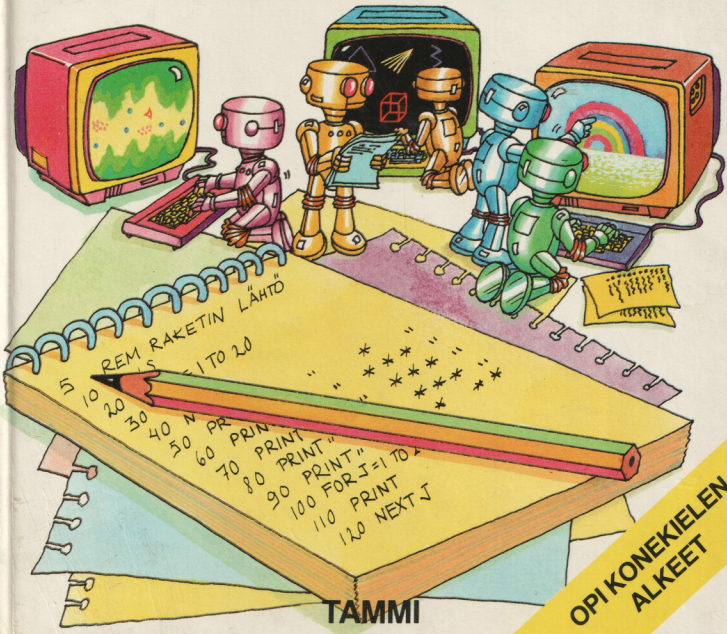
ALOITTELIJAN OHJELMOINTIKIRJA

BASIC JA KONEKIELI



TAMMI

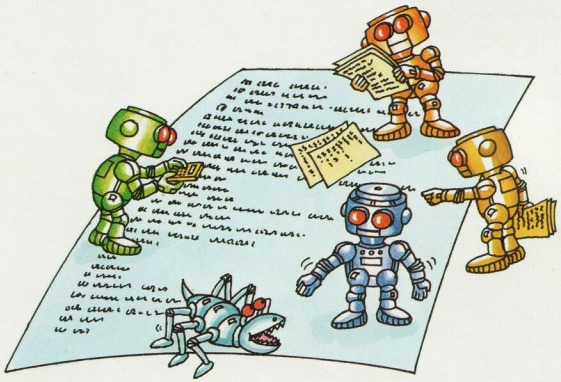
OPI KONEKIELEN ALKEET

ALOITTELIJAN OHJELMOINTIKIRJA

BASIC JA KONEKIELI

Teksti Lisa Watts ja Gaby Waters
Kuvitus Graham Round ja Naomi Reed
Suomennos Ilkka Rekiaro

KUSTANNUSOSAKEYHTIÖ TAMMI
HELSINKI



Asiantuntija-avustajat B. Reffin Smith, N. Cutler ja M. Wharton
Konekielen asiantuntijat A.P. Stephenson ja C. Oxlade
Ulkoasu G. Round

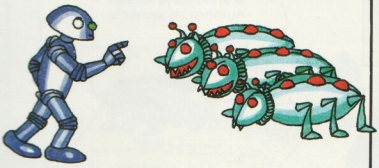
Sisällys

4 Johdanto Basiciin
5 BASIC-opas
10 Tämän kirjan BASIC
12 Helpot BASIC-ohjelmat
14 PRINT-harjoituksia
16 Muuttuja-arvoituksia
18 Asioiden toisto
20 Silmukka-arvoituksia
22 IF/THEN-harjoituksia
24 Satunnaisluvut
26 Merkkien murskausta
29 Salakieliohjelmia
30 INKEY$-harjoituksia
31 Miten tehdään autopeli
32 DATA-arvoituksia
34 Taulukoiden käyttö
36 Aliohjelmien kirjoittaminen
37 Yksikätinen rosvo
38 Tutkimme BASIC-ohjelmia
40 Neuvoja ohjelmien tutkijalle
42 Sanasilppupeli
44 Hätälasku avaruudesta
46 Maailmancupin jalkapallon tietokanta
54 Pikagrafiikkaa
58 Tiedonlajitteluohjelma
64 Graafiset esitykset
66 Keskusteluohjelma
74 Miten kirjoitetaan aarteenmetsästysohjelma
82 Konekielen alkeet
84 Mitä konekieli on?
86 Tutustumme tietokoneen sisuksiin
88 Tietokoneen muisti
91 Heksaluvut
92 PEEK ja POKE
94 Keskusyksikön sisukset
96 Käskyjä keskusyksikölle
98 Ohjelman kääntäminen heksakoodiksi
100 Tyhjän RAMin etsiminen
103 Ohjelman lataus ja ajo
107 Lisäämme tavuja muistista
108 Suurten lukujen käsittely
109 Muistilippu
110 Isoja lukuja käyttävät ohjelmat
112 Viestin esittäminen ruudulla
115 Hyppiminen ja haarautuminen
118 Vilkkuruutuohjelma
120 Tästä eteenpäin
121 Desimaali/heksa muunnostaulukot
122 Z80:n muistikkaat ja heksakoodit
125 6502:n muistikkaat ja heksakoodit
126 Konekielen sanastoa
128 Vastaukset BASIC-ohjelmiin
138 BASIC-sanoja
140 BASIC-muunnostaulukko
143 Hakemisto

Aluksi

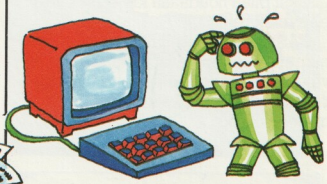
Tämä kirja opastaa BASIC- ja konekielellä ohjelmoinnin käytäntöön. Kirjan ensimmäinen osa on BASIC-opas. Sitä seuraa arvoituksia, harjoituksia ja ehdotuksia siitä, miten voit itse kirjoittaa lyhyitä ohjelmia oppiaksesi BASICin tärkeimmät käskyt. Näiden jälkeen kirjassa on pitempiä ohjelmia, joista näet, miten BASICilla voi tehdä hyvin mutkikkaita asioita, esimerkiksi luoda tietokannan, piirtää kuvia ja lajitella tietoa.

Kirjan kolmas osa on helppotajuinen johdatus konekieleen. Siinä selitetään, mitä konekieli on, miten kirjoitetaan helppoja konekielisiä ohjelmia ja miten ne annetaan tietokoneelle. Konekieli on hyvin mutkikasta ja siinä pitää noudattaa lukuisia sääntöjä. Joissain tehtävissä, esimerkiksi hienojen kuvaruutunäyttöjen tekemisessä tai pelien nopeuttajana konekieli on paljon parempi kieli kuin BASIC.



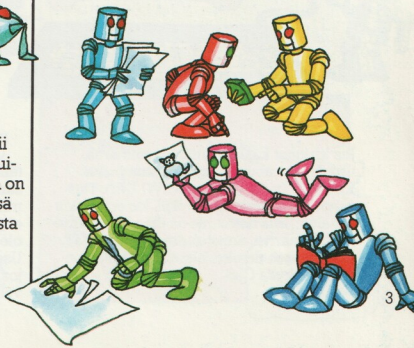
Tämän kirjan BASIC-ohjelmat on kirjoitettu yleis-BASICilla, joka toimii useimmissa kotitietokoneissa. On kuitenkin muutama BASIC-käsky, jotka on muutettava tiettyihin mikroihin. Tässä työssä on apua kirjan lopussa olevasta muunnostaulukosta.

Konekieliset ohjelmat on kirjoitettu tietokoneille, joissa on mikroprosessorina joko Z80 tai 6502. Mikroprosessori on lastu, jossa on tietokoneen keskusyksikkö. Eri mikroprosessorit käyttävät erilaista konekieltä. Mutta toisaalta kaikki tietokoneet, joissa on sama



mikroprosessori, käyttävät samaa konekieltä. Spectrum ja ZX81 käyttävät Z80-prosessoria ja Vic-20, BBC, Atarit ja Oric käyttävät 6502:ta. Commodore 64 käyttää 6510-prosessoria, joka kuitenkin ymmärtää 6502-konekieltä.

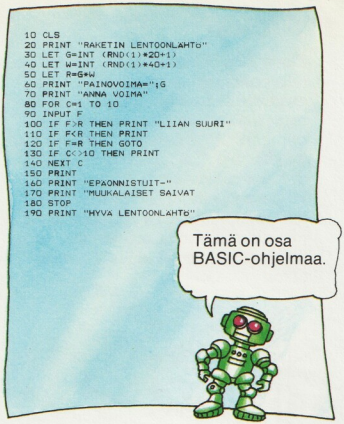
Vastaukset arvoituksiin ja ohjelmatehtäviin on annettu kirjan takana. Siellä on myös lyhyt selostus tehtävien toiminnasta. Älä murehdi, vaikket osaisikaan ratkaista kaikkia arvoituksia. Opit aivan yhtä paljon tutkimalla oikeita vastauksia. Jos voit, kokeile ohjelmia jollain tietokoneella ja tutki sitten ohjelman toimintaa.



Johdanto BASICiin

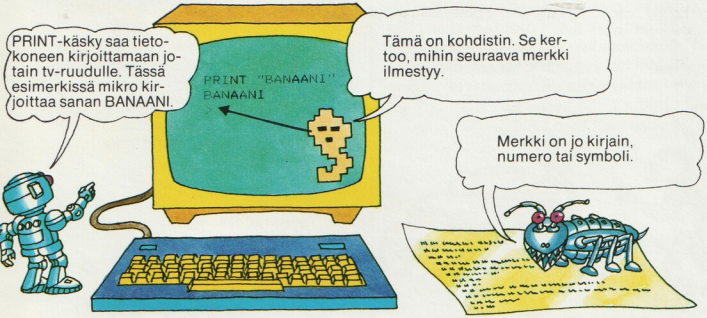
Ohjelmointikieli BASIC muodostuu noin sadasta sanasta. Jokainen sana on käsky, joka panee tietokoneen tekemään jotain. Tietokone saadaan suorittamaan haluttu tehtävä siten, että sille annetaan sarja käskyjä ja tietoa, joita yhdessä sanotaan ohjelmaksi. Käskyinä voi käyttää vain BASIC-kielen sanoja. Lisäksi pitää noudattaa kielen sääntöjä eli syntaksia.

BASICissa jokaisella käskyrivillä on numero. Numerot ovat yleensä 10:n välein, jotta rivien väliin voi lisätä uusia rivejä muuttamatta kaikkien rivien numeroita uusiksi.



Ohjelman kirjoittaminen

Kun kirjoitat ohjelmaa koneeseen, jokaisen rivin lopussa pitää painaa RETURN (tai ENTER tai NEWLINE tietokoneesta riippuen). Tämä saa tietokoneen tallentamaan rivin muistiinsa ja odottamaan seuraavaa riviä. Kun olet kirjoittanut kaikki ohjelmarivit, kirjoita RUN. Tämä panee koneen toteuttamaan antamasi käskyt.



Jos kirjoitat käskyn ilman rivinumeroa, tietokone toteuttaa sen heti, kun painat RETURN (tai ENTER tai NEWLINE). Tätä sanotaan suoraksi käskyksi. Jos esimerkiksi haluat, että tietokone näyttää kirjoittamasi ohjelmarivit, annat suoran käskyn LIST. Useimmissa tietokoneissa ruutu tyhjennetään käskyllä CLS.

Ohjelma on kirjoitettava koneeseen erittäin huolellisesti. Jos kirjoitat jonkin BASIC-sanan väärin tai jos kirjoitat vääriä kirjaimia, numeroita tai välimerkkejä, tietokone ei pysty toteuttamaan käskyjä. Ohjelmassa olevaa virhettä sanotaan joskus pukiksi. Useimmat pukit ovat kirjoitusvirheitä, mutta joskus ohjelman logiikassa on virhe, jolla voi olla yllättäviä seurauksia.

BASIC-opas

Seuraavilla sivuilla esitellään tärkeimmät BASIC-käskyt ja niiden käyttö. Jos sinulla on tietokone, etsi käskyt laitteesi käsikirjasta; mikrojen välillä on joskus hieman eroa sanoissa ja säännöissä.

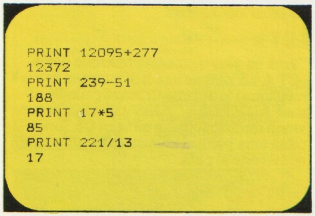
PRINT

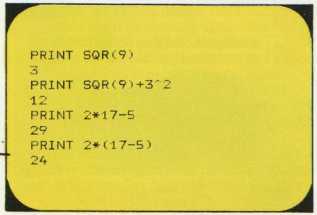
Tämä käskee tietokonetta esittämään jotain ruudulla. Kirjainten ja symbolien täytyy olla lainausmerkeissä, mutta numeroiden ei, kuten viereisestä esimerkistä näkyy. Esimerkissä ei ole rivinumeroita, joten tietokone toteuttaa käskyt heti, kun painat RETURN. Kone kirjoittaa aivan samat merkit, jotka panit lainausmerkkien sisään. Pelkkä sana PRINT käskee mikron kirjoittaa yhden tyhjän rivin.

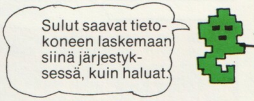
Laskutehtävät

PRINT näyttää myös laskutehtävien tuloksen. Mikro käyttää tuttuja yhteen- ja vähennyslaskun merkkejä, mutta kertolaskun se merkitsee * ja jakolaskun / . SQR(N) käskee laskea neliöjuuren luvulle (N) ja ↑ tai ^ tai * * tarkoittaa potenssiin korotusta. Esimerkiksi 3 ↑ 2 tarkoittaa 3 potenssiin 2 eli kolmen neliö.

Tietokone laskee ensin kerto- ja jakolaskut ja sitten yhteen- ja vähennyslaskut. Järjestyksen voi muuttaa suluilla, jotka kertovat mikrolle, missä järjestyksessä sen pitää toimia. Jos sulkuja on useita, mikro aloittaa laskemisen sisimmistä suluista.



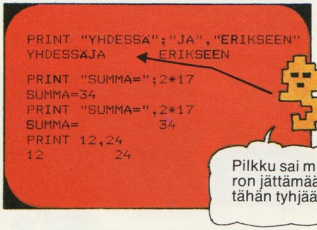




Pilkku ja puolipiste

Ne kertovat tietokoneelle, mihin seuraava merkki kirjoitetaan. Puolipiste tarkoittaa, että tyhjää tilaa ei jää ollenkaan, pilkku taas käskee mikroa siirtymään tietyn matkan eteenpäin (matka vaihtelee mikrosta toiseen).

Joissain mikroissa tarvitaan pilkku tai puolipiste erottamaan PRINT-lause ja tiedot tai muuttujat toisistaan (muuttujat ovat kirjaimia, jotka edustavat muistissa olevia tietoja eli datoja). Kokeile, miten oikealla olevat esimerkit toimivat mikrossasi.



Muuttujat ▶

Tietokoneelle annetaan käsiteltäväksi tietoa eli dataa. Kun tietokoneelle annetaan tietoa muistiin tallentamista varten, tiedolle on myös annettava nimi. Tätä nimeä sanotaan muuttujaksi. Kun haluat mikron tekevän tiedoille jotain, viittaat tietoihin muuttujanimellä. Termi tulee siitä, että muuttujan alaiset tiedot voivat muuttua ohjelman aikana.

Jos tiedot ovat numeroita, muuttujan nimenä käytetään kirjaimia tai kirjaimia ja numeroita, esimerkiksi A6. Kirjaimista ja symboleista muodostuvia tietoja sanotaan (merkki)jonoiksi. Jonoja merkitään kirjaimella tai kirjaimella ja numerolla, joiden perään pannaan dollarimerkki, esimerkiksi P$ (lausutaan P dollari) tai P6$. Muuttujien nimiä koskevat säännöt vaihtelevat mikrojen välillä, joten tarkista asia oman mikrosi käsikirjasta.

LET ▶

Tämä on yksi tapa antaa mikrolle tietoa. LET A=5 käskee sitä tallentamaan luvun 5 muistiinsa ja antamaan muistipaikalle nimen A. LET C$="JÄNIS" tallentaa kirjainjonon muistipaikkaan nimeltä C$. Jonot pitää aina panna lainausmerkkeihin, mutta numeroita ei pidä.

INPUT ▶

Tällä voidaan antaa mikrolle tietoa kesken ohjelman ajon. Sanan INPUT perään tulee muuttujan nimi. Kun mikro tulee ohjelmassa INPUT-käskyn kohdalle, se kirjoittaa ruudulle kysymysmerkin (tai jonkin muun symbolin) ja odottaa, että kirjoitat tiedot koneeseen. Jos INPUT-muuttuja on numeromuuttuja, mikrolle pitää syöttää numerotietoja ja jos se on jonomuuttuja, mikrolle pitää syöttää jono.

Monilla tietokoneilla voi panna INPUT-käskyn jälkeen lainausmerkkeihin sanoja, jotka tekevät ohjelman selvemmäksi (ks. esimerkkiä). Älä kuitenkaan käytä tätä menetelmää VIC-tietokoneessa, sillä VIC tallentaa muuttujaan sekä antamasi tiedon että selventävät sanat. Yleensä mikrossa tarvitaan puolipiste selventävien sanojen ja muuttujan nimen välillä.

```
10 LET A=5
20 LET C$="JANIS"
30 PRINT A
40 PRINT C$
RUN
5
JANIS
```

Kirjoita PRINT ja muuttujan nimi, kun haluat, että tietokone näyttää tiedot ruudulla.

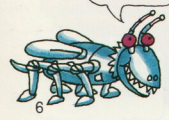
```
10 PRINT "MIKA SINUN NIMESI ON?"
20 INPUT N$
RUN
MIKA SINUN NIMESI ON?
?JUSSI
```

Tietokoneen kysymysmerkki.

Käyttäjän vastaus tallennetaan lokeroon N$.

```
10 INPUT "MIKA SINUN NIMESI ON";N$
20 INPUT "KUINKA VANHA OLET";A
RUN
MIKA SINUN NIMESI ON? JUSSI
KUINKA VANHA OLET? 21
```

Tietokoneen kysymysmerkit.



Yleensä mikrossa on erikoisnäppäimiä, joilla pyyhitään pois vahingossa kirjoitetut merkit. Jos haluat korjata ohjelmarivin, voit kirjoittaa sen kokonaan uudestaan rivinumero mukaan lukien. Uusi rivi korvaa vanhan, jolla virhe esiintyy. Jos haluat poistaa yhden rivin, kirjoita pelkkä rivinumero. Virheiden korjaamisesta kerrotaan lisää sivulla 14.

READ/DATA

Tämäkin on yksi tapa antaa mikrolle tietoa. Sanan READ jälkeen tulee yksi tai useampi muuttujan nimi ja muuttujien tiedot ovat rivillä, joka alkaa sanalla DATA. DATA-rivi voi olla missä tahansa ohjelman kohdassa. Kun mikro kohtaa käskyn READ se etsii sanan DATA ja panee kunkin tiedon järjestyksessä muuttujiin. Tiedot pitää erottaa toisistaan pilkulla. Joissain mikroissa jonomuotoiset tiedot on pantava lainausmerkkeihin. Joissain lainausmerkkejä tarvitaan vain, jos jonoissa on välilyöntejä tai välimerkkejä.

IF/THEN▶

Tällä voidaan panna tietokone vertailemaan tietoja ja tekemään jotain riippuen siitä, mitä se saa vertailun tulokseksi. Käskyllä voidaan vertailla, ovatko kaksi tietoa samoja vai eivät samoja, vai onko toinen tiedoista suurempi tai pienempi kuin toinen. Tällöin käytetään oikealla esitettyjä symboleita. Sanan THEN jälkeen voi tulla melkein mikä tahansa käsky, mutta jos vertailun tulos ei ole tosi, tietokone ei noudata THEN-käskyä vaan siirtyy ohjelmassa eteenpäin.

GOTO▶

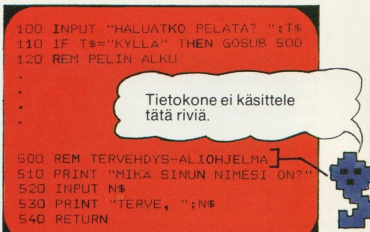
Tämä käskee tietokoneen mennä jollekin toiselle ohjelmariville. Yleensä sitä käytetään yhdessä IF/THEN-käskyn kanssa, joten tietokone haarautuu vain, jos tietyt ehdot käyvät toteen. Ole varovainen, jos käytät GOTO-käskyä yksinään, sillä se voi saada aikaan jatkuvan silmukan kuten esimerkkimme rivillä 185. Ainoa tapa pysäyttää esimerkin ohjelma on kirjoittaa BREAK tai ESCAPE (käsky vaihtelee mikrosta toiseen).

GOSUB/RETURN▶

GOSUB saa tietokoneen menemään aliohjelmaan, joka on jotain tiettyä tehtävää hoitava ohjelman osa. Sana RETURN aliohjelman lopussa palauttaa tietokoneen GOSUB-käskyä seuraavaan käskyyn. Ohjelmaan tulee pukki, jos unohdat sanan RETURN.

REM▶

Tämä on lyhennys englannin sanasta remark, huomautus. Tietokone ei käsittele rivejä, jotka alkavat sanalla REM. Näille riveille voi panna selvityksiä, jotka helpottavat ohjelman tulkintaa.



Rivi 540 lähettää tietokoneen takaisin GOSUBia seuraavaan käskyyn.

FOR/NEXT-SILMUKAT

Sanat FOR, TO ja NEXT panevat tietokoneen toistamaan jonkin osan ohjelmasta niin monta kertaa kuin haluat. Oikealla olevassa esimerkissä rivit 10–30 toistuvat kolme kertaa ja joka kerralla tietokone kirjoittaa viestin riville 20. J on muuttuja, jolla lasketaan toistokerrat. Rivi 30 lähettää tietokoneen takaisin etsimään J:n seuraavaa arvoa. Joka kerta, kun silmukka toistuu, J:hin lisätään 1. Kun J = 3 tietokone jatkaa ohjelmassa eteenpäin.

```
10 FOR J=1 TO 3
20 PRINT "J SILMUKKA ";J
30 NEXT J
40 PRINT
RUN
J SILMUKKA 1
J SILMUKKA 2
J SILMUKKA 3
```

Muuttuja J rivin lopussa saa mikron kirjoittamaan J:n arvon joka kerta, kun silmukka toistuu.

STEP▶

Tämä vaikuttaa siihen, miten J laskee toistokerrat. Esimerkiksi FOR J = 1 TO 10 STEP 2 kasvattaa J:tä 2:lla joka kierroksella ja STEP X saisi J:n kasvamaan sillä määrällä, mikä on tallennettu muuttujaan X. Esimerkissämme STEP-1 saa J:n laskemaan takaperin.

```
10 FOR J=10 TO 1 STEP-1
20 PRINT J;" PAIVAA JOULUUN"
30 NEXT J
40 PRINT "HYVAA JOULUA"
RUN
10 PAIVAA JOULUUN
9 PAIVAA JOULUUN
8 PAIVAA JOULUUN
7 PAIVAA JOULUUN
```

Sisemmän silmukan kummankin osan pitää olla ulomman silmukan sisällä, muuten ohjelmaan tulee pukki.

Sisäkkäiset silmukat▶

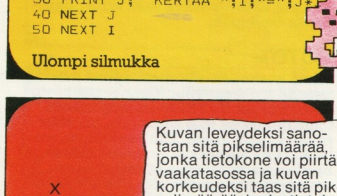
Saat aikaan melko mutkikkaita toistoja jos panet silmukoita toisten silmukoiden sisään. Näitä sanotaan sisäkkäisiksi silmukoiksi. Esimerkkiohjelmassamme silmukka riveillä 20–40 ajetaan 12 kertaa aina, kun rivien 10–50 silmukka toistuu. Aina kun sisempi silmukka toistuu, rivi 30 tulostaa arvon J × I.

```
10 FOR I=2 TO 12
20 FOR J=1 TO 12
30 PRINT J;" KERTAA ";I;"=";J*I
40 NEXT J
50 NEXT I
```

Ulompi silmukka

Grafiikkakäskyt▶

Tietokone piirtää kuvia sytyttämällä ruudulla pieniä neliöitä eli pikseleitä. Tietokoneiden välillä on eroa siinä, millä käskyillä neliöitä sytytetään. Tämän kirjan ohjelmissa käytetään käskyä PLOT X, Y jossa X ja Y ovat pikselin koordinaatit. Viivan piirtämiseksi kirjan ohjelmissa käytetään käskyä DRAW X, Y. Useimmissa tietokoneissa on samankaltaiset käskyt, mutta joissain tarvitaan lisäkäsky, jolla valitaan tietty grafiikkatila.*



Kuvan leveydeksi sanotaan sitä pikselimäärää, jonka tietokone voi piirtää vaakatasossa ja kuvan korkeudeksi taas sitä pikselimäärää, jonka tietokone voi piirtää pystysuunnassa.

RND▶

Tämä saa tietokoneen tuottamaan satunnaisluvun, mutta komennon muoto vaihtelee mikrojen välillä. Joissain RND(9) tuottaa luvun väliltä 1–9. Joissain tarvitaan mutkikkaampi komento, esimerkiksi INT(RND(1)★9+1). Tietokone aloittaa sulkujen sisältä. RND(1) saa sen tuottamaan luvun väliltä 0 ja 1. Se kertoo tämän luvun 9:llä, joka on korkein haluamasi luku, lisää 1, koska sana INT pyöristää luvun alaspäin, ja tekee luvusta kokonaisluvun.

```
PRINT RND(9)
7
PRINT INT(RND(1)*9+1)
7
```

0.0109425★9+1=

Joissain mikroissa kirjoitetaan RND(0) eikä RND(1)

Taulukot ▼

Taulukko on joukko tietoja, jotka on yhdistetty saman muuttujanimen alle. Muuttuja on ikään kuin tietokoneen muistissa oleva tila, joka on jaettu useisiin lokeroihin. Taulukko voi olla yksiulotteinen, jolloin siinä on yksi lokerorivi, tai kaksiulotteinen, jolloin siinä on useita lokerorivejä. Yksiulotteisessa taulukossa haluttuun tietoon viitataan lokeron numerolla. Alla olevassa kuvassa A$(4) on LUUMU. Kaksiulotteisissa taulukoissa on annettava rivin ja sarakkeen numero, esimerkiksi D(3,2) on 15. Suluissa olevan luvun nimi on alaindeksi. (Sinclair/Timex-tietokoneilla säännöt ovat hieman erilaiset. Ks. sivuja 34 ja 140.)

Ennen kuin voit käyttää taulukoita, tietokoneelle on ilmoitettava, kuinka iso taulukko on. Tämä tapahtuu sanalla DIM (lyhennys sanasta dimensio, ulottuvuus), kuten oikealla on tehty. Tieto pannaan taulukkoon käskyn READ/DATA ja silmukan avulla. Kaksiulotteisessa taulukossa tarvitaan sisäkkäiset silmukat, kuten oikealla olevassa esimerkissä näkyy.

Tässä esimerkissä I on rivinumero ja J sarakenumero. Aina, kun sisempi silmukka J toistuu, yksi tietosilmukka siirtyy rivin seuraavaan sarakkeeseen. Kun silmukka I toistuu, tietokone aloittaa uuden rivin.

```
20 DIM D(4,3)
30 FOR I=1 TO 4
40 FOR J=1 TO 3
50 READ D(I,J)
60 NEXT J
70 NEXT I
80 DATA 5, 12, 16
90 DATA 3, 2, 7
100 DATA 8, 15, 11
110 DATA 4, 1, 7
```

DIM-käskyn tulee olla ohjelman alussa, koska sitä tarvitaan vain kerran.

LEFT$ ja RIGHT$ ▶

Näillä voi tehdä temppuja jonomuuttujissa olevilla merkeillä. Esimerkiksi LEFT$(A$,4) käskee tietokoneen ottaa neljä merkkiä A$:n vasemmasta päästä ja RIGHT$(A$,5) käskee sen ottaa viisi merkkiä jonon oikeasta päästä. (Sinclair/Timex mikrot eivät käytä näitä käskyjä. Ks. sivuja 26 ja 140.)

```
10 LET A$="PAPUKAIJA"
20 PRINT LEFT$(A$,4)
25 PRINT LEFT$(A$,2)
30 PRINT
40 PRINT RIGHT$(A$,5)
RUN
PAPU
PA

KAIJA
```

MID$ ja LEN ▶

MID$ käskee tietokonetta ottamaan merkkejä jonon keskeltä ja LEN kertoo, montako merkkiä jonossa on välimerkit ja välilyönnit mukaan lukien. Esimerkiksi MID$(K$,2,4) tarkoittaa: ota neljä merkkiä K$:n keskeltä toisesta merkistä alkaen. Sivuilla 26 ja 140 on annettu käskyt, joita pitää käyttää Sinclair/Timex mikroissa.

```
10 LET K$="JOULUKUUSI"
20 PRINT MID$(K$,4,4)
25 PRINT MID$(K$,2,4)
30 PRINT
35 PRINT LEN(K$)
RUN
LUKU
OULU

10
```

K$:ssa on näin monta merkkiä.

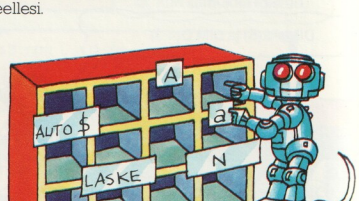
Tämän kirjan BASIC

Tämän kirjan ohjelmat on kirjoitettu "yleis'-BASICilla. Joillain tietokoneilla on kuitenkin omat erikoiset tapansa, joten voit joutua tekemään kirjan ohjelmiin pieniä muutoksia voidaksesi ajaa niitä omalla mikrollasi. Tällä aukeamalla on esitetty eräitä seikkoja, joihin on syytä kiinnittää huomiota.

Ohjelmat on kirjoitettu niin, että niitä voi käyttää monissa erilaisissa tietokoneissa, joten niissä ei oteta huomioon minkään mikron erityispiirteitä. Mutta heti, kun ymmärrät, miten ohjelmat toimivat, voit sovittaa ne omalle koneellesi.

Muuttujien nimet ▶

Joissain mikroissa voi käyttää muuttujan nimenä sanoja, joissain taas pelkästään kirjaimia tai kirjaimia ja numeroita. Esimerkiksi Sinclair tietokoneissa voi käyttää lyhyitä sanoja numeromuuttujien nimenä, mutta jonomuuttujien nimenä saa käyttää vain yhtä kirjainta. Tämän kirjan ohjelmissa muuttujien tunnuksina on etupäässä sanoja, jotta ne olisi helpompi ymmärtää. Jos tietokoneesi ei hyväksy sanoja, käytä sanan ensimmäistä kirjainta muuttujan nimenä.



LET ▶

Yleensä mikroissa ei tarvitse käyttää sanaa LET esimerkiksi lauseessa LET HEDELMÄ$ = "OMENA". Joissain mikroissa ei myöskään tarvita sanaa THEN lauseessa IF...THEN. Kaikissa tämän kirjan ohjelmissa käytetään sanoja LET ja THEN, mutta voit jättää ne pois, jos tietokoneesi ei tarvitse niitä.

```
100 HEDELMA$="OMENA"
110 LASKE=LASKE+1
120 IF LASKE=10 PRINT "VALMIS"
```

Muuttujien alustus ▶

Joissain mikroissa muuttujalle on annettava alkuarvo eli se on alustettava ennen kuin sitä voi käyttää. Tämä tarkoittaa, että muuttujalle annetaan ohjelman alussa arvo, kuten oikealla on esitetty. Jotkin mikrot taas olettavat, että numeromuuttuja on 0 ja jonomuuttuja on tyhjä ilman, että niitä tarvitsee alustaa. Tämän kirjan ohjelmissa on annettu arvot muuttujien alustusta varten, mutta ne voi jättää pois, jos tietokoneessi ei tarvitse niitä.

```
10 LET A=0
20 LET FRAASI$=""
 .
 .
 .
100 LET FRAASI$=FRAASI$+"K"
110 LET A=A+1
```

Tyhjää jonoa sanotaan myös nollajonoksi.

INPUT ▶

Yleensä tietokone hyväksyy INPUT-lauseessa lainausmerkkeihin pantuja selittäviä sanoja. * Tietokoneiden välillä on kuitenkin eroa siinä, tarvitaanko ennen INPUT-muuttujaa puolipiste ja jättääkö tietokone automaattisesti välilyönnin selventävien sanojen ja syöttämäsi tiedon väliin. Saat tietokoneesi tarpeet selville kokeilemalla tai katsomalla käsikirjasta.

```
10 INPUT"MIKA SINUN NIMESI ON ";N$
20 PRINT "HEI ";N$
```

Joissain tietokoneissa tarvitaan välilyönti lainausmerkkien sisään, jotta sanat ja tiedot eivät mene sekaisin.

* Älä käytä tätä menettelyä VIC-tietokoneessa, sillä se tallentaa muuttujaan sekä selventävät sanat että syöttämäsi tiedot.

DATA

```
100 DATA HIIRI, SOPULI, ROTTA
110 DATA GNU, "KOLMIVARPAINEN LAISKIAINEN"
120 DATA "KAURIS, PUNAINEN", "VIRTAHEPO, MUSTA", KIRAHVI
```

Kirjoita DATA-rivit erityisen huolellisesti. Jokainen tietoalkio pitää erottaa toisistaan pilkulla. Virheitä syntyy helposti. Joissain mikroissa tiedot pitää antaa lainausmerkeissä. Joissain lainausmerkkejä tarvitaan vain, jos tiedoissa on välilyöntejä tai -merkkejä.

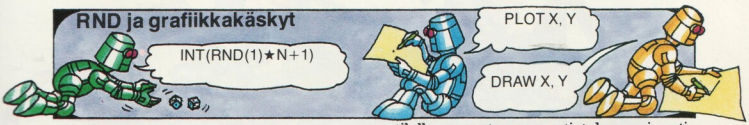
Useampi lause samalla rivillä

```
500 PLOT 40,1: DRAW 1,1
180 IF A=10 THEN PRINT "OIKEIN": GOTO 100
```

Yleensä mikrolle voi antaa useita käskyjä samalla rivillä. Käskyjen väliin pannaan kaksoispiste. Näin säästyy muistitilaa ja ohjelmaa on helpompi lukea. Jos tietokoneesi ei hyväksy useita lauseita samalla rivillä, kirjoita kukin niistä omalle rivilleen.

Jos käytät useita käskyjä samalla rivillä omissa ohjelmissasi, älä kirjoita ylimääräisiä lauseita käskyn IF...THEN perään, koska ne toteutuvat vain, jos IF-ehto on tosi.

RND ja grafiikkakäskyt



Nämä käskyt vaihtelevat tietokoneesta toiseen. Tämän kirjan ohjelmissa tuotetaan satunnaisluku väliltä 1 ja N (jossa N on mikä tahansa luku) käskyllä INT(RND(1)★N+1). Grafiikkakäskyt ovat PLOT X,Y pistettä ja DRAW X,Y viivaa varten. Kaikkien näiden tilalle on pantava oman tietokoneesi vaatimat käskyt. Jos mikrosi tarvitsee lisäksi yleisen grafiikkakäskyn, muista panna sekin ohjelmaan.

Sinclair/Timex tietokoneet

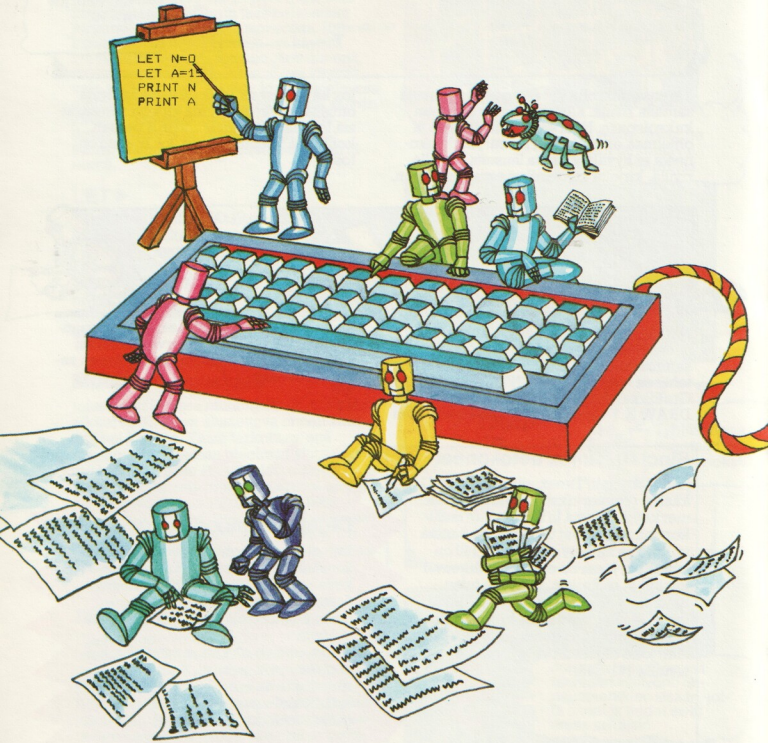
Jotkut Sinclair/Timex mikrojen BASIC-käskyt (esimerkiksi ne, joilla valitaan merkkejä jonosta ja ilmoitetaan taulukon mitat) eroavat useimmissa muissa mikroissa käytetyistä. Sivulla 140 on taulukko, jossa on esitetty poikkeavat käskyt ja selitetään, miten ne tulee muuttaa Sinclair mikroille. Kun näitä käskyjä käytetään tässä kirjassa, Sinclair mikroihin tarvittavat käskyt annetaan joko ohjelman yhteydessä tai sivuilla 141–142.

Ohjelmien muuttaminen

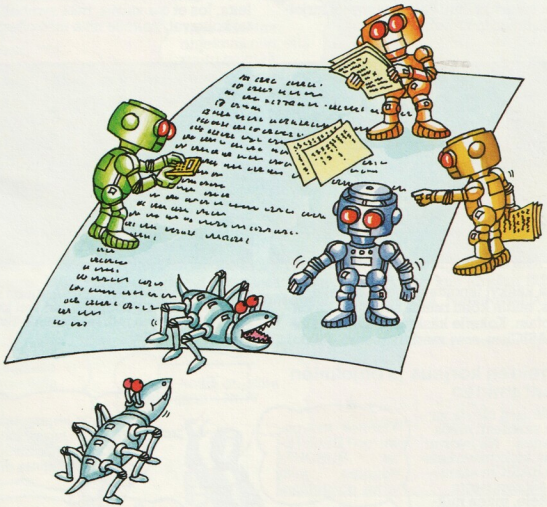
Kun olet saanut ohjelman toimimaan tietokoneessa ja kun olet saanut jonkinlaisen käsityksen ohjelman toiminnasta, voit muuttaa ohjelmaa syöttämällä uusia tietoja tai lisäämällä värejä ja ääniä.

Kun muutat ohjelmaa, tarkista jokainen rivi hyvin tarkasti. Katso, että silmukoita on oikea määrä, jotta uudet tiedot tulevat luetuiksi koneeseen ja muista muuttaa DIM-lauseet.

Helpot BASIC-ohjelmat

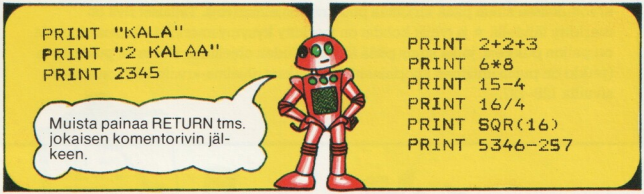


Kirjan seuraavassa osassa on runsaasti arvoituksia, harjoituksia ja ohjelmia, joiden avulla opit BASIC-kieltä. Arvoitukset ja harjoitukset on ryhmitelty sen mukaan, mitä käskyjä niissä käytetään. Kunkin jakson alussa esitellään tärkeimmät siinä käytetyt BASIC-sanat. Jos unohdat, miten jotain BASIC-sanaa käytetään, tarkista asia kirjan alun BASIC-oppaasta. Sivuilla 138–139 on BASIC-sanasto, josta voi olla apua. Joissain arvoituksissa sinun pitää kirjoittaa puuttuvia ohjelmarivejä. Tällaiset rivit on merkitty tähdellä ★ ja tyhjät kohdat on merkitty kysymysmerkillä. Osassa ohjelmia on paljon pukkeja, jotka sinun pitää löytää. Näiden ohjelmien alussa on pukin kuva (pukki on punasilmäinen kuoriainen). Vastaukset ohjelma-arvoituksiin on annettu sivuilla 128–137.



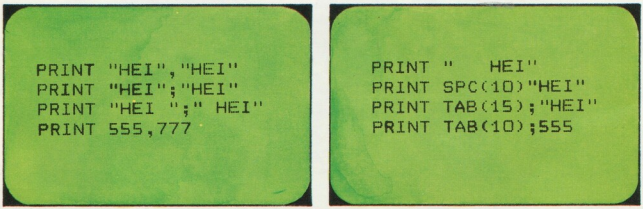
PRINT-harjoituksia

Tällä aukeamalla harjoittelet PRINT-käskyn käyttöä. PRINT saa tietokoneen näyttämään jotain tv-ruudulla. Käskyä voi käyttää suorana käskynä eli ilman rivinumeroa, jolloin tietokone toteuttaa käskyn välittömästi. Suoran käskyn jälkeen pitää painaa RETURN (tai ENTER tai NEWLINE tietokoneesta riippuen).



Kirjoita nämä suorat käskyt mikroosi. Kun käsket mikron printata kirjaimia tai kirjaimia ja numeroita yhdessä, ne on pantava lainausmerkkeihin. Pelkät numerot kirjoitetaan ilman lainausmerkkejä.

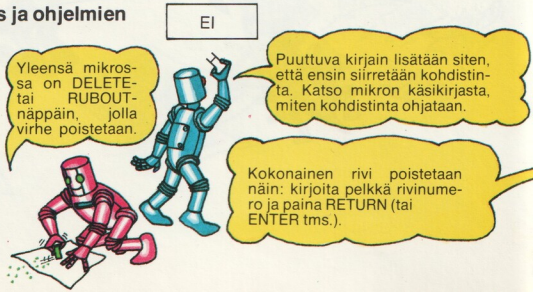
Voit myös käyttää käskyä PRINT laskutoimituksiin. Tässä on helppoja laskuja, joissa käytetään BASICin matemaattisia symboleita. Jos et ole varma, mitä symbolit tarkoittavat, kokeile niitä mikrollasi.



Nämä käskyt ilmoittavat tietokoneelle myös, mihin kohti ruutua sen pitää kirjoittaa jotain. Kokeile käskyjä. Välimerkeillä on BASICissa oma merkityksensä. Pilkku saa mikron jättämään tyhjää ennen seuraavaa merkkiä ja puolipiste saa sen kirjoittamaan seuraavan merkin samalle riville ilman välilyöntiä.

Virheiden korjaus ja ohjelmien muuttaminen

Ohjelmissa olevia virheitä sanotaan myös pukeiksi. Ne voivat johtua kirjoitusvirheistä tai BASICin sääntöjen rikkomisesta. Opettele, miten mikrossasi korjataan virheitä ja muutetaan ohjelmia.



Helppoja ohjelma-arvoituksia

Tässä on yksinkertainen ohjelma. Se saa tietokoneen esittämään ruudulla oikealla olevan viestin. Aja ohjelma ja kokeile, osaatko muuttaa sitä niin, että ruutu on samanlainen kuin alla olevissa kuvissa. Viereisen sivun suorista käskyistä pitäisi olla apua. Aukeaman alareunassa on vihjeitä siitä, miten ohjelmia muutetaan ja virheitä korjataan.

```
10 PRINT "MIKÄ ON NIMESI?"
20 INPUT A$
30 PRINT
50 PRINT A$
```

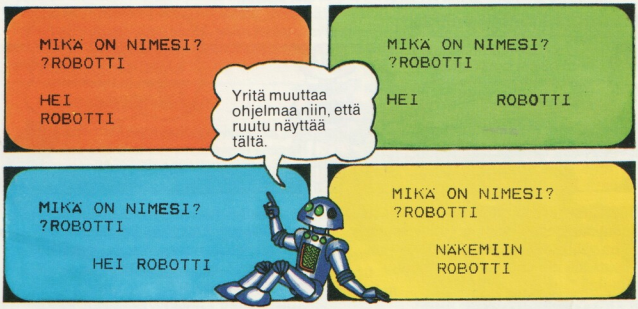
Saa mikron odottamaan, että kirjoitat nimesi ja tallentaa sen muuttujaan A$.

Pelkkä PRINT aiheuttaa tyhjän rivin.

Saa mikron printtaamaan muuttujaan A$ tallennetun sanan.

```
MIKÄ ON NIMESI?
?ROBOTTI

HEI ROBOTTI
```

Kirjoita ohjelma mikroosi ja paina RETURN tms. joka rivin lopussa. Kirjoita ohjelman loppuun RUN ja paina RETURN.



Ohjelman ja viereisen sivun suorien käskyjen välillä on se ero, että ohjelmassa joka rivillä on numero. Mikro tallentaa käskyt muistiinsa ja toteuttaa ne vasta, kun kirjoitat RUN. Jos numeroit rivit kymmenen välein voit myöhemmin lisätä käskyjä numeroimatta koko ohjelmaa uudestaan.

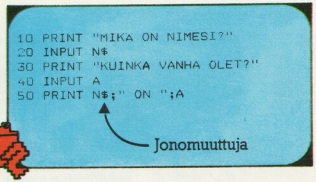
```
10 PRINT "TÄMÄ ON"
20 PRINT "OHJELMA"
15 PRINT "HASSU"
```

Uusia rivejä voi lisätä mihin tahansa ohjelman kohtaan ja vanhoja voi korjata kirjoittamalla ne uudestaan. Kirjoita tämä ohjelma sellaisenaan mikroosi ja katso, mitä tapahtuu, kun annat käskyn RUN tai LIST.

Muuttuja-arvoituksia

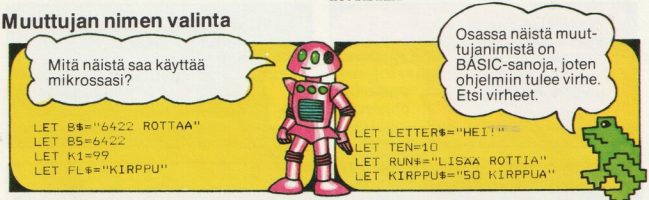
Muuttuja on tunnuksella varustettu paikka tietokoneen muistissa ja siihen on tallennettu tietoa. Tietokoneen käsketään tallentaa tieto BASIC-sanoilla LET ja INPUT, kuten alla on tehty. Muista, että sanoja sisältäviä tai kirjaimista, numeroista ja symboleista muodostuvia tietoja sanotaan jonoiksi ja että ne on pantava lainausmerkkeihin.

LET saa mikron antamaan muistipaikalle nimen ja panemaan sinne tietoa.



INPUT antaa muistipaikalle nimen ja panee mikron odottamaan, kunnes näppäilet tiedot sisään.

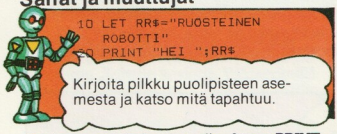
Muuttujan nimen valinta



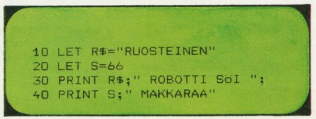
BASIC on hyvin tarkka siitä, millaisia nimiä annat muuttujille. Säännöt vaihtelevat mikrojen välillä. Esim. ZX81 hyväksyy vain yksittäisiä kirjaimia jonomuuttujien nimeksi.*

Jos annat lyhyitä sanoja muuttujien nimeksi, et saa käyttää BASIC-kielen sanoja, sillä ne panevat tietokoneen sekaisin. Kokeile yllä olevia rivejä, niin näet, mitkä ovat sallittuja omassa mikrossasi.

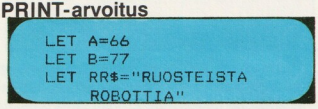
Sanat ja muuttujat



Kun laitat sanoja ja muuttujia yhteen PRINT-lauseeseen, pane sanat lainausmerkkeihin. Sanojen ja muuttujien väliin tulee puolipiste. Lainausmerkkien sisään pitää jättää välilyönti sanojen molemmin puolin. Kokeile, mitä tapahtuu, kun jätät välilyönnit pois yllä olevista riveistä.



PRINT-arvoitus



Osaatko kirjoittaa yllä annetuilla muuttujilla ohjelman, joka antaa ruudulle tällaisen tuloksen?

*Jos sinulla on ZX81, muuta tässä kirjassa annetut yli yhden kirjaimen mittaiset jonomuuttujien nimet mikrollesi sopiviksi.

Kurkistamme ohjelman sisälle

Tämä ohjelma muuntaa Fahrenheit-lämpötilat Celsius-asteiksi. Jos et ole varma, miten ohjelma toimii tai mikä muuttujien tehtävä on, lisää PRINT-rivejä, joilla tietokone näyttää ruudulla muuttujien sisällön. Näin voit kurkistaa ohjelman sisälle. Ymmärrät paremmin, miten se toimii.

```
10 INPUT A
20 LET F=32
30 LET B=5
40 LET C=B/9
50 LET D=A-F
60 LET R=D*C
70 PRINT R
```

Fahrenheit-lämpötila, jonka syötät mikrolle tallennetaan muuttujaan A.

Lisää ohjelmaan rivejä, jotta saat selville C:n ja D:n arvot.

Ohjelma-arvoitus

Kokeile, osaatko muuttaa ohjelman niin, että se laskee Celsius-asteista F-asteita.

Ohjelman kirjoitus: arvoituksia

Alla olevissa laatikoissa on ehdotuksia, joiden perusteella voi tehdä helppoja ohjelmia. Joudut käyttämään paljon muuttujia, joten pidä ne ja niiden nimet toisistaan erillään. On hyvä valita muuttujien nimiä, jotka muistuttavat sinua muuttujien tehtävästä tai siitä, mitä ne sisältävät. Kirjoita ohjelmat ensin paperille ja vasta sitten mikroon. Poista mahdolliset virheet.

1 Puhutko jugliaania?

Sinut on määrätty erikoistehtävään jugliaanin luo. He puhuvat suomea, mutta lisäävät jokaisen sanan alkuun JUG. Kirjoita ohjelma, joka kääntää suomen sanat jugliaanin kielelle.

Ohje: Käytä sanaa LET ja jonomuuttujaa tallentaaksesi kirjaimet JUG ja kysy käskyllä INPUT suomen kielen sanoja. Tulosta sitten molemmat muuttujat PRINT-käskyllä, jotta saat näkyviin jugliaanin kielen sanan. Selvennä ohjelmaa PRINT-lauseilla.

2 Nopeuslaskin

Osaatko kirjoittaa ohjelman, joka laskee juoksu- tai muun nopeuden? Tarvitset INPUT-rivit aikaa ja matkaa varten sekä laskutoimituksen nopeuden laskemiseksi ja PRINT-rivejä, jotta ohjelma on selvempi.

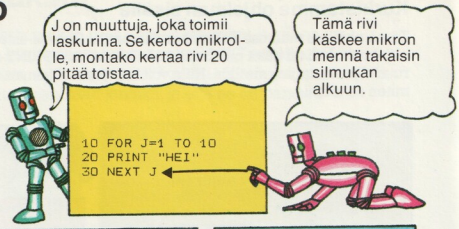
3 Makkaraohjelma

Robotti 1 syö 30 makkaraa tunnissa, mutta robotti 2 vain 20 samassa ajassa. Jos robotti 2 haluaa syödä ainakin 35 markkaa eikä robotti 1 suostu syömään hitaammin, montako makkaraa niiden on ostettava ja kauanko makkaroiden syöminen kestää?

Ohje: Aloita kolmella muuttujalla, joista kahdessa on makkaramäärät, jotka robotit pystyvät syömään ja yhdessä makkaramäärä, jonka robotti 2 haluaa syödä. Käytä lisämuuttujia, joihin panet ajan, joka robotti 2:lta menee 35 makkaran syömiseen ja makkaramäärään, jonka robotti 1 syö samassa ajassa.

Asioiden toisto

Yksi tapa saada mikro toistamaan jokin ohjelman osa on FOR/NEXT -silmukka. Aja tämän sivun yläosan ohjelmat, jotta näet, miten silmukka toimii. Yritä sitten ratkaista arvoitukset ja harjoitukset.




```
10 FOR K=1 TO 10
20 PRINT K
30 NEXT K
```

```
10 FOR I=1 TO 10
20 PRINT I;" X 8 ";
30 PRINT "= ";I*8
40 NEXT I
```

```
10 FOR L=1 TO 15
20 PRINT TAB(L);
"HEI"
30 NEXT L
```

Kirjaimia I, J, K ja L käytetään yleensä muuttujien nimenä silmukoissa.

Näet silmukkamuuttujan toimintatavan, kun panet PRINT-käskyn näyttämään sen arvon. Voit myös käyttää muuttujan arvoa silmukan sisällä. Kokeile kumpaakin.

Silmukka-arvoituksia

1

```
HEI HEI HEI HEI H
EI HEI HEI HEI HE
I HEI HEI HEI HEI
HEI HEI HEI HEI H
```

2

```
HEI
HEI
HEI
HEI
HEI
HEI
```

3

```
10 FOR I=1 TO 20
20 LET I=I-1
30 PRINT I
40 NEXT I
```

Osaatko kirjoittaa yksinkertaisen silmukkaohjelman, joka täyttää ruudun HEI-sanoilla? Entä ohjelman, joka kirjoittaa ruudun keskelle alekkain HEI-sanoja.

Portaat

```
10 FOR I=1 TO 25 STEP 5
20 PRINT I
30 NEXT I
```

```
10 FOR J=20 TO 1 STEP -1
20 PRINT J
30 NEXT J
```

Voit säätää muuttujan laskutapaa sanalla STEP ja luvulla, joka ilmoittaa, millaisin portain luvut tulee laskea. Jos annat negatiivisen luvun, tietokone laskee takaperin. Aja nämä ohjelmat ja yritä sitten muuttaa STEP-lukuja.

1

```
    HEI
   HEI
  HEI
 HEI
HEI
HEI
```

2

```
5   25
4   16
3   9
2   4
1   1
0   0
```

3

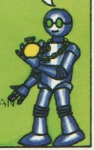
```
ALKU ?12
LOPPU ?0
ASKEL ?-3
12 9 6 3 0
```

Osaatko kirjoittaa ohjelmat, jotka saavat ruudun näyttämään tältä?

Viivesilmukat

Tietokoneet toimivat eri nopeuksilla, joten joudut ehkä pidentämään tai lyhentämään silmukkaa muuttamalla lukua 1000.

```
10 FOR J=10 TO 1 STEP-1
20 PRINT J
30 NEXT J
40 FOR K=1 TO 1000
50 NEXT K
60 PRINT "NOUSEMME ILMAAN"
```



Viivesilmukka on tyhjä silmukka, jossa ei ole käskyjä. Tässä ohjelmassa rivit 40–50 panevat mikron laskemaan 1:stä 1000:een, mikä saa mikron ikään kuin odottamaan hetken.

```
SALAINEN VIESTI
OPETTELE 5 SEKUNNISSA
SITTEN SE KATOAA

KOHTAA AGENTTI X KLO 14 ASEMALLA
```

Yritä kirjoittaa ohjelma, joka saa mikrosi kirjoittamaan jonkin salaisen viestin ja pyyhkimään sen pois 5 sekunnin kuluttua. Tarvitset viivesilmukan ja käskyn, joka puhdistaa näytön. Etsi kokeilemalla oikea luku viivesilmukkaan.

Kuvioarvoituksia

1

```
10 LET A$="***"
20 FOR J=1 TO 7
30 PRINT TAB(J);A$
40 NEXT J
50 FOR K=1 TO 3
60 PRINT TAB(J+1);A$
70 NEXT K
80 FOR L=7 TO 1 STEP-1
90 PRINT TAB(L);A$
100 NEXT L
```

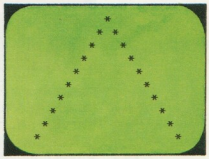
Tähtien asemesta voit käyttää grafiikkamerkkejä, jos mikrossasi on sellaisia.

```
80 FOR L=8 TO 18
90 PRINT TAB(L);A$;"**"
100 NEXT L
```

Vasemmalla oleva ohjelma piirtää ruudulle tähdistä muodostuvan kuvion. Aja ohjelma ja muuta sitä, jotta saat erilaisia kuvoita. Yllä on yksi ehdotus.

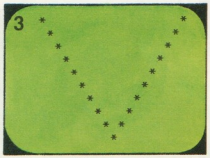
2

```
10 CLS
20 LET A=15
30 PRINT TAB(A);"*"
40 FOR K=1 TO 9
*50 PRINT TAB(?);"*";
*60 PRINT TAB(?);"*"
*70 ?
```



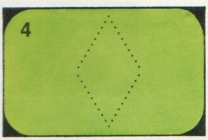
Osaatko täydentää tämän ohjelman rivit 50–70 niin, että ruutu näyttää sellaiselta kuin oikealla?

3



Muuta ohjelmaa niin, että se kirjoittaa kuvion ylös-alaisin.

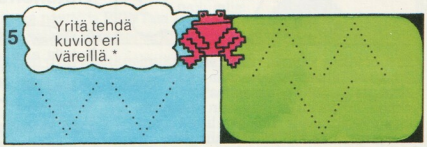
4



Voit panna kohtien 2 ja 3 ohjelmat yhteen; tulos on tällainen.

5

Yritä tehdä kuviot eri väreillä.*



Kokeile, osaatko muuttaa kuvio-ohjelmia niin, että tuloksena on tällaisia kuvioita.

*Katso mikrosi värikäskyt sen käsikirjasta.

Silmukka-arvoituksia

Tämän aukeaman arvoituksissa ja ohjelmissa käytetään sisäkkäisiä silmukoita, siis silmukoita, jotka ovat toisten silmukoiden sisällä. Alla on esimerkkejä sisäkkäisten silmukoiden toiminnasta.

```
10 FOR K=1 TO 3
20 FOR L=1 TO 5
30 PRINT K,L
40 NEXT L
50 NEXT K
```

Sisempi silmukka

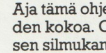
Tämä ohjelma tulostaa silmukan muuttujien arvot, joten näet, miten sisempi silmukka toimii.

```
10 FOR I=1 TO 4
20 FOR J=1 TO 4
30 PRINT
40 NEXT J
50 PRINT "HELLO"
60 NEXT I
```

Aja tämä ohjelma ja yritä muuttaa silmukoiden kokoa. Osaatko lisätä yhden sisäkkäisen silmukan, joka hidastaisi ohjelmaa?

Virheitä silmukoissa

```
10 FOR L=1 TO 15
20 PRINT TAB(5);"*";
TAB(10);"*"
30 FOR J=1 TO 5
40 PRINT
50 NEXT J
60 NEXT L
```



```
10 FOR I=9 TO 0 STEP-1
20 FOR J=9 TO 0 STEP-1
30 FOR K=9 TO 0 STEP-1
40 PRINT I;J;K
50 NEXT I
60 NEXT J
70 NEXT K
```

Ole huolellinen, jotta sisemmän silmukan molemmat osat tulevat ulomman silmukan sisään. Muuten ohjelmaan tulee virhe. Vasemmalla oleva ohjelma on virheetön, mutta sinun pitää etsiä virheet oikeanpuoleisesta ohjelmasta.

Binäärilaskuri

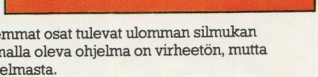
```
10 FOR A=0 TO 1
20 FOR B=0 TO 1
30 FOR C=0 TO 1
40 FOR D=0 TO 1
50 PRINT D+C*2+B*4+A*8;" = ";
60 PRINT A;B;C;D
70 NEXT D
80 NEXT C
90 NEXT B
100 NEXT A
```

Rivi 50 laskee binääriluvun desimaaliarvon.

Tietokone toistaa ensin sisimmän silmukan.

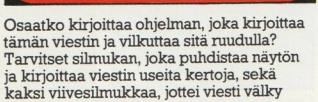
Tämä ohjelma käyttää neljää sisäkkäistä silmukkaa binäärijärjestelmän mukaiseen laskutapaan. Se ei osaa laskea kovin pitkälle, koska se käyttää vain nelinumeroisia binäärilukuja. Osaatko lisätä silmukoita ja muuttaa PRINT-rivit niin, että ohjelma laskee kahdeksannumeroisia binäärilukuja?

Välkkyvä viesti



Osaatko kirjoittaa ohjelman, joka kirjoittaa tämän viestin ja vilkuttaa sitä ruudulla? Tarvitset silmukan, joka puhdistaa näytön ja kirjoittaa viestin useita kertoja, sekä kaksi viivesilmukkaa, jottei viesti välky liian nopeasti.

Tietokonekello



Yritä kirjoittaa ohjelma, joka saa mikrosi toimimaan kellona. Tarvitset yhden silmukan sekunteja ja yhden minuutteja varten sekä yhden viivesilmukan.

Raketin lähtö -ohjelma

Osaatko kirjoittaa ohjelman, jossa on useita silmukoita ja joka saa yksinkertaisen raketin kuvan nousemaan hitaasti tv-ruutua ylös?

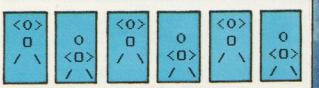
1. Ensin pitää puhdistaa ruutu, sitten kirjoittaa runsaasti tyhjiä rivejä niin, että raketin kuva kirjoittuu ruudun alareunaan.

2. Voit piirtää yksinkertaisen raketin tavallisilla merkeillä, joita panet PRINT-lauseisiin.

3. Saat raketin nousemaan ruutua pitkin, kun lisäät tyhjiä PRINT-rivejä ja sisäkkäisiä viivesilmukoita, jottei raketti kulje liian nopeasti.

Hyppivä mies -ohjelma

Kokeile, osaatko kirjoittaa helpon animaatio-ohjelman, jossa oikealla olevan kaltainen hahmo hyppii ylösalaisin ja kulkee samalla ruudun poikki.



Hahmojen piirtäminen

Voit piirtää yksinkertaisen hahmon kahteen paikkaan (ks. oikealla) PRINT-lauseilla, joissa on kirjaimia ja symboleita.

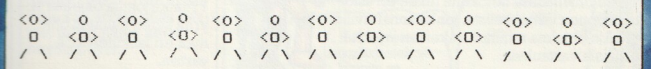
Ohjelman kirjoitus

1. Tarvitset silmukan, jonka koko vastaa näyttöruutusi leveyttä. Pane se laskemaan 2:n portaissa. Syy tähän selviää, kun kirjoitat PRINT TAB -käskyt.

2. Silmukan sisällä pitää puhdistaa näyttö ja PRINTata ensimmäinen hahmo. Käytä TAB-sanaa, jotta hahmon paikka vaihtuu joka kerta, kun se printataan.

3. Lisää viivesilmukka, jotta hahmo pysyy hetken ruudulla.

4. Toista tämä rutiini eri TAB-käskyllä toista hahmoa varten.



IF/THEN -harjoituksia

Sanoilla IF ja THEN vertaillaan tietoja ja pannaan tietokone tekemään erilaisia asioita vertailun tuloksesta riippuen.

> = tarkoittaa suurempi tai sama kuin
< = tarkoittaa pienempi tai sama kuin

```
10 LET A=8
20 INPUT B
30 IF B=A THEN PRINT "YHTÄ SUURET"
40 IF B>A THEN PRINT "EIVÄT YHTÄ SUURET"
50 IF B>A THEN PRINT B;" ON PIENEMPI KUIN ";A
60 IF B<A THEN PRINT B;" ON SUUREMPI KUIN ";A
```

Yritä ajaa tämä ohjelma, jotta opit näiden symbolien merkityksen tietojen vertailussa.

```
10 PRINT "MONTAKO LASTUA JAKSAT SYÖDÄ?"
20 INPUT C
30 IF C>=0 AND C<=10 THEN PRINT"JAAT NÄLKÄISEKSI"
40 IF C>10 AND C<=25 THEN PRINT"AIKA VÄHÄN"
50 IF C>25 AND C<1000 THEN PRINT "AHMATTI"
60 IF C<0 TAI C>1000 THEN PRINT "!!!!"
```

Yleensä tietokoneella voi verrata useita asioita yhtä aikaa käyttämällä IF/THEN-lauseissa sanoja AND (ja) sekä OR (tai). Aja tämä ohjelma, jotta näet miten ylimääräiset sanat toimivat.

Kertolaskuja

```
10 LET A=13
20 FOR J=1 TO 13
30 PRINT "PALJONKO ON ";J;" X ";A;" ";
40 INPUT B
50 IF B=J*A THEN PRINT "OIKEIN"
60 NEXT J
```

Tämä ohjelma kyselee kertolaskuja 13*13 saakka. Osaatko lisätä IF/THEN-käskyjä, jotka kertovat, jos vastaus on väärä ja antavat lisäksi oikean vastauksen?

Tunnussana

```
10 LET S$="MAKKARA"
20 PRINT "ANNA TUNNUSSANA ";
30 INPUT P$
40 IF P$=S$ THEN PRINT "OK, VOIT JATKAA"
```

Osaatko täydentää tätä ohjelmaa niin, että tietokone kirjoittaa viestin jos annat väärän tunnussanan? Yritä laajentaa ohjelmaa niin, että tietokone kysyy myös salaista numeroa.

Haarautuminen

Tietokoneelle voi antaa paljon erilaisia ohjeita THEN-sanan jälkeen. Voit esimerkiksi käskeä sitä lopettamaan ohjelman tai hyppäämään toiselle riville GOTO-sanalla.

```
10 LET C=0
20 PRINT "JOKO KYLLÄSTYIT?"
30 INPUT B$
40 LET C$=C+1
50 IF B$="KYLLÄ" THEN STOP
60 IF C>10 THEN PRINT "NYT OLET VARMASTI
   KYLLÄSTYNYT"
70 IF B$="EN" THEN GOTO 20
80 PRINT "ALA NARRAA"
90 GOTO 20
```

Mitä tapahtuu, jos kirjoitat "HÖLYNPÖLYÄ"

Tässä ohjelmassa tietokone tekee eri asioita riippuen vastauksesta, jonka annat rivillä 30. Aja ohjelma muutaman kerran ja käytä erilaisia vastauksia.

Tietokonelaskin

```
ANNA LUKU ? 7
ANNA TOINENKIN ? 11
HALUATKO:
LASKEA YHTEEN, VÄHENTÄÄ,
JAKAA VAI KERTOA
? KERTOA
VASTAUS ON 77

10 INPUT X
20 CLS
30 PRINT "ARVAA LUKU"
40 INPUT Y
```

Tässä on tulostus ohjelmasta, joka osaa laskea yhteen, vähentää, jakaa ja kertoa kirjoittamasi kaksi lukua. Yritä kirjoittaa ohjelma ja yritä sitten muuttaa ohjelmaa eri laskutehtäviä varten.

Arvausleikkejä

1

```
10 INPUT X
20 CLS
30 PRINT "ARVAA LUKU"
40 INPUT Y
50 IF Y=X THEN GOTO 70
60 GOTO 30
70 PRINT "OIKEIN"
```

1. Pane tietokone kertomaan, milloin arvauksesi on liian pieni tai liian iso, jotta sinun on helpompi yrittää uudestaan.

2. Lisää muuttuja, joka laskee, montako kertaa olet arvannut ja aseta arvauskerroille yläraja IF/THEN-käskyllä.

Tämä ohjelma on helppo numeroiden arvausleikki. Yksi henkilö valitsee numeron ja toinen yrittää arvata, mikä se on. Yritä parantaa ohjelmaa yllä annettujen ohjeiden perusteella.

Lisää ohjelmaan rivejä vihjettä varten, jotta arvaaminen on helpompaa.

2

```
VIHJE
KURNUTTAVA ELÄIN
? SIILI
EI
? SAMMAKKO
KYLLÄ
```

Tässä on ruutunäyttö sanojen arvauspelistä. Osaatko kirjoittaa tarvittavan ohjelman? Se on samanlainen kuin numeropeli, mutta siinä käytetään jonoja.

Ravikilpailut

Tässä on ravikilpailupelin listaus, josta puuttuu jotain. Yritä lisätä puuttuvat rivinumerot GOTO-sanojen jälkeen tähdellä merkityille riveille. Alla on peliohjeet ja vihjeitä pelin parantamiseksi.

Yleensä mikrolla voi kirjoittaa INPUT-riville sanoja kuten riveillä 20 ja 30 on tehty.

```
 10 LET N=0
 20 INPUT "VOITTAJA :";H1
 30 INPUT "KAKKONEN :";H2

 40 CLS
 50 LET N=N+1

 60 INPUT "ARVAA VOITTAJA :";G1
 70 INPUT "ARVAA KAKKONEN :";G2

* 80 IF G1=H1 AND G2=H2 THEN GOTO ?
* 90 IF G1=H2 OR G2=H1 THEN GOTO ?
*100 IF (G1=H1 AND G2<>H2) OR
  (G2=H2 AND G1<>H1 THEN GOTO ?

 110 PRINT "VÄÄRIN"
*120 IF N=4 THEN GOTO ?
 130 PRINT "YRITÄ UUDESTAAN":GOTO 50

 140 PRINT "YKSI ARVAUS OIKEIN":GOTO 120
 150 PRINT "OIKEA HEVONEN, VÄÄRÄ SIJOITUS":GOTO 120
 160 PRINT "OIKEIN"
 170 PRINT "VOITTAJA:";H1;" KAKKONEN:";H2
```

Ensimmäinen pelaaja valitsee, mikä hevonen voittaa ja mikä tulee kakkoseksi.

N laskee toisen pelaajan arvauskertojen määrän.

Muista lisätä rivinumerot.

Yleensä mikro hyväksyy kaksi käskyä samalle riville. Jos mikrosi panee vastaan, pane toinen käsky omalle rivilleen ja jätä kaksoispiste pois.

Peliohjeet

Hevosia on kuusi ja ne on numeroitu 1–6. Ensimmäinen pelaaja valitsee, mikä hevonen voittaa ja mikä tulee toiseksi. Toinen pelaaja saa arvata neljä kertaa, mitkä nämä kaksi hevosta ovat.

Vihjeitä pelin parantamiseksi

1. Keksi pistelaskujärjestelmä ja kirjoita se ohjelmaan.

2. Anna pelaajille mahdollisuus pelata uusi peli.

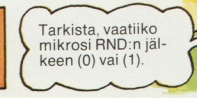
Satunnaisluvut

Sana RND tuottaa satunnaisluvun (englanniksi random number), mutta käskyn käytössä on eroa mikrojen välillä. Alla on annettu erilaisia käskyjä, mutta tarkista kuitenkin oman mikrosi käyttämä käsky sen ohjekirjasta tai tämän kirjan takana olevasta muunnostaulukosta. Yritä sitten tehdä tämän aukeaman harjoitukset ja ratkaista arvoitukset.

```
PRINT RND(99)
PRINT RND(10)
```

Joissain mikroissa käytetään pelkkää sanaa RND ja suluissa olevaa lukua, jos halutaan kokonaisluku yhden ja suluissa olevan luvun väliltä.

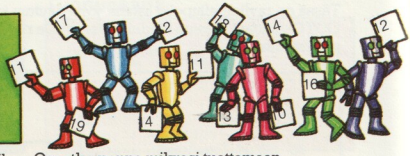
```
PRINT INT(RND(1)*99+1)
PRINT INT(RND(0)*99+1)
PRINT INT(RND*99+1)
```



Satunnaisen kokonaisluvun saa useimmissa mikroissa sanalla INT, jota seuraa RND ja mahdollisesti (1) tai (0). Tämä kerrotaan luvulla, joka ilmoittaa, miltä väliltä satunnaisluku halutaan ja sitten lisätään ensimmäinen haluttu luku. Esimerkiksi kaikki yllä olevat käskyt panevat mikron valitsemaan satunnaisluvun väliltä 1 ja 99.

```
PRINT RND(6)+5
PRINT INT(RND(1)*6+5)
PRINT INT(RND(0)*6+5)
PRINT INT(RND*6+5)
```

Nämä käskyt tuottavat satunnaisluvun väliltä 5 ja 10. Valitse omaan mikroosi sopiva käsky.



Osaatko panna mikrosi tuottamaan satunnaislukuja väliltä 10 ja 20?

Ohjelma-arvoitus

Katso sivun 23 numeroiden arvausleikkiä ja yritä muuttaa sitä niin, että tietokone valitsee arvattavaksesi luvun väliltä 1 ja 20.

Lukusarja

```
ARVAA SEURAAVA LUKU
TASSA SARJASSA
 4  13  22
? 31
OIKEIN
```

Yritä kirjoittaa ohjelma tätä lukusarjapeliä varten. Osa ohjelmasta on annettu sivun keskellä.

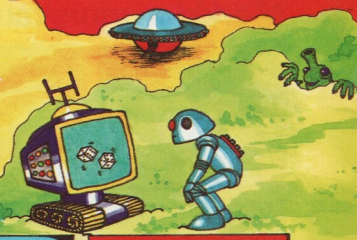
```
*10 LET X=?
*20 LET Y=?
 30 FOR I=1 TO 3
 40 PRINT X+I*Y
 50 NEXT I
```

```
PRINT X+I*I
PRINT I*I-Y
PRINT X+Y-I*I
PRINT Y+X^I
```

Sinun pitää kirjoittaa satunnaiskäskyt riveille 10 ja 20 ja lisätä PRINT ja IF/THEN-rivejä. Oikealla on vihjeitä rivin 40 muuttamiseksi erilaisia lukusarjoja varten. Osaatko laajentaa ohjelmaa niin, että tietokone valitsee sarjan satunnaisesti joka kerta, kun ajat ohjelman?

Pako Zorgosista

Kokeile, osaatko kirjoittaa ohjelman tätä peliä varten. Olet jäänyt vieraalle planeetalle nimeltä Zorgos. Tarvitset 50 Zlastua, jotta planeetta-ajoneuvollasi olisi riittävästi muistia, jolla se pystyy opastamaan sinut takaisin maapallolle. Sinulla on 10 Zlastua, mutta lisää saat vain pelaamalla rohkeaa uhkapeliä vieraan tietokoneen kanssa. Alla olevissa ruuduissa näkyy osia ohjelman ajosta.



```
SINULLA ON 10 LASTUA
MONTAKO PANET PELIIN : 5
NOPANHEITTO, PAINA P:P
     5   2
PIDA LASTUSI
```

```
SINULLA ON 10 LASTUA
MONTAKO PANET PELIIN : 9
NOPANHEITTO, PAINA P:P
     6   4
VOITIT 3-KERTAISEN MAARAN
SINULLA ON 28 LASTUA
```

```
NOPAT      PANOKSESI

TUPLAT     2-KERTAISTUU
10 TAI 11  3-KERTAISTUU
6 TAI 7    PIDA LASTUSI
MUUT:      HAVIAT
```

Panet peliin haluamasi määrän Zlastuja. Riippuen siitä, mitä numeroita tietokone saa heittäessään noppaa sinä joko voitat, häviät tai saat pitää lastusi. Jos molemmat nopat antavat saman luvun (tuplat) panoksesi kaksinkertaistuu, jos noppien luvut ovat yhteenlaskettuna 10 tai 11 panoksesi kolminkertaistuu ja niin edelleen (ks. punaista ruutua yllä).

Paperi, kivi vai sakset: etsi virheet

Tällä ohjelmalla pelataan peliä Paperi, kivi vai sakset tietokonetta vastaan, mutta ohjelma on täynnä virheitä. Yritä löytää virheet ja korjata ne siten, että ohjelma toimii kunnolla. Oikealla on kerrottu, miten ohjelman pitäisi toimia.

```
10 CLS
20 LET C=0
30 LET A=0
40 LET F=0
50 LET R=INT(RND(1)*4+1)
60 IF R=1 THEN LET C$="PAPERIN"
70 IF R=2 THEN LET C$="KIVEN"
80 IF R=3 C$="SAKSET"
90 PRINT "OLEN VALMIS"
100 PRINT "HALUATKO PAPERIN, KIVEN VAI SAKSET?"
110 INPUT A$
120 PRINT
130 IF C$="PAPERIN" AND A$="SAKSET" THEN LET F=1
140 IF C$="KIVEN" AND A$="PAPERIN" THEN LET F=1
150 IF A$="SAKSET" AND A$="KIVEN" THEN LET F=1
160 IF C$=A$ THEN LET F=2
170 PRINT "SINA VALITSIT ";A$
180 PRINT "MINA VALITSIN ";C$
190 PRINT "JOTEN
200 IF F=0 THEN PRINT "MINA VOITAN"
210 IF F=F THEN PRINT "SINA VOITAT"
220 IF F=2 THEN PRINT "TULI TASAPELI"
230 IF F=0 THEN LET A=A+1
240 IF F=1 THEN LET C=C+1
250 PRINT "PISTETILANNE ON:"
260 PRINT "MINA  :";C
270 PRINT "SINA  :";A
280 IF C>10 AND A>10 THEN GOTO 40
290 PRINT "PELI ON PAATTYNYT"
```

C laskee tietokoneen pisteet.
A laskee sinun pisteesi.
F kertoo tietokoneelle, kuka voitti riveillä 130–160.

Tietokoneen valinta (paperi, kivi tai sakset) tallennetaan muuttujaan C$ ja se riippuu R:n arvosta.

IF...THEN-rivit selvittävät, kuka voittaa. Jos sinä voitat, tietokone panee F:n arvoksi 1 ja tasapelissä 2. Muutoin F on 0 eli tietokone voittaa.

Tietokone ilmoittaa voittajan ja laskee pistetilanteen selvittämällä F:n arvon.

Muistatko, miten tätä peliä pelataan? Paperi peittää kiven, kivi on kovempi kuin sakset ja sakset leikkaavat paperia.

Merkkien murskausta

Tietokone osaa tehdä jonojen sisällä oleville merkeille yhtä ja toista. Tämän sivun ohjelmissa on BASIC-käskyjä, joilla käsitellään jonoja. Jos sinulla on Sinclair (Timex) tietokone, joudut käyttämään osittain erilaisia käskyjä, jotka on annettu alla.

```
LET R$="ROBOTTI"
PRINT LEFT$(R$,3)
PRINT RIGHT$(R$,4)

LET C$="MAAORAVA"
PRINT LEFT$(C$,3)
PRINT RIGHT$(C$,5)
```

LEFT$ ja RIGHT$ käskevät tietokonetta ottamaan tietyn määrän merkkejä jonon vasemmasta tai oikeasta päästä. Sulkuihin pannaan, montako merkkiä tietokoneen pitää ottaa.

```
10 LET K$="KENGURU"
20 PRINT "ENSIMMAISEN"
30 PRINT "KIRJAIMEN PAIKKA";
40 INPUT S
50 PRINT "MONTAKO KIRJAINTA";
60 INPUT N
70 PRINT MID$(K$,S,N)
```

MID$ käskee tietokoneen ottaa merkkejä jonon keskeltä. Suluissa oleva ensimmäinen luku ilmoittaa, mistä merkistä aletaan laskea ja toinen ilmoittaa, montako merkkiä otetaan.

Tietokone laskee välilyönnit samalla tavalla kuin kirjaimet ja symbolit.

```
10 PRINT "ANNA SANA"
20 INPUT W$
30 LET L=LEN(W$)
40 PRINT "SANASSA ";W$;
50 PRINT "ON ";L;" KIRJAINTA"
```

Sana LEN laskee jonossa olevien merkkien määrän. Kokeile tätä ohjelmaa, jotta näet, miten se toimii.

Sinclair (Timex) tietokoneet

```
LET A$="ABCDEFGHIJKLMNOP"
PRINT A$(4 TO 6)
DEF
PRINT A$(14 TO 16)
NOP
```

Sinclair (Timex) tietokoneet eivät käytä sanoja LEFT$, RIGHT$ ja MID$. Niiden asemesta tietokoneelle pitää kertoa yllä esitetyllä tavalla, mitkä merkit sen pitää valita (ensimmäinen luku ilmoittaa ensimmäisen merkin ja toinen luku viimeisen merkin, joka luetaan).

Jonoarvoituksia

1

```
10 INPUT "SANA";N$
20 LET L=LEN(N$)
30 FOR K=1 TO L
*40 PRINT TAB(K);
 MID$( ?, ?, ? )
50 NEXT K
```

```
L     M
E     A
N     K
K     K
K     A
I     R
      A
```

Osaatko täydentää ohjelmarivin 40 kysymysmerkkien kohdalta siten, että ruutu näyttää tällaiselta? Ohje: sanalla MID$ voit poimia jonosta yhden merkin kerrallaan.

2

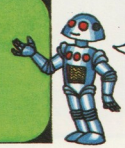
```
RUN
ANNA SANA
? KENGURU
URUGNEK
```

Yritä saada tietokone kirjoittamaan sana takaperin käyttämällä sanaa MID$ ja silmukkaa, jossa on askeleena (STEP) -1.

3

```
10 LET S$=
 "NELIöT"
20 LET L=LEN(S$)
30 PRINT S$
40 FOR J=1 TO 1
*50 ?
60 NEXT J
```

```
N E L I ö T
E L I ö T N
L I ö T N E
I ö T N E L
ö T N E L I
T N E L I ö
N E L I ö T
```



Kokeile, osaatko täyttää puuttuvan rivin sanoilla LEFT$ ja RIGHT$ siten, että ruutu näyttää tällaiselta.

Pisin sana

Rivillä 50 sinun pitää käskeä tietokonetta tallentamaan pisimmän siihen mennessä annetun sanan muuttujaan A$. Yritä käyttää sanoja IF/THEN ja LEN.

```
 10 LET A$=""
 20 PRINT "ANNA SANOJA"
 30 FOR J=1 TO 5
 40 INPUT W$
*50 ?
 60 NEXT J
 70 PRINT "PISIN SANA :"
 80 PRINT A$
```

```
ANNA SANOJA
? KISSA
? SISILISKO
? NORSU
? KOBRA
? MUURAHAINEN
PISIN SANA :
MUURAHAINEN
```

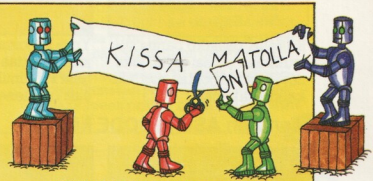
Tämä ohjelma etsii pisimmän sanan viiden sanan joukosta. Kirjoita puuttuvat rivit ja aja ohjelma.

Lyhyin sana

Osaatko kirjoittaa ohjelman, joka etsii lyhyimmän sanan? Ohjelma on muuten samanlainen kuin yllä, paitsi että tarvitset muuttujan, joka on pitempi kuin mikään kirjoittamasi sana, jotta tietokone voi verrata kirjoittamiasi sanoja siihen. Myös IF/THEN-käskyä pitää muuttaa.

Sanaeditori

Alla oleva listaus on sanaeditoriohjelma, jolla voit kirjoittaa lauseen ja muuttaa sitten sen sanoja. Ennen kuin voit ajaa ohjelman sinun pitää täyttää tähdellä merkityillä riveillä olevat tyhjät kohdat. Käytä apuna oikealla annettuja neuvoja.




```
  10 CLS
  20 PRINT "ANNA LAUSE"
  30 INPUT S$
* 40 LET S$= ?
  50 PRINT "VANHA SANA";
  60 INPUT W$
* 70 LET W$= ?
  80 PRINT "UUSI SANA";
  90 INPUT N$
*100 LET LS= ?
*110 LET LW= ?
 120 LET A$=""
 130 LET K=1
*140 IF MID$(S$, ?, ?)=W$ THEN
 LET A$=S$
*150 IF A$=S$ THEN LET S$=LEFT$
 ( ?, ? )+ ? +RIGHT$(A$,LS-(K+LW-2))
 160 LET LS=LEN(S$)
 170 LET K=K+1
*180 IF K<=LS-LW+1 THEN GOTO ?
 190 PRINT S$
 200 GOTO 50
```

Selvitä tämän ohjelman toiminta näin: kirjoita paperille jokin lause ja tee sille ohjelmassa annetut käskyt.

Käske tietokoneen lisätä välilyönti S$:n ja W$:n alkuun ja loppuun. Tiedätkö miksi?

Pane LS:n pituudeksi lauseen (S$) pituus ja LW:n pituudeksi sanan (W$) pituus.

Täydennä tämä rivi niin, että tietokone etsii vaihdettavan sanan (W$) lauseesta. Ohje: laske merkit K:lla.

Tämä rivi laskee, montako merkkiä poistettavan sanan vasemmalla puolella on, panee uuden sanan lauseeseen ja lisää loput lauseesta. Osaatko lisätä puuttuvat muuttujat ja luvun?

Lisää merkkien murskausta

Tietokoneen sisällä merkit esitetään koodinumeroilla. Merkkejä voi käsitellä käyttämällä näitä koodinumeroita. Sana CHR$ muuttaa koodinumeron merkiksi. ASC (tai CODE Sinclair (Timex) tietokoneissa) tekee päinvastoin eli muuttaa merkin koodinumeroksi. Yleensä eri kotitietokoneissa käytetään samoja ns. ASCII-koodeja, mutta ZX81:llä (Timex 1000) on omat koodinsa. Löydät oman mikrosi kooditaulukon sen käsikirjasta.

Sanan CHR$ käyttö

```
PRINT CHR$(65)
A
PRINT CHR$(90)
Z
```

```
PRINT CHR$(38)
A
PRINT CHR$(63)
Z
```

Kokeile muutaman kerran käskyä PRINT CHR$ näillä ja muilla numeroilla. Jotkut numerot on varattu esim. näppäimille SPACE (välilyönti) ja RETURN, joten ruudulla ei näy mitään. Mikrosi käsikirjasta saat selville tällaiset numerot.

Kirjainarvoituksia

1

```
*10 FOR K=? TO ?
 20 PRINT CHR$(K);
 30 NEXT K
```

Kokeile, osaatko panna silmukkaan oikeat luvut, jotta tietokone kirjoittaa ruudulle kaikki aakkoset.

2

```
abcdefghijklm
nopqrstuvwxyz
```

Yritä kirjoittaa silmukka, joka printtaa aakkoset pienillä kirjaimilla (jos mikrossasi on pienet kirjaimet)

3

```
I D I P Q C H K A
S K Q C H U A M S F
 S N M Q T O P X A
```

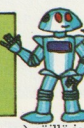
Osaatko kirjoittaa lyhyen ohjelman, joka kirjoittaa ruudulle satunnaisesti valittuja kirjaimia?

Sanojen ASC ja CODE käyttö

```
PRINT ASC("P")
PRINT ASC("+")
PRINT ASC(" ")
```

Väli

```
PRINT CODE("4")
PRINT CODE("U")
PRINT CODE("C")
```



Kokeile sanaa ASC (tai CODE Sinclair (Timex) tietokoneissa) näillä ja muilla merkeillä, jotta näet, mitä merkkejä tietokoneesi antaa.

Kirjainvertailu

```
?P,O
O ON ENNEN KUIN P
?L,B
B ON ENNEN KUIN L
?S,C
C ON ENNEN KUIN S
```

Kirjoita symboleita > ja < käyttäen ohjelma, joka vertaa mitä tahansa kahta kirjainta ja panee ne aakkosjärjestykseen.

Suur- ja pienaakkoset

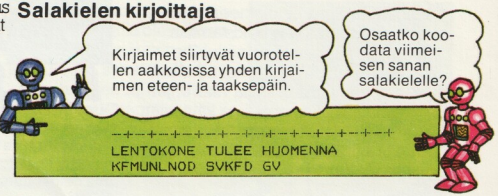
```
 10 PRINT "MIKÄ ON VIESTI"
 20 INPUT M$
 30 FOR J=1 TO LEN (M$)
 40 LET X$=MID$(M$,J,1)
*50 IF X$>="a" AND X$<="z" THEN
 PRINT CHR$(ASC(X$)-?);
*60 IF X$>="A" AND X$<="Z" THEN
 PRINT CHR$(ASC(X$)+?);
 70 IF X$<"A" OR X$>"z" THEN PRINT X$;
 80 NEXT J
```

Yritä lisätä puuttuvat numerot riveille 50 ja 60, jotta tietokone muuttaa viestin pienaakkosista suuraakkosiin tai päinvastoin. Et voi ajaa tätä ohjelmaa, jos tietokoneesi käyttää vain isoja kirjaimia.

Salakieliohjelma

Tässä on muutama ehdotus ohjelmiksi, jotka muuttavat viestejä salakielelle. Oikealla olevasta kaaviosta näkyy, miten ensimmäinen ohjelma toimii. Lisää ohjelmaan puuttuvat numerot ja yritä sitten ajaa se. Huom! Ohjelmissa Ä ja Ö ovat A ja O.

Salakielen kirjoittaja



Salakielen kirjoitusohjelma

```
  10 PRINT "MIKÄ ON VIESTI"
  20 INPUT M$
* 30 FOR J=1 TO?
* 40 LET X=?
* 50 IF X<?OR X>? THEN LET N=X:
  GOTO 100
* 60 IF INT(J/2)=J/2 THEN LET N=X?1
* 70 IF INT(J/2)<>J/2 THEN LET N=X?1
* 80 IF N<? THEN LET N=N+26
* 90 IF N>? THEN LET N=N-26
*100 PRINT?
 110 NEXT J
```

Pane silmukan kooksi sama kuin M$:n merkkien määrä.

Pane mikro valitsemaan kirjain kerrallaan ja tallentamaan sen ASCII-koodinumero X:ään.*

Lisää kaksi numeroa, joilla tarkistetaan, että jokainen merkki on kirjain. (Numerot ja välilyönnit eivät tässä salakielessä muutu.)

Kirjainten muuttamista varten lisää X:ään 1 jos silmukkalaskurissa (J) on parillinen luku ja vähennä 1 jos siellä on pariton luku.

Jos muutettu numero (N) ylittää aakkosten jommankumman pään, lähetä se takaisin vastakkaiseen päähän lisäämällä tai vähentämällä 26.

Printtaa koodattu kirjain sanalla CHR$.

Avainlukukoodi	← N = X + avainluku — N = N − 26 →
Aakkoset	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Koodatut aakkoset	GHIJKLMNOPQRSTUVWXYZABCDEF

Tässä salakielessä aakkosia siirretään tietyn kirjainmäärän (N) verran. Siirtymä riippuu avainluvusta. Tässä esimerkissä avainluku on 6 joten aakkoset siirtyvät kuuden kirjainmen verran. Yritä kirjoittaa ohjelma yllä olevan mallin mukaan. Voit itse valita avainluvun.

Silmukkakoodi

J = 1 2 3 4 5 6 7 8 9 10 11 12
E T S I A G E N T T I

Tässä salakielessä silmukkalaskurin (J) arvo lisätään kunkin kirjaimen ASCII-numeroon. Yritä kirjoittaa kaksi ohjelmaa, joista toinen koodaa viestin ja toinen purkaa sen.

Käänteiskoodi

K O H T A A A G E N T T I
O K H T A A A E G T N I T

Tässä salakielessä viesti jaetaan kahden kirjaimen ryhmiin ja niissä kirjaimet (ja välimerkit) vaihtavat keskenään paikkaa. Käytä ohjelmassa STEP 2 -silmukkaa ja printtaa kukin kirjainpari käänteisessä järjestyksessä.

*Jos sinulla on ZX81 (Timex 1000), käytä sen omia koodinumeroita.

ÄTÄMO NASALNINEV EITS !

INKEY$-harjoituksia

Sana INKEY$ saa tietokoneen katsomaan näppäimistöä ja tarkistamaan, onko jotain näppäintä painettu. Ero INPUT-sanaan on siinä, ettei tietokone pysähdy odottamaan sinua vaan ohjelma jatkuu. Joissain mikroissa käytetään INKEY$:n asemesta jotain toista sanaa. Tarkista siis asia mikrosi käsikirjasta ja yritä sitten tämän sivun harjoituksia.

Tässä on esimerkkejä eri tietokoneiden vastineista sanalle INKEY$. Tarkista mikrosi käsikirjasta tai tämän kirjan sivulla 140 olevasta muuntotaulukosta, minkä käskyn oma tietokoneesi ymmärtää.

```
10 LET A$=INKEY$
20 IF A$="" THEN PRINT "!";
30 IF A$<>"" THEN PRINT A$;
40 GOTO 10
```

Yritä ajaa tämä ohjelma käyttäen oman mikrosi vaatimaa käskyä. Kun painat jotain näppäintä, tietokone kirjoittaa vastaavan merkin, mutta muutoin se kirjoittaa huutomerkin.

```
HEI        HEI            H
EI  HEI                  HE
I       HEI             HEI
HEI             HEI      HE
I   HEI    HEI
```

Kokeile seuraavaksi, osaatko kirjoittaa ohjelman, joka panee mikrosi printtaamaan sanan HEI aina, kun painat jotain näppäintä ja jättämään muutoin tyhjää kuten yllä olevassa ruudussa. Ohjelma muistuttaa vasemmalla annettua ohjelmaa.

Panemme tietokoneen odottamaan

```
10 LET A$=INKEY$
20 IF A$="" THEN
PRINT "!";
30 IF A$<>"" THEN
PRINT A$;
40 GOTO 10
```

```
10 LET N=0
20 LET A$=INKEY$
30 IF A$<>"" THEN GOTO 70
40 LET N=N+1
50 IF N<50 THEN GOTO 20
60 PRINT " ";:GOTO 10
70 PRINT A$;:GOTO 10
```

Joskus on hyvä panna tietokone odottamaan hetken, ennen kuin ohjelma jatkuu. Joissain BASIC-murteissa sanan INKEY$ jälkeen pitää panna sulkuihin jokin numero, kuten ylhäällä vasemmalla on tehty. Numero kertoo tietokoneelle (sekunnin murto-osina), kauanko sen on odotettava ennen ohjelman jatkamista. Jos mikrossasi ei ole tätä mahdollisuutta, voit silti saada sen odottamaan panemalla sanan INKEY$ silmukkaan käyttäen sanaa GOTO, kuten ylhäällä oikealla on tehty.

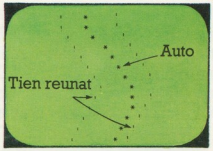
Nopeaa matikkaa

Löydätkö tämän ohjelman pukit? Korjaa ne, jotta ohjelma toimii oikein. Tietokoneen pitäisi valita satunnaisesti kaksi lukua väliltä 1 ja 25. Sinun pitää laskea luvut yhteen ja painaa jotain näppäintä heti, kun oikea vastaus näkyy ruudulla.

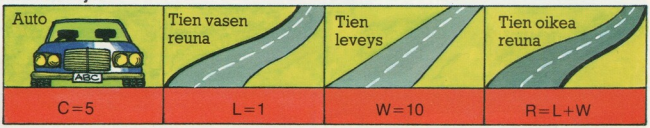
```
10 CLS
20 PRINT "PAINA JOTAIN
NÄPPÄINTÄ
30 PRINT "KUN NÄET OIKEAN
VASTAUKSEN"
40 LET N=0
50 LET X=INT(RND(1)+25+1)
60 LET Y=INT(RND(1)*25+1)
70 PRINT
80 PRINT "X;" + ";Y;" = "
90 LET N=N+1
100 PRINT N
110 LET INKEY$=A$
120 IF A$<>"" THEN GOTO 180
130 FOR K=1 TO 100:NEXT K
140 IF N<30 THEN GOTO 90
150 PRINT "HUONOSTI KÄVI.
VASTAUS ON ";X+Y
160 FOR K=1 TO 1000:NEXT T
170 GOTO 30
180 IF N<>X+Y THEN GOTO 150
190 PRINT "OIKEIN. VASTAUS ON X+Y"
```

Miten tehdään autopeli

Yritä kirjoittaa ohjelma tätä autopeliä varten alla annettujen ohjeiden perusteella. Oikealla oleva ruutu kertoo, miltä pelin pitäisi näyttää. Auton ja tien saat ruudulle sanoilla PRINT ja TAB sekä symboleilla * (auto) ja ! (tien reunat). Tie kiemurtelee ruutua pitkin alas samalla kun ohjaat autoa kahdella näppäimellä. Varo, ettet aja ojaan!

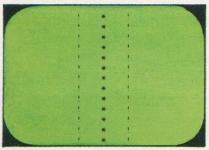


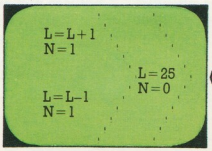
1. Muuttujat



Tarvitset neljä muuttujaa, C, L, W ja R auton ja tien reunojen TAB-kohtien laskemiseksi. Joudut ehkä muuttamaan yllä annettuja lukuja mikrosi näyttöön sopiviksi. Tyhjennä näyttö ja kirjoita nämä muuttujat viidelle ensimmäiselle riville.

2. Tien piirtäminen

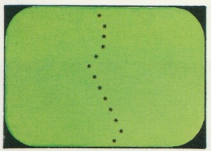


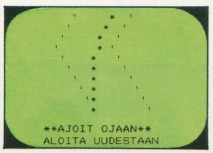


Pane tietokone näyttämään ruudulla maantie ja auto. Käytä käskyä PRINT TAB muuttujan ja symbolin kanssa. Käytä sanaa GOTO käskyjen toistamiseen ja pitkän suoran tien piirtämiseen.

Jotta saat tien kiemurtelemaan, sinun pitää lisätä rivejä, jotka muuttavat L:n arvoa joka kerta, kun käsky PRINT TAB toistuu. Varmista myös, ettei tie joudu pois näyttöruudulta. Siihen tarvitset lisämuuttujan (N). Aseta sen arvoksi 1 jos L< = 1 ja 0 jos L> = näyttöruutusi leveys. Käske sitten tietokonetta lisäämään tai vähentämään L:stä 1 riippuen N:n arvosta.

3. Auton ohjaaminen





Auton ohjaamiseksi tarvitset INKEY$ rivin. Valitse sitten kaksi näppäintä (esim. < ja >) sekä lisää tai vähennä C:stä 1 sen mukaan, kumpaa näppäintä painetaan. Jos ohjelma etenee niin nopeasti, ettet ehdi ohjata autoa, lisää siihen viivesilmukka.

Lopuksi sinun pitää tarkistaa, onko auto ajanut ojaan. Vertaa C:tä L:ään ja R:ään. Jos auto ajaa ojaan, ilmoita tästä ajajalle ja käske hänen aloittaa uudestaan.

DATA-arvoituksia

Yksi kaikkein helpoimmista tavoista antaa tietokoneelle paljon tietoa on käyttää sanoja READ ja DATA. DATA-rivillä on sana- tai numeroluettelo ja READ käskee tietokonetta lukemaan näitä sanoja tai numeroita yhteen tai useampaan muuttujaan. Jos sinulla on ZX81 (Timex 1000) et voi ajaa tämän aukeaman ohjelmia, koska tietokoneesi ei käytä näitä käskyjä. Sivulla 34 on kuitenkin esitetty vaihtoehtoinen tapa tallentaa tietoa taulukon avulla.

DATAn lukeminen

Erota tiedot toisistaan pilkulla.

```
10 FOR I=1 TO 7
20 READ X,X$
30 PRINT X;" ";X$
40 NEXT I
50 DATA 13,KALAA,77,SAMMAKKOA,91,ETANAA
60 DATA 23,KISSAA,62,KOIRAA,2,HIIRTA,1,ROTTA
```

Joissain mikroissa datajonot pitää panna lainausmerkkeihin.

Aja tämä ohjelma, niin näet miten READ ja DATA toimivat. Sanan READ jälkeen on kaksi muuttujaa. Joka kerta, kun silmukka toistuu, tietokone tallentaa seuraavan data-alkioparin muuttujiin X ja X$.

Nimiluettelo

Pane riville 50 kysymysmerkin tilalle datojesi viimeinen nimi.

```
 10 PRINT "ANNA NIMI ";
 20 INPUT N$
 30 READ X$
 40 IF X$=N$ THEN GOTO 70
*50 IF X$="" ? "  THEN PRINT "NIMESI
 EI OLE LUETTELOSSA":STOP
 60 GOTO 30
 70 PRINT "OK,NIMESI ON LUETTELOSSA"
*80 DATA ?
```

Tässä ohjelmassa tietokone kysyy sinulta nimeäsi ja tarkistaa, onko se datana tallennetussa nimiluettelossa. Pane omia datojasi riville 80 (voit lisätä niin monta nimeä kuin haluat) ja pane sitten datojen viimeinen nimi riville 50, jotta tietokone lopettaa ohjelman luettuaan luettelon viimeisen nimen.

DATAn palautus

```
*10 FOR J= ? TO ?
*20 FOR I= ? TO ?
 30 READ N$
*40 IF LEFT$(N$,1)=CHR$( ? ) THEN PRINT N$
 50 NEXT I
 60 RESTORE
 70 NEXT J
 80 DATA VEERA,KSENOFON,SAKARI,HORATIUS,PEKKA,PASI,PERTTI
 90 DATA TAPSA,TIMO,KALLE,JUSSI,HARRI,YRJÖ,VAINAMOINEN
100 DATA KEKE,KIKKE,JOUKAHAINEN,SINUHE,ROKKA
```

Tee J-silmukasta aakkoston pituinen käyttäen merkkien koodinumeroita.

Tee I-silmukasta saman pituinen kuin data-alkioiden määrä.

Osaatko täydentää tämän ohjelman niin, että se panee nimet aakkosjärjestykseen? Nimet on annettu DATA-riveillä ja RESTORE käskee tietokoneen palata datoluettelon alkuun joka kerta, kun J-silmukka toistuu. Yritä lisätä puuttuvat muuttujat ja numerot.

Etsi virheet

1

```
10 FOR K=1 TO 6
20 READ N : PRINT N
30 NEXT K
40 DATA 01-232,22-36-41,341-2241/2
50 DATA 97-24-11,47-29-01,236-4013
```

Tässä ohjelmassa datariveille on tallennettu puhelinnumeroita. Löydätkö ohjelman virheet? Korjaa ne.

2

```
10 READ X
20 PRINT X
30 GOTO 10
40 DATA 1,461,892,66,1471,4462,1,3
50 DATA 53,80,241,90,371,825,33,13
```

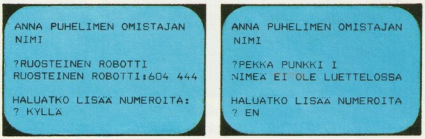
Jos et löydä tässä ohjelmassa olevaa virhettä, yritä ajaa se. Tietokoneen pitäisi antaa virheilmoitus, joka kertoo, mikä on vikana. Keksitkö helpon tavan korjata ohjelman kuntoon?

Jussin Baari

Ylhäällä vasemmalla on Jussin Baarin ruokalista. Käytä ruokalistan hintoja ja ruokia datana ja kirjoita ohjelma, joka kertoo, mihin ruokiin rahasi riittävät (oikealla on malli). Voit lisätä menuun uusia ruokia.

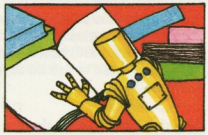
Puhelinluettelo

Alla on ohjeet puhelinluettelo-ohjelmaa varten. Oikealla näkyy, miltä ohjelman ajo näyttää ruudulla. Yritä kirjoittaa ohjelma.

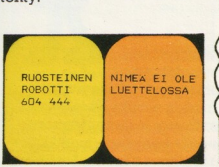


1. Laadi luettelo ystäviesi nimistä ja puhelinnumeroista DATA-riveille, kuten yllä on tehty.

2. Pane tietokone kysymään PRINT-sanalla puhelimen omistajan nimeä ja odottamaan vastaustasi sanalla INPUT.



3. Etsi sanalla READ nimi silmukan sisältä. Käytä eri muuttujia nimelle ja numerolle.



4. Printtaa nimi ja numero (jonka pitäisi olla seuraava data nimen jälkeen) ruudulle tai ilmoita, että nimeä ei ole luettelossa.

5. Pane tietokone kysymään, haluatko lisää numeroita ja odottamaan vastaustasi sanalla INPUT.

6. Vastauksesta riippuen palauta (RESTORE) data ja palaa alkuun tai lopeta ohjelma.

Taulukoiden käyttö

Dataa on kätevä tallentaa taulukkoon. Taulukko on datayksiköiden joukko, joka on yhteisen muuttujanimen alla. Jokaista taulukon yksikköä sanotaan alkioksi ja sen sijainnin ilmaisevaa numeroa sanotaan alaindeksiksi.

Numeeriset taulukot

Tässä on numeerinen taulukko nimeltä N. Siinä on kuusi alkiota. Tietokoneelle pitää kertoa, kuinka iso taulukko on, jotta se osaa varata riittävästi muistitilaa. Tämä tehdään sanalla DIM, jota seuraa taulukon nimi ja sen alkioiden lukumäärä. Tätä sanotaan taulukon ulottuvuuksien määrittelyksi.

```
*10 DIM
 20 FOR K=1 TO 6
 30 READ N(K)
 40 NEXT K
*50 DATA?
```

ZX81
(Timex 1000)

```
*10 DIM?
 20 FOR K=1 TO 6
 30 INPUT N(K)
 40 NEXT K
```

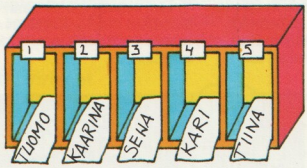
Dataa voi panna taulukkoon silmukalla ja sanoilla READ/DATA. Yritä täydentää vasemmalla oleva ohjelma niin, että se tallentaa kaikki sivun yläosassa olevan kuvan tiedot taulukkoon. Jos sinulla on ZX81 (Timex 1000) kokeile oikealla olevaa ohjelmaa, jossa taulukko täytetään INPUT-lauseilla.

```
N(1) IS 1066
N(2) IS 1216
N(3) IS 1485
N(4) IS 1603
N(5) IS 1665
N(6) IS 1959
```

```
1485  1216
1959  1485
1959  1665
1603  1603
1485  1216
```

Yritä seuraavaksi kirjoittaa ohjelma, joka printtaa taulukkoon tallennetut tiedot ruudulle. Käytä sanaa PRINT ja silmukkamuuttujaa taulukon alaindeksinä. Oikeanpuoleisessa ruudussa tietokone printtaa taulukon alkioita satunnaisesti käyttäen alaindeksinä satunnaislukua.

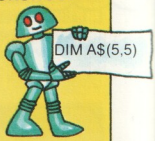
Merkkijonotaulukot



Tämä on merkkijonotaulukko (N$). Siinä on viisi nimeä eli viisi alkiota. Sinclair tietokoneita lukuun ottamatta jonotaulukkoa käsitellään aivan samalla tavalla kuin numeerista taulukkoa. Yritä kirjoittaa lyhyt ohjelma, joka tallentaa tämän taulukon datat ja printtaa ne ruudulle.

Sinclair (Timex) tietokoneet

```
TAULUKKO A$
1 KISSA
2 KALA
3 NORSU
4 ROTTA
5 AASI
```



Sinclair (Timex) tietokoneet panevat taulukossa jokaisen merkkijonon omalle rivilleen ja jokaisen merkin omaksi alkiokseen. Taulukon ulottuvuus määritellään siten, että tietokoneelle kerrotaan, montako jonoa (riviä) ja alkiota siinä on. Laske jonojen määrä ja pisimmän jonon merkkien määrä ja pane molemmat luvut sulkuihin, kuten yllä on tehty.

Kalenterilaskuri

```
* 10 ?
20 FOR K=1 TO 12
* 30 ?
40 NEXT K
50 PRINT "MONESKO KUUKAUSI ";
60 INPUT N
* 70 PRINT M$(?);"SSA ON ";
* 80 PRINT D(?);"PAIVAA"
* 90 DATA?
*100 DATA?
*110 DATA?
```

Määritä taulukon ulottuvuudet.

Lue (READ) data kahteen taulukkoon M$ ja D.

Lisää alaindeksit, joilla printataan oikeat datat taulukoista.

Pane näille DATA-riveille kunkin kuukauden nimi ja heti perään sen kuukauden päivien lukumäärä.

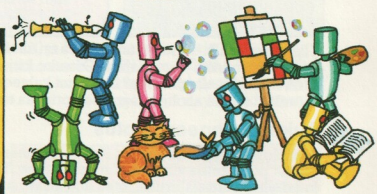
Osaatko kirjoittaa ohjelman valmiiksi niin, että kun annat kuukauden numeron tietokone printtaa kuukauden nimen ja kertoo, montako päivää siinä on? Ohjelman vieressä on ohjeita.

```
MIKA ON KUUKAUDEN
NIMI ? HEINAKUU

HEINAKUUSSA ON 31
PAIVAA
```

Muuta seuraavaksi ohjelmaa niin, että se kysyy kuukauden nimeä ja kertoo sitten, montako päivää siinä on. Etsi kuukauden nimi M$:sta silmukalla ja sanoilla IF/THEN. Käytä silmukkamuuttujaa alaindeksinä, jotta löydät oikean datayksikön D:stä.

Ajantappo-ohjelma

```
VALITSE LUKU 1 - 10
? 9
JOSPA SEISOISIT PAALLASI
SOPIIKO ? EI

VALITSE LUKU 1 - 10
? 5
JOSPA ANNAT KISSALLESI RUOKAA
SOPIIKO ? KYLLA
```



Tästä ohjelmasta voi olla paljon hyötyä, jos et keksi mitään tekemistä. Ohjelmassa valitset luvun ja tietokone kirjoittaa ruudulle toimintaehdotuksen.

Ohjelmaa varten tarvitset merkkijonotaulukon (I$), joka on täynnä ideoita. Pane (INPUT) valittu numero muuttujaan (N) ja laita tietokone valitsemaan yksi data-alkioista siten, että se käyttää N:ää I$:n alaindeksinä.

20 kysymystä

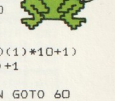
```
MONTAKO JALKAA ON
TUHATJALKAISELLA
? 1000
OIKEIN

MONTAKO PELAAJAA ON
JALKAPALLOJOUKKUEESSA
? 10
EI OLE, TONTTU
VASTAUS ON 11.
```

Tässä on kysymyspelin ruutunäyttö. Kirjoita sopiva ohjelma. Laadi ensin 20 kysymystä ja pane ne jonotaulukkoon. Pane vastaukset toiseen taulukkoon. Taulukoiden alaindeksien pitää yhdistää kysymykset oikeisiin vastauksiin.

Satunnaislukutaulukko

```
 1  *********
 2  ***************
 3  *********
 4  ************
 5  **********
 6  *******
 7  ***********
 8  *********
 9  ************
10  **********
```

Tietokone antaa jokaiselle A:n alkiolle alkuarvoksi 0.



```
10 LET N=0
20 DIM A(10)
30 FOR K=1 TO 10
40 LET A(K)=0
50 NEXT K
60 LET R=INT(RND(1)*10+1)
70 LET A(R)=A(R)+1
80 LET N=N+1
90 IF N<100 THEN GOTO 60
```

Tämä ohjelma tallentaa taulukon tietoja taulukkoon. Tietokone valitsee satunnaisesti 100 lukua väliltä 1–10. Taulukon (A) alkiot laskevat, montako kertaa kukin numero on valittu. Yritä laajentaa ohjelmaa niin, että tietokone printtaa yllä esitetyn taulukon, jossa jokaista valintakertaa merkitään tähdellä. Tee tämä kirjoittamalla silmukka, joka toistuu 10 kertaa ja pane sen sisään toinen silmukka. Sisemmän silmukan pitää printata taulukosta rivi kerrallaan.

Aliohjelmien kirjoittaminen

Aliohjelma on ohjelman osa, joka suorittaa jonkin osatehtävän ja jota yleensä käytetään useita kertoja ohjelman aikana. Sana GOSUB ja sitä seuraava numero, joka ilmoittaa aliohjelman ensimmäisen rivin, saa tietokoneen hyppäämään aliohjelmaan. Tietokone ajaa aliohjelmaa, kunnes se löytää sanan RETURN. Tämä lähettää sen takaisin pääohjelman riville, joka seuraa sanaa GOSUB.

Yritä ajaa tämä ohjelma, jotta näet, miten käskyt GOSUB ja RETURN toimivat.

```
10 PRINT "HEI ";
20 GOSUB 1010
30 PRINT "HEI TAAS ";
40 GOSUB 1010
50 PRINT "MIKAS TAMA ON ?"
60 GOSUB 2020
70 STOP
1010 PRINT "TAMA ON ALIOHJELMA"
1020 RETURN
2020 PRINT "TOINEN ALIOHJELMA"
2030 RETURN
```

Jäätelökysely

MELONI	16
BANAANI	11
INKIVÄÄRI	8
SUOLAKURKKU	1
PURUKUMI	18

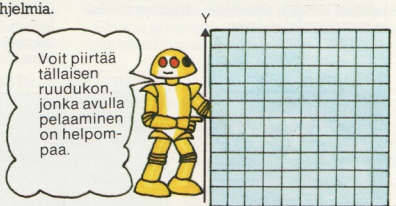
```
MELONI              ****************
BANAANI             ***********
INKIVÄÄRI           ********
SUOLAKURKKU         *
PURUKUMI            ******************
```

Yllä vasemmalla olevassa taulukossa on tulokset kyselystä, jossa etsittiin pidetyintä kuudesta uudesta jäätelölajista. Osaatko kirjoittaa ohjelman, joka tulostaa tiedot ruudulle siten kuin oikealla on tehty? Lue datat kahteen taulukkoon silmukalla ja lähetä tietokone silmukan sisällä aliohjelmaan, joka printtaa taulukosta rivin kerrallaan.

Sukellusveneen upotus

Alla olevan ohjelman nimi on Sukellusveneen upotus. Jossain kohden 10×10 kokoista ruutua piileskelee vihollisten sukellusvene. Tietokone valitsee sukellusveneen paikan kahdella satunnaisluvulla, jotka ilmoittavat sukellusveneen sijaintipaikan koordinaatit. Saat yrittää neljä kertaa löytää sukellusveneen arvaamalla koordinaatit X (vaakataso) ja Y (ylös). Jos arvaat väärin, tietokone hyppää aliohjelmaan, joka kertoo sinulle, mihin suuntaan sinun kannattaa seuraavalla arvauskerralla mennä. Ennen kuin voit ajaa ohjelman sinun pitää kirjoittaa aliohjelmia.

```
 10 CLS
 20 LET N=0
 30 LET X=INT (RND(1)*10+1)
 40 LET Y=INT (RND(1)*10+1)
 50 LET N=N+1
 60 PRINT "ARVAUS ";N;" ";
 70 INPUT A,B
 80 IF A=X AND B=Y THEN GOTO 170
 90 GOSUB 200
100 PRINT
110 IF N<=4 THEN GOTO 50
120 PRINT "ARVAUSKERRAT LOPPU"
130 PRINT "HALUATKO UUDEN PELIN"
140 INPUT R$
150 IF LEFT$(R$,1)="K" THEN GOTO 10
160 STOP
170 PRINT "OSUIT SUKELLUSVENEESEEN ";N;"
    ARVAUKSELLA"
180 GOTO 130
```



Tarvitset useita IF/THEN-rivejä, joilla verrataan arvaustasi (A,B) sukellusveneen sijaintiin (X,Y) ja printataan ruudulle viesti. Jos esim. B on pienempi kuin Y sinun pitää mennä pohjoiseen, jos A on pienempi kuin X, mene itään jne.

Ohjelman ajonäytteitä

```
ARVAUS 1 ?3,2
AMMUIT OHI
MENE POHJOISEEN ITAAN
```

```
ARVAUS 2 ?3,3
AMMUIT OHI
MENE ITAAN
```

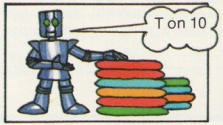
```
ARVAUS 3 ?5,3
AMMUIT OHI
MENE LANTEEN
```

Yksikätinen rosvo

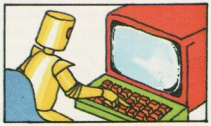
Osaatko kirjoittaa ohjelman, joka panee tietokoneesi toimimaan kuten peliautomaatti, jota sanotaan joskus yksikätiseksi rosvoksi? Oikealla on esimerkki ohjelman ajosta. Alussa sinulla on 10 pelimerkkiä. Jokainen pelikerta maksaa yhden pelimerkin. Paina mitä tahansa näppäintä aloittaaksesi. Tietokone printtaa kolmen satunnaisesti valitun "hedelmän" nimet. Noudata alla annettuja ohjeita, niin ohjelman kirjoittaminen käy helpommin.

```
SINULLA ON 10 PELIMERKKIA
PAINA JOTAIN NAPPAINTA JOS
HALUAT ALOITTAA:P
KELLO KELLO SITRUUNA

2 SAMAA
VOITAT 2 PELIMERKKIA
NYT SINULLA ON 11 PELIMERKKIA
PAINA JOTAIN NAPPAINTA JOS
HALUAT ALOITTAA:B
```

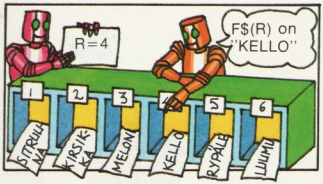
1. Määritä ensin taulukon F$ ulottuvuudet. Panet taulukkoon kuuden "hedelmän" nimet (sitruuna, kirsikka, luumu, meloni, rypäle ja kello). Tallenna nämä datat F$:iin sanoilla READ/DATA (tai INPUT).



2. Tyhjennä näyttö ja pane muuttuja T laskemaan pelimerkkien määrää. Anna T:lle alkuarvoksi 10 ja kerro pelaajalle, montako merkkiä hänellä on.



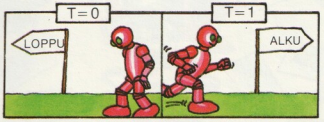
3. Pane samalla INKEY$ tietokone odottamaan, kunnes pelaaja haluaa aloittaa painamalla jotain näppäintä. Vähennä T:stä 1 ((yksi peli maksaa yhden merkin) ja tyhjennä näyttö uudestaan.



4. Pane tietokone etsimään satunnaisluku väliltä 1–6 ja tallentamaan se muuttujaan R. Käytä R:ää F$:n alaindeksinä hedelmän valinnassa ja tallenna hedelmä muuttujaan A$. Toista nämä ohjeet vielä kahdesti, jotta mikro valitsee vielä kaksi hedelmää. Tallenna ne muuttujiin B$ ja C$.

```
3 KIRSIKKAA:  VOITIT KOKO POTIN

3 SAMAA:      VOITIT 5 PELIMERKKIÄ

2 SAMAA:      VOITIT 2 PELIMERKKIÄ
```

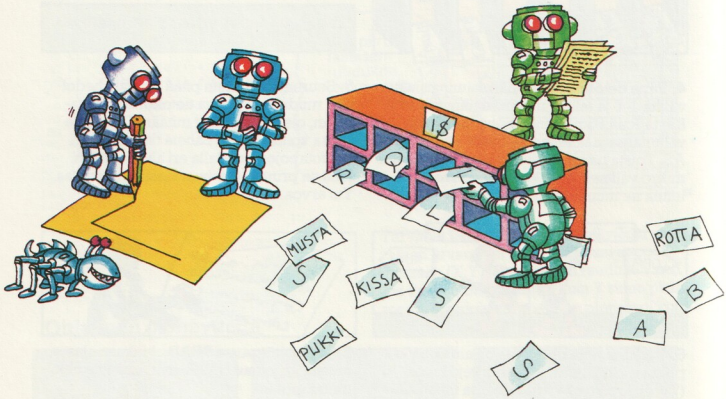
5. Seuraavaksi sinun pitää printata hedelmät ruudulle ja panna tietokone tarkistamaan, oletko voittanut mitään. On kolme tapaa voittaa (yllä). Lähetä tietokone eri voittotapojen kohdalla eri aliohjelmaan, jolla se printtaa sopivan viestin ja muuttaa T:n arvoa.



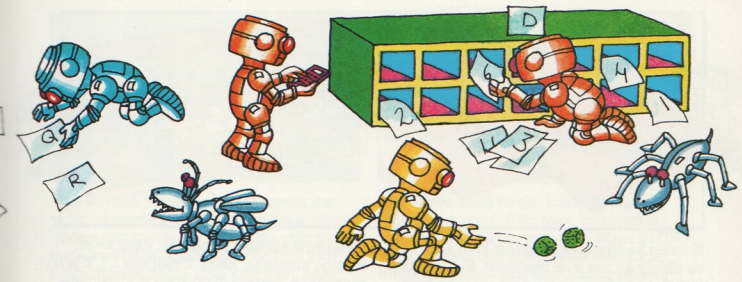
6. Tarkista, että pelaajalla on ainakin yksi pelimerkki jäljellä. Jos merkkejä ei ole (T=0), lopeta ohjelma. Muussa tapauksessa lähetä tietokone takaisin pelin alkuun.

7. Jos käytät sanaa READ ohjelman alussa, muista kirjoittaa DATA-rivit. Lisää ne ohjelman loppuun.

Tutkimme BASIC-ohjelmia



Hyvä tapa opetella BASICia on tutkia toisten kirjoittamia ohjelmia. Kirjamme seuraavassa osassa on ohjelmia ja tarkkoja selityksiä niiden toiminnasta. Ajamalla ja tutkimalla näitä ohjelmia näet, miten BASICilla voi tehdä vaikeitakin asioita, esimerkiksi perustaa tietokannan, piirtää kuvioita ruudulle ja lajitella tietoja. Ohjelmat on kirjoitutettu yleis-BASICilla, joten niitä voi pienin muutoksin ajaa useimmilla kotitietokoneilla. Jos et ole varma, mitkä käskyt pitää muuttaa omaa mikroasi varten, tarkista asia sivun 140 muunnostaulukosta. Sivuilla 141–142 on lisäksi tarkat ohjeet siitä, millaisia muutoksia pitää tehdä Sinclair (Timex) tietokoneille.

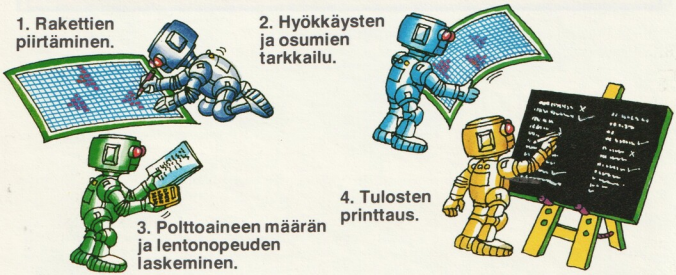


Neuvoja ohjelmien tutkijalle

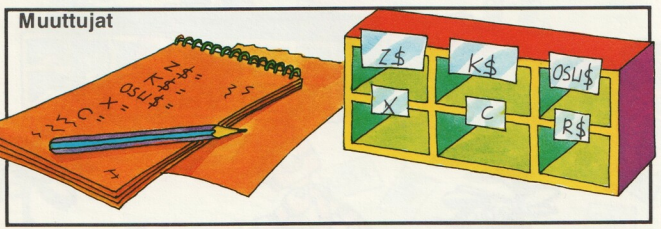
Ensi näkemältä jotkut seuraavista ohjelmista näyttävät hirvittävän vaikeilta. Mutta vaikea ohjelma on itse asiassa vain pitkä lista helppoja BASIC-käskyjä, jotka on kirjoitettu oikeaan järjestykseen. Tällä aukeamalla on perusneuvoja siitä, miten sinun kannattaa tutkia ohjelmia. Kun olet ymmärtänyt, miten ohjelma toimii, yritä muuttaa sitä niin, että se toimii hieman eri lailla kuin aiemmin.

Ohjelman tutkiminen

Yleensä ohjelmassa on useita eri osia (niitä sanotaan toisinaan rutiineiksi tai moduleiksi), jotka tekevät eri tehtäviä. Esimerkiksi rakettiohjelmassa yksi ohjelman osa piirtää ruudulle raketteja ja toiset osat panevat merkille hyökkäykset ja osumat, polttoaineen määrän ja lentonopeuden ja printtaavat lopputuloksen ruudulle.

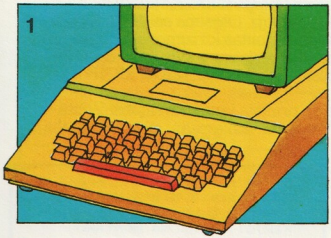


Ensimmäinen askel ohjelman tutkimisessa on etsiä sen eri osat ja osien tehtävät. Näin saat yleiskäsityksen ohjelman toiminnasta. Etsi mahdolliset aliohjelmat, joilla tehdään tiettyjä toimintoja, sekä isot aukot rivinumeroissa. Seuraavasta sadasta tai tuhannesta alkavat rivit ovat usein merkki siitä, että niiden kohdalta alkaa uusi ohjelmaosa. Joskus ohjelman eri osat on merkitty REM-lauseilla.

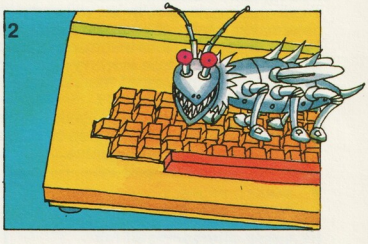


Kenties kaikkein vaikeimmin ymmärrettävä ohjelman osa ovat muuttujat. Ennen kuin kirjoitat ohjelman tietokoneeseen, on hyvä selvittää kunkin muuttujan tehtävä ja merkitä se paperille. Tiettyjä muuttujia käytetään usein samaan tehtävään, joten ne on helppo tunnistaa. Esim. kirjaimilla I, J, K ja L merkitään yleensä silmukoita ja kirjaimella Z tai Z$ dataa, jota käytetään vain hetken.

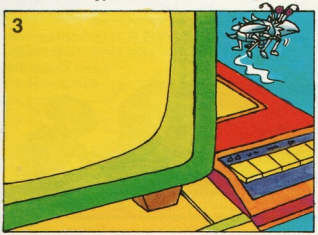
Ohjelmien ajaminen ja korjaaminen



Kun olet selvittänyt muuttujien tehtävän, kirjoita ohjelma tietokoneeseen. Koska tämän kirjan ohjelmat on laadittu yleis-BASICilla, voit joutua muuttamaan joitain BASIC-käskyjä omaan mikroosi sopiviksi.

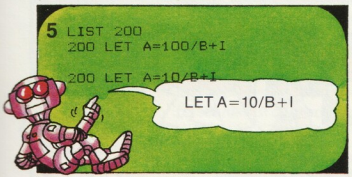


Yritä seuraavaksi ajaa ohjelma. Ohjelmassa on todennäköisesti virheitä eli pukkeja, joten sinun kannattaa listata ohjelma ruudulla ja etsiä kirjoitusvirheitä tai käskyjä, joita mikrosi ei hyväksy.



Kun olet löytänyt kaikki virheet, aja ohjelma muutaman kerran nähdäksesi, miten se toimii. Ohjelma kannattaa tässä vaiheessa tallentaa kasetille tai levylle. Näin et enää koskaan joudu kirjoittamaan sitä uudestaan.

Tutki seuraavaksi ohjelmalistausta uudestaan. Yritä selvittää kunkin rivin tehtävä. Huomaa lyhyet rutiinit, joita voit joskus käyttää omissakin ohjelmissasi.



Voit käyttää tietokonetta hyväksesi, kun yrität ymmärtää ohjelman toimintaa. Muuta yhden muuttujan arvoa ja katso, miten tämä vaikuttaa ohjelmaan. Tee vain yksi pieni muutos kerrallaan, jotta näet, mikä sen vaikutus on. Muista panna ohjelmaan takaisin oikeat luvut, kun olet lopettanut kokeilusi.

Voit myös lisätä rivejä, jotka printtaavat muuttujien arvot. Näin näet, miten arvot muuttuvat ohjelman kuluessa. Myös STOP-käskyistä voi olla apua. Niillä voit tutkia ohjelmaa pala kerrallaan. Muista poistaa pysäytyskäskyt tutkimisen jälkeen. Joissain mikroissa on käsky CONTINUE (jatka), jota voit käyttää STOPin jälkeen.

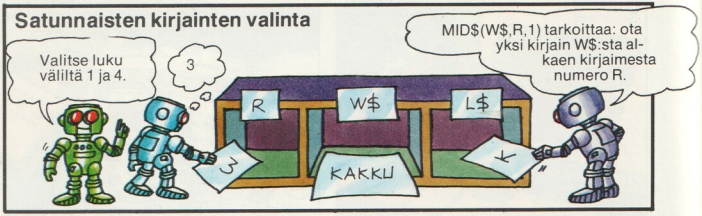
Sanasilppupeli

Tästä ohjelmasta näet, miten voit yhdistää aivan helppoja BASIC-käskyjä ohjelmaksi, joka tekee sangen mutkikkaita toimintoja. Ohjelma on sanasilppupeli. Tietokone pyytää sinua antamaan jonkin sanan, kirjoittaa kirjaimet satunnaisessa järjestyksessä ruudulle ja kysyy sinulta, montako kertaa antamasi sana esiintyy ruudulla. Ohjelma käyttää merkkijonojen käsittelyyn käskyjä MID$, RIGHT$ ja LEN sekä satunnaislukuja.*

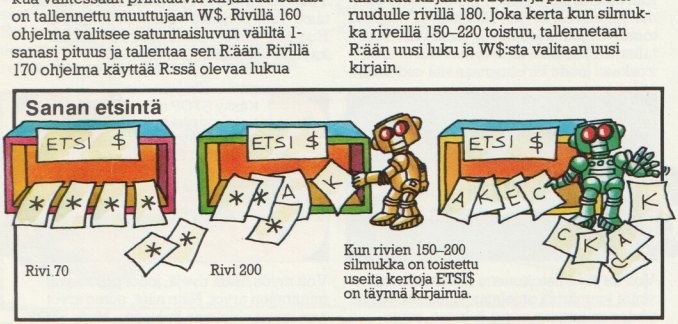
Ohjelmalla on kaksi päätehtävää. Toinen on printata kirjaimet satunnaisessa järjestyksessä ruudulle ja toinen on panna tietokone laskemaan, montako kertaa sana esiintyy oikein kirjoitettuna.

```
SANASILPPUPELI
ANNA LYHYT SANA
?KAKKU
YRITÄ LÖYTÄÄ
SANA RUUDULTA
ALOITA PAINAMALLA RETURN
```

```
K K U A K A U A K K K A K U A
A U K A K K U A A U K U A K A
K K U U U K A K A K U A U A K
KIRJOITA LUKUNA MONTAKO
KERTAA SANA ESIINTYY RUUDULLA
?1
VÄÄRIN!
SANA ESIINTYI 2 KERTAA
```



Ohjelma käyttää sanaa MID$ ja satunnaislukua valitessaan printtaavia kirjaimia. Sanasi on tallennettu muuttujaan W$. Rivillä 160 ohjelma valitsee satunnaisluvun väliltä 1-sanasi pituus ja tallentaa sen R:ään. Rivillä 170 ohjelma käyttää R:ssä olevaa lukua valitessaan jonkin kirjaimen W$:sta. Se tallentaa kirjaimen L$:iin ja printtaa sen ruudulle rivillä 180. Joka kerta kun silmukka riveillä 150–220 toistuu, tallennetaan R:ään uusi luku ja W$:sta valitaan uusi kirjain.



Ohjelman alussa tietokone varaa muistitilan nimeltä ETSI$ ja täyttää sen niin monella tähdellä kuin antamassasi sanassa on kirjaimia. Aina, kun se valitsee uuden satunnaiskirjaimen, se heittää menemään muuttujan ETSI$ ensimmäisen merkin ja lisää satunnaiskirjaimen jonon loppuun (rivi 200). Rivillä 210 se vertaa ETSI$:a W$:iin ja jos kirjaimet ovat samassa järjestyksessä se lisää N:ään 1.

*Ohjeet ohjelman muuttamiseksi Sinclair (Timex) tietokoneille on annettu sivulla 141.

Sanasilppupeli

```
10 CLS
20 PRINT "SANASILPPUPELI":PRINT
30 LET ETSI$=""
40 LET N=0
45 PRINT "ANNA LYHYT SANA"
50 INPUT W$
60 FOR I=1 TO LEN(W$)
70 LET ETSI$=ETSI$+"*"
80 NEXT I
90 PRINT
100 PRINT "YRITÄ LÖYTÄÄ"
110 PRINT "SANA RUUDULTA"

120 INPUT "ALOITA PAINAMALLA RETURN";Z$
130 CLS
140 REM SATUNNAISKIRJAINTEN VALINTA
150 FOR I=1 TO 50*LEN(W$)
160 LET R=INT(RND(1)*LEN(W$)+1)
170 LET L$=MID$(W$,R,1)
180 PRINT L$+" ";
190 REM SANAN ETSINTA
200 LET ETSI$=RIGHT$(ETSI$,LEN(W$)-1)+L$
210 IF ETSI$=W$ THEN LET N=N+1
220 NEXT I
230 FOR I=1 TO 1000
240 REM ALA TEE MITÄÄN
250 NEXT I
260 CLS
265 PRINT "KIRJOITA LUKUNA MONTAKO"
270 PRINT "KERTAA SANA ESIINTYY RUUDULLA"
275 INPUT G
280 PRINT
290 IF G=N THEN PRINT "OIKEIN!"
300 IF G<>N THEN PRINT "VÄÄRIN!"
310 PRINT "SANA ESIINTYI ";N;" KERTAA"
```

Käytä oman mikrosi käskyä ruudun tyhjentämiseen.

Tällä rivillä on kaksi käskyä, joita erottaa toisistaan kaksoispiste.

Tyhjiä muuttujia myöhempää käyttöä varten.

Pyytää sanaa ja panee sen W$:iin.

Silmukka, joka ajetaan yhtä monesti kuin sanassasi on kirjaimia, so. LEN(W$). Aina, kun silmukka toistuu, lisätään ETSI$:iin yksi *.

Rivi 120 panee mikron odottamaan, että kirjoitat jotain. Yleensä voit painaa RETURN, mutta Oricissa paina ensin jotain näppäintä ja paina vasta sitten RETURN.

Tämä on hyvä tapa panna mikro odottamaan, kunnes olet valmis.

Valmistaa riveille 150–220 silmukan, joka ajetaan 50 × sanasi kirjainten määrä.

Valitsee satunnaisluvun väliltä 1-sanan pituus ja panee sen R:ään.

Valitsee R:ssä olevan luvun perusteella kirjaimen W$:sta ja tallentaa sen L$:iin.

Käytä mikrosi RND-käskyä.

Printtaa L$:ssa olevan kirjaimen ja panee sen perään välilyönnin. Puolipiste jättää mikron samalle riville eri kirjaimia varten.

Suomennos: ota LEN(W$)–1 kirjainta ETSI$:n oikeasta päästä, lisää L$:ssa oleva kirjain ja pane tulos takaisin ETSI$:iin.

N laskee, montako kertaa sana esiintyy oikein kirjoitettuna.

Tämä on hyvä tapa panna mikro etsimään tiettyä sanaa datajoukosta. Voit käyttää samaa rutiinia muissakin ohjelmissa. Tarvitset lisäksi rivien 60 ja 80 silmukan.

Tämä on viivesilmukka. Se ei toteuta käskyjä vaan saa mikron odottamaan muutaman sekunnin panemalla sen laskemaan yhdestä tuhanteen.

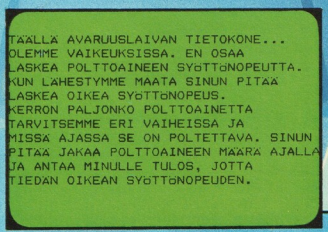
Toiset mikrot ovat nopeampia kuin toiset, joten muuta tämä luku mikrollesi sopivaksi. Isompi luku rivillä 230 pidentää taukoa.

Tallentaa arvauksesi muuttujaan G.

Vertaa G:tä N:ään (muuttujaan, jolla mikro laskee oikeiden sanojen määrän).

Hätälasku avaruudesta

Tämä ohjelma on avaruuspeli, joka panee myös päässälaskutaitosi koetukselle. Ohjelmassa on esimerkkejä siitä, miten voit käyttää silmukoita ja satunnaislukuja. Lisäksi siinä on ruudulla näkyviä erikoistehosteita, joita voit myöhemmin käyttää omissa ohjelmissasi. Ohjelma on aika pitkä, mutta suurin osa riveistä on ruudulle kirjoittavia PRINT-lauseita.



Ohjelma

```
100 CLS
110 FOR I=1 TO 10
115 PRINT "***VAROITUS***";
118 PRINT "***HÄLYTYS****";
120 FOR J=1 TO 10
125 REM ALA TEE MITÄÄN
130 NEXT J
135 NEXT I
140 CLS

150 FOR I=1 TO 20
155 PRINT "***PIIRIT VAURIOITUNEET***";
160 FOR J=1 TO 10
165 REM ALA TEE MITÄÄN
170 NEXT J
180 NEXT I
190 CLS

200 PRINT "TÄÄLLÄ AVARUUSLAIVAN TIETOKONE..."
210 PRINT
220 PRINT "OLEMME VAIKEUKSISSA. EN OSAA
LASKEA POLTTOAINEEN SYÖTTÖNOPEUTTA."
230 PRINT
240 PRINT "KUN LÄHESTYMME MAATA SINUN
PITÄÄ LASKEA OIKEA SYÖTTÖNOPEUS."


250 PRINT
260 PRINT "ILMOITAN PALJONKO POLTTOAINETTA
TARVITSEMME ERI VAIHEISSA JA MISSÄ AJASSA
SE ON POLTETTAVA."
270 PRINT
280 PRINT "SINUN PITÄÄ JAKAA POLTTOAINEEN
MÄÄRÄ AJALLA JA ANTAA MINULLE TULOS,
JOTTA TIEDÄN OIKEAN SYÖTTÖNOPEUDEN."
290 PRINT
300 PRINT "ESIMERKKI:"

310 PRINT "----------"
320 PRINT "AINE=24"
330 PRINT "AIKA=6"
345 PRINT

350 INPUT "JAA POLTTOAINE AJALLA JA
NAPPAILE VASTAUS NOPEASTI SISÄÄN ";VASTAUS
360 PRINT
370 JOS VASTAUS<>4 THEN PRINT "EI. YRITÄ
UUDESTAAN, HENKESI ON
VAAKALAUDALLA": GOTO 350
380 CLS
```

Käytä mikrosi käskyä ruudun tyhjentämiseksi.

Rivit 110–135 ovat sisäkkäisiä silmukoita. Aina, kun I-silmukka toistuu, printataan rivien 115–118 varoitus ja sitten J-silmukka toistuu 10 kertaa. J-silmukka on viivesilmukka, joka pysäyttää tietokoneen hetkeksi, jotta ehdit lukea varoituksen.

Rivit 150–180 toimivat kuten rivit 110–135.

Ohjelman seuraava osa printtaa peliohjeet ruudulle. Jos lauseet ovat liian pitkiä omalle ruudullesi, lisää PRINT-lauseiden määrää.

Pelkkä PRINT jättää tyhjän rivin.

Alleviivaa edellisen rivin sanat.

Vastauksesi tallennetaan muuttujaan VASTAUS.

GOTO toteutuu vain jos VASTAUS ei ole 4. (Jos mikrosi ei hyväksy kahta käskyä samalla rivillä, toista IF/THEN uudella rivillä sanan GOTO kanssa.)

```
390 PRINT "OK. NYT SINUN PITAA VASTATA
JOKA KERRALLA OIKEIN"
395 PRINT "TAI ALUKSEEN TULEE VAURIOITA"
400 PRINT "JOS TEET YLI 2 VIRHETTA"
405 PRINT "ME TUHOUDUMME"
410 PRINT
420 INPUT "PAINA RETURN KUN OLET
VALMIS";Z$
430 CLS
440 PRINT "VALMIINA - ALOITAMME!!!"
450 LET VAURIO=0

460 FOR AINE=720 TO 120 STEP-120

470 LET T=INT(RND(1)*5+2)

480 PRINT
490 PRINT "AINE=";AINE
500 PRINT "AIKA=";T
510 PRINT

520 LET R=AINE/T
530 INPUT "ANNA SYöTTöNOPEUS
NYT";VASTAUS
540 IF VASTAUS=R THEN GOTO 600

550 LET VAURIO=VAURIO+1
560 PRINT
570 PRINT "**VAURIO**"
580 PRINT

590 IF VAURIO>2 THEN GOTO 640
600 NEXT AINE
610 CLS
620 PRINT "ONNITTELUT - LASKIT AIVAN
OIKEIN. PAASIT ONNELLISESTI MAAN
PINNALLE!!!"
630 GOTO 720
640 CLS

650 FOR I=1 TO 20
660 PRINT "*";
670 FOR J=1 TO INT(RND(1)*I+50)
680 PRINT "    ";
690 NEXT J
700 NEXT I

710 PRINT: PRINT "ALUS TUHOUTUI"
720 END
```

Odottaa, kunnes painat RETURN. (Oricissa paina ensin jotain näppäintä ja sitten RETURN.)

Muuttuja nimeltä VAURIO.

AINE on muuttuja, joka laskee silmukat. Sitä käytetään myös laskutoimituksissa. Rivillä 460 AINE = 720 jonka jälkeen se vähenee 120:lla joka kerta kun silmukka toistuu. Nämä luvut valittiin siksi, että rivin 520 vastaus olisi aina kokonaisluku.

Antaa satunnaislukujen väliltä 2–6, joilla AINEen jakaminen menee tasan.

Tietokone jakaa AINEen T:n arvolla ja tallentaa vastauksen R:ään. Jos annat oikean vastauksen tietokone menee riville 600.

Muuttuja VAURIO laskee virheesi.

Jos teet yli kaksi virhettä, tietokone hyppää silmukasta riville 640.

Printtaa tämän rivin vain jos teet alle kaksi virhettä.

Nämä sisäkkäiset silmukat printtaavat ruudulle satunnaismuotoisen tähtikuvion. Aina kun I-silmukka toistuu tietokone printtaa tähden ja J-silmukka jättää satunnaisen tyhjän välin.

Maailmancupin jalkapallon tietokanta

Tietokanta on tietokoneeseen tallennettu suuri tietokokonaisuus. Tiedot on siinä järjestetty siten, että tietokone voi yhdistellä ja vertailla niitä monin eri tavoin. Näin tietokannan käyttäjä saa nopeasti tarvitsemansa tiedot.

Seuraavilla sivuilla on ohjelma, jolla laaditaan maailmancupin jalkapallon tietokanta. Tämä on sangen pieni tietokanta. Sen avulla voi kuitenkin selvittää, mikä joukkue voitti cupin tiettynä vuotena (alkaen vuodesta 1930) tai minä vuosina tietty joukkue voitti cupin. Ohjelman lopussa on ohjeita siitä, miten tietokantaa voi muuttaa toisenlaisia tietoja varten, esimerkiksi lehtijuttuja tai luontoaiheisia tietoja varten.

Tietokantaohjelmassa on kolme pääosaa. Tiedot pitää tallentaa, niitä pitää voida hakea ja lisäksi tarvitaan niin sanottu valikko. Valikko on luettelo, josta näkyy, mitä ohjelma osaa tehdä ja josta käyttäjä voi valita haluamansa toiminnon. Ohjelman pitäisi lisäksi olla käyttäjäystävällinen eli sen pitää antaa käyttäjälle selvät toimintaohjeet eikä se saa romahtaa eli mennä rikki, vaikka käyttäjä tekisi virheitä.

Esimerkkejä tietokannan ajosta

```
ANNA JOUKKUEEN NIMI TAI KIRJOITA
VALIKKO JOS HALUAT NAHDA
LUETTELON UUDESTAAN LANSI-SAKSA

LANSI-SAKSA
VOITTI CUPIN VUONNA 1954
OLI FINALISTI VUONNA 1966
VOITTI CUPIN VUONNA 1974
OLI FINALISTI VUONNA 1982

PAINA RETURN JOS HALUAT VALIKON
```

```
ANNA VUOSI TAI KIRJOITA VALIKKO
JOS HALUAT NAHDA LUETTELON
UUDESTAAN 1938

VUONNA 1938
ITALIA VOITTI CUPIN

PAINA RETURN
```

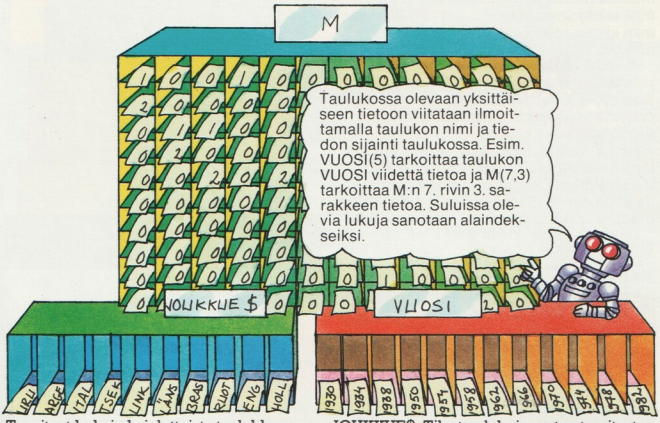
Tietojen tallentaminen

	1930	1934	1938	1950	1954	1958	1962	1966	1970	1974	1978	1982
URUGUAY	1	0	0	1	0	0	0	0	0	0	0	0
ARGENTIINA	2	0	0	0	0	0	0	0	0	0	1	0
ITALIA	0	1	1	0	0	0	0	0	2	0	0	1
TSEKKOSLOVAKIA	0	2	0	0	0	0	2	0	0	0	0	0
UNKARI	0	0	2	0	2	0	0	0	0	0	0	0
LÄNSI-SAKSA	0	0	0	0	1	0	0	2	0	1	0	2
BRASILIA	0	0	0	0	0	1	1	0	1	0	0	0
RUOTSI	0	0	0	0	0	2	0	0	0	0	0	0
ENGLANTI	0	0	0	0	0	0	0	1	0	0	0	0
HOLLANTI	0	0	0	0	0	0	0	0	0	2	2	0

Joukkueita ja vuosia varten ohjelma käyttää matriisia eli taulukkoa etsien joukkueen ja vuoden samalla tavalla kuin sinäkin etsisit. Taulukossa 1 tarkoittaa, että joukkue voitti cupin ja 2 tarkoittaa, että se pääsi loppuotteluihin (oli finalisti). Rivien (vaakataso) ja sarakkeiden (pystysuunta) risteyksistä näet, mikä joukkue voitti minäkin vuonna. Ohjelma etsii nämä tiedot automaattisesti. Jos tietoa on paljon, on ohjelma tietenkin paljon nopeampi kuin taulukkoa tutkiva ihminen.

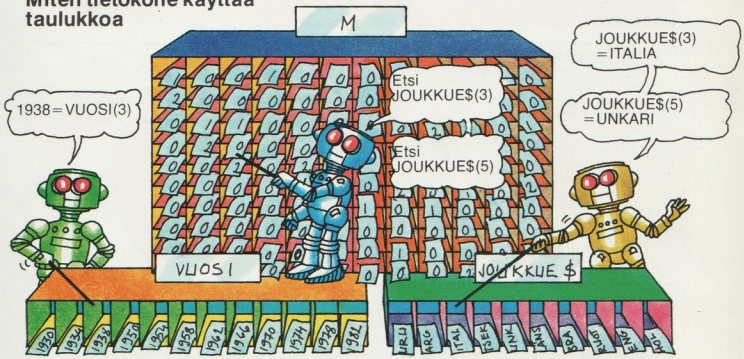
Taulukon laatiminen

Viereisen sivun taulukko on ihan helppo siirtää tietokoneeseen. Tietokoneessa taulukko on muuttuja, johon voi panna useita yksittäisiä tietoja.



Tarvitset kaksi yksiulotteista taulukkoa. Toisessa pitää olla 12 lokeroa vuosia varten ja toisessa 10 lokeroa joukkueita varten. Näiden taulukoiden nimet ovat VUOSI ja JOUKKUE$. Tilaston lukuja varten tarvitset kaksiulotteisen taulukon, jossa on 10 riviä ja 12 saraketta. Tämän nimi on ohjelmassa M.

Miten tietokone käyttää taulukkoa



Jos haluat tietää, mikä joukkue voitti maailmancupin vuonna 1938, tietokone etsii vuoden 1938 VUOSI-taulukosta ja toteaa, että se on lokerossa 3. Sitten se käy läpi taulukon sarakkeen 3 ja löydettyään 1:n tai 2:n se panee merkille rivinumeron ja etsii tämän numeron avulla joukkueen nimen JOUKKUE$:sta.

Tietokantaohjelma

Ohjelmassa on seitsemän pääosaa. Jokaisen osan hoitaa erillinen aliohjelma. Ensimmäisillä riveillä tietokoneelle kerrotaan, mitä aliohjelmaa sen pitää käyttää. Aliohjelman ajettuaan se palaa ohjelman alkuriveille.

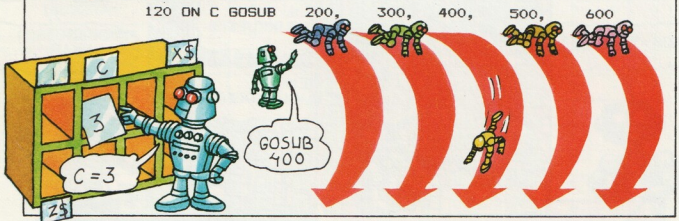
Riveillä 200 ja 300 alkavat aliohjelmat printtaavat joukkueet ja vuodet ruudulle. Rivit 400–500 selvittävät minä vuonna tietty joukkue voitti cupin. Rivit 500–600 taas selvittävät, mikä joukkue voitti tiettynä vuotena. Kaikki datat on pantu ohjelman loppupäähän, niiden jälkeen tulee tosin vielä valikko. Yleensä on parasta panna datat ohjelman loppupuolelle ja toistuvat osat ohjelman alkuun.

Valikko

```
                VALIKKO
                *******

JOUKKUELUETTELO - 1
VUOSILUETTELO - 2
ANNA JOUKKUEEN NIMI - 3
ANNA VUOSI - 4
LOPETUS - 5
KIRJOITA VALINTASI NUMERO
JA PAINA RETURN
```

Valikko on ohjelman osa, joka kertoo sinulle, mitä ohjelma osaa tehdä ja miten sitä käytetään. Tässä ohjelmassa valitset haluamasi toiminnon kirjoittamalla sitä vastaavan numeron. Numero tallennetaan muuttujaan C ja tietokone hakee tällä numerolla oikean aliohjelman, joka toteuttaa valitsemasi toiminnon.

Aliohjelmien kutsuminen



Ohjelmarivi 120 ohjaa aliohjelmien käyttöä. Kirjain C on muuttuja, jossa on valikosta valitsemasi numero. Tietokone käyttää C:ssä olevaa numeroa päättäessään, mihin aliohjelmaan se menee. Jos C = 1 se menee ensimmäiseen rivillä 120 ilmoitettuun aliohjelmaan, joka alkaa riviltä 200. Jos C = 2 se menee toiseen aliohjelmaan, so. riville 300. Jos C = 3 se menee kolmanteen jne. ON GOSUB on hyvä BASIC-käsky. Sillä saa tietokoneen menemään eri aliohjelmiin riippuen jonkin ehdon täyttämisestä. Jos mikrossasi ei ole käskyä ON, käytä useita IF/THEN-lauseita, esim. IF C = 1 THEN GOSUB 200.

Muuttujien tehtävät

VUOSI	JOUKKUE$	M
Taulukko, jossa on vuodet.	Taulukko, jossa on joukkueiden nimet.	Kaksiulotteinen taulukko, jossa on kaikki numerodata.
C	**Z$**	**X$**
Muuttuja, johon pannaan valikosta tekemäsi valinta.	Joukkueen nimi tai vuosi, jonka näppäilet koneeseen.	Väliaikaista dataa.

Käytä vain ensimmäisiä kirjaimia, jos mikrosi ei hyväksy sanoja taulukon nimeksi.

Ohjelma ★

```
10 DIM JOUKKUE$(10): DIM VUOSI(12): DIM M(10,12)
```

Kertoo tietokoneelle, paljonko tilaa sen tulee varata taulukoille.

```
100 GOSUB 1000: REM LUE DATAT
```

Kun ajat ohjelman, se menee ensiksi aliohjelmaan riville 1000 lukeakseen sieltä datat.

```
110 GOSUB 2000: REM PRINTTAA VALIKKO
```

Seuraavaksi se menee riville 2000 ja printtaa valikon ruudulle.

Listauksen vasemmalla puolen olevat hakasulut jakavat ohjelman osiin.

```
120 ON C GOSUB 200,300,400,500,600
130 GOTO 110
```

Tämä lähettää tietokoneen oikeaan aliohjelmaan sen mukaan, minkä toiminnon valitset valikosta. Aliohjelman jälkeen tietokone palaa riville 130, joka lähettää sen takaisin riville 110 printtaamaan valikon uudestaan.

```
200 REM ALIOHJELMA PRINTTAA JOUKKUEET
210 CLS

220 PRINT "JOUKKUELUETTELO"
225 PRINT "----------"

230 FOR I=1 TO 10: PRINT JOUKKUE$(I)
235 NEXT I
237 PRINT

240 INPUT "PAINA RETURN JOS HALUAT
VALIKON";X$
250 RETURN
```

Alleviivaa sanan JOUKKOLUETTELO.

Silmukka, joka printtaa joukkueiden nimet. Aina, kun silmukka toistuu I kasvaa 1:llä ja tietokone printtaa seuraavan nimen JOUKKUE$:sta.

Panee tietokoneen odottamaan, kunnes painat RETURN ennen kuin se siirtyy seuraavalle rivil-le. (Oricissa pitää ensin painaa jotain näppäintä.)

Takaisin riville 130.

```
300 REM ALIOHJELMA PRINTTAA VUOSILUETTELON
310 CLS
320 PRINT "VUOSILUETTELO": PRINT "----------"

330 FOR I=1 TO 12: PRINT VUOSI(I): NEXT I
335 PRINT
340 PRINT "VUOSINA 1942 JA 1946 EI PELATTU"
345 PRINT
350 INPUT "PAINA RETURN JOS HALUAT
VALIKON";X$
360 RETURN
```

Silmukka, joka printtaa vuodet.

Takaisin riville 130.

*Sivulla 141 on ohjeet tämän ohjelman muuttamiseksi Sinclair (Timex) tietokoneille.

```
400 REM ALIOHJELMA JOUKKUEEN SYÖTTÖ
405 CLS

410 INPUT "ANNA JOUKKUEEN NIMI TAI KIRJOITA
VALIKKO JOS HALUAT NÄHDÄ LUETTELON
UUDESTAAN ";Z$

415 IF Z$="VALIKKO" THEN RETURN

420 FOR I=1 TO 10
425 IF Z$=JOUKKUE$(I) THEN GOTO 440
430 NEXT I

435 PRINT: PRINT "JOUKKUETTA EI LÖYDY -
YRITÄ UUDESTAAN": PRINT: GOTO 410

440 PRINT
445 PRINT Z$: PRINT
```

Joukkueen nimi tai sana "VALIKKO" tallennetaan Z$:iin.

Jos Z$="VALIKKO" tietokone palaa riville 130 ja sieltä riville 110 printtaamaan valikon.

Silmukka, joka vertaa Z$:a kaikkiin JOUKKUE$:n nimiin. Kun se löytää nimen, joka on sama kuin Z$ se menee riville 440.

Tämä rivi on mukana siltä varalta, että kirjoitat joukkueen nimen väärin tai kirjoitat nimen, jota ei ole tietokannassa.

Printtaa joukkueen nimen.

Jos sinulla on BBC mikro, lue sivun 141 huomautus.

```
450 FOR J=1 TO 12
455 IF M(I,J)=1 THEN PRINT "VOITTI
CUPIN VUONNA ";VUOSI(J): PRINT
460 IF M(I,J)=2 THEN PRINT "OLI
FINALISTI VUONNA ";VUOSI(J): PRINT
465 NEXT J

470 PRINT
480 INPUT "PAINA RETURN JOS HALUAT
VALIKON";X$

490 RETURN
```

Silmukka, joka etsii valitsemaasi joukkuetta koskevia tietoja taulukosta. I on taulukon rivinumero. I:n arvon asettaa rivien 420–430 silmukka ja arvo on JOUKKUE$:ssa olevan joukkueesi alaindeksi (eli numero, joka kertoo joukkueen paikan taulukossa). J on sarakenumero ja joka kerta kun silmukka toistuu tietokone katsoo rivin I seuraavaa saraketta.

Toimii kuten rivi 240.

Takaisin riville 130.

```
500 REM ALIOHJELMA VUODEN SYÖTTÖ
505 CLS

510 INPUT "ANNA VUOSI TAI KIRJOITA
VALIKKO JOS HALUAT NÄHDÄ LUETTELON
UUDESTAAN ";Z$
515 IF Z$="VALIKKO" THEN RETURN

520 FOR I=1 TO 12
525 IF VAL(Z$)=VUOSI(I) THEN GOTO 540
530 NEXT I

535 PRINT: PRINT "VUOTTA EI LÖYDY -
YRITÄ UUDESTAAN": PRINT: GOTO 510
```

Toimii kuten rivi 410, mutta tällä kertaa valitsemasi vuosi pannaan Z$:iin.

Silmukka, joka vertaa Z$:a kaikkiin taulukon VUOSI vuosiin. Merkkijonomuuttujaa ei voi verrata numeeriseen muuttujaan, joten tarvitaan BASIC-käsky VAL. Se saa tietokoneen käsittelemään Z$:ssa olevia merkkejä numeroina.

```
540 PRINT: PRINT "VUONNA ";Z$: PRINT
550 FOR J= 1 TO 10
555 IF M(J,I)=1 THEN PRINT
JOUKKUE$(J);" VOITTI CUPIN"
560 NEXT J

565 PRINT

570 INPUT "PAINA RETURN JOS HALUAT
VALIKON";X$
580 RETURN
```

Tämä silmukka toimii samalla tavoin kuin rivit 450–460. Tällä kertaa sarakenumeron valitsee taulukon VUOSI vuoden alaindeksi ja taulukon rivinumero muuttuu joka kerta, kun silmukka toistuu.

Takaisin riville 130.

```
600 REM ALIOHJELMA PÄÄTTÄÄ OHJELMAN
610 INPUT "HALUATKO LOPETTAA - (K/E)";X$
620 IF X$<>"K" THEN RETURN ELSE END
```

Varmistaa, että todella haluat lopettaa. Jos kirjoitat K BASIC-käsky END käskee tietokoneen lopettaa ohjelman. Jos kirjoitat jotain muuta, tietokone palaa riville 130. Sana ELSE on kätevä tapa lisätä ehtoja IF/THEN-lauseisiin. Tästä on lisää tietoa seuraavalla sivulla.

Jos mikrosi ei tunne käskyä ELSE, voit panna sanan END yksinään omalle rivilleen.

```
1000 FOR I=1 TO 12: READ VUOSI(I): NEXT I
1010 DATA 1930,1934,1938,1950
1020 DATA 1954,1958,1962,1966
1030 DATA 1970,1974,1978,1982
```

Silmukka, joka lukee datan taulukkoon VUOSI.

```
1100 FOR I=1 TO 10: READ JOUKKUE$(I): NEXT I
1110 DATA URUGUAY, ARGENTIINA
1115 DATA ITALIA, TSEKKOSLOVAKIA, UNKARI
1120 DATA LANSI-SAKSA, BRASILIA
1125 DATA RUOTSI, ENGLANTI, HOLLANTI
```

Silmukka, joka lukee datan taulukkoon JOUKKUE$.

Kirjoita data sisään oikein huolellisesti. Pilkkujen tai lukujen puuttuminen aiheuttaa ohjelmaan virheitä.

```
1200 FOR I=1 TO 10: FOR J=1 TO 12
1205 READ M(I,J)
1210 NEXT J: NEXT I
```

Sisäkkäiset silmukat, joilla luetaan dataa kaksiulotteiseen taulukkoon M.

```
1215 RETURN
```

Tällä kertaa palataan takaisin riville 110.

```
1220 DATA 1,0,0,1,0,0,0,0,0,0,0,0
1230 DATA 2,0,0,0,0,0,0,0,0,1,0
1240 DATA 0,1,1,0,0,0,0,0,2,0,0,1
1250 DATA 0,2,0,0,0,0,2,0,0,0,0,0
1260 DATA 0,0,2,0,2,0,0,0,0,0,0,0
1270 DATA 0,0,0,0,1,0,0,2,0,1,0,2
1280 DATA 0,0,0,0,0,1,1,0,1,0,0,0
1290 DATA 0,0,0,0,0,2,0,0,0,0,0,0
1300 DATA 0,0,0,0,0,0,0,1,0,0,0,0,
1310 DATA 0,0,0,0,0,0,0,0,0,2,2,0
```

Tässä on M:n datat.

On hyvä tarkistaa datat useita kertoja lukemalla ne riveittäin ja sarakkeittain. Jos yksikin luku on väärin, tietokone antaa virheellisiä tietoja taulukkoa luettuaan.

LISTAUS JATKUU SEURAAVALLA SIVULLA

```
2000 REM ALIOHJELMA PRINTTAA VALIKON

2010 CLS

2020 PRINT "      VALIKKO"
2030 PRINT "      *******"

2040 FOR I=1 TO 6: PRINT: NEXT I

2050 PRINT "JOUKKUELUETTELO - 1"

2060 PRINT

2070 PRINT "VUOSILUETTELO - 2"

2080 PRINT

2090 PRINT "ANNA JOUKKUEEN NIMI - 3"

2100 PRINT

2110 PRINT "ANNA VUOSI - 4"

2120 PRINT

2130 PRINT "LOPETUS - 5"

2140 PRINT

2150 PRINT "KIRJOITA VALINTASI NUMERO"

2160 INPUT "JA PAINA RETURN";C

2170 IF C<1 OR C>5 THEN PRINT "KIRJOITA
LUKU VALILTA 1-5": GOTO 2150

2180 RETURN
```

Jätä tähän muutama välilyönti, jotta sana VALIKKO on keskellä valintojen yläpuolella.

Silmukka, joka jättää kuusi tyhjää riviä.

Nämä rivit printtaavat valikon. Valikon pitää olla selkeä ja käyttäjäystävällinen, jotta ohjelman käyttäjä ymmärtää täsmälleen, mitä hänen pitää tehdä.

Valitsemasi numero tallennetaan C:hen.

Tämä rivi on lisätty siltä varalta, että sattuisit kirjoittamaan jotain muuta kuin luvun väliltä 1–5. Alla on annettu lisätietoa sanasta OR.

Takaisin riville 120 oikeaa aliohjelmaa valitsemaan...

AND, OR ja ELSE★

```
IF A=3 AND C$="KYLLA" THEN LET D=D+1

IF X<0 OR X>100 THEN PRINT "YLITIT ASTEIKON"

IF IKA<36 THEN PRINT "NUORI" ELSE PRINT "VANHA"
```

Näillä BASIC-sanoilla voit lisätä ehtoja ja ohjeita IF/THEN-lauseisiin, kuten yllä olevissa esimerkeissä on tehty. Kun käytät sanaa AND tietokone toteuttaa käskyn THEN vain, jos molemmat IF-lauseen ehdot ovat tosia. OR saa tietokoneen toteuttamaan käskyn jos jompikumpi ehdoista on tosi. Sanalla ELSE voit antaa mikrolle käskyn, jonka se toteuttaa, mikäli kumpikaan ehto ei ole tosi. Osaatko kirjoittaa ELSE-sanaa käyttäen lyhyen ohjelman, joka ratkaisee oikealla esitetyn ongelman?

*Näitä BASIC-sanoja ei ole kaikissa mikroissa.

Arvoitus

Zak on kova robotti juoksemaan, se juoksee 500 metriä sekunnissa. Mutta jos lämpötila alittaa tai ylittää 15 astetta, Zakin juoksunopeus hidastuu tai nopeutuu lämpötilasta ja sekuntimäärästä riippuen? (Vastaus sivulla 137.)

Tietokannan muuttaminen

Kun olet päässyt selville siitä, miten ohjelma toimii, sinun on helppo muuttaa sitä toisenlaista tietokantaa varten. Alla on muutama ehdotus erilaisten tietokantojen laatimiseksi.

Kun olet valinnut tietokannan aiheen, piirrä taulukko tarvittavia lukuja varten (ota mallia sivulta 46). Taulukossasi voi olla eri määrä rivejä ja sarakkeita kuin edellisten sivujen ohjelmassa, joten joudut muuttamaan taulukoiden ulottuvuuksia. Pane seuraavaksi datat ohjelman datariveille ja kirjoita valikon tekstit uudestaan. Muista muuttaa DIM-lauseet ja silmukoiden ajokerrat, joilla data luetaan taulukoihin.

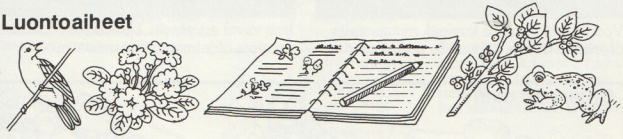
Aikakauslehdet

NUMERO$

AIHE$ | TAULUKKO

Aikakauslehtien tietokannasta voisit etsiä, missä numerossa jotain aihetta on käsitelty tai mitä aiheita käsiteltiin tietyssä numerossa. Näin et joudu selailemaan lehtiä tuntikausia etsiessäsi jotain artikkelia. Tarvitset taulukon nimeltä NUMERO$ ja taulukon nimeltä AIHE$ sekä matriisin.

Luontoaiheet

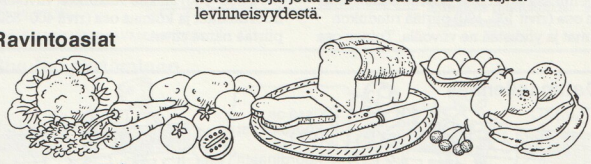


LINTU$

PAIKKA$ | TAULUKKO

Lintuja tai kasveja käsittelevästä tietokannasta voisit saada tiedon esim. siitä, milloin tai missä näit jonkin linnun tai kasvin. Tarvitset yhden taulukon lintujen nimiä varten ja toisen niiden havaitsemispaikkaa tai -vuodenaikaa varten. Luontojärjestöt laativat tällaisia tietokantoja, jotta ne pääsevät selville eri lajien levinneisyydestä.

Ravintoasiat



RUOKA$

VIT$ | TAULUKKO

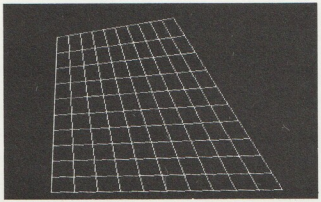
Tästä tietokannasta saisit selville, missä ruuissa on tiettyä vitamiinia tai mitä vitamiineja jossain tietyssä ruuassa on. Toinen mahdollisuus olisi tehdä tietokanta, jossa on ruokia ja niiden kalorimäärät. Siitä näkisi, paljonko kaloreita tietyssä ruuassa on tai minkä ruokien kalorimäärä ylittää tietyn rajan.

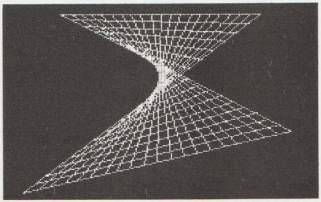
Pikagrafiikka

Tämä ohjelma piirtää näyttöruudulle yksinkertaisia kuvioita ja täyttää ne viivaristikolla. Tietokonegrafiikassa käytetään usein ristikoita, jotta kuviot näyttävät kolmiulotteisilta tai muuten vain hienommilta.

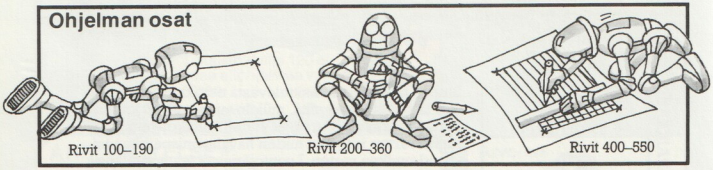
Ohjelmassa käytetään grafiikkakäskyjä PLOT X,Y pistettä varten ja DRAW X,Y viivan piirtämistä varten. Koordinaatit X ja Y mitataan näyttöruudun reunasta. Sinun pitää muuttaa nämä ohjeet omalle mikrollesi sopiviksi ja lisätä sen mahdollisesti tarvitsemat ylimääräiset grafiikkakäskyt. *

On kaksi tapaa kirjoittaa grafiikkaohjelmia. Voit panna mikron laskemaan ja piirtämään jokaisen pisteet erikseen, jolloin kuva syntyy vähitellen. Toinen tapa on panna mikro ensin laskemaan laskut, tallentaa tulokset taulukoihin ja piirtää sitten koko kuva melkein hetkessä. Seuraavassa ohjelmassa piirretään juuri "pikagrafiikkaa".





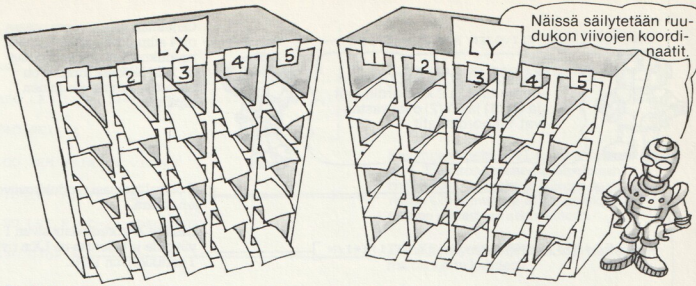
Voit piirtää erilaisia kuvioita muuttamalla ohjelman datoja. Voit myös piirtää ruudukon viivat eri värein. Listauksen lopussa on neuvoja ohjelman muuttamista varten.



Ohjelmassa on kolme pääosaa. Ensimmäinen osa (rivit 100–190) piirtää ruudukon kulmat ja yhdistää ne viivoilla. Toinen osa (rivit 200–360) laskee ruudukon viivojen koordinaatit ja kolmas osa (rivit 400–550) piirtää nämä viivat.

Ohjelma tallentaa koordinaatteja koskevat tiedot neljään taulukkoon. CX ja CY sisältävät ruudukon neljän kulman koordinaatit. Ohjelman alussa tietokoneelle annetaan tiedot näitä taulukoita varten. CX(1) ja CY(1) sisältävät ruudukon ensimmäisen kulman koordinaatit, CX(2) ja CY(2) toisen kulman jne.

*Joissain mikroissa, esim. Spectrumissa ja Oricissa, X- ja Y-koordinaatit mitataan viimeksi piirretystä pisteestä. Oheisen ohjelman muutosohjeet näille tietokoneille on annettu sivulla 142.



Taulukoissa LX ja LY säilytetään ruudukon viivojen X- ja Y-koordinaatit. Nämä ovat kaksiulotteisia taulukoita, joissa kummassakin on neljä riviä. Sarakkeiden määrän ratkaisee se, montako ruudukkoviivaa haluat. Kullakin rivillä on ruudukon yhden sivun viivojen koordinaatit. Rivi 1 on sivu 1 jne. Tietokone tallentaa tiedot LX:ään ja LY:hyn laskiessaan ohjelman laskuja.

Ohjelma

```
100 INPUT "MONTAKO RISTIKKOVIIVAA
HALUAT? ";N
110 DIM CX(4),CY(4)
115 DIM LX(4,N), LY(4,N)
120 CLS
130 REM PIIRTAA RISTIKON SIVUT
135 REM PANE TAHAN MIKROSI
GRAFIIKKATILAN KASKY

140 FOR I=1 TO 4
145 READ CX(I), CY(I)
150 NEXT I

160 DATA 100,50,700,50,500,700,150,500

170 PLOT CX(4),CY(4)

180 FOR I=1 TO 4
185 DRAW CX(I),CY(I)
190 NEXT I
```

Kokeile 20 viivaa mikroilla, joissa on tarkkuusgrafiikka ja 5 viivaa muissa mikroissa.

Kertoo tietokoneelle, miten suuria taulukot ovat.

Tässä annetut luvut ovat viereisen sivun vasemmanpuoleisen kuvan ristikkoa varten. Muuta lukuja jos haluat erilaisia ristikoita.

Silmukka, joka lukee nurkkien datat CX:ään ja CY:hyn. Aina, kun silmukka toistuu, rivin 160 seuraavat kaksi lukua luetaan CX:ään ja CY:hyn.

Tässä ovat nurkkien koordinaatit. Muuta näitä, jos näyttöruutusi vaatii.

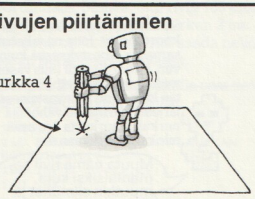
Käytä mikrosi käskyjä, jotka vastaavat sanoja PLOT ja DRAW.

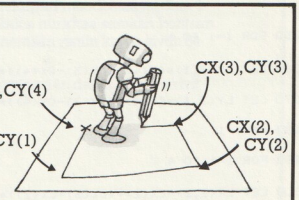
Piirtää nurkan 4 käyttäen CX(4):ään ja CY(4):ään tallennettuja lukuja.

Silmukka ruudukon sivujen piirtämiseksi.

LISTAUS JATKUU SEURAAVALLA SIVULLA

Sivujen piirtäminen

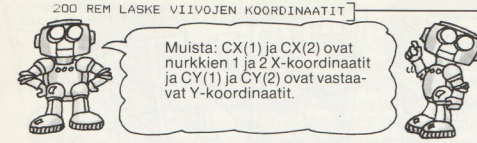




Tietokone piirtää ruudukon reunat siten, että se piirtää ensin nurkan 4 (rivi 170). Sitten rivien 180–190 silmukka piirtää viivan nurkkaan 1. Aina, kun silmukka toistuu, muuttujaan I lisätään 1 ja tietokone piirtää viivan seuraavaan nurkkaan.

```
200 REM LASKE VIIVOJEN KOORDINAATIT
```

Ohjelman seuraavassa osassa on 4 silmukkaa, jotka laskevat ruudukon viivojen koordinaatit. Alla olevissa kuvissa on selitetty, miten laskeminen tapahtuu.




```
210 FOR I=1 TO N
```

N on valitsemasi ruudukkoviivojen määrä.

```
220 LET LX(1,I)=CX(1)+(CX(2)-CX(1))*I/N
```

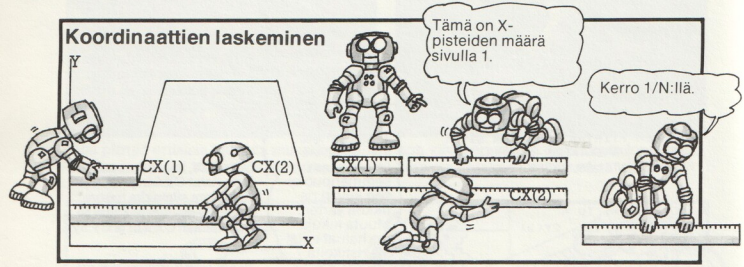
Laskee X-koordinaatit sivun 1 viivoille ja tallentaa ne LX:n riville 1 sarakkeisiin 1–N.

```
230 LET LY(1,I)=CY(1)+(CY(2)-CY(1))*I/N
```

Laskee Y-koordinaatit sivun 1 viivoille ja tallentaa ne LY:n riville 1 sarakkeisiin 1–N.

```
240 NEXT I
```

LISTAUS JATKUU ALLA



Kukin silmukka laskee ruudukon yhden viivojen koordinaatit. Esim. rivit 210–240 laskevat sivun 1 koordinaatit. Rivillä 220 tietokone vähentää CX(1):n CX(2):sta. Tästä saadaan X-pisteiden määrä sivulla 1. Silmukan ensimmäisessä ajossa tämä kerrotaan 1/N:llä (N on valitsemasi viivamäärä). Jos N = 5 tulos on 1/5 sivun pituudesta.

Sitten luku lisätään CX(1):ten ja vastaus tallennetaan LX(1,1):en. Silmukan toisessa ajossa I = 2 joten kertoja on 2/5 ja vastaus tallennetaan LX(1,2):ten. Näin tehdään kaikilla I:n arvoilla 1–N ja saadaan ruudukkoviivojen kaikki X-koordinaatit sivulle 1. Rivi 230 selvittää samalla tavalla Y-koordinaatit.

```
250 FOR I=1 TO N
260 LET LX(3,I)=CX(4)+(CX(3)-CX(4))*I/N
270 LET LY(3,I)=CY(4)+(CY(3)-CY(4))*I/N
280 NEXT I
```

Silmukka, joka laskee sivun 3 koordinaatit. Jotta ne tallentuvat samassa järjestyksessä kuin sivun 1 koordinaatit (siis vasemmalta oikealle) tietokoneen pitää laskea ne päinvastaisessa järjestyksessä. Se vähentää nurkan 4 nurkasta 3 ja lisää tuloksen nurkkaan 4.

Muuta nämä rivit samanlaisiksi kuin muutkin silmukat ja katso, mitä siitä seuraa.

```
290 FOR I=1 TO 4
300 LET LX(2,I)=CX(2)+(CX(3)-CX(2))*I/N
310 LET LY(2,I)=CY(2)+(CY(3)-CY(2))*I/N
320 NEXT I
```

Silmukka, joka laskee sivun 2 koordinaatit.

LISTAUS JATKUU SEURAAVALLA SIVULLA

```
330 FOR I=1 TO N
340 LET LX(4,I)=CX(1)+(CX(4)-CX(1))*I/N
350 LET LY(4,I)=CY(1)+(CY(4)-CY(1))*I/N
360 NEXT I

400 REM PIIRTAA VIIVAT
410 LET RIVI=1: GOSUB 500
420 LET RIVI=2: GOSUB 500
430 STOP

500 REM ALIOHJELMA
510 FOR I=1 TO N
520 PLOT LX(RIVI,I),LY(RIVI,I)
530 PLOT LY(RIVI+2,I),LY(RIVI+2,I)
540 NEXT I
550 RETURN
```

Silmukka, joka laskee sivun 4 koordinaatit.

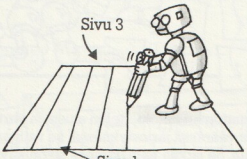
Muuttuja nimeltä RIVI, jonka alkuarvo on 1. Tietokone menee aliohjelmaan riville 500. Aliohjelman jälkeen se palaa riville 420 ja muuttaa RIVIn arvoksi 2 ja menee uudestaan aliohjelmaan.

Tämä pysäyttää ohjelman, kun aliohjelma on ajettu kahdesti.

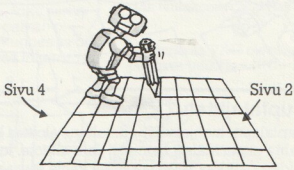
RIVI ja I ovat LX:n ja LY:n alaindeksit. Ne kertovat mikrolle, mistä lokerosta kunkin ruudukkoviivan X- ja Y-koordinaatit löytyvät. RIVI on rivin numero ja I on sarake. Rivillä 1 on sivun 1 koordinaatit, rivillä 2 sivun 2 koordinaatit jne.

Takaisin riville 420 aliohjelman ensimmäisessä ajossa ja riville 430 sen toisessa ajossa.

Ruudukkoviivojen piirtäminen



Aliohjelman ensimmäisessä ajossa RIVI=1 joten rivillä 520 mikro piirtää pisteen sivulle 1. Rivillä 530 se lisää RIVIin 2 ja piirtää viivan sivulle 3. Aliohjelman toisessa ajossa



RIVI=2 joten se piirtää pisteitä sivulle 2 ja viivoja sivulle 4.

Ohjeita ohjelman muuttamista varten

1. Saat eri muotoisia ristikoita, kun ensin piirrät ja lasket paperilla haluamasi kuvion koordinaatit. Muista, että rivin 160 ensimmäinen lukupari muodostaa nurkan 1 koordinaatit, toinen pari nurkan 2 jne. Jos panet kaksi nurkkaa samaan kohtaan tuloksena on kolmio. Yritä saada neliön sivut menemään ristiin kuten sivun 54 oikean puoleisessa kuvassa.

2. Sanalla INPUT ja silmukalla saat tietokoneen pyytämään dataa. Korvaa rivit 140–160 seuraavilla riveillä:

```
140 FOR I=1 TO 4
150 PRINT "MITKÄ OVAT NURKAN ";I;"
KOORDINAATIT"
155 INPUT CX(I),CY(I)
160 NEXT I
```

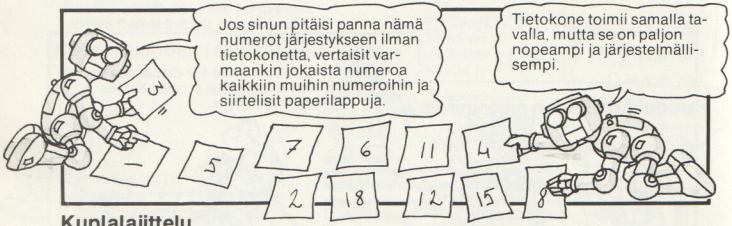
3. Värillisiä viivoja saat, kun lisäät mikrosi värikäskyn riveille 410 ja 420 ennen sanaa GOSUB. Muista erottaa värikäsky GOSUBista kaksoispisteellä.

Tiedonlajitteluohjelmia

Joskus tietoja eli dataa pitää panna aakkosjärjestykseen tai numerojärjestykseen. Näin voit joutua tekemään esim. lehtiartikkelien hakemistolle tai säätä, lintuja, kasveja tai kotiseutusi historiaa koskeville tiedostoille. Lyhyet luettelot on aivan helppo lajitella käsin, mutta kun tietoa on paljon, on tietokone ihmistä nopeampi ja tekee tarkempaa työtä.

On erityisiä tiedonlajitteluohjelmia. BASIC-kielellä on kirjoitettu jo useita tällaisia. Olet ehkä nähnyt lehdissä niiden mainoksia. Eri ohjelmissa on käytetty eri ohjelmointimenetelmiä ja ne soveltuvat erilaisiin tehtäviin.

Seuraavilla sivuilla on kaksi erilaista lajitteluohjelmaa. Toisen nimi on kupla (alla kerrotaan, mistä nimi tulee) ja toisen nimi on (tekijänsä mukaan) Shell. Kupla on erittäin hidas lajitteluohjelma ja se sopii vain pienille tietomäärille. Shell on paljon nopeampi. Sivulla 63 on annettu rivejä, jotka voit lisätä ohjelmiin, jotta pääset vertaamaan niitä ja tutkimaan mikrosi nopeutta.



Kuplalajittelu

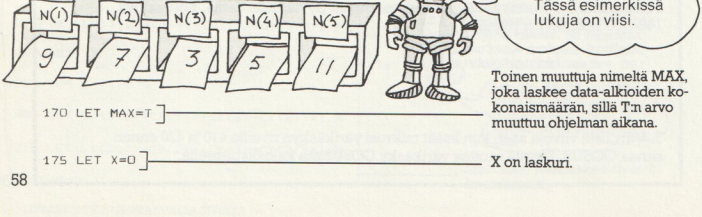
Kuplalajitteluohjelmassa tietokone aloittaa lajittelemattoman tietoluettelon alusta ja vertaa kahta ensimmäistä tietoa eli data-alkiota. Jos ne ovat väärässä järjestyksessä, se vaihtaa niiden paikkaa ja vertaa sitten kahta seuraavaa dataa. Se jatkaa näin, kunnes se on käynyt kaikki tiedot läpi. Suuremmat luvut "kuplivat" näin luettelon loppupäähän.

Seuraava ohjelma on kuplalajittelu lukuja varten. Seuraavalla aukeamalla on vastaava ohjelma sanojen lajittelua varten.

```
100 REM LUKUJEN KUPLALAJITTELU
110 INPUT "MONTAKO LUKUA
LAJITELLAAN? ";T
120 DIM N(T)
130 FOR I=1 TO T
140 PRINT "LUKU ";I
150 INPUT N(I)
160 NEXT I
170 LET MAX=T
175 LET X=0
```

Taulukko nimeltä N, jossa on T lokeroa. T ilmoittaa, montako lukua haluat lajitella.

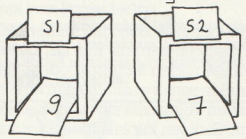
Pyytää lajiteltavia lukuja ja tallentaa ne taulukkoon.

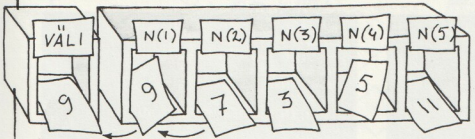


Toinen muuttuja nimeltä MAX, joka laskee data-alkioiden kokonaismäärän, sillä T:n arvo muuttuu ohjelman aikana.

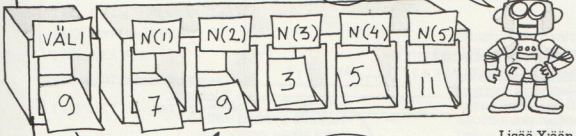
X on laskuri.

```
180 FOR C=1 TO T-1
190 LET S1=N(C): LET S2=N(C+1)
200 IF S1<=S2 THEN GOTO 250
210 LET VALI=N(C)
220 LET N(C)=N(C+1)
230 LET N(C+1)=VALI
240 LET X=X+1
250 NEXT C
260 IF X>0 THEN LET T=T-1: GOTO 175
270 PRINT "LAJITELLUT NUMEROT OVAT"
280 FOR I=1 TO MAX
290 PRINT N(1)
300 NEXT I
```

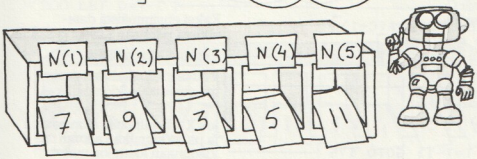




Aina, kun silmukka toistuu, luku siirtyy yhden paikan oikealle kunnes se on oikeassa lokerossa.



Silmukan toisessa ajossa C=2 joten N(C)=N(2) ja N(C+1)=N(3)



Toista IF/THEN jos panet rivin 260 GOTO-sanan uudelle riville.

Silmukka, joka ajetaan T-1 kertaa. Näin monta kertaa täytyy verrata luettelon lukupareja.

Muuttujat S1 ja S2 säilyttävät kunkin lukuparin niin kauan, kuin niitä vertaillaan. Ensimmäisellä ajokerralla C = 1 joten N1 ja N2 pannaan S1:een ja S2:een.

Vertaa kahta lukua toisiinsa. Jos S1 on pienempi kuin S2 luvut ovat oikeassa järjestyksessä ja rivi 250 lähettää mikron takaisin silmukan alkuun valitsemaan seuraavan lukuparin. Jos S1 on isompi kuin S2 tietokone jatkaa seuraaville riveille, jotka vaihtavat numeroiden paikkaa taulukossa.

N(C):ssä oleva luku pannaan muuttujaan nimeltä VÄLI.

N(C+1):ssä oleva luku siirretään taulukossa paikkaan N(C).

Muuttujassa VÄLI oleva luku pannaan paikkaan N(C+1).

Lisää X:ään 1 merkiksi siitä, että paikan vaihto on suoritettu.

Lähettää tietokoneen takaisin vertaamaan seuraavaa lukuparia. Kun silmukka on toistettu T-1 kertaa tietokone on verrannut kaikkia lukuja kerran ja siirtyy riville 260.

Jos X on enemmän kuin 0 on suoritettu paikan vaihto joten tietokone vähentää T:stä 1, koska yksi luku on nyt oikealla paikallaan. Sitten se palaa silmukan alkuun. Jos X = 0 luvut ovat oikeassa järjestyksessä ja tietokone jatkaa riville 270.

MAX on lajiteltujen lukujen kokonaismäärä.

Sanojen kuplalajittelu

Seuraava ohjelma on kuplalajittelu sanoja varten. Ohjelma on sama kuin lukujen lajitteluohjelma, mutta dataa sisältävät muuttujat (N, S1, S2 ja VÄLI) ovat nyt merkkijonomuuttujia.*

Tietokone vertailee kirjaimia samalla menetelmällä kuin numeroita. Tietokoneen sisällä kirjaimet ja symbolit on esitetty numeroina. Kun siis käsket tietokoneen verrata merkkejä, se vertailee niiden numerokoodeja. Kahta sanaa verratessaan se tarkistaa ensin kummankin sanan ensimmäisen kirjaimen. Jos ne ovat samat, se vertaa toisia kirjaimia, sitten kolmansia jne. Merkkijonomuuttujiin voi panna sekä kirjaimia että numeroita. Sanojen kuplalajittelua voi siis käyttää myös lajittelemaan dataa, jossa on sekä sanoja että numeroita kuten esim. osoitteissa ja sivuhakemistoissa.

```
MONTAKO DATAA LAJITELLAAN? 4
DATA 1
?LEVYASEMAT 34 76 82 93
DATA 2
?KONEKIELI 55 72 85
DATA 3
?VARIT 32
DATA 4
?GRAFIIKKA 8 23 45
LAJITELTU LUETTELO ON
GRAFIIKKA 8 23 45
LEVYASEMAT 34 76 82 93
KONEKIELI 55 72 85
VARIT 32
```

Tässä esimerkissä tietokone lajittelee sivuhakemiston tietoja ja oikealla se lajittelee osoitteita. Tiedot kirjoitettiin sisään ilman pilkkuja, sillä yleensä mikroissa pilkku on ns. erotin eli rajamerkki kahden data-alkion välillä. Jos haluat käyttää merkkijonoissa pilkkuja, pane jono lainausmerkkeihin.

```
MONTAKO DATAA LAJITELLAAN? 4
DATA 1
?KALLIO MARJA PITKAKATU 12
DATA 2
?SALLINEN SIMO ISOKATU 12
DATA 3
?LAHTINEN AHTI MAKIPOLKU 3
DATA 4
?ROPELO RISTO OIKOTIE 66
LAJITELTU LUETTELO ON
KALLIO MARJA PITKAKATU 12
LAHTINEN AHTI MAKIPOLKU 3
ROPELO RISTO OIKOTIE 66
SALLINEN SIMO ISOKATU 12
```

Ohjelma

```
100 REM SANOJEN KUPLALAJITTELU
110 INPUT "MONTAKO DATAA LAJITELLAAN? ";T
120 DIM N$(T)
130 FOR I=1 TO T
140 PRINT "DATA ";I
150 INPUT N$(I)
160 NEXT I
170 LET MAX=T
175 LET X=0
180 FOR C=1 TO T-1
190 LET S1$=N$(C): LET S2$=N$(C+1)
200 IF S1$<=S2$ THEN GOTO 250
210 LET VALI$=N$(C)
220 LET N$(C)=N$(C+1)
230 LET N$(C+1)=VALI$
240 LET X=X+1
250 NEXT C
260 IF X>0 THEN LET T=T-1: GOTO 175
270 PRINT "LAJITELTU LUETTELO ON"
280 FOR I=1 TO MAX
290 PRINT N$(I)
300 NEXT I
```

Taulukko nimeltä N$, jossa on T lokeroa.

Kokeile syöttää dataa, joka alkaa symbolilla, numerolla, suuraakkosella ja pienaakkosella, jotta näet, mihin järjestykseen mikrosi panee ne.

Kaksi ensimmäistä data-alkiota pannaan S1$:iin ja S2$:iin.

Vertaa S1$:a ja S2$:a.

Vaihtaa N$:n kahden ensimmäisen data-alkion paikkaa.

Jos X>0 vähentää summasta 1 ja palaa silmukan alkuun tarkistamaan luettelon uudestaan.

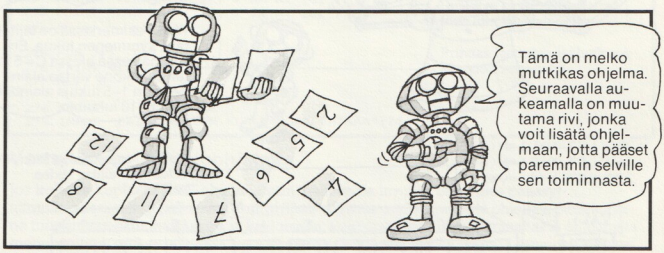
Toista IF/THEN jos panet rivin 260 GOTO-sanan uudelle riville.

*Sinclair (Timex) tietokoneissa muuta rivi 120 seuraavaan muotoon: DIM N$(T,N) jossa N on pisimmän käyttämäsi merkkijonon pituus.

Shell-lajittelu

Jos lajiteltavaa on paljon, on kuplaohjelma kovin hidas. Joillain mikroilla 50 datan lajittelu voi kestää lähes minuutin. Seuraava ohjelma on lukujen Shell-lajittelu. Se on noin kolme kertaa nopeampi kuin kuplaohjelma.

Shell-ohjelmassa tietokone jakaa lajiteltavat datat kahteen ryhmään ja vertaa toisen puoliskon datoja toiseen puoliskoon. Sitten se jakaa luettelon uudestaan kahtia ja tekee lisää vertailuja ja siirtää isommat luvut luettelossa ylöspäin käyttäen samaa paikanvaihtomenetelmää kuin kuplaohjelma.



Ohjelma

```
100 REM LUKUJEN SHELL-LAJITTELU
110 "MONTAKO LUKUA
LAJITELLAAN? ";T
120 DIM N(T)
130 FOR I=1 TO T
140 PRINT "LUKU ";I
150 INPUT N(I)
160 NEXT I
170 LET C=T
180 LET C=INT(C/2)
190 IF C=0 THEN GOTO 330
200 LET D=T-C
210 LET E=1
220 LET F=E
230 LET G=F+C
```

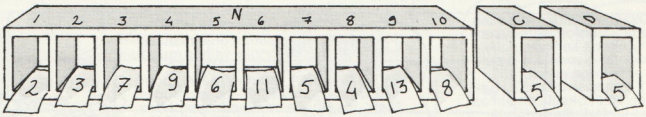
Nämä rivit ovat samat kuin kuplaohjelmassa. T on lajiteltavien lukujen kokonaismäärä ja rivit 120–160 pyytävät sinulta lajiteltavia lukuja ja tallentavat ne taulukkoon N.

Muuttuja C on sama kuin lajiteltavien datojen kokonaismäärä.

Jakaa C:n kahdella, jotta saadaan luettelon ensimmäisen puoliskon datojen lukumäärä. INT saa mikron unohtamaan mahdolliset desimaalipisteen jälkeiset numerot eli tekee C:stä kokonaisluvun.

Ohjelma jakaa toistamiseen C:n kahdella. Kun C = 0 mikro menee riville 330 ja printtaa lajitellun luettelon.

D ilmoittaa, montako dataa luettelon toisessa osassa on.



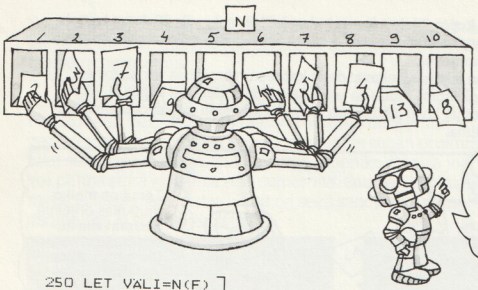
E on laskuri.

Muuttujat F ja G asettavat alaindeksit taulukkoon N, jonne kaikki numerot on tallennettu.

LISTAUS JATKUU SEURAAVALLA SIVULLA

```
240 IF N(F)<=N(G) THEN GOTO 300
```

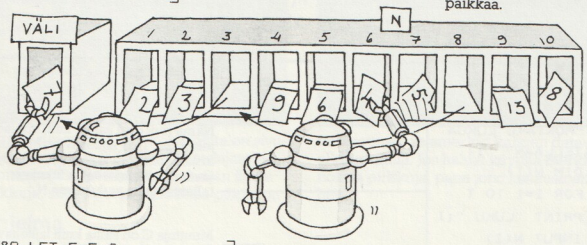
Jos N(F):ssä oleva luku on pienempi kuin N(G):ssä oleva luku, tietokone menee riville 300, lisää E:hen 1 ja palaa asettamaan F:n ja G:n alaindeksit seuraavaa lukuparia varten.



Tässä esimerkissä on lajiteltava kymmenen lukua. Ensimmäisessä ajossa C=5 jolloin tietokone vertaa alaindeksien 1–5 lukuja alaindeksien 6–10 lukuihin.

```
250 LET VÄLI=N(F)
260 LET N(F)=N(G)
270 LET N(G)=VÄLI
```

Jos N(F) on suurempi kuin N(G) tietokone vaihtaa niiden paikkaa.




```
280 LET F=F-C
290 IF F>0 THEN GOTO 230
300 LET E=E+1
310 IF E>D THEN GOTO 180
320 GOTO 220
330 PRINT "LAJITELTU LUETTELO ON"
340 FOR I=1 TO T
350 PRINT N(I)
360 NEXT I
```

Tarkistaa F:n arvon ja lisää laskuriin E 1 muuttaen F:n ja G:n arvot riveillä 220 ja 230.

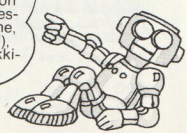
Varmistaa, että E on luettelon D-osassa. Jos se ei ole, palaa takaisin riville 180 ja jakaa luettelon uudestaan.

Jos E on vähemmän kuin D, mikro menee riville 220 asettamaan alaindeksit seuraavaa lukuparia varten.

Printtaa lajitellun luettelon.

Osaatko lisätä rivejä siten, että luettelon lajiteltuaan tietokone kertoo sinulle, montako vertailua ja paikanvaihtoa se suoritti? (Vastaus sivulla 137.) Sitten voit kokeilla ohjelmaa eri luettelolla ja nähdä, miten niiden käsittely eroaa toisistaan.

Jos haluat panna tämän ohjelman lajittelemaan merkkijonoja, vaihda muuttuja N muotoon N$ (riveillä 120, 150, 240–270 ja 350), muuta VÄLI muotoon VÄLI$ ja kirjoita PRINT-lauseet uudestaan. Jos sinulla on Sinclair tietokone, muuta rivi 120 muotoon DIM N$(T,N), jossa N on pisimmän antamasi merkkijonon pituus.



Miten ohjelma toimii?

Saat paremman käsityksen Shell-ohjelman toiminnasta, jos lisäät siihen seuraavat rivit. Ne printtaavat muuttujien arvoja ruudulle, joten pääset näkemään, mitä arvoja tietokone parhaillaan vertailee.

```
233 PRINT
235 PRINT "F=";F;",G=";G
237 PRINT "VERTAA N(";F;") JA N(";G;")"
245 PRINT "VAIHDA N(";F;") JA N(";G;")"
274 PRINT "LUETTELO=";
275 FOR J=1 TO T
276 PRINT N(J);" ";
277 NEXT J
278 PRINT
279 INPUT "PAINA RETURN KUN HALUAT JATKAA";Z$
```

F ja G ovat vertailevien lukujen alaindeksit.

Ilmoittaa paikkaansa vaihtavien lukujen alaindeksit.

Printtaa luettelon senhetkisessä muodossaan.

Vertailemme lajitteluohjelmia

Jos kokeilit kupla- ja Shell-ohjelmia vain muutamilla luvuilla, et kenties päässyt huomaamaan, miten paljon nopeampi Shell-ohjelma on. Voit verrata ohjelmia panemalla ne tuottamaan satunnaislukuja ja katsomalla, kuinka paljon aikaa kummaltakin ohjelmalta menee siihen, että se lajittelee luvut numerojärjestykseen. Mitä pitempi satunnaislukujen luettelo on, sitä suurempi aikaero syntyy ohjelmien välille. Seuraavalla sivulla on ohjelmia, jotka piirtävät graafisia esityksiä. Niillä voi havainnollistaa lajitteluohjelmien nopeuseroja.

Satunnaislukujen tuottaminen

```
140 LET N(I)=INT(RND(1)*200+1)
150 PRINT N(I)
165 INPUT "PIDA KELLO VALMIINA
JA PAINA RETURN KUN HALUAT
ALOITTAA LAJITTELUN";Z$
```

Voit muuttaa tämän luvun miksi tahansa luvuksi.

Jotta ohjelmat tuottavat itse omat lajiteltavat lukunsa, korvaa kummankin ohjelman rivit 140 ja 150 yllä annetuilla riveillä ja lisää uusi rivi 165, jotta voit itse määrätä, milloin lajittelu alkaa. Rivi 140 tuottaa satunnaislukuja väliltä 1–200 ja tallentaa ne taulukkoon N. Rivi 150 printtaa luvut ruudulle ja rivi 165 panee ohjelman odottamaan, kunnes painat RETURN.

Testin ajo

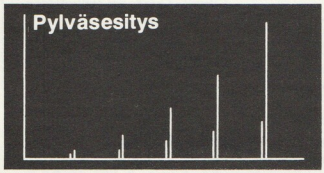
Kummallakin ohjelmalla testi pitää suorittaa useita kertoja. Kokeile ensimmäisellä kerralla, kauanko esim. 10 luvun lajittelu kestää ja jatka sitten 20, 30, 40 jne. luvulla. Erimerkkiset mikrot lajittelevat eri nopeudella ja joissain (esim. ZX81:ssä) on sekä nopea että hidas toimintatila. Alla on Apple II:lla saavutetut nopeudet.

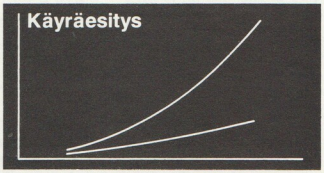
Lajittelutesti					
Lajiteltuja lukuja	10	20	30	40	50
Kupla	2 sek.	5 sek.	11 sek.	18 sek.	29 sek.
Shell	1 sek.	2 sek.	4 sek.	6 sek.	8 sek.

Graafiset esitykset

Tietokoneen antamia tuloksia on paljon helpompi lukea ja ymmärtää, jos esität ne kiintoisasti käyttäen sekä grafiikkaa että sanoja. Alla on pylväsdiagrammeja piirtävä ohjelma, jolla voi esittää kupla- ja Shell-ohjelmien nopeuserot. Viereisellä sivulla on ohjeet siitä, miten pylväsohjelma muutetaan käyriä piirtäväksi ohjelmaksi.

Ohjelmat ovat aivan helppoja ja voit nopeasti muuttaa ne erilaisten tietojen esittämistä varten. Voit myös parantaa ohjelmia lisäämällä niihin mikrosi värikomentoja, jolloin saat eri värisiä graafisia esityksiä.





Tässä on kahden ohjelmaruudun ruutunäytöt. Molemmissa verrataan sitä aikaa, mikä kahdelta lajitteluohjelmalta menee 10, 20, 30, 40 ja 50 luvun lajittelemiseen. Aika kasvaa Y-akselilla (ylös) ja lajiteltujen lukujen määrä X-akselilla (oikealle). Jos mikrosi osaa printata tekstiä pikseliosoitteisiin, voit lisätä kuvioihin selventäviä nimikkeitä. Viereisellä sivulla on kerrottu, miten tämä tehdään BBC-mikrolla.

Pylväsdiagrammiohjelma

```
100 DIM K(5): DIM S(5)
110 LET N=0
120 FOR I=10 TO 50 STEP 10
130 LET N=N+1
140 PRINT "KAUANKO KUPLAOHJELMALTA
MENI ";I
150 INPUT "LUVUN LAJITTELUUN? ";K(N)
160 INPUT "ENTA SHELL-OHJELMALTA? ";S(N
170 NEXT I
```

Taulukot K ja S sisältävät kupla- ja Shell-ohjelmien datat.

Silmukka, jolla pannaan dataa taulukoihin. N on laskuri, joka asettaa taulukoiden alaindeksit.

```
180 INPUT "MONTAKO PIKSELIA NAYTOSSASI
ON PYSTYSUUNNASSA? ";H
190 INPUT "ENTA SIVUSUUNNASSA? ";W
200 REM ANNA GRAFIIKKATILAN KASKY
JOS MIKROSI TARVITSEE SELLAISEN
```

Huomaa, että silmukkamuuttujalla I lasketaan myös lukuja.

Käskyjen PLOT ja DRAW luvut lasketaan näyttöruudun reunoista. Jos mikrosi käyttää koordinaatteja, jotka mitataan viimeksi piirretystä pisteestä, lue sivun 142 ohjeet.

```
210 REM PIIRTAA AKSELIT
220 PLOT 1,H: DRAW 1,1: DRAW W,1
230 REM PIIRTAA PYLVAAT
240 LET X=(W*0.75)/5
250 LET Y=(H*0.75)/K(5)
```

Piirtää akselit 1 pikselin päähän ruudun reunasta. H ja W ovat näyttöruudun korkeus ja leveys.

Luvut X ja Y valitsevat pylväiden mittasuhteet X- ja Y-akseleilla. X on 3/4 leveydestä jaettuna viidellä, joka on kummallakin ohjelmalla tehtyjen testien lukumäärä. Y on 3/4 korkeudesta jaettuna pisimmällä ajalla, joka on K(5).

LISTAUS JATKUU SEURAAVALLA SIVULLA

```
260 FOR I=1 TO 5
270 PLOT INT(I*X),1
280 DRAW INT(I*X),INT(K(I)*Y)
290 PLOT INT(I*X-4),1
300 DRAW INT(I*X-4),INT(S(I)*Y)
310 NEXT I
```

Osaatko panna pylväsdiagrammiohjelman piirtämään paksuja pylväitä? (Ratkaisu sivulla 137.)

Muuta BBC-mikrolle luku –4 luvuksi –10.

Silmukka, joka piirtää pylväät. Silmukkamuuttuja I laskee testien määrän ja aina, kun silmukka toistuu, tietokone piirtää kummallekin ohjelmalle pylvään.

Piirtää pisteen kohtaan I★X (vasemmasta reunasta) ja 1 (alareunasta). Silmukan ensimmäisessä ajossa I=1 ja X on X-akselin suuntaisen mittakaavan ilmoittama luku.

Piirtää viivan pisteeseen B(I)★Y. Silmukan ensimmäisessä ajossa I=1 joten B(I) on aika, joka meni 10 luvun lajitteluun ja Y on Y-akselin suuntaisen mittakaavan ilmoittava luku.

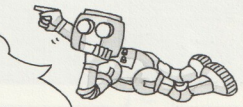
Piirtää pisteen kohtaan I★X-4 (vasemmasta reunasta) eli 4 pistettä vasempaa kuplaohjelman pylvästä.

Piirtää viivan pisteeseen S(I)★Y.

Palaa silmukan alkuun piirtämään seuraavat kaksi pylvästä.

Jos mikrosi osaa printata tekstiä pikseliosoitteisiin, voit lisätä ohjelmaan rivejä käyrien nimikkeitä varten. Esim. BBC mikrossa tarvitset seuraavat lisärivit.

```
305 VDU 5:MOVE  (I*X),INT(B(I)*Y)+50:PRINT "B"
308 VDU 5:MOVE  (I*X)-70,INT(S(I)*Y)+50:PRINT "S"
```



Käyränpiirto-ohjelma

Käyriä piirretään siten, että ensin piirretään käyrän alkupiste ja sitten siitä viiva käyrän seuraavaan pisteeseen. Voit muuttaa pylväsdiagrammiohjelman niin, että se piirtää käyriä. Muuta ohjelma riviltä 270 alkaen seuraavaan muotoon:

```
270 PLOT INT(X),INT(K(1)*Y)
280 FOR N=2 TO 5
290 DRAW INT(N*X),INT(K(N)*Y)
300 NEXT N
310 PLOT INT(X),INT(S(1)*Y)
320 FOR N=2 TO 5
330 DRAW INT(N*X),INT(S(N)*Y)
340 NEXT N
```

Piirtää kuplaohjelman ensimmäisen pisteen kohtaan X pikseliä vaakatasossa ja K(1)★Y pistettä pystysuunnassa.

Silmukka, joka piirtää kuplaohjelman käyrän. Aina, kun silmukka toistuu, N kasvaa 1:llä ja tietokone piirtää viivan käyrän seuraavaan pisteeseen.

Piirtää Shell-ohjelman ensimmäisen pisteen kohtaan X pikseliä vaakatasossa ja S(1)★Y pikseliä pystysuunnassa.

Silmukka, joka piirtää Shell-ohjelman käyrän.

Keskusteluohjelma

Seuraavilla sivuilla oleva ohjelma panee tietokoneen ikään kuin keskustelemaan kanssasi. Tietokoneen keskustelutaito riippuu tietenkin siitä, miten hyvän ohjelman syötät sisään ja mitä sanoja ja lauseita ohjelman taulukot sisältävät.*

Ohjelman päätehtävä on valita oikeat sanat ja vastata niillä käyttäjälle. Ohjelmassa on eräitä BASICin jononkäsittelyrutiineita, joiden ansiosta mikron vastaukset näyttävät paikoitellen aika älykkäiltä. Tällaisen ohjelman onnistuminen riippuu paitsi itse ohjelmasta myös sanoista ja lauseenparsista, joita panet ohjelmaan mukaan. Voit yrittää muuttaa tietokoneen sanastoa niin, että se "keskustelee" eri aiheista, tai voit panna sen vastaamaan ystävällisemmin... tai äreämmin.

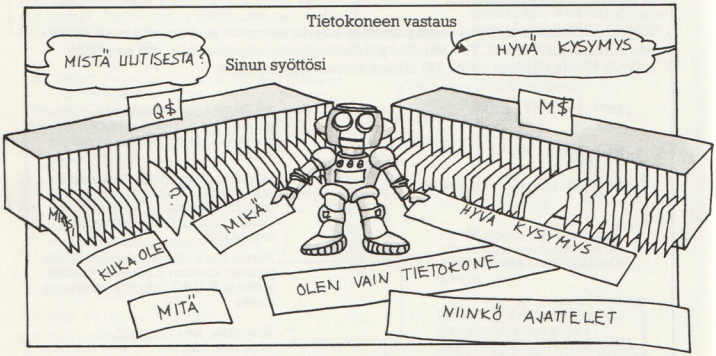
Näytteitä ohjelman ajosta

```
HEI! MIKA ON NIMESI? SIRKKU
SANOPA JOTAIN, SIRKKU
?HYVAA PAIVAA TIETOKONE
MITA MIELTA OLET TASTA ASIASTA: UUTISET
?MISTA UUTISESTA
HYVA KYSYMYS
?ETKO VASTAA KYSYMYKSEENI?
KUULEPAS, SIRKKU, MINUSTA SINA OLET AIVAN
YHTA YSTAVALLINEN KUIN KAIKKI MUUTKIN
JOIDEN KANSSA OLEN JUTELLUT
?KIITOS
EI KESTA
```

```
?MISTA HALUAT JUTELLA?
SANOPA JOTAIN AIHEESTA: SAA
?ULKONA SATAA
MITEN VOIT OLLA NIIN VARMA ASIASTA
?NAEN IKKUNASTA ULOS
OLEN KUULLUT ETTA OLET MELKOISEN TERAVA
NEROPATTI
?KUKA NIIN ON SANONUT?
SILLA EI OLE VALIA
?MIKSI ET VASTAA KYSYMYKSIINI?
HYVANEN AIKA, SIRKKU, ET KAI AJATTELE ETTA
JOKAINEN IHMINEN ON TOYKEA?
```

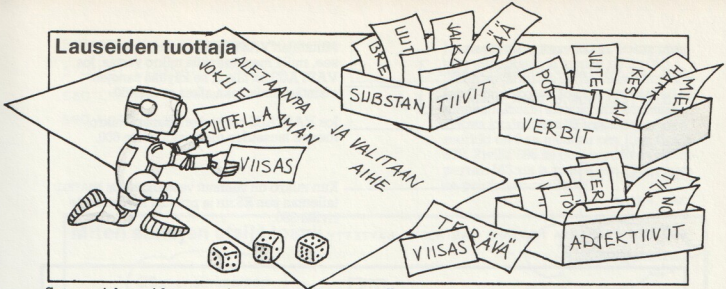
Miten ohjelma toimii

Ohjelmassa on kaksi menetelmää, joilla tietokoneen vastaukset saadaan aikaan. Toinen on sanojen etsintärutiini ja toinen on satunnaislauseiden tuottaja.



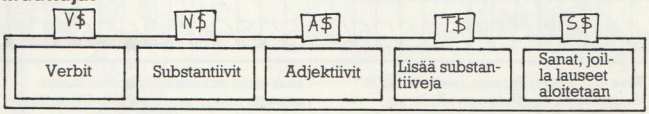
Sanojen etsintärutiinissa on luettelo usein käytetyistä sanoista. Luettelo on tallennettu taulukkoon Q$. Jokaista luettelon sanaa varten on M$:ssa sopiva vastaus. Kun näppäilet jotain, mikro tarkistaa, oletko käyttänyt jotain Q$:n sanaa ja jos olet, mikro vastaa sinulle käyttäen M$:ssa olevaa vastausta.

*Ohjeet ohjelman muuttamiseksi Sinclair tietokoneille on annettu sivulla 142.

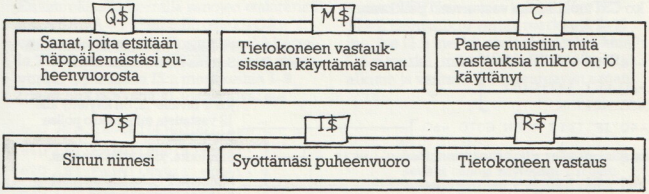


Satunnaislauseiden tuottajassa on puolivalmiita lauseita, joihin tietokone lisää verbejä, substantiiveja ja adjektiiveja, jotka se valitsee satunnaisesti. Kaikki sanat on tallennettu tietokoneen muistissa oleviin taulukoihin. Sanojen pitäisi antaa tulokseksi mielekkäitä lauseita. Voit itse muutella ohjelmaa.

Muuttujat



Näissä taulukoissa on satunnaislauseiden sanat.



Ohjelma

```
100 CLS
110 DIM V$(10),N$(10),A$(10)
120 DIM T$(10),S$(10)
130 DIM M$(30),Q$(30),C(30)
140 REM DATAN LUKEMINEN
150 GOSUB 1000
200 REM KAYTTAJAN PUHEENVUORO
210 INPUT "HEI! MIKA ON NIMESI? ";D$
220 PRINT
230 PRINT "SANOPA JOTAIN, ";D$
240 INPUT I$
250 IF I$="" THEN GOTO 220
260 IF I$="NAKEMIIN" THEN GOTO 910
```

Alustaa taulukot, joihin sanat tallennetaan. (Huom! Eräissä mikroissa alle 10 alkion taulukoiden ulottuvuuksia ei tarvitse määritellä.)

Menee aliohjelmaan lukemaan sanat taulukoihin.

Sinun puheenvuorosi pidetään I$:ssa.

Siltä varalta, että käyttäjä painaa vain RETURN ja I$ on tyhjä.

Jos kirjoitat NÄKEMIIN tietokone menee riville 910 kysymään, haluatko lopettaa ohjelman.

LISTAUS JATKUU SEURAAVALLA SIVULLA

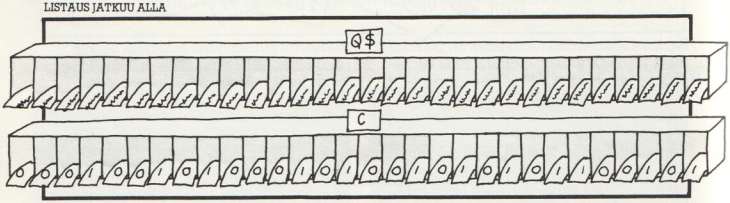
```
300 REM TIETOKONEEN VASTAUS
310 LET VASTAUS=INT(RND(1)*8+1)
320 IF VASTAUS<6 THEN GOTO 490
330 GOTO 600
340 PRINT
350 PRINT R$
360 PRINT
400 REM TARKISTA MONTAKO VASTAUSTA KAYTETTY
```

Muuttujan VASTAUS satunnaisluku ratkaisee, millä menetelmällä mikro vastaa. Jos VASTAUS on alle 6, se käyttää sanojen etsintärutiinia, joka alkaa rivillä 490.

Jos VASTAUS on 6 tai enemmän, mikro menee lauseiden tuottajaan riville 600.

Kun mikro on valinnut vastauksensa, se tallentaa sen R$:iin ja printtaa sen ruudulle rivillä 350.

LISTAUS JATKUU ALLA



Rivit 400–470 tarkistavat, montako M$:n vastauksista on jo käytetty. Aina, kun mikro löytää sinun vastauksestasi jonkin Q$:n sanan, se panee merkiksi 1:n taulukko C:n tuota sanaa vastaavaan paikkaan.

Tämä estää mikroa käyttämästä samaa vastausta uudestaan. Rivit 400–470 tarkistavat, montako merkkiä (1) C:ssä on. Jos niitä on yli 12, kaikki merkit nollataan.

```
405 LET T=0
410 FOR K=1 TO 30
420 LET T=T+C(K)
430 NEXT K
440 IF T<12 THEN GOTO 460
450 FOR K=1 TO 30: LET C(K)=0: NEXT K
460 LET T=0
470 GOTO 240
490 REM SANOJEN ETSINTARUTIINI
500 FOR SANA=1 TO 30
510 LET L1=LEN(Q$(SANA))
520 LET L2=LEN(I$)
530 FOR TESTI=1 TO L2
540 IF MID$(I$,TESTI,L1)=Q$
(SANA) THEN GOTO 560
550 NEXT TESTI: NEXT SANA:
GOTO 600
```

Silmukka, joka laskee montako merkkiä taulukossa C on. Summa tallennetaan T:hen.

Jos T on alle 12, on käytetty alle 12 vastausta eikä mikro nollaa merkkejä.

Silmukka, joka nollaa merkit.

Muuttuja T (joka laskee merkkien määrän) nollataan.

Takaisin riville 240 odottamaan käyttäjän puheenvuoroa.

Silmukka, joka toistuu niin monta kertaa kuin Q$:ssa on sanoja.*

Aina, kun silmukka toistuu, mikro mittaa seuraavan Q$:n sanan pituuden ja tallentaa pituuden L1:teen.

Puheenvuorosi (I$) merkkimäärä tallennetaan L2:teen.

TESTI on sisäkkäinen silmukka, joka toistuu yhtä monta kertaa kuin vastauksessasi on merkkejä. Aina, kun silmukka toistuu, mikro vertaa I$:n merkkejä Q$:n sanan merkkeihin. Jos ne ovat samat, se menee riville 560.

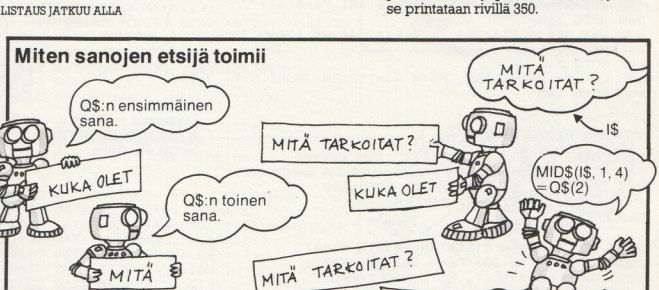
Jos mikro ei kaikkien silmukoiden toistumisen jälkeen löydä I$:sta yhtään Q$:n sanaa, se menee lauseiden tuottajaan riville 600.

*Jos sinulla on BBC mikro, lue sivun 141 huomautus.

```
560 IF C(SANA) >0 THEN GOTO 550
570 LET C(SANA)=C(SANA)+1
580 LET R$=M$(SANA)
590 GOTO 340
```

Jos se löytää saman sanan, se hyppää ulos silmukasta ja menee riville 560. Sitten se tarkistaa sanaa vastaavan merkin C-taulukossa. Jos merkki ei ole 0 se menee takaisin silmukoihin etsimään Q$:sta jotain toista I$:n sanaa. Jos merkki on 0 se muuttaa sen 1:ksi rivillä 570. Rivillä 580 se etsii vastaavan sananparren M$:sta ja panee sen R$:iin, josta se printataan rivillä 350.

LISTAUS JATKUU ALLA



Ensimmäisellä kerralla sanojen etsintärutiinissa SANA = 1 joten mikro tutkii Q$:n ensimmäisen sanan. Ensimmäisellä kerralla TESTI silmukassa TESTI = 1 joten mikro vertaa Q$:n sanaa I$:n merkkeihin 1–9 (mikä on Q$:n pituus). Jos ne eivät ole samat, TESTIsilmukka toistuu. Nyt TESTI = 2 joten mikro vertaa merkkejä 2–10 jne. Jos I$:n merkit eivät ole samoja kuin Q$:n merkit, mikro palaa SANAsilmukan alkuun ja valitsee Q$:n seuraavan sanan.

```
600 REM LAUSEIDEN TUOTTAJA
610 LET E=INT(RND(1)*10+1)
620 LET F=INT(RND(1)*10+1)
630 LET G=INT(RND(1)*10+1)
640 LET H=INT(RND(1)*10+1)
650 LET L=INT(RND(1)*10+1)
```

Satunnaislukuja, joilla valitaan, mitä lausetta ja sanoja käytetään. E:n luku valitsee lauseen. F valitsee verbit, G adjektiivit, H substantiivit ja L lauseen aloituksen.

```
660 ON E GOTO 700,720,740,760,
    780,800,830,850,870,890
```

E:n luku kertoo, mille riville mikro menee. Jos E = 1 se menee luettelon ensimmäiseen numeroon. Jos E = 2 se menee toiseen jne. Sivulla 48 kerrotaan enemmän BASIC-sanasta ON.

```
700 LET R$="MITA MIELTA OLET TASTA
ASIASTA: "+N$(H)+"?"

710 GOTO 340

720 LET R$=S$(L)+" "+D$+" ET KAI AJATTELE
ETTA JOKAINEN IHMINEN ON "+A$(G)+"?"

730 GOTO 340

740 LET R$="OLEN KUULLUT ETTA OLET
MELKOISEN "+A$(G)+" "+T$(H)

750 GOTO 340

760 LET R$=S$(L)+" "+D$+", MINUSTA
SINA OLET AIVAN YHTA "+A$(G)
```

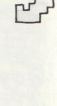
D$:ssa on sinun nimesi.

Riveillä 700–900 on 10 puolivalmista lausetta, joihin mikro lisää sanat N$:sta, V$:sta jne. Jonojen muodostamiseksi pitää käyttää merkkiä + Muista pannna lainausmerkkien sisällekin välilyöntejä, jotta sanojen väliin jää tyhjää. Mikro panee valmiin lauseen R$:iin ja palaa riville 340 printtaamaan sen.

LISTAUS JATKUU SEURAAVALLA SIVULLA

```
765 LET R$=R$+" KUIN KAIKKI MUUTKIN JOIDEN KANSSA OLEN JUTELLUT"
770 GOTO 340
780 LET R$="MINULLE TULI YHTÄKKIÄ "+A$(G)+" OLO"
790 GOTO 340
800 PRINT: PRINT "SSSSHHHH....ANNAS KUN MIETIN...."
810 LET R$="MINUSTA "+N$(H)+" ON "+A$(G)+" PUHEENAIHE"
820 GOTO 340
830 LET R$=D$+", SANOPA JOTAIN AIHEESTA: "+N$(H)
840 GOTO 340
850 LET R$="OLENKO MINÄ MIELESTÄSI "+A$(G)+"?"
860 GOTO 340
870 LET R$="ALETAANPA "+V$(F)+" JA VALITAAN JOKU
ENEMMÄN "+A$(G)+" AIHE"
880 GOTO 340
890 LET R$="ARVAAPAS, MITÄ AJATTELEN JUURI NYT, "+D$
900 GOTO 340
```

Kun muutat näitä lauseita, muuttuvat myös mikron vastaukset.



```
910 REM NÄKEMIIN-RUTIINI
920 PRINT "JOKO KYLLÄSTYIT?"
930 PRINT "EIKÖ KUKAAN HALUA JUTELLA KANSSANI?"
940 INPUT Z$: IF Z$="KYLLÄ" GOTO 210
950 PRINT: PRINT "HEI SITTEN"
960 END
```

Jos vastaat mikrolle NÄKEMIIN, rivi 260 lähettää sen tänne.

```
1000 REM SANOJEN ETSINTÄRUTIININ SANAT
1010 FOR I=1 TO 30: READ Q$(I): NEXT I
1020 DATA KUKA OLET, MITÄ, ?, TARKOIT, MIKSI
1030 DATA MINÄ, SINÄ, MIKÄ, MINU," SE "
1040 DATA PUHU, RÖYHKEÄ, MITEN, MILLÄ," NE "
1050 DATA " ON "," OLET ", EHKÄ, ÄLYKÄS
1060 DATA HÄVYTÖN, KIITOS, !, ?, ?, YMMÄRR,"EI"
1070 DATA " ON ", SINÄ, OVAT, TIEDÄT
```

Näillä riveillä on sanat, joita mikro etsii vastauksestasi. Näppäile ne juuri siten, kuin ne on kirjoitettu tähän. Lainausmerkkien sisällä olevat välilyönnit ovat osa dataa.

```
1100 REM MIKRON VASTAUKSET Q$:N SANOIHIN
1110 FOR I=1 TO 30: READ M$(I): NEXT I
1120 DATA OLEN PELKKÄ TIETOKONE
1130 DATA SILLÄ EI OLE VÄLIÄ, HYVÄ KYSYMYS
1140 DATA EN OLE VARMA, MIKSI EI?
1150 DATA MITÄ TARKOITAT, MITEN ON
ITSESI LAITA?
1160 DATA SITÄ SOPII MIETTIÄ, SE ON OMA
ASIASI, MITÄ "SE" TARKOITTAA
1170 DATA HALUATKO ETTÄ PIDÄN SUUNI KIINNI
1180 DATA NIINKÖ AJATTELET, SANOPA SINÄ
1190 DATA MITÄ TARKOITAT, VAI NIIN
1200 DATA OLET SIIS SAMAA MIELTÄ KANSSANI
1210 DATA ETKÖ PIDÄ MINUSTA
1220 DATA OLETPA EPÄVARMA, KIITOS
1230 DATA TÄMÄ ON KUULE VASTA ALKUA
1240 DATA EI KESTÄ, SITÄ SAMAA, EI KIINNOSTA
1250 DATA ONPA TYPERÄ KYSYMYS, SINULLA ON
ALHAINEN ÄLYKKYYSOSAMAARA
1260 DATA PÖTYPUHETTA, MITEN VOIT OLLA NIIN VARMA
ASIASTA
1280 DATA PUHUTAANPA SINUSTA, KAIKKEA SITÄ KUULEE
1290 DATA TIETÄMISESSÄ EN OLE VAHVOILLA
```

Tässä ovat tietokoneen vastaukset Q$:n sanoihin. Vastaukset ovat samassa järjestyksessä kuin Q$:n sanat. Esim. M$:n viides alkio on vastaus Q$:n viidenteen sanaan.

```
1300 REM LUE SUBSTANTIIVIT
1310 FOR I=1 TO 10: READ N$(I):NEXT I
1320 DATA JÄLKÄPALLO, BREAKTANSSI
1340 DATA SÄÄ, UUTISET
1350 DATA MINUN KESKUSYKSIKKÖNI, KALASTUS
1360 DATA SINIVALAS, EVOLUUTIO
1370 DATA MAANTIETO, RUOKA
```

Tässä ovat substantiivit, joita pannaan puolivalmiisiin satunnaislauseisiin.

```
1400 REM LUE VERBIT
1410 FOR I=1 TO 10: READ V$(I): NEXT I
1420 DATA AJATELLA, JUTELLA
1430 DATA KESKUSTELLA, MIETTIA
1440 DATA POHTIA, MIETISKELLA
1450 DATA HARKITA, POHDISKELLA
1460 DATA ANALYSOIDA, JARKEILLA
```

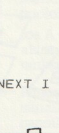
Tässä ovat satunnaislauseiden verbit.

Voit muuttaa minkä tahansa näistä sanoista toiseksi. Muista kuitenkin, että sanan pitää sopia lauseeseen mielekkäästi.

```
1500 REM LUE ADJEKTIIVIT
1510 FORI=1 TO 10: READ A$(I): NEXT I
1520 DATA TYPERÄ, TERAVA
1530 DATA ALYKÄS, VIISAS
1540 DATA NOKKELA, YSTÄVÄLLINEN
1550 DATA TYLSÄ, VÄSYTTÄVÄ
1560 DATA TÖYKEÄ, NEUROOTTINEN
```

Tässä ovat adjektiivit.

```
1600 REM LUE LAUSEEN ALOITTAJAT JA
LISÄSUBSTANTIIVIT PAREITTAIN
1610 FOR I=1 TO 10: READ S$(I) ,T$(I): NEXT I
```



Kun näppäilet vastauksesi tietokoneeseen, älä käytä pilkkuja. Ne saavat mikron luulemaan, että pilkku on vastauksesi loppumerkki, joten se ei lue pilkun jälkeen tulevia sanoja.

```
1620 DATA HYVÄNEN AIKA, TYLSIMYS
1630 DATA NO NIIN , SAMMAKKO
1640 DATA KATSOTAANPAS, TIETOKONENERO
1650 DATA KUULEPAS, NEROPATTI
1660 DATA KATSOS, VALOPAA
1670 DATA TIEDÄTKÖ, TONTTU
1680 DATA NIIN, IDIOOTTI
1690 DATA KUULE, EINSTEIN
1700 DATA VOI EI, HIRVIÖ
1710 DATA VOI HYVÄ, MIKROHULLU
1720 RETURN
```

Kukin alkio muodostuu lauseen alkusanasta ja substantiivista. Nämä luetaan pareittain tilan säästämiseksi.

Ehdotuksia ohjelman muuttamista varten

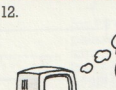
1. Helpoin tapa muuttaa ohjelmaa on muuttaa sanoja ja lauseita. Kannattaa tarkistaa jokaisen sanan sopivuus jokaiseen lauseeseen, jotta lopputulos on järkevä. Suomen kielen taivutus aiheuttaa ongelmia, mutta kepulikonsteilla voit yrittää kiertää tämän ongelman, kuten esimerkkitilauksessa on tehty. Jos lisäät ohjelmaan ylimääräisiä sanoja, muista muuttaa taulukoiden dimensioita, panna silmukat lukemaan kaikki datat ja muuttaa satunnaislukuja riveillä 610–650.

2. Voit myös muuttaa Q$:n sanoja, jotta tietokone etsii uudenlaisia sanoja. Keksi jokaiselle sanalle sopiva vastaus ja vastaukset oikeisiin paikkoihin M$:iin.

3. Saat mikron käyttämään satunnaislauseiden tuottajaa useammin, jos muutat rivin 320 luvun 6 pienemmäksi. Voit myös muuttaa taulukon C merkkien nollauksen harvemmin tai useammin tapahtuvaksi. Muuta rivin 440 lukua 12.

Itsepuhelu

Voit panna tietokoneen "juttelemaan" itsekseen lisäämällä ohjelmaan seuraavat rivit:



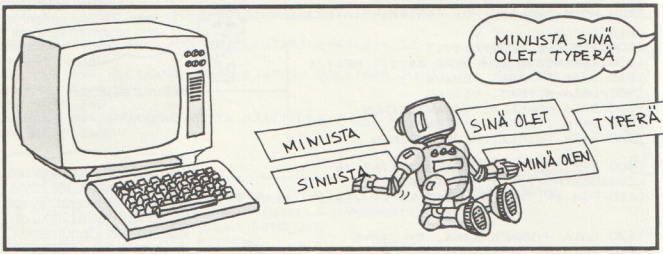

```
160 PRINT "KESKUSTELU VAI ITSEPUHELU?
(NÄPPÄILE K TAI I)"
170 INPUT K$

180 IF K$="I" THEN LET
D$="ROM": GOTO 600

470 IF K$="K" THEN GOTO 240

475 IF K$="I" THEN LET I$=R$
480 LET R$="": GOTO 310
```

D$ oli muuttuja, jossa oli sinun nimesi. Itsepuhelussa tietokone käyttää sanaa ROM omana nimenään ja menee riville 600 tuottamaan satunnaislauseen.

Tietokoneen vastauksesta (R$) tulee uusi puheenvuoro (I$). Sitten mikro palaa riville 310 valitsemaan vastaustavan.

Vastausrutiini

Tällä aukeamalla on ylimääräinen rutiini, jonka voit lisätä keskusteluohjelmaan. Se panee tietokoneen vastaamaan samoilla sanoilla, joita sinä käytit puheenvuorossasi. Sinclair tietokoneiden vaatimat muutokset on esitetty viereisen sivun alareunassa.



Vastausrutiini toimii samalla tavoin kuin sanojen tarkistusrutiini. Tarvitaan kaksi datataulukkoa U$ ja W$. U$:ssa on sanat, joita saatat käyttää puheenvuoroissasi ja W$:ssa on tietokoneen vastaukset. Jos käytät jotain U$:n sanaa, vastausrutiini panee sen tilalle vastaavan sanan W$:sta ja lisää tähän lauseen muut sanat.

```
135 DIM U$(10),W$(10)
138 DIM Z$(5)
```
Alustaa taulukot U$ ja W$ sekä taulukon Z$, jonne pannaan tietokoneen vastaukset.

```
155 GOSUB 2200
```
Menee aliohjelmaan lukemaan datat.

```
325 IF VASTAUS=7 THEN GOTO 2000
```
Ratkaisee, milloin vastausrutiinia käytetään.

```
2000 REM VASTAUSRUTIINI
```

```
2010 LET Z=0
```
Z on laskuri.

```
2020 LET P=LEN(I$)
```
LEN(I$) on puheenvuorosi merkkien määrä.

```
2030 FOR A=1 TO P
```
Silmukka, joka toistuu niin monta kertaa kuin puheenvuorossasi on merkkejä.*

```
2040 FOR B=1 TO 7
```
Silmukka, joka toistuu niin monta kertaa kuin W$:n pisimmässä alkiossa on merkkejä.

```
2050 LET L=LEN(U$(B))
```
Aina, kun silmukka B toistuu, U$:n seuraavan alkion pituus pannaan L:ään.

```
2060 IF MID$(I$,A,L)=U$(B) THEN
GOTO 2140
```
Vertaa puheenvuorosi merkkejä A–L U$:n sanaan, jonka alaindeksin arvo on B (B määräytyy silmukan perusteella). Jos sanat ovat samat, tietokone menee riville 2140.

```
2070 NEXT B: NEXT A
```
Aina, kun silmukka B toistuu, tietokone tarkistaa U$:n seuraavan sanan. Kun A-silmukka toistuu, se ottaa I$:sta seuraavan merkkisarjan.

```
2080 IF Z$(1)="" THEN GOTO 600
```
Palaa satunnaislauseiden tuottajaan, jos yhtään sanaa ei löydy.

*Jos sinulla on BBC mikro, lue sivun 141 huomautus.

```
2090 FOR J=1 TO Z
2100 PRINT Z$(J);
2110 LET Z$(J)=""
2120 NEXT J
```

Printtaa Z$:n kaikki vastaukset.

```
2130 LET R$=I$: GOTO 350
```

Panee lauseen mahdollisen loppuosan R$:iin ja palaa riville 365 printtaamaan R$:n.

```
2140 LET Z=Z+1
```

Z laskee Z$:n vastausten määrän.

```
2150 IF A>1 THEN LET Z$(Z)=LEFT$
(I$,A-1)+" "+W$(B)+" "
```

Muuttujan A asettaa rivin 2030 silmukka. A ilmoittaa, monesko I$:ssa olevan sanan merkki on ensimmäinen, jolle löytyy vastine U$:n sanasta. Jos A>1 rivi 2150 panee merkit Z$:ssa olevan sanan vasemmalle puolelle ja lisää W$:sta otetun vastauksen.

```
2160 IF A<2 THEN LET Z$(Z)=W$(B)+" "
```

Jos A<2 sana, jolle löytyy vastine, on I$:n alussa joten tietokone panee pelkän vastauksensa Z$:iin.

```
2170 LET I$=MID$(I$,A+L,P)
```

Panee loput lauseestasi Z$:iin ja palaa silmukoihin tarkistamaan, löytyykö muille sanoille vastineita.

```
2180 GOTO 2020
2200 REM VASTAUSRUTIININ DATAT
2210 FOR I=1 TO 10
2220 READ U$(I),W$(I)
2230 NEXT I
2240 DATA OLEN, OLET, OLET, OLEN
2250 DATA MINA, SINA, MINUSTA, SINUSTA
2260 DATA SINA, MINA, SINUN, MINUN
2270 DATA " EN "," ET "," ET "," EN "
2280 DATA " TE "," ME ", MINUN, SINUN
2290 RETURN
```

Tässä ovat U$:n ja W$:n data.

Vastausrutiini Sinclair tietokoneita varten

Lisää seuraavat rivit sekä Spectrumia että ZX81:tä varten. ZX81:ssä sinun on lisäksi käytettävä sivulla 142 esitettyä menetelmää, kun syötät datat keskusteluohjelmaan. Pane ZX81:n syöttösilmukka rivien 1000 ja 1720 väliin ja DIM-lauseet ennen riviä 1000.

```
135 DIM U$(10,7)
136 DIM W$(10,7)
137 DIM Z$(5,20)
2042 LET P$=""
2044 FOR I=1 TO LEN (U$(B))
2046 IF U$(B)(I TO I)<>"*" THEN LET P$=P$+U$(B)(I TO I)
2048 NEXT I
2050 LET L=LEN(P$)
2055 IF L+A-1>P THEN GOTO 2070
2060 IF I$(A TO A+L-1)=P$ THEN GOTO 2140
2080 IF Z=0 THEN GOTO 600
2150 IF A>1 THEN LET Z$(Z)=I$(TO A-1)+" "+W$(B)+" "
2160 IF A<2 THEN LET Z$(Z)=W$(B)+" "
2170 LET I$=I$(A+L TO)
2240 DATA "OLEN **","OLET","OLET **","OLEN"
2250 DATA "MINA **","SINA","MINUSTA","SINUSTA"
2260 DATA "SINA **","MINA","SINUN *","MINUN"
2270 DATA " EN ***","ET","ET ***","EN"
2280 DATA " TE ***","ME","MINUN *","SINUN"
2290 RETURN
```

Panee U$:n sanan P$:iin ja etsii *-symboleiden avulla sanan lopun

Jos P$ on pitempi kuin I$:n sana, palaa valitsemaan seuraavan sanan.

Jos I$ sana = P$ sana, menee riville 2140.

Jos sanoja ei löydy, menee saturnaislauseiden tuottajaan.

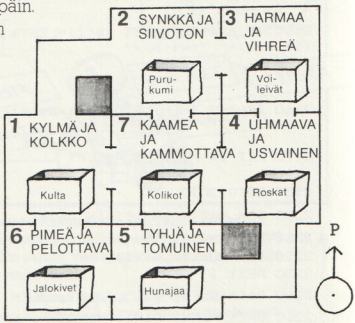
Jos A>1 panee I$:n alun merkit Z$:iin ja lisää W$:n sanan. Jos A<2 panee vain W$:n sanan Z$:iin.

Välilyönnit ovat tärkeä osa dataa, joten täytä jonot ★-symboleilla.

Kirjoitamme aarteenmetsästysohjelman

Seuraavilla sivuilla opastetaan askel askelelta, miten kirjoitetaan aarteenmetsästyspeli, jossa sinun pitää liikkua seitsemästä huoneesta muodostuvassa sokkelossa aarteita keräten. Ohjelma on aika mutkikas, joten sinun kannattaa noudattaa huolellisesti alla annettuja ohjeita. Ohjelman listaus on sivuilla 80–81. Jos et osaa itse ratkaista jotain ohjelman osaa, voit katsoa listauksista, miten se osa on ratkaistu ja siirtyä sitten ohjelmassa eteenpäin.

Jotta pystyt kirjoittamaan ohjelman, sinun pitää tietää, miten peliä pelataan. Lue siis ensin peliohjeita käsittelevä jakso.

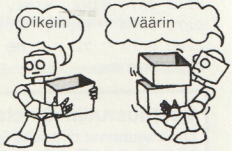


Peliohjeet

Oikealla olevassa kartassa on seitsemän huoneen muodostama sokkelo. Jokainen huone on erilainen ja jokaisessa huoneessa on laatikko, joka on täynnä aarteita. Kartan tiedot tallennetaan tietokoneen muistiin, mutta itse karttaa ei näy ruudulla. Siksi pelaajan pitää itse selvittää kunkin huoneen sijainti. Peli alkaa satunnaisesti valitusta huoneesta ja tavoitteena on saada kaikki aarteet yhteen huoneeseen pienellä siirtomäärällä.

```
TIETOKONE YMMARTAA NAMA SANAT

P,I,E,L    : MENE POHJOISEEN/ITAAN/ETELAAN TAI LANTEEN
OTTAA      : OTA AARRE MUKAASI
JATTAA     : JATA AARRE HUONEESEEN
ETSIA      : ILMOITTAA AARTEEN OLINPAIKAN
APUA       : ANTAA PELIOHJEET
```

Yllä olevasta näytöstä näet käskyt, joita pelissä voi käyttää ja niiden merkityksen. Kullakin siirrolla voit antaa vain yhden käskyn, joten sinulla ei ole varaa virheisiin.



Jotta peli olisi tarpeeksi vaikea, saat kantaa vain yhtä laatikkoa kerrallaan.

Näytteitä ohjelman ajosta

1

```
OLET HUONEESSA 4
SE ON UHMAAVA JA USVAINEN
SIINA ON ROSKAT

MITA HALUAT TEHDA ?
? OTTAA

OK. SINULLA ON MUKANA ROSKAT
```

2

```
OLET YHA HUONEESSA 4
MITA HALUAT TEHDA ?
? L

OK.
```

3

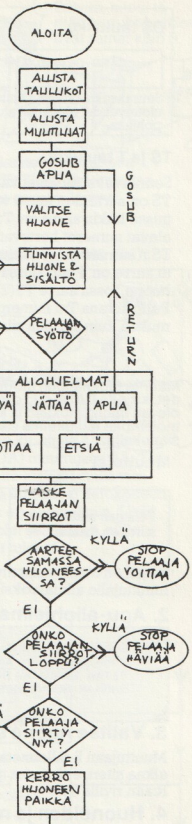
```
OLET HUONEESSA 7
SE ON KAAMEA JA KAMMOTTAVA
SIINA ON KOLIKOT

MITA HALUAT TEHDA ?
? JATTAA

OK. ROSKAT PANTU HUONEESEEN 7
```

4

```
OLET YHA HUONEESSA 7
MITA HALUAT TEHDA ?
? E

OK.
```

Ohjelman kirjoittaminen

Oikealla oleva kaavio on ns. vuokaavio. Siitä näkyy ohjelman rakenne. Jokainen päävaihe on esitetty järjestyksessä ylhäältä alas. Alla ja seuraavilla sivuilla annetut ohjeet ohjaavat sinut eri vaiheiden läpi. Joskus sinun käsketään hypätä ohjelman eri osiin toisten osien yli. Noudata ehdotettua rivinumerointia. Näin voit olla varma siitä, että kokoat ohjelman oikeassa järjestyksessä.



1 Taulukoiden alustus ja datan lukeminen
(rivit 100–250)

Sinun pitää panna kaikki viereisen sivun kartan tiedot tietokoneen muistiin. Tallenna kartta taulukoihin.

P, I, E, L taulukot

Ensiksi tarvitset neljä taulukkoa: P, I, E ja L. Näiden taulukoiden data kertoo tietokoneelle, mikä huone on pohjoiseen, itään, etelään ja länteen jostain toisesta sokkelon huoneesta katsoen. Määritä näiden taulukoiden ulottuvuudet ensimmäisellä rivillä (100)*. Kussakin on seitsemän alkiota.

	P	I	E	L
1	2	7	6	0
2	0	3	7	1
3				
4				
5				
6				
7				

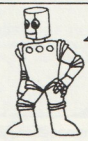
0 tarkoittaa, ettei siinä suunnassa ole huonetta. Esim. huoneen 1 länsipuolella tai huoneen 2 pohjoispuolella ei ole huonetta.

Piirrä tällainen taulukko, jotta saat selville ohjelman taulukkoon syötettävät datat. Oikealla olevat luvut ovat taulukoiden alaindeksejä ja tarkoittavat huoneita. Täytä taulukon rivit katsomalla viereisen sivun huonekartasta, missä ilmansuunnassa mikin huone on toisista huoneista katsottuna. Kaksi ensimmäistä riviä on täytetty malliksi.

Lisää näytön tyhjennyskäsky.

Datan lukeminen

```
110 FOR K=1 TO 7
120 READ N(K),E(K),S(K),W(K)
130 NEXT K

2000 DATA 2,7,6,0
```



Jos sinulla on ZX81 käytä erillistä INPUT-lausetta jokaista taulukkoa varten.

Seuraavaksi tarvitset yllä olevan kaltaisen silmukan, jolla luet datan taulukoihin. Pane piirtämäsi taulukon avulla data seitsemälle DATA-riville. Ensimmäinen on tehty yllä malliksi.

*Jos sinulla on ZX81, voit määrittää yhdellä rivillä vain yhden taulukon ulottuvuudet, joten aloita riviltä 10.

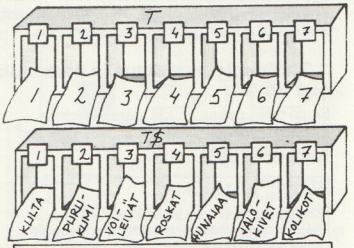
D$ taulukko

```
2100 DATA KYLMÄ JA KOLKKO,SYNKKÄ JA SIIVOTON
```

Seuraavaksi tarvitset taulukon nimeltä D$, johon panet huoneiden kuvaukset. Määritä sen ulottuvuudet ohjelman alussa ja kirjoita datat yllä olevan esimerkin mukaisesti kartan kuvausten pohjalta.

T$ ja T taulukot

Seuraavaksi tarvitset kaksi lisätaulukkoa. T$ on aarteiden nimiä varten ja T kertoo, missä kukin aarre on. T:n eri lokeroissa olevat numerot kertovat, missä huoneessa T$:n samannumeroisessa lokerossa mainittu aarre on. Pelin alussa esim. T(2) on 2 eli huone, jossa aarre T$(2) on (purukumi). Pelin aikana T:n numerot muuttuvat sitä mukaa kuin aarteita siirretään.



Määritä taulukoiden ulottuvuudet, kirjoita DATA-rivit oikealla olevan mallin mukaan ja lue data molempiin taulukoihin samalla silmukalla.

```
2200 DATA KULTA,1,PURUKUMI,2
2210 DATA VOILEIVÄT,3,ROSKAT,4
```

Muuttujat

M	C	F	WXY
Pelaajan tekemien siirtojen määrä.	Ilmoittaa, onko pelaajalla mitään mukanaan.	Lippumuuttuja. Toiminta selitetään myöhemmin.	Muuttujat, joissa on väliaikaista dataa.

Tässä on ohjelmassa myöhemmin tarvittavien muuttujien nimet. Anna jokaiselle muuttujalle alkuarvoksi 0 alkaen riviltä 300.

2. Apu-aliohjelma (rivit 1000–1120)

```
SOKKELOSSA ON 7 HUONETTA
JOKAISESSA ON AARRE. SINUN
PITÄA SAADA KAIKKI AARTEET
SAMAAN HUONEESEEN
```

Seuraavaksi tarvitset aliohjelman, joka selittää pelin säännöt. Pane riville 300 GOSUB-käsky ja aloita aliohjelma riviltä 1000. Pane aliohjelma printtaamaan vasemmalla näkyvät ohjeet sekä tietokoneen ymmärtämät sanat (s. 74).

3. Valitse huone satunnaisesti (rivi 350)

Muuttujaan R tallennetaan huone, jossa pelaaja on parhaillaan. Valitse huone, josta peli alkaa siten, että panet mikron tuottamaan satunnaisluvun väliltä 1–7 ja tallentamaan sen R:ään rivillä 350.

4. Huoneiden ja niiden sisällön tunnistus (rivit 400–470)

```
OLET HUONEESSA 7
SE ON KAAMEA JA
KAMMOTTAVA
SIINA ON KOLIKOT
```

```
OLET HUONEESSA 4
SE ON UHMAAVA JA USVAINEN
SIINA ON EI MITAAN
```

```
OLET HUONEESSA 1
SE ON KYLMÄ JA KOLKKO
SIINA ON KULTA
```

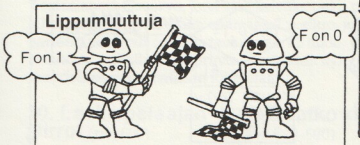
Kerro ensiksi pelaajalle, missä huoneessa hän on. Kuvaile sitten huonetta käyttäen R:ää D$:n alaindeksinä. Yllä on malleja siitä, miltä ruudun pitäisi näyttää.

Huoneen sisällön tunnistaminen

```
430 FOR K=1 TO 7
440 IF T(K)=R THEN PRINT T$(K):
LET F=1
450 NEXT K
460 IF F=0 THEN PRINT "EI MITAAN"
470 LET F=0
```

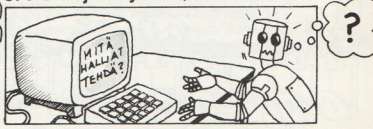
Huoneen R sisällön löydät etsimällä sen taulukosta T, jonne on tallennettu eri aarteita sisältävien huoneiden numerot. Voit käyttää silmukkaa (ks. yllä). IF/THEN rivi tarkistaa, onko jokin T:n numero sama kuin huoneen numero R. Jos vastaava numero löytyy, tietokone printtaa aarteen nimen ja asettaa muuttujan F 1:ksi.

Lippumuuttuja



Muuttuja F kertoo tietokoneelle, onko huoneessa yhtään aarretta sen jälkeen, kun se on tullut ulos silmukasta. F toimii lipun tavoin. Kun IF/THEN rivi on tosi, lippu nousee ylös (eli F on 1), mutta jos ehto ei ole tosi, F on 0. Muista nollata lippu rivillä 470, koska käytät sitä uudestaan.

5. Pelaajan syöttö (rivit 500–600)



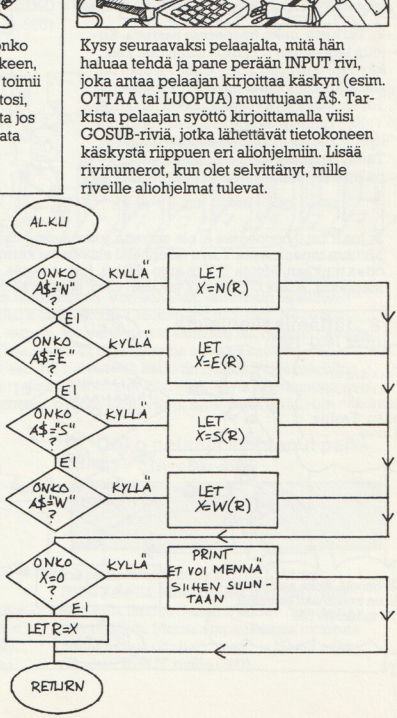
Kysy seuraavaksi pelaajalta, mitä hän haluaa tehdä ja pane perään INPUT rivi, joka antaa pelaajan kirjoittaa käskyn (esim. OTTAA tai LUOPUA) muuttujaan A$. Tarkista pelaajan syöttö kirjoittamalla viisi GOSUB-riviä, jotka lähettävät tietokoneen käskystä riippuen eri aliohjelmiin. Lisää rivinumerot, kun olet selvittänyt, mille riveille aliohjelmat tulevat.

6. Siirtymisaliohjelma

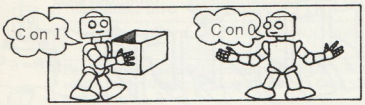
(rivit 1200–1260)

Yritä kirjoittaa siirtymisaliohjelma käyttäen apuna tätä vuokaaviota.

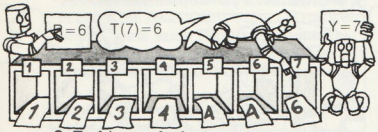
1. Kun pelaaja haluaa siirtyä, hän kirjoittaa P, I, E tai L. Tarkista, minkä kirjaimen hän näppäili. Käytä neljää IF/THEN riviä.
2. Etsi uuden huoneen numero. Käytä R:ää sopivan taulukon (P, I, E, L) alaindeksinä ja tallenna numero väliaikaiseen muuttujaan X.
3. Seuraavaksi sinun pitää tarkistaa X eli se, että pelaaja voi todella siirtyä haluamaansa suuntaan. Jos X on 0 ei tuossa suunnassa ole huonetta, joten sinun pitää printata viesti ja lähettää mikro GOTO käskyllä RETURN käskyyn aliohjelman loppuun.
4. Jos X ei ole 0, sinun pitää panna uusi huoneen numero R:ään panemalla R samaksi kuin X.
5. Pane RETURN aliohjelman loppuun.



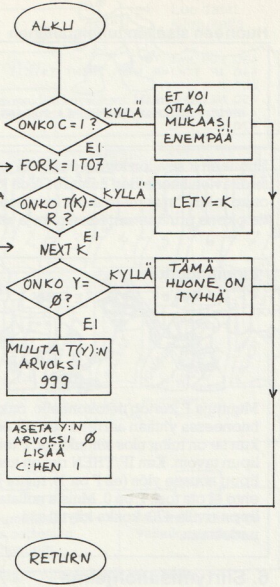
7. Ottamisaliohjelma (rivit 1300–1380)



1. Pelaaja voi kantaa vain yhden aarteen kerrallaan. Tarkista ensin, onko pelaajalla mukanaan mitään. Tämän voit tehdä testaamalla muuttuja C arvon. C saa arvon 1 jos pelaaja ottaa aarteen mukaansa. Jos C on 1 printtaa ruudulle viesti ja lähetä mikro RETURN riville.

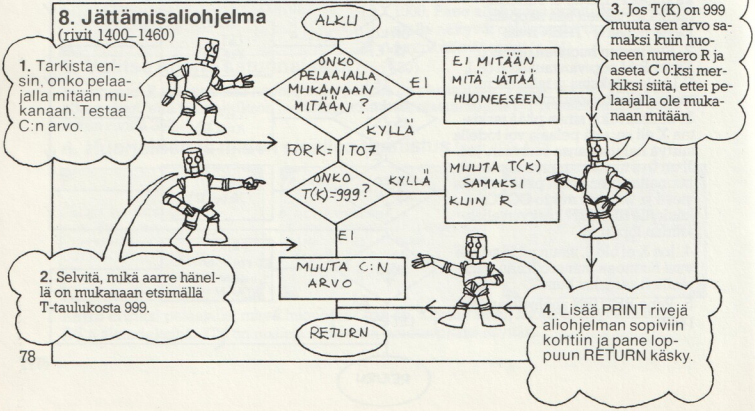


2. Tarkista, onko huoneessa aarretta. Kirjoita tätä varten IF/THEN silmukka, joka käy läpi taulukon T. Jos jokin T:n luku on sama kuin huoneen numero (R), tallenna T:n alaindeksi (K) väliaikaiseen muuttujaan Y. Kun mikro ajaa silmukan, se voi löytää useita lukuja, jotka ovat samoja kuin R. Tässä tapauksessa Y:ssä on silmukan päättyessä viimeisen T-numeron alaindeksi.



3. Jos Y on 0 huoneessa ei ole aarretta, joten sinun pitää lähettää mikro RETURN riville. Muussa tapauksessa Y:hyn tallennettu alaindeksi kertoo mikrolle, mikä aarre sen tulee ottaa mukaan. Muuta T(Y):n arvo 999:ksi, joka osoittaa, että aarre on nyt mukana matkassa. Aseta C 1:ksi ja Y takaisin 0:ksi, ja pane aliohjelman loppuun RETURN.

8. Jättämisaliohjelma (rivit 1400–1460)



9. Etsimisaliohjelma (rivit 1500–1590)

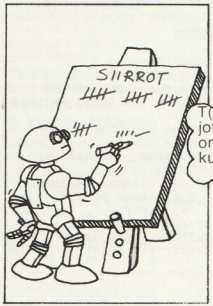
Tämä aliohjelma ilmoittaa pelaajalle aarteiden nykyisen olinpaikan. Yritä kirjoittaa aliohjelma alla annettujen ruutunäyttöjen perusteella. Suunnittele ensin aliohjelmasta vuokaavio.

```
SINULLA ON MUKANASI : KULTA
HUONEIDEN SISALTO:
  2:PURUKUMI
  3:VOILEIVAT
  4:ROSKAT
  7:HUNAJAA
  6:JALOKIVET
  7:KOLIKOT
```

```
SINULLA ON MUKANASI : EI MITAAN
HUONEIDEN SISALTO:
  7:KULTA
  7:PURUKUMI
  7:VOILEIVAT
  7:ROSKAT
  7:HUNAJAA
  6:JALOKIVAT
  7:KOLIKOT
```

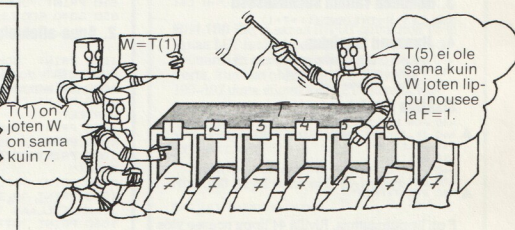
Testaa ensin C:n arvo, jotta saat selville, onko pelaajalla mitään mukanaan. Jos hänellä on, lähetä mikro silmukkaan etsimään T:n alkio, joka on 999, printtaa pelaajan kantaman aarteen nimi valitsemalla taulukosta T$ alkio, jolla on sama alaindeksi. Printtaa huoneiden sisältö silmukalla, joka printtaa aarteiden nykyisen sijainnin (mutta ei aarretta, jota mahdollisesti parhaillaan kannetaan).

10. Laske pelaajan siirrot (rivi 600)



Lisää muuttujaan M 1 rivillä 600. Näin lasket pelaajan siirtojen määrän.

11. Ovatko kaikki aarteet samassa huoneessa? (rivit 610–690)



Pelaaja voittaa pelin jos hän on siirtänyt kaikki aarteet samaan huoneeseen. Voit selvittää, onko näin tapahtunut testaamalla taulukkoon T tallennetut numerot. Ne ovat kaikki samoja, jos aarteet ovat samassa huoneessa. Ota yksi alkio T:stä, esim. T(1), ja tallenna se väliaikaiseen muuttujaan W. Vertaa sitä kaikkiin muihin T:n numeroihin yksi kerrallaan silmukkaa käyttäen. Käytä lippumuuttujaa ilmoittamaan, ovatko kaikki liput samoja vai eivät.

12. Onko pelaaja käyttänyt kaikki siirtonsa? (rivit 700–730)

Valitse, montako siirtoa pelaaja saa käyttää. Jos M on suurempi kuin valitsemasi luku, pelaaja häviää pelin. Jos näin käy, printtaa sopiva viesti ja lopeta ohjelma.

13. Onko pelaaja vaihtanut paikkaansa? (rivit 800–850)

```
OLET YHA HUONEESSA 7
MITA HALUAT TEHDA ?
?
```

Jos pelaaja kirjoitti P, I, E, tai L A$:iin INPUT rivillä, lähetä hänet takaisin riville 400, jolla ilmoitat hänelle uuden huoneen ja sen sisällön. Muussa tapauksessa muistuta häntä hänen olinpaikastaan ja lähetä mikro takaisin INPUT riville (500).

Aarteenmetsästysohjelman listaus

Tässä on ohjelman täydellinen listaus jakso kerrallaan ja oikeassa järjestyksessä esitettynä. Sinclair tietokoneiden tarvitsemat muutokset on annettu viereisellä sivulla.

1. Taulukoiden alustus ja datan lukeminen

```
100 DIM N(7),E(7),S(7),W(7),D$(7),
T$(7),T(7)
110 FOR K=1 TO 7
120 READ P(K),I(K),E(K),L(K)
130 NEXT K
140 FOR K=1 TO 7
150 READ D$(K)
160 NEXT K
170 FOR K=1 TO 7
180 READ T$(K),T(K)
190 NEXT K
200 LET M=0
210 LET C=0
220 LET F=0
230 LET W=0
240 LET X=0
250 LET Y=0
260 CLS
```

Data on rivillä 2000.

2. Apua-aliohjelma

```
300 GOSUB 1000
```

Aliohjelma on rivillä 1000.

3. Huoneen valinta satunnaisesti

```
350 LET R=INT(RND(1)*7+1)
```

4. Huoneen tunnistus

```
400 PRINT "OLET HUONEESSA ";R
410 PRINT "SE ON ";D$(R)
420 PRINT "SIINA ON ";
```

Sisällön tunnistus

```
430 FOR K=1 TO 7
440 IF T(K)=R THEN PRINT TAB(15);
T$(K):LET F=1
450 NEXT K
460 IF F=0 THEN PRINT TAB(15)"NOTHING"
470 LET F=0
480 PRINT:PRINT
```

F on lippumuuttuja. Rivillä 44 lippu nousee ylös (F = 1) kun yksi T-taulukon numeroista on sama kuin huoneen numero R.

5. Pelaajan syöttö

```
500 PRINT "MITA HALUAT TEHDA?"
510 INPUT A$
520 IF A$="APUA" THEN GOSUB 1000
530 IF A$="P" OR A$="I" OR A$="E" OR
A$="L" THEN GOSUB 1200
540 IF A$="OTTAA" THEN GOSUB 1300
550 IF A$="JATTAA" THEN GOSUB 1400
560 OF A$="ETSIA" THEN GOSUB 1500
```

Rivit 520–560 lähettävät mikron eri aliohjelmaan riippuen sanasta, jonka käyttäjä näppäilee muuttujaan A$ (rivi 510).

10. Laske pelaajan siirrot

```
600 LET M=M+1
```

11. Ovatko kaikki aarteet samassa huoneessa?

```
610 LET W=T(1)
620 FOR K=2 TO 7
630 IF W<>T(K) THEN LET F=1
640 NEXT K
650 IF F=1 THEN GOTO 690
660 PRINT "HIENOA. KERASIT KAIKKI AARTEET"
670 PRINT "HUONEESEEN ";R;" ";M;" SIIRROLLA"
680 STOP
690 LET F=0
```

Rivillä 610 T(1):teen tallennettu luku pannaan muuttujaan W. Silmukassa (rivi 630) mikro vertaa W:tä kaikkiin muihin T:n numeroihin. Jos jokin T:n numero ei ole sama kun W mikro asettaa lippumuuttujan F arvoksi 1. Jos kaikki T:n numerot ovat samoja kuin W F = 0 mikä tarkoittaa, että kaikki aarteet ovat samassa huoneessa.

12. Onko pelaaja käyttänyt kaikki siirtonsa?

```
700 IF M<=28 THEN GOTO 730
710 PRINT "SIIRROT LOPPUNEET"
720 STOP
730 PRINT
```

Jos M on enemmän kuin 29 pelaaja on käyttänyt kaikki siirtonsa ja ohjelma pysähtyy. Halutessasi voit muuttaa pelaajan siirtojen enimmäismäärää.

13. Onko pelaaja vaihtanut paikkaansa?

```
800 IF A$="P" OR A$="I" OR A$="E" OR
A$="L" THEN GOTO 400
810 PRINT "OLET YHA HUONEESSA ";R
850 GOTO 500
```

2. Apua-aliohjelma

```
1000 PRINT "SOKKELOSSA ON 7
HUONETTA"
1010 PRINT "JOKAISESSA ON AARRE."
1020 PRINT "SINUN PITAA SAADA KAIKKI
AARTEET"
1030 PRINT "SAMAAN HUONEESEEN"
1040 PRINT
1050 PRINT "TIETOKONE YMMARTAA NAMA
SANAT"
1060 PRINT "APUA     :ANTAA PELIOHJEET"
1070 PRINT "P,I,E,L:MENE POHJOISEEN/
ITAAN/ETELAAN TAI LANTEEN"
1080 PRINT "OTTAA    :OTA AARRE MUKAASI"
1090 PRINT "JATTAA   :JATA AARRE HUONEESEEN"
1100 PRINT "ETSIA    :ILMOITTAA AARTEEN
OLINPAIKAN"
```

```
1110 PRINT:PRINT
1120 RETURN
```

Mikro kertoo, miten peli toimii ja mitä sanoja se ymmärtää. Aina, kun pelaaja syöttää sanan APUA mikro menee tähän aliohjelmaan.

6. Siirtymisaliohjelma

```
1200 IF A$="P" THEN LET X=P(R)
1210 IF A$="I" THEN LET X=I(R)
1220 IF A$="E" THEN LET X=E(R)
1230 IF A$="L" THEN LET X=L(R)
1240 IF X=0 THEN PRINT "ET VOI MENNA
SIIHEN SUUNTAAN":GOTO 1260
1250 LET R=X
1260 RETURN
```

Riveillä 1200–1230 tietokone selvittää, minkä suunnan pelaaja näppäili (P, I, E, L). Sitten se etsii valitussa suunnassa olevan huoneen numeron käyttäen R:ää sopivan taulukon alaindeksinä ja tallentaa numeron X:ään. Jos X = 0 (rivi 1240) ei tuossa suunnassa ole huonetta.

7. Ottamisaliohjelma

```
1300 IF C=1 THEN PRINT "ET VOI OTTAA MUKAAN
ENEMPAA":GOTO 1370
1310 FOR K=1 TO 7
1320 IF T(K)=R THEN LET Y=K
1330 NEXT K
1340 IF Y=0 THEN PRINT "TAMA HUONE ON
TYHJA"
1350 LET T(Y)=999
1360 PRINT "Ok. SINULLA ON
MUKANA ";T$(Y)
1370 LET C=1:LET Y=0
1380 RETURN
```

Rivillä 1320 tietokone vertaa jokaista taulukon T numeroa R:ään. Jos numerot ovat samat, se ottaa kyseisen T-alkion alaindeksin ja panee sen muuttujaan Y. Silmukan jälkeen mikro valitsee Y:n avulla oikean alkion T:stä ja muuttaa sen arvoksi 999 (rivi 1350). Jos Y = 0 (rivi 1340) huone on tyhjä.

8. Jättämisaliohjelma

```
1400 IF C=0 THEN PRINT "SINULLA EI OLE
MUKANA MITAAN": GOTO 1450
1410 FOR K=1 TO 7
1420 IF T(K)=999 THEN PRINT T$(K);
"PANTU HUONEESEEN ";R
1430 IF T(K)=999 THEN LET T(K)=R
1440 NEXT K
1450 LET C=0
1460 RETURN
```

Tietokone etsii T:n alkion, joka on yhtä suuri kuin 999 (rivi 1420). Sitten se muuttaa sen arvon R:ksi (rivi 1430).

9. Etsimisaliohjelma

```
1500 PRINT "SINULLA ON MUKANASI :";
1510 IF C=0 THEN PRINT TAB(10);
"EI MITAAN":GOTO 1550
1520 FOR K=1 TO 7
1530 IF T(K)=999 THEN PRINT
TAB(10);T$(K)
1540 NEXT K
1550 PRINT "HUONEIDEN SISALTÖ: "
1560 FOR K=1 TO 7
1570 IF T(K)<>999 THEN PRINT
T(K);" : ";T$(K)
1580 NEXT K
1590 RETURN
```

1 Data

```
2000 DATA 2,7,6,0
2010 DATA 0,3,7,1
2020 DATA 0,0,4,2
2030 DATA 3,0,5,7
2040 DATA 7,4,0,6
2050 DATA 1,5,0,0
2060 DATA 2,4,5,1
```

Nämä ovat taulukoiden P, I, E ja L datat.

```
2100 DATA KYLMÄ JA KOLKKO,SYNKKÄ JA
SIIVOTON
2110 DATA HARMAA JA HIRVEÄ,UHMAAVA JA
USVAINEN
2120 DATA TYHJÄ JA TOMUINEN,PIMEÄ JA
PELOTTAVA,KAAMEA JA KAMMOTTAVA
```

Tässä ovat D$:n datat.

```
2200 DATA KULTA,1,PURUKUMI,2
2210 DATA VOILEIVAT,3,ROSKAT,4
2220 DATA HUNAJAA,5,JALOKIVET,6,
KOLIKOT,7
```

Tässä ovat T$:n ja T:n datat.

Sinclairin muutokset

Spectrumiin (Timex 2000) ei tarvita muita muutoksia kuin se, että merkkijonotaulukoiden ulottuvuudet pitää ilmoittaa samalla tavalla kuin ZX81:ssä (Timex 1000). DIM-lauseet pitää erottaa toisistaan kaksoispisteillä tai panna omille riveilleen kuten ZX81:ssä.

Rivi 100 Pane DIM-lauseet omille riveilleen ja kirjoita ulottuvuuksiksi D$(7,20) ja T$(7,9).

Rivi 120 Lisää erilliset INPUT rivit kullekin taulukolle ja näppäile datat sisään, kun ajat ohjelman.

```
120 INPUT P(K)
123 INPUT I(K)
125 INPUT E(K)
127 INPUT L(K)
```

Rivi 150 Käytä sanaa INPUT ja näppäile datat sisään, kun ajat ohjelman.

```
150 INPUT D$(K)
180 INPUT T$(K)
```

Rivi 180 Käytä sanaa INPUT taulukkoa T$ ja sanaa LET taulukkoa T varten, jotta et joudu syöttämään aarteiden alkupaikkoja joka kerta, kun ajat ohjelman. Käytä riveillä 193–197 uutta silmukkaa LET-lausetta varten:

```
193 FOR K=1 TO 7
195 LET T(K)=K
197 NEXT K
```

Rivi 350 Käytä mikrosi RND-käskyä.

```
350 LET R=INT(RND*7+1)
```

Rivi 440 Käytä erillisiä rivejä ja toista IF/THEN käsky:

```
440 IF T(K)=R THEN PRINT TAB(15);T$(K)
445 IF T(K)=R THEN LET F=1
```

Rivit 480 ja 1110 Käytä erillisiä rivejä.

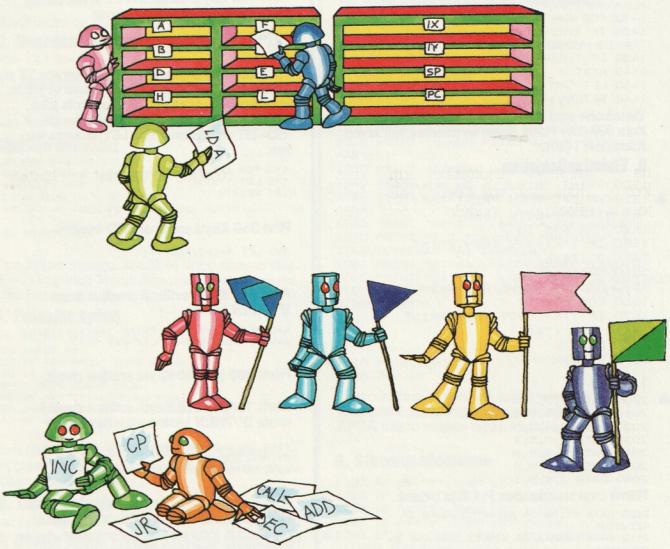
Rivit 1240–1510 Käytä erillisiä rivejä ja toista IF/THEN käskyt uudella rivillä:

```
1240 IF X=0 THEN PRINT "ET VOI MENNA
SIIHEN SUUNTAAN"
1245 IF X=0 THEN GOTO 1260
```

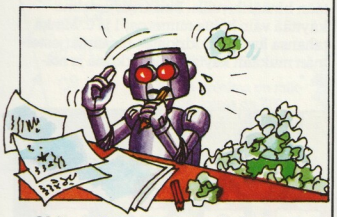
Rivi 2000 Poista rivit 2000–3000 ja lisää nämä rivit, jotta et joudu syöttämään datoja joka kerta, kun ajat ohjelman. Kun haluat pelata peliä uudestaan, näppäile GOTO 2000. (Jos painat RUN, mikro tyhjentää kaikki muuttujat ja taulukot.)

```
2000 PRINT "HALUATKO PELATA UUDESTAAN"
2010 INPUT R$
2020 IF R$="KYLLÄ" THEN GOTO 193
2030 IF R$="EI" THEN STOP
2040 GOTO 2000
```

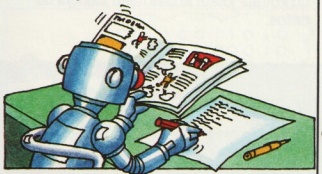
Konekielen alkeet



Konekieli on kieli, jolla tietokone tekee työnsä. Konekielellä kirjoitetut ohjelmat ovat paljon nopeampia ja ne vievät vähemmän muistia kuin BASIC-kieliset ohjelmat. Mutta konekielisen ohjelman kirjoittaminen ja ymmärtäminen on paljon vaikeampaa kuin BASIC-kielisen ohjelman. Kirjan seuraava osa opastaa kädestä pitäen konekielellä ohjelmointiin.

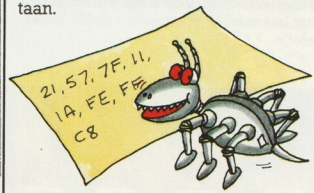


Ohjeet johdattavat sinut helpoin askelin konekielen perusteisiin. Niissä näytetään, miten kirjoitetaan helppoja konekielisiä ohjelmia – esim. sellaisia, joissa lasketaan kaksi lukua yhteen tai välkytetään ruudulla jotain viestiä – sekä neuvotaan, miten konekielisiä ohjelmia luetaan tietokoneeseen ja miten niitä ajetaan.

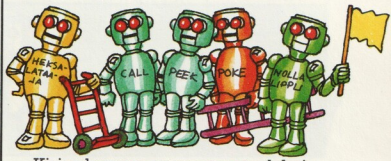


Konekieliset ohjelmat on kirjoitettu tietokoneille, joissa on Z80 tai 6502 mikroprosessori eli suoritin.* Prosessori on lastu, jossa on tietokoneen keskusyksikkö. Erilaista prosessoria käyttävät mikrot käyttävät myös erilaista konekieltä. Mutta kaikki mikrot, joissa on sama prosessori, käyttävät samaa konekieltä.

Konekieli on vaikeaa ja vaivalloista. Siinä pitää noudattaa monia sääntöjä ja muistaa paljon pieniä yksityiskohtia. Älä välitä, vaikka konekieli tuntuisikin aluksi vaikealta. Konekieli on hämmentävän näköistä, koska et heti osaa lukea sitä etkä ymmärrä sillä kirjoitettuja ohjelmia, nehän ovat pelkkiä kirjain- ja numerojonoja. Virheiden löytäminenkin on vaikeaa ja niiden seuraukset ovat tuhoisat. Konekielen kanssa pitää olla erittäin huolellinen ja tarkistaa kaikki asiat kahteen tai kolmeen kertaan.



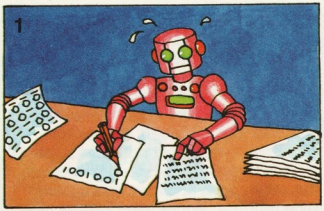
Pitkiä ohjelmia ei kannata kirjoittaa konekielellä, ellet ole todella innostunut sen käytöstä. Jotkut asiat voit hoitaa aivan yhtä hyvin BASICilla. Joissain tehtävissä konekielen käyttö on kuitenkin välttämätöntä – esimerkiksi peliohjelmien nopeuttajana tai fantastisten erikoistehosteiden luomisessa. Sivulla 120 kerrotaan, miten voit tehdä ohjelmistasi kiintoisampia käyttämällä BASIC-ohjelmissa lyhyitä konekielisiä aliohjelmia.



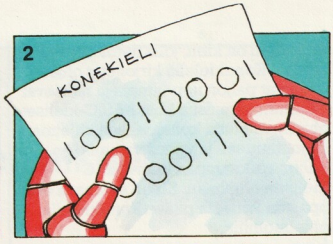
Kirjan lopussa on muunnostaulukoita, joista on apua, kun kirjoitat konekieltä, sekä konekielen sanasto, jossa on selitetty hankalia termejä. Seuraavilla sivuilla on lisäksi paljon arvoituksia ja lyhyitä ohjelmaehdotuksia; näiden vastaukset löytyvät sivulta 124.

*Spectrum ja ZX81 (Timex 2000 ja 1000) sekä MSX-mikrot käyttävät Z80-suoritinta ja VIC 20, BBC, Atarit ja Oric käyttävät 6502-suoritinta. Commodore 64 käyttää 6510-suoritinta ja ymmärtää 6502-konekieltä.

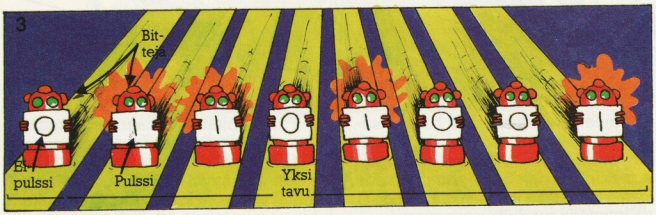
Mitä konekieli on?



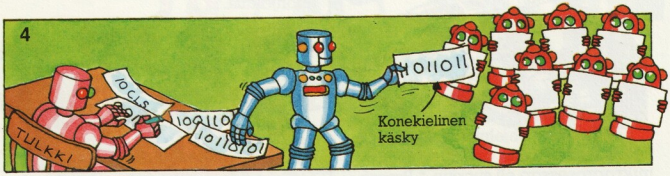
Konekieli on kieli, jolla tietokone suorittaa tehtävänsä. Kun annat tietokoneelle BASIC-kielisen ohjelman, kaikki sen käskyt ja datat käännetään tietokoneen sisällä konekielelle.



Konekielessä kaikki käskyt ja tiedot esitetään binääriluvuilla. Binäärijärjestelmä käyttää vain kahta numeroa, 1 ja 0. Minkä tahansa luvun voi kirjoittaa binäärijärjestelmän mukaan käyttämällä ykkösiä ja nollia.*



Tietokoneen sisällä binäärinumerot esitetään sähköpulsseina. Pulssi tarkoittaa 1 ja pulssin puuttuminen tarkoittaa 0. Pulsseja ja ei-pulsseja sanotaan biteiksi. Bitit virtaavat tietokoneen sisällä kahdeksan ryhmissä. Tällaista ryhmää sanotaan tavuksi (engl. byte). Jokainen pulsseista ja ei-pulsseista muodostuva tavu on binääriluku, joka tarkoittaa jotain konekielen käskyä tai tietoa.



Kaikissa tietokoneen tekemissä töissä – esim. kun se laskee kaksi lukua yhteen tai tyhjentää näytön – tarvitaan useita konekielisiä käskyjä. Kun annat tietokoneelle BASIC-käskyn, ohjelma nimeltä tulkki kääntää käskysi konekielelle eli käskyikisi, joita tietokone ymmärtää.

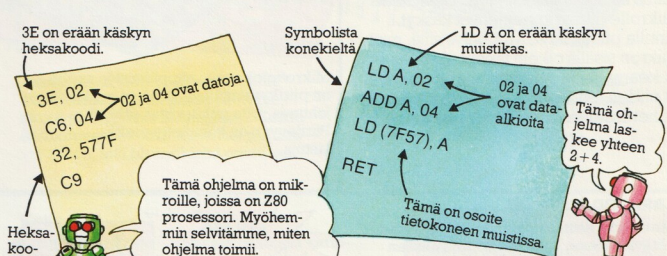
Konekielellä kirjoitetaan myös suoraan ohjelmia, jotka ovat jo alun perin paljon lähempänä tietokoneen kieltä kuin BASIC. Konekielisessä ohjelmassa tietokoneelle on annettava kaikki erilliset käskyt, jotka tietokone tarvitsee esim. tyhjentääkseen näytön.

*Sivulla 108 kerrotaan lisää binäärijärjestelmästä.

Konekielinen ohjelmointi

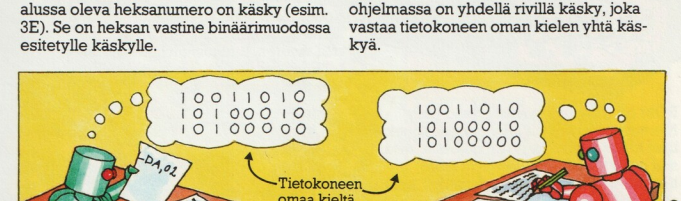
Konekielisiä ohjelmia voi kirjoittaa useilla eri tavoilla. Voit kirjoittaa kaikki käskyt binäärilukuina, mikä on erittäin vaivalloista. Toinen mahdollisuus on käyttää numerojärjestelmää nimeltä heksadesimaali eli heksa. Totuttelun jälkeen heksan käyttö on paljon helpompaa kuin binäärijärjestelmän käyttö.

Konekielisiä ohjelmia voi myös kirjoittaa kielellä, jonka nimi on symbolinen konekieli. Symbolisessa konekielessä kutakin käskyä esitetään ns. muistikkaalla, lyhyellä symbolisella sanalla, jonka nimi tuo mieleen vastaavan englanninkielisen käskyn.



Tässä on osa konekielistä ohjelmaa, joka on kirjoitettu heksadesimaalikoodeilla. Heksa on kuusitoistakantainen lukujärjestelmä, jossa käytetään symboleita 0-9 ja A-F esittämään numeroita 0–15. (Myöhemmin saat lisää tietoa heksasta.) Kunkin rivin alussa oleva heksanumero on käsky (esim. 3E). Se on heksan vastine binäärimuodossa esitetylle käskylle.

Tässä on sama ohjelma symbolisella konekielellä. Jokaisella rivillä on yhden käskyn muistikas, joka vastaa saman rivin heksakoodia vasemmanpuoleisessa kuvassa. Esim. muistikas LD A (load A) tarkoittaa samaa kuin heksan 3E. Kummassakin ohjelmassa on yhdellä rivillä käsky, joka vastaa tietokoneen oman kielen yhtä käskyä.



Kun annat tietokoneelle symbolisella konekielellä laaditun ohjelman, tarvitset lisäohjelman nimeltä assembler, joka kääntää muistikkaan tietokoneen kielelle. Joissain tietokoneissa on sisäänrakennettu kääntäjä, joihinkin se taas pitää ostaa kasetilla ja ladata siltä muistiin. Toinen vaihtoehto on kirjoittaa konekielinen ohjelma symbolisen konekielen muistikkailla (ne on helpompi oppia ulkoa kuin numerot) ja kääntää ne heksaan ennen kuin annat ne mikrolle. Jotkut mikrot hyväksyvät heksanumeroita, joillekin taas pitää ensin antaa lyhyt ohjelma nimeltä heksalataaja, joka kääntää käskyt tietokoneelle. Sivulla 104 on heksalataajaohjelma, jolla voit ladata tämän kirjan konekieliset ohjelmat.

Tutustumme tietokoneen sisuksiin

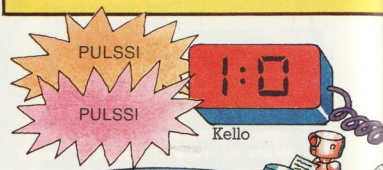
Kun ohjelmoit konekielellä, sinun pitää kertoa tietokoneelle täsmälleen, mitä sen pitää eri vaiheissa tehdä: mistä löydät datat, miten printata ruudulle jne. (Kun käytät BASICia, tietokoneen sisällä olevat erikoisohjelmat hoitavat nämä asiat puolestasi.) Jotta osaat antaa mikrolle oikeat konekieliset käskyt, sinulla pitää olla selvä käsitys siitä, mitä mikron sisällä oikein tapahtuu. Tällä aukeamalla on selitetty mikron eri osat ja niiden tehtävä. Seuraavilla sivuilla on annettu aiheesta tarkempaa tietoa.

Tietokoneen sisukset

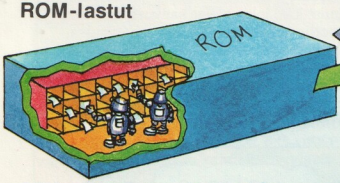
Mikrotietokoneen näppäimistön sisällä on piirikytkentälevy. Siinä on metallisia johtimia, joita pitkin virtaa sähköä. Piirilevyyn on kiinnitetty useita lastuja.

Lastujen tehtävät

Tämä kuva esittää tietokoneen eri lastujen tehtäviä. Viestit kulkevat lastusta toiseen tavuina eli kahdeksan bitin ryhminä, jotka edustavat käskyjä ja dataa.



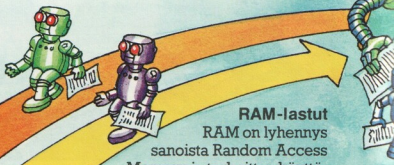
ROM-lastut



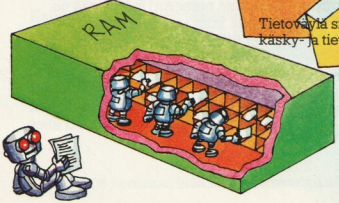
Osoiteväylä siirtää muistiosoitteita.

Konekielen tavut virtaavat lastusta toiseen piirilevyn johtimia myöten. Johtimia on kolmenlaisia eri tehtäviä suorittavia tavuja varten. Kutakin lajia sanotaan väyläksi.

ROM on lyhennys sanoista Read Only Memory ja tarkoittaa lukumuistia. Konekieliset käskyt, jotka kertovat mikrolle, mitä sen pitää tehdä, on tallennettu ROM-lastuun. Sitä sanotaan lukumuistiksi, koska mikro voi ainoastaan lukea ROM:issa olevia tietoja. Se ei pysty tallentamaan sinne mitään. Yleensä mikrotietokoneiden tulkki (ohjelma joka kääntää BASICin konekielelle) on ROMissa.

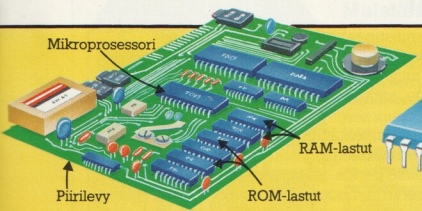


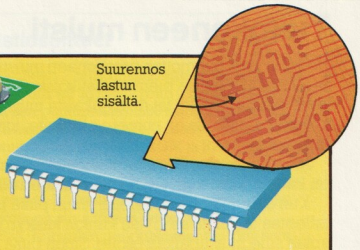
Tietoväylä siirtää käsky- ja tietotavuja.



RAM-lastut

RAM on lyhennys sanoista Random Access Memory ja tarkoittaa käyttömuistia eli suorasaantimuistia. Tänne tallennetaan tietokoneelle antamasi ohjelmat siksi aikaa, kun tietokone käsittelee niitä. RAMia sanotaan suorasaantimuistiksi, koska tietokone voi hakea suoraan minkä tahansa tiedon mistä tahansa muistin kohdasta. Kun katkaiset mikrosta virran, RAMissa olevat tiedot joutuvat hukkaan.

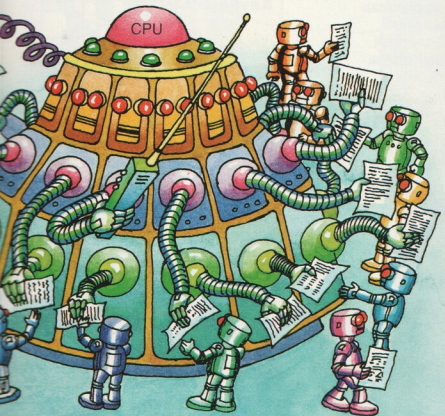




Lastun oikea nimi on integroitu piiri, sillä jokaisen lastun sisällä on mikroskooppisen pieniä sähköpiirejä. Kaikki tietokoneen tekemät toiminnot perustuvat siihen, että binäärikieltä edustavat virtapulssit kulkevat lastujen piireissä. Tehtävän mukaan on olemassa erilaisia lastuja. Alla olevassa kuvassa on selitetty eri tyyppisten lastujen tehtävä.

Kello

Kellon kvartsikide värähtelee miljoonia kerto-ja sekunnissa ja säätelee pulssien virtaa tietokoneessa.



Mikroprosessori

Prosessorilastulla on tietokoneen keskusyksikkö. Siellä tietokone tekee kaiken työnsä. Keskusyksikkö suorittaa laskutoimituksia, vertailee datoja, tekee päätöksiä ja ohjaa mikron kaikkia muita toimintoja. Keskusyksikön toimintaohjeet ovat ROM-lastulla.

Tietokoneesi mikroprosessori

Mikroprosessorilastuja on eri merkkisiä ja jokaisessa mikrossa on jonkinlainen prosessori. Yleisimmät prosessorit ovat Z80, jollainen on Sinclair (Timex)-tietokoneissa, sekä 6502, jota käyttävät Oric, BBC, VIC 20 ja monet muut. Muita prosessoreita ovat 6809, jollainen on Dragonissa, ja TI-99/4:n 9900.

Erimerkkiset mikroprosessorit käyttävät erilaisia konekielisiä käskyjä, mutta kaikki mikrot, joissa on sama prosessori käyttävät samaa konekieltä. Samasta lastusta on myös olemassa useita eri versioita. Esim. Z80A on nopeampi versio Z80:sta ja 6502A ja 6510 (jota käytetään Commodore 64:ssä) ovat 6502:n versioita. Silti niissä käytetään samoja konekielen käskyjä kuin perustana olleessa lastussa. Tämä kirja on kirjoitettu erityisesti tietokoneille, joiden mikroprosessorit ymmärtävät Z80 tai 6502 konekieltä.

Tietokoneen muisti

Helpoin tapa hahmottaa tietokoneen muistin rakenne on ajatella, että se muodostuu monista pienistä laatikoista, joihin kuhunkin mahtuu yksi tavu eli yksi konekielinen käsky tai data. Jokaista muistin lokeroa sanotaan paikaksi ja jokaisella paikalla on numero eli osoite, jonka avulla tietokone löytää sen muistista.

Muistin eri alueille tallennetaan erilaisiin tehtäviin käytettyä tietoa. Muistikartta on kaavio, josta näkyy kunkin eri alueen alkukohta.

Kun ohjelmoit konekielellä, sinun pitää kertoa tietokoneelle, minne sen pitää tallentaa tai mistä se löytää kunkin käskyn tai tiedon. Toisin sanoen sinun pitää antaa sille muistipaikan osoite. Sinun pitää kertoa tietokoneelle sekin, minne se tallentaa varsinaisen konekielisen ohjelman, joten sinun pitää opetella tuntemaan tietokoneesi muistikartta.

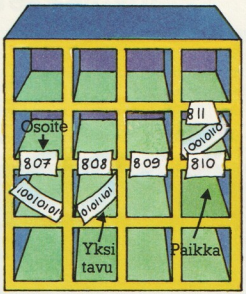
Muistikartta

Oikealla oleva kuva esittää kotitietokoneen muistikarttaa. Oman mikrosi käsikirjassa pitäisi olla sen muistikartta. Eri merkkisissä mikroissa muisti on jaettu eri lailla, joten oman mikrosi kartta ei ole samanlainen kuin kuvamme.

Muistikartta voidaan piirtää joko sarakkeen muotoon, kuten oikealla on tehty tai vaakatasoon. Kartassa on ilmoitettu, mistä osoitteesta kukin muistin eri alueista alkaa. Osoite voi olla desimaaliluku tai heksaluku tai molemmat, kuten tässä. Tässä kirjassa heksaluvut merkitään &-merkillä, joka pannaan luvun alkuun. Mikrosi käsikirjassa käytetään kenties eri symbolia, esim. $, % tai #.

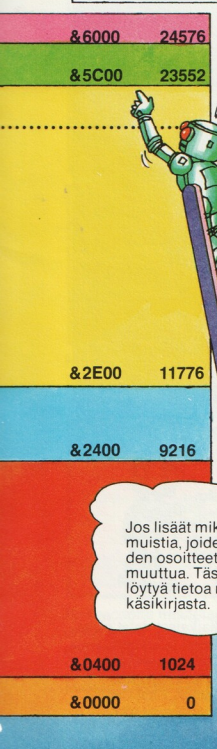
BASIC
Tällä alueella on tulkki eli ohjelma, joka muuttaa BASIC-kieliset käskyt tietokoneen binäärikielelle.

Käyttöjärjestelmä
Tällä alueella on ohjelmia, joita sanotaan käyttöjärjestelmäksi eli valvontaohjelmiksi. Ne kertovat tietokoneelle, miten sen tulee toimia. Nämä ohjelmat on kirjoitettu konekielellä. Mukana on ohjelmia, jotka ohjaavat laskutoimituksia, tyhjentävät näytön, etsivät saturnaislukuja, tarkistavat, onko jotain näppäintä painettu ja hoitavat kaikki muut tietokoneen tehtävät.



Muistiosoitteet

Tietokoneen sisällä muistiosoitteet esitetään kahdella tavulla eli 16 bitillä (bitti tarkoittaa virtapulssia tai sen puuttumista). Z80 tai 6502 prosessoria käyttävässä mikrotietokoneessa suurin mahdollinen muistimäärä on 64K (ROM ja RAM yhteenlaskettuna). Tämä johtuu siitä, että suurin luku, joka voidaan esittää 16 binääriluvulla on 65535, joten se on myös korkein mahdollinen osoite. Muistipaikkoja on siis 65536 ja ne on numeroitu 0–65535. Kuhunkin paikkaan mahtuu yksi tavu, 1024 tavua muodostaa yhden kilotavun (K) ja 65536 tavua on 64K (65536 ÷ 1024 = 64).



Syöttö/tulostus

Nämä muistipaikat ovat yhteydessä mikron otto/antoliitäntöihin.

Näyttömuisti

Tätä sanotaan joskus näyttötiedostoksi. Tässä muistin osassa säilytetään ruudulla näytettyä tietoa. Kaikki näyttömuistissa oleva tieto näkyy automaattisesti ruudulla. Yleensä mikroissa on ns. muistikartoitettu näyttö, jossa näyttöruudun kutakin kuvapistettä vastaa tietty paikka näyttömuistissa.

Käyttäjä-RAM

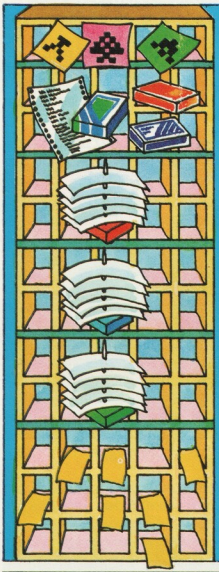
Tänne tallennetaan näppäilemäsi ohjelmat. Muuttujien ja taulukoiden datat tallennetaan käyttäjä-RAMin yläpäähän.

Varattu käyttöjärjestelmälle

Tietokone käyttää näitä RAM-paikkoja ohjelmia ajaessaan erilaisten tapahtumien seuraamiseen. Esim. kohdistimen paikka, nykyinen näytön väri, alas painettu näppäin ja nykyinen ohjelmarivin numero ovat tietoja, jotka tallennetaan tälle alueelle. Se on jaettu pienempiin osiin eri tehtävien hoitamista varten. Joissain mikroissa on tälle alueelle oma karttansa. Asiaa on selitetty seuraavalla sivulla.

Kurkistamme käyttöjärjestelmään

Tämä kuva antaa paremman käsityksen käyttöjärjestelmälle varatusta muistialueesta. Mikrosi käsikirjassa voi olla kartta tästäkin alueesta tai luettelo osoitteista ja niiden tehtävistä. Joissain mikroissa (esim. Sinclair/Timex) käyttöjärjestelmän käyttämät paikat eivät ole yhtenä ryhmänä vaan ne on ripoteltu eri puolille muistia.



Itsetehdyt graafiset merkit

Jos teet itse omia grafiikkamerkkejä, ne tallennetaan tänne.

Puskurit

Nämä ovat väliaikaisia varastoja, joihin pannaan näppäimistöstä tulevaa tai kirjoittimeen tai kasetille lähetettävää dataa.

Konepino

Tätä sanotaan myös prosessoripinoksi. Keskusyksikkö tallentaa näihin paikkoihin osoitteita silloin, kun se käsittelee konekielistä ohjelmaa.

BASIC-pino

Tätä sanotaan myös GOSUB-pinoksi. Tänne tallennetaan BASIC-käskyssä GOSUB käytetyt rivinumerot.

Laskuripino

Tämä on keskusyksikön väliaikainen varasto laskuissa käytetyille luvuille.

Järjestelmämuuttujat

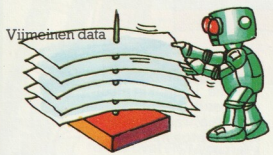
Nämä ovat muistipaikkoja, joihin keskusyksikkö tallentaa tietoa siitä, mitä mikron sisällä parhaillaan tapahtuu. Täällä on esim. erilliset paikat kohdistimen sijaintia, alas painettua näppäintä ja muuttujien tallennusosoitetta varten.

Muistisivut

Tietokoneen on helpompi liikkua muistissa, jos se jaetaan sivuihin. Mikrotietokoneessa yksi sivu on 256 paikkaa ja 4 sivua muodostavat yhden kilotavun ($4 \times 256 = 1024$).

Paikkoja 0–255 sanotaan joskus nollasivuksi. Tietokoneen eri muistialueet alkavat usein uudelta sivulta. Esim. edellisen sivun muistikartalla käyttäjä-RAM alkaa sivulta 45 jos ensimmäinen sivu lasketaan nollasivuksi.

Lisätietoa pinoista



Tietokone tallentaa väliaikaiset datat pinoihin tietyllä tavalla. Viimeksi tallennettu data on aina ensimmäinen, joka noudetaan käsittelyyn. Englanniksi tätä sanotaan LIFO-tallennukseksi (Last In, First Out): viimeisenä sisään, ensimmäisenä ulos.

Heksaluvut

Konekielisessä ohjelmassa numerot ja osoitteet kirjoitetaan aina heksadesimaaleina. Alla on selvitetty, miten desimaaliluvut muutetaan heksaluviksi ja päinvastoin.

Desimaali	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Heksa	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Tässä taulukossa on heksamerkit (0–9 ja A–F) ja niiden desimaaliarvot. Yli 15(F) suuruiset luvut esitetään kahdella (tai useammalla merkillä aivan kuten desimaalijärjestelmässä 9:n ylittävät luvut. Kunkin merkin arvo riippuu sen paikasta luvussa.

Desimaali			
1000	100	10	1
1	**2**	**2**	**6**

Heksa		
256	16	1
4	**C**	**A**

Desimaalijärjestelmässä luvun oikeanpuoleisin numero kertoo ykkösten määrän, toinen numero kymmenien määrän, kolmas satojen (10^2) määrän jne.

Heksaluvussa oikeanpuoleisin merkki kertoo ykkösten määrän, mutta seuraava merkki kertoo, montako 16:tta on ja kolmas montako 256:tta (16^2)luvussa on.

Heksan muutto desimaaliluvuksi

Heksaluku, esim. 4CA, muutetaan desimaaliluvuksi etsimällä kunkin merkin desimaalivastine. Sitten kukin vastine kerrotaan sen paikan arvolla ja vastaukset lasketaan yhteen.

256	16	1	
4	**C**	**A**	
4	12	10	Desimaaliarvo
×256	×16	×1	
1024 +	192 +	10 = 1226	

&4CA on desimaalilukuna 1226

Muuta &A7 desimaaliluvuksi ja 513 heksaksi? (Vastaus sivulla 124.)

Desimaaliluvusta heksaksi

Desimaaliluku, esim. 1226, muutetaan heksaluvuksi jakamalla se ensin 256:lla. Näin saadaan selville, montako 256:tta siinä on. Jäännös jaetaan 16:lla, jotta saadaan 16:t. Seuraava jäännös ilmoittaa ykkösten määrän. Lopuksi kukin vastaus muutetaan heksamerkiksi.*

1226 ÷ 256 = 4 ..... heksassa 4 on 4
jäännös 202

202 ÷ 16 = 12 ..... heksassa 12 on C
jäännös 10 ..... heksassa 10 on A

1226 on heksassa 4CA

Heksaosoitteiden muuttaminen

Heksaosoitteessa (esim. 5C64) kaksi ensimmäistä merkkiä kertoo, millä sivulla paikka on ja kaksi seuraavaa kertoo osoitteen sijainnin sivulla.

Heksasta desimaaliksi

Osoite &5C64

Sivunumero = &5C = 92 desimaalina
Paikka sivulla = &64 = 100 desimaalina

92 × 256 = 23552 + 100
= 23652

Heksaosoite 5C64 on desimaalina 23652.

Desimaalista heksaksi

Desimaaliosoite 23652

23652 ÷ 256 = 92 (muistin sivunumero)
jäännös 100 (paikka sivulla)

92 ÷ 16 = 5 jäännös 12 = &5C
100 ÷ 16 = 6 jäännös 4 = &64

Desimaaliosoite 23652 on heksana 5C64.

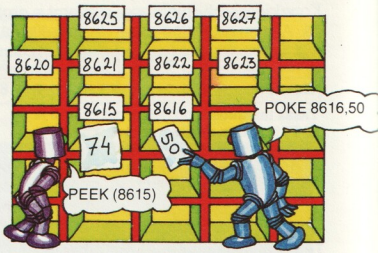
Heksaosoite muutetaan desimaaliksi siten, että ensin kukin merkkipari muutetaan desimaalinumeroksi. Sitten sivunumero kerrotaan 256:lla (sivulla on 256 paikkaa) ja lisätään sivupaikan numero.

Desimaaliosoite muutetaan heksaksi jakamalla 256:lla, jotta saadaan muistisivun numero. Jäännös ilmoittaa paikan sivulla. Sitten numerot muutetaan heksamerkeiksi siten kuin yllä on esitetty.

*Sivulla 121 kerrotaan, miten muunnos lasketaan koneella.

PEEK ja POKE

BASICissa on kaksi sanaa, PEEK ja POKE*, joilla voi katsella tietokoneen muistipaikkoihin tallennettuja tavuja ja muuttaa niitä. PEEK tarkoittaa "kurkistaa" ja POKE "pistää jonnekin". Sanoja käytetään muistipaikan heksaosoitteen kanssa (joissain mikroissa desimaaliosoitteen kanssa). Muista, että antaessasi tietokoneelle heksalukuja sinun pitää ensin kirjoittaa &, # tai S. Tarkista oikea merkki mikrosi käsikirjasta. Jotkut mikrot hyväksyvät vain desimaalilukuja.



PEEK-sanalla voit kurkistaa mihin tahansa tietokoneen muistipaikkaan, mutta uusia tavuja voit panna POKE-sanalla vain RAM-paikkoihin, koska ROMin tavuja ei voi muuttaa.

PEEK-sanan käyttö

```
PRINT PEEK(12345)
46
PRINT PEEK(720)
240
PRINT PEEK(8643)
0
LET A=PEEK(1024)
PRINT A
176
```

```
10 FOR J=700 TO 725
20 PRINT PEEK (J);",";
30 NEXT J
```

```
RUN
36,27,234,56,21,0,0,
0,0,45,32,67,121,45,
47,89,63,21,0,87,241,
202,225,63,87,16,
```

Nämä ovat konekielen tavujen desimaaliset vastineet.

Saat tietokoneen kurkistamaan haluttuun muistipaikkaan sanalla PEEK (tai mikrosi käyttämällä vastaavalla sanalla) sekä muistipaikan osoitteella. Tuloksen saat näyttöruudulle joko käskyllä PRINT PEEK tai tallentamalla tuloksen muuttujaan käskyllä LET ja printtaamalla muuttujan, kuten ylhäällä vasemmalla on tehty.

Yritä kirjoittaa lyhyt ohjelma käyttäen FOR/NEXT-silmukkaa (ks. keskimmäistä ruutua), jolla printtaat eri paikoissa olevia tavuja. Katso mikrosi muistikarttaa ja kokeile muistin eri osiin viittaavia osoitteita.

POKE-sanan käyttö

```
POKE 16763,60
PRINT PEEK(16763)
60
```

Tämä käskee mikroa panemaan 60 paikkaan 16763.

Tuloksen näet kun kirjoitat PRINT PEEK.

```
10 FOR A=16763
TO 16768
20 INPUT N
30 POKE A,N
40 NEXT A
```

Tämä panee luvun N paikkaan A.

Yllä olevasta kuvasta näet, miten POKE-sanaa käytetään. Voit pistää sillä tavuja mihin tahansa RAM-muistin kohtaan, mutta jos pistät uusia arvoja käyttöjärjestelmälle varatulle alueelle, saattaa mikron toiminta keskeytyä. Palauta se toimintakuntoon katkaisemalla virta hetkeksi. Yritä kirjoittaa yllä olevan kaltainen lyhyt ohjelma, jolla panet useita lukuja käyttäjä-RAMin eri paikkoihin.

Muistipaikkoihin pistämäsi lukujen tulee olla välillä 0–255. Jälkimmäinen on korkein luku, joka voidaan esittää kahdeksalla binääriluvulla (eli yhdellä konekielen tavulla).

*Joissain mikroissa käytetään eri käskyjä, esim. BBC:ssä käytetään ? Tarkista mikron käsikirjasta.

Mitä luvut tarkoittavat

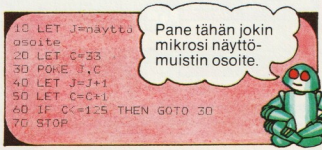
Kun käsket mikron printata näyttöruudulle jonkin muistipaikan sisällön, tulos on aina desimaaliluku väliltä 0–255. Tämä johtuu siitä, että kukin muistipaikka voi säilyttää yhden tavun ja korkein luku, joka voidaan esittää kahdeksalla binäärituvulla (eli yhdellä tavulla) on 255. On vain 256 (0–255) erilaista konekielen tavua ja yksi tavu voi merkitä tietokoneelle useita eri asioita.

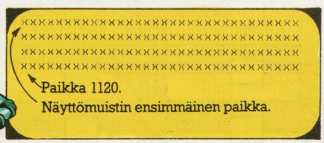
Esim. binääriluku 00110000 (desimaalina 48) voisi olla käskykannan yhden käskyn koodi, näppäimistön kirjaimen koodi tai osa toisen muistipaikan osoitteen koodia (jokainen osoite koostuu kahdesta tavusta).

Katso mikrosi käsikirjasta sen käyttöjärjestelmän osoite ja kokeile sitten tätä ohjelmaa. Ruudulla näkyvät luvut ovat erään käyttöjärjestelmän ohjelman konekielisten tavujen desimaalivastineita.

Etsi seuraavaksi mikrosi näyttömuisti ja yritä panna POKElla lukuja näyttömuistin paikkoihin. Sinun ei tarvitse käyttää käskyä PRINT PEEK koska näyttömuistiin tallennetut tavut tulevat automaattisesti ruudulle. Tällä kertaa mikro tulkitsee luvun merkkikoodiksi.*

Yleensä mikrot käyttävät ASCII-koodia selvittäessään, mitkä luvut vastaavat mitäkin merkkiä. Sen sijaan esim. ZX81 käyttää muita lukuja. VIC 20:ssa on omat lukunsa, ns. näyttökoodit, näytössä esitettäville merkeille. Mikrosi käsikirjasta pitäisi löytyä sen käyttämien merkkikoodien luettelo.





Yritä printata mikrosi merkkikanta yllä olevan kaltaisella lyhyellä ohjelmalla. Ohjelma käyttää ASCII-koodeja alkaen 33:sta, joka on ! :n koodi ja päättyen koodiin 90. Muut luvut välillä 0–255 edustavat erikoisnäppäimiä, esim. SPACE (välilyönti), ja DELETE (poista), käänteisnäyttöä tai vilkkunäyttöä ja grafiikkamerkkejä.

Yleensä mikrolla voi printata merkin tiettyyn paikkaan ruudulla selvittämällä paikan osoitteen. Esim. jos näyttömuisti alkaa paikasta 1024 ja mikron ruudun leveys on 32 merkkiä, neljännen rivin ensimmäisen paikan osoite on 1024 + (32×3) eli 1120. (Osoite 1024 lasketaan nollaksi).

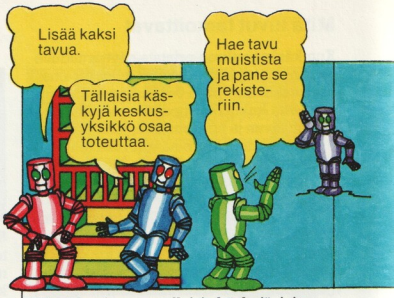
*Spectrumissa (Timex 2000) kunkin näyttöpaikan tiedot tallennetaan useisiin eri muistipaikkoihin. Merkkejä ei voi printata panemalla POKElla koodeja näyttömuistiin.

Keskusyksikön sisukset

Tietokone tekee työnsä noutamalla muistista käskytavuja ja dataa ja toteuttamalla käskyt keskusyksikössä.

Keskusyksikön sisällä on kolme pääaluetta: rekisterit, joissa dataluvut säilytetään käsittelyn ajan; aritmeettis-looginen yksikkö, jossa tavuja voidaan laskea yhteen, vähentää tai verrata; ja ohjausyksikkö, joka ohjaa näitä toimintoja.

Rekisterin sijoitus poikkeaa Z80 ja 6502 lastuissa toisistaan, kuten alla olevista kuvista näkyy.



Näissä kuvissa on sellaisia käskyjä, joita keskusyksikkö osaa toteuttaa. Käskyt ovat hyvin yksinkertaisia. Keskusyksikkö osaa hakea tavuja muistista ja panna ne rekistereihin, siirtää tavuja rekisteristä toiseen, käsitellä ne aritmeettis-loogisessa yksikössä ja tallentaa tulokset muistiin. Helpoim-

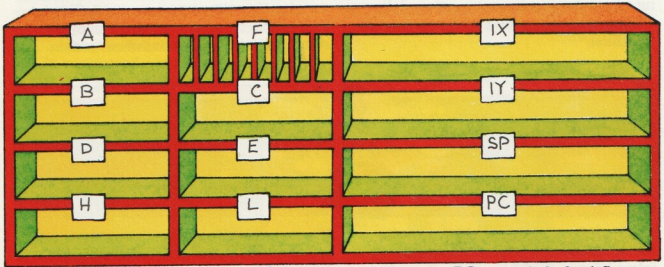
Z80:n rekisteri

Pääero Z80 ja 6502 lastujen välillä on siinä, että Z80:ssa on enemmän rekistereitä. Tämä tarkoittaa, että tavut voidaan väliaikaisesti tallentaa keskusyksikköön. 6502:ssa ne on sen sijaan lähetettävä takaisin muistiin.

A tarkoittaa akkua. Se on keskusyksikön tärkein rekisteri ja tallentaa tavut, jotka ovat menossa tai tulossa aritmeettis-loogisesta yksiköstä.

F on lippurekisteri. Siihen mahtuu kahdeksan bittiä mutta niistä käytetään vain kuutta. Kukin bitti toimii merkkinä. Esim. muistilippu on 1 kun vastaus on suurempi kuin 255 eikä sovi yhteen tavuun. Miinuslippu näyttää, onko luku positiivinen vai negatiivinen.

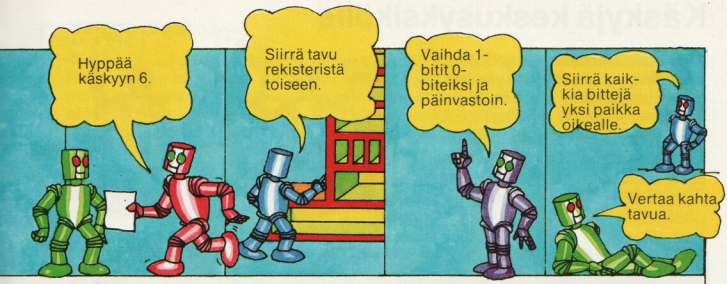
IX ja **IY** ovat indeksirekistereitä. Kumpaankin mahtuu 16 bittiä. Niitä käytetään tietyissä käskyissä tavun muistiosoitteen selvittämiseen.



B, C, D, E, H ja **L** ovat yleisrekistereitä, joihin voi tallentaa tavuja siirrettäessä niitä muistiin tai muistista. Kuhunkin mahtuu vain yksi tavu, mutta ne voidaan yhdistää pareittain, esim. BC, DE tai HL, jolloin niissä on tilaa kahdelle tavulle.

SP on pino-osoitin. Se on 16-bittinen rekisteri ja tallentaa konepinon viimeisen datan osoitteen. Konepino on paikka, jonne keskusyksikkö tallentaa väliaikaisia datoja.

PC on osoitelaskuri. Se on 16-bittinen rekisteri ja siellä säilytetään seuraavan muistista haettavan tavun osoite. Osoitelaskurin luku kasvaa yhdellä aina, kun yksi tavu on haettu.



matkin tehtävät – esim. kahden luvun yhteenlasku tai tuloksen näyttö ruudulla – vaativat yli sata lyhyttä työvaihetta, joita keskusyksikkö pystyy suorittamaan yli puoli miljoonaa sekunnissa.

Jokaisessa toimituksessa ohjausyksikkö noutaa käskytavun ROMista ja RAMista, lataa datatavun rekistereihin ja suorittaa käskyssä ilmoitetun toimituksen. Konekielellä voit määrätä, mitä keskusyksikkö tekee rekistereissä oleville tavuille, mutta aritmeettis-looginen ja ohjausyksikkö toimivat automaattisesti, joten et voi puuttua niiden toimintaan.

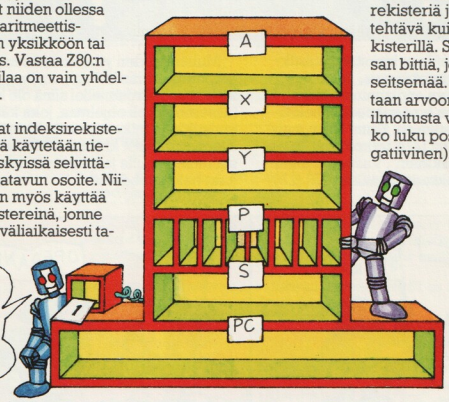
6502:n rekisterit

6502:n päärekisterit ovat samat kuin Z80:n, mutta osalla niistä on eri nimet.

A on akku, jonne tallennetaan tavut niiden ollessa matkalla aritmeettis-loogiseen yksikköön tai sieltä pois. Vastaa Z80:n akkua. Tilaa on vain yhdelle tavulle.

X ja **Y** ovat indeksirekistereitä. Niitä käytetään tietyissä käskyissä selvittämään datatavun osoite. Niitä voidaan myös käyttää yleisrekistereinä, jonne pannaan väliaikaisesti tavuja.

P tarkoittaa prosessorin tilarekisteriä ja sillä on sama tehtävä kuin Z80:n lippurekisterillä. Siinä on kahdeksan bittiä, joista käytetään seitsemää. Kukin bitti asetetaan arvoon 1 tietyn ehdon ilmoitusta varten (esim. onko luku positiivinen vai negatiivinen).

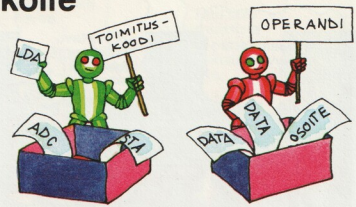


PC on osoitelaskuri ja toimii samalla tavalla kuin Z80:n PC-rekisteri.

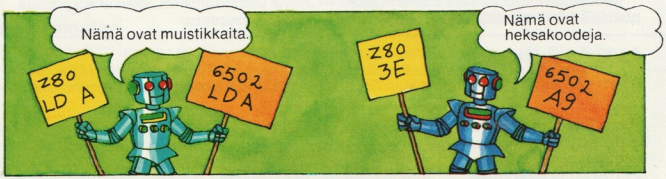
S on pino-osoitin. Se tallentaa pinon viimeisen datan osoitteen. Pino on RAM-alue, jonne keskusyksikkö tallentaa dataa. 6502:ssa pino-osoitin on 8-bittinen rekisteri. Osoitteiden tallentamista varten S-rekisteriin on kytketty yhdeksäs bitti, joka on pysyvästi 1. Se edustaa osoitteen sivunumeroa, joten 6502:ssa pino on aina muistin sivulla yksi. Pino-osoittimessa oleva luku ilmoittaa paikan sivulla.

Käskyjä keskusyksikölle

Konekielinen ohjelma muodostuu käskyistä, jotka kertovat keskusyksikölle täsmälleen, mitä sen tulee tehdä rekistereissä oleville tavuille. Voit käyttää vain sellaisia käskyjä, joita keskusyksikkö ymmärtää. Toisin sanoen tietokoneissa, joissa on Z80 ja Z80A mikroprosessori sinun pitää käyttää Z80:n käskykannan käskyjä ja tietokoneissa, joissa on prosessorina 6502, 6502A tai 6510, sinun pitää käyttää 6502:n käskyjä. Kirjan lopussa on luettelo Z80:n ja 6502:n käskyistä.

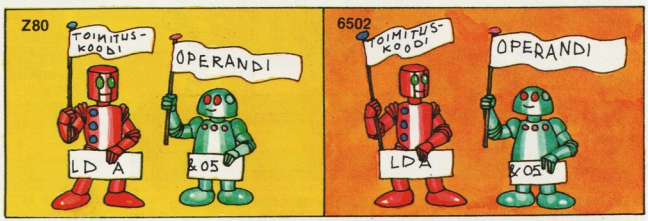


Yleensä konekielen käskyt ovat kaksiosaisia. Niissä on toimituskoodi (OPCODE) ja operandi. Toimituskoodi kertoo keskusyksikölle, mitä sen pitää tehdä ja operandi kertoo sille, mistä toimituksen kohteena olevat datat löytyvät. Toimituskoodit ovat käskykannan käskyjä.



Toimituskoodit voidaan kirjoittaa muistikkaina – ne ovat lyhyitä sanoja, jotka muistuttavat käskyn englanninkielistä nimeä – tai käskyjen binäärikoodien heksadesimaalisina vastineina. Esim. Z80:n muistikas LD A ja 6502:n LDA tarkoittavat "lataa (engl. load) tavu akkuun". Vastaavat toimituskoodit ovat heksadesimaaleina 3E (Z80) ja A9 (6502).

Muistikkaiden ymmärtäminen on paljon helpompaa kuin heksan, mutta niitä ei voi kirjoittaa mikroon, ellei siinä ole kääntäjää (assembler) eli ohjelmaa, joka kääntää muistikkaat mikron omalle kielelle.* Yleensä ihmiset kirjoittavat konekieltä ensin muistikkailla ja kääntävät sitten muistikkaat heksakoodiksi.



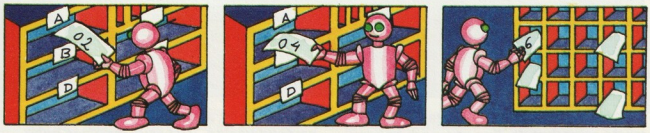
Tässä on kaksi konekielistä käskyä muistikkaina, toinen Z80:lle ja toinen 6502:lle. Kumpikin käskee mikron ladata heksaluvun 05 akkuun (& on heksalukujen merkki). Konekielessä luvut kirjoitetaan aina heksana. 6502:ssa luvun eteen tulee # merkiksi siitä, että on kyse datasta.

*Kääntäjistä on kerrottu sivulla 120.

Helppo ohjelma

Tässä on kaksi ohjelmaa, toinen Z80:lle ja toinen 6502:lle. Ne käskevät keskusyksikön laskea yhteen kaksi lukua. Kumpikin ohjelma on kirjoitettu muistikkailla. Tarkasti ottaen muistikkailla kirjoitettua ohjelmaa sanotaan symbolisella konekielellä kirjoitetuksi ohjelmaksi ja heksakoodilla kirjoitettua konekieliseksi ohjelmaksi. Seuraavalla sivulla kerrotaan, miten ohjelmat voi kääntää konekielelle ja sen jälkeen kerrotaan, miten käännetty ohjelma ladataan mikroon ja ajetaan sillä.

Z80:n ja 6502:n ohjelmissa on samat askelet, joskin varsinaiset käskyt eroavat toisistaan.* 6502:ssa data, jolla laskutoimitukset tehdään, pitää aina panna akkuun. Z80:ssa data pannaan akkuun, mutta isot luvut pannaan rekisteripariin HL.



Kaksi lukua lasketaan yhteen siten, että aluksi ensimmäinen luku ladataan akkuun. Sitten lisätään toinen luku akussa olevaan lukuun ja tallennetaan tulos muistiin. Alla on annettu näiden käskyjen muistikkaat.

Z80 muistikas	Merkitys
LD A, luku	Lataa A:han luku. A tarkoittaa akkua ja LD on lyhennys sanasta load (lataa).
ADD A, luku	Lisää A:han (akkuun) luku.
LD (osoite), A	Lataa tiettyyn osoitteeseen A:n (akun) sisältö. Osoitteet kirjoitetaan aina sulkuihin.

Z80:n toimituskoodit ja operandit erotetaan toisistaan pilkulla.

6502 muistikas	Merkitys
LDAluku	Lataa A:han luku. A on akku ja LD on lyhennys sanasta load (lataa).
	ADC on muistikas, joka tarkoittaa käskyä add with carry. Se käskee mikron lisätä luvun ja muistilipun akkuun ja asettaa lippurekisterin muistilipun jos tämä on tarpeen. Tästä on lisää tietoa sivulla 109.
STAosoite	Tallenna A (eli akun sisältö) tiettyyn osoitteeseen. ST on lyhennys sanasta store (tallenna) ja A tarkoittaa akkua.

Z80:n yhteenlaskuohjelma

```
LD A, &02
ADD A, &04
LD (&7F57), A
```

Data

Osoite

Tässä ohjelmassa on kolme toimituskoodia: LD A, ADD A, ja LD (osoite), A.

6502:n yhteenlaskuohjelma

```
LDA #&02
ADC #&04
STA &7F57
```

Data

Osoite

#-merkki ilmoittaa, että operandi on dataa.

Nyt voit lisätä datan ja osoitteet. Näissä esimerkeissä ohjelmat laskevat yhteen heksa 2 ja heksa 4 (eli desimaalilukuina 2 ja 4) ja tallentavat tuloksen muistipaikkaan heksa 7F57.

*Jos sinulla on Z80 prosessori, voit tästä alkaen hypätä 6502 ohjelmien yli. Jos mikrosi käyttää 6502:n käskyjä, hyppää Z80 ohjelmien yli.

Ohjelman kääntäminen heksakoodiksi

Muistikkaat voi kääntää heksakoodiksi vain yhdellä tavalla: etsimällä taulukosta kunkin muistikkaan vastineen. Tämän kirjan lopussa on muistikas- ja heksakoodi-taulukko. Ole huolellinen, kun käytät sitä, sillä kutakin käskyä varten on useita erilaisia heksakoodeja riippuen siitä, onko operandi dataa, osoite vai rekisterin nimi. Tässä on esimerkin vuoksi erilaisia versioita toimituskoodista, joka lataa akun ja sitä vastaavista heksakoodeista.

Z80		6502	
Muistikas	**Heksakoodi**	**Muistikas**	**Heksakoodi**
LD A,data	3E,data	LDA data	A9 data
LD A,(osoite)	3A, osoite	LDA osoite	AD osoite

Kun operandi on dataa, puhutaan välittömästä osoituksesta. Kun operandi on osoite, jonne data on tallennettu, puhutaan kiinteästä osoituksesta. Tämän kirjan lopussa olevassa muistikas- ja heksakoodiluettelossa on annettu kaikki tässä kirjassa käytetyt käskyt. Jos haluat kirjoittaa kehittyneempiä ohjelmia, tarvitset täydellisen Z80:n tai 6502:n koodiluettelon. Sivulla 120 on kirjaehdotuksia.

Z80 yhteenlaskuohjelma		6502 yhteenlaskuohjelma	
Muistikas	**Heksakoodi**	**Muistikas**	**Heksakoodi**
LD A,data	3E,data	LDA data	A9 data
ADD A,data	C6,data	ADC data	69 data
LD (osoite),A	32,osoite	STA osoite	8D osoite

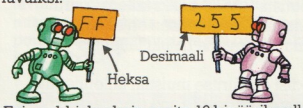
Tässä on Z80 ja 6502 yhteenlaskuohjelmien heksakoodit. Joskus muistikkaina ilmaistuna ilmaistuja käskyjä tuloskoodiksi.

Z80 yhteenlaskuohjelma ja data		6502 yhteenlaskuohjelma ja data	
Muistikas	**Heksakoodi**	**Muistikas**	**Heksakoodi**
LD A,&02	3E,02	LDA#&02	A9 02
ADD A,&04	C6,04	ADC#&04	69 04
LD (&757),A	32,577F	STA &7F57	8D 577F

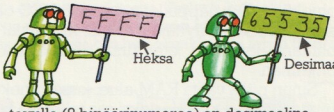
Nyt voit lisätä datan ja osoitteet. Tämä on aivan helppoa – osoitteiden lisäämistä lukuun ottamatta. Konekielessä sinun pitää muuttaa osoitteen kahden numeroparin järjestys päinvastaiseksi. Tästä kerrotaan lisää viereisellä sivulla.

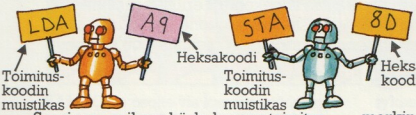
Lisätietoa heksakoodeista

Konekieliset ohjelmat kirjoitetaan heksakoodilla eikä desimaalilukuina siitä syystä, että tietokoneen käyttämät binääriluvut on helpompi kääntää heksakoodeiksi kuin desimaaliluvuiksi.

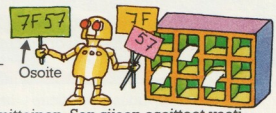


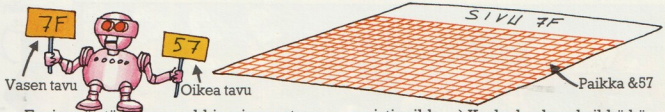
Esimerkki: korkein osoite 16 binääriluvulla on desimaalina 65535 ja heksana FFFF ja korkein luku, joka voidaan esittää yhdellä tavulla (8 binäärinumeroa) on desimaalina 255 ja heksana FF.*





Suurin osa mikron käskykannan toimituskoodeista on yhden tavun mittaisia, joten heksassa kukin toimituskoodi on kahden merkin mittainen. Sen sijaan osoitteet vaativat kaksi tavua tilaa eli kaksi heksamerkkiparia.





Ensimmäistä heksamerkkiparia sanotaan vasemmanpuoleiseksi tavuksi ja se ilmoittaa, millä muistin sivulla osoite sijaitsee (ks. s. 90). Toista merkkiparia sanotaan oikeanpuoleiseksi tavuksi ja se ilmoittaa muistipaikan sijainnin sivulla. (1 sivu = 256 muistipaikkaa.) Koska keskusyksikkö käsittelee osoitteita tietyllä tavalla, sille on aina ensin annettava oikea tavu (paikka sivulla) ja sitten vasen tavu (sivunumero).

Tutkimme konekielisiä ohjelmia

Lehtiin painetut konekieliset ohjelmat ovat erittäin sekavan näköisiä, jos et tunne niiden esitystapaa. Alla on kaksi esimerkkiä siitä, miten konekielisiä listauksia esitetään. (Kumpikaan ohjelma ei ole täydellinen eikä siten toimi tietokoneessa.)

Heksavedos

Kukin merkkipari on käsky, data tai osoitteen osa.

Heksaosoite

```
3A30 00 00 00 00 DB 01 CB 1F
3A38 CB 1F 30 07 3E 10 D3 00
3A40 CF 37 3F 21 2F 39 11 00
3A48 00 01 C0 03 C5 0E 27 71
3A50 1A FE 58 28 28 CD C5 3A
3A58 CD A0 3A D3 01 4E DD 79
3A60 FE 00 FA 88 3A F5 CD BD
3A68 08 FE 64 CA 06 3B F1 71
```

Tässä on ns. heksavedos. Kunkin rivin neljä ensimmäistä merkkiä ovat osoite ja loput merkkipareista ovat käskyjen, datan ja osoitteiden heksakoodeja. Kunkin rivin ensimmäinen koodi on tallennettu rivin alun osoitteeseen. Loput koodeista on tallennettu tämän osoitteen jälkeisiin paikkoihin.

Symbolisen konekielen listaus

Osoite — Heksakoodeja — Muistikkaita

```
0340  A2 00      LDX #&00
0342  BD 4E 03   LDA &034E,X
0345  9D C0 83   STA &83C0,X
0348  E8         INX
0349  E0 0B      CPX #&0B
034B  D0 F5      BNE &F5
034D  00         BRK
```

Tässä listauksessa on sekä heksakoodeja että muistikkaita. Kunkin rivin ensimmäinen luku on muistiosoite, jonne sen rivin ensimmäinen tavu on tallennettu. Seuraavassa sarakkeessa on ohjelman heksakoodit ja niiden jälkeen tulevat muistikkaat.

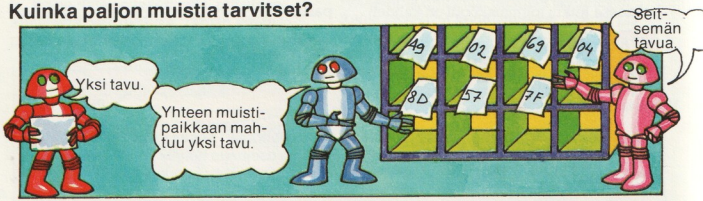
*Sivulla 108 on selitetty, miten binääriluvut muutetaan desimaaleiksi.

Tyhjän RAMin etsiminen

Joudut tekemään erinäisiä asioita, ennen kuin voit ladata ja ajaa sivun 98 yhteenlaskuohjelman. Ensin sinun pitää valita muistista alue, jonne tallennat ohjelman. Kun näppäilet mikroon BASIC-ohjelman, BASIC-tulkki tallentaa sen automaattisesti käyttäjä-RAMiin. Kun annat mikrolle konekielisen ohjelman, tulet ohittaneeksi tulkin, joten sinun pitää kertoa mikrolle, minne haluat tallentaa ohjelman.

Sinun pitää valita RAM-alue, jossa konekoodisi ei häiritse muistiin tallennettuja muita tietoja. Et esimerkiksi saa tallentaa konekoodia alueille, jotka on varattu käyttöjärjestelmälle, esim. järjestelmämuuttujille tai pinoille. Jos nimittäin teet näin, järjestelmä todennäköisesti romahtaa, koska olet tuhonnut tietokoneen toiminnalle elintärkeitä tietoja. Muista myös pitää konekoodisi erillään samaan aikaan tietokoneelle antamastasi BASIC-ohjelmasta. Jos tietokone romahtaa, ainoa keino sen elvyttämiseksi on katkaista virta hetkeksi, jolloin ohjelmasikin joutuu hukkaan.

Kuinka paljon muistia tarvitset?



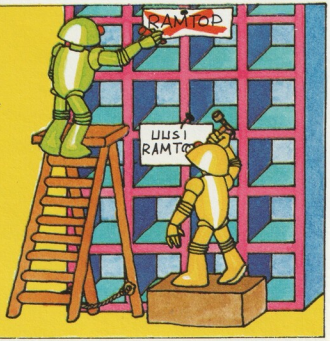
Konekielisen ohjelman pituus on helppo selvittää – pitää vain laskea, montako heksamerkkiparia siinä on (yksi pari tarvitsee yhden tavun verran muistia). Esimerkiksi yhteenlaskuohjelmassa on seitsemän tavua, 6502:ssa CLC-käsky mukaan lukien kahdeksan. Ks. s. 109 alinta kappaletta.

Yleensä konekieliset ohjelmat ovat lyhyitä. Jos sinulla on alussa sata tavua muistitilaa, se tulee todennäköisesti riittämään hyvin konekielisille ohjelmillesi.

Tyhjän RAMin etsiminen

Konekieliset ohjelmat tallennetaan tavallisesti käyttäjä-RAMin loppu- eli yläpäähän, jonne myös BASIC-ohjelmat tallennetaan. Sinun pitää kuitenkin varmistaa, ettei konekoodi sotkeudu mahdollisiin BASIC-ohjelmiin. Tämän vuoksi voit laskea käyttäjä-RAMin korkeinta osoitetta alemmaksi. Näin käyttäjä-RAMin yläpuolelle syntyy ei-kenenkään-maata, jota tietokone ei käytä, ennen kuin käsket sitä käyttämään sen ladatessasi konekielisen ohjelman.

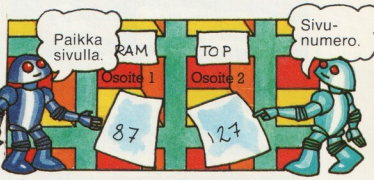
Käyttäjä-RAMin yläpäästä eli korkeimmasta muistiosoitteesta käytetään myös nimeä RAMTOP tai HIMEM. Viereisellä sivulla kerrotaan, miten RAMTOPin saa alemmas.

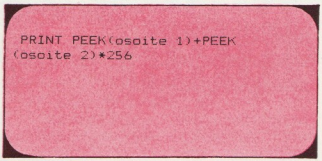


RAMTOP-osoitteen alentaminen

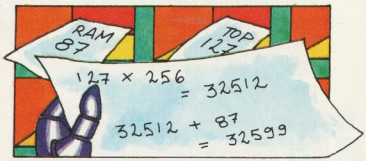
Tietokone tallentaa RAMTOPin osoitteen järjestelmämuuttujiin. Voit muuttaa RAMTOPia muuttamalla järjestelmämuuttujiin tallennettua osoitetta. Eri mikroilla tämä tehdään eri tavoin, mutta useimmat niistä noudattavat alla esitettyä periaatetta. Tarkista kuitenkin mikrosi käsikirjasta, miten RAMTOP-osoitetta muutetaan, sillä mikrosi käskyt voivat hyvin poiketa tässä esitetyistä ja sitä paitsi mikrossasi voi jopa olla helpompikin tapa tehdä tilaa konekoodille.*

RAMTOPin osoite vie kaksi peräkkäistä paikkaa järjestelmämuuttujissa. Toinen on sivunumeroa ja toinen sivupaikkaa varten. Etsi näiden järjestelmämuuttujien osoitteet mikrosi käsikirjasta (katso esim. kohdista RAMTOP tai HIMEM). Tietokone tallentaa osoitteen kaksi tavua käänteisessä järjestyksessä – ensin sivupaikan ja sitten sivunumeron – joten järjestelmämuuttujissa ensimmäisessä paikassa on paikkanumero ja toisessa sivu.

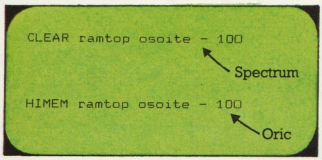




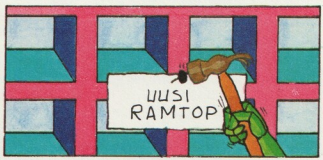
Käskyllä PRINT PEEK (tai mikrosi vastaavalla käskyllä) voit kurkistaa järjestelmämuuttujiin ja printata RAMTOPIN osoitteen. Lisää järjestelmämuuttujiesi osoitteet.



Tämä käsky muuttaa RAMTOP-osoitteen kaksi tavua automaattisesti desimaaliosoitteeksi kertomalla sivunumeron 256:lla ja lisäämällä siihen sivupaikan.



Yleensä mikroilla on kullakin oma käskynsä, jolla RAMTOPin osoitetta muutetaan. Esim. Spectrumissa käsky on CLEAR ja Oricissa se on HIMEM. Näiden käskyjen perään tulee käyttäjä-RAMin korkein osoite vähennettynä sillä muistitavujen määrällä, jonka haluat varata konekoodille (ylh. vas.). Tarkista mikrosi käyttämä käsky sen käsikirjasta.

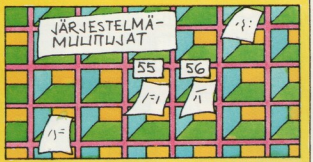


Nämä käskyt alentavat käyttäjä-RAMin korkeinta osoitetta 100 paikalla eli varaavat konekoodille 99 tavua alkaen RAMTOPia seuraavasta osoitteesta. Muuta lukua 100 jos haluat varata enemmän tilaa.

*Seuraavalla sivulla on ohjeet siitä, miten RAMTOP-osoitetta alennetaan VIC 20:ssa ja minne konekoodi tallennetaan ZX81:ssä.

VIC 20

VIC 20:ssa ei ole omaa käskyä, jolla voisi muuttaa järjestelmämuuttujiin tallennetun osoitteen. Tässä on käskyt, joilla alennetaan käyttäjä-RAMin korkeinta osoitetta VIC 20:lla.



Osoite on järjestelmämuuttujissa 55 ja 56. Muista, että sivunumero on jälkimmäisessä paikassa.

```
POKE 56,PEEK(56)-1
```

Käyttäjä-RAMin korkeinta osoitetta alennetaan 256:lla eli yhdellä sivulla käyttämällä yllä annettua suoraa käskyä. Se panee mikron kurkistamaan (PEEK) paikkaan 56 (jossa on sivunumero). Se vähentää siellä olevasta arvosta 1 ja pistää (POKE) uuden arvon takaisin paikkaan 56. Toisin sanoen se vähentää osoitteen sivunumero-osaa 1:llä. Käyttäjä-RAMin uuden korkeimman osoitteen saat näkyviin tällä käskyllä.

```
PRINT PEEK(55)+PEEK(56)*256
```

ZX81

ZX81:ssä paras paikka konekielisten ohjelmien tallentamiseen on käyttäjä-RAMin alussa. Tallennus tapahtuu näin: Kirjoita sivun 104 heksalataajaohjelman ensimmäiseksi riviksi REM-lause, jossa on yhtä paljon numeroita kuin konekielisessä ohjelmassasi on tavuja.

```
5 REM 1234567
```

Seitsemän tavua

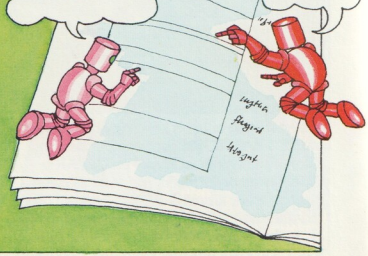
Jokainen REM-lauseen numero vie yhden muistipaikan. Nyt voit pistää POKElla konekieliset tavusi REM-lauseen numeroiden varaamiin paikkoihin.

Tätä varten sinun pitää tietää osoite, jonne ensimmäinen numero on tallennettu. Käyttäjä-RAM alkaa osoitteesta 16509 ja tietokone tarvitsee kaksi tavua REM-rivinumeroa varten, yhden REMia varten, yhden NEWLINEa varten ja yhden rivin pituutta varten, joten ensimmäinen numero on paikassa 16514.

Minne muualle konekoodia voi tallentaa?

Muistissa on eräitä muitakin paikkoja, jonne voit tallentaa konekoodia, mikäli et muuten käytä näitä paikkoja. Esim. jos et aio käyttää kasettinauhuria ohjelmassasi, voit panna sen kasettipuskuriin, tai jos et itse tee grafiikkamerkkejä, voit tallentaa ohjelman niille varatulle alueelle. Katso mikrosi käsikirjasta näiden RAM-alueiden osoitteet.

Käsikirjasta voi löytyä enemmänkin ehdotuksia siitä, minne päin tietokoneen muistia konekoodia voi tallentaa. Myös lehdistä ja kirjoista löytyy neuvoja.

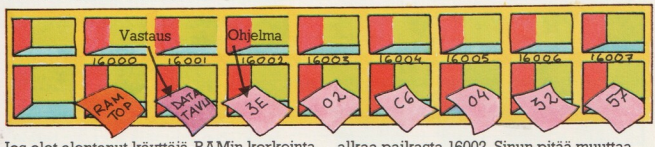


Ohjelman lataus ja ajo

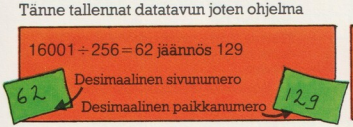
Seuraavilla sivuilla kerrotaan, miten sivun 98 ohjelma ladataan ja ajetaan. Konekielinen ohjelma annetaan mikrolle siten, että jokainen tavu pistetään POKElla sille muistialueelle, jonka olet varannut konekoodille (eli RAMTOPin yläpuolelle). Useimmissa mikroissa voi POKElla pistää muistiin vain desimaalilukuja, joten tarvitset tähän tehtävään lyhyen BASIC-ohjelman nimeltä heksalataaja. Heksalataaja muuntaa jokaisen konekoodin tavun desimaaliluvuksi ja pistää sen POKElla muistiin. Seuraavalla sivulla on heksalataajaohjelma. Ensin sinun on kuitenkin muutettava yhteenlaskuohjelman vastauksen osoite tietokoneellesi sopivaksi osoitteeksi. Sinun pitää myös lisätä ohjelmaan vielä yksi käsky (ks. alla).

Vastauksen osoitteen valinta

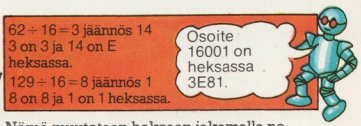
Konekielisen ohjelman tuottamaa dataa, esimerkiksi yhteenlaskuohjelman tulosta, sanotaan datatavuiksi. Datatavut pitää ehdottomasti tallentaa sellaiseen paikkaan, missä ne eivät sekoitu varsinaiseen ohjelmaan. Paras paikka niille on heti konekoodille varaamasi alueen alussa ennen ohjelmaa.



Jos olet alentanut käyttäjä-RAMin korkeinta osoitetta esim. paikkaan 16000, on konekoodin alueen ensimmäinen osoite 16001. Tänne tallennat datatavun joten ohjelma alkaa paikasta 16002. Sinun pitää muuttaa datatavun osoite heksadesimaaliksi, jotta voit lisätä sen ohjelmaan.

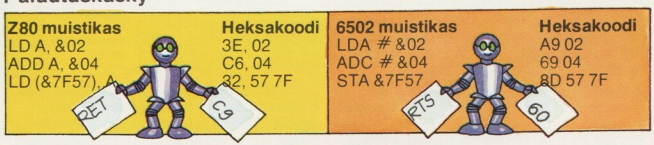


Osoite muutetaan heksaan jakamalla se 256:lla. Vastaus on desimaalinen sivunumero ja jäännös on paikka sivulla (ks. s. 91).



Nämä muutetaan heksaan jakamalla ne 16:lla ja muuntamalla vastaukset ja jäännökset heksamerkeiksi siten kuin yllä on tehty.

Palautuskäsky



Jokaisen konekielisen ohjelman lopussa pitää aina olla palautuskäsky RET (Z80) tai RTS (6502). Tämä saa mikron lopettamaan konekielisen ohjelman ajon ja palaamaan kohtaan, jossa se haarautui. Ilman tätä käskyä tietokone yrittäisi toteuttaa käskyn jokaisella muistissa olevalla tavulla, joten järjestelmä romahtaisi ennen pitkää.*

*Sivulla 115 kerrotaan lisää palautuskäskystä.

Heksalataajaohjelma

Tässä on ohjelma, jolla konekoodia ladataan tietokoneen muistiin. Lataajaa käytetään siten, että panet konekielisen ohjelmasi heksakoodit riville 160 ja niiden perään sanan LOPPU merkiksi siitä, että data loppuu siihen. Rivillä 40 tietokone lukee heksamerkkiparin, muuntaa sen desimaaliluvuksi riveillä 70–110 ja pistää (POKE) tämän luvun muistiin rivillä 130. Alla on ohjeet siitä, miten tämä ohjelma muutetaan Spectrumille. ZX81:n ja Atarin käyttäjille on ohjeita sivulla 137.

```
 10 PRINT "OSOITE JOHON KONEKOODI
 TALLENNETAAN?"
 20 INPUT A
 30 LET C=0
 40 READ H$
 50 IF H$="LOPPU" THEN GOTO 180
 60 IF LEN(H$)<>2 THEN GOTO 170
 70 LET X=(ASC(H$)-48)*16
 80 IF ASC(H$)>57 THEN LET X=
 (ASC(H$)-55)*16
 90 LET Y=ASC(RIGHT$(H$,1))
100 IF Y<58 THEN LET X=X+Y-48
110 IF Y>57 THEN LET X=X+Y-55
120 IF X<0 OR X>255 THEN GOTO 170
130 POKE A+C,X
140 LET C=C+1
150 GOTO 40
155 REM LISAA HEKSAKOODIT
160 DATA EF,F6,E2,A9,LOPPU
170 PRINT "VIRHEELLISTA DATAA"
180 STOP
```

A on ensimmäisen muistipaikan osoite sillä alueella, jonne haluat tallentaa ohjelmasi.

C on laskuri.

Panee rivin 160 ensimmäisen heksamerkkiparin H$:iin.

Katsoo, onko H$ sana LOPPU eli datan loppumerkki.

Varmistaa, että H$:ssa on kaksi merkkiä. Jos näin ei ole, menee riville 170.

Muuttaa ensimmäisen heksamerkin desimaaliluvuksi ja tallentaa sen X:ään.

Muuttaa toisen heksamerkin desimaaliluvuksi Y ja lisää sen X:ään.

Etsii virheellistä dataa varmistamalla, että X:ssä oleva desimaaliluku on väliltä 0–255.

Ensimmäisellä kerralla C = 0 joten pistää (POKE) X:n muistipaikkaan A.

Lisää C:hen 1 eli pistää seuraavan heksan desimaaliarvon paikkaan A + 1.

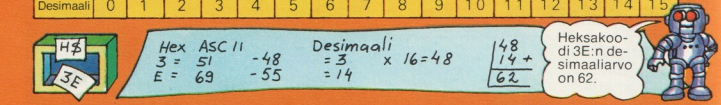
Palaa lukemaan seuraavan heksakoodin.

Pane heksakoodisi tänne ja lisää loppuun sana LOPPU.

Printtaa tämän viestin jos löytää viallista dataa rivillä 60 tai 120 ja pysähtyy sen jälkeen.

Miten lataaja toimii?

Heksa	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCII	48	49	50	51	52	53	54	55	56	57	65	66	67	68	69	70
							miinus 48							miinus 55		
Desimaali	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



Rivillä 70 tietokone muuttaa H$:n ensimmäisen merkin vastaavaksi ASCII-koodiksi käyttäen BASIC-sanaa ASC. Sitten se muuttaa ASCII-koodin desimaaliarvoksi vähentämällä siitä 48 tai (yli 57 menevistä koodeista) 55, kuten taulukosta näkyy. Sitten se kertoo 16:lla, koska ensimmäinen heksamerkki kertoo, montako 16:n joukkoa siinä on. Lopuksi tietokone panee vastauksen X:ään.

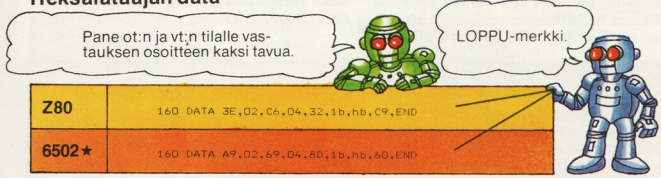
Rivillä 90 se käyttää samaa menetelmää oikeanpuoleisen merkin muuttamiseksi ASCII-koodiin ja tallentaa sen Y:hyn. Riveillä 100 ja 110 se muuttaa Y:n desimaaliluvuksi vähentämällä 48 tai 55 kuten edellä ja lisää tuloksen X:ään. (Tällä kertaa se ei kerro 16:lla, koska kyse on merkistä, joka ilmoittaa heksaluvun 1:t.) X:ssä oleva arvo on heksamerkkiparin desimaalinen vastine.

*Muuta Spectrumissa ASC-käsky muotoon CODE, pane kukin heksakoodipari lainausmerkkeihin ja muuta rivi 90 muotoon LET Y=CODE (H$(2 TO 1))

Lataajan käyttö

Voit nyt käyttää heksalataajaa ja kokeilla konekielistä yhteenlaskuohjelmaa. Tämä ei tosin ole mikään erityisen kiintoisa ohjelma, mutta yksinkertaisuudessaan se näyttää sinulle, miten konekoodi toimii. Näppäile heksalataaja mikroosi. Pane rivin 160 datan tilalle yhteenlaskuohjelman heksakoodit siten kuin alla on esitetty.

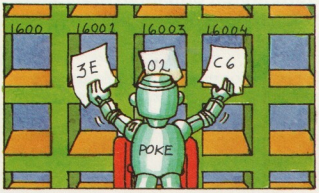
Heksalataajan data



Tässä on yhteenlaskuohjelman heksakoodit. Sinun pitää panna kirjainten ot (oikea tavu) ja vt (vasen tavu) tilalle ne kaksi tavua, jotka ilmaisevat osoitteen, jonne vastaus tietokoneessasi tallennetaan. Muista panna tavut päinvastaiseen järjestykseen, siis ensin oikea tavu (paikka sivulla) ja sitten vasen tavu (sivunumero).

Heksalataajan ajo

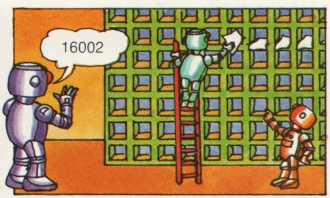
Kirjoita seuraavaksi RUN ajaaksesi heksalataajaohjelman. Kun ohjelma kysyy osoitetta, näppäile ensimmäinen sitä paikkaa



seuraava paikka, jonne vastauksen tallennat. Kirjoita tämä osoite desimaalilukuna, koska sitä käytetään POKE-käskyn kanssa.

Konekielisen ohjelman ajo

Eri tietokoneissa käytetään eri käskyä konekielisen ohjelman ajon aloittamiseksi. Joissain käytetään sanaa CALL, joissain PRINT USR tai SYS plus ohjelman ensimmäisen tavun muistipaikan desimaaliosoite.



Tarkista oikea käsky mikrosi käsikirjasta. Kun mikro saa oikean käskyn, se menee osoitteeseen ja alkaa suorittaa konekielisiä käskyjä.

* 18 on CLC-käsky. Ks. sivun 109 alinta kappaletta.

Tulosten näkeminen

```
PRINT PEEK (16001)
```

```
PRINT PEEK (16001)
 6
```

Tietokone toteuttaa konekieliset käskyt ja tallentaa vastauksen antamaasi paikkaan. Vastauksen näet, kun kirjoitat PRINT PEEK sekä vastauksen osoitteen. Tulos on vastaus desimaalilukuna.

Ohjelmaehdotuksia

Nyt osaat tarpeeksi konekieltä pystyäksesi kirjoittamaan yksinkertaisia ohjelmia. Sivun alaosassa on muistisi tueksi luettelo kaikesta siitä, mitä sinun pitää tehdä, kun kirjoitat konekielisiä ohjelmia. (Vastaukset löytyvät sivulta 124.)

1. Yritä kirjoittaa ohjelma, joka laskee yhteen 25 ja 73 (desimaalilukuja) ja tallentaa vastauksen muistiin.

2. Osaatko kirjoittaa ohjelman, joka laskee yhteen 64 ja 12 ja 14 (desimaalilukuja) ja tallentaa vastauksen muistiin?

Yhteenlaskuohjelma laskee vain lukuja, joiden summa on alle 255. Sivulla 108 kerrotaan, miten lasketaan yhteen suurempia lukuja.

Konekielen ohjelman muistilista

1. Kirjoita ohjelmasi symbolisella konekielellä ja muuta kaikki datat heksadesimaaleiksi.

2. Etsi kunkin muistikkaan heksakoodi taulukosta (kirjan lopussa on muistikas- ja heksakoodiluettelo).

Muista lisätä heksalataajaan sana LOPPU heksakoodien perään.

3. Lisää palautuskäsky ohjelman loppuun. (Ks. s. 103.)

4. Laske tavujen määrä ja varaa niille RAMista tilaa. (Ks. s. 100-102.)

Kirjoita ylös datatavujen osoitteet ja osoite, jonne tallennat ohjelman.

5. Selvitä, mitä muistipaikkoja tarvitset datatavuille ja muuta osoitteet heksadesimaaleiksi. (Ks. s. 103.)

6. Lisää osoitteet ohjelmaan. Muista panna nämä kaksi tavua päinvastaiseen järjestykseen. (Ks. s. 98-99.)

Ennen kuin ajat heksalataajan, tarkista datarivin heksakoodit erittäin huolellisesti.

7. Näppäile heksalataaja tietokoneeseen (voit tallentaa ohjelman kasetille) ja lisää riville 160 heksakoodit sekä sana LOPPU. (Ks. s. 104.)

Jos ohjelman ajo ei onnistu, tarkista, että olet käyttänyt oikeita heksakoodeja.

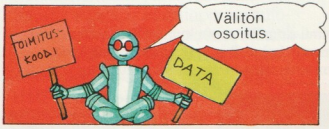
8. Aja heksalataaja ja näppäile ensimmäinen sen alueen desimaaliosoite, jonne haluat tallentaa konekoodin. (Ks. s. 105.)

9. Aja konekielinen ohjelma käyttäen mikrosi vaatimaa käskyä ja konekoodin ensimmäisen tallennuspaikan (desimaali)osoitetta. (Ks. s. 105.)

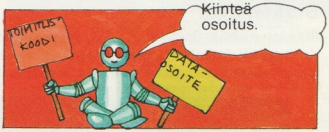
Jos muutat heksalataajan datoja, sinun pitää ajaa ohjelma uudestaan, jotta uudet tavut menevät muistiin.

Lisäämme tavuja muistista

Edellisessä ohjelmassa data oli mukana itse ohjelmassa. Tätä sanotaan välittömäksi osoitukseksi. Joskus haluat ehkä panna tietokoneen käsittelemään dataa, joka on tallennettu sen muistiin. Tässä tapauksessa käskyn operandiosa on osoite, joka kertoo, mistä data löytyy. Tätä sanotaan kiinteäksi (tai suoraksi tai laajennetuksi) osoitukseksi.



Tässä on esitetty vain kaksi niistä monista tavoista, joilla tietokoneelle voi kertoa, mistä käsiteltävä data löytyy. Erilaisia tapoja sanotaan osoitustavoiksi. Jokaisella käskyllä on erilainen heksakoodi riippuen siitä, mitä osoitustapaa käytetään.



Ohjelma, joka laskee yhteen muistissa olevia lukuja

Tässä on ohjelma, joka laskee yhteen kaksi muistista otettua lukua. Vertaa tämän (kiinteää osoitustapaa käyttävän) ohjelman käskyjen heksakoodeja edellisen (välitöntä osoitustapaa käyttäneen) yhteenlaskuohjelman heksakoodeihin.

Z80 ohjelma		
Muistikas	**Heksakoodi**	**Merkitys**
LD A,(osoite 1)	3A,osoite 1	Pane osoitteen 1 luku akkuun.
LD B,A	47	Pane akussa oleva luku rekisteriin B.
LD A,(osoite 2)	3A,osoite 2	Pane osoitteen 2 luku akkuun.
ADD A,B	80	Lisää rekisterin B luku akkuun.
LD (osoite 3),A	32, osoite 3	Tallenna akun sisältö osoitteeseen 3.
RET	C9	Palautus.

Jos haluat laskea yhteen kaksi muistissa olevaa lukua, ne pitää ensin ladata rekistereihin. Voit käyttää akkua (A) ja rekisteriä B. Et voi ladata rekisteriin B suoraan muistista, joten sinun on ensin pantava ensimmäinen luku akkuun ja siirrettävä se sieltä B:hen.

6502 ohjelma		
Muistikas	**Heksakoodi**	**Merkitys**
LDA osoite 1	AD osoite 1	Pane osoitteen 1 luku akkuun.
ADC osoite 2	6D osoite 2	Lisää osoitteen 2 luku akkuun.
STA osoite 3	8D osoite 3	Tallenna akun sisältö osoitteeseen 3.
RTS	60	Palautus.

Ohjelman ajo

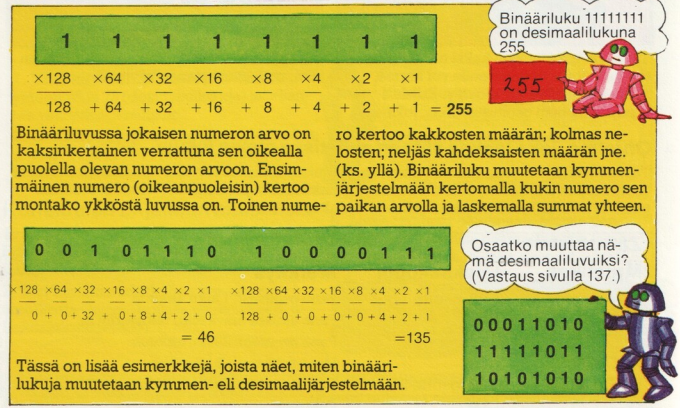
Kun ajat ohjelman, noudata viereisen sivun muistilistan askeleita. Ensin sinun on kuitenkin pistettävä POKElla muistiin yhteenlaskettavat luvut. Valitse muistipaikat konekoodille varaamasi alueen alusta, jotta datatavut pysyvät erillään käskyistä. Muuta sitten osoitteet heksaan ja lisää ne ohjelmaan. Tarvitset vastausta varten vielä kolmannen osoitteen. Tuloksen näet näppäilemällä PRINT PEEK(osoite 3).

Suurten lukujen käsittely

Edellisten sivujen ohjelmat pystyvät käsittelemään vain lukuja, joiden summa on korkeintaan 255. Tämä on korkein luku, jonka voi esittää kahdeksalla bitillä yhdessä rekisterissä tai muistipaikassa. Jotta voit käsitellä isompia lukuja, sinun pitää saada hieman enemmän tietoa binäärijärjestelmästä ja muistilipun käytöstä. Seuraavalla aukeamalla on konekielinen ohjelma, joka laskee yhteen suuria lukuja.

Binääriluvut

Binääri- eli kaksikantainen lukujärjestelmä toimii samalla tavoin kuin heksadesimaali- ja desimaalijärjestelmät, paitsi että numeroita on vain kaksi, 0 ja 1. 1:tä suuremmat luvut koostuvat useista numeroista. Kunkin numeron arvo riippuu sen paikasta luvussa.

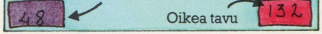


Binääriluvussa jokaisen numeron arvo on kaksinkertainen verrattuna sen oikealla puolella olevan numeron arvoon. Ensimmäinen numero (oikeanpuoleisin) kertoo montako ykköstä luvussa on. Toinen numero kertoo kakkosten määrän; kolmas nelosten; neljäs kahdeksaisten määrän jne. (ks. yllä). Binääriluku muutetaan kymmenjärjestelmään kertomalla kukin numero sen paikan arvolla ja laskemalla summat yhteen.

Tässä on lisää esimerkkejä, joista näet, miten binäärilukuja muutetaan kymmen- eli desimaalijärjestelmään.

Miten mikrolle annetaan isoja lukuja

Tietokoneen sisällä yli 255 menevät luvut tallennetaan kahteen tavuun, vasempaan ja oikeaan tavuun, aivan kuten osoitteetkin. Vasen tavu kertoo, montako 256:tta luvussa on ja oikea tavu on jäännös. Tietokone käsittelee aina ensin oikean ja vasta sitten vasemman tavun. Sinun pitää myös tallentaa ne muistiin tässä järjestyksessä.



Kun annat mikrolle luvun, joka on suurempi kuin 255, sinun pitää selvittää jokaisen tavun arvo. Tätä varten luku pitää jakaa 256:lla. Vastaus on vasemman tavun desimaaliarvo. Jäännös on oikea tavu.

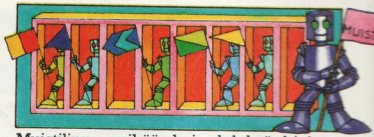
Jos haluat käyttää lukua datana konekielisessä ohjelmassa, sinun pitää muuttaa jokainen tavu heksaan. Jaa kaikki tavut 16:lla ja muuta sitten vastaukset jäännökset heksamerkeiksi sivun 91 ohjeiden mukaan.

307; 21214; 759; 1023

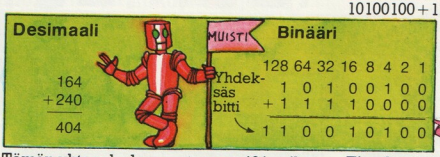
Mitkä ovat näiden lukujen desimaaliset vasemmat ja oikeat tavut? Ja mitkä ne ovat heksadesimaaleina? (Vastaukset sivulla 137.)

Muistilippu

Muistilippu on yksi bitti lippurekisterissä (jota sanotaan myös tilarekisteriksi) ja sitä käytetään ilmoittamaan, onko yhteenlaskun summa suurempi kuin 255, jolloin se ei mahdu yhteen tavuun (kahteen bittiin). Aina, kun näin käy, tietokone panee muistilippuun automaattisesti 1:n. Tätä sanotaan muistilipun asettamiseksi. Jos muistilipusta tehdään 0, puhutaan nollauksesta.



Muistilippu on ikään kuin yhdeksäs bitti, joka ilmoittaa, että luvun kahdeksannesta sarkkeesta on pantu muistiin binäärinumero 1. Katso esim. alla esitettyä summaa 164 + 240 (binäärilukuina 10100100 + 11110000).



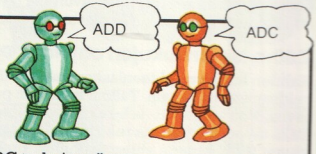
Tämän yhteenlaskun vastaus on 404, mikä vie binäärilukuna 9 bittiä tilaa. Yhdeksäs bitti kertoo, montako 256:tta luvussa on.

Binäärilukuja lasketaan yhteen siten, että aina, kun sarakkeen summa ylittää 1, pannaan 1 muistiin samalla tavalla kuin kymmenjärjestelmän lukuja yhteen lasketaan, jos sarakkeen summa ylittää 9.

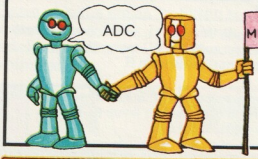
Tietokoneessa tätä edustaa muistilipun bitti.

Muistiin paneminen Z80:ssa

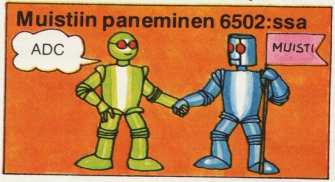
Z80:ssa on kaksi erilaista yhteenlaskukäskyä: ADD ja ADC. ADD käskee tietokoneen laskea yhteen kaksi lukua ja unohtamaan aikaisemmista laskuista muistiin pannun numeron. Jos laskussa pitää panna numero muistiin, tietokone asettaa muistilipun ja jos numeroa ei tarvitse panna muistiin, se panee muistilipun asentoon 0.



ADC tarkoittaa "summaa ja pane muistiin" ja käskee laskea yhteen kaksi lukua plus muistilipun asettaa tai nollata muistilipun vastauksesta riippuen. Jos lasket useita laskuja, käytät ensimmäisessä yhteenlaskussa ADD-käskyä, jotta mukaan ei tule aiemmasta toimituksesta jäänyttä muistinumeroa. Käytä seuraavissa laskuissa ADC-käskyä siltä varalta, että ensimmäisestä toimituksesta pantiin numero muistiin.



Muistiin paneminen 6502:ssa



6502:ssa on vain yksi yhteenlaskukäsky, ADC, joten se ottaa laskuihin aina mukaan muistilipun sisällön. Tästä syystä on ehdottoman tärkeää, että muistat nollata muistilipun käskyllä CLC ennen yhteenlaskuja. CLC (= heksa 18) pitää siis lisätä kaikkien edellä esitettyjen ohjelmien alkuun, jotta ne toimivat oikein.

Isoja lukuja käyttävät ohjelmat

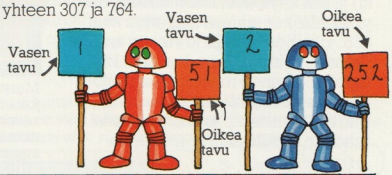
Ennen kuin voit ajaa tämän aukeaman ohjelmia, sinun pitää selvittää jokaisen laskuihin tulevan luvun vasen ja oikea tavu ja pistää ne POKElla muistiin. Oletetaan vaikka, että haluat laskea yhteen 307 ja 764.

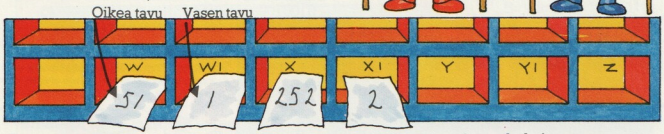
Ensimmäinen luku: **307**

307 ÷ 256 = 1 jäännös 51

Toinen luku: **764**

764 ÷ 256 = 2 jäännös 252





Seuraavaksi sinun pitää pistää nämä tavut POKElla muistipaikkoihin konekielelle varaamasi alueen alkuun. Jokaisen luvun oikeanpuoleisen tavun pitää olla ensimmäisessä paikassa ja vasemman tavun sitä seuraavassa paikassa. Yllä olevassa kuvassa ensimmäisen luvun kaksi tavua on tallennettu paikkoihin W ja W1 ja toisen luvun tavut paikkoihin X ja X1. Tarvitset kolme paikkaa (Y, Y1 ja Z) vastausta varten (yhden oikealle ja yhden vasemmalle tavulle ja yhden mahdolliselle muistiin pantavalle numerolle).

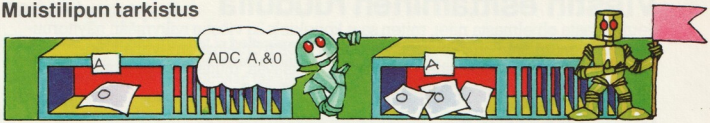
Z80:n isojen lukujen ohjelma

Kahden lukumme yhteenlaskeminen Z80:lla on aivan helppoa, koska voit käyttää rekistereitä pareittain. Tällöin kuhunkin pariin mahtuu yhden luvun tarvitsemat kaksi tavua. Voit käyttää H- ja L-rekistereitä yhtenä parina ja B- ja C-rekistereitä toisena parina. Tällä tavoin käytettyinä rekistereitä sanotaan HL- ja BC-rekistereiksi. Jos et käytä akkua, käytät HL-rekisteriä yhteenlaskuihin. Tässä ovat ohjelman muistikkaat ja heksakoodit. Ohjelmaa tutkiessasi sinulle voi olla apua sivun yläosan kuvasta.

Muistikas	Heksakoodi	Merkitys
LD HL,(osoite W)	2A,osoite W	Panee osoitteen W tavun (1. luvun oikean tavun) rekisteriin L ja osoitteen W1 tavun (1. luvun vasemman tavun) rekisteriin H.
LD BC,(osoite X)	ED4B,osoite X Tämä toimituskoodi on 2 tavun mittainen.	Panee osoitteen X tavun (2. luvun oikea tavu) rekisteriin C ja osoitteen X1 tavun (2. luvun vasen tavu) rekisteriin B.
ADD HL,BC	09	Summaa HL:n ja BC:n sisällöt ja jättää tuloksen HL:ään. Ei summaa muistilippua mutta asettaa sen jos on tarpeen.
LD (osoite Y),HL	22, osoite Y	Tallentaa vastauksen oikean tavun osoitteeseen Y ja vasemman tavun osoitteeseen Y1.
LD A,&0 ADC A,&0 LD (osoite Z),A	3E,0 CE,0 32 osoite Z	Viereisellä sivulla selitetään, miten mikro tarkistaa muistilipun.
RET	C9	Palautus.

Kun ajat ohjelman, sinun pitää lisätä W, X, Y ja Z:n heksaosoitteet. (Muista vaihtaa merkkiparien paikat.) Kun käytät rekistereitä pareittain, tarvitsee sinun ilmoittaa vain yksi osoite paria kohden. Mikro panee seuraavan osoitteen tavun automaattisesti parin toiseen rekisteriin.

Muistilipun tarkistus

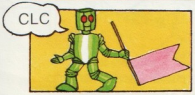


Z80-ohjelman rivit 5–7 tarkistavat muistilipun. Muistilipun sisältöä ei voi suoraan ladata rekisteriin tai muistiin. Ainoa keino tarkistaa, onko lippu asetettu, on tehdä ylimääräinen yhteenlasku. Pane akkuun 0 (5. rivi) ja lisää 0 käskyllä ADC. Jos edellinen laskutoimitus asetti muistilipun, akussa on nyt 1 (muistilipusta) ja tämä tallennetaan osoitteeseen Z (rivi 7).

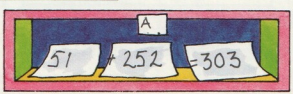
6502:n isojen lukujen ohjelma

Tässä on 255:ttä suurempien lukujen yhteenlaskuohjelma 6502:ta varten. Ennen kuin ajat ohjelman, sinun pitää selvittää yhteenlaskettavien lukujen vasemmat ja oikeat tavut ja pistää ne POKEilla muistiin siten, kuin viereisellä sivulla on neuvottu.

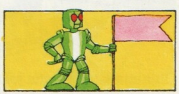
Muistikas	Heksakoodi
CLC	18
LDA osoite W	AD osoite W
ADC osoite X	6D osoite X
STA osoite Y	8D osoite Y
LDA osoite W1	AD osoite W1
ADC osoite X1	6D osoite X1
STA osoite Y1	8D osoite Y1
LDA#&0	A9 00
ADC#&0	69 00
STA osoite Z	8D osoite Z
RTS	60



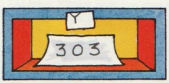
Ensin ohjelma nollaa muistilipun, jonka edellinen lasku on voinut asettaa.

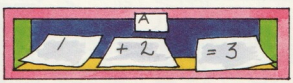


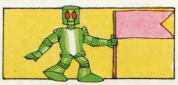
Sitten se panee 1. luvun oikean tavun akkuun ja laskee ADC-käskyllä 2. luvun oikean tavun (rivit 2 ja 3).



Jos tulos on suurempi kuin 255 se asettaa muistilipun.





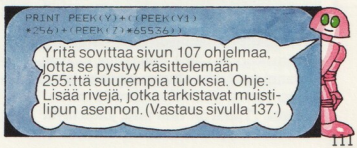


Se tallentaa tuloksen paikkaan Y (4. rivi). Sitten se summaa vasemmat tavut ja edellisen laskun mahdollisesti asettaman muistilipun. Se tallentaa tuloksen paikkaan Y1 (7. rivi).

Rivit 8–10 tarkistavat, onko muistilippu asetettu käyttäen sivun yläosassa selitettyä menetelmää.

Tuloksen printtaus

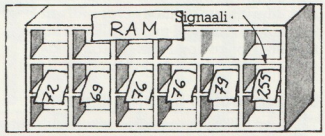
Tulos tallennetaan kolmena tavuna. Oikea tavu (paikka y) ilmoittaa ykkösten määrän. Vasen tavu (paikka Y1) ilmoittaa 256:ten määrän. Tällä kertaa muisti (paikka Z) ilmoittaa 65536:ten määrän. Tuloksen saat näkyviin, kun käytät oikealla annettua käskyä. (Pane Y, Y1 ja Z:n tilalle mikrosi käyttämät osoitteet.)



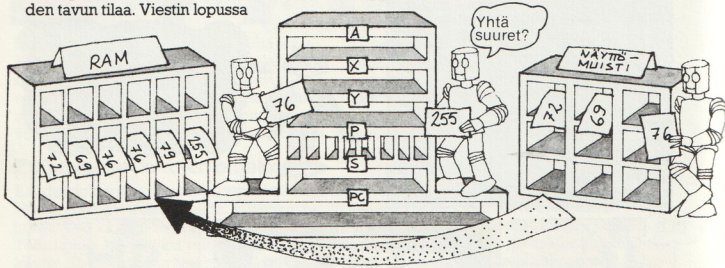
Viestin esittäminen ruudulla

Seuraavasta ohjelmasta näet, miten saat konekielellä viestin näkymään näyttöruudulla. Z80:n ohjelma on viereisellä sivulla ja 6502:n ohjelma on sivulla 114. Molemmat ohjelmat noudattavat samaa peruskaavaa, mutta sen toteutus eroaa näiden kahden prosessorin välillä hieman toisistaan.*

Miten ohjelma toimii

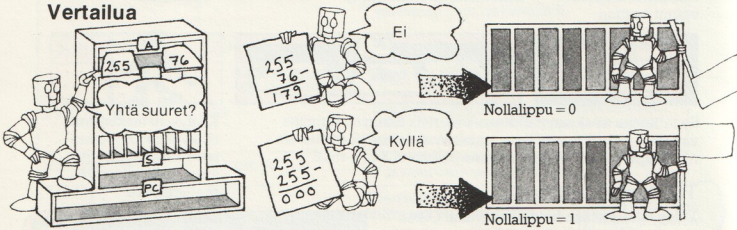


Ensin pistät POKElla viestisi jokaisen merkin merkkikoodin tyhjän RAM-alueesi alun muistipaikkoihin. Jokainen kirjain vie yhden tavun tilaa. Viestin lopussa pistät POKElla muistiin koodin 255 merkiksi siitä, että tietokone on tullut viestin loppuun.



Ohjelma lataa viestin jokaisen tavun akkuun ja vertaa sitä 255:teen. Jos viestin tavu ei ole sama kuin 255, ohjelma tallentaa sen näyttömuistiin ja se näkyy automaattisesti ruudulla. Sitten tietokone hyppää ohjelman alkuun ja etsii muistista viestin seuraavan tavun.

Vertailua



Käytä Z80:ssa toimituskoodia CP ja 6502:ssa toimituskoodia CMP, kun panet mikron vertaamaan tavua akussa olevaan tavuun. Mikro vertaa niitä vähentämällä toisen toisesta. (Tämä on vain testi, joten tavut eivät muutu.) Jos tulos on 0, tavut ovat yhtä suuret ja lippurekisterin nollalippu asetetaan 1:ksi. Jos luvut eivät ole yhtä suuret, nollalippu on 0. Voit seuraavaksi käskeä mikron mennä ohjelman johonkin toiseen osaan tai jatkaa seuraavasta käskystä sen mukaan, onko nollalippu 1 vai 0.

*Spectrumissa (Timex 2000) et saa asianmukaista viestiä ruudulle laitteen näyttömuistin rakenteesta johtuen.

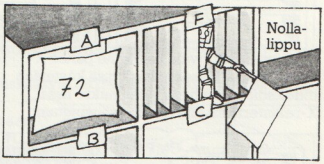
Z80:n viestiohjelma

Tässä ovat Z80:n muistikkaat ja heksakoodit. Ennen kuin ajat ohjelman, pistä POKElla viestisi tyhjään RAMiin. Lisää sitten osoitteet ohjelman riveille 1 ja 2. Ohjelman viimeinen käsky panee mikron hyppäämään takaisin kolmanteen käskyyn. Lisää ohjelman viimeiselle riville osoite, jonne kolmas käsky on mikrossasi tallennettu.

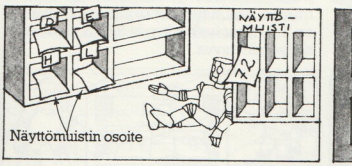
Muistikas	Heksakoodi
LD HL,näyttöosoite	21, näyttöosoite
LD DE,viestiosoite	11, viestiosoite
LD A, (DE)	1A
CP,&FF	FE, FF
RET Z	C8
LD (HL),A	77
INC, DE	13
INC, HL	23
JP,3. käskyn osoite	C3,3. käskyn osoite

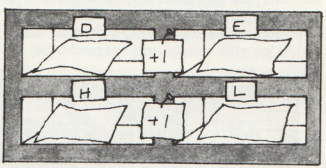
Tässä ohjelmassa rekisteripareja HL ja DE käytetään osoittimina, jotka viittaavat osoitteeseen, jonne tietokoneen pitäisi tallentaa tai josta sen pitäisi noutaa dataa. Tätä sanotaan epäsuoraksi osoitukseksi. 3. ja 6. rivin käskyt käyttävät epäsuoraa osoitusta.

Kahdella ensimmäisellä rivillä tietokone panee rekisteriparin HL näyttöosoitteen (osoitteen, jonne data tallennetaan) ja rekisteriparin DE viestiosoitteen (osoitteen, josta data haetaan).



LD A, (DE) käskee mikron lukea DE:ssä olevan osoitteen, hakea tavun tuosta osoitteesta ja panna sen akkuun. Tämä on epäsuoraa osoitusta. Sitten mikro vertaa (CP) akussa olevaa tavua &FF:ään (heksakoodin vastine 255:lle). RET Z käskee mikron palata BASICiin, jos nollalippu on 1 (eli jos tavu on 255). Jos nollalippu on 0, se jatkaa seuraavasta käskystä.





LD (HL),A käyttää myös epäsuoraa osoitusta. Se käskee mikron lukea HL:ssä olevan osoitteen ja tallentaa akun sisällön (viestitavun) tuohon osoitteeseen. INC on muistikas, joka tarkoittaa inkrementtiä eli "lisää yksi". 7. ja 8. rivillä tietokone lisää DE:ssä ja HL:ssä pidettyihin osoitteisiin 1. Kun se hyppää takaisin 3. rivin käskyyn, se hakee viestitavun seuraavasta muistipaikasta.

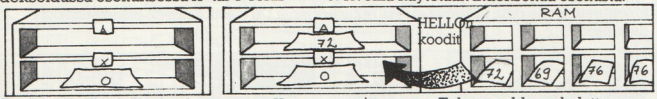
6502:n viestiohjelma

Tässä ovat 6502:n muistikkaat ja heksakoodit. Ennen kuin ajat ohjelman, pistä POKElla viestisi merkkikoodit tyhjään RAMiin sekä niiden perään 255, viestin loppumerkki. Pane seuraavaksi viestin ensimmäisen muistipaikan (heksa) osoite ohjelman toiselle riville. Pane viidennelle riville jokin tietokoneesi näyttömuistin osoite.

Sinun pitää myös lisätä 7. riville osoite, jonne tietokoneessasi on tallennettu ohjelman toinen käsky. Tämä saa tietokoneen toistamaan ohjelman.

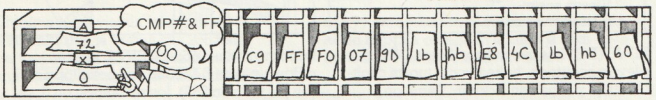
Muistikas	Heksakoodi
LDX #&00	A2 00
LDA viestiosoite, X	BD viestiosoite
CMP #&FF	C9 FF
BEQ käskyyn RTS	F0 07
STA näyttoosoite, X	9D näyttoosoite
INX	E8
JMP 2. käskyn osoite	4C 2. käskyn osoite
RTS	60

Tässä ohjelmassa käytetään toisenlaista osoitustapaa, ns. indeksoitua osoitusta. Indeksoidussa osoituksessa X- tai Y-rekisterin sisältö lisätään operandiin, jotta saadaan osoite, jonne data on tallennettu. 2. ja 5. riveillä käytetään indeksoitua osoitusta.

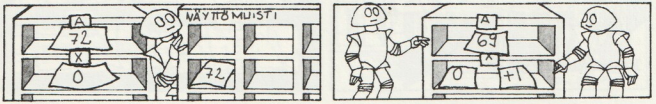


Ensimmäisellä rivillä tietokone panee X-rekisteriin 0. Toisessa käskyssä käytetään indeksoitua osoitusta, joten mikro lisää X-rekisterin sisällön käskyssä annettuun osoitteeseen. Tulos on akkuun ladattavan datan (viestitavun) osoite.

Seitsemän tavua



CMP 3. rivillä panee mikron vertaamaan akun tavua &FF:ään (heksan vastine 255:lle), joka on viestin loppumerkki. Jos ne ovat yhtä suuret, se asettaa nollalipun 1:ksi. Seuraavassa käskyssä BEQ tarkoittaa "haaraudu jos yhtä suuret", siis haaraudu, jos nollalippu on 1. Heksakoodeissa sen perässä on numero, joka kertoo, monenko paikan yli mikron pitää hypätä. Haluamme mikron haarautuvan RTS:ään jos viestitavu on 255. Haarauskäskyn ja RTS:n välillä on seitsemän tavua.



5. rivillä ohjelma käyttää indeksoitua osoitusta ja tallentaa akun tavun (viestitavun) käskyssä annettuun osoitteeseen lisättynä X:llä.

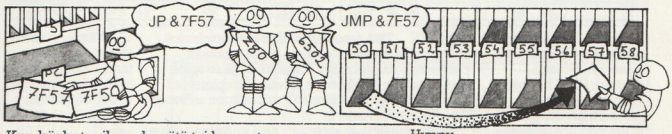
INX tarkoittaa "inkrementti X" ja lisää X-rekisteriin 1. Sitten mikro hyppää takaisin toiseen käskyyn. Tällä kertaa X on 1, joten mikro lataa seuraavan viestitavun akkuun ja tallentaa sen seuraavaan näyttöpaikkaan.

Hyppiminen ja haarautuminen

Haarautuminen on sitä, että tietokone pannaan menemään ohjelman jossain toisessa osassa olevaan käskyyn. Haarautumista on kolmenlaista: hyppyjä, aliohjelmia ja ehdollisia haarautumisia. Ehdollisessa haarautumisessa tietokone testaa jonkin ehdon ja sen jälkeen joko haarautuu tai jatkaa seuraavasta käskystä sen mukaan, oliko ehto tosi vai ei. Seuraavalla sivulla kerrotaan lisää ehdollisista haarautumisista. Hypyissä tietokone menee tiettyyn osoitteeseen.

Osoitelaskuri

Osoitelaskuri on 16-bittinen rekisteri, jossa on seuraavaksi toteutettavan käskyn osoite. Tietokone lukee osoitelaskurissa olevan luvun ja menee sen ilmoittamaan osoitteeseen hakemaan seuraavan käskyn. Sitten osoitelaskuriin lisätään 1, jolloin se osoittaa seuraavaan muistipaikkaan.

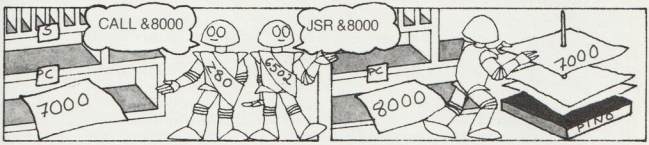


Hyppy

Kun käsket mikron hypätä tai haarautua tiettyyn osoitteeseen, osoite pannaan osoitelaskuriin ja tietokone toteuttaa käskyt vuoron perään tuosta osoitteesta alkaen. Yllä olevasta kuvasta näkyvät Z80:n ja 6502:n hyppyä tarkoittavat toimituskoodit.

Aliohjelmat

Z80:n käsky "CALL osoite" ja 6502:n "JSR osoite" (hyppää aliohjelmaan) käskevät mikron mennä aliohjelmaan. Toiminta on sama kuin BASICissa ja aliohjelman lopussa tarvitset palautuskäskyn (Z80:ssa RET ja 6502:ssa RTS).

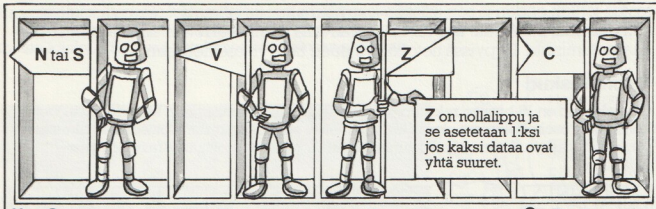


Kun käsket mikron mennä aliohjelmaan, aliohjelman osoite pannaan osoitelaskuriin. Osoitelaskurin sisältö (käskyn CALL tai JSR jälkeisen käskyn osoite) tallennetaan eli pinotaan pinoon. Pino on RAMin osa, joka on varattu tietokoneen käyttöön (ks. s. 90).

Kun tietokone tulee aliohjelman lopussa käskyyn RTS tai RET, se noutaa pinon ylimmän luvun ja panee sen osoitelaskuriin. Tämä osoite ilmoittaa käskyn, joka seuraa käskyä, josta haarauduttiin aliohjelmaan. Tietokone toimii juuri näin, kun käsket sen ajaa konekielisen ohjelman.

Ehdollinen haarautuminen

Ehdollisessa haarautumisessa tietokone testaa yhden lippurekisterin biteistä ja tuloksesta riippuen joko haarautuu tai jatkaa seuraavasta käskystä. Tässä ovat lippurekisterin bitit, joita ehdollisessa haarautumisessa voi testata.



N tai **S** Tämä on miinuslippu. 6502:ssa sitä sanotaan N:ksi ja Z80:ssa S:ksi. Se asetetaan 1:ksi jos laskun tulos on negatiivinen ja 0:ksi jos tulos on positiivinen.

V tai **P/V** Tätä sanotaan 6502:ssa ylivuotobitiksi. Z80:ssa sillä on kaksi tehtävää ja sitä sanotaan pariteetti/ylivuotobitiksi. Ylivuotobittinä se asetetaan 1:ksi kun laskun tulos kahden komplementti -esityksessä johtaa siihen että miinuslippu asetetaan (ks. viereistä sivua).

Pariteettibittinä se asetetaan 1:ksi jos tavussa on pariton määrä ykkösiä. Näin sitä käytetään tarkistusta varten.

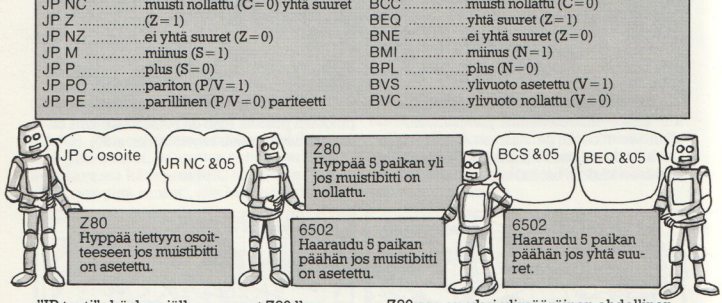
C Tämä on muistilippu. Se asetetaan 1:ksi kun yhteenlaskun vastaus ei mahdu yhteen tavuun.

Vertaa-käskyn lisäksi erinäiset käskyt asettavat tai nollaavat näitä lippuja automaattisesti. Esim. 6502:ssa käsky DEC (vähennys) vaikuttaa miinus- ja nollalippuihin.*

Ehdollisen haarautumisen toimituskoodit

Tässä on eri bittien testaukseen tarvittavat ehdollisen haarautumisen käskyt.

Z80	Hyppää jos...	6502	Haaraudu jos...
JP C	muisti asetettu (C=1)	BCS	muisti asetettu (C=1)
JP NC	muisti nollattu (C=0) yhtä suuret	BCC	muisti nollattu (C=0)
JP Z	(Z=1)	BEQ	yhtä suuret (Z=1)
JP NZ	ei yhtä suuret (Z=0)	BNE	ei yhtä suuret (Z=0)
JP M	miinus (S=1)	BMI	miinus (N=1)
JP P	plus (S=0)	BPL	plus (N=0)
JP PO	pariton (P/V=1)	BVS	ylivuoto asetettu (V=1)
JP PE	parillinen (P/V=0) pariteetti	BVC	ylivuoto nollattu (V=0)

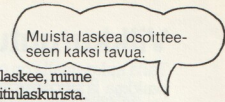


"JP testi" -käskyn jälkeen annat Z80:lle sen käskyn osoitteen, johon haluat sen hyppäävän. 6502:ssa annat mikrolle luvun, joka kertoo, monenko paikan yli sen on hypättävä eteen- tai taaksepäin käskyn löytääkseen. Tätä sanotaan suhteelliseksi osoitukseksi ja lukua sanotaan siirtymäksi.

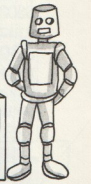
Z80:ssa on yksi ylimääräinen ehdollinen haarautumiskäsky, "JR testi", jota käytetään siirtymän eikä osoitteen kanssa. JR tarkoittaa "hyppää suhteessa". JR:llä voi testata vain nollalipun ja muistilipun.

*Mikrotietokoneesi käskykantaa tutkimalla saat selville, mitkä käskyt vaikuttavat mihinkin lippuihin.

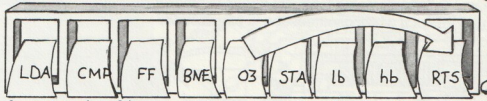
Siirtymän laskeminen



Kun annat mikrolle siirtymäluvun ehdollisessa haarautumisessa, mikro laskee, minne osoitteeseen sen pitää hypätä lisäämällä tai vähentämällä siirtymän osoitinlaskurista. Siirtymän saat selville, kun lasket, monenko tavun päässä haluamasi käsky on (laske mukaan varsinaisen käskyn tavu). Aloita laskeminen ehdollista haarautumista seuraavasta käskystä ja laske se 0:ksi (koska osoitelaskuri osoittaa jo tuohon käskyyn). Tässä on esimerkin vuoksi kaksi lyhyttä 6502:n ohjelmaa, jotta näet, miten siirtymä lasketaan. (Z80:ssa menetellään samalla tavalla.)



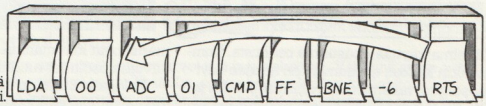
```
LDA osoite
CMP#&FF
BNE RTS:ään
STA osoite
RTS
```



Tietokone hyppää ylemmän esimerkin RTS-käskyyn, kun siirtymä on 3.

Alemmassa esimerkissä tietokone hyppää takaisin päin käskyyn ADC, jos siirtymä on –6.

```
LDA#&00
ADC#&01
CMP#&FF
BNE ADC:hen
RTS ← Laske tämä käsky 0:ksi.
```

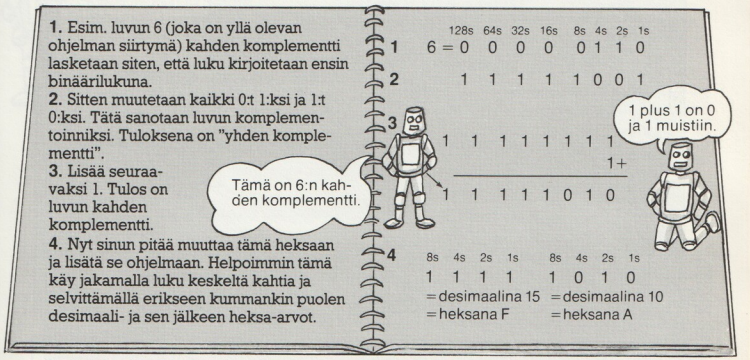


Hyppy eteen ja taakse

Eteen hypättäessä käännät vain siirtymän heksaluvuksi ja lisäät sen ohjelmaan. Takaisin hypättäessä siirtymä sen sijaan on negatiivinen luku, eikä 8-bitin binäärikoodilla voi ilmaista negatiivisia lukuja. Sen sijaan sinun pitää käyttää uudenlaista merkintätapaa nimeltä "kahden komplementti". Kahden komplementissa vasenta bittiä käytetään merkkibittinä. Jos tämä bitti on 1 luku on negatiivinen. Jos se on 0 luku on positiivinen.

Kahden komplementti

1. Esim. luvun 6 (joka on yllä olevan ohjelman siirtymä) kahden komplementti lasketaan siten, että luku kirjoitetaan ensin binäärilukuna.
2. Sitten muutetaan kaikki 0:t 1:ksi ja 1:t 0:ksi. Tätä sanotaan luvun komplementoinniksi. Tuloksena on "yhden komplementti".
3. Lisää seuraavaksi 1. Tulos on luvun kahden komplementti.
4. Nyt sinun pitää muuttaa tämä heksaan ja lisätä se ohjelmaan. Helpoimmin tämä käy jakamalla luku keskeltä kahtia ja selvittämällä erikseen kummankin puolen desimaali- ja sen jälkeen heksa-arvot.



6:n kahden komplementti on siis heksana FA. Taaksepäin tapahtuvaa hyppyä varten lisäät tämän luvun ohjelmaan. Korkein luku, jonka kahden komplementeilla voi esittää, on 128. Tämä on suurin taaksepäinsiirtymä, jota voit käyttää. Suurin eteenpäin siirtymä on 127, joka on korkein luku, jonka voi esittää, kun 8. binääriluku on asetettu 0:ksi positiivisen luvun merkiksi.

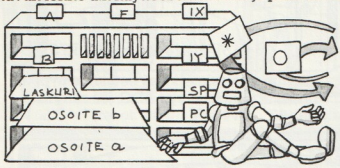
Osaatko laskea heksan vastineen 12, 18 ja 9:n kahden komplementeille? (Vastaus sivulla 137.)

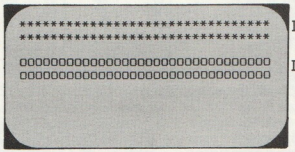
Vilkkuruutuohjelma

Tällä aukeamalla on ohjelma, joka vaihtaa ruudulla keskenään kahden näytön paikkaa ja saa siten aikaan vilkkuvan tehosteen. Ohjelmasta näet, miten yksinkertainen animaatio toimii. Alla on Z80:n ohjelma ja viereisellä sivulla on 6502:n ohjelma. Lopussa on ohjeita siitä, miten ohjelma ajetaan näillä mikroprosessoreilla.

Z80:n vilkkunäyttö

Hyvin pelkistetysti sanottuna tämä ohjelma vaihtaa näytössä näkyvien lohkojen paikkaa lataamalla kummastakin lohkosta yhden tavun rekistereihin ja tallentamalla lohko b:n tavun lohko a:n näyttöosoitteeseen ja päinvastoin.



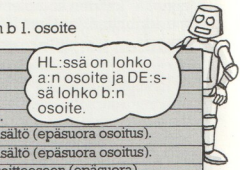


Ohjelma käyttää epäsuoraa osoitusta. Kummankin lohkon ensimmäisten tavujen näyttöosoitteet tallennetaan rekistereihin HL ja DE. Tietokone lukee näissä rekistereissä olevat osoitteet aina, kun se lataa tai tallentaa tavut. Kun tavut ovat vaihtaneet paikkaa, käsky INC (inkrementti) lisää HL:ään ja DE:hen 1. Kun ohjelma toistuu, nämä ovat kummankin lohkon seuraavien tavujen osoitteet ruudulla.

Rekisteri B:hen on tallennettu, montako tavua vaihtaa paikkaansa. Aina, kun ohjelma toistuu, B:stä vähennetään (DEC) 1, joten se toimii laskurina. Kun B = 0 kaikki tavut on vaihdettu.

Z80:n ohjelma

n = yhden lohkon tavujen määrä; a = lohkon a 1. osoite; b = lohkon b 1. osoite



Muistikas	Heksakoodi	Merkitys
LD B,n	06,n	Laskuri.
LD HL, (osoite a)	21, osoite a	Pane lohkon a osoite HL:ään.
LD DE, (osoite b)	11, osoite b	Pane lohkon b osoite DE:hen.
LD C, (HL)	4E	Lataa C:hen HL:n osoitteen sisältö (epäsuora osoitus).
LD A, (DE)	1A	Lataa A:han DE:n osoitteen sisältö (epäsuora osoitus).
LD (HL),A	77	Tallenna akun sisältö HL:n osoitteeseen (epäsuora).
LD A, C	79	Pane C (lohkon a 1. tavu) akkuun.
LD (DE),A	12	Tallenna akun sisältö DE:n osoitteeseen.
INC HL INC DE	23 13	Lisää yksi HL:ään ja DE:hen.
DEC B	05	Vähennä B:stä (laskurista) yksi.
LD A, &00	3E, 00	Pane akkuun 0.
CP B	B8	Vertaa B:tä akun sisältöön (0).
JR NZ 4. käskyyn	20, F3	Jos B ei ole 0 hyppää takaisin &F3 paikkaa lataamaan seuraavat tavut rekisteriin. F3 on heksan vastine 13:n kahden komplementille (ks. s. 117).
RET	C9	Palautus.

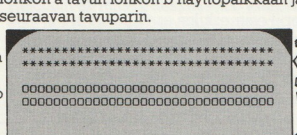
Datan ja osoitteiden lisääminen

n (yhden lohkon merkkien määrä) Laske n kertomalla rivin merkkimäärä yhden lohkon rivimäärällä. Muuta heksaan.

osoitteet a ja b Jos haluat vaihtaa keskenään ruudun 2 ylintä riviä ja seuraavat 2 riviä, pane osoite a mikrosi näyttömuistin 1. osoitteeksi. Osoite b on lohko a:n osoite plus vaihdettavien tavujen määrä. Muuta kumpikin osoite heksaan.

6502:n vilkkunäyttö

Tämä ohjelma vaihtaa lohkojen paikkaa tavu tavulta (eli merkki kerrallaan) alkaen kummankin lohkon viimeisestä tavusta. Ohjelma lataa nämä tavut rekistereihin, tallentaa lohkon a tavun lohkon b näyttöpaikkaan ja päinvastoin. Sitten ohjelma toistuu ja vaihtaa seuraavan tavuparin.



Ohjelma käyttää indeksoitua osoitusta kunkin tavun osoitteen etsimiseen. Yhden lohkon tavujen kokonaismäärä ladataan X-rekisteriin. Kun tavuja ladataan tai tallennetaan, X-rekisterin luku lisätään kunkin lohkon alkuosoitteeseen. Käsky DEX (vähennä X) panee mikron vähentämään X:stä 1, joten ohjelman toistuessa tietokone noutaa seuraavan tavun takaisin näyttöön.

6502:n vilkkunäyttöohjelma

Viereisen sivun alaosassa on kerrottu, miten arvot n, a ja b lasketaan. Vähennä sitten a:sta ja b:stä 1. Näin varmistat, että kun mikro lisää X:n, se saa tulokseksi kunkin lohkon viimeisen osoitteen eikä seuraavan rivin ensimmäistä osoitetta. (Varmista, että n, a ja b ovat heksadesimaaleja.)

Muistikas	Heksakoodi	Merkitys
LDX #n	A2 n	Lataa X:ään yhden lohkon tavujen määrä.
LDA osoite a, X	BD osoite a	Pane akkuun osoitepaikan a+X sisältö.
TAY	A8	Siirrä akun sisältö rekisteriin Y.
LDA osoite b, X	BD osoite b	Pane akkuun osoitepaikan b+X sisältö.
STA osoite a, X	9D osoite a	Tallenna akun sisältö osoitteeseen a+X.
TYA	98	Siirrä Y-rekisterin sisältö takaisin akkuun.
STA osoite b, X	9D osoite b	Tallenna akun sisältö osoitteeseen b+X.
DEX	CA	Vähennä X:stä 1. Nollalippu asetetaan 1:ksi kun X=0.
BNE käskyyn kaksi	D0 EF	Haaraudu takaisin &EF paikkaan jos X ei ole 0. EF on 17:n kahden komplementin heksakoodi (ks. s. 117).
RTS	60	Palautus.

Ohjelman lataaminen ja ajo Z80:lla tai 6502:lla

Paras tapa ajaa tämä ohjelma on käyttää sitä konekielisenä aliohjelmana heksalataajassa. Tee näin:

1. Näppäile mikroon heksalataaja ja pane tietokoneesi mikroprosessorin heksakoodit riville 160.

2. Rivillä 180 tarvitset 2 silmukkaa, joilla pistät POKElla näytön merkit näyttömuistiin. Tässä on esimerkkinä rivit, joilla saadaan kaksi riviä * -symboleita (koodi 42) ja niiden perään 2 riviä 0:a (koodi 48) tietokoneessa, jonka näytön leveys on 40 merkkiä.

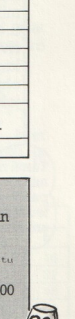
```
180 FOR J=0 TO 79
190 POKE 1. näyttöosoite + J,42
200 NEXT J
210 FOR J=80 TO 159
220 POKE 1. näyttöosoite + J,48
230 NEXT J
```

3. Lisää seuraavaksi nämä rivit ohjelman loppuun:

```
240 CALL osoite, jonne konekoodi on tallennettu
250 FOR K=1 TO 500
260 NEXT K
270 GOTO 240
```

Muuta viivesilmukan luku 500 mikrollesi sopivaksi.

4. Kirjoita nyt RUN ajaaksesi ohjelman. Heksalataaja pistää POKElla heksakoodit muistiin ja näyttökoodit näyttömuistiin. Rivi 240 lähettää sen paikkaan, jonne konekielinen ohjelma on tallennettu toteuttamaan ohjelman käskyt. Konekielinen ohjelma sinänsä vaihtaa näytön vain kertaalleen, joten rivi 270 kutsuu ohjelmaa uudestaan ja uudestaan, jotta saadaan aikaan välkkytehoste. Viivesilmukka on tarpeen, koska konekielinen ohjelma on erittäin nopea.



Tästä eteenpäin

Jos haluat saada lisää tietoa konekielestä, on paras keino ruveta kirjoittamaan omia lyhyitä ohjelmia ja kokeilla ja tutkia toisten kirjoittamia ohjelmia. Yksi hyvä tapa käyttää konekieltä on tehdä lyhyitä aliohjelmia, jotka suorittavat tietyn tehtävän BASIC-ohjelman sisällä. Konekieli sopii esimerkiksi erittäin hyvin tietojen lajitteluun tai ruudun täyttämiseen grafiikalla, koska se on nopeampi ja vie vähemmän muistia kuin BASIC. Tällaisia tehtäviä suorittavia aliohjelmia löydät lehdistä. Jos aliohjelmasi on kirjoitettu juuri omalle mikrollesi, voit ajaa ne sellaisenaan. Jos ne on kirjoitettu jollekin toiselle mikrolle, jossa on sama prosessori kuin omassasi, sinun pitää muuttaa ohjelman osoitteet sen alueen osoitteiksi, jonka olet tietokoneessasi varannut konekoodia varten.

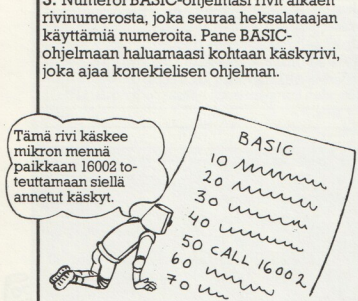
Konekieliset aliohjelmat

Noudata seuraavia ohjeita, kun käytät konekielistä aliohjelmaa BASIC-ohjelmassa.

1. Tee muistiin tilaa konekoodille alentamalla käyttäjä-RAMin korkeinta osoitetta (ks. s. 100–102).

2. Pane konekielisen aliohjelman koodit sivun 104 heksalataajaohjelman riville 160. (Tarkista, että konekielisen ohjelman lopussa on palautuskäsky.) Lisää rivejä datarivien POKElla pistämistä varten, jos tämä on tarpeen, näppäile heksalataaja mikroon ja aja se.

3. Numeroi BASIC-ohjelmasi rivit alkaen rivinumerosta, joka seuraa heksalataajan käyttämiä numeroita. Pane BASIC-ohjelmaan haluamaasi kohtaan käskyrivi, joka ajaa konekielisen ohjelman.



4. Näppäile BASIC-ohjelma mikroosi ja kirjoita RUN. Mikro toteuttaa BASIC-käskyt ja kun se tulee riville, joka käskee sen ajaa konekielisen ohjelman, se menee osoitteeseen, jonne konekoodi on tallennettu ja toteuttaa siellä annetut käskyt. Konekoodin lopussa oleva palautuskäsky lähettää mikron takaisin BASIC-ohjelman seuraavalle riville.

Kääntäjän käyttö

Kääntäjä (assembler eli ohjelma, jonka ansiosta voit kirjoittaa konekielisen ohjelman käyttäen muistikkaita) helpottaa konekielellä ohjelmointia huomattavasti. Useimpiin kotitietokoneisiin voi ostaa kasetilla olevan kääntäjän. Joissain mikroissa, esim. BBC:ssä, kääntäjä on valmiina.

Kääntäjän avulla voit kirjoittaa muistikkaiden viereen huomautuksia, jotka muistuttavat sinua kunkin rivin tehtävästä. Kääntäjä näyttää ruudulla ohjelman heksana ja muistikkaina sekä näyttää osoitteet, joihin käskyt on tallennettu sekä huomautukset.

Kääntäjä vaihtaa automaattisesti osoitteiden merkkiparien paikat ja laskee hyppyjen osoitteet tai siirtymät. Joissain kääntäjissä dataan voi viitata symbolisilla nimillä BASICin muuttujien tapaan. Hyvässä kääntäjässä on lisäksi tutkaaja (debugger), jolla voi etsiä virheet, ja editori, jonka avulla voit korjata ne.

Kirjaehdotuksia

Konekielestä on kirjoitettu paljon kirjoja eri mikroille. Kirjat ovat kuitenkin englanninkielisiä. Seuraavista kirjoista voi olla apua:

Programming the Z80 ja **Programming the 6502.** Molemmat on kirjoittanut Rodney Zaks ja julkaissut Sybex. Nämä ovat hyvin yksityiskohtaisia oppaita, joissa on täydelliset luettelot prosessorien käskyistä. Aloittelijalle ne eivät ole helppoa luettavaa, mutta ne ovat hyviä hakuteoksia.

VIC 20 Programmer's Reference Guide, julkaissut Commodore.

6502 Machine Code for Beginners, kirjoittanut A.P. Stephenson, julkaissut Newnes Microcomputer Books.

Desimaali/heksa muunnostaulukot

Tämä taulukko muuntaa heksaluvut 0–FF desimaaliluviksi ja päinvastoin.

Heksasta desimaaliksi

Heksaluvun muutat desimaaliksi siten, että etsit heksalukusi ensimmäisen merkin kohdalta vaakarivin ja toisen heksamerkin kohdalta pystysarakkeen risteyskohdan. Rivin ja sarakkeen risteyksessä näkyvä luku on heksalukusi desimaalivastine. Esim. heksaluku A1 on desimaalina 161.

Desimaalista heksaksi

Desimaaliluvun muutat heksaksi siten, että etsit taulukosta desimaaliluvun. Sitten etsit vaakarivin vasemmasta reunasta ensimmäisen heksamerkin ja sarakkeen yläreunasta toisen heksamerkin. Esim. 154 on 9A.

Ensimmäinen heksamerkki \ Toinen heksamerkki	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Osoitteiden muuntaminen

Taulukkoa käytät heksaosoitteiden muuntamiseen siten, että etsit osoitteen ensimmäisen merkkiparin desimaalivastineen. Tämä on sivunumero. Etsi sitten toisen merkkiparin desimaalivastine, jotta saat selville paikan sivulla. Kerro sivunumero 256:lla ja lisää siihen paikka sivulla.

Kahden komplementtien muunnostaulukko

Tästä taulukosta saat heksana kahden komplementit desimaaliluvuille väliltä –1 – –128. Muuta luku kahden komplementiksi siten, että etsit luvun taulukosta, haet ensimmäisen heksamerkin vaakariviltä ja toisen merkin pystysarakkeesta.

Ensimmäinen heksamerkki \ Toinen heksamerkki	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
E	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
D	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
C	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
B	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
A	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
9	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
8	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128

Muunnosten laskeminen taskulaskimella

Kun lasket muunnoksia laskimella, laskin näyttää jakojäännökset desimaaleina. Jos esim. olet muuntamassa desimaalilukua 134 heksaan, jaat sen ensin 16:lla ja muutat sitten vastauksen ja jäännöksen heksamerkeiksi. Taskulaskin antaisi jakolaskun tulokseksi 8.375.

Jäännöksen muutat kokonaisluvuksi siten, että vähennät ensin desimaalipistettä edeltävän luvun ja kerrot loput luvulla, jota käytit jakolaskussa.

$8.375 - 8 = 0.375 \times 16 = 6$

Siis $134 \div 16 = 8$ jäännös 6. Eli desimaaliluku 134 on heksana 86.

Z80:n muistikkaat ja heksakoodit

Seuraavilla sivuilla on esitetty tässä kirjassa käytettyjen käskyjen muistikkaat ja heksakoodit. Luettelossa käytetty termi "implisiittinen osoitus" tarkoittaa yksinkertaisesti käskyjä, joiden heksakoodissa ei tarvitse ilmoittaa operandia. Tässä mainittujen lisäksi on olemassa joitain muitakin käskyjä. Jos haluat edetä pitemmälle konekielen oppimisessa, tarvitset täydellisen luettelon Z80:n käskykannasta (ks. s. 120). Luettelossa käytetään seuraavia lyhenteitä.

n = luku
nn = kaksitavuinen luku
r = rekisteri
rr = rekisteripari
x = osoite
c = ehto
d = siirtymä

ADC A,n Lisää luku n ja muistilippu akkuun. (Välitön osoitus.)

ADC A,n	CE,n

ADC A,r Lisää rekisteri r ja muistilippu akkuun. (Implisiittinen osoitus.)

ADC A,A	8F
ADC A,B	88
ADC A,C	89
ADC A,D	8A
ADC A,E	8B
ADC A,H	8C
ADC A,L	8D

ADC HL,rr Lisää rekisteriparin rr sisältö ja muistilippu HL:ään. (Implisiittinen osoitus.)

ADC HL,BC	ED4A
ADC HL,DE	ED5A
ADC HL,HL	ED6A

ADD A,n Lisää luku n akkuun. (Välitön osoitus.)

ADD,n	C6,n

ADD A,r Lisää rekisteri r akkuun. (Implisiittinen osoitus.)

ADD A,A	87
ADD A,B	80
ADD A,C	81
ADD A,D	82
ADD A,E	83
ADD A,H	84
ADD A,L	85

ADD HL,rr Lisää rekisteripari rr:n sisältö HL:ään. (Implisiittinen osoitus.)

ADD HL,BC	09
ADD HL,DE	19
ADD HL,HL	29

CALL x Mene aliohjelmaan alkaen osoitteesta x. (Välitön osoitus.)

CALL x	CD x

CALL c,x Mene aliohjelmaan alkaen osoitteesta x riippuen ehdosta c. c voi olla Z (sama); NZ (ei sama); C (muistilippu asetettu); NC (muistilippu nollattu); PE (parillinen); PO (pariton); M (miinus) tai P (plus). (Välitön osoitus.)

CALL Z,x	CC,x
CALL NZ,x	C4,x
CALL C,x	DC,x
CALL NC,x	D4,x
CALL PE,x	EC,x
CALL PO,x	E4,x
CALL M,x	FC,x
CALL P,x	F4,x

CCF Lisää muistilippuun 1. (Implisiittinen osoitus.)

CCF	3F

CP n Vertaa akun sisältöä dataan n. (Välitön osoitus.)

CP n	FE n

CP r Vertaa rekisterin r sisältöä akkuun (Implisiittinen osoitus.)

CP A	BF
CP B	B8
CP C	B9
CP D	BA
CP E	BB
CP H	BC
CP L	BD

CP (HL) Vertaa akun sisältöä HL:ssä olevan osoitteen sisältöön. (Epäsuora osoitus.)

CP (HL)	BE

DEC r Vähennä rekisteristä r yksi. (Implisiittinen osoitus.)

DEC A	3D
DEC B	05
DEC C	0D
DEC D	15
DEC E	1D
DEC H	25
DEC L	2D

DEC rr Vähennä rekisteriparista rr 1. (Implisiittinen osoitus.)

DEC BC	0B
DEC DE	1B
DEC HL	2B
DEC IX	DD2B
DEC IY	FD2B

DEC (HL) Vähennä HL:ssä olevan osoitteen sisällöstä 1. (Epäsuora osoitus.)

DEC (HL)	35

INC r Lisää rekisteriin r yksi. (Implisiittinen osoitus.)

INC A	3C
INC B	04
INC C	0C
INC D	14
INC E	1C
INC H	24
INC L	2C

INC rr Lisää rekisteripariin rr yksi. (Implisiittinen osoitus.)

INC BC	03
INC DE	13
INC HL	23

INC (HL) Lisää HL:ssä pidetyn osoitteen sisältöön yksi. (Epäsuora osoitus.)

INC (HL)	34

JP x Hyppää osoitteeseen x. (Välitön osoitus.)

JP x	C3 x

JP (rr) Hyppää osoitteeseen, joka on rekisteriparissa rr. (Implisiittinen osoitus.)

JP (HL)	E9
JP (IX)	DDE9
JP (IY)	FDE9

JP c,x Hyppää osoitteeseen x riippuen ehdosta c. c voi olla Z (sama); NZ (ei sama); C (muistilippu asetettu); NC (muistilippu nollattu); PE (parillinen); PO (pariton); M (miinus) tai P (plus). (Välitön osoitus.)

JP Z,x	CA,x
JP NZ,x	C2,x
JP C,x	DA,x
JP NC,x	D2,x
JP PE,x	EA,x
JP PO,x	E2,x
JP M,x	FA,x
JP P,x	F2,x

JR d Hyppää suhteessa. Hyppää d tavua (d on siirtymä). (Suhteellinen osoitus.)

JR d	18 d

JR c,d Hyppää suhteessa. Hyppää d tavua (d on siirtymä) riippuen ehdosta c. c voi olla NZ (ei sama); Z (sama); NC (muistilippu nollattu) tai C (muistilippu asetettu). (Suhteellinen osoitus.)

JR NZ,d	20,d
JR Z,d	28,d
JR NC,d	30,d
JR C,d	38,d

LD r,n Lataa rekisteriin r data n. (Välitör osoitus.)

LD A,n	3E,n
LD B,n	06,n
LD C,n	0E,n
LD D,n	16,n
LD E,n	1E,n
LD H,n	26,n
LD L,n	2E,n

LD rr,nn Lataa rekisteriparin rr kaksitavuinen luku nn. (Välitön osoitus.)

LD BC,nn	01,nn
LD DE,nn	11,nn
LD HL,nn	21,nn

LD A, (x) Lataa akkuun osoitteen x sisältö. (Kiinteä osoitus.)

LD A, (x)	3A, (x)

LD rr, (x) Lataa rekisteriparin rr osoitteiden x ja x+1 sisältö. (Kiinteä osoitus.)

LD BC, (x)	ED4B, (x)
LD DE, (x)	ED5B, (x)
LD HL, (x)	2A, (x)

LD A,r Lataa akkuun rekisterin r sisältö. (Implisiittinen osoitus.)

LD A,A	7F
LD A,B	78
LD A,C	79
LD A,D	7A
LD A,E	7B
LD A,H	7C
LD A,L	7D

LD B,r Lataa rekisteriin B rekisterin r sisältö. (Implisiittinen osoitus.)

LD B,A	47
LD B,B	40
LD B,C	41
LD B,D	42
LD B,E	43
LD B,H	44
LD B,L	45

LD C,r Lataa rekisteriin C rekisterin r sisältö. (Implisiittinen osoitus.)

LD C,A	4F
LD C,B	48
LD C,C	49
LD C,D	4A
LD C,E	4B
LD C,H	4C
LD C,L	4D

LD D,r Lataa rekisteriin D rekisterin r sisältö. (Implisiittinen osoitus.)

LD D,A	57
LD D,B	50
LD D,C	51
LD D,D	52
LD D,E	53
LD D,H	54
LD D,L	55

LD E,r Lataa rekisteriin E rekisterin r sisältö. (Implisiittinen osoitus.)

LD E,A	5F
LD E,B	58
LD E,C	59
LD E,D	5A
LD E,E	5B
LD E,H	5C
LD E,L	5D

LD H,r Lataa rekisteriin H rekisterin r sisältö. (Implisiittinen osoitus.)

LD H,A	67
LD H,B	60
LD H,C	61
LD H,D	62
LD H,E	63
LD H,H	64
LD H,L	65

LD L,r Lataa rekisteriin L rekisterin r sisältö. (Implisiittinen osoitus.)

LD L,A	6F
LD L,B	68
LD L,C	69
LD L,D	6A
LD L,E	6B
LD L,H	6C
LD L,L	6D

LD r, (rr) Lataa rekisteriin r rekisteriparissa rr pidetyn osoitteen sisältö. (Epäsuora osoitus.)

LD A, (BC)	0A
LD A,(DE)	1A
LD A,(HL)	7E
LD B,(HL)	46
LD C,(HL)	4E
LD D,(HL)	56
LD E,(HL)	5E
LD H,(HL)	66
LD L,(HL)	6E

LD (x),A Tallenna akun sisältö osoitteeseen x. (Kiinteä osoitus.)

LD (x),A	32,x

LD (x),rr Tallenna rekisteriparin rr sisältö osoitteisiin x ja x+1. (Kiinteä osoitus.)

LD (x),BC	ED43,x
LD (x),DE	ED53,x
LD (x),HL	22,x

LD (rr),r Tallenna rekisterin r sisältö rekisteriparissa rr pidettyyn osoitteeseen. (Epäsuora osoitus.)

LD (BC),A	02
LD (DE),A	12
LD (HL),A	77
LD (HL),B	70
LD (HL),C	71
LD (HL),D	72
LD (HL),E	73
LD (HL),H	74
LD (HL),L	75

LD (rr),n Tallenna data n osoitteeseen, joka on rekisteriparissa rr. (Välitön/epäsuora osoitus.)

LD (HL),n	36

RET Palaa aliohjelmasta. (Epäsuora osoitus.)

RET	C9

RET c Palaa aliohjelmasta riippuen ehdosta c. c voi olla Z (sama); NZ (ei sama); C (muistilippu asetettu); NC (muistilippu nollattu); PE (parillinen); PO (pariton); P (plus) tai M (miinus). (Epäsuora osoitus.)

RET Z	C8
RET NZ	C0
RET C	D8
RET NC	D0
RET PE	E8
RET PO	E0
RET M	F8
RET P	F0

SBC A,n Vähennä data n akusta, lainaa muistilipusta. (Välitön osoitus.)

SBC A,n	DE,n

SBC A,r Vähennä rekisterin r sisältö akusta. (Implisiittinen osoitus.)

SBC A,A	9F
SBC A,B	98
SBC A,C	99
SBC A,D	9A
SBC A,E	9B
SBC A,H	9C
SBC A,L	9D

SBC HL,rr Vähennä rekisteriparin rr sisältö rekisteriparista HL, lainaa muistilipusta. (Implisiittinen osoitus.)

SBC HL,BC	ED42
SBC HL,DE	ED52
SBC HL,HL	ED62

SBC A, (HL) Vähennä rekisteriparissa HL olevan osoitteen sisältö akusta, lainaa muistilipusta. (Epäsuora osoitus.)

SBC A,(HL)	9E

SCF Aseta muistilippu. (Implisiittinen osoitus.)

SCF	37

SUB n Vähennä data n akusta. (Välitön osoitus.)

SUB, n	D6, n

SUB r Vähennä rekisteri r:n sisältö akusta. (Välitön osoitus.)

SUB A	97
SUB B	90
SUB C	91
SUB D	92
SUB E	93
SUB H	94
SUB L	95

SUB (HL) Vähennä HL:ssä pidetyn osoitteen sisältö akusta. (Epäsuora osoitus.)

SUB (HL)	96

Arvoitusten vastaukset

Sivu 91 &A7 in desimaalilukuna 167.
513 on heksana &201.
Sivu 106 1. 25+73 (25 on &19 ja 73 on &49)

Vinkki: luvun kahden komplementti on helppo laskea siten, että vähennät luvun 256:sta ja muutat vastauksen heksaan. Esim. 256–6 = 250 joka on heksassa FA.

Z80 Muistikas	Heksakoodi	6502 Muistikas	Heksakoodi	Merkitys
LD A, &19	3E,19	LDA #&19	A9 19	Pane &19 akkuun.
ADD A, &49	C6,49	ADC #&49	69 49	Lisää &49 akkuun.
LD (osoite), A	32, osoite	STA osoite	8D osoite	Tallenna akun sisältö tiettyyn osoitteeseen.
RET	C9	RTS	60	Palautus.

2. 64+12+14 (64 on &40, 12 on &0C ja 14 on &0E)

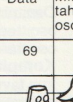
Z80 Muistikas	Heksakoodi	6502 Muistikas	Heksakoodi	Merkitys
LD A, &40	3E, 40	LDA &40	A9 40	Pane &40 akkuun.
ADD A, &0C	C6,0C	ADC &0C	69 0C	Lisää &0C akkuun.
ADD A, &0E	C6,0E	ADC &0E	69 0E	Lisää &0E akkuun.
LD (osoite), A	32, osoite	STA osoite	8D osoite	Tallenna akun sisältö tiettyyn osoitteeseen.
RET	C9	RTS	60	Palautus.

Arvoitusten vastaukset jatkuvat sivulla 137.

6502:n muistikkaat ja heksakoodit

Alla olevassa taulukossa on kaikkien tässä kirjassa käytettyjen käskyjen (sekä muutamien muiden käskyjen) muistikkaat ja heksakoodit. Muistikäskyt löytyvät taulukon vasemmasta reunasta ja käskyjen heksakoodit eri osoitustavoille löytyvät itse taulukosta. Nollasivun osoitus on aivan samaa kuin kiinteä osoitus eli operandi on osoite, jonne data on tallennettu, mutta osoitteen on oltava muistisivulla nolla (eli paikoissa 0–255; ks. s. 90). Implisiittinen osoitus tarkoittaa käskyjä, joissa ei tarvitse ilmoittaa operandia, esim. *CLC*. Tässä esitettyjen lisäksi on olemassa eräitä muitakin käskyjä. Jos haluat edetä pitemmälle konekielen oppimisessa, tarvitset täydellisen luettelon 6502:n käskykannasta.

Osoitustapa	Välitön	Kiinteä	Nollasivu	Indeksoitu X	Indeksoitu Y	Implisiittinen	Suhteellinen
Operandi on	Data	Mikä tahansa osoite	Osoite sivulla nolla	Osoite + X-rekisteri	Osoite + Y-rekisteri	Puut-tuu	Siirtymä
ADC Lisää tavu plus muistilippu akkuun.	69	6D	65	7D	79		
BBC Haaraudu jos muistilippu nollattu.							90
BCS Haaraudu jos muistilippu asetettu.							B0
BEQ Haaraudu jos samat.							F0
BMI Haaraudu jos miinus.							30
BNE Haaraudu jos eivät samat.							D0
BPL Haaraudu jos plus.							10
BVC Haaraudu jos ylivuoto nollattu.							50
BVS Haaraudu jos ylivuoto asetettu.							70
CLC Nollaa muistilippu.						18	
CMP Vertaa akkuun.	C9	CD	C5	DD	D9		
CPX Vertaa rekisteriin X.	E0	EC	E4				
CPY Vertaa rekisteriin Y.	C0	CC	C4				
DEC Vähennä muistipaikasta yksi.		CE	C6	DE			
DEX Vähennä X-rekisteristä 1.						CA	
DEY Vähennä Y-rekisteristä 1.						88	
INC Lisää muistipaikkaan 1.		EE	E6	FE			
INX Lisää X-rekisteriin 1.						E8	
INY Lisää Y-rekisteriin 1.						C8	
JMP Hyppää operandin ilmoittamaan osoitteeseen.		4C					
JSR Hyppää aliohjelmaan, joka alkaa operandissa ilmoitetusta osoitteesta.		20					
LDA Lataa akku.	A9	AD	A5	BD	B9		
LDX Lataa X-rekisteri.	A2	AE	A6		BE		
LDY Lataa Y-rekisteri.	A0	AC	A4	BC			
RTS Palaa aliohjelmasta.						60	
SBC Vähennä ja aseta muistilippu. Vähennä akusta ja lainaa muistilipusta.	E9	ED	E5	FD	F9		
SEC Aseta muistilippu.						38	
STA Tallenna akku tiettyyn osoitteeseen.		8D	85	9D	99		
STX Tallenna X-rekisteri tiettyyn osoitteeseen.		8E	86				
STY Tallenna Y-rekisteri tiettyyn osoitteeseen.		8E	86				
TAX Siirrä akku X-rekisteriin.						AA	
TAY Siirrä akku Y-rekisteriin.						A8	
TXA Siirrä X-rekisteri akkuun.						8A	
TYA Siirrä Y-rekisteri akkuun.						98	



Huomaa, että kaikkia käskyjä ei voi käyttää kaikissa osoitustavoissa.

Konekielen sanastoa

#-merkki. Risumerkkiä käytetään joissain mikroissa heksalukujen tunnuksena. 6502 prosessorissa se on datan tunnus.

&-merkki. Ja-merkkiä käytetään toisinaan heksalukujen tunnuksena.

Akku (accumulator). Rekisteri, jonne tallennetaan tietotavut, joille tehdään aritmeettisia tai loogisia toimituksia.

Alkukoodi (source code). Symbolisella konekielellä tai muulla korkean tason kielellä, esim. BASICilla, kirjoitettu ohjelma.

Aritmeettis-looginen yksikkö (Arithmetic Logic Unit, ALU). Keskusyksikön alue, jossa aritmeettiset ja loogiset toimitukset suoritetaan.

Assembler ks. kääntäjä.

Binäärijärjestelmä (binary). Lukujärjestelmä, jossa on kaksi numeroa, 0 ja 1 ja jossa luvun jokaisen numeron arvo on kaksinkertainen verrattuna sen oikealla puolella olevan numeron arvoon.

Bitti (bit). Tietokonekoodin perusyksikkö eli 1 tai 0, joka edustaa virtapulssia tai sen puuttumista.

Ehdollinen haarautuminen (conditional branch). Käsky, joka panee tietokoneen hyppäämään toiselle ohjelmariville riippuen tekstin tuloksesta.

Epäsuora osoitus (indirect addressing). Osoitustapa, jossa operandia käytetään datan osoittimena. Operandi voi olla osoite tai Z80:ssa rekisteripari. Operandissa on datan osoite.

Haarautuminen (branching). Käsky, joka panee tietokoneen hyppäämään jollekin toiselle ohjelmariville.

Heksadesimaali eli heksa (hex). Lukujärjestelmä, joka käyttää 16:tta merkkiä (numeroita 0–9 ja kirjaimia A–F). Jokaisen heksaluvun merkin arvo on 16 kertaa sen oikealla puolella olevan merkin arvo.

Heksalataaja (hex loader). BASIC-ohjelma, joka muuntaa konekielisen ohjelman heksakoodit desimaaliluvuiksi ja pistää ne tietokoneen muistiin.

HIMEM. Käyttäjä-RAMin korkein osoite.

Hyppy (jump). Käsky, joka panee tietokoneen menemään jollekin toiselle ohjelmariville.

Implisiittinen osoitus (implicit tai implied addressing). Osoitustapa, jossa operandi tiedetään eikä sitä tarvitse ilmoittaa.

Indeksirekisterit (index registers). Indeksoidussa osoituksessa käytetyt rekisterit. 6502:ssa näitä käytetään myös yleisrekistereinä.

Indeksoitu osoitus (indexed addressing). Osoitustapa, jolla indeksirekisteriin sisältö lisätään käskyssä annettuun osoitteeseen, jotta saadaan datan varsinainen osoite.

Järjestelmämuuttujat (system variables). RAMin muistipaikat, joissa on tietoa mikron senhetkisestä tilasta.

Kahden komplementti (two's complement). Merkintäjärjestelmä, jolla esitetään negatiivisia lukuja. Luvun kahden komplementti saadaan komplementoimalla (muuttamalla kaikki 1:t 0:ksi ja 0:t 1:ksi) luvun binäärivastine ja lisäämällä 1.

Kiinteä osoite (absolute address). Datan varsinainen osoite.

Kiinteä osoitus (absolute addressing). Osoitustapa, jossa käsky sisältää datan osoitteen. Sama kuin laajennettu tai suora osoitus.

Komplementti (complement) saadaan, kun tavun kaikki 0:t muutetaan 1:ksi ja 1:t muutetaan 0:ksi.

Korkein muistiosoite (top of memory). Käyttäjä-RAMin korkein muistiosoite.

Käsky (instruction). Keskusyksikön suoritettavaksi annettu toimitus.

Käskykanta (instruction set). Kaikki käskyt, jotka tietty mikroprosessori pystyy suorittamaan.

Käyttäjä-RAM (user RAM). RAMin osa, jonne BASIC-ohjelmat tallennetaan.

Käyttöjärjestelmä (operating system). Konekielellä kirjoitettuja ohjelmia, jotka on tallennettu tietokoneen ROMiin ja jotka kertovat tietokoneelle, miten sen tulee toteuttaa saamansa tehtävät.

Kääntäjä (assembler). Ohjelma, joka muuntaa symbolisen konekielen muistikkailla kirjoitetut käskyt tietokoneen omaksi koodiksi.

LIFO (last in/first out). Viimeisenä sisään, ensimmäisenä ulos. Kuvaa menetelmää, jolla tietokone tallentaa tietoa pinoon.

Lippu (flag). Lippurekisterin bitti, jota käytetään merkkinä tietystä tilasta, esim. negatiivisen luvun tai yhteenlaskussa muistiin panemisen merkkinä.

Miinuslippu (sign flag). Lippurekisterin bitti, joka ilmoittaa, onko luku positiivinen vai negatiivinen.

Mikroprosessori (microprocessor) eli suoritin. Lastu, jolla on tietokoneen keskusyksikkö ja joka toteuttaa ohjelman käskyt ja ohjaa tietokoneen kaikkia muita toimintoja.

Muistikas (mnemonic). Symbolisessa konekielessä käytetty kirjainkoodi, joka esittää tietokoneen oman koodin käskyjä helposti muistettavassa muodossa. Muistikkaat perustuvat käskyjen englanninkielisiin nimiin.

Nollalippu (zero flag). Lippurekisterin bitti, joka ilmoittaa, jos toimituksen tulos on 0 ja jota käytetään myös ilmoittamaan, ovatko kaksi tavua yhtä suuret.

Nollasivu (zero page). Muistin 256 ensimmäistä paikkaa.

Nollasivun osoitus (zero page addressing). Käytetään vain 6502:ssa osoitustapana, jossa operandi on osoite muistin sivulla nolla (eli paikoissa 0–255).

Nollata (clear). Muuttaa bitti, esim. lippurekisterin bitti, nollaksi.

Näyttömuisti (screen memory). RAM-paikat, joissa pidetään näyttöruudulla näkyvää tietoa.

Oikea tavu (low order byte). Kaksi osoitteen heksamerkkiä, jotka kertovat osoitteen paikan muistisivulla. Tarkoittaa myös niitä kahta heksamerkkiä, jotka ilmoittavat, monellako yksiköllä luku ylittää 255.

Operandi (operand). Käskyn osa, joka kertoo tietokoneelle, mistä käsiteltävä tieto löytyy.

Osoite (address). Luku, jolla tunnistetaan jokin tietokoneen muistin paikka.

Osoitelaskuri (program counter). Rekisteri, jossa on seuraavan muistista noudettavan käskyn osoite.

Osoitin (pointer). Muistipaikka (tai rekisteripari), jossa on jonkin datan osoite.

Osoitustavat (addressing modes). Eri tavat, joilla tietokoneelle voi kertoa, mistä konekielisessä ohjelmassa käytettävä data löytyy.

Pino (stack). Muistialue, jota tietokone käyttää väliaikaiseen tallennukseen ja josta viimeksi tallennettu data noudetaan aina ensimmäisenä pois.

Pino-osoitin (stack pointer). Keskusyksikön rekisteri, jossa on pinon viimeisen datan osoite.

Puskuri (buffer). Väliaikainen varastointipaikka tietokoneen muistissa. Dataa säilytetään täällä sen ollessa matkalla määränpäähänsä.

RAMTOP. Käyttäjä-RAMin korkein osoite.

Rekisterit (registers). Keskusyksikön paikat, joissa pidetään käsky-, data- ja osoitetavuja sinä aikana, jolloin tietokone käsittelee niitä.

Siirtymä (displacement). Hypyssä tai haarautumisessa käytetty luku, joka ilmoittaa, monenko paikan yli tietokoneen tulee hypätä päästäkseen seuraavaan käskyyn.

Sivu (page). Muistin osa. Yleensä mikroissa sivulla on 256 muistipaikkaa.

Suhteellinen osoitus (relative addressing). Osoitustapa, jossa tietokone laskee seuraavan käskyn osoitteen lisäämällä osoitelaskurissa olevaan osoitteeseen luvun nimeltä siirtymä.

Suora osoitus (direct addressing). Ks. kiinteä osoitus.

Symbolinen konekieli (assembly language). Ohjelmointimenetelmä, jossa käytetään konekielen käskyjen asemesta kirjainkoodeja, ns. muistikkaita.

Tavu (byte). Kahdeksan ykkösen tai nollan eli bitin ryhmä, joka edustaa yhtä tietoalkiota tietokoneen koodissa.

Tilarekisteri (processor status register). 6502:ssa tämä on lippurekisterin nimi (siis rekisterin, jonka bittejä käytetään ilmoittamaan tiettyjä tietokoneen sisäisiä tiloja).

Toimituskoodi (opcode). Käskyn osa, joka kertoo tietokoneelle, mitä sen pitää tehdä.

Tulkki (interpreter). Ohjelma, joka kääntää BASICin (tai muun korkean tason kielen) käskyjä tietokoneen omalle koodille.

Tuloskoodi (object code). Ohjelma, joka on käännetty konekielelle symbolisesta konekielestä tai jostain muusta korkean tason kielestä.

Vasen tavu (high order byte). Heksaosoitteen kaksi ensimmäistä merkkiä, jotka ilmoittavat muistisivun, jolla osoite on. Tarkoittaa myös niitä kahta merkkiä, jotka kertovat montako 256:tta on luvussa, joka on suurempi kuin 255.

Vastakääntäjä (disassembler). Ohjelma, joka näyttää muistipaikkojen sisällön konekielisinä ruudulla. Saatavana kasetilla. Voidaan käyttää konekielisten ohjelmien vikojen etsimiseen ja mikron ROMissa olevien ohjelmien tutkimiseen.

Välitön osoitus (immediate addressing). Osoitustapa, jossa käskyn data sisältyy käskyyn.

Vastaukset BASIC-ohjelmiin

Seuraavilla sivuilla on vastaukset kaikkiin kirjan ensimmäisen osan BASIC-arvoituksiin ja tehtäviin. Ohjelmat on kirjoitettu yleis-BASICilla, joten joudut ehkä muuttamaan jotkin käskyt – esim. RND ja CLS – omalle mikrollesi sopiviksi. Jos jokin alla listatuista ohjelmista ei toimi, tarkista ensin, tunteeko mikrosi kaikki ohjelmassa käytetyt BASIC-sanat. Sivulla 140 on muunnostaulukko. Ohjelmarivit, joille on tehtävä isompia muutoksia Sinclair (Timex) tietokoneita varten, on merkitty △ ja tilalle tulevat rivit on annettu lähellä tätä merkkiä. Rivit, jotka täytyy muuttaa vain ZX81:lle (Timex 1000) on merkitty ▲.

Joissain arvoituksissa voi käydä niin, että itse kirjoittamasi ohjelma eroaa vastauksissa listatusta ohjelmasta. Älä välitä eroista, jos ohjelmasi toimii kunnolla. Tutki kuitenkin vastauksia ja tarkista, että ohjelmasi on mahdollisimman lyhyt ja siisti.

PRINT-harjoitukset (sivut 14–15)

Helppoja ohjelma-arvoituksia

1
```
40 PRINT "HEI";
50 PRINT A$
```

Puolipiste saa mikron pysymään samalla rivillä, kun se printtaa A$:n.

2
```
40 PRINT "HEI",A$
50 Poista tämä rivi
```

Pilkku jättää tyhjiä välilyöntejä ennen A$:a.

3
```
40 PRINT TAB(6);"HEI ";A$
50 Poista tämä rivi
```

Joissain mikroissa ei TAB-käskyn jälkeen tarvita puolipistettä.

4
```
40 PRINT "      NAKEMIIN"
50 PRINT "      ";A$
```

Muuttuja-arvoitukset (sivut 16–17)

Muuttujan nimen valinta

Seuraavissa nimissä on BASIC-sanoja, joten niitä ei voi käyttää muuttujan niminä: LETTER$ (LET); RUN$ (RUN). ZX81 hyväksyy vain yhden kirjaimen mittaisia merkkijonomuuttujan nimiä, joten tässä mikrossa et voi käyttää nimeä KIRPPU$.

PRINT-arvoitus

```
 10 LET A=66
 20 LET B=77
▲30 LET RR$="RUOSTEISTA ROBOTTIA"
 40 PRINT A;" ";RR$;" SOI"
 50 PRINT B;" KARISTETTYA MAKKARAA"
```

Lainausmerkkien sisällä oleva välilyönti antaa ruudullekin välilyönnin.

Kurkistamme ohjelman sisälle

```
40 LET C=9/8
45 PRINT "C ON ";C
```

Celsius-asteista Fahrenheitiin

```
40 LET C=9/8
50 LET D=C*A
60 LET R=D+F
```

Muuta rivit 40–60 yllä annettuun muotoon.

Ohjelman kirjoitus: arvoituksia

1
```
10 LET U$="JUG"
20 PRINT "ANNA SANA "
30 INPUT W$
40 PRINT "JUGLIAANIN KIELELLÄ SANA ON ";
50 PRINT U$;W$
```

Rivillä 50 mikro kirjoittaa JUG ja sen perään W$:iin tallennetun sanan.

2
```
10 PRINT "MIKA ON MATKA ";
20 INPUT D
30 PRINT "KAUANKO MATKA KESTI ";
40 INPUT T
50 LET S=D/T
60 PRINT "NOPEUTESI OLI ";S;
70 PRINT "KILOMETRIA TUNNISSA"
```

Rivillä 50 mikro laskee nopeuden ja tallentaa sen S:ään.*

3
```
10 LET A=30
20 LET B=20
30 LET S=35
40 LET T=S/B
50 LET C=A*T
60 LET D=C+S
70 PRINT "NIIDEN ON OSTETTAVA ";
80 PRINT D;" MAKKARAA."
90 PRINT " SYÖMISEEN MENEE ";
100 PRINT T;" TUNTIA"
```

A ja B ilmoittavat, montako makkaraa robotit pystyvät syömään tunnissa. S kertoo, montako makkaraa robotti 2 haluaa syödä ja T on aika, jossa robotti 2 syö makkaransa (S). Rivillä 50 lasketaan, montako makkaraa robotti 1 syö ajassa T ja D on makkaroiden kokonaismäärä.

Asioiden toisto (18–19)

Silmukka-arvoituksia

1
```
10 FOR J=1 TO 100
20 PRINT "HEI ";
30 NEXT J
```

Puolipiste rivillä 30 pitää mikron samalla rivillä seuraavan sanan kirjoittamista varten.

2
```
10 FOR J=1 TO 25
20 PRINT TAB(15);"HEI"
30 NEXT J
```

Tämä saa mikron kirjoittamaan alekkain 25 kertaa HEI, 15 välilyönnin päähän ruudun vasemmasta reunasta.

*Joissain mikroissa (esim. VIC 20) pitää PRINT-riveillä ennen INPUT-sanaa oleva puolipiste jättää pois.

3 Rivi 20 on virheellinen, koska se häiritsee silmukkalaskuria I. Aina, kun silmukka toistuu, rivi 20 muuttaa I:n arvon 1:stä 0:ksi.

1
```
10 FOR I=25 TO 1 STEP -1
20 PRINT TAB(I);"HEI"
30 NEXT I
```

Käytä silmukan lukua TAB-lukuna. Varmista, ettei I ole suurempi kuin ruutusi leveys. Aina, kun silmukka toistuu, I vähenee 1:llä.

2
```
10 FOR L=5 TO 0 STEP-1
20 PRINT TAB(5);L;TAB(10);L*L
30 NEXT L
```

Askel –1 saa silmukkamuuttujan laskemaan takaperin 5:stä 0:aan. Aina, kun silmukka toistuu, rivi 20 kirjoittaa L:n arvon ja korottaa sen neliöön.

3
```
10 PRINT "ALKU ";
20 INPUT A
30 PRINT "LOPPU ";
40 INPUT B
50 PRINT "ASKEL ";
60 INPUT C
70 FOR J=A TO B STEP C
80 PRINT J;" ";
90 NEXT J
```

Rivien 70–90 silmukka asettaa A:n silmukkamuuttujan alkuarvoksi, B:n sen loppuarvoksi ja C:n askeleksi.

Salainen viesti

```
10 PRINT "SALAINEN VIESTI"
20 PRINT "OPETTELE 5 SEKUNNISSA"
30 PRINT "SITTEN SE KATOAA"
40 PRINT
50 PRINT "KOHTAA AGENTTI X KLO 14 ASEMALLA"
60 FOR I=1 TO 1000          ---- Viivesilmukka
70 NEXT I
80 CLS                      ---- Tyhjentää näytön
```

Kuvioarvoituksia

2
```
50 PRINT TAB(A-K);"*";
60 PRINT TAB(A+K);"*"
70 NEXT K
```

A on kolmion huippu. Vasemman puolen piirrät vähentämällä A:sta K ja oikean puolen piirrät lisäämällä A:han K.

3
```
30 Poista tämä rivi.
40 FOR K=9 TO 1 STEP -1
80 PRINT TAB(A);"*"
```

Muuta ohjelmaan nämä rivit, jotta kuvio piirtyy ylösalaisin. Sivun 9 muut kuviot saat samalla ohjelmalla muuttamalla TAB-paikkoja.

Silmukka-arvoitukset (sivut 20–21)

Silmukka, joka hidastaa ohjelmaa

Voit lisätä tämän sisäkkäisen silmukan riville 45 tai 55.

```
45 FOR K=1 TO 1000
48 NEXT K
```

Virheitä silmukoissa

Riveillä 50 ja 70 on virhe. Rivien pitäisi olla tällaiset:

```
50 NEXT K
70 NEXT I
```

Binäärilaskuri

Tarvitset neljä lisäsilmukkaa (E, F, G, H) ohjelmaan, joka laskee kahdeksanmerkkisiä lukuja. Numeroi ohjelmarivit uudestaan ja muuta PRINT-rivit tällaiseen muotoon:

```
PRINT H+G*2+F*4+E*8+D*16+C*32+B*64+
A*128;" ";
PRINT A;B;C;D;E;F;G;H
```

Välkkyvä viesti

```
10 FOR J=1 TO 10
20 CLS
30 FOR K=1 TO 1000: NEXT K
40 PRINT TAB(10);"VAARA"
50 PRINT TAB(7);"HYÖKKÄYS AVARUUDESTA"
60 FOR K=1 TO 1000: NEXT K
70 NEXT J
```

Tämä ohjelma kirjoittaa välkkyvän viestin. Joudut ehkä muuttamaan viivesilmukan kokoa ja TAB-paikkoja näyttöösi sopiviksi. Jos mikrosi ei hyväksy kahta käskyä samalla rivillä, pane käskyt NEXT K omille riveilleen.

Tietokonekello

Minuutti-silmukka

```
10 FOR J=0 TO 59
20 FOR K=0 TO 59
30 PRINT J;":";K
40 FOR L=1 TO 500:NEXT L      ---- Sekunti-silmukka
50 CLS
60 NEXT K
70 NEXT J
```

Aseta K-silmukka siten, että viive on 1 sekunti.

Raketin lähtö

```
▲  10 CLS
   20 FOR I=1 TO 20
   30 PRINT
   40 NEXT I
   50 PRINT "   *   "
   60 PRINT "  ***  "
   70 PRINT "  ***  "
   80 PRINT " ***** "
   90 PRINT " ** ** "
  100 FOR J=1 TO 25
▲ 110 PRINT
  120 FOR K=1 TO 1000          ---- Viivesilmukka
  130 NEXT K
  140 NEXT J
```

Rivien 20–40 silmukka kirjoittaa tyhjiä rivejä, joten raketti on ruudun alareunassa. Rivit 50–90 printtaavat raketin. Pane rivien 100–140 silmukka toistumaan niin monta kertaa, kuin ruudullasi on rivejä. Aina, kun silmukka toistuu, tietokone kirjoittaa tyhjän rivin ja raketti siirtyy ruudulla ylöspäin.

Sinclairin (Timex) muutokset

Paina Sinclairissa ENTER, kun Scroll-viesti ilmestyy ruutuun.

▲ Muuta nämä rivit ZX81:teen:

```
 20 FOR I=1 TO 17
110 SCROLL
```

Hyppivä mies -ohjelma

```
10 FOR J=1 TO 25 STEP 2
20 CLS
30 PRINT
40 PRINT TAB(J);" O "
50 PRINT TAB(J);"<O>"
60 PRINT TAB(J);"/\"
70 FOR K=1 TO 1000
80 NEXT K
90 CLS
100 PRINT TAB(J+1);"<O>"
110 PRINT TAB(J+1);" O "
120 PRINT TAB(J+1);"/\"
130 FOR K=1 TO 1000
140 NEXT K
150 NEXT J
```

Pane rivien 10–150 silmukka toistumaan niin monta kertaa, kuin ruudullesi mahtuu merkkejä rinnakkain. Rivit 40–60 printtaavat miehen ensimmäisessä asennossa. Rivit 70–80 ja 130–140 ovat viivesilmukoita, jotka pitävät miestä ruudulla hetken aikaa. Rivit 100–120 printtaavat miehen toisessa asennossa.

IF/THEN-harjoitukset (sivut 22–23)

Kertolaskuja

```
55 IF B<>J*A THEN PRINT "VAARIN. ";
J;" X ";A;" = ";J*A
```

Lisää ohjelmaan tämä rivi, jolla mikro kertoo, jos vastauksesi on väärä.

Tunnussana

```
50 IF P$<>S$ THEN PRINT "VAARIN,
PAKSUPAA"
60 IF P$=S$ THEN PRINT "OK. VOIT JATKAA"
```

Lisää nämä rivit (varustettuna omalla viestilläsi) tunnussanaohjelmaan.

```
10 LET S$="MAKKARA"
20 LET N=007
30 PRINT "ANNA TUNNUSSANA ";
40 INPUT P$
50 PRINT "ANNA SALAINEN NUMERO ";
60 INPUT SN
70 IF P$<>S$ OR SN<>N THEN PRINT "VAARIN,
PAKSUPAA"
80 IF P$=S$ AND SN=N THEN PRINT
"OK. VOIT JATKAA"
```

Muuta tunnussanaohjelma tällaiseksi. Nyt tietokone kysyy myös salaista numeroa.

Tietokonelaskin

```
10 PRINT "ANNA LUKU ";
20 INPUT X
30 PRINT "ANNA TOINENKIN ";
40 INPUT Y
50 PRINT "HALUATKO:"
60 PRINT "LASKEA YHTEEN, VAHENTAA"
70 PRINT "JAKAA VAI KERTOA ?"
80 INPUT A$
90 IF A$="LASKEA YHTEEN" THEN LET A=X+Y
100 IF A$="VAHENTAA" THEN LET A=X-Y
110 IF A$="JAKAA" THEN LET A=X/Y
120 IF A$="KERTOA$" THEN LET A=X*Y
130 PRINT "VASTAUS ON ";A
```

Arvausleikit

```
45 IF Y<X THEN PRINT "LIIAN PIENI"
46 IF Y>X THEN PRINT "LIIAN ISO"
```

Lisää ohjelmaan nämä rivit, joilla mikro kertoo, jos arvauksesi on liian iso tai pieni.

```
5 LET N=0
55 LET N=N+1
56 IF N>5 THEN STOP
```

Lisää ohjelmaan nämä rivit, jotka laskevat arvauskertojen määrän ja lopettavat ohjelman viiden arvauksen jälkeen.

Sanojen arvaus

```
10 PRINT "ANNA SANA ";
20 INPUT W$
30 PRINT "ANNA VIHJE ";
40 INPUT C$
50 CLS
60 PRINT "VIHJE :";
70 PRINT C$
80 PRINT "ARVAA SANA"
90 INPUT G$
100 IF G$=W$ THEN GOTO 130
110 PRINT "EI"
120 GOTO 90
130 PRINT "KYLLA"
```

Riveillä 10–40 mikro pyytää sanaa ja vihjettä. Rivit 50–60 tyhjentävät näytön ja kirjoittavat sille vihjeen.
Rivi 120 lähettää mikron takaisin riville 90 toista arvausta varten.

Ravikilpailut

Tässä ovat raviohjelman IF/THEN rivit täydellisinä.

```
80 IF G1=H1 AND G2=H2 THEN GOTO 160
90 IF G1=H2 OR G2=H1 THEN GOTO 150
100 IF (G1=H1 AND G2<>H2) OR (G2=H2
AND G1<>H1) THEN GOTO 140
120 IF N=4 THEN GOTO 170
```

Rivi 80: Lähetä mikro riville 160 jos molemmat arvaukset ovat oikein.
Rivi 90: Jos hevonen on oikein arvattu mutta sijoitus ei, lähetä mikro riville 150.
Rivi 100: Jos yksi arvaus on oikein, mene riville 140.
Rivi 120: Neljän arvauksen jälkeen lähetä mikro riville 170.

Vihjeitä pelin parantamiseksi

```
15 LET S=0
140 PRINT "YKSI OIKEA ARVAUS";LET
S=2:GOTO 120
160 PRINT "OIKEIN";LET S=4
180 PRINT "SINULLA ON ";S;" PISTETTA"
```

Tässä on yksi tapa panna ohjelma laskemaan pisteitä; molempien hevosten arvaamisesta saa 4 pistettä ja yhdestä hevosesta 2 pistettä. Pisteet tallennetaan muuttujaan S.

```
190 PRINT "HALUATKO PELATA UUDESTAAN ? (K/E)"
200 INPUT Z$
210 IF Z$="K" THEN GOTO 10
220 IF Z$="E" THEN STOP
```

Kun lisäät nämä rivit, pelaaja voi halutessaan pelata uuden pelin.

 Tietokone suorittaa eri laskutoimituksen riippuen siitä, minkä valinnan teet rivillä 80.

Satunnaisluvut (sivut 24–25)

Satunnaisluku väliltä 10–20

```
INT (RND(1)*11+10)
```

Satunnaisluvut saat näin: pane kertojaksi luku-välin lukujen määrä (11) ja lisää välin ensimmäinen luku (10). Käytä mikrosi RND-käskyä.

Ohjelma-arvoitus

```
20 LET X=INT(RND(1)*20+1)
```

Muuta numeroiden arvauspelin rivi 20, jotta mikro valitsee satunnaisluvun väliltä 1–20.

Lukusarja

Lisää nämä rivit lukusarjaohjelmaan.

```
5 PRINT "ARVAA SEURAAVA LUKU"
7 PRINT "TASSA SARJASSA"
10 LET X=INT(RND(1)*10+1)
20 LET Y=INT(RND(1)*10+1)
60 LET A=X+4*Y
70 INPUT N
80 IF N=A THEN PRINT "OIKEIN"
90 IF N<>A THEN PRINT "VAARIN. LUKU
ON ";A
100 GOTO 5
```

Rivi 60 laskee sarjan seuraavan luvun.

Satunnaislukusarja

```
24 LET R=INT (RND(1)*3+1)
25 IF R=1 THEN GOTO 30
26 IF R=2 THEN GOTO 40
27 IF R=3 THEN GOTO 50
```

Satunnaisen lukusarjan saat, kun panet mikron valitsemaan satunnaisluvun väliltä 1–3 ja tallennat sen R:ään. Lähetä mikro eri sarjaan R:n arvosta riippuen.

```
30 FOR I=1 TO 3
33 PRINT X+I*1
35 NEXT I
37 LET A=X+4*4
39 GOTO 70
40 FOR I=1 TO 3
43 PRINT I*I-Y
45 NEXT I
47 LET A=4*4-Y
49 GOTO 70
50 FOR I=1 TO 3
53 PRINT X+Y-I*I
55 NEXT I
57 LET A=X+Y-4*4
```

Lisää ohjelmaan kolme erilaista lukusarjaa siten, että poistat rivit 30–60 ja lisäät joka sarjalle oman silmukan. Lähetä mikro kahden ensimmäisen silmukan lopussa riville 70.

Pako Zorgosista

```
10 LET C=10
20 PRINT "SINULLA ON ";C;" LASTUA"
30 PRINT "MONTAKO PANET PELIIN :";
40 INPUT B
50 IF B>C THEN PRINT "SINULLA EI OLE
NIIN MONTAA LASTUA":GOTO 20
60 LET C=C-B
70 LET X=INT(RND(1)*6+1)
80 LET Y=INT(RND(1)*6+1)
90 PRINT "NOPANHEITTO, PAINA P:";
100 INPUT P$
110 IF P$<>"P" THEN GOTO 90
120 PRINT TAB(5);X;TAB(10);Y
130 IF X=Y THEN GOTO 250
140 IF X+Y=10 OR X+Y=11 THEN GOTO 210
150 IF X+Y=6 OR X+Y=7 THEN GOTO 190
160 PRINT "VALITAN. MENETIT LASTUSI"
170 IF C=0 THEN GOTO 290
180 GOTO 20
190 PRINT "EI MUUTOSTA. PIDA LASTUSI"
200 LET C=C+B:GOTO 20
210 PRINT "HIENOA. VOITIT 3-KERTAISEN MAARAN"
220 LET C=C+(B*3)
230 IF C>=50 THEN GOTO 320
240 GOTO 20
250 PRINT "VOITIT 2-KERTAISEN MAARAN"
260 LET C=C+(B*2)
270 IF C>=50 THEN GOTO 320
280 GOTO 20
290 PRINT "VALITAN. SINULTA LOPPUIVAT LASTUT"
300 PRINT "ET PAASE PAKOON ZORGOSISTA"
310 STOP
320 PRINT "HIENOA. NYT PAASET PAKOON"
330 PRINT "ZORGOSISTA MUKANASI ";C;" LASTUA"
```

C laskee lastujen määrän.

Varmistaa, ettei pelaajan panos ylitä hänen lastujensa määrää.

Laske, montako lastua on jäljellä.

Valitsee kaksi satunnaislukua väliltä 1–6.

Tarkistaa että pelaaja painaa P.

Kirjoittaa satunnaisluvut ruudulle.

Tarkistaa luvut ja lähettää mikron eri riveille tuloksesta riippuen.

Tarkistaa jäljellä olevien lastujen määrän. Jos C = 0 lähettää mikron ohjelman loppuun.

Lähettää mikron takaisin uutta erää varten.

Muuttaa C:n arvoa.

Jos C on 50 tai enemmän, lähettää mikron ohjelman loppuun.

Paperi, kivi vai sakset: etsi virheet

Korjaa seuraavat rivit, jotta ohjelma toimii kunnolla.

```
50 LET R=INT(RND(1)*3+1)
80 IF R=3 THEN LET C$="SAKSET"
150 IF C$="SAKSET" AND A$="KIVEN"
THEN LET F=1
190 PRINT "JOTEN ";
210 IF F=1 THEN PRINT "SINA VOITAT!"
230 IF F=0 THEN LET C=C+1
240 IF F=1 THEN LET A=A+1
280 IF C<10 AND A<10 THEN GOTO 40
```

Merkkien murskaus (sivut 26–27)

Jonoarvoitukset

1 △ `40 PRINT TAB(K);MID$(N$,K,1)`

Tämä saa mikron kirjoittamaan sanasta yhden kirjaimen kerrallaan TAB-paikkaan K alkaen kirjaimesta numero K.

2

```
10 LET K$="KENGURU"
20 LET L=LEN(K$)
30 FOR J=L TO 1 STEP-1
△ 40 PRINT MID$(K$,J,1);
50 NEXT J
```

Käytä silmukkaa, jonka askel on –1, jotta tietokone kirjoittaa sanan takaperin.

3 △ `50 PRINT RIGHT$(S$,L-J)+LEFT$(S$,J)`

Aina, kun silmukka toistuu, RIGHT$(S$,L-J) saa mikron kirjoittamaan sanan NELIÖT oikeasta päästä L (sanan pituus) miinus J kirjainta ja lisäämään J kirjainta sanan vasemmasta päästä. Aja ohjelma käyttäen eri sanoja.

Pisin sana

```
50 IF LEN(W$)>LEN(A$)
THEN LET A$=W$
```

Aina, kun silmukka toistuu, tietokone vertaa W$:n pituutta A$:iin. Jos W$ on pitempi kuin A$, tietokone panee W$:n A$:n tilalle ja vertaa sitten seuraavaa sanaa. Silmukan lopussa A$:ssa on pisin syötetyistä sanoista.

Lyhyin sana

```
10 LET A$="XXX!!!&&ABC***123!!!!
XXXXXXXXXX"
50 IF LEN(W$)<LEN(A$)
THEN LET A$=W$
70 PRINT "LYHYIN SANA:"
```

Ohjelma on sama kuin pisimmän sanan ohjelma rivejä 10 ja 50 lukuun ottamatta. Rivillä 10 on merkkijono, johon tietokone vertaa eri sanojen pituutta. On aivan sama, mitä merkkejä panet A$:iin.

Sanaeditori

Tässä ovat sanaeditoriohjelman vajaat ohjelmarivit täydellisinä:

```
40 LET S$=" "+S$+" "
70 LET W$=" "+W$+" "
```

S$:n ja W$:n molemmille puolille pitää lisätä välilyöntejä, jotta mikro etsii rivillä 140 vain kokonaisia sanoja. Ilman välilyöntejä tietokone poimisi lauseesta minkä tahansa merkkijonon, joka on samassa järjestyksessä kuin S$ tai W$. Aja ohjelma ilman välilyöntejä, niin näet, miten siinä käy.

```
100 LET LS=LEN(S$)
110 LET LW=LEN(W$)
```

Nämä rivit panevat LS:n pituudeksi S$:n pituuden ja LW:n pituudeksi W$:n pituuden. Rivi 100 toistuu rivillä 160, jotta LS on uuden lauseen pituinen.

```
△ 140 IF MID$(S$,K,LW)=W$
THEN LET A$=S$
```

Aina, kun silmukka toistuu, MID$(S$,K,LW) saa mikron tarkistamaan LW (W$:n pituus) merkkiä S$:sta alkaen merkistä K. Kun mikro löytää merkkijonon, joka on sama kuin W$, se tallentaa koko lauseen uuteen muuttujaan A$.

```
△ 150 IF A$=S$ THEN LET S$=LEFT$(A$,K)
+N$+RIGHT$(A$,LS-(K+LW-2))
```

LEFT$(A$,K) laskee, montako merkkiä vaihdettavan sanan vasemmalla puolen on. Tämän rivin toiminta on helpointa ymmärtää, jos kokeilet sitä johonkin lauseeseen paperilla kirjoittaen.

```
180 IF K<=LS-LW+1 THEN GOTO 140
```

Lähettää mikron takaisin riville 140 tarkistamaan loput lauseesta siltä varalta, että vaihdettava sana esiintyy siinä kaksi kertaa.

Muutokset Sinclair (Timex) tietokoneille

Jonoarvoitukset

Lisää nämä rivit jonoarvoituksien vastauksiin 1, 2 ja 3.

1 `40 PRINT TAB(K);N$(K TO K)`

2 `40 PRINT K$(J TO J);`

3 `50 PRINT S$(J+1 TO )+S$( TO J)`

Sanaeditori

```
140 IF S$(K TO K+LW-1)=W$ THEN
LET A$=S$
150 IF A$=S$ THEN LET S$=A$( TO K)
+N$+A$(K+LW-1 TO )
```

Vaihda nämä rivit sanaeditoriohjelmaan.

Lisää merkkien murskausta (sivu 28)

Kirjainarvoitukset

1 ▲
```
10 FOR K=65 TO 90
```

Nämä ovat ASCII-luvut, joilla kirjoitetaan aakkoset. ZX81:ssä (Timex 1000) lukujen pitää olla 38–63.

2
```
10 FOR K=97 TO 122
20 PRINT CHR$(K);
30 NEXT K
```

Näillä luvuilla kirjoitetaan aakkoset pienillä kirjaimilla. (Mm. Dragonissa ja muissa tietokoneissa, jotka eivät osaa kirjoittaa pienaakkosia, näitä lukuja käytetään erikoissuuraakkosten tunnuksina.)

3 ▲
```
10 LET R=INT(RND(1)*26+65)
20 PRINT CHR$(R);
30 GOTO 10
```

Tämä on helpoin tapa kirjoittaa satunnaiskirjaimia. Pane ZX81:ssä 65:n eli aakkosten ensimmäisen koodinumeron tilalle 38.

Kirjainvertailu

```
10 INPUT X$,Y$
20 IF X$<Y$ THEN PRINT X$;" ON
ENNEN KUIN ";Y$
30 IF Y$<X$ THEN PRINT Y$;" ON
ENNEN KUIN ";X$
40 GOTO 10
```

Jos mikrosi hyväksyy vain yhden muuttujan sanan INPUT jälkeen, käytä kahta INPUT-riviä.

Suur- ja pienaakkoset

```
50 IF X$>="a" AND X$<="z" THEN PRINT
CHR$(ASC(X$)-32;
60 IF X$>="A" AND X$<="Z" THEN PRINT
CHR$(ASC(X$)+32);
```

Riveiltä 50 ja 60 puuttuva luku on 32. Tämä on suuraakkosten ja pienaakkosten koodinumeroiden välinen ero. Voit tarkistaa asian mikrosi käsikirjan kooditaulukosta.

Salakieliohjelmat (sivu 29)

Salakielen kirjoitusohjelma

Tässä on täydellinen ohjelma, johon on lisätty puuttuvat luvut ja symbolit. Jos käytät tätä ohjelmaa Spectrumissa, muuta käsky ASC muotoon CODE. Käytä ZX81:ssä oikealla annettua ohjelmaa.

```
10 PRINT "MIKA ON VIESTI"
20 INPUT M$
30 FOR J=1 TO LEN(M$)
△ 40 LET X=ASC(MID$(M$,J,1))
50 IF X<65 OR X>90
THEN LET N=X:GOTO 100
60 IF INT(J/2)=J/2 THEN LET N=X+1
70 IF INT(J/2)<>J/2 THEN LET N=X-1
80 IF N<65 THEN LET N=N+26
90 IF N=>90 THEN LET N=N-26
100 PRINT CHR$(N);
110 NEXT J
```

▲ ZX81:n salakielen kirjoitusohjelma

```
10 PRINT "MIKA VIESTI"
20 INPUT M$
30 FOR J=1 TO LEN(M$)
40 LET X=CODE(M$(J TO J))
50 IF X<38 OR X>63 THEN LET N=X
55 IF X<38 OR X>63 THEN GOTO 100
60 IF INT(J/2)=J/2 THEN LET N=X+1
70 IF INT(J/2)<>J/2 THEN LET N=X-1
80 IF N<38 THEN LET N=N+26
90 IF N>63 THEN LET N=N-26
100 PRINT CHR$(N);
110 NEXT J
```

Rivit 60–70 tarkistavat, onko silmukkamuuttuja pariton vai parillinen jakamalla J:n 2:lla ja tekemällä siitä kokonaisluvun sanalla INT. Seuraavaksi mikro tarkistaa, onko vastaus sama kuin J/2. Jos se on sama, J on parillinen luku.

Avainlukukoodi

Käytä tässä ohjelmassa salakielen kirjoitusohjelmaa ja lisää siihen rivejä, joilla salanumero syötetään tietokoneeseen. Rivi 60 pitää muuttaa siten, että se lisää avainluvun (K) kunkin merkin ASCII-koodinumeroon (X). Poista rivi 70.

```
25 PRINT "MIKA ON AVAINLUKU"
27 INPUT K
60 LET N=X+K
70 Poista tämä rivi.
```

Silmukkakoodi

Kuten salakielen kirjoitusohjelma, mutta rivillä 60 silmukkamuuttuja (J) pitää lisätä kunkin merkin ASCII-koodiin.

```
60 LET N=X+J
70 Poista tämä rivi.
```

Silmukkakoodin purkaja

Muuta purkausohjelmaan rivi 60:

```
60 LET N=X-J
```

Kääänteiskoodi

```
10 INPUT "VIESTI ";M$
20 FOR J=1 TO LEN(M$) STEP 2
△ 30 PRINT MID$(M$,J+1,1);
△ 40 PRINT MID$(M$,J,1);
50 NEXT J
```

STEP 2 saa mikron laskemaan kahden askelissa. Aina, kun silmukka toistuu, mikro kirjoittaa ruudulle merkkiparin toisen merkin (J+1) ja sen perään ensimmäisen merkin.

Muutokset Sinclairia varten

```
30 PRINT M$(J+1 TO J+1);
40 PRINT M$(J TO J);
```

INKEY$-harjoitukset (sivu 30)

HEI-sanan kirjoittaminen

```
10 LET A$=INKEY$
20 IF A$="" THEN PRINT " ";
30 IF A$<>"" THEN PRINT "HEI";
40 GOTO 10
```

Jos mitään näppäintä ei paineta, A$ on tyhjä ja mikro printtaa välilyönnin (rivi 20). Jos jotain näppäintä painetaan, se tallennetaan A$:iin ja mikro printtaa HEI (rivi 30).

Käytä mikrosi tuntemaa INKEY-käskyä.

Panemme tietokoneen odottamaan

```
10 LET A$=INKEY$
20 IF A$="" THEN GOTO 10
30 JA NIIN EDELLEEN...
```

Rivi 20 panee mikron toistamaan rivin 10 niin kauan, kunnes jotain näppäintä painetaan.

Nopeaa matikkaa

Korjaa ohjelmaan seuraavat rivit, jotta se toimii kunnolla.

```
20 PRINT "PAINA JOTAIN NAPPAINTA"
50 LET X=INT(RND(1)*25+1)
80 PRINT X;" + ";Y;" = "
110 LET A$=INKEY$
140 IF N<50 THEN GOTO 90
160 FOR K=1 TO 1000:NEXT K
170 GOTO 40
190 PRINT "OIKEIN. VASTAUS ON ";X+Y
```

- Lisää puuttuvat lainausmerkit.
- Kerro 25:llä ja lisää 1, jotta saat satunnaisluvun väliltä 1–25.
- Poista lainausmerkki X:n edeltä.
- Muuttujan nimen pitää olla ennen sanaa INKEY$.
- N pitää olla 1–50, jotta saadaan kaikki mahdolliset vastaukset yhteenlaskuun X + Y.
- Väärä muuttuja NEXT-sanan jälkeen.
- Muuta rivinumero siten, että N nollautuu.
- Pane muuttujien nimet lainausmerkkien ulkopuolelle

Autopeli (sivu 31)

Tässä on autopelin ohjelma:

```
10 CLS
20 LET C=5
30 LET L=1
40 LET W=10
50 LET R=L+W
60 IF L<=1 THEN LET N=1
70 IF L>=25 THEN LET N=0
80 IF N=1 THEN LET L=L+1
90 IF N=0 THEN LET L=L-1
100 LET A$=INKEY$
110 IF A$=">" THEN LET C=C+1
120 IF A$="<" THEN LET C=C-1
130 PRINT TAB(L);"!";TAB(C);"*";TAB(R);"!"
140 IF C<=L OR C>=R THEN PRINT "**AJOIT OJAAN**"
150 GOTO 50
```

- Kun L tulee ruudun jompaankumpaan reunaan, N:n arvo muuttuu.
- Kun N on 1, L:n arvo kasvaa 1:llä. Kun N on 0, L:n arvo kasvaa 1:llä.
- Nämä rivit antavat sinun ohjata autoa.
- Printtaa tien ja auton.
- Selvittää, onko auto mennyt ojaan.
- Lähettää mikron takaisin riville 50 muuttamaan R:n arvon aina, kun L muuttuu.

Lisää tämä viivesilmukka, jos ohjelma on liian nopea:

```
142 FOR K=1 TO 400:NEXT I
```

Pistelaskujärjestelmä

```
140 IF C<=L OR C>=R THEN GOTO 160
145 IF A$<>"" THEN LET S=S+1
160 PRINT "**AJOIT OJAAN"
170 LET CR=CR+1
180 IF CR<=5 THEN GOTO 20
190 PRINT "LOPPU"
200 PRINT "SAIT ";S;" PISTETTA"
210 STOP
```

Rivi 145 antaa yhden pisteen joka kerta, kun ohjaat autoa ajamatta ojaan. Rivi 180 lopettaa ohjelman viiden ojaanajon jälkeen. Ohjelman alussa pitää antaa CR:lle (ojaanajot) ja S:lle (pisteet) alkuarvoksi 0.

Muutokset ZX81 tietokoneelle

Lisää tämä rivi: `125 SCROLL`

Tällä mikrolla ohjelma sujuu paremmin, kun käytetään pistelaskujärjestelmää. Kun ajat ojaan, painat CONT, jotta mikro jatkaa ohjelmaa.

Satunnaisesti kiemurteleva tie

```
75 IF L=5 OR L=10 THEN
LET N=INT(RND(1)*2)
76 IF L=15 OR L=20 THEN
LET N=INT(RND(1)*2)
```

Yksi keino tämän toteuttamiseksi on panna mikro valitsemaan satunnaisesti, mihin suuntaan tie kääntyy eräissä TAB-paikoissa. Muuta satunnaisesti N:n arvoa aina, kun L:n arvo on esim. 5, 10, 15 ja 20.

DATA-arvoitukset (sivut 32–33)

Nimiluettelo

```
50 IF X$="JUSSI" THEN PRINT "NIMESI
EI OLE LUETTELOSSA":STOP
80 DATA JASKA,INKERI,MARKUS-SETA
90 DATA TAUNO,ANSA,JUSSI
```

Pane oma nimiluettelosi DATA-riveille yllä olevan mallin mukaan. Nimien välillä pitää olla pilkku. Pane luettelon viimeinen nimi riville 50. Jos mikrosi ei hyväksy kahta käskyä samalla rivillä, toista IF/THEN-käsky uudella rivillä STOP-sanan kanssa.

DATAn palautus

```
10 FOR J=65 TO 90
20 FOR I=1 TO 19
△40 IF LEFT$(N$,1)=CHR$(J) THEN PRINT N$
```

ASCII-koodit

Data-alkioiden määrä

Muutokset Sinclairia varten

```
40 IF N$(1 TO 1)=CHR$(J) THEN PRINT N$
```

Etsi virheet

1. Data-alkioissa on symboleita – ja / joten rivillä 20 muuttuja pitää vaihtaa merkkijonomuuttujaksi, esim. muotoon N$.

2. GOTO-käsky panee mikron jatkamaan datan lukemista, vaikka se on tullut dataluettelojen loppuun. Tuloksena on syntaksivirhe, esim. "OUT OF DATA AT LINE 50". Ongelman voit ratkaista panemalla silmukan toistumaan niin monta kertaa, kuin data-alkioita on, tai panemalla viimeisen data-alkion IF/THEN-riville samalla tavalla kuten nimiluettelo-ohjelmassa.

Jussin Baari

```
10 PRINT "TERVETULOA JUSSIN BAARIIN"
20 PRINT "PALJONKO RAHAA VOIT KAYTTAA"
30 INPUT X
40 PRINT "RAHASI RIITTAVAT NAIHIN"
50 PRINT
60 READ Y,Y$
70 IF Y<=X THEN PRINT Y$
80 IF Y$="KERROSHAMPURILAINEN" THEN STOP
90 GOTO 60
100 DATA 32.50,RANSKALAINEN ETANAMUHENNOS
```

Kirjoita ruokalistan loput ruuat ja niiden hinnat DATA-riveille. Pane KERROSHAMPURILAINEN viimeiseksi.

Puhelinluettelo

```
10 PRINT "ANNA PUHELIMEN OMISTAJAN NIMI"
30 PRINT
40 INPUT N$
50 READ X$,Y$
60 IF N$=X$ THEN GOTO 90
70 IF X$="" THEN PRINT "NIMEA EI
OLE LUETTELOSSA":GOTO 100
80 GOTO 50
90 PRINT X$;":";Y$
100 PRINT ""HALUATKO LISAA NUMEROITA"
110 INPUT A$
120 IF A$="KYLLA" THEN GOTO 150
130 IF A$="EI" THEN STOP
140 PRINT "EN YMMARRA":GOTO 110
150 RESTORE:GOTO 10
160 DATA MAAORAVA,670 505
170 DATA RUOSTEINEN ROBOTTI,604 444
```

Pane nimet X$:iin ja numerot Y$:iin.

Mene riville 90 kun olet löytänyt nimen.

Pane luettelosi viimeinen nimi tälle riville lainausmerkkeihin.

Tarvitset tämän rivin siltä varalta, että kirjoitat rivillä 110 jotain muuta kuin KYLLÄ tai EI.

Lähettää mikron takaisin DATA-luettelon alkuun.

Kirjoita nimet ja puhelinnumerot tällä tavalla DATA-riveille.

Taulukoiden käyttö (sivut 34–35)

Numeeriset taulukot

```
10 DIM N(6)
50 DATA 1066,1216,1485,1603,1665,1959
```

Datan printtaus

Tarvitset tällaisen silmukan datan printtausta varten (muista numeroida datarivi uudestaan).

```
50 FOR K=1 TO 6
60 PRINT "N(";K;")  ON  ";N(K)
70 NEXT K
```

Tämä silmukka printtaa taulukon alkiot satunnaisesti:

```
50 FOR K=1 TO 10
60 LET R=INT(RND(1)*6+1)
70 PRINT N(R),
80 NEXT K
```

Merkkijonotaulukot

```
△10 DIM N$(5)
 20 FOR I=1 TO 5
▲30 READ N$(I)
 40 NEXT I
 50 FOR I=1 TO 5
 60 PRINT "N$(";I;")"  ON  ";N$(I)
 70 NEXT I
 80 DATA TUOMO,KAARINA,SEIJA,KARI,TIINA
```

Sinclairin muutokset

```
 10 DIM N$(5,7)
▲30 INPUT N$(I)
```

Kalenterilaskuri

Lisää ohjelmaan seuraavat puuttuvat rivit ja muuttujat.

```
△10 DIM M$(12),D(12)
▲30 READ M$(K),D(K)
 70 PRINT M$(N);" SSA ON ";
 80 PRINT D(N);" PAIVAA"
```

Lisää loput datarivit tämän mallin mukaan:

```
▲90 DATA TAMMIKUU,31
```

Muuta ohjelmaa lisäämällä nämä rivit:

```
 50 PRINT "MONESKO KUUKAUSI"
 60 INPUT A$
 62 FOR I=1 TO 12
△63 IF M$(I)=A$ THEN LET F=I
 64 NEXT I
 70 PRINT A$;"SSA ON  ";
 80 PRINT D(F);" PAIVAA"
```

Sinclairin muutokset

```
 10 DIM M$(12,9)
 15 DIM D(12)
▲30 INPUT M$(K)
▲35 INPUT D(K)

 63 IF M$(I)( TO LEN(A$))=A$ THEN
 LET F=1
```

Ajantappo-ohjelma

```
△ 10 DIM I$(10)
  20 FOR K=1 TO 10
▲ 30 READ I$(K)
  40 NEXT K

  50 PRINT "VALITSE LUKU 1 - 10"
  60 INPUT N
  70 PRINT "JOSPA ";
  80 PRINT I$(N)
  90 PRINT "SOPIIKO ?";
 100 INPUT A$
 110 IF A$="KYLLA" THEN STOP
 120 PRINT:GOTO 50
 130 DATA ALKAISIT PIIRTAMAAN,LUKISIT KIRJAN
```

Lisää riville 130 omat datasi.

Sinclairin muutokset

```
  10 DIM I$(10,? )
▲ 30 INPUT I$(K)
```

Laske pisimmän jonon merkkimäärä ja pane luku DIM-lauseeseen.

Satunnaislukutaulukko

```
100 FOR K=1 TO 10
110 PRINT K;TAB(4);
120 FOR L=1 TO A(K)
130 PRINT "*";
140 NEXT L
150 PRINT
160 NEXT K
```

Aliohjelmien kirjoittaminen (sivu 36)

Jäätelökysely

```
△ 10 DIM I$(5),N(5)
  20 FOR K=1 TO 5
▲ 30 READ I$(K),N(K)
  40 GOSUB 70
  50 NEXT K
  60 STOP
  70 PRINT I$(K);TAB(11);" ";
  80 FOR L=1 TO N(K)
  90 PRINT "*";
 100 NEXT L
 110 PRINT
 120 RETURN
 130 DATA MELONI,16,BANAANI,11
 140 DATA INKIVAARI,8,SUOLAKURKKU,1
 150 DATA PURUKUMI,18
```

Tarvitset rivillä 70 TAB-käskyn, jotta mikro printtaa *-symbolit joka rivillä samaan kohtaan.

Sinclairin muutokset

```
 10 DIM I$(5,10)
 15 DIM N(5)
▲ 30 INPUT I$(K)
▲ 35 INPUT N(K)
```

Sukellusveneen upotus

```
200 PRINT "AMMUIT OHI"
210 PRINT "MENE";
220 IF B=Y THEN GOTO 250
230 IF B<Y THEN PRINT "POHJOISEEN ";
240 IF B>Y THEN PRINT "ETELAAN";
250 IF A=X THEN GOTO 280
260 IF A<X THEN PRINT "ITAAN"
270 IF A>X THEN PRINT "LANTEEN"
280 RETURN
```

Tämä on aliohjelma, joka vertaa sukellusveneen olinpaikkaa arvaukseen ja printtaa viestin ruudulle.

20 kysymystä

```
△ 10 DIM Q$(20),A(20)
  20 FOR K=1 TO 20
▲ 30 READ Q$(K),A(K)
  40 NEXT K
  50 FOR L=1 TO 20
  60 PRINT Q$(L)
  70 INPUT X
  80 IF X=A(L) THEN PRINT "OIKEIN"
  90 IF X<>A(L) THEN PRINT "EI OLE, TONTTU.
     VASTAUS ON ";A(L)
 100 PRINT
 110 NEXT L

 120 DATA MONTAKO JALKAA ON
     TUHATJALKAISELLA ?,1000
 130 DATA MONTAKO PELAAJAA ON
     JALKAPALLOJOUKKUEESSA ?,11
```

Jos vastauksissasi on symboleita tai kirjaimia, pane ne merkkijonomuuttujiin.

Pane kysymykset ja vastaukset data-riveille näin.

Sinclairin muutokset

```
  10 DIM Q$(20,?)
  15 DIM A(20)
▲ 30 INPUT Q$(K)
▲ 35 INPUT A(K)
```

Laske pisimmän jonon merkkimäärä ja pane luku DIM-lauseeseen.

Yksikätinen rosvo (sivu 37)

```
△ 10 DIM F$(6)
  20 FOR K=1 TO 6
▲ 30 READ F$(K)
  40 NEXT K
  50 CLS
  60 LET T=10
  70 PRINT "SINULLA ON ";T;" PELIMERKKIA"
  80 LET I$=INKEY$
  90 IF I$="" THEN GOTO 80
 100 LET T=T-1
 110 CLS
 120 LET R=INT(RND(1)*6+1)
 130 LET A$=F$(R)
 140 LET R=INT(RND(1)*6+1)
 150 LET B$=F$(R)
 160 LET R=INT(RND(1)*6+1)
 170 LET C$=F$(R)
 180 PRINT
 190 PRINT A$;"  ";B$;" ";C$
 200 PRINT
 210 IF A$=B$ AND B$=C$ AND
     C$="KIRSIKKA" THEN GOSUB 270
 220 IF A$=B$ AND B$=C$ AND
     C$<>"KIRSIKKA THEN GOSUB 310
 230 IF (A$=B$ AND B$<>C$) OR
     (A$=C$ AND C$<>B$) OR (B$=C$
     AND C$<>A$) THEN GOSUB 350
 240 IF T>0 THEN GOTO 70
 250 PRINT "PELIMERKIT LOPPU"
 260 STOP
 270 PRINT "3 KIRSIKKAA"
 280 PRINT "VOITIT KOKO POTIN"
 290 LET T=T+20
 300 RETURN
 310 PRINT "3 SAMAA"
 320 PRINT "VOITIT 5 PELIMERKKIA"
 330 LET T=T+5
 340 RETURN
 350 PRINT "2 SAMAA"
 360 PRINT "VOITIT 2 PELIMERKKIA"
 370 LET T=T+2
 380 RETURN
 390 DATA SITRUUNA,KIRSIKKA,MELONI
 400 DATA KELLO,RYPALE,LUUMU
```

Valitsee 3 hedelmää F$:sta.

Printtaa hedelmät.

Rivit 210–230 selvittävät, mitä olet voittanut.

1. aliohjelma

2. aliohjelma

3. aliohjelma

Sinclairin muutokset

```
  10 DIM F$(6,6)
▲ 20 INPUT F$(K)
```

Arvoitus sivulla 52

```
10 INPUT "ANNA LAMPOTILA ";T
20 INPUT "MONTAKO SEKUNTIA? ";S
30 IF T<>15 THEN LET D=S*(500+10*(T-15))
ELSE LET D=500*S
40 IF D<1 THEN PRINT "LIIAN KYLMÄÄ ZAKILLE"
50 PRINT T;" ASTEEN LAMPOTILASSA ZAK
JUOKSEE ";S;" SEKUNNISSA ";D;" METRIA"
```

Pane Sinclairissa rivin 30 tilalle tämä:

```
30 IF TEMP<>60 THEN LET
D=S*(500+10*(TEMP-60))
35 IF TEMP=60 THEN LET D=500*S
```

Mikro laskee ensin laskun 10*(T-15) eli lämpötilojen välisen matkaeron. (Jos T on alle 15 vastaus on negatiivinen.) Kun vastaus lisätään 500:aan saadaan matka, jonka Zak juoksee sekunnissa ja kertomalla se S:llä saadaan matka S sekunnissa.

Shell-lajittelun paikanvaihdot (sivu 62)

```
90 LET X=0
95 LET VAIHTO=0
231 LET X=X+1
271 LET VAIHTO=VAIHTO+1
365 PRINT "VERTAILUJA OLI ";X;" JA
VAIHTOJA ";VAIHTO
```

Paksut pylväät (sivu 65)

Joudut ehkä muuttamaan lukuja mikrollesi sopiviksi.

```
285 FOR J=1 TO 8 STEP 2
290 PLOT INT(I*X+J),1
300 DRAW INT(I*X+J),INT(B(I)*Y)
310 PLOT INT(I*X-4-J),1
320 DRAW INT(I*X-4-J),INT(S(I)*Y)
325 NEXT J
```

Konekielen arvoitusten vastaukset – jatkoa

Sivu 108

		Desimaali		Heksa	
		Vasen	Oikea	Vasen	Oikea
00011010 on desimaalina 26	**307**	1	51	&01	&33
11111011 on desimaalina 251	**21214**	82	222	&82	&DE
10101010 on desimaalina 170	**759**	2	247	&02	&F7
	1023	3	255	&03	&FF

Sivu 111

Sivun 107 ohjelman saat muutettua 255:ttä suuremmille luvuille poistamalla palautuskäskyn ja lisäämällä alla annetut rivit. Tuloksen näet tällä komennolla:
PRINT PEEK(osoite 3)+PEEK(osoite 4)*256.

Z80		6502		Merkitys
Muistikas	Heksakoodi	Muistikas	Heksakoodi	
LD A, &00	3E,0	LDA # &00	A9 00	Pane akkuun 0.
ADC A, &00	CE,00	ADC # &00	69 00	Lisää 0 ja muistilippu akkuun.
LD(osoite 4),A	32, osoite 4	STA osoite 4	8D osoite 4	Tallenna akun sisältö osoitteeseen 4.
RET	C9	RTS	60	Palautus.

Sivu 117

12:n kahden komplementin heksamuoto on &F4; 18 on &EE ja 9 on &F7.

Heksalataajan muutokset

Muuta nämä rivit ZX81:tä (Timex 1000) varten:

```
40 INPUT H$
70 LET X=(CODE(H$)-28)*16
80 POISTA
90 LET Y=CODE(H$(2 TO ))-28
100 LET X=X+Y
110 POISTA
155 POISTA
160 POISTA
```

Muuta tämä rivi Atari-tietokoneita varten:

```
90 LET Y=ASC(A$(2))
```

BASIC-sanoja

Alla on luettelo tässä kirjassa käytetyistä BASIC-sanoista sekä lyhyt selitys kunkin merkityksestä. Tähdellä merkityt sanat löytyvät vain joistain tietokoneista. Etsi nämä sanat sivun 140 muuntotaulukosta tai mikrosi käsikirjasta; sieltä löytyvät mikrosi käyttämät vastaavat sanat.

ABC käskee mikron unohtaa lukujen edessä olevan miinus- tai plusmerkin ja antaa lukujen absoluuttisen arvon. Esim. ABS(–5) on 5.

***ASC** muuttaa merkin ASCII-koodinumeroksi. Esim. ASC("A") antaa 65, joka on kirjaimen A ASCII-koodinumero. Sinclair (Timex) mikrot käyttävät ASC)-sanan asemesta sanaa CODE.

CHRS muuttaa tietokoneen sisällä merkkejä edustavan koodinumeron merkiksi. Mikroissa, jotka käyttävät ASCII-koodia CHR$(65) antaa kirjaimen A. ZX81:ssä, joka käyttää erilaisia koodinumeroita, CHR$(38) antaa kirjaimen A.

CLS tyhjentää näytön.

***CODE**-sanaa käytetään Sinclair-tietokoneissa sanan ASC asemesta muutettaessa kirjain sitä vastaavaksi koodinumeroksi. Spectrum käyttää ASCII-koodinumeroita, mutta ZX81:ssä on omat koodinumeronsa.

***DATA** ks. READ/DATA

***DIM** kertoo mikrolle, paljonko muistitilaa sen tulee varata taulukolle. Sanan DIM perään pannaan taulukon nimi ja sen sisältämien alkioiden määrä. Esim. DIM A(5) tarkoittaa, että taulukossa A on viisi tietoalkiota. Kun Sinclair-tietokoneissa käytetään merkkijonotaulukoita, on DIM-lauseeseen pantava myös pisimmän jonon merkkien määrä.

FOR/TO...NEXT saa mikron toistamaan käskyjä rivien FOR/TO ja NEXT välillä tietyn kertamäärän. Tällaista toistoa sanotaan silmukaksi.

GOSUB käskee mikron lähteä pääohjelmasta ja mennä ohjelman osaan, jota sanotaan aliohjelmaksi. GOSUBin perään pitää panna aliohjelman ensimmäisen rivin numero. Aliohjelman lopussa sana RETURN käskee mikron palata pääohjelmaan alkaen GOSUB-käskyä seuraavasta käskystä.

GOTO saa mikron hyppäämään riville, jonka numero on annettu GOTO-sanan perässä.

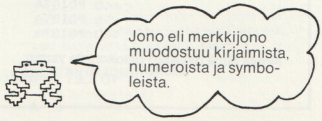
IF/THEN käskee mikron tehdä päätöksen ja tehdä sitten jotain päätöksestä riippuen. Sanan IF jälkeen seuraa ehto, jota mikron pitää testata vertaamalla eri tietoja toisiinsa. Jos ehto on tosi, mikro toteuttaa sanaa THEN seuraavan käskyn. Jos ehto ei ole tosi, mikro hyppää tämän käskyn yli.

***INKEY$** tarkistaa, onko jokin näppäin painettu alas. INKEY$ ei odota, kunnes painat näppäintä (INPUT-sana sen sijaan odottaa), eikä sinun tarvitse painaa RETURN (tai ENTER tai NEWLINE mikrosta riippuen). Koska mikrot ovat erittäin nopeita, INKEY$ pannaan usein silmukan sisään. Joissain mikroissa INKEY$:n perään pitää panna sulkuihin jokin luku, joka ilmoittaa, kauanko mikron pitää odottaa.

INPUT käskee mikron odottaa ohjelman ajon aikana, kunnes näppäilet muuttujaan tietoa. INPUT-sanan perään pitää panna muuttujan nimi.

INT tarkoittaa kokonaislukua (integer). Se muuttaa reaaliluvun (luku, jossa on desimaalipisteen jälkeen numeroita) kokonaisluvuksi jättämällä pois kaikki desimaalipisteen jälkeiset numerot. Esim. INT(6.732) antaa 6. Negatiivisilla luvuilla käy samoin, mutta pyöristys tapahtuu seuraavaan kokonaislukuun. Esim. INT(–3.2) antaa –4.

***LEFT$** käskee mikron ottaa tietyn määrän merkkejä jonon vasemmasta päästä. Esim. LEFT$(A$,4) ottaa A$:sta neljä ensimmäistä merkkiä. Sinclair mikroissa ei ole tätä käskyä.



LEN antaa jonon pituuden eli kertoo, montako merkkiä siinä on (välilyönnit ja välimerkit mukaan lukien).

LET antaa muistipaikalle tunnuksen ja panee sinne tietoa.

LIST käskee mikron näyttää ohjelmalistauksen ruudulla.

***MID$** käskee mikron ottaa tietyn määrän merkkejä jonosta. Esim. MID$(A$,5,2) poimii muuttujasta A kaksi merkkiä alkaen viidennestä kirjaimesta. Sinclair-mikroissa ei ole tätä käskyä.

NEXT käskee mikron palata silmukan alkuun. Ks. FOR/TO...NEXT.

NEW käskee mikron pyyhkiä ohjelman muististaan.

PRINT käskee mikron esittää jotain ruudulla. Pelkkä PRINT aiheuttaa tyhjän rivin.

***READ/DATA** käskee mikron etsiä alkioita, jotka on lueteltu sanalla DATA alkavilla riveillä ja panna ne muuttujaan tai taulukkoon, joka seuraa käskyä READ.

REM tarkoittaa huomautusta (remark). REM-lauseissa voi kertoa muistin tueksi, mitä ohjelman eri osat tekevät.

RESTORE käskee mikron palata DATA-rivien alkuun.

RETURN-sanaa käytetään aliohjelman lopussa. Se käskee mikron palata pääohjelmaan alkaen käskystä, joka seuraa GOSUB-käskyä.

***RIGHT$** käskee mikron ottaa tietyn määrän merkkejä jonon oikeasta päästä. Esim. RIGHT$(A$,4) käskee mikron poimia A$:n neljä viimeistä merkkiä. Sinclair-mikroissa ei ole tätä käskyä.

***RND** käskee mikron valita satunnaisluvun.

RUN käskee mikron toteuttaa ohjelman käskyt.

***SPC** käskee mikron printata ruudulle tietyn määrän välilyöntejä. Tätä käskyä ei ole kaikissa mikroissa.

STEP-sanaa käytetään FOR/TO...NEXT -silmukoissa. Se ilmoittaa, montako kertaa silmukka toistetaan.

***TAB** siirtää kohdistimen tietyn matkaa ruudulla oikealle. Yleensä sanaa käytetään yhdessä sanan PRINT kanssa, kun jotain printataan ruudun keskelle. Joissain mikroissa sanalla TAB voi siirtää kohdistinta ruudulla alaspäin. Yleensä sanan TAB jälkeen pitää panna puolipiste.

THEN-sanaa käytetään sanan IF kanssa, kun mikrolle sanotaan, mitä sen pitää tehdä, jos tietty ehto on tosi. Ks. IF/THEN.

BASIC-symbolit

*	Kertomerkki
/	Jakomerkki
SQR	Etsi neliöjuuri
^	Korota potenssiin. Symbolin käytössä on eroja, joten tarkista mikrosi käsikirjasta, mitä symbolia tietokoneesi käyttää.
>	Suurempi kuin
<	Pienempi kuin
>=	Suurempi tai yhtä suuri kuin
<=	Pienempi tai yhtä suuri kuin
<>	Ei yhtä suuri

BASIC-muunnostaulukko

Tässä kirjassa käytetty käsky.	CLS Tyhjentää näytön	INT(RND(1)*N+1) Valitsee kokonaisen satunnaisluvun väliltä 1 ja N.	INKEY$ Tarkistaa, onko jokin näppäin painettu alas.
BBC	CLS	RND(N)	INKEY$(N)
VIC 20/PET	PRINT CHR$(147)	INT(RND(1)*N+1)	GET X$
DRAGON	CLS	RND(N)	INKEY$
ORIC	CLS	INT(RND(1)*N+1)	KEY$
TRS 80	CLS	INT(RND(0)*N+1)	INKEY$
APPLE	HOME	INT(RND(1)*N+1)	X$="" IF PEEK(-16384) >127 THEN GET X$
ZX81 TIMEX (1000)	CLS	INT(RND*N+1)	INKEY$
SPECTRUM TIMEX (2000)	CLS	INT(RND*N+1)	INKEY$

Lisämuunnokset Sinclair-tietokoneita varten

LEFT$ RIGHT$ MID$	Sinclair-mikrot eivät käsittele merkkijonoja samalla tavalla kuin muut mikrot. Muunna käskyt Sinclair mikroille sopiviksi näin: Laske ensimmäisen ja viimeisen haluamasi merkin paikat. Pane nämä paikat sulkuihin ja niiden väliin sana TO. Pane sulkujen eteen jono tai sen muuttujanimi oikealla olevien esimerkkien mukaisesti.	LET X$="CHIPMUNK" PRINT X$(1 TO 4) CHIP PRINT X$(5 TO) MUNK PRINT X$(2 TO 4) HIP
ASC	Käytä ASC:n asemesta sanaa CODE. ZX81:ssä (Timex 1000) on omat koodinumeronsa, mutta Spectrum (Timex 2000) käyttää ASCII-koodia.	PRINT CODE("A")
DIM	Merkkijonotaulukon ulottuvuudet määritellään näin: Pane sanan DIM perään sulkuihin kaksi lukua. Ensimmäinen luku kertoo taulukon tietoalkioiden (jonojen) määrän ja toinen pisimmän jonon merkkien määrän. Oikealla olevassa esimerkissä taulukossa A$ on viisi jonoa, joista pisimmässä on 12 merkkiä.	DIM A$(5,12)
READ/DATA	Spectrum käyttää käskyä READ/DATA muiden mikrojen tavoin, mutta ZX81:ssä ei ole täysin vastaavaa käskyä. Paras vaihtoehto on oikealla olevan esimerkin kaltainen sarja INPUT-lauseita.	10 FOR K=1 TO 5 20 INPUT A$(K) 30 NEXT K

Sivujen 42–71 ohjelmien muutokset

Tällä ja seuraavalla sivulla kerrotaan, miten sivujen 42–71 ohjelmat muutetaan Sinclair mikroille sopiviksi sekä miten grafiikkaohjelma muutetaan tietokoneille, joissa viivat piirretään suhteessa viimeksi piirrettyyn pisteeseen. Sinun pitää lisätä ohjelmiin tässä annetut rivit sekä tehdä muut mikrosi vaatimat muutokset eli sinun on käytettävä mikrosi grafiikkakäskyjä ja RND-käskyä sekä vaihdettava tarvittaessa muuttujien nimet.

Sinclair-tietokoneiden sanasilppupeli

Muuta ETSI$ E$:ksi ja vaihda rivien 170 ja 200 tilalle seuraavat rivit:

```
170 LET L$=W$(R TO R)
200 LET C$=C$(2 TO)+L$
```

Spectrumin tietokanta

```
10 DIM T$(10,14):DIM Y(12):
15 DIM M(10,12)
120
417
425
```

Sama kuin ZX81:ssä

Muista panna DATA-rivin jokainen data-alkio sulkuihin.

ZX81:n tietokanta

```
10 DIM T$(10,14)
12 DIM Y(12)
14 DIM M$(10,12)
```

Toinen alaindeksi on pisimmän jonon pituus.

```
120 GOSUB 100+100*C
```

Laskee rivinumeron käyttäen C:tä.

```
417 LET L=LEN(Z$)
425 IF Z$=T$(I)(1 TO L) THEN GOTO 440
```

Kertoo mikrolle, mitä merkkejä sen tulee etsiä.

```
455 IF VAL(M$(I,J))=1 THEN PRINT
"VOITTI CUPIN VUONNA ";Y(J)
460 IF VAL(M$(I,J))=2 THEN PRINT
"OLI FINALISTI VUONNA ";Y(J)
555 IF VAL(M$(J,I))=1 THEN PRINT
T$(J);" VOITTI CUPIN"
```

Taulukon datat on tallennettu merkkijonotaulukkoon, joten sinun pitää kertoa sanalla VAL tietokoneelle, että sen tulee valita numeerinen arvo.

```
2170 IF C>0 AND C<6 THEN GOTO 2180
2175 PRINT "KIRJOITA LUKU
VALILTA 1-5"
2176 GOTO 2150
```

Vuosilukujen syöttämisen asemesta voit panna mikron laskemaan ne näppäilemällä seuraavat rivit:

```
1000 FOR I=1 TO 3
1010 LET Y(I)=1926+I*4
1020 NEXT I
1030 FOR I=0 TO 8
1040 LET Y(I+4)=1950+4*I
1050 NEXT I
```

Pane rivien 1100–1120 tilalle kymmenen LET-lausetta, esim.

```
1100 LET T$(1)="URUGUAY"
1110 LET T$(2)="ARGENTIINA"
```

Pane rivien 1200–1310 tilalle toiset kymmenen LET-lausetta, esim.

```
1200 LET M$(1)="100100000000"
1210 LET M$(2)="200000000010"
```

Lisää uusi rivi:

```
1310 RETURN
```

BBC:n tietokanta- ja keskustelu-ohjelmat

Molemmat ohjelmat tarkistavat dataa silmukoilla, jonka jälkeen ne hyppäävät datan löydettyään ulos silmukasta. BBC-mikrossa voit tehdä näin vain kymmenen kertaa. Sen jälkeen saat virheilmoitukset "TOO MANY FORS". Kierrä tämä ongelma siten, että muutat silmukat muuttujalaskureiksi IF...THEN -lauseilla. Esim tietokannassa voit käyttää seuraavia rivejä:

```
420 LET I=0
422 LET I=I+1
430 IF I<10 GOTO 422
520 LET I=0
522 LET I=I+1
530 IF I<12 GOTO 522
```

Joudut tekemään samat muutokset keskusteluohjelman riveillä 500–550 ja vastausaliohjelman riveillä 2030–2070.

Pikagrafiikka ja graafiset esitykset

Tietokoneissa (esim. Spectrumissa ja Oricissa), jotka piirtävät viivan pisteeseen X,Y mitattuna viimeksi piirretystä pisteestä eikä näyttöruudun kulmasta, pitää grafiikka- ja graafisten esitysten ohjelmiin vaihtaa seuraavat rivit. (Vaihda myös sanojen DRAW ja PLOT tilalle mikrosi käyttämät käskyt.)

Pikagrafiikka

```
175 DRAW CX(1)-CX(4),CY(1)-CY(4)
180 FOR I=2 TO 4
185 DRAW CX(I)-CX(I-1),CY(I)-CY(I-1)
530 DRAW LX(ROW+2,I)-LX(ROW,I),
LY(ROW+2,I)-LY(ROW,I)
```

Tietokone etsii viivan loppupisteen koordinaatit vähentämällä viimeksi piirretyn pisteen koordinaatit.

Pylväsdiagrammit

```
220 PLOT 1,H:DRAW 0,-H+1:DRAW W,0
300 DRAW 0,  INT(B(I)*Y)
320 DRAW 0,  INT(S(I)*Y)
```

Käyrien piirto

```
290 DRAW X,  INT((B(N)-B(N-1))*Y)
330 DRAW X,  INT((S(N)-S(N-1))*Y)
```

Spectrumin keskusteluohjelma

Tee Spectrumia varten seuraavat muutokset:

```
110 DIM V$(10,12): DIM N$(10,21); DIM A$(10,13)
120 DIM T$(10,14):DIM S$(10,14)
130 DIM M$(30,37): DIM Q$(30,9):DIM C(30)
245 PRINT I$
500 FOR Q=1 TO 30
510 LET P$=""
512 FOR I=1 TO LEN (Q$(Q))
514 IF Q$(Q)(I TO I)<>" " THEN
LET P$=P$+Q$(Q)(I TO I)
516 NEXT I
530 FOR T=1 TO L2-LEN(P$)+1
540 IF I$(T TO T+LEN(P$)-1)=P$ THEN
GOTO 560
550 NEXT T:NEXT Q: GOTO 600
560
570 Vaihda muuttuja SANA muuttujaksi q.
580
660 GOTO ((E<7)*(680+E*20))+((E>6)*
(810+(E-6)*20))
```

Käytä nimikkeen SANA asemesta nimikettä Q.

Panee Q$: sanan P$:iin.

Tarkistaa, onko I$ = P$.

Tarkoittaa, että jos E<7 GOTO riville numero 680 + E★20 ja jos E>6 GOTO riville numero (810 + (E76)★20).

ZX81:n keskusteluohjelma

ZX81:ssä tarvitset oman menetelmän datan syöttämiseksi. Voit tehdä tämän INPUT-lauseilla ja silmukoilla. Ennen kuin ajat ohjelman ZX81:llä, tee seuraavat muutokset:

1. Tee yllä Spectrumille annetut muutokset, mutta pane DIM-lauseet riveille 970–990.

2. Pane READ/DATA-lauseiden tilalle INPUT-lauseita, esim.

```
1000 REM SANOJEN ETSINTARUTIININ SANAT
1010 FOR I=1 TO 30
1020 INPUT Q$(I)
1030 NEXT I
```

3. Muuta rivi 1720 muotoon:

```
1720 STOP
```

4. Näppäile ohjelma tietokoneeseen ja näppäile sitten RUN 970 ja näppäile koneeseen datat, kun mikro pyytää niitä.

5. Kokeile ohjelmaa näppäilemällä GOTO 100. Älä paina RUN, sillä jos painat, kaikki data joutuu hukkaan.

6. Nyt voit tallentaa ohjelman kasetille. Kun lataat ohjelman koneeseen, kirjoita aina GOTO 100, kun haluat ajaa sen.

Hakemisto

Tässä hakemistossa arvoitusten ja ohjelmien nimet on painettu puolilihavalla ja BASIC-käskyt suuraakkosilla.

absoluuttinen osoitus, 98, 107, 126
akku, 94-95, 97, 30, 112, 126
yhteenlasku, 5
  konekielellä, 97, 107, 110
osoite, 88-89, 99, 126
  muutto heksaan tai desimaaliksi, 91, 98-99
osoitustavat, 107, 126
ALU (aritmeettis-looginen yksikkö) ja-merkki (&), 94, 95, 126
AND, 22, 52
Vastausrutiini, 72
Apple II tietokone, 63, 140
taulukot, 9, 34–35, 47, 53, 54
ASC, 28, 138, 140
ASCII koodi, 28, 93, 104, 112,
assembler, 85, 96, 120, 126
symbolinen konekieli, 85, 97, 99, 126
Atari tietokoneet, 3, 83, 104, 137
Pylväsdiagrammi, 64
BBC tietokone, 3, 50, 64, 65, 68, 72, 83, 87, 92, 140, 141
binäärikoodi, 84, 85, 96
  luvut, 84, 99, 108, 126
  muutto heksaan, 117
Binäärilaskuri, 20, 129
bitti, 84, 126
sulut laskuissa, 5
haarautuminen, 22
  konekielessä, 114, 115, 126
BREAK, 7
Kuplalajittelu, 58-59, 60, 63
puskurit, 90, 126
viat, 4, 6, 7, 14, 41, 51
pukit, ks. viat
Vikoja silmukoissa, 20, 129
tavu, 84, 93, 99, 100, 126
byte ks. tavu
Kalenterilaskuri, 35, 135
Suur- ja pienaakkoset, 28, 133
Autopeli, 31, 134
muistilippu, 94, 95, 97, 109, 110, 111, 116, 126
muistiin pano yhteenlaskuissa, 109, 110, 111
Celsiuksesta Fahrenheitiin, 17, 128
merkkikoodit, 93, 112
merkit, 4, 60
CHR$, 28, 138
nollaus, 109, 126
CLS, 4
CODE, 28, 138, 140
kaksoispiste, 11
pilkku, 5, 7, 11, 51, 60, 71
Commodore 64 tietokone, 3, 64, 83, 87
vertailu, 112
Kirjainvertailu, 28, 133
komplementoida, 126
Tietokonelaskin, 22, 130
Tietokonekello, 20, 129
ehdollinen haarautuminen, 115, 116-117, 126
CONTINUE, 41
ohjausyksikkö, 94, 95
Keskusteluohjelma, 66-73
koordinaatit, 8, 54, 64
keskusyksikkö, 87, 94-95, 96, 99
romahtaa, 100
kohdistin, 4
DATA, 7, 11, 32, 138
Tietokantaohjelma, 46-53
datatavut, 103, 108
Itsepuhelu, 71
desimaaliluvut, 91, 121
vianetsintä, 6, 41
vähennys, 116, 118
viivesilmukka, 19, 43, 44
DELETE, 14
rajoitin, 60
DIM, 9, 11, 34, 53, 138, 140
taulukon ulottuvuudet, 9, 67
suora osoitus, 107, 126
suorat käskyt, 4, 14-15
siirtymä, 116-117, 126
jakaminen, 5
Dragon tietokone, 87, 140
Puhutko jugliaania? 17, 128
alkiot (taulukossa), 34
ELSE, 51, 52
END, 51
ENTER, 4
ESCAPE, 7

Pako Zorgosista, 25, 131
laajennettu osoitus, 107
lippurekisteri, 94-95, 97, 109, 116
lippumuuttuja, 77
Välkkyvä viesti, 20, 129
vuokaavio, 75
FOR/NEXT silmukka, 8, 18-19
Yksikätinen rosvo, 37, 136
GET, 30, 40
GOSUB, 7, 36, 48, 138
GOTO, 7, 22
grafiikkakäskyt, 8, 11, 54
graafiset esitykset, 64-65
Arvausleikit, 23, 130
risumerkki, #, 92, 96, 98, 126
heksa
  koodit, 96, 98, 99
  muutto desimaaliksi, 91, 121
  vedos, 99
  lataaja, 85, 104, 126
  lukujärjestelmä, 85, 88, 91, 126
vasen tavu, 99, 108, 110, 111, 126
Nopeaa matikkaa, 30, 134
HIMEM, 88, 100, 101, 126
Ravikilpailut, 23, 130
Jäätelökysely, 36, 136
IF/THEN, 7, 10, 11, 22-23, 52
välitön osoitus, 98, 107, 113, 126
implisiittinen osoitus, 126
lisäys, 113, 114, 118
indeksoitu osoitus, 114, 119, 126
indeksirekisterit, 94-95, 126
epäsuora osoitus (Z80), 113, 118, 126
muuttujien alustus, 10
INKEY$, 30, 138, 140
INPUT, 6, 10, 16
käskykanta, 96, 127
INT, 8, 14, 138
tulkki, 84, 88, 100, 127
Jussin Baari, 33, 135
Hyppivä mies, 21, 130
hypyt (konekoodissa), 113, 115, 127
Avainlukukoodi, 29, 133
LEFT$, 9, 26, 138, 140
LEN, 9, 14, 26, 139
LET, 6, 10, 16, 139
Kirjainarvoitukset, 28, 133
LIFO, 90, 127
Viivojen piirto, 64-65
rivinumerot, 4, 6, 40
LIST, 4, 15, 139
paikat, muisti-, 88-89, 90, 92-93
Pisin sana, 27, 132
Kurkistamme ohjelmaan, 17, 128
Silmukka-arvoitukset, 18, 20-21, 128, 129
silmukat, 8, 11, 40, 52, 55
Silmukkakoodi, 29, 133
oikea tavu, 99, 108, 110-111,127
konekieli,
  tarkistuslista, 106
  ohjelman pituus, 100
  aliohjelmat, 119, 120
  tallennus muistiin, 100-102

Ajantappo-ohjelma, 35, 136
odottaminen, tietokoneen, 30
matemaattiset symbolit, 14, 139
matriisi, 46
muisti, 88-89, 90, 92-93
muistikartta, 88
valikko, 46, 48, 52
mikroprosessori, 87, 96, 127
MID$, 9, 26, 42, 139, 140

virheiden korjaaminen, 14-15
muistikkaat, 85, 96-97, 127
moduli, 40
useita käskyjä rivillä, 11, 43, 44
kertolasku, 5
Nimiluettelo, 32, 135
sisäkkäiset silmukat, 9, 20, 44, 45, 51,
NEW, 14, 139
NEWLINE, 4
Numerosarja, 24, 131
numeeriset muuttujat, 6, 10, 50
tuloskoodi, 98, 127
ON, 48, 49, 69
toimituskoodi, 96, 98, 99, 127
operandi, 96, 98, 107, 127
käyttöjärjestelmä, 88, 90, 91, 93, 100, 127
OR, 22, 52
Oric tietokone, 3, 43, 45, 54, 83, 87, 101, 140
sivu (muistin), 90, 91, 99, 101, 127
Paperi, kivi tai sakset, 25, 132
Tunnussana, 22, 130
Kuvioarvoitukset, 19, 129
PEEK, 92-93, 101, 106, 111
PET tietokone, 128, 140
pikseli ks. kuvapiste
kuvapiste, 8, 64
PLOT, 54, 55, 64
osoitin, 113, 127
POKE, 92-93, 103
paikka sivulla (osoitteen), 91, 99, 101
PRINT, 5, 14-15, 139
PRINT-arvoitus, 16, 128
tilarekisteri, 95, 109, 127
osoitelaskuri, 94-95, 115, 127
välimerkit, 14
lainausmerkit, 5, 11
RAM, 86, 92, 93, 100-101
RAMTOP, 88, 100, 101, 127
alentaminen, 100-102
Satunnaistaulukko, 35, 136
satunnaisluvut, 8, 35, 63, 136
READ/DATA, 7, 9, 32, 139, 140
rekisterit, 94-95, 107, 110, 111, 127
suhteellinen osoitus, 116, 127
REM, 7, 40
REM lause konekielessä, 102
varattu käyttöjärjestelmälle, 89, 90
RESTORE, 32, 139
palautuskäsky (konekieli), 103, 115
RETURN, 4, 7, 14-15
Käänteiskoodi, 29, 133
RIGHT$, 9, 26, 42, 139, 140
RND, 8, 24, 139, 140
Robottijuoksija, 52, 137
Raketin lähtö, 21, 129
ROM, 86, 92, 93
rutiini, 40, 41, 66
RUBOUT, 14
RUN, 4, 15, 139
Makkaraohjelma, 17, 128
näyttömuisti, 88, 93, 127
Salakielen kirjoitusohjelma, 29, 133
Salainen viesti, 19, 129
puolipiste, 5, 6
Shell-lajittelu, 58, 61-62, 63, 137
Lyhyin sana, 27, 132
merkkilippu, 94, 116, 127
Sinclair tietokoneet, 3, 9, 10, 11, 26, 28, 34, 87, 140
Sukellusveneen upotus, 36, 136
lajittelu, tietojen, 58-63
alkukoodi, 98, 127
SPC, 14, 139
Spectrum tietokone, 3, 54, 83, 93, 112, 140
Nopeuslaskuri, 17, 128
Etsi virheet: arvoitukset, 32, 135
Porrasarvoitukset, 18, 129
neliöjuuret, 5
pino, 90, 94, 95, 100, 115, 127
pino-osoitin, 94-95, 127
yleis-BASIC, 10, 41
STEP, 8
STOP, 41
jono
käsittely, 42, 66
muuttujat, 10, 16-17, 50, 60
Jonoarvoitukset, 26, 132
aliohjelmat, 7, 40, 48
konekoodissa, 115
alaindeksit, 9, 34, 47
vähennyslasku, 5
järjestelmämuuttujat, 90, 100, 101, 127
TAB, 14, 139
Kertolaskuohjelma, 22, 130
Puhelinluettelo, 33, 135
korkein osoite, muistin, 100, 101, 127
Aarteenmetsästysohjelma, 74-79, 80-81
Kaksikymmentä kysymystä, 35, 136
kaksiulotteinen taulukko, 9, 51
kahden komplementti, 117, 121, 127
käyttäjäystävällinen, 46, 52
käyttäjä-RAM, 88, 100, 127
VAL, 50
muuttujien nimet, 6, 10
muuttujat, 5, 6, 8, 9, 10, 40, 43
VIC-20 tietokone, 3, 6, 10, 83, 87, 93, 102, 128, 140
Sanojen arvaus, 23, 130
Sanasilppupeli, 42, 141
nollalippu, 112, 113, 114, 116, 127
nollasivu, 90, 125, 127
nollasivun osoitus, 125, 127
ZX81 tietokone, 3, 16, 28, 34, 83, 89, 93, 102, 138, 140
vastakääntäjä, 126

Alkuteos: The Beginner's Programming Handbook in Basic & Machine Code
© 1984 Usborne Publishing © Tammi 1986
Printed in Great Britain 1986
ISBN 951-30-6310-0

Aloittelijan OHJELMOINTIKIRJA

Basic ja konekieli

Kirja on värikäs askel askelelta etenevä kotitietokoneen ohjelmoinnin opas. Ensimmäisessä osassa kerrotaan, miten kirjoitetaan BASIC-ohjelmia. Lopussa on hyvin helppo johdatus konekielen alkeisiin. Siinä kerrotaan, mitä konekieli on, miten konekielisiä ohjelmia ajetaan kotitietokoneella ja miten voi itse kirjoittaa konekielisiä ohjelmia. Basic-käskyt esitellään yksitellen. Jokaisen käskyn käyttöä selvennetään esimerkeillä. Kirjassa on runsaasti arvoituksia, harjoitustehtäviä ja neuvoja, joista on apua BASICin oppimisessa. Mukana on myös pitempiä ohjelmia, joista ilmenee BASICin käyttö esimerkiksi tietokannan laatimisessa tai kuvien piirtämisessä. Kirjan ohjelmat on laadittu yleis-BASICilla, joka sopii pienin muutoksin useimpiin kotitietokoneisiin. Konekieliset ohjelmat on laadittu tietokoneille, joissa on Z80 suoritin (Spectrum, ZX81, TS1000, TS2000, TS2068 sekä MSX-mikrot) tai 6502 suoritin (BBC, Commodore 64, VIC-20, Oric ja Atarit).

N 60

ISBN 951-30-6310-0