastaavalla tavalla kuin PLAY-lauseissa, alimerkkijonoja voidaan suorittaa X-komennolla, jota seuraa merkkijonomuuttuja (XA\$) ja puolipiste (;).

```
10 PMODE3,1:SCREEN1,1:PCLS
20 S$="L8E4F4"
30 D$="A0;XS$;A1;XS$;A2;XS$;A3;XS$;"
40 DRAW "S24"+D$
50 GOTO 50
```

Kolmio tallennetaan alimerkkijonoon S\$, jota käytetään rivillä 30 kuvion laatimisessa. Puolipiste on välttämätön ainoastaan dollarimerkin (\$) jälkeen.

Viimeinen parametri on N, joka tarkoittaa, ettei piirtämis-
kohtaa pidetä muistissa. Määriteltä viiva piirretään mutta
sen loppupistettä ei käytetä uutena alkupisteenä. NU10L5
piirtää 10 pistettä pitkän viivan ylöspäin, palaa sitten läh-
töpisteeseen ja piirtää 5 pisteen pituisen viivan oikealle
(siis L-muoto).

```
10 PMODE3,1:SCREEN1,1:PCLS
```

```
20 DRAW "BM128,96;NU25ND25NR25NL25;NE17NF17NG17NH17"
```

```
30 GOTO 30
```

Ylläoleva esimerkki piirtää viivoja ruudun keskipisteestä
alkaen ja palaa aina keskipisteeseen aloittaessaan uuden
viivan.

Usein halutaan piirtää toinen kuvio edellisen viereen. Uu-
den kuvion sijainti suhteessa aikaisempaan kuvioon tiedetään,
mutta ei haluta selvittää koordinaatteja. Ongelma voidaan
ratkaista suhteellisen siirtymän avulla, esimerkiksi 5 pistet-
tä oikealle ja 10 pistettä ylöspäin. Liikeparametriä (M)
käyttämällä tarvitsee vain määritellä etäisyys tämänhetkises-
tä sijaintikohdasta joko plus tai miinuspuolella, esimerkiksi
M+5,-10. Muista käyttää parametriä B tarpeettomien viivojen
välttämiseksi.

```
25 DRAW "BM-25,-25,U10R25D10L25"
```

Lisää tämä rivi edelliseen esimerkkiin, ja edellisen kuvion
yläpuolelle piirtey suorakulmio. Viimeinen kohta oli 128,96,
koska käytettiin N-parametriä. Nyt on liikuttu 25 pistettä
vasemmalle ja 25 pistettä ylös (muista, että y=0 on ruudun
ylälaidassa), ja suorakulmion piirtäminen aloitetaan pistees-
tä 103,71.

DRAW-lauseella piirrettävät kuviot voidaan yhdistää LINE ja CIRCLE-lauseilla piirrettäviin muotoihin, mutta on muistettava, että kaikki myöhemmät mittakaavaa (S), väriä (C) tai kulmaa (A) koskevat muutokset vaikuttavat vain kuvan DRAW-osaan.

PSET ja PRESET- sekä PAINT-komennoilla voidaan lisätä kuvi-oon yksityiskohtia ja väriä. Etenkin PAINT-käskyä käytet- täessä on kuitenkin syytä olla varovainen, sillä jos kuvan DRAW-osaan on tehty muutoksia, väri saattaa tulla väärään kohtaan.

KUVAN SIIRTELY

Kun olet laatinut piirroksen LINE-, CIRCLE- ja DRAW-käskyjen avulla, haluat ehkä siirrellä sitä ruudulla. Voit tietenkin pyyhkiä kuvan pois ja piirtää sen uudelleen toiseen paikkaan samalla tavoin kuin aikaisemminkin. Tämä vie kuitenkin aikaa, ellei kuva ole aivan yksinkertainen. Se ratkaistaan kahdella uudella käskyllä. Kuvasta tarvitsee vain ottaa (GET) kopio, joka sijoitetaan (PU) toiseen paikkaan. GET-lauseella voi- daan kopioida suorakulmion muotoinen alue ruudulta taulukkoon, joka voidaan myöhemmin sijoittaa takaisin ruudulle.

GET (x1,y1)-(x2,y2), taulukkonimi, G

x1,y1 ja x2,y2 ovat sen suorakulmion muotoisen alueen vasem- man yläkulman ja oikean alakulman koordinaatit jonka sisään tallennettava kuvio jää. Taulukkonimi on sen aikaisemmin mi- toitetun taulukon nimi, johon kuva tallennetaan. Jos olet unohtanut, miten taulukoita tallennetaan, kertaa kappaleen 6

alku. Taulukon koon on vastattava ruudulta tallennettavan suorakulmion kokoa. Taulukon ensimmäinen dimensio on suorakulmion leveys (x_2-x_1) ja toinen suorakulmion pituus (y_2-y_1). Viimeinen parametri G on valinnainen ja määrittää tallennettavien yksityiskohtien määrän. G-parametri on välttämätön tiloissa PMODE0, 1 ja 3, muutoin PUT-käskyllä aikaansaattava vaakasuora liike saattaa jäädä epätarkaksi.

Osoitamme nyt rakettipiirroksen avulla, kuinka tämä tapahtuu. Ensiksi on laskettava tarvittavan taulukon koko. Piirros alkaa kohdasta (120,96), vasen siiveke ulottuu 6 pistettä yli raketin rungon leveyden, runko ja oikea siiveke ovat 13+6, joten piirroksen leveys on 25 pistettä: kohdasta (114, 96) kohtaan (139,96). Korkeus on 26 pistettä plus kärki, joka on 5 pistettä, joten korkeus on siis yhteensä 31 pistettä. Lisätään vielä muutama piste joka suuntaan ja tehdään siitä 30 x 40 suorakulmio, jonka vasen yläkulma on kohdassa (112,60) ja oikea alakulma kohdassa (142,100).

```
10 PMODE3,1:SCREEN1,1:PCLS:DIM R(29,39)
20 R$="C8BM120,96;U26R13D26L13;C7L6U6E6BR13
    F6D6L6BU26H6G6"
30 DRAW R$
40 GET (112,60)-(142,100),R,G
100 GOTO 100
```

Ylläoleva esimerkkiohjelma piirtää raketin kuten aiemminkin (kaikki on nyt yhdessä merkkijonossa), ja tallentaa sen taulukkoon R. Tarvittavan taulukon koko on vain 29 x 39, koska voimme käyttää taulukossa nolla-alkioita.

Tallennettuamme piirroksen meidän pitäisi nyt saada se takaisin ruudulle. Tähän käytetään PUT-lausetta, jonka muoto on samankaltainen kuin GET-lauseen.

PUT (x1,y1)-(x2,y2), taulukkonimi, toiminto

(x1,y1)-(x2,y2) ovat kuten aiemminkin suorakulmion koordinaatit, mutta tällä kertaa ne viittaavat alueeseen, johon piirros halutaan sijoittaa (PUT) eivätkä sen aikaisempaan sijaintiin. Taulukkonimi on sen taulukkomuuttujan nimi, johon piirros on tallennettu. Toiminto-parametri on välttämätön vain silloin, kun GET-lauseessa on käytetty G-parametriä. Toiminto on jokin seuraavista sanoista ja se määrittää, miten piirros tulostetaan uuteen sijaintikohtaansa ruudulla.

PSET Sytyttää jokaisen pisteen niin kuin ne ovat taulukossa. Tulostaa kuvan siis sellaisena kuin se on tallennettu.

PRESET Sammuttaa jokaisen lähtötaulukon pisteen. Tämä joko hävittää kuvan tai vaihtaa sen väriä riippuen käytettävästä taustaväristä ja kuvi-
on väristä.

AND Vertaa alkuperäisen piirroksen pisteitä sijoituskohdan pisteisiin ruudulla. Jos molemmat ovat päällä, piste sytytetään. Jos jompi kumpi ei ole päällä, pistettä ei sytytetä. Jos siis kaksi kuvaa sijoitetaan päällekkäin, vain päällekkäin osuvat pisteet näkyvät.

OR Vertaa pisteitä kuten edellä. Jos joko lähtö-
taulukon tai tulokohdan piste on kytketty pääl-
le, piste ruudulla syttyy. Kuva siis peittää
toisen kuvan.

NOT Tämä parametri muuttaa päinvastaiseksi jokai-
sen näyttöalueen pisteen ja kuvion väri muut-
tuu siten taustaväriksi.

PUT-lausetta on aina käytettävä samassa tilassa kuin GET-lau-
setta muutoin saattaa syntyä outoja kuvioita. Palaamme nyt
esimerkkiin ja sijoitamme (PUT) raketin toiseen paikkaan.

Lisää seuraavat rivit:

```
50 Y=150:FOR X=10 TO 210 STEP 40
60 PUT(X,Y)-(X+30,Y+40),R,PSET
80 NEXT X
```

Ruudun alareunassa on nyt vierekkäin useita raketteja. Jot-
ta raketti saataisiin liikkumaan ruudun alareunaa pitkin,
lisätään vain rivi:

```
70 FOR D=1 TO 200:NEXT D:PCLS
```

Käyttämällä GET- ja PUT-lauseiden yhteydessä ohjausvipuja
voit liikutella piirrosta haluamallasi tavalla.

```
10 PMODE3,1:SCREEN1,1:PCLS:DIM S(48,48)
20 DRAW "BM24,12,S8;C4;E2H2D4D8R8H8G8R2NR6F3R6E3"
30 GET (0,0)-(48,48),S
40 A$=INKEY$:IF A$=""THEN 40
50 PCLS:A=JOYSTK(0)*3.25:B=JOYSTK(1)*2.25
60 PUT (A,B)-(A+48,B+48),S:GOTO 50
```

Esimerkki piirtää kuvan ruudun vasempaan yläkulmaan. Ruutu voidaan nyt tyhjentää painamalla mitä tahansa näppäintä ja kuvaa voidaan liikutella vasemman ohjausvivun avulla.

Jos haluat säästää tilaa, voit määritellä taulukon koon pienemmäksi GET- ja PUT-lauseita varten. Taulukon koko määrätään hienografiikkapisteinä lasketun suorakulmion alan mukaan. Ala $((x_2-x_1)*(y_2-y_1))$ jaetaan D:llä, jolloin D=160 tilassa PMODE0, D=80 tiloissa PMODE1 ja 2, D=40 tiloissa PMODE3 ja 4. Pyöristä tulos lähimpään kokonaislukuun ja lisää vielä 1.

Olemme nyt käsitelleet graafiset toiminnot. Antamamme esimerkit ovat väistämättä rajallisia eivätkä anna kovin hyvää kuvaa siitä, miten kaikkea voidaan tehdä vähäisellä ajatus-työllä ja kärsivällisyydellä. Kuvien piirtämistä voidaan helpottaa huomattavasti, jos ne suunnitellaan etukäteen ja piirretään ensin graafisille luonnosarkeille.

DRAW

DRAW-lauseella voidaan piirtää viiva tai useita viivoja merkkijonossa annettujen ohjeiden mukaan. Se toimii ainoastaan hienografiikkatilassa.

DRAW merkkijono

Merkkijono voi olla joko merkkijonovakio tai merkkijonomuuttuja ja siihen voi sisältyä seuraavia aliohjeita:

Mx,y siirry piirtokohtaan x,y ruudulla

U n ylös n pistettä

D <u>n</u>	alas <u>n</u> pistettä
L <u>n</u>	vasemmalle <u>n</u> pistettä
R <u>n</u>	oikealle <u>n</u> pistettä
E <u>n</u>	45 asteen kulmassa <u>n</u> pistettä
F <u>n</u>	135 asteen kulmassa <u>n</u> pistettä
G <u>n</u>	225 asteen kulmassa <u>n</u> pistettä
H <u>n</u>	315 asteen kulmassa <u>n</u> pistettä
X	suorittaa alimerkkijonon ja palaa takaisin
C <u>m</u>	antaa viivalle värin <u>m</u>
A <u>k</u>	kiertää seuraavaa viivaa annetussa kulmassa (angle)
	k=0 0 astetta k=1 90 astetta
	k=2 180 astetta k=3 270 astetta
S <u>k</u>	muuttaa kuvan mittakaavaa 1/4-asteikolla
	k=1 on neljäsosa k=8 kaksinkertainen
	k=4 on standardiarvo
N	piirtokohtaa ei pidetä muistissa (no update)
B	tyhjä (siirtyminen viivaa piirtämättä)

Suhteellinen siirtyminen voidaan määrittää M-parametrillä seuraavasti:

M x-siirtyminen, y-siirtyminen

Tällöin x-siirtyminen ja y-siirtyminen ovat lukuja, jotka ilmoittavat etäisyyden tämänhetkisestä sijainnista. Mo-

lemmilla luvuilla on oltava etumerkki, joko plus(+) tai miinus(-).

Esimerkki DRAW-lauseen käytöstä on esitetty lokerossa PUT.

GET

GET-lausetta voidaan käyttää vain hienografiikkatilassa. GET kopioi määrätyn suorakulmion muotoisen alueen sisällön ruudulta ja tallentaa sen taulukkoon. Taulukko on määritettävä etukäteen ja sen on oltava oikean kokoinen.

GET (x1,y1)-(x2,y2), taulukkonimi, G

x1,y1 ja x2,y2 ovat ruudulta kopioitavan suorakulmion vasemman yläkulman ja oikean alakulman koordinaatit.

Taulukkonimi on sen etukäteen määritellyn taulukon nimi, johon suorakulmion sisältö tallennetaan.

G on valinnainen parametri. Se ilmoittaa tallennettavien graafisten yksityiskohtien asteen.

Ks. PUT-käskyä käsittelevästä lokerossta esimerkki GET-lauseen käytöstä.

PUT

PUT-lausetta käytetään tulostettaessa GET-käskyllä tallennettua graafisen taulukon sisältöä.

PUT-lausetta on käytettävä samassa tilassa, jota käytettiin taulukkoa muodostettaessa, muutoin saadaan sekavia tuloksia.

PUT (x₁,y₁)-(x₂,y₂), taulukkonimi, toiminto

x₁,y₁ ovat ruudulta valitun alueen vasemman yläkulman koordinaatit ja x₂,y₂ oikean alakulman koordinaatit. Taulukkonimi on aikaisemmin määritetty taulukko, joka sisältää graafiset yksityiskohdat. Toiminto on valinnainen parametri, mutta sitä on käytettävä, jos GET-lauseessa käytettiin parametria G.

PSET tulostaa pisteet ruudulle lähtötaulukon mukaisesti.

PRESET sammuttaa kaikki lähtötaulukossa päälle kytketyt pisteet

AND vertaa lähtötaulukkoa ja tulokohtaa. Jos molemmissa piste on kytketty päälle, se sytytetään, muussa tapauksessa se sammutetaan.

OR vertaa pisteitä kuten edellä, ja jos jompi kumpi on päällä, ruudun piste sytytetään.

NOT kääntää päinvastaiseksi kaikki pisteet kohdealueella lähtötaulukosta riippumatta.

Valitun tulostusalueen ruudulla on oltava samansuuruinen kuin taulukko, muutoin ruudulle tulostuu pelkkää sotkua.

```
10 PCLEAR 4:Pmode3,1:PCLS:SCREEN1,1:DIM W(30,30)
```

```
20 DRAW "BM10,12;S8;R1U3R1D2R2U2R1D3R1D2L1  
D2R1D1L2U3L4D3L2U1R1U2L1U2R1U2BR1BD1D2R2  
U2NL2R2D2L2U2"
```

```
30 PAINT (11,13),6,5:GET (0,0)-(30-30),W
```

```
40 A$=INKEY$:IF A$=""THEN 40
50 PCLS:FOR C=0 TO 100 STEP 20
60 FOR A=0 TO 200 STEP 20
70 PUT (A,C)-(30+A,30+C),W
80 PUT (A,C+30)-(30+A,60+C),W
90 PUT (A,C+60)-(30+A,90+C),W
100 PLAY "T255;ABFGBA":PCLS:NEXT A,C
```

11. LOPPUSILAUS

LISÄÄ TULOSTUKSESTA

Vaikka tulostusta ruudulle voidaankin ohjailla melko hyvin PRINT ja PRINT@-lauseilla, on olemassa vielä yksi lisätoiminto. PRINT USING-lauseella voidaan määritellä täsmällisesti, miten kukin rivi pitäisi kirjoittaa. Tämä käsky on erittäin hyödyllinen laadittaessa taulukoita, kaavioita ja kirjanpilotulostuksia.

```
PRINT USING muotolause; tulostuslista
```

Muotolause on merkkijonovakio tai merkkijonomuuttuja, joka sisältää ohjeita tulostuslistan kirjoittamista varten.

Tulostuslista on tavallinen luettelo, joka muodostuu vakioista ja muuttujista samalla tavoin kuin PRINT-lauseissa. Muotolause sisältää 'kenttätarkenteita'. Ne muodostuvat merkeistä, jotka ilmoittavat koneelle, kuinka monta merkkipaikkaa käytetään lukua tai merkkijonoa tulostettaessa.

tarkenne

Tällä merkillä ilmoitetaan luvun jokaisen numeron merkkipaikka.

```
PRINT USING"###.##";A
```

Ylläoleva lause käskee tulostaa A:n arvon siten, että ennen desimaalipistettä on 3 numeroa ja pisteen jälkeen 2 numeroa. Jos desimaalipisteen jälkeen on enemmän kuin kaksi numeroa, luku pyöristetään. Käyttämättä jääneet paikat desimaalipisteen vasemmalla puolella merkitään tyhjillä merkeillä. Jos

luku on liian suuri eikä mahdu annettuun tilaan, tietokone yrittää silti tulostaa luvun ja merkitsee sen eteen % ilmoittaakseen ylityksestä.

```
PRINT USING"###.##";13.4695
```

```
∇13.47
```

```
PRINT USING"###.##";1492.878
```

```
%1492.88
```

```
PRINT USING"###.###";146
```

```
146.00
```

```
PRINT USING"### ";18.76
```

```
∇19
```

* tarkenne

Rahasummaa käsiteltäessä ei ole hyvä, että lukujen eteen jätetään tyhjiä kohtia, etenkin sekkejä kirjoitettaessa. Tämä voidaan ratkaista käyttämällä tähtimerkkiä (*). Jos lisäät kaksi tähteä numeerisen kentän eteen, tyhjät kohdat merkitään tähdillä.

```
PRINT USING"*# ## ,###";1.492
```

```
***1.49
```

+ tarkenne

Jos numeerisen kentän eteen merkitään +-merkki, luvun etumerkki kirjoitetaan näkyviin.

```
PRINT USING"+ ###.##";14.7
```

```
∇+14.70
```

```
PRINT USING"+** ###.##";-7.4
```

```
***-7.40
```


Jos plusmerkki kirjoitetaan numeerisen kentän loppuun, etumerkki kirjoitetaan luvun perään.

```
PRINT USING"###.##+";27.86
```

```
∇27.86+
```

```
PRINT USING"###.##+";-1.6
```

```
∇∇1.60-
```

Jos luvun perään merkitään miinusmerkki, kaikkien negatiivisten lukujen jälkeen merkitään miinusmerkki, ja kaikkien positiivisten lukujen perään jätetään tyhjä.

```
PRINT USING"xx###.##-";-12.418
```

```
xxx12.42-
```

```
PRINT USING"###.##-";47.25
```

```
∇47.25∇
```

↑↑↑ tarkenne

Tämä merkki tarkoittaa, että luku kirjoitetaan eksponentiaalissa muodossa. Neljä ylöspäin osoittavaa nuolta merkitään lukukentän perään.

```
PRINT USING"##.###↑↑↑↑";123456
```

```
1.2346E+05
```

! tarkenne

Tätä merkkiä käytetään merkkijonojen yhteydessä, jolloin vain merkkijonon ensimmäinen merkki tulostetaan.

```
PRINT USING"!";"KREDIT"
```

```
K
```

% tarkenne

Merkkijonoja tulostettaessa on määriteltävä tulostuksen vaatiman kentän leveys. Tämä tehdään kahdella %-merkillä, joiden väliin jätetään tarvittava määrä tyhjiä lyönnejä. Kentän leveys on tällöin tyhjien lukumäärä plus kaksi. Jos merkkijono on pitempi kuin käytettävissä oleva kenttä, tulostetaan vain ensimmäiset n merkkiä, ja n on tällöin kentän leveys.

```
PRINT USING"%VVVVV%";"DEBET"
```

```
DEBETVV
```

```
PRINT USING"%V%";"SALDO"
```

```
SAL
```

\$ tarkenne

Dollarimerkkiä (\$) käytetään rahayksikön symbolina. Jos se merkitään numeerisen tarkenteen eteen, se näkyy myös tulostuksessa.

```
PRINT USING"$ ### . ##"+2.87
```

```
$VV2.87
```

Jos käytetään kahta dollarimerkkiä, merkki tulostetaan aivan luvun eteen.

```
PRINT USING"$ $ ### . ##";2.87
```

```
VVV$2.87
```

Jos dollarimerkin yhteydessä käytetään kahta tähteä, tulostus on seuraava:

```
PRINT USING"*x$ # . ##";14.9
```

```
*$14.90
```

Tyhjät välilyönnit ja muut merkit, joita muotolausejonossa esiintyy, tulevat myös tulostukseen:

```
PRINT USING"KESKIARVOVV ##.###VVVYHTEENSÄVV ###.##";
```

3.4,40.8

```
KESKIARVOVVV3.40VVVYHTEENSÄVVV40.80
```

Jos tulostuslistassa on enemmän alkioita kuin muotolauseessa on kenttiä, muotolause aloitetaan uudelleen alusta

```
PRINT USING" ###.##VVV";7.84,142.5,.234
```

```
VV7.84VVV142.50VVVVVV0.23
```

Ruudulle tulostettaessa PRINT USING-lauseella tulostettavan rivin pituus on tietenkin edelleen enintään 32 merkkiä. Jos se on pitempi, se jatkuu automaattisesti seuraavalle riville. Kirjoitinta käytettäessä rivi voi olla huomattavasti pitempi (useimmat kirjoittimet kirjoittivat ainakin 80 merkkiä riville). Tällöin lauseen muoto on seuraava:

```
PRINT #-2, USING muotolause; tulostuslista
```

Muotolause ja tulostuslista ovat samat kuin edellä, #-2 tarkoittaa, että tieto lähetetään kirjoitinkanavaa pitkin eikä ruudulle. Jos halutaan tulostaa sekä ruudulle että kirjoittimelle, on käytettävä kahta PRINT USING-lausetta. Voit käyttää myös PRINT#V-lausetta, jossa V on muuttuja, jonka arvo on -2 kirjoittimelle ja 0 (nolla) ruudulle tulostettaessa.

KASETTISYÖTTÖ JA -TULOSTUS

Tähänastisissa ohjelmissa olemme joutuneet näppäilemään koneeseen kaikki tarvitsemamme tiedot (tai lukemaan ne DATA-

lauseesta) ja kaikki tulostus on tapahtunut ruudulle. Tietoa voidaan kuitenkin myös tallentaa kasetille samoin kuin ohjelmiakin. Tallennettu tieto voidaan sitten lukea tarvittaessa. Kasettinauhuri kytketään tietokoneeseen ja saatetaan käyttö- kuntoon aivan samoin kuin ohjelmia tallennettaessa. Tietokoneelle on ilmoitettava, että se käyttää tiedostoja. Tämä tehdään OPEN-lauseella.

OPEN a, #-1, tiedoston nimi

Muuttujan a on oltava joko "O" tai "I". "O" tarkoittaa tulostusta (output), siis tiedon siirtymistä koneesta kasettinauhalle. "I" tarkoittaa syöttöä (input), siis tiedon siirtymistä kasetilta koneeseen.

#-1 ilmoittaa koneelle, että käytetään kasettinauhuria.

Tiedoston nimi on se nimi, joka tiedostolle halutaan antaa (mikä tahansa nimi, joka alkaa kirjaimella ja on enintään 8 merkkiä pitkä).

Seuraavaksi kirjoitetaan tieto kasettihauhalle. Tämä tapahtuu PRINT-lauseella seuraavasti:

PRINT #-1, tulostuslista

Ainoa ero aikaisempiin PRINT-lauseisiin verrattuna on #-1. Se kertoo tietokoneelle, että tulostuslista kirjoitetaan nauhalle eikä ruudulle. Kun kirjoittaminen lopetetaan, tiedosto on suljettava CLOSE-lauseella.

CLOSE #-1

PRINT USING

PRINT USING-lauseella voidaan paremmin ohjailta tulostusta ruudulle tai kirjoittimelle.

PRINT USING muotolause; tulostuslista

Muotolause on merkkijonovakio tai -muuttuja, johon sisältyvät "kentätarkenteet", jotka ilmoittavat, miten tulostuslista kirjoitetaan. Tulostuslista on merkkijono- tai numeromuuttujien (tai vakioiden) luettelo, jossa muuttujat on erotettu toisistaan pilkuilla. Kentätarkenteita ovat:

<u>Merkki</u>	<u>Toiminto</u>	<u>Esimerkki</u>	<u>Tulos</u>
#	Sieventää lukuja	"####";147.2	147
.	Desimaalipiste kiinteässä paikassa	"###.###";34.678	34.68
,	Kirjoittaa pilkun joka kolmannen numeron vasemmalle puolelle	"###.###.#";123456	123,456
##	Täyttää tyhjät paikat luvun edessä tähdillä	"##.###.###";1.47	###1.470
\$	Kirjoittaa dollarimerkin (\$) luvun eteen	"\$###.###";12.689	\$V12.69
##\$	Liukuva dollarimerkki	"##\$.###.#";12.689	###\$12.69
+	Ensimmäisenä merkinä käskee kirjoittaa etumerkin luvun eteen, viimeisenä merkinä käskee kirjoittaa etumerkin luvun perään	"#.##+";-12.689	12.69-
↑↑↑	Kirjoitetaan eksponentinmuodossa	"###.###↑↑↑";12.689	1.27E+01
!	Kirjoittaa vain jonon ensimmäisen merkin	"!";"KREDIT"	K

```
%välilyöntejä% Merkkijonokenttä. Kentän pi-
tuus on välilyöntien määrä
plus 2 "%VVV%";SALDO SALDO
```

Jokaisen kenttätarkenteen perään voidaan lisätä haluttu määrä väli-
lyöntejä, jotka näkyvät tulostusrivillä tyhjinä paikkoina.

```
10 CLS:INPUT"NÄPPÄILE VIIMEINEN SALDO";B:C=0:D=0
20 CLS:T$="VVVV%VVV%VVV%VVVV%VVV%VVV%"
30 L$="VVV####.## VV ## ## .## VV ## ## .## +"
40 PRINT USING T$;"DEBET","KREDIT","SALDO"
50 PRINT USING L$;D,C,B
60 OPEN "I", #-1,"SEKKI"
70 IF EOF(-1)THEN 110
80 INPUT #-1,A:D=0:C=0
90 IF A<0 THEN D=ABS(A)ELSE C=A
100 B=B+C-D:PRINT USING L$;D,C,B:GOTO 70
110 CLOSE #-1:END
```

TULOSTUS KIRJOITTIMELLE

Käytettäessä kirjoitinta, joka on kytketty I/O-rinnakkaisporttiin, tulostus voidaan ohjata ruudun sijasta kirjoittimelle eräiden käskyjen muunnelmia käyttäen.

```
PRINT #-2, tulostuslista
PRINT #-2, USING muotolause; tulostuslista
```

Muotolause ja tulostuslista ovat samat kuin ruudulle tulostettaessa. POS(-2) antaa kirjoituspään sijainnin.

LLIST listaa ohjelman suoraan kirjoittimelle. Sitä käytetään kuten LIST-käskyä. Painamalla yhtäikaa SHIFT-näppäintä ja nollaa(0)

kirjoittimelle voidaan kirjoittaa pienillä kirjaimilla. Pieniä kirjaimia voidaan käyttää vain merkkijonoissa ja REM-lauseissa, sillä kaikki ohjelmalauseet ja koneelle annettavat käskyt on kirjoitettava isoilla kirjaimilla. Ellei kirjoittimessa ole rivinvaihtoa, voidaan toimia kahdella tavalla: Joko pysäyttää kirjoitin (e) tai näppäillä POKE 330,2.

Kirjoitin saadaan kirjoittamaan suomalaisilla aakkosilla näppäilemällä seuraavasti:

<u>Englanti</u>	<u>Suomi</u>	<u>Painettavat näppäimet</u>
[Ä	[SHIFT] - [↓]
\	Ö	[SHIFT] - [CLEAR]
]	A	[SHIFT] [+]

Tieto luetaan tiedostosta takaisin koneeseen käyttäen samoja toimintoja, mutta nyt tiedosto avataan syöttöä varten ("I") ja PRINT-käskyn sijasta käytetään muotoa:

```
INPUT #-1, syöttötietolista
```

CLOSE-käsky on molemmissa tapauksissa sama.

Seuraava esimerkki osoittaa, miten tämä tapahtuu. Aseta ensin kasettinauhuri käyttökuntoon ja kelaat nauha kohtaan, johon haluat tallentaa tiedoston. Käytä SKIPF-käskyä. Paina sitten samanaikaisesti PLAY- ja RECORD-näppäimiä.

```
10 CLS:PRINT "LAADI PUHELINLUETTELO"
```

```
20 OPEN "0", #-1,"PUHELIN":PRINT "NÄPPÄILE XXX,XXX LOPETTAESSASI"
```

```
30 PRINT@128,"";:INPUT "NIMI V",N$;
```

```
40 INPUT "PUH.NO.";T$:IF N$="XXX"OR T$="XXX"THEN 60
50 PRINT # -1,N$,T$:PRINT@ 128,"":GOTO 30
60 CLOSE #-1:END
```

Kun ohjelma suoritetaan, nauhuri käynnistyy ja avaa tiedoston. Aina, kun näppäilet nimen ja puhelinnumeron, ne kirjoitetaan tiedostoon. PRINT@128-lause rivillä 30 tyhjentää ruudun. Tämä jatkuu, kunnes näppäillään XXX,XXX, joka sulkee tiedoston, ja ohjelma päättyy. Nyt tarvitsee vain lukea tiedot takaisinkoneeseen. Tärkein ero tulostuksen ja syötön välillä on se, että syötettäessä on varottava ohittamasta tiedoston loppua. Tämä varmistetaan ylimääräisellä lauseella, jota käytetään syötettäessä. EOF-lause tarkistaa, onko tiedoston loppu saavutettu.

Kelaa kasettinauha takaisin alkuun ja paina alas vain PLAY-näppäin.

```
10 CLS:PRINT "LUE PUHELINLUETTELO"
20 OPEN "I", #-1,"PUHELIN"
30 PRINT "NIMI","NUMERO"
40 IF EOF(-1)THEN 60
50 INPUT #-1,A$,B$:PRINT A$,B$:GOTO 40
60 CLOSE #-1:END
```

Kun ohjelma nyt suoritetaan, nauha käynnistyy ja etsii tiedoston "PUHELIN". (Tämä saattaa kestää jonkin aikaa, jos tiedosto on nauhan lopussa). Ohjelma lukee nyt sisään nimen ja puhelinnumeron ja näyttää ne ruudulla. Sinun ei tarvitse käyttää samoja muuttujanimiä, joita käytit tietoja kirjoittaessasi. Muuttujatyypin on kuitenkin oltava sama. Kun

tullaan tiedoston loppuun, tiedosto suljetaan ja ohjelma päättyy.

EOF-käskyn on oltava ennen INPUT -1-lauseetta, muuten kone antaa IE-virheen (yritetty lukea tiedoston lopun ohi). Muista aina sulkea tiedosto (CLOSE), muutoin saattaa syntyä ongelmia erityisesti tiedostoon kirjoitettaessa.

KONEKIELI

Seuraava askel Basic-kielen jälkeen on konekieli. Tämä on tietokoneen oma kieli. Tähän asti tietokoneen kanssa on keskusteltu Basic-kieltä ymmärtävän tulkkiohjelman välityksellä. Konekielen etu on, että ohjeet suoritetaan huomattavasti nopeammin, ne vaativat vähemmän muistitilaa ja niillä voidaan tehdä asioita, jotka eivät onnistu Basicin avulla.

Konekieleen voidaan parhaiten tutustua ohjekirjan avulla, joka koskee erityisesti 6800-sarjan mikroprosessoreita. Ks. kirjallisuusluetteloä tämän kirjan lopussa.

Tietokoneessa on useita rutineja, joiden avulla voit käyttää hyväksesi konekielellä kirjoitettuja ohjelmia. Niitä kuvataan lyhyesti seuraavassa.

USRn-komennolla voit kutsua kymmentä (0-9) konekielistä ohjelmaa. Lauseen muoto on:

USR n(argumentti)

Argumentti on tällöin merkkijono tai numerolauseke. Kun ohjelma tulee USR-kutsun kohdalle, se haarautuu osoitteeseen, joka on annettu DEF USR n-lauseessa. Osoite ilmoittaa konekielisen rutiinin aloituskohdan. DEF USR n-lauseetta käytte-

tään USR n-funktion osoitteen määrittämiseen. Muoto on seuraava:

```
DEF USR n = osoite
```

jossa n on jokin luku väliltä 0-9 ja sama kuin n USR-kutsussa. Osoitteen on oltava välillä 0-65535 ja sen tulee sisältää USR n-kutsun aloitusosoite.

```
CLEAR s,h
```

CLEAR-lauseella varataan muistitilaa USR-funktiolle. s varaa tilaa merkkijonoille, kuten aikaisemminkin. h ilmoittaa suurimman muistiosoitteen, jota Basic voi käyttää. Konekielisille rutiineille on siten varattu tila osoitteesta h+1 eteenpäin.

POKE-komentoa käytetään, kun halutaan kirjoittaa jokin arvo tiettyyn paikkaan muistissa.

```
POKE osoite, arvo
```

Osoite on kuten edellä ja arvon oltava välillä 0-255.

VARPTR on Basic-muuttujan osoitin, jota voidaan käyttää argumenttina USR-funktiossa. Se mahdollistaa USR-funktion pääsyn taulukon sisältöön.

```
VARPTR (muuttujanimi)
```

Muuttujanimi on tässä Basic-muuttuja, johon halutaan päästä.

VARPTR-osoitinta käytetään USR-argumentin osana:

```
USRØ (VARPTR(X))
```

Konekieliset rutiinit voidaan tallentaa kasetille ja ladata kasetilta takaisin koneeseen CSAVEM- ja CLOADM-komentoja käyttäen.

CSAVEM nimi, alku, loppu, tulokohta

CLOADM nimi, siirtymä

Nimi on nauhalla olevan tiedoston nimi, alku on muistissa olevan rutiinin aloitusosoite, loppu on viimeinen rutiinin varaama osoite ja tulokohta on ohjelman aloituskohta. Siirtymä CLOADM-komennossa antaa mahdollisuuden ladata rutiini uudelleen muistiin parametrien alku+siirtymä antamaan osoitteeseen.

Kun rutiini on luettu muistiin, se voidaan suorittaa EXEC-komennolla:

EXEC osoite

Osoite on rutiinin alku. Jos osoite jätetään pois, kone käyttää edellisessä CLOAD-komennossa annettua alku-arvoa.

Alla oleva ohjelmaesimerkki osoittaa, miten muuttujien arvoja voidaan siirtää Basic- ja konekielisten ohjelmien välillä. Konekielinen ohjelma luetaan muistiin DATA-lauseista ja se muorittaa kertolaskun kahdella luvulla, joiden maksimiarvot ovat 127 ja 255.

```
10 REM KONEKIELINEN OHJELMOINTI
20 CLEAR 20,3199:ALKU=3200:LKM=8
40 DEF USR5=ALKU
50 CLS
60 FOR OSOITE=ALKU TO ALKU+LKM-1
70 READ ARVO
80 POKE OSOITE,ARVO
90 NEXT OSOITE
```

```
100 'KONEKOODI MUISTISSA
110 INPUT"ANNA KAKSI LUKUA";T1,T2
120 T3=256*T1+T2
130 T4=USR05(T3)
140 PRINT T1;"*T2;"=";T4
150 END
160 'HAKEE ARVON JSR $8B2D
170 DATA &HBD,&H8B,&H2D
180 'KERTOMINEN
190 DATA &H3D
200 'PALAUTTAA ARVON JSR $8C37
210 DATA &HBD,&H8C,&H37
220 'PALUU BASIC-OHJELMAAN RTS
230 DATA &H39
```

LIITE A

ASCII-koodit

<u>Näppäin</u>	<u>Desimaaliarvo</u>	<u>Desimaaliarvo</u>
	<u>ilman SHIFT-näppäintä</u>	<u>SHIFT-näppäimen avulla</u>
[BREAK]	3	3
[CLEAR]	12	92
[ENTER]	13	13
[VÄLILYÖNTI]	32	32
!	33	-
"	34	-
#	35	-
\$	36	-
%	37	-
&	38	-
'	39	-
(40	-
)	41	-
*	42	-
+	43	-
,	44	-
-	45	-
.	46	-
/	47	-
0	48	18
1	49	-
2	50	-
3	51	-
4	52	-
5	53	-
6	54	-
7	55	-
8	56	-
9	57	-
:	58	-
;	59	-
<	60	-
=	61	-
>	62	-
?	63	-
⊙	64	19
A	97	65
B	98	66
C	99	67
D	100	68
E	101	69
F	102	70
G	103	71
H	104	72
I	105	73
J	106	74
K	107	75
L	108	76
M	109	77
N	110	78

<u>Näppäin</u>	<u>Desimaaliarvo</u>	<u>Desimaaliarvo</u>
	<u>ilman SHIFT-näppäintä</u>	<u>SHIFT-näppäimen avulla</u>
O	111	79
P	112	80
Q	113	81
R	114	82
S	115	83
T	116	84
U	117	85
V	118	86
W	119	87
X	120	88
Y	121	89
Z	122	90
↑	94	95
↓	10	91
+ →	8	21
→	9	93

Jos painetaan samanaikaisesti SHIFT- ja Ø-näppäintä seuraavat kirjaimet tulostuvat pieninä kirjaimina kirjoittimella. Kun painetaan uudelleen SHIFT ja Ø, siirrytään takaisin normaalille tasolle. Seuraavat alatasolla (pienet kirjaimet) olevat merkit saadaan käyttöön CHR\$()-funktioiden avulla:

```
[ CHR$(123)      ↑ CHR$(126)
\ CHR$(124)      + CHR$(127)
] CHR$(125)
```

GRAAFISET MERKIT

Merkit 128-255 ovat seuraavanlaisia graafisia merkkejä:



128



129



130



131



132



133



134



135



136



137



138



139



140



141



142



143

Nämä merkit saadaan käyttöön, kun CHR\$-funktioon liitetään asianomainen lukukoodi. Väritelevisiossa harmaa alue on vihreä. Muut värit saadaan lisäämällä haluttua väriä vastaava luku. Esimerkiksi PRINT CHR\$(142+112) tulostaa merkin 142, jossa harmaa alue on oranssi.

+16 keltainen

+32 sininen

+48 punainen

+64 valkoinen

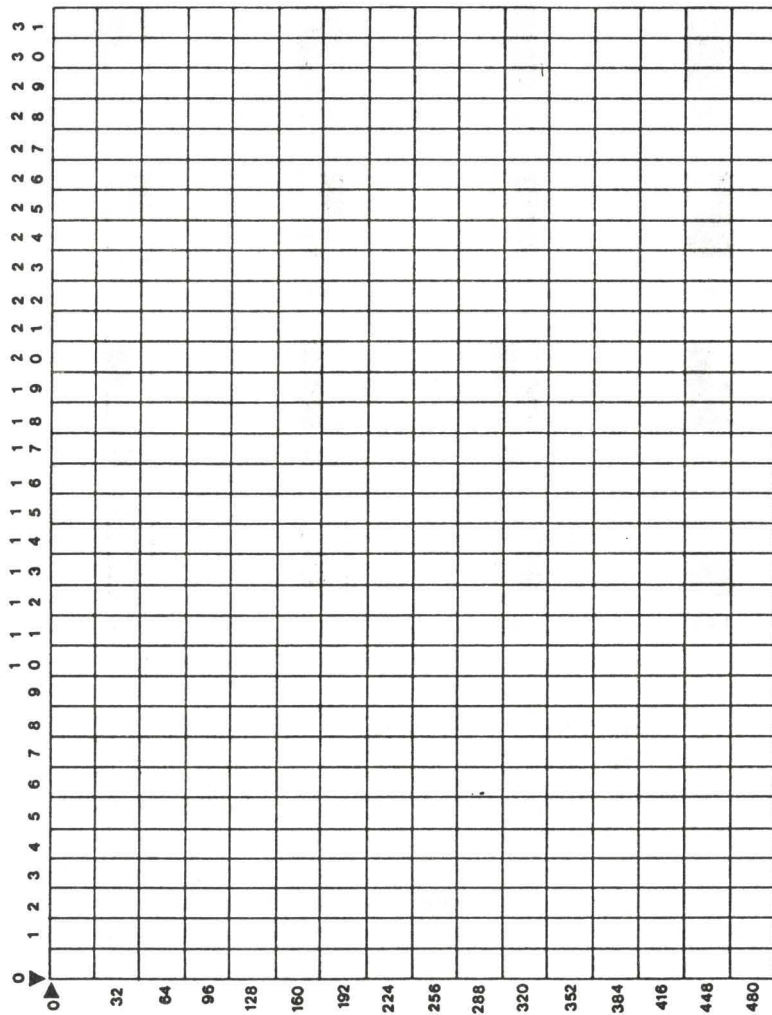
+80 turkoosi

+96 sinipunainen

+112 oranssi

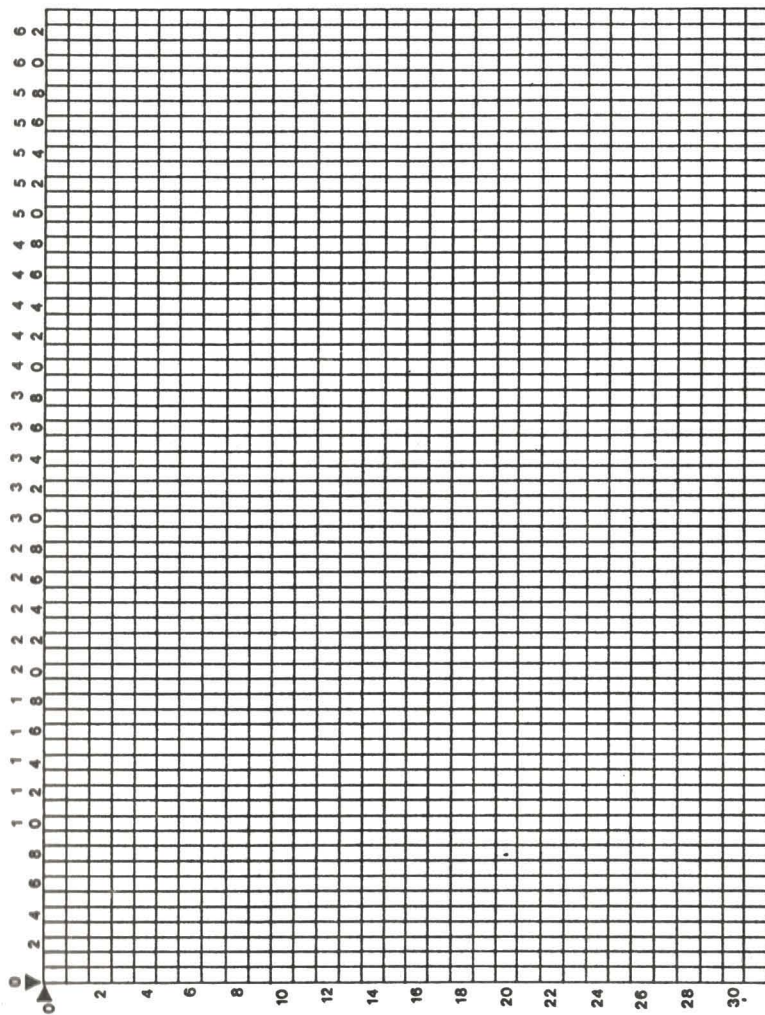
LIITE B

Ruudukko PRINT®-lausetta varten



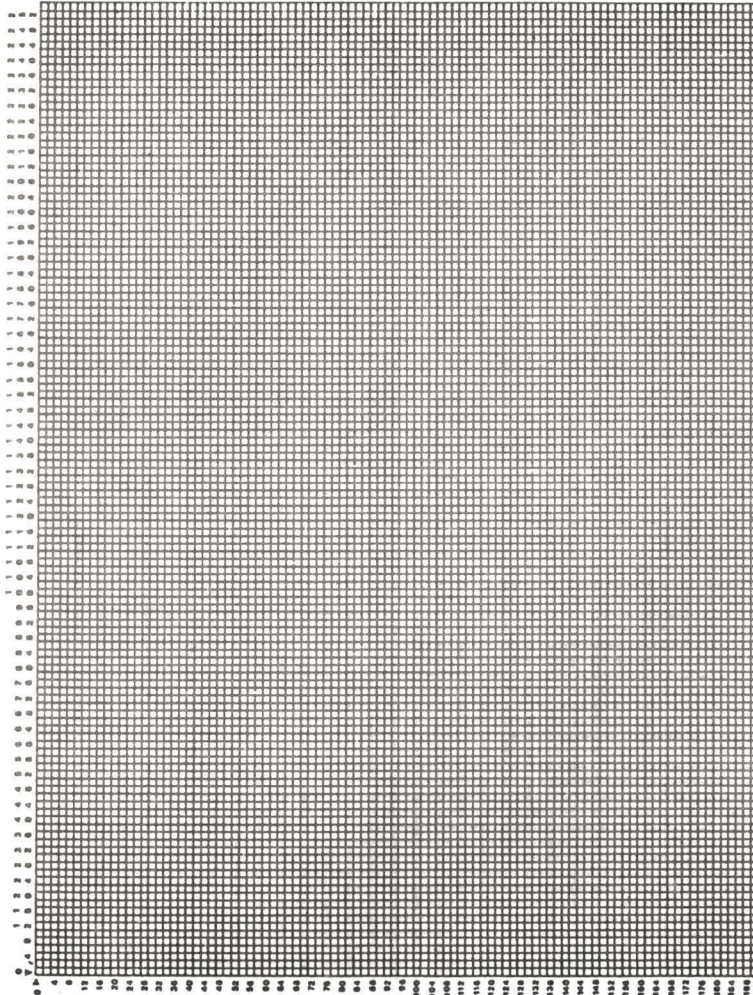
Puoligrafiikkaruudukko

Tämä toinen ruudukko on tekstiruudun puoligrafiikkatiloja varten, joissa käytetään SET ja RESET-käskyjä.



Ruudukko hienografiikkaa varten

Kolmas ruudukko on hienografiikkaruutu, ja sitä voidaan käyttää kaikkien hienografiikkakomentojen yhteydessä.



LIITE C

Virhekoodit

<u>Koodi</u>	<u>Selitys</u>
/Ø	Yritetty jakaa nollaa
AO	Yritetty avata jo avoinna oleva tiedosto. Yleisin syy: painettu RESET-näppäintä tiedostoja käyttävän ohjelman keskeyttämiseksi. Kytke laite pois päältä ja uudelleen päälle.
BS	Virheellinen indeksi. Yleisin syy: indeksin arvo on suurempi kuin taulukon ilmoitetut mittasuhteet.
CN	Ei voi jatkaa. Yritetty käyttää CONT-käskyä ohjelman lopussa.
DD	Yritetty antaa uudet mittasuhteet taulukolle. Taulukon koko voidaan määritellä vain kerran samassa ohjelmassa.
DN	Väärä laitenumero (device number). Esim. PRINT A kun A>Ø tai A<-2.
DS	Komentovirhe. Yritetty ottaa CLOAD tiedostosta.
FC	Virheellinen funktiokutsu. Yleensä väärä parametriarvo tai väärä muuttujatyyppi.
FD	Virheellinen tieto tiedostossa. Kasettitiedostossa virheellistä tietoa, jota ei voida lukea merkkijonomuuttujaan.
FM	Virheellinen tiedostotila. Yritetty syöttää (INPUT) tiedostosta, joka on avattu (OPEN) tulostusta (O) varten tai kirjoittaa (PRINT) tiedostoon, joka on avattu (OPEN) syöttöä (I) varten.
ID	Virheellinen suora käsky. Yritetty käyttää käskyä, jota voidaan käyttää vain ohjelmassa. Esimerkiksi INPUT, DEF FN.
IE	Yritetty lukea ohi tiedoston lopun. Käytä lausetta IF EOF (-1) tämän estämiseksi.
IO	Syöttö/tulostus -virhe. Kasettinauhuria ei ole asennettu oikein tai nauha on huono.
LS	Merkkijono liian pitkä. Maksimi on 255 merkkiä.
NF	NEXT-lause ilman FOR-osaa. Esiintyy yleensä, kun sisäkkäiset FOR-silmukat ovat epäjärjestyksessä.
NO	Tiedosto ei ole avattu. Tiedoston lukeminen ja siihen kirjoittaminen on mahdollista vasta, kun tiedosto on avattu (OPEN).
OD	Ei enempää tietoa. READ-lause on lukenut kaikki DATA-lauseet.
OM	Muistitila täynnä. Kaikki muistitila on käytetty tai varattu.

- OS Muistissa ei tilaa merkkijonomuuttujille. Varaa CLEAR-lauseella lisää tilaa, mikäli mahdollista.
- OV Liian pitkä luku. Luku on niin pitkä, ettei kone pysty käsittelemään sitä (ABS(X)>1E38).
- RG RETURN-lause ilman GOSUB-lausetta. Pääohjelma on ilmeisesti "pudonnut" aliohjelmaan. Tämä vältetään käyttämällä pääohjelmassa END-lausetta. Syynä saattaa olla myös haarautuminen suoraan aliohjelmaan.
- SN Syntaksivirhe. Syynä yleensä kirjoitusvirhe tai välimerkkivirhe.
- ST Merkkijonolauseke liian monimutkainen. Jaa lauseke pienempiin osa-alueisiin.
- TM Väärä tyyppi. Yritetty liittää merkkijonoja numeromuuttujan tai päinvastoin.
- UF Määrittelemätön funktio. FNA(X)-lause ilman DEF FNA(X)-lauseetta.
- UL Määrittelemätön rivi. Haarautumiskäskyn kohderiviä ei ole olemassa.

LIITE D

Trigonometriset funktiot

Suorakulmaisessa kolmiossa ABC

AB on kulman α viereinen sivu.

BC on kulman α vastainen sivu.

AC on hypotenuusa.

Kulman α SINI, KOSINI ja TANGENTTI määritellään seuraavasti:

$$\text{SIN } \alpha = \frac{\text{vastainen sivu}}{\text{hypotenuusa}}$$

$$\text{COS } \alpha = \frac{\text{viereinen sivu}}{\text{hypotenuusa}}$$

$$\text{TAN } \alpha = \frac{\text{vastainen sivu}}{\text{viereinen sivu}}$$

Kaikissa Basicin trigonometrisissä funktioissa oletetaan, että kulma ilmoitetaan radiaaneina. Radiaani on kulman mitta ympyrän yksikköinä. Ympyrässä on 360° tai 2π radiaania (π on kreikkalainen kirjain "pii", jonka arvo on vakio 3.1415926). Yksiköt voidaan siten muuntaa seuraavasti:

$$\text{Astetta}/(180/\pi) = \text{radiaania}$$

$$\text{Radiaania} \times (180/\pi) = \text{astetta}$$

Käänteisfunktio vastakkainen toiminto funktiolle. Esimerkiksi 1,5 radiaanin kulman tangentti on:

$$\text{TAN}(1.5) = 14.101419$$

Tangentin käänteisarvo on ATN. Sen avulla voidaan määrittää kulma, jonka tangentti tiedetään:

$$\text{ATN}(14.101419) = 1.5$$

Sinin ja kosinin käänteisarvot eivät ole Basic-kielen standardifunktioita, mutta ne voidaan saada ATN-funktion avulla seuraavasti:

$$\text{Sinin käänteisarvo} = \text{ATN}(X/\text{SQR}(-X \times X + 1))$$

$$\text{Kosinin käänteisarvo} = (\text{TN}(X)\text{SQR}(-X \times X + 1)) + 1.5708$$

LIITE E

Kirjoittimen käyttö

Dragon 32:n kanssa voidaan käyttää monia markkinoilla olevia kirjoittimia. Kirjoitin kytketään Dragonissa olevaan liitoskohtaan, joka on merkitty P.I/O. Tämä 20-napainen pin-liitäntä, joka lähettää signaaleja standardille "centronics"-tyyppiselle kirjoittimelle. Kytkeä tapahtuu seuraavasti:

Ylärivä (numerointi oikealta vasemmalle)		Alarivi
Pin 1	Kirjoitustiheys	Pin 2 +5 voltia
Pin 3	Databitti 0	Pin 4 +5 voltia
Pin 5	Databitti 1	Pin 6 0 voltia
Pin 7	Databitti 2	Pin 8 0 voltia
Pin 9	Databitti 3	Pin 10 0 voltia
Pin 11	Databitti 4	Pin 12 0 voltia
Pin 13	Databitti 5	Pin 14 0 voltia
Pin 15	Databitti 6	Pin 16 0 voltia
Pin 17	Databitti 7	Pin 18 0 voltia
Pin 19	ACK (negativ)	Pin 20 BUSY(negativ)

Seuraavia kirjoittimia on käytetty Dragonin kanssa:

Ambler 2400, Centronics 351, Epson MX80, OKI microline 80 ja 93, Seikosha GP100A ja GP250A.

LIITE F

Muistikartta (memory map)

Desimaaliosoite	Sisällys	Heksadesimaaliosoite
0-1023	Systeemin käytössä	0000-03-FF
255	"Direct page" RAM	00FF
1023	"Extended page" RAM	003F
1024-1535	Tekstiruutumuisti	0400-05FF
	Grafiikkaruutumuisti	
1536-3071	Sivu 1	0600-0BFF
3072-4607	Sivu 2	0C00-11FF
4608-6143	Sivu 3	1200-17FF
6144-7679	Sivu 4	1800-1DFF
7680-9215	Sivu 5	1E00-23FF
9216-10747	Sivu 6	2400-29FF
10748-12287	Sivu 7	2A00-2FFF
12288-13824	Sivu 8	3000-35FF
13824-32767	Ohjelma- ja muuttujamuisti	3600-7FFF
32768-49151	Basic-tulkki	8000-BFFF
49152-65279	Ohjelmamodulimuisti	C000-FEFF
65280-65375	Syöttö/tulostus -alue	FF00-FF5F
65376-65503	SAM kontrollibitit	FF60-FFDF
65504-65535	CPU-vektorit	FEE0-FFFF

HAKEMISTO

A-parametri DRAW-lauseessa	160, 166
Aakkostus	94
ABS-funktio	97
Algoritmi	39
Alimerkkijonon suorittaminen DRAW-lauseessa	160
Alimerkkijonon suorittaminen PLAY-lauseessa	152
Alimerkkijonot PLAY-lauseessa	148
Aliohjelma	85
Aliohjelma, rivinnumero	86
Aliohjelmakirjasto	87
Aliohjelmien kutsuminen	86
Aliohjelmarivien numerointi	87
Aloituskohhta DRAW-lauseessa	157
Ammuntapeli	120
Ampuma-ase (esimerkki)	123
AND IF-lauseessa	73
AND PUT-komennossa	164
AND-operaattori	73
Argumentit funktiossa	97-104
Aritmeettiset operaattorit	10
Arvojen tallentaminen muuttujaan	19
Arvon kopiointi muuttujaan	19
ASC-funktio	102
ASCII-koodit	135
Askelpalautin	7
Asteiden muuttaminen radiaaneiksi	193
Asteikko DRAW-lauseessa	159
ATN-funktio	97, 193
AUDIO OFF -lause	147, 154
AUDIO ON -lause	147, 154
AUX-liitäntä	51
B-parametri DRAW-lauseessa	166, 161
Basic	8
Basic-muuttujan osoitin	182
BREAK-näppäin	27
BREAK-näppäin, käyttö	69
C-asteikko	149
C-parametri DRAW-lauseessa	166
CHR\$-funktio	100, 186
CHR\$, graafinen merkki	113, 117
CIRCLE-lause	138, 144
CLEAR-lause	107, 111, 18
CLEAR-näppäin	7
Clementine (esimerkki)	152
CLOAD-komento	54, 63
CLOADM-komento	183
CLOSE-lause	176
CLS-lause	27, 44
COLOR-lause	135
CONT-komento	67, 68
COS-funktio	97, 193
COSIN1, määritelmä	193
CSAVE-komento	53, 63
D-parametri DRAW-lauseessa	158, 166
DATA-lause	106, 107

DATA-osoitin	107
DEF FN -lause	103, 108
DEF USRn -lause	182
DEL-komento	61
DIM-lause	92, 95
DIN-pistoke	51
DRAW ja LINE -komentojen yhdistäminen	162
DRAW-lause	157, 166
E-parametri DRAW-lauseessa	158, 166
EAR-liitäntä	51
EDIT, askelpalautin	57
EDIT, etsiminen	57
EDIT, H-komento	64
EDIT, K-komento	64
EDIT, kursorin liikkuminen rivillä	57
EDIT, lisääminen	58
EDIT, muutokset	58
EDIT, poistaminen	57
EDIT, rivin pidentäminen	57
EDIT-komento	56, 64
EDIT-tila, käyttö	56
EDIT-tilasta poistuminen	58, 64
Ehdollinen haarautuminen	69
Ehdollinen haarautuminen aliohjelmiin	87
Ehdon tutkiminen	79
Ehto	70
Ekspontiaalimuodossa kirjoittaminen, PRINT USING	173, 177
END-lause	68
ENTER-näppäin	8, 13
EOF-funktio	180, 181
Epätosi ehto	71
Etsiminen (tekstiesimerkki)	107b
Etsiminen EDIT-tilassa	57
EXEC-komento	97
EXP-funktio	87
F-parametri DRAW-lauseessa	158, 166
FIX-funktio	98
FOR-lause	81-83, 88
Funktionimet	96
Funktioryhmät	97
Funktiot	97-103
Funktiot, 1. ryhmä	97
Funktiot, 2. ryhmä	100
Funktiot, 3. ryhmä	101
Funktiot, 4. ryhmä	102
Funktiot, 5. ryhmä	103
Funktiot, sekafunktiot	102
Funktiotyypit	96
G-asteikko	149
G-parametri DRAW-lauseessa	158
G-parametri GET-lauseessa	163
GET-lause	165-166, 168
GOSUB-lause	86, 89
GOTO-lause	42, 69, 86
Grafiikka, käytettävissä olevat värit	113
Grafiikka, merkkijonojen käyttö	114
Grafiikkaruudukko	188-190
Grafiikkatila	128
Grafiikkatilan käyttö GET- ja PUT-lauseissa	168
Graafiset merkit, CHR\$	187

Graafiset sivut	128
Graafisten sivujen kopiointi	140, 146
Graafisten sivujen muuttaminen	140-142, 146
H-parametri DRAW-lauseessa	158
Haarautumiskäskyt	69
Haarautumiskäskyt aliohjelmiin	86
Haarautumiskäskyt silmukoissa	84
Heittomerkki REM-lauseen merkkinä	45
HEX\$-funktio	100
Hienografiikka	112, 127
Hienografiikkatila	127
I/O ERROR	53
IF-lause, ehto	71
IF-lause, lauseke	79
IF-lause, loogiset operaattorit	72
IF-lause, merkkijonot	72
IF-lause, suhde	72
IF-lause, suhdeoperaattorit	72
IF THEN ELSE -lause	71, 79
Indeksit, taulukoissa	91
INKEY\$, käyttö	73, 80
INKEY\$-funktio	73, 103
INPUT-lause	27, 35
INSTR-funktio	102
INT-funktio	98
Jakolaskuoperaattori	11
Janojen piirtäminen	130, 143
JOYSTK-funktio	98
JOYSTK-funktio, käyttö	121
Kaarien piirtäminen	140
Kaksiulotteinen taulukko	95
Kaksoispiste välimerkkinä	25, 28
Kasetti, tulostus	180
Kasettinauhalle tallentaminen	51-56
Kasettinauhuri, kaukosäätö	51
Kasettinauhuri, käyttö	4, 51
Kasettinauhuri, MOTOR-komento	154
Kasettinahurin liitännät	51
Kasettinahurin ohjaaminen	147
Kasettinahurityypit	51
Kaukosäätöinen käynnistyskytkin	51
Kenttätarkenteet	171
Kertolasku	10
Keskeytetyn ohjelman jatkaminen	67, 68
Konekieli	181
Konekielinen ohjelma	183
Konekielisen ohjelman suorittaminen	181
Konekielisen ohjelman tallentaminen	182
Kommenttien lisääminen ohjelmaan	40, 45
Konkatenaatio (merkkijonojen)	22
Korttien jakaminen (esimerkki)	107a
Korttipakan sekoittaminen (esimerkki)	94
Kromaattinen asteikko	150
Kromaattinen asteikko PLAY-komennossa	148
Kulman kierto DRAW-lauseessa	166
Kulmaparametri DRAW-lauseessa	160
Kursori	6
Kursorin käyttö EDIT-tilassa	57
Kuulokeliitettä	51

Kuvien piirtäminen	112
Kuvion väri	129
Käyttäjakohtaiset funktiot	103, 108
L-parametri DRAW-lauseessa	158
L-parametri PLAY-lauseessa	150
Lainausmerkki	15
Lajittelu taulukoiden avulla	94
Lajitteluesimerkki	95
Laskentasäännöt	9-14
Laskujärjestys	9, 10
Laskujärjestys, operaattorit	9
Laskujärjestyksen muuttaminen	13
Laskutoimitukset	9
Lauseke IF-lauseessa	72
Lauseke, merkkijono	32
Lauseke, muuttujan käyttö	32
LEFT\$-funktio	101
LEFT-parametri DRAW-lauseessa	158
LEN-funktio	102
Liike grafiikassa	140
Liike ohjausvivun avulla	121
Liikkuvat kuvat	116-119
LINE IN -liitäntä	51
LINE INPUT -komento	106, 109
LINE INPUT, käyttö	106
LINE-lause	130-133
LINE-lause DRAW-lauseen yhteydessä	162
Linna (esimerkki)	114
LIST-komento	30
LOG-funktio	98
Loogiset operaattorit	72
Luettelot	91
M-parametri DRAW-lauseessa	158
Matemaattiset lausekkeet	12
MEM-funktio	102
Merkin lisääminen EIDT-tilassa	58
Merkin muuttaminen EDIT-tilassa	58
Merkin poistaminen EDIT-tilassa	57
Merkkijonofunktiot	101
Merkkijonoissa käytettävät merkit	14
Merkkijonojen ja lukujen yhdistäminen	22
Merkkijonojen tallentaminen	15
Merkkijonomuistitilan varaaminen	107
Merkkijonomuuttuja lausekkeessa	19
Merkkijonomuuttujan nimi	22, 24
Merkkijonomuuttujat	17
Merkkijonomuuttujien tallentaminen	107
Merkkijonot IF-lauseessa	73
Merkkijonotaulukot	92
MID\$-funktio	101
Miinusmerkki DRAW-lauseessa	161
MOTOR OFF -lause	154
MOTOR ON -lause	154
Muistiosoite	182
Muistitilan varaaminen grafiikassa	127
Muistitilan varaaminen konekielisille ohjelmille	182
Muotojen värittäminen	137, 144
Muotolauseet, PRINT USING	171
Musiikki	148
Musiikkimerkkijonot	149

Muuttajat	17
Muuttujatyypit	17
Muuttujatyypit IF-lauseessa	72
Muuttujien arvot	19
Muuttujien nimet	18
N-parametri DRAW-lauseessa	166
Negatiivinen luku	10
NEW-lause	31
NEXT-lause	81,88
Nolla	7
Nopanheiton simulointi	48,50
NOT PUT-lauseessa	168
Numeeriset funktiot	97-98
Numeeriset merkit merkkijonossa	92
Numerolausekkeet	69
Numeromuuttujien nimet	18
Nuolinäppäimet	117, 118
Nuotin pituus	150
Nuotit	148
Näppäimistö	7
O OPEN-lauseessa	176
O:n ja nollan ero	7
O-parametri PLAY-lauseessa	153
Ohjausvivun näppäin	121
Ohjelma, määrittelmä	27
Ohjelma, suoritusjärjestys	27
Ohjelmakomennot	25
Ohjelman haarautuminen	69
Ohjelman kulun seuraaminen	67
Ohjelman laatiminen	38
Ohjelman lukeminen kasetilta	54
Ohjelman osat	48
Ohjelman pysäyttäminen	27,67
Ohjelman rakenne	85
Ohjelman rivit	25
Ohjelman suunnittelu	38
Ohjelmarivien tallentaminen	26
Ohjelmarivien toistaminen	81
Ohjelmarivien uudelleennumerointi	61-62,66
Ohjelmarivin lisääminen (muutokset)	37
Ohjelmarivin muuttaminen	37,57
Ohjelmarivin poistaminen	61,65
Ohjelmien tallentaminen kasetille	53
Ohjelmointiesimerkki	48-50
Oikea ohjelmaversio	49
OK-ilmoitus	6
Oktaavi	153
ON GOSUB -lause	86-87,89
ON GOTO -lause	69,78
OPEN-lause	176
Opetusohjelma	73
OR-operaattori	73
OR PUT-lauseessa	168
P-parametri PLAY-lauseessa	151
PAINT-käsky DRAW-lauseessa	162
PAINT-lause	137, 144
PCLEAR-lause	127, 133
PCLS-lause	132, 135
PCOPY-lause	142

PEEK-funktio	198
PEEK-komento, käyttö	123
Pienet kirjaimet	7
Piirtäminen	157
Pisteellä merkitty nuotti	150
Pisteet hienografiikassa	128, 134
Pisteet TV-ruudulla	112
Pistokeliitännät	51
PLAY, käyttö soitettaessa	155
PLAY-lause	148, 153, 155
PLAY-parametri	148
PLAY-näppäin kasettisoittimessa	53
Plus-merkki DRAW-lauseessa	168
Plus-merkki merkkijonoissa	23
PMODE-lause	129
POINT-funktio	98
POINT-komento, käyttö	124
POKE-komento	182
POS-funktio	99
Potenssiin korottaminen	9
PPOINT-funktio	99
PPOINT-komento	129
PRESET-lause	131
PRESET DRAW-lauseessa	162
PRESET PUT-lauseessa	164
PRINT	8
PRINT@ grafiikassa	113-123
PRINT@ -lause, ruudukko	188
PRINT-lause	25, 33
PRINT-lause STOP-käskyn jälkeen	67
PRINT USING -lause	174, 177
PRINT USING# -1 -lause	176
PRINT USING# -2 -lause	178
PRINT USING, merkkijonot	177
PSET-käsky DRAW-lauseessa	162
PSET-lause	131, 136
PSET PUT-lauseessa	164
Puoligrafiikka	112
Puoligrafiikkaruudukko	189
Puolipiste DRAW-lauseessa	158
Puolipiste PLAY-lauseessa	150
PUT-lause	164, 166
PUT-lause, parametrit	164, 168
Pystysuora liike	117
Päätteet PLAY-lauseessa	153
R-parametri DRAW-lauseessa	158
Radiaanien käyttö funktioissa	105
Radiaanien muuttaminen asteiksi	193
Radiaanit	97, 100
Radiaanit, määritelmä	193
Raketin piirtäminen (esimerkki)	159, 163
READ-lause	106, 109
RECORD-painike kasettinahurissa	53
REM-lause	45
RENUM-komento	61, 66
RESET-lause	119, 126
RESTORE-lause	107, 109
RETURN-lause	86, 89
RIGHT\$-funktio	101
Rivien numeroiminen	158

Rivin kirjoittaminen EDIT-tilassa	58
Rivinumero	26
Rivinumero, askelväli	61-62,66
Rivinumero ON GOTO -lauseessa	69
Rivin pituus	72
RND-funktio	28
RND, käyttö	28
RUN-komento	31
Rutiinin osoite	183
Ruutu, tyyppi	134
Ruutu, värit	28
S-parametri DRAW-lauseessa	159
SCREEN-lause	128,134
SET-lause	119,125
SGN-funktio	99
SHIFT-näppäin	7
Siirtyminen viivaa piirtämättä DRAW-lauseessa	161,166
Sijoituslause	32
Silmukan hidastaminen	82
Silmukat	81,88
Silmukat, sisäkkäiset	83
Silmukkalaskuri	82
SIN-funktio	99
SINI, määritelmä	193
Sisäkkäiset silmukat	83,84
Sivujen selaaminen	140
SKIPF-komento	53,63
SKIPF, käyttö	179
Soikioiden piirtäminen	138-140,144
SOUND-lause	29,41
SQR-funktio	99
STEP FOR...NEXT-lauseessa	82
STEP, poisjättäminen	82
STOP-lause	67,68
STRING\$-funktio	101
STR\$-funktio	100
Suhde IF-lauseessa	72
Suhteellinen siirtyminen DRAW -lauseessa	167
Sulkumerkkien käyttö	13
Suorakulmioiden piirtäminen	131
Syntaksivirhe	8
Systeemikomentoja	60
Sävelasteikot	149
Sävelkulkujen toistaminen musiikkiohjelmissa	152
Sävelmä, Clementine	151
Sävelmä, Greensleeves	154
Sävelmä, Lavender Blue	152
T-parametri PLAY-lauseessa	151
Talon piirtäminen (esimerkki)	132
TAN-funktio	100,193
Tangentti, määritelmä	193
TAPE-liitântä	51
Tauko musiikkimerkkijonossa	151
Taulukko, indeksi	92
Taulukko, koko GET-lauseessa	164
Taulukko, käyttö	92-94
Taulukko, käyttö GET-lauseessa	162
Taulukko, merkkijono	92
Taulukko, numeerinen	92

Taulukkomuuttujan nimeäminen	92
Taulukon koko	91
Taustan väri	126, 136
Tekstiruutu	127
Tekstiruutuarkki	188
Tempo	151
Tiedon lukeminen ohjelmasta	107, 109
Tiedon siirto kasetille	175
Tiedon tallentaminen kasetille	175
Tiedon tallentaminen ohjelmaan	107
Tiedostojen lukeminen	180
Tiedoston nimi	53, 176
Tiedoston nimi, kasettinauhalla	53
Tiedoston sulkeminen	176
TIMER-funktio	103
Toimitusohjelma (EDITOR)	56-60
Tosi ehto	71
Trigonometriset funktiot	97-100, 193
TROFF-komento	67, 68
TRON-komento	67, 68
TV-ruutu	6, 112
Tyhjä merkki merkkijonossa	14
Tyhjä merkki PRINT USING -lauseessa	175
U-parametri DRAW-lauseessa	158
Useampien ohjelmien tallentaminen	55
USRn-funktio	182
Vaakaliike	117
Vakiot	17
VAL-funktio	102
Valemuuttuja DEF FN -lauseessa	103
Valinta	69
VARPTR-funktio	182
Video-RAM	127
Vihjeitä tallentamista varten	55
Viivan väri DRAW-lauseessa	159
Viivästetty toiminta	7
Vinoviivan piirtäminen DRAW-lauseessa	158, 166
Virhe, selitykset	191
Virheen korjaaminen ohjelmassa	40, 56
Virheen korjaaminen rivillä	7
Virhekoodit	191
Vähennyslasku	11
Välilyönti Basic-lauseessa	14
Välilyöntinäppäin, käyttö EDIT-tilassa	57, 64
Välitön toiminta	7
Värien säätäminen TV-vastaanottimessa	113
Värikoodit, CHR\$	187
Värityskomento DRAW-lauseessa	160, 166
Värivalikoima	128, 134
X-parametri DRAW-lauseessa	160
X-parametri PLAY-lauseessa	152
x,y-piste	125
Yhteenlasku	11
Yksinkertaiset muuttujat	17, 18, 24
Ympyrät	138
Äänen lisääminen grafiikkaan	147
Äänen voimakkuuden säätäminen kasettinauhurissa	52, 54
Äänitehosteet	147

Tekstiruutu	127
Tempo	151
Nopanheiton simulointi	48,50
Ehdon tutkiminen	79
Sijoituslause	32
TIMER-funktio	103
Kaksiulotteinen taulukko	95
Trigonometriset funktiot	97-100,193
TROFF-komento	67,68
TRON-komento	67,68
TV-ruutu	6,112
U-parametri DRAW-lauseessa	158
Opetusohjelma	73
Epätosi ehto	71
USRn-funktio	182
Tiedon siirto kasetille	175
Alimerkkijonon suorittaminen DRAW-lauseessa	160
Alimerkkijonon suorittaminen PLAY-lauseessa	152
Lauseke IF-lauseessa	72
Lauseke, muuttujan käyttö	32
Lauseke, merkkijono	32
VAL-funktio	102
Valinta	69
Muuttujien nimet	18
Muuttujatyypit	17
Muuttujatyypit IF-lauseessa	72
Muuttujat	17
Muuttujien arvot	19
VARPTR-funktio	182
Pystysuora liike	117
Video-RAM	127
Ehto	70
Kulmaparametri DRAW-lauseessa	160
Äänvoimakkuuden säätäminen kasettinauhurissa	52,54
x,y -piste	125
X-parametri DRAW-lauseessa	160
X-parametri PLAY-lauseessa	152

© **FREMICS** OY

KIERRETIE 20 01650 VANTAA 65

Helisten-offset, Hki 1984

www.DragonData.co.uk