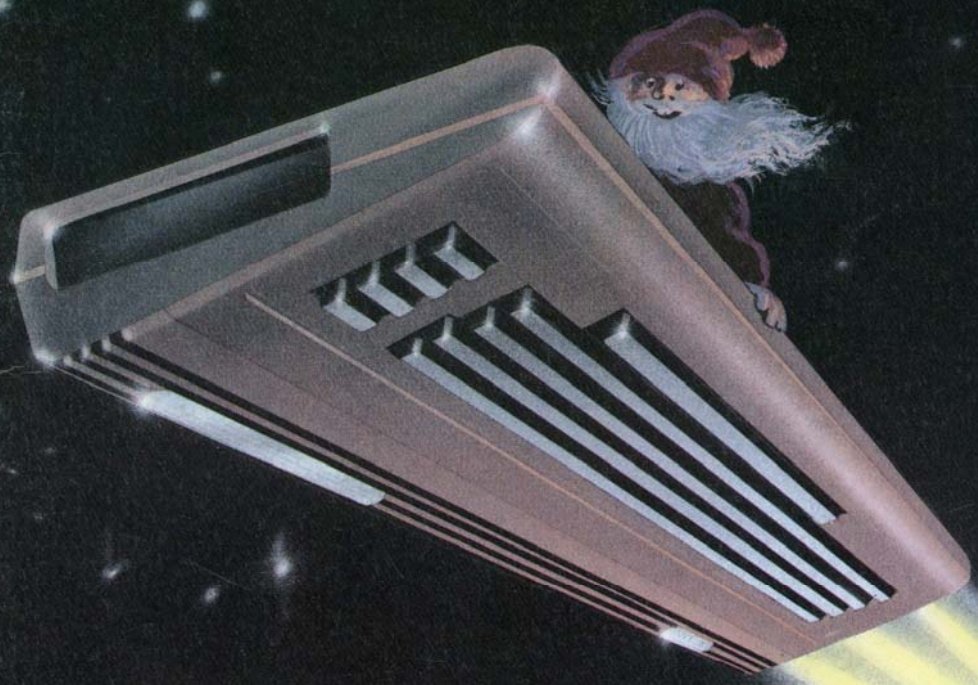
VIC rapport

ÅRGÅNG 2 NR 5/6
PRIS 20:- inkl moms



En VIC-20 som räddar liv
Musikaliskt Julkort på VIC-20
Tips-program
Program på postorder
Ekonomiprogram för VIC-20
Egna tecken på VIC-64
Kalenderprogram
Klubbnyheter m.m.

473—5/6

DIGICOM NYHETER



HEXPERT/64 Spännande 3D spel för hela familjen med fin grafik och mycket action. ca pris 125:-

MINI KONG/20 En ny variant av Krazy Kong för oexpanderad VIC 20. Bästa hitintills? ca pris 99:-

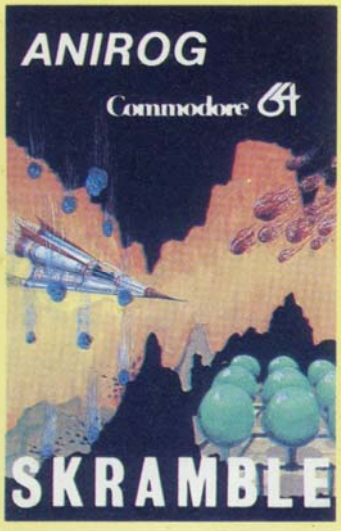
MOON BUGGY/64 Hur långt klarar du köra din moonbuggy över kratrar och under ständig attack. ca pris 125:-

DRACULA/LOST IN THE DARK/20 Två grafikadventures för oexpanderad VIC 20. Vågar du möta DRACULA? ca pris 99:-

KRAZY KONG/64 Äntligen ett KONG till 64-an som utnyttjar alla möjligheter. ca pris 125:-

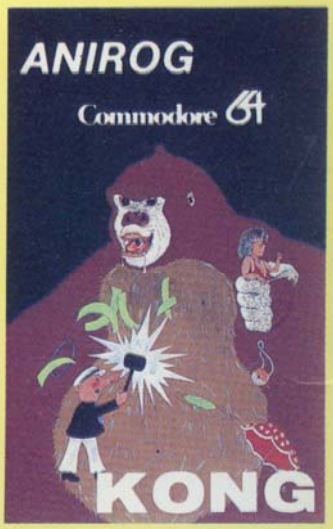
SKRAMBLE/64 Otroliga grafik och ljudeffekter i detta actionspel för 64-an. Vår bestseller!! ca pris 125:-











DU KÖPER SPELEN HOS

VIC SPECIALISTEN

VI HAR ALLT TILL VIC 20 OCH VIC 64

DISKETT

DAVIDSHALLSG 27
MALMÖ TEL 97 20 26

VIC**ATARI**SPECTRAVIDEO**SHARP**VIC-20**VIC-64**DATA**PRINT**SPRITE**BUSS**BINÄR**SOYSTICK**CP/M**CHIP**RETURN**STRÄNG**ERROR**PEEK**MODEM**INPUT**ROM**BINÄR**RAM**DIM**POKE**CPU**BASIC**MIKRO**ROM**BYTE**NEW**

DATORSPECIALISTEN MITT I SKÅNE

MASSOR AV PROGRAM & TILLBEHÖR
FRÅN LEDANDE LEVERANTÖRER

VIC*ATARI
SPECTRAVIDEO

EGEN PROGRAMUTVECKLING INOM MÄKLARBRANSCH MM.
AVANCERAT BUTIKSDATASYSTEM.KASSAREGISTER "ON-LINE"

även postorder !! SNABBA LEVERANSER FRAKTFRITT ÖVER 500

DATORBUTIKEN i ESLÖV
ASB--MIKRO AB
Köpmansgatan 2
241 00 ESLÖV
0413 – 105 70

IMPORTÖR & DISTRIBUTÖR
HTD DIGICOM AB
Box 194 243 00 HÖÖR Tel. 0413/210 30

Våra program, böcker och tillbehör köper du hos välsorterade fackhandlare, eller om ingen sådan finnes på din ort, direkt från oss. Du når oss på telefon 0413/210 30. Nya återförsäljare antages.

VI SKICKAR VÅR KATALOG MOT DUBBELT SVARSPORTO

Datorernas intåg - en transport-revolution!

På pressöppningen vid Personal Computer World Show i Barbican Centre, London, presenterade huvudtalaren några intressanta tankar om den otroligt snabba utvecklingen på mikrodatorområdet.

Han talade om våra tre revolutioner på transportområdet och nämnde att den första var när järnvägen kom. Då började även de som inte var så rika att de hade egna hästar och kärror att kunna resa, t.ex. till ett arbete i en stad medan de bodde kvar på landet. Eller för att resa och hälsa på vänner som bodde långt borta. Denna revolution var första steget i att föra människorna närmare varandra.

Nästa revolution var när bilen uppfanns. Då kunde människorna själva styra dit de ville resa eller transportera gods. Vi kom ännu närmare varandra, jorden blev mindre. Flyget var en naturlig vidareutveckling på transportområdet och är i dag en naturlig del i vår vardag. Vi flyger inrikes på rabattbiljetter under lågprisperioder och kommer ännu snabbare fram än med tåg eller bil. Vi har ännu lättare att flytta oss till avlägsna arbetsplatser och man har fått en ny sorts människor, pendlare, som bor kanske i ett land och arbetar i ett annat.

Den senaste revolutionen kan innebära ett steg tillbaka mot ursprungstillståndet. Mikrodatorrevolutionen gör ju, att människor kan stanna kvar hemma och sköta sina arbeten via en terminal. Det enda som finns centraliserat är en stor databas, till vilken decentraliserade terminaler är anslutna. För datorrevolutionen innebär ju att information och kunskap kan transporteras med ljusets hastighet från ena ändan av världen till den andra, ibland med hjälp av satelliter så att det inte längre är samma behov av att människorna finns på samma arbetsplats för att kunna sköta sitt jobb. *forts. sid. 5*

Innehållsförteckning

Ledare: Datorernas intåg - en transportrevolution 1
En VIC-20 som räddar liv! 2
God Hul från VIC-rapport 5
Tippa stryktips med din expanderade VIC-20 6
Å Ä och Ö på en expanderad VIC-20 11
Spela julsångerna på din VIC-20 12
Kräv bättre programvara och datortidningar 13
Bättre bild och några andra tips till VIC-20 14
Spelrekord 15
Användarvänlighet 16
"Bättre service åt både konsument och återförsäljare" 17
Europas största utställning i år? 18
Nyheter utifrån 20
Är VIC:s tidning bedrövlig? 23
Vad har man egentligen för nytta av en hemdator? 24
Lägg på en rem! 25
Planera din ekonomi 26
Konsten att sortera 30
Sorteringsförsök 30
Sortering 34
Sorteringsprogram för VIC-20 34
Variabellagring i Basic/CBM-64/VIC-20 36
Bidrag från läsarna 39
Gör egna tecken på VIC-64 42
Kapten Storöra 46
"För tid och evighet..." Kalenderprogram 48
Färger 50
Strukturerade Basic-program 52
Krångliga tecken i listorna? 53
Prentice Hall 53
Calc Input 54
Lättfil 55
Nya läroböcker till VIC-datorerna 56
The Working Commodore 64 58
Ännu en bok om VIC-20, VIC-20 exposed 59
Ännu en bok om VIC-64, Commodore 64 exposed 59
Bokrecension 60
Från bokförlaget Osborne/McGraw-Hill 61
Grossistronden 61
VIC-20 inklusive introduktionskurs 61
Hösten - mässornas förlovade tid 62
Detta vet man väl 63
Gömma rader i ett BASIC-program? 64
Programmeringshandbok till VIC 64 65
Nybörjarskolan, del 2 66
Frågor och svar 70
Mister Tid-Bit: Gränslöst roligt? 73
Gratisannonser 74
Tipsrutan 74
För VIC-64 "Blindbingo" 76
Nya klubbar 79
Låt VIC-64 räkna ut speltid på kassettbandade grammofonskivor! 80
Assemblerskolan 3 82
Spelnytt 84
Wizard 85
Speltestare söktes! Stopp, stopp, stopp 86

Redaktionsruta sid. 55

En VIC-20 som räddar liv!



Nu har jag varit med om något fantastiskt! Jag har sett en VIC-20 fungera som livräddare eller i varje fall som ett effektivt hjälpmdel för dem, som sköter om att hitta och rädda folk i sjönöd.

Programmet, som gör att VIC-20 kan användas, heter "DATA-SAR", och det är skrivet av en av sjöräddningsledarna på Göteborg Radio, Ingvar Wihlborg, Kungsbacka. Efter hans demonstration är jag helt övertygad om att den datakraft, som finns i en VIC-20 är värd ett långt bättre öde än att enbart användas för spel och lek!!

Ingvar berättar vidare om hur han till grund för sitt program haft mycket noggranna undersökningar, där man legat ute till sjöss under långa perioder och mätt upp basfaktorer som t.ex. vindstyrkans effekt på drivande farkoster av olika slag och med olika belastningar och utrustningar, strömmarnas påverkan, vindens inverkan på strömmarna och många andra viktiga förhållanden. Även detta material har bearbetats i VIC-20!

Men jag låter Ingvar själv beskriva sitt program:

VIC-20 i sjöräddningens tjänst

I Onsala, ca. 1.5 mil från Kungsbacka, ligger Göteborg Kustradiostation, vardagligt kallad Göteborg Radio. Stationen fungerar bl.a. som västkustens sjöräddningscentral. Denna verksamhet är integrerad i stationens normal-arbetsfält, dvs. radiokommunikation med fartyg på alla världens hav.

Sjöräddningscentralens ansvarsområde sträcker sig från Bohusläns gräns mot Norge i norr till Falsterbo i söder. Stationen är bemannad dygnet runt och totalt arbetar ca. 75 personer i skift. En tredjedel av dessa personer fungerar vid sjöräddningslarm som sjöräddningsledare. I stationsbyggnadens mitt ligger sjöräddningscentralen belägen. Här inne står sedan 15 månader en VIC-20 alltid startad och lagrad med en för sjöräddningen unik programvara, benämnd "DATA-SAR". (SAR står för "Search And Rescue"). Under denna tidsperiod med ca. 1200 behandlade sjöräddningsfall, har VIC:en använts i en tiondel av fallen.

Det kan fastslås, att programvaran "DATA-SAR" endast används av västkustens central. Förmodligen är den så unik, att västkustens räddningscentral är ensam i världen om denna facilitet. Programvaran är framtagen av artikelförfattaren, en av sjöräddningsledarna vid centralen.

Då en detaljrik beskrivning av "DATA-SAR" skulle bli en lång historia och förmodligen verka mycket krånglig för den oinvigde, skall jag försöka så kortfattat som möjligt beskriva hur programvaran används. Programmet, som till stora delar handlar om olika beräkningar, är uppbyggt kring en meny, där de olika programsekvenserna presenteras. Huvudrubrikerna i menyn är följande:

```
**DATA-SAR**
◆PROGRAM◆

#1=SÖKOMRÅDE
#2=URSPRUNG
#3=PLANERING
#4=SEKTORSÖKNING
#5=RENDEZ-VOUS
#6=NÖDRAKETER

PROGRAMVAL #?
```

I korthet skall nu beskrivas, vad de olika underprogrammen avser:

```
DATA-SAR
PROGRAMVAL #: 1
SØKOBJEKT #: 2

DS-TIDP : 06.00
LATITUD : 57.253
LONGITUD: 11.424
DS.RADIE: .5

OBS# 1
TIDPUNKT    : 9
VIND,GRADER: 225
VIND,KNOP   : 20
VS/BS/TS    : VS

OBS# 2
TIDPUNKT    : 12
VIND,GRADER: 270
VIND,KNOP   : 15

OBS# 3
TIDPUNKT    : 14 SØKST.
VIND,GRADER: 315
VIND,KNOP   : 12

OBS# 4
TIDPUNKT    : #

          SØKOMRÅDE KL: 14

MELLAN PARALLELLER:

LAT: 57.312 - 57.22
LON: 12.032 - 11.461

CENTRUM

LAT: = 57.266
LON: = 11.546

RADIE= 4.6 NM

SJÖRADDN.FALL #..... DATUM.../.../... SMC:.....
```

SÖKOMRÅDE:

Som namnet anger, gäller detta beräkning av sökområde, dvs. beräkna ett geografiskt område dit en saknad fritidsbåt, flotte etc. kan ha drivit. Olika faktorer och element styr denna drift. Vad gäller faktorerna, finns sådana medtagna i beräkningen såsom djupgående, vindfaktor och vindriktningsavvikelse. Samtliga dessa faktorer är speciella för varje typ av farkost. Elementen som påverkar driften är bl.a. historiska strömmar och d:o vindar. Tiden har också en avgörande betydelse. Datorn matas med drivstarttidpunkt (kan vara = olyckstidpunkt) och drivstartposition. Vindar och strömmar matas in från denna tidpunkt till sökstart. VIC-20 presenterar nu ett svar som anger mellan vilka LAT. resp. LONG.-paralleller det sökta föremålet kan finnas. Svaret anges också med sökområdets centrumpunkt samt områdets radie. I denna programsekvens kan beräkningen bygga på olika strömalternativ och bl.a. kan nämnas, att programmet alltid beräknar genererad vindström och corioliseffekt. Vad gäller de speciella faktorerna för olika farkoster kan nämnas, att i ett underprogram omhändertags alla avdriftsprov som görs och där farkosters olika "uppförande" beräknas. Då DATA-SAR körs, anges endast typ av farkost i klartext.

```
                AVDRIFTSPROV (831024)

PLASTEKA                              DJPG: 3 DM

STARTTID: 13.53                 SLUTTID: 18.02
STARTLAT: 55.5752               SLUTLAT: 55.5828
STARTLON: 15.1781               SLUTLON: 15.6925

                TOTALDRIFT      VERKLIG VINDAVDRIFT
RIKTNING(GRADER):  288              278
DISTANS(NM):       4.4              3.5
MEDELHAST.(KNOP)   .96              .76

                        OBSAR

OBS-TID  VINDRIKTN. VINDSTYRKA TS-RIKTN. TS-STYRKA    TS-RIKTN. TS-STYRKA
14       94         11         312       .24        312       .24
18.3     88         10         312       .24        312       .24

                TOTALSTRÖM UNDER DRIVPERIODEN=  312     .24
                MEDELVIND  UNDER DRIVPERIODEN=  88      10
                BS - STRÖM UNDER DRIVPERIODEN=  348     .16

AVDRIFTSFAKTORER /VIND 15 KNOP/  (VÄRDEN VID AKTUELL MEDELVIND)
VINDRIKTN.AVVIKELSE (GRADER): LW= 11              ( 10 )
VINDAVDRIFTSFAKTOR (PROCENT): F = 6               ( 7.6 )

                        K U R V O R

LW=-5  10  15  20  25  30  35  40  45  50  55  60  65  70  75°
F=: 1   2   3   4   5   6   7   8   9  10  11  12  13  14  15%
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
KNOP
```

URSPRUNG:

I grova drag kan detta program beskrivas som en omvänd sökområdesberäkning. En tom drivande båt etc. upphittas och man är intresserad av att veta varifrån den kan ha kommit. Knarksäckarna i somras var ett uppdrag för DATA-SAR:s ursprungsberäkning.

PLANERING:

Denna programsekvens finner sin användning när väl ett s.k. sökområde bestämts. Detta område skall avspanas av sjöräddningsenheter, räddningskryssare, kustbevakningsbåtar och helikoptrar. Sjöräddningsledaren önskar snabbt planlägga det hela och dimensionera avspaningen. Faktorer som åtgående tid, antal enheter, sökavstånd och täckfaktor beräknas. Även spaningens effektivitet beräknas och en standardfras "POD" (probability of detection) beräknas. Ingångsvärden i denna programsekvens är områdets yta, enheternas prestanda samt rådande meteorologiska förhållanden som

```
UTÖKNING (V): PLAN
SIDA:A: 9.2
SIDA:B: 7.2
YTA: 66.24 NM

OBJEKT#: 3

SØKENHET: TV275
MAX SØKTID: 2
SØKHASTIGH: 15
SPANHØJD(M): 0
MET.SIKT(NM): 10
VINDSTYRKA: 15
MOLNMÄNGD,%: 100

SØKENHET: TV257
MAX SØKTID: 2
SØKHASTIGH: 15
SPANHØJD(M): 0

SØKENHET: #
ALT.VAL #: 3
MCA

                                    PLANERING
                                      MCA
                #&ENHET                      ERF
                 S    C    POD  YTA  TID
1 TV275
  2.1  1.4  95%  33   1.03
2 TV257
  2.1  1.4  95%  33   1.03

UPPNÅDD YTA= 66 NM
OMR= 9.2 X 7.2 NM
URSPR.YTA= 66.2 NM

SJÖRADDN.FALL #..... DATUM.../.../... SMC:.....
```

sikt, molnmängd etc. Programmet fördelar vidare området i lämpligaste avsnitt för respektive deltagande spaningsenhet.

```
DATA-SAR
PROGRAMVAL #: 4
RADIE(NM): 2.5
FART(KTS): 20
STARTTID : 11.37

                SEKTORSØK

OMR. RADIE = 2.5 NM
TOTAL DIST.= 22.5 NM
STYRB.GIRAR= 120 GRAD.

GÅNGTID:   /STRAECKA/
15 (M.SEK) /1;3;5/
7.3 (M.SEK) /2;4;6/

STARTTID    = 11.37
AVSPAN.KLAR= 12.44

SJÖRADDN.FALL #..... DATUM.../.../... SMC:.....
```

SEKTORSÖK:

Denna programsekvens utför nödvändiga beräkningar i ett s.k. sektorsök. Sektorsök innebär ett sökmönster som kan jämföras med ekrarna i ett hjul. Olika manövrar beräknas samt olika gångtider.

RENDEZ-VOUS:

Denna sekvens låter sig bäst beskrivas av att relatera till ett sjöräddningsfall. Ute på nödenhet X finns en allvarligt sjuk besättningsmedlem som snarast måste föras till sjukhus med en snabbgående räddningsenhet. Nödenheten går på viss kurs med viss hastighet. Dess läge

forts nästa sida

```
VRELJ ETL#: 2

                    RENDEZ-VOUS
                    DETALJER
                    ENL.ETL# 2

POLISB RTVK=    207
GG123 RTVK=     27
RDVZ:TIDP=      9.53
RDVZ:LAT=       57.031
RDVZ:LON=       12.037

ETA GBG KL 10.36

SJÖRADDN.FALL #:..... DATUM:.../.../... SMC:.....
```

samt aktuell tidpunt inmatas. Räddningsenhetens startläge och prestanda inmatas. Ett obegränsat antal tilltänkta landningsplatser kan inläggas. Inom loppet av några sekunder beräknar datorn kurser för de båda enheterna, anger mötespunkt och mötestid samt visar räddningsenhetens ankomsttid till de olika landningsplatserna. Härigenom kan sjöräddningsledaren snabbt bestämma lämpligaste landningsplats med avsikt på tidsvinst samt till enheterna anvisa kurser etc.

```
DATA-SAR
PROGRAMVAL #: 6
ALT.VAL #: 1
RAKETHØJD:
OBS.PLATS:LAT: 57
OBS.PLATS:LON: 12
OBS.PLATS:HØH: 9
RAKETBRERING : 235
HØJDVINKEL   : 3

                    NØDRAKET
                    ALT:# 1

TEORETISK AVFYRNINGS-
PLATS:

LAT.  :  56.583
LON.  :  11.555

FELMARGINAL:
RADIE:  .7 (NM)

DIMENSIONERA SØKOMR.
EFTER OBSERVATIONENS
KVALITET !

SJÖRADDN.FALL #:..... DATUM:.../.../... SMC:.....
```

NÖDRAKETER:

Som rubriken anger gäller det här nödraketer. Programmet kan, om anmälaren lämnar goda uppgifter, bestämma avfyrningspositionen samt ange ett lämpligt sökområde. Programmet kan vidare analysera raketerna och så att säga sortera bort larm, som gäller observationer, som inte kan hänföras till kategorin "nödraketer".

Detta var i stora drag en presentation av en seriös användning av VIC-20. Datorn har fungerat klanderfritt under tiden den varit i drift. Den är kopplad till avbrottsfri kraft och har inte haft en enda "drop out".



Där slutar Ingvar sin redogörelse. Jag, som hade det stora nöjet att få se prov på programmet blev mest av allt imponerad av den oerhörda snabbhet med vilken datorn presenterade svaren och alltså gav sökområden eller andra uppgifter som sedan kunde användas av spaningsledningen. En annan sak, som Ingvar påpekade var mycket bra med VIC-20 var att just det förhållandet att man bara hade 22 tecken på bredden i bildskärmen medförde att man kunde läsa av skärmen från långt håll. Alltså kunde man stå vid sjökortsbordet och rita ut sina bäringar och ändå se uppgifterna på bildskärmen och detta utan att använda glasögon eller flytta sig närmare. Ingvar berättade också att de flesta medarbetarna var i åldrarna 35-55 och att man även därför inte kunde ha alltför liten stil på skärmtexterna. Vi vet ju alla hur synen försämras med åren.

För att ge en liten inblick i detta fantastiska program, bad jag Ingvar få återge de utskrifter, som löpande producerades under arbetet med "DATA-SAR". Dessa utskrifter tjänar som dokumentation och som kontroll av de uppgifter man ger in till programmet. Dessutom används de som underlag för TELEFAX-sända kopior till t.ex. befälhavaren på en räddningskryssare eller helikopter, Han får då alltså en skriven dokumentation bara några minuter efter ett larm och kan därför med större sannolikhet komma att lyckas fullgöra sitt uppdrag.

De utskrifter som visas här, är endast prover som gjordes under demonstrationen och alltså inte från ett verkligt sjöräddningsuppdrag. Därför visas också ur handboken en av de menyer, som visas på bildskärmen under arbetets gång.

Om själva programmet berättade Ingvar att det var skrivet med tanke på att absolut inte kunna gå att "krascha". Alla möjligheter till feltryckningar har eliminerats och med tanke på att programmet skall användas i en miljö, där inga fel får göras - det kan ju gälla människoliv - är det ännu viktigare att programmet när det används aldrig får "falla ur". Och det gör det inte heller, det har Ingvar sett till genom mycken klurig programmering, med noggrann analys av alla möjligheter till fel som kan göras i en pressad situation av en mindre van sjöräddare.

En mycket viktig fördel med användningen av VIC-20 är att man numera kan få fram de beräkningar som tidigare tog 2 ½ timme att göra manuellt, på ca 2,5 sek.!!

Det är roligt att kunna säga, att även om Göteborg Radio är ensamma om detta sätt att nyttja en VIC-20 och Sjöfartsverket ännu inte installerat systemet vid någon annan station så finns det andra som är mycket intresserade ute i Europa! Hösten och vintern lär visa i hur stor grad detta system kommer att accepteras som det effektivaste - prisbilligaste och enklaste "Search And Rescue"-systemet i världen. Kanske får vi ännu en svensk programprodukt att exportera! Ingvar Wihlborg önskas lycka till med Europa-lanseringen.

Åke Fredriksson

forts från sid. 1

Det skulle kunna innebära att datorrevolutionen kommer att medföra småstädernas och byarnas pånyttfödelse! Med terminaltekniken behöver vi inte längre flytta hela vår kropp för att på en central plats samlas för att utföra en gemensam arbetsuppgift. Vi kan sitta hemma vid terminalen och göra allt – utom möjligen fika tillsammans och umgås ansikte mot ansikte. Den sociala, mellanmänskliga delen i vår tillvaro kommer också att förändras i och med denna datorrevolution.

Man räknar med att utvecklingen inom datorområdet till 24/25:edelar kommer från små företag. Detta borde rimligen också leda till att dagens politiker, som är så frälsta på "storskalighet" måste lära om och tänka om. Visserligen är ett stort företag lättare att styra för staten i och med att man bara behöver ha kontakter med ett fåtal högt uppe i ledningen för att kunna styra utvecklingen, men det måste vakna ett intresse för, och en medvetenhet om, att det dock är "de svårstyrda" som starkast bidrar till utveckling och förändring i vår tillvaro. Det är ju den, som inte är nöjd med sakernas nuvarande tillstånd och som gör något åt det, som skapar förändring. Inte alla de, som i medgörlig apati bara finner sig i det ena övergreppet efter det andra från dem som makten hava.

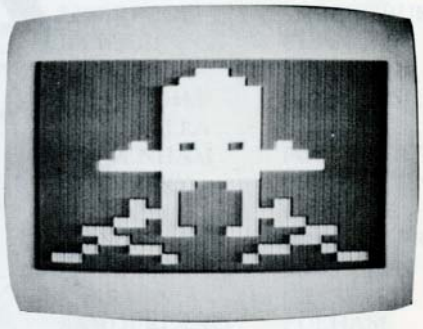
Alla de nya moderörelserna: "tillbaka till naturen", "småskalighet", "decentralisering", "miljörörelsen" och många fler borde också vakna till för de möjligheter datoriseringen innebär. Datorrevolutionen ger dem ju deras stora chans! Med en baskunskap om vad datorrevolutionen egentligen kan innebära har de ju ännu större chanser att förverkliga sina mål.

Därför är det ett gott råd jag vill ge så här till jul och nyår: Gack och skaffa dig en dator (naturligtvis håller jag på VIC!) och lär dig mer om vad den kan användas till! Vi kommer att göra allt vi kan i VIC-rapports spalter för att göra erfarenheten till en rolig och nyttig sysselsättning, det hoppas vi detta vårt stora dubbelnummer som är årets sista, ger exempel på. Och välkommen att prenumerera för 1984!

Åke Fredriksson

GOD JUL från VIC-rapport





Har du använt det extra skärm- och färgminnet som finns på din VIC-20? Om inte så har du chansen nu. Samtidigt får du en glad julhälsning från oss på VIC-rapport!

Kontrolläs programmet noga och lagra undan det på band eller disk innan du kör det. Eventuella felmeddelanden kan vara dolda på skärmen. Skriv då POKE 36879,27 och tryck på (RETURN)-tangenten (eventuellt får 27 bytas mot annat tal). Före rättning, tryck på (RUN) och (RESTORE) samtidigt.

Nu lite om skärmväxlingarna. POKE 56,28 : CLR skapar utrymme i minnet för extraskärmen. På ordinarie skärmbild (start vid 7680 och färgminne start 38400) ritas en figur (raderna 110-320 och extraskärmen på rad 380-410 (start vid 7168 och färgminne start 37888). Observera att skärm växlas först när PRINTkommando utförs.

Vill du använda extraskärmen för mycket utskrifter måste minnescellerna 217-240 uppdateras. Innehållet i dessa celler informerar om ifall en rad är startrad eller fortsättningsrad. Studera dessa värden med POKE och försök komma underfund med funktionen!

Som trevlig juluppgift kan du ju försöka göra om texterna till att visa mycket stora bokstäver på skärmen. Det skulle väl se trevligt ut med ett jättestort "GOD JUL" när du slår på TV:n på julaftonen före Kalle Anka & Co.?

Lycka till önskar **Lars Moeschlin**

```
100 POKE 56,28:CLR:PRINT CHR$(147)
110 POKE 36879,76
120 PRINT" ■■  ■     ■ "
130 PRINT"   ■ ■  ■  "
140 PRINT"    ■  "
150 PRINT"   ■  ■   ■  "
160 PRINTTAB(5)"■  ■ ■ "
170 PRINTTAB(6)"■  "
180 PRINTTAB(6)"■ ■ ■ "
190 PRINTTAB(8)"■ "
200 PRINTTAB(8)"■  "
210 PRINTTAB(8)"■▙   "
220 PRINTTAB(4)"■■▌  ▙  "
230 PRINTTAB(5)"■■▌ ▙  "
240 PRINTTAB(6)"■■ ■▙■ ■ "
250 PRINTTAB(8)"■▙ ▐▙ "
260 PRINTTAB(8)"■  "
270 PRINTTAB(8)"■  "
280 PRINTTAB(8)"■  "
290 PRINTTAB(8)"■ ▌▙ "
300 PRINTTAB(8)"■ ▐ "
310 PRINTTAB(9)"■▙  "
320 PRINTTAB(10)"■■  "
330 L=7680:GOSUB 350
340 L=38400:GOSUB 350:GOTO 380
350 FOR J=1 TO 21:K=L+J*22:FOR I=1 TO 10
360 POKE K+21-I,PEEK(K+I):NEXT I,J
370 RETURN
380 A=648:B=28:C=30
390 F=36866:G=22:H=150
400 POKE A,B:POKE F,G
410 PRINT CHR$(147)
420 S2=36876:V =36878
430 PRINT"■■■■■■■■■■"
440 POKE 36879,90
450 PRINT "■■■■    GOD JUL"
460 POKE 36879,90
470 POKE V,10
480 GOTO 630
490 FOR I=1 TO 2
500 FOR J=1 TO 4
510 READ P
520 IF P=-1 THEN 700
530 READ D
540 POKE S2,P+228
550 FOR N=1 TOD
560 FORM=1TO150:NEXT M
570 NEXT N
580 POKE S2,0
590 FOR N=1 TO 20:NEXT N
600 NEXT J:POKE A,C:POKE F,H
610 NEXT I
620 RETURN
630 GOSUB 490
640 GOSUB 740
650 GOSUB 490
660 GOSUB 780
670 GOSUB 490
680 GOSUB 820
690 GOSUB 490
700 POKE S2,0
710 FOR I=1 TO 1000: NEXT I
720 RESTORE
730 GOTO 400
740 POKE 36879,45:POKE A,B:POKE F,G
750 PRINT "■■■■■■■■■■"
760 PRINT "■■■■    GOD JUL"
770 RETURN
780 POKE 36879,90:POKE A,B:POKE F,G
790 PRINT "■■■■■■■■■■"
800 PRINT "■■■■    GOD JUL"
810 RETURN
820 POKE 36879,42:POKE A,B:POKE F,G
830 PRINT "■■■■■■■■■■"
840 PRINT "■■■■OCH GOTT NYTT ■■■■R"
850 RETURN
860 DATA  0,2, 7,2
870 DATA  7,1, 9,1
880 DATA  7,1, 6,1
890 DATA  3,2, 3,2
900 DATA  3,2, 9,2
910 DATA  9,1,11,1
920 DATA  9,1, 7,1
930 DATA  6,2, 0,2
935 REM
940 DATA  0,2,11,2
950 DATA 11,1,12,1
960 DATA 11,1, 9,1
970 DATA  7,2, 3,2
980 DATA  3,1, 3,1
990 DATA  0,2, 7,2
1000 DATA 6,2, 7,2
1010 DATA -1
1020 REM KONTROLLERA
1030 REM LAGRA UNDAN
1040 REM SEN RUN
```



Tippa stryktips med din expanderade VIC-20.

Förhoppningsvis finns det stryktipstippare bland VIC-20-användarna. Det här redovisade tipsprogrammet förutsätter att din VIC har expanderats med minst 8 K. Inledningsvis skall också sägas, att programmet vänder sig till de tippare, som kan offra tid och intresse åt att kontinuerligt följa upp säsongens matchresultat. Detta är nödvändigt, då programmet vid sin tipsradframtagning tar hänsyn till just dessa saker. Programmet som presenteras fungerar som program. Huruvida det ger storkovan till brukarna, får framtiden utvisa. Tipsraden kan "färgas" av användaren inom vissa gränser, men mer om detta längre fram. En s.k. "slumpfunktion" ingår också.

Programmet, enligt bifogad listning är avsett för stryktips med engelska lag ur div. 1 och div. 2. Med mycket små ändringar, kan programmet fungera för svenska lag. Tipsprogrammet är skrivet i Basic och av den anledningen lätt att följa. Programmet har byggts upp enligt följande:

1. Inmatning av resultat
2. Status
3. Tippning
4. Tabeller

Programkörning

(Lagra programmet på bandsida#1 och matchdata på sida#2). Programmet läggs in och startas med "RUN". På skärmen frågas: "FINNS MATCHDATA LAGRAT I DATAFIL (Y/N)?". Är ditt svar (N), vilket det måste vara vid allra första användningstillfället, går programmet till en underrutin för öppnandet av en datafil till kassettstationen. De 44 lagen matas nu in och besvara samtliga frågor som ställs. Svaren på frågorna finner du i varje dagstidning med tipsservice. Efter avslutad inmatning och lagring återgår programmet till "RUN" och inledningsfrågan dyker upp på nytt. Från och med nu och vid framtida användning, besvaras denna fråga med (Y). (Du har nu matchdata i datafil).

Observera vidare följande. Vid vissa tillfällen stannar programmet och visar en blinkande markör utan att någon fråga ställs. Avsikten är att ge dig en liten paus eller presentera ett meddelande på skärmen. Tryck på "RETURN" och programmet fortsätter.

Efter Y-svaret läses datafilen in och du kommer så småningom fram till menyn. Väljer du nu#4 visas efter ca 20 sekunder en serietabell. Jämför den med tidningens. Vid överensstämmelse innebär detta, att din datafil för matchdata är OK.

För att bena upp de olika detaljerna i menyn, väljer du nästa gång STATUS. Skriv in lagets namn och nästan omedelbart visas lagets under säsongen spelade matcher. Bilden avslutas med frågan KORREKT (Y)? Är den korrekt svarar du med Y och nästa lag kan plockas fram. Är

inte "statusbilden" ok, tryck på return. Du har nu möjlighet att korrigera eventuellt glömda inmatningar etc. Lag för lag kan plockas fram och när alla korrektioner gjorts, åstadkommer ett dubbeltryck på return att du uppmanas lagra korrektionerna på bandet.

Menyn #1 innebär resultatinmatning. I denna sekvens förväntar sig datorn resultatinmatning typ "Arsenal - Stoke 3-0". Resultaten sorteras och lagras på rätt ställe. Ett dubbeltryck på "return" uppmanar dig ånyo att "lagra i bandfil".

Under sekvensen "TIPPNING" bestämmer du först antalet matcher. (i regel 13). Därefter matas tipsomgången in, match för match. Då en match skrivits in, visas två fält - ett för hemmalaget och ett reverserat för bortalaget. Du uppmanas att fördela fem tecken under 1a, X och 2a. Du kan ignorera denna uppmaning och fördela hur många tecken du vill. Kanske vill du mata in "tio tidningars tips" eller kanske "London tippar". Just i detta avsnitt kommer din personliga "färgning" av raden.

Vi förutsätter nu att kupongens alla tretton matcher matats in. På skärmen visas nu vilka system du kan tippa. I princip kan du välja alla multiplikationssystem från 8 till 243 rader. Stämmer inte ditt val med vad som finns i datorn eller du väljer "1" visas en enkelrad och du kan gardera efter behag.

I detta skede skall också påpekas, att teckenframtagning - den egentliga tippningen - bygger på just mina värderingar. Genom att ändra koefficienter och formler och låta gjorda mål, insläppta mål, hemma- resp. bortavinster etc. få annan betydelse än jag valt, kan du själv konvertera programmet till dina önskemål. En sak kan dock fastslås, att först när "formpoängen" kompletterats, dvs. programmet körts så länge att formpoängen bestämts, tippar datorn ganska bra. Datorn väljer vissa överraskningar.

Några viktiga punkter att beakta:

Du kan tippa hur många system du önskar. Allt ligger kvar tills "RUN" exekveras. Run-kommandot tömmer ju alla variabler. Behöver du någon gång verkställa "RUN/STOP RESTORE", skriv därefter GOTO 78. På denna adress ligger menystarten.

Programförklaring

(istället för "REM-satser" i listan)

Rad	000- 077:	Inläsning från data-fil.
Rad	078- 099:	Meny-delen.
Rad	100- 151:	Inmatning av resultat.
Rad	152- 180:	Status.
Rad	181- 188:	Korrektioner.
Rad	200- 290:	Tippning, del:1
Rad	810-1055:	Tippning, del:2
Rad	1100-1125:	Nystart (öppnandet av data-fil).
Rad*1300-1335:		Systemval.
Rad*2000-2125:		Tabellsortering och utskrift.

Naturligtvis är det viktigt att du använder rätt lagbeteckning i programmet. Av den anledning har jag skrivit lagnamnen som de skall användas och som datorprogrammet hittar:

MANCHEST.U	SHEFF.W
WESTHAM	MANCHEST.C
QPR	CHELSEA
SOUTHAMPT.	NEWCASTLE
LIVERPOOL	SHREWSBURY
IPSWICH	CHARLTON
LUTON	BARNSLEY
NOTTINGH.F	HUDDERSF.
ASTONVILLA	GRIMSBY
TOTTENHAM	PORTSMOUTH
COVENTRY	BLACKBURN
WBA	CARLISLE
BIRMINGHAM	MIDDLESBR.
ARSENAL	BRIGHTON
EVERTON	CRYSTAL P
SUNDERLAND	FULHAM
NORWICH	CARDIFF
WATFORD	LEEDS
STOKE	OLDHAM
NOTTS CO	CAMBRIDGE
WOLVES	DERBY
LEICESTER	SWANSEA

Skulle du tycka att detta långa program är jobbigt at skriva in, kan du köpa programmet på kassett. Vid sådant köp, finns programmet på kassettens sida:1 och matchdata fram till aktuellt datum på sida: 2. Programmet säljs av: Ingvar Wihlborg, Syrengatan 14B, 434 00 Kungsbacka. Tel: 0300/119 80.

*** DATA-TIPS ***

```
0 POKE36879,25:PRINT"□":A$="DATA-TIPS"
1 REM *DATA-TIPS* AV  INGVAR WIHLBORG,      KUNGSBACKA
2 PRINTTAB(6)A$:PRINTTAB(6)"          ":PRINT
30 DIMH$(13):DIMG$(13):DIMBB(44)
40 DIMET(13):DIMKR(13):DIMTV(13):DIMFP(44,4):DIMPF(44)
50 DIMT$(13):DIMGA$(13)
51 DIML$(44):DIMV(44):DIMO(44):DIMF(44):DIMGM(44):DIMIM(44):DIMMS(44):DIMP(44)
52 DIMHV(44):DIMHO(44):DIMHF(44):DIMBV(44):DIMBO(44):DIMBF(44)
53 PRINT:Y$="":PRINT"FINNS MATCHDATA LAGRAT":PRINT:INPUT"I DATAFIL(Y/N)";Y$:IFY$
="N"THEN1100
54 PRINT"□":PRINT:PRINT"DATA ON TAPE(S:2) #000":INPUT"<CR>";ZZ$
55 REM INLASN. AV MATCH-DATA FR. BAND
56 OPEN1,1,0
60 FORN=1TO44
61 INPUT#1,L$(N)
62 INPUT#1,V(N)
63 INPUT#1,O(N)
64 INPUT#1,F(N)
65 INPUT#1,GM(N)
66 INPUT#1,IM(N)
67 INPUT#1,MS(N)
68 INPUT#1,P(N)
69 INPUT#1,HV(N)
70 INPUT#1,HO(N)
71 INPUT#1,HF(N)
72 INPUT#1,BV(N)
73 INPUT#1,BO(N):INPUT#1,BF(N):INPUT#1,FP(N,1):INPUT#1,FP(N,2)
74 INPUT#1,FP(N,3):INPUT#1,FP(N,4):INPUT#1,PF(N)
75 NEXTN:INPUT#1,A:CLOSE1
76 W$="(S)":SC=6:IFA=3THENW$="(E)":SC=9
77 I=0:PRINT"TAPE TO 000":INPUT"<CR>";TP$
78 PRINT"□":PRINT:PRINTTAB(7)"DATA-TIPS";W$:PRINTTAB(7)"          ":PRINT:CLOSE9:
CLOSE1
79 PRINT"#1:RESULTATINM.":PRINT
80 PRINT"#2:STATUS":PRINT
85 PRINT"#3:TIPPNING":PRINT
87 PRINT"#4:TABELLER"::PRINT:PRINT
90 INPUT"#-VAL";Y
```

forts. nästa sida

```
95 ONYGOTO100,152,200,2000
100 PRINT"■";TAB(7)A$;W$:PRINTTAB(7)"──────":PRINT
101 PRINTTAB(4);"RESULTATINMATN.":PRINTTAB(4);"──────":PRINT:PRINT:OPEN
9,0
102 PRINT"MATCH":PRINT"──────":PRINT:X$=""
103 INPUT#9,X$:PRINT"- ";
104 IFX$=""THENCLOSE9:GOTO132
105 INPUT#9,Y$:PRINT:PRINT:PRINT:PRINT
106 FORN=1TO44:IFX$=L$(N)THENH=N:GOTO108
107 NEXTN:CLOSE9:GOTO100
108 FORN=1TO44:IFY$=L$(N)THENB=N:GOTO110
109 NEXTN:CLOSE9:GOTO100
110 PRINT"■■■■■■";:INPUT#9,X:PRINT"- ";
111 INPUT#9,Y
112 CLOSE9
115 GM(H)=GM(H)+X:GM(B)=GM(B)+Y:IM(H)=IM(H)+Y:IM(B)=IM(B)+X
116 MS(H)=GM(H)-IM(H):MS(B)=GM(B)-IM(B)
117 IFX=YTHENO(H)=O(H)+1:O(B)=O(B)+1:HO(H)=HO(H)+1:BO(B)=BO(B)+1:GOTO120
118 IFX>YTHENV(H)=V(H)+1:F(B)=F(B)+1:HV(H)=HV(H)+1:BF(B)=BF(B)+1:GOTO121
119 IFX<YTHENF(H)=F(H)+1:V(B)=V(B)+1:HF(H)=HF(H)+1:BV(B)=BV(B)+1:GOTO122
120 FP(H,4)=1:FP(B,4)=1:GOTO123
121 FP(H,4)=A:FP(B,4)=0:GOTO123
122 FP(H,4)=0:FP(B,4)=A
123 P(H)=V(H)*A+O(H):P(B)=V(B)*A+O(B)
124 FP(H,1)=FP(H,2):FP(H,2)=FP(H,3):FP(H,3)=FP(H,4):FP(B,1)=FP(B,2)
129 FP(B,2)=FP(B,3):FP(B,3)=FP(B,4):PF(H)=FP(H,1)+FP(H,2)+FP(H,3)
130 PF(B)=FP(B,1)+FP(B,2)+FP(B,3)
131 GOTO100
132 OPEN1,1,2:PRINT"■","MATCHRES. OCH EVT.":PRINT"KORREKTIONER LAGRAS NU"
133 FORN=1TO44
134 PRINT#1,L$(N)
135 PRINT#1,V(N)
136 PRINT#1,O(N)
137 PRINT#1,F(N)
138 PRINT#1,GM(N)
139 PRINT#1,IM(N)
140 PRINT#1,MS(N)
141 PRINT#1,P(N)
142 PRINT#1,HV(N)
143 PRINT#1,HO(N)
144 PRINT#1,HF(N)
145 PRINT#1,BV(N)
146 PRINT#1,BO(N)
147 PRINT#1,BF(N)
148 PRINT#1,FP(N,1):PRINT#1,FP(N,2):PRINT#1,FP(N,3):PRINT#1,FP(N,4)
149 PRINT#1,PF(N)
151 NEXTN:PRINT#1,A:CLOSE1:GOTO78
152 PRINT"■":N=0:PRINTTAB(15)"STATUS■■■■■■■■■■■■■■■■"
153 LX$="":INPUT"LAG";LX$:L=LEN(LX$):IF(LX$="")AND(K0=1)THENK0=0:GOTO132
154 IFLX$=""THEN78
155 N=N+1
156 IFLX$=L$(N)THENPRINT"■";LX$:FORP=1TOL:PRINT"■";:NEXTP:PRINT"■ F■:"PF(N);"■
":GOTO160
158 IFN=44THEN152
159 GOTO155
160 PRINT" V:";TAB(4)" O:";TAB(8)" F:";TAB(12)" MS:  P:
165 PRINT"──────────"
170 PRINTV(N);TAB(4)O(N);TAB(8)F(N);TAB(12)MS(N);TAB(17)P(N)
171 PRINT"----------──────"
175 PRINT"H:";TAB(12)"| "
176 PRINTHV(N);TAB(4)HO(N);TAB(8)HF(N);TAB(12)"|  TOT.MA■■L.■L"
177 PRINT"B:";TAB(12)"| ";"──────"
178 PRINTBV(N);TAB(4)BO(N);TAB(8)BF(N);TAB(12)"| "STR$(GM(N));" -";STR$(IM(N))
179 PRINT"──────────"
180 YY$="":PRINT"■■";:INPUT"KORREKT?(Y)";YY$:IFYY$="Y"THENPRINT"■■":GOTO152
181 K0=1:PRINT:OPEN9,0:PRINT"■■";" V   O   F          P  ":PRINT" ";
182 INPUT#9,V(N):PRINTTAB(5);:INPUT#9,O(N):PRINTTAB(9);:INPUT#9,F(N):PRINTTAB(17
);
183 P(N)=V(N)*3+O(N):PRINTP(N)
184 PRINT"■■";" HV  HO  HF          "
185 PRINT" ";:INPUT#9,HV(N):PRINTTAB(5);:INPUT#9,HO(N):PRINTTAB(9);:INPUT#9,HF(N
):PRINT
186 PRINT"■■";" BV  BO  BF   GM  IM ":PRINT" ";
187 INPUT#9,BV(N):PRINTTAB(5);:INPUT#9,BO(N):PRINTTAB(9);:INPUT#9,BF(N):PRINTTAB
(14);
188 INPUT#9,GM(N):PRINTTAB(18);:INPUT#9,IM(N):MS(N)=GM(N)-IM(N):CLOSE9:PRINT"■■"
:GOTO152
200 PRINT"■"
201 PRINTTAB(7)A$;W$:PRINTTAB(7)"──────":PRINT:PRINT
202 INPUT"ANTAL MATCHER";J
204 FORI=1TOJ:PRINT"■"
205 PRINTTAB(7)A$:PRINTTAB(7)"──────":PRINT:PRINT:OPEN9,0
210 PRINT"MATCH #";I:PRINT"──────":PRINT
215 INPUT#9,H$(I):PRINT"- ";
220 INPUT#9,G$(I)
225 FORN=1TO44
```

```
230 IFH$(I)=L$(N)THENH=N:GOTO235
231 NEXTN:GOTO210
235 FORN=1TO44
240 IFG$(I)=L$(N)THENB=N:GOTO245
241 NEXTN:GOTO210
245 CLOSE9:S=INT(RND(TI)*7+7):W=RND(TI)*.7+.7:T=INT(RND(TI)*5+1)
246 OT=HV(H)+BF(B)+HO(H)+BO(B)+BV(B)+HF(H)
247 ET=(HV(H)+BF(B))/OT*S:ET=ET+ET*PF(H)/SC
248 KR=(HO(H)+BO(B))/OT*(S+T):KR=KR+KR*(PF(H)+PF(B))/(2*SC)
249 TV=(BV(B)+HF(H))/OT*(S+T):TV=TV+TV*PF(B)/SC
250 IFP(H)>P(B)THENET=ET+P(H)/P(B)
251 IFP(H)=P(B)THENKR=KR+2
252 IFP(B)>P(H)THENTV=TV+P(B)/P(H)
253 IFGM(H)>GM(B)THENET=ET+W
254 IFGM(H)=GM(B)THENKR=KR+W+T*.1
255 IFGM(B)>GM(H)THENTV=TV+W
256 IFIM(B)>IM(H)THENET=ET+W/2
257 IFIM(B)=IM(H)THENKR=KR+W/2+T*.05
258 IFIM(H)>IM(B)THENTV=TV+W/2
259 IFMS(H)>MS(B)THENET=ET+1:GOTO262
260 IFMS(H)=MS(B)THENKR=KR+1.5:GOTO263
261 IFMS(B)>MS(H)THENTV=TV+1:GOTO264
262 IF(MS(H)-MS(B))>9THENET=ET+1:GOTO265
263 IFMS(H)>9THENKR=KR+1:GOTO265
264 IF(MS(B)-MS(H))>9THENTV=TV+1
265 OPEN9,0
266 PRINT"■";L$(H);"(HEMMALAG)":PRINT"──────────────────────":K=H
267 PRINT" HV  HO  HF  P:T FORM":PRINT"──────────────────────"
268 PRINTHV(K);TAB(4)HO(K);TAB(8)HF(K);TAB(12)P(K);TAB(17)PF(K):PRINT:PRINT"──────────":PRINT
269 PRINT"■";L$(B);"(BORTALAG)":PRINT"■";"                    ":PRINT"──────────":K=B:PRIN
270 PRINT"■";" BV  BO  BF  P:T FORM":PRINT"■";"──────────────────────"
T
271 PRINTBV(K);TAB(4)BO(K);TAB(8)BF(K);TAB(12)P(K);TAB(17)PF(K):PRINT:PRINT"──────────"
272 K=0:PRINT:PRINT"■■■
```

```
997 GOTO994
1000 IFQ<ANTHENS=INT(RND(TI)*J+1):GOTO1002
1001 GOTO1010
1002 IFLEN(GA$(S))<2THEN1004
1003 GOTO1000
1004 IFT$(S)="1"THENGA$(S)="1X":GOTO1009
1005 IFT$(S)="X"THENGA$(S)="1X":GOTO1009
1006 IFT$(S)="2"THENGA$(S)="X2"
1009 Q=Q+1:GOTO1000
1010 FORN=1TOJ
1015 IFGA$(N)=""THENGA$(N)=T$(N):NEXTN
1035 IFXL=>1THENXL=XL-1:GOTO1037
1036 GOTO1041
1037 XY=INT(RND(TI)*13+1)
1038 IFLEN(GA$(XY))<>2THEN1037
1039 GA$(XY)="1X2":GOTO1035
1041 PRINT"MATCH";TAB(18)"RAD":PRINT"──────────────────":FORN=1TOJ
1042 H$=LEFT$(H$(N),7)
1043 G$=LEFT$(G$(N),7)
1044 IF(N=4)OR(N=7)OR(N=10)THENPRINT"--------------------"
1045 PRINTH$;TAB(7)"-";G$;TAB(15)GA$(N):NEXTN
1046 PRINT
1050 PRINT"SYSTEM:";SV;"RADER":AV$=STR$(SV*.60)+"0:-":PRINT"KRONOR: ";AV$
1055 INPUT"<CR>";Y$:GOTO1300
1100 PRINT"□";"MATCH-DATA VID NYSTART":PRINT
1103 PRINT"REW. TAPE(S:2) TO #000":PRINT:INPUTZZ$
1105 OPEN1,1,2:OPEN9,0
1109 FORN=1TO44:PRINT"□";"MATCH-DATA #";N:PRINT
1110 PRINT"LAG          ";:INPUT#9,X$:PRINT#1,X$:PRINT
1111 PRINT"VINSTER      ";:INPUT#9,X:PRINT#1,X:PRINT
1112 PRINT"ÅRVGJORDA    ";:INPUT#9,Y:PRINT#1,Y:PRINT
1113 PRINT"FØRLUSTER    ";:INPUT#9,Z:PRINT#1,Z:PRINT
1114 PRINT"GJORDA MÅL   ";:INPUT#9,V:PRINT#1,V:PRINT
1115 PRINT"INSL.  MÅL   ";:INPUT#9,W:PRINT#1,W:PRINT:PRINT#1,V-W:PRINT#1,X*3+Y
1116 FRINT"HEM.VINST.   ";:INPUT#9,X:PRINT#1,X:PRINT
1117 PRINT"HEM.ÅRVGJ.   ";:INPUT#9,X:PRINT#1,X:PRINT
1118 PRINT"HEM.FØRL.    ";:INPUT#9,X:PRINT#1,X:PRINT
1120 PRINT"BORTAVINST   ";:INPUT#9,X:PRINT#1,X:PRINT
1121 PRINT"BORTAÅRVGJ   ";:INPUT#9,X:PRINT#1,X:PRINT
1122 PRINT"BORTAFØRL.   ";:INPUT#9,X:PRINT#1,X:PRINT
1123 PRINT#1,0:PRINT#1,0:PRINT#1,0:PRINT#1,0:PRINT#1,0:NEXTN
1125 PRINT#1,3:CLOSE1:CLOSE9:RUN
1300 PRINT"□":U$="":Z=0:AZ=0:AN=0
1301 PRINTTAB(5)"SYSTEMVAL":PRINTTAB(5)"──────────":PRINT
1302 PRINT"  8 RADER":PRINT"  9 RADER":PRINT" 16 RADER":PRINT" 24 RADER":PRINT"
27 RADER"
1303 PRINT" 32 RADER":PRINT" 36 RADER":PRINT" 48 RADER":PRINT" 64 RADER"
1304 PRINT" 81 RADER":PRINT" 96 RADER":PRINT"128 RADER":PRINT"243 RADER":PRINT
1305 SV=0:INPUT"RADER (S)";U$:SV=VAL(U$):IFU$=""THENGOTO78
1306 IFSV=8THENXL=0:AN=3:GOTO1330
1307 IFSV=9THENXL=2:AN=2:GOTO1330
1308 IFSV=16THENXL=0:AN=4:GOTO1330
1309 IFSV=24THENXL=1:AN=4:GOTO1330
1310 IFSV=27THENXL=3:AN=3:GOTO1330
1311 IFSV=32THENXL=0:AN=5:GOTO1330
1312 IFSV=36THENXL=2:AN=4:GOTO1330
1313 IFSV=48THENXL=1:AN=5:GOTO1330
1314 IFSV=64THENXL=0:AN=6:GOTO1330
1315 IFSV=81THENXL=4:AN=4:GOTO1330
1316 IFSV=96THENXL=1:AN=6:GOTO1330
1317 IFSV=128THENXL=0:AN=7:GOTO1330
1318 IFSV=243THENXL=5:AN=5:GOTO1330
1320 XL=0:AN=0:GOTO1330
1330 Q=0:FORN=1TOJ:T$(N)="":GA$(N)="":NEXTN
1335 GOTO810
2000 PRINT"□";TAB(6)"TABELLER":PRINTTAB(6)"──────────":PRINT
2005 PRINT"#1: DIVISION 1":PRINT
2010 PRINT"#2: DIVISION 2":PRINT
2020 INPUT"TABELLVAL";TA
2025 ONTAGOTO2030,2035
2030 PRINT"□";TAB(5)"DIVISION 1":PRINTTAB(5)"──────────":PRINT:TX=1:TY=22:GOTO20
45
2035 PRINT"□";TAB(5)"DIVISION 2":PRINTTAB(5)"──────────":PRINT:TX=23:TY=44:GOTO2
045
2045 PRINT"LAG:";TAB(13)"MS:";TAB(17)"P:":PRINT"──────────────────"
2047 FORN=0TO44:BB(N)=0:NEXTN:CC=0
2050 FORN=TXTOTY:SM=400-(200-MS(N)):U$=STR$(N/100)
2051 IFGM(N)<10THENUM$="0"+STR$(GM(N)):GOTO2055
2052 UM$=STR$(GM(N))
2055 AA$=STR$(P(N))+STR$(SM)+UM$+U$
2060 AA=VAL(AA$)
2065 IFBB(CC)=<AATHEN2075
2070 CC=CC+1
2075 IFBB(CC)<>AATHEN2085
2080 CC=CC+1
2085 DD=BB(CC):BB(CC)=AA:AA=0
2090 IFDD<>0THENAA=DD:GOTO2065
2095 CC=DD
2100 NEXTN
2105 FORN=0TO21
2110 LX=INT((BB(N)-INT(BB(N)))*100+.5)
2115 PRINTL$(LX);TAB(12)MS(LX);TAB(16)P(LX)
2120 NEXTN
2125 INPUTZZ$:GOTO78

READY.
```

Å Ä och Ö på en expanderad VIC-20

Behovet av att kunna skriva de "svenska" bokstäverna Å Ä och Ö kan föreligga hos en del VIC-20 användare. Det här under beskrivna programmet ger dig denna möjlighet. Maskinprogrammet, som utför detta, ligger på adress 673–767, ett minnesområde som inte normalt används av VIC-20.

Den nya teckengeneratorn ligger på adress 4096–5119. Teckengeneratorn omfattar 128 tecken och upptar exakt 1 K RAMarea. Reverserade tecken samt gemener (små tecken) ingår inte i den nya teckengeneratorn. Skärmarean har flyttats till adr. 5120–5625 varvid Basicstarten flyttats till 5632. Har du expanderat din VIC med 16 K, har du ändå kvar ca. 19000 bytes. De nya tecknen finns på din VIC enligt följande:

] =Å : [=Ä : £ =Ö .

Programmet skrivs in enl. listan och lagras på band. Skriv därefter "RUN". På bildskärmen visas hur många K-bytes som är "free". Basicprogrammet, som "pokade" in maskinprogrammet, har nu suddats ut, men maskinprogrammet med start på adress 673 finns kvar tills du stänger av din VIC. Skriv nu SYS673. Markören ändrar form från "tegelsten" till ett vertikalt streck. Tryck på exempelvis £-tecknet och på skärmen skrivs Ö.

Du kan nu använda den nya teckengeneratorn i dina program, antingen du skriver nya eller har sparade på kassett. Kom bara ihåg, att du måste först spela in detta program. Vill du återgå till maskinens hela, ursprungliga teckengenerator verkställ RUN/STOP RESTORE. Markören återgår nu till normal form. Ett nytt SYS 673 ger Dig Å Ä och Ö.

Å Ä Ö

```
0 FORI=673TO767:READC:POKEI,C:NEXTI:SYS716:SYS64624
1 DATA169,0,133,87,133,89,169,128,133,88,169,16
2 DATA133,90,160,0,177,87,145,89,192,255,208,9
3 DATA230,88,230,90,160,0,76,195,2,200,169,132
4 DATA197,88,240,3,76,177,2,169,220,141,5,144
5 DATA169,22,141,130,2,169,20,141,136,2,162,0
6 DATA189,233,2,157,216,16,232,224,23,208,245,96
7 DATA36,0,24,36,126,66,66,0,36,0,60,66
8 DATA66,66,60,0,24,0,24,36,126,66,66
```


Computer Books

Böcker till VIC-20/64

	(priser exkl. porto)
Getting acquainted with your VIC 20	116:—
VIC Innovative Computing	125:—
ZAP! POW! BOOM! Games for VIC 20	145:—
Symphony for VIC 20	116:—
VIC 20 Exposed	135:—
The Working Commodore 64	115:—
Commodore 64 Games Book	115:—
Commodore 64 Exposed	135:—
Commodore Machine Code Master	135:—
Commodore 64 Adventures	135:—

Beställ katalog!

Studieförlaget
Box 386, 751 06 Uppsala 1, Telefon 018-15 53 90



Spel för oexpanderad VIC 20

☐ Frog	85 Kr
☐ Galaxzions	90
☐ VIC Rescue	90
☐ Alien Attack	95
☐ Fantazia	95
☐ Crazy Kong	95
☐ Jupiter Defender	95

Spel för VIC 64

☐ Frogger 64	125
☐ Panic 64	125
☐ Scramble 64	125
☐ Spriteman 64	125
☐ Star Trek 64	125
☐ Crazy Kong 64	125

Kryssa för de spel du vill ha och skicka annonsen till PROGRAMBODEN, Box 225, 13154 Nacka
☐ Jag betalar i förskott via postgiro 4767997-2
☐ Jag betalar via postförskott (frakt tillkommer)

Namn

Adress

Tel

Skatteplanera med VIC-20/64

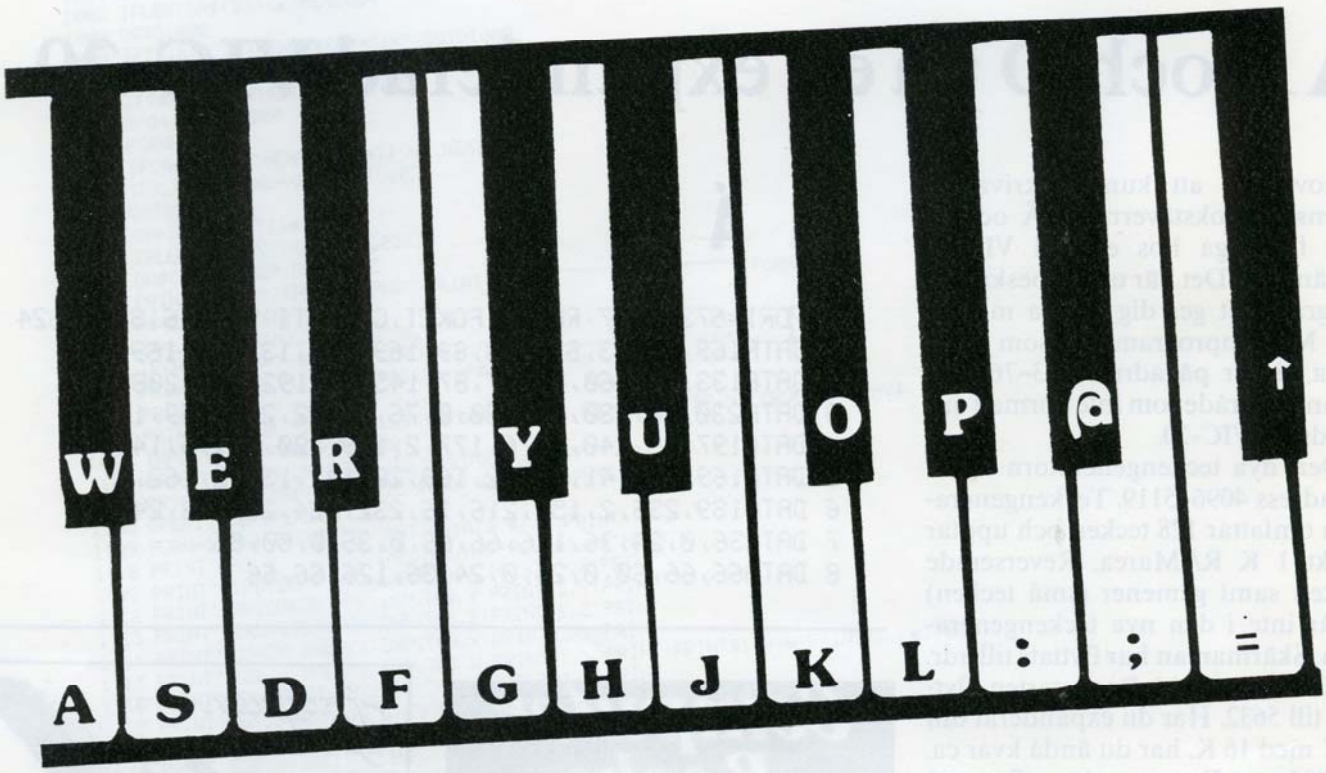
Fyra program, framtagna av Expressens Anders Andersson som räknar ut din skatt, hjälper dig sänka skatten och visar hur skattereformen slår 1983—1985. Sätt in 180:- på pg. 80 94 71-6 så kommer programmen med utförlig instruktion.

EKONOMIDATA Poppelvägen 55, 135 52 Tyresö. 08-712 46 46

VIC-20			VIC-64	
GRIDRUNNER	oexp	75:-	ATACK OF MUT. CAMELS	115:-
LASER ZONE	+8-16K	90:-	GRIDRUNNER 64	110:-

inventec
Box 33015 200 50 MALMÖ

Postförskott: +10:-
Postgiro 430 61 48-0: inga tillägg
Även exp.kort, extra minne m.m.
Gratis spel- o tillbehörskatalog



Spela julsångerna på din VIC-20



VIC-20 är ganska enkel att göra om till ett litet piano. Tangenterna i mittersta bokstavsraden ger de vita tangenterna eller heltonerna på pianot. För halvtonerna, de svarta tangenterna, använder man w e r, y u och o p å. Dvs i det program som återges i den här artikeln! Naturligtvis kan du programmera din VIC-20 så att du får användning för de tangenter som passar dig bäst, men då får du allt titta efter i Programmers Reference Guide till VIC-20 först.

VIC-ljudet

Inbyggd i maskinvaran finns ett ljudgenereringssystem på VIC-20. De signaler som kommer därifrån leds in till ljudkanalen på din TV och du kan återge realistiska effekter med ett minimum av programmering.

Tre oberoende tongeneratorer kan aktiveras och blandas. Var och en med ett frekvensomfång på tre oktaver. På så sätt kan trestämmig musik återges, om du ändrar frevensen på tongeneratorerna. Dessutom kan volymen kontrolleras med program-kommandon. För vissa speciella ljudeffekter som t.ex. explosioner eller laserskott, finns också en s.k. "vitt" ljud- eller brusljudsgenerator.

För att få kontroll över ljudmöjligheterna i din VIC-20 skall du POKEa registren i VIC-chipet så här:

Decimal	Hex	Funktion
36874	$900A	Tongenerator 1. 0 ger inget ljud, 128-255 ger en ton ifrån sig (se nedan)
36875	$900B	Tongenerator 2.
36876	$900C	Tongenerator 3.
36877	$900D	Brusljudsgenerator. Uteffekten varierar från surr av flygplan till vindens vinande.
36878	$900E	Volym. 0 ger lägsta volym och 15 den högsta. OBS! glöm inte dra på volymen på TV:n!

Hur man använder tongeneratorn

Även om ljud kan skapas med enkla POKE-kommandon i direkt-mod, är den vanligaste metoden att använda ett BASIC-program. Toner läses in som textsträngar från data-satser, omvandlas till POKE-värden och aktiveras därefter i FOR...NEXT-slingor för att få rätt taktart och längd. Alternativt kan ett litet program skrivas för att GETta tecken från tangentbordet, så att VIC-20 fås att simulera en enkel trestämmig elektronisk orgel! Alltefter det tiden går, kommer våra VIC-användare säkert att bygga upp ett aktningsvärt bibliotek med subrutiner för generering av specialeffekter och spel-ljud.

Piano-programmet

För att du skall få en idé om vilken ljudkapacitet VIC-20 har, tänkte vi oss att du skulle kunna roa dig med det följande enkla pianoprogrammet. Det börjar på tonen F och går ungefär en och en halv oktav upp till C*. Pianots vita tangenter "spelas" på tangenterna A, S, D, F, G, H, J, K, L, :, ; och =. De svarta tangenterna har, som framgår av bilden, lagts på bokstäverna W, E, R, Y, U, O, P, @ och Å. Programmet kommer inte att göra dig till en Mozart, men det bör ge dig ett gott begrepp om hur du kan använda din VIC-20 till ljudframställning. Mer detaljerad information finns i VIC-20 Programmers Reference Guide, som nu publiceras på svenska från Handic/Förlagsgruppen i tre delar där den första redan finns ute i butikerna.

OBS! Vi fick in ett bidrag från Ingarö som ger dig ännu bättre ljudmöjligheter. Titta på det programmet också!

```
10 S2=36874
20 V=36878
30 DIMA(36)
50 FORN=0TO36
60 READA(N)
70 NEXTN
80 GETA$:IFA$=""THEN80
90 IFA$="←"THEN200
100 N=ASC(A$):IFN<58ORN>94THEN80
110 POKES2,0
120 POKEV,15
130 POKES2,A(N-58)
140 GOTO80
200 POKEV,0
210 END
300 DATA237,239,0,240,0,0,238,209,0,0
310 DATA219,217,223,225,228,0,231,232,235,0
320 DATA0,233,236,0,221,215,0,229,0,212
330 DATA0,227,0,0,0,0,241

READY.
```

ASSEMBLER MONITOR.EDITOR M.M.
PLUGG-IN KASSETT FÖR OEXP. ELLER EXP. VIC
SNART ÄVEN TILL CBM 64
VIC SOFT BOX 2 52100 FALKÖPING
PRIS 500:-

Brev från en läsare:

Kräv bättre programvara och datortidningar

Under helgen försökte jag köra in ett program från en datortidning. Det var allt annat än lätt. För att få plats med så mycket eget bildmateriel som möjligt, så hade de förminskat programlistan. Det var nästan omöjligt att läsa texten. När det sedan gällde de grafiska tecknen som det blir av CLR, CRSR, RVS m.m. så var det bara att gissa sig fram. Följden blev att programmet inte gick att köra. Även programmakaren hade gjort sitt för att försvåra inprogrammeringen. Inga som helst mellanslag fanns mellan kommandona.

Det är många som tycks tro att det finns en obegränsad marknad att ösa ur. "Bara det står data på omslaget, så säljer det". Marknaden för datortidningar komer att bli stenhård. Det finns ingen som helst anledning att köpa dåliga produkter. Gör som jag. Gå tillbaka till handlaren med tidningen och förklara att den inte går att läsa.

Så några råd till dig som programmerar; använd mellanslag mellan kommandona. Då blir listan lättare att läsa.

Använd svenska när texten skall visas på skärmen. Det är inte alla som kan engelska på gymnasienivå. Ett program som bara ett fåtal begriper är värdelöst.

När grafiska tecken, såsom CRSR m.fl. används, så förklara med en REM-sats vilka tecken du har använt alla har inte en skrivare så de kan pröva sig fram. Är sedan färgbandet dåligt, eller tidningstrycket suddigt. Då kan ingen läsa ditt program.

Pröva programmet på din moster, eller någon annan som inte är datafrälst. Begriper de programmet, då är det bra. Det finns ingen som helst marknad för obegripliga program.

Alf Olsson

Bättre bild och några andra tips till VIC-20

Hur ser färgerna på din skärm ut egentligen? Är den gula färgen mer lik en hårdkokt äggula, eller är den rent av grön?

I den här artikeln skall du få lära dig hur man åtgärdar detta. Låt mig dock få säga på en gång att om du inte tillhör dem som reparerar de trasiga kontakterna i ditt hem eller gör andra små arbeten så LÅT NÅGON ANNAN GÖRA DE SAKER SOM BESKRIVS HÄR. I annat fall finns det risk att du blir ägare till en trasig VIC 20. Läs också igenom HELA artikeln INNAN du utför något av det som står här.

Om man har dåliga färger på sin skärm eller en kontrast som inte överensstämmer med det man normalt kallar kontrast så finns det faktiskt möjligheter att åtgärda detta på VICKE utan att man behöver göra några större ingrepp. Detta är något som är särskilt intressant för dig som vill koppla din dator direkt till en monitor utan den RF-generator som krävs när man kör på en vanlig TV. När man gör detta brukar nämligen bilden vara både sned och vind och uppvisa de mest konstiga färger. Detta beror naturligtvis inte på att det är något fel på VICKE utan helt enkelt på att man på fabriken trimmar datorn så att den ger en bra bild kopplad till en vanlig TV över den RF-generator som medföljer. Eftersom nu olika TV-apparater varierar kraftigt i utförande så passar denna inställning inte för alla.

Lösningen på problemet är helt enkelt att man öppnar själva lådan på VICKE och lokaliserar två trimpotentiometrar som används för att trimma bilden. Själva justeringen går till så här.

1) Dra ur kontakten ur väggen. Ta sedan ut kontakten från transformatorn ur VICKE. Försäkra dig om att INGEN KONTAKT ÄR INKOPPLAD IN TILL VIC:

2) Vänd nu datorn upp och ned. På undersidan i framkanten kan du hitta tre skruvar som sitter i rad. Skruva upp alla desa.

3) Vänd nu återigen på datorn så att den står som vanligt. Lyft på övre delen av lådan så att du kommer åt innandömmet. Det är lämpligt att ställa en låda eller något bakom så att man kan luta tangentbordet mot denna.

4) Inuti datorn så hitar du en "silver-skiva" som du viker upp mot dig och din dator är frilagd in på bara benen. Nu kliar det kanske i dig att börja peka på alla de kretsar som sitter här. RÖR INGEN AV DE KRETSAR SOM DU SER VARE SIG MED FINGRAR ELLER ANDRA FÖREMÅL. Gör du det och har en smula otur så kan du förstöra din dator.

Det bör kanske tilläggas att det krävs ganska mycket otur men det är ju onödigt att chansa eller hur? Ungefär i mitten på tangentbordet hittar du en stor krets som det står 6561 på. Detta är den krets som sköter om bildhanteringen på din dator och också gett VIC-20 dess namn. Medan vi i alla fall har öppnat lådan så kan vi ta och titta vad som mer finns här. Till höger om 6561:an kan du hitta två andra kretsar som är lika stora. Detta är kretsar som sköter all kommunikation med omvärlden i ditt system och det är hit joysticks, användarport och kassett-kontakten är kopplad. Själva hjärnan i din dator, Processorn har du mot dig. Detta är också en krets med 40 ben och det står 6502 på den. Det är denna krets vi lär oss att styra i den assemblerskola som finns i detta nummer. Till höger om denna kan du hitta två kretsar med 24 ben. Här ligger minnen som innehåller BASIC och KERNAL dvs själva operativsystemet. Dessa ligger i s.k. ROM-minnen. Mot dig räknat från 6561:an finns en annan krets som har 24 ben. Denna innehåller definitioner av alla de tecken som man kan skriva på tangentbordet och kallas följdaktligen teckengenerator. I själva verket är också detta bara ett ROM-minne precis som de andra två 24-bens-kapslarna.

Till vänster om tecken generatorn ligger prydliga rader av mindre IC-kretsar som det står 2114 på. Detta är det RAM minne, läsminne, som finns i din dator. Och i dessa lagras dina program samt färg; bild-information för skärmen. Resterande kretsar som finns är till för att anpassa spänningsnivåer till bestämda värden och avkoda t.ex. en bestämd minneskrets. Låt oss då återvända till vår uppgift.

Ovanför 6561:an hittar du en hel hög av komponenter men mitt ibland dessa kan du se två trimpotentiometrar. Dessa ligger bakom varandra i en rak linje precis bakom serieutgången på baksidan. De är gjorda i vit plast och har en skåra i sig som precis passar för en liten skruvmejsel. Med dessa kan vi justera bilden enligt önskan. Med den vänstra av dessa kan du ställa in amplituden för den signal som skall sändas till din TV, dvs ungefär den topp till topp spänning som skall vara på TVutgången. Den till höger reglerar DCnivån på signalen och hissar i praktiken upp/ner TVsignalen.

Dessa båda egenskaper hos TV-singnalen påverkar varandra så rör man på en av dem så måste man i de flesta fall även justera den andra.

5) koppla nu in strömmen till din dator igen. Dvs sätt i både nätkontakten i väggen och kontakten i VICKE. Spänningen in till VICKE är bara på 10 volt så att den är inte alls farlig att vidröra. INNAN DU SLÅR PÅ STRÖMMEN TILL DATORN så försäkra dig om att

video sladden är kopplad till datorn. Det är mycket lätt att en säkring går om du glömmer bort att göra detta. Skulle detta ändå ske så kan du hitta säkringen vid högra kortsidan precis där nätsladden går in. Här skall det sitta en 3A trög radiosäkring, en modell som du kan få tag i på i alla radio/TV-affärer. Är tråden i denna säkring av så har den alltså gått sönder och måste bytas. OBS! om denna säkring går igen när du har bytt den så är något allvarligt fel och det är dags att skicka in din dator för reparation.

Vid denna justering behöver du dock inte bekymra dig om detta så länge du inte har datorn påkopplad med videosladden urdragen.

6) Knäpp nu på din dator och justera in en bra bild. Detta gör du genom att justera fininställningen till kanalväljaren och ser till att denna ligger på UHF kanal 36. Har du en monitor eller en direkt videoingång så får du utföra justeringen med de kontroller som just din monitor har. När du är nöjd med bilden så skriv ut reverserade fyrkanter på skärmen i alla tillgängliga färger. Detta gör du genom att slå på RVS ON och sedan CNTRL + färg samt trycker på mellanslag. Alla åtta färgerna skall synas på skärmen. Genom att sedan växelvis vrida på de två potentiometrarna så justerar du sedan in en bra bild. Bli inte rädd om den börjar flimra eller helt försvinner, genom att justera "pottarna" kan du återigen få en bra bild. Ha stort tålamod och vrid inte för mycket varje gång. Inställningen är mycket känslig och små justeringar ger en stor förändring. Som ett riktmärke kan man ha den blå ramen, så länge denna inte är blå så har man inte justerat in korrekt. För övrigt är det lättast att titta på den gula färgen som ofta tenderar at krypa mot grönt.

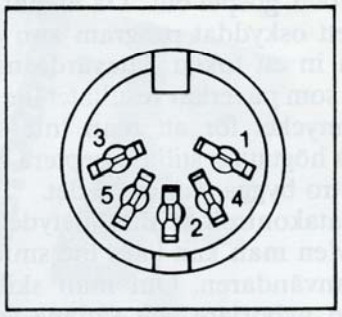
När du är nöjd med din bild är det bara att slå av datorn igen, stänga locket och skruva i de tre stjärnskruvarna och njuta av en skärm som överensstämmer med de krav som du har på en datorskärm.

För dig som tänker inhandla en monitor kanske det är intressant att veta vad det är för signal som finns i video kontakten. Signalen är av en standardtyp vad det gäller videosignaler som kallas "composite video" vidare innebär att sync, färg och bildinformation finns i en gemensam ledare.

I fig 1 kan du hitta information om de signaler som man kan hitta i denna kontakt. Till en monitor behövs bara två av dessa plus nr 3, Ljudsignalen om din monitor har denna finess. Nr 2, Jord, används alltid och skall vara kopplad till strumpan på den kabel som är kopplad till monitorn. Själva videosignalen kan man plocka ut från ETT AV stiften 4 eller 5. Handic serviceavdelning rekommenderar att stift 4 används men i egna experiment har det visat sig att stift 5 har fungerat lika bra.

Figur 1.

LJUD/VIDEO



PIN #	TYP	ANM.
1	+6V	10mA MAX
2	GND	
3	LJUD	
4	VIDEO LÅG	
5	VIDEO HÖG	

Testa vad som passar bäst för din monitor. Videosignalen har ett topp till topp värde på 0.5 volt, men kan justeras något med den vänstra potentiometern enligt ovan. Till slut vill jag påminna om en punkt på handics garantisedel "GARANTIN GÄLLER EJ om apparaten utsatts för ändring, ingrepp eller reparation utförd av annan än garantilämnaren."

Har du garanti kvar så låt alltså bli att göra det som beskrivs här, i detta fall tycker jag att du skall prata med den affär där du köpte din dator och höra om dom kan hjälpa dig.

Åke Hedman

SPELREKORD!

Lars Dahlin, Östersund, har fått 46900 poäng på VIC 1907 Superlander! Grattis! Låt oss få in fler spelrekord!

Använd gärna denna kupong för rapporteringen:

Namn: .....

Adress: .....

Postnr, postadress: .....

Spel: .....

Uppnådda poäng: .....

Intygas av: .....

(vittnet bör vara myndig person, alltså över 18 år!)

Kupongen sänds in till VIC-rapport, Spelredaktionen, Gustavsbergsgatan 8, 431 37 Mölndal.

Användarvänlighet

Din dator kan vara användarvänlig på flera sätt. Den kan t.ex. som VIC ha ett stort och lättarbetat tangentbord och ge stora och lättlästa tecken på skärmen. Men utöver dessa goda förutsättningar har du själv en stor roll att göra dina program sådana att datorn framstår som vänlig mot användaren.

Det fanns t.ex. en gång en FORTRAN-kompilator som skrev ut 'SUCCESS!' när man lyckades, det kändes härligt. Man vänlighet är mer än applåder, användaren mår bra av lite hjälp också.

Det är på tre områden som programmerarens omtanke särskilt behövs:

In- och utdatahanteringen, dvs meny och fråge/svarshantering samt presentation av resultatet.

Struktur, läsbarhet och kommentering i programmet.

Rimlighetskontroll av indata, utdata och viktiga mellanresultat inuti programmet.

När det gäller indata skall du låta datorn göra jobbet att översätta vanligt språk och den skall också presentera vilka svar den accepterar. Här finns alla varianter av klurigheter i bruk, men jag tror att man bör leva efter valspråket 'too clever is dumb'. Om du kräver att få strängarna 'ja' eller 'nej' och repeterar frågan när du får annat skräp så är du ganska säker på att den som kör programmet vet vad han vill ha gjort. Ja, det är klart att man kan bli otålig på en dum dator, man vill skriva 'j' eller 'n' eller helst bara nicka eller skaka på huvudet. Jag tror ändå att konversationen mår bra av fasta regler. Men låt oss se vad det är man ska svara på?

Det är vänligt och omtänksamt att ha definierat ofarliga, normala siffervärden i programmet, visa dem på skärmen och fråga om de passar. Om de inte stämmer med problemet tackar man 'nej' och kommer då till en inputrutin som börjar med att visa gränserna för tillåtna siffervärden. När man har knappat in sin data så gör programmet lämpligen en liten test att de inte faller utanför det tillåtna intervallet. Tycker du det verkar ofint, eller? – det är bara så att om man verkligen vill lita på datorn är det bäst att göra den lite misstänksam. Det finns tyvärr otaliga exempel på saker som gått åt skogen på grund av indata som varit så tokiga att man inte föreställt sig vilken inverkan de skulle kunna ha, långt mindre skyddat sig. Du har väl hört om GIGO-principen 'garbage in = garbage out'? Ännu värre säger erfaret folk och ryser, är det med 'garbage in' och 'gospel out'. Då menar de att i ett oskyddat program kan det slinka in ett tokigt datavärde nånstans, som påverkar resultatet lite lagom mycket för att man inte ska hoppa högt utan stilla acceptera och i god tro bygga vidare på det.

Indatakontroll är alltså betydelsefull. Men man kan vara lite smidig mot användaren. Om man skiljer mellan nybörjare och sådana som anser sig vara experter genom en fråga, så kan man i fortsättningen låta experten slippa en del förmaningar, använda förkortningar och mata in flera data åt gången, ja t.o.m. gå utanför begränsningarna. Men testerna bör man inte hoppa över, utan i det sista fallet lägger man i så fall in utskrift av ett meddelande att programmet går vidare 'under protest'. Skriv t.ex. 'gränsvärde överskridet x=...', 'outside limit x=...' eller något i den stilen på skärmen.

Efter programstart eler input av data kan det hända att resultatet dröjer. Då bör man av ett modernt begåvat program få kvittensen 'vänta lite' eller 'ta en paus medan jag räknar'. Det är mycket trevligare än att bli sittande framför en tom skärm och undra vad som hänt.

När resultatet kommer beror det förstås väldigt mycket på vad slags databehandling det gäller vilka vänliga åtgärder man ska tänka sig. Om du programmerar ett spel så är presentationen antagligen en del av spelets ide och vi diskuterar inte den här. Men en trevlig vinkel på ett spel är att det tar in namnet på den som spelar och kommer ihåg allas resultat till ett protokoll som dateras upp och visas efter varje körning.

Ordbehandling är också ett speciellt område där användarvänlighet nog är ungefär detsamma som att programmet helt enkelt är lättarbetat.

Om det är tekniska eller ekonomiska beräkningar eller datasorteringar du programmerar så tänk på att samla ihop alla data som använts och visa dem eller åtminstone en översikt av något slag så att man kan gå tillbaka och kontrollera. Sidodata som datum och benämningar är viktigt i det sammanhanget. Beräknade värden skall ställas upp i tabeller eller visas i grafik så att användaren får en god överblick.

När du som användare av t.ex. ett Basicprogram finner att du skulle vilja ändra lite på det så kommer du i kontakt med den 'inre användarvänligheten'. Det är bara om programmet är uppdelat i lättförståeliga delar och har förklaringar till variabelnamnen som du har en god chans att lyckas med att rota i ett program som någon annan gjort från början. Lägg därför själv ned omtanke på detta och tänk särskilt på två saker:

1) Skilj mellan väsentliga, permanenta variabelnamn och s.k. slaskvariabler. Med det senare menar jag variabelnamn som du tillfälligt använder t.ex. varje gång du beräknar en summa med

```
S=0
FOR J=1 TO N
S=S + X(J)
NEXT J
```

Efteråt räddar du summan till en permanent variabel med t.ex. 'SU=S', och sedan kan S användas igen i någon annan betydelse. I Basic där variabelnamnen gäller genom hela programmet bör man egentligen behandla alla variabler man träffar på genom 'GOSUB' som slaskvariabler och skriva 'anhoppet' på subrutinen i den här stilen:

```
4711 X=X1:Y=Y1:S=0:GOSUB 5011:SU=S
4712 PRINT"SUMMAN AV X1 OCH Y1";SU eller
4712 REM SUMMAN AV X1 OCH Y1 = SU
5011 S=X+Y:RETURN
```

På det sättet får du en struktur som hjälper dig att läsa och förstå programmen flera veckor efter att du gjort dem. (Det här exemplet är naturligtvis en alldeles för enkel subrutin.)

2) Lämna aldrig kvar kommentarer från en gammal programversion utan att kontrollera att de fortfarande är meningsfulla och riktiga. Det är så frestande att tro på kommentarerna men datorn struntar som bekant i dem. Vi kanske får Artificiell Intelligens i framtiden som gör självkommenterande program, men än så länge är den bästa vägen att välja variabelnamn så förklarande som möjligt och skriva kommentarer inte om vad man gör utan om varför!

Rimlighetskontroll slutligen har jag nämnt i inputsammanhang. Basic-tolken i VIC-20 t.ex. är ju barsk men vänlig när den stoppar felskrivna kommando. Programmet kan bli vänligare = säkrare om du går igenom dina variabler, ser vilka talvärden de kan variera mellan och lägger in test som slår larm för orimliga värden. Och, väl värt att observera, det är ofta i sådana handgripliga formella studier av programmen som logiska galenskaper uppdagas; gör det tidigt och bli vänlig användare!

Frenne Söderberg

"Bättre service åt både konsument och återförsäljare"



säger Charlotte Bille-Edholm, ansvarig för det nyaste datatronic-bolaget handic POST ab.

handic POST ab är namnet på ett nytt, helägt dotterbolag till datatronic. Det kommer att per postorder sälja och leverera program, utrustningsdetaljer och litteratur till datorerna VIC-20 och VIC-64. Företaget är det första datatronic-företaget med kvinnlig chef - den första i en tidigare mansdominerad koncern.

"Från handic POST kommer vi att kunna leverera ALLA program och tillbehör till VICdatorerna - oberoende av om de tagits fram inom landet eller utomlands."

"Vi kommer att göra Sveriges mest kompletta katalog över program för spel eller nytta och för utbildning eller övningsändamål. Dessutom kommer vår katalog att med utförliga beskrivningar - både av program och tillbehör - göra det lätt för konsumenten/VIC-ägaren att välja i den otroligt rika programfloran!"

Charlotte fortsätter: "En annan stor fördel med handic POST-katalogen är att återförsäljarna kan använda den (och handic POST) som en utökning av sina egna lager! Man kan låta sina kunder välja i katalogen hemma, i lugn och ro. När de sedan kommer till butiken vet de oftast exakt vad de vill ha. Antingen plockar handlaren det från hyllan eller så beställer han från handic POST för omedelbar leverans per post till kundens hemadress. Själva betalningen kan klaras av i butiken, per postförskott eller med QVIC-konto. Kunden behöver inte vänta medan säljaren kanske är upptagen med demonstrationer för andra kunder när han/hon redan vet vilket program man vill ha."

Men det viktigaste nämner Charlotte sist: "Genom att göra en katalog över allt som finns till VIC-datorerna kan programmerarna få en överblick över marknaden, se vad som saknas - eller vad de kan förbättra. handic POST-katalogen blir därför en mötesplats för programköpare och programskrivare, för utrustningsköpare och tillverkare, för elever och läroboksförfattare."

Vi frågar om inte återförsäljarna misstänker att handic POST kommer att ta ifrån dem försäljningsvolym, och då svarar Charlotte: "Nej, inte allts! handic POST är tänkt att fungera så att återförsäljarna skall kunna ÖKA sin försäljning genom att kunna erbjuda ALLT, som finns i katalogen. Inte bara de produkter man lagt upp ett lager på. handic POST-katalogen fungerar som en utökning av deras lager - utan att de behöver binda pengar i lagervaror."

"POST-katalogen kommer att samordnas med QVIC-konto så att kunderna kan välja hur de vill betala, kontant i butiken, mot postförskott eller på QVIC-konto. Detta är en annan av våra idéer, som vi tror kommer att göra tilvaron lättare för både återförsäljare och kunder/VIC-ägare. Den nya katalogen ligger klar att distribueras före årskiftet", lovar Charlotte när vi avslutar vårt samtal.

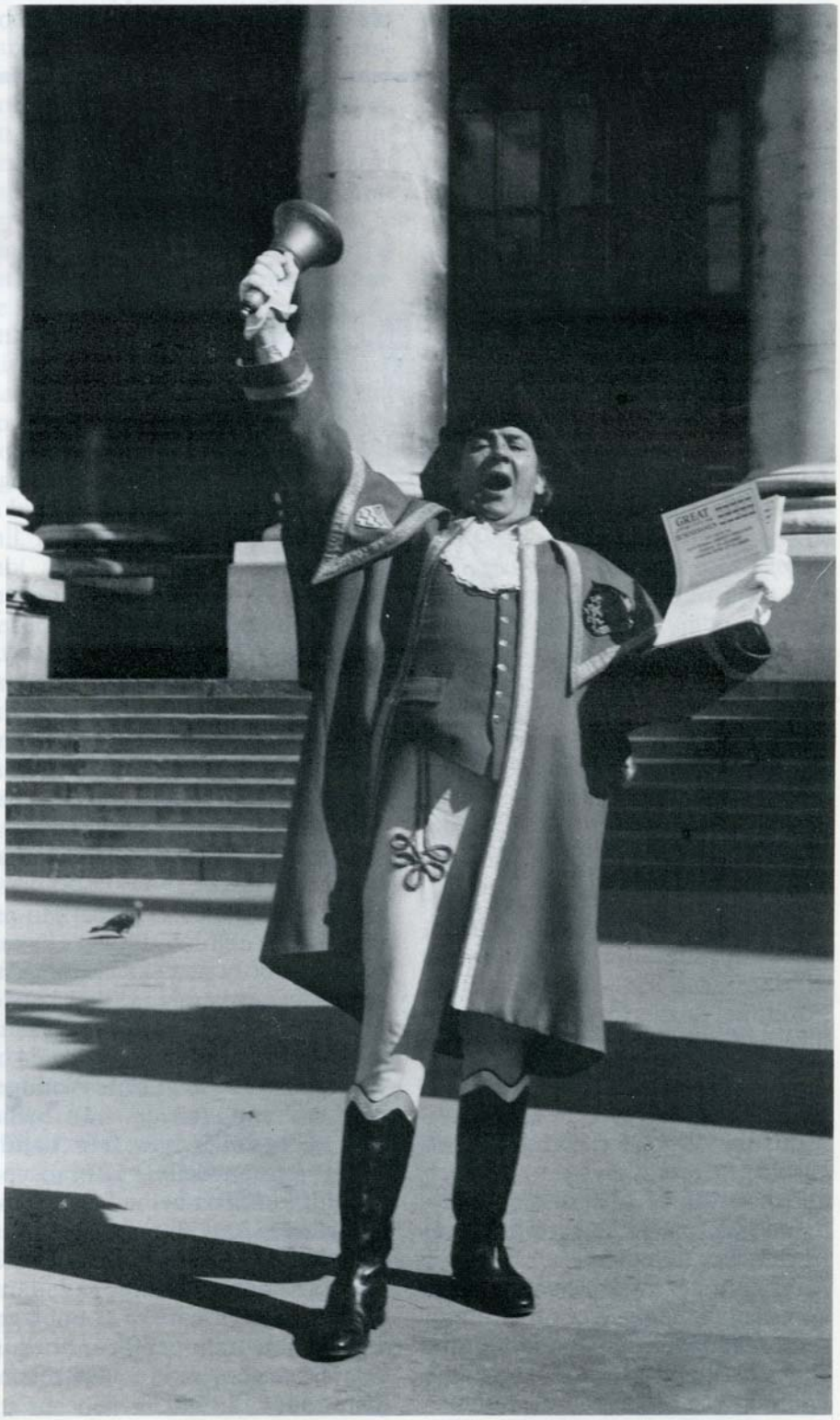
Europas största utställning i år?

PCW-mässan i London med sina hundratals utställare var kanske den största datormässan hittills. I bästa engelska stil utropades den av en traditionellt klädd "Town Cryer" när den avhölls under tiden 29 september till 2 oktober.

Vi hade rest över till London ett gäng unga och gamla datorintresserade "gossar" för att under en vecka riktigt få vårt lystmäte av datorprylar och program tillfredsställt. Det var VIC-klubben i Göteborg, som även om där fanns förespråkare för andra datormärken också, i huvudsak bestod av VIC-entusiaster. Praktiskt taget hela klubben plus nyvärvade medlemmen ÅF hade satsat en semestervecka för att "göra London". Det var flera av resenärerna som varit i London förut och de fungerade som utomordentliga guider för oss andra.

Första dagen vi var där, måndagen, då vi tog oss ner till något man kallade för "elektronik-gatan" därför att där låg så många affärer som sålde elektronik och datorer. För den, som är van vid det smala utbudet av böcker och datorprylar samt program här i Sverige, var det ju rena julafton och pengarna smalt snabbt ihop när man handlade. (Hade inte kreditkorten funnits med vete fåglarna hur det hade gått!)

Tisdagen blev en sorts återhämtningsdag, då i varje fall jag tog det mycket lugnt. Några av "grabbarna" skulle naturligtvis ut och handla eller titta då också, men andra passade på att titta på Londons sevärdheter. Vi sparade oss till onsdagen, då mässan öppnade med Pressens Dag. Eftersom vi hade två tidningsredaktörer i VIC-klubben, dels Bosse Svensson från Allt om Hemdatorer och dels ÅF med VIC-rapport fixade vi naturligtvis presskort för alla VIC-klubbsmedlemmarna!



Och det var nog tur det, för maken till utställning att vara tättbefolkad har jag inte sett sedan jag sist besökte Dagens Hushåll! T.o.m. på pressdagen var det trångt! Och det var åtskilligt att titta på och pröva. Mängder med nya spel och nya applikationsprogram för smådatorer inom affärsvärlden fanns att se. Synd att så många är svåra att anpassa till svenska förhållanden och lagar. På annan plats här i VIC-rapport skall vi försöka presentera nyheterna som finns och som kan väntas hit så småningom.

Ett annat tydligt märkbart förhållande var den stora variationsrikedomen på smådatorer som bjöds ut. Där fanns säkerligen mer än tjoget olika märken, av vilka endast ett fåtal någonsin kommer att nå Sverige Och här får de stark konkurrens av de även i England alltmer populära VIC-20 och 64:orna!

När man trött i futterna och huvud äntligen kom tillbaka till hotellet och fått en matbit och satte sig för att titta på TV en stund kom ett specialprogram från utställningen! Där fick vi träffa två unga män som redan hade stort anseende som programmerare och höra deras synpunkter på framtiden. Den ene var ganska kritisk till ett alltför oförbehållsamt accepterande av datoriseringen, den andre såg inga som helst risker utan tyckte det var bara att fortsätta. Båda var dock ense om att man behöver en ökad kunskap om vad datorer kan och inte kan utföra hos den stora allmänheten, som i dag alltför ofta skräms av okunniga och kanske även negativt inställda journalister och nyhetsförmedlare.

Under programmet berättades också att Central TV (man har ju flera bolag som levererar TV i England!) skulle sätta igång en speciell utbildningsserie som riktade sig till föräldrar och barn och täcka in alla "hemdator"-tillämpningar.

En speciell nyhet från mässan var 3-dimensionella spel, som presenterades av Postern och som krävde speciella 3-D-glasögon. Tyvärr har jag själv ingen glädje av sådana spel med bara ett helt perfekt öga, men jag förstår (och minns de gamla s.k. anaglyf-bilderna som var så populära förr) att den tredje dimensionen är fascinerande i en spelsituation. Vid kontakt med Postern sade man visserligen att just detta program inte fungerade lika bra på VIC:arna som på andra märken, men någon kommer säkert fram med en variant som drar nytta av 64:ans grafik i högupplösning och dess goda färgmöjligheter.

I Commodore's monter visades inte bara VIC-maskiner utan även PET-anläggningar av olika slag liksom man körde mängder av program från VIC Software, det postorderföretag, som Commodore England startat och som förser marknaden med både program och utrustningsdetaljer från centrallagret i Slough, en bit utanför London. Vi lär ju få något liknande i Sverige nu, när Datatronic sätter igång sin egen postorder-avdelning.

Man hade flera nya program för VIC:arna, både spelprogram och nyttoprogram. Vilka som kommer att säljas av Handic i försvenskat skick är väl inte helt klart ännu, men de blir i varje fall presenterade här i VIC-rapports spalter när man bestämt sig och fått programmen klara för distribution.

Bland de intressantare nyheterna fanns en speciell väggkontakt som tog hand om s.k. transienter eller strömstötar på nätet. Jag hade önskat att vi hade en sådan här i går under höststormen, när strömmen föll bort inte mindre än fyra gånger medan vi höll på att skriva in text till tidningen på vår egen redaktionsdator! Man skall undersöka om det går att göra en motsvarande kontakt med svensk standard - vi kommer att hålla våra läsare underrättade!

En annan nyhet var ett förnämligt diskettförvaringssystem med hängmappsliknande fickor som enkelt kunde låsas fast så att man inte riskerade att tappa fickorna när man lyfter ur disketten ur sin ficka. Man har ett genomtänkt system med registreringskort och varje låda rymmer 80 disketter och är låsbar.

Besök på Commodore i Slough

En av dagarna for jag ut till Commodore, en halvtimmes tågresa från Paddington-stationen ute på den engelska landsbygden. Commodore har en helt nybyggd anläggning med plats för alla de centrala funktionerna: lager och expeditioner, testsalar och exportavdelning - postorderföretag och allmänt kontor. Min ciceron berättade att man hade en mycket intensiv verksamhet med testning av program, som levererades från de olika programvaruhusen och programmerarna ute i landet, och som man ville ha upptagna i katalogerna under rubriken "Approved Products" dvs godkända för användning på Comodore-datorer. Varje månad kommer 100-120 olika program och man brukar hinna med 50-60 per månad! I två stora salar fullspäckade med datorer sitter folk hela dagarna och testar programmen men man hade just bestämt sig för att ytterligare utöka denna verksamhet för att kunna "beta av" den ständigt växande eftersläpningen (den är nu uppe i cirka 8 veckor!). Commodore köper också många program för egen distribution och man har en kopieringsavdelning där ett tjugotal diskstationer används för i huvudsak kopiering av demoprogram till återförsäljare.

Av nyheter såg jag bara den portabla 64:an, som döpts till SX-10 - det är den som kallats VIP tidigare - och som har en enkel tunn diskdrive. Den var under utprovning och såg riktigt nätt ut med sin lilla färgbildskärm bredvid diskdriven. Man har ännu inte bestämt om den kommer att tillverkas med dubbla diskdrivar, men i så fall kommer den att kallas DX-10. Min ciceron berättade vidare att det egna postorderföretaget, ComSoft, säljer program och datorer med tillbehör per postorder till orter och kunder som inte har återförsäljare i närheten och därmed lyckas uppnå en än större marknadstäckning. En stor fördel för ComSoft är det register över VIC- och PET-kunder som man får i och med att garantikorten sänds in efter datorköpet. Ytterligare fördelar är att man lyckats etablera samarbete med återförsäljarnätet och därigenom förbättrat den service som VIC- och PET-kunderna får.

forts. nästa sida

Nyheter utifrån

Föreningarna på PCW-mässan

En del av mässan var helt ägnad de olika datorföreningarna, som är så rikligt förekommande i England. Varje förening eller snarare deras centrala organisation hade en monter, där man visade program, sålde tidningar och överhuvudtaget informerade om föreningsverksamheten. Själv hälsade jag på i ICPUG-montern, alltså hos Independent Commodore Products Users Group. Jack Cohen välkomnade mig varmt som en av de få svenska föreningsmedlemmarna och visade vad hans son åstadkommit med lite komponenter och en lödkolv. Det visade sig vara en plug-in cartridge till VIC-64 med namnet PICTOGRAPH och som underlättade grafikanvändning på 64:an. Med en joystick kunde man rita figurer och fylla i dem, flytta dem, spegelvända dem och förstora dem. Jag har lånat ut mitt exemplar och vi återkommer med en riktig testrapport framöver här i VIC-rapport. I en annan monter hittade jag en ljuspenna till överkomligt pris. Den är också utlämnad på testning och även här återkommer vi när testaren är klar med sin rapport.

Å. F-n.



OBSERVERA!

Alla program och nyheter, som presenteras i VIC-rapport, finns inte att få i Sverige. Ibland får vi nyheter utifrån av program, som aldrig kommer, ibland kan det dröja bara. Hör med våra annonsörer eller – ännu bättre, kontrollera i handic POST-katalogen.

Här skall vi titta på det material som jag fick med mig från den stora datormässan i London, Det är i de flesta fall program och tillsatser eller böcker, som inte finns i Sverige ännu, men genomgången avser visa det breda utbud av VIC-inriktat material som finns.

COMPUPRINT i England har en serie bokföringsprogram för PET, VIC-64 samt förenklade versioner för VIC-20. Programmen är kanske inte så attraktiva i Sverige, då de dels ligger ganska högt i pris (omräknat i SEK mellan 3 och 6 tusen kr/st) och dels är anpassade för engelska förhållanden.

En annan intressant produkt är en speciell ställning, som är avsedd för bl.a. VIC-arna. Den är utförd i rökfärgat akrylglas och avsedd att ställas på ett bord. Den ger utrymmen för dator, skrivminne och Datasett samt har plats för monitor eller TV högst upp. Sladdar och sådant göms bra och man får en prydlig arbetsplats. Kanske någon svensk tillverkare gör något liknande?

COMPUTER CHOICE är en ny datortidning som skall komma ut under november i England. Man säger sig vara den "första och enda nybörjarguiden" för datorer under 2500 kronor i inköpspris. Prenumeration kostar ca 120 kr/år.

BUBBLE BUS heter ett engelskt programföretag, som lanserade nya program för 64:an. Bl.a. KICK-OFF (ett nytt fotbollsspel, som vi också får ett nytt från Handic under namnet CUP FINAL), EXTERMINATOR (ett arkad-spel, där man skall ta kål på maskar och annat i en trädgård utan att själv bli offer för en farlig spindel och/eller en slående örn). HUSTLER är ett pool-spel med sex olika spel för en eller två spelare.

"Vi tror att många hemdatorägare börjar bli lite trötta på de många enklare spelen och ser sig omkring efter mer praktiska om än enkla program för utbildning eller inriktade på affärstillämpningar", säger chefen för ett av de större programvaruhusen, MARKETING MICRO SOFTWARE LTD. Därför har man nu lanserat INVENTORY 64, ett lagerbokföringssystem på disk för 64:an. Man får plats med 650 artiklar med tillhörande uppgifter som tex. lagerplats, säljare, enhetsstorlek, orderpunkter, minimim-orde och kostnad för lagerhållet material. SPINTYPER är ett annat program, som avser att lära ut maskinskrivning. Det finns på kassett för VIC-20. TINY TUTOR heter ett läroprogram i räkning, som är avsett för barn mellan 2 och 7 år(!) som har en VIC-20. Bland nya spel från Marketing Micro Software kan nämnas ZEPPELIN RESCUE för VIC-64 och CONQUEROR för VIC-20.

QUICKSILVA är namnet på ett distributionsföretag för spel, som tidigare mest ägnat sig åt program för

ZX81 och BBC-datorer, men nu går in för att ta fram spel till de även i England allt populärare VIC:arna. I den senaste katalogen har man ett program för en oexpanderad VIC-20 med namnet TORNADO, ett annat program som heter TRADER, där man spelar en sorts intergalaktisk dörrknackare på sin VIC20+16K. Det med Quicksilva samarbetande företaget PIXEL ger ut STAR-QUEST & ENCOUNTER, SUB-SPACE STRIKER, HARVESTER & BRAINSTORM samt ett som fått namnet PIXEL POWER och ger arbetsverktyg åt den, som vill göra fin grafik på sin VIC-20 med 8K eller större utbyggnad av minnet. Direkt i posten kom ytterligare några spel för 64:an PURPLE TURTLES, AQUQ-PLANE, RING OF POWER samt QUINTIC WARRIOR.

VIRGIN GAMES är ett ungt och nytt programföretag med unga programmerare. Man har ett par nya program för VIC-20: ENVAHI och CREEPERS samt ett med namnet FALCON PATROL för 64:an. Under vintern 83 skall ytterligare spel komma enligt deras katalog, bl.a. MISSION MERCURY för VIC-20 och BITMANIA för 4:an.

AUDIOGENIC är ett av de mer kända programföretagen i England. Man har introducerat ett nytt informationshanteringsprogram med namnet MAGPIE (jag tror det betyder "skata"). Dessutom har man ett ordbehandlingsprogram med namnet WORDCRAFT 40. (Från Handic finns ju redan ute TEXT 64 och CALC RESULT EASY och CALC RESULT ADVANCED för 64:an, så det är föga troligt att dessa engelska program importeras. Det kan ju vara roligt att veta att man har mängder av program ute i världen, eller hur?

Från PRECISION SOFTWARE kommer SUPERBASE 64 - ett nytt styrsystem för 64:an, som låter dig skapa egna format och records, ändra layout och datafält.

COMMODORE SOFTWARE har också kommit med lite nya program. En del av dessa tas hem till Sverige och modifieras till svenska förhållanden av Handic Software för distribution genom Handic.

HIGH FLYER - ett simuleringsprogram där man leder ett flygföretag och som går att spara mellanresultat i. Man kan alltså avsluta spelet tillfälligt och ta upp det från det läge man slutade när man vill fortsätta!

FUTURE FINANCE - endast i England ännu och är en motsvarighet till CALC RESULT, som vi har här.

PETSPEED BASIC COMPILER finns nu också till 64:an. EASY SCRIPT har vi som TEXT 64. GOR-TEK-programmet finns också i handeln i Sverige.

En stor nyhet är att nu äntligen finns en fungerande COMAL-disk till 64:an! Från Commodore Danmark fick vi häromdagen den första riktiga COMAL-en till vår 64:a och kastade oss över att prova den. Versionen heter 00.12 och är speciellt anpassad till VIC-64 i väntan på den plug-in cartridge, som skall komma på nyåret. Man kan räkna med att disketter (jo, man får två för att ha en i reserv, de går nämligen inte att kopiera!) och handbok kommer att kosta runt 500 kr, men då har man möjlighet att utan extra kostnad få byta en skadad diskett av de två, märk väl bara om man registrerats som köpare av disketterna, vilket görs genom att man sänder in ett garantikort till säljaren. COMAL-versionen har fullt utbyggda grafiska kommandon och en enkel rutin för att starta upp även musiken. Det musikprogram som låg på disken var helt förtjuuuuusande och lät ibland som piano och ibland som spinett. Faktiskt riktigt snyggt! Det enda som egentligen saknas på disketten är den utbyggda strånghanteringen, som kommer att finnas på cartridgen när den kommer. Översättning av handboken från danska pågår som bäst.

Boknyheter

SYBEX BOOKS har två titlar: Advanced 6502 och 6502 Applications Book. Man har dessutom fem böcker för 64:an: Commodore 64 - Computing, -Games Book, - Programmers Reference Guide, - Working Commodore och - Elementary 64. För VIC-20 har man ännu fler böcker, men den maskinen har ju funnits längre. GRANADA heter ett förlag som har två titlar för VIC, VIC-20 Games Book och Commodore 64 Games Book. INTERFACE har Mastering Commodore 64 och SUNSHINE säljer Commodore 64 Adventures.

En lista vi fick från Computer Bookshops Ltd i Birmingham angav dessutom följande program till VIC 64 och VIC 20: VIC-20-program: Cyclons, Martian Raider, Moons of Jupiter, Pakacuda. VIC-64-program: Centropods, Cyclons, Database, Dickys Diamonds, Escape MCP, Fools Gold, Pakacuda, Stellar Triumph, Tombs Xeiope och Zappy Zooks. Vi får väl se, om någon tar hem dem till Sverige...?

HOFACKER heter ett förlag, som har en hel del för VIC-arna också. Man har ett ordbehandlingsprogram som heter Bliztext, till vilket finns även maskinvarutillsatser som t.ex. interface för en parallell printer, en sats för direkt modem, ett universellt experimentkort och ett expansionskort med plats för fyra experimentkort. Bland övriga program kan nämnas Macrofire, en editor/assembler för 64:an. I bokväg har man t.ex. "How to program in 6502 Machine Language on your 64, Commodore 64 Tune-up, Small Business Programs for the 64, The Great Book of Games, More on the 64, Tricks for VICs. Man har också de flesta programmen som finns i böckerna på kassett eller disk om man vill betala för att slippa knappa in långa listningar.

DUCKWORTH är namnet på ett annat förlag, som har böcker till 64:an, bl.a. en titel "Using the Commodore 64" och en annan "Exploring Adventure Games on the Commodore 64", Commodore 64 Games och A Pocket Handbook for the Commodore 64 är två andra böcker inom vårt intresseområde. "Will you still love me when I am 64" är en komplett guide till musik på 64:an, Commodore 64 Graphics behandlar grafiken. Dessutom finns ju VIC-20-böcker: Exploring Adventure Games on the VIC, A Pocket Handbook for the VIC, VIC Games, Twelve Simple Electronic Projects for the VIC. Man har många fler böcker på gång, men mer därom när vi sett dem.

Å. F-n.

Vics tidning är bedrövlig

■■ Vic är Sveriges mest köpta hemdator. Folkdatorn kallas den i annonserna.

— Fast egentligen säljer vi inte en dator utan en idé, att lära svenska folket förstå datorer, säger Leif Dahlberg, marknadschef för Vic hos generalagenten Handic.

Så säger han. Men i Handics egen tidning för alla Vicägare gör han och hans företag inte ett dugg för att förverkliga dessa idéer.

Tidningen är en bedrövlig produkt. För den som inte är dataproffs är den nästan obegriplig — facktermer och varubeteckningar på Handicprodukter i en kompakt textmassa sida upp och sida ned.

Vi som inte är proffs får ingen hjälp alls för att få mer nytta och nöje av vår Vic.

För att få hjälp måste vi i stället köpa Handics instruktionsböcker som säljs till oförskämt höga priser på mellan 80 och 250 kr.

I tidningen får vi inte ens klarhet i vilken tidning vi egentligen läser. På omslaget står det att den heter Vic-Nytt. Men inne i tidningen kallas den Vic-News.

Jag och 12-åringen där hemma läser med förtjusning de nya datortidningarna Allt om Hemdatorer, Dator Hobby och Min Hemdator. I dessa finns lättlästa artiklar och bra förslag till program, proffsigt redigerat med bra bilder och klatschiga rubriker.

Inget intressant

Men i senaste numret av Vic-Nytt (Vic-News?) hittar vi inget intressant.

Mycket av vad som står i den orkar vi inte försöka tränga oss in i. Annat förstår vi inte, även om vi försöker begripa.

Hur många nyblivna Vicägare kan till exempel förstå ett råd som det här:

"Vic 64:ans User Port kan användas som en interface till RS 232C eller V.24 om man lägger till en Vic 1011a RS232C nivåomvandlare."

Vi begriper det i varje fall inte. Och vi borde vara typiska Vicägare — en 12-åring som suttit vid datorn mest varje dag sedan vi köpte den i februari och en far som ibland också försöker sig på litet egna program.

Men vilka frågor och svar får vi i Victidningens frågespalt? Jo, svar av den här typen:

"Mikael Persson i Visby vill ha spelprogramlista och prisuppgifter. Vi har separat sänt broschyrmaterial. I övrigt hänvisar vi till våra annonsörer och till annonser i såväl dagspress som fackpress."

Ingen som helst information alltså — mer än för den person som frågat. Och han kan ju lika gärna få svar i ett vanligt brev.

"Vi ska bli bättre"

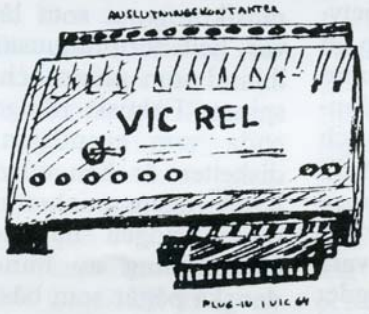
I tidningen finns också recension av olika spelprogram. Recensenten, som fått i uppdrag av Handic att testa Handics spel i Handics egen tidning, kommer till följande slutsats:

"Dessa åtta program var verkligen roliga och bra, de ställde stora krav utan att vara alltför besvärliga. Håll ögonen öppna för nya spelprogram från HANDIC SOFTWARE."

På sista sidan finns en sida med radannonser. Men tro inte att det är läsarna som skickat in. Nej, så här står det ovanför annonserna:

"Då avdelningen är helt nystartad, ger vi här några exempel, som vi lånat från andra svenska tidningar och hoppas att annonsörerna inte blir arga på oss för att de får ytterligare en gratisannons."

Men varför sviker då Handic alla de tusentals nyblivna Vicägare som skulle lära sig datorer? Varför får vi ingen information som vi begriper?



• I hela tidningen finns bara ett fåtal illustrationer. En av dem ser ut så här.

Det är just sådana som vi som Vic 20 främst är avsedd för. De riktiga proffsen nöjer sig inte med en så liten dator.

För oss dyker det ständigt upp frågor när vi sitter och försöker göra program. Så en frågespalt är verkligen viktig i en datortidning.

— Tidningen ska bli bättre. Jag lovar, säger Leif Dahlberg på Handic.

— Men vi måste också tillfredsställa ett informationsbehov hos alla de entusiaster som haft Vic länge. Det är dessa som skrivit det mesta i senaste numret.

Det märks. De är säkert duktiga på datorer. Men inte på att skriva begripligt.

- Anders Andersson

Är VIC:s tidning bedrövlig?

Det stod att läsa i Expressen den 9/10 i år och då vi på redaktionen inte tyckte detta var särskilt roligt att läsa bestämde vi oss för att höra vad våra läsare i större antal ansåg.

Här följer artikeln, något omklippt för att passa vårt sidformat, tillsammans med en uppriktig fråga från redaktionen:

Vad tycker DU, som läser VIC-rapport?

Vad anser du vi bör göra för att du skall få ännu större glädje och nytta av tidningen? Vi lottar ut ett dussin T-shirts med VIC-motiv bland dem som skickar in vårt frågeformulär.

Mina önskemål om VIC-rapport

Först vad du tycker om tidningen i dess nuvarande utförande: (gradera de olika punkterna med kryss för de betyg du vill ge i tabellerna.

Första kolumnens betyg gäller de olika artiklarnas innehåll, om du tycker det är bra eller ej. Andra kolumnen gäller HUR du tycker författarna skriver, om det är lätt eller svårt att förstå (svårt att förstå = 1, lättläst = 5).

Betyg
1 (dåligt)
2 (inte helt bra)
3 (acceptabelt)
4 (ganska bra)
5 (mycket bra)

GLÖM INTE FYLLA I NAMN OCH ADRESS!
Du KAN ju vinna!

	INNEHÅLLET DÅLIGT / BRA					TEXTEN SVÅR / LÄTT				
	1	2	3	4	5	1	2	3	4	5
Ledaren										
Nybörjarskolan										
Kurs i strukt.BASIC										
Assemblerskolan										
Comalkursen										
Programmeringsteknik (olika artiklar)										
Nyhetspresentationer										
Frågor och svar										
Kapten Storöra										
Annonser										
Småtips										

För att vi skall få någon ledning av dina svar ber vi dig tala om för oss hur länge du själv hållit på med datorer:

Nybörjare: (0–1 år)
Hållt på någon tid: mellan 1 och 2 år
Erfaren användare: mer än 2 år
(stryk det som inte gäller)

Här kan du skriva vad du vill att VIC-rapport skall innehålla mer av i framtiden. Skriv din egen "önskelista", den kommer vi att studera mycket noga! Ange också gärna med siffror ordningsföljden på de olika ämnena, om du skriver fler än ett, så att det för dig viktigaste får nr 1 och sedan i ordning.

Namn: .....
Adress: .....
Postadress: .....

Kupongen sänds till VICrapports redaktion, Gustavsbergsgatan 8, 431 37 Mölndal, senast den 10 januari 1984. Dragning av vinnare av VIC-tröjor sker den 15 januari och vinnarna meddelas direkt per post.



Vad har man egentligen för nytta av en hemdator?

Detta är en artikel främst för dig som är pappa eller mamma till en hacker. Vad kan du själv göra med hemdatorn när du lyckats slita din unga datafantast ifrån familjens VIC? Säkert har du många gånger frågat dig: Vad har man egentligen för nytta av en hemdator?

Expressen-medarbetaren Anders Andersson sände oss den här artikeln. som vi gärna tar in. Den tar ju upp en väsentlig fråga för alla föräldrar i denna datorernas förlovade tid.

Det verkar som om många dataföretag tidigare stått handfallna inför frågan:

– Den är bra att lägga in matrecept i, har någon svarat.

Som om någon vill ladda in ett program och leta i datorn när det är middagsdags – istället för att slå i kokboken.

En del säger att man ska lägga in sina grammofonskivor eller möblera om vardagsrummet genom att flytta små teckningar av bord och soffor på skärmen.

Det där tror i varje fall inte jag på. Jag tror att det största användningsområdet inom några år är hemdatorn i kombination med telefonen.

Du ringer upp postorderföretaget, kollar på skärmen vad som finns och beställer genom att knappa in på din terminal. På samma sätt kan du beställa teaterbiljetter, beställa radannonser, hyra specialprogram för att plocka in egna siffror i.

Experiment kring sådant här har redan satt igång.

I väntan på att detta startar på allvar är det viktigt att så många som möjligt lär sig grunderna för hur en dator fungerar.

Det kan vara nytta nog för en vuxen som inte själv är en hacker och ständigt vill konstruera olika spel.

För båtklubben

Den som har en liten firma eller jobbar i någon förening kan ha god användning för olika program för bokföring, ordbehandling och registrering. En båtklubb kan till exempel lägga in sina medlemmar i ett registerprogram. Det går att få datorn

att hålla reda på vilka som betalat medlemsavgiften, när det är dags för sjösättning för olika sorters båtar, vilka som lagt ut pengar för klubben etc, etc.

Fast för sådant räcker inte en VIC-20. Och till den VIC-64 som man måste ha krävs också en floppydisk.

Men i en vanlig familj, vad har man där för användning av en hemdator?

Det verkar som om datorföretagen nu äntligen har börjat fundera litet över den frågan. Det finns nu program att köpa för t.ex. hembokföring, språkstudier, skatteberäkning och lottospel.

VIC-rapport har också börjat tänka på andra än datafantasterna. I det här numret finns ett program för din egen hembudget. Det är bara att knappa in det på datorn.

Travprogram

Men det finns mycket annat som datorn kan hjälpa dig med. Fundera över saker som tar tid att räkna ut för hand. Tänk efter om inte datorn kan hjälpa dig att räkna.

Busschaufförен Gustav Antal i Stockholm fann t.ex. att det inte fanns något program som jämfört travhästar med varandra. Så nu håller han på att konstruera ett som snart är ute till försäljning.

VICs egen marknadschef Leif Dahlberg tyckte det var svårt att räkna ut för hand hur mycket färg som skulle gå åt när han skulle måla om huset. Han gjorde ett enkelt program för att räkna ut det och kunde se hur många burkar det gick åt.

Deklarationsprogram

Själv har jag de senaste åren skrivit om skatter och privatekonomi på Expressens sidor "dina pengar". Då har jag sett hur svårt det är att räkna ut skatten. Den saken har jag nu lärt min VIC 20.

Att räkna ut skatten för hand blir bara värre och värre. I februari nästa år, när vi för första gången ska deklarera efter nya skattereformen, kommer halva svenska folket att sitta där hjälplösa. I Expressen ska vi naturligtvis lära ut så gott det går hur man räknar.

Men det är inte lätt. Först består statliga skatten av ett grundbelopp och sedan ett tilläggsbelopp, som ibland reduceras och ibland inte.

Och det är inte lättare nu vid slutet av året när det är dags att skatteplanera, för att se om det lönar sig att betala in ränta i förskott eller litet extra på en pensionsförsäkring.

Jag hade just gjort mina skatteplaneringsprogram klara och fått datorn att tala om vad avdragen kostar efter skatt. Då plötsligt sa regeringen att skatteskalan för 1984 ska ändras.

Jag fick börja om på nytt. Men nu när det äntligen blivit klart ger datorn sekundsnabba svar.

Det är just för att göra beräkningar som tar tid att göra för hand som hemdatorn är till nytta. Fundera steg för steg hur det går till att räkna för hand. lägg in variabler för det datorn måste veta och för det du vill få fram, lägg in de matematiska formlerna i programmet och lär av de fel du gör.

Man behöver inte vara hacker för att klara det. Det är lättare än man först tror - om man bara har tålamod.

Anders Andersson, Expressen

Lägg på en rem!

När man programmerar i BASIC är det viktigt att lägga in ordentligt med REM-satser, så att andra kan se hur programmet fungerar när de listar det. Detta har väl i alla fall de som läser data i skolan fått veta. Däremot är det inte lika väl känt, att man kan försvåra listningar ganska avsevärt med lämpliga REM-satser! I julinumret av Compute!, fanns en artikel som demonstrerade en del egenheter med Commodore-basicens REM-kommando. Om du (för åskådlighetens skull i gemena = små bokstäver, lower case) slår in följande rader:

```
10 rem print
20 rem PRINT
```

och sedan LISTar dessa rader, får du resultatet:

```
10 rem print
20 rem input#dimright$nextlet
```

Detta var ju inte precis vad vi skrev in. Tecknen +, -, *, a, L och alla bokstäver utom L ger, skiftade i REM-satser, BASIC-kommandon när man listar! Ett skiftat L *) gör att datorn skriver ut: "?syntax error" och avslutar listningen.

Om man skriver reverserade tecken i REM-satserna återges de vid listningar normalt, dvs reverseringen försvinner. Om du sedan skriver REM-satser med "fnuttar" dvs citationstecken, blir det ännu underligare. Både oskiftade och reverserade tecken återges som man skrev in dem, t.o.m. markörkommandon m.m. återges som de skrevs in. Men, vissa reverserade tecken fungerar som styrkommandon för utskrift: /rev/ /skift/t motsvarar "insert" t.ex.

```
10 rem""/del/ /rev/tttt/off/print a$
```

ser ut så här när du har slagit in dem på skärmen:

```
10 rem "ttttprint a$, förutom att t:na
```

är reverserade. När du LISTar kommer det däremot att se ut så här:

```
10 print a$
```

Datorn behandlar dock fortfarande raden som en REM-rad, dvs den hoppar över raden när programmet körs.

Förvirrad? Det är ingenting emot vad din kompis, som inte har läst den här artikeln kommer att bli när han listar dina REMmade program!

På liknande sätt kan man få fram: inskjutning, rensning av skärmen och markörförflyttning m.m. I många fall måste man lägga in ett skiftat reverserat M (skiftad return), för att kommandona skall fungera fritt. Med lite fantasi kan man få ett helt litet program i programmet, som körs när du LISTar i stället för när du skriver RUN!

*) Skiftat L motsvaras på maskiner med PET 4.0 BASIC av skiftad vänster hakparentes.

J.M.L.

Planera din ekonomi

Av Lars Moeschlin
S. Fiskebäcksvägen 84
421 58 Västra Frölunda
Telefon 031-29 38 72 (efter kl 18)

Vi går mot ekonomiskt bistrare tider. Priser och avgifter höjs och det blir allt väsentligare att ha god överblick över sin ekonomi. Har man möjlighet att göra en semesterresa till sommaren? Hur mycket skall man spara? Måste några utgifter minskas?

Är ekonomin komplicerad och kärv är det motiverat att använda ett datorprogram för att räta ut alla frågetecken. Med programmet nedan kan man bolla med inkomster, utgifter och sparande tills en godtagbar lösning erhållits.

Programmet är uppbyggt med konton där man lägger in sina fasta utgifter (hyra, el etc.) och inkomster (lön, barnbidrag etc.).

Vid avlöningstillfället ger programmet besked om hur mycket som skall sparas eller tas ut från bankkontot. Med de disponibla pengarna betalas räkningarna och resten används för rörliga utgifter (mat, kläder etc.).

Programmet som listats nedan är avsett för VIC 20. Med modifieringar av skärmhanteringen bör det även gå att köra på 64:an. Minst 3 K extra minnesexpansion krävs. När det är inprogrammerat körs det enligt körinstruktionen. Denna behövs bara till en början eftersom programmet är självinstruerande.

```
80 POKE36879,99:PRINT"■":MK=50:GOTO100
90 GETX$:IFX$=""GOTO90
91 RETURN
100 ML=13:NP=12:MX=13:DIMT$(MK),B%(MK),S%(MK),H%(MK),V(ML),U(ML),P(ML),A(MX),R(N
P)
105 DIMQ%(8):FORI=1TO8:READQ%(I):NEXT:G=214
106 DATA100,82,70,64,67,68,69,99
110 C$=CHR$(13):NK$=" NYTT KONTO ":GK$=" GAMMALT KONTO ":X=PEEK(648):K$="KONTOHA
NTERING"
115 R$="TIDIGARE SPARLÄGE":L$="AKTUELLA KONTON"
120 S$="FRAMTIDA SPARLÄGE":SK=37888:TK=4096:IFX=30THENSK=38400:TK=7680
121 N$="■TRYCK NGN TANGENT"
130 PRINT"■■■■EKONOMISK PLANERING■"
140 INPUT"■LAGRADE DATA IN";X$:IFLEFT$(X$,1)="N"GOTO220
150 INPUT"■FILNAMN";F$:OPEN1,1,0,F$:INPUT#1,NK,NL,NP,RM
160 FORK=1TONK:INPUT#1,T$(K),B%(K),S%(K),H%(K):NEXT
180 FORI=1TONL:INPUT#1,V(I),U(I),P(I):NEXT
185 FORI=1TONP:INPUT#1,R(I):NEXT:CLOSE1
190 PRINT"■■■■■■■■MENY■■":PRINT"■1 "K$","■2 KALKYLERING","■3 "R$","■4 "S$","■5 LAGRA"
200 INPUT"■1,2,3,4 ELLER 5";X:IFX<1ORX>5GOTO190
210 ONXGOTO260,540,1000,952,1100
220 PRINT"■■ NY START":NK=0:NL=0:GOTO290
260 PRINT"■■ ■"K$","■■■■■■MENY■","■1";NK$,"■2";GK$,"■3 LISTA KONTON","■4 "L$
270 PRINT"■5 INGETDERA":INPUT"■1,2,3,4■ELLER5";X:IFX<1ORX>4GOTO190
280 ONXGOTO290,420,460,1200
290 NK=NK+1:K=NK:M=1
300 PRINT"■■"NK$"NR="STR$(K):GOSUB310:GOSUB370:GOTO390
310 INPUT"■KONTONAMN";T$(K)
312 IFLEFT$(T$(K),1)<>"←"GOTO320
314 INPUT"■RADERA KONTOT";X$:IFLEFT$(X$,1)<>"J"GOTO320
316 FORI=K+1TONK:J=I-1:T$(J)=T$(I):B%(J)=B%(I):S%(J)=S%(I):H%(J)=H%(I)
318 NEXT:NK=NK-1:K=K-1:RETURN
320 INPUT"■UTGIFT/PERIOD=";B%(K)
330 INPUT"■STARTPERIOD=";S%(K):IFS%(K)>NPGOTO330
340 INPUT"■PERIODINTERVALL=";H%(K):IFH%(K)>0THENRETURN
360 GOTO340
370 PRINT"■"RIGHT$(STR$(K),2)" "T$(K):PRINTSPC(6-LEN(STR$(B%(K))))B%(K)"■KR STÄR
T";
380 PRINTRIGHT$(STR$(S%(K)),2)" INT";:RETURN
381 PRINTRIGHT$(STR$(H%(K)),2);:RETURN
390 INPUT"■KORREKT INSKRIVET";X$:IFLEFT$(X$,1)="N"THEN ONMGOTO300,440
400 PRINT"■KONTOT LAGRAT":INPUT"■FLER KONTON";X$:IFLEFT$(X$,1)="J"THENONMGOTO290
,420
410 GOTO260
420 M=2:INPUT"■■KONTONR";K:IFK<1ORK>NKGOTO420
430 GOSUB370
440 PRINT"■OM GAMLA DATA SKALL   VARA KVAR TRYCK <RETURN> EFTER ?-T
ECKEN"
```

```
445 PRINT"RADERA MED KONTONAMN ←"
450 GOSUB310:GOSUB370:GOTO390
460 K=1
470 PRINT"KONTOLISTA":J=0
480 GOSUB370:K=K+1:J=J+1:IFK>NKGOTO500
490 IFJ<6GOTO480
500 PRINTN$;
510 GOSUB90
520 IFK>NKGOTO260
530 GOTO470
540 PRINT"    AVKALKYLERING=":M=3:IFNL=0THENINPUT"VILKET PERIODNR";PN:GOTO560
550 PN=P(NL)+1:IFRIGHT$(STR$(PN),2)>RIGHT$(STR$(NP),2)THENPN=PN+100-NP
560 PRINT"PERIODNR";PN:INPUT"KORREKT PERIODNR";Z$:IFLEFT$(Z$,1)="J"GOTO590
570 INPUT"VILKET PERIODNR";Z:IFNL=0THENPN=Z:GOTO560
575 IFZ=P(NL)THENM=4:PN=P(NL):PRINT"KALKYL FÖR PERIOD";Z"GÖRS OM":GOTO590
580 PRINT"KALKYLERING OMÖJLIG":GOTO560
590 X=0:Y=0:FORI=0TONP:R(I)=0:NEXT:PRINT"  ":FORK=1TONK:PRINT"KONTO NR="K:I=S%(
K)
600 Y=Y+B%(K):R(I)=R(I)+B%(K):IFI=VAL(RIGHT$(STR$(PN),2))THENX=X+B%(K)
610 I=I+H%(K):IFI<NP+1GOTO600
620 NEXT:RM=INT(Y/NP+.5):D=X-RM:PRINT"VERKLIG UTGIFT=";X:PRINT"MEDELUTGIFT=";R
M
630 IFD<0THENPRINT"SPARA";-D;"KR
640 IFD>=0THENPRINT"TAG UT"D"KR
650 E=0:INPUT"EXTRA SPARANDE=";E:IFM=4GOTO680
660 IFNL<MLTHENNL=NL+1:GOTO680
670 FORI=2TONL:V(I-1)=V(I):U(I-1)=U(I):P(I-1)=P(I):NEXT
680 P(NL)=PN:U(NL)=U(NL-1)-D+E:PRINT"PLANERAT SPAR=";U(NL)
690 INPUT"VERKLIGT SPAR=";V(NL):PRINT"VERKL-PLAN="V(NL)-U(NL)"KR"
702 PRINT,,N$
705 GOSUB90
710 GOTO900
720 X=1019:AA=-X:AB=X:IFAT=0THENAA=0:AB=0:E=18:M=2
721 FORI=1TONA:IFAA<A(I)THENAA=A(I)
730 IFAB>A(I)THENAB=A(I)
740 NEXT:D=(AA-AB)/E:IFD<200THEND=200
750 Y=LEN(STR$(INT(D+1)))-2:Z=D/10^Y:B=1:C=2:IFZ>1THENB=2:C=10
760 IFZ>2THENB=5
770 IFZ>5THENB=10:C=20
780 AD=B*10^Y:CC=C*10^(Y-3):L=-1:FORI=MTOE+M:LK=SK+22*I:K=L:FORJ=5TO4+NA
785 K=NOTK:IFKTHENPOKELK+J,7
790 NEXTJ:L=NOTL:NEXTI:AI=AD*INT(AB/AD):AC=AI:FORI=E+M+1TOM+1STEP-1
800 X=INT(AC/200+.5)/5:IFX-CC*INT(X/CC)=0THENPOKEG,I:PRINT"XTAB(4)CHR$(175)
810 AC=AC+AD:NEXTI
820 FORI=1TONA:X=(A(I)-AI)/AD:K=INT(X):L=INT(8*(X-INT(X)))+1
830 POKETK+(E+M-K)*22+1+4,8*(L):NEXTI:RETURN
840 POKEG,22:PRINT"  STARTAL":FORI=1TONL:X$=STR$(P(I)):X=VAL(RIGHT$(X$,2))
850 X$=MID$(X$,2,2):IFX<12GOTO880
855 POKETK+488+I,103:IFI<3GOTO890
860 IFI<12ANDX>ITHENX=I
870 PRINTTAB(I-X/2+4)X$;:GOTO890
880 IFI=NLANDX>1ANDI>1GOTO860
890 NEXTI:RETURN
900 PRINT"  PER  UTGIFT DIFFERENS":FORI=1TONP:X=R(I):Y=R(I)-RM
910 PRINTSPC(3-LEN(STR$(I)))ISPC(8-LEN(STR$(X)))XSPC(8-LEN(STR$(Y)))Y:NEXT
920 PRINTN$:GOSUB90
931 AT=0:NA=NP:FORI=1TONP:A(I)=R(I):NEXT:PRINT"  1234567890123 PER-PERIOD";
932 POKEG,23:PRINTTAB(5)"
940 GOSUB90:GOTO190
952 L=VAL(RIGHT$(STR$(P(NL)),2)):X=U(NL):A(1)=X:Y=0:FORI=2TONP:L=L+1
954 IFL>NPTHENL=1
956 IFL>NPTHENL=1
960 A(I)=A(I-1)+RM-R(L):IFA(I)<THENX=A(I):Y=I-1
965 NEXT:PRINT"  "S$," BLIR MIN EFTER"Y"PER","MIN SPARAT ="X"KR"
967 PRINTN$:GOSUB90
970 PRINT"  TKR "S$:AT=0:NA=NP:GOSUB720
975 POKEG,23:PRINT"  EFTER0123456789012 PERIODER";
980 GOSUB90:GOTO190
1000 PRINT"  R$"  ","  PER VERKL PLANE DIFF.";
1010 FORI=1TONL:X$=RIGHT$(STR$(P(I)),4):Y$=RIGHT$(STR$(V(I)),6)
1020 Z$=RIGHT$(STR$(U(I)),6):A$=RIGHT$(STR$(V(I)-U(I)),6)
1030 PRINTSPC(4-LEN(X$))X$SPC(6-LEN(Y$))Y$SPC(6-LEN(Z$))Z$SPC(6-LEN(A$))A$:NEXT
1040 PRINTN$:GOSUB90
1050 PRINT"  TKR VERKLIGT SPARLOGE":AT=0:NA=NL:FORI=1TONL:A(I)=V(I)
1051 NEXT:AT=0:GOSUB720:GOSUB840
1060 GOSUB90
1070 PRINT"  TKR VERKL-PLAN SPARL.":AT=1:FORI=1TONL:A(I)=V(I)-U(I):NEXT
1080 GOSUB720:GOSUB840:GOSUB90:GOTO190
1100 PRINT"  LAGRING AV DATA":X$=F$:INPUT"FILNAMN";X$:IFX$<>""THENF$=X$
1105 INPUT"  BAND I KORREKT POS";X$:IFLEFT$(X$,1)<>"J"GOTO1100
```

forts nästa sida

```
1110 OPEN1,1,1,F$:PRINT#1,NK;C$;NL;C$;NP;C$;RM;C$;
1115 FORK=1TONK:PRINT#1,T$(K);C$;B%(K);C$;S%(K);C$;H%(K);C$;:NEXT
1120 FORI=1TONL:PRINT#1,V(I);C$;U(I);C$;P(I);C$;:NEXT
1130 FORI=1TONP:PRINT#1,R(I);C$;:NEXT:CLOSE1
1140 INPUT"■KORREKT LAGRAT";X$:IFLEFT$(X$,1)<>"J"GOTO1100
1150 END
1200 PRINT"■■■L$"■":INPUT"VILKEN PERIOD";I:I=VAL(RIGHT$(STR$(I),2))
1210 PRINT"■ NR KONTO          BELOPP■":Y=0:FORK=1TONK:J=S%(K)
1215 IFI<>JGOTO1230
1220 Y=Y+B%(K):PRINTTAB(3-LEN(STR$(K)))KLEFT$(T$(K),11)TAB(21-LEN(STR$(B%(K))))B
%(K);
1230 J=J+H%(K):IFJ<NP+1GOTO1215
1240 NEXT:PRINTTAB(8)"■SUMMA"TAB(21-LEN(STR$(Y)))Y;:PRINT" "N$:GOSUB90:GOTO260
```

Körinstruktion

START

Starta med RUN. Programmet frågar då: "LAGRADE DATA IN?" Här kan man läsa in eventuella tidigare lagrade data på kassettband (vid svar JA eller bara J). Vid första starten svarar vi NEJ eller bara N (vi har ju inga lagrade data då!) och trycker på (RETURN)-tangenten.

Programmet är gjort för 12 perioder (månader) per år. En period kan t.ex. vara från den 25 jan till den 24 febr – om lönen erhålls den 25 januari.

Kontohantering

Efter denna inledning läses kontona in (max 50 st). Kontonamnet bör begränsas till 17 bokstäver. I genomsnitt bör ej fler än 9 bokstäver användas. Utgiften per period kan vara positiv eller negativ. Ett negativt värde, t.ex. -500, innebär en inkomst på 500 kronor.

Startperiod är den första period på året som utgiften skall betalas (eller som inkomsten erhålls). Periodintervall är det intervall i antal perioder mellan utgifterna i fortsättningen (till årets slut). Om en period är lika med en månad och startperioden är 4 och periodintervallet 3 betalas utgiften månaderna 4, 7 och 10.

När kontoinläsningen är klar skrivs data ut och kan rättas. Vid rättning tryck på (RETURN) om gamla data skall behållas. Skall kontot raderas så tryck på (←)+(RETURN) när namnet läses in. Härefter visas föregående konto (OBS att konto nr 0 ej används!) När inläsning av konto är klar visas grundmenyn:

Meny

1. KONTOHANTERING
2. KALKYLERING
3. TIDIGARE SPARLÄGE
4. FRAMTIDA SPARLÄGE
5. LAGRA

Väljes nu 1 visas i nästa meny:

1. NYTT KONTO
2. GAMMALT KONTO
3. LISTA KONTON
4. AKTUELLA KONTON
5. INGETDERA

Fall 1, nytt konto, har vi redan klarat av. Fall 2, GAMMALT KONTO ger oss mjlighet att rätta visst konto. Fall 3 listar konton, vilket ibland är nödvändigt för att få reda på kontonummer m.m. Fall 4 tar fram de konton som är aktuella för en viss period (kontonamn och belopp). Här har man möjlighet att se om beloppen stämmer innan en kalkyl utförs.

Med 5 (RETURN) återvänder man till grundmenyn. Efter att ha knappat in kontona är det dags för en första kalkyl. Återvänd till grundmenyn och tryck 2(RETURN).

KALKYLERING

Har man inte läst in lagrade data från kassettband frågar datorn efter periodnr. Detta anges med fyra siffror, årtal och period, t.ex. 8311. Är detta korrekt inslaget så visas efter viss beräkningstid verklig utgift för perioden och medelutgift per period. Dessutom föreslår datorn hur mycket man skall spara eller ta ut denna period för att få lika mycket pengar över varje period för rörliga utgifter. Notera detta värde!

Nu är det dags

att läsa in det extra sparandet. Här kan man påverka det planerade sparandet. Detta skall visa hur mycket pengar som måste finnas på bankkontot, för att man skall klara av utgifterna under året utan att ekonomin är sämre efter ett år. Vid den första körningen inläses här det verkliga sparläget i slutet av föregående period. Därefter görs extra sparande endast när man gjort insättning eller uttag från bankkontot för saker som ej är medtagna i kontona. (T.ex. bör gåva på 1000 kr noteras som 1000 kr extra sparande. Ett köp av symaskin för 2000 kr bör noteras som -2000 kr extra sparande = uttag av 2000 kr).

Det ovannämnda skulle räcka om inte inflationen fanns. Inflationen leder till att kontobeloppen då och då höjs utan att pengar tidigare satts undan. Detta botas genom att varje period göra ett extra sparande så att en buffert erhålles mot senare kostnadsökningar. En god tumregel är att detta extra sparande bör vara på 5% av den inflationskänsliga delen av periodmedelutgiften (vid 10%-ig inflation). Om t.ex. hyra, el, telefon etc ger en månadsmedelutgift på 3000 kr så bör ett extra sparande på 150 kr ske varje period.

Härefter visar datorn planerat sparläge, dvs hur mycket man bör ha på bankkontot i slutet av perioden. Verkligt sparläge på banken läses därefter in. Om man befinner sig i början av perioden får man här ev. gissa och i så fall göra om kalkylen i slutet av perioden.

Så småningom visas aktuella uppgifter för samtliga perioder på året

och skillnaden gentemot periodmedelutgiften. Detta plottas ut (sedan man tryckt ner någon tangent) som ett stapeldiagram på bildskärmen. När nu kalkylen är klar, är det dags att kontrollera den framtida sparsituationen. Räcker pengarna på bankkontot? Krävs extra banklån eller skall utgifterna minskas???

Framtida sparande

Tryck nu 4(RETURN) så beräknar datorn när sparläget enligt planerna blir sämst och hur stort sparat belopp man då har. Är beloppet negativt är situationen dålig och måste åtgärdas, t.ex. genom att minska utgifterna, att låna eller göra extra sparande (man får då mindre summa över för de rörliga utgifterna).

Om åtgärd skall ske denna period genom att ändra i konto, t.ex. ta bort någon utgift, eller genom extra sparande, då måste kalkylen göras om för denna period. Efter tryck på någon tangent visas sparsituationen ett år framåt (normalt efter 0 till 11 perioder). Vill man sen även se hur sparsituationen varierat tidigare så återvänd till grundmenyn och tryck på 3(RETURN).

Sparsituation

Här listas sparsituationen för de senaste perioderna (max 13 st). Dels visas vad man verkligen hade sparat men även vad man hade planerat att spara samt skillnaden dem emellan. Efter tryck på någon tangent plottas även dessa värden ut.

När man efter ett antal varv på detta sätt känner sig nöjd med den ekonomiska situationen skall data lagras.

Lagra

Tryck 5(RETURN) och knappa in ett filnamn, som noteras för användning nästa gång data skall läsas. Lägg i ett kassettband och positionera detta rätt. Nu är du klar att lagra data. Om data önskas lagrade på flera ställen eller på olika band svarar man bara NEJ på frågan "KORREKT LAGRAT?" varefter även filnamn kan ändras.

Efter detta är programmet slut, men kan köras på nytt utan radering av data genom GOTO190(RETURN). **Används RUN så raderas alla data!**

Metod

MEDELUTGIFTEN per period erhålls som summan av alla utgifter under ett år delat med antalet perioder per år. AKTUELL UTGIFT för en viss period erhålls genom att testa vilka konton som belastar denna period och summera beloppen. PLANERAT SPARLÄGE erhålls genom att först summera planerat sparläge för föregående period med medelutgiften och det extra sparandet. Därefter minskas med aktuell utgift.

Programbeskrivning

Följande variabler används:

K	=	kontonummer (max MK)
L	=	löpnummer (max ML)
M	=	mod-nummer (1=nytt konto, 2=gammalt konto, 3=kalkyl, 4=kalkyl görs om)
MK	=	maximalt antal konton
ML	=	maximalt antal lagrade perioder
NK	=	antal konton
NP	=	antalet perioder per år
A()	=	fält för plottning
B%(k)	=	periodutgift för konto K
H%(k)	=	intervall mellan utgifterna
P(L)	=	periodnummer, t.ex. 8311
R(I)	=	summa utgifter för period I
S%(k)	=	startperiod för konto k
T$(k)	=	kontonamn för konto k
U(L)	=	planerat sparläge
V(L)	=	verkligt sparläge

Programmet

RAD 80: Sätter om skärmen för gemensam förgrund och variabel bakgrund. Normalt vit bakgrund och blå text. För rubriker och plottning används även gul bakgrund.
RAD 100-121: Diverse initieringar och kontroll av minnet för skärmhanteringen.
RAD 140-185: Inläsning av data från kassettband.
RAD 190-210: Grundmenyn
RAD 220-250: Startprocedur
RAD 260-530: Kontohantering
RAD 540-940: Kalkylering
RAD 720-830: En subrutin för plottning. Automatisk axeljustering i höjdled med stegen 2,5 eller 10 vid graderingen. Med AT=0 kommer alltid värdet noll med. Placering av markören (cursorn) på rätt rad I direkt sker genom
POKE 214,I:PRINT "(markör upp)";
Rad 840-890: En knepig subrutin för utskrift av årtal på axeln i sidled.
RAD 950-990: Beräknar planerat sparläge i framtiden.
RAD 1000-1150: Lagring av data på kassettband. C$=CHR$(13) används för kompakt lagring.
RAD 1200-1240: Framtagning av aktuella konton.

Hela programmet kan levereras färdigt på kassett av Lars Moeschlin till den som sätter in 45:- på en postanvisning till adressen i början av artikeln.

DIN VIC HEMDATOR KÖPER DU BÄST HOS
COMPUTER EXPRESS
Den professionelle distributören av datorer och mikrodatorprogram
ORDER TEL: 08-54 00 10

Jag beställer er sortimentkatalog
Namn: _____
Adress: _____
Skicka in till Computerexpress, Box 49035, 100 28 Stockholm

Konsten att sortera

Det är en av de allra vanligaste uppgifterna för en dator detta att sortera tal eller textsträngar i någon bestämd ordning. Kanske vill man hålla reda på att de tal man använder finns i stigande eller fallande ordning, kanske vill man ha en adresslista utskriven efter postnummerordning. Eller man kanske vill sortera ut alla som heter Andersson och Eriksson från en stor adresslista. Användningarna är nästan lika många som programmen.

Det är ganska roligt därför att vi här fått in flera sorteringsprogram samtidigt, olika på ett eller annat sätt. Tillsammans ger de en hel mängd olika procedurer och metoder för sortering som vi vill låta VIC-rapports läsare ta del av.

En lärare i Åmål, Sven Sjöberg, har t.o.m. gjort försök med olika sorteringsrutiner och undersökt hur lång tid de tar för en viss sorteringsuppgift. En annan medarbetare, Ingvar Wihlborg, har skrivit en alldeles egen sorteringsrutin, som lär vara en av de kortaste tänkbara. Ja, se själva!

Sorteringsförsök

av Sven Sjöberg

Vi saxar ur författarens egna förord: "SLASKMETODEN" är ett slags referensprogram vid jämförelserna med tidsåtgången vid sorteringarna. Tiderna skulle naturligtvis kunna minskas ytterligare om heltalsvariabler används. Bortsett från grundprogrammet "Slaskmetoden" är de övriga programmen egna funderingar och reflexioner. Förmodligen kan texten om "termometermetoden" ge uppslag åt andra intresserade, liksom det, som jag tror, helt annorlunda och nya betraktelsesättet i SORTERINGSFÖRSÖK. Detta är ett intressant område och det finns säkert mycket att göra här."

Det finns många metoder för sortering av tal. Den kanske vanligaste är den här kallade "Slaskmetoden". Denna finns återgiven i program 1. Där anges också tiderna för sortering av olika tal. Sorteringen har här bestått i att rättvända en talutskrift, som börjar i N och slutar med 1. Några dubletter förekommer således inte vid dessa förenklade körningar. De tider, som uppmätes för sorteringarna, är således minimitider, eftersom dubbla eller flera tal av samma sort medför en något längre sorteringsrutin (inom programmet).

A. Slaskmetoden visar sig vara snabb då det gäller att sortera ett mindre antal tal. Sorteringstiden växer däremot mycket snabbt för stigande antal element. Den är proportionell mot kvadraten på N.

10 sorteringar tar ung 1 sek medan 500 tar hela 39 min.

B. Ett sätt att nedbringa sorteringstiden vore att sortera från början och slutet samtidigt. Detta har gjorts i program 2. (Här behandlas även den för bilddimensionering värdefulla "termometermetoden".)

10 sorteringar tar ung. 1.2 sek och 500 tar nu 34 min.

C. Eftersom den tidigare delen av sorteringen är mindre tidskrävande, så borde man få snabbare körning, om programmet först sorterade ena halvan och sedan den andra - var för sig - och sedan sammanställde resultatet. Se "Slaskmetoden med halvering". Vi finner att denna metod är betydligt snabbare än den första.

10 sorteringar tar ung 0.6 sek och 500 tar nu 15.5 min.

D. Ett försök att nedbringa sorteringstiden genom indelning i fyra delar, som sorteras var för sig har gjorts. Mycket riktigt blir sorteringstiden ännu kortare. Tidsvinsten står dock inte i rimlig proportion till den ökade BYTES-åtgången. Jag anger därför inte programmet utan nämner endast resultatet.

10 sorteringar tar ung. 1.1 sek medan 500 tar 719 sek, dvs 12 min.

E. En kombination av B och C skulle ge ännu kortare sorteringstid. Men det är tydligt att man måste använda NÅGON HELT ANNORLUNDA metod för att nedbringa sorteringstiden för större termantal. En idé vore ju att slå in de N talen som A(I), där I går från 1 till N, och sedan KALLA dessa tal B(J), där J går FRÅN LÄGSTA A(I) TILL STÖRSTA A(I) och sedan låta DESSA SKRIVA UPP SIG SJÄLVA. Det hela visade sig fungera utmärkt - och snabbt! Det gäller således för positiva heltal. Men om två eller flera tal var lika förlorades dubletterna. Därför har jag nu tillfogat även en rutin, som klarar av den detaljen. Se program 3. (Det har visat sig att rutinen även klarar decimaltalssortering, om dessa inte har samma heltalsdel!)

10 sorteringar tar ung. 0.3 sek och 500 tar bara knappt 22 sek.

Således en avsevärd tidsvinst (från *39 minuter* till *22 sekunder* för sortering av de 500 talen).

Sortering ("Slask-metoden")

I följande uppställning kan talen A(I) anta vilka värden som helst. Flera av samma storlek får förekomma.

Program 1

```
10 PRINT CHR$(147)
20 INPUT"N=";N
30 DIM A(N)
40 FOR I=1 TO N
50 INPUT"A(I)=";A(I)
60 next I
61 T1=TI
70 FOR I=1 TO N-1
80 FOR J=1+I TO N
90 IF A(I)<=A(J) THEN GOTO 130
100 S=A(I)
110 A(I)=A(J)
120 A(J)=S
130 NEXT J
140 NEXT I
150 PRINT"SORTERAT"
160 FOR I=1 TO N
170 PRINT A(I)
180 NEXT I
190 PRINT (TI-T1)/60;"SEK"
200 END
```

Vid körning har använts 50 A(I)= N+1-1 som sedan rättvändes av rutinen. Samtliga talvärden är således olika.

N=	TID (sek)	K=LN(T2/T1)/ LN(N2/N1)
10	1.05	1.95
50	24.07	1.96
100	93.93	1.97
150	208.88	1.98
200	368.77	1.98
250	574.1	2.01
300	828.6	2.04
350	1134.7	2.04
400	1489.4	2.03
450	1892.8	2.03
500	2345.2	

ger ung. 5.7 BYTES/tal

Tid för N=500 är *39 min*

$T \approx 0.0094 * N$

Max- Min-termometern ("Slask" både början och slutet samtidigt)

Vi skall här använda oss av den möjlighet som finns, då det gäller att bestämma en talföljds största och minsta värde med hjälp av dator. I matematiken brukar man bestämma en funktions maxima och minima samt randvärdena i början och slutet av intervallet, när man vill få funktionens högsta och lägsta värde inom intervallet. Dessutom skulle då fordras en sorteringsrutin. Vid bl.a. dimensionering av bildens storlek har ju funktionens största och lägsta värde stor betydelse.

Vi skall se att dessa värden kan vara intressanta även ur sorteringssynpunkt.

Jag visar här idén för detta. Det ligger samma tanke bakom denna sortering som den vid maximi-minimitermometern.

Vi låter A vara ett godtyckligt värde som dock är större än det minsta värde, som funktionen eller talföljden har. Så låter vi funktionen genomlöpas av x från början till slutet av intervallet. Det inträffar då, att funktionsvärdet blir mindre än A men DÅ LÅTER VI A BLI = FUNKTIONSVÄRDET. Sedan stiger kanske funktionsvärdet medan A ligger kvar tills funktionen kanske antar ett ännu lägre värde. Återigen låter vi nu A bli = funktionsvärdet osv. När funktionen (talföljden) har genomsökts från början till slut, så anger således A funktionens (talföljdens) lägsta värde inom intervallet. På liknande sätt förfares med funktionens högsta värde.

På Basic kan detta få följande formulering:

```
10  LET A=10000:B=-10000                (exempelvis      )
20  FOR X=1 TO 50                       (allmänt: TO   N  )
30  IF A(X) < A  THEN A=A(X)
40  IF A(X) > B  THEN B=A(X)
50  NEXT X
60  PRINT "HÖGST";B,"LÄGST";A
```

Detta förfaringssätt tillämpar vi generellt på sorteringen nedan. Den tillgår så, att första och sista värdena utsorteras först. Därefter utsorteras näst första och näst sista osv. tills alla talen blivit sorterade.

```
10  PRINT CHR$(147)                     (Dimensionering av A(I),B(I) )
20  INPUT"ANTAL";N
30  DIM A(N),B(N)
40  FOR I=1 TO N                        ┐
50  INPUT"A(I)";A(I)                    │ Inmatning
60  IF I<=N/2 THEN B(I)=10000           │          (Jämför rad 10 ovan)
70  NEXT I                              ┘
80  FOR J=1 TO N/2+0.5                  ┐
90   FOR I=J TO N+1-J                   │
100   IF A(I)<B(J) THEN B(J)=A(I)       │ Sortering
110   IF A(I)>B(N+1-J) THEN B(N+1-J)=A(I)│  (0.5 har tillfogats för det
120  NEXT I                             │   fall att N är udda tal)
130 NEXT J                              ┘
140 FOR J=1 TO N                        ┐
150  PRINT B(J)                         │ Utskrift
160 NEXT J                              ┘
```

forts. nästa sida

Vid provkörning av programmet har rad 50 A(I)=N+1-I använts. Tidmätning införd genom 75 T1=TI och PRINT (TI-T1)/60;"SEK"

10	sorteringar tar	1.2 sek
50		23 sek
100		85 sek
200		330 sek
500		2042 sek

Tid för N=500 är *34 min*
(ger ung. 10.7 BYTES/tal)

Sortering "Slaskmetoden med halvering"

I följande uppställning kan talen A(I) anta vilka värden som helst. Flera av samma storlek får förekomma. Intervallet har halverats en gång.

```
10  REM  SORTERING (HALVERAD SLASK)
14  D=100000:A=D:B=-D
15  PRINT CHR$(147)
16  K=0:L=0:M=0:N=0:F=0
20  INPUT"ANTAL";N
40  DIM A(N),B(N)
50  PRINT"SKRIV TALEN
    (RETURN EFTER VARJE)"
60  FOR I=1 TO N
70  INPUT"A(I)";A(I)
75  IF A(I)<A THEN A=A(I)
77  IF A(I)>B THEN B=A(I)
80   NEXT I
82  M=(A+B)/2
84  FOR R=1 TO N
85  IF A(R)<=M THEN GOSUB 300
86  IF A(R)  M THEN GOSUB 350
87   NEXT R
88  PRINT"SORTERAS"
89  T1=TI          (för tidmätningen)
90  FOR V=1 TO 2
91 IF V=1 THEN P=1:N=K
92 IF V=2 THEN P=K+1:N=K+L
96  FOR I=P TO N-1
100 FOR J=I+1 TO N
110 IF B(I)<B(J) THEN 150
120 LET S= B(I)
130 LET B(I)=B(J)
140 LET B(J)=S
150  NEXT J
160  NEXT I
180 FOR I=P TO N
195 PRINT B(I)
200  NEXT I
202  NEXT V
205 PRINT (TI-T1)/60;"SEK"
210 END
300 REM   DELNING
305 K=K+1
310 B(K)=A(R)
315 RETURN
350 L=L+1
355 B(N-L+1)=A(R)
300 RETURN
```

N=	TID (sek)	K=
10	0.6	
50	10.63	
100	39.53	
150	86.67	
200	151.68	
250	235.07	
300	335.5	
400	594.83	
500	929.25	

TID för 500 är *15.5 min*

Program 3

```
10  PRINT CHR$(147)
20  A=10000:B=-10000
30  INPUT"N=";N                    (Antalet tal)
40  DIM A(N),B(400)                (Dimensionering ex)
50  FOR I=1 TO N
60  PRINT"A";I;
70  INPUT A(I)
80  IF A(I)<A THEN A=A(I)
90  IF A(I)>B THEN B=A(I)
100 J=A(I)
110 IF INT(B(J))<>0 THEN K=K+1:C(K)=B(J)
120 LET B(J)=A(I)
130 NEXT I
140 FOR J=INT(A) TO B
150 IF B(J)=0 THEN 200
160 FOR S=1 TO K
170 IF INT(C(S))=INT(B(J)) THEN PRINT C(S)
180 NEXT S
190 PRINT B(J)
200 NEXT J
210 END
```

I följande uppställning antas talen A(I) vara heltal. Antalet tal är N st. Flera tal av samma storlek får förekomma. (Anm. Denna rutin sorterar även alla positiva decimaltal, som inte har samma heltalsdel).

a)
Tidmätning kan göras genom
123 T1=TI
203 PRINT (TI-T1)/60;"SEK"

b)
Vid provkörning av ett stort antal termer kan införandet av A(I) förenklas så att de N talen rättvändes från omvänd placering med
60 A(I)=N+1-I
70 borttages

c)
Vid sortering av HELTAL behövs inte INT beteckningarna (ex. i stället för INT(B(J)) skrives då B(J) osv).
500 sorteringar tar då *22 sek.* Detta kan jämföras med
500 "slaskmetod"-sorteringar vilket tar *39 min.*

10	sorteringar tar ung	0.33 sek
50		2.0
100		4.23
200		8.7
300		13.2
400		17.8
500		22.35 sek

$T \approx 0.043 \quad N$

EN BRA LAPP!







Ja, det är kanske lite dyrt för en T-shirt, men du får ju faktiskt helårsprenumeration av VIDEOTIME-HEMDATOR på köpet. Och visst kan det väl vara skönt att få tidningen hem i brevlådan och slippa springa till kiosken i det ruggiga höstvädret! Åh du, om du absolut måste ut i höststormen så har du ju i alla fall den där sköna bomullströjan närmast kroppen. Prenumerera NU, och låt Videotime-Hemdator förgylla dina hemmakvällar.



Ja, jag vill prenumerera ett år på Videotime-Hemdator för 100:— och får då en T-shirt på köpet.

Jag betalar först när inbetalningskortet kommer, och min T-shirt ska vara

	GUL	VIT	LJUSBLÅ
och storlek	S	M	L XL

Namn: _____

Adress: _____

Postadress: _____

Frankeras ej VIDEOTIME HEMDATOR betalar portot

VIDEOTIME
LBA

Svarsförsändelse

Kontonr: 15935000
443 01 Lerum

VIDEO TIME HEMDATOR

SORTERING

Att sortera listor är en uppgift som datorer ofta ställs inför, och följaktligen finns det många olika sätt att sortera på. Olika sorteringsalgoritmer är bra på olika saker, vissa sorterar en totalt oordnad lista relativt snabbt, och vissa är bra på att sortera in ett par element i en ordnad lista. Bubble-sort, den algoritm som jag nu kommer att beskriva, är faktiskt inte särskilt bra på varken det ena eller det andra. Den har emellertid den fördelen att den är kort och lättförståelig, något som kan vara väl så trevligt. Bubble-sort kallas den för efter sättet att sortera. Vissa element kommer att flyta uppåt (som bubblor) i listan tills att den är sorterad.

Tekniskt sett går sorteringen till på följande sätt. Först jämförs det översta elementet med det andra, och om de ligger fel i förhållande till varandra byter de plats. Det första elementet jämförs sedan med det tredje, och proceduren upprepas tills att det sista elementet har nåtts. Detta resulterar i att det största (eller minsta, beroende på hur jämförelsen går till och hur listan skall sorteras) elementet nu ligger överst i listan. Det andra elementet jämförs sedan med de övriga elementen (ej med det översta som ju ligger rätt!) på samma sätt som förut. Detta upprepas tills det att det näst nedersta elementet har jämförts med det nedersta. Listan är då sorterad. Det är många jämförelser som måste göras, och ofta också många byten, så det tar sin lilla tid. Om en någorlunda väl ordnad lista skall sorteras, är det färre byten som måste göras än med en rörig lista och det går lite fortare.

Programmet nedan ger ett exempel på hur en bubble-sort kan fungera. Det sorterar tal som har lästs in till en vektor, men om man byter variabeltypen kan man naturligtvis använda rutinen till att sortera strängar eller något annat. Listan kommer att sorteras så att högsta elementet (talet) kommer att komma överst. Man ändrar detta genom att ändra jämförelsen på rad 190. Observera att när man skall byta värden på två variabler på rad 200, är man tvungen att ha en hjälpvariabel (i det här fallet X).

MOGF

```
100 DIM A(100)
110
115 REM-- LAS IN TAL
117
120 I=I+1
130 : INPUT A(I)
140 IF A(I)<>0 THEN 120
150 AN=I-1
160
165 REM-- SORTERA LISTAN
167
170 FOR I=1 TO AN-1
180 : FOR J=I+1 TO AN
190 : : IF A(I)>=A(J) THEN 210
200 : : X=A(I): A(I)=A(J): A(J)=X
210 : NEXT J
220 NEXT I
230
235 REM-- SKRIV UT LISTAN
237
240 FOR I=1 TO AN
250 : PRINT A(I)
260 NEXT
```

Sorteringsprogram för VIC-20

I händelse av att någon behöver sorteringsrutin i något program, visas här två relativt snabba Basicvarianter. Den ena varianten visar sortering av numeriska värden (tal) och den andra visar sortering av strängar.

Programmen sorterar med lägsta värdet på topp (bokstaven a etc. för strängar) men kan ändras till omvänd sortering, genom att villkorssatsen på rad 20 istället för större än > = får mindre än < =.

Programbitarna kan finna olika användningar. Se t.ex. serietabellsortering i tipsprogrammet.

Ingvar Wihlborg

Sortering (numeriska värden).

```
0 PRINT"J":POKE36879,25
5 PRINT"HUR MA■i.XNGA TAL SKALL":INPUT"SORTERAS";X:DIMB(X)
10 PRINT"J":INPUT"TALET";A$
15 A=VAL(A$):IFA$="*"THENC=0:GOTO100:REM  (TILL UTSKRIFT)
20 IFB(C)>=ATHEN40
30 C=C+1
40 IFB(C)<>ATHEN60
50 C=C+1
60 D=B(C):B(C)=A:A=0
70 IFD<>0THENA=D:GOTO20
80 C=D
90 GOTO10
100 C=C+1:IFB(C)<>0THENPRINTB(C):GOTO100
110 END

READY.
```

Sortering (strängar).

```
0 PRINT"J":POKE36879,25
5 PRINT"HUR MA■i.XNGA STRANGAR":INPUT"SKALL SORTERAS";X:DIMB$(X)
10 PRINT"J":INPUT"STRANG";A$
15 IFA$="*"THENC=0:GOTO100:REM  (TILL UTSKRIFT)
20 IFB$(C)>=A$THEN40
30 C=C+1
40 IFB$(C)<>A$THEN60
50 C=C+1
60 D$=B$(C):B$(C)=A$:A$=""
70 IFD$<>""THENA$=D$:GOTO20
80 C=VAL(D$)
90 GOTO10
100 C=C+1:IFB$(C)<>""THENPRINTB$(C):GOTO100
110 END

READY.
```

VIC 20 VIC 64



GRANA

SOFTWARE

SPEL och NYTTOPROGRAM i MÄNGDER
plus hela HANDIC-sortimentet.



Gratis katalog!
Ring 018–11 30 03

Återförsäljare sökes!

GRANA SOFTWARE Box 26051, 750 26 Uppsala, Sweden ★ Telefon 018-39 80 77, 39 80 57

Variabellagring i Basic/ VIC-20/-64

I den här lilla artikeln ska vi ta oss en titt på hur Basic lagrar sina variabler och själv ligger lagrad i minnet. Denna information är ofta mycket värdefull då man vill göra något utöver vad den normala programmeringen tillåter. Det som sägs här gäller också för CBM-64, men då jag inte har tillgång till en 64a har jag inte haft någon chans att verifiera uppgifterna, så du som äger denna maskin får ta informationen med "en nypa salt".

När en VIC-20 slås på så testar den (initieringsrutinen), hur mycket minne den har och sätter pekarna A och F (se fig. 1). Huruvida detta sker för 64an vet jag inte, men vi kan i alla fall vara säkra på att 64ans initieringsrutin sätter samma pekare. I VIC-20's fall beror de värden som pekarna får på vilken typ av expansion man har.

Nästa steg som utförs är att pekare E sätts lika med pekare F, vilket betyder att inga strängar är definierade ännu. Sedan sätts pekarna B, C och D lika med A's värde, det betyder att inget Basic-program finns i minnet (B) samt att inga variabler och arrayer är definierade (C, D). När rutinen efter att ha lagrat diverse andra variabler lägger den sig i direkt mod, dvs. väntar på att du ska ge den ett kommando (cursorn blinkar och READY står på skärmen). Från detta stadium har vi två vägar att gå:

1. Vi laddar ett program.
2. Vi skriver in ett program från tangentbordet.

I båda dessa fall gäller att pekarna B, C och D följs åt och flyttas uppåt, dvs. de pekar alla på Basic-slutet så länge inga variabler är definierade. Låt oss nu studera hur C, D varierar när variabler definieras. Antag att vid körningen av programmet så ser rad tio ut så här:

10 A = 7000

Pekarna C, D flyttas nu uppåt 7 bytes och definitionen av A, dvs. A får värdet 7000, läggs i området för enkla variabler. Detta betyder att C nu pekar på nästa fria byte i variabeltabellen, alltså det ställe där nästa variabel kommer att ligga. Notera dock att C även anger starten för arrayerna, har vi en eller flera arrayer definierade så pekar C på ett element i en av dessa arrayer. Arrayområdet flyttas alltså också uppåt när en enkel variabel definieras, detta betyder att även pekare D flyttas uppåt 7 bytes och det fria minnet krymper. Antag nu att rad 20 ser ut så här:

20 B$="TOK-STRÄNG"

Vad som nu händer är att vår sträng läggs i strängområdet och pekare E flyttas nedåt (i detta fall 10 bytes). Efter det att strängen lagts in, läggs strängdefinitionen in i variabelområdet, vilket medför att pekarna C, D flyttas uppåt 7 bytes. Som du ser så minskas området mellan D och E i storlek, det fria minnet krymper. Det resultat du får efter operationen FRE(X) är just skillnaden mellan dessa två variabler, men först görs något som kallas "Garbage Collect" (ung. sophantering). Denna rutin ordnar upp strängområdet så att gamla strängar inte ligger och skräpar.

Låt oss ta ett exempel för att förstå hur detta fungerar. Antag att rad 30 definierar om vår strängvariabel B$

30 B$="ÄNNU TOKIGARE STRÄNG"

B$ finns nu redan definierad i variabelområdet så denna behöver inte skapas. Dock ska strängen läggas in i strängområdet. Vår gamla definition av B$ ersätts nu inte, utan den nya definitionen börjar läggas in på pekare E's position och nedåt, samtidigt som variabel B$'s pekare ändras så att den pekar hit istället. Detta betyder att vår gamla sträng "TOK-STRÄNG" nu är skräp, för ingen pekare pekar på denna. Detta förfaringssätt används för att man ska kunna använda strängar av olika längd och på så sätt spara minne, men som du inser så innebär förfaringssättet även att en hel del skräp blir kvar i form av "gamla" strängar. Det är dessa som "garbage collecten" tar bort. Så fort pekare E=pekare D eller FRE(0) körs, så görs en sådan "garbage collect" dvs. alla skräpsträngar tas bort och respektive pekare i variabelområdet ändras. De strängar som har pekare på sig läggs sedan i en snygg ordning med början från pekare F. Sist så ändras pekare E till den nya "botten" av sträng-

Figur 1



Decimal	Hex	Pekare
55/56	$37/$38	F
51/52	$33/$34	E
49/50	$31/$31	D
47/48	$2F/$30	C
45/46	$2D/$2E	B
43/44	$2B/$2C	A

lagringsområdet. Effekten är att det fria minnet ökar. Den stora nackdelen med detta förfarande är att operationen tar en "hiskelig" massa tid, tider upp till flera minuter är inte ovanliga för stora program, under vilken din maskin är helt "död". Tyvärr finns det inte så mycket att göra åt detta, men man kan i alla fall slippa garbage collecten när FRE(X) körs, genom att göra följande definition:

Def.

```
FNF(X)=PEEK(52)x256+PEEK(51)-
PEEK(50)x256-PEEK(49)
```

När du sedan skrivit PRINT FNF(X) (X är valfritt tal), så får du samma resultat som FRE(X), men ingen garbage collect utförs. För dig som är ägare av en 64'a ger denna definition ett korrektare värde än när du skriver FRE(X) eftersom denna rutin inte är gjord för lyckliga datoranvändare med mer än 32 K minne (ett heltal STÖRRE ÄN 32767 ses av Basic som ett negativt tal). Du vill kanske att garbage-rutinen ska köras så att det korrekta antalet fria bytes anges i så fall kan du testa:

Def.

```
FNF(X)=FRE(X)-FRE(X)+PEEK(52)x256
+PEEK(51)-PEEK(50)x256-PEEK(49)
```

Gör du denna definition när du startar upp ditt system så fungerar FNF(X) på samma sätt som FRE(X), utom att den första ger ett korrekt värde.

Ojdå! Det här blev en ganska lång utvikning, låt oss fortsätta med variabellagringen. Förutom variabel-bearbetningar i program finns det tre Basickommandon som påverkar pekarna.

CLR

Pekaren E sätts lika med pekare F dvs. inga strängar definierade.
Pekare C, D sätts lika med pekare B dvs. inga arrayer/variabler definierade.

Effekten är alltså att variabelminnet rensas, men utan att variablerna egentligen försvinner. En räddningsinstruktion av typ UNCLR som återställer dessa pekare skulle alltså vara fullt möjligt att konstruera.

NEW

Pekare E sätts lika med pekare F, som i CLR. Pekare B, C och D sätts lika med pekare A dvs. både Basicprogrammet och arrayer/variabler ser ut för interpretatorn som om de har upphört att existera, men finns egentligen kvar i minnet en egenskap som OLD använder sig av.

DIM

Dimensionerar arrayer och pekare beroende på arraytyp och storlek, och påverkar alltså pekare C, D och eventuellt E.

Låt oss nu studera hur de olika variabeltyperna lagras i variabeltabellen. Gemensamt för dem alla är att varje definition upptar 7 bytes.

Heltalsvariabler (Integer)

-32768 till 32767
Byte 1 Första tecken av variabelnamn +128 (högsta biten är satt)
Byte 2 Andra tecken av variabelnamn +128 (om inget andra tecken, 128)
Byte 3 Högbytet
Byte 4 Lågbytet
Byte 5-7 innehåller nollor
Observera att högbytet av heltalet ligger före lågbytet.
Exempel:
Antag att vi har raden:
10 A1%=7000

I byte 1 ligger då ASCII-koden för "A" 65, plus 128, dvs. 65+128=193.

I byte 2 ligger ASCII-koden för "1" 49, plus 128, dvs. 49+128=177.

Byte 3 ska innehålla högbytet av 7000 dvs. 7000/256=27. Byte 4 innehåller lågbytet av 7000 alltså 7000-256x27=88. Byte 5, 6 och 7 innehåller alla nollor.

Det är lätt att förstå hur Basicinterpretatorn förstår att variabler är heltal ty denna har den mest signifikanta biten satt i båda sina variabelnamn's byte.

Flyttalsvariabler (Floating point)

Byte 1 Första tecken av variabelnamn med MSB-biten nollställd.
Byte 2 Andra tecken av variabelnamn (0 om inget) med MSB-biten nollställd.
Byte 3 Binär exponent +128.
Byte 4-7 MANTISSA. MSB-biten i första byten används för att indikera Mantissan's tecken, 1=negativ Mantissa, 0=positiv Mantissa.
Observera! att i flyttals-ackumulatorn skiljer sig lagringsformatet och har här formen:

FAC	AFAC	
97	105	exponent
98	106	Mantissa
99	107	Mantissa
100	108	Mantissa
101	109	Mantissa
102	110	MSB innehåller tecken för Mantissa 1=negativ 0=positiv

Exempel:
Antag att vi definierar flyttalsvariabeln 10 A1=7000
på binär form har denna utseendet 00011011 01011000.

Punkten efter uttrycket är den s.k. RADIX-PUNKTEN, som kan jämföras med decimalpunkten i det decimala systemet. (jmf. 7000.0000).
Detta uttryck kan nu skrivas på formen 00011011 01011000.$x2^0$
(jmf. $7000.0000x10^0$)

För att alla flyttal ska få ett enhetligt utseende flytar vi nu decimalpunkten så att den ligger före den första ettan:
.11011010 11000000$x2^{13}$
(jmf. $0.7000x10^4$=7000)

Till slut fyller vi på med nollor i slutet så att vi får formen

Byte 2	11011010	218
Byte 3	11000000	192
Byte 4	00000000	0
Byte 5	00000000	0

Exponenten som ska ligga i byte 1 är här alltså 13, men eftersom den är positiv så ska 128 adderas till värdet dvs. Byte 1=13+128=141

Mantissan's tecken ska ligga i MSB-biten på byte 6 och eftersom vi har ett positivt tal så är nu denna bit nollställd. Dvs. om vi antar att vårt

forts. nästa sida

tal ligger i flyttals-ackumulator 1, FAC, så ser innehållet ut så här:

97	141	exponent+128
98	218	
99	192	Mantissa
100	0	
101	0	
102	0	sign

Nu är det dock inte så lätt som att det är denna representation som lagra i variabelområdet. Detta är ganska naturligt, eftersom vi slösar med minnesutrymme genom att lagra byte 6 (tecknet) där vi endast har en bit som ger information. Om vi tänker efter vad normaliseringsprocessen innebär så inser vi snart att den högsta biten i Mantissan (biten längst till vänster) alltid är satt. Eftersom vi vet att detta alltid är fallet kan vi använda denna bit till att ange tecknet på Mantissan dvs. i minnet lagrar vi aldrig byte 6, utan låter den högsta biten i Mantissan representera tecknet.

Låt oss se hur vår ansättning (A1 = 7000), lagras i variabelområdet:
Byte 1 ASCII-koden för "A" dvs. 65
Byte 2 ASCII-koden för "1" dvs. 49
Byte 3 exponenten + 128 dvs. 141
Byte 4 218-128=90 talet är positivt dvs. högsta biten är nollad
Byte 5 192
Byte 6 0 återstoden av Mantissan
Byte 7 0
Det här blev ett "ganska" långt exempel men låt oss fortsätta.

Strängvariabler

Byte 1 Första tecken av variabelnamn i ASCII-koden
Byte 2 Andra tecken av variabelnamn i ASCII +128 (128 om inget andra tecken)
Byte 3 Strängens längd (0-255)
Byte 4 Pekare till strängen (lågbyte)
Byte 5 Pekare till strängen (högbyte)
Byte 6, 7 Nollad

Pekarna i byte 4 och 5 pekar på strängområdet eller in i Basictexten, där själva lagringen av strängen sker.

Exempel:
Antag att
10 A$ ="HEMDATOR"
exekveras
Byte 1 Innehåller då 65 dvs. ASCII-koden för "A"
Byte 2 Innehåller 128 eftersom variabelnamnet endast består av ett tecken.
Byte 3 Innehåller 8 vilket är längden av "HEMDATOR"
Byte 4, 5 Innehåller en pekare som pekar på "H" i "HEMDATOR", på den position där den ligger lagrad.

Funktionsdefinitioner

När vi gör en definition av typ:
Def. FNX(X)=X^2+2*X-28
så måste även den lagras någonstans. Formatet är det som följer och ligger i den normala variabelarean.
Byte 1 Funktionsnamnets första tecken (här X) + 128
Byte 2 Funkionsnamnets andra tecken eller 0 (här 0)
Byte 3, 4 Pekare i låg/hög byte format som pekar in på definitionen av funktionen in i Basic-texten
Byte5, 6 Pekare till variabel i hög/låg byte format (i ovanstående fall X) (OBS! pekar på definitionen ej namnet).
Byte 7 Initialvärde för variabel (alltid ASCII-koden för första tecknet i variabelnamnet)

Till sist tittar vi på hur arrayer lagras. Att dessa lagras med början på den position dit pekare C pekar (arrayområdet) är väl helt klart. Låt oss studera formatet:
Byte 1, 2 Har samma format som för de enklare variablerna och innehåller arraynamnet, de möjliga typerna är
1. Heltalsarrayer,
2. Flyttalsarrayer
3. Strängarrayer
Byte 3, 4 Innehåller den totala längden av arrayen. Observera dock att strängen inte ligger i arrayen utan ligger lagrad i strängvariabelområdet eller i Basic-texten och själva arrayen innehåller bara pekare till strängarna.
Byte 5 Anger arrayens dimension 1 för t.ex. X(i), 2 för t.ex. X(i, j) etc.
Byte 6, 7 Här ligger par av värden som innehåller respektive dimensions storlek + 1 (dimensionen börjar ju räknas från 0) i hög/låg format med den sista dimensionen först t.ex. skulle vi i fallet X(a, b, c) byte 6, 7 innehålla a+1, byte 8, 9 b+1, byte 10, 11 c+1.

Efter detta ligger data eller pekarna i arrayen. För heltalen består varje element av ett hög/låg byte dvs. de upptar vardera 2 byte. För flyttal har vi 5 bytes för Mantissa och exponent. Strängen har 3 bytes bestående av en count och en pekare på låg/hög formatet som pekar till den aktuella strängen. Låt oss titta på två exempel av lagring:

Ex. 1
A(2,2) lagras i ordningen:
(0,0), (0,1), (2,0), (0,1), (1,1), (2,1), (0,2), (1,2), (2,2)

Ex. 2
A(1,1,2) lagras i ordningen:
(0,0,0), (1,0,0), (0,1,0), (1,1,0), (0,1,1), (1,0,1), (0,1,1), (1,1,1), (0,0,2), (1,0,2)

Ett allmänt utryck för att bestämma elementets position för en array på formen X(a, b, c) är:
a+b x(l+dim_1) + c x (l+dim_2) x (l+dim_2)
c=0 för två dimensionella arrayer.

För att beräkna längden av en given array använder vi:
5+2 x antal dimensioner + (dim_1+1) x (dim_2+1) x .....
x (dim_N+1) x 2, 3 eller 5
där 2, 3 eller 5 beror på variabeltyp.
Låt oss se på två exempel:
Ex. 1
Strängarrayen A$(500) definieras från början så att varje sträng får längden 10 bytes dvs. det antal bytes som tas upp är:
5+2+50001x3+500x10=6510

Ex. 2
För heltals arrayen A%(50,50) som totalt innehåller 51x51=2601 heltal blir det totala antalet bytes som återgår:
5+2x2+51x51x2=5211 bytes

Så hoppas jag att du fått en litet mer ingående uppfattning om hur din maskin lagrar sina variabler. Hoppas också att artikeln har givit dig några impulser till nya härliga programideer. Är det något annat som du undrar över eller har kommentarer till denna artikel så skriv en rad.

Åke Hedman

Bidrag från läsarna

Här har nu den listige läsaren ytterligare ett musikprogram, som kan användas för att producera alla julsånger och andra visor, som kommer att spelas och sjungas under de kommande helgdagarna!

Björn Hovdell på Ingarö hade läst att vi fått dåligt med material från våra läsare, så han skyndade sig att sända in två rejäla program. Björn hoppas att många får glädje av programmen, som går att köra på oexpanderad VIC-20. Det ena programmet är en omvandling av VICs tangentbord till en orgel med ett vibrerande ljud.

När man kör programmet frågar datorn först efter en övre resp. undre gräns för hur mycket ljudet skall vibrera. Volymen kan få värden mellan 0 och 15 och om man väljer den övre gränsen till 15 och den undre till 8, kommer volymen att svänga fram och tillbaka mellan 15 och 8 hela tiden. Om stabilt ljud önskas väljer man övre gräns till 15 och undre till 14.

Nedersta tangentraden innehåller den lägsta oktaven, mellersta raden tangenter den mittre oktaven och tredje raden den högsta oktaven. Sifferraden ger övertoner till den översta raden. Normalt är det den låga tongeneratorn som hörs, men om man håller (SHIFT) nere så aktiveras den mittre tongeneratorn och om COMMODORE-tangenten hålls nere så sätts den översta tongeneratorn i aktion.



```
5 PRINT"■■■JAG LADDAR              MASKINKODSPROGRAMMET"
7 PRINT"■■HAV TALAMOD !!!!!"
10 POKE52,28:POKE56,28
20 L=7168
30 READA$:IFA$="*"THEN80
40 C=LEN(A$):A=ASC(A$)-48:B=ASC(RIGHT$(A$,1))-48
50 IFA>16THENA=A-7
60 IFB>16THENB=B-7
70 N=A*16+B:POKEL,N:L=L+1:GOTO30
80 L=7424:Q=Q+1:IFQ=2THEN200
85 PRINT"■■JAG LADDAR TON-TABELL"
90 GOTO30
95 REM PROGRAMM
100 DATA18,D8,78,A9,2F,8D,14,03,A9,1C,8D,15,03,58,60,18,D8,A9,0F,8D,0E,90,A6,C5,
BD
110 DATA00,1D,8D,0A,90,A9,0A,6D,8D,02,8D,1C,1C,A5,C5,38,E9,20,18,D0,E8,60,18,D8,
AD
120 DATAFF,1D,F0,17,EE,0E,90,AD,0E,90,38,E9,0F,10,F0,03,4C,BF,EA,A9,00,0D,FF,1D,
4C
130 DATABF,EA,CE,0E,90,AD,0E,90,38,E9,0E,18,F0,03,4C,BF,EA,A9,01,8D,FF,1D,4C,BF,
EA,*
140 REM TON-TABELL
150 DATADD,E4,E8,ED,F0,F4,00,00,00,E3,E7,EC,EF,F3,F7,00,00,BF,CC,D5,DD,E3,02,00,
00
160 DATA00,8D,9F,B3,BF,CC,00,00,80,99,AA,BB,C6,00,00,00,C6,CF,D9,DF,E6,EA,00,DF,
E6
170 DATAEA,EE,F1,F5,F9,00,E1,E7,EB,EF,F2,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,*
200 SYS7168
210 POKE36874,230:PRINT"■"
220 FORTR=0TO20:GETT$:NEXT:PRINT"■■■ANGE OVRE GRANS FOR   SVANGNINGEN (1-15)":IN
PUTO
230 PRINT"■■■ANGE UNDRE GRANSEN FORSVANGNINGEN":INPUTU
240 POKE7230,O:POKE7253,U
250 PRINT"■■■M  NOJD"
260 IFPEEK(197)=28THENPRINT"■TTTTTTTTTTTTTTT■":GOTO220
270 IFPEEK(197)=20THEN300
280 GOTO260
300 PRINT"■■■NU GAR DET BRA ATT    SPELA PA TANGENTBORDET"
310 PRINT"■■TRYCK PA SPACE OM DU  VILL ANDRA SVANGNINGEN"
312 PRINT"■'SHIFT' OCH COMMODORE- TANGENTEN GER EN RESPTVA OKTAVER HOGRE LJUD"
320 SYS7183
322 POKE36879,27:POKE36865,38:POKE36864,12
330 PRINT"■":POKE36874,230:FORGR=0TO30:GETT$:NEXT:GOTO220

READY.
```

PUNK(T) - PLOTT

Det andra programmet är ett BASIC-program, som låter dig styra en prick på skärmen och som lämnar kvar en kopia av sig själv när man trycker på (MELLANSLAG) (SPACE). Sedan kan man gå vidare och plotta punkter tills man har en figur. Trycker man då på (F1) räknar datorn ut vilka värden varje rad i figuren har och visar även hur figuren ser ut i den storlek den kommer att få på bildskärmen om man använder de erhållna värdena i en DATA-sats.

Styrfunktioner:

A	= upp
Z	= ner
DEL	= höger
HOME	= vänster
F1	= visa figur + värden

```
0 PRINT"■"
2 FORX=0TO511:POKE7168+X,
  PEEK(32768+X):NEXT
4 POKE36869,255
9 C=30720
10 REMPOKE36879,122
20 FORX=6TO15
30 FORY=7812TO8010STEP22
40 POKEX+Y,35:POKEX+Y+C,5
50 NEXT:NEXT
60 FORX=7TO14
70 FORY=7834TO7988STEP22
80 POKEX+Y,32:POKEX+Y+C,0
90 NEXT:NEXT
100 REM PLOTTA
120 X=7841:A=32
130 R=0:P=PEEK(197)
140 IFP=7THENR=1
150 IFP=62THENR=-1
160 IFP=17THENR=-22
170 IFP=33THENR=22
180 IFPEEK(X+R)=35THENR=0
190 POKEX,A:X=X+R
195 IFP=32THENPOKEX,42
197 IFP=39THENPOKEX,32:GOTO300
200 A=PEEK(X):POKEX,42
210 GOTO130
300 PRINT"■■■■■■■"
305 B=128:U=1
310 FORY=7841TO7995STEP22
315 B=128
320 FORX=0TO7
330 IFPEEK(X+Y)=42THENA(U)=A(U)+B
340 B=B/2
350 NEXT
360 PRINTTAB(16)A(U):U=U+1
370 NEXT
380 PRINT"■■■■■■VERKLIG STORLEK"
390 FORX=1TO8:Z=Z+1
400 POKE7167+X,A(Z)
410 NEXT
420 POKE8130,0:POKE38850,0
430 IFPEEK(197)=64THEN430
435 FORJ=1TO8:A(J)=0:NEXT
440 PRINT"■":Z=0:GOTO4
```

Mer än ett vanligt kreditkort!

Nu finns ett nytt skräddarsytt kontokort, för Dig som vill hänga med i datorutvecklingen – QVIC-KONTO.
Med QVIC-KONTO kan du, när du vill, köpa VIC-datorer och alla datortillbehör du behöver.

QVIC-KONTO innebär:

- **Ingen handpenning**
- **Fasta månadsbetalningar**
- **Rabatt på VIC-rapport-prenumeration**
- **Gratis information om nyheter**
- **Specialerbjudanden**



PRENUMERERA!



VIC-rapport är specialtidningen för dig, som har en VIC-20 eller VIC-64.

Den kommer 10 gånger under 1984 med nyheter,tips,programexempel mm och ger dig alla möjligheter att utnyttja din dator på bästa möjliga sätt.

Tidningen säljs genom Pressbyrån och i alla butiker som säljer VIC-datorer. Den kostar 15:-/nr (utom för två dubbelnummer/år som är prishöjda och dubbelt innehållsrika). Du som prenumererar får alla nummer under ett år för endast 120:-.

Ur innehållet kan nämnas:
Nybörjarskolan
Frågor och svar
Programtips
Bok- och spelrecensioner
Programlistningar
Artiklar om användningsområden
Reportage
Undervisningsnytt
Nyhetstips
Information om tillbehör
Kurser i programmering, bl.a. i COMAL, Assembler och i Strukturerad BASIC
Gratisannonser för läsarna

prenumerera nu, före nyår!

Vi lottar ut 100 st T-shirts bland de prenumerationer, som kommit in på vårt postgirokonto **84 646-9** senast den 31/12 1983! Glöm inte skriva namn och adress på talongen samt att ange vilken storlek du skulle vilja ha på din T-shirt, om du vinner en. (De storlekar vi har är: S, small = liten, M, medium = mellan, L, large = stor och XL, eXtra Large = extrastor).

VIC-rapport utges av handic PRESS ab, Box 42054, 126 12 Stockholm.
Redaktion: Gustavsbergsgatan 8, 431 37 Mölndal. Tel 031-27 62 75.

Gör egna tecken på VIC-64

I en artikel i förra numret av VIC-rapport (s. 14-21) beskrevs video-chippet och hur man kan kontrollera det på olika sätt. Det nämndes då att man kan ändra en pekare till teckengeneratorn, dvs. till det ställe där video-chippet hämtar information om hur tecknen skall se ut. Jag skall här beskriva hur man kan använda detta till att skapa sina egna tecken. Förutom denna pekare är det en del andra saker man bör känna till, och jag kommer därför att repetera dessa (ni som läste artikeln i förra numret får stå ut med denna repetition).

Vilket block används?

Video-chippet kan bara hämta information som ligger inom ett och samma 16K-block. Vilket av fyra möjliga block som video-chippet "ser" bestäms av de två minst signifikanta bitarna (bit 0-1) i adress 56576. I tabell 1 kan du se vilket block som används vid olika värden på dessa bitar. Genom att ändra dessa bitar kan man alltså byta mellan de olika blocken. Följande BASIC-rad ändrar dessa bitar (A är värdet på bitarna): POKE 56576, (PEEK(56576) AND 252) OR A. Först ANDas förra innehållet med 252 (11111100, binärt) vilket "nollar" de aktuella bitarna medan övriga bitar inte ändras, och sedan sätts bitarna genom att ORa med A.

Placering av skärmminnet

I adress 53272 finns en pekare som talar om var någonstans i det aktuella 16K-blocket bildskärmsminnet ligger. För att få fram adressen till bildminnet tar man värdet av bitarna 4-7 (de fyra mest signifikanta bitarna), multiplicerar med 1024 (1K) och lägger till 16Kblockets startadress. Tabell 2 visar var bildminnet ligger inom aktuellt 16K-block för olika värden på bitarna. Man kan ändra värdet på dessa bitar med följande rad: POKE 53272, (PEEK (53272) AND 15) OR A*16. A är värdet som bitarna skall ha (0-15).

Det räcker emellertid inte att tala om för video-chippet var det skall hämta data för att skriva ut skärmen. Man måste också tala om för BASICens skärmeditor var den skall arbeta med skärmen någonstans. Den pekare som bestämmer detta ligger i adress 648, och pekar på vilken "sida" i minnet som används. Eftersom en "sida" är 256 bytes fås startadressen genom att multiplicera innehållet med 256. Denna pekare är absolut, och beror alltså inte på vilket block video-chippet ser. Följande rad sätter pekaren (A är startadressen till bildminnet): POKE 648, A/256.

Vad är en teckengenerator?

Bilskärmsminnet består av 1000 bytes, där varje byte innehåller en kod som motsvaras av ett bestämt tecken. För att ett tecken skall kunna skrivas ut, måste det finnas information om hur tecknet ser ut. Den area i minnet där denna information finns kallas för teckengenerator. Ett tecken består av en matris om 8x8 punkter som kan vara tända eller släckta. I teckengeneratorn beskrivs därför ett tecken med 8 bytes, i vilka varje bit (en byte består av 8 bitar) motsvarar en punkt i teckenmatrisen. Om en bit är satt (lika med 1) betyder det att motsvarande punkt skall vara tänd, och om biten är lika med 0 skall motsvarande punkt vara släckt. Eftersom det finns två olika teckenuppsättningar (stora bokstäver/grafik och små/stora bokstäver) som vardera innehåller 256 olika tecken, måste teckengeneratorn bestå av 2*8*256 = 4096 bytes för att alla tecken skall kunna skrivas ut. En sådan teckengenerator finns också, den ligger i ROM och startar på adress 53248 (D000 hexadecimalt). I tabell 3 kan du se vilka delar av ROM-teckengeneratorn som innehåller vilka tecken. I denna teckengenerator ligger tecknen definierade i ordning efter bildkoden. Först ligger de 8 bytes som definierar tecken 0 (@), sedan kommer 8 bytes som definierar tecken 1 (A), osv. Som exempel kan vi ta och titta på hur informationen för "A" ser ut. I figur 1 kan du se vilka olika bitar som är tända, och vilka värden detta medför för de 8 bytes som tecknet består av. Jag skall senare beskriva hur man gör för att läsa i ROMteckengeneratorn.

Placering av teckengeneratorn

Teckengeneratorns placering bestäms, liksom bildminnets, av en pekare i adress 53272, men nu är det de fyra minst signifikanta bitarna (bitarna 0-3) som gör jobbet. Samma regler som för att ta fram adressen till bildminnet gäller också här (man multiplicerar värdet av bitarna med 1024 och lägger till 16K-blockets startadress). Observera dock att bit 0 inte används, utan antas alltid ha värdet 0 (även om den är satt till 1 när du PEEKar på adressen). För att ändra pekaren kan man använda följande rad (där A som vanligt är värdet på bitarna): POKE 53272, (PEEK(53272) AND 240) OR A. Tabell 4 visar teckengeneratorns placering för olika värden på pekaren.

En viktig sak att ta hänsyn till, är att när video-chippet ser block 0 eller block 2 (se tabell 1) ligger ROM-teckengeneratorn på startadressen för blocket + 4096. Att den ligger där betyder bara att video-chippet ser den där. Det är också bara för video-chippet den ligger där, om man vill läsa i den får man läsa på adress 53248 och framåt där den "egentligen" ligger. (Mer om hur man kan läsa i teckengeneratorn senare). Resultatet av att video-chippet ser ROM-teckengeneratorn på denna adress, är att om bitarna 0-3 i 53272 pekar dit kommer data för tecken att hämtas från ROM-teckengeneratorn och inte från RAM-adresserna. För att bättre belysa detta kan vi undersöka hur det ser ut när datorn slås på. Bitarna 0-1 i 56576 är då lika med 3 (11, binärt), och video-chippet ser adresserna 0-16383 (block 0). Bitarna 0-3 i 53272 har värdet 4, vilket betyder att teckengeneratorn ligger på adresserna 4096-8191 (0+4*1024 =

4096). Enligt vad vi diskuterade ovan, ser video-chippet ROMteckengeneratorn just på denna adress inom block 0 och 2, och teckendata kommer alltså att hämtas från denna.

Skriv egna tecken

Om man sätter pekaren till teckengeneratorn till att peka på en area där video-chippet inte ser ROM-teckengeneratorn, kommer data till de olika tecknen att hämtas från den RAM-area som pekas på. Detta medför att man kan definiera sina egna tecken. Man kan också kopiera ROM-teckengeneratorn (eller en del av den) till RAM-arean, och endast ändra en del tecken. ROM-teckengeneratorn ligger på adresserna 53248-57343. Men om du PEEKar där märker du snart att det inte är teckeninformation du läser. "Över" teckengeneratorn ligger nämligen de olika i/o-chippens kontrollregister. Om man skall läsa i ROM-teckengeneratorn skall man göra på följande sätt:

1. Stäng av all I/O-interrupt. Anledningen till att man måste göra det är att I/O-chippens kontrollregister måste switchas bort för att man skall kunna "se" teckengeneratorn (se punkt 2). Interrupten stänger man av genom att nolla bit 0 i 56334. Följande BASIC-rad gör detta: POKE 56334, PEEK(56334) AND 254.

2. Switcha bort I/O så att teckengeneratorn "syns". Detta görs genom att nolla bit 2 i adress 1. POKE 1, PEEEK(1) AND 251 fixar det.

3. Nu kan man läsa den teckeninformation man vill, och kopiera den till en RAM-area (se tabell 3 för var de olika tecknen ligger). Adressen till ett tecken fås av startadressen + teckenkod * 8 (i det här fallet, när man läser från ROM-teckengeneratorn, är startadressen lika med 53248).

4. Switcha tillbaks I/O igenom genom att sätta bit 2 i adress 1. POKE 1, PEEK(1) OR 4 gör detta.

5. Sätt slutligen på interrupten igen genom att sätta bit 0 i 56334. Det gör man med: POKE 56334, PEEK (56334) OR 1. Se program 1 och 2 för exempel på hur man läser.

När man sedan har kopierat tecknen till en RAMarea, kan man ändra i dem som man vill. I program 1 kopieras de första 64 tecknen till adress 14336 och framåt. Sedan ändras lika-med-tecknet (=) till en höger-pil. Figur 2 visar vilka koder som använts för att åstadkomma en pil. Startadressen för tecknet får man genom att multiplicera bildkoden för tecknet med 8 och lägga till startadressen för teckengeneratorn. Slutligen, på rad 250, sätts pekaren till teckengeneratorn så den pekar på adress 14336 där den "nya" teckengeneratorn ligger. Observera att minnes-top-pekaren sätts till 14336 för att skydda arean från att bli överskriven av BASIC. När du kört programmet kommer en höger-pil att skrivas varje gång du trycker på lika-med-tangenten. Alla lika-medtecken i programlistningen kommer också att ha ändrats till höger-pilar. Detta är emellertid bara en synvilla, det enda som programmet ändrar är hur tecknet ser ut på skärmen. För BASIC fungerar det fortfarande som ett =, även om det inte ser ut så. Pröva själv!

Å, Ä och Ö

Program 2 gör att du får å, ä och ö på din 64a. De tangenter som används är: höger hakparentes (Å), vänster hakparentes (Ä) och pund-tecknet (Ö). Anledningen till att jag valt dessa tecken är att VIC-skrivaren använder dem. En programbeskrivning följer.

Rad 10
variabel BE innehåller startadressen för det 16K-block som video-chippet ser. Om du vill ändra placeringen av teckengeneratorn behöver du bara ändra denna variabel (den måste vara lika med någon av startadresserna i tabell 1). Om du lägger den i minne som också används av BASIC, måste du också skydda minnet (se program 1)

Rad 30-40
skärmen rensas och ett meddelande skrivs ut (för att man inte skall tro att datorn har dött).

Rad 100-160
på dessa rader kopieras hela ROM-teckengeneratorn till en RAM-area (samma teknik som i program 1 används).

Rad 170-180
vilket 16K-block video-chippet skall se bestäms och bildminnets skärmsi-

Rad 190
här bestäms att bildminnet alltid startar på första adressen i aktuellt 16K-block, och att teckengeneratorn börjar 12*4096 = 12288 bytes fram.

Rad 210
rensa den nya skärmen, samt skriv ut meddelande.

Rad 290-370
först läses antalet tecken som skall ändras in, för varje tecken läses teckenkoden in, och slutligen POKEas 8 nya värden in i teckengeneratorn. Reverse av tecknen POKEas också in (för att få reverse av ett tecken vänder man naturligtvis på alla bitarna i bitmönstret). De tre teckenkoder som är större än 255 medför att tecken i andra teckenuppsättningen ändras (verklig kod fås genom att subtrahera 255).

Rad 400-470
första DATA-satsen innehåller antalet tecken som skall ändras, i det här fallet 6. Följande DATA-satser innehåller först teckenkoden (bildkoden) och sedan de 8 värden som bestämmer hur tecknet skall se ut.

För att inget minne skall gå förlorat läggs den nya teckengeneratorn i det RAM som ligger under operativsystemets ROM på adress 61440 och framåt. (Video-chippet är nämligen så smart att det fattar att det skall läsa i RAM, och inte i ROMmet som ligger ovanpå.) Om teckengeneratorn skall läggas däruppe måste man sätta pekaren i 56576 till att peka på block 3, och skärmminnet måste då också flyttas. Detta sker också som beskrivet ovan. När man har kört program 2, är skärmminnets startadress 32768. Jag hoppas att programbeskrivningen ovan, samt REM-satserna i programmet, räcker för att du skall förstå programmet. Om så inte är fallet föreslår jag att du jobbar dig igenom det steg för steg. Jag bör kanske tillägga att tecknen inte är så väldigt vackra, gör gärna om dem (det är bara att ändra i DATA-satserna).

Experimentera själv

Det bästa sättet att lära sig vilka olika möjligheter som det innebär att kunna definiera sina egna tecken, är att experimentera själv. Du behöver

forts. nästa sida

ju inte hålla dig till ett tecken, utan kan definiera flera stycken som bildar en större figur när de skrivs bredvid varandra. Möjligheterna är många. Om det är något du undrar över (som inte förklarats tillräckligt bra) skriv gärna och fråga. Om du skriver något roligt program som använder teknikerna ovan, eller kommer på något intressant, skriv gärna till VIC-rapport så fler får ta del av det. Det enda krav vi ställer på inskickade program är att de skickas hit på kassettband eller skiva!

MOGF

Tabell 1. Pekare till 16K-block (bitarna 0-1 i 56575)

Bitvärde	Blocknr.	RAM-area
0 (00)	3	49152-65535
1 (01)	2	32768-49151 *
2 (10)	1	16384-32767
3 (11)	0	0-16383 (normalt) *

*För dessa block ser video-chippet ROM-teckengeneratorn på adresserna 4096-8191 (plus startadressen).

Tabell 2. Pekare till bildminnet (bitarna 4-7 i 53272).

Bitvärde	Startadress
0 (0000)	0
1 (0001)	1024
2 (0010)	2048
3 (0011)	3072
4 (0100)	4096
5 (0101)	5120
6 (0110)	6144
7 (0111)	7168
8 (1000)	8192
9 (1001)	9216
10 (1010)	10240
11 (1011)	11264
12 (1100)	12288
13 (1101)	13312
14 (1110)	14336
15 (1111)	15360

Tabell 3. Uppdelning av ROM-teckengeneratorn.

Adresser	Innehåll
53248-53759	Stora bokstäver
53760-54271	Grafiska tecken
54272-54783	Reverse stora bokstäver
54784-55295	Reverse grafiska tecken
55296-55807	Små bokstäver
55808-56319	Stora bokstäver/grafiska tecken
56320-56831	Reverse små bokstäver
56832-57344	Reverse stora bokstäver/grafiska tecken

Tabell 4. Pekare till teckengeneratorn (bitarna 0-3 i 53272).

Bitvärde	Startadress
0 (000x)	0
2 (001x)	2048
4 (010x)	4096 (normalt) *
6 (011x)	6144
8 (100x)	8192
10 (101x)	10240
12 (110x)	12288
14 (111x)	14336

*Video-chippet ser ROM-teckengeneratorn från denna startadress i block 0 och 2.

Figur 1. Bitmönster för A.

	7	6	5	4	3	2	1	0	Binärt	Decimalt
0				X	X				00011000	24
1			X	X	X	X			00111100	60
2		X	X			X	X		01100110	102
3		X	X	X	X	X	X		01111110	126
4		X	X			X	X		01100110	102
5		X	X			X	X		01100110	102
6		X	X			X	X		01100110	102
7									00000000	0

Figur 2. Bitmönster för höger-pil.

	7	6	5	4	3	2	1	0	Binärt	Decimalt
0									00000000	0
1					X				00001000	8
2					X	X			00001100	12
3		X	X	X	X	X	X		01111110	126
4		X	X	X	X	X	X		01111110	126
5					X	X			00001100	12
6					X				00001000	8
7									00000000	0

Program 1

```
100 POKE 52,56: POKE 56,56 : CLR   : REM SKYDDA MINNE DAR TECKEN SKALL LIGGA
110
120 POKE 56334,PEEK(56334) AND 254 : REM STANG AV INTERRUPT
130 POKE 1,PEEK(1) AND 251         : REM SWITCHA IN TECKENGENERATORN
140 FOR I=0 TO 504                 : REM KOPIERA DE 64 FORSTA TECKNEN
150 : POKE 14336+I,PEEK(53248+I)
160 NEXT                           : REM SWITCHA BORT TECKENGENERATORN IGEN
170 POKE 1,PEEK(1) OR 4            : REM SATT PA INTERRUPT IGEN
180 POKE 56334,PEEK(56334) OR 1
190                                : REM STARTADDRESS FOR TECKEN NR 61 (=)
200 AD=14336+61*8                  : REM ANDRA 8 BYTE
210 FOR I=AD TO AD+7
220 : READ A : POKE I,A
230 NEXT
240
250 POKE 53272,(PEEK(53272) AND 240) OR 14
260
300 DATA 0,8,12,126,126,12,8,0

READY.
```

Program 2

```
10 BE=49152 : REM FÖRSTA ADRESS I AKTUELLT MINNESBLOCK (FOR VIDEOCHIPPET)
20
30 PRINT CHR$(147)   : REM RENSA SKÄRMEN
40 PRINT "TECKENGENERATORN KOPIERAS TILL RAM, DET TAR EN LITEN STUND...."
90
100 POKE 56334,PEEK(56334) AND 254     : REM STÄNG AV INTERRUPT
110 POKE 1,PEEK(1) AND 251             : REM SWITCHA IN TECKENGENERATORN
120 FOR I=0 TO 4095                    : REM KOPIERA ALLA TECKEN TILL AREA I RAM
130 : POKE BE+12288+I,PEEK(53248+I)    : REM LÄGG TECKNEN HÖGST UPP I BLOCKET
140 NEXT
150 POKE 1,PEEK(1) OR 4                : REM SWITCHA IN I/O IGEN
160 POKE 56334,PEEK(56334) OR 1        : REM SÄTT PÅ INTERRUPT
170 POKE 56576,(PEEK(56576) AND 252) OR (3-BE/16384) : REM BESTÄM VIDEO-BLOCK
180 POKE 648,BE/256  : REM SIDA I MINNET FÖR SKÄRMEN (FÖR SKÄRMEDITORN)
190 POKE 53272,12    : REM PLACERING AV SKÄRM OCH TECKENGENERATOR
200
210 PRINT CHR$(147): PRINT "NU LÄGGER JAG TILL Å,Ä OCH Ö."
270
280
290 READ AN : REM ANTAL TECKEN ATT ÄNDRA
300 FOR T=1 TO AN                      : REM TECKENKOD
310 : READ TK                          : REM POSITION I TECKENGENERATORN
320 : P=BE+12288+8*TK                  : REM LÄGG IN 8 NYA BYTE
330 : FOR P=P TO P+7
340 : : READ A: POKE P,A
350 : : POKE P+8*128, (NOT A) AND 255 : REM LÄGG IN REVERSE AV TECKNET
360 : NEXT
370 NEXT
380 PRINT: PRINT "FÄRDIG!!"
390
395
400 DATA 6 : REM ANTAL TECKEN ATT ÄNDRA
410
420 DATA  27, 102,24,60,102,126,102,102,0
430 DATA  28, 102,0,60,102,102,102,60,0
440 DATA  29, 24,36,60,102,126,102,102,0
450 DATA 283, 102,0,60,6,62,102,62,0
460 DATA 284, 0,102,0,60,102,102,60,0
470 DATA 285, 8,20,60,6,62,102,62,0

READY.
```

Kapten Storöra



En liten polisongprydd gubbe med kaptensmössa på huvudet och några sällsynt stora öron lär ha varit synlig i Stockholm, Askin, Horsens och Slough (i England).

När vi försökte få reda på mer om denne intressante figur, hände det sig att SAPU (Statens Avdelning för Polisong-Utveckling) fick höra talas om vårt sökande. Vi märkte det på att telefonerna började knäppa litet konstigt och att vi ibland hörde ekon av våra egna röster. Speciellt när vi talade med utlandet. Man gick så långt, att man t.o.m. ringde oss mitt i natten för att höra om vi möjligen var fullt sysselsatta då också. (Det var vi!) Visserligen försökte man kamouflera sin verksamhet med att låtsas vilja testa spelprogram, men vi vet nog vi...

Med hänsyn till SAPU:s verksamhet har vi därför fått skaffa ett hemligt telefonnummer som vi kan ringa ut från redaktionen på! Dessutom har vi skaffat ett nytt telefonnummer som går till den riktiga redaktionen. Om du lovar att inte avslöja det för alltför många så skall du få veta det här: 031-27 62 75!

Så småningom ebbade tydligen SAPU:s intresse för oss ut och det har bara varit tre samtal denna söndag. Skönt, tänkte vi, nu kan vi fortsätta och leta efter Kapten Storöra, för det var namnet vi gett den storörade gentlemannen. Man vi behövde inte leta länge, för häromdagen - det var en stormig förmiddag. Då kom brevbäraren släpande på ett ovanligt tjockt kuvert, som fastnade i redaktionsbrevlådan. När brevbäraren försvunnit smög vi oss ut och lyckades lossa kuvertet ur brevlådans järngrepp. Vi var mycket nyfikna vad som kunde finnas i detta tjocka kuvert.

Döm om vår förvåning, som det heter i alla bättre romaner, när vi fann att det var en rapport från just Kapten Storöra. Han förklarade att han avlyssnat SAPU:s telefoner och den vägen fått reda på att vi sökte honom. (Storöra avslöjade också att hans avlyssningsutrustning var otroligt enkel = hans två öron, och det kunde ju inte bli olagligförklarat av SAPU!). Vi har tänkt oss att - sedan vi rensat hans rapport från de allra värsta överdrifterna och det mesta av åtalbar natur - så låter vi kaptenen själv berätta:

Det är jag som är Kapten Storöra och det är jag som hör det mesta och rapporterar nästan allting jag hör.

Den här gången skall jag berätta vad som hände när jag smög runt bland alla maskinerna och programmerarna och folket som sprang som myror med papper i händerna och verkade så förfärligt sysselsatta ute i Askim. Det blev så jäktigt och ingen pratade med varandra så jag stolpade över till det andra företaget, dom sålde ljudutrustningar. Där var det lugnt och prydligt. Visserligen stod en dam och plockade ihop kataloger på ett ställe, men annars verkade det stillsamt. Då plötsligt rann det ut en massa papper ur Telefaxen och genast såg rummet ut som flustret på en bikupa. Man hade fått en leverans av apparater, som skulle distribueras omedelbart. Alla jobbade med allt och efter en stunds kutande var det återigen lugna gatan. Stämningen var skön och avspänd, fjärran från jäktet på det första stället jag kom till och där dom visst pysslade med datorer eller vad det var.

Eftersom lugnet inte gav mig några tillfällen att tjuvlyssna på intressant snack så drog jag mig tillbaka till myrstacken igen. Där kunde man höra hur man på alla vis försökte förklara för någon i telefon varför han inte fått leverans, på ett annat ställe varför han fått sånt som beställts för så förfärligt länge sedan att han glömt bort det själv och i en tredje telefon var det en stackars dam som försökte förklara att man på grund av datafel inte fått ut alla prenumeranternas tidningsexemplar i tid, ja, det var en kakafoni som knappt går att återge på en VIC-64 ens.

Man talade i ett hörn om det nya QVIC-kontot, som handlarna blivit så förtjusta i och som konsumenterna började ringa och fråga om. (Red.anm. Jodå, dom har t.o.m. ringt VIC-rapports redaktion - folk är ju galna i kontokort!)

I ett annat hörn diskuterade man just leveranssituationen för VIC-datorer till julen. Det var så stor efterfrågan fortfarande, att man hade svårigheter att tillfredsställa den. Och värre väntade man den skulle bli när VIC-MODEMET blev klart och folk kunde koppla upp sig till Teledata.

Där var det någon som plötsligt avbröt sig mitt i en mening och då spetsade jag de stora extra mycket. Vad hade han sagt? Jag letade i minnescellerna och kom till slut rätt: man skulle visst starta ett eget teleät. som VIC-ägare kunde koppla upp sig till och umgås med varandra genom. Men hur jag än ansträngde mig kunde jag inte få fram fler detaljer - inte heller tidpunkten för start.

Nu blev jag upptäckt och man ledde mig vänligt men bestämt till kaffebordet och tvingade i mig fyra koppar starkt kaffe under intensiv övervakning. Inte ens socker fick jag till! Efter en timme hade kaffet gjort sitt och jag blev tvungen att utgå. (Jag reste hem till mig och gamla Farmor på Svärdslundsvägen).

Nästa vecka befann jag mig i England och andades London-nånting, för mycket till luft var det ju inte. Jo, det luktade diesel förstås, men annars var vädret fint. När jag tröttnat på att gå i butiker på Portobello Road och besöka alla PUB-ar därför att mina pilar tog slut och butikerna var stängda så köpte jag en returbiljett från Paddington Station till Slough. Dit anlände jag mitt i lunchen och tog mig till världshuset White Horse, men inte sjöng dom där inte... Visserligen fanns det en sorts Josepha med vacker röst och snygg blus, som serverade mig en hot roll och en öl, men det var det närmaste en operett jag kom den gången.

När jag väl blev insläppt efter noggrann kontroll i entren fick jag en "barnsköterska" med mig för att inte gå vilse - sa dom. Det var väl för att jag inte skulle få se och höra för mycket kan jag tänka mig. Lite fick jag reda på i alla fall och däribland att den

portabla 64:an nu skall heta SX-10 och att man ännu inte bestämt sig för om den skall tillverkas med dubbla diskstationer och kallas DX-10. Man lär få veta't framåt jul om det blir nån DX-ing med Commodore-maskiner under granen. Programmen till VIC-20 och 64:an såldes genom den postorderavdelning som fanns, VIC-SOFT, och där man sålde både egna och andras program till alla kunder ute i landet direkt och i samarbete med sina återförsäljare. Man hade mycket stor framgång med denna verksamhet och det ryktades att något liknande skulle kanske starta i Sverige också. Det fick mig att spetsa utväxterna på sidan av huvudet igen och ställa några försåtliga frågor. Man svarade bl.a. att av alla de hundratals program som kom in till VIC-SOFT för testning före eventuell försäljning. så var det en tredjedel, som var säljbara. Men dessa var istället desto bättre. Man talade också om att man märkt en gradvis förbättring av program från fristående programmerare under den senaste tiden. Men man talade inte om när man fick fram COMAL-80 till 64:an, för det visste man inte. Det är en affär för Commodore/Danmark, sa man.

Ja vad kunde jag väl göra annat än att försöka ta mig till Danmark också? Där träffade jag en liten servitris på en trevlig dansk Kro, och sen var den dan slut. Nästa dag försökte jag igen, att komma in på Commodore Danmark's kontor, men då hade alla gått till lunch. Skam den som ger sig – efter lunchen, när alla är på gott humör av smörrebröd och danskt öl. då skall det väl gå lätt att komma in, tänkte jag.

Mycket riktigt, nu gick det bra. Så bra att jag t.o.m. fick se en, tänk det, se en och hålla i en diskett med COMAL-80 till 64:an!!! Äntligen. Så fick jag se en dansk handbok om COMAL på 64:an, skriven av Börge Christensen och dessutom hörde jag att man talade i telefon med en kille i Göteborg, som skulle översätta denna handledning till svenska! Här kan alltså hända saker i höst eller snarare i vinter. Men man märkte att jag hade ovanligt flaxande hudflikar på huvudets utsida och lät mig genomgå kaffebehandlingen även här. Samma resultat – hemresa.
Sen flög jag – tänka sig en sjöman till lufts – till Kgl Huvudkommunen och skulle besöka Mr G på Huvudkontoret. Honom missade jag, men fick träffa flera av hans närmaste under en paus i slitet med att hålla ihop alla tyglarna. För om man skall ta en hästterm så är det väl inte fyrspann utan snarare åttaspann som Mr G helst kör. Han kanske borde ställa upp vid körtävlingarna på Tjolöholm?

Men det var ju inte om detta jag skulle berätta utan om vad jag lyckades höra under ivrigt hoppande mellan vrårna på det gamla kontoret. Det första jag hörde var att alla nu förväntade sig förfärligt mycket av det nya kontoret, dit man trodde att man skulle få flytta under november. Visserligen hördes någon röst som trodde att "det var nog överbefolkat redan när vi kommer dit", men denna röst var det ingen som brydde sig om. En annan sak som jag bara råkade höra till hälften var att man just höll på att bygga upp en egen postorderverksamhet där man tänkte sig att sälja program och böcker och tillbehör så att återförsäljarna inte skulle behöva ligga med så mycket pengar bundna i lager av sådana produkter. Man hade något speciellt knep för att få återförsäljarna att gilla verksamheten, men vilket lyckades jag inte lura ut...

När jag sedan promenerade i sakta mak till orderavdelningen, kunde jag av förbiflygande ordersedlar se, att det verkade som om skolorna i mycket stor omfattning köper Commodore-datorer. Man har ju redan i dag på många ställen PET-ar och nu kommer 64:orna stort. Det verkar som om lärarna och skolledningarna kommit underfund med att det är en ruskans bra liten dator till ett ruskans bra litet pris! Och dessutom kommer ju COMAL-80 nu, det nya skolspråket just till 64:an. Jo, en sak jag glömde berätta från Danmark, man tror att en cartridge (plug-in) med den stora COMAL-80, version 2.0, kraftfull och fin, kommer i december!!!!

Kommer sedan nu bara ordentliga undervisningsprogram också, då får nog alla som tänker på sina compisar se upp!

Men nu måste jag rusa, båten går snart!

Er vän **Kapten STORÖRA**

DATA AID

presenterar

* * VIC-20 PROGRAM * *
* * VIC-64 PROGRAM * *

Brett programutbud inom:

- Statistik
- Matematik
- Geometri
- Lantmäteri
- Privatekonomi
- Spel
- Navigation
- Kalenderberäkning
- Chiffrering
- Resultatberäkning för sällskapsspel, tips, lotterier

Sänd efter kostnadsfri programförteckning!

DATA AID
Eriksgatan 14
521 00 Falköping

VIC-böcker!

Spela VIC!
av Tim Hartnell och Mark Ramshaw

Ca 75 tydliga spelprogramlistor för VIC-20. Bildskärmstexter på svenska. En översättning av böckerna *Zap Pow Boom* och Symphony For A Melancholy Computer i en volym. Pris 135:— inkl moms och frakt.

Mastercode för VIC-64
av David Lawrence och Mark England

Ett programvarupaket med bl a disassembler och assembler, här presenterat i form av kompletta, fylligt kommenterade listor. Maskinspråksrutiner för 14 nya BASIC-kommandon ingår. Utkommer i januari. Pris 145:— inkl moms och frakt.

Ring 018/32 05 75 och beställ!

applica

BOX 9014, 75009 UPPSALA 018/320575

"För tid och evighet..." KALENDERPROGRAM

Almanackor är ju oumbärliga hjälpmedel för att orientera sig i tillvaron. Felet med dem är att de bara gäller för ett år i taget. Den som vill ha en evighetskalender kan faktiskt programmera in en sådan på sin VIC, fast det blir förstås ett ganska vidlyftigt program.

Problemet är för det första att vi bytte "stil" år 1753, då vi fick den gregorianska kalendern i stället för den julianska (som alltjämt används i Sovjet och en del andra länder). På så sätt infördes en ny skottårsräkning bl.a. Det är ju skottåren som ställer till trassel i ordningsföljden, bl.a. när man ska räkna ut veckodag för ett visst datum.

Det finns olika sätt att lösa skottårsproblemet i ett kalenderprogram. Ett är att utgå från regeln att samma veckodag återkommer på samma datum vart 28:de år. Det stämmer så länge man bara håller sig inom ett århundrade. Överskrider man sekelgränserna så blir det fel.

Det program som presenteras här är i princip mycket enkelt. Det baserar sig nämligen på fenomenet, att året inte har jämt 365 dagar utan faktiskt omspänner 365,2425 dygn. Avkortat till 365,25 fås automatiskt ett tillägg för skottdagen vart fjärde år utan några krångliga IF-satser. De enda undantagen gäller sekelskiftena 1800 och 1900, som inte är skottår. Där hoppas skottårsmånaden februaris 29 dagar över med en IF-sats.

Programmet, som täcker alla år mellan 1753 och 2100 är ganska långsamt, eftersom alla räkningar sker från 1753. Men det är i stället så mycket säkrare.

När infaller påsken

Det andra programmet räknar ut på vilket datum påskdagen infaller under valfritt år. Det är ju som bekant lite olika med den saken. Det var på kyrkomötet i Nicea år 325 som det bestämdes att påsken skulle inträffa *första söndagen efter första fullmånen efter vårdagjämningen, dock inte senare än den 25 april.*

Det där kan ju låta lite avskräckande för en programmerare, men problemet är till största delen redan löst av den geniale tyske matematikern Karl Friedrich Gauss, som år 1800 konstruerade en formel för uträkning av påskterminen.

Den här formeln är ganska invecklad. Det tar tid att handskas med den för hand med block och penna, och med fickräknare går det inte mycket lättare. Däremot är den tämligen enkel att programmera på datorn.

Formeln lyder:
Önskat årtal divideras i tur och ordning med talen 19, 4 och 7. När det är gjort tar man *resterna* efter divisionerna och kallar dem förslagsvis A, B och C. Till 19*A lägger man ett tal, som vi kallar M och som är olika beroende på århundrade. Ligger årtalet före 1800, så är M=15. För åren 1800-1899 har M värdet 23 och för 1900 fram till 2099 är M=24.

19*A+M divideras med 30 och *resten* kallar vi D. Sen tar vi 2*B+4*C+

PET COMMODORE CENTER
Skandinaviens största datacenter för PET- och VIC-datorer
Vid Fridhemsplan (Hantverkargatan 87)
hittar Du PCC,
specialisterna på datorer från COMMODORE
KUNSKAP KVALITET KÖPTRYGGHET
PET COMMODORE CENTER AB
Hantverkargatan 87 112 38 Stockholm Tel. 08/54 01 35

VILKEN VECKODAG?

```
1 PRINT"□"
5 PRINT"*****VILKEN VECKODAG?*****"
10 INPUT "VILKET ÅR";A
11 INPUT "VILKEN MÅNAD";B
12 INPUT "VILKET DATUM";C
30 FOR I=1753 TO A
40 D=A-1753
50 S=D*365.25
60 IF D>47 THEN S=S-1
70 IF D>147 THEN S=S-1
75 S=INT(S)
80 NEXT I
90 FOR J=1 TO 12
100 READ D
110 IF J=B THEN 132
120 S=S+D
130 NEXT J
132 IF A=1800 OR A=1900 THEN 145
135 IF J=2 AND A/4=INT(A/4)THEN D=29
140 IF J>2 AND A/4=INT(A/4) THEN S=S+1
145 GOTO 150
150 DATA 31,28,31,30,31,30,31,31,30,31,30,31
152 PRINT:PRINT
155 PRINT"DEN";C;"/";B;"ÄR";A;"ÄR EN ";
160 S=S+C
165 IF C>D THENPRINT"LÄR DIG MÅNADERNA!"
170 S1=S-INT(S/7-0.1)*7
180 ON S1 GOSUB 201,202,203,204,205,206,207
190 END
201 PRINT"MÅNDAG":RETURN
202 PRINT"TISDAG":RETURN
203 PRINT"ONSDAG":RETURN
204 PRINT"TORSDAG":RETURN
205 PRINT"FREDAG":RETURN
206 PRINT"LØRDAG":RETURN
207 PRINT"SØNDAG":RETURN
```

6*D och lägger till ett tal N, som skiftar värde precis som M. Före 1800 är N=6, 1800-1899 är N=4 och mellan 1900 och 2099 har N värdet 5.

2*B+4*C+6*D+N ska sen divideras med 7. *Resten* kallar vi E. För att få påskdagens datum adderar man till sist D+E+22. Om talet man får är > =31 infaller påsken på givet datum i mars. Är talet 31 minskar man med 31 och får datum i april. Skulle det mot förmodan inträffa att svaret blir den 26 april, vilket det följande programmet inte är garderat mot, så infaller påskdagen i verkligheten en vecka tidigare.

Conny Jonsson

NÄR INFALLER PÅSKEN?

```
1 PRINT"□"
5 PRINT"NÄR INFALLER PÅSKEN?"
10 INPUT"VILKET ÅR GÄLLER DET";S
20 A=S-INT(S/19)*19
30 B=S-INT(S/4)*4
40 C=S-INT(S/7)*7
50 IF S<1800 THEN M=15
55 IF S<1800 THEN N=6
60 IF S<1900 THEN M=23
65 IF S<1900 THEN N=4
70 IF S=>1900 THEN M=24
75 IF S=>1900 THEN N=5
80 D=19*A+M-INT((19*A+M)/30)*30
90 E=2*B+4*C+6*D+N-INT((2*B+4*C+6*D+N)/7)*7
100 P=D+E+22
110 IF P>31 THEN GOTO 130
120 PRINT"AR ";S;"INFALLER PÅSKEN DEN ";P;"MARS
125 GOTO 5
130 PA=P-31
140 PRINT"ÅR ";S;"INFALLER PÅSKEN DEN ";PA;"APRIL"
145 GOTO 5
150 END
```

FÄRGER

Varför skall alla program man skriver vara användbara? Ofta är det roligare (och ibland nyttigare) att skriva program som inte går att använda till något särskilt. Programmet nedan är ett typiskt sådant exempel. Det är inte särskilt avancerat men visar ändå några intressanta tekniker som man kan ha användning av i mer seriösa program. Vad som i korthet händer när man kör programmet är: först fylls skärmen med reverse mellanslag av slumpmässigt valda färger, och sedan väljs en punkt på skärmen slumpmässigt och de angränsande rutorna ändras till samma färg som den valda rutans. Allt eftersom vissa färger utrotas kommer det (sannolikt) att bli färre och färre färger på skärmen tills att det bara är en färg kvar. Meningslöst? Javisst!

Versioner finns i BASIC och maskinkod, för både VIC 20 (program 1 och 3) och VIC 64 (program 2 och 4). När jag upptäckt att det gick mycket långsamt i BASIC (vilken nyhet!) satte jag mig ner för att skriva ett, vad jag trodde, enkelt och kort maskinkodsprogram. Jag kom inte så långt innan jag började förstå att det kanske inte var så enkelt i alla fall. Det stora problemet var att ta reda på hur man använder slumptal i maskinspråk. Hur jag löste det finns beskrivet i programbeskrivningarna nedan. Jag lär emellertid inte ut maskinspråksprogrammering nu, så ni som inte ännu behärskar detta får hålla er till BASICvarianten.

De båda BASIC-programmen för VIC 20 och 64 är relativt lika (vissa adresser samt antalet olika färger skiljer sig) så jag går igenom dem samtidigt. På raderna 100-120 läses först värden in till en vektor D%. Dessa värden är de som skall läggas till skärmadressen för att komma åt alla angränsande rutor. På raderna 140-160 fylls sedan skärmen med reverse mellanslag (bildskärmskod 160), och motsvarande adresser i färgminnet sätts till slumpmässigt valda färger (värden mellan 0 och 7 för VIC 20, och mellan 0 och 15 för VIC 64). Därefter väljs en punkts adress P i färgminnet slumpmässigt, och adressens färg C tas fram (rad 180). På raderna 190-220 adderas värdena i vektorn D% till P i en loop, och om den resulterande adressen är rätt (dvs ej utanför färgminnet) sätts den till färgen C. Alla angränsande rutor till den valda punkten P kommer alltså att få samma färg som P. Slutligen, på rad 230, hoppar programmet tillbaka till rad 180 och väljer en ny punkt.

Programmen är korta och enkla och erbjuder antagligen inget nytt för den mer avancerade användaren. Men för den som inte hållt på så länge med programmering kan användningen av en vektor kanske vara intressant. Hur som helst är programmen skrivna med tanke på att vara lättlästa och lättförståeliga. I många fall är detta att föredra jämfört med program som visserligen är lite snabbare men därför också mycket krångligare. Om ni inte visste det innan (jag tror att jag har sagt det förut), får man en tomrad genom att först skriva radnumret, sedan ett grafiskt tecken, ett mellanslag och ytterligare ett grafiskt tecken. När man sedan slår RETURN och programraden läggs in i minnet, kommer de grafiska tecknen att ignoreras och raden alltså bestå av endast ett mellanslag.

Ur programmeringsteknisk synpunkt är maskinkodsprogrammet mer intressant. Där används en del intressanta rutiner för att få fram slumpal inom rätt gränser. Programmen är listade i form av BASIC-program som POKEar in maskinkodsprogrammet och sedan sätter igång det. Eftersom det är relativt små delar av programmet som är intressanta, har jag inte tagit med någon assemblerlistning. Ni som är intresserade av att se allt i detalj kan ju lista programmet med en maskinkodsmonitor. De addresser som ges i följande beskrivning är addresserna till VIC 64, med VIC 20:s address i parentes efter.

Hur genererar programmet då sina slumptal? Det använder sig naturligtvis av BASICens RND-rutin (kan BASIC, kan maskinkod!). Startadressen för RND är $E097 ($E094 för VIC 20). Det första som görs är att tecknet på flyttals-ackumulator nr 1 (fac#1), som antas innehålla argumentet, kollas. Beroende på om det var positivt, negativt eller noll, hoppar sedan datorn till olika delar av RND-rutinen. Dessa delar har sedan en gemensam "utgång" där de möts igen på adress $E0E3 ($E0E0). Om argumentet är positivt används 4 byte från timer A och B på I/O-block 1 till att bilda slumptal. Eftersom två av dessa ändras varje klockcykel blir det inga riktiga slumptal. Om argumentet är negativt byts bara innehållet i fac#1:s mantissa om. Posivt argument resulterar i "riktigt" slumptal. Först multipliceras fac*1 med ett konstant värde, och sedan läggs ett annat konstant värde till resultatet. (Följaktligen kommer RND(1) alltid att ge samma resultat varje gång man sätter på sin dator. Konstantvärdena är dock olika för olika datorer.) Sedan

Exempel 1.

```
LDA #$03
LDY #$E9              ;$03E9 = 1001
JSR $B391 ($D391)     ;LAGRA 1001 I FAC#1, (A,Y) INNEHALLER TALET
LDX #<TAL
LDY #>TAL             ;(X,Y) PEKAR PA TAL
JSR $BBD4 ($DBD4)     ;LAGRA FAC#1 SOM FLYTTAL I ADRESS PEKAD PA AV (X,Y)
```

Exempel 2.

```
JSR $E0BE ($E0BB)     ;FIXA SLUMPTAL
LDA #<TAL
LDY #>TAL             ;(A,Y) PEKAR PA TAL
JSR $BA28 ($DA28)     ;MULTIPLICERA FAC#1 MED ETT TAL PEKAT PA AV (A,Y)
JSR $B7F7 ($D7F7)     ;KONVERTERA FAC#1 TILL ETT HELTAL SOM LAGRAS I ($14)
```

körs den "negativa delen" (omkastning av byten i fac#1:s mantissa) som ligger direkt efter. När man skall använda sig av slumptal i maskinkod kan man alltså direkt anropa den del man är intresserad av. Startadresserna följer:
RND(pos): $E0BE ($E0BB)
RND(neg): $E0D3 ($E0D0)
RND(0): $E09E ($E09B)

I programmen nedan används startaddressen för positivt argument. Eftersom resultatet av RND lagras i fac#1 kan man använda en av byten i dess mantissa som slumptal. För att få en slumpfärg gör programmet följande:

JSR $E0BE ($E0BB)
LDA $62
AND#$0F (#$07)

Anledningen till att ackumulatorn ANDas med olika värden för VIC 64 och VIC 20 är att de har olika antal färger (16 respektive 8 stycken). När sedan skärmen har fyllts, skall adresser slumpmässigt väljas i färgminnet. Detta skulle man naturligtvis kunna göra genom att använda $62 och multiplicera (rotera) ett antal gånger. Programmet gör emellertid inte på det sättet. Det lägger istället upp ett flyttal motsvarande 1001 eller 506 i minnet, och multiplicerar fac#1 med detta varje gång RND har använts. Multiplikationen såväl som lagringen av talet i minnet gör programmet med hjälp av BASICens rutiner. Programexempel 1 lagrar talet 1001 i minnet (med label TAL). Exempel 2 tar fram ett slumptal och multiplicerar detta med talet lagrat under labeln TAL, samt omvandlar resultatet till ett heltal som lagras i adresserna $14 och $15.

Martin Fredriksson

Program 1.

```
100 FOR I=0 TO 7
110 : READ D%(I)
120 NEXT
130
140 FOR I=7680 TO 8185
150 : POKE I,160: POKE 30720+I,INT(RND(1)*8)
160 NEXT
170
180 P=38400+INT(RND(1)*506): C=PEEK(P)
190 FOR I=0 TO 7
200 : P2=P+D%(I)
210 : IF P2>=38400 AND P2<=38905 THEN POKE P2,C
220 NEXT
230 GOTO 180
240
250 DATA -23,-22,-21,1,23,22,21,-1
```

Program 2.

```
100 FOR I=0 TO 7
110 : READ D%(I)
120 NEXT
130
140 FOR I=1024 TO 2023
150 : POKE I,160: POKE 54272+I,INT(RND(1)*16)
160 NEXT
170
180 P=55296+INT(RND(1)*1001): C=PEEK(P)
190 FOR I=0 TO 7
200 : P2=P+D%(I)
210 : IF P2>=55296 AND P2<=56295 THEN POKE P2,C
220 NEXT
230 GOTO180
240
250 DATA -41,-40,-39,1,41,40,39,-1
```

Program 3.

```
10 PRINT "■PROGRAMMET LADDAS."
20
30 FOR I=7424 TO 7424+155
40 : READ A: POKE I,A
50 NEXT
60
70 SYS 7424: REM STOPPA MED RUN/STOP + RESTORE
80
90
100 DATA 169,1,160,250,32,145,211,162,156,160,29,32,212,219,169,30,133,64
110 DATA 169,150,133,66,169,0,133,63,133,65,160,0,169,160,145,63,152,72
120 DATA 32,187,224,104,168,165,98,41,7,145,65,165,64,201,31,208,4,192
130 DATA 249,176,9,200,208,226,230,66,230,64,16,218,169,148,133,63,169,29
140 DATA 133,64,32,187,224,169,156,160,29,32,40,218,32,247,215,165,21,24
150 DATA 105,150,133,21,162,0,160,7,165,21,133,66,177,63,24,101,20,133
160 DATA 65,177,63,48,4,144,6,230,66,176,2,198,66,165,66,201,150,144
170 DATA 16,240,10,201,151,208,10,165,65,201,250,176,4,161,20,129,65,136
180 DATA 16,208,48,182,233,234,235,1,23,22,21,255
```

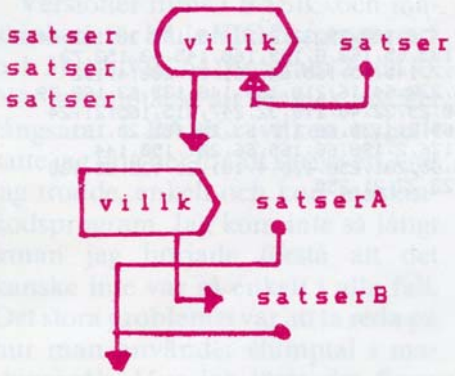
Program 4.

```
10 PRINT "■PROGRAMMET LADDAS."
20
30 FOR I=7*4096 TO 7*4096+158
40 : READ A: POKE I,A
50 NEXT
60
70 SYS 7*4096: REM STOPPA MED RUN/STOP + RESTORE
80
90
100 DATA 169,0,141,32,208,169,3,160,233,32,145,179,162,159,160,112,32,212
110 DATA 187,169,4,133,64,169,216,133,66,169,0,133,63,133,65,160,0,169
120 DATA 160,145,63,152,72,32,190,224,104,168,165,98,145,65,165,64,201,7
130 DATA 208,4,192,231,176,9,200,208,228,230,66,230,64,16,220,169,151,133
140 DATA 63,169,112,133,64,32,190,224,169,159,160,112,32,40,186,32,247,183
150 DATA 165,21,24,105,216,133,21,162,0,160,7,165,21,133,66,177,63,24
160 DATA 101,20,133,65,177,63,48,4,144,6,230,66,176,2,198,66,165,66
170 DATA 201,216,144,16,201,219,144,8,208,10,165,65,201,232,176,4,161,20
180 DATA 129,65,136,16,208,48,182,215,216,217,1,41,40,39,255
```

Strukturerade BASIC-program

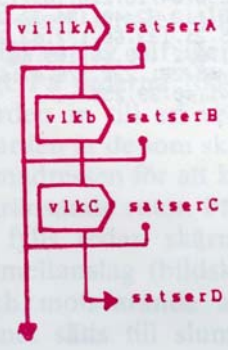
Här fortsätter Lars-Erik Bengtsson sin serie om hur man med lite eftertanke och planering kan dra nytta av moderna strukturerade principer i sin programmering - även om man använder det ostrukturerade språket BASIC. I nästa nummer av VIC-rapport kommer hans sista, avslutande artikel. Där går Lars-Erik igenom ett större program, som gjorts upp efter de principer han lär ut.

En dellösning skall bara ha en väg in och en väg ut! Denna form kan beskrivas på många sätt. Vi har tidigare studerat grundformerna:



Alla moderna språk (Pascal, Simula, C, Comal och Ada) uttrycker dessa på ett enkelt sätt. Det finns emellertid andra varianter och de senaste språken (Comal och Ada) tillåter former som t.ex. inte finns i Pascal.

Vägvalet med flera "vägvisare".



"Om villkor A är uppfyllt utförs satser A.
Om villkor A inte är uppfyllt undersöks villkoren och om detta är uppfyllt utförs satser B annars undersöks villkor C osv.
Om inget av villkoren (A, B, C) är uppfyllt utförs satser D".

I Basic skriver vi:

```
xxxx IF villkor A THEN satser A: GOTO wwww
yyyy IF villkor B THEN satser B: GOTO wwww
zzzz IF villkor C THEN satser C: GOTO wwww
---- satser D
---- .....
wwww REM ENDIF
```

Problemen såg vi redan i det enkla vägvalet. Enradsvarianten av IF ger ett begränsat utrymme. "Vänder" vi villkoret får vi istället:

```
xxxx IF NOT villkor A THEN yyyy
---- satser A.....
---- ......: GOTO wwww
yyyy IF NOT villkor B THEN zzzz
---- satser B.....
---- ......: GOTO wwww
zzzz IF NOT villkor C THEN vvvv
---- satser C.....
---- ......: GOTO wwww
vvvv satser D.....
---- .....
wwww REM ENDIF
```

som ger obegränsat utrymme för varje alternativ.

Upprepningar med "inre utgångar".



"Utför satser A - satser B så länge villkoret inte är uppfyllt. Observera att satser A alltid utförs i gång mer än satser B."

I Basic:

```
xxxx REM LOOP
---- satser A
---- .....
yyyy IF villk THEN zzzz
---- satser B
---- ......: GOTO xxxx
zzzz REM ENDLOOP
```

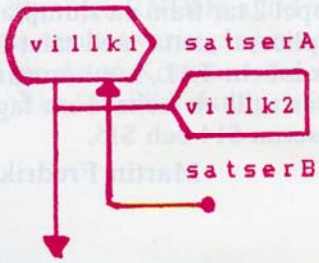
Strukturen har ett specialfall av särskilt intresse:




```
xxxx REM LOOP
---- satser A
---- .....
yyyy IF villk THEN zzzz
---- GOTO xxxx
zzzz REM ENDLOOP
```

Denna form kallas i Pascal REPEAT-UNTIL. Den generella formen är däremot inte tillåten i Pascal.

Ada tillåter specialfall som tidigare inte ansågs vara strukturerade:




```
xxxx IF NOT villk1 THEN zzzz
---- satser A
---- .....
yyyy IF villk2 THEN zzzz
---- satser B
---- ......: GOTO xxxx
zzzz REM ENDLOOP
```

Observera att villkor1 skall vändas och att villkor2 bibehålls oförändrat.

Avslutning:

Vi har nu beskrivit de flesta former av styrande satser som vi kan återfinna hos de moderna språken. Löser vi våra problem med dessa som utgångspunkt får vi likartade programstrukturer oberoende av språket vi använder. Ett Basic-program som följer dessa regler blir därmed lätt att översätta till t.ex. Comal. I våra programlistor känner vi snart igen dessa former. Detta gör det lättare för oss att följa och förstå texten. Det är emellertid viktigt att vi är konsekventa. Kan vi inte vara säkra på att programtexten är strängt strukturerad förlorar vi viktig information. Gör först en strukturerad flödesmodell. Beskriv sedan denna med hjälp av språket.

PRENTICE HALL

Från Prentice-Hall kommer ett antal intressanta böcker om VIC-20 och 64 varav några recenseras i detta nummer. Enligt katalogen har man följande titlar klara:

Elementary Commodore 64 *by William Sanders.*

Using the Commodore 64 in the Home *by William Behrendt & Hank Liubrach.*

Commodore 64 User's Guide *by Tim Onosko.*

Commodore 64 Computing *by Ian Sinclair.*

Sprite Graphics for the Commodore 64 *by Sally Larsen.*

VIC BASIC: A User-Friendly Guide *by Ramon Zamora.*

The Elementary VIC *by William Sanders.*

Getting the Most From Your VIC 20 *by Hank Librach.*

Kids and the VIC *by Edward Carlson.*

Man har även ett par program med tillhörande böcker:

Start with BASIC on the Commodore VIC-20 *by Don Monro.*

Miner 2049'er *by Jerry Brechner.*

Krångliga tecken i listorna?

Eftersom många har efterlyst information om hur man får fram de konstiga svarta tecken, som syns i listorna ibland, har vi här tagit fram för VIC-20 ,exempel på vad som händer när man använder olika styrtangenter innanför citationstecken.

För det är ju bara där som tecknen kommer fram! Vi beklagar att vi inte tidigare uppmärksammat problemet, men vi bättrar oss ju..., eller hur? Här följer i alla fall listningen och tangenterna i klartext. Nedanstående listning avser endast s.k. versaler (stora bokstäver). Står maskinen så inställd att den ger små bokstäver får vi ytterligare några tecken, men de hann vi inte med före pressläggningen, så dem spar vi till nästa nummer av VIC-rapport.

```
10 PRINT" ■ " CTRL-1
20 PRINT" ■ " CTRL-2
30 PRINT" ■ " CTRL-3
40 PRINT" ■ " CTRL-4
50 PRINT" ■ " CTRL-5
60 PRINT" ■ " CTRL-6
70 PRINT" ■ " CTRL-7
80 PRINT" ■ " CTRL-8
90 PRINT" ■ " RVS ON
100 PRINT" ■ " RVS OFF
110 PRINT" ■ " HOME
120 PRINT" ■ " CLEAR (SHIFT HOME)
130 PRINT" ■ " CRSR DOWN
140 PRINT" ■ " CRSR UP
150 PRINT" ■ " CRSR RIGHT (MARKÖR ÅT HÖGER)
160 PRINT" ■ " CRSR LEFT (MARKÖR ÅT VÄNSTER)
170 PRINT" ■ " F1
180 PRINT" ■ " F2 (SHIFT F1)
190 PRINT" ■ " F3
200 PRINT" ■ " F4 (SHIFT F3)
210 PRINT" ■ " F5
220 PRINT" ■ " F6 (SHIFT F5)
230 PRINT" ■ " F7
240 PRINT" ■ " F8 (SHIFT F7)
250 PRINT" NU ÄR DET SLUT"
```

Bland kommande nyheter kan nämnas:

Gordon - 100 Programs for Commodore 64.

Gratte - Starting with Comal.

Winter - Computer playground: Commodore 64/VIC 20.

Zamora VIC BASIC.

Bush - VIC 20

VIC 20: Fifty easy to run computer games.

Sovego - VIC 20 programmers notebook.

VIC 20 Games, Graphics and Applications plus Cassette.

Hakell - Commodore 64/VIC 20

Ruston - Pascal with your Basic micro.

Calc Input

Hur ofta har man inte önskat att man kunnat ge ett uttryck som input när man velat ta in ett värde för en variabel. En dator är ju till för att utföra beräkningar åt oss och man skall inte behöva sitta med en räknedosa bredvid sig för att kunna svara på de frågor som ges på skärmen. Jag fick för en tid sedan till uppgift att lösa denna uppgift och skall här redovisa vad jag kom fram till. Rutinen är avsedd för VIC 20 men jag tror att det räcker att ändra på raderna 60030, 60040 och 60050 för att den skall passa för 64:an. Detta har jag dock inte kunnat testa så ni som vill anpassa rutinen för denna maskin får experimentera lite med dessa värden.

Vad innebär då ett calc input?

Den normala input rutinen fodrar ju att man antingen ger ett flyttal eller en sträng som input. Eftersom vi här bara skall titta på flyttal så låter vi input rutinen behålla sin gamla form vad det gäller strängar. När man skall mata in ett flyttal till ett program använder man vanligen en sats i stil med

10 INPUT A

och datorn kommer då att "putta" ut ett frågetecken samt tända cursorn för att indikera att den väntar på data från operatören. För oss är det då bara att skriva in det tal vi vill mata in och sedan är allt frid och fröjd.

Tänk dig nu ett fall när vi istället för att skriva 2E 2 som input hellre skulle vilja skriva

2 SIN(2*π)-2 * X↑2 - 88

Med den normala inputrutinen så går detta inte särskilt bra utan vi får beräkna det aktuella uttrycket på t.ex. räknedosa och sedan mata in svaret till VICKE. Den rutin som redovisas här löser detta problem och du kan kasta räknedosan all världens väg.

Någon kanske tycker att uttrycket ovan inte alls behöver beräknas innan man matar in det till datorn, det är ju så lätt att programmera in den aktuella formeln i programmet.

Detta är ju också mycket riktigt sant men ibland inträffar fall när man behöver "inputta" *olika* beräkningsintryck varje gång och att då göra om programmet varje gång för att möta den nya situationen är både tidsödande och tråkigt. Låt oss titta på ett exempel där en rutin av detta slag skulle vara användbar.

Antag att vi vill göra ett program som tar in tio olika värden på en effekt och lagrar det i en vektor X(10). Detta program skulle man då kunna skriva t.ex. så här

```
10 FOR I = 0 TO 9
20 INPUT P(I)
30 NEXT
```

Den elektriska effekten kan nu ges av tre olika uttryck:

1) Mäter vi spänningen (U) och strömmen (I) i ett kretsnät så ges effekten (P) av uttrycket P=U*I

2) Mäter vi en resistans (R) och strömmen (I) så ges effekten av uttrycket P=(I↑2)*R.

3) Mäter vi en resistans och spänningen så ges effekten av uttrycket P = (U↑2)/R.

```
10 FORI=0TO10
20 GOSUB60000:P(I)=IN
30 NEXT
60000 INPUT A$
60010 PRINT"?":IFA$=""THENA$="0"
60020 PRINT"■■■IN=";A$;":GOTO60070
60030 IF LEN(A$)>11THENPRINT"?";
60040 IF LEN(A$)>33THENPRINT"?";
60050 IF LEN(A$)>55THENPRINT"?";
60060 PRINT"???":POKE198,1:POKE631,13:END
60070 PRINT"??";IN:A$="":RETURN
```

Om alla dessa situationer uppstår lika ofta skulle vi behöva tre olika inputrutiner eller en enda där vi beräknade uttrycket på t.ex. en räknedosa innan vi slog in det på datorn. Med subrutinen som ges här har vi möjlighet att ge själva beräkningsuttrycket som input och slipper alltså tänka på hur vi erhåller det värde som datorn skall matas med. En annan egenskap hos rutinen är att man kan använda variabler vid input. Antag t.ex. att flyttalsvariabeln Q är definierad redan tidigare i programmet. Ett input på formen

$LOG(Q*2E_2)$

är då fullt möjlig. Detta kan vara bra att ha när det man "inputar" skall multipliceras med en konstant som är vald bland några standardvärden. Man behöver då bara definiera dessa standardvärden tidigare i sitt program och kan använda dem fritt när man vill låta programmet arbeta åt sig.

Hur använder man då programmet?

Detta är enkelt. Man ersätter bara den aktuella INPUT med ett subrutin-anrop till 60000 eller var man nu har lagt denna rutin. När rutinen anropas så lägger den ut ett frågetecken och tänder cursorn på vanligt manér. Du kan då ge ditt input på den form du önskar. Som vanligt, dvs ett flyttal eller som ett beräkningsuttryck av godtycklig komplexitet. Beräkningsuttrycket bör dock inte vara längre än ca 3 rader, men denna längd räcker gott och väl i de flesta fall. När man återvänder från rutinen så innehåller variabeln IN resultatet av beräkningsuttrycket. Strängvariabeln A$ är också upptagen men denna sätts till längden noll i den sista raden och han har alltså inte tillgång till uttrycket.

```
10 FOR I = 0 TO 9
20 GOSUB 60000 : P(I)=IN
30 NEXT
```

En förklaring av hur programmet fungerar kanske är på sin plats. Den normala inputrutinen ersätts i subrutinen av ett input av en sträng. Denna godkänner alla typer av input och säger inte till om man t.ex.

svarar "KALLE" istället för ett flyttal. Resultatet av att ge t.ex. "KALLE" som input blir samma som att slå return eller bara skriva "0". I alla dessa fall så är det alltså noll som du matar in.

På rad 60010 så rensas skärmen och på den efterföljande raden skrivs IN, ansatt till det aktuella uttrycket, ut.

Raderna 60030-60050 sköter om att flytta cursorn uppåt ett antal rader beroende på hur långt beräkningsuttrycket var. På rad 60060 flyttas cursorn uppåt fyra rader varje gång. Men poke 198,1 talar vi sedan om att det finns ett tecken i tangentbordsbufferten. Poke 631,13 lägger in koden för RETURN i denna buffert. Raden avslutas som du ser med END. Detta kan kanske vara lite förvånande. När vi når END så går programmet ut i DIREKT MOD dvs det förbereder sig för att lägga READY på skärmen och börja blinka med cursorn. Samtidigt börjar VICKE dock kolla om någon tangent har tryckts ned och genom våra pokar så har vi lurat systemet att tro att RETURN har slagits ned, eftersom koden för RETURN (13) ligger i tangentbordsbufferten. Efter våra cursorförflyttningar så ligger cursorn på den rad där vi skrev ut IN = beräkningsuttryck:goto 60070. och effekten blir att datorn tror att detta är vad vi skrivit på tangentbordet och sedan, genom att slå return, vill att den skall behandla.

Detta sätt att lura datorn är användbart i en mängd fall t.ex. när man vill skapa ett program som skriver sig själv och kan vara nyttigt att lägga på minnet. Vad man gör är alltså att lägga antalet tecken man tänker poka in i tangentbordsbufferten i minnescell 198 och sedan lägga koderna för det man vill ha utfört i minnescell 631 och framåt. Du kan maximalt lägga 10 tecken i bufferten.

Den sista raden slutligen skriver ut resultatet av beräkningsuttrycket så att man får koll på detta och återvänder sedan genom RETURN till det subrutin-anrop som kallade på rutinen. Observera att denna rad exekveras bara efter det att END har körts i rad 60060.

Åke Hedman

LÄTTFIL

- ett registerprogram för VIC-64



Compsoft Lättfil heter ett nytt registerprogram från samma företag, som säljer DMS och DELTA, två program som är välkända och där t.ex. DMS utnämnts till årets programvara i England 1982.

Programmet, som ursprungligen hette Easy File (det blir ju lättfil på svenska!) kommer från Simple Software i England. Det har genomgått en grundlig omarbetning beträffande skärmhantering och mycket annat som gör det användarvänligt. Som kronan på verket har man lagt in en liten modul som gör säkerhetskopiering på enkelfloppy, dvs en standard 1541-floppy till 64:an.

Program och data ligger på samma disk (poster lagras på disken allteftersom de skrivs in för att få största säkerhet och bekvämlighet). Största poststorlek är 256 tecken, dvs en post per block.

För att räkna ut maximala antalet poster gör man så här: Antalet fält multipliceras med 2. Därtill läggs antalet tecken i posten. Summan man får är poststorleken. Posterna är dynamiska, vilket gör Lättfil till ett ekonomsikt program. I en post kanske efternamnet tar stor plats, i en annan är kanske adressen lång.

Programmoduler för etikettutskrift och för beräkningar i registret finns också. "Höj medlemsavgiften med 10:- för alla seniorer" är ett av de enklare tricken som Lättfil kan utföra.

Rapportgeneratorn har även den processmöjligheter. "Skriv en lista på alla i fotbollssektionen och skriv också ut spelarnas genomsnittsålder" är en annan uppgift, som Lättfil klarar galant. Eller "Ge vinstlistan med värdet inklusive och exklusive moms samt totala vinstvärdet".

Sortering och sökning kan ske på vilket fält som helst. Urval kan ske på hela fält eller delar av fält (t.ex. alla som har efternamn på -son).

Lättfil säljs av Compsoft i Farsta.

J. Stiernborg

Ansvarig utgivare: Peter Elmlund
Redaktör: Åke Fredriksson
Övr. medarbetare: Lars-Erik Bengtsson, Martin Fredriksson, Åke Hedman, Jörgen Stiernborg, Peter Tegelberg
Annonser: Telefon 031–87 70 84
Tryck: Ölandstryckarna AB, Borgholm
Layout: Ottoson's Reklam

Annonsorder och annonsmaterial (heloriginal eller negativ film) enligt överenskommelse under 1983.

Tryckförfarande: Offset
Upplaga 30.000 ex.
Nästa nummer av VIC-rapport utkommer den 7 febr. 1984. Manusstopp den 29 dec. 1983.

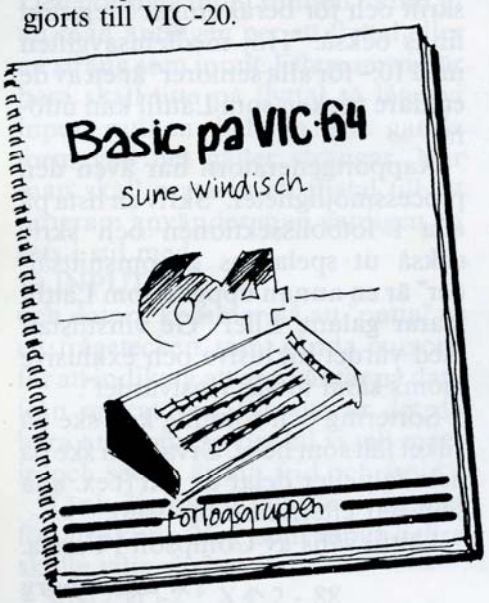
Nya läroböcker till VIC-datorerna

Som alla datorintresserade i dag vet, finns till VIC-20 ett av marknadens mest kompletta läromedelspaket på svenska. De flesta av dessa böcker har framtagits av Förlagsgruppen som är ett litet specialförlag inom datalitteratur.

Nu har man kompletterat bokserien med ett stort antal böcker till VIC-64. En VIC-64 tillsammans med de här böckerna borde vara en av marknadens mest prisvärda hemdatorer i dag.

Grundböcker till VIC-64

Tre grundböcker har tagits fram till VIC-64 på samma sätt som tidigare gjorts till VIC-20.



BASIC PÅ VIC-64

Den första boken för nybörjare beskriver uppkoppling och tangentord. För den som inte kan basicprogrammering är det här en bok som lättförståeligt beskriver grunderna i BASIC. Boken är genomgående skriven med programexempel som praktiskt skall köras på datorn.

Hela tiden beskrivs nya detaljer i programmeringen noggrannt. På grund av bokens uppläggning anser förlaget att den är utmärkt för både självstudier och olika former av gruppstudier, som studiecirklar, gymnasieskola m.m.

Detta gäller för övrigt alla Förlagsgruppens böcker. De är medvetet skrivna för att passa i de flesta sammanhang.

Här får du lära dig alla grundinstruktioner, kommandon och funktioner. Du får också lära dig olika begrepp inom programmering, som subrutiner, flödesplaner, strängar, indexerade variabler m.m.



Grafik och ljud på VIC-64

Det finns många finesser och specialfunktioner hos VIC-64. En hel del av dessa beskrivs i den här boken och för dig som har en VIC-64 är det här en naturlig fortsättning på grunderna i basicprogrammering.

Titeln på den här boken kan inte omfatta allt du får lära dig. Givetvis är grafik ett omfattande avsnitt. Det finns mängder att lära inom grafiken. Först har du alla standardgrafiktecknen (de som finns på tangentbordet). Det beskrivs hur du bäst och lättast använder dessa. En rolig detalj är att man samtidigt lär sig arbeta med färg för givetvis ska grafiska program vara gjorda i effektfulla färger.

Med programexempel beskrivs hur du bygger upp olika former av diagram och hur du plottar kurvor med hjälp av grafiken.

Sprite

En effektfull variant av grafik är de s.k. "sprites". Det är förprogrammerade grafiska figurer som man med enkla kommandon kan använda för "rörliga bilder" på bildskärmen.

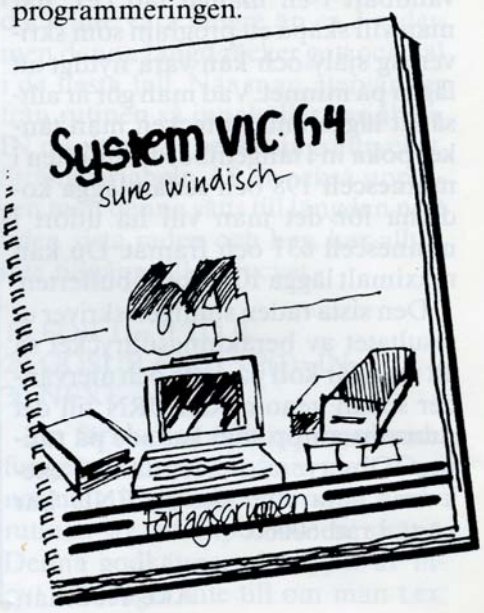
För att riktigt förstå sprites måste man vara lite orienterad i binära talsystemet, samt veta lite om minnesuppbyggnaden. Allt detta beskrivs i det här avsnittet tillsammans med programexempel på hur du bygger dina egna sprites. Det här är tekniskt avancerat men beskrivet ovanligt lättfattat.

Du får också en beskrivning på hur du kan bygga en helt egen teckengenerator med egna grafiktecken.

Ljud

Ljuddelen på VIC-64 är mycket avancerad. Användandet av den beskrivs ganska kortfattat, men fullt tillräckligt för den som vill gå vidare och göra avancerade ljudillustrationer.

En positiv detalj i boken är att den har 7 appendix längst bak i boken. Dessa fungerar som uppslagsverk för adresser, teckenkoder och andra viktiga detaljer som ofta behövs vid programmeringen.



Den här boken kompletterar de andra två grundböckerna för att ge dig en komplett uppfattning om vilken komplett dator VIC-64 är. För dig som köpt kringutrustning till VIC-64 är SYSTEM VIC-64 oumbärlig om du vill få ut det mesta av din utrustning.

Här beskrivs med lättfattliga programexempel hur du bäst använder skrivare, kassettbandspelare och flexskiveminnet. Speciellt flexskiveminnet är mycket detaljerat beskrivet. Dels lite tekniskt hur det fungerar och dels hur man programmässigt nyttjar det (bl.a. med ett litet registerprogram).

En kort orientering om hur paddle och joysticks fungerar ges också i den här boken.

SYSTEM VIC-64 avslutas med att beskriva en utökad BASIC som heter TOOL 64. För att kunna använda den måste man köpa en cartridge som heter TOOL 64. Det är en kassett som trycks in i datorn och ökar ut BASICen till en helt otrolig nivå för en sådan här typ av dator.

Förlagsgruppen har producerat ett mycket trevligt ADB-paket till ett relativt billigt pris. Det innehåller övningsprogram för Redovisning, Registerhantering samt Ord- och Textbehandling.

Man kan köpa varje del för sig eller hela utbildningspaketet, vilket ger ett betydligt reducerat pris. Köper man hela ADB-paketet levereras det i en trevlig samlingspärm där man förvarar både program och böcker.

Pärmen innehåller då Registerprogam, Redovisningsprogrammet V-bok med programnyckel och TEXT 64. Dessutom en teoribok i ADB samt arbetsböcker till ovanstående program. Register- och redovisningsprogrammen är utbildningsprogram, vilket innebär samma kraftfulla program som proffsvarianten men strypta i antal poster respektive verifikat.



Redovisning liksom de övriga böckerna startar med en uppkopplings/uppstartsbeskrivning. Då du startat upp programmet arbetar du i ett litet modellföretag där du själv fungerar som kamrer.

Med hjälp av en mängd verifikat gör du en grundboksregistrering med en valfri kontoplan där du kan välja mellan förening, företag, jordbruk eller egen kontoplan.

Med styrda övningar kan du studera olika rapporter som resultatrapport, balansrapport och momsrapport. Du får lära dig att göra periodavslut och årsbokslut.

Det här materialet är mycket prisvärt och i dag lär det vara svårt att för samma kostnad hitta motsvarande utbildning. Materialet är upplagt så att det passar såväl skolor, studieförbund, större företags internutbildning och mindre företagares introduktion till ADB.



Med hjälp av 25 övningar arbetar du dig praktiskt igenom de moment som brukar förekomma i registersammanhang.

Man kan tycka att ett register är ett register, det skall man inte behöva träna på. Faktum är att en genomkörare av den här modellen ger både inblick i programmets möjligheter och ideer som man kan ta med till de register man arbetar med.

Alla böckerna är rikt illustrerade med bilder på utskrifter. Det är inte alltid dessa bilder stämmer riktigt på grund av den snabba utveckling och att programmet förbättrats flera gånger under produktionen.



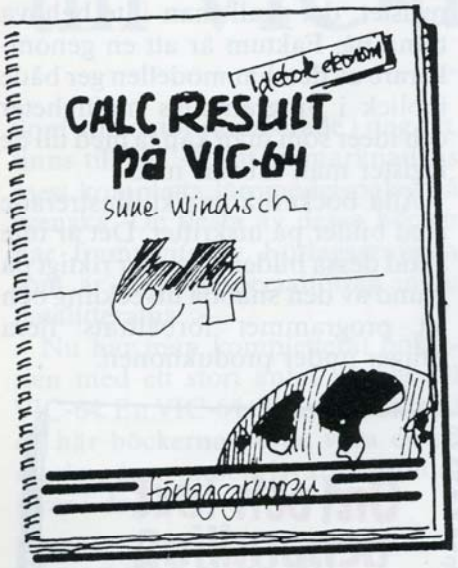
Den här boken är som de två övriga en praktiskt upplagd bok med mängder av övningar.

Programmet som används är TEXT 64 och helt professionellt, alltså ingen utbildningsvariant utan helt användbart inom företaget.

Boken är upplagd för att likna verkligheten så mycket som möjligt. Första övningen är ett handskrivet brev. Därefter följer övningar där man går in och korrigerar i texten. Flera övningar visar texten två gånger. Båda med felaktig text men den ena med inlagda korrektionstecken. Givetvis både för att du skall se alla fel och för att du ska få träning i hur text korrigeras.

forts. nästa sida

Med de övningar som ingår får du träna i normal texthantering, som bläddring i texten, utbyte, radering, insättning av ord eller texter m.m.



Calc result på VIC-64

Handic Software har tagit fram ett kalkyleringsprogram till bl.a. VIC-64. Det heter Calc Result och beskrivs i flera undersökningar som mer avancerat än VisiCalc. Till 64:an finns den i två versioner, dels en enkel: Calc Result Advanced. Det är avsett att användas tillsammans med egna applikationsprogram för att göra lite mer avancerade kalkyleringsarbete med snygg presentation av resultatet. Att få resultatet snyggt presenterat är nog så viktigt vid sådana här arbeten.

Sune Windisch på Förlagsgruppen har gjort en arbetsbok till den här programvaran för att öka förståelsen för vad Calc Result kan användas till. Boken är i första hand avsedd att användas till Calc Result Advanced tillsammans med den manual som följer programmet.

Då man köper boken ingår en flexskiva som innehåller ett antal applikationsprogram av samma typ som man shälv skall skriva. Avsikten med boken är inte att ge dig dessa program så att du slipper göra egna, utan att visa hur man gör sådana här applikationsprogram.

Efter den generella uppstartsrutinen och beskrivning av grunderna i Calc Result kommer övningarna i var sitt kapitel. Alla applikationsprogram finns på flexskiva, men avsikten är egentligen att du ska skriva in programmen själv enligt beskrivningarna. Detta för att lära dig mer.

Flexskivan kan du ha som en säkerhet för att studera hur resultatet ska bli.

Följande övningar finns med i boken:

Varulager, Invantarieförteckning, körjournal, priskalkyl och deklaration.

Deklaration är inte avsett att utföra din deklaration även om programmet kan det. I stället är det med för att demonstrera s.k. global rekalkylering, vilket innebär att man kalkylear över flera sidor. Programmet är konstruerat för att beskriva själva tekniken.

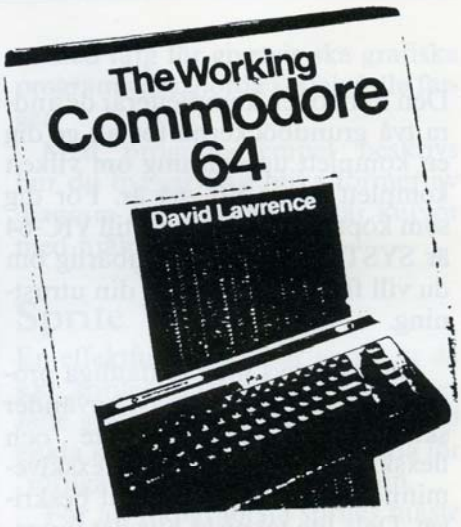
Finska böcker till VIC-64

Våra finska läsare kanske gläds åt att flera av böckerna kommer på finska.

I dag finns endast Basicboken tryckt på finska men både Grafik och ljud och System på VIC-64 är översatta och håller på att tryckas i en finsk variant.

I övrigt håller man på Förlagsgruppen på med ett tekniskt paket till VIC-64 på samma sätt som tidigare gjorts till VIC-20. Det lär dock inte bli färdigt förrän till våren. Även till Calc Result har man planer på en arbetsbok till, men nu med inriktning på teknik.

Alla Förlagsgruppens böcker säljs genom Handic's återförsäljare. Den som vill veta mer om böckerna kan ringa Förlasgruppen i Norrköping 011-13 40 80, som kan svara på de flesta frågor.



The Working Commodore 64

Denna bok av David Lawrence innehåller 20 BASIC-program för 64:an. Kan det vara något att ha?

Ja, om du programmerar i BASIC, så får du dels 20 nya program att skriva in, dels en mängd förslag på rutiner som går att använda i allehanda program. Alla program är skrivna i modulform, och alla programrader är beskrivna i separata kommentarer. Detta gör det lätt att förstå vad varje del av programmet gör. Det gör också att programmen enkelt kan modifieras för egna ändamål, och med hjälp av MERGE, RENUMBERoch DELETE-rutinerna som finns beskrivna i boken, kan du sedan inkorporera de nya modulerna i andra program. Nu frågar sig naturligtvis den ivrige läsaren vad det är för program som beskrivs så väl.

Jo, det är en ganska brokig blandning listningar med allt möjligt, från sprite-program och musikprogram till filhantering och ordbehandling. Och det fina i kråksången är just flexibiliteten i programmen.

Naturligtvis är BASIC-program för högupplösningsgrafik och ordbehandling både långsammare och omständligare än dito maskinspråksprogram, men å andra sidan hade åtminstone inte jag kunnat utnyttja modulerna på samma sätt om de hade varit skrivna i kryptisk maskinkod. Till sist bör väl nämnas att boken naturligtvis är skriven på engelska och att den kostar cirka 120 kr.

Magnus Lyckå

Ännu en bok om VIC-20
VIC 20 EXPOSED

Boken är författad av John vander Reyden och utgiven av Melbourne House (Publishers) Ltd., Melbourne House, Church Yard, Tring, Hertfordshire HP23 5LU, England. (ISBN 0-86161-132-2). Den är på 166 sidor i pocketupplaga och kostar cirka 7 pund i England. Den importeras veterligen inte till Sverige men kan köpas direkt från England.

Innehållet är uppdelat i 9 kapitel och ett appendix. Till största delen ägnar sig boken åt programmering i BASIC och något i maskinkod. Dessutom ges en översiktlig beskrivning av hårdvaran, dock ej hur man använder t.ex. user port. Innehållet är väl disponerat och med hjälp av indexet i slutet bör boken kunna användas som uppslagsverk för programmering av VIC-20. Några programexempel är behäftade med enklare fel. Att rätta dessa kan ses som övningsuppgifter.

I kapitel 1 görs en allmän genomgång av VIC-BASIC. Först beskrivs direktmod och programeringsmod, därefter datastrukturer med byte, karaktärer (tecken) och variabler av olika typer. Slutligen beskrivs operatorerna (+, -, and, or, etc.)

Samtliga BASIC-kommandon avhandlas med exempel i kapitel 2. Vad händer t.ex. med följande printkommando:
PRINT"(BLANK)" (CURSOR LEFT 2 GGR)(CTRL RVS ON) N (RETURN)

Kapitel 3 går in på mer avancerad användning av BASIC. Där beskrivs hur man laddar in nya program under exekvering utan att variablerna nollställs och hur man slår ihop olika program eller subrutiner lagrade på band. Tangentbordsbufferten används i två exempel för att lista varje rad stegvis och för automatisk radnumrering. Till sist ges en maskinkodsrutin för MERGE (inläses i BASIC-datasatser).

Kapitel 4 handlar om grafiken i låg och högupplösningsutförande samt om hur man i flerfärgsmod kan få fyra olika färger i varje tecken. Konstruktionen av egna tecken beskrivs väl och avslutningsvis ges en plotrutin i BASIC för högupplösningsgrafik (176*184 punkter med 3 K minnesexpansion). En finess som avslöjas är hur man kan lagra två olika skärmbilder och blixtsnabbt växla mellan dem.

Därefter följer ett kort kapitel om ljusgenerering. I kapitel 6 avhandlas yttre enheter som kassettbandspelare, skivminne och skrivare. Det mesta utrymmet ägnas åt skivminnet.

Användning av tangentbord och joystick för att styra ett program beskrivs i kapitel 7. Nästa kapitel tar upp hur VIC:en är konstruerad. Vad är en mikroprocessor? Vad gör VIC-kretsen och hur lagras data?

Sista kapitlet går in på maskinspråksprogrammering, lagring av program och avstängning av STOP- och RESTORE-knapparna är bra at kunna. Processorns olika register,

forts. nästa sida

Ännu en bok om VIC-64
Commodore 64 exposed

Författare är Bruce Bayley och boken är utgiven av Melbourne House (Publishers) Ltd., Melbourne House, Church Yard, Tring, Hertfordshire HP23 5LU, England. (ISBN 0-86161-133-0). Pris cirka 7 pund i pocketupplaga (195 sidor).

Detta är en trevlig, välskriven bok i behändigt format, som ägnar sig åt programmering i BASIC och maskinkod. Dessutom beskrivs datorns fina ljud och grafikmöjligheter. Ett flertal nyttiga program är infogade i texten. Genom att studera dessa närmare kan läsaren lära sig mycket om programmering, ljud och grafik. Till viss del beskrivs hur datorn arbetar och hur man kan lägga in egna kommandon och funktioner (t.ex. stegvis listning).

Kapitel 1, 2 och 3 handlar om BASIC-programmering och är nästan identiska med de i boken VIC-20 Exposed (se recension på annan plats i tidningen). Här finns bland annat rutiner för automatisk radnumrering och sammanslagning av två olika program på band.

Ljudgenereringen beskrivs därefter med flera programexempel. I ett demonstrationsprogram (på sju sidor) plottas ljudets vågform på skärmen samtidigt som man hör ljudet. En tabell visar innehållet i de 29 olika ljudregistren, varefter varje register gås igenom separat. Allt är väl avhandlat utom filtreringsdelen, som borde varit utförligare.

Grafikkapitlet är hämtat från VIC-20 Exposed (modiferade adresser fram till sprite-avsnittet. I detta senare finns en liten käck rutin för sprite-generering, samt tabell som förklarar de 47 st olika sprite-registren.

Kapitel 6 handlar om maskinkodning. Eftersom processorerna i VIC-20 och VIC-64 har samma instruktionslista har även detta avsnitt delvis kunnat hämtas från VIC-20 Exposed. Ett antal nya program finns med för att läsaren skall bli bekant med processorns olika instruktioner och adresseringsmoder. Bland dessa program kan nämnas en enkel maskinkodsmonitor. Olika sätt att disponera 64:ans RAM och ROM minnen visas. Därefter följer några program, som använder 64:ans maskinkodsrutiner i ROM (Kernal och BASIC). Bland annat för definition av funktionstangenter och plottning i låg- och högupplösningsgrafik. Sista delen handlar om raster-interrupt, dvs hur man kan använda linjegenereringen av TV-bilden för diverse effekter, t.ex. generera fler än 8 sprites.

Sista kapitlet behandlar kassettbandspelare, skivminne och printer. I appendix visas koder för CHR$, skärm och ASCII, en lista över de olika minnesregistren, hur man får olika grafiska tecken, 33 st ROM-Kernal-rutiner, BASIC felmeddelanden samt tangentbordskoder.

L. Moeschlin

adresseringsmoder och instruktioner genomgås. Materialet räcker för den som vill lära sig lite om maskinspråksprogrammering. Som exempel på användning ges en rutin som rör ett tecken över skärmen alltmedan det ordinarie BASIC-programmet körs. Ett annat exempel är ett plot-program som placerar markören på en godtycklig plats med kommandot SYS PL(X,Y). Slutligen finns en BASIC-rutin för omvandling av maskinkod till datasatser.

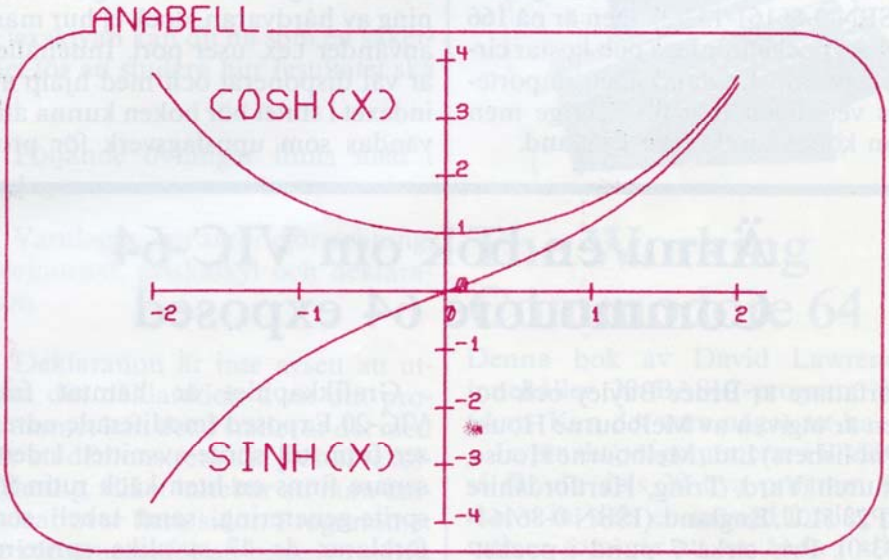
Allra sist finns ett antal appendix. Det första beskriver CHR$( )-koderna, skärmkoderna och ASCII-koderna. Nästa går igenom vad som lagras i de olika minnescellerna. Därefter visas hur man trycker på tangenterna för att få ett visst grafiktecken.

Appendix D går igenom 33 st användbara ROM-rutiner, som t.ex. PLOT för att placera markören på godtycklig plats på skärmen. Färgkoder för bakgrund och ram, uppdelning av minnet, felmeddelanden i BASIC och vilket värde som erhålls vid PEEK(197) vid olika tangenttryckningar följer sedan. Sista appendix går igenom 22 st maskinkodsrutiner för BASIC, t.ex. hur man skriver en sträng på skärmen.

L. Moeschlin

VIC-redaktionen beklagar, att
TIAL TRADING
- som levererar alla spelprogram, som stod först i programkatalogen i nummer 3 i år, inte nämndes. Vi ber läsarna observera, att alla spel, som har nummer före namnet (på sidorna 41, 42, 43 och 44), kommer från TIAL!

Bokrecension



ANABELL®, ISSN 0280-6959 och ISBN 91-970463-0-2. Den analoga tabellen visar exakt utseendet av alla miniräknarens funktioner...

Ja så står det på framsidan till ett bokalster som jag tagit del av. Boken riktar sig i första hand till skolelever som i sitt dagliga arbete med bl a matematik kommer i kontakt med matematiska funktioner men som kanske inte är säkra på vad de egentligen är. Jag tycker att boken ger bra förståelse för vad som döljer sig bakom de ”konstiga” uttryck som Du stöter på även när Du använder din hemdator. Tänk att kunna slå upp vad egentligen SIN, COS, SQR, EXP och de andra funktionerna står för.

Boken ger ett vederhäftigt intryck och är välgjord i alla delar. Till varje datorplot, för det är nämligen sådana som används, finns ett litet stycke matematikhistoria som förklarar och ger bakgrunden till kurvorna. En fördel är att man kan använda en vanlig linjal för att mäta i figurerna, de är nämligen måttsatta så att det går att läsa av värdet på funktionerna med hjälp av linjalen. Boken tar upp de vanligaste elementära matematiska funktionerna såsom SIN(x), COX(x), TAN(x), ARCTAN(x), ARCSIN(x), ARCCOS(x), EXP(x), LN(x), 10x, ^{10}LOG(x), X^2, SQR(x), 1/x, Ax, SINH(x) och COSH(x).

Ett orginellt grepp är att boken har sidorna så att säga på tvären vilket gör att figurerna är stora och lättlästa samtidigt som det är lättare att bläddra i boken med ena handen och arbeta med miniräknaren eller hemdatorn med den andra.

Boken är tryckt på svagt färgat papper och det gör figurerna lättlästa samtidigt som man inte blir bländad.

ANABELL® ges ut av GRANDET DATA i Lindome och kan rekvireras genom att sätta in 39 SEK på deras postgiro nr 70 61 64-1. Adressen till dem är för dig som vill skriva dit GRANDET DATA, Box 123, 437 00 LINDOME.

I dessa digitala tidevarv är det bra att kunna göra sig en bild av vad som egentligen döljer sig bakom siffrorna. Behovet av grafik och geometri i vår numeriska värld täcks till en del av ANABELL®. Läs och njut av kurvornas mjuka rundning.

GUF

Från bokförlaget Osborne/McGraw-Hill har vi fått en katalog, där vi hittar några intressanta titlar:

Trade Secrets: How to Protect Your Ideas and Assets av James Pooley,
CBM Professional Computer Guide av Adam Osborne med Jim & Ellen Strasma,
VIC-20 User Guide av John Heilborn med Ran Talbott,
PET Personal Computer Guide av Adam Osborne med Jim & Ellen Strasma,
Discover FORTH av Thom Hogan,
6502 Assembly Language Programming av Lance A Leventhal
6502 Assembly Language Subroutines av Lance A Leventhal och Winthrop Saville,
6809 Assembly Language Programming av Lance A Leventhal
6800 Assembly Language Programming av Lance A Leventhal,
Some Common BASIC Programs av Lon Poole och Mary Borchers,
Practical BASIC Programs, redigerad av Lon Poole,
Some Common BASIC Programs, PET/CBM Edition av Poole, Borchers & Donahue,
PET Fun and Games av Ron Jeffries och Glen Fisher,
PET/CBM and the IEEE-bus (GPIB), andra upplagan av Fisher och Jensen.

Dessa böcker kan köpas i bokhandeln eller i din datorbutik eller direkt från Europa-representanten: McGraw-Hill Book Company GmbH, Postfach 630520, D-2000 Hamburg 63, Västtyskland. Tel. 0049-40-538208.

Om någon av VIC-rapports läsare inte vill ha besväret att beställa någon av böckerna själv direkt från Tyskland, skulle vi – förutsatt att tillräckligt många ber om det – kunna hjälpa till från redaktionen. SKRIV I SÅ FALL OCH TALA OM VILKEN ELLER VILKA BÖCKER DU ÄR INTRESSERAD AV! Adressen är:
VIC-rapport,
Gustavsbergsgatan 8,
431 37 Mölndal.

Grossistronden

Här tänker vi presentera de olika företag, som arbetar med dels egna importerade program och utrustningar till VICdatorer och dels med Handic-sortimentet. De flesta säljer huvudsakligen genom återförsäljare, men några har viss postorderverksamhet – det ser du i så fall av deras annonser. Vi försöker berätta litet om människorna bakom företaget och vad som driver dem.

SABA

SABA har funnits i Sverige sedan 1970 och sålts till radiohandeln genom SABA Radio Products AB. 1983 ändrades namnet i samband med omorganisation till SDN Distribution AB.

SABA har också flyttat in i nya, större och betydligt rationellare lokaler på Frövingegatan i Västra Frölunda. SABA är välkänt för sina TV- och Videoapparater men har även komplett sortiment för radiohandlaren.

För oss, som är intresserade av datorer är väl viktigaste produkten de monitorer, som man introducerat och som annonserats tidigare i tidningen. Enligt Handics tekniker är det de hittills bästa som finns för VIC-20 och 64, som man provat.

SABAs adress är Box 2053, 421 02 Västra Frölunda, tel. 031-490 900, men man har ingen direktförsäljning – endast genom återförsäljare. (som finns på de gula sidorna i telefonkatalogen under namnet SABA-återförsäljare) eller via Handics återförsäljarenät.

VIC-20 inklusive introduktionskurs

Utbildningsföretaget LEXICON erbjuder för våren -84 ett kurspaket bestående av en dags utbildning inklusive dator. I priset 2.175:- (inkl. moms) ingår lunch, kursmaterial och datorn VIC-20.

Kursen syftar till att lära deltagarna hur en hemdator fungerar. Man lär sig de enklaste grunderna i programmering, vilket är en grund för vidare självstudier. Olika färdiga program demonstreras, och deltagarna får också skriva egna enkla program. Man informerar också om övrig utrustning som behövs för en effektiv användning av datorn, t.ex. anslutning till en TV, samt användning av en kassettbandspelare.

Kurser äger rum på följande tider och platser:

HV-12	84-01-09	STHLM
HV-15	84-02-02	STHLM
HV-19	84-02-27	STHLM, GBG
HV-112	84-03-19	STHLM, MALMÖ
HV-116	84-04-16	STHLM
HV-119	84-05-10	STHLM, GBG
HV-123	84-07-07	STHLM, MALMÖ

Hösten - mässornas förlovade tid

I slutet av september hölls HEMELEKTRONIKMÄSSAN i Sollentuna. Som namnet säger, ville man markera en skillnad till MIKRODATORMÄSSAN som hålls på våren i samma lokaler.

Hemelektronik tycks vara samma som ZAPP-DING-BANG-KRASH, för det var så det lät. Hemdatorer överallt, och nästan bara spel. Och VIC-20 eller CBM-64 i var och varannan monter. Den svenska marknaden tycks ha bestämt sig.

Här och var fanns små montrar med telefonapparater och miniräknare. Trängseln där var ansenlig, men bleknar till ett intet jämfört intresset för hemdatorerna.

Med intresse konstaterade jag trenne företelser:

* Studieförbund och andra utbildningsanordnare visade upp sig.
* Några läromedelsleverantörer och fack-bokförlag visade att de förstår vad som gäller.
* Flera renodlade programleverantörer fanns representerade, varav många med program för flera olika datorer. Håller mönstret på att sammanfalla med tendensen i andra länder?

PCW - mässan som räknas

Från Sollentuna åkte vi raka spåret till London, där Personal Computer World hade sin sjätte show i rad, för året nyinflyttade i the Barbican. Lokalerna är ungefär fem gånger ytan i Sollentuna, och montrarna genomsnittligt hälften så stora, så nog fanns det mycket att titta på.

Halva ytan var vikt för hemdatorer, och där var det det ZAPP-DING-KRASCH-BANG-SPJING-BONG, eftersom man i England kommit längre i användningen av ljud. Det är också utom all tvekan att grafiken hos de flesta programmen i Sollentuna inte ens hade renderat dem en plats i tamburen på OCW. Med ett lysande undantag: Llamasoft's Grid-Runner, som nog var det mest kopierade programmet av alla. Som du säkert vet, är det Llamasoft's original som fått äran att bära Handic-etiketten i Sverige, och det är verkligen utan konkurrens när det gäller aggressionsspel där man tänker med avtryckarfingret.

Underfundiga spel

Microsoft har en flygsimulator som är något i hästväg. Dessvärre kräver den rejält med minnesutrymme, men vi kan hoppas att den kommer att finnas till den nya 700-PETen.

Hover-Bovver var nog det spel som tilltalade mig mest. Titeln står för ett slarvigt uttal av 'Låna Gräsklippare'. Man går in till grannen och lånar hans gräsklippare. Medan sagde granne jagar en, klipper man för allt vad tygen håller. Men körs gräsklipparen för hårt, blir den överhettad och måste vila en stund. Om man råkar klippa rabatterna också, tillkommer en rasande trädgårdsmästare.

Till all lycka har vi vår trofasta hund, som i viss utsträckning kan bussas på såväl granne som trädgårdsmästare. Efter en viss tid tar dock hundens lojalitet slut, och kräket sällar sig till de jagande. Mycket ljud och en otrolig grafik. Snälla Handic, ta hit den till julen!

Channel 8 Software visade några äventyrsprogram, du vet när datorn talar om hur det ser ut runtomkring dig, och du skall undersöka omgivningen och ta dig genom terrängen. Det som var nytt med dessa spel, varav två fanns färdiga för 64-an, var grafik. I färg. Så du inte bara kan läsa om vad som händer, utan du kan se det också. Helt fantastiskt, hoppas de kommer hit snart.

Äventyrsspel har jag faktiskt inte sett på den svenska marknaden. På Datatronic har man två, icke publicerade spel för 3000-serien. Finns det ingen efterfrågan?

Nätverk

I Commodores monter visades en behändig burk, vilken gör det möjligt att koppla ihop upp till 128 Commodore-enheter. En CBM-64 kan hämta ett program från en 200, skicka programmet vidare till en 3000 som listar programmet på en skrivare samtidigt som 64:an sparar programmet på en 8050. Såvitt jag kunde se, fanns det inte några begränsningar. Ytterligare en möjlighet för dem som har en PET för administrativa rutiner och en 64 för CALC-RESULT att låta maskinerna bolla siffror mellan sig.

Underligt nog visade Commodore inte upp några administrativa program med undantag för CALC-RESULT. Och även om det gick femton Spread-sheet-program på dussinet, var det just CALC-RESULT som drog folk likt flugor samlas kring en sockerbit. Det programmet har verkligen gjort succé, inte bara i Sverige.

Andra intressanta Commodoreprodukter var VIP, en CBM-64 med inbyggd bildskärm och floppy. Mindre, lättare och billigare än Osborne 1. För att inte tala om en liten fyrfärgsplotter till klart överkomligt pris.

Strömproblem

I en monter visades magnetiska nätstabilisatorer, så man tycks uppmärksamma detta problem i Storbrittannien också. Det jag fann mest intressant av allt var dock en elbackup. De stabbar som finns i dag klarar strömavbrott på någon hundradelssekund. Den här lådan gav 700 mA i upp till 20 minuter! Fullt tillräckligt för att i lugn och ro avsluta faktureringen eller vad man håller på med, alltmedan de övriga på kontoret grävt fram gamla ljusstumpar för att få ledljus. Jag lovar att det kändes underligt att slita sladden ur väggurtaget och fortsätta som om ingenting hade hänt. Detta lilla under kommer till Sverige i höst.

J. Stiernborg

Detta vet man väl?

Det är väl verkligen sorgesamt, men allt som står här är säkert sådant som alla redan vet - eller? Naturligtvis kommer många att säga "Javisst, men det vet ju alla". Ganska fånigt då att skriva om det, om det inte vore för att just det som alla tror att alla vet aldrig blir nämnt och att det därför egentligen är så att de flesta inte ens har en aaaaning om att det är så här det förhåller sig.

Det är så många saker som alla vet, att jag här bara skall nämna några få utvalda ämnen och försöka klara ut vad som gäller.

Luddiga siffror, osäkra tal

Alla vet att datorn inte kan rymma decimaler på ett exakt sätt. Inom datorn förvaras decimaler som oändliga upprepade talserier (på samma sätt som en tredjedel är 0,33333333 när det omvandlas till decimaltal). Allt du behöver göra för att kontrollera detta är att be din dator PRINT 8,13 och se vad den svarar.

Så vi vet alla att för noggranna beräkningar måste vi ta hand om detta luddighetsfenomen med noga avvägda avrundningar.

Alla vet att exponentiering eller rotutdragning genomgår en konstig beräkningsrutin som bland annat omfattar logaritmer och exponentialfunktioner och sånt. Följdaktligen vet vi att sådana resultat är nära, men inte exakta. Vi kan få bekräftelse härpå genom att knappa in: FOR J=1 TO 10: PRINT J↑2:NEXT J. Och vi vet naturligtvis att multiplikation inte har detta problem, så att vi borde eliminera detta luddighetsproblem genom att skriva J * J.

Det är också välkänt, att tal som skrivs exakt inte nödvändigtvis är exakta. Om jag ber datorn att PRINT SQR(25) kommer den lydigt att skriva ut kvadratroten ur 25 eller 5. Men vi låter inte datorn lura oss, även om det skrev ut exakt 5 så såg vi SQR-funktionen och har vett nog att vara misstänksamma. Jovisst, om vi skriver PRINT SQR(25) - 5 får vi bekräftelse. Istället för noll får vi ett mycket litet tal till svar!

På tal om det, vi känner väl alla till dom små talen med bokstaven E i sig. Om E:et följs av ett plustecken är det ett mycket stort tal, om det följs av ett minustecken ett mycket litet sådant. E betyder "exponent-skrivning" eller "vetenskapligt skrivsätt" på miniräknare (scientific notation på engelska), men det är naturligtvis

forts. nästa sida

HEMDATORN ÄR HÄR!



Datorer och programmering är inte längre något hokus pokus förbehållet övermänskliga dataknuttar. Äntligen kan familjen få bekanta sig med datorer, framtidens teknik.
Hemdatorn ansluter du lätt själv till teven och sedan kan hela familjen göra egna program, tävla, leka och spela.
Mytech Smådatorer AB är en specialbutik för hemdatorer. Hos oss hittar du hela nio olika modeller. Dessutom har vi givetvis program, tillbehör, böcker och tidningar.
Ta med dig familjen hit och titta, testa och fråga!

MYTECH SMÅDATORER AB

Viktoriagatan 26, Göteborg, 031-11 51 38
Öppet: Vard. 10—13, 14—18, Lörd. 9—14

välkänt... Om du säger till datorn PRINT .00000000123 så får du se talet skrivet i E-form. Men det kände vi ju alla till, eller hur?

Så nu känner vi till så mycket att vi ser upp med när man jämför likheter eller tal som genomgått en "luddighetsskapande" process. Med andra ord måste vi undvika att skriva IF SQR(25)=5.... för datorn kommer förmodligen aldrig att finna de två värdena riktigt lika! I stället för att jämföra likhet bör vi alltså jämföra om talen är varandra "tillräckligt nära" med t.ex.
IF ABS(SQR(25)-5<.0001... och vi får bra resultat av våra jämförelser. Men det visste du väl redan?

Vad händer när man trycker på (RETURN)?

Intuitivt känns det för den mest flyktige betraktaren som om (RETURN)-tangenten egentligen borde hetat "GÖR DET"-tangenten. När du trycker på denna tangent säger du till datorn att göra, utföra, allt som står på den rad du just skrivit och skall gå ifrån. Detta är väl bra, när du skrivit in en rad vill du att datorn skall utföra den.

Det finns emellertid tillfällen, när du inte vill ha en rad utförd (utom möjligen genom dörren). Du kanske har ritat en hemsk Grojka, eller en del av honom, på bildskärmen och vill fortsätta till nästa del av skärmen. Kanske har du försökt rätta en BASIC-sats och klantat till det... vilket ibland händer om du kommer in i det s.k. "programmed cursor mode" eller det sätt datorn arbetar mellan dubbelfnuttar ("). Kanske höll du på och skrev in ett program när någon ringde och du bara skrev upp telefonmeddelandet på en ledig del av bildskärmen. Du vill inte att datorn skall försöka göra allt som står på bildskärmen. Du vill bara hoppa ur till nästa rad utan att göra något annat.

Och naturligtvis vet du att man kan lämna den aktuella raden genom att hålla nere SHIFT-tangenten och trycka på (RETURN). Skiftrad (RETURN) kan också sägas vara en "GÖR DET INTE"-tangent. Det kan hjälpa dig fri ur en massa snåriga lägen.

"Renhållning" eller Sophämtning (Garbage Collection)

Vem känner inte till att vissa slag av strängar kan ge dig en elakartad fördröjning som uppstår vid oberäkneliga tidpunkter. Om du inte visste detta kunde det ju tänkas att du fick för dig att din dator hade "krasch!andat" eftersom den blir helt tyst och inte svarar på tryckning på RUN/STOP-tangenten. Pröva nedanstående lilla program:

```
100 DIM A$(500)
110 FOR J = 1 TO 500
120 A$(J)=STR$(J)
130 NEXT J
```

Kör programmet med RUN, det kommer att ta sju sekunder eller så. När du gjort detta och skärmen säger "READY" skriver du in följande direktkommando:

```
PRINT "DET FINNS"; FRE(0); "BYTES LEDIGA"
```

Vi vet ju alla att här kommer "sophämtningen" att vara inblandad och att det kommer att ta denna rad nästan tjugo sekunder för att skrivas färdig. Under pausen fungerar inte RUN/STOP-tangenten. Du får kort eller ingen paus om du har en PET med BASIC 4.0, men det visste du naturligtvis förut?

Och det är välkänt att fördubbling av antalet strängar kan förlänga "sophämtningen" med faktor fyra, dvs fyra gånger längre än tidigare. Pröva och ändra 500 till 1000 i ovanstående program t.ex. Som väntat torde du få se att fördröjningen kommer att bli mer än en minut!

Naturligtvis vet vi om att det finns metoder att komma runt sådana här fördröjningar. Vi har inte plats för dem just här, men när vi väl vet om dem kan kunskapen ge oss möjligheter att gå runt dem.

Slutsatser

Allt ovanstående visste du förstås redan tillsammans med en förfärlig mängd andra väldigt självklara informationsbitar. Men, och detta är det roliga, du har säkert inte tänkt på att nämna dem för din klasskompis, granne eller klubbkompis. För detta är väl sånt som alla redan vet?

Det visar sig nämligen, att det finns mängder med människor, som aldrig har hört talas om dessa saker. Därför att ingen någonsin tänkt på att berätta för dom om sånt som är självklart – för dom själva!

Gömma rader i ett BASIC-program?

Visste du att det finns många små knep när det gäller att gömma rader i ett program och göra livet surt för den som försöker LISTa dina klurigaste programsteg?

Vi hittade en liten rolig grej i en av våra korresponderande tidningar från andra sidan det stora havet.

Har du någonsin önskat att du kunde "gömma undan" vissa delar av ditt program även om det är helt i BASIC och går att kopiera utan vidare? Här presenteras en enkel och effektiv metod:

Ladda helt enkelt in ditt program i minnet och placera fem kolon (:::::) framför varje programrad som du vill gömma. Sätt kolonen efter programradnumret och framför själva BASIC-satsen. Sedan kan du bara knappa in satserna som följer och utföra dem med ett GOTO. Lista programmet och du får se att din "hemliga" kod har försvunnit! För att få raderna att synas igen behöver du bara GOTO det radnummer som du gav åt den andra halvan av programsatserna nedan.

Pröva det här knepet på det lilla programmet som vi återger här bara för att se hur det fungerar, det är roligt och det kostar dig bara litet mer utrymme för de fem kolonen och de extra programsatserna.

```
10 rem "ett enkelt program"
20 print "denna rad skyddas inte"
```

```
30 print "inte heller den här"
40 :::::print "men det blir denna rad"
50 :::::print "och även den här..."
55 stop: rem "antag att detta är programmets logiska slut
60 rem "programmet på raderna 60000-60020 gömmer raderna
70 rem "och raderna 60030-60050 låter dem titta fram igen
60000 :::::fori = 1024 to 15000
60010 :::::ifpeek(i) = 58andpeek(i+2) =58then pokei,0:i=i+5
60020 :::::next:stop
60030 :::::fori=1024 to 15000
60040 ::::ifpeek(i)=0 andpeek(i+1)= 58andpeek(i+2)=58thenpokei,58:i= i+5
60050 :::::next:stop
```

LÄGG MÄRKE TILL att värdet på 15000 på raderna 60000 och 60030 bara behöver vara ett värde vilket som helst, som är större än antalet bytes som din kodning behöver så att du är säker om att programmet letar efter dina fem kolon genom hela programmet!

Programmeringshandbok till VIC 64

Efter att ha tröttnat på den väldiga grundläggande användarmanualen till VIC 64, ville jag gärna ha en lite mer avancerad bok som beskrev 64ans finesser i detalj. "Programmer's Reference Guide" är en sådan bok. Den är publicerad av Commodore och innehåller i stort sett allt man behöver veta om 64an (den räcker i alla fall ett tag). Handic har nu översatt den till svenska, och ger ut den i form av tre delar. Del 1 är nu helt färdig och finns att köpa.

Vad de tre delarna i stort kommer att innehålla är följande:

Del 1 BASIC
* Programmeringsregler för Basic
* Programmeringstips
* En komplett Basic-ordlista
* Appendix med förkortningar, koder, minneskartor, m.m.

Del 2 GRAFIK OCH LJUD
* Olika grafiska moder och deras användning
* Sprite-grafik
* Ljud- och musikprogrammering
* Appendix

Del 3 MASKINSPRÅK OCH KOMMUNIKATION
* Maskinspråksprogrammering
* Beskrivning av Kernal
* Programmering av in/ut-kommandon
* Appendix

De tre delarna bildar en mycket kraftfull och innehållsrik informationskälla för såväl nybörjaren som den erfarne användaren. Böckerna är inte avsedda att vara läroböcker i programmering, det är meningen att de skall användas som referenskälla när det är något man undrar över. Det mesta man vill veta finns som sagt också där. De innehåller också många exempel och små program som man kan lära sig mycket av. Det är naturligtvis inte möjligt att här beskriva allt som de innehåller, men följande beskrivning visar något av allt det som finns att vänta.

Del 1 BASIC

Denna del består av två kapitel som beskriver 64ans BASIC i detalj. I kapitel 1 kan man lära sig hur operativsystemet (det som tar hand om dina kommandon) i stort är uppbyggt och hur det arbetar. Vidare klargörs för hur man använder variabler och vilket format konstanter skall ha. De speciella tecken som BASICen styrs av gås igenom. De olika regler som bestämmer hur man skall skriva olika uttryck, utföra aritmetiska beräkningar och olika andra operationer beskrivs också. Kort sagt så ger detta kapitel en grundläggande och klar beskrivning av hur man beräknar och lagrar data i BASIC. Allting beskrivs från grunden, så nybörjaren kan lära sig mycket här. Lite programmeringsteknik och tips på hur man gör sina program kortare och snabbare finns också.

Kapitel 2 beskriver alla BASIC-kommandon och funktioner i detalj. Kapitlet består till stor del av en alfabetisk lista över alla kommandon och funktioner. I denna lista förklaras ingående, och ges exempel på, hur man använder varje kommando. Om man vill använda ett kommando men inte är riktigt säker på hur det fungerar, är denna lista en guldgruva. Efter listan beskrivs tangentbordets olika egenskaper, och hur skärmeditorn fungerar. Även om dessa två kapitel är väldigt enkla och inte går in på några mer avancerade saker, är de bra att ha till hands även för den vane programmeraren som vill ha något speciellt klarlagt eller färska upp minnet om något. De olika exemplen kan också ge vissa tips som man inte har tänkt på förut.

Slutligen kommer appendix A-P som faktiskt är ett kapitel i sig. Här kan man hitta allt från olika teckenkoder och matematiska formler till ingående beskrivningar av de olika kretsarna som 64an består av. För den som vill fördjupa sig i t.ex. ljudkretsen, finns mycket att hämta. Några minneskartor samt tabeller över skärm- och ASCII-koder finns också.

Del 2 GRAFIK OCH LJUD

Om det var något man undrade över när det gäller att använda färger och musik på 64an så finns det förklarat här. En ingående beskrivning av olika grafik-moder och hur man kan göra sina egna tecken finns. Hur man använder Sprites förklaras, både för nybörjare och mer erfarna. Man kan ha mycket roligt med 64ans grafik, och det mesta klargörs här. En utförlig genomgång av hur man åstadkommer musik och olika andra ljud kommer sedan. Flera exempel på detta finns, både enkla och avancerade.

Del 3 MASKINSPRÅK OCH KOMMUNIKATION

Detta är en instruktionsbok om hur man programmerar i maskinspråk, och hur och var man använder detta

forts. sid. 70

Nybörjarskolan, del 2

Som vanligt inleder vi med en varning! Detta är rena nybörjarskolan och var och en som följer den bör vara nybörjare. För de följder som kan komma för mer erfarna programmerare svarar vi INTE! Nu drar vi igång igen:

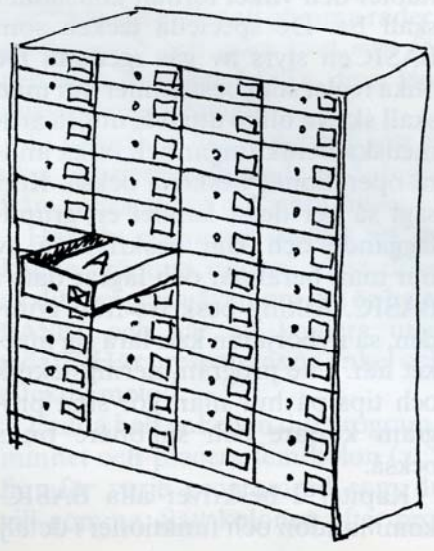
I första delen lärde vi oss litet om olika tangenterna på VIC: Vi lärde oss om markören, om shift-tangenten och Return-tangenten. Vi lärde oss rensa skärmen från "skrotbokstäver och dito tecken", vi lärde oss vad RVS/ON/OFF-tangenterna gjorde och vi fick lära oss ett svar som maskinen gav oss när vi gjorde fel: "Syntax Error" (eller "VIC förstår inte"). Vi fick lära oss skriva in ord och att måla med färgtangenternas hjälp. Vi fick också lära oss vad CLR/HOME-tangenten gjorde och vad mellanslagstangenten användes till. Som avslutning fick vi rita en regnbåge och allra sist fick vi göra ett "Mystiskt experiment".

Vad hände då när du gjorde detta "Mystiska experiment"? Jo, om du knappat in allting riktigt, så fick du bokstäverna JQK "flygande" snett över skärmen som en oändlig mängd flyttfåglar! Det var väl inte märkvärdigt kanske, men roligt, eller hur?

Du använde din VIC som skrivare, när du skrev "?", för det är så att just frågetecknet är en förkortning av kommandot "PRINT" som betyder "skriv". Om du läste artikeln om hur man gör snygga skärmtexter på sidan 32 i förra numret (nr 4) så fick du ännu fler exempel på PRINT-satser.

I förra avsnittet beskrev vi hur du arbetar med tangentbordet. Det är din styrcentral mot datorn. Men datorn har ett par andra viktiga delar också. Det är dels den del som är den egentliga datorn, den som räknar och minns. Den kallas för CENTRALENHETEN och finns på ett s.k. kretskort i samma låda som tangentbordet. Den har ett antal s.k. chips eller kretsar, som har olika funktioner. Vissa av dessa kallas för ROM (kommer från Read Only Memory, som betyder "minne bara för läsning"). Det är kretsar som du bara kan hämta saker från, aldrig lägga in eller ändra något i själv. Andra kallas RAM (kommer från Random Access Memory, vilket betyder "minne som du kommer åt som du vill"). Det program, som översätter det du skriver på tangentbordet till instruktioner som datorn förstår, detta program ligger i ROM-minne så det kan du aldrig förstöra eller ändra. De uppgifter som du skriver in, t.ex. talvärden som du vill räkna med eller texter som du vill spara, det hamnar i RAM-minnen och där kan du både läsa av och ändra innehållet.

Du skulle kunna betrakta minnet i datorn som en stor byrå med mängder av "lådor". Varje låda är en cell i minnet och den har också en etikett, som talar om var den finns, det kallas för ADRESS. När du skall använda minnet måste du antingen känna till den rätta adressen eller också ha talat om för datorn att lådan har ett särskilt namn som du har gett den i ett program. Hur det går till skall vi tala om litet senare i sitt rätta sammanhang. Nu vet du i alla fall hur datorns minne ser ut i stora drag.



Utom minnen finns på kretskortet också kretsar, som utför beräkningar och sådana som styr "samtalet" mellan datorn och eventuella s.k. yttre enheter. Som yttre enheter brukar man tala om tangentbordet, kassettbandspelaren och printern, skrivaren. Även en floppy-disk-enhet är en yttre enhet. En annan är din TV-skärm, där allt du skriver in visas och där det synliga resultatet av dina program kommer fram. För att datorn skall veta hur den skall arbeta måste den styras av särskilda kretsar som sitter på kortet vi talade om förut. Yttre enheter ansluts till datorn med någon av de utgångar eller "portar" som finns på sidan och bakpå datorn. Oftast måste man ha speciella kontakter med många små kontaktställen och du ser en sådan om du tittar på bandspelarkontakten.

Men nu skall vi titta litet på hur din VIC arbetar med beräkningar innan vi går vidare.

VIC-20 som räkneapparat

Vi skall nu se hur VIC-20 går att använda som en superduktig räknemaskin. Samtidigt får vi lära oss om de speciella tecken som VIC använder för att tala om för datorn att den skall addera, subtrahera, dividera, multiplicera eller exponentiera, dvs multiplicera ett tal ett visst antal gånger med sig självt. Dessa tecken kallas för OPERATORER och de utförs av datorn i en bestämd ordning. Precis som i matematiken kan dock ordningen ändras genom att man använder parenteser ( ) för att säga ifrån att något skall göras före något annat.

De operatorer vi nämnde ovan kallas för ARITMETISKA OPERATORER. När man programmerar har man ofta även behov av att kunna göra jämförelser, se på rela-

tioner mellan olika tal eller strängar ("STRÄNG" kallas en följd av tecken som hör samman på ett eller annat sätt, t.ex. bokstäverna i ett namn. En sträng kan skrivas ut på datorn om man omger den med citationstecken: PRINT "namn" ger utskrift av bokstäverna "n","a","m","n" på bildskärmen).

Jämförelse-operatorer kallas också för RELATIONS- eller RELATIONELLA OPERATORER. De är "<" (mindre än,) ">" (större än) "=" (lika med) och kombinationer av dessa som t.ex. "<=" (mindre än eller lika med). Man kan också skriva "<>" (som betyder mindre än eller större än och alltså är det samma som "inte lika med"). Först talar vi om de aritmetiska operatorerna.

Datorn vill endast utföra operatorerna i en bestämd följd som är samma som används inom den vanliga matematiken. Vi listar dem nedan i den ordning de utförs (om man alltså inte ändrat ordningsföljden genom att använda parenteser:
Först utförs exponentiering (↑)
Sedan utförs multiplikation och division (*, /)
Därefter addition och subtraktion (+, -)

Innan vi går vidare måste vi tala om, att VIC-20 inte godkänner kommatecken (,) för att ange decimaler som en del av ett tal. Inte heller kan du använda punkt(.) för att , som vi ibland gör i Sverige, skilja tusental från resten av talet: 34.000 är inte lika med 34000 när datorn ser det. Skriver du 34,00 är detta inte 34 hela och noll hundradelar för datorn utan den tror att du vill skilja 34 från 00 med en tabulering. Kommatecknet används nämligen för att ange en tabulering eller framflyttning på raden av datorn.

Du kan göra beräkningar direkt eller i ett program. I s.k. DIREKT-MODE eller s.k. IMMEDIATE MODE ("omedelbart-utfört") använder du det tidigare kända PRINT-kommandot.

Försök med det här: Knappa in PRINT 6+4 (RETURN).

VIC svarar genom att omedelbart utföra additionen och visa (PRINTa) svaret på skärmen: 10.

Lägg märke till att du INTE ANVÄNDER CITATIONSTECKEN när du använder PRINT för att göra beräkningar. Pröva dessa två exempel för att se varför:

PRINT"6+4" (RETURN). VIC visar 6+4 i stället för 10. PRINT"Summan av 6+4 är ";6+4 (RETURN). Nu skriver VIC ut texten inom citationstecken samt summan enligt vad som skrivits UTANFÖR citationstecknen. Detta är viktigt att komma ihåg!

Nu skall vi pröva litet matematikexempel.

ADDITION

PRINT 500+1000

Du kan själv göra flera exempel med varierande tal. Pröva och se!

SUBTRAKTION

PRINT 6-4

Försök med några fler exempel själv!

MULTIPLIKATION

PRINT 5*5 (OBS! Datorn kräver asterisktecken (*) i stället för X för multiplikation)

Du kan säkert hitta på ännu några multiplikationer.

DIVISION

PRINT 10/5 (Snedstreck används för division).

Gör fler exempel själv.

BRÅKTAL

Datorn kan inte utföra beräkningar med bråktal, utan ser alla tal som decimaltal. Om du vill använda bråktal i dina beräkningar får du innesluta dem med parenteser och se dem som enskilda divisions-uträkningar. Du kan addera 5/10 och 10/5 på följande sätt:
PRINT (5/10+(10/5)

Resultatet blir 2.5. Observera att parenteserna ovan egentligen inte behövs (division utförs ju före addition) utan endast är till för att förtydliga.

DECIMALTAL

PRINT 5.2/5.8 (Lägg märke till att datorn läser punkten (.) som decimaltecken - det är en av anledningarna till att vi ej kan skriva 35.000 när vi menar 35000!). När vi tryckt (RETURN) efter PRINT 5.2/5.8 får vi svaret .896551724. VIC hanterar decimala svar i området från 0,01 till 999999999 precis som vanligt, men tal utanför detta område omvandlas automatiskt till s.k. "Scientific Notation" eller E-form. Detta betyder att om ett tal är för stort att visa eller arbeta med blir det automatiskt omvandlat till en form som datorn kan hantera lättare. (Se mer om detta senare).

NEGATIVA TAL

PRINT -5*10 ... eller ... PRINT -5*-10

VIC känner igen och tar hand om negativa tal, men invecklade formler kan kräva parenteser för att hålla ordning så att negativa tal inte sammanblandas med subtraktioner. T.ex. -5 kanske behöver slås in som (-5) som vi har många operatorer i en enda beräkning.

EXPONENTER

PRINT 2↑2

Den uppåtriktade pilen innebär att 2 skall multipliceras med sig själv en gång eller 2 * 2. I vanligt matematiskt skrivsätt brukar man skriva 2^2. 10 upphöjt till 3 eller 10 * 10 * 10 skrivs alltså i BASIC, som är det språk datorn använder "normalt", så här: 10↑3. Om du inte känner till exponentiering så kan vi här endast säga att man läser "10 upphöjt till 3" som "tag det första talet (10) och multiplicera det med sig självt tre gånger" eller 10*10*10 (=1000). 5↑2 innebär alltså 5*5 (=25). Exponentiering är viktigt att känna till därför att VIC använder exponenter för att återge tal, som är för stora att återge i normal form dvs med alla siffror utskrivna. Vi återkommer med en detaljerad beskrivning.

ANVÄNDNING AV PI

Ett av specialtecknen på din VIC är symbolen "pi". Pi ser ut så här π och den finns på FRAMSIDAN AV PIL-UPPÅT-TANGENTEN. Du kan PRINTa eller visa pi-symbolen på samma sätt som varje annat tecken, men det kan också användas som ett VÄRDE på samma sätt som ett tal. Värdet på pi är 3,14159265... osv och det är förhållandet mellan en cirkels diameter och dess omkrets. Om du multiplicerar diametern på en cirkel med pi får du cirkelns omkrets. Om

forts. nästa sida

du dividerar omkretsen med pi får du cirkelns diameter. För att få fram pi-tecknet på VIC behöver du bara hålla ner SHIFT-tangenten och trycka på "PIL-UPP"-tangenten. Pröva att skriva in följande:
PRINT π

VIC svarar med att visa talet 3.14159265. Tänk dig att du vill beräkna omkretsen på en cirkel med en diameter på 5 cm. Skriv då bara
PRINT π *5 och tryck (RETURN)

VIC svarar med att visa talet 15.7079633. På samma sätt kan du beräkna en cirkels diameter genom att dividera en känd omkrets med pi. Gör några prover själv! Försök t.ex. dividera omkretsen 3145 meter och se vilken diameter den cirkeln har.

Beräkningsordningen

Om du har en invecklad formel eller en lång beräkning med många matematiska operationer, kan det hända att du vill ha beräkningarna utförda i en bestämd följd. VIC räknar alltid på samma sätt som vi nämnde ovan, nämligen i en bestämd ordningsföljd mellan de olika grupperna av operatorer. (Alltså med exponentieringar först, sen multiplikation/division och sist addition/subtraktion). Finns det flera beräkningar i SAMMA GRUPP utför VIC dem med början längst till vänster och går åt höger.

Det är speciellt viktigt om du använder VIC för att lösa ekvationer och utföra flerstegsberäkningar.

Du kan tvinga VIC att göra beräkningarna i den ordning DU vill, om du använder dig av parenteser ( ) för att isolera de beräkningar du vill skall ske före andra. Det är då också möjligt att skriva flera parentespar inuti varandra och då börjar VIC att räkna ut det innersta parentesparets innehåll först.

Låt oss titta på ett exempel. Knappa in följande och tryck på (RETURN)
PRINT 10 * 2↑2/4+3-2

I denna formel kommer VIC att utföra beräkningen i följande ordning: först räknas exponentieringen ut, alltså 2↑2. Sedan kommer multiplikationen 10*4 (längst till vänster alltså före divisionen, som kommer sedan) resultatet av 10*4 delas med 4. Allra sist, i ordning från vänster det också, följer addition av 3 och subtraktion av 2. Om det finns flera operationer i samma grupp (man kallar det också för prioritet eller ordning) utförs de alltså från vänster åt höger.

Vi kan använda parenteser för att tydligare visa hur VIC utför denna beräkning i verkligheten. Här följer en tabell, som visar arbetsgången:

Beräkningssteg	Den beräkning som utförs
10*(2↑2)/4+3-2	2↑2 = 4
(10*4)/4+3-2	10*4 = 40
(40/4)+3-2	40/4 = 10
(10+3)-2	10+3 = 13
(13-2)	13-2 = 11

Du kan ÄNDRA denna beräkning genom att lägga till parenteser så här:
PRINT 10*2↑2/(4+3-2)

I det här fallet kommer VIC först att utföra exponentieringen, därefter multiplikationen, men ... före divisionen kommer den att beräkna uttrycket inom parentesen! Beräkningsstegen ser nu ut så här:

10*2↑2/(4+3-2)	2↑2 = 4
10*4/(4+3-2)	10*4 = 40
40/(4+3-2)	4+3-2 = 5
40/5	40/5 = 8

Hur man använder tal i program

Den här delen av vår kurs skall visa dig hur du använder tal i program.

Du kan använda tal och nummer i dina BASIC-program - inte bara direkt, som du använder tal när du räknar - men även indirekt i form av VARIABLER. Variabler är lättast att beskriva med ett exempel: Antag att du har en storhet i en beräkning och inte vet exakta värdet på storheten. Då kan du använda en variabel för att ändå beskriva storheten i en beräkningssats. Du har kanske behov att räkna ut ytan på olika cirkelytor. Cirkeldiametern kan variera mellan 10 cm och 100 cm. Då kan du i din beräkning införa variabeln "RADIE" och ställa upp din beräkningssats eller -formel utan siffervärden på RADIE: Ytan = pi * RADIE * RADIE.

När datorn räknar med variabler skulle man kunna säga att den ger variabelnamnet åt en särskild plats i minnet, dit programmet sedan går och tittar när det får order att utföra en beräkning. Finns det något värde som variabeln har fått? Då används detta värde vid beräkningen. Finns det inget värde, då måste variabeln få ett värde och det sker med en s.k. TILLDELNING.

Variabler kan alltså betraktas som ett slags koder som representerar tal - de kallas då för NUMERISKA VARIABLER.

Det finns LEGALA och ILLEGALA variabelnamn. Det betyder att din VIC inte tillåter dig att använda vilka namn som helst, inte heller får du skriva variabelnamnen hur som helst. De måste börja med en bokstav! Därefter får följa bokstäver eller siffror. De speciella reglerna för förbjudna variabelnamn hänger samman med de ord som programmeringsspråket BASIC använder för olika kommandon. Inga sådana s.k. RESERVERADE ORD får förekomma i variabelnamn när du skriver program i BASIC!

Vanliga variabelnamn, som börjar på en bokstav, t.ex. AA, BIL, B3, C99 kan anta vilka decimala värden som helst. Det finns en speciell typ av variabelnamn som endast kan anta heltalsvärden och de har ett procenttecken som sista tecken: A%, AA% eller ABCDE%, R4% eller T56%. Observera dock att det bara är de två första tecknen som används för att skilja variablerna åt i BASIC, även om du har längre namn.

En tredje typ av variabelnamn kallas STRÄNGVARABLER. De kan användas för att spara ORD eller TECKEN i stället för värden eller tal. Strängvariabler har ett dollartecken som sista tecken: A$, ABCDE$, R4$, T135$ osv.

Skillnaden mellan de olika variabeltyperna kanske bäst beskrivs om du tänker dig att du ger en variabel värdet 43137. Är det då en vanlig numerisk variabel, t.ex. ABC1, så tror VIC att du vill spara just värdet 43137. Den kan du ändra genom att addera till den eller dra ifrån eller multiplicera eller dividera osv. Om du istället tilldelar en strängvariabel tecknen "43137" är detta inget nume-

riskt värde utan en följd av tecken (i detta fall siffror).

Hur fungerar variabler?

Om du skriver X = 10 så har du just skapat en variabel "X" och tilldelat den värdet "10". Det betyder att överalllt i ditt program när du använder "X" så har du värdet 10 i verkligheten som programmet räknar med.

Säger du nu till VIC att den skall PRINT X. så kommer datorn att skriva "10". Låt oss pröva med något annat, knappa först in NEW (och tryck (RETURN) så raderar du allt gammalt i minnet.) Skriv sedan X = 15 (och tryck på (RETURN)).

Nu har du sagt till din VIC att variabeln X är detsamma som 15. Detta kan ske i s.k. DIREKT MODE utan radnummer. VIC lagrar automatiskt de inslagna variablerna i minnet... även om variablerna nollställs varje gång vi kör ett program med RUN. Vi fortsätter, knappa nu in
Print X (och tryck (RETURN))

VIC svarar med att visa talet 15. Detta sker därför att du skapade variabeln X med TILLDELNINGSSATSEN X = 15. Du kan skapa andra varaibler: Slå in
Y = 3 (och tryck (RETURN))

Nu har vi definierat två variabler, X och Y, där X=15 och Y=3. Fördelen med att kunna göra så här skall du få se nu. Knappa in det här:
PRINT X * Y (och tryck på (RETURN))

VIC multiplicerar blixtsnabbt värdet av X (15) med värdet av Y (3) och visar svaret "45" på skärmen. Utöver direktberäkningar som vi visat här, kan du använda din VIC för att konstruera alla möjliga slags beräkningsprogram med numeriska variabler. Här kommer ett kort program som låter dig slå in två tal som skall multipliceras och sedan visar svaret:

```
10 INPUT X: INPUT Y: PRINT X*Y: GOTO 10
```

Här använder du ett nytt kommandoord, INPUT (som betyder "ge in" eller "skriv in"). Vi skall senare lära oss mer om detta kommando, nu räcker det att du stavar det rätt när du knappar in programraden.

Härmed har vi pratat färdigt för den här gången om VIC som räkneapparat. Som slutexempel skall du få nedanstående program, som ställer upp en multiplikationstabell åt dig. Försök själv att fundera ut hur du skall få fram en annan tabell än den som visas! I nästa avsnitt skall du få ett exempel på ett program där du kan få vilken tabell du vill.

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


VIC-JOUR

PÅ HORNSGATAN I STOCKHOLM!

En söndagsmorgon när du tröttnat på Gridrunner och längtar efter ett schysst adventure — då har vi öppet! En fredagskväll när flexskivorna plötsligt är slut — då har vi öppet! Kan du inte komma ifrån på dagarna för att testa ordbehandlingen på 64:an — kom till oss efter jobbet!

Computer Center på Hornsgatan (vid Mariatorget) har öppet 7 dagar och 5 kvällar i veckan. För din skull.

Vi är specialister på 64:an. Vare sig det gäller skrivare eller diskettstationer. Spel- eller nyttoprogram. Böcker eller utbildning. Så klarar vi dina problem.

Titta in till oss en kväll och provkör Nightdriver eller testa ett bokföringsprogram. Se vad du tycker innan du köper. Det är därför vi har vår testbänk.

Vill du spara 10% på allt du köper? Skaffa vårt rabattkort, det kostar bara 250:— och räcker i flera år!

Så alla hackers & proffs, kvinnor & barn, amatörer & företagare sluta stressa runt. Välkomna till Hornsgatan!

Computer Center Hornsgatan 66. Tel 08-84 07 55.

Öppet vardagar 10.00-19.00
Öppet lördagar 10.30-14.30
Öppet söndagar 10.30-14.30

Svenska Computer Center

— öppet när du har tid —

forts. från sid. 65

Programmeringshandbok till VIC 64

i VIC 64. De olika addresseringsmoderna beskrivs, och en fullständig lista över de olika maskinkodsinstruktionerna finns. De olika sätt på vilka man kan styra minnesorganisationen i datorn beskrivs också. Vidare så ges en lista på de Kernal-rutiner man kan använda i sina program. För varje rutin anges vad den gör, vilka register som ändras och vilka rutiner man eventuellt skall ha anropat innan man använder den. En minneskarta med Commodore's egna labelar finns också. Denna del är ovärderlig för den som skriver i maskinkod. Det enda som man kan sakna är en ingående beskrivning på BASICens och operativsystemets olika rutiner. Men man kan ju inte få allt! M.F.



Från **Conny Jonsson, Herrviksvägen 15, 139 00 Värmdö,** fick vi ett långt och omväxlande brev med både ris och ros och dessutom ett programbidrag! Först till svaren på dina frågor Conny: Du kommer att få alla under 1983 utkommande nummer, och vi beklagar bara, att det strulade till sig vid distributionen av de tidigare numren. Orsaken ligger i personalbrist som delvis berott på att VIC-datorerna blivit så förfärligt eftersökta och att man helt enkelt inte hunnit med ute på Handic Elektronik att få fram och utbilda folk i behövlig takt. Nu börjar dock eftersläpningarna att hämtas in efter en massiv insats under våren, sommaren och hösten. Vi på redaktionen hoppas verkligen detta nummer kommer till dig i tid!

Tack för ditt programbidrag, som publiceras på annan plats här i tidningen. Vi ser fram mot fortsatt samarbete, det vore trevligt om fler följer ditt exempel och passar på och skickar programbitar samtidigt med brev med frågor till oss...

Tage Haglund, Lavendelgatan 4, 440 45 Nödinge, frågade om VIC RABBIT redan i förra numret, men nu har vi fått fram uppgifterna:

VIC RABBIT är ett höghastighets s.k. kassettinterface. Det är ett ROM (Read Only Memory)-plug-in-cartridge som tillverkas av Eastern House Software i USAmerika. Det sätts i VICs expansion port och lägger till 12 kommandon till VICs BASIC. Med cartridgen (finns det förresten inte nån därute som har ett bra namn på plug-in-grejerna till VIC-arna? Cartridge är ju egentligen patron, men det låter inte så trevligt på svenska och "plug-in" låter ännu sämre. Kom igen nu med ett kort och träffande namn så kan du vinna ett spelprogram ur Handic Softwares katalog!) Med VIC RABBIT skall följa en åttasidig instruktionsbok, som dock ansågs av en recensent som något förvillande då den även talar om VIC RABBIT i förbindelse med 64:an, med PET 2001, 4001 och 8032! Någon svensk bruksanvisning finns inte och troligen ingen svensk importör heller.

Adressen är i alla fall:

Eastern House Software, 3239 Linda Drive, Winston-Salem, NC 27106, USA. Angivet pris är US$ 39.95. OBS! Skicka inte pengar på vinst och förlust, skriv först och hör om man vill sända något och vad portot o så fall är. Tänk på att det kan dröja flera månader om du låter dem sända s.k. surface mail (ytfrakt) och att luftpostportot i USA är dyrare än här!

Fråga: Hur är det med de olika minnena i 64:an? Jag har hört att man haft flera olika ROM-kapslar och att de skiljer sig åt. Hur tar man reda på detta och vilka effekter har de olika ROM-arna?

Svar: Man har haft tre olika Kernal ROM i 64:an. Ändringarna mellan dem innebar vissa förbättringar av maskinfunktionen, men det finns en bi-effekt. När man POKEar till skärmen med den första ROM-kapseln, fick man tecken synliga därför att färgen i färgminnet var ställd till VIT. När det gäller det andra ROM-et hade man satt färgminnet till samma färg som skärmbakgrunden, så man såg inga POKE-ar om inte man även POKEade bakgrundsfärgminnet.

I den tredje och sista ROM-kapseln har färgen satts till samma som markörens. Färgminnet i ROM ändras bara när man använder CLEAR SCREEN (eller CHR$(147)). Detta påverkar vissa av de färdiga program som finns. Commodore i USA har informerat de olika programvaruhusen om möjliga problem och hjälper också till där med att korrigera programmen. Själva ändrar man sina 656 public-domain-program och de torde alla nu vara anpassade till den sista typen av ROM.

För att kontrollera om du har kernal 1 eller kernal 2 kan du PEEKa 65408 (hex $FF08) Kernal 1 håller värdet 170, kernal 2 värdet 0.

VIC-klubben, var blev den av?

Fråga: hur är det med programbytarverksamheten i den gamla VIC-klubben? Finns den inte alls, eller är det något ni kommer att ta upp i tidningen?

Svar: Programbytarverksamheten kom aldrig riktigt igång inom VIC-klubben. Dels berodde det väl på att ingen hade tid att hålla i verksamheten och dels på att intresset inte var så stort. Klubbverksamheten var också mycket låg under 1982, kanske delvis därför att ingen ägnade sig åt arbetet på heltid – och det är nästan ett måste, om en rikstäckande klubb skall bli till någon större nytta för medlemmarna. Naturligtvis berodde det också på att med-

lemmarna i en rikstäckande förening eller klubb med nödvändighet inte träffas så ofta som behövs för att man skall kunna dela upp arbetsuppgifterna så inte en enda människa blir pressad till utmattningens gräns av allt som behöver göras.

Nu har man i alla fall bestämt, att någon central VIC-klubb skall inte finnas i den gamla meningen. Däremot kommer vi i VIC-rapports spalter att på allt sätt uppmuntra till lokala klubb-bildningar. Vi har för den saken börjat plocka ihop material, som skall ingå i en s.k. "startsats" eller ett startpaket, som kan rekvireras av den, som vill starta igång en lokal VIC-klubb. Du kan läsa om startsatsen på annat håll i tidningen.

Man bestämde också i höstas att alla medlemmarna i VIC-klubben skulle erbjudas samtliga nummer av tidningen som utkommer under 1983 och att de skulle kunna prenumerera på VIC-rapport för 1984 för en kostnad om 120:- för de 10 nr som beräknas komma under 1984. Det blir två dubbelnummer, ett under sommaren och ett till jul, men annars skall tidningen komma punktligt en gång i månaden - det har vi fått lova Pressbyrån!).

Till alla VIC-klubbmedlemmar kommer det att sändas ut en inbjudan till prenumeration, så är du medlem nu kan du vänta lugnt, det

Alla andra, som också vill prenumerera, kan göra det genom att sätta in 120:- på postgiro 846 46-9, VIC-rapport. Glöm inte att skriva (texta) namn och adress på talongen på postgirokortet. Blir det fel där så vet ju ingen vart vi skall sända tidningen! Skriv också gärna "Prenumeration 1984-A" också, så vet vi säkert vad pengarna avser.

Beställning av gamla tidningsnummer

Den som prenumererar, kan samtidigt beställa gamla nummer av VIC-rapport och dess föregångare VIC-News. Tyvärr är årgång 1 nr 1 och årgång 2 nr 2 helt slut, men nu fins det kvar av årgång 1 nr 2/3 samt årgång 2 nr 1, 3 och 4.

Dessa gamla nummer säljs så länge de finns för 10:-/st (dvs nr 2/3 årg 1 kostar 20:- då där sitter två speciallistningar också till VIC Super Expander och till VIC Maskinkodsmonitor) PLUS PORTO: Posten vill för 1 tidning ha 6,25, för 2-3 tidningar 9:- och för fler än 3 tidningar 12,50!).

Samma postgironummer gäller och om du bara skriver tydligt på postgirotalongen vad du beställer så behöver det inte bli några missförstånd. Glöm bara inte portot!

Innehållet i gamla nummer

För att göra tillvaron lite lättare för alla dem, som inte fått tag i de första numren av VIC-News tänker vi någon gång under 1984 ta fram "godbitarna" ur dessa äldre tidningar. Det får du läsa om i VIC-rapport under våren!

Fråga: Telefonmodem till VIC 20/64, kommer det någon gång och vart kan man ringa för att ansluta sig till något telenät och vad kostar det?

Svar: Hela frågan om anslutningsmöjligheter för VIC-modem, databaser och Teledata-nät håller för närvarande på att utredas. Det kommer någonting, men vi kan ännu inte ge någon detaljbeskrivning. Du, som är intresserad, följ md i VIC-rapports spalter - vi skall försöka gå ut tidigt med nyheterna när de är verklighet.

Fråga: VIC-klubb i Falun, finns det?

Svar: Vi vet inte. I varje fall har ingen talat om det för oss på tidningen. Kanske är det på sin plats att åter upprepa:

KLUBBAR HÖR AV ER!

Ni får fri annonsplats i tidningen på 1/4 enkelspalt, dvs 59 mm brett och 60 mm högt utrymme, där ni kan annonsera om möten, verksamheter, byten, m.m.

UTNYTTJA DENNA MÖJLIGHET!

Fråga: Jag kan inte hitta uppgifter om PEEK(197)-tangentkoder för min 64. Jag har tittat både i användarhandboken och i Programmers Reference Guide.

Svar: Det verkar som om någon glömt bort denna tabell. Var inte ledsen ändå, för nu kommer den!:

Nr	Tangent	Nr	Tangent	Nr	Tangent	Nr	Tangent
0	[INST]	17	R	34	J	51	[CLEAR]
1	[RTRN]	18	D	35	0	52	[RIGHT SHIFT]
2	[CRSR R]	19	6	36	M	53	=
3	[F7]	20	C	37	K	54	↑
4	[F1]	21	F	38	O	55	/
5	[F3]	22	T	39	N	56	1
6	[F5]	23	X	40	+	57	←
7	[CRSR D]	24	7	41	P	58	[CNTRL]
8	3	25	Y	42	L	59	2
9	W	26	G	43	-	60	[SPACE]
10	A	27	8	44	.	61	[CMDR]
11	4	28	B	45	:	62	Q
12	Z	29	H	46	@	63	[RUN/STOP]
13	S	30	U	47	,	64	NO KEY
14	E	31	V	48	[POUND]		
15	[SHFT]	32	9	49	*		
16	5	33	I	50	;		

Nr	Tangent	Nr	Tangent	Nr	Tangent	Nr	Tangent
0	[INST]	17	R	34	J	51	[CLEAR]
1	[RTRN]	18	D	35	0	52	[RIGHT SHIFT]
2	[CRSR R]	19	6	36	M	53	=
3	[F7]	20	C	37	K	54	↑
4	[F1]	21	F	38	O	55	/
5	[F3]	22	T	39	N	56	1
6	[F5]	23	X	40	+	57	←
7	[CRSR D]	24	7	41	P	58	[CNTRL]
8	3	25	Y	42	L	59	2
9	W	26	G	43	-	60	[SPACE]
10	A	27	8	44	.	61	[CMDR]
11	4	28	B	45	:	62	Q
12	Z	29	H	46	@	63	[RUN/STOP]
13	S	30	U	47	,	64	NO KEY
14	E	31	V	48	[POUND]		
15	[SHFT]	32	9	49	*		
16	5	33	I	50	;		

Fråga: Kan VICMON och SUPER EXPANDER användas samtidigt med hjälp av expansionskortet?

Svar: Nej, även om en del funktioner fungerar korrekt, är vissa minneslägen i dessa båda insatser samma vilket kommer att påverka programmeringen.

Fråga: Kan man få skärmen att "rulla" baklänges, dvs nedåt på VIC-20?

Svar: Det är möjligt att få "rullningen" att gå så att säga tvärtom, men då måste du skriva en maskinspråksrutin för detta. Det finns ingen rutin i ROM som har denna effekt. Vi hittade i en amerikansk tidning som vi samarbetar med en rutin som just gör en "Backward scroll". Man säger där att för att få "baklängesrullning" på VIC-20 skall man först skriva Print "Home Down Back Insert Space". Sedan skall man POKEa 218,158 för att tala om för VIC:en att den andra raden inte är en fortsättning på första raden. Följande exempel visar hur man kan göra:

```
10 PRINT "■■■■ "
20 POKE 218,158
30 GOTO 10

READY.
```

Fråga: Hur kopplar man ur "RUN/STOP"-tangenten på VIC20? Jag håller på och arbetar med ett förskoleprogram och använder GET-satser i stället för INPUT när jag vill ha ett användarsvar.Olyckligtvis nog kan ett barn bryta ner programmet genom att oavsiktligt trycka på RUN/STOP-tangenten. Jag kan inte koppla ur denna tangent på det vanliga sättet med GET A$ och CHR$ ("X")-funktionen. Antagligen måste man POKEa nånstans, men var?

Svar: Du har helt rätt. Stäng av med POKE 788,194 och sätt på igen med POKE 788,191. Observera dock att då stoppar även realtidsklockan (TI och Yi$).

Michael Schöberl i Degerfors undrar hur man bär sig åt för att sortera tal. Han nämner att han är nybörjare och vill gärna ha hjälp med data. I det här numret finns flera artiklar som behandlar just sortering. Varsågod!

Kopiering av DOS

Per Ljungberg i Norrköping frågar hur man kan kopiera programmen "C-64 WEDGE" och "DOS 5.1". Dessa program följer med då man köper skivminnet 1541, och lägger in en "wedge" som gör att man lättare kan använda olika disk-kommandon. "C-64 WEDGE" är ett kort BASIC-program som helt enkelt läser in och startar upp "DOS 5.1". Eftersom "C-64 WEDGE" är skrivet i BASIC är det inga problem att kopiera detta till en annan skiva, det är bara att ladda in det, byta skiva, och SAVEa det. Med "DOS 5.1" är det emellertid inte lika enkelt. Det är ett maskinkodsprgram som laddas in på adresserna 52224 (CC00 hexadecimalt) till 53082 (CF5A hexadecimalt). Om man har en maskinkodsmonitor kan man med hjälp av den spara programmet med följande rad: .S "DOS 5.1",08,CC00,CF5A. Om man inte har en sådan monitor kan man spara det från BASIC genom att sätta några pekare till rätt värden. Två pekare används när BASICen sparar ett program. Adresserna 43-44 pekar på programmets första byte, och 45-46 pekar på den sista byten plus ett (den byte som 45-46 pekar på sparas ej). Genom att sätta dessa pekare kan man SAVEa vilken area i minnet man vill. En sak som man då måste känna till är att pekare lagras med formatet låg-byte, hög-byte, där adressen är hög-byte * 256 + låg-byte. För att 43-44 skall peka på adress 52224 skriver man följande: POKE 43,0: POKE 44,204 (204*256+0 = 52224). Låter det krångligt? Om du tycker det, följ bara följande instruktioner för att kopiera programmet.

1. Ladda in programmet med LOAD "DOS 5.1", 8,1: NEW (eller genom att köra "C-64 WEDGE").

2. Skriv POKE 43,0: POKE 44,204: POKE 45,90: POKE 46, 207 och slå RETURN.

3. Spara programmet med SAVE "DOS 5.1",8.

4. Återställ pekarna med NEW.

Fråga: Vad blev det av med serien "Listskydd" som började i nr 1, årgång 2, sid 8? Jag skulle bli glad över en fortsättning.

Svar: Vi vet inte var författaren blivit av, men vi kanske kan få letat fram honom under våren när snön smälter igen? Skämt åsido som charkuteristen sa, jag lovar att gräva ner mig i problemet och komma med en utförlig rapport när jag vet något. (Red.)

Lagring av maskinspråksprogram

Hur lagrar man maskinspråksprogram i form av DATA-satser är en av frågorna Christer Jonsson har skickat in. Ett maskinspråksprogram består av olika koder som talar om för processorn (datorns "hjärna") vad den skall göra. Dessa koder utförs sekventiellt, och naturligtvis måste programmet börja någonstans. Att man lagrar ett maskinspråksprogram i DATA-satser innebär helt enkelt att de olika koderna lagras i DATA-satser. Dessa koder läses sedan och POKEas in från startadressen och uppåt. Följande lilla BASIC-program är ett exempel på hur man på detta sätt laddar in ett maskinspråksprogram i kassett-bufferten (den börjar på adress 828). Programmet som körs med SYS 828, reversar alla tecken på skärmen. Programmet är skrivet för VIC 64. Lägg till följande rad för att det skall fungera för VIC 20: 120 POKE 829,30: POKE 852,32.

```
10 FOR I=828 TO 855
20 : READ A: POKE I,A
30 NEXT
40
50 PRINT "■■STARTA MASKINKODSPROGRAM MED : ■SYS 828■."
60
100 DATA 169,4,133,79,169,0,133,78,160,0,177,78,73,128,145,78,200,208
110 DATA 247,230,79,165,79,201,8,144,237,96

READY.
```

Mister Tid-Bit:

GRÄNSLÖST ROLIGT?

Hur klarar man sig i livet utan en hemdator?

Ska man tro alla lovord om denna 80-talets välsignelse (och de är, det ska gudarna veta, många nuförtiden), så riskerar man att bli något av en andlig krympling, om man avstår från intelligent konversation på bildskärmen.

Och vem vill vara själsligt förkrympt i en tidsålder som kräver av oss stackars medborgare att vi ska kunna både räkna och skriva? Ställd inför sådana ohemula krav är det skönt med en hjälpare i nöden: en liten hemdator vars intelligenskvot åtminstone inte behöver ifrågasättas.

Nu är det ju det här med språket, förstås.

Förr i världen räckte det med att tala med bönder på bönders vis och med lärde män på latin. Det torde heller inte ha berett allmogen några större svårigheter. Men ska man tala med sin dator, så förstår inte han (eller kan det möjligen vara en hon?) ett enkelt "Ave Vic" som artig inledning på konversationen.

Som svar får man nämligen ett "Syntax Error" och det låter varken trevligt eller särskilt uppmuntrande så här goda vänner emellan. Alltså tvingas man lära sig Basic i stället. Det är datorns eget språk det. Latin och Bond-språk göre sig inte besvär.

Basic ska det vara – dataålderns eget esperanto!

Sedan har vi det här med VERSALER också.

Första gången man ser sina enkla små instruktioner till datorn spruta fram på bildskärmen blir man smått förskräckt. Nästan allting kommer nämligen fram med STORA BOKSTÄVER. Vid andra andhämtningen ger det en känsla av makt. Det är ju kraft och tyngd i det man säger.

En sådan versalernas mästare var Torsen Tegnér, gamla Idrottsbladets skapare och främste skribent. Hans språk vimlade av kursiver, understykningar men framförallt av versaler. Därmed markerade han allt han ansåg särskilt viktigt i sina texter. Hade han fått uppleva en hemdator med möjlighet till ordbehandling hade han kunnat fira formliga excesser.

Nu kan den mest försynta lilla människa göra det i stället. Sällan eller aldrig behöver man tillämpa den gamle prostens marginalanteckninar i manuskriptet till predikan: "Argumenteringen svag, höj rösten!"

Inför Vår Dator är alla ord lika!

Därmed borde också all rädsla vara försvunnen.

I den trygga förvissningen om att det dessutom finns statsmannaord på att man "ingenting har att frukta utom fruktan själv" borde vem som helst våga slå sig ner vid tangentbordet utan att rädas tekniken.

Datorn bits ju inte. I dess föreställningsvärld är nämligen "bits" detsamma som bitar i dess minnesutrymme. På sin höjd kan man därför endast beskylla sin hemdator för att vara långsint.Till skillnad från människan glömmer den inte men gör heller inga moraliska värderingar av rätt eller fel. Lär man sig bara acceptera den lätt förebrående tonen i "Syntax Error" har man inget att frukta.

Annars kan man ju söka hjälp på annat håll. Björn Vinberg i Expressen t.ex. beskrev för dryga miljonen läsare hur han sökte och fann tröst hos en expert på området, en docent i Lund som tillika var hans svåger.

Denne lär numera ha hemligt telefonnummer.

Vårt behov av tröst är omättligt, skriver skalden.

Men det var uppenbarligen innan både svågern och hemdatorn uppfanns. Forna tiders radioter och TV-frälsta har ersatts av Hackers. De lär enligt förljudande vara vårt nya datafrälse. De skriver eller "hackar" på tangentbordet, inte på andra. Det senare har de inte tid med.

En Hacker befinner sig på Bit-nivå, säger de invigda, och poängterar med eftertryck att ordet "Bit" ska uttalas med kort i-ljud, ungefär som i "ritt". Kanske beror det på att man på denna avancerade kunskapsnivå bildligt talat sitter säkert i sadeln.

Den som ännu inte nått dit kan möjligen trösta sig med att ordet "hack" också betyder hästkrake.

Det finns inga gränser. Det är bara gränslöst roligt, säger reklamen om hemdatorn och dess möjligheter.

Om nu detta är sant, så bör det heller inte finnas några hinder för en Hacker att bli förmögen en dag. Rimligen skulle det gå att göra ett program som ökade sannolikheten för 13 rätt på tipset upp till maximum. Sedan borde det inte vara svårare för hemdatorn att producera en vinstrad än att göra en hushållsbudget. Den lär ju kunna göra det även i svångremstider.

Lars-Gunnar Björklund tvingas säkerligen rådfråga processorn, själva hjärnan i Tipstjänsts egen dataanläggning, för att komma tillrätta med alla inströmmande, hemdatabaserade tipsrader. Frågan är om han skulle finna det "gränslöst roligt" med detta Hjärnornas Krig.

Det finns ju gränser – för tålamodet.

Kanske är det bättre att låta hjärnorna slåss i rymden.

Det är bara att låta sig förflyttas fram till år 2190 för att få nöjet att helt riskfritt skjua ner droider med plasmakanoner. Det är inte så ruskigt som det låter. Man har en Joystick till sin hjälp. Som namnet säger, är det den som svarar för det roliga. Några krav på att man själv måste känna glädje finns inte.

Detta innebär givetvis en fasansfull begränsning hos den gränslösa hemdatorn: den tar inget som helst ansvar för att man själv blir glad. I det fallet har den en andlig kusin i svensk TV.

Som kompensation borde hemdatorns produktutvecklare försöka konstruera en Sadstick. Den kunde med fördel användas i t.ex. deklarationstider. Man kan ju nämligen underlätta sin självdeklaration med den förträffliga datorns hjälp. Att deklarera är trots detta ingen upplyftande sysselsättning. En Sadstick skulle svara för det ställföreträdande lidandet under denna eljest så

forts. nästa sida

melankoliska period.

Så långt har dock inte Riksdagen tänkt.

När man fattade beslut om att investera i att datautbilda hela svenska folket hade man vare sig glädje eller sorg i tankarna. De rationella motiven inför vårt inträde (och kvarstannande) i datasamhället avgjorde säkerligen det kloka beslutet.

Sannolikt var åhörarläktaren fylld av data-intresserat folk under debatten. För dem hemtama uttryck bör ha korsat luften när Riksdagen talade Basic, ett rakt och exakt språk, fritt från dialekter, substantivsjuka och krånglig meningsbyggnad av tysk, grammatikalisk modell.

För en gångs skull blev riksdagsspråket begripligt. För en gångs skull behövdes inget Syntax Error.

T.o.m. Torbjörn Fälldin lär ha varit till sin fördel.

Nu är det bara att hoppas att Riksdagen själv föregår med gott exempel.

I stället för en betald bridge-resa till England skulle man t.ex. kuna utrusta kortintresserade ledamöter med varsin hemdator. På sikt skulle det bli mycket billigare om de satt i sina bänkar och spelade med datorn som motståndare. Hela riksdagsarbetet skulle för övrigt bli mycket intressantare för dem. Vanan att "knappa" har de ju redan från den elektroniska voteringsmaskinen.

Finge man sedan se våra folkvalda ombud i lite gladare kläder också, så kunde Riksdagen bli en riktigt anslående institution. En färgglad folktröja på varje riksdagsman och -kvinna skulle lysa upp den grå vardagen.

Vid åsynen av denna nya ämbetsdräkt skulle heller ingen behöva sakna Karl XI:s färgstarka drabanter vid Riksdagens högtidliga öppnande. I stället kunde alla begrunda Riksdagens välbehövliga nya motto på våra folkvaldas bringor:

Kunskap ger försprång!

Säljes VIC-20
Spelprogram till VIC-20 säljes till löjligt låga priser t.ex. TRAXX: 20 kr, DEFENDA: 15 kr, SPACE PHREEKS: 20 kr m.m. Tel. 018-32 46 67 ring efter kl. 18 fråga efter Franz.

GRATISANNONSER

VIC20 program och spel säljes, ex: Databas 70:-, Ormen 20:-, Othello 20:-, Frogrun 20:-, Saucer Attack 2:-, Backgammon 20:-, Logger (+3K) 20:-. Vid likvid till pg 459 26 40-9 fritt. Skicka efter lista eller beställ direkt från Tore Gullstrand, Norrgårdsv. 127, 184 00 Åkersberga.

Nytto- och spelprogram
Svenska VIC-20-program. Sänd efter katalog nr 2 1983. Rego Granlund, Kalkvägen 3, 902 42 Umeå.

VIC-20 ** VIC-64
Vi säljer adventures och arcadespel av högsta klass. T.ex. Secret of Kublai, Secret of Neptune, Time Travel 1 = adventures utan like. Space Trouble, Star War, Bubbles = arcadspel som inte går av för hackor. Ring 0570-141 67 fråga efter Johan.

VIC-20 tillbehör
8 K RAM + Programmer's AID 675:-. Maskinkodsmonitor + manual 300:-. Handbok "Grafik och ljud" 50:-. Telefon 031-44 62 25, Fredrik Olsson, Partille.

Diamond Heist
Högklassigt amerikanskt grafikäventyr för oexpanderad VIC-20. Gör inbrott genom en rad livsfarligt bevakade rum! Garanterat fängslande! Sätt in 55:- på pg 58 89 07-6 (OBS! ange Joystick eller Keyboard), Svernlöv, Linköping.

Travspel till VIC-64
Spela vinnare resp. plats. Bjud in dina vänner och testa era spelnerver, men sitt inte uppe hela natten. Bra, välgjort, realistiskt. 9 KB på kassett.
Biorytm - beräkna din och dina vänners Biorytm. Båda programmen för 49:- + frakt. OBS!! Ej listskydd. R Larsson, Furutåvägen 16 B, 352 54 Växjö, tel. 0470-813 23. Pg 456 05 17-7 = fraktfritt.

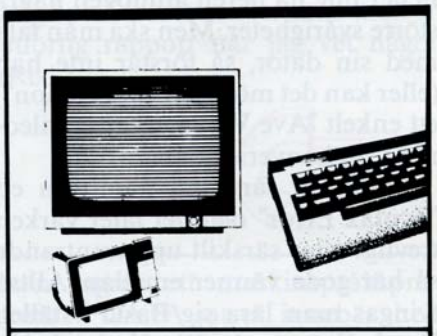
VIC-20 Assembler MM
Assembler. Gör det lättare att skriva i ren maskinkod. Å; Ä, Ö samt ytterligare 60 nya tecken, t.ex. fonetiska alfabet, bilar, UFO m.m. Paketpris endast 149:-. Lars Olsson, L. Anrås 6511, 444 00 Stenungsund. Tel. 0303-776 15.

VIC-20
Säljes: Road Race Pluginspel (130 kr), BASIC på VIC-20 (50 kr), 6502 games (10 maskinkodspel, 100 kr) och Termometer (50 kr). Håkan, tel. 0500-410 67.

Säljes
Säljes VIC-20 tillb. 32 K minne 750 kr, 3 K minne 200 kr. Exp.låda 6 fack 750 kr - paketpris 1400 kr. Tel. 031-23 01 44, Robert.

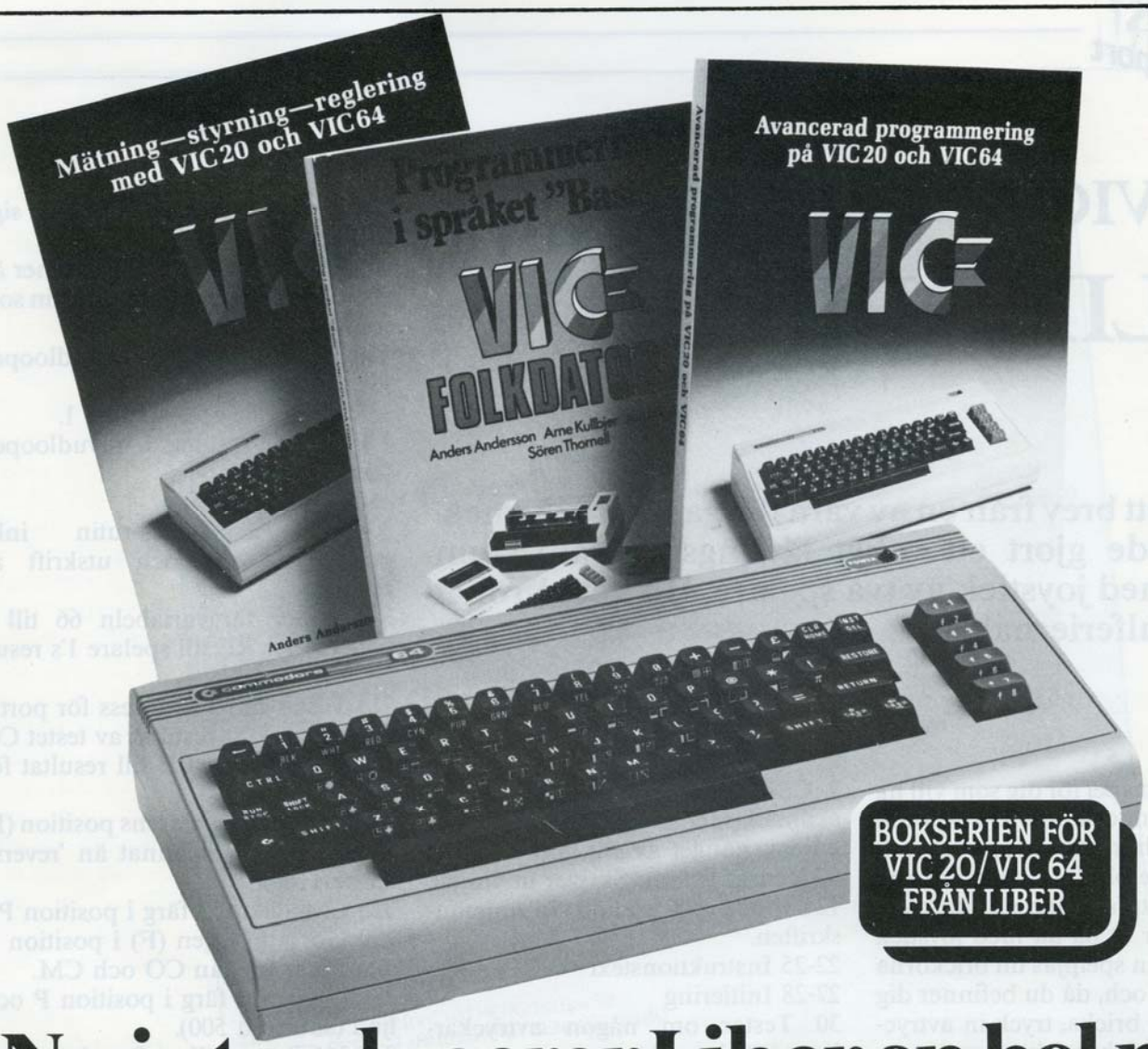
Skatteplaneringsprogram
för VIC-20/VIC-64 - räknar ut skatten vid deklarationen, planerar för skattereformen 1983, 1984. 1985.

180 kr för fyra program med utförlig instruktion. Anders Andersson, Poppelv. 55, 135 52 Tyresö, Pg: 80 94 71-6, tel: 08/7124646.



TIPSRUTAN

Från Erik Sölving i Spånga kommer tipset att VIC kan användas som ett ytterst bra supplement för en videofilmare, som vill ha en textapparat! Han undrar också varför dator- och film-folk inte funnit varandra. Men där är det väl så att de har funnit varandra i de kommersiella sammanhangen men inte i hobbysvängen. Tänk bara på alla roliga effekter man brukar leka med på Sveriges TV! Många av dem är gjorda med datorkraft. Är det någon där ute, som känner sig hågad att skriva en vers eller två så är hon/han välkommen i våra spalter!



Nu introducerar Liber en hel ny serie böcker för VIC 20/VIC 64!

* Sannolikt de viktigaste – och mest omfattande – böcker som någonsin skrivits för VIC-datorerna!
* Endast 3 delar som täcker allt: från BASIC till avancerad programmering och tekniska tillämpningar.
* Passar alla: nybörjare, mer erfarna programmerare, tekniker.
* Oumbärliga för alla som vill lära sig behärska VIC 20/VIC 64 till 100%.

Liber

Finns i bokhandeln eller hos din VIC-handlare!

För VIC-64

"BLINDBINGO"

Vi fick ett brev från en av våra flitiga läsare i Umeå. Han hade gjort ett roligt tävlingsprogram, som spelas med joystick av två spelare. Ha det så roligt under julferierna!

Det här är ett spel för dig som vill ha någonting annorlunda. Du och din motspelare har var sin joystick och tävlar om de olikfärgade brickorna som finns utplacerade på skärmen. Det hela går ut på att med joystick styra din egen spelpjäs till brickorna på skärmen och, då du befinner dig mitt över en bricka, tryck in avtryckarknappen och brickans poäng är din. Enkelt?? Nja, om du kunde se din spelpjäs vore det ingen konst men eftersom du inte kan se spelpjäsen så måste du lita till din hörsel. För varje steg din spelpjäs tar så avger den ett karaktäristiskt ljud.

För varje prickad bricka får du poäng - olika poäng för olika färg på brickan. För missad bricka får du avdrag på poängen. Framgång i BLINDBINGO kräver framför allt snabba strategiska beslut men också ett visst handlag med joystick. Spelet är ganska svårt så träna på egen hand först innan du utmanar en motståndare.

Först till 100 poäng - BINGO!

Eftersom programmet är ganska långt så kan du som vill slippa mödan med att själv knacka in det i maskin få en kopia på kassett genom att sätta in 35:- på pg 476 96 08-3, Gunnar Berglund. Skriv 'BLINDBINGO' på inbetalningskortet.

För klaring av programlistningen till Blindbingo.

7 CS= startadress på ljudchip & skillnad mellan skärm- och färgadress: stänger av allt ljud.
10 Rensar skärm: Skriver ut vinjett.
12 Subrutin 500 ger ljud vid vinjettutskriften.
22-25 Instruktionstext
27-28 Initiering
30 Testar om någon avtryckarknapp är intryckt
40 Testar om 'S' är intryckt.
60 Testar om avtryckarknapp för port 1 är intryckt och släcker spelpjäs 1 som resultat, Subrutin 455 ger ljud för pjäs 1.
70 Testar om avtryckarknapp för port 2 är intryckt och släcker spelpjäs 2 som resultat: Subrutin 459 ger ljud för pjäs 2.
75 Tänder spelpjäs 1 och 2.
98-145 Huvudloop
98 Sätter J till adress för port 1: Sätter variabler för X- och Y-position för spelare1.
100 Jfr 98 fast för spelare 2 (port 2).
110-116 Läser joystick och anger som resultat förändringar i x- och yläge för spelpjäsen.
117 Ny x- och y-position.
118-121 testar om spelfältets ram är nådd.
124 Beräknar skärmadresen.
125 Testar om joystick inte är i neutralläge och går till 'stegljud' som resultat.
126 Testar om avtryckarknappen är intryckt och går till träff/missrutin som resultat.
128 Testar om spelare 1 befinner sig i huvudloopen
129 Testar om spelare 2 fått mer än 99 poäng och går till triumfrutin som resultat av testet.
130 Sparar resultaten i huvudloopen för spelare 2.
138 Jfr 129 fast för spelare 1.
140 Sparar resultat i huvudloopen för spelare 1.
150-220 Triumfrutin.
300-445 Träff/miss-rutin inkl. poängberäkning och utskrift av poäng.
310 Sätter färgvariabeln 66 till 1 (vitt): sätter RE till spelare 1's resultat.
315 Testar om J = adress för port 2 och sätter som resultat av testet CO till 2 (svart) och RE till resultat för spelare2.
320 Testar om spelarens position (P) innehåller något annat än 'reverse space' (160).
325 CM= aktuell färg i position P.
330 Gör att färgen (F) i position P alternerar mellan CO och CM.
335 Lägger ut färg i position P och ljud (subrutin 500).
340-345 Testar vilken färg den tagna målbrickan har och anger resultatet (R) som resultat av testet.
350 Släcker position P: flyttar den tagna brickan till slumpmässig plats (subrutin 1100) på skärmen.
360 Summerar resultatet: Testar om svart spelare är inblandad.
405 Tänder pjäs i position P: producerar 'flopp'-ljud (miss).
410 Tidsfördröjning: stänger av ljud: släcker pjäs i position P.
415 Räknar ner resultatet med 10 för missen.
420-445 Räknar fram och skriver ut resultaten för spelarna.
450 Sätter karaktären för ljudet av 'steg' för spelare 2.
455 Som 450 fast för spelare 1.
460-470 Producerar 'steg'-ljud för spelarna.
500-520 Träff-ljud.
610 Stänger av ljud för voice 1.
1000-1100 Ritar ut målbrickorna på spelplanen.
2000-2175 Flyttar tecken från ROM till RAM och definierar nya tecken (Å, Ä, Ö).
2400-2500 Instruktionstext.

```
1 REM*******************
2 REM****BLINDBINGO****
3 REM****AV          ****
4 REM****GUNNAR      ****
5 REM*****BERGLUND   ****
6 REM*******************
7 CS=54272:FORL=CSTOCS+20:POKEL,0:NEXT:POKECS+24,15
10 PRINT"■":FORL=0TO10:PRINT "-BLINDBINGO- ETT SPEL AV GUNNAR BERGLUND"
12 GOSUB500
13 NEXTL
15 PRINT"■":PRINT"■■■■ FLYTTAR TECKEN "
20 PRINT"■■ STRAX KLAR!      ■":GOSUB2000
22 PRINT"■■■TESTA DIN SPELPJ:S MED"
23 PRINT"■■■AVTRYCKKARKNAPPEN - NU!!"
25 PRINT"■■■■STARTA SPELET - TRYCK P@ 'S'■"
27 P1=1024:P2=2023:DX=0:DY=0:X1=0:Y1=0:X2=39:Y2=24:SC=1024:POKE53280,0
28 R1=0:R2=0
30 IF(PEEK(J1)AND16)=0OR(PEEK(J2)AND16)=0THEN60
40 GETA$:IFA$<>"S"THEN30
50 GOTO90
60 IF(PEEK(J1)AND16)=0THENPOKESC+CS,6:GOSUB455
70 IF(PEEK(J2)AND16)=0THENPOKE2023+CS,6:J=J2:GOSUB450
75 POKESC,160:POKESC+CS,1:POKE2023,160:POKE2023+CS,0:GOTO40
90 GOSUB1000
98 J=J1:X=X1:Y=Y1:GOTO105
100 J=J2:X=X2:Y=Y2
105 DX=0:DY=0
110 IF(PEEK(J)AND15)=7THENDX=1
112 IF(PEEK(J)AND15)=11THENDX=-1
114 IF(PEEK(J)AND15)=13THENDY=1
116 IF(PEEK(J)AND15)=14THENDY=-1
117 X=X+DX:Y=Y+DY
118 IFX<0THENX=0
119 IFY<0THENY=0
120 IFX>39THENX=39
121 IFY=>24THENY=24
124 P=SC+40*Y+X
125 IFDX<>00RDY<>0THENGOSUB450
126 IF(PEEK(J)AND16)=0THENGOSUB300
128 IFJ=J1THEN138
129 IFR2>99THEN150
130 P2=P:X2=X:Y2=Y
135 GOTO98
138 IFR1>99THEN150
140 P1=P:X1=X:Y1=Y
145 GOTO100
150 B=5:REM BINGO!
155 PRINT"■■":IFR1>R2THENPRINT"■■"
160 PRINT"■■■■■"
162 PRINT"■■■■■■■■■■  **  **  **  ****  ****  "
163 PRINT"■■■■■■ **  **  **  **  *  **  **  "
164 PRINT"■■■■■■ **  **  ***  **  **  **  **  "
165 PRINT"■■■■■■■■■  **  **  * **  **  ***  **  **  "
166 PRINT"■■■■■■ **  **  **  ***  **  **  **  "
167 PRINT"■■■■■■ **  **  **  **  **  **  **  "
168 PRINT"■■■■■■■■■  **  **  **  *****  ****  "
170 PRINT"■■■":IFR1>R2THENGOSUB455:GOTO172
171 IFR2>R1THENGOSUB450
172 IFB>1THENB=B-1:FORT=0TO200:NEXT:PRINT"■":FORT=0TO100:NEXT:GOTO155
220 B=0:FORT=0TO1500:NEXT:PRINT"■":GOTO22
300 REM FB ON
310 CO=1:RE=R1
315 IFJ=J2THENCO=0:RE=R2
320 IFPEEK(P)<>160THENGOTO400
325 CM=(PEEK(P+CS)AND15):FORL=1TO15:F=CO
330 IFL/2=INT(L/2)THENF=CM
335 POKEP+CS,F:GOSUB500:NEXTL
340 R=5:IFCM=5THENR=10
345 IFCM=8THENR=15
350 POKEP,32:POKEP+CS,6:GOSUB1100:POKEM,160:POKEM+CS,CM
360 RE=RE+R:IFCO=1THEN440
370 GOTO430
```

forts. nästa sida

```
400 REM MISS
405 POKEP+CS,CO:POKEP,160:POKECS+1,125:POKECS+5,12:POKECS+4,129
410 FORL=1TO1000:NEXT:GOSUB610:POKEP,32:POKEP+CS,6
415 RE=RE-10
420 REM SCORE
425 IFCO=1THEN440
430 R2=RE
434 PRINT"                    "TAB(34)"     "R2"
435 RETURN
440 R1=RE:PRINT"     "R1"
445 RETURN
450 IFJ=J2THENPOKECS+1,12:GOTO460:REM LJUD 2
455 POKECS+1,100:REM LJUD 1
460 POKECS,125:POKECS+5,9:POKECS+2,255:POKECS+4,65
470 FORL=0TO35:NEXT:GOSUB610
490 RETURN
500 REM LJUDRUTIN
510 POKECS,125:POKECS+1,(RND(1)*255):POKECS+5,8:POKECS+4,17
515 FORT=0TO25:NEXTT:GOSUB610
520 RETURN
600 REM LJUD OFF VOICE 1
610 FORLJ=CSTOCS+6:POKELJ,0:NEXTLJ:RETURN
1000 PRINT"     ":REM RITA SPELPLAN
1010 FORL=1TO20:GOSUB1100
1015 POKEM,160:POKEM+CS,12:NEXT
1020 FORL=1TO10:GOSUB1100
1025 POKEM,160:POKEM+CS,5:NEXT
1030 FORL=1TO5:GOSUB1100
1035 POKEM,160:POKEM+CS,8:NEXT
1040 RETURN
1100 M=INT((2019-1027)*RND(1))+1027:GOSUB500:RETURN
2000 REM FLYTTA TECKEN
2005 J1=56321:J2=56320
2110 PRINTCHR$(142):POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251
2115 FORI=0TO69:FORJ=0TO7
2120 POKE14336+I*8+J,PEEK(53248+I*8+J)
2125 NEXTJ:NEXTI:FORJ=0TO7
2130 POKE14336+160*8+J,PEEK(53248+160*8+J):NEXT
2135 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1
2140 POKE53272,(PEEK(53272)AND240)+14
2145 FORBY=0TO7:READA:POKE14336+BY,A:NEXT
2150 FORCH=58TO59:FORBY=0TO7:READA
2155 POKE14336+(8*CH)+BY,A:NEXTBY,CH
2160 RESTORE
2165 DATA60,36,126,102,126,102,102,0
2170 DATA102,0,126,102,126,102,102,60,0
2175 DATA102,0,60,102,102,102,60,0
2400 PRINT"     BLINDBINGO!!
2405 PRINT"
2410 PRINT"  SPELAS AV TV@ SPELARE MED
2415 PRINT"  VAR SIN JOYSTICK.
2420 PRINT"  STYR DIN OSYNLIGA SPELPJ:S
2425 PRINT"  KVERTIKALT/HORISONTELLT) TILL
2430 PRINT"  N@GON AV SPELF:LTETS M@LBRICKOR
2435 PRINT"  YSSNA TILL DINA STEG!!
2440 PRINT"  NV:ND AVTRYCKAREN D@ DU TROR ATT
2445 PRINT"  DU BEFINNER DIG P@ EN M@LBRICKA
2450 PRINT"  =5P.  =10P.  =15P."
2455 PRINT"  MISSAD BRICKA=-10P.
2460 PRINT"  BINGO=F;RST TILL 100=BINGO."
2500 RETURN
```

NYA KLUBBAR

Vi har efterlyst nya VIC-klubbar eller gamla VIC-klubbar som är okända för oss och för läsekretsen. Några få svar har kommit in, men vi tror att det finns åtskilliga fler klubbar därute i buskarna! HÖR AV ER! Ni får utrymme i tidningen för att annonsera Era aktiviteter och ni får gärna skriva och göra reportage om er verksamhet. Skriv bara och berätta, det går bra med handskrivna manuskript. Vi kan både stava och skriva rent här på redaktionen! Ni får gärna också skicka med svartvita fotografier (ja i nödfall kan vi ta diabilder eller färgkopior också!) från någon intressant klubbkväll eller som visar hur ni har det.

Här nedan följer de första.

I de brev, som följde med anmälningarna berättar man synnerligen kortfattat om verksamheterna, men litet skall vi kunna återge:

MALMÖ Här berättar ordföranden Mats Nordström att man bildat en VIC-klubb dit alla VIC:are i malmötrakten är välkomna att höra av sig. Man är hittills ett tjugotal medlemmar och ger ut egen tidning med tips och artiklar. Vi skulle gärna skriva mer om tidningen, men vi väntar tills vi får se ett exemplar:....

OLOFSTRÖM Kontaktmannen Sven-Gösta Wallden berättar att man till sin datorklubb inköpt 3 st VIC-20, har stort intresse och räknar med att det vid årsskiftet skall finnas ett tiotal VIC-20-maskiner i klubben. Man har stor åldersspridning och den yngste är 10, den äldste 45. De flesta av de vuxna medlemmarna jobbar på Volvo Personvagnar AB, där man ju ser datorer i andra sammanhang.

GÄVLE-rapportören är Per Wretman, som berättar att deras VIC-klubb har ca 10 medlemmar som alla går första året i gymnasiet. De flesta har VIC-20 men två har nyss skaffat sig VIC-64. Man har hållit på med datorer ca 1 år och anser sig som ganska erfarna programmerare med en genomsnittlig "arbetstid" vid dator på 10–12 timmar varje vecka(!). Dessutom träffas man i klubben en gång per vecka för att byta erfarenheter, program och annat som kommit fram under veckan. Man har byggt upp ett stort programförråd med många bra program. Vi får väl hoppas, att några av dessa program finner sin väg till VIC-rapports spalter i framtiden...?

SPÅNGA-klubben är en avläggare till Unga Örnar. Man lär ut programmering och kringliggande ämnen i grupp. Terminsavgiften är 100 kr och man har tänkt sig att medlemmarna skulle kunna vara från 11 år och uppåt. Det språk som lärs ut är BASIC och man använder enbart VIC-20- och VIC-64-maskiner.

I EKSJÖ har man just bildat en VIC-klubb med målet att skapa kontakt med olika VIC-klubbar i Sverige. Dessutom vill man samla ihop alla data-intresserade killar och tjejer i Eksjö. Man är ca 30 medlemmar redan och har både VIC-20 och VIC-64.
Adressen är Eksjö VIC-klubb, Norra Storgatan 29, 575 00 Eksjö.

GÄVLE:
Klubbnamn: THE VICKERS
Kontaktman: Per Wretman, Sandstensvägen 16
802 26 Gävle. Tel. 026-18 64 17.

SPÅNGA:
Klubbnamn: Skäftingebackens DATOR-ÖRNAR
Kontaktman: Alf Olsson, Skäftingebacken 36, 4 tr
163 64 Spånga. Tel. 08-761 08 98

OLOFSTRÖM:
Klubbnamn: OCC - Olofströms Computer Club
Kontaktman: Sven-Gösta Wallden, Höjdaliden 13
293 00 Olofström.

MALMÖ:
Klubbnamn: MALMÖ VIC-KLUBB
Kontaktman: Mats Nordström, Box 25067,
200 47 Malmö.

Vill starta VICklubb:

STOCKHOLM: Tore Gullstrand, Norrgårdsv. 127,
184 00 Åkersberga. Tel. 0764-603 25.

Nu har vi sett några få exempel på vilka VIC-klubbar som finns och litet vad de sysslar med. VAKNA UPP NU - alla ni där ute! Skynda er och skicka informationer om era klubbar till VIC-rapports redaktion: Gustavsbergsgatan 8, 431 37 Mölndal.

Glöm inte att skriva litet om vad ni gör och hur ni har det med klubbmöten och sådant, om det är något ni önskar av VIC-rapport som ert samlande medlemsblad eller som komplement till ert eget. Tala också om vad nu vill ha i VIC-rapports spalter, vilken typ av artiklar som ni gillar bäst - och gärna vilka ni tycker sämst om också!

Red.

Låt VIC-64 räkna ut

speltid på kassettbandade grammofonskivor!

Emellanåt kanske även du vill spela in din favorit-LP på band för att ta med i bilen eller på semestern. Och då ställs du inför problemet att beräkna hur mycket som får plats på kassett-bandet.

Skivtillverkarna har också ett litet hyss för sig: de gör skivans B-sida lite längre än A-sidan. Förmodligen för att göra livet surt för oss som vill spela in bara en skiva per band, och där man vänder på bandet som om man vände på en skiva.

Fylld av desperation ritade jag därför ihop detta lilla program, som numera gör livet lättare för mig. Programlistningen hittar du här intill. Och eftersom programmet inte innehåller några kommentarer, får du dem här.

Spara programnamnet

En kommentarrad finns för resten: rad 1. Den viker jag alltid i mina program för programmets namn och datum då jag senast var inne och rotade i det. Därigenom vet jag alltid programmets exakta namn, och kan efter en bearbetning spara det på rätt sätt. Det är bara att skriva LIST 1, flytta upp markören, blanka över 1 REM med mellanslag och trycka på RETURN. Snabel-a gör att du inte får FILE EXISTS, utan katalogen kommer att visa datorn till den senast sparade versionen. Enligt Programming the PET/CBM finns det en bug i operativsystemet, varför man får använda snabel-a på egen risk. Själv har jag använt det i åtskilliga år på 3040, 4040 och 8050 utan problem, men på senare tid har jag misslyckats några gånger på 1541. Programmet sparas korrekt, men BAM (Block Availability Map) tycks inte vara felfri, varför programmet kan komma att delvis skrivas över annat på skivan. Kan någon ge en förklaring?

På rad 200 skaffar jag mig en vektor för vertikal tabulering i bildskärmen. Mycket mer praktiskt än att dribbla runt med 'markör upp' och annat. Vill jag ha 'Hej!' utskrivet på rad 16, ger jag bara instruktionen PRINT V$(16)"HEJ!"

Det är bara vid ett tillfälle du måste använda 'markör upp', och det är efter utskrift på rad 24 i skärmen. Om du då inte avslutar med 'markör upp', kommer skärmen at scrolla, så att texten i stället kommer på rad 22 eller 23 och text på rad ett försvinner upp i det blå.

SP$ är en annan praktisk variabel för radering i skärmen eller för formattering i skärm eller på skrivare. På rad 210 låter jag den bestå av 40 mellanslag.

Rad 2 sätter färger till CBM 64: POKE 53280 för ramen, 532281 för skärmen och 646 för texten. I handboken hitar du värdet för de olika färgerna (0=svart, 1=vitt och så vidare upp till 15). Ger du ett värde högre än 15, blir det svart. 9 står för brunt och 7 för gult, och det är den färgkombination som passar mig bäst.

På rad 300 skaffar du dig en rubrik, RU$, som tömmer skärmen och skriver rubriktexten i omvänd video på översta raden. Att rubriken läggs i en variabel går av gammal vana: man kan aldrig vara säker på att man inte behöver använda rubriken flera gånger. Du öppnar också en fil till tangentbordet.

Input utan frågetecken

Om du kommer till en INPUT på en PET och trycker på RETURN utan att ha matat in något, ramlar du ur programmet. Med en CBM 64 ramlar du inte ur programmet, men går vidare utan något inmatat värde, vilket kan få sina konsekvenser.

Har du i stället öppnat en fil till tangentbordet, ramlar du inte ur programmet på en PET. Genom en enkel kontroll, kan du se om operatören givit RETURN utan annan inmatning. En sådan kontroll har du på rad 150. Finessen är då, att markören kommer att placera sig själv på samma rad som den ursprungligen stod. Dessutom slipper du ifrån, att systemet sätter ut ett frågetecken. Skall det vara ett frågetecken i skärmen, vill jag sätta ut det själv, och på den plats jag själv valt.

En annan sak om INPUT: låt alltid input ske i en strängvariabel, så slipper du REDO FROM START. Gör sedan om strängen till ett numeriskt värde som på rad 160.

Raderna 110-180 är en subrutin som används för både A-sidan och för B-sidan. Om uppgifter skall skrivas till vänster eller till höger på skärmen styrs av variabeln KOL, som för A-sidan är 2 och för B-sidan 22. På rad 110 börjar en varvräknare, L, som står för 'Låtnummer'. För varje ny låt räknas den upp ett steg. Den används också för att styra utskriften i skärmen. På rad 170 använder du den för att avgöra vilken konstant i vektorn V$(0..24) som skall användas. På samma rad kan du se hur du kan använda SP$ för formattering.

Du vet ju, att man kan sätta ihop strängar med ett plus-tecken. Om A$="CBM" och B$=" 64", så blir A$+B$="CBM 64". På rad 170 sätter vi ihop SP$ (40 mellanslag) med strängfunktionen av ett tal. Därefter skriver vi ut sex tecken räknat från höger i den sammansatta strängen, alltså de sex sista tecknen. Skriver vi ut talen på detta sätt, kommer vi att få en prydlig spalt med entalen under varandra. 'Vänsterutfyllnad med blanktecken' kallas det av standardiseringskommissionen, men englsmännen kallar det för 'padding'.

Minuter och sekunder

På rad 160 ser du en omvandling av minuter och sekunder till enbart sekunder. Vi använder heltalsvaria-

bler (med %-tecken), eftersom inget värde kommer att överstiga 32 767, och vi vill slippa alla decimaler. Som du kanske har märkt, är inte operativsystemet helt pålitligt när det gäller att lagra heltal i en flyttalsvariabel.

Låt säga att ett inslaget värde är 3.45. Låter vi M% anta detta värde, kommer datorn att skala bort decimalerna, så att M%=3. Därefter drar vi M% från 3.45 och multiplicerar resultatet med 100. Resultatet, 45, lagras i S%, varefter S% ökas med minuterna uttryckta i sekunder: S% = M% * 60 + S%. S1% används för den ackumulerade tiden för A-sidan, och därför ökar vi S1% med S%.

Slutligen skrivs S% ut, högerställt på samma plats som tiden matades in, följt av tre mellanslag som raderar onödiga decimaler. Slingan går tillbaka till rad 110 om vi inte skrivit in 'SLUT' i stället för tid.

När vi angivit 'SLUT', så sker ett uthopp från rad 140 i subrutinen. Sista raden raderas och vi får utskrivet på rad 20 i skärmen hur lång A-sidan är. Varvräknaren L nollställs också. Därefter beräknas B-sidan på samma sätt, men med alla utskrifter 20 tecken längre till höger i skärmen.

På rad 420 beräknas den totala tiden, vilken skrivs ut på rad 430. Där tar vi också hand om den egenheten, att då datorn skriver ut ett numeriskt värde, lämnar den ett blanksteg efter värdet. Skriver du då på en rad som inte är tom, raderas inte tecknet under blanksteget. Receptet för att lösa detta är, att först backa ett steg och därefter göra ett mellanslag.

Skenbart beslut

Som du vet, är datorn det dummaste som finns. Men om du tänker dig in i alla alternativa resultat som rimligtvis kan uppkomma, kan du få datorn att verka intelligent.

På rad 500 beräknas först skillnaden i tid mellan de två sidorna och S1$ kommer att innehålla texten 'A-sidan' och S2$ 'B-sidan'. Om B-sidan är den längre, får innehållet i S1$ byta plats med S2$ på rad 510, en så kallad 'swap'. När vi kommit till rad 520 vet vi med säkerhet, att den längre sidans namn finns i S1$. Om A-sidan är den längre är D% positiv, och då behöver vi inte manipulera med inspelningen. Rad 24 på skärmen raderas då. Men om B-sidan är den längre, halverar vi differensen mellan sidorna och skriver ut ett glatt budskap. Slutligen ställer sig datorn på rad 550 och väntar på att du trycker på en tangent.

J. Stiernborg

```
1 REM      SAVE"@0:SKIVTID",8:                    REM      831016  JS
2 POKE53280,9:POKE53281,9:POKE646,7:GOTO200
100 REM                                          >>> INPUT
110 L=L+1
120 PRINTV$(2+L)SPC(KOL)"LÄT"+RIGHT$(SP$+STR$(L),3)"    ";
130 INPUT#9,A$
140 IFLEFT$(A$,4)="SLUT"THENPRINTV$(2+L)SPC(KOL)LEFT$(SP$,20):RETURN
150 IFVAL(A$)=0THEN120
160 M%=INT(VAL(A$)):S%=(VAL(A$)-M%)*100:S%=M%*60+S%:S1%=S1%+S%
170 PRINTV$(2+L)SPC(KOL+8)RIGHT$(SP$+STR$(S%),6)"    "
180 GOTO110
190 :
200 DIMV$(24):V$(0)="■":FORI=1TO24:V$(I)=V$(I-1)+"■":NEXT
210 SP$="":FORI=1TO40:SP$=SP$+" ":NEXT
220 :
300 RU$="■R      ***   RÄKNA TID PÅ SKIVA   ***     ":PRINTRU$:OPEN9,0
310 PRINTV$(24)" ANGE TID SOM 3.20   SLUT=SLUTA■"
320 PRINTV$(2)"A-SIDAN"
330 S1%=0:L=0:KOL=2:GOSUB100
340 SA%=S1%:S1%=0:L=0
350 PRINTV$(22)"A-SIDAN ÄR"SA%" SEKUNDER"
360 :
400 PRINTV$(2)SPC(20)"B-SIDAN"
410 KOL=22:GOSUB100
420 S2%=SA%+S1%:M%=S2%/60:S2%=S2%-M%*60
430 PRINTV$(20)"SAMMANLAGT"M%"■ MINUTER OCH"S2%"■ SEKUNDER"SP$
440 :
500 D%=SA%-S1%:S1$="A-SIDAN":S2$="B-SIDAN"
510 IFD%<0THENA$=S1$:S1$=S2$:S2$=A$
520 PRINTV$(22)S1$" ÄR"STR$(ABS(D%))" SEK. LÄNGRE ÄN "S2$
530 IFD%>=0THENPRINTV$(24)LEFT$(SP$,35)"■":GOTO550
540 PRINTV$(24)"FLYTTA EN BIT SOM ÄR MINST"INT(ABS(D%)/2+.5)"■ SEK. ■"
550 GETA$:IFA$=""THEN550
```

Assemblerskolan 3

Hej "VICare"! Nu skall vi ta tjuren vid hornen igen. Låt oss tillsammans tränga djupare in i assemblerdjungeln. De flesta är väl ganska sugna på att programmera nu och se lite resultat av allt slit med talbaser och annat. Men först en kort men användbar genväg för omvandling från en given bas till decimalt samt en lika snabb metod för omvandlingar åt andra hållet.

A) Algoritm för omvandling från bas X till decimalt.

Antag att talet X består av siffrorna X3, X2, X9, X0 (t.ex. %1000). Skriv detta tal på en rad med ganska stort mellanrum mellan siffrorna:

```
X3      X2      X1      X0
```

Flytta nu ned X3 till raden under, dvs vi får:

```
X3      X2      X1      X0
X3
```

Multiplicera det nedflyttade talet med basen (X). Addera nästa siffra (X2). Detta ger:

```
X3      X2              X1      X0
X3    X3*X+X2
```

Gör nu samma sak med nästa tal. Multiplicera talet som står under X2 med basen X och addera nästa siffra X1:

```
X3      X2              X1                      X0
X3    X3*X+X2      (X3*X+X2)*X+X1
```

Gör sedan samma sak igen och du får:

```
X3      X2              X1                      X0
X3    X3*X+X2      (X3*X+X2)*X+X1    ((X3*X*X2)*X+X1)*X+X0
```

Det tal som nu står under X0 är det decimala resultatet av talomvandlingen. Låt oss belysa tekniken ytterligare med två exempel.

Ex.1. omvandla %10100000 till decimaltal

	föregående värde	* basen	+ nästa	= värde
1	0	* 2	+ 1	= 1
0	1	* 2	+ 0	= 2
1	2	* 2	+ 1	= 5
0	5	* 2	+ 0	= 10
0	10	* 2	+ 0	= 20
0	20	* 2	+ 0	= 40
0	40	* 2	+ 0	= 80
0	80	* 2	+ 0	= 160

dvs. % 10100000 = 160 decimalt

Ex. 2. omvandla $C13F decimaltal. Basen vi nu multiplicerar med är 16.

```
C (12)                =   12
1          12 * 16 +  1 =  193
3         193 * 16 +  3 = 3091
F (15)   3091 * 16 + 15 = 49471
```

dvs. $C13F är decimalt lika med 49471.

B) Algoritm för omvandling från decimaltal till tal i given bas.

Här dividerar vi istället för att multiplicera. Resten vid divisionen är den aktuella siffran. Låt oss förklara metoden med två exempel.

Ex. 3. Omvandla 200 till ett binärt tal.

```
2 / 200 = 100   rest = 0
2 / 100 =  50   rest = 0
2 /  50 =  25   rest = 0
2 /  25 =  12   rest = 1
2 /  12 =   6   rest = 0
2 /   6 =   3   rest = 0
2 /   3 =   1   rest = 1
2 /   1 =   0   rest = 1
```

Vi sätter sedan ihop de rester vi erhållit och omvandlingen är klar:

```
% 1 1 0 0 1 0 0 0
  ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑
rest 8, 7, 6, 5, 4, 3, 2, 1
```

Ex. 4. Omvandla 200 till ett hexadecimalt tal.

```
16 / 200 = 12   rest =  8
16 /  12 =  0   rest = 12 (=$C)
```

dvs. $ C 8 är den hexadecimala motsvarigheten till 200.

Ett annat tips vi måste ge är hur man lätt gör omvandlingar mellan den hexadecimala och den binära basen. Antag att vi har

% 1 0 1 1 0 1 0 1

Dela upp detta tal i grupper om fyra, dvs i 2 nibbles (nibble = 1/2 byte):

% 1 0 1 1 0 1 0 1

Omvandla var och en av dessa grupper till hex-tal.

```
% 1 0 1 1  =  $ B
% 0 1 0 1  =  $5
```

Sätt ihop dem och hextalet är konstruerat. I detta fall $ B 5.

Omvänt är förfarandet lika lätt. Omvandlar vi t.ex. $ C) till binärt gör vi så här:

```
$ C = 12 = % 1 1 0 0
$ 9 =  9 = % 1 0 0 1
```

Sätt ihop dessa och vi har det binära talet % 1 1 0 0 1 0 0 1.

Från och med nu är det slut med talkonverteringar för vår del (pust, pust). Vi förutsätter alltså att ni i fortsättningen klarar av den detaljen själva.

De programexempel som kommer att ges här kommer alla att vaa avsedda för VIC-20.

Men även du som har en VIC-64 kan dra nytta av det som skrivs här om än inte direkt använda de rutiner som listas. Tekniken är ju trots allt densamma för båda maskinerna.

En annan sak vi måste påpeka innan vi hastar oss in i programmerandet är att du som har tänkt följa den här serien behöver en maskinspråksmonitor av något slag. HESMON (som rekommenderas varmt) är den vi själva använder, men även handics VICMON går bra.

Det skulle vara fullt möjligt att lista en monitor skriven i BASIC. Men nackdelarna med en sådan monitor är så stora att vi avstår från detta.

Om någon av er som läser detta har tillgång till "public-domain"-monitorn SUPERMON eller liknande, skicka gärna en listning till oss så skall vi publicera denna för alla er fattiga hackers där ute, som får svälta vid köp av en kommersiell monitor.

All programmering i serien kommer att bygga på användbara rutiner. Genom att studera hur de konstruerats skall även vi lära oss att göra liknande rutiner.

Den första rutinen vi skall titta på är avsedd för att programmera funktionstangenterna på VIC-20. Rutinen är gjord av Jim Wilcox (publicerad 1982 i novembernumret av COMPUTE!, sidan 196). En loader för detta program hittar du i slutet av denna artikel.

(En assemblerlistning av programmet går vi igenom i nästa nummer. Ditt uppdrag till dess blir att kika litet på det på egen hand.)

När du knappat in programmet skall du först spara det (på band/disk) innan du kör det. Ett fel i en av DATA-satserna kan nämligen få systemet att "dyka". Fungerar det inte första gången, gå igenom DATAsatserna och kolla att de är korrekt inslagna.

När du kör programmet skrivs F1=? ut på skärmen. Skriv in den funktion du vill att F1 skall ha, t.ex.
F1 = List (vänsterpil) (RETURN)
där (vänsterpil) betyder att en automatisk return skall ske när du trycker F1. När du slår (RETURN) kommer nästa fråga:
F2=?

Fortsätt på samma sätt tills alla funktionstangenter definierats. När detta skett skrivs
READY
på skärmen och Dina funktionstangenter kan nu göra en hel del nytta.

Raskt över till litet assembler.

Låt oss titta på vad LDA, som förekommer flitigt i assemblerprogram, gör
LDA Laddar ackumulatorn (A) med data från minnet.

Titta i manualen till din monitor och undersök hur A- (assembler)-kommandot fungerar. På HESMON/VICMON skriver du:
A adress mnemonic operand (RETURN)

Sedan skrivs efterföljande adresser ut automatiskt.

Skriv nu in

adress	mnemonic	operand
1200	LDA	$FF
1202	BRK	

(skrivs ut automatiskt)

Efter den (RETURN) du slår efter BRK skrivs 1203 ut. Slå då en (RETURN) till så att markören (cursorn) hamnar mot vänsterkanten.

Skriv nu
G 1200 (RETURN)
och det lilla program vi skrev kommer att köras. När programmet är klart skrivs registerinnehållen ut på skärmen. På ett ställe här står AAC och därunder ser du $FF. Detta betyder att vårt program fungerade och vi har laddat ackumulatorn med $FF. Byt ut $FF mot andra värden och kör programmet. Vad händer?

Byt nu ut LDA$#$FF eller det värde du ersatt $FF med, mot LDA $FF.

"Hage"-tecknet (#) tas inte med denna gång. Kör ditt program igen. Vad blev nu innehållet i A+

Kanske blev det $FF men sannolikheten är betydligt större att det blev något annat tal. Skriv upp detta tal på ett papper. Använd nu M-kommandot på din monitor och skriv
M 00FF (RETURN)
dvs be monitorn att visa vad som ligger i minnesposition $FF och framåt.

Effekten av kommandot blir att en hel radda av tal visas på skärmen (8 i HESMON, 5 i VICMON).

Jämför nu det första av dessa tal med det tal du antecknade. De ser misstänkt lika ut, eller hur?

Skriv nu över det tal som står först med 10 och slå (RETURN). Kör sedan programmet igen.

Under AC står nu $10, dvs programmet
LDA $ FF
BRK
har nu laddat ackumulatorn med $10 i stället för $FF eller vad du tidigare hade. Ändra nu LDA$FF återigen till LDA#$FF och försäkra dig om att minnesposition $FF innehållr $10, detta skrivs vanligen ($FF) = $10.

Kör programmet!

Under AC står åter $FF. Testa med att ändra innehållet i minnesposition $FF. Hur du än beter dig blir resultatet att A laddas med $FF.

Vad du just bevittnat är användandet av EN instruktion på assemblernivå men du har använt TVÅ olika adresseringsmoder.

LDA betyder att A skall laddas med data från minnset och är vad vi kallar en MNEMONIC (ungefär "minneshjälp"). Det som står efter LDA är OPERANDER, dvs $FF och $FF i vårt fall. Operanden är data som säger hur LDA skall jobba.

När "hage"-tecknet (#) är med, säger vi åt monitorn att adresseringsmode IMMEDIATE (ungefär = direkt) skall användas. Denna adressmode laddar ackumulatorn direkt med det tecken som följer, dvs
LDA# $FF laddar A med $FF
LDA# $10 laddar A med $10
osv.

När vi inte använder hage-tecknet säger vi i stället åt monitorn att ladda ackumulatorn med INNEHÅLLET i den adress som följer, dvs
LDA $ FF laddar A med INNEHÅLLET i $FF
LDA $ 10 laddar A med INNEHÅLLET i $10

Detta är som du förstår också en adressmode men kan delas upp i två. För att förstå varför tar vi ett exem-

forts. nästa sida

pel. Skriv nu in nedanstående program på monitorn:

```
1200 LDA#$ F F
1202 LDA  $ F F
1204 LDA  $ F F 0 0
1207 BRK
```

Adresserna som står till vänster om ditt program talar om var ditt program ligger. Som du ser börjar det i minnescell $1200 och slutar vid $1207

Skriv nu

M $ 1 2 0 0 (RETURN)

Nu skrivs

1200 A9 FF A5 FF AD 00 FF 00 på HESMON (på VICMON måste du ta fram en rad till).

Ändra nu alla $FF till $10 genom att skriva över dem och slå (RETURN). Skriv sedan:

D 1 2 0 0 (RETURN)

och tryck på Cursor ned tills du fått:

```
1200 LDA#$10
1202 LDA  $10
1204 LDA  $1000
1207 BRK
```

på skärmen. (Du kan även skriva: "D 1200 1207" så får du samma resultat direkt).

Som du ser har alla $FF bytts ut mot $10. Vad du gjorde när du ändrade alla $FF till $10 var att ändra i det maskinspråksprogram som din monitor skapade av din assemblerkod.

Varje mnemonic presenteras i maskinkodsformen av ett åtta bitars ord (en byte).

LDA=$A9, LDA=$A5, LDA=$AD och BRK=$00.

Nä, stopp nu! Här är något fel. LDA representeras ju i maskinkodsformen av $A9, $A5 och $AD. Här är något på tok!

Lugn, bara lugn.

Att LDA får olika kod i de olika fallen beror just på att vi använt olika ADRESSMODER.

$A99, $10

representerar just LDA $10, dvs vi laddar A med det värde som följer operationskoden ($A9). Detta är som sagt adressmode IMMEDIATE.

$A5 $10

representerar LDA $10, dvs A laddas med innehållet i den adress som följer operationskoden ($A5).

$AD $00 $FF

representerar LDA $FF00 (OBS! "låg/hög-formatet" på adressen.) A laddas även här med innehållet i den adress som följer operationskoden ($AD).

Du skarpsynte ser direkt att vi här har två olika opkoder IGEN för samma sak. Besvärligt det här med maskinkods-) (läs assembler-) programmering!

Lugn, bara lugn. Faktum är att

LDA $FF och

LDA $FF00

också tillhör två olika adressmoder.

Den första av dessa säger att data skall tas från en byte som ligger på den första sidan, eller sidan noll (page zero eller xero page på utrikiska) ($00-$FF).

För att representera en adress på denna adressarea behövs bara en byte, dvs högbyten som ju här alltid är lika med noll behövs inte i maskinspråksprogrammet.

Adressmoden kallas ZERO-PAGE eftersom den adress, som är operand, dvs följer LDA, skall ligga på denna area.

För att peka på en given adress i datorns minne krävs två bytes:

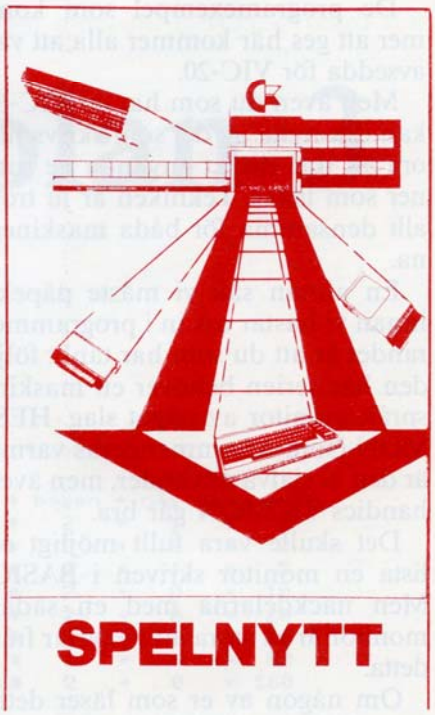
LDA $FF00

är ett exempel där vi pekar ut minnescellen $FF00. Denna adressmode kallas den ABSOLUTA. Som du säkert insett övergår denna till ZERO-PAGE när adressen är mindre än $100.

Detta var en liten titt på programmering. Du vet nu hur man kan ladda data i ackumulatorn på några olika sätt. Öva detta till nästa nummer så att du behärskar den ABSOLUTA; ZERO-PAGE och IMMEDIATE adressmoden av LDA. Ha det så kul! Nästa gång skall vi lära oss hur man lägger ut data i en minnesposition, titta på assemblerlistningen till ovan presenterade program och en hel del andra spännande saker.

Assemblerskolan del 3

Copyright 1983 Peter Tegelberg och Åke Hedman.



Cosmic Jail Break

Detta spel går ut på att med en laserkanon skjuta ned 26 st monster som skall befria 3 st fångar från ett fängelse. 13 st monster står stilla på vänster och höger sida i en vertikal rad och strax efter det att spelet har börjat förflyttas slumpvis utvalda monster mot fängelset som ligger i spelets mitt. Monstren fäller under denna förflyttning bomber.

Fängelset består av 9x9 block och mitt inne i fängelset sitter 3 st fångar. När monstren kommer fram till fängelsets vägg demonterar de fängelset blockvis genom att ta med sig ett block på återvägen till sin ursprungliga plats i det vertikala ledet.

Man kan endast skjuta på monstren när de rör sig från och till fängelset, således aldrig när de står stilla i vänster resp. höger vertikala led.

För varje fånge som monstren tar med sig tillbaka förlorar man en av dom tre ursprungliga laserkanonerna som man har från spelets början.

Monstren vinner spelet när de tagit ned tillräckligt med block för att ta med alla fångarna ut, eller om deras bomber träffat laserkanonerna 3 ggr.

Ovanför laserkanonerna finns 3 st betongskydd som både dina skott

och monstrens bomber delvis förstör vid varje träff. Man får poäng för träff på moderskepp och monster. Har man lyckats skjuta ned alla monstren i första spelomgången startar automatiskt en ny spelomgång med nya bunkers och 26 st nya monster. Man får även ett nytt fängelse.

Om däremot första spelomgången tar lång tid lämnas fängelset över till den nya spelomgången i det skick det befann sig när första spelomgången slutade dvs. med flera block borttagna.

När man genom träning lyckats nå även tredje spelomgången dyker ett rymdskepp upp mellan laserkanonerna och monstren och följer laserkanonens rörelser, det blir då svårare att träffa monstren.

Vid träff i ytterkant på rymdskettet försvinner det och man får poäng. Men rymdskeppet återkommer upprepade gånger i tredje och fjärde spelomgången.

Min uppfattning om spelet är:

Spelet är både roligt och engagerande då man måste vara helskärpt för att hinna undan alla bomber som fälls och samtidigt försöka träffa monstren.

Dessutom varierar spelets svårighetsgrad genom att under spelets gång utan förvarning ett monster arbetar dubbelt så fort för att försöka få ut fångarna och är då mycket svårt att träffa.

Speltips:

Vid 2000, 3500 och 5000 poäng får man en extra laserkanon.

Skjut bort bunkern först det blir då lättare att följa och skjuta på monstren med laserkanonen.

Dick Augustsson

WIZARD

Du befinner dig i en labyrint fylld med farliga monster. Din uppgift är att förinta dessa monster innan dom förintar dig... Du har tre försök på dig att förinta alla monsterna och erövra labyrinterna. När du 20000 poäng har du ett försök till. Spelet är uppbyggt i fyra olika labyrinter. När du klarat av sista labyrinten ökas svårighetsgraden och du börjar om på första labyrinten igen.

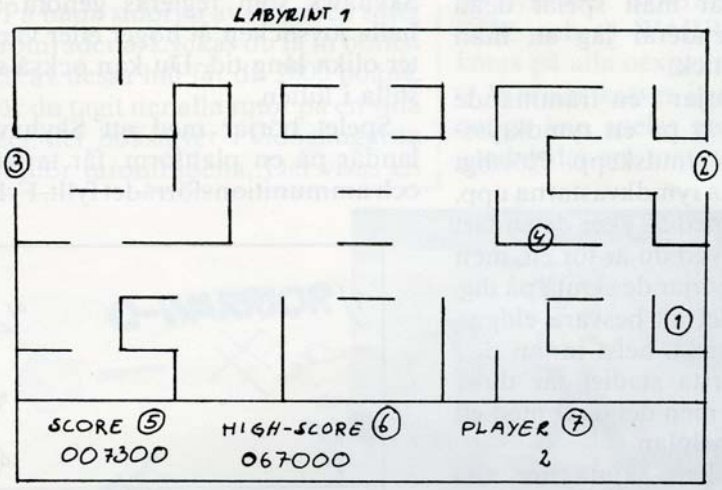
Du kan styra din "gubbe" med joystick eller med tangentbordet. Om du styr din "gubbe" med joystick avfyrar du din laserpistol med avfyrningsknappen på joysticken. Om du använder tangentbordet att styra med använder du L för vänster. ; för höger, P för upp, . för ned och du avfyrar laserpistolen med A .

En spelomgång går till så här: Du börjar med att skjuta ned de blå, gula och de gröna monsterna. Ju längre tid du tar på dig att förinta monsterna ju snabbare rör sig monsterna och ju saktare kan du röra dig. Efter du har förintat alla monster kommer ett ensamt vitt monster med ökad svårighetsgrad. Efter det monstret kan eventuellt ett rött monster komma, han är ännu svårare än det vita monstret. Antalet vita monster som kommer ökas för varje labyrint du har klarat av, detsamma gäller för de röda monsterna när du klarat av några labyrinter.

Varje gång du lyckas förinta ett monster får du poäng. Hur mycket poäng du får beror på vilken fiende du förintar:

Blått		100 poäng
Gult		200 poäng
Grönt		300 poäng
Vitt		400 poäng
Rött		varierande poäng

Monsterna är inte så välgjorda grafiskt sett, men detta kompenseras av den höga hastigheten på spelet. Bildskärmen är bra utnyttjad, figurerna är stora och tydliga. Det finns inte heller någon onödig text på bildskärmen.



1. Här startar du, även på de andra kartorna.
2. 3. Här kan man gå emellan.
4. Labyrintvägg.
5. Ditt resultat.
6. Det högsta resultatet.
7. Hur många reservspelare du har.

En bra taktik är att man håller sig på ett ställe hela tiden. Ju närmare väggen är som du skjuter mot ju snabbare kan du avfyra laserpistolen.

Wizard innehåller allt ett dataspel skall innehålla: action, spänning, skicklighet och det krävs taktik. Spelet är svårt vilket leder till att man inte tröttnar på det så fort. LYCKA TILL!

TEAMWORK

Speltestare söktes! Stopp, stopp, stopp!

Som svar på vårt anrop i nummer 3 av VIC-rapport, där vi efterlyste speltestare, har vi fått så otroligt många svar att vi inte ens hunnit skriva till dem alla ännu. Men alla ni som anmält er, var lugna, ni får ett brev så småningom. Det är bara så att redaktörens tid inte räcker till att både göra tidning och svara på 250 brev samtidigt! Vi behöver i alla fall inga fler speltestare på ett tag!

Nya spel - och testrapporter

En läsare i Falun, Kent Asp, skickade oss en ny test på ett spel vi hade infört i nummer 3 och den tyckte vi var så bra att den återges här:

Arcadia

För att vara ett spel till en oexpanderad VIC så är detta spel helt enkelt fantastiskt. När man spelar detta spel rekommenderar jag att man använder joystick.

Det hela börjar i en främmande galax där du är på en rymdkryssning med ditt rymdskepp. Plötsligt dyker de otäcka rymdkvastarna upp. Till att börja med så åker de endast runt för att se vad du är för en, men ganska snart börjar de skjuta på dig.

Nu gäller det att besvara eldgivningen illa kvickt, helst innan den börjar! På första stadiet får du 1 poäng/fiende, men det ökar med ett för varje ny spelplan.

När du lyckats skjuta ner alla rymdkvastarna så kommer nämligen: DE HEMSKA RYMDFJÄRILARNA.

Till skillnad från rymdkvastarna anfaller dessa genast och fäller betydligt fler bomber. Här gäller det alltså att vara på sin vakt varje sekund. Skulle du klara av även detta stadium så har du fortfarande sex svårighetsgrader kvar, nämligen:

* De fruktade rymdvandrarna
* Phonixerna
* Virvlarna
* Långe orm
* De pulserande solarna
* De hemska bumburangerna

Om du klarar av alla dessa omgångar så fortsätter spelet med start från början.

Spelet får 4 poäng av 5 möjliga.

Från QUICKSILVA i England kom ett nytt spel till VIC-20:

Skyhawk

Spelet är för VIC-20. Det kräver joystick och antingen 3k, 8k eller 16k RAM expansion. Spelet ligger på band och tar cirka 3 minuter att ladda in.

Din uppgift i spelet är att försvara en stad som bombas av fientliga flygplan. Med hjälp av ditt flygplan, Skyhawk, kan du skjuta ner fiendeflygplanen. Det finns två hastigheter på Skyhawk som regleras genom att hålla joysticken åt höger eller vänster olika lång tid. Du kan också stå stilla i luften.

Spelet börjar med att Skyhawk landar på en plattform, får tanken och ammunitionsförrådet fyllt. Fylld tank innebär 900 enheter bränsle och fyllt ammunitionsförråd innebär 100 robotar. Du är tvungen att då och då under spelets gång landa på någon av plattformarna för att fylla på förråden. Fiendeplanen bombar sönder plattformarna under spelets gång. Se därför till att tanka mellan anfallen. När bränslet kommer under 100 enheter ger datorn en ljudsignal. Om bränslet tar slut störtar du.

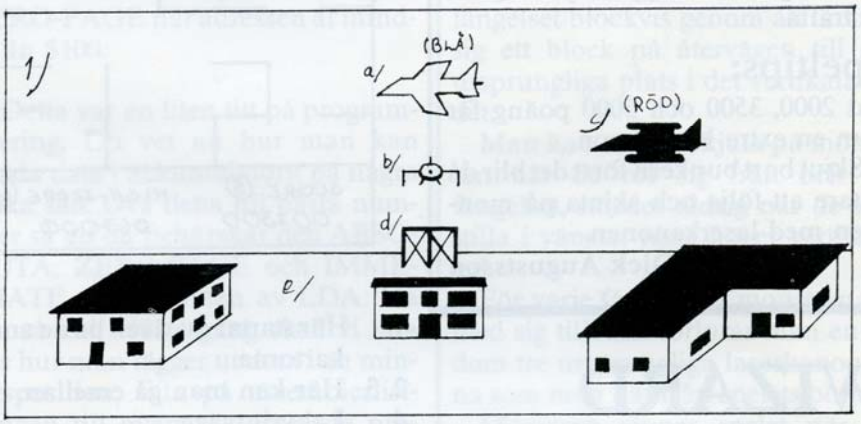
Under den första nivån anfaller fienden i formation om två plan. Under den andra med tre plan och under den tredje med fyra plan. När du fått 400 poäng börjar andra nivån och efter 1500 poäng börjar tredje nivån.

Poängsättning sker enligt följande:

första planet i formation	25 poäng
andra planet i samma formation	50 poäng
tredje planet i samma formation	100 poäng
fjärde planet i samma formation	200 poäng

När du fått 3000 poäng får du ett extraplan.

En bra taktik är att åka mot flygplanen på lägsta höjd, vända när de passerar och anfalla bakifrån. Se upp för lågtflygande plan och titta ofta på radarn (se bild).



1. Stridsrummet
   a) Skyhawk flygande
   b) Skyhawk stillastående
   c) Fiendeplan
   d) Plattform
   e) Landskap
2. Din poäng
3. High Score
4. Radar
   a) Var du är på radarn
   b) Plattform
   c) Fiendeplan
5. Bränsletank (full 900 enheter)
6. Ammunitionsförråd (fullt 100 robotar)

En av de få nackdelarna med spelet är att man kan åka igenom explosionen av ett träffat fiendeflygplan. Nosen på Skyhawk blinkar när man skjuter.

Spelet har en skrollad skärm. den är bredare än vanligt. En enkel uppbyggnad med en välgjord högupplösningsgrafik i många färger. Stridsrummets bakgrund är grå!?! Ljudet är bra och passar till spelet. Landskapet är gjort i perspektiv med fina detaljer. Flygplanen är grövre men rör sig mjukt.

Spelet har en enkel idé och är omväxlande. Mycket nöje och lycka till!!

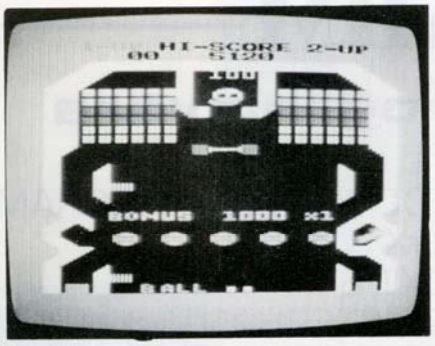
Jesper och Joakim

Handic Software har ett nytt spel, som vi också fått en testrapport på:

Pinball Spectacular

Spelet är för en VIC-20 och ligger på cartridge. Det kräver paddles. I början av spelet är bilden flyttad åt vänster så att en del av bilden inte syns. Detta kan du inte justera förrän instruktionerna visats klart.

Starta spelet genom att trycka på F1. Nu kan du välja mellan 1 eller 2 spelare genom att trycka på F1 respektive F3. Nu börjar spelet. Bollen studsar omkring i det högra turområdet (se bild). Du har tre bollar. För



att få bollen i spel trycker du på avfyrningsknappen på paddeln. För att hindra bollen att "rinna" har du två skivor som du styr med paddeln. En mitt på skärmen och en vid skärmens botten. På mitten av skärmens övre del finns ett spökhus. I spökhuset finns ett till tre spöken på samma gång. Mängden spöken regleras genom att låta bollen studsa på den nedre skivan. Poängen för dessa, som står ovanför spökhuset, varierar mellan 100-5000 poäng. På båda sidorna av spökhuset finns det gula, röda och violetta väggar. Gula rutor får du 30 poäng för, röda 40 poäng och violetta 50 poäng. Om du träffar en röd ruta ökas hastigheten på bollen lite och om du träffar en violett ruta ökas den ännu mer. Bonusen som från början är 1000*1 poäng ökas med 1000 poäng för varje färgfält du lyckats ta ner. Om du lyckas ta ner alla rutor på en sida kommer det fram en rymdvarelse. Rymdvarelsen går tvärs över området där rutorna förut var. När den nått andra kanten försvinner den. Om du träffar rymdvarelsen får du 500*1 poäng till 500*9 poäng. När rymdvarelsen försvunnit kommer det fram nya rutor. Under spökhuset sitter spinnern, du får 20 poäng per varv den snurrar. Ovanför den nedre skivan sitter fem ansikten. Vi början av spelet är de blåa och ledsna. När bollen passerar över dem blir de gula och glada. Om du åker över ett gult ansikte blir det blått igen. Om du lyckas med att få alla ansikten gula blir dom röda och du får bonusen*2 istället. Om du ännu en gång lyckas få ansiktena gula blir de blåa igen och du får din bonus*3.

På båda sidorna av skärmen sitter turområdena. Lyckas du få in bollen i ett av dessa hål får du 2000 poäng. När du tagit ner alla rutor på en sida lyser det bokstäver i sidogångarna ovanför turområdena. Det visas en bokstav i taget av ordet EXTRA. Om du lyckas tända alla bokstäverna får du en extraboll. Du får också en extraboll vid 50 000 poäng. I nedre hörnen finns rinnhålen. Första gången bollen åker ner i hålet skyddar en grön kloss så att bollen studsar upp igen. Nästa gång finns det ingen kloss som skyddar. När du tagit ner alla rutor på en sida får du en ny kloss, om den har försvunnit. Ett bra tips är att göra alla ansikten glada innan bollen går för fort.

Detaljerna på spelet är grova men välgjorda. En miss i spelet är att bollen studsar på mitten av skivan i höjdled. Figurerna rör sig på ett fint sätt.

Ett välgjort program som bygger på en bra idé!
Lycka till!!!

Joakim och Jesper

Från spel-leverantörerna har vi fått litet informationer om de program, som man just håller på att distribuera och det vill vi också låta VIC-rapports läsare ta del av:

3 nya VIC-20-spel

Från Programdistributören kommer tre nya spel: BUGBASHER, PLAGUE och Q WARRIOR. De kan köras på alla oexpanderade VIC-20. Q Warrior kräver joystick, de båda övriga kan spelas med enbart tangentbordet om man vill.





ADB-UTBILDNING

för studiecirklar, skolor, företagsutbildning och för enskilda studier.

ALLT I EN PÄRM

Läroböcker:

- REDOVISNING PÅ VIC-64
- ORD OCH TEXTBEHANDLING PÅ VIC-64
- REGISTER HANTERING PÅ VIC-64
- ADB VAD ÄR DET?

Programvara

- REDOVISNINGSPROGRAMMET V-BOX
- TEXT 64
- REGISTERHANTERING
- PROGRAMNYCKEL

förlagsgruppen 011-13 40 80

handic 031-28 97 90

BÖCKERNA SÄLJES HOS DIN RADIO-TV-DATAHANDLARE SOM SÄLJER VIC

Bygg ut Din VIC 20 – VIC 64. Stor & Liten har sortimentet.



VIC 1005. Expansionsenhet för VIC 20.
Stor & Litens pris **595:-**





Kalender till VIC 64. Idealisk för adress och telefonregister, förteckning över sammanträden m.m.
Stor & Litens pris **495:-**

Spelet **Jupiter Lander** på VIC 64. Landa Din farkost på främmande planet.
Stor & Litens pris **249:-**

ÅÄÖ till VIC.
Stor & Litens pris **249:-**



VIC 64 är en kraftfull dator med fantastiska möjligheter.
Stor & Litens pris **3.995:-**

Forth 64. Innehåller strukturerad, kompakt och extremt snabb kod. Virtuellt minne m.m.
Stor & Litens pris **495:-**

Easy Calc Result på VIC 64. För avancerat budgetarbete, kalkylering m.m.
Stor & Litens pris **995:-**

VIC 1541. Floppy Disk för VIC 20 och 64. Skivminne för upp till 176.000 tecken.
Stor & Litens pris **3.495:-**



VIC 1530. Bandstation för VIC 20 och 64. Yttre minne för lagring av program och data.
Stor & Litens pris **589:-** exkl. kassett.





Grafik och ljud på VIC 20. Lär Dig komponera melodier och rita grafiska bilder.
Stor & Litens pris **97:-**

Basic på VIC 20. Allmänna grunder om basic.
Stor & Litens pris **97:-**

Programmeringshandbok till VIC 20.
Stor & Litens pris **151:-**



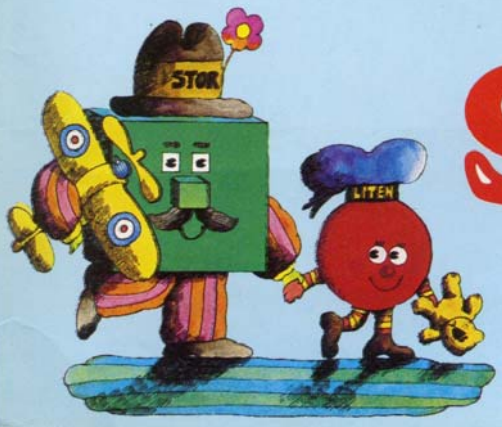
Introduktions till Basic för VIC 20. Innehåller bok och kassettband. 2 delar.
Del 1. Stor & Litens pris **249:-**
Del 2. Stor & Litens pris **269:-**



Kunskap ger försprång. Lär med VIC 20.
Stor & Litens pris
1.995:-

VIC 1111. Ökar minnet på Din VIC 20 med 16K RAM.
Stor & Litens pris **795:-**

VIC 1210. Ökar minnet med 3K RAM.
Stor & Litens pris **349:-**



Stor & Liten

FramtidsButiken

Gallerian, Täby och Arninge Centrum, Ringen Söder.

08/231390 08/758890 08/756893 08/440585



Störst på VIC program!

Vi har Skandinaviens största sortiment av program till VIC hemdatorer. Spel- och nyttoprogram på diskett och kassett för VIC och Spectrum, hemdatorer med tillbehör alltid till bästa dagspris; allt är samlat i den första fullständiga postorder-katalogen.
Beställ den 90-sidiga katalogen, posta kupongen redan idag!



Återförsäljare sökes till vårt oslagbara programsortiment. Kontakta oss!

DATAMÄKLAREN

Vasavägen 61 C, Linköping · Tel 013-10 31 74

Insändes till: DATAMÄKLAREN, BOX 3045, 580 03 LINKÖPING

Jag beställer nya katalogen

Namn _____

Adress _____

Postnr/Ort _____

Tel _____