

SALORA **Manager**



**BASIC-
KÄSIKIRJA**

SALORA MANAGER KOTITIETOKONE



BASIC-KÄSIKIRJA

JOHDANTO

ALOITA TÄSTÄ, JOS ET TIEDÄ MITÄÄN TIETOKONEISTA

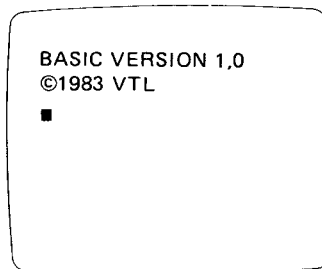
Jos et tiedä eroa BASIC-kielten välillä, tämä kirja on sinua varten. Tai jos olet yksi tuhansista, joka on yrittänyt oppia BASIC-kielen joskus aikaisemmin mutta epäonnistunut surkeasti, tämä kirja selvittää asiaa jonkin verran.

MITÄ VALMISTELUJA SINUN PITÄÄ TEHDÄ, ENNENKUIN LUET TÄSTÄ ETEENPÄIN

Tämä kirja on kehitetty erikoisesti Salora Manageria varten. Kun luet tätä kirjaa, sinun pitäisi olla jo

- 1) purkanut Salora Manager-tietokoneen laatikosta
- 2) liittänyt sen televisioosi
- 3) yhdistänyt sähköverkkoon sekä television että Salora Managerin
- 4) kytkenyt virran molempiin koneisiin
- 5) varannut yhden TV:n ohjelmanvalintapainikkeista tietokonekäyttöön ja virittänyt television Salora Manageria varten

... ja nyt television kuvaruudun pitäisi näyttää tältä:



Kaikki se informaatio, jonka tarvitset päästäksesi tähän vaiheeseen, löytyy PIKAOPPAASTA. Joten, jos et ole kytkenyt vielä tietokonetta televisioon, tee se nyt. Jos olet, jatka lukemista.

OIKEA ASENNE ALOITETTAESSA

Tämä kirja opettaa sinulle MICROSOFT* BASICin laajennetun version. Vaikka se on helppo oppia ja sitä on helppo käyttää, on hetkiä, jolloin tunnet olevasi maailman tyhmin ihminen. Kun näin käy, huolimatta siitä kuinka tyhmäksi tunnet itsesi, on hyvä muistaa tärkeä asia:

ÄLYKKÄINKIN TIETOKONE ON VIELÄ TYHMEMPI KUIN SINÄ OLET.

* Microsoft on Microsoft Inc:n rekisteröity tavaramerkki

SISÄLLYSLUETTELO

LUKU 1

Tunne tietokoneesi

— Ensimmäinen tietokoneohjelmasi	10
— Asiaa, jota Sinun ei tarvitse tietää	12
— Asiaa, joka Sinun on tiedettävä	13
— Ohjelmointiteoriaa	19
— Ohjelmointikäytäntöä	21
— Pelätty syntaksivirhe	23
— Ruutueditointi	25
— BASIC-ohjelmoinnin kymmenen käskyä	28

LUKU 2

Managerin BASIC-käskyt

Grafiikkakäskyt

— GR	30
— TEXT	31
— COLOR =	32
— PLOT	33
— UNPLOT	36
— CIRCLE	37
— RECT	38
— HOME	38
— VPEEK	39
— VPOKE	40
— VIDEO-RAM TEKSTITILASSA	40
— VIDEO-RAM GRAFIIKKATILASSA	42

LUKU 3

Managerin BASIC-komennot

Helpot komennot

— END	44
— NEW	45
— LET	46
— REM	47
— PRINT	48
— INPUT	49
— LIST	50
— STOP	51
— CONT	52

— BREAK	53
— TRACE ja NOTRACE	54
— Tehokkaat oikotiet	55

LUKU 4

Managerin BASIC-komennot

Fiksujen komennot

— IF...THEN	58
— AND, OR ja NOT	59
— FOR..TO..STEP..NEXT	60
— READ...DATA	61
— RESTORE	62
— GOTO	63
— GOSUB...RETURN	64
— ON...GOTO/ON...GOSUB	65
— CLEAR	66
— SPC ja TAB	66
— Piikun ja puolipisteen käyttö PRINT-käskyssä	67

LUKU 5

Managerin BASIC-komennot

Ääni- ja musiikkikäskyt

— Miksi tietokoneessa on ääni	70
— SOUND	70
— SGEN	72

LUKU 6

Epämiellyttävä aritmetiikka, jonka toivoit välttäväsi

Vakiot, numeeriset muuttujat ja kokonaisluvut	74
— Mitä ovat operaattorit?	75
Aritmeettiset operaattorit	75
Loogiset operaattorit	76
Relaatio-operaattorit	76
— Mikä on funktio?	
ABS	77
ATN	77
COS	77
EXP	77
INT	77
LOG	77
RND	77
SGN	78
SIN	78

SQR	78
TAN	78

LUKU 7

Kamat ja käsittämättömät merkkijonot

— Merkkijonofunktiot ja -muuttujat	80
— Merkkijonofunktiot	80
ASC	80
CHR\$	80
GET	80
LEFT\$	80
LEN	80
MID\$	80
RIGHT\$	80
STR\$	81
VAL	81
— Merkkijonojen vertailuoperaattorit	82

LUKU 8

Managerin BASIC-komennot

Kovat käskyt

— DIM	84
— PEEK	84
— POKE	85
— CALL	85
— RAM (0)	88
— &	88
— GET	89
— ONERROR GOTO	90

LUKU 9

Managerin BASIC-komennot

Kasettinauhuriin liittyvät komennot

— Mitä voit tehdä nauhurilla?	92
— CLOAD	92
— CSAVE	93
— CRUN	93
— BLOAD	93
— BSAVE	94
— BRUN	94
— STORE	95
— RECALL	95

LUKU 10

Managerin BASIC-komennot

Kirjoitinkomennot

— LLIST	98
— LPRINT	99

LUKU 11

— Virheiden numerokoodit	102
— Virheilmoitukset ja miksi ne kiusaavat sinua	103
— ASCII-kooditaulukko	109
— Yhteenveto BASIC-komennoista	111
— Käsitteitä ja termejä	114

LIITE 1

— Aakkosellinen hakemisto	127
---------------------------------	-----

LIITE 2

— Heksadesimaali-desimaalimuunnostaulukko	133
---	-----

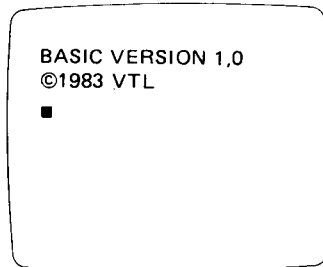
LUKU1

TUNNE TIETOKONEESI

- ENSIMMÄINEN TIETOKONEOHJELMASI
- ASIAA, JOTA SINUN EI TARVITSE TIETÄÄ
- ASIAA, JOKA SINUN ON TIEDETTÄVÄ
- OHJELMOINTITEORIAA
- OHJELMOINTIKÄYTÄNTÖÄ
- PELÄTTY SYNTAKSIVIRHE
- RUUTUEEDITOINTI
- BASIC-OHJELMOINNIN KYMMENEN KÄSKYÄ

ENSIMMÄINEN TIETOKONEOHJELMASI

Mennään suoraan asiaan. Kuten johdannossa mainittiin, kaikki liitännät pitäisi olla tehtynä ja virran kytkettynä, jotta ollaan valmiita jatkamaan. Kuvaruudun pitäisi näyttää tältä:



Kirjoita nyt seuraava ohjelma täsmälleen samanlaisena kuin se on alla. Älä tee virheitä. Älä jätä pois ainuttakaan pilkkua. Älä jätä pois ainuttakaan rivin alussa olevista numeroista. Älä lisää mitään.

Kun näet RETURN-merkin, älä kirjoita sitä. Paina vain Managerin näppäimistöllä olevaa näppäintä, jolla lukee "RETURN". Tämä siirtää pienen vilkkuvan neliön alas, jotta voit aloittaa seuraavan rivin.

Kaikki valmiina? Aloita kirjoittaminen:

```
10 GR RETURN
20 COLOR = 1, 3 RETURN
30 CIRCLE (75, 50), 20 RETURN
40 CIRCLE (175, 50), 20 RETURN
50 PLOT 70, 145 TO 75, 150 RETURN
60 PLOT 180, 145 TO 175, 150 RETURN
70 FOR X = 75 TO 175 STEP 1 RETURN
80 PLOT X, 150 RETURN
90 NEXT X RETURN
100 CIRCLE (85, 60), 5 RETURN
110 CIRCLE (165, 60), 5 RETURN
120 END RETURN
```

Siinä kaikki. Olet juuri kirjoittanut ohjelman. Ei kovinkaan vaikeaa loppujen lopuksi, vai kuinka? Mutta nyt tulee pahin. Ajetaan se ja katsotaan mitä tapahtuu. Jos olet tehnyt kirjoitusvirheen, niin nyt se paljastuu.

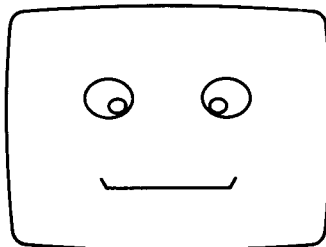
Valmis? Kirjoita:

RUN RETURN

Ruudulle pitäisi ilmestyä vihreä pohjaväri, jolle mustat viivat piirtävät iloiset kasvot. Jos näet tällaisen, olet tehnyt kaikki oikein.

Saadaksesi kuvan häviämään, paina näppäintä, jolla lukee RESET. Tämän avulla saat jälleen vihreät kirjaimet mustalla pohjalla ja ohjelmasi tulostuu siististi ruudulle. Tallettaaksesi ohjelman kasettinauhalle katso ohjeita tästä kirjasta CSAVE-komennon kohdalta.

Saadaksesi ohjelman häviämään kirjoita NEW tai katkaise virta Managerista ja kytke virta uudelleen.



Jollet ole vielä nähnyt iloisia kasvoja, johtuu se tekemästäsi virheestä. Paina RESET-näppäintä ja kirjoita LIST. Vertaa ruudulla näkyviä ohjelmarivejäsi kirjassa oleviin ohjelmariveihin. Varmistu, että ne ovat täsmälleen samat. Et näe RETURNia ruudulla koska et kirjoittanut sitä. Painoit vain RETURN-näppäintä.

Jollet huomaa selviä virheitä, katkaise virta Managerista ja kytke virta uudelleen. Aloita sitten aivan ohjelman alusta.

Kun olet menestyksellisesti nähnyt iloiset kasvot, jatka lukemista ja ymmärrät, miten ne tehtiin.

ASIAA, JOTA SINUN EI TARVITSE TIETÄÄ

Miksi Sinun ei tarvitse tietää tätä asiaa?

Kaikki tässä oleva informaatio on oleellisesti tarpeetonta. Se ei auta sinua vähimmässäkään määrin opiskellessasi BASICia tai opetellessasi kirjoittamaan ohjelmia. Se on tässä mukana vain, jotta voit tehdä vaikutuksen perheeseesi ja ystäviisi laajoilla tiedoillasi tietokone-epäoleellisuuksista.

MIKÄ TIETOKONE ON?

Ei tarvita hienoja kaavioita tai selityksiä. Tietokone on se pieni näppäimistöllä varustettu kapine, joka sinulla pitäisi olla edessäsi.

MIKÄ BASIC ON?

BASIC on lyhenne sanoista Beginner's All-purpose Symbolic Instruction Code (aloittelijoiden kaikkiin tarkoituksiin sopiva symbolinen ohjelmointikieli). Sen kehittivät Dartmouth Collegessa professorit John G. Kemeny ja Thomas E. Kurtz vuonna 1964 tehdäkseen ohjelmoinnin mahdolliseksi ilman vaikean binäärisen tai konekielisen ohjelmoinnin oppimista. Joten jos löydät visaisia BASIC-ongelmia, tiedät, keitä voit syyttää.

MIKÄ ON MS-BASIC?

BASIC on laajimmin levinnyt ohjelmointikieli maailmassa. Nykyisin on lukemattomia BASIC-murteita. MS-BASIC on BASIC-versio, jonka on luonut Microsoft, suuri amerikkalainen ohjelmistotalo. Se on suosituin mikrotietokoneiden BASIC ja kun opit sen Salora Manageria varten, kykenet ohjelmoimaan lähes kaikkia maailmalla olevia mikrotietokoneita.

JOS TUNNET JO MS- BASICIN

Managerin käyttämää MS-BASICia on Salora laajentanut yhdessä Video Technologyn kanssa. Grafiikkaa ja äänen muodostamista varten on lisätty joukko erityiskäskyjä.

Jos tunnet jo MS-BASICin, siirry yksinkertaisesti sivulle, jolta lisäkomennot alkavat.

ASIAA, JOKA SINUN ON TIEDETTÄVÄ

Miksi Sinun pitää tietää tämä asia?

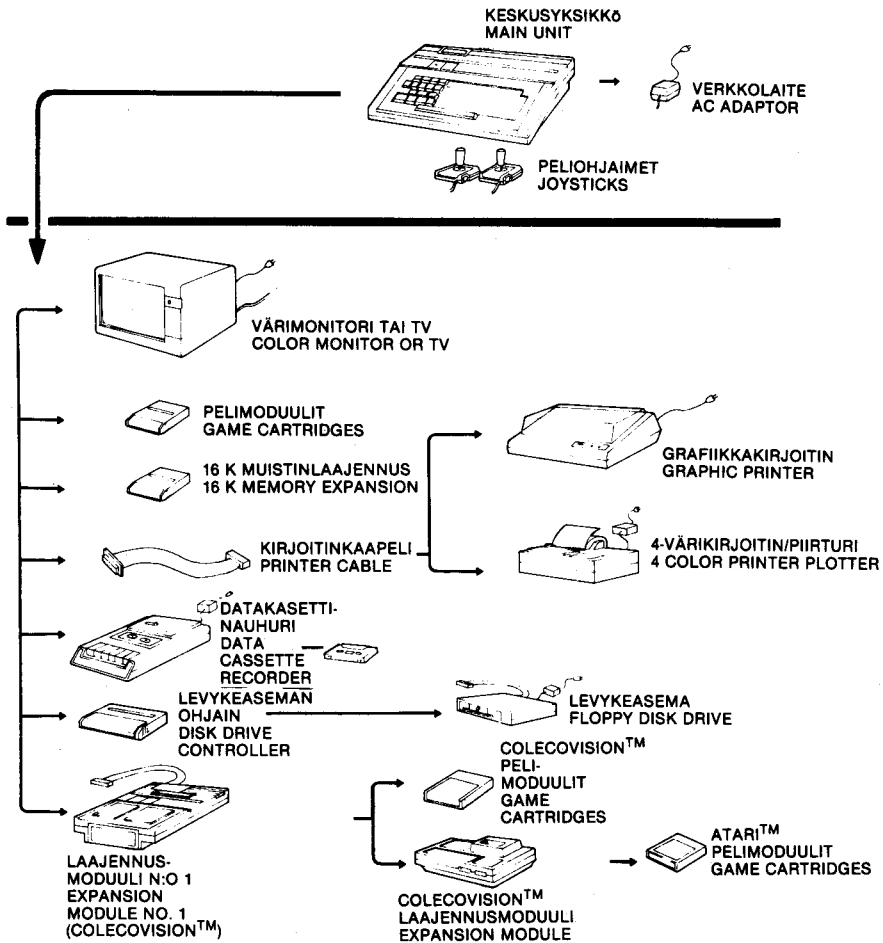
Kaikki tässä luvussa oleva asia pitää ymmärtää ennenkuin voit jatkaa seuraaviin osiin. Älä yritä kiirehtiä ja aloittaa laajoja ohjelmia, jollet ole läpikäynyt kaikkea tässä olevaa. Muuten ei kaikki yrittämäsi onnistu ja turhaudut. Vihastut ja mahdollisesti menetät mielenkiinnon ohjelmointiin. Silloin et milloinkaan saa tietää mikä suuri ohjelmoiija sinusta olisi tullut.

AVAIMET MAAILMANKAIKKEUTEEN

Salora Managerin näppäimistö on tehty niin yksinkertaiseksi ja helppokäyttöiseksi kuin mahdollista. Mutta on eräitä tärkeitä asioita, jotka sinun pitää oppia sen käytöstä, jos haluat tehdä sillä muutakin kuin katsoa sitä. Kun olet oppinut temput, tiedät, mitä kaikkea voit tehdä sillä.

MIKSI ET TARVITSE KAAVIOTA

Jos muistat, sinulla oletetaan olevan Salora Manager käyttövalmiina edessäsi. Joten viimeinen asia maailmassa, jota tarvitset, on kaavio, eikö totta?



USEIMMILLA NÄPPÄIMILLÄ ON KOLME ERI TOIMINTOA

Managerissa on 49 eri muotoista ja kokoista näppäintä. Ne ovat kaikki automaattisella toistolla toimivia. Se tarkoittaa, että jos pidät näppäintä alhaalla, tulostuu niissä oleva merkki ruudulle kunnes lopetat näppäimen painamisen. Useimmat näistä 49 näppäimestä toimivat kolmella eri tavalla:

1) Painamalla näppäintä

Jokaiselle näppäimelle on painettuna iso musta kirjain, numero tai sana. Jos painat mitä tahansa näppäintä koskettamatta muuta näppäintä, näppäin tulostaa isoa, mustalla painettua. Jos esimerkiksi painat näppäintä jolla on merkki 1, numero "1" ilmestyy ruudulle. Jos painat A:lla merkittyä näppäintä, iso "A" ilmestyy kuvaruudulle.

2) Yhdessä SHIFT-näppäimen kanssa

18 näppäimellä — koko ylärivillä ja oikeassa reunassa olevilla kuudella näppäimellä — on painettuna pieni merkki näppäimen yläosaan ison mustan merkin lisäksi. Saadaksesi tämän merkin ilmestymään kuvaruudulle sinun pitää painaa toista SHIFT-näppäintä samaan aikaan kun painat näppäintä, jolla on tällainen pieni merkki. Silloin tämä pieni merkki ilmestyy ruudulle. Jos esimerkiksi painat SHIFT-näppäintä ja 1-näppäintä saman aikaisesti, "!" ilmestyy kuvaruudulle.

Jos painat SHIFT-näppäintä ja mitä tahansa kirjainnäppäintä, A:sta Ö:hön, ruudulle tulostuu pieni kirjain. Jos esimerkiksi painat SHIFT-näppäintä ja A-näppäintä samanaikaisesti, ruudulla näkyy pieni "a"-kirjain.

Älä käytä pieniä kirjaimia ohjelmointiin. Salora Manager ymmärtää vain isoja kirjaimia BASIC-ohjelmissa ja saatat tuhjata paljon aikaa ja sinun on tehtävä kaikki uudelleen. Siksi Manager tulostaa vain isoja kirjaimia, jollet erityisesti halua käyttää pieniä kirjaimia.

SHIFT-näppäimet on helppo löytää. Ne ovat niin tärkeitä, että niitä on kaksi kappaletta, yksi näppäimistön alimman rivin molemmissa päissä. Jotta asia olisi vielä helpompaa, niissä lukee "SHIFT".

3) CTRL-näppäimen kanssa

Tämä näppäin aktivoi näppäimien kolmannen toiminnon. Sen avulla voit syöttää BASIC-komentoja yhdellä painalluksella. Tämä voi säästää tunteja kirjoittamisessa ja grafiikka ja musiikkikäskyjen ollessa kyseessä, myös tunteja ohjelmoinnissa.

Näet BASIC-komennot kirjoitettuina jokaisen näppäimen yläpuolella. Esimerkiksi Q-näppäimen yläpuolella on "CSAVE"-komento.

Kirjoittaaksesi nämä BASIC-komennot näppäimeltä sinun pitää painaa CTRL- näppäintä ja halutun komennon sisältävää näppäintä samanaikaisesti. Syöttääksesi esimerkiksi BASIC-komennon CSAVE sinun pitää painaa sekä CTRL- että Q-näppäintä samaan aikaan. Komento CSAVE näkyy ruudulla.

Tietysti voit halutessasi kirjoittaa komennon CSAVE myös näppäimistöltä, kirjain kerrallaan. Mutta CTRL-näppäimellä aikaansaatu yhden painalluksen komento säästää sekä aikaa että vaivaa ja varmistaa, ettet tee kirjoitusvirheitä.

CTRL tarkoittaa kontrollia (CONTROL). CTRL-näppäin sijaitsee Manager-näppäimistön kolmannen näppäinrivin vasemmassa reunassa. Se löytyy helposti, sillä sana CTRL on kirjoitettu siihen.

MUUT VALTAKUNNAN AVAIMET

RETURN-näppäin

Tämä näppäin sijaitsee Managerin toisen näppäinrivin oikeassa reunassa. Siinä lukee RETURN.

RETURN-näppäimen avulla ilmoitetaan tietokoneelle että olet lopettanut rivin kirjoittamisen ja haluat aloittaa uuden rivin. Jollet paina RETURN-näppäintä halutessasi lopettaa rivin, tietokone yhdistää sen seuraavaan riviin ja mahdollisesti joudut kirjoittamaan molemmat rivit uudelleen.

RETURN tarkoittaa vaunun palautusta (CARRIAGE RETURN) ja on lyhenne, joka on peräisin vanhasta konekirjoitustekniikasta. Silloin tällöin lukiessasi BASIC-ohjelmakirjoja näet symbolin CR. Kirjoissa, joissa halutaan välttää miellelyhtymää kirjoituskoneeseen, näet merkinnän ENTER. Älä murehdi kumpaakaan näistä. Ne molemmat tarkoittavat yksinkertaisesti RETURNia.

VÄLILYÖNTI-näppäin

Tämä on alhaalla näppäimistössä oleva iso pitkä näppäin. Se muodostaa tyhjän välilyönnin kirjoittamiesi sanojen välille. Muuten tuloksena olisi yksi pitkä sana, jonka lukemisessa jopa sinulla itselläsi olisi vaikeuksia.

VÄLILYÖNTI-näppäin tulostaa tyhjän välilyönnin ruudulle joko painaessasi sitä tai kun painat sitä samanaikaisesti SHIFT-näppäimen kanssa.

RESET-näppäin

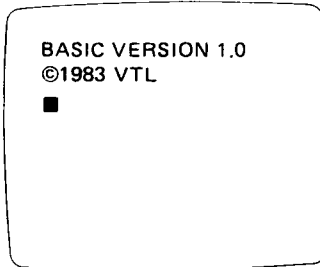
Tämän näppäimen kanssa tulet hyvin tutuksi. Sen avulla tietokone keskeyttää ikuiset silmukat ja ratkaisemattomat ongelmat tuhoamatta muistissa olevaa ohjelmaa.

RESET-näppäin sijaitsee väppäimistön yläosassa, aivan virran merkkivälön vieressä. Etsi neliönmuotoinen näppäin, jossa lukee RESET ja siinä se on.

MIKÄ VILKKUVA PIENI NELIÖ OIKEIN ON?

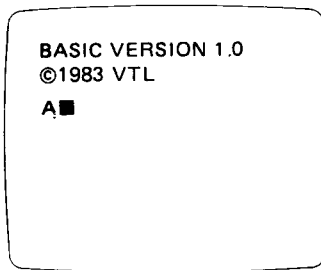
Jos olet kokeillut Salora Managerin eri näppäimiä, tv-kuvaruutu on ilmeisesti täyttynyt oudoilla sanoilla ja symboleilla.

Päästäksesi eroon tästä kaameasta sekasotkusta katkaise virta Managerista. Sitten kytke virta uudelleen. (Katkaise mielummin virta tietokoneesta kuin televisiosta, koska muuten tietokone muistaa kaiken roskan, jonka olet syöttänyt ja yksinkertaisesti tulostaa sen uudelleen kuvaruudulle.) Kun olet päässyt eroon kaikesta tarpeettomasta, joka oli ruudulla, pitäisi kuvaruudun näyttää tältä:



Vilkkuvaa valkoista neliötä kutsutaan KURSORIksi. Tälle nimelle ei ole mitään erityistä syytä, mutta se on yleisesti käytössä, joten yritä muistaa se. Joissakin yhteyksissä kursorista käytetään myös suomenkielisempää nimitystä KOHDISTIN. Kursori ja kohdistin tarkoittavat täsmälleen samaa asiaa.

KURSORin tarkoituksena on yksinkertaisesti osoittaa missä kohtaa ruutua työskentelet. Jos kirjoitat kirjaimen "A", niin se ilmestyy ruudulle täsmälleen siihen kohtaan, missä KURSORI on juuri nyt. Sitten KURSORI siirtyy yhden position eteenpäin osoittamaan paikkaa, johon seuraava merkki ilmestyy. Kuvaruutu näyttää nyt tällaiselta.



MITEN KURSORIA SIIRRELLÄÄN?

Siirtääksesi kursoria ruudulla vasemmalle tai oikealle paina CTRL-näppäintä ja nuolinäppäimiä ← tai → (Ä ja Å).

Jotta kursori siirtyisi ylöspäin, paina CTRL-näppäintä ja ↑ -näppäintä samanaikaisesti (L).

Jotta kursori siirtyisi alaspäin, paina CTRL-näppäintä ja ↓ -näppäintä samanaikaisesti (Ö).

Kokeile. Me odotamme.

KURSORI ON MYÖS TIEDOTTAJA

Kun ajat ohjelmia, vilkkuva kursori ilmoittaa, että tietokone on suorittanut ohjelman ja on nyt valmis ottamaan vastaan uusia ohjeita. Et voi syöttää mitään ennenkuin vilkkuva kursori ilmestyy näkyviin, koska tietokone voi tehdä vain yhtä asiaa kerrallaan.

OHJELMOINTITEORIAA

Pidä aina mielessäsi, että tietokone on tyhmä. Salora Manager on fiksumpi kuin useimmat, mutta siitä huolimatta kuka tahansa kaksi- tai kolmevuotias lapsi voi omatoimisesti ajatella ja kehittää toimintasarjoja, joihin kalliimmatkaan tietokoneet eivät pysty.

Asia on niin, että tietokoneet eivät opi mitään kokemuksesta tai päättelämällä. Ne eivät tiedä mitään. Sinun on kerrottava niille kaikki. Joten kirjoittaessasi mitä tahansa ohjelmaa, älä pidä mitään itsestään selvänä.

Kuvittele, että sinulla on typerä hovimestari

Jos sinulla olisi hyvin tyhmä hovimestari ja haluaisit hänen hakevan sanomalehden ulkona olevasta postilaatikosta ja tuovan sen sinulle, sinun pitäisi suunnitella asiat hyvin huolellisesti.

Ohjeesi hovimestarille saattaisivat näyttää seuraavilta.

- Vaihe 1: Älä tee mitään ennenkuin sanon "TEE"
- Vaihe 2: Mene ulko-ovelle
- Vaihe 3: Avaa ovi
- Vaihe 4: Mene ulos
- Vaihe 5: Sulje ovi
- Vaihe 6: Mene postilaatikoille
- Vaihe 7: Avaa postilaatikko
- Vaihe 8: Ota sanomalehti
- Vaihe 9: Sulje postilaatikko
- Vaihe 10: Käänny ympäri
- Vaihe 11: Tule takaisin taloon
- Vaihe 12: Avaa ovi
- Vaihe 13: Tule sisälle
- Vaihe 14: Sulje ovi
- Vaihe 15: Tule takaisin olohuoneeseen
- Vaihe 16: Anna sanomalehti minulle
- Vaihe 17: Mene pois ja odota, kunnes kutsun sinua uudelleen
- Vaihe 18: "TEE"

Tämä käskyjoukko olisi ohjelma. Sinun on tärkeää ottaa vaihe 1 mukaan, muuten typerä hovimestarisi juoksisi ulko-ovelle heti kun olisit antanut

ensimmäisen käskysi ja seisoisi siellä odottaen lisäohjeita. On myös tärkeää muistaa vaihe 18. Muuten tyhmä hovimestarisi vain seisoisi katsellen sinua.

Sinun pitää huolehtia hyvin eri vaiheiden järjestyksestä. Jos käskisit hovimestarin avata oven ja sitten sulkea oven, hän ei menisi ulos. Hän vain seisoisi siellä ja avaisi ja sulkisi oven. Joten kun seuraavassa vaiheessa käsket hänen mennä postilaatikoille, hän kävelisi oven läpi.

Tietokoneet ovat vielä tyhmempiä kuin tyhmä hovimestarisi. Jos antaisit nämä ohjeet tietokoneelle, sinun pitäisi myös kertoa miten kävellään. Sinun pitäisi sanoa "Vasen jalka eteenpäin, oikea jalka eteenpäin" koko matkan postilaatikoille. Sinun pitäisi käskää sitä liikuttamaan kättään, kääntämään ovenkahvaa ja muuta vastaavaa, sillä tietokone unohtaa kaiken minkä se on tiennyt joka kerran kun aloitat uuden ohjelman.

Sellaisten asioiden suorittamiseen kuin sanomalehden hakemiseen sinun pitäisi suunnitella tuhansia vaiheita, jotka tietokone suorittaisi. Vain yksi erehdys, ja voit lähettää sen Etelänavalle sanomalehden kanssa. Joten monessa tapauksessa sinusta on helpompaa noutaa itse sanomalehti.

TÄRKEÄTÄ MUISTETTAVAA OHJELMOINNISTA

- 1) Suunnittele joka vaihe etukäteen
- 2) Varmistaudu siitä, että tietokone suorittaa ne oikeassa järjestyksessä.

OHJELMOINTIKÄYTÄNTÖÄ

Sinun pitää oppia vielä lisää tätä tyypäisvää asiaa ennenkuin pääset tekemään omia ohjelmia. Älä anna houkutukselle valtaa ja hyppää tämän asian yli. Jos teet niin, Manager ilmoittaa meille laiskuudestasi ja me lähetämme kosmisen tuhoamissäteen taloosi ja se muuttaa tietokoneesi savuavaksi muovilätäköksi aivan silmiesi edessä. Joten lue tottelevaisesti tämäkin luku ja välttä tuhoamismahdollisuus.

MITEN OHJELMARIVI ALOITETAAN?

Jos haluat kirjoittaa oikean BASIC-ohjelman, sinun pitää aloittaa joka vaihe rivinumerolla. Muuten käy kuten typerälle hovimestarillesi edellisessä kappaleessa; tietokone suorittaa käskyn heti kun olet kirjoittanut sen ja se saattaa sotkea koko tehtävän. Muista siis laittaa numerot joka käskyn eteen.

Tyypillinen ohjelma näyttää tällaiselta:

```
10 TEE JOTAKIN  
20 TEE JOTAKIN MUUTA  
30 TEE JOTAKIN AIVAN ERILAISTA
```

MIKSI TARVITSET TILAA OHJELMAN RIVINUMEROIDEN VÄLILLE?

Sinun kannattaa aloittaa luvulla 10 ja laittaa seuraavalle riville numero 20, ja niin edelleen. Voit numeroida ne 1, 2, 3..., jos haluat. Mutta jopa huippu-taitavat ammatikseen ohjelmoivat numeroivat rivinsä 10 välein. Syy tähän välin jättämiseen on valmistautuminen virheisiin etukäteen. Jos kirjoitat 50 tai 60 ohjelmariviä ja huomaat sitten, että olet jättänyt pois tärkeän vaiheen ohjelman alussa, tarvitset tilaa laittaaksesi tuon vaiheen oikealle paikalle.

Itse asiassa koska olet vasta aloittamassa, saattaa olla hyvä idea kirjoittaa ohjelmat näin:

```
100 TEE JOTAKIN  
200 TEE JOTAKIN MUUTA  
300 TEE JOTAKIN AIVAN ERILAISTA
```

Tällä tavalla jätät itsellesi runsaasti tilaa lisäyksiä varten. Sinun ei tarvit-

se pelätä rivinumeroiden loppuvan. Salora Manager voi käsitellä rivinumeroita 1:stä aina 63999:ään asti eikä rivinumeroiden välisellä erolla ole merkitystä, tietokone suorittaa ne yksi kerrallaan kasvavassa numeerisessa järjestyksessä.

Voit vaikka kirjoittaa rivinumeron 10 kirjoitettua rivinumeron 90. Tietokone lajittelee ne ja suorittaa ne rivien alussa olevien numeroiden mukaisessa järjestyksessä.

MITEN OHJELMARIVI LOPETETAAN?

Kun riville tulevat käskyt on kirjoitettu, sinun pitää painaa RETURNia. Muuten tietokone ei tiedä, että olet lopettanut ja sotkee tämän rivin seuraavaan riviin ja et voi ikinä erottaa niitä. Joten tosiasiasa kuvitteellinen ohjelmamme näyttää tältä:

```
100 TEE JOTAKIN RETURN
200 TEE JOTAKIN MUUTA RETURN
300 TEE JOTAKIN AIVAN ERILAISTA RETURN
```

RETURN ei näy kuvaruudulla, mutta älä anna sen hämätä. JOKAINEN RIVI TÄYTYY PÄÄTTÄÄ RETURN-NÄPPÄIMEN PAINALLUKSELLA. Tai muuten ohjelmoinnistasi ei tule mitään.

OHJELMIEN AJAMINEN

Riippumatta siitä kuinka monta riviä ohjelmakäskyjä kirjoitat, tietokone ei tee niillä mitään ennenkuin käsket sitä. Se tapahtuu käskyllä RUN. Tämä käsky pitää kirjoittaa, kun ohjelma on valmis suoritettavaksi, ILMAN RIVINUMEROA. Jos haluamme suorittaa ylläolevan ohjelmamme, meidän pitää kirjoittaa tällä tavoin:

```
100 TEE JOTAKIN RETURN
200 TEE JOTAKIN MUUTA RETURN
300 TEE JOTAKIN AIVAN ERILAISTA RETURN
RUN RETURN
```

KIELIOPPI JA VÄLIMERKIT

Jos olet elämässäsi tavannut ihmisiä, jotka vaativat täydellistä kielioppia joka paikassa — ihmisiä kuten opettajat, päälliköt ja byrokraatit — niin tiedät jo kuinka turhauttavia, ahdasmielisiä, jopa typeriä he voivat olla. Mutta jos ajattelet, että he ovat vaikeita, odotapa, kunnes alat työskentelemään tietokoneen kanssa.

Jokainen tietokone, mukaanluettuna Salora Manager, tulee hulluksi, jos yksikin pieni pilkku puuttuu. Tai on tehty yksikin pieni kirjoitusvirhe, se sekoaa niin täysin, että se pysähtyy täsmälleen siihen kohtaan eikä tee mitään ennenkuin korjaat virheen.

Ainoa keino välttää tämä turhauttava kokemus on opetella kaikki väli-merkkien käyttö, käyttää niitä oikein ja tarkkailla sanojen kirjoittamista huolella.

PELÄTTY SYNTAKSIVIRHE

Managerissa on joukko sisäänrakennettuja virheilmoituksia, joita tietokone kirjoittaa iloisesti ruudulle aina tehdessäsi virheen. Älä ole harmisiasi tästä. Useat ihmiset jotka työskentelevät erilaisten tietokoneiden kanssa, eivät saa minkäänlaista virheilmoitusta. Heidän ohjelmansa eivät vain toimi. Sinä ainakin tiedät mistä aloittaa.

Useimmissa tapauksissa tietokone jopa kertoo sinulle millä ohjelmarivillä virhe on tehty, joten voit tarkistaa sen rivin komennolla LIST X, jossa X on kyseinen rivinumero ohjelmassa. Sen jälkeen sinun tarvitsee vain selvittää mikä on vikana. On olemassa joukko virheilmoituksia, mutta me käsittelemme tällä kertaa vain yhtä. Muut selitetään myöhemmin, kun sinulla on vähän enemmän kokemusta.

SYNTAKSI- ELI MUOTOSÄÄNTÖVIRHE (SYNTAX ERROR)

Jollet ole nähnyt tämän ilmestyvän vielä kuvaruudullesi, älä välitä; tulet näkemään sen. Sen ilmoituksen saat todennäköisesti useammin kuin jonkun muun. Vielä senkin jälkeen kun sinusta on tullut mestariohjelmioija.

Salora Manager on erityisesti ohjelmoitu olemaan mukava. Kun se ilmoittaa "SYNTAX ERROR", se itse asiassa tarkoittaa "TEIT VIRHEEN, SINÄ TYPERÄ IHMISPOLOINEN!"

Kun saat tämän ilmoituksen, johtuu se yleensä siitä, että olet tehnyt virheen välimerkkien käytössä. Jokaisella BASIC-komennolla on oma tarkka muotonsa ja jos teet siinä virheen, tietokone ei tunnista sitä käskyä.

MITEN SIITÄ SELVIÄÄ?

Tietokone ilmoittaa, että rivillä 80 on muotovirhe tekstillä "SYNTAX ERROR IN 80". (Kyseessä voi tietenkin olla mikä tahansa rivinumero, mutta me käytämme lukua 80, jotta emme sekoittaisi asiaa käyttämällä esimerkiksi rivinumerona X:ää tai jotakin muuta vastaavaa). Kun näet tämän ilmoituksen, kirjoita LIST 80. Silloin ohjelmarivi, jonka numero on 80, tuostuu ruudulle ja voit yrittää selvittää, mikä siinä on vikana. Jos siinä ei ole selvää kirjoitusvirhettä, tarkista BASIC-käsky jota olet käyttänyt sillä rivillä ja katso ovatko välimerkit oikein. Jos et huomaa mitään virhettä, yritä kirjoittamalla koko rivi uudelleen ja laita välilyönti jokaisen sanan ja symbolin väliin. Kirjoita sitten RUN. Jos tämä ei toimi, mene ulos ja käy vaikka lenkillä. Kun tulet takaisin, olet saattanut selvittää mikä meni vikaan.

RUUTUEEDITOINTI

On niin monia mahdollisuuksia sotkea BASIC-ohjelma, että emme edes yritä listata niitä kaikkia tässä. Mutta Managerissa on sisäänrakennettu ruutueditori, joka voi auttaa sinua selvittämään minkälaisen sotkun tahansa.

Käytetäänpä havainnollistamiseen ohjelmaa, jolla muodostettiin iloiset kasvot:

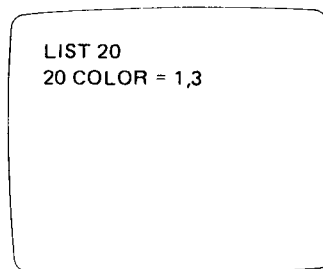
```
10 GR RETURN
20 COLOR = 1, 3 RETURN
30 CIRCLE (75, 50), 20 RETURN
40 CIRCLE (175, 50), 20 RETURN
50 PLOT 70, 145 TO 75, 150 RETURN
60 PLOT 180, 145 TO 175, 150 RETURN
70 FOR X = 75 TO 175 STEP 1 RETURN
80 PLOT X, 150 RETURN
90 NEXT X RETURN
100 CIRCLE (85, 60), 5 RETURN
110 CIRCLE (165, 60), 5 RETURN
120 END RETURN
```

RIVIN HAKU

Kun huomaat, että sinun pitää muuttaa jotakin, olet yleensä kaukana ongelmasta. Useat BASIC-ohjelmat ovat useiden sivujen pituisia. Sinun on saatava eteesi rivi, jota haluat muuttaa. Oletetaan, että haluamme muuttaa ohjelmassamme taustaväriä. Tiedämme, että väri määritellään rivillä 20, joten kirjoitamme

LIST 20 RETURN

Nyt kuvaruutu näyttää tällaiselta:



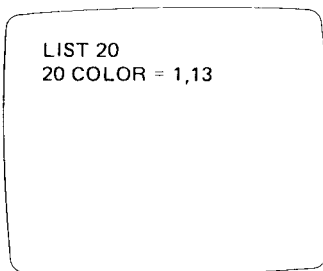
```
LIST 20
20 COLOR = 1,3
```

RIVILLE LISÄÄMINEN

Entäpä jos haluamme muuttaa taustavärin purppuraksi, jonka värinnumero on 13. Meidän tarvitsee ainoastaan lisätä numero 1 numeron 3 eteen rivillä 20. Ihan helppoa.

Siirrämme kursorin riville 20 käyttämällä CTRL- ja L-näppäintä yhdessä (käytämme itseasiassa näppäimen ↑ -toimintaa, kuten aiemmin kerroimme SHIFT ja CTRL-toimintojen yhteydessä). Sitten siirrämme kursoria rivillä käyttäen → -näppäintä (CTRL ja Å-näppäimiä) kunnes se on numeron 3 kohdalla.

Nyt painamme CTRL- ja P-näppäimiä samaan aikaan. Tätä kutsutaan liisäys eli upotustilaksi "INSERT". Sen avulla voimme lisätä yhden merkin muuttamatta mitään rivillä olevaa. Sen jälkeen kirjoitamme vain numeron 1 ja painamme RETURNia. Editointi on suoritettu. Kuvaruutemme näyttää nyt tältä:



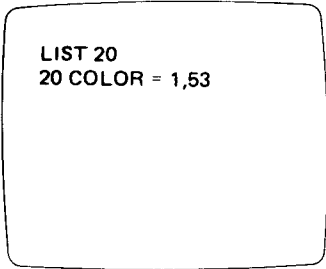
Jos haluat nyt ajaa tämän muutetun ohjelman, kirjoita RUN ja paina RETURNia.

Ohjelman saat takaisin kuvaruutuun painamalla RESET-näppäintä.

MERKIN POISTO

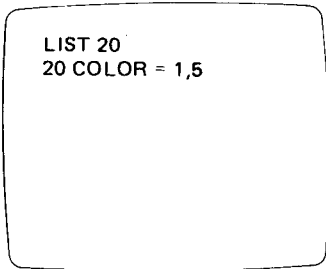
Entäpä jos haluamme muuttaa taustavärin punaiseksi? Punaisen värikoodi on 5. Mitään taikatemppua emme voi tehdä 13:lle. Meidän on muutettava yksi viideksi ja poistettava numero 3.

Siirrä kohdistin ykkösen kohdalle. Kirjoita sen kohdalle 5. Nyt kuvaruutu näyttää seuraavalta.



LIST 20
20 COLOR = 1,53

Nyt kohdistin on kolmosen kohdalla. Älä siirrä sitä. Nyt sinun pitää painaa CTRL ja 0-näppäimiä samanaikaisesti. Numero 3 häviää ja rivi tiivistyy automaattisesti. Nyt painamme RETURNia ja editointi on suoritettu. Kuvaruutu on nyt seuraavan näköinen.



LIST 20
20 COLOR = 1,5

Helppoa, vai mitä?

BASIC-OHJELMOINNIN KYMMENEN KÄSKYÄ

- 1) KÄYTÄ AINA ISOJA KIRJAIMIA
- 2) ALOITA RIVI AINA RIVINUMEROLLA
- 3) PÄÄTÄ RIVI AINA RETURNILLA
- 4) NUMEROI RIVIT HALUAMAASI SUORITUSJÄRJESTYKSEEN
- 5) ÄLÄ KIRJOITA RIVINUMEROA RUN/LIST/NEW/CONT -KOMENTOJEN
ETEEN
- 6) ÄLÄ TEE PIENEN PIENTÄKÄÄN KIRJOITUSVIRHETTÄ
- 7) ÄLÄ EDES AJATTELE VÄLIMERKKIEN KÄYTÖSSÄ VIRHEITÄ
- 8) JOS OHJELMA EI TOIMI, ÄLÄ TURHAAN HIILLY TAI MURSKAA
TIETOKONETTASI KIRVEELLÄ
- 9) PIDÄ TÄMÄ KIRJA AINA SAATAVILLASI
- 10) ÄLÄ LUOVUTA IKINÄ

LUKU 2

MANAGERIN BASIC-KÄSKYT GRAFIIKKAKÄSKYT

- GR
- TEXT
- COLOR =
- PLOT
- UNPLOT
- CIRCLE
- RECT
- HOME
- VPEEK
- VPOKE
- VIDEO-RAM TEKSTITILASSA
- VIDEO-RAM GRAFIIKKATILASSA

MIKSI ALOITAMME GRAFIIKKAKÄSKYILLÄ?

Useimmissa ohjekirjoissa grafiikkakäskyt on kätkeyty kirjan loppuun heksadesimaalilukuja ja konekieltä käsittelevien lukujen väliin. Tämä sen vuoksi, että grafiikan käyttö on yleensä vaikeaa.

Uusi Salora Manager on tehnyt grafiikasta lastenleikkiä ja voit luoda erilaisia kuvia ja animaatioita ilman että sinun tarvitsee opiskella ohjelmointia ja puhdasta matematiikkaa kymmentä vuotta.

Managerin avulla voit tehdä kaiken yksinkertaisilla, selvillä käskyillä. On paljon mukavampaa aloittaa grafiikalla kuin yrittää laskea sinejä, kosinejä tai logaritmisia funktiota, eikö totta?

GR

MITÄ SE TARKOITTA?

Komennolla GR tietokoneesi muuttaa kuvaruudun grafiikkatilaan. Sinun pitää tehdä tämä ennenkuin piirrät minkäänlaisia kuvia. Muuten ei tapahdu mitään.

Kun GR-käsky on suoritettu, ei ruudulle voi tulostaa sanoja PRINT-käskyllä. Ne eivät yksinkertaisesti näy ruudulla.

MITEN KIRJOITAT SEN?

10 GR

MITÄ VOIT TEHDÄ SILLÄ?

Tässä tilassa voit tehdä piirroksia, kuvia, kaavioita, käyriä, animaatioita ja pelejä. Jollet kirjoita GR-komentoa, mitään ylläolevista ei näy kuvaruudulla.

TEXT

MITÄ SE TARKOITTAÄ?

Tämän komennon avulla voit palata tekstitilaan grafiikkatilasta. Mutta muista, että kaikki grafiikka katoaa ja sen sijaan ohjelmarivit näkyvät jälleen.

Itse asiassa jos laitat TEXT-käskyn samaan ohjelmaan grafiikkasi kanssa, niin töin tuskin näet grafiikan. Se vain välähtää kuvaruudussa ja olet taas tekstitilassa sekunnin murto-osassa.

MITEN KIRJOITAT SEN?

10 TEXT

MITÄ VOIT TEHDÄ SILLÄ?

Voit käyttää sitä ohjelmissa, joissa siirrytään grafiikkatilaan näyttätessä jokin kuva tai kaavio, jonka jälkeen käytetään TEXT-käskyä, jotta palataan takaisin tekstitilaan.

COLOR =

MITÄ SE TARKOITTA?

Tämä komento ohjaa tietokonettasi muuttamaan kuvaruudun taustaväri määrätyn väriksi ja muuttamaan edustaväri (kirjaimet ja piirrokset) eri värisiksi.

VÄRIKODI VÄRI

1. MUSTA
2. VIHREÄ
3. VAALEANVIHREÄ
4. TUMMANSININEN
5. VAALEANSININEN
6. TUMMANPUNAINEN
7. SYAANI (SINIVIHREÄ)
8. PUNAINEN
9. VAALEANPUNAINEN
10. TUMMANKELTAINEN
11. VAALEANKELTAINEN
12. TUMMANVIHREÄ
13. MAGENTA (PURPPURA)
14. HARMAA
15. VALKOINEN
16. LÄPINÄKYVÄ

KUINKA KIRJOITAT SEN?

10 COLOR = 1, 3

Ensimmäinen numero edustaväri, jälkimmäinen on taustaväri.

MITÄ VOIT TEHDÄ SILLÄ?

COLOR = -käsken avulla voit käyttää haluamaasi väriyhdistelmää työkennellessäsi ja sen avulla voit värittää piirroksesi. Kuvaruudulla voi näkyä samanaikaisesti vain kahta väriä.

PLOT

MITÄ SE TARKOITTAÄ?

Tämän käskyn avulla piirretään viivoja. Se käskee tietokonetta vetämään viivan ensinmainitusta koordinaatista toiseksi mainittuun koordinaattiin.

Grafiikkatilassa kuvaruutu muodostuu pistejoukosta, jossa on 256 pistettä vaakasuoraan ja 192 pistettä pystysuunnassa. Kuvaruudun vasemman yläkulman koordinaatit ovat 0, 0. Oikea alakulma on vastaavasti 255, 191.

Vaakasuoran (horisontin) koordinaattiakselia kutsutaan aina X-akseliksi ja se merkitään aina ensin. Pystyakselia kutsutaan Y-akseliksi ja se on aina toisena. Joten koordinaatit 35, 100 tarkoittavat 35 pistettä vasemmalta oikealle (X-akselilla) ja 100 pistettä ylhäältä alaspäin (Y-akselilla).

Sinun pitää olla grafiikkatilassa käyttäessäsi PLOT-komentoa.

MITEN KIRJOITAT SEN?

```
10 GR
20 COLOR = 1, 3
30 PLOT 100, 100 TO 150, 100
40 REM TULOSETTIIN SUORA VIIVA
50 PLOT 150, 100 TO 125, 50 TO 100, 100
60 REM VIIVA JATKETTIIN KOLMIOKSI
70 END
```

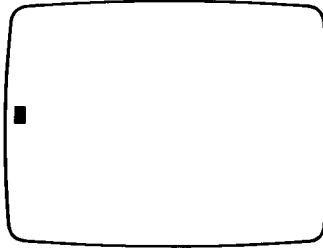
Huom: Kaikki koordinaatit erotetaan toisistaan pilkulla. Kaikki koordinaattiparit (X, Y) erotetaan toisistaan sanalla TO. Huomaa riviltä 50, että voit lisätä uusia pisteitä lisäämällä vain TO X, Y niin kauan kunnes kaikki pisteet ovat yhdistetty.

MITÄ VOIT TEHDÄ SILLÄ?

Kaikkea, mitä voit kuvitella, voidaan piirtää PLOT-käskyn avulla. Voit tehdä viivoista paksuja käyttämällä FOR...TO...STEP...NEXT-käskyjä.

```
10 GR
20 COLOR = 1, 3
30 FOR X = 10 TO 15 STEP 1
40 PLOT X, 100 TO X, 120
50 NEXT X
60 END
```

Esimerkkimme piirtää paksun viivan kuvaruudulle X-koordinaateille 10...15 (ruudun vasen reuna) ja Y-koordinaateille 100...120 (pystysuunnassa suunnilleen ruudun keskelle).
Kokeile ja toteuta itse!



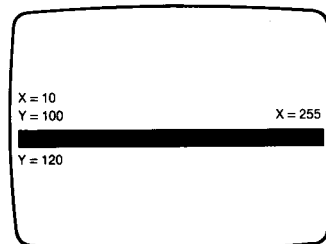
Jos muutat ohjelmaa seuraavasti, havaitset miten helppoa PLOT-käskyn käyttäminen on. Kirjoita rivi 30 uudelleen

```
30 FOR X = 10 TO 255 STEP 1
```

ja paina RETURN. Listaa ohjelma, paina CTRL ja L, sitten RETURN. Kuvaruudussa on nyt tämän näköinen ohjelma:

```
10 GR
20 COLOR = 1, 3
30 FOR X = 10 TO 255 STEP 1
40 PLOT X, 100 TO X, 120
50 NEXT X
60 END
```

Aja uusi ohjelma ja katso mitä tapahtuu!



Muutimme riviä 30 siten, että tietokone piirtää jatkuvan viivan X-akselille 10...255 ja viivan sijainti Y-akselilla on 100...120.

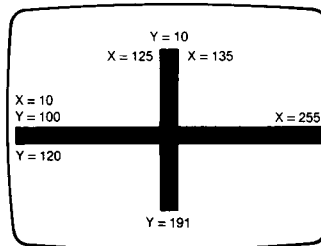
Jatketaan vielä ohjelman muuttamista. Kirjoita

```
60 FOR Y = 10 TO 191 STEP 1
70 PLOT 125, Y TO 135, Y
80 NEXT Y
90 END
```

Listaa taas koko ohjelma ja vertaa, onko ohjelmasi tällainen?

```
10 GR
20 COLOR = 1, 3
30 FOR X = 10 TO 255 STEP 1
40 PLOT X, 100 TO X, 120
50 NEXT X
60 FOR Y = 10 TO 191 STEP 1
70 PLOT 125, Y TO 135, Y
80 NEXT Y
90 END
```

Minkä näköinen kuvio nyt syntyy, kun ohjelma ajetaan. Kokeillaan! Kirjoita RUN ja paina RETURN.



Havaitset, että PLOT-käskyä voi kätevästi käyttää esimerkiksi omien peliohjelmien tekemiseen. Älä kuitenkaan vielä ryhdy ohjelmoimaan pelejä. Pystyt tekemään huomattavasti monipuolisempia ohjelmia, kunhan ensin opettelet myös muut grafiikkakäskyt.

UNPLOT

MITÄ SE TARKOITTAÄ?

Tämän käskyn avulla voit poistaa antamasi koordinaattien välillä olevia viivoja. Joten tehtyäsi kömmähdyksen voit poistaa sen ja jatkaa ilman että koko ruutu pitää tyhjentää.

MITEN KIRJOITAT SEN?

50 UNPLOT X1, Y1 TO X2, Y2

Samat säännöt koskevat sekä PLOT- että UNPLOT-käskyjä. X-akseli ilmoitetaan aina ensin ja vasta sen jälkeen Y-akseli. Voit kirjoittaa UNPLOT...TO...TO...TO niin kauan kunnes haluamasi viivan osat on poistettu.

MITÄ VOIT TEHDÄ SILLÄ?

Sen lisäksi että se on mukava tapa korjata virheitä, voit käyttää UNPLOT-komentoa animaation tekoon. Saat kuvat ilmestymään tai katoamaan käyttämällä PLOT- ja UNPLOT- käskyjä.

CIRCLE

MITÄ SE TARKOITTAÄ?

Tämä ohjaa tietokonetta piirtämään ympyrän antamasi pisteen X, Y ympärille säteen ollessa haluamasi. Muista, että säde on vain puolet ympyrän halkaisijasta, joten jos annat säteen pituudeksi 20, on ympyräsi halkaisija 40 pistettä, kun se on piirretty.

Sen sijaan että istuisit miettimässä haluamasi koordinaattien tarkkaa arvoa, voit usein antaa tietokoneen tehdä työn puolestasi. Tiedät esimerkiksi, että kuvaruudun koko on 256 x 192 pistettä. Jos haluat piirtää ympyrän aivan ruudun keskelle säteen ollessa 10, tietokone voi toimia vaikka allaolevan ohjelman mukaan.

MITEN KIRJOITAT SEN?

```
10 GR
20 COLOR = 1, 3
30 X = 256/2
40 Y = 192/2
50 CIRCLE (X, Y), 10
60 END
```

MITÄ VOIT TEHDÄ SILLÄ?

Piirtää ympyröitä. Mitäpä muutakaan.

Pienen esimerkin CIRCLE-käskyn grafiikkaominaisuuksista saat, jos kirjoitat seuraavat rivit uudelleen:

```
50 FOR I = 10 TO 80 STEP 1
60 CIRCLE (X,Y), I
70 NEXT I
80 END
```

RECT

MITÄ SE TARKOITTAÄ?

Tämän komennon avulla tietokone piirtää suorakulmion, esimerkiksi neliön. Sinun tarvitsee ainoastaan ilmoittaa vasemman yläkulman ja oikean alakulman X- ja Y-koordinaatit.

MITEN KIRJOITAT SEN?

```
10 GR
20 COLOR = 1, 3
30 RECT (130, 30), (90, 90)
40 END
```

Kuten kaikissa muissakin grafiikkakäskyissä, ensin ilmoitetaan X-koordinaatin arvo, sen jälkeen Y:n arvo. Tätä käskyä varten pitää antaa kaksi X:n ja kaksi Y:n arvoa, yksi kumpaisenkin sulkeen sisällä. Älä unohda pilkkuja eikä sulkumerkkejä.

HOME

MITÄ SE TARKOITTAÄ?

Tämä komento tyhjentää tekstiätilassa kuvaruudun. Kohdistin siirtyy ruudun vasempaan yläkulmaan. Kaikki kuvaruudulla näkynyt teksti häviää. Kuitenkaan se ei vaikuta muistissa olevaan ohjelmaan.

MITEN KIRJOITAT SEN?

```
50 HOME
```

VPEEK

MITÄ SE TARKOITTA?

Tämä on samanlainen kuin PEEK-komento, paitsi että PEEK-komento lukee dataa käyttäjän RAM-muistista, ja VPEEK-komento lukee dataa video-näytön RAM- muistista.

Sen avulla voidaan esimerkiksi ohjelmallisesti tulostaa tekstiilan kuva-ruutunäyttö kirjoittimella.

MITEN KIRJOITAT SEN?

10 PRINT VPEEK(A)

tai

10 X = VPEEK(A)

Lauseessa A on haluamasi muistipaikan osoite. Osoitteen pitää viitata video- RAM-muistikarttaan, jonka osoitteet ovat nollan ja 16383:n välillä.

VIDEO-RAM MUISTIKARTTA

0	TALLETA BITTIKUVA GRAFIIKKATILASSA
6114	
8192	TALLETA LIIKKUVA KOHDE GRAFIIKKATILASSA
14236	TALLETA GRAFIIKKATILAN VÄRIKOODI
15360	TALLETA TEKSTITILAN TEKSTINÄYTTÖ
16128	TALLETA BITTIKUVAN SIJAINTI GRAFIIKKATILASSA
16384	TALLETA LIIKKUVAN KOHTEEN SIJAINTI

VPOKE

MITÄ SE TARKOITTA?

Tämä on samanlainen kuin POKE-komento, paitsi että POKE-komento kirjoittaa dataa käyttäjän RAM-muistiin ja VPOKE-komento kirjoittaa dataa videonäytön RAM-muistiin.

Tekstitilassa voit muodostaa sen avulla uusia grafiikkamerkkejä.

MITEN KIRJOITAT SEN?

10 VPOKE A, B

missä A on haluamasi muistipaikan osoite ja B on talletettava data, joka on nollan ja 255:n välillä. Osoitteen pitää viitata video-RAM-muistikarttaan (0... 16383).

VIDEO-RAM TEKSTITILASSA

Tekstitilassa voit kontrolloida video-RAM-muistialuetta 6400 — 7167. Tällä alueella sijaitse merkkivalikoiman bittikuvataulukko (koodit heksadesimaalisena \$20 — \$7F, desimaalisena 32 — 127). Merkki koostuu kahdeksan tavun muodostamasta 8x8 -bittimatriisista. Tekstitilassa ei käytetä bittejä 0 ja 1, joten niiden arvo on nolla.

Alueella 14336 — 15295 sijaitsee kuvaruudun RAM-kuva. Merkin ASCII-koodin syöttäminen siirtää merkin heti kuvaruudulle. Kuvaruutu koostuu 24 rivistä, joille mahtuu 40 merkkiä. Sarakkeita 1, 2, 39 ja 40 ei käytetä. Rivin 1 sarake 1 vastaa video-RAM- osoitetta 14336 ja rivin 24 sarake 40 osoitetta 15295.

ESIMERKKI VIDEO-RAM-MUISTIKARTAN KÄYTÖSTÄ

```
10 FOR I = 0 TO 7
20 READ A
30 VPOKE 6920 + I,A
40 NEXT I
50 DATA 56, 68, 68, 68, 68, 68, 40, 108, 0
```

Tämän ohjelman suorituksen jälkeen tulostuu omega-merkki (Ω), kun painetaan A-näppäintä yht'aikaa SHIFT-näppäimen kanssa.

RESETin käytön jälkeen tulostuu jälleen "a".

Merkki muodostuu ruudulla 8x8 pisteen ruudukosta, kuten alla on huomattavasti suurennettuna esitetty.

	VASEMMAN- PUOLEISET LOHKOT	OIKEAN- PUOLEISET LOHKOT
1. RIVI		
2. RIVI		
3. RIVI		
4. RIVI		
5. RIVI		
6. RIVI		
7. RIVI		
8. RIVI		

Jokainen rivi on jaettu kahteen neljän pisteen lohkoon.

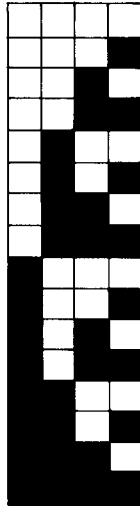
RIVI							

Luvut edellisellä sivulla olevaan DATA-lauseeseen on saatu seuraavasti:

	Heksa- desimaali- luku	Desimaali- luku
	38	56
	44	68
	44	68
	44	68
	44	68
	28	40
	6C	108
	0	0

Seuraavassa esimerkissä on esitetty merkkilohkojen binääri- ja heksadesimaalikoodit. Täydellinen heksadesimaali-desimaalikooditaulukko löytyy kirjan lopusta liitteestä 2, sivulta 135. Leikkaa tarvittaessa sivu irti ja aseta taulukko tietokoneesi viereen, jos aiot tehdä runsaasti eri grafiikkamerkkejä.

Lohkot



Binäärikoodi

1 = näkyvissä
0 = ei näkyvissä

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

Heksadesimaali-koodi

0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F

VIDEO-RAM GRAFIIKKATILASSA

Kuvaruudun kuva on jaettu video-RAM-muistissa kolmeen 256-tavun ryhmään.

Ryhmä 1: \$3C00 — \$3CFF
2: \$3D00 — \$3DFF
3: \$3E00 — \$3EFF

(32 merkkiä 24 rivillä = 768 tavua)

Jokainen ryhmä muodostaa kaksi vastaavaa taulukkoa. Toinen sisältää 256 bittikuvaa 8x8 -bittimatriisinä, toinen jokaista bittikuvan pisteriviä vastaavan edusta- ja taustavärin.

Taulukon alkuosoitteet ovat:

	Bittikuvataulukko	Värikooditaulukko
Ryhmä 1	0000 — 07FF	2000 — 27FF
2	0800 — 0FFF	2800 — 2FFF
3	1000 — 17FF	3000 — 3FFF

Värikooditaulukossa neljä merkitsevintä bittiä ilmaisevat edustavärin ja neljä vähiten merkitsevää bittiä ilmoittavat taustavärin.

Alustus GR-komennolla asettaa kaikki taulukot täyteen nolliä, ja kuvaruutalue alustetaan kolmesti arvoilla nollasta 255:een.

LUKU 3

MANAGERIN BASIC-KOMENNOT HELPOT KOMENNOT

- END
- NEW
- LET
- REM
- PRINT
- INPUT
- LIST
- STOP
- CONT
- BREAK
- TRACE JA NOTRACE
- TEHOKKAAT OIKOTIET

MIKSI NÄMÄ KOMENNOT OVAT HELPPOJA?

Tässä luvussa olevat Managerin BASIC-komennot eivät käytä matematiikkaa. Jos ajattelet ja toimit kuten useimmat meistä, jotka olemme jo hankineet henkilökohtaisen tietokoneen, pidät varmasti ostosi yhtenä syynä tietokoneen erinomaista matematiikkakykyä. Joten sinun ei tarvitse itse olla matematiikkanero. Näiden kommentojen avulla voi ohjelmoida ilman monimutkaista matematiikkaa. Sen takia nämä käskyt ovat helppoja.

END

MITÄ SE TARKOITTA?

Tietokoneet ovat niin tyhmiä, että sinun pitää kertoa niille milloin ohjelma loppuu. Muuten ne eivät tiedä sitä. Joten lopettaessasi BASIC-ohjelman viimeinen ohjelmarivi, JOLLA ON RIVINUMERO, on END-komento.

MITEN KIRJOITAT SEN?

1000 END RETURN

Älä käytä rivinumeroa 1000. Käytä mitä tahansa numeroa, johon ohjelmasi päättyy.

MITÄ VOIT TEHDÄ SILLÄ?

Saat tietokoneen lopettamaan ohjelman suorituksen. Jollet lopeta ohjelmaasi sillä, tietokone olettaa, että ohjelma ei ole vielä päättynyt ja mitä tahansa vielä kirjoitatkin on osa sama ohjelmaa.

NEW

MITÄ SE TARKOITTA?

Tietokoneet ovat niin typeriä, että vaikka sinä tiedätkin milloin aloitat aivan uutta ohjelmaa, ne eivät sitä tiedä. Sinun pitää kertoa se niille NEW-komennolla.

Jos et kerro niille, ne sotkevat uuden ohjelmasi vanhaan, muistissa olevaan ohjelmaasi ja et ikinä kykene erottamaan rivejä toisistaan.

Kun kirjoitat NEW-komennon, muistissa oleva ohjelma häviää. Varmistaudu, että olet tallettanut sen nauhalle tai et tarvitse sitä enää.

MITEN KIRJOITAT SEN?

NEW RETURN

Rivinumeroa ei tarvita.

MITÄ VOIT TEHDÄ SILLÄ?

Voit kirjoittaa sen jokaisen kirjoittamasi ohjelman alkuun. Sitten kun ajat ohjelman, et saa yllätyksekseksi vanhoja ohjelmarivejä uuteen ohjelmaasi.

LET

MITÄ SE TARKOITTAÄ?

LET tarkoittaa "KUVITTELE". Käsket tietokoneen kuvitella, että jokin on totta. Kuten $X = 5$. Tietokone tietää vain sen, jonka sinä kerrot sille. Joten se "kuvittelee", että $X = 5$.

Tietokoneen ohjelmointi ei ole koulussa oppimasi algebran kaltaista. Kun sanot algebrassa $X = 5$, niin et voi muuttaa sitä. Mutta tietokoneen ohjelmoinnissa voit.

Tämä siksi, että sinä vain "kuvittelet". Joten heti seuraavalla rivillä voit käskää tietokonetta kuvittelemaan, että nyt $X = 287$. Se "kuvittelee", että näin on, kunnes kerrot sille jotain muuta.

KUINKA KIRJOITAT SEN?

10 LET $X = 5$

MITÄ VOIT TEHDÄ SILLÄ?

Lähes kaikki on mahdollista LET-komennon avulla. Muista, että tietokone suorittaa käskyt tarkasti rivinumeroiden mukaisessa järjestyksessä. Kuten yllä on rivillä 10, $X = 5$. Seuraavalla rivillä voidaan sanoa LET $X = 2096$ ja tietokone hyväksyy tämän, kun se suorittaa kyseisen käskyn.

REM

MITÄ SE TARKOITTAA?

Pidä aina mielessäsi, että tietokoneet ovat tyhmiä. Ne yrittävät käsitellä kaiken sen tiedon, jonka niille syötät. Jos sinulla on tekstiä, jota et halua niiden ymmärtävän ohjelmana, sinun on käskettävä sitä olla välittämättä tästä tekstistä. Tällaisia tekstejä voivat olla esimerkiksi ohjelmien nimet, aliohjelmien kommentit ja muut ohjelman seurantaan helpottavat tiedot. Tämän takia on REM-käsky. Se tarkoittaa itseasiassa kommenttia (RE-MARK), mutta se neuvoo typerää tietokonetta, että tätä informaatiota ei tarvitse käsitellä.

MITEN KIRJOITAT SEN?

10 REM PALLOPELIOHJELMA

MITÄ VOIT TEHDÄ SILLÄ?

Voit käyttää REM-käskyä, jotta muistat ohjelman tarkoituksen. Se on myös käytännöllinen huomautus muille ohjelmoijille keskellä ohjelmaa, se kertoo heille mitä teet ohjelman eri vaiheissa. Voit käyttää sitä myös etsiessäsi ohjelmasta virheitä, lisää REM-käsky rivinumeron ja rivillä olevan tekstin väliin. Tietokone ei huomioi sitä riviä ja näet, miten tietokone toimii, kun kyseinen rivi on poistettu ilman, että sinun tarvitsee kirjoittaa sitä kokonaan uudelleen, mikäli tarvitset kyseistä riviä. Jos tarvitset vielä kyseistä riviä, poistat vain REM-käskyn.

PRINT

MITÄ SE TARKOITTAÄ?

Se tarkoittaa "NÄYTÄ MINULLE VASTAUS KUVARUUDULLA". Tämä komento ohjaa tietokonetta tulostamaan vastauksen ohjelman ajon aikana. Tai voit käyttää sitä, jotta kuvaruudulle tulostuu tekstiä ohjelman eri vaiheissa. Muuten tietokone vain suorittaa laskutehtäviä tai muita toimeskiantoja, saa vastaukset mutta ei näytä niitä sinulle. Tietokone vain nököttää siinä edessäsi typeränä.

MITEN KIRJOITAT SEN?

10 PRINT "VIHAAN TIETOKONEITA"

Tällä tavoin tulostetaan tekstiä. Sen pitää olla lainausmerkkien " " sisällä. Tässä tapauksessa kuvaruudulle tulostuu teksti VIHAAN TIETOKONEITA.

```
10 LET X = 5 RETURN
20 LET Y = 5 RETURN
30 PRINT X + Y RETURN
40 END RETURN
RUN RETURN
```

Tällä tavoin saat vastauksia. Lainausmerkkejä ei käytetä. Tässä tapauksessa tulos on 10.

MITÄ VOIT TEHDÄ SILLÄ?

Voit tulostaa kirjaimia kaikkialle kuvaruudulla tai voit saada vastauksia kysymyksiisi. Voit myös käyttää PRINT-komentoa asetettaessa kysymyksiä keskelle ohjelmia.

Kun ohjelmaa ajetaan, voit saada sen yhtäkkiä pysähtymään ja kysymään käyttäjältä vaikka MILLOIN OLET SYNTYNYT? Tietokone ei tee mitään ennenkuin tähän kysymykseen vastataan.

INPUT

MITÄ SE TARKOITTAÄ?

INPUT-komento käskää tietokoneen odottamaan syöttöä näppäimistöltä ennenkuin se jatkaa eteenpäin. Syöttö voi olla joko numeroita tai tekstiä.

Kuitenkin jollet ilmoita, että haluat syötteen tekstiä, tietokone hyväksyy syötteen vain numeroita.

Jos haluat syötteen tekstiä, sinun pitää käyttää merkkijonomuuttujia. (Katso tarkemmin luvusta 7, jossa käsitellään merkkijonoja.)

MITEN KIRJOITAT SEN?

50 INPUT A

Tämä sallii vain yhden numeerisen syötön. Kaikenlainen muu syöttö aikaansaa kuvaruudulle tekstin "REENTER", joka tarkoittaa, uudelleen.

50 INPUT A, B, C, D, E

Tämä vaatii viisi numeerista syöttöä näppäimistöltä ennenkuin ohjelman suoritus jatkuu. Kaikki luvut kelpaavat, jos et vaadi toisin.

50 INPUT A\$

Tämä odottaa syötteen tekstiä. Käyttääksesi tätä dataa myöhemmin voit tehdä jotain tällaista:

```
10 COLOR = 1,3
20 PRINT "ANNA NIMESI"
30 INPUT A$
40 PRINT "TERVE", A$
50 END
```

MITÄ VOIT TEHDÄ SILLÄ?

Voit antaa käyttäjän keskustella ohjelman kanssa tai voit tehdä ohjelmia, jotka vaativat jatkuvasti päivityksiä.

LIST

MITÄ SE TARKOITTAÄ?

Se tarkoittaa "NÄYTÄ MINULLE OHJELMA KUVARUUDULLA". Sen avulla voit listata kaikki ohjelmarivit tai minkä tahansa osan ohjelmasta, jotta voit etsiä virheitä tai tehdä lisäyksiä tai poistoja.

Pitkissä ohjelmissa se auttaa myös muistamaan mitä aikaisemmilla riveillä tehtiin, mikäli sinun tarvitsee viitata niihin.

KUINKA KIRJOITAT SEN?

LIST `RETURN` Tämä näyttää koko ohjelman

LIST 10,100 `RETURN` Tämä näyttää rivit 10 — 100

LIST 50, `RETURN` Tämä näyttää rivit 50:stä loppuun

LIST, 50 `RETURN` Tämä näyttää rivit alusta riville 50

LIST 50 `RETURN` Tämä näyttää ainoastaan rivin 50

MITÄ VOIT TEHDÄ SILLÄ?

Voit listata ohjelmarivejä uudelleentarkastelua tai korjausta varten. Ymmärtääksesi miten ammattimaiset ohjelmoijat tekevät ohjelmia. Voit myös käyttää LIST-komentoa tutkiaksesi osia ohjelmista, jotka lataat kasettinauhalta. Tämä on hyvä keino oppia arvokkaita vinkkejä grafiikan, animaation ja musiikin ohjelmoinnista.

LIST-komennon käyttö on myös hyvä testi, kun tarkistat onko muistissa ohjelmaa. Kirjoita pelkkä LIST ja tiedät välittömästi, onko tietokoneen muistissa ohjelmarivejä.

STOP

MITÄ SE TARKOITTAÄ?

Se tarkoittaa juuri sitä mitä se sanookin. STOP. Se on komento, jota käytetään ohjelmassa, kun halutaan tietokoneen pysäyttävän kaikki toiminnot, kunnes sitä näppäimistöltä käsketään jatkamaan.

MITEN KIRJOITAT SEN?

```
10 LET X = 5
20 LET B = 23
30 PRINT X + B
35 STOP
40 PRINT 2*X + 2*B
RUN
28
BREAK IN 35
```

MITÄ VOIT TEHDÄ SILLÄ?

Voit käyttää STOP-komentoa virheiden etsintään ohjelmasta, nähdäksesi mitkä ohjelman osat toimivat ja mitkä eivät. STOP-käskyn käyttö on paljon helpompi tapa kuin yrittää keskeyttää ohjelman suoritus oikeassa kohdassa näppäimistön avulla (katso BREAK-komentoa). Sinun tarvitsee vain lisätä STOP-käskyjä ohjelman eri vaiheisiin. Kun näet, että ohjelma toimii, poistat STOP-käskyt. Saadaksesi ohjelman suorituksen jatkumaan STOP-käskyn jälkeen kirjoita CONT-komento.

CONT

MITÄ SE TARKOITTAÄ?

Se tarkoittaa "JATKA" (CONTINUE). Se käskee tietokonetta jatkamaan ohjelman suoritusta siitä mihin se jäi sen sijaan, että suoritus alkaisi ohjelman alusta.

MITEN KIRJOITAT SEN?

CONT RETURN

MITÄ VOIT TEHDÄ SILLÄ?

Tätä komentoa käytät, kun joko BREAK- tai STOP-komento on pysäyttänyt ajettavan ohjelman. Sen avulla ohjelman suoritus on mahdollista aloittaa siitä kohdasta mihin se pysäytettiin tai missä se keskeytettiin. Joten se säästää sinut ohjelman suorittamiselta alusta alkaen. Sitä käytetään yleensä virheiden etsinnässä, mutta voit käyttää sitä myös syöttönä ohjelmissa, jotka alkavat ja pysähtyvät käyttäjän tutkimuksia varten.

BREAK (CTRL-C)

MITÄ SE TARKOITTAÄ?

Tietokone suorittaa käskyjä, vaikka ne ovat loogisesti vääriä. Se ei tiedä. Joten pysäyttääksesi sen näppäimistöltä ohjelman ajon aikana käytä BREAK-komentoa.

MITEN KIRJOITAT SEN?

CTRL-C

(Paina sekä CTRL- että C-näppäimiä samaan aikaan).

MITÄ VOIT TEHDÄ SILLÄ?

Voit pysäyttää ajossa olevan ohjelman. Voit keskeyttää ikuiset silmukat. Voit selviytyä tuntemattomista ongelmista, kuten sellaiset, jotka ilmenevät siten, että kirjoitettuasi ohjelman ja painettuasi RETURNia ei tapahdu mitään. BREAK-komennon avulla säilytät ohjelmasi ja toivottavasti saat korjattua sen, jotta se toimisi oikein.

Tehdään ohjelma, joka ei lopu, ja ajetaan se.

```
1000 PRINT 1234  
2000 GOTO 1000  
RUN
```

Kun keskeytät ohjelman suorituksen BREAK-komennon avulla, kuvaruudulla näkyy keskeytyksestä johtuva ilmoitus, joka on tämän tapainen:

```
BREAK IN 1000
```

Tämä tarkoittaa, että ohjelman suoritus on pysähtynyt ohjelmariville 1000. Jos et näe tätä ilmoitusta, se tarkoittaa, ettei ole keskeyttänyt ohjelmaa. Sinun pitää yrittää uudelleen.

TRACE JA NOTRACE

MITÄ NE TARKOITTAVAT?

TRACE-komennon johdosta kuvaruudulle tulostuu jokaisen suoritettavan ohjelmarivin rivinumero. Se näyttää ohjelman suorittamat silmukat, haurautumiset ja muut työt. PRINT-käskyt suoritetaan, samoin kuin GET ja INPUT. Rivinumerot tulostuvat muodossa (**#** xxxx).

Paluu normaaliin tilaan tapahtuu NOTRACE-komennon avulla. Sen jälkeen ohjelman suoritus tapahtuu tavalliseen tapaan ilman tulostuvia rivinumeroita.

MITEN KIRJOITAT SEN?

TRACE
RUN

ja

NOTRACE
RUN

MITÄ VOIT TEHDÄ NIILLÄ?

Yhdessä STOP/CONT-käskyparin avulla voit helposti selvittää ohjelman kulkua etsittäessä virheitä.

PAHEKSUTTAVAT MUTTA TEHOKKAAT OIKOTIET

Tämän tyyppisten kirjojen tarkoituksena on opettaa oikea ja kelvollinen asioiden suoritustapa. Tämä kirja ei ole sellainen. Tämän kirjan nimenomaisena tarkoituksena on yrittää tehdä ohjelmointi sinulle mahdollisimman helpoksi. Tämän johdosta tässä esitellään sinulle joukko oikoteitä, joita voit käyttää. Muuten voisit löytää ne kokemuksen kautta. Näitten oikoteiden ansiosta on ohjelmoinnin aloittaminen sinulle helpompaa, mutta ne voivat myös opettaa sinulle huonoja tapoja.

Valinta on sinun. Voit unohtaa tämän kappaleen ja oppia perinteisten ohjelmointien hyvät tavat. Tai sitten voit käyttää näitä oikoteitä ja tehdä asioista helpompia. Älä lue tästä yhtään pitemmälle ennenkuin ratkaiset tämän moraalisen ongelman omalta kohdaltasi.

SINUN EI TARVITSE ALOITTA A NEW-KOMENNOLLA

Se ei ole välttämätöntä. Muista kirjoittaa NEW-komento, kun aloitat ja tarkista, että se on tyhjentänyt muistin käyttäen LIST-komentoa. Tiedät, että muistissa ei ole ohjelmaa, jos LIST ei tulosta mitään.

SINUN EI TARVITSE LOPETTA A END-KOMENTOON

Jos ajat erän ohjelmia yhdessä (jotka käynnistät komennoilla RUN 10, RUN 200 jne), tarvitset tietenkin END-käskyn. Muuten sillä ei ole merkitystä. Tietokone lopettaa, kun se on suorittanut kaikki ohjelmarivit.

SINUN EI TARVITSE KÄYTTÄÄ LET-KOMENTOA LAINKAAN

Sen sijaan että kirjoitat $10 \text{ LET } X = 5$ voit kirjoittaa $10 \text{ X} = 5$. se voi säästää sinua suuren LET-määrän kirjoittamiselta ja tietokone ymmärtää käskyn yhtä hyvin.

VOIT KORVATA PRINT-KOMENNON KYSYMYSMERKILLÄ

Print komento voidaan korvata ?-merkillä. PRINT A ja ?A tekevät saman asian.

LISÄÄ OIKOTEITÄ

SINUN EI TARVITSE KÄYTTÄÄ REM-KÄSKYÄ PALJOAKAAN

Vaikka sinun oletetaankin laittavan sen jokaisen ohjelman alkuun kerto-
maan ohjelman tarkoituksesta, ei sinun tarvitse vaivautua ensimmäisissä
ohjelmissasi. Sinä muistat ne muutenkin. Siinä vaiheessa kun olet tehnyt
ison joukon ohjelmia, tiedät tarpeeksi, jotta luettuasi muutamia rivejä
ymmärrät ohjelman.

JOKAISEN RIVIN EI TARVITSE ALKAA RIVINUMEROLLA

Säästääksesi aikaa ja ohjelmarivejä voit laittaa kaksi tai kolme käskyä
samalle riville. Sinun pitää vain erottaa ne kaksoispisteellä, joten sen
sijaan että kirjoittaisit

10 X = 5

20 Y = 23

30 A = X + Y

voit kirjoittaa

10 X = 5:Y = 23:A = X + Y

Voit laittaa riville niin monta eri käskyä kuin sille sopii. Yhdelle riville voi
laittaa 255 merkkiä.

VOIT KÄYTTÄÄ RESET-NÄPPÄINTÄ BREAK-KOMENNON SIJASTA

Se on helpompaa. Sinun tarvitsee painaa vain yhtä näppäintä eikä kahta.
Ohjelmasi on tallessa RESETin painamisen jälkeen Salora Managerissa,
mutta ole varovainen. Monissa muissa tietokoneissa se ei ole. Useimmissa
tietokoneissa RESET poistaa kaiken muistista ja aloittaa aivan alkuti-
lanteesta. Joten katso, että RESETin käytöstä tulee tapa vain käyttäes-
säsi Manageria.

Kun käytät RESETiä Managerissa, et voi aloittaa ohjelman ajoa CONT-
komennolla. Sinun pitää kirjoittaa RUN ja aloittaa ohjelman alusta.

LUKU 4

MANAGERIN BASIC-KOMENNOT FIKSUJEN KOMENNOT

- IF...THEN
- AND, OR JA NOT
- FOR..TO..STEP..NEXT
- READ...DATA
- RESTORE
- GOTO
- GOSUB...RETURN
- ON...GOTO/ON...GOSUB
- CLEAR
- SPC, TAB, PUOLIPISTE JA PILKKU PRINT-LAUSEESSA

MIKSI NÄMÄ KÄSKYT OVAT FIKSUILLE?

Nämä komennot säästävät aikaa. Tyhmät tekevät paljon työtä, mutta fiksit pääsevät helpolla käyttäessään näitä käskyjä. Ne ohjaavat tietokonetta tekemään raskaat, toistuvat tehtävät, jollaisissa tietokoneet ovat hyviä. Sen sijaan että kirjoittaisit sivukaupalla ohjelmaa, voit saada tietokoneen tekemään paljon töitä yhdellä tai kahdella käskyllä, ja voit käyttää oman aikasi hyödyllisemmin.

IF...THEN

Tällä komennolla annat tietokoneelle vallan päätöksentekoon, jossa voi olla kaikenlaisia haarautumisia. Sen toiminta-ajatus on, että jos (IF) jokin ehto on tosi, sitten (THEN) tee jotakin.

(Tästä seuraa: jos jokin ei ole tosi, sitten jatka ohjelman suoritusta).

MITEN KIRJOITAT SEN?

10 A = 3

20 B = 7

30 IF A + B > 9 THEN PRINT "SE ON SUUREMPI"

Voit laittaa minkä tahansa haluamasi ehdon IF-lauseeseen.

Voit laittaa minkä tahansa komennon THEN-osaan.

Rivi 30 voisi olla:

30 IF A + B > 9 THEN END

tai

30 IF A + B > 9 THEN A = —3

MITÄ VOIT TEHDÄ SILLÄ?

Voit käyttää tätä käskyä lähes kaikissa tilanteissa, joissa haluat tietokoneen tekemän valinnan. Esimerkiksi voit käyttää sitä animaatiossa siten, että jos (IF) liikkuva kohde saapuu kuvaruudun reunalle, niin silloin (THEN) se kääntyy ympäri ja liikkuu toiseen suuntaan. Heti kun olet tottunut siihen, keksit kymmeniä tapoja, joilla voit käyttää sitä. Itse asiassa olet mahdollisesti jo törmännyt siihen tietokoneessasi. Todennäköisesti Managerin BASICissä on ohjelmarivi, jonka sisältö on jotakin tällaista:

10998 IF typerä ihminen sotkee asioita THEN PRINT "SYNTAX ERROR"

AND, OR ja NOT

MITÄ NE TARKOITTAVAT?

Ne ovat loogisia operaattoreita, joilla voit yhdistää lausekkeita. Niiden antamilla tuloksilla on kaksi mahdollista arvoa: tosi (1) tai epätosi (0).

Esimerkki: 10 A = 2
20 B = 3
30 PRINT A > B
40 PRINT A < B
RUN
0
1

Lauseke A > B antaa tulokseksi nollan, koska se on epätosi. Vastaavasti lausekkeen A < B tulos on yksi, sillä se on tosi. Loogiset operaatiot suoritetaan aritmeettisten ja relaatio-operaatioiden jälkeen.

Operaattori	Esimerkki	Tarkoitus
NOT (EI)	NOT A	Jos A on tosi, on NOT A epätosi
AND (JA)	A AND B	A AND B on tosi vain silloin, kun sekä A että B ovat tosia, muulloin se on epätosi.
OR (TAI)	A OR B	A OR B on epätosi vain silloin, kun sekä A että B ovat epätosia, muulloin se on tosi.

Esimerkki: 10 INPUT A, B, C
20 IF A > B AND B > C THEN PRINT "A > B > C"
30 IF NOT (A > B) OR NOT (B > C) THEN PRINT
"A > B > C ON EPÄTOSI"
40 GOTO 10

RUN
? 10, 5, 7
A > B > C ON EPÄTOSI

MISSÄ KÄYTÄT NÄITÄ OPERAATTOREITA?

Kaikkialla, missä täytyy yhdistää useampia ehtoja: KUN sataa JA minun täytyy mennä ulos, SILLOIN otan mukaan sateenvarjon.

FOR...TO...STEP...NEXT

MITÄ SE TARKOITTAÄ?

Se ei ole niin monimutkainen komento kuin miltä se näyttää. Se on yksinkertaisesti sarja askeleita (STEP), jotka saavat tietokoneen suorittamaan paljon työtä lähes ilmaiseksi.

Se antaa sinulle mahdollisuuden suorittaa saman operaation kasvavalla tai vähenevällä muuttujalla. Jos esimerkiksi haluat jostakin ihmeen syystä lisätä luvun 3 jokaiseen lukuun 1:n ja 100:n välillä, tarvitset vain 3 ohjelmariviä.

MITEN KIRJOITAT SEN?

```
10 FOR X = 1 TO 100 STEP 1
20 PRINT X + 3
30 NEXT X
```

Sama juttu joka 7:nnele luvulle 1:n ja 100:n välillä tapahtuu näin:

```
10 FOR X = 1 TO 100 STEP 7
```

Voit käyttää minkä kokoista askelta tahansa, vaikkei se jakaannu edes loppuarvoon tasan. Voit käyttää askeleina desimaalilukuja kuten .5 tai .3. Jos haluat askeleen pituuden olevan 1, sinun ei tarvitse kirjoittaa sitä. Tietokone käyttää automaattisesti yhden mittaista askelta. (Esimerkkirivi olisi 10 FOR X = 1 TO 100). Sinun tarvitsee ilmoittaa ainoastaan yhdestä poikkeavasta askelista. Jos esimerkiksi menet taaksepäin, sinun pitää kirjoittaa FOR X = 100 TO 1 STEP -1.

Älä koskaan unohda komennosta NEXT-osaa. Muuten tietokone lisää 3 vain ensimmäiseen lukuun (1) ja lopettaa vastaukseen 4. Jollet käskä tietokonetta ottamaan seuraavaa (NEXT) lukua, se ei tee sitä.

Jos yrität olla ovela ja käyttää NEXT-komentoa ilman FOR-komentoa, tietokone nauraa sinulle ja antaa seuraavan ilmoituksen: NEXT WITHOUT FOR ERROR IN 30 (NEXT-komento ilman FORia rivillä 30)

MITÄ VOIT TEHDÄ SILLÄ?

Tätä komentoa voidaan käyttää epämiellyttävään työhön kuten laskemiseen tai sitä voidaan käyttää apuna esimerkiksi grafiikassa. Sen avulla voidaan piirtää viivat niin hitaasti, että voit nähdä niiden syntyvän.

READ...DATA

MITÄ SE TARKOITTAÄ?

Tämä on toinen työtä säästävä komentopari. Sen avulla voit tehdä saman operaation joukolle eri arvoja ja se vaatii vain muutaman ohjelmarivin kirjoittamisen.

DATA-komento kertoo tietokoneelle oikeat arvot. Ne voivat olla lukuja tai ne voivat olla sanoja tai sanontoja. Kaikki arvot, olivatpa ne lukuja tai tekstiä, on erotettava toisistaan pilkuilla.

READ-komento k  skee tietokoneen lukemaan dataa. Ensimm  inen muuttuja (X, Y tai mik   tahansa) READ-komennossa k  ytt     ensimm  ist   DATA-komennon arvoa. Seuraava READin muuttuja saa arvokseen toisen DATA-komennossa olevan arvon ja niin edelleen. Jokainen READ-muuttuja p  t     erottaa toisista muuttujista pilkulla.

DATA-komennossa olevien arvojen lukum      n t  ytyy olla sama tai suurempi kuin READ-komennossa olevien muuttujien lukum      . Muuten aikaan saat kuvaruudulle virheilmoituksen "OUT OF DATA ERROR" (liian v  h     dataa).

MITEN KIRJOITAT SEN?

```
10 DATA 50, 30, "AAMU", 2.5
20 READ A, B, C$, D
30 PRINT A, B, C$, D
40 END
RUN
```

Tietokone tulostaa: 50...30...AAMU...2.5

MIT   VOIT TEHD   SILL  ?

Voit s    st     paljolta ty  lt  , kun joudut tekem         kurjaa ty  t   kuten keskiarvojen laskeminen. Voit my         sille mielenkiintoisia vaihteluita. T  m     komennon avulla voit tulostaa saman kirjeen vain vastaanottajan nimen vaihtuessa.

RESTORE

MITÄ SE TARKOITTAÄ?

RESTORE-komento käskee tietokoneen ottamaan DATAn, joka on jo käytetty READ-komennossa, uudelleenkäytettäväksi seuraavassa READ-komennossa. Kun teet näin, on aina hyvä ajatus käyttää uusia muuttujanimiä toisessa READ-komennossa. Jos käytit X:ää ja Y:tä ensimmäisessä READissä, käytä A:tä ja B:tä toisessa. Ei tietokone kylläkään sekoita niitä keskenään. Se hyväksyy uudet aivan tyytyväisenä. Mutta sinä saatat sotkea ne, etkä ole aivan varma kummasta X:stä puhut ohjelman edetessä.

MITEN KIRJOITAT SEN?

```
10 DATA 13, "AAMU", 2.7
20 READ X, Y$
30 RESTORE
40 READ A, B$
50 PRINT X, Y$
60 PRINT A, B$
RUN
```

Vastaus, jonka saat:

```
13   AAMU
13   AAMU
```

RESTORE-komennon johdosta ensimmäinen READ-lause, joka sitä seuraa, lukee datat ensimmäisestä DATA-lauseesta. Tämä DATA-lause voi olla ohjelmassa ennen RESTORE-komentoa kuten esimerkissä. Se voi olla myös RESTORE-komentoa myöhemmin. Jos vaikka haluat, se voi olla aivan ohjelman lopussa.

MITÄ VOIT TEHDÄ SILLÄ?

RESTOREn avulla voit suorittaa erilaisia operaatioita samalle DATA-joukolle tarvitsematta kirjoittaa dataja uudelleen. Oiva työkalu laiskurille. Mutta fiksut saavatkin luvan ottaa rennommin, eikö totta?

GOTO

MITÄ SE TARKOITTAÄ?

Aivan kuten olettaa sopii, GOTO-komento ohjaa tietokonetta menemään (GO TO) sinne ja tänne sekä tekemään sen, mikä pitää tehdä perille päästyä. Kun ohjaat tietokonetta GOTO-komennolla, sinun pitää lähettää se ohjelmassa olevaan rivinumeroon, vaikka on aikoja, jolloin mieluummin käskisit sen menemään ... vaikkapa jonnekin, jossa pippuri kasvaa.

Toisin kuin GOSUB/RETURN-komennolla, tietokone ei palaa takaisin GOTO-komennon jälkeen. Kun se kerran menee, niin se on mennyt, jollei lähellä sen määränpäättä ole toista GOTO-käskyä, joka käsksee sen palata takaisin.

KUINKA KIRJOITAT SEN?

```
10 PRINT "TERVE"  
20 PRINT "TERVE, TERVE"  
30 PRINT "MITEN MENEÄ"  
40 PRINT "IHAN MUKAVASTI BASICIN OPISKELU SUJUU"  
50 PRINT "KIVA KUULLA, TERVE SITTEN"  
60 PRINT "NO TERVE"  
70 GOTO 10
```

Tällaista ohjelmaa kutsutaan ikuiseksi silmukaksi. Jos todella ajat sen, huomaat, että se ei lopu koskaan. Ainoa tapa lopettaa sen suoritus on käyttää joko BREAK-komentoa tai RESET-näppäintä. GOTO-komennot eivät pääty aina tällaiseen silmukkaan, mutta ne saattavat tehdä niin, jollet ole varovainen niiden käytössä.

MITÄ VOIT TEHDÄ SILLÄ?

Voit hypellä edestakaisin ohjelmassasi suorittaaksesi toistuvia tehtäviä, kuten matemaattisia operaatioita. Kuten ohjetekstien tulostus. Ja samanlaisen grafiikan tulostus yhä uudelleen. Sitä käytetään myös usein tietokoneen päätöstilanteissa IF...THEN-komennossa (IF X=Y THEN GOTO 50).

GOSUB...RETURN

MITÄ SE TARKOITTAÄ?

Tällä komennolla käsket tietokoneen menemään ja suorittamaan ALIRU-TIININ tai aliohjelman, joka on tietyssä kohdassa ohjelmassa, eikä se saa palata ennenkuin sen suoritettavaksi tulee RETURN-lause.

Tietokone siirtyy määrätyle rivinumerolle ja suorittaa siitä eteenpäin olevat käskyt, kunnes se kohtaa RETURN-käskyn. Silloin se siirtyy ohjelmassa takaisin käskyyn, joka seuraa välittömästi sitä lausetta, jossa se suoritti GOSUB-komennon. Se on niin loistava komento, joten on sääli, ettet voi soveltaa sitä joihinkin ihmisiin.

MITEN KIRJOITAT SEN?

```
10 PRINT "JOS ANNAT KAKSI LUKUA"  
20 PRINT "NIIN LASKEN NIIDEN SUMMAN PUOLESTASI"  
30 PRINT "ANNA LUVUT"  
40 INPUT A, B  
50 GOSUB 70  
60 GOTO 30  
70 C = A + B  
80 PRINT C  
90 RETURN
```

Tämäkin ohjelma jatkaa ikuisesti, jollet keskeytä sitä BREAKillä tai RESETillä. Tietenkin käyttäessäsi oikein suuria lukuja, se tulee hulluksi ja keskeyttää itse ohjelman suorituksen. Mutta ethän sinä tee vahingossa-kaan niin ... ethän?

Huom: Tässä tapauksessa sana RETURN pitää kirjoittaa. Muuten yllätyt perusteellisesti.

ON...GOTO/ON...GOSUB

MIHIN SITÄ KÄYTETÄÄN?

Ohjaamaan ohjelman suoritusta riippuen lausekkeen arvosta.

MITEN KIRJOITAT SEN?

ON (lauseke) GOTO (rivinumero 1, rivinumero 2,...)

ON (lauseke) GOSUB (rivinumero 1, rivinumero 2,...)

MITEN KÄYTÄT SITÄ?

Lausekkeen arvon pitää aina olla kokonaisluku, joka on pienempi tai yhtäsuuri kuin 255. Kun sen arvo lasketaan, se ohjaa ohjelmansuorituksen riville, jonka numero on vastaavalla paikalla rivinumerolistalla, joka seuraa joko GOTO- tai GOSUB-lausetta. Jos esimerkiksi lausekkeen arvo on viisi, silloin ohjelman suoritus haarautuu viidennelle rivinumerolle, ja jos sen arvo on yhdeksän, se siirtyy yhdeksännelle rivinumerolle.

Esimerkki:

```
200 PRINT "1 = YHTEENLASKU"
210 PRINT "2 = VÄHENNYSLASKU"
220 PRINT "3 = KERTOLASKU"
230 PRINT "4 = JAKOLASKU"
240 PRINT
250 INPUT "VALINTASI"; A%
260 IF A % < 1 OR A % > 4 THEN 250
270 ON A % GOSUB 300, 400, 500, 600
280 PRINT: GOTO 200
```

Tämä ohjelma on esimerkki valikko-ohjelmasta, jossa annettu valintavastaus siirtää ohjelman suorituksen alirutiiniin.

ON-lauseessa käytetyn muuttujan sallittujen arvojen pitää olla kokonaislukuja alkaen yhdestä ja kasvaen yhden välein.

CLEAR

MITÄ SE TARKOITTAÄ?

Tämä komento poistaa kaikki ohjelman muuttujat, vapauttaa tietokoneen FOR ja GOSUB-tilasta sekä asettaa datat uudelleen käytettäviksi.

KUINKA KIRJOITAT SEN?

```
10 CLEAR
```

SPC, TAB, PUOLIPISTE (;) JA PILKKU (,) PRINT-LAUSEESSA

SPC(xx)

SPC(xx) erottaa kaksi tulostettavaa arvoa toisistaan xx:n ilmoittamalla välilyöntimäärällä.

Esimerkki:

```
10 A = 1 : B = 2 : C = 3
20 PRINT A; SPC(4); B; SPC(6); C

RUN
1      2          3
```

TAB(xx)

TAB(xx) siirtää tulostuksen xx:n määrittelemään sarkainkohtaan, joka lasketaan aina vasemmasta reunasta.

Esimerkki:

```
10 PRINT 3; TAB(8); 4; TAB(12); 6 + 2

RUN
3          4      8
```

PILKKU (,)

Pilkku siirtää tulostuksen seuraavaan ennaltamääritelyyn sarakainkohan, jotka kuvaruudulla sijaitsevat sarakkeissa 1, 13 ja 25.

Esimerkki:

```
PRINT 1, 2, 3, 4, 5, 6
1          2          3
4          5          6
```

PUOLIPISTE (;)

PRINT-käskyn suorituksen jälkeen kursori siirtyy seuraavan rivin alkuun. Seuraavalla PRINT-käskyllä tulostetaan tälle riville. Puolipisteen avulla estetään tämä siirtyminen seuraavalle riville.

Esimerkki:

```
10 PRINT 1 + 2
20 PRINT 2*3;
30 PRINT "KUUSI"

RUN
3
6KUUSI
```

MIHIN VOIT NÄITÄ KÄYTTÄÄ?

PRINT-lauseella saadaan kuvaruuturiville 36 merkkiä. Merkkipositiot numeroidaan vasemmalta oikealle 1...36. Välimerkeillä ja sarkaimilla voit määrätä merkkien sijainnin ruuturivillä ja tehdä ulkoasusta haluamasi.

Puolipiste saa aikaan sen, että kirjoitettavat tiedot tulevat ruudulle vierekkäin, ilman tyhjää tilaa niiden välissä.

TAB-käskyllä voit muodostaa sarakkeita haluamallasi tavalla. Jos teet koko kuvaruudun täyttäviä taulukoita, määrittele lukujen tai merkkien paikat TAB-käskyllä.

Puolipisteellä erotetut merkit tulostuvat kuvaruudulle merkkipositioihin 1, 13 ja 25. Tämä toiminto vastaa samaa kuin TAB-käskyt TAB(1), TAB(13) ja TAB(25). Voidaan ajatella, että merkkipositiot tai sarakkeet 1, 13 ja 25 ovat tietokoneen sisäisesti määrätty näihin TAB-paikkoihin ja nämä sarkaimet ovat koneessasi aina valmiina.

SPC-käskyllä voit erottaa tulostettavat tekstit haluamallasi välilyöntimäärällä. Jos sarakkeisiin tulevat merkkimäärät ovat kaikki yhtä isoja, myös SPC-käskyllä aikaansaadaan siistejä taulukoita.

Näiden käskyjen käyttöä kannattaa harjoitella, sillä hallittu kuvaruutunäyttö on vaivan arvoinen asia. Kun kerran on tehnyt työn huolella, näiden käytöstä tulee tapa ja voit osoittaa todella hallitsevasi tietokoneen.

LUKU 5

MANAGERIN BASIC-KOMENNOT ÄÄNI- JA MUSIIKKIKÄSKYT

- MIKSI TIETOKONEESSA ON ÄÄNI
- SOUND
- SGEN

MIKSI TIETOKONEESSA ON ÄÄNI?

Salora MANAGERin on sisällä pieni integroitu piiri, jonka avulla voit muodostaa mitä erilaisempia ääniä. Tehdäksesi omia erityisiä ääniefektejä tai musiikkia voit käyttää komentoja SGEN tai SOUND. Integroitua piiriä kutsutaan äänigeneraattoriksi. Tässä äänigeneraattorissa on kolme äänikanavaa ja yksi kohinakanava. Voit siis muodostaa kaikenlaista ääntä, kohinaa tai musiikkia käyttämällä 4 eri kanavaa.

SOUND

MITÄ SE TARKOITTAÄ?

Tämä komento voi kytkeä päälle minkä tahansa äänigeneraattorin kanavista aivan halusi mukaan. Voit määrätä taajuuden, keston ja voimakkuuden.

KUINKA KIRJOITAT SEN?

10 SOUND (S, K, V), (S, K, V), (S, K, V), (P/J, K, V, KAN)
 kanava kanava kanava kohinakanava
 1 2 3

S = sävelkorkeus välillä 0 — 255; 0 on matalin taajuus

K = kesto välillä 0 — 255; 0 on pienin kesto

V = voimakkuus välillä 0 — 15; 15 on äänekkäin

P/J = puhdas kohina tai jaksottainen kohina; 1 on jaksottainen kohina
 2 on puhdas kohina

KAN = voit valita minkä tahansa kanavista 1 — 3 tai kohinakanavan.

Esimerkki:

```
10 FOR K = 0 TO 255 STEP 10
20 SOUND (K, 15, 10)
30 NEXT K
```

TAAJUUDEN MUUTOSTAULUKKO ON SEURAAVA:

TODELLINEN TAAJUUS = $62500/(255 - \text{sävelkorkeus})$

Taulukko on allaolevan mukainen.

SÄVELKORKEUDEN MUUTOS TAAJUUDEKSI

SÄVELKORKEUS	TAAJUUS
0	- 245.098039
10	- 255.102041
20	- 265.957447
30	- 277.777778
40	- 290.697674
50	- 304.878049
60	- 320.512821
70	- 339.837838
80	- 357.142857
90	- 378.787879
100	- 403.225807
110	- 431.034483
120	- 462.962963
130	- 500
140	- 543.478261
150	- 595.238095
160	- 657.894737
170	- 735.294118
180	- 833.333333
190	- 961.538461
200	- 1136.36364
210	- 1388.88889
220	- 1785.71429
230	- 2500
240	- 4166.66667
250	- 12500

SGEN

MITÄ SE TARKOITTAÄ?

Tämä komento antaa sinun ohjata suoraan äänigeneraattoria.

MITEN KIRJOITAT SEN?

10 SGEN A, B, C, D...

missä A, B, C, D... on datajoukko, joka lähetetään ohjaamaan äänigeneraattoria suoraan.

Seuraava ohjelma on esimerkki siitä, kuinka SGEN-komento toimii. Voit muuttaa SGEN-lauseessa olevaa dataa nähdäksesi mitä tapahtuu.

```
10 REM KELLO-OHJELMA
20 REM KYTKE POIS KAIKKI ÄÄNIKANAVAT
30 SGEN 159, 191, 223, 255
40 REM ASETA KAN1 = 679 HZ, KAN2 = 694 HZ
50 SGEN 140, 5, 170, 5
60 FOR B = 0 TO 11
70 REM B = KELLON ÄÄNEN NUMERO
80 FOR I = 140 TO 170
90 SGEN I, I + 32
100 FOR D = 0 TO 75: NEXT D
110 NEXT I
120 NEXT B
130 PRINT "ÄÄNEN LOPPU"
```

LUKU 6

EPÄMIELLYTTÄVÄ ARITMETIIKKA, JONKA TOIVOIT VÄLTTÄVÄSI

- VAKIOT, NUMEERISET MUUTTUJAT JA KOKONAISLUVUT
- MITÄ OVAT OPERAATTORIT?
 - ARITMEETTISET OPERAATTORIT
 - LOOGISET OPERAATTORIT
 - RELAATIO-OPERAATTORIT
- MIKÄ ON FUNKTIO?
 - ABS
 - ATN
 - COS
 - EXP
 - INT
 - LOG
 - RND
 - SGN
 - SIN
 - SQR
 - TAN

VAKIOT, NUMEERISET MUUTTUJAT JA KOKONAISLUVUT

VAKIOT

Aiemmissa luvuissa olet jo käyttänyt vakioita. Vakiot ovat arvoja, jotka eivät muutu. Luku 6.32 (Salora Manager käyttää desimaalipistettä eikä -pilkkaa) on vakio kuten 6.33.

Luvut voivat olla $-10E38:n$ ja $10E38:n$ välillä.

Pienin positiivinen luku on $10E-38$.

NUMEERISET MUUTTUJAT

Muuttujalle annetaan muuttuva arvo. Esimerkiksi lauseessa $Y = X + 3$ muuttujalle Y annetaan muuttujan X arvo lisättynä kolmella. X ja Y ovat muuttujanimiä. Nimi voi olla kirjain (A — Ö), kaksi kirjainta tai kirjain ja numero.

Muuttujanimen ensimmäisen merkin täytyy aina olla kirjain (A — Ö), esimerkiksi A, AB ja A6.

Jotta ohjelmalistauksia olisi helpompi ymmärtää, voi muuttujanimen pituus olla 5 merkkiä, joista kaksi ensimmäistä määrittelevät muuttujan. Esimerkiksi AA1 ja AA2 ovat sama muuttuja.

KOKONAISLUVUT

Kokonaisluvut voivat olla $-32767:n$ ja $32767:n$ välillä. Kokonaislukumuuttujan nimessä on %-merkki.

Esimerkki:

```
A% = 32767
PRINT A%
32767
```

```
A% = 12.35
PRINT A%
12
```

Kokonaisluvut tarvitsevat vähemmän muistitilaa kuin desimaaliluvut. Lisäksi niiden käsittely on nopeampaa, joten niiden käyttöä suositaan esimerkiksi grafiikkaohjelmissa.

MITÄ OVAT OPERAATTORIT?

Operaattorit työskentelevät yhdessä lausekkeen tai muuttujan kanssa tuottaakseen yhden arvon.

On olemassa kolmen tyyppisiä operaattoreita

- aritmeettisia operaattoreita
- loogisia operaattoreita
- relaatio-operaattoreita

ARITMEETTISET OPERAATTORIT

Nämä ovat tavallisia matemaattisia operaattoreita ja niiden suoritussy järjestys on määrätty. Ne ovat suoritussy järjestyksessä:

Operaattori	Operaatio	Esimerkki
**	Potenssiin korotus	$A^{**}B$
—	Negaatio	$-A$
*, /	Kertolasku	$A*B$
	Jakolasku	A/B
+, —	Yhteenlasku	$A + B$
	Vähennyslasku	$A - B$

Operaatioiden suoritussy järjestystä voi muuttaa käyttämällä sulkumerkkejä. Sulkeissa olevat lausekkeet suoritetaan ensin.

Sulkeiden sisällä noudatetaan ylläolevaa suoritussy järjestystä.

Esimerkki:	Lauseke	Tulos
	$2 + 10/2$	7
	$(2 + 10)/2$	6

LOOGISET OPERAATTORIT

Nämä operaattorit käsittelevät arvoja bitti bitiltä saadakseen tuloksen, joka on joko 1 (tosi) tai 0 (epätosi).

Loogisilla operaattoreilla on suoritusjärjestys, joka, kuten vastaava bit-tioperaatio, on seuraava:

Operaattori	A	B	Tulos
NOT	1	—	0
	0	—	1
AND	1	1	1
	1	0	0
	0	1	0
	0	0	0
OR	1	1	1
	1	0	1
	0	1	1
	0	0	0

Esimerkkejä:

63 AND 16 = 1663 on binäärisenä 111111 ja 16 on binäärisenä 10000, joten ANDin tulos on 10000 eli 16

4 AND 2 = 0 4 on binäärisenä 100 ja 2 on binäärisenä 10, on tulos 0

4 OR 2 = 6 Binääriluvuilla 100 OR 10 = 110 eli 6

NOT 0 = —1 0:n bittikomplementti on 1111 1111 1111 1111 eli desimaalisena —1

RELAATIO-OPERAATTORIT

Nämä operaattorit ovat jossain määrin samanlaisia kuin loogiset operaattorit, sillä niiden tuloksena on vain kaksi mahdollista arvoa: tosi (—1) tai epätosi (0).

Operaattori	Relaatio	Esimerkki
=	Yhtäsuuruus	$A = B$
< >	Erisuuruus	$A < > B$
<	Pienempi kuin	$A < B$
>	Suurempi kuin	$A > B$
< = tai = <	Pienempi tai yhtäsuuri kuin	$A \leq B$
> = tai = >	Suurempi tai yhtäsuuri kuin	$A \geq B$

Jos lauseke sisältää sekä aritmeettisiä, loogisia tai relaatio-operaattoreita, ne suoritetaan järjestyksessä aritmeettiset, relaatio- ja viimeiseksi loogiset operaatiot.

Sekä loogisia että relaatio-operaattoreita voidaan käyttää hyväksi tehässä päätöksiä ohjelman haaraantumisesta.

MIKÄ ON FUNKTIO?

Funktio on sääntö, joka antaa tulokseksi uuden arvon, kun sitä sovelletaan tiettyyn arvoon.

Esimerkiksi SQR on neliöjuurifunktio, joten jos kirjoitamme

PRINT SQR (9)

saamme vastaukseksi 3.

LISTA NUMEERISISTA FUNKTIOISTA JA LYHYT SELITYS NIISTÄ

- ABS(X) Antaa lausekkeen X itseisarvon
- ATN(X) Antaa tulokseksi lausekkeen X arkustangentin radiaaneissa
- COS(X) Antaa tulokseksi lausekkeen X kosinin radiaaneissa
- EXP(X) Antaa e^x :n arvon; $e = 2.71828$
- INT(X) Antaa suurimman kokonaisluvun, joka on pienempi tai yhtäsuuri kuin lauseke X
- LOG(X) Antaa lausekkeen X luonnollisen logaritmin
- RND(X) Generoi satunnaisluvun nollan ja yhden väliltä. X ohjaa satunnaislukujen generointia. $X > 0$ generoi uuden satunnaisluvun. $X = 0$ antaa aina saman satunnaisluvun. $X < 0$ aloittaa uuden satunnaislukusarjan.

- SGN(X) Antaa lausekkeen X etumerkin. Positiivinen X antaa arvon 1, negatiivinen X antaa -1 . X :n ollessa nolla on funktion arvo 0.
- SIN(X) Antaa tulokseksi lausekkeen X sinin arvon radiaaneissa
- SQR(X) Antaa X :n neliöjuuren
- TAN(X) Antaa tulokseksi lausekkeen X tangentin radiaaneissa.

LUKU 7

KAMALAT JA KÄSITTÄMÄTTÖMÄT MERKKIJONOT

- MERKKIJONOFUNKTIOT JA -MUUTTUJAT
- MERKKIJONOFUNKTIOT
 - ASC
 - CHR\$
 - GET
 - LEFT\$
 - LEN
 - MID\$
 - RIGHT\$
 - STR\$
 - VAL
- MERKKIJONOJEN VERTAILUOPERAATTORIT

MERKKIJONOFUNKTIOT JA -MUUTTUJAT

Merkeistä koostuvaa listaa kutsutaan merkkijonoksi. Esimerkiksi "MANAGER", "SUORITETAAN TESTI" ja "ABCD" ovat kaikki merkkijonoja. Aivan kuten numeerisille muuttujille voidaan merkkijonomuuttujille antaa haluttuja arvoja. Merkkijonomuuttujien muuttujanimen loppuun liitetävä "\$" -merkki erottaa ne numeerisista muuttujista.

Yritä esimerkiksi seuraavaa tekstiä:

```
A$ = "SALORA MANAGER"  
PRINT A$  
SALORA MANAGER
```

Tässä esimerkissä asetamme merkkijonomuuttujan A\$ arvoksi merkkijonon "SALORA MANAGER". Huomaa, että A\$:lle annettava merkkijono on lainausmerkkien sisällä.

Lauseessa PRINT "TIETOKONE" on teksti TIETOKONE merkkijonovakio.

MERKKIJONOFUNKTIOT

NÄIDEN FUNKTIOIDEN ARGUMENTTINA ON MERKKIJONO

ASC (X\$)	Palauttaa merkkijonolausekkeen X\$ ensimmäisen merkin ASCII-koodin desimaalilukuna (katso liitteestä ASCII-koodit sivulta 109)
CHR\$ (I)	Palauttaa argumentin I ASCII-koodia vastaavan merkin (katso liitteestä ASCII-koodit sivulta 109)
GET A\$	Lukee yhden merkin näppäimistöltä. Jos dataa tulee näppäimistöltä, se asetetaan GET-lauseessa olevaan muuttujaan. Jos dataa ei ole saatavissa, BASIC-ohjelma jatkaa odottamista.
LEFT\$(X\$,I)	Palauttaa jonolausekkeen X\$ ensimmäisestä merkistä I:nteen merkkiin ulottuvan osajonon.
LEN (X\$)	Palauttaa jonolausekkeen X\$ pituuden. Sekä tyhjät välilyönnit että merkit, jotka eivät tulostu, lasketaan mukaan pituuteen.
MID\$ (X\$,I,J)	Palauttaa jonolausekkeesta X\$ osajonon, joka alkaa I:nestä merkistä ja jonka pituus on J merkkiä
RIGHT\$ (X\$,I)	Palauttaa jonolausekkeesta X\$ osajonon, joka alkaa I:nestä merkistä lopusta lukien ja päättyen viimeiseen.

STR\$ (X)	<p>Palauttaa merkkijonon, joka on numeerisen lausekkeen X merkkijonoesitys.</p> <p>Esimerkiksi $\text{STR\\$}(1.23) = "1.23"$</p>
VAL (X\$)	<p>Palauttaa merkkijonon X\$ muunnettuna luvuksi.</p> <p>Esimerkiksi $\text{VAL}("1.23") = 1.23$</p>

MERKKIJONOJEN VERTAILUOPERAATTORIT

Merkkijonojen vertailu perustuu ASCII-koodeihin, siinä verrataan merkki kerrallaan kunnes ero löytyy. Jos kahta merkkijonoa verrattaessa toinen niistä loppuu, niin lyhyempää merkkijonoa pidetään pienempänä.

Merkkijonojen vertailuoperaattorit ovat samat kuin numeeriset relaatio-operaattorit, jotka ovat

=

<

>

< = tai = <

> = tai = >

Merkkijonojen yhdistämiseen käytetään symbolia "+". Esimerkiksi A\$=B\$+C\$. Tulokseksi saatavan merkkijonon pituuden pitää olla vähemmän kuin 256 merkkiä.

LUKU 8

MANAGERIN BASIC-KOMENNOT KOVAT KÄSKYT

- DIM
- PEEK
- POKE
- CALL
- RAM (0)
- ET-MERKKI (&)
- GET
- ONERR GOTO

DIM

MITÄ SE TARKOITTAÄ?

DIM-komento varaa muistitilaa matriiseja (taulukkoja) varten. Kaikkien matriisin alkioden arvot asetetaan nolaksi tällä komennolla.

MITEN KIRJOITAT SEN?

1. DIM muuttuja (koko 1, koko 2...)
Esim. 100 DIM A(3)
2. Matriiseissa voi olla yhdestä 255:een ulottuvuutta.
Esim. 100 DIM R3(5,5), D\$(2,2,2)
3. Matriisit voidaan dimensioida dynaamisesti ohjelman suorituksen aikana.
Esim. 100 DIM Q1(N)

Jos matriisia ei dimensioida erityisesti DIM-lauseella, niin sen oletetaan olevan yksiulotteinen matriisi, jonka ainut indeksi voi vaihdella nolasta kyhmmeneen.

PEEK

MITÄ SE TARKOITTAÄ?

PEEK-funktio palauttaa desimaalilukuna ilmoitetussa muistiosoitteessa (— 32768:n ja 32768:n välillä) olevan arvon. Palautettu arvo on välillä 0 — 255. Yritys lukea muistiosoitetta, jota ei ole olemassa, palauttaa tuntemattoman arvon.

MITEN KIRJOITAT SEN?

10 PRINT PEEK (I)

tai

10 A = PEEK(I)

jossa I on muistiosoitte desimaalisena

POKE

MITÄ SE TARKOITTAÄ?

POKE-lause tallettaa tavun muistiosoitteeseen, joka on —32768:n ja 32768:n välillä. Talletettavan tavun pitää olla välillä 0 — 255.

MITEN KIRJOITAT SEN?

10 POKE I, J

jossa I on muistipaikan osoite ja J on talletettava tavu.

VAROITUS!

POKE-lauseen huolimattoman käytön seurauksena ohjelma saattaa toimia virheellisesti, ohjelma saattaa jumiutua tai ohjelma saattaa tuhoutua.

MIHIN SITÄ KÄYTETÄÄN?

Yhdessä PEEK-funktion kanssa. Voit siirtää argumentteja konekielisiin aliohjelmiin. Voit myös käyttää PEEK- ja POKE-käskyjä muistidiagnostiikan kirjoittamiseen tai BASIC-assembleriin.

CALL

MITÄ SE TARKOITTAÄ?

Tämä käsky on yksi mahdollisuus siirrettäessä ohjelman suoritus assembler-kieliseen alirutiiniin. Alirutiinin pitää päättyä assembler-käskyn RTS, jonka seurauksena ohjelmansuoritus siirtyy BASIC-ohjelmaan.

MITEN KIRJOITAT SEN?

10 CALL I

jossa I on muistiosoite, josta konekielinen alirutiini alkaa.

PELIOHJAINTEN OHJELMOINTI

Muutaman käskyn avulla voit käyttää peliohjaimia BASIC-ohjelmassa.

```
10 CALL — 1536
20 LAUKAISU = PEEK(21)
30 VASEN = PEEK(24)
40 OIKEA = PEEK(19)
50 GOTO 10
```

Kolme muuttujaa, LAUKAISU, VASEN ja OIKEA sisältävät tiedot peliohjaimen käytöstä.

LAUKAISU

Vasen peliohjain,	vasenta laukaisunäppäintä painettu	=	17
	oikeaa	=	34
	molempia	=	51
Oikea peliohjain,	vasenta laukaisunäppäintä painettu	=	68
	oikeaa	=	136
	molempia	=	204

VASEN

Liikesuunta	ylöspäin	=	8
	alaspäin	=	2
	vasemmalle	=	32
	oikealle	=	4

OIKEA

Liikesuunta	ylöspäin	=	73
	alaspäin	=	65
	vasemmalle	=	77
	oikealle	=	69

Kaksinpelissä laukaisupainikkeiden käytön tarkistus on mahdollista rekisterien 22 ja 23 avulla.

LAUKAISE vasen peliohjain	oikea	= 1
	vasen	= 2
	molemmat	= 3

LAUKAISE oikea peliohjain	oikea	= 17
	vasen	= 34
	molemmat	= 51

Seuraavassa esimerkissä voit käyttää vasenta peliohjainta viivojen piirtämiseen kuvaruudulle. Kirjoita ohjelma, näppäile RUN ja kokeile.

```
10 HOME:GR
20 RECT (0,0), (255,191)
30 X = 128:Y = 96
40 CALL — 1536
50 A = PEEK (24) : REM VASEN PELIOHJAIN
60 IF A = 2 THEN 200
70 IF A = 4 THEN 250
80 IF A = 8 THEN 300
90 IF A = 32 THEN 350
100 GOTO 40
200 PLOT X, Y
210 Y = Y + 1
220 IF Y = 192 THEN Y = 96
230 GOTO 40
250 PLOT X, Y
260 X = X + 1
270 IF X = 256 THEN X = 128
280 GOTO 40
300 PLOT X, Y
310 Y = Y — 1
320 IF Y = — 1 THEN Y = 96
330 GOTO 40
350 PLOT X, Y
360 X = X — 1
370 IF X = — 1 THEN X = 128
380 GOTO 40
```

RAM (0)

MITÄ SE TARKOITTAÄ?

Se ilmoittaa vielä vapaan, BASIC-ohjelman käytettävissä olevan muistitavujen lukumäärän.

MITEN KIRJOITAT SEN?

10 PRINT RAM (0) — 16384

Suluissa oleva luku (0) on valeargumentti, eli sen arvo voi olla mikä tahansa mahdollinen luku tai kirjain.

Funktion antama tulos on 16 ktavua suurempi kuin vapaa BASIC-muisti.

MIHIN SITÄ KÄYTETÄÄN?

Kirjoittamalla komennon A = RAM(A\$) saat käyttöösi lisää muistia, sillä se yhdistää erillään olevat, vapaat muistitavut yhdeksi jatkuvaksi muisti-alueeksi. Lisäksi ohjelman suoritus nopeutuu.

ET-MERKKI (&)

MITÄ SE TARKOITTAÄ?

Sillä siirretään ohjelman suoritus assembler-kieliseen alirutiiniin, joka alkaa muistiosoitteesta \$03F4.

MITEN KIRJOITAT SEN?

10 &NIMI 100

MIHIN SITÄ KÄYTETÄÄN?

Se antaa assembler-taitoiselle ohjelmoijalle mahdollisuuden määritellä omia käskyjä.

GET

MITÄ SE TARKOITTAÄ?

GET-käskyn avulla luetaan merkki (kirjain, numero tai erikoismerkki) näppäimistöltä. Ohjelman suoritus pysähtyy siihen asti, kunnes painetaan jotakin näppäintä (RETURNia ei tarvita, se on yksi erikoismerkeistä). Kuvaruudulla ei näy syötettyä merkkiä.

MITEN KIRJOITAT SEN?

10 GET A

tai

10 GET A\$

Esimerkki:

```
10 INPUT "ANNA KAKSI LUKUA"; A, B
20 PRINT "*" = KERTOLASKU"
30 PRINT "/" = JAKOLASKU"
40 GET A$
50 IF A$ = "*" THEN PRINT A*B:GOTO 10
60 PRINT A/B
70 GOTO 10
```

Huom: Käsky GET A odottaa syötteeksi vain lukua nollasta yhdeksään.

ON ERR GOTO

MITÄ SE TARKOITTA?

Tämän lauseen avulla voit itse käsitellä ohjelmassa tapahtuvia virheitä. Käselyn pitää olla ohjelmassa ennen mahdollisen virheen esiintymistä ja sen avulla haaraudutaan ohjelman osaan, jossa virhe käsitellään.

MITEN KIRJOITAT SEN?

10 ON ERR GOTO 1000

MIHIN SITÄ KÄYTETÄÄN?

Voit korjata ohjelmassa olevat virheet oikeilla BASIC-käskyillä tai tulostaa virheilmoituksen kuvaruudulle.

Esimerkki:

```
10 ON ERR GOTO 100
20 A = 0 : B = 0
30 PRINT B/A
40 GOTO 20
90 REM VIRHEEN KÄSITTELY
100 IF PEEK(222) = 133 THEN A = 1E-37
110 POKE 216,0 : GOTO 30
```

```
RUN
1E+38
? DIVISION BY ZERO ERROR IN 30
```

Virheenkäsittelyosassa tutkitaan muistipaikasta 222 virheen syy (virheiden koodit löytyvät kirjan lopussa olevasta liitteestä, sivulla 102). Mahdollisesti tarpeellinen ON ERR -funktion paluulause on mahdollinen POKE 216,0 -lauseen avulla.

LUKU 9

MANAGERIN BASIC-KOMENNOT KASSETTINAUHURIIN LIITTYVÄT KOMENNOT

- MITÄ VOIT TEHDÄ NAUHURILLA?
- CLOAD
- CSAVE
- CRUN
- BLOAD
- BSAVE
- BRUN
- STORE
- RECALL

MITÄ VOIT TEHDÄ NAUHURILLA?

Se antaa sinulle jälleen mahdollisuuden olla laiska. Voit panna tietokoneesi ohjelmiesi kimppuun tekemään kaikenlaisia pikku askareita ja saada sen väsymään ja rasittumaan liikaa samalla kun sinä vain istut myhäilemässä. Tässä on taas yksi syy, jonka takia kaikki turhauttavat ohjelmointitunnit kannattivat.

TARVITSET KASETTINAUHURIN, ENNENKUIN TÄLLÄ LUVULLA ON MERKITYSTÄ SINULLE

Näiden komentojen avulla voit tallettaa ohjelmia kasettinauhurille sekä ladata niitä sieltä. PIKAOPPAASSA kerrotaan, miten yhdistät nauhurin Salora Manageriin. Kun se on liitetty ja se toimii, voit käyttää tässä luvussa olevia komentoja.

CLOAD

MITÄ SE TARKOITTAÄ?

Se tarkoittaa aivan yksinkertaisesti lataamista (LOAD). C tarkoittaa kasettia (Cassette). Jos olet valmis ajamaan nauhalla olevia ohjelmia, vaikkapa joitakin omia ohjelmiasi, jotka olet tallettanut, laitat vain nauhurisi valmiiksi ja painat toistonäppäintä (play). Sen jälkeen kirjoitat CLOAD näppäimistöltäsi.

Kasetin pitäisi alkaa liikkua ja tekstin "LOADING PROGRAM" pitäisi ilmestyä kuvaruudulle.

OHJELMIEN LATAUS

Kirjoita

CLOAD

CSAVE

MITÄ SE TARKOITTAÄ?

Tämän komennon avulla voit tallettaa (SAVE) ohjelmia kasetilla. C tarkoittaa kasettia (Cassette).

Kirjoitat vain

CSAVE

ja tietokone siirtää ohjelman nauhalle. Nauhurin pitää olla nauhoitustilassa.

CRUN

MITÄ SE TARKOITTAÄ?

Tämä komento ohjaa tietokonetta lataamaan ohjelman kasettinauhalta ja välittömästi latauksen jälkeen aloittamaan sen suorituksen (RUN). Kun käytät CLOAD-komentoa, tietokone suorittaa vain latauksen. Sinun pitää itse lisätä RUN-komento.

Kirjoitat vain

CRUN

BLOAD

BLOAD-komento ohjaa tietokonetta lataamaan (LOAD) BINÄÄRIohjelman tai datatiedoston nauhalta. Enimmäkseen binääriohjelmat ovat pelejä, mutta ne voivat sisältää muuta ohjelmistoa kuten tekstinkäsittelyä tai taulukkolaskentaohjelmia.

Syy siihen, että sinun pitää käyttää BLOAD-komentoa, on se, että tietokoneesi käyttää enimmäkseen BASICia ja se ei tunnista binääristä kieltä, jollet kerro sitä.

Ladataksesi binääriohjelman nauhalta kirjoita

BLOAD

tai

BLOAD \$A

jossa A on muistipaikan heksadesimaalinen alkuosoite, josta alkaen haluat tallettaa ohjelmasi.

BSAVE

MITÄ SE TARKOITTAA?

Tämän komennon avulla voit tallettaa (SAVE) BINÄÄRIohjelman kasetille. Ennenkuin voit tallettaa binääriohjelman, sinun pitää tietää missä ohjelma sijaitsee sekä kuinka pitkä se on.

Tallettaaksesi binääriohjelman kirjoitat

BSAVE \$A, \$L

missä A on ohjelman alkuosoite heksadesimaalisena ja L on ohjelman pituus heksadesimaalisena.

BRUN

MITÄ SE TARKOITTAA?

Aivan kuten CRUN, tämä komento ohjaa tietokonetta aloittamaan ohjelman suorituksen heti kun se on ladattu kasetilta. Ainoa ero on se, että BRUN lataa ja käynnistää binääriohjelmia eikä BASIC-ohjelmia.

Kirjoitat vain

BRUN

tai

BRUN \$A

jossa A on muistipaikan heksadesimaalinen alkuosoite, josta alkaen haluat tallettaa ohjelmasi.

STORE

MITÄ SE TARKOITTA?

Tämä komento lähettää määrätyn muuttujan arvon tai dataa kasetille. Sitä suoritettaessa pitää nauhurin luonnollisesti olla liitettynä kunnolla tietokoneeseen ja sen pitää olla nauhoitustilassa, kun tämä komento suoritetaan.

MITEN KIRJOITAT SEN?

10 STORE A\$

MIHIN SITÄ KÄYTETÄÄN?

Tätä komentoa käytetään talletettaessa dataa kasetille. Se on varsin so-piva sellaisissa ohjelmissa kuten esimerkiksi postitusluettelo. Yleensä kannattaa varmuuden vuoksi muuttaa numeeriset muuttujat merkkijo-noiksi STR\$-funktion avulla.

RECALL

MITÄ SE TARKOITTA?

Se lukee määrättyä kasetille talletettua dataa sekä asettaa näin saadut arvot annetuille muuttujille.

MITEN KIRJOITAT SEN?

10 RECALL A\$

MIHIN SITÄ KÄYTETÄÄN?

Tätä komentoa käytetään yhdessä STORE-komennon kanssa luettaessa ja kirjoitettaessa dataa kasetille.

LUKU 10

MANAGERIN BASIC-KOMENNOT KIRJOITINKOMENNOT

- LLIST
- LPRINT

TARVITSET KIRJOITTIMEN, ENNENKUIN SINULLA ON HYÖTYÄ TÄSTÄ LUVUSTA

Heti kun olet hankkinut kirjoittimen, katso PIKAOPPAASTA miten se liitetään. Kun kirjoitin on liitetty oikein, voit näiden kommentojen avulla tulostaa ohjelmia, tekstiä ja jopa grafiikkaa.

LLIST

Tätä käytetään LIST-komennon sijasta, jotta saadaan listaus koko ohjelmasta paperille. Jos rivinumerot on määrätty, voit käyttää LLIST-komentoa ja saat haluamasi ohjelmanpätkän kirjoittimellesi.

MITEN KIRJOITAT SEN?

LLIST

MITÄ VOIT TEHDÄ SILLÄ?

Tällä komennolla on kaksi pääfunktiota. Toinen on se, että saat paperikopion ohjelmastasi dokumentointia varten. Toinen on virheiden etsintä ohjelmasta, koska useimmat BASIC-ohjelmat ovat liian pitkiä, jotta ne mahtuisivat kerralla kuvaruudulle. Kun sinulla LLIST:illä otettu paperikopio, sinulla on koko ohjelma tulostettuna silmiesi edessä sen sijaan, että katselet sitä pieninä pätkinä kuvaruudulla.

LPRINT

MITÄ SE TARKOITTAÄ?

Tämä komento käskee tietokoneen kirjoittamaan kirjoittimelle eikä kuvaruudulle. Voit käyttää LPRINT-käskyä ohjelmassa, saadaksesi tulostuksia paperille ohjelman suorituksen aikana. Voit myös käyttää LPRINT-komentoa tulostaaksesi paperille kaiken sen mitä on kuvaruudulla (jälkimmäisessä tapauksessa et tarvitse rivinumeroa).

MITEN KIRJOITAT SEN?

```
10 LET A = 5
20 LET B = 6
30 LPRINT A + B
40 END
```

HUOM! Aivan kuin PRINT-komennossa, jos haluat tulostaa tekstiä, on sinun laitettava lainausmerkit tekstin ympärille. Esimerkiksi ylläolevassa ohjelmassa tulostuva vastaus on 11, mutta jos muutat rivin 30 riviksi

```
30 LPRINT "A + B"
```

vastaus olisi A + B.

MITÄ VOIT TEHDÄ SILLÄ?

Käyttääksesi kirjoitintasi siihen, mihin ostit sen.

LUKU 11

- VIRHEIDEN NUMEROKOODIT
- VIRHEILMOITUKSET
JA MIKSI NE KIUSSAavat SINUA
- ASCII-KOODITAUUKKO
- YHTEENVETO BASIC-KOMENNOISTA
- KÄSITTEITÄ JA TERMEJÄ

VIRHEIDEN NUMEROKOODIT ON ERR -TOIMINNOISSA

107 BAD SUBSCRIPT
210 CAN'T CONTINUE
133 DIVISION BY ZERO
191 FORMULA TOO COMPLEX
149 ILLEGAL DIRECT
53 ILLEGAL QUANTITY
0 NEXT WITHOUT FOR
42 OUT OF DATA
77 OUT OF MEMORY
69 OVERFLOW
120 REDIM'D ARRAY
22 RETURN WITHOUT GOSUB
176 STRING TOO LONG
16 SYNTAX ERROR
163 TYPE MISMATCH
90 UNDEFINED STATEMENT

VIRHEILMOITUKSET JA MIKSI NE KIUSAAVAT SINUA

107 BAD SUBSCRIPT (VIRHEELLINEN ALAINDEKSI)

Yritettiin viitata sääntiöalkioon (taulukon tai matriisin alkioon), joka on sääntiön tilavarauksen (DIM) ulkopuolella. Tämä virhe voi syntyä, jos viitattaessa matriisiin käytetään väärää indeksimäärää. Esimerkiksi $LET A(1,1,1) = Z$, kun A on dimensioitu käyttäen DIM A(2,2)

210 CAN'T CONTINUE (OHJELMAN SUORITUSTA EI VOI JATKAA)

Yritetään jatkaa CONT-komentoa käyttäen ohjelmaa, jota ei ole olemassa, tai virheen jälkeen tai kun ohjelmasta on poistettu rivi, siihen on lisätty rivi tai riviä on muutettu.

133 DIVISION BY ZERO (JAKOLASKU, JOSSA JAKAJA ON NOLLA)

Ohjelma-ajon aikana saavutaan lauseeseen, jonka jakajan arvo on nolla. Nollalla jakaminen on virhe ja tietokone antaa virheilmoituksen.

191 FORMULA TOO COMPLEX (LASKUKAAVA ON LIIAN MONIMUTKAINEN)

STRING FORMULA TOO COMPLEX (MERKKIJONON KAAVA ON LIIAN MONIMUTKAINEN)

Suoritetaan enemmän kuin kaksi lausetta, jotka ovat muotoa IF "XX" THEN tai merkkijonon kaava on liian pitkä tai monimutkainen. Merkkijonokaava on jaettava kahteen yksinkertaisempaan ilmaisuun.

149 ILLEGAL DIRECT (LAITON KOMENTO)

Et voi käyttää INPUT, GET tai DATA-lausetta välittömänä komentona (toisin sanoen näitä lauseita ei voi käyttää ilman rivinumeroa).

59 ILLEGAL QUANTITY (LAITON SUURE)

Antamasi parametri matemaattiselle tai merkkijonofunktiolle on sallittujen rajojen ulkopuolella. ILLEGAL QUANTITY voi tulla seuraavista syistä:

- a) Negatiivinen alaindeksi sääntiössä, esimerkiksi $LET A(-1) = 0$
- b) Kun LOG:n argumentti on nolla tai negatiivinen

- c) SQR:n argumentti on negatiivinen
- d) Kun lausekkeessa $A \cdot B$ arvo A on negatiivinen eikä B ole kokonaisluku
- e) Kutsutaan USR-toimintaa jolla on virheellinen alkuosoite.
- f) Virheellinen argumentti seuraavissa: MID\$, LEFT\$, RIGHT\$, PEEK, POKE, TAB, SPC, ON...GOTO tai missä tahansa grafiikkafunktiossa.

0 NEXT WITHOUT FOR (NEXT, JOTA EI EDELLÄ FOR)

NEXT-lauseessa oleva muuttuja ei vastaa suoritettavassa FOR-lauseessa olevaa muuttujaa tai nimetön NEXT ei vastaa ainoatakaan parast'aikaa suoritettavaa FOR- lausetta.

42 OUT OF DATA (LUETAAN READ-KÄSKYLLÄ ENEMMÄN DATAA KUIN DATA-LAUSEESSA ON)

Suoritetaan READ-lause, mutta kaikki ohjelmassa olevat DATA-lauseet on jo luettu. Ohjelma yrittää lukea liian paljon dataa tai ohjelmassa on riittämättömästi dataa.

77 OUT OF MEMORY (MUISTITILA LOPPU)

Mikä tahansa seuraavista voi aiheuttaa tämän virheen: Liian pitkä ohjelma, liian paljon muuttujia, sisäkkäisiä FOR...NEXT-silmukoita enemmän kuin 10 tasoa, sisäkkäisiä GOSUB-lauseita enemmän kuin 24 tasoa, liian monimutkainen lauseke, sisäkkäisiä sulkeita enemmän kuin 36 tasoa.

69 OVERFLOW (YLIVUOTO)

Laskutulos oli liian suuri, jotta se olisi pystytty esittämään BASICin numeromuodossa. Jos ylivuoto tapahtuu, tulokseksi tulee nolla ja ohjelman suoritus jatkuu ilman virheilmoituksen tulostusta.

120 REDIM'D ARRAY (SÄÄNTIÖ MÄÄRITELLÄÄN UUELLEEN)

Sen jälkeen kun sääntiö on dimensioitu (sääntiön alkioille on suoritettu tilavaraus), suoritetaan toinen DIM-lause samalle sääntiölle. Tämä virhe tapahtuu usein, kun sääntiölle on annettu oletusdimensio 10, koska lausetta kuten $A(1) = 3$ seuraa myöhemmin ohjelmassa $\text{DIM } A(100)$. Tämä virheilmoitus voi osoittautua hyödylliseksi, jos haluat selvittää millä ohjelmarivillä määrätty sääntiö dimensioidaan; laitat vain kyseistä sääntiötä koskevan DIM-lauseen ensimmäiselle riville, aja ohjelma ja tietokone kertoo sinulle missä alkuperäinen DIM-lause on.

22 RETURN WITHOUT GOSUB (RETURN, JOHON EI TULLA GOSUB-KÄSKYN KAUTTA)

Suoritetaan RETURN-lause ilman että vastaavaa GOSUB-lausetta ei ole suoritettu.

176 STRING TOO LONG (MERKKIJONO ON LIIAN PITKÄ)

Yritetään käyttää merkkijonojen yhdistämistä luotaessa merkkijono, joka on pitempi kuin 255 merkkiä.

16 SYNTAX ERROR (SYNTAKSIVIRHE)

Syntaksi- eli muutosääntövirhe ilmoitetaan jos kirjoitetussa käskyssä tai välittömässä komennossa on kirjoitusvirhe tai jos lausekkeesta puuttuu sulkumerkki, rivillä on laiton merkki, virheellinen välimerkki jne.

163 TYPE MISMATCH (TYYPIT EROAVAT TOISISTAAN)

Asetuslauseen vasen puoli oli numeerinen muuttuja ja oikea puoli oli merkkijono, tai päinvastoin; tai funktiolle, jolle odotettiin argumenttina merkkijonoa, annettiin numeerinen argumentti, tai päinvastoin.

90 UNDEF'D STATEMENT (MÄÄRITTELEMÄTÖN LAUSE)

Yritetään suorittaa GOTO-, GOSUB- tai THEN-lause riville, jota ei ole olemassa.

MITÄ NUMEROKOODIT TARKOITTAVAT?

Jokaiselle kuvaruudulle tulostuvalle merkille on tietty numerokoodinsa. Seuraavassa taulukossa on esitetty merkkejä vastaavat koodit.

MITEN VOIT KÄYTTÄÄ NÄITÄ KOODEJA?

Esimerkki:

```
PRINT CHR$(65)
```

```
A
```

```
PRINT "123"; CHR$(24); "456"
```

```
123
```

```
456
```

Tulostus siirtyy yhden rivin alaspäin (numerokoodi 24 on rivinsiirto alaspäin).

TAULUKKO:

HEX	0	1	2	3	4	5	6	7
LSB MSB	000	001	010	011	100	101	110	111
0	0000		SPC	0	@	P	↘	P
			32	48	64	80	96	112
1	0001		!	1	A	Q	a	q
			33	49	65	81	97	113
2	0010		”	2	B	R	b	r
			34	50	66	82	98	114
3	0011		#	3	C	S	c	s
			35	51	67	83	99	115
4	0100		\$	4	D	T	d	t
			36	52	68	84	100	116
5	0101	csr ↑ 5	csr ← 21	%	5	E	U	e
			37	53	69	85	101	117
6	0110		&	6	F	V	u	v
			38	54	70	86	102	118
7	0111		'	7	G	W	g	w
			39	55	71	87	103	119
8	1000	csr → 8	csr ↓ 24	(8	H	X	h
			40	56	72	88	104	120
9	1001)	9	I	Y	i	y
			41	57	73	89	105	121
A	1010		*	:	J	Z	j	z
			42	58	74	90	106	122
B	1011		+	;	K	[k	{
			43	59	75	91	107	123
C	1100		,	<	L	Ä	1	ä
			44	60	76	92	108	124
D	1101	CR 13	—	=	M	Ö	m	ö
			45	61	77	93	109	125
E	1110		.	>	N	Å	n	å
			46	62	78	94	110	126
F	1111		/	?	0	?	o	■
			47	63	79	95	111	127

Edellisen sivun taulukko on esitetty sivuilla 109 ja 110 hieman toisessa muodossa.

Katso tarvittaessa heksadesimaali- tai binäärikoodit sivun 107 taulukosta, sivuilla 109 — 110 olemme esittäneet ainoastaan merkkien desimaalikoodit sekä kyseisen toiminnon.

Jos haluat muodostaa omia kuvaruutumerkkejä kuten esitimme luvussa 2 sivulla 41, käytä liitteen 2 heksadesimaali-desimaalimuunnostaulukkoa. Voit tehdä omia merkkejäsi vaikka joka näppäimellä, mutta pidä mielessäsi — millä silloin jatkamme ohjelman tekoa?

ASCII-KOODITAUUKKO

ASCII-

koodi	Merkki
5	↑ (rivinsiirto ylös)
8	→ (siirto oikealle)
13	CR (Carriage Return, siirto rivin alkuun)
21	← (siirto vasemmalle)
24	↓ (siirto alas)
32	(tyhjä, välilyönti)
33	! (huutomerkki)
34	" (lainausmerkki)
35	# (lukumäärä, "risuaitamerkki")
36	\$ (dollari)
37	% (prosentti)
38	& (et-merkki)
39	' (apostrofi)
40	((alkusulkumerkki)
41) (loppusulkumerkki)
42	* (tähti, asteriski)
43	+
44	,
45	— (miinus)
46	.
47	/ (vinoviiva)
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7

ASCII-

koodi	Merkki
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W

**ASCII-
koodi Merkki**

88	X
89	Y
90	Z
91	Ä
92	Ö
93	Å
94	^ (circonflexi)
95	_ (alaviiva)
96	` (aksentti)
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	p
109	m
110	n
111	o

**ASCII-
koodi Merkki**

112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z
123	ä
124	ö
125	å
127	(merkin poisto)

YHTEENVETO BASIC-KOMENNOISTA

Kirjan lopussa olevassa liitteessä on lisäksi kaikkien komentojen aakkosellinen hakemisto.

FUNKTIOT:

1. Aritmeettiset operaattorit

+ , — , * , / , **

2. Relaatio-operaattorit

> , < , = , > = , < = , < >

3. Aritmeettiset funktiot

ABS - itseisarvo
ATN - arkustangentti
COS - kosini
EXP - e^x
INT - kokonaisosa
LOG - luonnollinen logaritmi
RND - satunnaisluku
SGN - etumerkki
SIN - sini
SQR - neliöjuuri
TAN - tangentti

4. Merkkijonofunktiot

ASC - ASCII-arvo
CHR\$ - merkki
GET - merkin luku näppäimistöltä
LEFT\$ - vasemmanpuoleiset merkit
LEN - pituus
MID\$ - merkit keskeltä
RIGHT\$ - oikeanpuoleiset merkit
STR\$ - numeerinen argumentti merkkijonoksi
VAL - merkkijonon numeerinen arvo

5. Loogiset operaattorit

AND Relaatio- ja loogisten lausekkeiden arvo on 1 tai —1 (ei nolla), jos
OR ne ovat tosia, 0, jos ne ovat epätosia.
NOT

KÄSKYT JA KOMENNOT

6. Grafiikkakäskyt

GR
TEXT
COLOR =
PLOT
PLOT...TO
UNPLOT
UNPLOT...TO
CIRCLE
RECT
HOME
VPEEK
VPOKE

7. Ääni- ja musiikkikäskyt

SOUND
SGEN

8. Komennot

END
NEW
LET
REM
PRINT
INPUT
LIST
STOP
CONT
BREAK
IF...THEN
FOR...TO...STEP...NEXT
RAM
READ...DATA
RESTORE
GOTO
GOSUB...RETURN
ON...GOTO/ON...GOSUB
ONERR
CLEAR
DIM

SPC
TAB
PEEK
POKE
CALL
TRACE
NOTRACE
&

9. Kasettinauhuriin liittyvät komennot

CLOAD
CSAVE
CRUN
BLOAD
BSAVE
BRUN
STORE
RECALL

10. Kirjoitinkomennot

LLIST
LPRINT

KÄSITTEITÄ JA TERMEJÄ

Salora Managerin BASIC-käsikirjassa ja Pikaoppaassa käytettyjä käsitteitä ja termejä

Tässä luvussa selostetaan tietokoneen mukana tulleissa kirjoissa käytettyjä termejä. Jokaisen suomenkielisen termin jäljessä on sulkeissa vastaava englanninkielinen sana. Tästä saattaa olla hyötyä tutustuttaessa alan runsaaseen kirjallisuuteen, joka on valtaosin englanninkielistä.

Aakkosmerkki (Alphabetic Character) — Aakkosiin kuuluva kirjain.

Aakkosnumeerinen (Alphanumeric, Alphameric) — Vrt. myös alfanumeerinen. Merkkien muodostama sarja, joka sisältää sekä kirjaimia että numeroita.

Ajaa (RUN) — Suorittaa muistissa oleva ohjelma tai sen osa.

Alaindeksi (Subscript) — Ks. määrite.

Alfanumeerinen (Alphanumeric, Alphameric) — Ks. aakkosnumeerinen.

Algoritmi (Algorithm) — Sarja toimintaohjeita, joilla tehtävä ratkaistaan askel askeleelta.

Aliohjelma (Subprogram) — Valmiiksiohjelmoitu yleiskäyttöinen toimintotarja, jonka käyttäjä voi kutsua BASICissa CALL-lauseella. Aliohjelmat lisäävät BASICin suorituskykyä, ja niiden ohjelmointi BASICilla on hankalaa.

Alirutiini (Subroutine) — Ohjelmaosa jota voidaan käyttää useaan kertaan ohjelmansuorituksen aikana. Alirutiineja voidaan käyttää mm. mutkikkaissa laskutoimitussarjoissa tai ohjelman tulostusosana. BASICissa alirutiini kutsutaan GOSUB-lauseella ja se päättyy paluulauseeseen (RETURN).

Alkio (Element) — Joukon osa, tavallisesti yksi sääntiön tai taulukon arvoista.

Alustaa I (Format) — Toimenpiteet, jolla levyke otetaan käyttöön eli formatoidaan laitteelle sopivaksi.

Alustaa II (Initialize) — Toimenpide, jolla laskurit, kytkimet, osoitteet ja muistin sisältö asetetaan alkuarvoihinsa ohjelman tai rutiinin alussa.

Argumentti (Argument) — Arvo, joka on siirrettävissä ohjelmasta toiseen tai riippumaton muuttuja funktiossa.

ASCII (ASCII) — Lyhenne sanoista American Standard Code for Information Interchange. Useimmissa henkilökohtaisissa tietokoneissa käytetty aakkosnumeeristen merkkien sekä erikoismerkkien esitykseen tai siirtoon käytetty 8-bittinen koodi.

Asychroninen (Asynchronous) — Tahdistamaton. Viittaa yleensä tiedonsiirtoon liittyviin oheislaitteisiin.

Attribuutti (Attribute) — Määrite, ominaisuus.

Avainsana (Keyword) — Ohjelmointikielelle määritetty sana, ks. varattu sana.

BASIC (BASIC) — Helppokäyttöinen ja suosittu ohjelmointikieli, jota käytetään useimmissa kotitietokoneissa. "BASIC" on lyhenne sanoista Beginners All Purpose Symbolic Instruction Code. Microsoft* BASIC on ehkä maailman laajimmin käytössä oleva BASIC-"murre". Kaikissa BASIC-kielissä käskykanta on samantapainen, mutta lauseet ja komennot poikkeavat jonkin verran toisistaan.

Baudi (Baud) — Tiedonsiirtonopeuden mittayksikkö, bittiä sekunnissa.

Binääri (Binary) — Kaksiarvoinen, kaksijärjestelmään kuuluva. Kaksijärjestelmässä käytetään pelkästään lukuja (bittejä) 0 ja 1. Tietokoneen sisäinen kieli käyttää kaksijärjestelmää.

Bitti (Bit) — Binäärisen luvun osa, joko 0 tai 1.

Boolean arvo (Boolean Value) — Boolean algebraan perustuva looginen arvo, joka on joko tosi tai epätosi.

Bps (Bits per Second) — Bittiä sekunnissa, ks. baudi.

Data (Data) — Tietokoneen käsiteltävissä oleva tieto.

Edusta (Foreground) — Kuvaruudun alue, jota merkki edustaa itse. Ks. myös tausta.

Eksponentti (Exponent) — Luku joka ilmoittaa mihin potenssiin luku tai lauseke on korotettava. Se merkitään tavallisesti korotettavan luvun oikeaan ylänurkkaan. Esimerkiksi $2^8 = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$. Eksponentti syötetään joko "***"-merkin tai E-kirjaimen jäljessä. Esimerkiksi $2^8 = 2^{**}8$, ja $1,3 \times 10^{25} = 1.3E25$.

End-of-file (End-of-file) — Tila jossa kaikki tiedostossa oleva data on luetu.

Eroin (Delimiter) — Ks. myös rajamerkki. Merkki, joka jakaa ryhmiin tai erottaa arvot toisistaan.

Etu- (Leading) — Tietueen tai kentän vasemmanpuoleisin osa (Etunollat, Leading Zeros).

Erottelukyky, erotuskyky (Resolution) — Tietokonegrafiikan tarkkuuden mitta. Erottelukyky (resoluutio) ilmaistaan luvulla, joka kertoo montako erillistä merkkiä (pistettä tai viivaa) kuvaruudulle voidaan tulostaa, esim. 256x192 merkkiä. Jos näyttölaitteena käytetään monitoria eikä tavallista TV-vastaanotinta, erottelukyky voi olla huomattavasti parempi mikäli tietokoneen ominaisuudet sen sallivat.

Funktio (Function) — Määritelmä joka antaa tietyn arvon joko yhdelle tai useammalle riippumattomalle muuttujalle. Esimerkiksi sini ja kosini ovat numeerisia funktioita jotka määräytyvät kulman suuruuden mukaan.

Funktionäppäin (Function Key) — Henkilökohtaisissa tietokoneissa yleensä näppäimistön yläosassa olevat näppäimet, jotka tavallisesti on merkitty esim. F1...F10.

Grafiikka (Graphics) — Aineisto, joka saadaan kuvaruudulle normaalien kirjainmerkkien lisäksi kaaviot, liikkuvat ja liikkumattomat kuvat.

Haarautuminen — Ks. Hyppy.

Heksadesimaalinen (Hexadecimal) — Kuusitoistajärjestelmään kuuluva. Heksadesimaalijärjestelmässä on kuusitoista merkkiä: 0-9 ja A-F. Binäärijärjestelmän luku 1010 on heksadesimaalilukuna A. Binääriluku 1111111 on 16-järjestelmässä FF.

Hertsi, Hz (Herz, Hz) — Taajuuden yksikkö, yksi sykäys (tapahtuma) sekunnissa.

Hyppy (Branch) — Poikkeaminen ohjelman suorituksessa lauseiden numerojärjestyksestä. Pakollinen hyppylause saa ohjelmasuorituksen siirtymään poikkeuksetta hyppylauseeseen annettuun osoitteeseen. Ehdolli-

nen hyppy tapahtuu jonkin aritmeettisen tai loogisen päätelmän perusteella.

Iterointi (Iteration) — Menetelmä jossa toistetaan joitakin ohjelmalauseita. Yhtä toistokertaa nimitetään iterointikierrökseksi. Ks. myös silmukka.

Itselataava ohjelma (Bootstrap Program) — Ohjelma, jolla laitteisto saadaan toimintakuntoon.

Jono I (Queue) — Rivi tai lista kohtia, jotka odottavat toimenpiteitä.

Jono II (String) — Ks. merkkijono.

Kahden komplementti (Two's Complement) — Negatiivisen luvun esitysmuoto binäärijärjestelmässä.

Kaksisuuntainen (Duplex) — Tiedonsiirrossa ja viestinnässä tapahtuva kaksisuuntainen kommunikointi, jossa samanaikaisesti voi tapahtua lähetys kumpaankin suuntaan.

Kanava (Channel) — Väylä, jota pitkin tieto siirretään.

Kantaluku-100 (Radix-100) — Sataan perustuva lukujärjestelmä.

Kello (Clock) — Tietokoneessa oleva laite, jolla generoidaan säännöllisiä pulsseja. Signaalia kutsutaan kellosignaaliksi, ja sen avulla sovitetaan sähköiset toiminnot toisiinsa.

Kenttä (Field) — Tiedoston määrätty osa nauhalla tai levykkeellä ja johon talletetaan vain tietyn tyyppistä tietoa.

Kerros, kerrosta (Overlay) — Sama muistialue käytetään hyväksi ohjelman eri osissa ja eri aikaan.

Keskeytys (Interrupt) — Toiminnan tai ohjelman keskeytys siten, että se voidaan käynnistää uudelleen.

Keskeytyskohta (Breakpoint) — Ohjelmakohta, johon BREAK-komento saa ohjelmasuorituksen pysähtymään. Keskeytyksen aikana käyttäjä voi käyttää tietokonetta komentotilassa. Ohjelmakeskeytystä voidaan käyttää mm. ohjelmien vianetsintään. Ohjelmansuoritus jatkuu CONT-komennolla.

Kilo, kilotavu (K, Kbyte) — Tietokoneen tai massamuistivälineen muistin määrä. 1 kilotavu on 1024 (2^{10} tavua tai merkkiä).

Kirjasin, kirjasinlaji (Font) — Merkki, joka on ulkoasultaan tietynlainen.

Kirjoite (Hard Copy) — Kirjoittimella tai piirturilla tehty tulostus.

Kohdistin (Cursor) — Ks. myös kursori. Symboli joka ilmoittaa mihin kohtaan kuvaruutua seuraava näppäimenpainallus tuo merkin.

Kokonaisluku (Integer) — Luku jolla ei ole desimaaliosaa. Se voi olla joko positiivinen, negatiivinen tai nolla.

Komento (Command) — Tietokoneen välittömästi suorittama toiminto. Komennot eivät voi sisältyä ohjelmaan ja ne syötetään ilman rivinumeroa.

Komentotila (Command Mode) — Kun tietokone ei suorita ohjelmaa, se on komentotilassa. Käyttäjän syöttämät komennot suoritetaan välittömästi.

Kommunikointi (Communication) — Ks. viestintä.

Komplementti (Complement) — Tietokone muodostaa binääriluvusta toisen vaihtamalla luvun nollat ykkösiksi ja ykköset nolliksi.

Kontrollimerkki (Control Character) — Erikoismerkki, joka kontrolloi oheislaitetta, esim. kirjoitinta.

Koordinaatisto (Co-ordinates) — Kahden tai useamman numeron sarja, joka määrittelee pisteen paikan kaksi- tai useampiulotteisessa avaruudessa.

Korjailu (Debug) — Virheiden etsintä ja poistaminen ohjelmasta.

Kovo (Hardware) — Katso laitteisto.

Kursori (Cursor) — Ks. myös kohdistin. Symboli, joka ilmoittaa mihin kohtaan kuvaruutua seuraava näppäimen painallus tuo merkin.

Kutsu (Call) — Ohjelman tai aliohjelman kutsuminen, ks. aliohjelma.

Laitteisto (Hardware) — Tietokone ja siihen yhteydessä olevat laitteet, kuten lisämuisti, levy-yksikkö, kirjoitin jne.

Laskentakapasiteetin ylitys — Ks. Ylitys.

Lause (Statement) — Ohjelmakäsky, jota edeltää rivinnumero. Samalla rivillä voi olla useampia lauseita.

Lauseke (Expression) — Vakioiden, muuttujien ja operaattoreiden yhdistelmä, joka voidaan ratkaista siten, että saadaan jokin merkityksellinen tulos. Lausekkeita ovat numero-, jono- ja suhteelliset lausekkeet.

Levy, levyke (Disc, Floppy Disc) — Massamuistina käytettävä tietoväline. Joko ohut kalvolevy (Floppy Disc, Diskette) tai levy (Disc).

Liittäminen (Concatenation) — Ks. Yhdistäminen (koskee jonojen yhdistämistä).

Liitäntä, liityntä (Interface) — Väline tai laitteisto, jolla kaksi laitetta liitetään toisiinsa.

Lippu, liputtaa (Flag) — Erityyppiset ilmaisimet, joita käytetään määrittäessä ohjelman kulkua.

Lisäys (Increment) — Luku jolla muuttujan arvo kasvaa tai vähenee esim. ohjelmasilmuksessa.

Literaali (Literal) — Vakio merkkijonoarvo.

Lohko (Segment) — Tietty muistin osa.

Loppumerkki (Stop Bit) — Signaali, joka seuraa lähettyä merkkiä tai tietuetta. Saatuaan loppumerkin vastaanottava laite on valmis seuraavan merkin tai tietueen vastaanottoon.

Lukualue (Infinity, Machine Infinity) — Suurin numero, joka voidaan esittää tietokoneen sisäisessä muodossa.

Mantissa (Mantissa) — Eksponenttimuodossa esitetyn luvun kantaluku. Esimerkiksi luvun $3.264E + 2$ mantissa on 3.264.

Massamuisti (Mass storage device) — Oheislaitte, kuten kasettinauhuri tai levykeasema, johon ohjelma ja/tai data voidaan tallettaa myöhemmin käytettäväksi. Tieto talletetaan tavallisesti muodossa joka on tarkoitettu tietokoneen, ei ihmisen, ymmärrettäväksi.

Matriisi (Matrix) — Sääntiö, jossa on kaksi tai useampi ulottuvuus.

Matriisikirjoitin (Matrix Printer) — Kirjoitin, jossa kukin merkki muodostuu pienistä pisteistä. Pistematriisi voi muodostua esim. vaakasuunnassa 7 pisteestä ja pystysuunnassa 8 pisteestä (7x8-matriisi).

Megatavu (Mega Byte, MByte) — Tietokoneen tai massamuistin määrä. 1 megatavu on 1 048 576 tavua.

Merkki (Character) — Kirjain, numero, välimerkki, ohjausmerkki tai graafinen erikoismerkki, jota tavu esittää.

Merkkijono (String) — Yhtenä kokonaisuutena käsiteltävä kirjainten, numeroiden ja symbolien joukko.

Moduuli (Module, Command Module, Video Game Module) — Valmiiksi ohjelmoitu moduuli tai laajennusmoduuli, joka voidaan liittää tietokoneeseen sen toimintojen monipuolistamiseksi. Moduulilla voi olla esim. lisämuisti tai videopeli.

Muisti (Memory) — Ks. RAM, ROM, ja massamuisti.

Muoto (Format) — Tiedon asettelu tai järjestely joko kuvaruudulla tai levykkeellä.

Muotosääntö (Syntax) — Ohjelmointikielen rakennetta ja oikeinkirjoitusta määräävät säännöt.

Muuttuja (Variable) — Luku- tai kirjainmerkkiarvolle annettu nimi. Ko. arvo voi muuttua ohjelman suorituksen aikana. Muuttujan voi ajatella olevan muistipaikka, jonka sisältöä voidaan muuttaa ohjelmakäskyillä.

Määrite (Subscript) — Numerolauseke joka yksilöi taulukon, matriisin tai sääntiön alkion.

Nuolisymboli (Prompt) — Kuvaruutumerkki joka osoittaa uuden syötetävän komennon tai ohjelmalauseen aloituskohdan. Tietokoneen tiedustelumerkki.

Näennäinen satunnaisluku (Pseudo-random number) — Luku, jonka tietokone muodostaa tietyllä laskutoimitussarjalla (algoritilla). Lukuarvo on riittävän satunnainen normaalikäytössä, vaikkakaan se ei täytä todellisen satunnaisluvun tunnusmerkkejä. Absoluuttisia satunnaislukuja luo vain sattuma.

Oheislaite (Accessory Device, Peripheral Device) — Tietokoneeseen liitettävä, sen suorituskykyä lisäävä laite. Mm. valmisohjelmamoduuli (Command Module, Memory Expansion Module, Video Game Module), tai tietoa lähettävä tai vastaanottava laite, kuten kasettinauhuri, kirjoitin, levy-yksikkö jne.

Ohjelma (Program) — Tietokoneen muistiin syötetyt ohjelmalausekkeet, jotka määräävät mitä tietokoneen tulee tehdä.

Ohjelmariivi (Program Line) — Tietokoneen muistiin syötetty rivi, joka sisältää yhden tai useamman lauseen.

Ohjelmamoduuli, valmisohjelmamoduuli (Command Module) — Valmiiksi ohjelmoitu ROM-muistimoduuli, joka voidaan liittää tietokoneeseen.

Ohjelmisto (Software) — Ohjelmat, jotka tietokone pystyy suorittamaan. Esimerkiksi moduuliohjelmat, kasettiohjelmat ja käyttäjän näppäimistöstä syöttämät ohjelmat.

Oktaali- (Octal) — Lukujärjestelmä, jossa kantaluku on 8.

Oletusarvo (Default value) — Vakioarvo jonka tietokone ottaa käyttöön jos ohjelmalauseesta puuttuu jokin tekijä tai jos ohjelmassa on valintamahdollisuus.

Operandi (Operand) — Operaattorin suorittama toimitus.

Operaattori (Operator) — Vakiolaskutoimituksissa käytettävät symbolit (laskutoimitusoperaattorit) ja vertailuoperaattorit. Laskutoimitusoperaattoreita ovat +, −, *, /, ja **. Vertailu- eli relaatiooperaattoreita ovat <, >, =, <=, >=, ja <>.

Osoite (Address) — Rekisterin paikka, muistin osa tai joko RAM- tai ROM-muistissa olevan tiedon (datan) osa.

Parametri (Parameter) — Arvo tai arvot, jotka määräävät ohjelmalauseen tai funktion tuottaman lopputuloksen.

Pehmo (Software) — Ks. ohjelmisto.

Peliohjain (Joystick) — Tietokoneeseen liitettävät langalliset ohjaimet, joilla voi ohjata pelitoimintoja tai ohjelmallisesti suorittaa eri toimenpiteitä kuvaruudulla.

Peräkkäissaanti (Sequential Access) — Saantimuoto, jossa tietueet lue-
taan samassa järjestyksessä kuin ne on kirjoitettu.

Portti (Port) — Lisälaiteväylä joko datan vastaanottoa tai lähettämistä
varten.

Puskuri (Buffer) — Muistialue, jossa tietokone säilyttää tilapäisesti siir-
rätätietoja (lähetettäviä tai vastaanottavia tietoja). Puskurin avulla voi-
daan liittää nopea tuloportti hitaaseen lähtöporttiin. Puskuria voidaan
verrata vesisäiliöön, johon vesi tilapäisesti varastoidaan.

Päällekirjoitus (Overwriting) — Massamuistilaitteen tai muistissa olevan
vanhan ohjelman tai tiedon päälle tallettaminen siten, että aiempaa tie-
toa ei enää voi lukea.

RAM-muisti (RAM memory) — Random Access Memory, suorasaanti-
muisti, luku/kirjoitusmuisti. Puolijohdemuisti, josta tieto voidaan hakea
osoitteen perusteella. Kun virta katkeaa, RAM-muistin sisältö nollautuu.

Rajamerkki (Delimiter) — Ks. erotin.

Rivinsiirto eteen (Line Feed, LF) — Kontrollimerkki, jolla kirjoitin tai kuva-
ruutunäyttö siirtyy seuraavalle riville ja jatkaa tulostusta siirrytystä koh-
dasta eteenpäin (ei palaa rivin alkuun).

Rivinsiirtomerkki (Carriage Return, CR) — Kontrollimerkki, jolla kirjoitin
tai kuvaruutunäyttö siirtyy rivin alkuun.

ROM-muisti (ROM memory) — Read Only Memory, lukumuisti. Mikropiiri
johon tallennettua tietoa käyttäjä ei voi muuttaa, se voidaan pelkästään
lukea. ROM-muistin sisältö säilyy vaikka se ei saisi sähkövirtaa.

Satunnaisluku — Ks. Näennäinen satunnaisluku.

Segmentti (Segment) — Tietty muistin osa tai levykkeen osa, segmentti.

Selata (Scan) — Tutkia järjestelmällisesti, esim. rivi riviltä.

Seurata (Trace) — Ohjelmakäskyjen suoritusjärjestyksen seuraaminen.
Seuranta on hyödyllistä esim. ohjelmointivirheitä etsittäessä.

Silmukka (Loop) — Tietyn ohjelmaosan suorittaminen toistuvasti. Tavalli-
sesti jokin ohjelmoitu laskuri määrää suorituskertojen lukumäärän.

Sisäinen esitysmuoto (Internal data-format) — Tietokoneen sisäisesti käyttämä datan muoto.

Sisäkkäinen (Nest, Nested) — Ohjelmarakenne voi sisältää sisäkkäin samanlaisia toimintoja. Hyvin usein silmukoita käytetään sisäkkäin (silmukan sisäiset silmukat) tai alirutiineihin saattaa sisältyä omia alirutiineja.

Sivu (Page) — Osa muistista, joka voidaan tulostaa omana kokonaisuutena joko kuvaruudulle tai kirjoittimella paperille.

Sivunvaihtomerkki (Form Feed, FF) — Kontrollimerkki, jolla kirjoitin tai kuvaruutunäyttö siirtyy seuraavalle sivulle.

Sovellusohjelma, sovellutusohjelma (Application Program) — Ohjelma, joka on sovellettavissa tiettyyn tarkoitukseen.

Suorittaa (Execute) — Ajaa tietokoneen muistissa oleva ohjelma tai toteuttaa käyttäjän syöttämä komento.

Syntaksi (Syntax) — Ks. muotosääntö.

Syöttää (Input) — Syöttää tietoa tietokoneen muistiin (joko näppäimistöstä tai oheislaittelta).

Syöttö/tulostus (I/O, Input/Output) — Tietojen siirtäminen tietokonejärjestelmään kuuluvien laitteiden välillä.

Syöttörivi (Input line) — Kerrallaan syötettävän datan suurin pituus.

Sääntiö (Array) — Kuten taulukot. Yleensä taulukko on kaksiulotteinen, mutta sääntiössä voi olla useampia ulottuvuuksia. Nimityksiä taulukko ja sääntiö käytetään usein samassa tarkoituksessa ulottuvuuksista riippumatta.

Tahdistamaton (Asynchronous) — Ks. asynkroninen.

Tarkkuus I (Precision) — Laskutoimitusten luotettavuuden määre, tavallisesti luvussa käytettävien desimaalien määrä.

Tarkkuus II (Resolution) — Ks. erottelukyky.

Taulukko (Table, Array) — Numero- tai jonomuuttujien järjestetty joukko, jossa muuttujat on järjestetty riveihin ja sarakkeisiin. Tietokone valitsee

tietoalkiot käsiteltäväksi niiden sijainnin perusteella. Tietoalkion sijainnin ilmoittaa ohjelmalauseeseen kuuluva määrite. Taulukot ovat kaksiulotteisia sääntiöitä.

Tausta (Background) — Kuvaruudun merkkiä ympäröivä osa.

Tavu (Byte) — Yhtenä kokonaisuutena käsiteltävä 8 bitin joukko jota kutsutaan myös merkiksi. Tietokoneen muistikapasiteetti ilmoitetaan usein tavulukumäärällä (kilotavua tai ktavua).

Tiedosto (File) — Ulkoiseen muistiin (kasettinauhalle tai levykkeelle) tallennettu, toisiinsa liittyvien tietueiden joukko. Tiedostoja käytetään ohjelmien yhteydessä.

Tieteellinen esitystapa, kymmenpotenssiesitys (Scientific Notation) — Hyvin suurten ja hyvin pienten lukujen esitystapa. Se rakentuu mantissasta (kantaluku) ja haluttuun potenssiin korotetusta luvusta kymmenen. Luvut kirjoitetaan muodossa ETUMERKKI, MANTISSA, kirjain-E, KYMMENEN EKSPONENTTI (ja sen edellä tarvittaessa miinusmerkki). Esim. 3.264E4 tai 22.47E—17. Englanninkielessä ja BASIC-kielessä luvun kokonais- ja desimaaliosan välissä on piste (pilkkua käytetään tekstissä, ei BASICissä) erottamaan tuhansia, miljoonia ym. kolmijakoisia luvun osia toisistaan).

Tietue (Record) — Samaan yhteyteen liittyvät, rajattuun tapahtumaan liittyvät luku- tai kirjainmuotoiset tiedot.

Tulkita (Interpret) — Ohjelmakielen lauseiden kääntäminen toiseen kieleen sitä mukaa kuin alkukielistä ilmaisua saadaan käsiteltäväksi (BASIC-tulkki).

Tulostaa, kirjoittaa (Display) — Tuoda merkkejä näkyviin kuvaruudulle.

Tulostus (Output) — Tiedon siirtäminen tietokoneen muistista oheislaitteelle, esim. kuvaruudulle, kirjoittimelle tai massamuistiin.

Tyhjä jono (Null String) — Jono, johon ei sisälly yhtään merkkiä, ja jonka pituus on nolla.

Tyhjä lause (Null, Null Statement) — Tyhjä lause ilman merkitystä, esimerkiksi merkkijono jossa ei ole merkkejä. Voidaan käyttää esimerkiksi ohjelmasuorituksen hidastamiseksi.

Täyttää (Pad) — Lohkon tai osan täyttäminen valedatalla, tavallisesti nolilla tai välilyönneillä.

Vakio (Constant) — Määritelty luvun tai jonon arvo. Numerovakio voi olla reaali-luku, esimerkiksi 256.8976 tai —9054. Jonovakio on lainausmerkkien sisällä oleva merkkiyhdistelmä, kuten "KAUNIS ILMA" tai "SALO-RANKATU 5 -7"

Vale- (Dummy) — Keksitty argumentti, jolla ei ole merkitystä. Valeargumentti on tarpeellinen joissakin toiminnoissa ennenkuin toiminta voidaan alkaa.

Valikko (Menu) — Valintalista, josta valitaan seuraava toiminto.

Valokynä (Light Pen) — Valoherkkä lisälaite, jota käytetään kuvaruudulla tapahtuvaan hahmojen muodostamiseen tai muihin toimenpiteisiin.

Varattu sana (Reserved Word) — Ohjelmointikielissä erikoissana, jolla on ennalta määrätty merkitys. Varatun sanan tulee olla kirjoitettu oikein, sen täytyy sijaita oikeassa kohdassa lausetta tai komentoa, sitä voi käyttää muuttujanimenä eikä se voi sisältyä muuttujanimeen.

Varmistus (Backup) — Varmuuskopio ohjelmasta tai datasta. Voidaan käyttää, jos käyttöversio tuhoutuu.

Vianmääritys (Diagnostic) — Viittaus virhetoiminnon tai virheen etsimiseen ja erottamiseen.

Vierittää (Scroll) — Kuvaruudulla olevan tiedon tai sen osan siirtely pysty- tai vaakasuunnassa siten, että uusi data ilmestyy esiin toisesta reunasta ja vanha data häviää vastakkaiseen reunaan.

Viestintä (Communication) — Ks. kommunikointi.

Virhe (Bug) — Laitteistovika tai ohjelmavirhe, joka saa tietokoneen toimimaan virheellisellä tavalla.

Virheenmääritys (Diagnostic) — Ks. vianmääritys.

Vuorosuuntainen (Half Duplex) — Tiedonsiirto, jossa kommunikointi tapahtuu vain yhteen suuntaan kerrallaan (vrt. kaksisuuntainen).

Yhdistäminen, liittäminen (Concatenation) — Kahden jonon yhdistäminen yhdeksi, pitemmäksi jonoksi. Yhdistämisoperaattorina käytetään " + "-merkkiä. Yhdistetyn jonon pituus ei saa ylittää 256 merkkiä.

Ylitys, laskentakapasiteetin ylitys (Overflow) — Ylitys tapahtuu kun käyttäjä syöttää tai tietokone laskee luvun, joka on suurempi kuin rekisterin

tilavaraus sallii. Jos näin tapahtuu, tietokone antaa tulokseksi nollan ja ohjelman suoritus jatkuu ilman virheilmoituksen tulostusta.

Ylivuoto (Overflow) — Ks. ylitys.

Ääni (Sound) — Tietokoneen aikaansaamat äänet voivat olla joko säveliä tai hälyääniä.

LIITE 1

AAKKOSELLINEN HAKEMISTO

AAKKOSELLINEN HAKEMISTO

ABS	Itseisarvo, esim. ABS(X)	77
AND	Relaatio- ja loogisten lausekkeiden arvo on 1 tai —1 (ei nolla), jos ne ovat tosia, 0, jos ne ovat epätosia, katso myös OR ja NOT, esim. A AND B	59
ASC	ASCII-arvo, esim. ASC(X\$)	80
ATN	Arkustangentti radiaaneissa, esim. ATN(X)	77
BLOAD	Binääriohjelman lataaminen nauhalta	93
BREAK	Ohjelman pysäyttäminen kesken suoritusta	53
BRUN	Binääriohjelman lataaminen nauhalta ja ohjelman käynnistäminen	94
BSAVE	Binääriohjelman tallettaminen nauhalle	94
CALL	Siirtyminen assembler-kieliseen aliohjelmaan	85
CHR\$	Antaa argumentin ASCII-koodia vastaavan merkin, esim. CHR\$(I)	80
CIRCLE	Ympyrän piirtäminen, esim. CIRCLE (X,Y),Z	37
CLEAR	Muuttujien tai syötetyn tiedon poistaminen muistista sekä datan asettaminen uudelleen käytettäväksi	66
CLOAD	Ohjelman lataaminen kasetilta	92
COLOR =	Kuvaruudun taustavärin ja edustavärin muuttaminen, esim. COLOR = 3,15	32
CONT	Ohjelma-ajon jatkaminen pysäytetystä kohdasta	52
COS	Kosini radiaaneissa, esim. COS(X)	77
CRUN	Ohjelman lataaminen kasetilta ja ohjelman ajo välittö- mästi lataamisen tapahduttua	93
CSAVE	Ohjelman tallettaminen kasetille	93
DIM	Taulukon tai sääntiön tilavaraus, esim DIM A(3)	83
END	Ohjelman loppu	44
EXP	e ^x , esim. EXP(X)	77
FOR..TO..STEP..NEXT	Silmukka, joka suorittaa sarjan askeleita, esim. FOR I = 1 TO 100 STEP 5	60
GET	Merkin luku näppäimistöltä, esim GET A\$	87
GOSUB...RETURN	Siirtyminen GOSUB-komennon rivinumeron määrää- mään alirutiiniin tai aliohjelmaan ja palaaminen takaisin RETURN-komennolla. Ohjelma jatkuu GOSUB- komennon jälkeiseltä riviltä	64

GOTO	Ohjelman jatkaminen halutulta rivinumerolta, esim. GOTO 160	63
GR	Siirtyminen grafiikkatilaan	30
HOME	Kuvaruudun tyhjentäminen	38
IF...THEN	Haarautuminen ohjelmassa, esim. IF X + Y > 10 THEN PRINT "SUUREMPI"	58
INPUT	Datan syöttö näppäimistöltä ohjelmaa ajettaessa, esim. INPUT A,B	49
INT	Kokonaisosa, esim. INT(X)	77
LEFT\$	Merkkijonon vasemmanpuoleiset merkit haluttuun merkkiin asti, esim. LEFT\$(A\$,5)	80
LET	Muuttujan arvon asettaminen, esim. LET B = 6	46
LIST	Ohjelman tulostus eli listaus kuvaruudulle	50
LLIST	Ohjelman listaus kirjoittimella	98
LPRINT	Tulostus kirjoittimella	99
LEN	Merkkijonon pituus, esim. LEN(X\$)	81
LOG	Luonnollinen logaritmi, esim. LOG(X)	78
MID\$	Merkkijonon merkit keskeltä alkaen määrätystä merkistä ja otetaan haluttu määrä merkkejä, esim. MID\$(X\$,5,5)	81
NEW	Uuden ohjelman aloittaminen ja vanhan ohjelman poistaminen muistista	45
NOT	Relaatio- ja loogisten lausekkeiden arvo on 1 tai —1 (ei nolla), jos ne ovat tosia, 0, jos ne ovat epätosia, katso myös OR ja AND, esim. NOT B	59
NOTRACE	Paluu seurantatilasta normaaliin toimintaan, katso myös TRACE	54
ON...GOTO/ON...GOSUB	Siirtyminen halutulle rivinumerolle/haluttuun alirutiiniin tai aliohjelmaan sen mukaisesti mitä ON-lausekkeen arvo määrää	65
ONERR	Ohjelman haarautuminen virheen esiintyessä, esim ONERR GOTO 1000	90
OR	Relaatio- ja loogisten lausekkeiden arvo on 1 tai —1 (ei nolla), jos ne ovat tosia, 0, jos ne ovat epätosia, katso myös NOT ja AND, esim. A OR B	59
PEEK	Muistiosoitteessa olevan arvon lukeminen, esim. PRINT PEEK(X)	84
PLOT	Viivan piirtäminen	33

PLOT...TO	Viivan piirtäminen pisteestä toiseen, esim. PLOT X,Y TO Z,V	33
POKE	Arvon tallettaminen muistiosoitteeseen, esim. POKE X,Y	85
PRINT	Tulostus kuvaruudulle	48
RAM	Jäljellä olevan muistitilan ilmoittaminen, esim PRINT RAM(0) — 16384	88
READ...DATA	Datan luku arvo arvolta DATA-lauseesta	61
RECALL	Kasetille talletetun datan lukeminen	95
RECT	Suorakulmion piirtäminen, esim. RECT (X,Y),(Z,V)	38
REM	Kommentti, jota ei huomioida ohjelmaa ajettaessa, esim. REM **ALIOHJELMA TALONRAKENNUS**	47
RESTORE	Datan käyttäminen uudelleen READ...DATA-lauseiden yhteydessä	62
RIGHT\$	Oikeanpuoleiset merkit, esim. RIGHT\$(X\$,5), katso myös LEFT\$	81
RND	Satunnaisluku nollan ja yhden väliltä, esim RND(X)	78
SGEN	Ääni- ja musiikkikäsky, jolla ohjataan äänigeneraattoria suoraan, esim. SGEN 159, 161, 223, 255	72
SGN	Lausekkeen etumerkki, esim. SGN(X)	78
SIN	Sini radiaaneissa, esim. SIN(X)	78
STOP	Ohjelman pysäytys	51
SOUND	Ääni- ja musiikkikäsky	70
SPC	Tyhjä tila joka määräytyy välilyöntien määrän mukaan, esim. SPC(4)	66
SQR	Lausekkeen neliöjuuri, esim. SQR(X)	78
STORE	Datan talletus kasettinauhalle	95
STR\$	Numeerinen argumentti merkkijonoksi, esim. STR\$(X)	81
TAB	Sarkain,	66
TAN	Lausekkeen tangentti radiaaneissa, esim. TAN(X)	78
TEXT	Paluu grafiikkatilasta tekstitilaan	31
TRACE	Siirtyminen seurantatilaan. Kuvaruudulle tulostuu rivi- numerot siinä järjestyksessä, missä tietokone suorittaa ne ohjelma-ajon aikana. Komentoa voidaan käyttää virheiden etsinnässä, esimerkiksi paikannetta- essa päättymättömiä silmuikoita.	54
UNPLOT	Viivan poisto	36
UNPLOT...TO	Viivojen poisto annettujen koordinaattien väliltä, esim. UNPLOT X,Y TO Z,V	36

VAL	Merkkijonon numeerinen arvo, esim. VAL(X\$)	81
VPEEK	Datan luku videonäytön RAM-muistista	39
VPOKE	Datan kirjoittaminen videonäytön RAM-muistiin	40
&	Siirtyminen assembler-kieliseen alirutiiniin	88

LIITE 2

HEKSADESIMAALI — DESIMAALIMUUNNOSTAULUKKO

Käytä seuraavaa taulukkoa omien kuvaruutumerkkien muodostamisessa. Etsi heksadesimaaliluvun ensimmäinen numero vasemmasta pystysarakkeesta ja toinen numero vaakariviltä. Etsitty luku on näiden leikkauspisteessä.

Esim.

$$38_{\text{H}} = 56_{\text{D}}$$

$$44_{\text{H}} = 68_{\text{D}}$$

$$28_{\text{H}} = 40_{\text{D}}$$

$$6\text{C}_{\text{H}} = 108_{\text{D}}$$

$$00_{\text{H}} = 0_{\text{D}}$$

Desimaaliluvut ovat riveittäin kasvavassa numerojärjestyksessä. Jos haluat muuttaa desimaaliluvun heksadesimaaliseksi, etsi desimaaliluku ja katso vasemman reunan sarakkeesta ja yläriviltä luvun heksadesimaaliarvo.

Esim.

$$72_{\text{D}} = 48_{\text{H}}$$

$$222_{\text{D}} = \text{DE}_{\text{H}}$$

$$0_{\text{D}} = 00_{\text{H}}$$

$$256_{\text{D}} = \text{FF}_{\text{H}}$$

**HEKSADESIMAALI —
DESIMAALIMUUNNOSTAULUKKO**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
010	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
020	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
030	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
040	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
050	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
060	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
070	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
080	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
090	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A0	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B0	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C0	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D0	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E0	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F0	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255

SALORA

Salorankatu 5—7 24100 SALO 10

Myynti: Informaatiojärjestelmät, puh. 924-6151



SALORA

Salorankatu 5-7 24100 SALO 10
Myynti: Informaatiojärjestelmät, puh. 924-6151

ZB9470