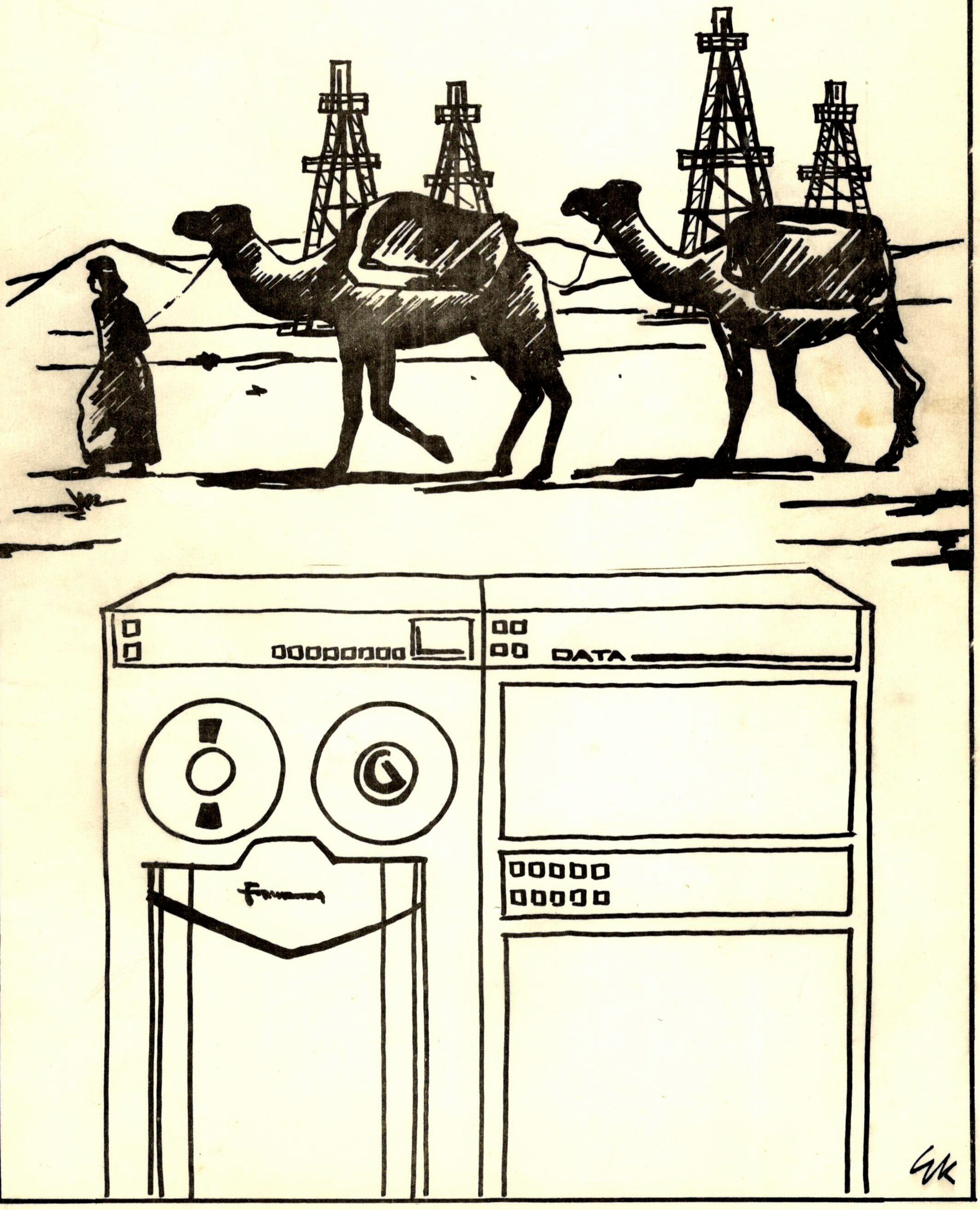
MPU-laren nr4 1980

TIDSKRIFT FÖR PRIVATDATAKLUBBEN PD68 ÅRGÅNG 3



REDAKTIONEN

MPU-larenredaktionen består av

Roger Holmstrand (RH)
Rapsodivägen 168
142 00 TRÅNGSUND 08/7713852
(FLEX2,FLEX9,Papperstape,KCS)

Carl Axel Bruno (CAB)
Timotejvägen 116
191 77 SOLLENTUNA 08/7543788
(D2-kit,FLEX2,KCS)

Lars Landqvist
Harpsundsvägen 113 1tr
124 40 BANDHAGEN 08/995141
(SMOKE 5",KCS)

Bernt Svensson (BS)
c/o Petzäll,Kocksg 43,1tr
116 29 STOCKHOLM 08/448864

Lennart Uusitalo (LU)
Skrakvägen 20
141 72 HUDDINGE
08/973559

ANNONSER

Helsida	200:-
Halvsida	100:-
Kvartsida	50:-
Bilagor	200:-

Priserna gäller tryckfärdigt material.Radannonser för medlemmar är gratis om det inte gäller firma.Tidningens upplaga är för närvarande 250 ex.
För material till tidningen ring eller skriv gärna till någon i redaktionen ovan eller sänd detta till,PD68
BOX 98 122 21 ENSKEDE 1

STYRELSEN

PD68 nuvarande styrelse.

Tommy Bladh (TBl) Ordförande
Norrholmsvägen 106
132 00 SALTSJÖ-BOO

Nils Ohlsson (NO) V. Ordförande
Sandfjärdsgatan 68, 12 tr
121 69 JOHANNESHOV

Åke Holm (ÅH) Suppleant
Långsjövägen 15B
135 54 TYRESÖ

Anders Karlsson Sekreterare
Fasanstigen 40
197 00 BRO

Bo Ljungblad (BLj) Kassör
Brännkyrkagatan 117
117 28 STOCKHOLM

Lars Bergstedt Suppleant
Forvägen 31
145 51 NORSBORG

Gunnar Lovius V. Sekreterare
Ragnebergsvägen 21
136 67 HANDEN

MEDLEMSINTRÄDE

Vill DU bli medlem i PD68 ?
Sätt in 75:- på postgiro nr
PG 960468-7 samt namn adress o tele nr så kommer medlemskap på posten.OBS GLÖM EJ NAMN!!!

MPU-laren söker annons-ackvisitör !

Vill du sköta MPU-larens annons-kontakter ? ?
Vi söker någon som vill komma med i vårt glada RED-aktions-gäng, och som vill prata med våra annonsörer.
Kontakta någon i red. om du är intresserad.

LU

— I huvudet på en ordförande —

Snart är 1980 utgånget och det blir dags att boota om för ett nytt år. Fast innan vi gör RESET måste dock tomten komma. Ni vet vem jag menar. Den rödklädde äldre herrn med en näsa som antyder böjelse för starka drycker. Han som brukar komma med både hårdvaru och mjukvaru paket. De senare mottas numera lika entusiastiskt som de förra. Inga sura miner längre alltså.

Så småningom drar sig den gamle mot dörren. Efter ett par vänliga POKar i bröstet är han borta. Vi PEEKar ivrigt igenom julklappshögen. Kanske far i huset tar ett paket under armen och säger: -"Ursäkta mig, men jag måste gå och DISKa en stund". När alla hissat upp hakorna igen, tillägger han illmarigt mot den förstummade vänkretsen "..på båda sidorna! ".

Medan far sitter på kammarn och NEWDISkar kan vi andra fundera på vad vi skall göra med det nya året. Två enkäter har vi skickat ut för att ge oss vägledning. Den ena rörde MPU-laren, den andra om vi skall skaffa en klubbdator. Mer än hälften av alla medlemmar har nu svarat.

Glädjande nog har många dessutom kommit med extra synpunkter och ideer utanför enkäternas ramar. Fram till nästa utgåva av MPU-laren skall vi försöka sammanställa svaren. Dock kan vi redan nu lova en ny utskrift av medlemsregistret. Kompletterat med telefonnummer för de som angett att vi fick trycka det. Även datautrustningen kommer att tryckas. Det senare är ett önskemål från flera medlemmar. Eftersom vissa system har unika egenskaper (och problem), håller vi med om att denna extra information är värdefull. Onödig sekretess skall inte vara vår musik.

En kretiv avslutning på det gamla året och friska tankar inför det nya, önskar er alla:

TBC

TIDNIGSANKOR I GÅSTIDER

Förra numret av tidningen blev knapphändig, med få knappnedslag på redaktionens tangentbord. En kraftig tidsbrist och sent inkommet material satte oss i denna sits.

För att få ut tidningen i "tid" lät vi den tryckas utan kommentarer och en del material som inte kom med.

Den här avdelningen av tidningen får därför bli ett komplement till förra numret och rättelser avtidigare nummer.

——•——

Först kommer en beskrivning av promprogrammeraren från förra numret. Endast listningen kom med.

——•——

Därefter ett tilläg till programmet WHIST i nummer 2 -80 sidorna 7 till 10. I slutet fattades programmets subrutiner på ca. 200 basicrader. Detta har inte gett något ERROR-meddelande som har nått fram till vår kännedom förrän nu. Vi tackar så mycket för responsen.

Håll till godo med subrutinerna och låt speldjävulen leda dig rätt.

——•——

Föresten fattade ni vad figuren i nummer 3 -80 sida 42 var för något.

Den visade anslutningarna mellan controller och MPI51/52-minidiskdrive.

——•——

Har ni knappat in programmet OTHELLO i förra numret? Har ni kört det? Gick det bra? Inte?

Ni som har börjat, lägg av och bergunda detta först. Programmet kan erhållas av klubben på disk eller kassett i källformat. Sänd en diskett eller kassett till klubben med ett meddelande om vilket program som önskas.

För att programmet ska få plats måste max minne (32k) användas och köras comprimerad. Ändock blir endast 500 bytes kvar. Använd COMPRESS.CMD som finns beskrivet i MPU- laren nummer 1 -80. Detta program kan även det fås genom klubben. Vi tar även emot frågor angående programmet.

MEDLEMSANNONSER

ABC 80 grundenhet
6000 kr + moms
Även Flexenhet finnes
Sven-Erik Berggren
Tel. 0590 / 13995

Shugart diskdrive SA 400
EPromar, minneskort
Tommy Riboe
Tel. 08 / 881683

Shugart diskdrive SA 400
Roger Holmstrand
Tel. 08 / 7713852

PR-40 printer 600 kr
AC-30 kassettinterface 250 kr
Åke Stenäng
Ljungbergsv.25
564 00 BANKERYD
Tel. 036 / 72491

8k S100 bus-kort utan chips 200 kr
G Månstrand
Box 75
175 22 JÄRFÄLLA
TEL. 0758 / 39951 efter kl 14

4k-ram SS 50 300kr
Tomt 8k-ramkort SS 50 50kr
Hans Persson 0278/40874 eft 17.

ELTRAC-IC-Testboard (det stora)
med ca.70st.testsladdar 600kr
(har kostat ca.1400kr)
NIFE-ACKUMULATOR (ny),12v
10 Ah, 300kr
(har kostat ca.600kr)
säljes eller bytes mot 2114,

Allan Bratt Forslundaväg 10 I
54151 SKÖVDE
TELE 0500/16026.efter.kl.17

PROMPROGRAMMERARE FÖR 2716

HÄR FÖLJER EN BESKRIVNING ÖVER HUR MAN BYGGER EN BILLIG OCH ENKEL PROMPROGRAMMERARE FÖR DE POPULÄRA OCH NUMERA MYCKET BILLIGA 2716-PROMMARNA.
PROGRAMMERAREN BESTÅR AV TVÅDELAR, HÅRDVARA OCH MJUKVARA. FÖRST SKALL JAG BESKRIVA HÅRDVARAN, (SE SKISSEN). FÖRUTOM SJÄLVA PROMMET BEHÖVER MAN ENDAST 2 IC, EN 6821 ADRESSERBAR FRÅN DATORN OCH EN LATCH, 74LS375 DESSUTOM 2 TRISSOR, 2 DIODER OCH 3 MOTSTÅND. 74LS375'AN ANVÄNDS FÖR ATT LATCHA DE TRE HÖGSTA ADRESSBITARNA OCH FÖR ATT STYRA OMKOPPLINGEN MELLAN LÄS- OCH PROGRAMMERINGSLÄGE
CB2-UTGÅNGEN PÄ PIA'N FUNGERAR DÄRVID SOM STROB, VILKEN STYRS AV PROGRAMMET. FUNKTIONEN AV TRANSISTORERNA T1OCH T2 ÄR ATT KOPPLA OM VPP-INGÅNGEN PÅ PROMMET MELLAN +25V (PROGRAMMERING) OCH +5V (LÄSNING). AV DET SISTA FRAMGÅR ATT MAN PÅ NÅGOT SÄTT MÅSTE GENERERA +25V-SPÄNNINGEN, OM MAN INTE HAR TILLGÅNG TILL DEN DIREKT FRÅN DATORNS SPÄNNINGSAGGREGAT.
NÄR UTGÅNGEN Q3 PÅ LATCHEN ÄR LÅG SÅ ÄR OE (20) PÅ PROMMET LÅG, VIDARE SÅ ÄR T1 STRYPT D V S. VPP (21) PÅ PROMMET ÄR +5V, DÅ HAR VI KOPPLAT FÖR LÄSNING Å ANDRA SIDAN NÄR Q3 ÄR HÖG, SÅ ÄR OE HÖG OCH VPP +25V, DÅ HAR VI PROGRAMMERINGSLÄGE.
VALET AV TRANSISTORER ÄR INTE KRITISKT, UTAN DET FUNKAR ÄVEN MED ANDRA TYPER.
SOM SOCKEL FÖR PROMMET REKOMMENDERAS EN 'TEXTOOL'-SOCKEL SOM GÖR DET LÄTT ATT MONTERA I OCH UR PROMMET UTAN ATT RISKERA BÖJDA PINNAR.
HELA RASKET KAN LÄMPLIGEN MONTERAS PÅ EN BIT VERO-BOARD ELLER LIKNANDE.
NU KASTAR VI OSS RASKT ÖVER TILL MJUKVARAN, D.V.S DET VIKTIGASTE! PROGRAMMET, SOM FÖR ÖVRIGT ÄR SKRIVET SÅ ATT DET ÄR PROM-BART, DET HAR STARTADRESS $1000 MEN DETTA KAN MAN ÄNDRA EFTER ÖNSKEMÅL.
VIKTIGT ATT KOLLA FÖRST ÄR HUR FORT SYSTEMKLOCKAN SNURRAR, PROGRAMMET FÖRUTSÄTTER EN KLOCKA PÅ 1MHZ FÖR TIDS-LOOPAR. SKULLE MAN HA NÅGON ANNAN FREKVENS, MÅSTE MAN ÄNDRA EN KONSTANT PÅ ADRESS $110A OCH (EVT.) EN PÅ $10E4. VIDARE MÅSTE MAN KOLLA ATT DEN PIA MAN ANVÄNDER ADRESSERAS AV PROGRAMMET, SKULLE MAN HA EN ANNAN ADRESS ÄN $FCE0 FÅR MAN GÖRA NÖDVÄNDIGA ÄNDRINGAR.
VIDARE KAN DE ANROP SOM SKER AV SUBRUTINER I MONITORN BEHÖVA BESKRIVAS.
MBUGE SKALL VARA ETT HOPP TILL EN KONTROLLSLINGA I MONITORN, CBCDHX SKALL GÖRA OM ETT ASCII-TECKEN TILL EN BCD-SIFFRA,
INCH, OUTCH, OUT2H, OUT4HS, PCRLF, PDATA, PDATA1 OCH PSPACE SKA FUNGERA SOM 'EXBUG' ELLER 'MIKBUG'
IFBRK ÄR EN RUTIN SOM KOLLAR ACIA-STATUS,
HAR ETT X KOMMIT IN BRYTER PROGRAMMET OCH ÅTERVÄNDER TILL P-PROMTEN HAR MAN INTE DEN FUNKTIONEN KAN MAN ERSÄTTA DEN MED EN SUBRUTIN SOM BARA ÅTERVÄNDER MED Z-BITEN NOLLSTÄLLD.
PC SKALL VARA DEN MINNESCELL DÄR MONITORN HÄMTAR ADRESSEN VID ETT 'GO'-KOMMANDO.
NÄR PROGRAMMET VÄL ÄR PÅ PLATS, OCH HÅRDVARAN FIXAD, GER PROGRAMMET EN HEL DEL TREVLIGA MÖJLIGHETER. MAN KAN INTE BARA PROGRAMMERA, UTAN OCKSÅ LÄSA PROMINNEHÅLLET TILL MINNET FÖR EVENTUELLA ÄNDRINGAR, MAN KAN OCKSÅ VISA PROMINNEHÅLLET ELLER MINNESINNEHÅLLET PÅ TERMINALEN.
MED KOMMANDOT 'CHCK' KAN MAN BLANKCHECKA VALFRITT ANTAL BYTES I PROMMET, OCH MED 'STAT' FÅR MAN EN UTSKRIFT ÖVER DE AKTUELLA ADRESSERNA OCH PROMOFFSET. MED 'VERF' KAN MAN VERIFIERA ETT PROMINNEHÅLL MOT ETT MINNESINNEHÅLL.
OBSERVERA ATT VID KOMMANDOT 'PROG' SKER FÖRST EN BLANKCHECK PÅ DE BYTES SOM SKA PROGRAMMERAS, OCH EFTER PROGRAMMERINGEN SKER EN KONTROLL AV PROMMET MOT MINNET, SKULLE NÅGON OLIKHET UPPTÄCKAS SKER UTSKRIFT AV DE FELANDE POSITIONERNA OM PROGRAMMET UNDER BLANKCHECKEN UPPTÄCKER NÅGON BYTE SOM REDAN ÄR PROGRAMMERAD

(<>$FF), SKRIVS FRÅGAN 'CHANGE?' UT OFTA BEROR DETTA PÅ ATT MAN KNAPPAT IN FEL OFFSET FÖR PROMMET, D V.S MAN FÖRSÖKER PROGRAMMERA EN REDAN ANVÄND DEL AV DETTA, OCH ÄR MAN DÅ INTE ABSOLUT SÄKER PÄ ATT MAN VILL FÖRSÖKA ÄNDRA DET TIDIGARE PROGRAMMERADE, SÅ SKA MAN SVARA 'N' VARVID PROGRAMMET BRYTER.
EN FINESS SOM SKA FRAMHÅLLAS ÄR ATT HAR MAN INTE ANGETT EN VISS PARAMETER, T.EX. OFFSET, BLIR DEN AUTOMATISKT DET VÄRDE SOM ANGES AV 'STAT'. HAR MAN GÅTT UR PROGRAMMET MED 'EXIT' SPARAS MJUKSTART-ADRESSEN, SÅ ATT MAN KAN GÅ TILLBAKA BARA GENOM ATT GE ETT 'GO'-KOMMANDO.
FÖR ÖVRIGT SKALL FRAMHÅLLAS ATT PROGRAMMET ÄR MYCKET ANVÄNDARVÄNLIGT, SÅ ACCEPTERAS T.EX INTE PROGRAMMERING MED ADRESSER SOM ÄR STÖRRE ÄN PROMMET, SKULLE MAN FÖRSÖKA, SKRIVS 'SIZE?' UT. FÅR ATT KLARGÖRA ANVÄNDNINGEN YTTERLIGARE, FÖLJER HÖR NÅGRA EXEMPEL:

```
P  PROG,100,3FF,400  =PROGRAMMERA FR.O M ADRESS $100 T O M. ADRESS $3FF OCH BÖRJA PROGRAMMERA FRÅN PROMADRESS $400
P  SHWP              =VISA DET NYSS PROGRAMMERADE.
P CHCK 100 O         =GÖR EN BLANKCHECK AV PROMMET, $100 BYTES FR.O.M. OFFSET O
P     READ    900,9FF,700    =LÄS PROMINNEHÅLLET FRÅN OFFSET $700 TILL MINNESADRESS $900-$9FF
P VERF               =KOLLA ATT DET KOM DIT (OPT )
P EXIT               =HOPPA TILL MONITOR (GÖR NÅGRA ÄNDRINGAR PÅ ADRESS $900-$9FF)
P SHWM  =VISA MINNET,$900-$9FF
P  PROG   =PROGRAMMERA $900-$9FF TILL OFFSET $700
CHANGE?    Y         =ÖNSKAR ÄNDRA INNEHÅLL
P STAT  =VISA VILKA ADRESSER M.M. SOM ÄR AKTUELLA.
```

LYCKA TILL MED BYGGANDET!!!

BO MOLANDER

Flex nyheter

I nr 4 av FLEX Newsletter (nov 1980) berättar TSC att man börjar skeppa iväg sitt 6809 UniFLEX operativsystem. Än så länge bara för SWTPc S/09 och Gimix 6809 med en SWTPc DMAF-2 disk kontroller. Man berättar om hur förträffligt systemet är. Tag till exempel disk transfer hastigheten.
För floppy disk är den i storleksordningen 24k i sekunden. Både skriv och läsning. Detta motsvarar en ökning med 8 gånger jämfört mot FLEX för läsning och 16 gånger vid skrivning (med verify på). Med hårdisk blir speeden obetydligt under den hos en PDP-11/70. UniFLEX är ett fullt multi-user system där användarna loggar in med namn och hemliga "passwords". Flera jobb kan pågå samtidigt i kärran. Man kan exempelvis starta upp flera samtidiga assembleringar och under tiden sitta och editera. Även swapping understöds. Dvs ett jobb kan kastas ut från primärminnet för att ge plats åt ett annat.
För närvarande finns i mjukvaruväg en BASIC, BASIC precompiler samt textprocessor för körning under UniFLEX. Editor och assembler ingår. Jobb som pågår är PASCAL, C kompilator, relokerande assembler och länkare.

Enkel filtestare

En myckel enkel metod att kontrollläsa en fil beskrivs också. Valfri fil, text eller binär, går att kontrollera med avseende på läsbarhet. Man använder helt enkelt den vanliga LIST. Genom att specificera ett tillräckligt högt radnummer för start av utskrift, kommer endast en läsning att ske. Ingen utskrift. Exempel på kontroll av COPY.CMD:

```
  +++LIST 0.COPY.CMD,9999
```

Fortsättning sidan 16

```
2280 END
2290 REM SUBRUTIN SOM FÖDELAR KORTEN
2300 I=INT(RND(0)*4)+1
2310 J=INT(RND(0)*13)+1
2320 IF A(I,J)>0 THEN 2300
2330 RETURN
2340 REM SUBRUTIN SOM LISTAR SPELARENS KORT
2350 PRINT "  "
2360 PRINT "DU HAR :"
2370 IF F=0 THEN 2400
2380 I=F
2390 GOTO 2410
2400 FOR I=1TO4
2410 PRINT "  "
2420 D$=B$(I)
2430 FOR J=1TO13
2440 IF A(I,J)<>1 THEN 2470
2450 PRINT D$,C$(J)
2460 D$="    "
2470 NEXT J
2480 IF F<>0 THEN 2500
2490 NEXT I
2500 RETURN
2510 REM SUBRUTIN SOM VÄLJER PASS ELLER SPEL HOS ÖVRIGA SPELARE
2520 Y=0
2530 FOR I=1TO4
2540 X=0
2550 FOR J=11 TO 13
2560 IF A(I,J)<>F THEN 2590
2570 G=G+1
2580 X=X+1
2590 NEXT J
2600 IF X<Y THEN 2630
2610 V(2,F)=I
2620 Y=X
2630 NEXT I
2640 Y=0
2650 FOR I=1TO4
2660 X=0
2670 FOR J=4TO 1 STEP -1
2680 IF A(I,J)=F THEN X=X+1
2690 NEXT J
2700 IF X<Y THEN 2730
2710 Y=X
2720 V(1,F)=I
2730 NEXT I
2740 IF V(1,F)=0 THEN S=INT(RND(0)*4)+1
2750 IF V(2,F)=0 THEN S=INT(RND(0)*4)+1
2760 RETURN
2770 REM SUBRUTIN SOM SPELAR FÖRDELAKTIGASTE KORT VID SPEL
2780 REM I=AKTUELL SPELARE
2790 FOR J=1TO13
2800 IF A(S,J)<>I THEN 2860
2810 IF J<Y THEN 2860
2820 A(S,J)=0
2830 Y=J
2840 O=I
2850 GOTO 3040
2860 NEXT J
2870 REM INGET KORT HÖGRE
2880 FOR J=1 TO 13
```

```
2890 IF A(S,J)<>I THEN 2920
2900 A(S,J)=0
2910 GOTO 3040
2920 NEXT J
2930 REM INGET KORT I RÄTT FÄRG
2940 FOR P=1TO4
2950 FOR J=1 TO 13
2960 IF A(P,J)<>I THEN 3020
2970 IF E(I)=2 THEN 3000
2980 IF T>9 THEN 3000
2990 IF J>10 THEN 3020
3000 A(P,J)=0
3010 GOTO 3040
3020 NEXT J
3030 NEXT P
3040 N=J
3050 RETURN
3060 REM SUBRUTIN SOM SPELAR FÖRDELAKTIGASTE KORT VID PASS
3070 REM I=AKTUELL SPELARE
3080 FOR J=13 TO 1 STEP -1
3090 IF A(S,J)<> I THEN 3130
3100 IF J>Y THEN 3130
3110 A(S,J)=0
3120 GOTO 3300
3130 NEXT J
3140 REM INGET KORT LÄNGRE
3150 FOR J=1 TO 13
3160 IF A(S,J)<>I THEN 3210
3170 A(S,J)=0
3180 Y=J
3190 O=I
3200 GOTO 3300
3210 NEXT J
3220 REM INGET KORT I RÄTT FÄRG
3230 FOR P=1 TO 4
3240 FOR J=13 TO 1 STEP -1
3250 IF A(P,J)<>I THEN 3280
3260 A(P,J)=0
3270 GOTO 3300
3280 NEXT J
3290 NEXT P
3300 N=J
3310 RETURN
3320 REM SUBRUTIN SOM KONTROLLERAR SLAGET KORT
3330 PRINT
3340 IF F=0 THEN 3370
3350 PRINT "ANGE BARA VALÖR PÅ KORT"
3360 GOTO 3410
3370 PRINT "ANGE FÖRST FÄRG, SEDAN VALÖR"
3373 PRINT
3375 PRINT "EXEMPEL  HJÄRTER,FEM ( GLÖM INTE KOMMAT)"
3380 PRINT
3390 PRINT "VILKET KORT VÄLJER DU";
3400 INPUT Q$,R$
3405 GOTO 3420
3410 INPUT R$
3415 PRINT
3420 FOR I=1 TO 4
3430 IF B$(I) <> Q$ THEN 3500
3440 FOR J=1 TO 13
```

```
3450 IF C$(J) <> R$ THEN 3490
3460 IF A(I,J)=1 THEN 3560
3470 PRINT "DET KORTET HAR DU INTE"
3480 GOTO 3520
3490 NEXT J
3500 NEXT I
3510 PRINT "DET KORTET KÄNNER JAG INTE TILL"
3520 PRINT
3530 PRINT "FÖRSÖK IGEN"
3540 GOSUB 2340
3550 GOTO 3330
3560 A(I,J)=0
3570 X=I
3580 N=J
3590 RETURN
3600 REM SUBRUTIN SOM SKRIVER UT SLAGET KORT
3610 IF P<>0 THEN 3650
3620 PRINT L$(I), "SLÅR ";B$(S),C$(N)
3630 PRINT
3640 GOTO 3680
3650 PRINT L$(I), "SAKAR ";B$(P),C$(N)
3660 PRINT
3670 P=0
3680 RETURN
3690 REM SUBRUTIN SOM TOLKAR DITT KORT
3700 IF X<> S THEN 3760
3710 IF J<Y THEN 3740
3720 Y=J
3730 0=1
3740 P=0
3750 GOTO 3770
3760 P=X
3770 I=1
3780 GOSUB 3600
3790 RETURN
3800 REM SUBRUTIN SOM VÄLJER FÄRG
3810 Y=0
3820 IF M=0 THEN J=1
3830 IF M>0 THEN J=2
3840 IF 0=3 THEN 3890
3850 IF 0=4 THEN 3920
3860 IF W(4)=0 THEN 3950
3870 S=V(J,4)
3880 GOTO 4030
3890 IF W(1)=0 THEN 3950
3900 S=V(J,1)
3910 GOTO 4030
3920 IF W(2)=0 THEN 3950
3930 S=V(J,2)
3940 GOTO 4030
3950 S=V(J,0)
3960 W(0)=1
3970 GOTO 3990
3980 S=INT(RND(0)*4)+1
3990 FOR I=1TO 13
4000 IF A(S,I)=0 THEN 4070
4010 NEXT I
4020 GOTO 3980
4030 FOR I=1 TO 13
4040 IF A(S,I)=0 THEN 4070
```

```
4050 NEXT I
4060 GOTO 3950
4070 X=S
4080 RETURN
4090 REM SUBRUTIN SOM STANNAR PROGRAMMET
4100 IF F<>0 THEN 4140
4110 PRINT "SKRIV N NÄR DU ÄR KLAR"
4120 INPUT Q$
4130 PRINT
4140 RETURN
4150 REM SUBRUTIN SOM TESTAR OM KORT FINNS I RÄTT FÄRG
4160 FOR J=1 TO 13
4170 IF A(S,J) <>1 THEN 4210
4180 F=S
4190 Q$=B$(S)
4200 GOTO 4230
4210 NEXT J
4220 F=0
4230 RETURN
4240 REM SUBRUTIN SOM VÄLJER PASS ELLER SPEL SUBRUTIN
4250 IF M>0 THEN GOSUB 2770
4260 IF M=0 THEN GOSUB 3060
4270 RETURN
```

***** NYTT ******

********** MEK6802D5E **********

Det nya kitet från MOTOROLA (R) skiljer sig på några punkter från MEK6800D2.

* Färdigbyggt.
* Nätdel medföljer.
* Kontakter till/från KCS-interfacet.
* Förbättrad monitor.
* Power on reset.
* Förberett för yttre terminal genom att plats finns för MC6850, MC14411+kristal, MC1488-89 samt 25 pol. kontakt.
* Allt på ett kretskort.

Några likheter finns tyvärr kvar.

* In/ut kretsarna ej fullt avkodade.
* Klockfrekvensen skiljer sig från 1 MHZ (859 kHZ).
* Displayerna ej latchade (släcks vid programexekvering).

Priset är 1365 + moms.

Sammanfattningen gjord av Kjell Karlsson.

Datainspektionens författningssamling

Ansvarig utgivare: Rabbe Wrede

DIFS 1980: 1

Föreskrifter rörande tillämpningen av 2 § tredje stycket datalagen (1973: 289) Personregister för privat bruk

utfärdade den 29 maj 1980.

Datainspektionen föreskriver följande.

1 Personregister som får inrättas och föras utan tillstånd

Enligt denna författning får enskild person för sitt privata bruk inrätta och föra personregister utan sådant tillstånd som fordras enligt 2 § första stycket datalagen om registret uppfyller följande två villkor.

1. Registret innehåller inte uppgifter som avses i 4 § datalagen om annan än den som lämnat samtycke därtill.

2. Registret förs med hjälp av en teknisk utrustning som inte innefattar annat än

- en centralenhet med högst 64 kbyte primärminne för lagring av instruktioner och data,
- ett tangentbord,
- en dataskärm (bild-/textskärm),
- två kassettbandenheter,
- två flexskivenheter,
- skrivare och
- mätinstrument.

2 Allmänna upplysningar

Genom denna författning undantas ovan beskrivna personregister från datalagens bestämmelse att personregister inte får inrättas och föras utan datainspektionens tillstånd. Övriga bestämmelser i datalagen gäller i tillämpliga delar.

För personregister som används i yrkesmässig verksamhet finns särskilda undantagsregler (se DIFS 1980: 2).

Dessa föreskrifter träder i kraft den 1 juli 1980.

DATAINSPEKTIONEN
Jan Freese

Nils Rydén

Datainspektionens författningssamling

Ansvarig utgivare: Rabbe Wrede

DIFS 1980: 2

Föreskrifter rörande tillämpningen av 2 § tredje stycket datalagen (1973: 289) Personregister i yrkesmässig verksamhet över kunder m fl

utfärdade den 29 maj 1980.

Datainspektionen föreskriver följande.

1 Personregister som får inrättas och föras utan tillstånd

Enligt denna författning får personregister inrättas och föras utan sådant tillstånd som fordras enligt 2 § första stycket datalagen om registret uppfyller följande tre villkor.

1. Registret utgör förteckning över

– anställda, elever, hotellgäster, hyresgäster, konferensdeltagare, kunder, kursdeltagare, leverantörer, medlemmar, tävlingsdeltagare eller därmed jämställda

och används för

- administration av de mellanhavanden som med hänsyn till typ av register uppstår mellan registeransvarig och registrerad,
- framställning av meddelanden eller underlag för meddelanden till bank eller annan penninginrättning för in- och utbetalningsuppdrag eller för rättsliga åtgärder avseende registeransvarigs fordran,
- redovisning av resultat och statistik rörande verksamheten samt framställning av underlag för bokslut och revision eller

– författare eller därmed jämställda

och används för

- åtkomst av dokument

2. Registret innehåller inte uppgifter som avses i 4 § datalagen.

3. Registret förs med hjälp av en teknisk utrustning som inte innefattar annat än

- en centralenhet med högst 64 kbyte primärminne för lagring av instruktioner och data,
- ett tangentbord
- en dataskärm (bild-/textskärm),
- två kassettbandenheter,
- två flexskivenheter,
- skrivare och
- mätinstrument.

Den här artikeln är hämtad från Creative Computing nr 8 1980. Den är skriven av Stephen Kimmel. Författaren påskinnar att programmen vi skriver endast exakt ersätter något vi redan gör. Och dessutom själva gör fortare och effektivare. Låt oss se vad denna hädare säger:

Det finns ett begränsat antal programmerare härute och ett begränsat tidskvanta att spendera. Så varför slösar vi bort en massa tid på att skriva värdelösa program? Trots att det finns miljoner ord om hur man skall programmera, verkar det inte finnas något om vad man skall programmera. Hur kan man börja att skriva ett datorprogram utan att först välja något som skall datoriseras?

Svaret på denna brist på vägledning är enkel. Man kan skriva goda program för ett obegränsat antal saker. Nå, inte helt obegränsat. Vägledningen är negativ snarare än positiv. Den säger "Skriv inte", snarare än "Skriv på". Mina studenter och jag har formulerat följande regler som vägvisare:

1. Skriv inte programmet såvida inte uppgiften är något du gör frekvent.

Med frekvent menar vi fler än två gånger om året. Skälet är tvåfaldigt. För det första, om du inte vet vad du gör, kommer du inte kunna lära en dator att göra det. Kan du verkligen känna uppgiften om du endast använder kunskapen en gång om året? För det andra kommer det att ta dig minst (och det om du är mycket snabb och mycket duktig), tre gånger så lång tid att programmera en dator att göra något, som det skulle tagit dig själv att göra det. Om du bara använder programmet en eller två gånger om året har du inte gjort dig själv någon större tjänst.

För varje datorprogram man skriver, måste man ha ett korrekt exempel att testa det emot. Ingen, jag repeterar, ingen skriver program som varje gång är felfria. Varje program måste checkas och testas. En del fel kommer att ge sig till känna på ett självklart sätt. Likt programmet som rådde dig att tillsätta negativa ägg i sufflen. Andra fel kommer att nästla sig in mer oförmärkt. I svar som verkar rimliga.

Det tål att upprepas. Man måste ha ett korrekt exempel att checka programmet mot.

Det klassiska exeplet på ett dåligt program som skrivs under denna regel är ett Inkomst Deklarations program.
Man blir tvungen att arbeta igenom sin deklaration för hand för att få ett korrekt exempel att testa programmet med. Som slutsummering kan man säga att det skulle ta mellan tre till sex gånger så lång tid att hantera deklarationen med ett egenskrivet program. Och man skulle bli tvungen att skriva om programmet varje år, eftersom de ändrar lagarna så ofta.

Undantaget är förstås om man tänker sig att säljaprogrammet. Eller starta en deklarations service. Men i så fall gör man deklarationer flera gånger än en om året. Jag rekommenderar inte ens inköp av ett deklarations program. Såvida inte säljaren är beredd att garantera svaren - och det är han inte - måste man själv checka svaren. Vilket betyder handarbete.

Det andra undantaget är regel nummer två:

2. Skriv inte programmet om inte uppgiften är något du skulle göra oftare om den var lättare att utföra.

I sin enklaste definition är detta faktiskt vad datakraft betyder.

3. Skriv inte programmet om

en människa kan göra samma sak på mindre tid än fem minuter.

Alla de tre första reglerna kan reduceras till en enda variabel - tid. Om datorn skall berika människornas liv, om datorn skall frigöra mänsklig tankekraft för betydelsefullare saker, måste den möjliggöra för oss att utföra uppgifterna på kortare tid. Om den missar denna test, om det faktiskt tar längre tid att utföra uppgiften med dator, då har den blivit ett hinder, snarare än en hjälp.

Låt oss begrunda ett exempel från ett kommersiellt tillgängligt program. Iden går ut på att konvertera recept. Du står i köket och skall tillaga en måltid för 46 av dina intimaste vänner. Ditt recept har emellertid endast portioner beräknade för tolv personer. Du kan nu:

Ta kokboken till dator- rummet (extra sovrummet).	30 sek
Slå på datorn (två knappar).	2 sek
Läsa in interpretern, utilitys etc.	1 min
Läsa in receptetomräk- ningsprogrammet.	30 sek
Kör programmet och skriv ned svaren.	1 min
Slå av datorn.	2 sek
Gå tillbaka till köket.	30 sek
Total tid:	3:34

Eller du kan göra så här:

Gå till din lya och hämta kalkylatorn.	20 sek
Slå på kalkylatorn, (en knapp).	1 sek
Utför beräkningarna och skriv ned svaren.	3min
Stäng av kalkylatorn.	1 sek
Återvänd till köket.	20 sek
Total tid:	3:42

Eller du kan tillgripa människans traditionellametod:

Fuska omräkningen men- talt.	10 sek

Eller så kan man använda den andra traditionella metoden:

Gör tillräckligt för 48 personer och servera nå- got större portioner.	0 sek

Netto besparingen med databehandling är mindre än 8 sekunder. Även om man har en disk så att programmet laddas omedelbart, har man fortfarande inte sparat mer än drygt två minuter i förhållande till normal behandling. Låt gå att vi sparat lite extra tid. Vad skall vi göra med dessa två extra friköpta minuter? Sitta och betrakta hur vår fyra fots tårta faller samman - eller spendera dem på en promenad tillbaka till datorn för att spara ytterligare tid?

Fem minuter siffran är helt godtycklig, men är en bra tumregel. Ju mindre din tid är värd,desto kortare kommer den att vara.

4. Skriv inte programmet för att göra något som kan utföras bättre med en:

a.50 kronors räknedosa - exempelvis att balansera checkkontot.

b.10 kronors kortlåda med några 148*105 kort. Exempelvis julkortsregistret. Såvida du inte planerar att låta datorn trycka adressetiketter ... vilket är häpnadsväckande ofint.

c. 25 kronors kalender och en 20 öres penna - till exempel din anteckningsbok.

Man kan inte bara balansera sitt ckeckkonto, uppdatera sin adresslista och göra noteringar om kommande händelser snabbare för hand än med datorn. Utan man kan dessutom göra det betydligt billigare. De fysiska handlingarna är ungefär desamma. Datorbiten innebär förutom all den extra overheaden (gå till datorn, ladda programmet etc), även andra hinder, såsom begränat minne och data former. Och du tillbringar fyra timmars programmering för att få den att

göra var och en av dessa saker. En dator i dessa tillämpningar är helt enkelt slöseri med pengar ochtid.

Varför använder någon en dator för dessa saker? Lätt! De köpte den, och de måste hitta på något att göra med den.

Slutligen:

5. Skriv inte ett datorprogram om noggrannhet är betydelselöst.

Datorn kan kanske tala om för dig att du behöver köpa 12,76 pizzor när du själv skulle gissat 13 stycken. Problemet är att det inte spelar någon roll. Man måste ändå köpa 13 pizzor. Det blir ingen besparing varken i tid eller pengar. Det finns bara den svagt förstärkta insikten att man är en bra gissare, och den obehagligt krypande känslan att man överhuvudtaget slösat sina pengar på ett datorköp.

Liknande exempel finns i myriader. Människor har överlevt i århundraden trots att de endast approximalt balanserat sin budget. Det är möjligt att besparingar på minskade överskott så småningom kan betala datorinköpet, men jag betvivlar det.

Det finns alldeles för många värdefulla saker att programmera för den tid och möda som åtgår vid skrivandet av program som kväver och hindrar människan. Låt datorn göra vad den antas göra, nämligen att frigöra människan för större och bättre uppgifter. Allt man behöver använda är en smula fantasi. Den tar oss förbi det likgiltiga och in till det värdefulla.

Översättning: TBl

I Creative Computing går en serie om sortering. Författare är Albert Nijenhuis. Den är uppdelad i tre artiklar som vardera behandlar en speciell metod. Eftersom just sortering är en vanligt förekommande process, kan kanske en översättning av artikelserien för MPU-larens läsare vara av intresse. Del ett som presenteras här behandlar Insertion Sort. Snarlik den sk "Bubble sort" metoden.

Varför behövs sortering

Förr eller senare - i regel tidigare än förväntat - blir man tvungen att utföra någon form av sortering. Den lista som skall sorteras kan vara en kompilators symboltabell som man vill ha i alfabetisk ordning. Eller också kan man vilja ha de största 100 enheterna i ordning från en lista på 10000 enheter. Du kanske är en student som vill uppdatera skoltidningens adressregister. Har man en rörelse vill man förmodligen ordna de senaste använda kontona innan den stora databasen uppdateras.

Hursomhelst, datorer ägnar en stor del av sin tid med att sortera listor. Att hitta den bästa metoden för varje jobb är därför naturligtvis av stort intresse.

Några vanliga metoder

I den här artikeln och i ytterligare två kommande delar, skall vi diskutera ett par av de många tillgängliga metoder för sortering och visa situationer där de är speciellt användbara.

Innan man läser vidare kan det vara nyttigt att fråga sig hur man skulle bära sig åt för att sortera en lista på, låt oss säga, sex tal i stigande ordning. Chansen är stor att man kommer fram till något liknande den "Insertion Sort" som vi behandlar i den här artikeln. Eller "Bubble Sort" som är väldigt likartad och ofta nyttjad som

programmeringsuppgift.

Insertion Sort är extremt enkelt uppbyggd och lätt att programmera. Den är speciellt lämplig när man måste sortera listor med längder upp till tio enheter. När listan blir längre kommer jämförelserna som måste göras att bli oerhört många. I sådana fall är metoden definitivt inte att rekommendera. För korta listor är den dock mycket snabb. Endast ett minimum av bokföring blir då aktuell.

Heapsort är en metod som kraftigt ekonomiserar det antal jämförelser som utförs. Den allmänna principen som gäller är att om man fastställt att "a < b" och "b < c" finns det ingen anledning att jämföra a och c. Då antalet saker per steg som måste hållas ordning på är något fler, arbetar dennna sorteringsmetod något långsammare på korta listor än insertion metoden.

Både insertion sort och heapsort metoderna använder inget array utrymme förutom det som datat är lagrat i. Den egenskapen gör att dessa metoder blir enkla att använda som del av större program.

Den tredje metoden kallas linked merge-sort (länkad sammanslagning). Den kommer att diskuteras i den tredje artikeln. Metoden är särskilt användbar i situationer då de enheter som skall sorteras är skrymmande och svåra att omflytta. Eller då input data redan innehåller så pass mycket av den önskade ordningen att det vore slöseri att inte dra nytta av detta. Denna sorteringsmetod behöver några egna arbetsareor, och programkoden blir större än för de båda övriga.

Insertion Sort

Vi förutsätter nu att de tal som ska sorteras är lagrade i en array med n celler, a(1),...a(n). Sorteringen består i att växla innehållet i dessa celler så att det kommer i stigande ordning.

Följande sätt att se på problemet är utomordentligt viktigt för att förstå algoritmerna. Vi förutsätter att de första j elementen a(1)...a(j) redan är i ordning. Vid första påseende verkar detta lite magiskt - det ser ut att påstå att problemet redan är löst. I själva verket är det inte så. Om det överhuvudtaget finns någon magi, ligger den i ersättandet av det fast givna indexet n med indexet j. Där j i princip kan anta vilket som helst av värdena 1 ..,n. Påståendet att "a(1),..a(j) redan är i ordning" kommer att vara sant för vissa värden på j men inte för andra. Om det verkligen är sant för j=n, är uppgiften klar. Med det är sällan fallet. Allt vi är tillåtna att anta, är att det är sant för j=1, eftersom en lista med endast en beståndsdel är automatiskt sorterad. Vad vi ska skapa är en metod som - förutsatt att påståendet är sant för vilket j som helst, kommer att även få det att bli sant för nästa värde på j. Metoden kommer därvid att leda oss steg för steg från j=1 till j=n.

Till exempel. Låt oss anta att vi har en lista bestående av talen 1,2,6,8,12,5. Då blir påståendet sant för j=5 (pekar på talet 12), men inte för j=6 (talet 5). Sant därför att 12 är större än 8. Falskt eftersom 5 onekligen är mindre än 12. För att det även skall bli sant för j=6, jämför vi 5 och 12 och växlar plats på dem, enär de är i oordning. Nu har vi listan 1,2,6,8,5,12. Fortfarande fokuserad på 5 tar vi och jämför den med 8 och växlar plats. Därefter kommer jämförelsen med i den nya listan 1,2,6,5,8,12 som ånyo leder till ett utbyte. Nästa jämförelse, med 2, leder inte till ett utbyte. Vihar därför uppnått det sanna i påståendet för j=6.

För generella värden på j blir historien märkvärdigt lik, även om vi formulerar om den något. Först lagrar vi a(j+1) i b för att underlätta proceduren. (Så

föregående stycke sätter vi b=5.) Jämför nu b med a(j), a(j-1) etc, eller för att göra det ännu bättre, jämför b med a(i), där i succesivt antar värdena j, j-1 ..till avslutningen sker. Om jämförelsen av b och a(i) utfaller så att b är den mindre, flyttas a(i) till a(i+1) Repetera sedan jämförelsen med i ersatt med i-1.

```
1000 'SUBR: INSERTION SORT OF  A
     OF DIM  N
1010 FOR J=1 TO N-1
1020 LET B=A(J+1)
1030 FOR I=J TO 1 STEP -1
1040 IF B>=A(I) GOTO 1080
1050 LET A(I+1)=A(I)
1060 NEXT I
1070 LET I=0
1080 LET A(I+1)=B
1090 NEXT J
1100 RETURN
```

Figur 1

Om emellertid b är större än eller lika med a(i), instoppas b i a(i+1). Slutligen, om i når värdet 0, insättes b i a(1). Insättningen av b avslutar processen och innehållet i a(1), ..,a(j+1) är nu i ordning.

Basic programmet i figur 1 utför insättningen enligt ovan (raderna 1020 till 1080) i en loop j=1 to n-1.

J=	1	2	3	4	5	6	7	8	9
3	3	3	2	1	1	1	1	1	1
7	7	4	3	2	2	2	2	2	2
4	4	7	4	3	3	3	3	3	3
2	2	2	7	4	4	4	4	4	4
1	1	1	1	7	7	7	6	6	5
9	9	9	9	9	9	8	7	7	6
8	8	8	8	8	8	9	8	8	7
6	6	6	6	6	6	6	9	9	8
10	10	10	10	10	10	10	10	10	9
5	5	5	5	5	5	5	5	5	10

Figur 2

Figur 2 visar sorteringen av en lista med j värdet ovanför varje kolumn. Ursprungslistan är kolumnen längst till vänster.

Litteratur referenser.

Litteraturen vad det gäller sortering är mycket rik. Lyckligtvis finns ett helt uppslagsverk i ämnet för allt som hände före 1973, i Vol. 3 av D E. Knuth's The Art of Computer programming. Där tillägnas en 350 sidor detta ämne. Under tiden fortätter utvecklingen. Resultat publiceras i flera journaler, såsom i Association for Computing Machinery. Håll utkik efter dem i ditt skolbibliotek eller i en närbelägen skola med teknikämnen på programmet. (Nästa gång: Heapsort).

Översättning: TB1

.FLEX nyheter forts...

Räddning ur editor krasch

Du har editerat i timmar. Nöjd har du knappat in det sista 'S'. Då kommer det mardrömslika felmeddelandet: "ALL AVAILABLE DISK SPACE USED". Och du är ute i FLEX igen. Kallsvetten tränger fram. Grova kötteder får videoskärmen att blekna. Du har hela maskineriet i händerna. Bara sekunder återstår innan smörjan ligger ute på gården Pulsen bultar i tinningen...

** L U C N **

Det är inte fullt så farligt. Mycket går att rädda. Än så länge ligger allt i minnet. Hoppa in på varmingången. Den är $0203 för 6800 och $0003 för 6809. Skriv ut bufferten med P!. Nu ser du hur mycket som kan räddas.

Skriv nu 'SAVE'. OBS hela ordet. Du får nu en fråga om disk eller tape. Svara 'D'. Ge sedan ett komplett filnamn. Sätt i en disk som kan rymma vad som ligger i minnet innan du ger vagnretur. Nu susar minnesinnehållet upp på skivan under det nya namnet. Med hjälp av APPEND på nya och gamla filer kan du sedan kanske editera samman en komplett fil. Den här grejen har klarat mig några gånger.

SPELPROGRAM I BASIC

```
10 PRINT "INSTRUKTIONS (1=Y,0=N)"
20 INPUT Z
30 IF Z=1 THEN GOSUB 1170
40 A=-1
50  B=-1
60  C=-1
70  D=-1
80  E=1
90  F=-1
100  G=-1
110  H=-1
120  I=-1
130  J=0
140 PRINT
150 IF J/3*3=J THEN PRINT "";
160 IF A=1 THEN PRINT "* ";
170 IF A=-1 THEN PRINT "- ";
180 IF B=1 THEN PRINT "* ";
190 IF B=-1 THEN PRINT "- ";
200 IF C=1 THEN PRINT "* ";J
210 IF C=-1 THEN PRINT "- ";J
220 IF D=1 THEN PRINT "* ";
230 IF D=-1 THEN PRINT "- ";
240 IF E=1 THEN PRINT "* ";
250 IF E=-1 THEN PRINT "- ";
260 IF F=1 THEN PRINT "*"
270 IF F=-1 THEN PRINT "-"
280 IF G=1THEN PRINT "* ";
290 IF G=-1 THEN PRINT "- ";
300 IF H=1 THEN PRINT "* ";
310 IF H=-1 THEN PRINT "- ";
320 IF I=1   THEN PRINT "*      ";
330 IF I=-1 THEN PRINT "-     ";
340 IF E=1 GOTO 370
350 IF A+B+C+D+F+G+H+I=8 GOTO 1040
360 IF A+B+C+D+F+G+H+I=-8 GOTO 1070
370 PRINT "SHOOT";
380 INPUT Y
390 IF Y=0 THEN 1150
400  Q=Y
410 IF Q<1 THEN 370
420 IF Q>9 THEN 370
430 ON Q GOTO 490,550,600,660,710,780,830,890,940
440 REM ENTRYPOINT
450  J=J+1
460 GOTO 140
470 PRINT "YOU GAVE UP ON ";J;" TRYS!!!"
480 GOTO 1080
490 IF A=-1 GOTO 1010
500  A=-A
510  B=-B
520  D=-D
530  E=-E
540 GOTO 440
550 IF B=-1 GOTO 1010
560  A=-A
570  B=-B
580  C=-C
590 GOTO 440
600 IF C=-1 GOTO 1010
```

```
600 IF C=-1 GOTO 1010
610  B=-B
620  C=-C
630  E=-E
640  F=-F
650 GOTO 440
660 IF D=-1 GOTO 1010
670  A=-A
680  D=-D
690  G=-G
700 GOTO 440
710 IF E=-1 GOTO 1010
720  B=-B
730  D=-D
740  E=-E
750  F=-F
760  H=-H
770 GOTO 440
780 IF F=-1 GOTO 1010
790  C=-C
800  F=-F
810  I=-I
820 GOTO 440
830 IF G=-1 GOTO 1010
840  D=-D
850  E=-E
860  G=-G
870  H=-H
880 GOTO 440
890 IF H=-1 GOTO 1010
900  G=-G
910  H=-H
920  I=-I
930 GOTO 440
940 IF I=-1 GOTO 1010
950  E=-E
960  F=-F
970  H=-H
980  I=-I
990 GOTO 440
1000 RETURN
1010 PRINT "HEY !! YOU CAN ONLY SHOOT"
1020 PRINT "STARS, NOT BLACK HOLES."
1030 GOTO 370
1040 PRINT
1050 PRINT "YOU WON WITH ";J;" SHOTS"
1060 GOTO 1080
1070 PRINT "YOU LOST WITH ";J;"TRYS "
1080 PRINT
1090 PRINT "TRY AGAIN (1=YES,0=NO)";
1100 INPUT X
1110 IF X=1 GOTO 40
1120 IF X=0 GOTO 1150
1130 PRINT "FOLLOW INSTRUCTIONS "
1140 GOTO 1090
1150 PRINT "HOPE YOU HAD FUN"
1160 STOP
1170 PRINT "";
1180 PRINT "* * *  THERE ARE STARS"
1190 PRINT "- - -  AND BLACK HOLES"
1200 PRINT "* * *  IN THE UNIVERSE"
```

```
READY

RUN

INSTRUKTIONS (1=Y,0=N)
? 1
* * *  THERE ARE STARS
- - -  AND BLACK HOLES
* * *  IN THE UNIVERSE

1 2 3  YOU SHOOT A STAR *
4 5 6  NOT A BLACK HOLE -
7 8 9  BY TYPING ITS NUMBER

EACH STAR IS IN A GALAXY.
WHEN YOU SHOOT A STAR, EVERY-
THING IN ITS GALAXY CHANGES.
ALL STARS BECOME BLACK HOLES,
ALL BLACK HOLES BECOME STARS.
GALAXY MAPS:

1 * -  * 2 *  - * 3
* * -  - - -  - * *
- - -  - - -  - - -

* - -  - * -  - - *
4 - -  * 5 *  - - 6
* - -  - * -  - - *
- - -  - - -  - - -
* * -  - - -  - * *
7 * -  * 8 *  - * 9

PATTERNS TO LOOK FOR:

START     WIN      LOSE

- - -    * * *    - - -
- * -    * - *    - - -
- - -    * * *    - - -

TYPE '0' TO END GAME

TYPE '4' TO GO ON? 4

- - -  0
- * -
- - -      SHOOT? 5

- * -  1
* - *
- * -      SHOOT? 2

* - *  2
* - *
- * -      SHOOT? 0
HOPE YOU HAD FUN

STOP AT LINE 1160

READY
```

```
1210 PRINT
1220 PRINT "1 2 3  YOU SHOOT A STAR *"
1230 PRINT "4 5 6  NOT A BLACK HOLE -"
1240 PRINT "7 8 9  BY TYPING ITS NUMBER"
1250 PRINT
1260 PRINT "EACH STAR IS IN A GALAXY "
1270 PRINT "WHEN YOU SHOOT A STAR, EVERY-"
1280 PRINT "THING IN ITS GALAXY CHANGES "
1290 PRINT "ALL STARS BECOME BLACK HOLES,"
1300 PRINT "ALL BLACK HOLES BECOME STARS "
1310 PRINT "GALAXY MAPS:"
1320 PRINT
1330 PRINT "1 * -  * 2 * - * 3"
1340 PRINT "* * -    - -  - **"
1350 PRINT "- - -    - - -  - - -"
1360 PRINT
1370 PRINT "* - -   * -  - - *"
1380 PRINT "4 - -  * 5 *  - - 6"
1390 PRINT "* - -  - * -    - *"
1400 PRINT
1410 PRINT "-- -  - - -  - - -"
1420 PRINT "* *-  - - -  - * *"
1430 PRINT "7 * -  * 8 *  - * 9"
1440 PRINT
1450 PRINT "PATTERNS TO LOOK FOR:"
1460 PRINT
1470 PRINT "START    WIN     LOSE"
1480 PRINT
1490 PRINT "- - -   * * *     - -"
1500 PRINT "-* -   * - *   - - -"
1510 PRINT "-- -     * * *     - -"
1520 PRINT
1530 PRINT "TYPE '0' TO END GAME"
1540 PRINT
1550 PRINT "TYPE '4' TO GO ON";
1560 INPUT Z
1570 RETURN
```

Evie Nef's law. There is a solution to every problem; the only difficulty is finding it.

Unnamed law. If it happens, it must be possible.

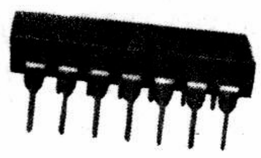
Miller's law. You can't tell how deep a puddle is until you step into it.

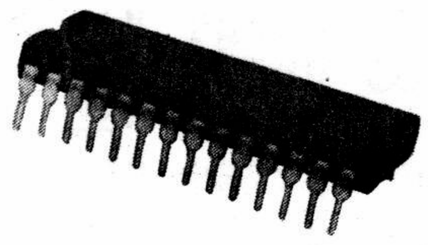
Uhlmann's razor. When stupidity is a sufficient explanation, there is no need to have recourse to any other.

Minnen!

SAMTLIGA PRISER INKL MOMS

MB8116E	16k dyn. RAM 200ns	32.-/st
	16 av ovanstående (32kByte)	475.-
2114L-3	4k stat. RAM 300ns	32.-/st
74LS240		7.20
74LS244		7.20
74LS245		15.70
74LS373		10.90
74LS374		10.90
6800	MPU 1MHz	30.-
6809	MPU 1MHz	120.-
6821	PIA	20.35
6875	klockkrets för 6800	25.35





FYND!

TTL-paket 90 IC, c:a 25 olika 74LS- 100.-

DIGITAL CONNECTION HB Box 288 175 25 JÄRFÄLLA

Senaste versionsnummer på TSC programvara

November 1, 1980.

Program Name	6809 Version	6800 Version
Extended BASIC	17	15
Extended BASIC Precompiler	4	2
BASIC	13	12
BASIC Precompiler	3	2
Multi-User BASIC	13	
Text Editing System	2	
Assembler	2	
Sort/Merge	2	2

RESA TILL SAUDIARABIEN.MIN ANDRA.

Av Alf Eriksson

Stockholm 3 mars 1980. Vaknade av väckarklockan 05.00. Efter frukost färd med bil till Arlanda.Vädret ganska bra, cirka 10 grader kallt men blåsigt.Avgång 07.30. Starten gick bra och väl uppe i luften var det dags för frukost.Resan till Köpenhamn tog ca en timme,temperaturen där 2 grader kallt. Ny avresa 09.45, bara 10 minuters försening. Flygvädret var fint,vi flög på ca 10000 meters höjd. Min litteratur under resan var Teknikens värld och en deckare. Efter mellanlandning i Athen, fortsatte vi till Jeddah. Vädret där var blåsigt och kändes kallt trots 20 grader Celsius. Efter tullen, som består av bord i U-form och ser till att ingen för in porr,fläskkött eller sprit,ringde jag till firman för hämtning. Kl 23.00 var jag "hemma" igen.Det var svårt att somna till ljudet av aircondition.

Efter frukost nästa morgon åkte jag till kontoret för vidare order. Hem och packa för vidare flygresa till Riyadh ungefär 110 mil från Jeddah. Efter bilfärd till flygplatsen checkade man in som vanligt. Ca en timmes väntan uppe i hallen, rökte och drack Pepsi Cola samt studerade den mängd av resande till alla delar av landet. Som vanligt på flygplatser,så är högtalaranläggningarna dåliga och därtill är arabernas engelska ej den bästa.När det var dags för avgång blev det en otrolig trängsel vid utgången,vilken hade formen av en kofålla.Det vore omöjligt att försöka klara en påse ägg genom denna promenad på ca 15 meter.Araberna nästan klättrar på varandra.Ganska roligt att studera inrikespassagerarna ombord, troligen ganska många som flög för första gången.Stora svårigheter med säkerhetsbältet.Jag studerade två stycken beduiner ,som försökte på alla sätt i många minuter. De fick till slut hjälp av flygvärdinnan, en liten Thailändska..

Jag fick sitta tillsammans med två Jemenbor,tror jag.Lukt av svett och stark parfym.Men resan varade bara 60 minuter.Jag skulle bli hämtad på flygplatsen av företaget,men detta fungerade ej,det fick bli taxi. Taxi i Saudi är inte det lättaste.Chaufförerna är dåliga på Engelska och det finns inga gatunamn.Återstod att visa vägen genom att be honom att köra antingen höger eller vänster.Tyvärr så missade jag en korsning så det blev en liten extra rundtur.Men resan gick bra och utan missöden.Annars är det hemskt att köra bil i Saudi. Bara i Riyadh dör det ca 400 personer i trafiken per månad.

Väl framme vid min blivande arbetsplats,morsade jag på sedan tidigare kända kamrater.Allt gick nästan enligt planerna utom att jag blev tvungen att jobba till Kl 21.00 samma kväll.Mitt arbete bestod i att lägga mattor i en del kontor och lägenheter samt en del takarbeten.

Näst sista dagen i Riyadh fick jag av en tillfällighet höra att det var kamelrace cirka 8 mil ut i öknen,vilket lät interessant. Ryktet spred sig så vi blev fyra man som for ut för att beskåda detta evenemang.Resan ut blev i bil,vägvisare fanns utsatta så det blev inga svårigheter att hitta.Vägen gick till en del ute i öknen så det var ganska dammigt.Ja detta var något man inte fick missa.Utefter ca 2 km av av banan var det bilar parkerade i 2 till 4 led med massor av folk, stående,liggande och sittande på bilarna med kameror och kikare för att kunna se och fotografera detta fantastiska. Bara att studera allt folk som var församlat var nästan tillräckligt. Man var ju i Saudi, så blickarna drogs ju inte bara till banan utan också till den kvinnliga delen av åskådarna. Det är ju en viss mystik med Saudiska beduinbrudar, slöjor och för oss ovana kläder. Religionen är ju när nere en

helt annan än vad vi är vana med,vilket för med sig en hel del, som jag inte nämner här.Härliga färger, svårt att förklara utan bilder. Jag såg på mina kamrater,de var nog mera interesserade av brudarna än av kamelerna.Dessa brudar med svarta slöjor utom för ögonen som är obeslöjade och djupa av mystik.Jag förstår unga grabbars förtjusning.

Åter till racet.Första missade vi,det gick tidigare på dagen och innehöll prinsarnas kameler,cirka ett hundratal.Nu det verkliga racet. 2704 startande kameler. Banan är 2,2 mil lång och går i en oval,cirka 30 meter bred med jordvallar på bägge sidor.Starten har gått,alla kameler på en gång i flera led men det är inga problem, öknen är stor. Vi stod ca 1 km från starten,men det gick förvånansvärt bra att trots allt damm och sand som rördes upp.Ja jag tror mig icke vara kunnig att berätta allt som berörde mina ögon.Bara att studera startnumren på kamelerna var roligt.Dessa var målade på kamelernas halsar med arabiska siffror.Ni kan väl dessa,0-9: .Bara detta är en vetenskap.Förarnas ålder kan jag tänka mig var från 10 år och uppåt. Klädseln kan jag tänka mig i de flesta fall var den typiska långskjortan.. Har Ni sett eller läst om kameler tidigare så vet ni säkert om deras egenskaper, vilka för förarnas del måste vara svårt.Ibland så gick dom tvärs och stannade samt bar sig åt precis som dom ville själva.det roliga var när dom helt plötsligt stannade och gick ned på knä.Vad jag kunde se var det en konst att få en kamel att starta på nytt,ty under loppet såg man många exempel på detta.Bara att komma upp och få kamelen att resa sig verkar svårt.Passerar de sedan jordvallen blir dom diskvalificerade och besprutade med rödfärg på halsen vilket drabbade ganska många.Nog om detta ,en vinnare korades. Priset bestod i en summa pengar i detta fall Saudi Rialer.Troligen hade inte pengarna någon betydelse utan äran var det viktigaste.Återresan från platsen var också en upplevelse,tänk kanske flera tusen bilar på en gång dammande genom klara detta. öknen för att nå fram till vägen. Man bör vara en van bilförare för att klara detta.

Efter återkomst till Jeddah fick jag nytt arbete, denna gång med montering av kontorsmöbler i ett nybyggt kontorshus.Trevligt arbete som som varade tre dagar. Nya order.Jag skulle fara till Taif cirka 20 mil sydost från Jeddah. Jag och en kamrat,som heter Gunnar,fick ta en av firmans pickupper för denna resa. Fick innan avfärden en rese beskrivning som visade sig fungera utan problem.Taif är en stad som ligger 2000 meter över havet med fint klimat,torr luft och inte överdrivet varmt,bara mellan 30 till 40 grader varmt i skuggan.Vägen dit innebar en hel del nya bekantskaper, olika typer av öken, berg, sand och stepp i olika former.Väl uppe bland bergen blev vädret helt plötsligt dåligt,regn och åter regn så mycket att vindrutetorkarna inte klarade av detta.Av vår resa återstod cirka en timmes resa så färden fick gå långsamt.Framme vid arbetsplatsen återstod att hitta våra kamrater som tidigare var här nere.Arbetsplatsen denna gång är ett palats åt kronprins Fahd.Palats är lindrigt sagt,ty det är ett område med ett trettiotal hus. Vårt arbete bestod i att lägga takmaterial,i detta fall en gummiduk. Väl uppe på taket blev synen en chock.Här var det mängder av fundament för Airconditionering och rörledningar m.m. Jag tror säkert att det var 500 stycken på en yta av 3000 kvm uppdelat i sektioner.Snacka om tak va.

Som Ni förstår finns det en hel del att göra och se här nere om man har interesse och tid.Djur och natur många för oss främmande växter och djur, bland annat scorpioner,ormar,ödlor och

skalbaggar stora som möss,trevligt eller vad tycks.Jag trodde jag skulle få se en scorpion i verkligheten men det gick inte trots mycket letande i berg och dal.Röda Havet låg för oss långt borta,cirka 18 mil,men trots det slog vi oss lösa en fredag för att bada och ha det riktigt skönt.Resan dit blev för oss ett nytt äventyr genom att vi åkte fel väg.När vi kommit underfund med detta frågade vi efter närmaste väg till havet.Enligt två infödda kunde man åka genom öknen 3 till 4 mil för att komma dit vi ville.Ja vi körde vidare ut i öknen enligt anvisning på en väg, ja väg är för mycket sagt För i princip var det bara spår i sanden bland buskar gräs och mindre träd.Träd med 3 till 4 cm långa taggar.Färden blev guppig,upp och ner. Den slutade efter cirka 3 mil då vi trodde att vi åkt vilse.Vi träffade där några beduiner,vilka förklarade för oss att det inte gick att åka vidare utan bil med fyrhjulsdrift.Nå vi vände och hittade denna gång på rätt väg.Äntligen framme,skönt,raskt på med badbyxor och ett par gamla skor på fötterna samt plums i vattnet.Där vi stannade var det väldigt långgrunt 3-400 meter,för att komma ut på 1,5 meters djup.Där börjar det verkliga korallrevet.Mycket har man sett i bilder och filmer om röda havet men detta överträffade nog allt detta.Med cyklop och snorkel blev det svårt att slita sig från vattnet.Fiskar koraller i alla former och färger helt fantastiskt,åtminstone för mig som förr sysslat en hel del med sportdykning.Utan tvekan kan man ligga i timmar och hela tiden upptäcka nya fiskar,stora och små. Ja än en gång ,helt otroligt. Ute i naturen uppe på land finns för mig som nordbo många nya bekantskaper,växter och stenar mm.För att nämna några som är kända hemma som krukväxter Hibiskus,Nerium,Trillingblomma fast här i gigantformat. Mimosan blommade här under tiden jag var nere. Stora buskar helt gula till en början utan blad.Sandsten i helt otroliga formationer och för oss en hel del ovanliga stenar,troligen av vulkaniskt ursprung.Ja, som Ni förstår är fynden många.

Min tid här nere är tre månader,och jag har begärt att få åka hem 31 maj. Min resa ned till Jeddah gick utan bekymmer,bara en felkörning. Lördag är avresedag.Jag drar iväg till firman strax före 7.Blir satt i arbete,frivilligt denna gång med att hjälpa till med en del kemiska laborationer för undersökning av den Saudiska jorden.Firman säljer nämligen ett jordförbättringsmedel som heter barkmull made in Sverige.

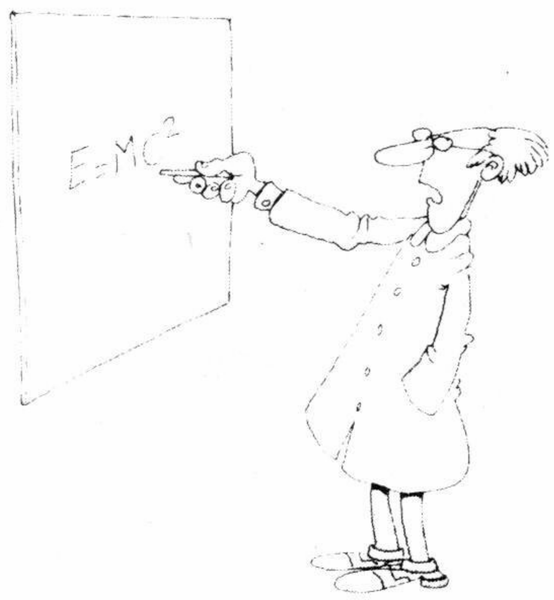
Jag oroar mig lite över att mitt visum har gått ut för tio veckor sedan.Utan visum är det svårt att lämna landet.

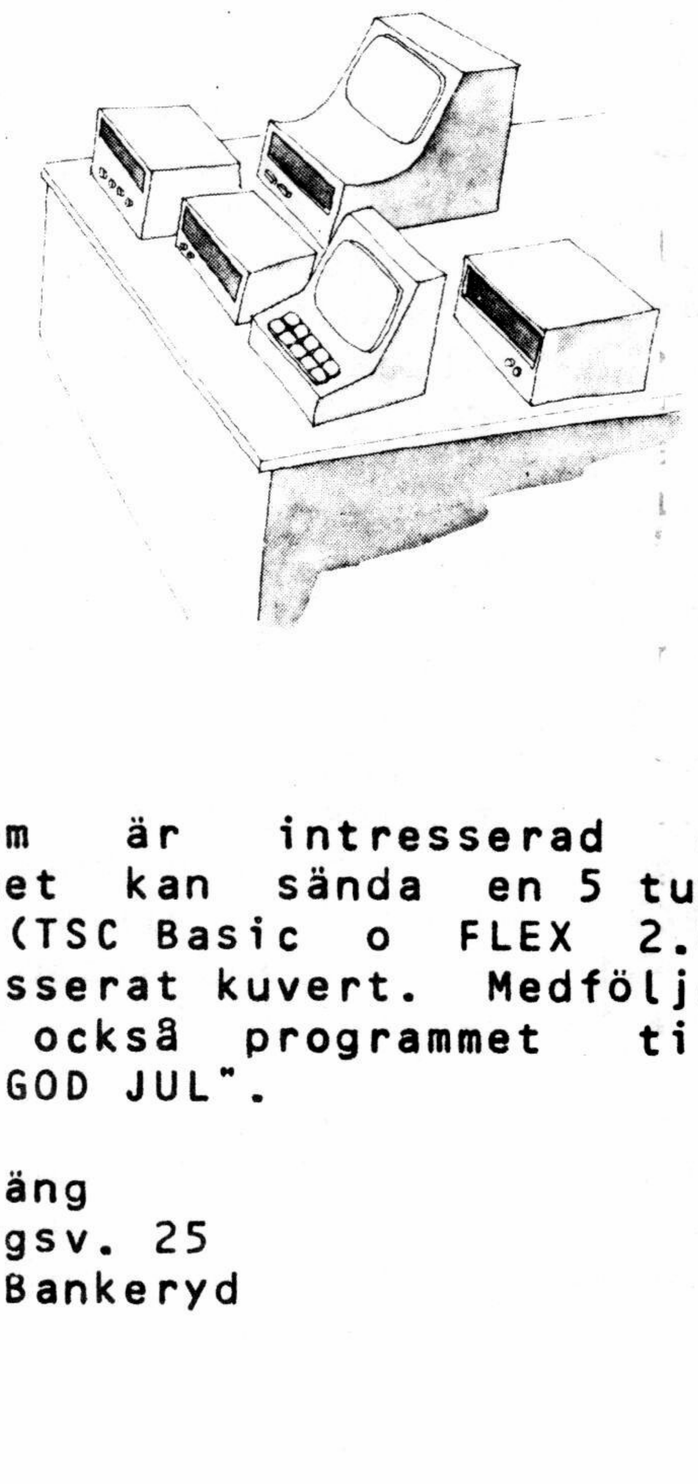
Väl ute på flygplatsen ordnade sig detta,jag var nog lite orolig innan. På flygplanet är det massor av folk,nästan fler än platser.Till min glädje blir jag tillfrågad om om jag vill flytta från min plats till förstaklassavdelningen.Vad skulle Ni ha gjort,flyttat förstås,så även jag .Middagen i första klass var en ny upplevelse för mig,vit duk på bordet, porslin och allt dukat med glas och bestick enligt alla regler. Till förrätt åt jag gravad laxforell,sedan grekisk sallad med ost,helt underbar. Varmrätt lamm med vitlökspotatis,perfekt.Ostbricka,camembert och roqefort. Kaffe med tårta.Rött vin att dricka till,portvin och ett glas likör till k kaffet.Maten helt perfekt,troligen till allas förnöjelse.Fast jag tror jag skulle kunna grava laxen bättre själv.

Efter förfrågan om jag kunde få besöka cockpiten,fick jag detta en kort stund,ty vi närmade oss köpenhamn.Detta besök var mycket interessant för en gammal dataamatör.Visserligen har jag sett bilder från flygförarnivå tidigare,men verkligheten är inte det samma.Färden gick över ett arbetsamt område bara en timme före

landning ,så besöket blev kort men intensivt.Landning 20.25 lokal tid alltså försenade en hel del. Min första blick ut på flygplatsen,grönt gräs och maskrosor.Härligt att vara hemma igen.
I väntan på avgång vidare till Stockholm så föll intresset på att studera folket i väntan.Det är egentligen ganska interessant,många olika typer, dom flesta med fötterna i ställning med inåtriktade tår.Tydande enligt gammal sägen på hemlängtan.
I planet hamnade kaptenen från förra planet bredvid mig.Han var på väg hem efter ett arbetsamt arbetspass i arabländerna. Tiden gick väldigt fort denna gång.Vi pratade hela tiden om våra fritidssysselsättningar. Det var interessant att få höra om hans fuskande i mitt yrke på sitt fritidshus vad gäller mattläggning i trappor. På Arlanda hämtas jag av min fru och min dotter och därmed var min andra period i Saudi över.Nu skall jag ha tre veckors semester.

Alf Eriksson
Tretum International
P.O. Box 2548 JEDDAH
Saudi Arabien





* D A T O R - K O N S T *

Ett julkort programmerat på en dator är det här programmet ett prov på. Kan kanske vara roligt att skicka ett helt annorlunda julkort och samtidigt visa ännu en sida av datorn, att framställa enklare konst!

Några kommentarer: Efter varje utskriven rad bör man backa valsen ett steg för att få tillräckligt tätt mellan raderna. Detta måste ske manuellt (rad 120). Maskinen måste ha BACKSPACE så att man kan skriva två bokstäver på samma ställe för den starkaste färgtonen. A4 formatet är förminskat hos en kopieringsfirma. Bilden blir då lättare att tyda på närmare håll.

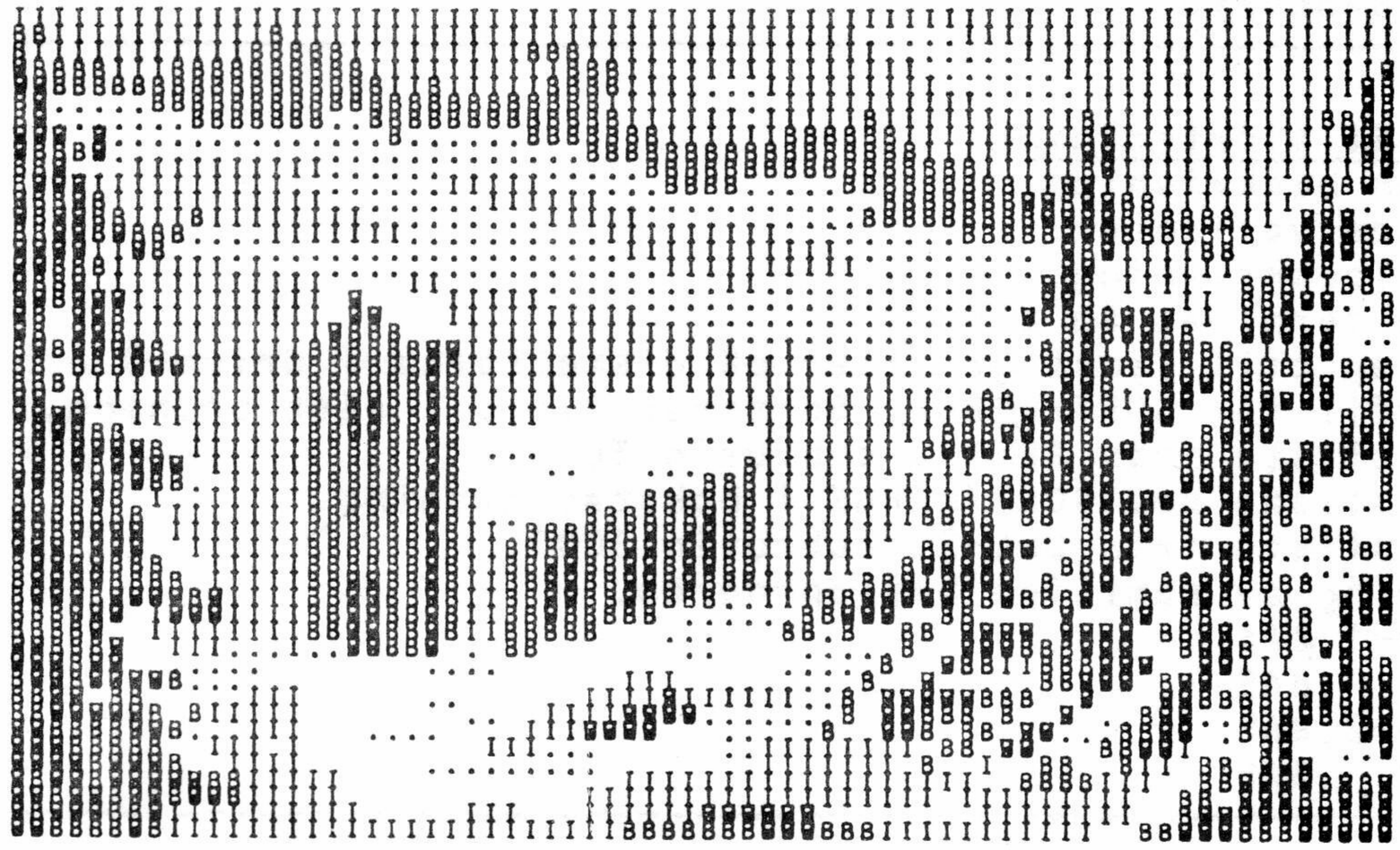
Den som är intresserad av programmet kan sända en 5 tums diskett (TSC Basic o FLEX 2.0) och adresserat kuvert. Medföljer gör då också programmet till texten "GOD JUL".

Åke Stenäng
Ljungbergsv. 25
564 00 Bankeryd

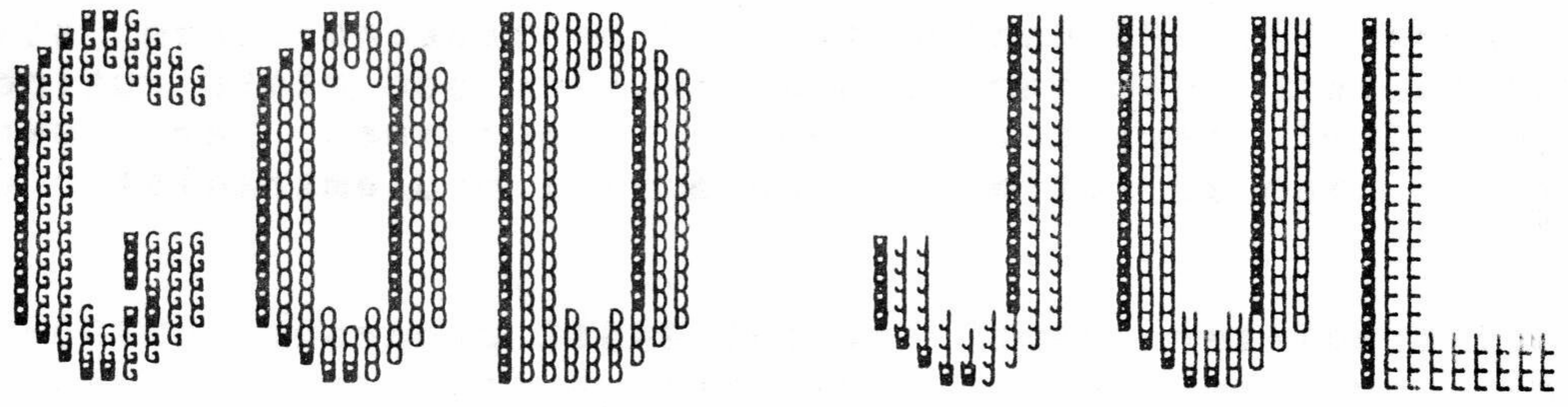
```
10 REM ************************************************************
20 REM *                  D A T O R  -  K O N S T                  *
30 REM * SKRIVEN AV LEE WILKINSON, USA. ÄNDRAD FÖR TSC-BASIC *
40 REM * OCH IBM I/O SKRIVMASKIN 73 AV ÅKE STENÄNG 80-08-03  *
50 REM *    För vidare information se KILOBAUD april 1980    *
60 REM ************************************************************
70 PRINT CHR$(12)
80 PRINT"* * * KYRKA * * *":PRINT:PRINT
90 PRINT"STÄLL IBM PÅ RADVASTÅND 1 OCH VÄNSTERMARGINAL PÅ 7!"
100 PRINT"TRYCK RETURN!":INPUTLINEAA$
110 OPEN "O.PRINT"AS 0
120 IF A$="C" THEN PRINT CHR$(7):PRINT"BACKA VALSEN ETT STEG!";:INPUTL
INEAB$
130 READ A$
140 IF LEN(A$)=3 THEN A=VAL(LEFT$(A$,2)):GOTO 160
150 A=VAL(LEFT$(A$,1))
160 A$=RIGHT$(A$,1)
170 IF A$="T" THEN A$=CHR$(08):GOTO 220:REM "TILLBAKA=BACKSPACE"
180 IF A$="C" THEN PRINT#0:GOTO120:      REM "CR"
190 IF A$="P" THEN A$=CHR$(46):GOTO 220:REM "PUNKT"
200 IF A$="S" THEN A$=CHR$(32):GOTO 220:REM "SPACE"
210 IF A$="E" THEN 260:                  REM "END"
220 FOR I=1 TO A
230 PRINT#0,A$;
240 NEXT I
250 GOTO 120
260 PRINT#0:PRINT#0
270 PRINT#0,TAB(26);"* *  K Y R K A  * *"
280 CLOSE 0:END
290 REM ************************************************
300 REM * Datasatserna till "KYRKA" komponerade *
310 REM *   och programmerade av Åke Stenäng    *
320 REM ************************************************
330 DATA 70I,1C,2B,11I,1B,29I,5P,22I,1C
340 DATA 1B,11I,5B,9I,3B,15I,7P,19I,1C
350 DATA 1B,1T,1W,4B,3I,10B,10I,3B,15I,7P,16I,1B,1T,1W,1C
360 DATA 2B,2T,2W,17B,2I,1B,5I,4B,4I,3P,9I,7P,14I,1B,1T,1W,1B,1C
370 DATA 2B,1T,1W,5P,23B,5I,4P,10I,5P,13I,1S,1B,1T,1W,1B,1C
380 DATA 1B,1B,1T,1W,7P,7B,2P,13B,6I,1P,5I,1B,10I,1B,11I,2B,1B,1T,1W,1
B,1C
390 DATA 3B,3T,3W,1S,1B,1T,1W,14P,1B,6P,7B,6I,6B,9I,1B,1B,1T,1W,11I,3B
,3T,3W,1C
400 DATA 1B,1T,1W,4B,1T,1W,24P,17B,8I,2B,2T,2W,12I,2B,2T,2W,1C
410 DATA 3B,1T,1W,3P,10I,16P,17B,5I,2B,1T,1W,12I,2B,1T,1W,1C
420 DATA 4B,3T,3W,10I,8P,5I,6P,4B,5P,9B,2I,2B,2T,2W,9I,1S,4B,1T,1W,1C
430 DATA 5B,5T,1W,2S,1W,1S,11I,8P,6I,14P,9B,2T,2W,1B,2B,2T,2W,2B,7I,1S
,1B,3P,1C
440 DATA 1B,1T,1W,3B,2T,2W,2B,3I,1B,8I,9P,5I,11P,13B,4T,4W,6B,2I,1S,3B
,3T,3W,2P,1C
450 DATA 4B,2T,2W,2B,1T,1W,3B,4P,7I,10P,9I,9P,8B,3T,3W,7B,2S,3B,3T,3W,
2B,1C
460 DATA 4B,4T,4W,2I,1B,1T,1W,1B,23P,11I,10P,1S,1B,1T,1W,1B,1B,1T,1W,4
I,2B,3S,1B,1T,1W,2B,2T,2W,1B,1C
470 DATA 2B,2T,2W,3B,6I,1P,1I,20P,10I,1S,8P,1S,2B,2T,2W,1S,6I,2S,2B,2T
,2W,1B,1S,1B,1T,1W,1B,1C
480 DATA 1B,1T,1W,2B,1B,1T,1W,12I,4S,2I,14P,2I,12P,2S,2B,2T,2W,1B,1S,4
I,2S,3B,2I,2B,1C
490 DATA 2B,2T,2W,1B,3B,3T,3W,10I,1S,1B,1T,1W,4S,5I,1S,23P,1S,2B,2T,2W
,1B,2P,2S,2I,1S,2B,3B,3T,3W,2P,1B,1C
500 DATA 1B,1T,1W,1B,1S,2B,2T,2W,1B,10I,1S,2B,2T,2W,3S,10I,1S,17P,1S,1
```

```
B,1T,1W,4B,3B,3T,3W,1S,1I,1S,2B,1B,1T,1W,4S,1B,1C
510 DATA 1B,1T,1W,1B,1S,2B,2T,2W,1B,10I,1B,1T,1W,3B,3S,12I,18P,1B,1B,1
T,1W,1B,2B,2T,2W,2B,2S,5B,5T,5W,1S,2P,1C
520 DATA 8B,8T,1S,1W,1S,4W,1S,7I,8B,2T,2W,14I,2P,1I,12P,3B,1T,1W,2I,3B
,3T,3W,2B,1S,2I,2B,2T,2W,1S,2P,1C
530 DATA 2B,1S,6B,3T,3W,6I,8B,2T,1W,1S,18I,10P,1S,4B,2T,2W,3B,1T,1W,7B
,1S,3B,1C
540 DATA 3B,12I,8B,2T,1W,1S,18I,2P,2I,6P,2S,2B,2T,2W,1B,2P,3B,3T,3W,3B
,1S,2B,2T,2W,1S,2B,1C
550 DATA 2B,2T,2W,1S,1B,11I,4B,2T,2W,2B,1B,1T,1W,1B,7I,5S,12I,2P,2B,1S
,2B,2T,2W,2B,2I,2B,2T,2W,1S,2B,4B,4T,4W,1S,2B,1C
560 DATA 4B,4T,4W,2S,9I,4B,2T,2W,2B,1B,1T,1W,1B,4I,10S,11I,2B,1S,3B,3T
,3W,2B,1S,2B,2T,2W,2S,1B,2B,2T,2W,3S,3B,1C
570 DATA 4B,3T,3W,2B,3S,6I,4B,2T,2W,4B,2T,1W,1S,11S,3P,1S,9I,5B,2T,2W,
3B,2P,1B,1T,1W,2S,1B,3B,3T,3W,3P,3B,3T,3W,1C
580 DATA 8B,7T,1W,1S,4W,1S,1S,6I,8B,2T,1W,2S,3P,11S,7I,1S,4B,3T,3W,2I,
5B,1T,1W,2S,4B,1T,1W,2S,5B,5T,5W,1C
590 DATA 9B,8T,1W,1S,1W,1S,4W,1S,5I,8B,2T,1W,5S,2P,3S,3P,2S,1B,7I,2S,2
I,2S,5B,3T,3W,3S,4B,2T,2W,1S,3B,3T,3W,1B,1S,1B,1C
600 DATA 9B,9T,2W,2S,1W,1S,1W,2S,2P,4I,8B,2T,1W,11S,1P,1S,3B,6I,1S,3I,
1S,7B,4T,1W,1S,2W,1S,2P,7B,7T,7W,3S,1B,1C
610 DATA 6B,2T,1W,2S,1I,2P,5I,8B,2T,1W,1S,1I,8S,6B,10I,5B,3T,2W,2S,5B,
5T,5W,2P,4B,4T,2W,3S,3P,1B,1C
620 DATA 7B,7T,2W,6S,7I,8B,2T,1W,1S,1I,2P,3S,9B,3T,1W,2S,8I,4B,1S,2B,1
S,4B,3T,3W,1S,6B,4T,2W,1C
630 DATA 7B,7T,2W,1S,4W,1S,7I,8B,2T,1W,1S,2I,1P,12B,6T,1W,1S,2W,2S,7I,
3S,2B,2T,2W,1S,3P,4B,2T,2W,1S,1B,2S,2B,2T,1W,2S,3B,1C
640 DATA 7B,7T,3W,1S,3W,2S,6I,8B,2T,1W,1S,2I,13B,10T,1W,2S,2W,1S,2W,2S
,6I,2S,6B,4T,4W,2S,3B,1T,1W,2S,5B,4T,3W,1S,3P,3B,1C
650 DATA 8B,7T,4W,5S,5I,8B,2T,1W,1S,2I,13B,11T,2W,2S,2W,1S,2W,2S,5I,2S
,6B,5T,5W,2S,4B,2T,2W,1S,3P,3B,3T,2W,2S,3P,1C
660 DATA 9B,8T,4W,5S,5I,8B,6T,2W,2S,1W,1S,2I,13B,11T,2W,2S,2W,1S,1W,3S
,3I,2S,3B,1T,1W,1I,4B,4T,4W,1S,1B,1S,2B,2T,2W
670 DATA 1S,1P,4B,2T,2W,4B,2S,2B,2T,2W,1C
680 DATA 11B,10T,3W,1S,4W,1S,1W,4I,8B,6T,2W,2S,1W,1S,2I,11B,9T,2W,2S,1
W,4S,2P,3I,2B,12B,12T,5W,2S,1W,3S,1W
690 DATA 2S,1B,1S,3B,3T,2W,1S,3I,2S,3B,3T,3W,1C
700 DATA 6B,3T,3W,1S,1I,3B,3T,3W,4I,8B,6T,2W,2S,1W,1S,2I,8B,6T,1W,6S,6
P,5B,3T,3W,2S,2B,4S,1B,1T,1W,2S,2B,2T,2W
710 DATA 1S,2B,2P,3B,1S,3B,3T,3W,1C
720 DATA 5B,1T,1W,2S,8I,8B,6T,2W,2S,1W,1S,2I,5B,3S,3P,3S,2B,1P,1B,2S,2
B,1S,4B,4T,4W,1S,4B,4T,4W,1S,4B,2T,2W,1S,3B,1S
730 DATA 1B,1T,1W,1C
740 DATA 6B,6T,2W,3S,1W,2S,3I,3P,1I,2P,5B,5T,2W,2S,1W,1S,1P,1S,2B,7S,2
P,4S,5P,1B,2S,9B,9T,3W,3S,3W,2S
750 DATA 4B,3T,2W,1S,1S,1B,1S,3B,3T,2W,1C
760 DATA 6B,6T,3W,1S,2W,2S,5P,8S,4P,14S,5P,1B,2S,3B,3T,1W,2S,2I,6B,3T,
3W,2S,2B,2T,2W,3I,2P,3B,3T,1W,1C
770 DATA 9B,8T,2W,1S,4W,1S,6P,16S,3I,2S,7P,1B,2S,1B,1T,1W,2S,3P,5B,3T,
3W,1S,3B,2T,2W,2S,1B,2S,4B,4T,4W,1C
780 DATA 4B,2T,2W,1S,2B,3S,2P,3I,6S,2P,6S,4I,1B,1T,1W,6I,2P,1B,1S,8B,8
T,5W,5S,1B,1T,1W,3S,12B,12T,2W,5S,2W,1S,2W,1C
790 DATA 8B,6T,5W,2S,1B,5I,8S,2P,2S,4I,4B,4T,4W,7P,1B,1S,5B,5T,2W,4S,1
B,2S,1B,1T,1W,1S,2P,3B,2T,2W,2P,8B,6T,2W,2S,2W,1C
800 DATA 9B,9T,2W,4S,2W,1S,3P,3I,3S,4P,4S,3I,4B,4T,4W,3S,3P,2S,1B,2S,3
B,3T,2W,1S,1S,2B,1S,2B,2T,2W,3P,4B,2T,2W,1S
810 DATA 1P,3B,1T,1W,3S,2B,2T,2W,1C
820 DATA 8B,8T,3W,7S,3I,1S,1I,9S,3I,3P,6S,2P,8I,1S,2B,1S,2B,2T,2W,3S,4
B,2T,2W,1I,3S,2B,2T,2W,2S,2P,1B,1T,1W,1C
830 DATA 9B,9T,4W,1S,3W,3S,4I,6S,9P,7S,1P,7I,1S,1B,2S,1I,2S,2B,2S,1B,2
```

```
I,1S,2B,1S,3B,3T,3W,3P,1C
840 DATA 12B,12T,1W,1S,2W,1S,5W,2S,5I,14S,18I,2S,4B,3I,2S,10B,8T,4W,1S
,1W,1S,1W,1C
850 DATA 12B,12T,2W,1S,2W,1S,2W,1S,2W,1S,5I,12S,18I,1S,9I,2S,2B,1S,8B,
8T,3W,2S,3W,1C
860 DATA 8B,8T,1W,1S,3W,2S,1W,10I,5S,3I,3S,6I,6B,6T,6W,8S,8I,2S,11B,10
T,7W,2S,1W,1C
870 DATA 8B,8T,2W,4S,1W,1S,23I,10B,3T,3W,3B,11I,2S,5B,3T,3W,8B,5T,5W,1
C
880 DATA 1E
```



* * K Y R K A * *



ÖNSKAR

Red: Carl Stael Åsa Lars Bert

Programmerat på en SWTPC 6800 dator av Åke Stenäng

* * ADRESSREGISTER av Åke Stenäng * *

Det här programmet är det första adressregistret i MPU-laren för TSC-basic och random datafil, men hoppas att det inte är det sista. Lösningarna är ju som alltid många och det finns förstås smartare sätt att lösa detta problem. Om Du har ett adressprogram så låt oss få se det!

För den som önskar finns en uppsättning av programmen på FD-68, eller skicka en diskett och adresserat kuvert till:
Åke Stenäng
Ljungbergsv. 25
564 00 Bankeryd

Innehållsförteckning:

A) Allmän beskrivning.
B) Använda variabler.
C) Detaljbeskrivning.
D) Annan terminal än CT-82.
E) Förslag till lämpliga filer på disketterna i drive 0 och drive 1.
F) Binärfil för SORT/MERGE. Sorterar Å,Ä,Ö i rätt ordning.
G) Adresslista från DEL-1, subrutin 3000.
H) Adresslista med SORT/MERGE i bokstavsordning.
I) Adresslista från DEL-2, subrutin 3000.
J) Listning av samtliga basicprogram.
   HUVUD-1.BAS
   DEL-1 .BAS
   HUVUD-2.BAS
   DEL-2 .BAS

A) * * Allmän beskrivning * *

Program 1. För max 320 namn (1 namn/sektor)
Program 2. För max 640 namn (2 namn/sektor)

Adressregistren är avsedda för FLEX 2.0, TSC:s basic, minifloppy, CT-82 terminal och en printer (CENTRONICS 779). Användarminnet måste vara 32 k plus minne för DOS. För den som har annan terminal än CT-82, se punkt D). På Centronics 779 är teckenbredden ställbar med en potentiometer. Har man bara en teckenbredd måste rutinerna 3000 i DEL-1 och DEL-2 arrangeras om. Programmen är skrivna som kundregister, men är lätta att ändra till medlemsregister eller annat register!

Varje adressregister består av två BASIC programm, HUVUD-program och DEL-program plus en DATA-fil. Man kan gå från det ena basicprogrammet till det andra genom ett kommando.

HUVUD-programmet har hand om inmatning av nya namn, ändring av variabler och borttagning av ett eller flera namn. Rutinerna LISTNING och ETT NAMN I TAGET kan visas på skärm. Ingen printerrutin finns i HUVUD-programmet.

DEL-programmet är endast ett utskriftsprogramm, antingen till printer eller på skärm. Rutinen LISTNING VERTIKALT kan användas för etiketter (en rad). LISTNING ETT NAMN/TVÅ RADER kan användas för en fullständig lista på alla namn.

Inget av registren har automatisk utökning av datafilens sektorer. Man får vid uppstartning reda på hur många sektorer som finns i datafilen och om den behöver utökas. Säkerhet finns också så att man inte kan mata in fler namn än det finns sektorer, utökning begärs då! Man får på detta sätt full kontroll över datafilens storlek.

Datafilen kan sorteras enligt nästan varje önskemål, t.ex i bokstavsordning (eller annan ordning), efter postnummer eller telefonnummer osv. För denna sortering används TSC:s SORT/MERGE. Att sortera inom basicprogrammen och med 32 k minne vore förstås smart! Kanske någon har en lösning på detta, t.ex. med "nycklar"!? I punkt H) har parametrarna för sorteringen satts i en binärfil som kallas PARA-1A för HUVUD-1.BAS. Kommandot för sortering i punkt H) blir då: +++P,PSORT,0.PARA-1A,REG-1.DAT. Filerna på disketterna måste då vara ordnade enl punkt E).

Om max antal namn skall skrivas in, får disketten endast innehålla COPY.CMD och CAT.CMD plus datafilen som då lämpligen utökas till full storlek vid första uppstartningen. En kort presentation av de båda adressregistrens kapacitet följer nu:

Adressregister för 320 namn.

Varje kund använder 1 sektor=252 tecken. Vid inmatningen tilldelas kunden ett unikt nummer. Om kunden tas bort blir numret vakant till nästa inmatning av ny kund. Datum för varje inmatning skrivs automatiskt in för varje kund. I detta register finns det plats för fler variabler då 87 tecken i varje sektor inte används i programmet.

Programmets inmatningsrutin innehåller:

Efter- o förnamn ell företag	max 30 tecken
C/O	max 20 tecken
Gata ell Box	max 30 tecken
Postadress	max 25 tecken
Tel nr	max 13 tecken
Land	max 25 tecken
Reserv	max 87 tecken

Adressregister för 640 namn.

Varje kund använder 1/2 sektor=126 tecken. Vid inmatningen tilldelas kunden ett unikt nummer. Om kunden tas bort blir numret vakant till nästa inmatning av ny kund. Datum skrivs ej in (får ej plats).

Programmets inmatningsrutin innehåller:

Efter- o förnamn ell företag	max 30 tecken
C/O	max 20 tecken
Gata ell Box	max 30 tecken
Postadress	max 25 tecken
Tel nr	max 13 tecken

B) * * Använda variabler * *

A,A$,AA$,A1$ = KUNDNUMMER (med början från 1001)
B$,BB$,A1$ = NAMN eller FÖRETAG
C$,CC$,C1$ = C/O
D$,DD$,D1$ = GATA eller BOX
E$,EE$,E1$ = POSTADRESS
F$,FF$,F1$ = LAND
G$,GG$,G1$ = TEL NR

```
     TR,TR$ = Totala antal använda records, inkl subrecords och borttagn
     BO,BO$ = Antal borttagningar.
     SE,SE$ = Antal sektorer i datafilen.

S$,SS$,S1$ = Datumsträng utan för- o efterspace, ÅÅ-MM-DD.
     YY,YY$ = År
     MM,MM$ = Mån
     DD,DD$ = Dag

          KA = Kundantal
          Z4 = Utökningsvariabel för sektorer i datafilen.
          Z5 = Recordräknare
           E = Recordnummer
           S = Väljer subrecord 1 eller 2 i program 2.
         DA$ = Datafilens namn i rad 85.
         FA$ = Firmans namn i rad 86.
         CR$ = Sätter 0D sist i varje record för SORT/MERGE.
          UT = Om UT=1 måste datafilen utökas.
           H = Menyvariabel
           G = Till skärm eller printer.
         (Y) = För minnesintagning.
B,D,Q,Q1     = Räknare för minnesintag. D ger antal namn till minne.
          SI = Sidnummer
         W,U = FOR NEXT variabel.
         AB$ = INPUTLINE variabel (div).

      T1-T5 = TAB-variabler i rutin 3000, rad 3315:

                                                          SID (T1=88)
NAMN          GATA (T3=31)          POSTADRESS (T4=62)    C/O (T5=88)
KUNDNR        DATUM                 TEL NR                LAND
```

C) * * Detaljbeskrivning * *

Rad 35-45: Ställer CT-82 i mode 82*20, baude 38400, keyp.=numerisk.
Rad 50: DP=0
Rad 55-75: Gör en datumsträng utan för- o efterspace, ÅÅ-MM-DD.

Rad 90,100,300-320: Programmet söker först efter en gammal datafil och om inte den finns går programmet genom rad rad 90 till rad 300. Där stängs den öppnade filen och OPEN gör en ny fil av storleken 4 sektorer och SE sätts då till 10 sektorer som i rad 315 skrivs in i:
program 1 i RECORD 1, SE$
program 1 i RECORD 2, subrecord 1 RB$

Rad 120: FIELD. I program 1 används FIELD en gång vid programmets uppstartning. I program 2 måste FIELD initieras varje gång subrecord skall ändras. Därför är FIELD som RETURN-rutin. För ökad snabbhet i inmatningen av nya namn finns en separat FIELD i 3161.

Genom att ställa S=1 eller 2 kan man få FIELD att välja subrecord 1 ell 2. I FIELD "126 ASZ$" får varje subrecord 126 tecken.

Rad 140-150: Tomma eller tidigare inskrivna variabler för:
TR= Totala antalet använda RECORDS,inkl subrecords o borttagning(ar) som ökas i inmatningsrutinen.
BO= Antal borttagningar.
SE= Antal sektorer i datafilen som skrivs in dels vid första uppstartningen i rad 315 och senare vid varje utökning i 190-200.

TR minus BO blir kundantalet KA.
Rad 153,162: Se INMATNING.
Rad 164: Om SE <= totala RECORDS + 10 (som utökningsreserv), så visas "BÖR DATAFILEN UTÖKAS".

SÖKT RECORD:
I program 1 ställs RECORD enkelt genom t.ex,
Z5=3:E=Z5:GET 1, RECORD E
I program 2 måste även subrecord 1 eller 2 bestämmas enl:
Ojämnt kundnummer= subrecord 1
Jämnt kundnummer= subrecord 2

Subrecord= S	S=1 kundnr	S=2
RECORD 1	1	2
RECORD 2	3	4
RECORD 3	5	6
RECORD 4	7	8
	osv	

ex. rutinen EN KUND I TAGET:
I rad 1100 (Z5=A-1000) får man ett tal som är ett kundnummer minus 1000. Om talet är större än TR går programmet tillbaks till 1080, annars undersöks i rad 1110 om talet är ojämnt "IF INT(Z5/2)-(Z5/2)<> 0 osv". Vid ojämnt tal blir subrecord=1 och sökt RECORD blir då det ojämna talet +1 dividerat med 2. (Z5+1)/2. Därefter går programmet till FIELD (GOSUB 120) för initiering.

Om talet är jämnt så ställs i rad 1120 S=2 och talet divideras med 2 så har man sökt RECORD och subrecord.

LISTNING VERTIKALT:
För att slippa ha floppyn igång hela tiden tar programmet in 25 namn i minnet, rad 2300-2390 och skickar sedan ut dessa namn till skärm eller printer. Rad 2335-2340 ordnar så att listningen ej överstiger kundantalet KA.

INMATNING:
Programmet går olika vägar beroende på om det är 1) Första inmatningen 2) Annan inmatning (utan BO), 3) Om någon borttagning skett.

Inmatningen hindras helt om vissa flaggor ställts! Om KA=0 kan endast inmatningsrutinen användas!

1) Första inmatning: Genom rad 3020 ställs RECORD och kundnummer (i program 2 även subrecord). I rad 3100 förs 20 namn (eller mindre) till minnet och sedan över till disketten. Genom LSET skrivs de olika variablerna in på disketten från inmatningsrutinen 3300. TR$, BO$, SE$, Z$ och RB$(program 2) sätts tomma i varje RECORD därför att inget fält får bestå av null om datafilen skall användas i SORT/MERGE. Till sist skrivs aktuella data till TR, BO, SE in i resp RECORD enl FIELD-listningen.

2) Annan inmatning: Om inmatning skett tidigare och ingen borttagning finns så går programmet till 3025 resp 3025,3026 (program 2). TR är nu större än 0 så därför används samma undersökning (i program 2) som i EN KUND I TAGET för att hitta RECORD resp subrecord. Här måste dock RECORDS och subrecords ökas med 1, då det ju är följande RECORD som skall användas.

3) Om borttagning skett tidigare används rutinen 3030-3085(3090). Här sker en sekventiell sökning i 3045(3046). Ny inmatning i det vakanta

numret sker sedan på vanligt sätt. Rutinen används så länge något vakant nummer finns (rad 3080 resp 3085). För att kunna skriva in nya namn efter borttagning och filen är slut (UT=1) finns rad 3002,3102. Genom att hoppa över 3007 (som stänger för inmatning vid UT=1) kan nu ett nytt namn läsas in på ett vakant nummer. Ett stopp måste dock sättas i 3102, annars kan man fortsätta att läsa in helt nya namn efter en inmatning i borttagningsrutinen. I 3085 sätts ju TR+1 och genom att då sätta TR+5 i 3102 så är stoppet klart och ny inmatning i 3105 kan aldrig nås. Från 3211 går programmet direkt till 520 (MENY), då man i detta fall inte behöver veta om datafilen är slut eftersom vakant(a) nummer finns.

Inmatningsrutinen 3300: Rad 3305-3327 skrivs ut direkt och sedan går cursorn till första variabeln. Snedstreck (/) visar max tecken som kan användas. I rad 3330 går cursorn upp 5 resp 6 rader och sedan till höger 46 steg, och i 3335 kopplas cursorn till. Aktuell variabel skrivs nu in och cursorn stängs av. Därefter går cursorn en rad ner, 46 till höger osv.

Programmen 1 och 2 kan ta emot max 320 resp 640 namn. För att hindra vidare inmatning (disketten slut), så sätts flaggor för detta i 162,3005,3132.

Inmatningen får ju inte heller överstiga antal sektorer i datafilen. Detta räknas ut och flaggor sätts för att förhindra inmatning och urspårning i 153,3007,3133 och i 3212. Om inmatning önskas i detta fall begärs utökning av datafilen. DOS-systemet ger datafilen 2 tomma sektorer allra först i filen och kan ej användas för data. Själva basicprogrammen sätter som säkerhet 2 tomma sektorer sist i varje datafil. Därför " +4 " i dessa flaggrutiner.

BORTTAGNING: Samtliga variabler enl inmatningsrutinen 3300 sätts här tomma. B0 ökas med 1 vid varje borttagning och B0 skrivs in i resp RECORD.

ÄNDRING: Varje inmatningsvariabel kan ändras i denna rutin.

LISTNING, ETT NAMN/TVÅ RADER: Principen med sekventiell sökning är densamma som i LISTNING,VERTIKALT. Utskriften är till för 72 raders papper i bana. Sidnumrering ingår.

D) * * Annan terminal än CT-82 * *

Ändra enl följande i HUVUD-programmen:
Ta bort rad 35-45.
Ta bort rad 2040.
2700 IF C=KA THEN 520
10010 END

Ändra enl följande i DEL-programmen:
Ta bort rad 35-45.
2070 IF G=1 THEN 2300
3070 IF G=1 THEN 3300
10010 END
Ta bort rad 2882 och 3882.

Ta bort 3300-3375 och 5800-5890 och ersätt med följande:
För HUVUD-1.BAS gäller hela rutinerna enl nedan.
För HUVUD-2.BAS , ta bort 3360-3365 och 5880-5890.

```
3300 PRINT:PRINT"* * INMATNING * *":PRINT
3305 PRINT"NR";A
3310 INPUT"MAX 30 TECKEN!  EFTER- o FÖRNAMN ell FÖRETAG ";BB$
3315 IF LEN(BB$)>30THENPRINTCHR$(7):PRINT">30 GÖR OM":GOTO3310
3320 PRINT"MAX 20 TECKEN!  C/O (skriv äv C/O)..... ";:INPUTLINECC$
3325 IF LEN(CC$)>20THENPRINTCHR$(7):PRINT">20 GÖR OM":GOTO3320
3330 PRINT"MAX 30 TECKEN!  GATA ell BOX (skriv äv BOX). ";:INPUTLINEDD$
3335 IF LEN(DD$)>30THENPRINTCHR$(7):PRINT">30 GÖR OM":GOTO3330
3340 PRINT"MAX 25 TECKEN!  POSTADRESS..... ";:INPUTLINEEE$
3345 IF LEN(EE$)>25THENPRINTCHR$(7):PRINT">25 GÖR OM":GOTO3340
3350 PRINT"MAX 13 TECKEN!  TEL NR..... ";:INPUTLINEGG$
3355 IF LEN(GG$)>13THENPRINTCHR$(7):PRINT">13 GÖR OM":GOTO3350
3360 PRINT"MAX 25 TECKEN!  LAND..... ";:INPUTLINEFF$
3365 IF LEN(FF$)>25THENPRINTCHR$(7):PRINT">25 GÖR OM":GOTO3360
3370 PRINT:PRINT"TRYCK RETURN FÖR FORTSATT INMATNING!"
3375 PRINT"SKRIV "S" FÖR SLUT!";:INPUTLINEAB$:PRINT:RETURN

5800 PRINT B$;"  -ÄNDRAS TILL";
5810 INPUT BB$:LSET B$=BB$:RETURN
5820 PRINT C$;"  -ÄNDRAS TILL";
5830 INPUTLINECC$:LSET C$=CC$:RETURN
5840 PRINT D$;"  -ÄNDRAS TILL";
5850 INPUTLINEDD$:LSET D$=DD$:RETURN
5860 PRINT E$;"  -ÄNDRAS TILL";
5870 INPUTLINEEE$:LSET E$=EE$:RETURN
5875 PRINT G$;"  -ÄNDRAS TILL";
5876 INPUTLINEGG$:LSET G$=GG$:RETURN
5880 PRINT F$;"  -ÄNDRAS TILL";
5890 INPUTLINEFF$:LSET F$=FF$:RETURN
```

5800-5890 bör kanske begränsas enl INMATNING med max tecken!

E) * * Förslag till lämpliga filer i drive 0 och drive 1. * *

```
CATALOG OF DRIVE NUMBER 0
DISK: ADRESS-1 #1

 NAME    TYPE   SIZE  PRT

ERRORS   .SYS     9
FLEX2    .SYS    25
PRINT    .SYS     1
CAT      .CMD     5
LIST     .CMD     3
COPY     .CMD     5
TTYSET   .CMD     2
RENAME   .CMD     1
DELETE   .CMD     2
ASN      .CMD     1
BUILD    .CMD     1
P        .CMD     1
EXEC     .CMD     1
DATE     .CMD     2
LINK     .CMD     1
NEWDISK  .CMD     7
MAKEDISK.TXT      1
BASIC    .CMD    56
RENUMBER.CMD      2
CT-82    .CMD     1    (önskad uppstartning av CT-82 terminal)
```

```
TIME     .CMD      1
STARTUP  .TXT      1     (t.ex.   ASN,W=1,S=0:BASIC,0.HUVUD-1 )
COLL     .BIN      1
SRTMRG   .SYS     19
SORT     .CMD     14
PSORT    .CMD      3
HUVUD-1  .BAC     36     (Kompilerat program)
DEL-1    .BAC     22
PARA-1A  .BIN      2

SECTORS LEFT = 114


CATALOG OF DRIVE NUMBER 1
DISK: REG-1 #2

 NAME    TYPE   SIZE   PRT

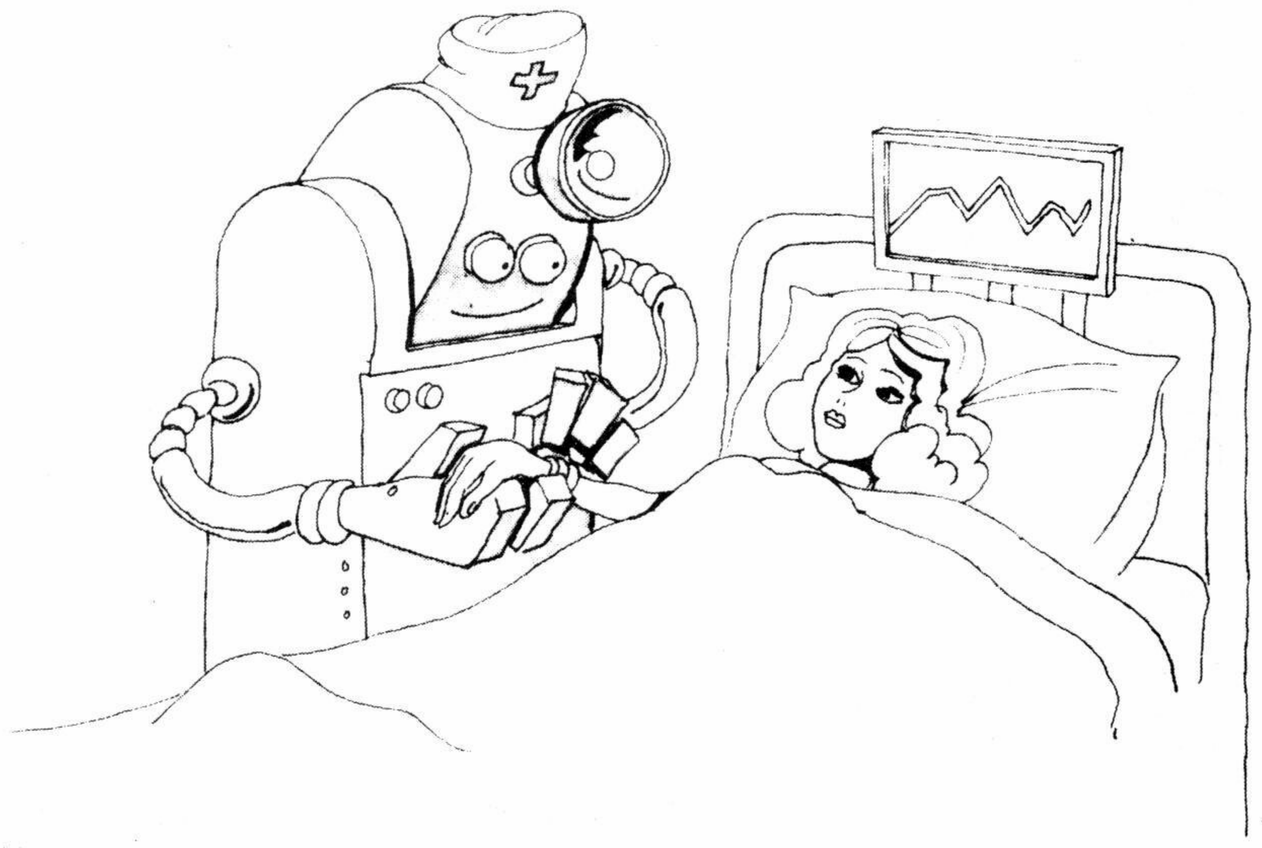
COPY     .CMD      5
CAT      .CMD      5
REG-1    .DAT    326

SECTORS LEFT = 4
```

F) * * Binärfil för SORT/MERGE * *

```
                * NAME: COLL.BIN
                * ALTERNATE COLLATING SEQUENCE FILE FOR SORT/MERGE
                * RIGHT SORT OF SWEDISH Å,Ä,Ö. By Å. Stenäng
                *
0600                    OPT     NOG
0600                    ORG     $0600
0600 41                 FCC     /ABCDEFGHIJKLMNOPQRSTUVWXYZÅÄÖ/
061D 61                 FCC     /abcdefghijklmnopqrstuvwxyzåäö/
063A 30                 FCC     /0123456789/
0644 21                 FCC     /!"$7o"()*+/
064F 2C                 FCB     $2C,$2D,$2E,$2F
0653 3A                 FCC     /:;<=>?e/
065A 5E                 FCC     /↑ /
065C 60                 FCB     $60,$7E,$7F
065F 00                 FCB     0,1,2,3,4,5,6,7,8,9,10,11,12
066C 0D                 FCB     13,14,15,16,17,18,19,20,21,22
0676 17                 FCB     23,24,25,26,27,28,29,30,31
                        END

NO ERROR(S) DETECTED
```



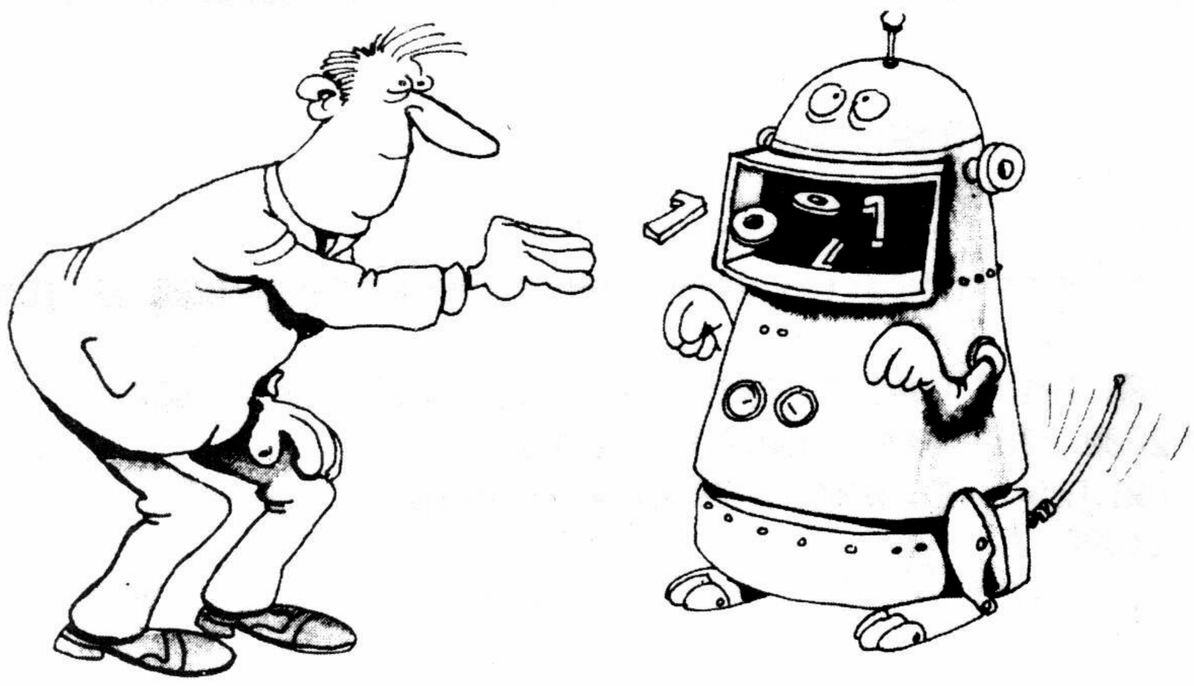
```
23 OCTOBER, 1980  TIME: 1750 27

10 REM ************************************
15 REM *       ADRESS-REGISTER            *
16 REM *       FöR MAX 320 NAMN           *
18 REM *       1 NAMN PER SEKTOR          *
20 REM * AV ÅKE STENÄNG 80-09-25 FöR       *
25 REM * TSC BASIC, MINIFLOPPY, CT-82      *
30 REM ************************************
31 REM *     HUVUD-1.BAS PROGRAM           *
32 REM ************************************
35 PRINT CHR$(28);CHR$(18)
40 PRINT CHR$(29);CHR$(29);CHR$(31)
45 PRINT CHR$(30);CHR$(30)
50 POKE HEX("AC03"),0
55 REM DATUMSTRÄNG ÅÅ-MM-DD
60 YY=PEEK(HEX("AC10")):MM=PEEK(HEX("AC0E")):DD=PEEK(HEX("AC0F"))
62 YY$=MID$(STR$(YY),2,LEN(STR$(YY))-2)
64 IF LEN(YY$)=1 THEN YY$="0"+YY$
66 MM$=MID$(STR$(MM),2,LEN(STR$(MM))-2)
68 IF LEN(MM$)=1 THEN MM$="0"+MM$
70 DD$=MID$(STR$(DD),2,LEN(STR$(DD))-2)
72 IF LEN(DD$)=1 THEN DD$="0"+DD$
75 SS$=YY$+"-"+MM$+"-"+DD$: REM DATUMSTRÄNG=SS$
80 DIM A1$(27),B1$(27),C1$(27),D1$(27),E1$(27),F1$(27),G1$(27),S1$(27)
85 DA$="REG-1":    REM DATAFILENS NAMN
86 FA$="MPU-LAREN":REM FIRMANS NAMN
90 ON ERROR GOTO 300
100 OPEN OLD DA$ AS1
120 FIELD £1,4ASA$,30ASB$,20ASC$,30ASD$,25ASE$,25ASF$,13ASG$,87ASZ$,8ASS$,3ASTR$,3ASBO$,3ASSE$,1ASCR
$
135 PRINT"* * ADRESSREGISTER (MAX 320 NAMN) FöR ";FA$;" * *":PRINT:PRINT
136 PRINT"* * HUVUD - PROGRAM * *":PRINT
140 GET£1,RECORD1:TR=VAL(TR$):BO=VAL(BO$):SE=VAL(SE$):KA=TR-BO
153 IF TR+4=SE THEN UT=1
155 PRINT DA$;". DAT FILEN INNEHÅLLER";SE;"SEKTORER. "
160 PRINT"ADRESSREGISTRET INNEHÅLLER";KA;"NAMN. ":PRINT
161 PRINT"PROGRAMMET ANVÄNDER ETT NAMN PER SEKTOR. ":PRINT
162 IF TR=320 THENPRINTCHR$(7):PRINT"OBS! DATAFILEN SLUT! ANTAL RECORDS = 320!":GOTO520
164 IF SE<=TR+10 THENPRINTCHR$(7):PRINT"OBS! DATAFILEN BöR EVENTUELLT UTöKAS!":PRINT
165 PRINT"öNSKAS DATAFILEN UTöKAD?  J/N  ";:INPUTLINEAB$
170 IF LEFT$(AB$,1)="J" THEN190
175 IF LEFT$(AB$,1)="N" THENPRINTCHR$(12):GOTO520
180 IFLEFT$(AB$,1)<>"J"ORLEFT$(AB$,1)<>"N" THENPRINTCHR$(7):GOTO165
190 PRINT:INPUT"HUR MÅNGA SEKTORER SKALL DATAFILEN UTöKAS MED ";Z4:Z5=Z4+SE
193 IF Z5>320 THEN Z5=320
195 E=Z5:PUT£1,RECORDE:SE=Z5+6:UT=0
200 GET£1,RECORD1:LSETSE$=MID$(STR$(SE),2,LEN(STR$(SE))-2):PUT£1,RECORD1
210 PRINTCHR$(12):GOTO520
300 IF ERR<>4 THEN ON ERROR GOTO
305 CLOSE1:OPEN DA$ AS1
310 PUT£1,RECORD4:SE=10
315 GET£1,RECORD1:LSETSE$=MID$(STR$(SE),2,LEN(STR$(SE))-2):PUT£1,RECORD1
320 RESUME 140
400 PRINTCHR$(12):PRINTCHR$(7):PRINT"REGISTRET ÄR TOMT! NR";H;"GÅR DÅ EJ ATT ANVÄNDA!"
520 REM * *  MENY  * *
522 PRINT:PRINT"REGISTRET INNEHÅLLER";KA;"KUNDER. "
525 PRINT:PRINT
530 ON ERROR GOTO 11000
540 PRINT TAB(5);"VÄLJ:"
550 PRINT TAB(5);"1.  EN KUND I TAGET       (SKÄRM)"
560 PRINT TAB(5);"2.  LISTNING, VERTIKALT   (SKÄRM)"
570 PRINT TAB(5);"3.  INMATNING"
580 PRINT TAB(5);"4.  BORTTAGNING"
```

```
590 PRINT TAB(5);"5.  ÄNDRING"
600 PRINT TAB(5);"6.  TILL DELPROGRAM MED DIV LISTNINGAR (PRINTER ELL SKÄRM)"
610 PRINT TAB(5);"7.  SLUT":PRINT
630 INPUT "VÄLJ NR";H
632 IF H>0 AND H<3 THEN IF KA=0 THEN 400
633 IF H>3 AND H<7 THEN IF KA=0 THEN 400
635 IF H<1 OR H>7 THEN PRINT CHR$(7):PRINT"FEL INLSLAGET!":GOTO 630
640 ON H GOTO 1000,2000,3000,4000,5000,6000,10000
1000 PRINTCHR$(12):PRINT"* * EN KUND I TAGET * *":PRINT
1080 ON ERROR GOTO 11000
1085 PRINT:INPUT"ANGE KUNDNUMRET ";A
1086 IF A<=1000 THENPRINTCHR$(7):PRINT"OBS! KUNDNUMRET SKALL VARA öVER 1000!":GOTO1085
1090 PRINT
1100 Z5=A-1000:E=Z5
1105 IF Z5>TR THENPRINTCHR$(7):PRINT"KUNDNUMMER";A;"FINNS EJ!":GOTO520
1140 GET £1, RECORD E
1150 IF B$="" THENPRINTCHR$(7):PRINT"NR";A;"ÄR BORTTAGEN!":GOTO520
1155 PRINTA$;TAB(7);S$
1160 PRINTB$:PRINTC$
1180 PRINTD$:PRINTE$;" ";F$:PRINTG$
1220 GOTO 520
2000 PRINTCHR$(12):PRINT"* * LISTNING VERTIKALT * *":PRINT
2040 PRINT CHR$(29);CHR$(29);CHR$(17)
2300 REM DISK TILL MINNE
2310 C=0:Q=0:B=0:Y=0:Z5=0
2330 D=25:REM ANTAL NAMN
2335 Q=Q+D
2340 IF Q>KA THEN Q1=Q-KA:D=D-Q1
2345 FOR B=1 TO D
2350 Z5=Z5+1:E=Z5
2355 GET £1,RECORD E
2360 IF B$="" THEN2350
2365 Y=B
2370 A1$(Y)=A$:B1$(Y)=B$:C1$(Y)=C$:D1$(Y)=D$
2375 E1$(Y)=E$:F1$(Y)=F$:G1$(Y)=G$:S1$(Y)=S$
2390 NEXT B
2400 REM NU TILL SKÄRM
2450 FOR Y=1 TO D
2455 C=C+1
2457 PRINTA1$(Y);TAB(7);S1$(Y)
2550 PRINTB1$(Y):PRINTC1$(Y)
2570 PRINTD1$(Y):PRINTE1$(Y);" ";F1$(Y):PRINTG1$(Y)
2575 PRINT
2670 NEXT Y
2700 IF C=KA THEN PRINTCHR$(29);CHR$(29);CHR$(31):GOTO 520
2710 GOTO 2330
3000 REM * INMATNING *
3002 IF BO>0 THEN 3010
3005 IF TR=320 THEN 162
3007 IF UT=1 THEN 3212
3010 PRINT CHR$(12):A=1000:C=0
3020 IF TR=0 THENZ5=1:E=Z5:A=A+1:GOTO3100
3025 IF BO=0 THENZ5=TR+1:E=Z5:A=A+Z5:GOTO3100
3030 Z5=0
3032 PRINT:PRINT"LETAR EFTER BORTTAGNING(AR)!"
3035 Z5=Z5+1:E=Z5:A=A+1
3040 GET £1,RECORD E
3045 IF B$<>"" THEN 3035
3050 BO=BO-1:GOSUB 3300
3062 PRINT:PRINT"NU FöRS 1 NAMN öVER TILL DISK (F.D.  VAKANT NUMMER)!"
3065 LSETB$=BB$:LSETC$=CC$:LSETD$=DD$:LSETE$=EE$:LSETF$=FF$:LSETG$=GG$:LSETZ$="":LSETS$=SS$
3070 PUT £1,RECORD E:KA=KA+1
3075 IF AB$="S" THEN 3200
3080 IF BO>0 THEN 3032
3085 Z5=TR+1:E=Z5:A=1000+Z5
```

```
3100 REM  TILL MINNE 20 NAMN ELL MINDRE
3102 IF TR+5>SE THEN 3200
3105 FOR I=1 TO 20
3115 C=C+1:GOSUB 3300
3120 AA$=MID$(STR$(A),2,LEN(STR$(A))-2)
3125 A1$(I)=AA$:B1$(I)=BB$:C1$(I)=CC$:D1$(I)=DD$:E1$(I)=EE$:F1$(I)=FF$:G1$(I)=GG$
3130 KA=KA+1:TR=TR+1
3132 IF TR=320 THENPRINTCHR$(7):PRINT"OBS! DATAFILEN SLUT! ANTAL RECORDS = 320!":GOTO3150
3133 IF TR+4=SE THEN UT=1:GOTO3150
3135 IF AB$="S" THEN 3150
3140 A=A+1
3145 NEXT I
3150 PRINT:PRINT"NU FöRS";C;"NAMN öVER TILL DISK!"
3160 FOR I=1 TO C
3165 GET £1,RECORD E
3170 LSETA$=A1$(I):LSETB$=B1$(I):LSETC$=C1$(I):LSETD$=D1$(I)
3172 LSETE$=E1$(I):LSETF$=F1$(I):LSETG$=G1$(I):LSETZ$=""
3173 LSETS$=SS$:LSETTR$="":LSETBO$="":LSETSE$="":LSETCR$=CHR$(13)
3180 PUT £1,RECORD E:Z5=Z5+1:E=Z5
3185 NEXT I
3200 GET £1,RECORD 1
3205 LSETTR$=MID$(STR$(TR),2,LEN(STR$(TR))-2)
3206 LSETBO$=MID$(STR$(BO),2,LEN(STR$(BO))-2)
3207 LSETSE$=MID$(STR$(SE),2,LEN(STR$(SE))-2)
3210 PUT £1,RECORD 1
3211 IF BO>0 THEN 520
3212 IF UT=1 THENPRINTCHR$(7):PRINT"OBS! DATAFILEN SLUT! MÅSTE UTöKAS INNAN NY INMATNING KAN SKE!":P
RINT:GOTO165
3215 IF LEFT$(AB$,1)="S" THEN 520 ELSE GOTO 3000
3300 PRINTCHR$(12):PRINTTAB(30);"* * INMATNING * *":PRINT
3302 PRINT"NR";A
3305 PRINT"MAX 30 TECKEN!  EFTER- O FöRNAMN ELL FöRETAG /                    /"
3310 PRINT"MAX 20 TECKEN!  C/O (SKRIV ÄV C/O)..... /                /"
3315 PRINT"MAX 30 TECKEN!  GATA ELL BOX (SKRIV ÄV BOX). /                    /"
3320 PRINT"MAX 25 TECKEN!  POSTADRESS..... /                    /"
3325 PRINT"MAX 13 TECKEN!  TEL NR..... /          /"
3327 PRINT"MAX 25 TECKEN!  LAND..... /                    /"
3330 PRINTCHR$(5);:FORW=1TO6:PRINTCHR$(1);:NEXTW
3335 FORW=1TO46:PRINTCHR$(9);:NEXTW:PRINTCHR$(21);:INPUT£0,BB$:PRINTCHR$(5);
3337 PRINTCHR$(2);:FORW=1TO46:PRINTCHR$(9);:NEXTW:PRINTCHR$(21);:INPUTLINE£0,CC$:PRINTCHR$(5);
3340 PRINTCHR$(2);:FORW=1TO46:PRINTCHR$(9);:NEXTW:PRINTCHR$(21);:INPUTLINE£0,DD$:PRINTCHR$(5);
3350 PRINTCHR$(2);:FORW=1TO46:PRINTCHR$(9);:NEXTW:PRINTCHR$(21);:INPUTLINE£0,EE$:PRINTCHR$(5);
3360 PRINTCHR$(2);:FORW=1TO46:PRINTCHR$(9);:NEXTW:PRINTCHR$(21);:INPUTLINE£0,GG$:PRINTCHR$(5);
3367 PRINTCHR$(2);:FORW=1TO46:PRINTCHR$(9);:NEXTW:PRINTCHR$(21);:INPUTLINE£0,FF$
3370 PRINT:PRINT:PRINT"TRYCK RETURN FöR FORTSATT INMATNING!  ";
3375 PRINT"SKRIV 'S' FöR SLUT! ";:INPUTLINE£0,AB$:PRINT:RETURN
4000 PRINTCHR$(12):PRINT"* * BORTTAGNING * *":PRINT
4025 ON ERROR GOTO 11000
4030 INPUT"ANGE MED KUNDNUMMER VILKEN KUND SOM SKALL TAS BORT";A
4032 IF A<=1000 THENPRINTCHR$(7):PRINT"OBS! KUNDNUMRET SKALL VARA öVER 1000!":PRINT:GOTO4030
4035 PRINT
4095 Z5=A-1000:E=Z5
4096 IF Z5>TR THENPRINTCHR$(7):PRINT"KUNDNUMMER";A;"FINNS EJ!":GOTO520
4105 GET £1, RECORD E
4107 IF B$="" THENPRINTCHR$(7):PRINT"KUNDNUMMER";A;"ÄR REDAN BORTTAGEN!":PRINT:GOTO4210
4115 PRINT B$;"  -TAS NU BORT UR REGISTRET!"
4116 KA=KA-1:BO=BO+1
4120 LSETB$="":LSETC$="":LSETD$="":LSETE$="":LSETF$="":LSETG$="":LSETS$=""
4125 PUT £1,RECORD E
4130 GET £1,RECORD 1
4135 LSETBO$=MID$(STR$(BO),2,LEN(STR$(BO))-2):PUT£1,RECORD1
4200 PRINT:PRINT"KLART!":PRINT
4205 IF KA=0 THEN 520
4210 INPUT"öNSKAR DU GöRA EN TILL BORTTAGNING  J/N  ";AB$
4220 IF LEFT$(AB$,1)="J" THENPRINT:GOTO4025
```

```
4230 GOTO 520
5000 PRINTCHR$(12):PRINT"* * ÄNDRING * *":PRINT
5010 ON ERROR GOTO 11000
5020 PRINT:INPUT"ANGE KUNDNUMRET";A
5025 IF A<=1000 THENPRINTCHR$(7):PRINT"OBS! KUNDNUMRET SKALL VARA öVER 1000!":GOTO5020
5030 PRINT
5050 Z5=A-1000:E=Z5
5060 IF Z5>TR THENPRINTCHR$(7):PRINT"KUNDNUMMER";A;"FINNS EJ!":GOTO520
5070 GET £1, RECORD E
5075 IF B$="" THENPRINTCHR$(7):PRINT"KUNDNUMMER";A;"ÄR BORTTAGEN!":GOTO 520
5083 PRINT
5090 PRINT "1.  ";B$
5100 PRINT "2.  ";C$
5110 PRINT "3.  ";D$
5120 PRINT "4.  ";E$
5125 PRINT "5.  ";G$
5130 PRINT "6.  ";F$:PRINT
5140 INPUT "VILKET NUMMER SKALL ÄNDRAS ";G:PRINT
5145 IF G<1 OR G>6 THEN PRINTCHR$(7):PRINT"FEL INSLAGET!":GOTO5140
5150 IF G=1 THEN GOSUB 5800
5160 IF G=2 THEN GOSUB 5820
5170 IF G=3 THEN GOSUB 5840
5180 IF G=4 THEN GOSUB 5860
5185 IF G=5 THEN GOSUB 5875
5190 IF G=6 THEN GOSUB 5880
5210 PRINT:PRINT"öNSKAR DU GÖRA NÅGON MER ÄNDRING I KUNDNUMMER";A;" J/N ";:INPUTAB$
5220 IF LEFT$(AB$,1)="J" THEN 5083
5230 PUT £1, RECORD E
5240 GOTO 520
5800 PRINT B$;"  -ÄNDRAS TILL /                                        /";
5810 FORW=1TO31:PRINTCHR$(4);:NEXTW:INPUT£0,BB$:PRINT:LSETB$=BB$:RETURN
5820 PRINT C$;"  -ÄNDRAS TILL /                                  /";
5830 FORW=1TO21:PRINTCHR$(4);:NEXTW:INPUTLINE£0,CC$:PRINT:LSETC$=CC$:RETURN
5840 PRINT D$;"  -ÄNDRAS TILL /                                        /";
5850 FORW=1TO31:PRINTCHR$(4);:NEXTW:INPUTLINE£0,DD$:PRINT:LSETD$=DD$:RETURN
5860 PRINT E$;"  -ÄNDRAS TILL /                                    /";
5870 FORW=1TO26:PRINTCHR$(4);:NEXTW:INPUTLINE£0,EE$:PRINT:LSETE$=EE$:RETURN
5875 PRINT G$;"  -ÄNDRAS TILL /                          /";
5876 FORW=1TO14:PRINTCHR$(4);:NEXTW:INPUTLINE£0,GG$:PRINT:LSETG$=GG$:RETURN
5880 PRINT F$;"  -ÄNDRAS TILL /                                    /";
5890 FORW=1TO26:PRINTCHR$(4);:NEXTW:INPUTLINE£0,FF$:PRINT:LSETF$=FF$:RETURN
6000 IF KA=0 THEN 400 ELSE CLOSE 1:CHAIN "DEL-1.BAS"
9999 REM * DIV SUBRUTINER *
10000 PRINT:PRINT "SLUT !":CLOSE 1
10010 PRINT CHR$(29);CHR$(29);CHR$(17):END
11000 REM * OM EN BOKSTAV SKRIVS I EN NUMERISK VARIABEL *
11005 IF ERR<>30 THEN ON ERROR GOTO
11010 PRINT CHR$(7):PRINT"FEL INSLAGET!"
11015 RESUME
```

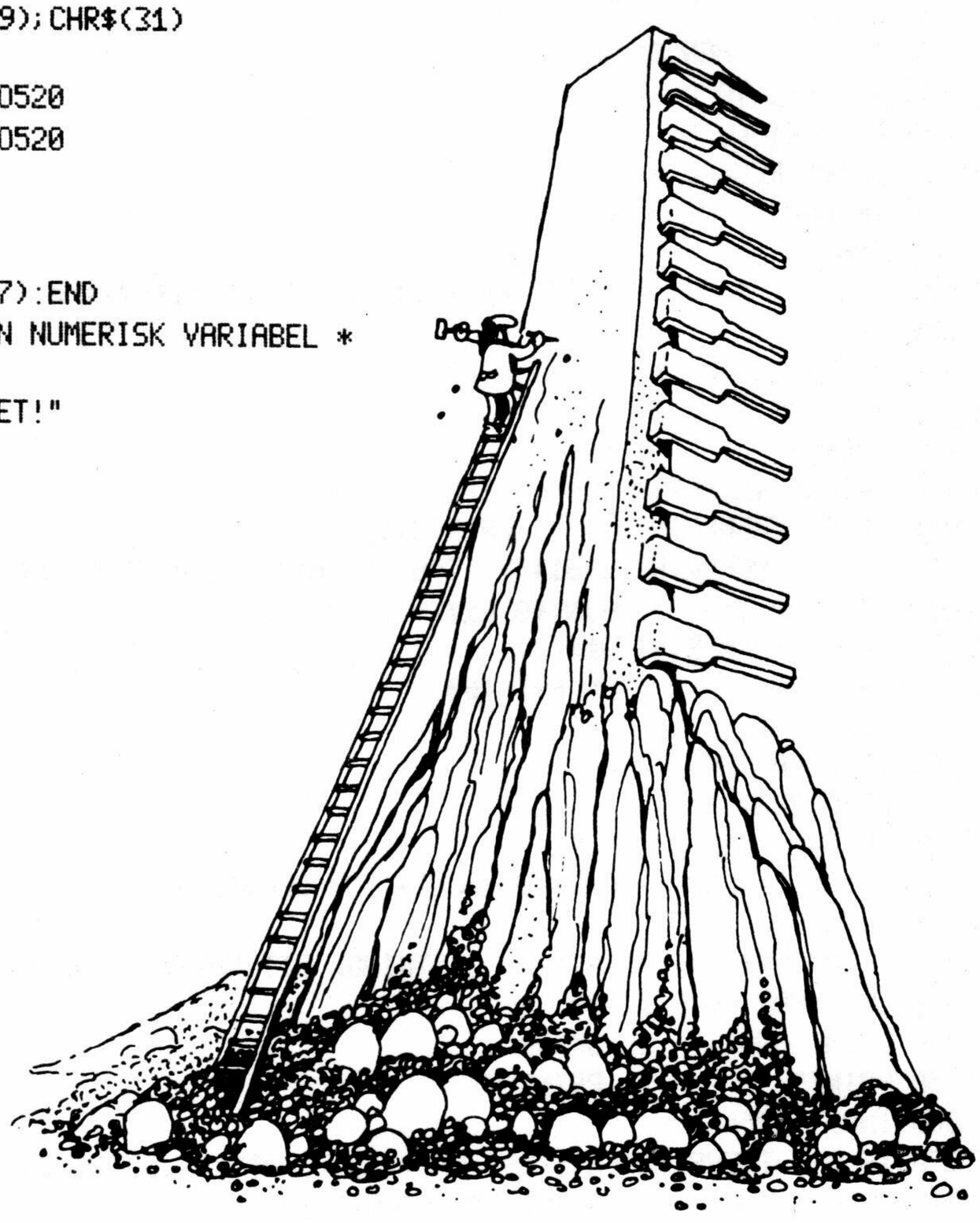


```
23 OCTOBER, 1980  TIME: 1755 46

10 REM ************************************
15 REM *       ADRESS-REGISTER            *
16 REM *       FöR MAX 320 NAMN           *
18 REM *       1 NAMN PER SEKTOR          *
20 REM * AV åKE STENÄNG 80-07-15 FöR      *
25 REM * TSC BASIC, MINIFLOPPY, CT-82     *
30 REM ************************************
31 REM *         DEL-1.BAS PROGRAM        *
32 REM ************************************
35 PRINT CHR$(28);CHR$(18)
40 PRINT CHR$(29);CHR$(29);CHR$(31)
45 PRINT CHR$(30);CHR$(30)
50 POKE HEX("AC03"),0
55 REM DATUMSTRÄNG åå-MM-DD
60 YY=PEEK(HEX("AC10")):MM=PEEK(HEX("AC0E")):DD=PEEK(HEX("AC0F"))
62 YY$=MID$(STR$(YY),2,LEN(STR$(YY))-2)
64 IF LEN(YY$)=1 THEN YY$="0"+YY$
66 MM$=MID$(STR$(MM),2,LEN(STR$(MM))-2)
68 IF LEN(MM$)=1 THEN MM$="0"+MM$
70 DD$=MID$(STR$(DD),2,LEN(STR$(DD))-2)
72 IF LEN(DD$)=1 THEN DD$="0"+DD$
75 SS$=YY$+"-"+MM$+"-"+DD$: REM DATUMSTRÄNG=SS$
80 DIM A1$(30),B1$(30),C1$(30),D1$(30),E1$(30),F1$(30),G1$(30),S1$(30)
85 DA$="REG-1":    REM DATAFILENS NAMN
86 FA$="MPU-LAREN":REM FIRMANS NAMN
87 AN$="å. STENÄNG, BANKERYD":AN=LEN(AN$): REM  ANVÄNDARENS NAMN ELL FIRMA
90 ON ERROR GOTO 300
100 OPEN OLD DA$ AS1
120 FIELD £1,4ASA$,30ASB$,20ASC$,30ASD$,25ASE$,25ASF$,13ASG$,87ASZ$,8ASS$,3ASTR$,3ASBO$,3ASSE$,1ASCR
$
135 PRINT"* * ADRESSREGISTER (MAX 320 NAMN) FöR ";FA$;" * *":PRINT:PRINT
150 PRINT"* * DEL - PROGRAM * *"
155 GET£1,RECORD1:TR=VAL(TR$):BO=VAL(BO$):SE=VAL(SE$):KA=TR-BO
160 GOTO 520
300 IF ERR<>4 THEN ON ERROR GOTO
310 PRINTCHR$(7):PRINT"DET FINNS INGEN DATAFIL! GÅ TILL HUVUDPROGRAMMET!"
320 RESUME 525
400 PRINTCHR$(7):PRINT"DET FINNS INGA NAMN I REGISTRET!"
520 REM * *  MENY  * *
522 PRINT:PRINT"REGISTRET INNEHåLLER";KA;"KUNDER. "
525 PRINT:PRINT
530 ON ERROR GOTO 11000
540 PRINT TAB(5);"VÄLJ:  (ALLA PRINTER ELL SKÄRM)"
550 PRINT TAB(5);"1. EN KUND I TAGET"
560 PRINT TAB(5);"2. LISTNING, VERTIKALT"
570 PRINT TAB(5);"3. LISTNING, ETT NAMN/TVå RADER + F:A 0 SIDNUMRERING"
580 PRINT TAB(5);"4. --"
590 PRINT TAB(5);"5. --"
600 PRINT TAB(5);"6. TILL HUVUDPROGRAM"
610 PRINT TAB(5);"7. SLUT":PRINT
630 INPUT "VÄLJ NR";H
632 IF KA=0 AND H<4 THENPRINTCHR$(12):PRINTCHR$(7):PRINT"REGISTRET ÄR TOMT! NR";H;"GÅR DÅ EJ ATT ANV
ÄNDA!":GOTO520
635 IF H<1 OR H>7 THEN PRINT CHR$(7):PRINT"FEL INLSLAGET!":GOTO 630
640 ON H GOTO 1000,2000,3000,4000,5000,6000,10000
1000 PRINTCHR$(12):PRINT"* * EN KUND I TAGET * *":PRINT
1005 ON ERROR GOTO 11000
1020 PRINT "ANGE:   1. SKÄRM"
1030 PRINT "        2. PRINTER"
1040 INPUT "NR";G
1045 IF G=1 THEN 1080
1050 IF G=2 THEN OPEN"0.PRINT"AS0:GOTO 1075
```

```
1070 PRINT CHR$(7):PRINT"FEL INSLAGET!":GOTO 1040
1075 PRINTCHR$(7):PRINT"ÄR PRINTERN KLAR - TRYCK RETURN";:INPUTLINEAB$
1080 ON ERROR GOTO 11000
1085 PRINT:INPUT"ANGE KUNDNUMRET ";A
1086 IF A<=1000 THENPRINTCHR$(7):PRINT"OBS! KUNDNUMRET SKALL VARA öVER 1000!":GOTO1085
1090 PRINT
1140 Z5=A-1000:E=Z5
1145 IF Z5>TR THENPRINTCHR$(7):PRINT"KUNDUMMER";A;"FINNS EJ!":GOTO520
1150 GET£1,RECORD E
1153 IF B$="" THENPRINTCHR$(7):PRINT"NR";A;"ÄR BORTTAGEN!":GOTO520
1155 IFG=1THENPRINTA$;TAB(7);S$
1160 PRINT£0,B$:PRINT£0,C$:PRINT£0,D$:PRINT£0,E$;" ";F$
1190 IFG=1 THENPRINT G$
1210 IF G=2 THEN CLOSE 0
1220 GOTO 520
2000 PRINTCHR$(12):PRINT"* * LISTNING VERTIKALT * *":PRINT
2020 ON ERROR GOTO 11000
2040 PRINT "ANGE:   1. SKÄRM"
2050 PRINT "        2. PRINTER"
2060 INPUT "NR";G
2070 IF G=1 THENPRINT:PRINTCHR$(29);CHR$(29);CHR$(17):GOTO2300
2080 IF G=2 THEN OPEN"0.PRINT"AS0:GOTO 2100
2090 PRINT CHR$(7):PRINT"FEL INSLAGET!":GOTO 2060
2100 REM FöR OLIKA VAL VID PRINTER
2110 PRINTCHR$(7):PRINT"ÄR PRINTERN KLAR - TRYCK RETURN";:INPUTLINEAB$
2300 REM DISK TILL MINNE
2310 C=0:Q=0:B=0:Y=0:Z5=0
2330 D=30:REM ANTAL NAMN
2335 Q=Q+D
2340 IF Q>KA THEN Q1=Q-KA:D=D-Q1
2345 FOR B=1 TO D
2350 Z5=Z5+1:E=Z5
2355 GET£1,RECORD E
2360 IF B$="" THEN2350
2365 Y=B
2370 A1$(Y)=A$:B1$(Y)=B$:C1$(Y)=C$:D1$(Y)=D$:E1$(Y)=E$:F1$(Y)=F$:G1$(Y)=G$:S1$(Y)=S$
2390 NEXT B
2400 REM TILL SKÄRM ELL PRINTER
2450 FOR Y=1 TO D
2455 C=C+1
2457 IF G=1 THENPRINTA1$(Y);TAB(7);S1$(Y)
2460 IF G=1THENPRINTB1$(Y):PRINTC1$(Y)
2550 IF G=2THENPRINT£0,B1$(Y);TAB(35);A1$(Y):PRINT£0,C1$(Y)
2570 PRINT£0,D1$(Y):PRINT£0,E1$(Y);" ";F1$(Y)
2572 IFG=1 THENPRINT G1$(Y)
2575 PRINT£0:IFG=2 THEN PRINT£0
2670 NEXT Y
2700 IF C=KA THEN 2880
2710 GOTO 2330
2880 IF G=2 THEN CLOSE 0
2882 IF G=1 THENPRINTCHR$(29);CHR$(29);CHR$(31)
2885 GOTO 520
3000 PRINTCHR$(12):PRINT"* * LISTNING, ETT NAMN/TVÅ RADER + F:A O SIDNUMRERING * *":PRINT
3020 ON ERROR GOTO 11000
3040 PRINT"ANGE:   1. SKÄRM"
3050 PRINT"        2. PRINTER"
3060 INPUT"NR";G
3070 IF G=1 THENPRINTCHR$(29);CHR$(29);CHR$(17):GOTO3300
3080 IF G=2 THEN OPEN"0.PRINT"AS0:GOTO3100
3090 PRINT CHR$(7):PRINT"FEL INSLAGET!":GOTO3060
3100 REM PRINTER HÄR
3110 PRINT:PRINTCHR$(7):PRINT"STÄLL IN PRINTERN PÅ 14 TECKEN/TUM - TRYCK RETURN";:INPUTLINEAB$
3300 REM DISK TILL MINNE
3310 C=0:Q=0:Q1=0:B=0:Y=0:Z5=0:SI=0
3315 T1=88:T3=31:T4=62:T5=88:REM TAB I UTMATNING
```

```
3330 D=30:REM ANTAL NAMN TILL MINNE
3335 Q=Q+D
3340 IF Q>KA THEN Q1=Q-KA:D=D-Q1
3345 FOR B=1 TO D
3350 Z5=Z5+1:E=Z5
3355 GET£1,RECORDE
3360 IF B$="" THEN 3350
3365 Y=B
3370 A1$(Y)=A$:B1$(Y)=B$:C1$(Y)=C$:D1$(Y)=D$:E1$(Y)=E$:F1$(Y)=F$:G1$(Y)=G$:S1$(Y)=S$
3390 NEXT B
3400 REM TILL PRINTER ELL SKÄRM
3402 SI=SI+1
3404 PRINT£0,"ADRESSREGISTER FöR ";FA$;"  ";SS$;"  REGISTRET=";KA;"KUNDER I KUNDNUMMERORDNING";TAB(T
1);"SID";SI
3406 FOR U=1TO105:PRINT£0,"-";
3407 NEXT U:PRINT£0
3408 PRINT£0,"NAMN";TAB(T3);"GATA ELL BOX";TAB(T4);"POSTADRESS";TAB(T5);"C/O"
3410 PRINT£0,"KUNDNR";TAB(T3);"DATUM";TAB(T4);"TEL NR";TAB(T5);"LAND"
3415 FOR U=1TO105:PRINT£0,"-";
3416 NEXT U:PRINT£0
3450 FOR Y=1 TO D
3455 C=C+1
3460 PRINT£0,B1$(Y);TAB(T3);D1$(Y);TAB(T4);E1$(Y);TAB(T5);C1$(Y)
3465 PRINT£0," ";A1$(Y);TAB(T3);S1$(Y);TAB(T4);G1$(Y);TAB(T5);F1$(Y)
3470 NEXT Y
3700 IF C=KA THEN FOR U=1 TO AN:PRINT£0,"-"; :NEXTU:PRINT£0:PRINT£0,AN$:GOTO3880
3710 FOR U=1 TO 7:PRINT£0:NEXT U:GOTO 3330
3880 IF G=2 THEN CLOSE 0
3882 IF G=1 THENPRINTCHR$(29);CHR$(29);CHR$(31)
3885 GOTO 520
4000 PRINT"FINNS INGET PÅ 4000!":GOTO520
5000 PRINT"FINNS INGET PÅ 5000!":GOTO520
6000 CLOSE 1:CHAIN "HUVUD-1.BAS"
9999 REM * DIV SUBRUTINER *
10000 PRINT:PRINT "SLUT !":CLOSE 1
10010 PRINT CHR$(29);CHR$(29);CHR$(17):END
11000 REM * OM EN BOKSTAV SKRIVS I EN NUMERISK VARIABEL *
11005 IF ERR<>30 THEN ON ERROR GOTO
11010 PRINT CHR$(7):PRINT"FEL INSLAGET!"
11015 RESUME
```



```
23 OCTOBER, 1980  TIME: 1759 22

10 REM ************************************
15 REM *      ADRESS-REGISTER             *
16 REM *      FöR MAX 640 NAMN            *
18 REM *      2 NAMN PER SEKTOR           *
20 REM * AV ÅKE STENÄNG 80-09-15 FöR      *
25 REM * TSC BASIC, MINIFLOPPY, CT-82     *
30 REM ************************************
31 REM *     HUVUD-2.BAS PROGRAM           *
32 REM ************************************
35 PRINT CHR$(28);CHR$(18)
40 PRINT CHR$(29);CHR$(29);CHR$(31)
45 PRINT CHR$(30);CHR$(30)
50 POKE HEX("AC03"),0
80 DIM A1$(27),B1$(27),C1$(27),D1$(27),E1$(27),G1$(27)
85 DA$="REG-2":    REM DATAFILENS NAMN
86 FA$="MPU-LAREN":REM FIRMANS NAMN
90 ON ERROR GOTO 300
100 OPEN OLD DA$ AS1:GOTO135
120 FIELD £1,(S-1)*126ASZ$,4ASA$,30ASB$,20ASC$,30ASD$,25ASE$,13ASG$,3ASRB$,1ASCR$:RETURN
135 PRINT"* * ADRESSREGISTER (MAX 640 NAMN) FöR ";FA$;" * *":PRINT:PRINT
136 PRINT"* * HUVUD - PROGRAM * *":PRINT
140 S=1:GOSUB120:GET£1,RECORD1:TR=VAL(RB$)
145 S=2:GOSUB120:GET£1,RECORD1:BO=VAL(RB$):KA=TR-BO
150 S=1:GOSUB120:GET£1,RECORD2:SE=VAL(RB$)
153 IF TR/2+4=SE THEN UT=1
155 PRINT DA$;".DAT FILEN INNEHåLLER";SE;"SEKTORER."
160 PRINT"ADRESSREGISTRET  INNEHåLLER";KA;"NAMN.":PRINT
161 PRINT"PROGRAMMET ANVÄNDER TVå NAMN PER SEKTOR.":PRINT
162 IF TR=640 THENPRINTCHR$(7):PRINT"OBS! DATAFILEN SLUT! ANTAL RECORDS = 640!":GOTO520
164 IF SE<=TR/2+10 THENPRINTCHR$(7):PRINT"OBS! DATAFILEN BöR EVENTUELLT UTöKAS!":PRINT
165 PRINT"öNSKAS DATA-FILEN UTöKAD?  J/N  ";:INPUTLINEAB$
170 IF LEFT$(AB$,1)="J" THEN 190
175 IF LEFT$(AB$,1)="N" THENPRINTCHR$(12):GOTO520
180 IFLEFT$(AB$,1)<>"J"ORLEFT$(AB$,1)<>"N" THENPRINTCHR$(7):GOTO165
190 PRINT:INPUT"HUR MåNGA SEKTORER SKALL DATA-FILEN UTöKAS MED ";Z4:Z5=Z4+SE
193 IF Z5>320 THEN Z5=320
195 E=Z5:PUT£1,RECORD E:SE=Z5+6:UT=0
200 S=1:GOSUB120:GET£1,RECORD2:LSETRB$=MID$(STR$(SE),2,LEN(STR$(SE))-2):PUT£1,RECORD2
210 PRINTCHR$(12):GOTO520
300 IF ERR<>4 THEN ON ERROR GOTO
305 CLOSE1:OPEN DA$ AS1
310 PUT£1,RECORD4:SE=10
315 S=1:GOSUB120:GET£1,RECORD2:LSETRB$=MID$(STR$(SE),2,LEN(STR$(SE))-2):PUT£1,RECORD2
320 RESUME 140
400 PRINTCHR$(12):PRINTCHR$(7):PRINT"REGISTRET ÄR TOMT! NR";H;"GÅR DÅ EJ ATT ANVÄNDA!"
520 REM * *  MENY  * *
522 PRINT:PRINT"REGISTRET INNEHåLLER";KA;"KUNDER."
525 PRINT:PRINT
530 ON ERROR GOTO 11000
540 PRINT TAB(5);"VÄLJ:"
550 PRINT TAB(5);"1.  EN KUND I TAGET      (SKÄRM)"
560 PRINT TAB(5);"2.  LISTNING, VERTIKALT  (SKÄRM)"
570 PRINT TAB(5);"3.  INMATNING AV NYA NAMN"
580 PRINT TAB(5);"4.  BORTTAGNING"
590 PRINT TAB(5);"5.  ÄNDRING"
600 PRINT TAB(5);"6.  TILL DELPROGRAM MED DIV LISTNINGAR  (PRINTER ELL SKÄRM)"
610 PRINT TAB(5);"7.  SLUT":PRINT
630 INPUT "VÄLJ NR";H
632 IF H>0 AND H<3 THEN IF KA=0 THEN 400
633 IF H>3 AND H<7 THEN IF KA=0 THEN 400
635 IF H<1 OR H>7 THEN PRINT CHR$(7):PRINT"FEL INLSLAGET!":GOTO 630
640 ON H GOTO 1000,2000,3000,4000,5000,6000,10000
```

```
1000 PRINTCHR$(12):PRINT"* * EN KUND I TAGET * *":PRINT
1080 ON ERROR GOTO 11000
1085 PRINT:INPUT"ANGE KUNDNUMRET ";A
1086 IF A<=1000 THENPRINTCHR$(7):PRINT"OBS! KUNDNUMRET SKALL VARA öVER 1000!":GOTO1085
1090 PRINT
1100 Z5=A-1000
1105 IF Z5>TR THENPRINTCHR$(7):PRINT"KUNDNUMMER";A;"FINNS EJ!":GOTO520
1110 IF INT(Z5/2)-(Z5/2)<>0 THEN S=1:Z5=(Z5+1)/2:E=Z5:GOTO1140
1120 S=2:Z5=Z5/2:E=Z5
1140 GOSUB120:GET£1,RECORDE
1150 IF B$="" THENPRINTCHR$(7):PRINT"NR";A;"ÄR BORTTAGEN!":GOTO520
1155 PRINT A$
1160 PRINTB$:PRINTC$:PRINTD$:PRINTE$:PRINTG$:GOTO520
2000 PRINTCHR$(12):PRINT"* * LISTNING (VERTIKALT) * *":PRINT
2040 PRINT CHR$(29);CHR$(29);CHR$(17)
2300 REM DISK TILL MINNE
2310 C=0:Q=0:B=0:Y=0:Z5=1:E=Z5:S=1
2330 D=25:REM ANTAL NAMN
2335 Q=Q+D
2340 IF Q>KA THEN Q1=Q-KA:D=D-Q1
2345 FOR B=1 TO D
2350 GOSUB120:GET£1,RECORDE
2355 IF S=1 THEN S=2:GOTO2360
2356 S=1:Z5=Z5+1:E=Z5
2360 IF B$="" THEN2350
2365 Y=B
2370 A1$(Y)=A$:B1$(Y)=B$:C1$(Y)=C$:D1$(Y)=D$:E1$(Y)=E$:G1$(Y)=G$
2390 NEXT B
2400 REM NU TILL SKÄRM
2450 FOR Y=1 TO D
2455 C=C+1
2460 PRINTA1$(Y):PRINTB1$(Y):PRINTC1$(Y):PRINTD1$(Y):PRINTE1$(Y):PRINTG1$(Y)
2465 PRINT
2480 NEXT Y
2700 IF C=KA THEN PRINTCHR$(29);CHR$(29);CHR$(31):GOTO 520
2710 GOTO 2330
3000 REM * INMATNING *
3002 IF BO>0 THEN 3010
3005 IF TR=640 THEN 162
3007 IF UT=1 THEN 3212
3010 PRINT CHR$(12):A=1000:C=0
3020 IF TR=0 THEN S=1:Z5=1:E=Z5:A=A+1:GOTO3100
3022 IF BO>0 THEN 3030
3025 IF INT(TR/2)-(TR/2)<>0 THEN S=2:Z5=(TR+1)/2:E=Z5:A=1000+TR+1:GOTO3100
3026 S=1:Z5=TR/2+1:E=Z5:A=1000+TR+1:GOTO3100
3030 Z5=0:S=1
3032 Z5=Z5+1:E=Z5
3033 PRINT:PRINT"LETAR EFTER BORTTAGNING(AR)!"
3035 GOSUB120:A=A+1
3040 GET £1,RECORD E
3045 IF B$<>"" THEN IF S=1 THEN S=2:GOTO3035
3046 IF B$<>"" THEN S=1:Z5=Z5+1:E=Z5:GOTO3035
3050 BO=BO-1:GOSUB 3300
3062 PRINT:PRINT"NU FöRS 1 NAMN öVER TILL DISK (F.D. VAKANT NUMMER)!"
3065 LSETB$=BB$:LSETC$=CC$:LSETD$=DD$:LSETE$=EE$:LSETG$=GG$:LSETRB$=""
3070 PUT£1,RECORDE:KA=KA+1
3075 IF AB$="S" THEN 3200
3080 IF BO>0 THEN IF S=1 THEN S=2:GOTO3033
3085 IF BO>0 THEN S=1:GOTO3032
3090 GOTO3025
3100 REM TILL MINNE 20 NAMN
3102 IF TR/2+5>SE THEN 3200
3105 FOR I=1 TO 20
3115 C=C+1:GOSUB 3300
3120 AA$=MID$(STR$(A),2,LEN(STR$(A))-2)
```

```
3125 A1$(I)=AA$:B1$(I)=BB$:C1$(I)=CC$:D1$(I)=DD$:E1$(I)=EE$:G1$(I)=GG$
3130 KA=KA+1:TR=TR+1
3132 IF TR=640 THENPRINTCHR$(7):PRINT"OBS! DATAFILEN SLUT! ANTAL RECORDS = 640!":GOTO3150
3133 IF TR/2+4=SE THEN UT=1:GOTO3150
3135 IF AB$="S" THEN 3150
3140 A=A+1
3145 NEXT I
3150 PRINT:PRINT"NU FöRS";C;"NAMN öVER TILL DISK!"
3160 FOR I=1 TO C
3161 FIELD£1,(S-1)*126ASZ$,4ASA$,30ASB$,20ASC$,30ASD$,25ASE$,13ASG$,3ASRB$,1ASCR$
3165 GET £1,RECORD E
3170 LSETA$=A1$(I):LSETB$=B1$(I):LSETC$=C1$(I):LSETD$=D1$(I):LSETE$=E1$(I):LSETG$=G1$(I):LSETRB$="":
LSETCR$=CHR$(13)
3175 PUT £1,RECORD E
3180 IF S=1 THEN S=2:GOTO3185
3182 S=1:Z5=Z5+1:E=Z5
3185 NEXT I
3200 S=1:GOSUB120:GET£1,RECORD1:LSETRB$=MID$(STR$(TR),2,LEN(STR$(TR))-2):PUT£1,RECORD1
3205 S=2:GOSUB120:GET£1,RECORD1:LSETRB$=MID$(STR$(BO),2,LEN(STR$(BO))-2):PUT£1,RECORD1
3210 S=1:GOSUB120:GET£1,RECORD2:LSETRB$=MID$(STR$(SE),2,LEN(STR$(SE))-2):PUT£1,RECORD2
3211 IF BO>0 THEN 520
3212 IF UT=1 THENPRINTCHR$(7):PRINT"OBS! DATAFILEN SLUT! MÄSTE UTöKAS INNAN NY INMATNING KAN SKE!":P
RINT:GOTO165
3220 IF LEFT$(AB$,1)="S" THEN 520 ELSE GOTO 3000
3300 PRINTCHR$(12):PRINTTAB(30);"* * INMATNING * *":PRINT
3302 PRINT"NR";A
3305 PRINT"MAX 30 TECKEN!  EFTER- O FöRNAMN ELL FöRETAG /                    /"
3310 PRINT"MAX 20 TECKEN!  C/O (SKRIV ÄV C/O)..... /                /"
3315 PRINT"MAX 30 TECKEN!  GATA ELL BOX (SKRIV ÄV BOX). /                    /"
3320 PRINT"MAX 25 TECKEN!  POSTADRESS..... /                /"
3325 PRINT"MAX 13 TECKEN!  TEL NR..... /            /"
3330 PRINTCHR$(5);:FORW=1TO5:PRINTCHR$(1);:NEXTW
3335 FORW=1TO46:PRINTCHR$(9);:NEXTW:PRINTCHR$(21);:INPUT£0,BB$:PRINTCHR$(5);
3337 PRINTCHR$(2);:FORW=1TO46:PRINTCHR$(9);:NEXTW:PRINTCHR$(21);:INPUTLINE£0,CC$:PRINTCHR$(5);
3340 PRINTCHR$(2);:FORW=1TO46:PRINTCHR$(9);:NEXTW:PRINTCHR$(21);:INPUTLINE£0,DD$:PRINTCHR$(5);
3350 PRINTCHR$(2);:FORW=1TO46:PRINTCHR$(9);:NEXTW:PRINTCHR$(21);:INPUTLINE£0,EE$:PRINTCHR$(5);
3360 PRINTCHR$(2);:FORW=1TO46:PRINTCHR$(9);:NEXTW:PRINTCHR$(21);:INPUTLINE£0,GG$
3370 PRINT:PRINT:PRINT"TRYCK RETURN FöR FORTSATT INMATNING!  ";
3375 PRINT"SKRIV 'S' FöR SLUT! ";:INPUTLINE£0,AB$:PRINT:RETURN
4000 PRINTCHR$(12):PRINT"* * BORTTAGNING * *":PRINT
4025 ON ERROR GOTO 11000
4030 INPUT"ANGE MED KUNDNUMMER VILKEN KUND SOM SKALL TAS BORT";A
4032 IF A<=1000 THENPRINTCHR$(7):PRINT"OBS! KUNDNUMRET SKALL VARA öVER 1000!":PRINT:GOTO4030
4035 PRINT
4095 Z5=A-1000
4096 IF Z5>TR THENPRINTCHR$(7):PRINT"KUNDNUMMER";A;"FINNS EJ!":GOTO520
4097 IF INT(Z5/2)-(Z5/2)<>0 THEN S=1:Z5=(Z5+1)/2:E=Z5:GOTO4105
4098 S=2:Z5=Z5/2:E=Z5
4105 GOSUB120:GET£1,RECORDE
4107 IF B$="" THENPRINTCHR$(7):PRINT"KUNDNUMMER";A;"ÄR REDAN BORTTAGEN!":PRINT:GOTO4210
4115 PRINT B$;"  -TAS NU BORT UR REGISTRET!"
4116 KA=KA-1:BO=BO+1
4120 LSETB$="":LSETC$="":LSETD$="":LSETE$="":LSETG$=""
4125 PUT £1,RECORD E
4130 S=2:GOSUB120:GET£1,RECORD1:LSETRB$=MID$(STR$(BO),2,LEN(STR$(BO))-2):PUT£1,RECORD1
4200 PRINT:PRINT"KLART!":PRINT
4205 IF KA=0 THEN 520
4210 INPUT"öNSKAR DU GöRA EN TILL BORTTAGNING  J/N  ";AB$
4220 IF LEFT$(AB$,1)="J" THENPRINT:GOTO4025
4230 GOTO 520
5000 PRINTCHR$(12):PRINT"* * ÄNDRING * *":PRINT
5010 ON ERROR GOTO 11000
5020 INPUT"ANGE KUNDNUMRET";A
5025 IF A<=1000 THENPRINTCHR$(7):PRINT"OBS! KUNDNUMRET SKALL VARA öVER 1000!":GOTO5020
5030 PRINT
```

```
5050 Z5=A-1000
5060 IF Z5>TR THENPRINTCHR$(7):PRINT"KUNDNUMMER";A;"FINNS EJ!":GOTO520
5065 IF INT(Z5/2)-(Z5/2)<>0 THEN S=1:Z5=(Z5+1)/2:E=Z5:GOTO5070
5066 S=2:Z5=Z5/2:E=Z5
5070 GOSUB120:GET£1,RECORDE
5075 IF B$="" THENPRINTCHR$(7):PRINT"KUNDNUMMER";A;"ÄR BORTTAGEN!":GOTO520
5083 PRINT
5090 PRINT "1.  ";B$
5100 PRINT "2.  ";C$
5110 PRINT "3.  ";D$
5120 PRINT "4.  ";E$
5125 PRINT "5.  ";G$:PRINT
5140 INPUT "VILKET NUMMER SKALL ÄNDRAS ";G:PRINT
5145 IF G<1 OR G>5 THEN PRINTCHR$(7):PRINT"FEL INSLAGET!":GOTO5140
5150 IF G=1 THEN GOSUB 5800
5160 IF G=2 THEN GOSUB 5820
5170 IF G=3 THEN GOSUB 5840
5180 IF G=4 THEN GOSUB 5860
5185 IF G=5 THEN GOSUB 5875
5210 PRINT:PRINT"ÖNSKAR DU GÖRA NÅGON MER ÄNDRING I KUNDNUMMER";A;" J/N ";:INPUTAB$
5220 IF LEFT$(AB$,1)="J" THEN 5083
5230 PUT £1, RECORD E
5240 GOTO 520
5800 PRINT B$;"  -ÄNDRAS TILL /                                    /";
5810 FORW=1TO31:PRINTCHR$(4);:NEXTW:INPUT£0,BB$:PRINT:LSETB$=BB$:RETURN
5820 PRINT C$;"  -ÄNDRAS TILL /                              /";
5830 FORW=1TO21:PRINTCHR$(4);:NEXTW:INPUTLINE£0,CC$:PRINT:LSETC$=CC$:RETURN
5840 PRINT D$;"  -ÄNDRAS TILL /                                    /";
5850 FORW=1TO31:PRINTCHR$(4);:NEXTW:INPUTLINE£0,DD$:PRINT:LSETD$=DD$:RETURN
5860 PRINT E$;"  -ÄNDRAS TILL /                                 /";
5870 FORW=1TO26:PRINTCHR$(4);:NEXTW:INPUTLINE£0,EE$:PRINT:LSETE$=EE$:RETURN
5875 PRINT G$;"  -ÄNDRAS TILL /                          /";
5876 FORW=1TO14:PRINTCHR$(4);:NEXTW:INPUTLINE£0,GG$:PRINT:LSETG$=GG$:RETURN
6000 IF KA=0 THEN 400 ELSE CLOSE 1:CHAIN "DEL-2.BAS"
9999 REM * DIV SUBRUTINER *
10000 PRINT:PRINT "SLUT !":CLOSE 1
10010 PRINT CHR$(29);CHR$(29);CHR$(17):END
11000 REM * OM EN BOKSTAV SKRIVS I EN NUMERISK VARIABEL *
11005 IF ERR<>30 THEN ON ERROR GOTO
11010 PRINT CHR$(7):PRINT"FEL INSLAGET!"
11015 RESUME
```

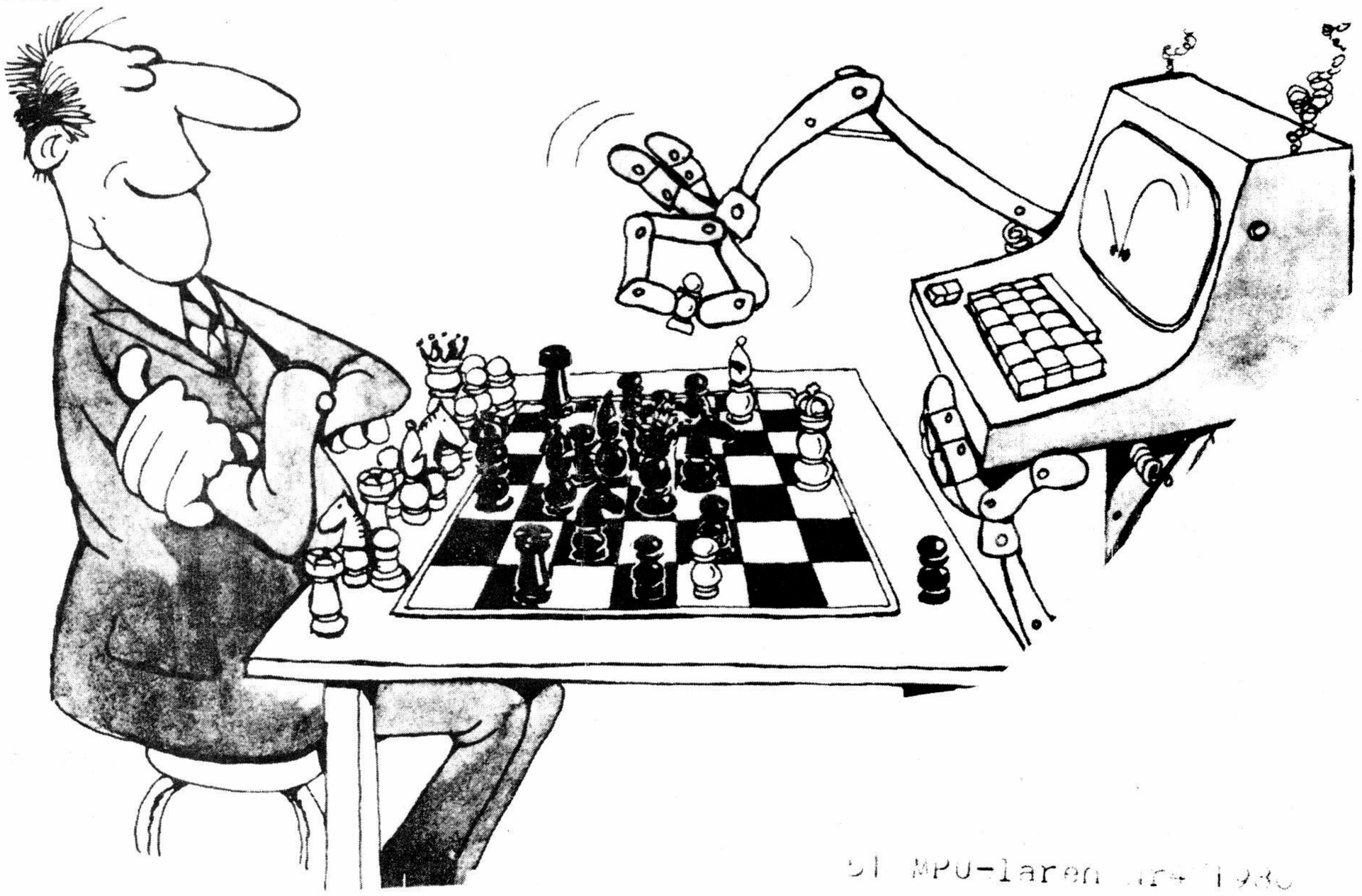


```
23 OCTOBER, 1980  TIME: 1804 39

10 REM ************************************
15 REM *      ADRESS-REGISTER             *
16 REM *      FÖR MAX 640 NAMN            *
18 REM *      2 NAMN PER SEKTOR           *
20 REM * AV ÅKE STENÄNG 80-07-15 FÖR      *
25 REM * TSC BASIC, MINIFLOPPY, CT-82     *
30 REM ************************************
31 REM *       DEL-2.BAS PROGRAM          *
32 REM ************************************
35 PRINT CHR$(28);CHR$(18)
40 PRINT CHR$(29);CHR$(29);CHR$(31)
45 PRINT CHR$(30);CHR$(30)
50 POKE HEX("AC03"),0
55 REM DATUMSTRÄNG ÅÅ-MM-DD
60 YY=PEEK(HEX("AC10")):MM=PEEK(HEX("AC0E")):DD=PEEK(HEX("AC0F"))
62 YY$=MID$(STR$(YY),2,LEN(STR$(YY))-2)
64 IF LEN(YY$)=1 THEN YY$="0"+YY$
66 MM$=MID$(STR$(MM),2,LEN(STR$(MM))-2)
68 IF LEN(MM$)=1 THEN MM$="0"+MM$
70 DD$=MID$(STR$(DD),2,LEN(STR$(DD))-2)
72 IF LEN(DD$)=1 THEN DD$="0"+DD$
75 SS$=YY$+"-"+MM$+"-"+DD$: REM DATUMSTRÄNG=SS$
80 DIM A1$(30),B1$(30),C1$(30),D1$(30),E1$(30),G1$(30)
85 DA$="REG-2":    REM DATAFILENS NAMN
86 FA$="MPU-LAREN":REM FIRMANS NAMN
87 AN$="Å. STENÄNG, BANKERYD":AN=LEN(AN$):REM ANVÄNDARENS NAMN ELL FIRMA
90 ON ERROR GOTO 300
100 OPEN OLD DA$ AS1:GOTO135
120 FIELD £1,(S-1)*126ASZ$,4ASA$,30ASB$,20ASC$,30ASD$,25ASE$,13ASG$,3ASRB$,1ASCR$:RETURN
135 PRINT"* * ADRESSREGISTER (MAX 640 NAMN) FÖR ";FA$;" * *":PRINT:PRINT
150 PRINT"* * DEL - PROGRAM * *"
155 S=1:GOSUB120:GET£1,RECORD1:TR=VAL(RB$)
160 S=2:GOSUB120:GET£1,RECORD1:BO=VAL(RB$):KA=TR-BO:GOTO520
300 IF ERR<>4 THEN ON ERROR GOTO
310 PRINTCHR$(7):PRINT"DET FINNS INGEN DATAFIL! GÅ TILL HUVUDPROGRAMMET!"
320 RESUME 525
520 REM * *  MENY  * *
522 PRINT:PRINT"REGISTRET INNEHÅLLER";KA;"KUNDER."
525 PRINT:PRINT
530 ON ERROR GOTO 11000
540 PRINT TAB(5);"VÄLJ:  (ALLA PRINTER ELL SKÄRM)"
550 PRINT TAB(5);"1.  EN KUND I TAGET"
560 PRINT TAB(5);"2.  LISTNING, VERTIKALT"
570 PRINT TAB(5);"3.  LISTNING, ETT NAMN/TVÅ RADER + F:A O SIDNUMRERING"
580 PRINT TAB(5);"4.  --"
590 PRINT TAB(5);"5.  --"
600 PRINT TAB(5);"6.  TILL HUVUDPROGRAM"
610 PRINT TAB(5);"7.  SLUT":PRINT
630 INPUT "VÄLJ NR";H
632 IF KA=0 AND H<4 THENPRINTCHR$(12):PRINTCHR$(7):PRINT"REGISTRET ÄR TOMT! NR";H;"GÅR DÅ EJ ATT ANV
ÄNDA!":GOTO520
635 IF H<1 OR H>7 THEN PRINT CHR$(7):PRINT"FEL INLSLAGET!":GOTO 630
640 ON H GOTO 1000,2000,3000,4000,5000,6000,10000
1000 PRINTCHR$(12):PRINT"* * EN KUND I TAGET * *":PRINT
1005 ON ERROR GOTO 11000
1020 PRINT "ANGE:   1.  SKÄRM"
1030 PRINT "        2.  PRINTER"
1040 INPUT "NR";G
1045 IF G=1 THEN 1080
1050 IF G=2 THEN OPEN"0.PRINT"AS0:GOTO 1075
1070 PRINT CHR$(7):PRINT"FEL INSLAGET!":GOTO 1040
1075 PRINTCHR$(7):PRINT"ÄR PRINTERN KLAR - TRYCK RETURN";:INPUTLINEAB$
```

```
1080 ON ERROR GOTO 11000
1085 PRINT:INPUT"ANGE KUNDNUMRET ";A
1086 IF A<=1000 THENPRINTCHR$(7):PRINT"OBS! KUNDNUMRET SKALL VARA öVER 1000!":GOTO1085
1090 PRINT
1140 Z5=A-1000
1145 IF Z5>TR THENPRINTCHR$(7):PRINT"KUNDUMMER";A;"FINNS EJ!":GOTO520
1150 IF INT(Z5/2)-(Z5/2)<>0 THEN S=1:Z5=(Z5+1)/2:E=Z5:GOTO1152
1151 S=2:Z5=Z5/2:E=Z5
1152 GOSUB120:GET£1,RECORDE
1153 IF B$="" THENPRINTCHR$(7):PRINT"NR";A;"ÄR BORTTAGEN!":GOTO520
1155 IFG=1THENPRINTA$
1160 PRINT£0,B$:PRINT£0,C$:PRINT£0,D$:PRINT£0,E$
1190 IFG=1 THENPRINT G$
1210 IF G=2 THEN CLOSE 0
1220 GOTO 520
2000 PRINTCHR$(12):PRINT"* * LISTNING VERTIKALT * *":PRINT
2020 ON ERROR GOTO 11000
2040 PRINT "ANGE:   1. SKÄRM"
2050 PRINT "        2. PRINTER"
2060 INPUT "NR";G
2070 IF G=1 THENPRINT:PRINTCHR$(29);CHR$(29);CHR$(17):GOTO2300
2080 IF G=2 THEN OPEN"0.PRINT"AS0:GOTO 2100
2090 PRINT CHR$(7):PRINT"FEL INSLAGET!":GOTO 2060
2100 REM FöR OLIKA VAL VID PRINTER
2110 PRINTCHR$(7):PRINT"ÄR PRINTERN KLAR - TRYCK RETURN";:INPUTLINEAB$
2300 REM DISK TILL MINNE
2310 C=0:Q=0:B=0:Y=0:Z5=1:E=Z5:S=1
2330 D=30:REM ANTAL NAMN
2335 Q=Q+D
2340 IF Q>KA THEN Q1=Q-KA:D=D-Q1
2345 FOR B=1 TO D
2350 GOSUB120:GET£1,RECORDE
2355 IF S=1 THEN S=2:GOTO2360
2356 S=1:Z5=Z5+1:E=Z5
2360 IF B$="" THEN2350
2365 Y=B
2370 A1$(Y)=A$:B1$(Y)=B$:C1$(Y)=C$:D1$(Y)=D$:E1$(Y)=E$:G1$(Y)=G$
2390 NEXT B
2400 REM TILL SKÄRM ELL PRINTER
2450 FOR Y=1 TO D
2455 C=C+1
2457 IF G=1THENPRINTA1$(Y)
2460 IF G=1THENPRINTB1$(Y):PRINTC1$(Y)
2550 IF G=2THENPRINT£0,B1$(Y);TAB(35);A1$(Y):PRINT£0,C1$(Y)
2570 PRINT£0,D1$(Y):PRINT£0,E1$(Y)
2572 IFG=1 THENPRINT G1$(Y)
2575 PRINT£0:IFG=2 THEN PRINT£0
2670 NEXT Y
2700 IF C=KA THEN 2880
2710 GOTO 2330
2880 IF G=2 THEN CLOSE 0
2882 IF G=1 THENPRINTCHR$(29);CHR$(29);CHR$(31)
2885 GOTO 520
3000 PRINTCHR$(12):PRINT"* * LISTNING, ETT NAMN/TVå RADER + F:A 0 SIDNUMRERING * *":PRINT
3020 ON ERROR GOTO 11000
3040 PRINT"ANGE:   1. SKÄRM"
3050 PRINT"        2. PRINTER"
3060 INPUT"NR";G
3070 IF G=1 THENPRINTCHR$(29);CHR$(29);CHR$(17):GOTO3300
3080 IF G=2 THEN OPEN"0.PRINT"AS0:GOTO3100
3090 PRINT CHR$(7):PRINT"FEL INSLAGET!":GOTO3060
3100 REM PRINTER HÄR
3110 PRINT:PRINTCHR$(7):PRINT"STÄLL IN PRINTERN PÅ 14 TECKEN/TUM - TRYCK RETURN";:INPUTLINEAB$
3300 REM DISK TILL MINNE
3310 C=0:Q=0:Q1=0:B=0:Y=0:Z5=1:E=Z5:S=1:SI=0
```

```
3315 T1=88:T3=31:T4=62:T5=88:REM TAB I UTMATNING
3330 D=30:REM ANTAL NAMN TILL MINNE
3335 Q=Q+D
3340 IF Q>KA THEN Q1=Q-KA:D=D-Q1
3345 FOR B=1 TO D
3350 GOSUB120:GET£1,RECORDE
3355 IF S=1 THEN S=2:GOTO3360
3356 S=1:Z5=Z5+1:E=Z5
3360 IF B$="" THEN 3350
3365 Y=B
3370 A1$(Y)=A$:B1$(Y)=B$:C1$(Y)=C$:D1$(Y)=D$:E1$(Y)=E$:G1$(Y)=G$
3390 NEXT B
3400 REM TILL PRINTER ELL SKäRM
3402 SI=SI+1
3404 PRINT£0,"ADRESSREGISTER FöR ";FA$;"  ";SS$;"  REGISTRET=";KA;"KUNDER I KUNDNUMMERORDNING";TAB(T
1);"SID";SI
3406 FOR U=1TO105:PRINT£0,"-";
3407 NEXT U:PRINT£0
3408 PRINT£0,"NAMN";TAB(T3);"GATA ELL BOX";TAB(T4);"POSTADRESS";TAB(T5);"C/O"
3410 PRINT£0,"KUNDNR";TAB(T3);"TEL NR"
3415 FOR U=1TO105:PRINT£0,"-";
3416 NEXT U:PRINT£0
3450 FOR Y=1 TO D
3455 C=C+1
3460 PRINT£0,B1$(Y);TAB(T3);D1$(Y);TAB(T4);E1$(Y);TAB(T5);C1$(Y)
3465 PRINT£0," ";A1$(Y);TAB(T3);G1$(Y)
3470 NEXT Y
3700 IF C=KA THEN FOR U=1 TO AN:PRINT£0,"-";:NEXTU:PRINT£0:PRINT£0,AN$:GOTO3880
3710 FOR U=1 TO 7:PRINT£0:NEXT U:GOTO 3330
3880 IF G=2 THEN CLOSE 0
3882 IF G=1 THENPRINTCHR$(29);CHR$(29);CHR$(31)
3885 GOTO 520
4000 PRINT"FINNS INGET PÅ 4000!":GOTO520
5000 PRINT"FINNS INGET PÅ 5000!":GOTO520
6000 CLOSE 1:CHAIN "HUVUD-2.BAS"
10000 PRINT:PRINT "SLUT !":CLOSE 1
10010 PRINT CHR$(29);CHR$(29);CHR$(17):END
11000 REM * OM EN BOKSTAV SKRIVS I EN NUMERISK VARIABEL *
11005 IF ERR<>30 THEN ON ERROR GOTO
11010 PRINT CHR$(7):PRINT"FEL INSLAGET!"
11015 RESUME
```



© MPU-laren



KA ANTALET LABLES I FLEX-ASSEMBLERN

Om man assemblerar stora program med FLEX 6800-assembler råkar man ut för att en del lables ej kommer med i lable-arean.Vi rakade dessutom ut för att ej få någon error-signalering för detta.

I FLEX-assemblern finns följande pointrar, som normalt räknas ut fran MEMEND i FLEX:

LBLBEG ($004B,$004C) Här börjar lablearean.

LBLEND ($004D,$004E) Slutet på lablearean = slutet på minnet(Max$77FF)

SRCBEG ($004F,$0050) Här börjar ett avsnitt av källkoden.

SRCEND ($0041,$0042) Här slutar ett avsnitt av källkoden.

Normalt delar assemblern lika mellan källkodsarea och lablearea.Detta beror troligen på att den ursprungligen är en kassettassembler, där man har behov av att få in så mycket källkod som möjligt på en gang. När man kör med disksystem blir nackdelen av liten källkodsarea att disken kommer att få jobba mycket,då man endast tar in en mindre del källkod åt gången.

Då lablearean tydligt var för liten minskade vi källkodsarean och ökade lablearean.Då rakade vi ut för att lablearean bara fylldes upp till LBLBEG + $2000,resten förblev tomt,och vårt resultat var oförändrat dåligt.

Lablelagringen i en assembler sker med något som kallas "Hashkodning" Det innebär att man räknar ut ett önskat läge i minnet för lablen. Därefter kontrollerar man om det är upptaget där, och om det är upptaget (<>00) kan man antingen "hasha om" eller söka sig till en ledig plats i närheten.

Vi letade reda på den rutin i assemblern,som skötte denna uträkning, genom att söka igenom pass 1. Hashrutinen räknade ut ett värde,varefter MSB (Most Significant Byte) maskades med $1F ($0738) och detta adderades till LBLBEG. Vi patchade in $3F,minskade källkodsarean och fick igenom vår assemblering.

CAB+RH

```
          NAM    FOBUG
*
* COPYRIGHT CÅ-ELEKTRONIK AB
*           BOX 2010
*           S-135 02 TYRESÖ SWEDEN
*
* 1980-09-15  08:20 REV B
*
*****************************************************
*
* MONITORPROGRAM FÖR MC6809
* PROGRAMMET INGÅR SOM EN DEL I ETT DATORBYGGE
* FÖR NYBÖRJARE OCH ENKLARE TILLÄMPNINGAR.
* DATORN BESTÅR AV :
* CPU-KORT CÅ-6819 MED MC6809 OCH 1 K DATAMINNE (RAM)
* MODERKORT CÅ-800 MED 2 KONTAKTER (J15 OCH J16).
* DISPLAYKORT CÅ-8006 MED TANGENTKORT
* PROGRAMMINNE FOBUG I EPROM (På KORT CÅ-6819).
*
*****************************************************
* FUNGERAR UNGEFÄR SOM MOTOROLAS D2-KIT
* MEN MED STATISK DISPLAY OCH FÖR 6809
*****************************************************
*
* TANGENTERNAS FUNKTIONER :
* 0-9 SAMT A-F HEXADECIMAL TECKENINMATNING
* A-F ÄR ÄVEN KOMMANDOTANGENTER
* A ALTER MEMORY, MINNESMANIPULERING
* B SÄTT IN EN BRYTPUNKT
* C CONTINUE, FORTSÄTT PROGRAMMET FRÅN BRYTPUNKT
* D DELETE BRKPT, TA BORT ALLA BRYTPUNKTER
* E EXECUTE PROGRAM, GÅ TILL ADRESS OCH STARTA PROGRA
* L LADDA MINNET FRÅN KASSETT S1-FORMAT
* P PUNCH MEMORY, SPELA IN DATA På KASSETT S1-FORMAT
* R DISPLAY REGISTERS
*
*
* FOBUG LIGGER I EPROM OCH ÄR POSITIONSOBEROENDE
* MED UNDANTAG AV VEKTORERNA
*
* FORDRAR RAM På ADRESS $C000
*
* DE SEX LYSDIODDISPLAYERNA ADRESSERAS SOM RAM-CELLER
* BIT 0-3 GER ETT HEXADECIMALT TAL MELLAN 0 OCH F
* BIT 4-6 ANVÄNDS EJ
* BIT 7 HÖG TÄNDER DECIMALPUNKTEN
*
* DATA FRÅN TANGENTBORDET HAR VÄRDET
* $00 - $13, BIT 4-6 MASKAS BORT
*
*
* OBS !!! DP-REGISTRET SKALL STå På
* $C0 OM SUBRUTINERNA SKALL ANVÄNDAS
* I EGNA PROGRAM !!!
*
DS1       EQU     $E0E0       MSB ADRESSINDIKATOR (VÄNSTRA
DS2       EQU     $E0E1
```

```
E0E2   DS3      EQU    $E0E2
E0E3   DS4      EQU    $E0E3      LSB ADRESS
E0E4   DS5      EQU    $E0E4      MSB DATA
E0E5   DS6      EQU    $E0E5      LSB DATAINDIKATOR
       *
E0E6   KEY      EQU    $E0E6      DATA FRÅN TANGENTBORD
E0E8   INFLG    EQU    $E0E8      BIT 7 ÄR NOLL OM TANGENT ÄR
E0EE   BELL     EQU    $E0EE      ACCESS GER PULS TILL BELLKOR
008C   SKIP2    EQU    $8C
E020   TAPE     EQU    $E020
E021   TAPED    EQU    $E021
       *
       * RAMCELLER FÖR DATALAGRING
       *
C000                  ORG    $C000
            C000 RAM  EQU    *
       *
C000        IRQV   RMB    2          IRQ-VEKTOR
C002        BEGA   RMB    2          STARTADRESS VID PUNCH MM
C004        ENDA   RMB    2          SLUTADRESS VID PUNCH MM
C006        NMIV   RMB    2          NMI VEKTOR
C008        SPEK   RMB    2          ANVÄNDARENS STACKPEKARE
C00A        CKSM   RMB    1          CHECKSUMMA
C00B        BYTECT RMB    1          BYTE RÄKNARE
C00C        XHI    RMB    2          TEMP LAGRING AV X-REGISTER
C00E        SWIVEK RMB    2          EGEN SWI-VEKTOR
C010        VFLAG  RMB    1          ANTAL INSATTA BRYTPUNKTER
C011        BPTAB  RMB    15         TABELL MED MAX 5 BRYTPUNKTER
C020        RDS1   RMB    1          RAMCELL FÖR DS1
C021        RDS2   RMB    1
C022        RDS3   RMB    1
C023        RDS4   RMB    1
C024        RDS5   RMB    1
C025        RDS6   RMB    1
C026        MCONT  RMB    1
C027        ADRESS RMB    2
C029        DSPEK  RMB    2          PEKARE FÖR AKTUELL DISPLAY
C02B        BLRAM  RMB    2          CELL FÖR BLINKANDE DISPLAY
C02D        BLFLG  RMB    1          FLAGGA FÖR BLINKNING
C02E        DPFLG  RMB    1          1=TÄND DEC.PUNKT, 0=SLÄCKT
C02F        BLTID  RMB    2          CELL FÖR BLINKHASTIGHET
C031        SWI2R  RMB    2          SWI2 ADRESS
C033        SWI3R  RMB    2          SWI3 ADRESS
C035        FIRQR  RMB    2          FAST IRQ ADRESS
C037        BUFF   RMB    10
       C07F STACK  EQU    RAM+$7F
       *
       00C0        SETDP  $C0
       *
F000                  ORG    $F000
            F000 PGM  EQU    *
       *
       * SERVICERUTIN FÖR IRQ
       *
       *
F000 6E   9F C000 IRQSER JMP    ÄIRQVÄ     ADRESS FÖR IRQ-RUTIN
       *
       * SERVICERUTIN FÖR NMI
       *
```

```
F004 6E   9F C006  NMISER  JMP     ÄNMIVÅ     ADRESS FÖR NMI-RUTIN
                   *
                   * SERVICERUTIN FÖR SWI
                   *
F008 6E   9F C00E  SWISER  JMP     ÄSWIVEKÅ   ADRESS FÖR SWI-RUTIN
                   *
                   * SERVICERUTIN FÖR SWI2
                   *
F00C 6E   9F C031  SWI2SE  JMP     ÄSWI2RÅ
                   *
                   * SERVICERUTIN FÖR SWI3
                   *
F010 6E   9F C033  SWI3SE  JMP     ÄSWI3RÅ
                   *
                   * SERVICERUTIN FÖR FIRQ
                   *
F014 6E   9F C035  FIRQSE  JMP     ÄFIRQRÅ
                   *
F018 8E   E0E0     CLEARD  LDX     #DS1       RUTIN FÖR ATT NOLLA INDIKATO
F01B C6   06       CLEAR   LDB     #6
F01D 6F   80       CL300   CLR     0,X+
F01F 5A                    DECB                6 POSITIONER
F020 26   FB               BNE     CL300
F022 39                    RTS
                   *
F023 86   03       START   LDA     #3         NOLLA ACIA FÖR KCS
F025 B7   E020             STA     TAPE
F028 10CE C07F             LDS     #STACK
F02C 86   C0               LDA     #$C0
F02E 1F   8B               TFR     A,DP
F030 30   8D 016D          LEAX    SFE,PCR
F034 9F   0E               STX     SWIVEK
F036 9F   06               STX     NMIV       SPARA FÖR EV. NMI-AVBROTT
F038 32   74               LEAS    -12,S
F03A 10DF 08               STS     SPEK
F03D 0F   2E               CLR     <DPFLG
F03F 0F   10       KILL    CLR     VFLAG
F041 10DE 08       CONTRL  LDS     SPEK
F044 86   C0               LDA     #$C0       SÄTT DP-REG PÅ $C000
F046 1F   8B               TFR     A,DP
F048 8D   CE               BSR     CLEARD
                   *
                   * ÅTERSTÄLL DISPLAYPEKAREN PÅ FÖRSTA DISPLAYEN OCH TÄ
                   * BLINKANDE DECIMALPUNKT
                   *
F04A 17   00A3             LBSR    RESPEK
F04D 8D   CC               BSR     CLEAR
F04F 8D   18               BSR     LBLINK     LÅNGSAMT BLINK
F051 8D   68               BSR     DEL200     LÄGG IN BLTID I BLRAM
F053 8D   1E       LOOP10  BSR     INCH       HÄMTA TECKEN
F055 80   0A               SUBA    #$0A       TESTA ATT GILTIGT TECKEN INM
F057 25   FA               BCS     LOOP10
F059 48                    ASLA                MULT MED 2
F05A 8D   10               BSR     SBLINK
F05C 30   8D 0349          LEAX    INTAB,PCR  HÄMTA KOMMANDOTABELLEN
F060 EC   86               LDD     A,X        VARS OFFSET LAGRAS I D
F062 30   8C FD    GETKOM  LEAX    GETKOM,PCR X PEKAR NU HÄR
F065 AD   8B               JSR     D,X        HOPPA TILL SUBRUTIN
```

```
F067 20   D8                BRA     CONTRL
                       *
                       * LÅNGSAM BLINKNING
                       *
F069 C6   27          LBLINK  LDB     #$27
F06B 8C                       FCB     SKIP2
                       *
                       * SNABB BLINKNING
                       *
F06C C6   07          SBLINK  LDB     #$07
F06E D7   2F                  STB     <BLTID
F070 D7   2D                  STB     <BLFLG
F072 39                       RTS
                       *
                       * SUBRUTIN FÖR ATT HÄMTA IN TECKEN FRÅN TANGENTBORDET
                       * DECIMALPUNKTEN PÅ DISPLAYEN VARS ADRESS FINNS I DSP
                       * BLINKAR MED EN HASTIGHET SOM BEROR AV VÄRDET I BLTI
                       *
 F073 34   14         INCH    PSHS    B,X           SPARA B-ACK
 F075 8D   3C         IN100   BSR     DELAY         BLINKRUTINEN
;?F077 0D   2D                TST     <BLFLG        SKALL PUNKTEN SLÄCKAS ELLER
 F079 27   02                 BEQ     IN250         INGEN ÄNDRING=0
 F07B 8D   16                 BSR     STODSP        LÄGG UT PÅ DISPLAYEN
                       *
 F07D B6   E0E8       IN250   LDA     INFLG         ÄR KNAPP INTRYCKT ?
 F080 2B   F3                 BMI     IN100         NEJ, GÅ RUNT IGEN
 F082 0F   2E                 CLR     <DPFLG        SLÄCK DEC.PUNKTEN
 F084 8D   0D                 BSR     STODSP
 F086 B6   E0E6               LDA     KEY           HÄMTA DATA
 F089 84   1F                 ANDA    #$1F          MASKA DE TRE HÖGSTA BITARNA
 F08B 35   94                 PULS    B,X,PC        ÅTERTAG OCH GÖR RTS
                       *
                       * RUTIN FÖR ATT LÄGGA A-ACK I RAM OCH DISPLAY
                       *
F08D 9E   29          STODS   LDX     <DSPEK        HÄMTA ADRESSEN
F08F A7   84                  STA     0,X           LÄGG I RAM
F091 20   0B                  BRA     DPOFF         OCH PÅ DISPLAY
                       *
F093 9E   29          STODSP  LDX     <DSPEK        HÄMTA ADRESSEN
F095 A6   84                  LDA     0,X           OCH TECKNET
F097 0D   2E                  TST     <DPFLG
F099 27   03                  BEQ     DPOFF
F09B 8B   80          DPON    ADDA    #$80          LÄGG TILL DECIMALPUNKTEN
F09D 8C                       FCB     SKIP2         HOPPA ÖVER NÄSTA INSTRUKTION
                       *
F09E 84   0F          DPOFF   ANDA    #$0F          INGEN DEC.PT.
F0A0 34   10                  PSHS    X
F0A2 34   02                  PSHS    A
F0A4 1F   10                  TFR     X,D           FLYTTA X TILL A+B
F0A6 C3   20C0                ADDD    #$20C0        ADRESS TILL DISPLAY
F0A9 1F   01                  TFR     D,X
F0AB 35   02                  PULS    A             ÅTERTAG A
F0AD A7   84                  STA     0,X           LÄGG UT PÅ DISPLAYEN
F0AF 0F   2D                  CLR     <BLFLG
F0B1 35   90                  PULS    X,PC
                       *
                       * SUBRUTIN FÖR BLINKFUNKTIONEN
                       *
```

```
FOB3 9E   2B       DELAY   LDX    <BLRAM     HÄMTA VÄRDET PÅ TIDSCELLEN
FOB5 30   1F               LEAX   -1,X       MINSKA
FOB7 9F   2B               STX    <BLRAM     LÄGG TILLBAKA
FOB9 26   08               BNE    DEL500     OM X EJ ÄR NOLL GÅ RTS
FOBB 9E   2F       DEL200  LDX    <BLTID     NYTT VÄRDE PÅ BLINKCELLEN
FOBD 9F   2B               STX    <BLRAM     LÄGG IN DET
FOBF 0C   2D               INC    <BLFLG     VÄXLA FLAGGAN
FOC1 03   2E               COM    <DPFLG
FOC3 39            DEL500  RTS
                   *
                   * TAG IN HEXADECIMALT TECKEN
                   *
FOC4 8D   AD       INHEX   BSR    INCH
FOC6 81   0F               CMPA   #$0F       KOLLA OM HEX
FOC8 23   F9               BLS    DEL500
FOCA 7D   E0EE             TST    BELL       GE PIIP OM EJ HEXADECIMALT
FOCD 20   F5               BRA    INHEX      OM EJ HEX SÅ FÖRSÖK IGEN
                   *
                   * BYGG ADRESS I X-REG + "ADRESS"
                   *
FOCF 8D   09       BADDR   BSR    BYTE       HÖGSTA 8 BITARNA
FOD1 97   27               STA    <ADRESS
FOD3 8D   05               BSR    BYTE       LÄGSTA 8 BITARNA
FOD5 97   28       BADDR7  STA    <ADRESS+1
FOD7 9E   27               LDX    <ADRESS
FOD9 39                    RTS
                   *
                   * TAG IN TVÅ HEX. TECKEN I A-ACK OCH LAGRA I DSPEK
                   *
FODA 8D   E8       BYTE    BSR    INHEX      IN ETT HEXADECIMALT TAL
FODC 8D   AF       BYTE2   BSR    STODS      LÄGG UT PÅ DISPLAY
FODE 8D   30               BSR    INCDSP     INCREMENTA DSPEK
FOE0 48                    ASLA
FOE1 48                    ASLA
FOE2 48                    ASLA
FOE3 48                    ASLA              HHHH 0000
FOE4 1F   89               TFR    A,B
FOE6 34   04               PSHS   B          SPARA B-ACK
FOE8 8D   DA               BSR    INHEX      TAG IN DET ANDRA TECKNET
FOEA 8D   A1               BSR    STODS      LÄGG UT PÅ DISPLAYEN
FOEC AB   E0               ADDA   0,S+       LÄGG IHOP A OCH B
FOEE 20   20               BRA    INCDSP
                   *
                   * ÅTERSTÄLL DISPLAYPEKAREN
                   *
FOF0 8E   C020     RESPEK  LDX    #RDS1
FOF3 9F   29               STX    <DSPEK
FOF5 39                    RTS
                   *
                   * LÄGG UT PÅ ADRESSDISPLAYEN
                   *
FOF6 8D   F8       UTADR   BSR    RESPEK
FOF8 30   07               LEAX   ADRESS-RDS1,X ADRESSEN TILL DET VÄRDE
FOFA 8D   03               BSR    OUT2H       MSB BYTE.. SOM SKALL UT
FOFC 8E   C028             LDX    #ADRESS+1
                   *
                   * MATA UT 2 HEX TECKEN
                   *
```

```
F0FF A6    84        OUT2H   LDA     0,X         HÄMTA EN BYTE
F101 34    02        OUT2H2  PSHS    A
F103 44              LSRA
F104 44              LSRA
F105 44              LSRA
F106 44              LSRA
F107 8D    84        BSR     STODS       LÄGG UT DET FÖRSTA TECKNET
F109 35    02        PULS    A
F10B 8D    03        BSR     INCDSP
F10D 17    FF7D      LBSR    STODS       LÄGG UT DET ANDRA TECKNET
F110 0C    2A        INCDSP  INC     <DSPEK+1    PEKA PÅ NÄSTA
F112 39              INHEX5  RTS
                     *
                     * MINNESMANIPULERING
                     *
F113 8D    BA        MEMCHG  BSR     BADDR       BEGÄR FÖRSTA ADRESS
F115 8D    DF        MEM10   BSR     UTADR       LÄGG UT ADRESS PÅ DISPLAY
F117 9E    27              LDX     <ADRESS
F119 8D    62              BSR     DATAIX      LÄGG UT DATA PÅ DISPLAYEN
F11B 17    FF55      MEM15   LBSR    INCH        HÄMTA TECKEN
F11E 81    12              CMPA    #$12        OM RETURN GÅ RTS "R"
F120 27    F0              BEQ     INHEX5      GÅ TILL RTS
F122 81    11        MEM20   CMPA    #$11        FÖREGÅENDE ? "P"
F124 27    39              BEQ     MEM88
F126 81    10              CMPA    #$10        NÄSTA ? "L"
F128 27    3D              BEQ     MEM94
F12A 81    13              CMPA    #$13        OFFSETBERÄKNING ? "S"
F12C 26    22              BNE     MEM60
F12E 9E    27              LDX     <ADRESS
F130 9F    02              STX     <BEGA       SPARA MINNESPOSITIONEN
F132 8D    BC              BSR     RESPEK
F134 8D    99              BSR     BADDR       BEGÄR IN HOPPADRESSEN
F136 30    1F              LEAX    -1,X        KORRIGERA FÖR OFFSET
F138 9F    04              STX     <ENDA       SPARA
F13A 8D    BA              BSR     UTADR       LÄGG UT PÅ DISPLAYEN
F13C 17    0143            LBSR    CMPADR      JÄMFÖR ADRESSERNA
F13F 9E    02              LDX     <BEGA
F141 9F    27              STX     <ADRESS
                     *
                     * OM B=00 INOM RACKHALL FRAMÅT OM A<80
                     * OM B=FF INOM RACKHALL BAKÅT OM A>7F
                     *
F143 4D              TSTA                    FRAM/BAKÅT
F144 2B    04              BMI     OK          HOPPA OM BAKÅT
F146 5D              TSTB                    =0 OM FRAMÅT
F147 26    11              BNE     MEMFEL      PIIP OM FÖR STOR OFFSET
F149 5A              DECB                    B=FF NU
F14A 5C              OK      INCB
F14B 26    0D              BNE     MEMFEL      FEL I OFFSET BAKÅT
F14D 8D    30              BSR     DATADS      SKRIV UT OFFSET PÅ DISPLAYEN
F14F 8C              FCB     SKIP2       HOPPA ÖVER NÄSTA INSTRUKTION
                     *
F150 8D    8A        MEM60   BSR     BYTE2       NYTT DATA
F152 9E    27              LDX     <ADRESS
F154 A7    84        MEM62   STA     0,X         LAGRA I RAM
F156 A1    84              CMPA    0,X         FASTNADE DET ?
F158 27    09              BEQ     MEM92       NIX
F15A 7D    E0EE      MEMFEL  TST     BELL        GE PIIP
```

```
F15D 20   0A              BRA    MEM95
                  *
                  * GÅ TILL FÖREGÅENDE ADRESS I MINNET
                  *
F15F 30   1F      MEM88   DEX
F161 20   06              BRA    MEM95
                  *
                  * STEGA TILL NÄSTA ADRESSPOSITION
                  *
F163 8D   08      MEM92   BSR    DELAYD
F165 9E   27      MEM93   LDX    <ADRESS
F167 30   01      MEM94   LEAX   1,X
F169 9F   27      MEM95   STX    <ADRESS   LAGRA NY ADRESS
F16B 20   A8              BRA    MEM10
                  *
                  * FÖRDRÖJNINGSLOOP FÖR ATT HINNA LÄSA
                  * AV ADRESS-DATA DISPLAYEN
                  *
F16D 34   14      DELAYD  PSHS   B,X        SPARA X OCH B
F16F C6   02              LDB    #2
F171 8E   AFFF    DELLDX  LDX    #$AFFF
F174 30   1F      DELDEX  LEAX   -1,X       MINSKA X-LOOPEN
F176 26   FC              BNE    DELDEX
F178 5A                   DECB
F179 26   F6              BNE    DELLDX
F17B 35   94              PULS   B,X,PC     ÅTERTAG
                  *
                  * LÄGG UT A-ACK PÅ DATADISPLAYEN
                  *
F17D A6   84      DATAIX  LDA    0,X        HÄMTA INDEXERAT
F17F 9F   02      DATADS  STX    <BEGA      SPARA X
F181 34   06              PSHS   B,A        SPARA B OCH A
F183 17   FF7B            LBSR   OUT2H2     LÄGG UT
F186 35   06              PULS   B,A
F188 8E   C024    DEC2DS  LDX    #RDS5
F18B 9F   29              STX    <DSPEK
F18D 9E   02              LDX    <BEGA      ÅTERTAG X
F18F 39                   RTS
                  *
                  *
                  * GÅ TILL ADRESS OCH STARTA PROGRAM
                  *
F190 17   FF3C    GOEXEC  LBSR   BADDR      HÄMTA IN EN ADRESS
F193 DE   08              LDU    <SPEK      LÄGG IN PC-VÄRDET
F195 AF   4A              STX    10,U       PÅ STACKEN
F197 6F   43              CLR    3,U        NOLLA DP-REG
                  *
                  * CONTINUE
                  *
F199 17   00CB    CONTIN  LBSR   BRKPIN
F19C DE   08              LDU    <SPEK      HÄMTA USER STACKPOINTER
F19E 1F   34              TFR    U,S        LÄGG IN USER STACK
F1A0 3B           RETINT  RTI               OCH GÖR EN RTI
                  *
                  * SWI SERVICERUTIN
                  *
F1A1 10DF 08      SFE     STS    SPEK
F1A4 86   C0              LDA    #$C0
```

```
F1A6 1F    8B              TFR     A,DP
F1A8 1F    41              TSX
F1AA 6D    0B              TST     11,X        DECREMENTERA
F1AC 26    02              BNE     *+4
F1AE 6A    0A              DEC     10,X
F1B0 6A    0B              DEC     11,X
F1B2 17    008F            LBSR    BRKP40      FINNS BREAKPOINTS ?
F1B5 27    13              BEQ     PRREG       PRINTA REGISTREN
                   *
                   * TAG BORT BREAKPOINTS I FOBUG
                   *
F1B7 34    10      BRKP00  PSHS    X
F1B9 A6    02              LDA     2,X         SKYDDA MOT MULTIDEF. BRKPT
F1BB 81    3F              CMPA    #$3F
F1BD 27    06              BEQ     BRKP10
F1BF AE    84              LDX     0,X
F1C1 A7    84              STA     0,X
F1C3 35    10              PULS    X
F1C5 30    03      BRKP10  LEAX    3,X
F1C7 5A                    DECB
F1C8 26    ED              BNE     BRKP00
                   *
                   * PRESENTERA REGISTREN
                   *
F1CA 17    FE4B    PRREG   LBSR    CLEARD      NOLLA INDIKATORERNA
F1CD 8D    B9              BSR     DEC2DS      PEKA Pa DS5
F1CF 9E    08              LDX     <SPEK
F1D1 8D    AA              BSR     DATAIX      UT MED CC-REG
F1D3 8D    57              BSR     INTECK      VÄNTA IN NYTT TECKEN
F1D5 30    01              LEAX    1,X
F1D7 8D    A4              BSR     DATAIX      UT MED A-ACK
F1D9 8D    51              BSR     INTECK
F1DB 30    01              LEAX    1,X
F1DD 8D    9E              BSR     DATAIX      UT MED B-ACK
F1DF 8D    4B              BSR     INTECK
F1E1 30    01              LEAX    1,X
F1E3 8D    98              BSR     DATAIX      UT MED DP-REG
F1E5 8D    45              BSR     INTECK
F1E7 30    01              LEAX    1,X
F1E9 9F    02              STX     <BEGA
F1EB AE    84              LDX     0,X
F1ED 9F    27              STX     <ADRESS     VÄRDET Pa X I STACKEN
F1EF 17    FF04            LBSR    UTADR       UT MED X-REG
F1F2 8D    38              BSR     INTECK
F1F4 9E    02              LDX     <BEGA
F1F6 30    02              LEAX    2,X
F1F8 9F    02              STX     <BEGA
F1FA AE    84              LDX     0,X
F1FC 9F    27              STX     <ADRESS     VÄRDET Pa Y I STACKEN
F1FE 17    FEF5            LBSR    UTADR       UT MED Y-REG
F201 8D    29              BSR     INTECK
F203 9E    02              LDX     <BEGA
F205 30    02              LEAX    2,X
F207 9F    02              STX     <BEGA
F209 AE    84              LDX     0,X
F20B 9F    27              STX     <ADRESS     VÄRDET Pa U-REG.
F20D 17    FEE6            LBSR    UTADR       UT MED U-REG
F210 9E    02              LDX     <BEGA
```

```
F212 30   02                LEAX   2,X
F214 9F   02                STX    <BEGA
F216 AE   84                LDX    0,X
F218 9F   27                STX    <ADRESS    VÄRDET PÅ PC
F21A 8D   10                BSR    INTECK
F21C 17   FED7              LBSR   UTADR      UT MED PC
F21F 9E   08                LDX    <SPEK      STACKVÄRDET
F221 9F   27                STX    <ADRESS
F223 8D   07                BSR    INTECK
F225 17   FECE              LBSR   UTADR      UT MED STACKPEKAREN
F228 8D   02                BSR    INTECK
F22A 20   9E                BRA    PRREG
                      *
F22C B6   E0E8      INTECK  LDA    INFLG      KOLLA OM KNAPP TRYCKT
F22F 2B   FB                BMI    INTECK
F231 B6   E0E6              LDA    KEY        HÄMTA TECKNET
F234 84   1F                ANDA   #$1F       MASKA D5-D7
F236 81   13                CMPA   #$13       KOLLA OM S-KNAPPEN
F238 27   03                BEQ    INTRTS     JO
F23A 16   FE04              LBRA   CONTRL     HOPPA TILLBAKS
                      *
F23D 34   10        INTRTS  PSHS   X
F23F 17   FDD6              LBSR   CLEARD     NOLLA DISPLAYEN FÖRST
F242 35   90                PULS   X,PC
                      *
F244 8E   C011      BRKP40  LDX    #BPTAB
F247 D6   10                LDB    <VFLAG     ANTAL BRKP'S
F249 39             BRKRTS  RTS
                      *
                      * V-KOMMANDO SÄTT IN EN BRYTPUNKT
                      *
F24A 17   FE82      SETBR   LBSR   BADDR      FIXA ADDRESSEN
F24D 8D   F5                BSR    BRKP40     FINNS LEDIG PLATS ?
F24F 27   0D                BEQ    BRKP60     OM LEDIG SÅ STOPPA IN DEN
F251 C1   05                CMPB   #5         MAX FEM BRYTPUNKTER
F253 25   04                BCS    BRKP50     OK
F255 7D   E0EE      BELRTS  TST    BELL       GE PIIP OM FÖR MÅNGA
F258 39                     RTS
                      *
F259 30   03        BRKP50  LEAX   3,X        ÖKA X MED 3
F25B 5A                     DECB
F25C 26   FB                BNE    BRKP50
                      *
F25E 0C   10        BRKP60  INC    <VFLAG     ÖKA ANTALSRÄKNAREN
F260 DC   27                LDD    <ADRESS    ADRESS TILL MINNET
F262 ED   84                STD    0,X        IN I TABELLEN
F264 16   FF06              LBRA   DELAYD     VÄNTA EN STUND
                      *
                      * STOPPA IN ALLA BRKPT'S I MINNET VID 'G'
                      *
F267 8D   DB        BRKPIN  BSR    BRKP40
F269 27   1F        BRKPRT  BEQ    CMPRTS
                      *
F26B 9F   0C        TGB     STX    <XHI
F26D AE   84                LDX    0,X        HÄMTA ADRESSEN
F26F A6   84                LDA    0,X        HÄMTA INSTRUKTIONEN
F271 34   02                PSHS   A
F273 86   3F                LDA    #$3F       OPCODE FÖR SWI
```

```
F275 A7   84              STA    0,X        I MINNET
F277 9E   0C              LDX    <XHI
F279 35   02              PULS   A
F27B A7   02              STA    2,X        SPARAD OPCODE
F27D 30   03              LEAX   3,X
F27F 5A                   DECB
F280 20   E7              BRA    BRKPRT
                  *
                  * JÄMFÖR START/STOPPADDRESSER
                  *
F282 96   05      CMPADR  LDA    <ENDA+1
F284 90   03              SUBA   <BEGA+1
F286 D6   04              LDB    <ENDA
F288 D2   02              SBCB   <BEGA
F28A 39           CMPRTS  RTS
                  *
                  * PUNCHA TILL KASSETT I S1-FORMAT
                  * ERFORDRAR KCS-MODEM MED ACIA PÅ $E020
                  * UNDER PUNCHNING VISAS ADRESS OCH DATA
                  * PÅ INDIKATORERNA
                  *
F28B 86   11      PNCH00  LDA    #$11       SÄTT UPP ACIAN FÖR
F28D B7   E020            STA    TAPE       TVÅ STOPPBITAR VID PUNCH
F290 17   FE3C            LBSR   BADDR      BEGÄR STARTADRESS
F293 9F   02              STX    <BEGA
F295 17   FED5            LBSR   DELAYD
F298 17   FE55            LBSR   RESPEK
F29B 17   FE31            LBSR   BADDR      BEGÄR STOPPADRESS
F29E 9F   04              STX    <ENDA
F2A0 8D   E0      PUN11   BSR    CMPADR     JÄMFÖR START/STOPP
F2A2 26   04              BNE    PUN22
F2A4 81   10              CMPA   #16        MINDRE ÄN 16 BYTES KVAR ?
F2A6 25   02              BCS    PUN23      HOPPA ISÅFALL ÖVER NÄSTA INS
                  *
                  * LÄGG UT 16 MINNESBYTES PER RECORD
                  *
F2A8 86   0F      PUN22   LDA    #15
F2AA 8B   04      PUN23   ADDA   #4         INKLUDERA ADRESS I BYTERÄKNA
F2AC 97   26              STA    <MCONT     SPARA RESULTATET
F2AE 80   03              SUBA   #3
F2B0 97   37              STA    BUFF       DETTA RECORDS BYTEANTAL
                  *
                  * SÄND CR,LF,NULL OCH S1(BBAAAADDDDDDDDDDDDCC)
                  *
F2B2 30   8D 00E7         LEAX   MTAPE1,PCR STRÄNG MED DATA
F2B6 8D   34              BSR    TDATA      IVÄG MED DET
F2B8 5F                   CLRB              CHECKSUMMA
                  *
                  * SÄND BYTEANTALET (S1)BB(AAAADDDDDDDDDDDDCC)
                  *
F2B9 8E   C026            LDX    #MCONT
F2BC 8D   38              BSR    PUNT2
                  *
                  * SÄND ADRESSEN (S1BB)AAAA(DDDDDDDDDDDDCC)
                  *
F2BE 9E   02              LDX    <BEGA      HÄMTA ADRESS FÖR RECORD
F2C0 9F   27              STX    <ADRESS
F2C2 34   04              PSHS   B
```

```
F2C4 17   FE2F            LBSR   UTADR      LÄGG UT PÅ DISPLAYEN
F2C7 35   04              PULS   B
F2C9 8E   C002            LDX    #BEGA      STARTADRESS FÖR VARJE RECORD
F2CC 8D   28              BSR    PUNT2      FÖRSTA 2 HEX
F2CE 8D   26              BSR    PUNT2      ANDRA 2 HEX
                  *
                  * SÄND DATA (S1BBAAAA)DDDDDDDDDDDD(CC)
                  *
F2D0 9E   02              LDX    <BEGA      ADRESSEN TILL DATAT
F2D2 17   FEA8    PUN32   LBSR   DATAIX
F2D5 8D   1F              BSR    PUNT2
F2D7 0A   37              DEC    BUFF
F2D9 26   F7              BNE    PUN32
                  *
                  * SÄND CHECKSUMMAN (S1BBAAAADDDDDDDDDDDD)CC
                  *
F2DB 53                   COMB                 CHECKSUMMAN INVERTERAS
F2DC 1F   98              TFR    B,A
F2DE 8D   1A              BSR    OUT2HT
F2E0 9F   02              STX    <BEGA
F2E2 30   1F              LEAX   -1,X
F2E4 9C   04              CPX    <ENDA       KLAR ?
F2E6 26   B8              BNE    PUN11
F2E8 30   8D 00BA         LEAX   S9STR,PCR
                  *
F2EC A6   80      TDATA   LDA    0,X+        HÄMTA TECKEN
F2EE 81   04              CMPA   #4          SLUT ?
F2F0 27   26              BEQ    TRTS
F2F2 8D   16              BSR    TAPEUT
F2F4 20   F6              BRA    TDATA
                  *
                  * SÄND 2 HEX TAL SAMT UPPDATERA CHECKSUMMAN
                  *
F2F6 EB   84      PUNT2   ADDB   0,X         UPPDATERA CHECKSUMMAN
F2F8 A6   80              LDA    0,X+        SÄND 2 HEX TECKEN
F2FA 34   02      OUT2HT  PSHS   A
F2FC 8D   1B              BSR    OUTHL       OMVANDLA TILL ASCII
F2FE 35   02              PULS   A
F300 84   0F      OUTHR   ANDA   #$0F        MASKA BORT MSB-DELEN
F302 8B   30              ADDA   #$30        TILLVERKA ASCII
F304 81   39              CMPA   #$39
F306 23   02              BLS    TAPEUT
F308 8B   07              ADDA   #7          FÖR A-F
                  *
                  * UTRUTIN FÖR TAPE
                  *
F30A 34   02      TAPEUT  PSHS   A
F30C B6   E020    TAPUT   LDA    TAPE
F30F 47                   ASRA
F310 47                   ASRA
F311 24   F9              BCC    TAPUT       LEDIG ?
F313 35   02              PULS   A
F315 B7   E021            STA    TAPED
F318 39           TRTS    RTS
                  *
F319 44           OUTHL   LSRA
F31A 44                   LSRA
F31B 44                   LSRA
```

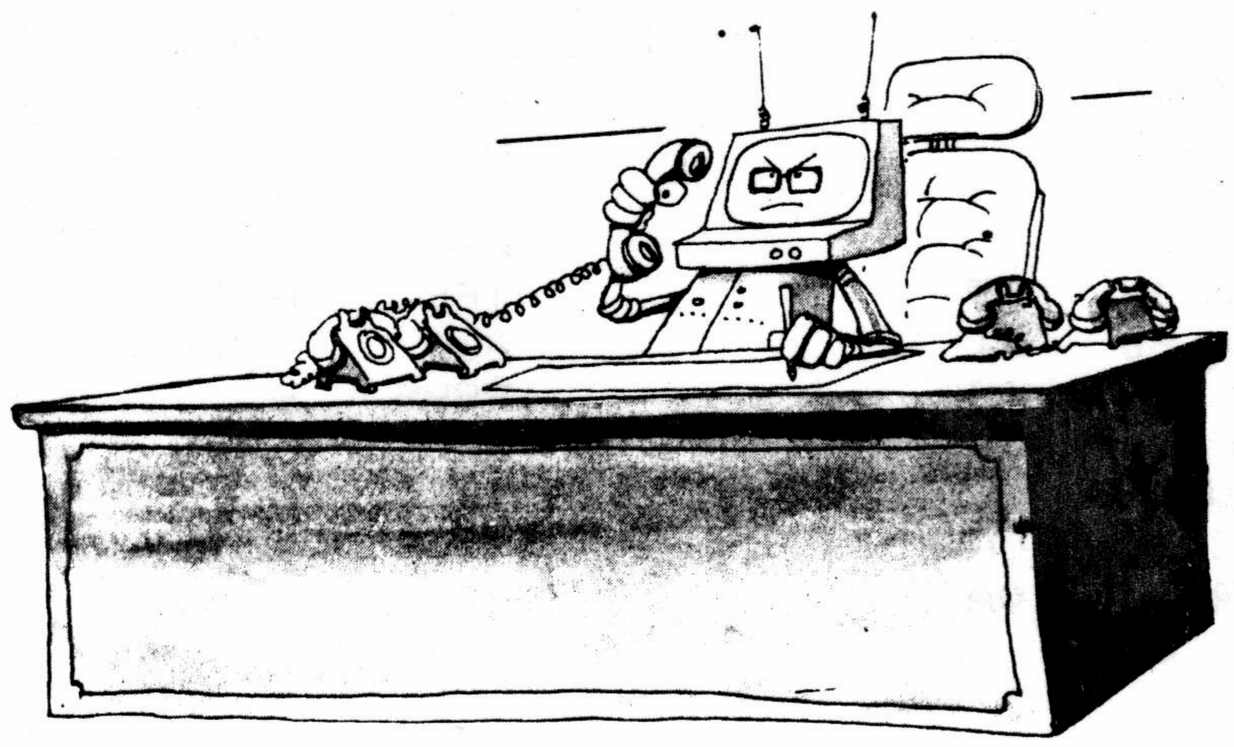
```
F31C 44                     LSRA
F31D 20   E1                BRA     OUTHR
                    *
                    * LADDA MINNET FRÅN KASSETT I S1-FORMAT
                    * UNDER LADDNINGEN VISAS MINNESADRESSEN
                    * PÅ ADRESSDISPLAYEN DS1-4
                    * OCH DATA PÅ DS5-6
                    *
F31F 86   15        LOADOO  LDA     #$15        SÄTT UPP ACIAN FÖR
F321 B7   E020              STA     TAPE        EN STOPPBIT
F324 8D   3D        LOAD3   BSR     TAPEIN      HÄMTA TECKEN
F326 81   53                CMPA    #'S         SÖK TYP AV RECORD
F328 26   FA                BNE     LOAD3
F32A 8D   37                BSR     TAPEIN
F32C 81   39                CMPA    #'9         ÄR DET EOF ?
F32E 27   E8                BEQ     TRTS
F330 81   31                CMPA    #'1         DATA RECORD
F332 26   F0                BNE     LOAD3
F334 0F   0A                CLR     <CKSM       NOLLA CHECKSUMMECELLEN
F336 8D   4A                BSR     TBYTE       LÄS IN ANTAL BYTES I RECORDE
F338 80   02                SUBA    #2          KOMMANDE ADRESS= 2 BYTE
F33A 97   0B                STA     <BYTECT
                    *
                    * BYGG ADRESSEN
                    *
F33C 8D   44                BSR     TBYTE       HÄMTA MSB
F33E 97   27                STA     <ADRESS
F340 8D   40                BSR     TBYTE       HÄMTA LSB
F342 97   28                STA     <ADRESS+1
F344 17   FDAF              LBSR    UTADR       LÄGG UT PÅ DISPLAYEN
F347 9E   27                LDX     <ADRESS     X HAR NU LADDADRESSEN
                    *
                    * LAGRA DATA
                    *
F349 8D   37        LOAD11  BSR     TBYTE       GÖR EN BYTE
F34B 0A   0B                DEC     <BYTECT     MINSKA ANTALSRÄKNAREN
F34D 27   0D                BEQ     LOAD15      OM NOLL SÅ GÅ TILL BÖRJAN
F34F 17   FE2D              LBSR    DATADS      LÄGG UT DATA PÅ DISPLAYEN
F352 A7   84                STA     0,X         LAGRA I MINNET
F354 A1   84                CMPA    0,X         FASTNADE DET
F356 26   08                BNE     LOAD19
F358 30   01                LEAX    1,X
F35A 20   ED                BRA     LOAD11
                    *
F35C 0C   0A        LOAD15  INC     <CKSM
F35E 27   C4                BEQ     LOAD3
F360 16   FEF2      LOAD19  LBRA    BELRTS
                    *
F363 B6   E020      TAPEIN  LDA     TAPE
F366 47                     ASRA
F367 24   FA                BCC     TAPEIN
F369 B6   E021              LDA     TAPED
F36C 39                     RTS
                    *
F36D 8D   F4        TINHEX  BSR     TAPEIN
F36F 80   30                SUBA    #$30
F371 2B   24                BMI     C1          EJ HEX
F373 81   09                CMPA    #$09
```

```
F375 2F   0A            BLE    IN1HG
F377 81   11            CMPA   #$11
F379 2B   1C            BMI    C1
F37B 81   16            CMPA   #$16
F37D 2E   18            BGT    C1          EJ HEX
F37F 80   07            SUBA   #7
F381 39         IN1HG   RTS
                *
F382 8D   E9    TBYTE   BSR    TINHEX
F384 48                 ASLA
F385 48                 ASLA
F386 48                 ASLA
F387 48                 ASLA
F388 1F   89            TFR    A,B
F38A 8D   E1            BSR    TINHEX
F38C 34   04            PSHS   B
F38E AB   E0            ADDA   0,S+
F390 1F   89            TFR    A,B
F392 DB   0A            ADDB   <CKSM
F394 D7   0A            STB    <CKSM
F396 39                 RTS
                *
F397 7D   E0EE  C1      TST    BELL        GE PIIP OCH ÅTERGÅ
F39A 16   FCA4          LBRA   CONTRL
                *
F39D 0D 0A 00 00 MTAPE1 FCB    $D,$A,0,0,0,0,'S,'1,4
                *
F3A6 53 39 04    S9STR  FCB    'S,'9,4
                *
                *
                * TABELL FÖR KOMMANDON
                *
F3A9 00B1       INTAB   FDB    MEMCHG-GETKOM MINNESMANIPULERING "M"
F3AB 01E8               FDB    SETBR-GETKOM
F3AD 0137               FDB    CONTIN-GETKOM
F3AF FFDD               FDB    KILL-GETKOM
F3B1 012E               FDB    GOEXEC-GETKOM
F3B3 031F               FDB    IN1HG-GETKOM RETURADRESS
F3B5 02BD               FDB    LOADOO-GETKOM
F3B7 0229               FDB    PNCHOO-GETKOM
F3B9 0168               FDB    PRREG-GETKOM
F3BB 031F               FDB    IN1HG-GETKOM
F3BD            TABEND  EQU    *
                *
                * VEKTORER FÖR START OCH INTERRUPT.
                * DESSA ÄR EJ POSITIONSOBEROENDE OCH
                * MÅSTE ÄNDRAS OM PROGRAMMET SKALL
                * FLYTTAS FRÅN $FOOO !!!!
                *
F3F2                    ORG    PGM+$3F2
F3F2 F010               FDB    SWI3SE
F3F4 F00C               FDB    SWI2SE
F3F6 F014               FDB    FIRQSE
F3F8 F000               FDB    IRQSER
F3FA F008               FDB    SWISER
F3FC F004               FDB    NMISER
F3FE F023               FDB    START
                        END
```

ORDNINGSFRAGA:
Av Carl Axel Bruno

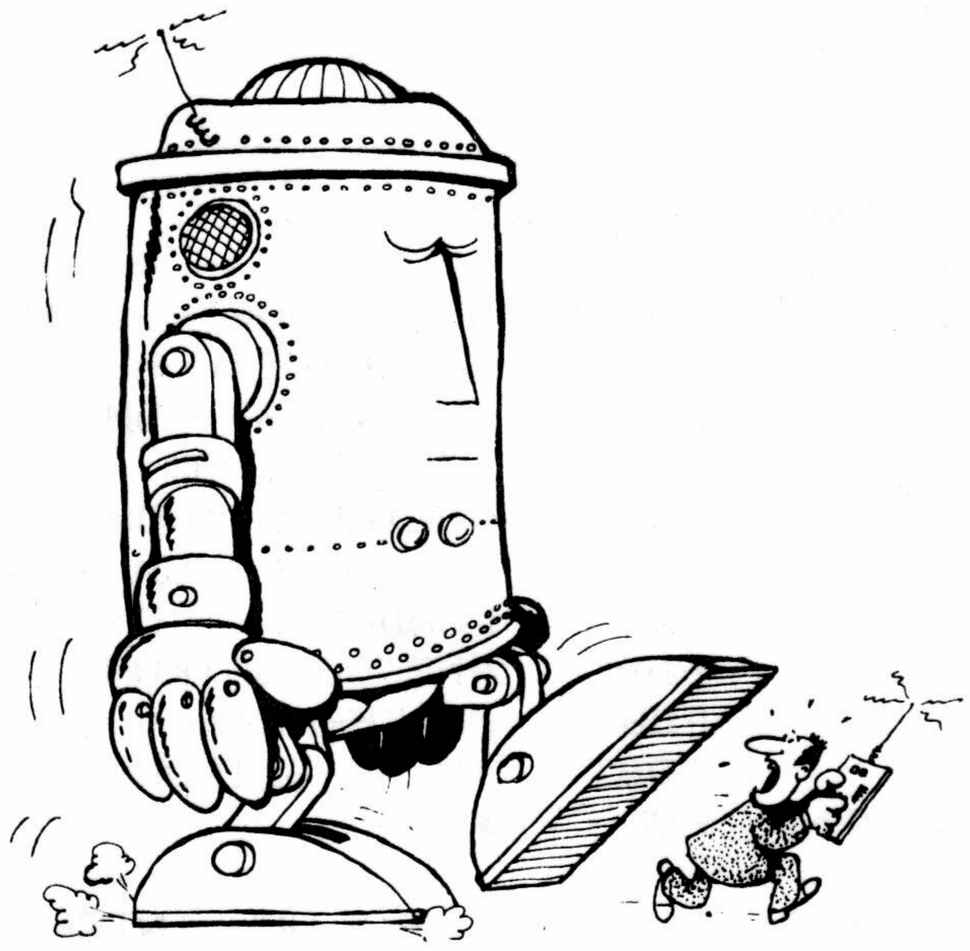
Som alla har jag svårt att hålla ordning bland mina filer på mina disketter Det är svårt att överblicka att man har tillräckligt med backupper,var en viss fil finns, och så vidare
Min lösning på detta problem är följande:
1 Jag editerade en fil med alla directoryn i slumpvis ordning,första disketten först,andra sedan osv:

FILIP1
CESAR1
ANDERS1
CESAR2
AXEL2
CESAR3
DAVID3
BERTIL3
DAVID4
CESAR4
DENNIS5
GUSTAV5
BERTIL5
ANDERS6
HARALD7
AXEL7
ANDERS8
ERIK8



2 Jag bearbetade filen med ett sorteringsprogram:

ANDERS1
ANDERS6
ANDERS8
AXEL2
AXEL7
BERTIL3
BERTIL5
CESAR1
CESAR2
CESAR3
CESAR4
DAVID3
DAVID4
DENNIS5
ERIK8
FILIP1
GUSTAV5
HARALD7



Genom att jag har lagt till en siffra efter varje namn kan jag hålla redapå vilken diskett resp fil finns på

3 Jag bearbetade filen med en textprocessor, och fick som slutresultat hela alltet på en halv A4sida Eftersom alla filer med samma innehåll låg intill varandra var det jättelätt att se om någon fil saknade backup Uppdatering sker i den osorterade filen och kan alltså lätt skötas från en färsk katalog.
ANDERS1 ANDERS6 ANDERS8 AXEL2 AXEL7 BERTIL3 BERTIL5 CESAR1 CESAR2 CESAR3CESAR4 DAVID3 DAVID4 DENNIS5 ERIK8FILIP1 GUSTAV5 HARALD7

CAB

****** NY VERSION ******

Presenterar en ny version av MORSE CODE send/recive programmet som fanns i MPU-LAREN förut men som visade sig,inte fungera så bra på RT-datorn. De flesta har kommit så långt att programmet har presenterat sig, men sen har det tagit stopp Det största problemet ligger i att terminal acian inte genererar IRQ.

Åtgärda genom att ansluta ACIA pin 7 - CPU pin 4. Då kommer acian att ge irq vid tecken från terminalen. Sedan har jag även gjort en del andra små förbättringar av mjukvaran, de viktigaste är startadressen $87FO i det här fallet, samt att även de svenska tecknen finns med.

Jag har provkört i en RT-datorn bestyckad med 16K ram, eprom-kort med morseprogrammet placerat på plats $8400, samt modiferat D2-kort enl RT. Finns det intresse att få en kopia av den nya version så är det bara att höra av sig.

Anders

```
1                       NAM     CWMK
2                 ********************************************************
3                 * MORSE CODE/ASCII SEND/RECEIVE PROGRAM
4                 * FOR THE MOTOROLA 6800 MICROPROCESSOR
5                 * WRITTEN BY:
6                 *    DON L JACKSON - W7KGU
7                 *    JIM BAINTER - WA7VKZ
8                 * TILLFIXAT AV ANDERS KARLSSON - SMOKDE
9                 * VERSION 2 5 - FOR USE ON MOTOROLA
10                *  DESIGN EVALUATION KIT (TBUG).
11                *
12                * VERSION DATE 80 04 04
13                *
14                ********************************************************
15                * I/O HARDWARE CONFIGURATION:
16                *    PIA ADDRESS - $8024
17                *       CA1 - INTERRUPT TIMING FOR XMIT CODE,
18                *             2-50 HZ
19                *       PA0 - T/R SWITCH,0=TX, 1=RCV
20                *       PA7 - RECEIVE CODE INPUT
21                *       PB0 - CODE OUT
22                *       PB1 - TRANSMIT LAMP
23                *         PB2 - RECEIVE LAMP
24                *    ACIA ADDRESS - $8008
25                ********************************************************
```

```
66  8400                    ORG     $8400
67                  *                 KN  BK  AR  SK   ACK
68  8400 00          CODE   FCB     0,$B4,$8B,$54,$16,0,$14,0
69                   *              ERR          SP
70  8408 01                 FCB     1,0,0,0,0,$21,0,0
71  8410 00                 FCB     0,0,0,0,0,0,0,0
72                   *                    AS
73  8418 00                 FCB     0,0,0,$44,0,0,0,0
74                   *               SP    "              '
75  8420 21          RTAB   FCB     $21,0,$4A,0,0,0,0,$7A
76                   *               (   )  * +  ,  -  .  /
77  8428 B6                 FCB     $B6,$B2,$44,$54,$CE,$86,$56,$94
78                   *               0   1   2   3   4  5  6   7
79  8430 FC                 FCB     $FC,$7C,$3C,$1C,$0C,4,$84,$C4
80                   *               8   9   :   ;    BT(=)  ?
81  8438 E4                 FCB     $E4,$F4,$E2,$AA,0,$8C,0,$32
82                   *                 A   B   C   D   E   F   G
83  8440 16                 FCB     $16,$60,$88,$A8,$90,$40,$28,$D0
84                   *               H   I   J   K   L   M   N   O
85  8448 08                 FCB     $08,$20,$78,$B0,$48,$E0,$A0,$F0
86                   *               P   Q   R   S   T   U   V   W
87  8450 68                 FCB     $68,$D8,$50,$10,$C0,$30,$18,$70
88                   *               X   Y   Z   Ä   Ö   Å
89  8458 98                 FCB     $98,$B8,$C8,$58,$E8,$6C

91                  ***RESTART ROUTINE***
92  845E 0F         RESRT   SEI
93  845F CE 00 7F           LDX     #$7F        CLR RAM 1-7F
94  8462 6F 00      L1      CLR     0,X
95  8464 09                 DEX
96  8465 26 FB              BNE     L1
97  8467 CE 0F 01           LDX     #$0F01
98  846A DF 0C              STX     SPEEDK      INZ SPEED CONS & RCHAR
99  846C 86 80              LDA A   #$80
100  846E 97 7F         STA A   BUFTOP      INZ BUFTOP
101  8470 CE A0 53      LDX     #BTOP       INZ BUFFER POINTERS
102  8473 DF 16         STX     P1
103  8475 09            DEX
104  8476 DF 18         STX     P2
105  8478 CE 84 00      LDX     #CODE
106  847B DF 00         STX     CVCX        INZ CVCX
107  847D CE 00 7F      LDX     #BUFTOP
108  8480 DF 02         STX     SAVEX
109  8482 CE 84 EC      LDX     #POLL       INZ IRQ VECTOR IN MIKBUG
110  8485 FF A0 00      STX     IRQVEC
111  8488 CE 80 24      LDX     #PIAAD
112  848B 6F 01         CLR     1,X         CLR PIAAC
113  848D 6F 03         CLR     3,X         CLR PIABC
114  848F 8E 00 07      LDS     #$0007      CA1=+ & ALLOWED
115  8492 AF 00         STS     0,X         (PIAAD & PIAAC)
116  8494 8E FF 34      LDS     #$FF34
117  8497 AF 02         STS     2,X         (PIABD & PIABC)
118  8499 A6 00         LDA A   0,X         CLR IRQA FLAGS
119  849B A6 02         LDA A   2,X         CLR IRQB FLAGS
120  849D 86 03         LDA A   #3
121  849F B7 80 08      STA A   ACIAC       RESET ACIA
122  84A2 86 95         LDA A   #$95        8 DATA,   1 STOPBIT, INTERUPT
123  84A4 B7 80 08      STA A   ACIAC       INZ ACIA
124  84A7 8E A0 7F      LDS     #STACK      INZ STACK POINTER
125  84AA 8D 29         BSR     SCRLF       PRINT CR/LF
```

```
125  84AA 8D 29          BSR    SCRLF     PRINT CR/LF
126  84AC CE 87 53       LDX    #MSG1
127  84AF 8D 27          BSR    SCS       PRINT MSG1
128  84B1 0E             CLI              CLEAR INTERRUPT FLAG

130                *** EXECUTIVE PROGRAM ***
131  84B2 96 06  EXEC   LDA A  BUFLAG    CHECK FOR CRLF
132  84B4 48            ASL A
133  84B5 2A 08         BPL    EX2       NO CRLF
134  84B7 8D 1C         BSR    SCRLF
135  84B9 96 06         LDA A  BUFLAG
136  84BB 84 BF         AND A  #$BF      ZERO B6 IN BUFLAG
137  84BD 97 06         STA A  BUFLAG
138  84BF 96 07  EX2    LDA A  CHCNT
139  84C1 81 41         CMP A  #65
140  84C3 23 0D         BLS    EX1
141  84C5 DE 02         LDX    SAVEX     66 CHAR FM KB
142  84C7 A6 00         LDA A  0,X
143  84C9 81 21         CMP A  #$21       LOOK FOR WORD SPACE
144  84CB 26 05         BNE    EX1        NO WORD SPACE
145  84CD 8D 06         BSR    SCRLF      CR/LF TO PRINTER
146  84CF 7F 00 07      CLR    CHCNT
147  84D2 3E      EX1   WAI
148  84D3 20 DD         BRA    EXEC

150                ***SUBR TO SEND CR/LF OR CHAR STRING TO PRINTER***
151  84D5 CE 87 DA SCRLFLDX    #CRLF
152  84D8 A6 00    SCS  LDA A  0,X
153  84DA 81 04         CMP A  #4
154  84DC 27 0D         BEQ    RTSR       END OF CHAR STRING
155  84DE F6 80 08 SCR1 LDA B  ACIAC
156  84E1 C5 02         BIT B  #2          TST FOR TX REG.EMPTY
157  84E3 27 F9         BEQ    SCR1        NOT EMPTY
158  84E5 B7 80 09      STA A  ACIAD       SEND CHAR TO PRINTER
159  84E8 08            INX
160  84E9 20 ED         BRA    SCS
161  84EB 39      RTSR  RTS

163                ***IRQ INTERRUPT ROUTINE***
164  84EC F6 80 25 POLL LDA B  PIAAC      SAVE INTR. FLAGS.
165  84EF B6 80 24      LDA A  PIAAD
166  84F2 46            ROR A
167  84F3 25 0A         BCS    RECM       REC MODE
168  84F5 5D      POLL1 TST B              CHECK FOR TX INTR.
169  84F6 2B 0E         BMI    POLL2      (CA1) YES
170  84F8 F6 80 08      LDA B  ACIAC      CHECK ACIA RDRF
171  84FB 56            ROR B
172  84FC 25 6D         BCS    COMMR1
173  84FE 3B            RTI

175  84FF 96 06  RECM   LDA A  BUFLAG
176  8501 2B F2         BMI    POLL1      DATA IN TX BUFFER
177  8503 7E 86 0B RECI JMP    REC        GO TO REC ROUTINE

179  8506 F6 80 26 POLL2LDA B  PIABD
180  8509 CA 02         ORA B  #2          XMIT LAMP ON
181  850B C4 FB         AND B  #$FB        REC LAMP OFF
182  850D F7 80 26      STA B  PIABD
183  8510 96 08         LDA A  COSTA       GET CODE OUTPUT STATUS
184  8512 2A 07         BPL    CFDAH       CHECK FOR DAH
185  8514 C6 7F         LDA B  #$7F
```

```
185   8514 C6 7F          LDA B  #$7F
186   8516 D7 05          STA B  RESMSK     DIT RESET
187   8518 5F      CES1   CLR B             DIT BEING SENT
188   8519 20 29          BRA    CKCNT      TEST COUNT

190   851B 48      CFDAH  ASL A             TEST FOR DAH
191   851C 2A 08          BPL    CFES       CHECK FOR ELEMENT SPACE
192   851E C6 BF          LDA B  #$BF
193   8520 D7 05          STA B  RESMSK     DAH RESET
194   8522 C6 04          LDA B  #4         DAH BEING SENT
195   8524 20 1E          BRA    CKCNT

197   8526 48      CFES   ASL A             TEST FOR ELEMENT SPACE
198   8527 2A 0A          BPL    CFWS       CHECK FOR WORD SPACE
199   8529 C6 DF          LDA B  #$DF
200   852B D7 05          STA B  RESMSK     ELEMENT SPACE RESET
201   852D 8D 2A          BSR    SOZERO
202   852F C6 01          LDA B  #1
203   8531 20 11          BRA    CKCNT

205   8533 48      CFWS   ASL A
206   8534 2A 08          BPL    CFCS       CHECK FOR SPACE
207   8536 C6 EF          LDA B  #$EF
208   8538 D7 05          STA B  RESMSK     WORD SPACE RESET
209   853A C6 08          LDA B  #8         WORD SPACE BEING SENT
210   853C 20 06          BRA    CKCNT

212   853E C6 F7   CFCS   LDA B  #$F7
213   8540 D7 05          STA B  RESMSK
214   8542 C6 04          LDA B  #4
215   8544 D1 04   CKCNT  CMP B  COUNT
216   8546 27 04          BEQ    CK1
217   8548 7C 00 04       INC    COUNT
218   854B 3B      RETRN  RTI

220   854C 7F 00 04  CK1  CLR    COUNT
221   854F 96 08          LDA A  COSTA
222   8551 94 05          AND A  RESMSK
223   8553 97 08          STA A  COSTA
224   8555 26 F4          BNE    RETRN
225   8557 20 14          BRA    GNEL

227                ***SEND OUT A ZERO***
228   8559 B6 80 26  SOZERO  LDA A  PIABD
229   855C 84 FE             AND A  #$FE
230   855E B7 80 26          STA A  PIABD
231   8561 39                RTS

233                ***SEND OUT A ONE***
234   8562 B6 80 26  SOONELDA A  PIABD
235   8565 8A 01             ORA A  #1         SET BIT ZERO
236   8567 B7 80 26          STA A  PIABD
237   856A 39                RTS

239   856B 20 53  COMMR1  BRA    COMMR

241                ***GET NEW ELEMENT FOR OUTPUT***
242   856D 96 7F  GNEL    LDA A  BUFTOP
243   856F 81 80          CMP A  #$80       TEST FOR CHAR END
244   8571 26 22          BNE    GNEL1
245   8573 86 08          LDA A  #$08       SEND OUT CHAR. SPACE
```

```
245  8573 86 08          LDA A  #$08      SEND OUT CHAR. SPACE
246  8575 97 08          STA A  COSTA
247  8577 8D E0          BSR    SOZERO
248  8579 DE 02          LDX    SAVEX
249  857B 8C 00 7F       CPX    #BUFTOP
250  857E 26 07          BNE    GNELR
251  8580 96 06          LDA A  BUFLAG    BUFFER EMPTY
252  8582 84 7F          AND A  #$7F
253  8584 97 06          STA A  BUFLAG    RESET FLAG
254  8586 3B             RTI
255                 *
256  8587 8D 27  GNELR   BSR    MOVUP
257  8589 96 7F          LDA A  BUFTOP
258  858B 81 21          CMP A  #$21      LOOK FOR SPACE
259  858D 27 16          BEQ    SWS
260  858F 7D 00 08       TST    COSTA
261  8592 27 01          BEQ    GNEL1
262  8594 3B             RTI
263                 *
264  8595 78 00 7F GNEL1 ASL    BUFTOP
265  8598 25 04          BCS    SODAH     SEND OUT DAH
266  859A 86 A0          LDA A  #$A0      SEND OUT DIT & ELEMENT SPAC
267  859C 20 02          BRA    SOEL
268                 *
269  859E 86 60  SODAH   LDA A  #$60      SEND OUT DAH &ELEMENT SPACE
270  85A0 97 08  SOEL    STA A  COSTA     SET STATUS
271  85A2 8D BE          BSR    SOONE
272  85A4 3B             RTI
273                 *
274  85A5 86 80  SWS     LDA A  #$80
275  85A7 97 7F          STA A  BUFTOP
276  85A9 86 10          LDA A  #$10
277  85AB 97 08          STA A  COSTA     SEND OUT WORD SPACE
278  85AD 8D AA          BSR    SOZERO
279  85AF 3B             RTI
280                 *
281  85B0 CE 00 65 MOVUP LDX    #BUFTOP-BUFBOT
282  85B3 A6 19  MOV1    LDA A  BUFBOT-1,X
283  85B5 A7 1A          STA A  BUFBOT,X
284  85B7 09             DEX
285  85B8 26 F9          BNE    MOV1
286  85BA DE 02          LDX    SAVEX     UPDATE INPUT POINTER
287  85BC 08             INX
288  85BD DF 02          STX    SAVEX
289  85BF 39     MOVRT   RTS

291                 ***COMM REC INTERRUPT ROUTINE***
292  85C0 DE 02  COMMR   LDX    SAVEX
293  85C2 7C 00 07       INC    CHCNT     INC CHAR COUNTER
294  85C5 B6 80 09       LDA A  ACIAD     GET KB DATA
295  85C8 8C 00 1A       CPX    #BUFBOT
296  85CB 27 32          BEQ    BUFFUL    MEMORY BUFFER FULL
297  85CD C5 18          BIT B  #$18      CK FOR FE OR OVRN
298  85CF 26 32          BNE    ERROR
299  85D1 81 0D          CMP A  #$0D      CK FOR CR
300  85D3 27 1F          BEQ    SCR       SEND CR
301  85D5 84 7F          AND A  #$7F      MASKA BORT BIT 7
302  85D7 81 5D          CMP A  #$5D
303  85D9 22 2C          BHI    BADCH     ILLEGAL CHAR
304  85DB 97 01  STM     STA A  CVCX+1
305  85DD DE 00          LDX    CVCX
```

```
305  85DD DE 00          LDX     CVCX
306  85DF E6 00          LDA B   0,X         DATA CONVERTED TO MORSE CODE
307  85E1 27 24          BEQ     BADCH       KICK OUT BAD CHAR'S
308  85E3 DE 02          LDX     SAVEX
309  85E5 09             DEX
310  85E6 E7 00          STA B   0,X         STORE DATA IN BUFFER
311  85E8 DF 02          STX     SAVEX       MOVE POINTER DOWN
312  85EA D6 06          LDA B   BUFLAG
313  85EC CA 80          ORA B   #$80        SET FLAG BIT 7
314  85EE D7 06          STA B   BUFLAG
315  85F0 B7 80 09 PRNT  STA A   ACIAD       PRINT CHARACTER
316  85F3 3B             RTI
317                 *
318  85F4 D6 06  SCR     LDA B   BUFLAG
319  85F6 CA 40          ORA B   #$40        SET BIT 6
320  85F8 D7 06          STA B   BUFLAG
321  85FA 7F 00 07       CLR     CHCNT       RESET COUNTER TO 0
322  85FD 20 DC          BRA     STM
323                 *
324  85FF 86 2A  BUFFUL  LDA A   #'*         PRINT * FOR BUFFUL
325  8601 20 ED          BRA     PRNT
326                 *
327  8603 86 3F  ERROR   LDA A   #'?         PRINT ? FOR ERROR
328  8605 20 E9          BRA     PRNT
329                 *
330  8607 86 07  BADCH   LDA A   #7          RING BELL FOR BAD CHAR
331  8609 20 E5          BRA     PRNT

333                 ***START OF RECEIVE ROUTINE***
334  860B B6 80 26 REC   LDA A   PIABD
335  860E 8A 04          ORA A   #4          REC LAMP ON
336  8610 84 FD          AND A   #$FD        XMIT LAMP OFF
337  8612 B7 80 26       STA A   PIABD
338  8615 86 FF  KEYUP   LDA A   #$FF        RESET INTERVAL COUNTER
339  8617 81 FE  KULOOP  CMP A   #$FE        INTERVAL COUNTER AT MAX?
340  8619 27 03          BEQ     NOINC       DO NOT INCR IF MAX
341  861B 4C             INC A               INCREMENT INTERVAL COUNTER
342  861C 20 0A          BRA     KUCONT

344  861E 97 12  NOINC   STA A   KUTIM       SAVE KU INTERVAL TIME
345  8620 D6 0D          LDA B   RCHAR       GET RCVD CHAR
346  8622 C1 01          CMP B   #1          ANYTHING THERE?
347  8624 27 02          BEQ     KUCONT
348  8626 8D 4E          BSR     COMPKU
349  8628 F6 80 24 KUCONT LDA B  PIAAD       CHECK INPUT
350  862B 2B 08          BMI     KD          BRANCH IF KEYDOWN
351  862D 8D 35          BSR     TIMER       TIME DELAY
352  862F 56             ROR B               CHECK T/R SWITCH
353  8630 25 E5          BCS     KULOOP
354  8632 7E 84 EC       JMP     POLL        RETURN TO TX ROUTINE

356  8635 97 12  KD      STA A   KUTIM       SAVE KU INTERVAL TIME
357  8637 8D 3D          BSR     COMPKU
358  8639 86 FF  KEYDWN  LDA A   #$FF        RESET INTERVAL TIMER
359  863B 81 FE  KDLOOP  CMP A   #$FE        INTERVAL COUNTER AT MAX?
360  863D 27 01          BEQ     MAXKD       DO NOT INCR IF MAX
361  863F 4C             INC A               INCREMENT INTERVAL COUNTER
362  8640 F6 80 24 MAXKD LDA B   PIAAD       CHECK INPUT
363  8643 2A 04          BPL     KU          BRANCH IF KEYUP
364  8645 8D 1D          BSR     TIMER
365  8647 20 F2          BRA     KDLOOP
```

```
365   8647 20 F2              BRA     KDLOOP

367   8649 97 11  KU          STA A   KDTIM       SAVE KD INTERVAL TIME
368   864B 81 04              CMP A   #$4         KD INTERVAL TIME TOO LOW?
369   864D 24 05              BCC     CKHI        BRANCH IF NOT TOO LOW
370   864F 96 0C              LDA A   SPEEDK
371   8651 44                 LSR A               DIVIDE SPEED CONSTANT BY 2
372   8652 20 07              BRA     UNZERO

374   8654 81 7F  CKHI        CMP A   #$7F        KD INTERVAL TIME TOO HIGH?
375   8656 25 07              BCS     CMPTKD      BRANCH IF OK
376   8658 96 0C                  LDA A   SPEEDK
377   865A 48                     ASL A               MULTIPLY SPEED CONSTANT
378   865B 8A 01      UNZERO      ORA A   #1          ASCERTAIN SPEED CONSTANT
379   865D 97 0C                  STA A   SPEEDK      NOT SET TO ZERO
380   865F 8D 32      CMPTKD      BSR     COMPKD
381   8661 7E 86 15               JMP     KEYUP

383                   ***SUBROUTINE TO CREATE TIME DELAY***
384   8664 37         TIMER       PSH B               SAVE B
385   8665 36                     PSH A               SAVE A
386   8666 D6 0C                  LDA B   SPEEDK
387   8668 86 40      DELOP2      LDA A   #$40
388   866A 4A         DELOOP      DEC A
389   866B 26 FD                  BNE     DELOOP
390   866D 5A                     DEC B
391   866E 26 F8                  BNE     DELOP2
392   8670 BD 87 03               JSR     PACH        PRINT CHAR IN BUFFER
393   8673 32                     PUL A               RESTORE A
394   8674 33                     PUL B               RESTORE B
395   8675 39                     RTS

397                   ***SUBROUTINE TO COMPUTE KU***
398   8676 91 0F      COMPKU      CMP A   TQLDAT
399   8678 25 18                  BCS     MOREL       BRANCH IF KUTIM < TQLDAT
400   867A 96 0D                  LDA A   RCHAR       GET CHAR BEING RECEIVED
401   867C 81 01                  CMP A   #1
402   867E 27 08                  BEQ     CKFSP
403   8680 8D 67                  BSR     GAFT        GET ASCII FROM TABLE
404   8682 8D 46                  BSR     PRINT
405   8684 86 01                  LDA A   #1
406   8686 97 0D                  STA A   RCHAR       READY FOR NEW CHAR
407   8688 96 10      CKFSP       LDA A   TLDAT       GET TWICE LAST DASH TIME
408   868A 91 12                  CMP A   KUTIM       COMPARE WITH KU INTERVAL
409   868C 24 04                  BCC     MOREL       BRANCH IF TLDAT >= KU IN
410   868E 86 20                  LDA A   #$20        ASCII SPACE
411   8690 8D 38                  BSR     PRINT       PRINT SPACE
412   8692 39         MOREL       RTS

414                   ***SUBROUTINE TO COMPUTE KD***
415   8693 96 0B      COMPKD      LDA A   TLETIM     GET TWICE LAST ELEM TIME
416   8695 91 11                  CMP A   KDTIM      COMPARE WITH KD INTERVAL
417   8697 25 10                  BCS     DASHEL      BRANCH IF TLETIM < KD IN
418   8699 96 0A                  LDA A   HLETIM     GET HALF LAST ELEM TIME
419   869B 91 11                  CMP A   KDTIM      COMPARE WITH KD INTERVAL
420   869D 24 04                  BCC     DOTEL       BRANCH IF HLETIM >= KD I
421   869F 96 09                  LDA A   LETYPE      CHECK LAST ELEMENT TYPE
422   86A1 26 06                  BNE     DASHEL      BRANCH IF LAST ELEM WAS
423   86A3 7F 00 09   DOTEL      CLR     LETYPE      MAKE LAST ELEM TYPE=00=D
424   86A6 0C                     CLC
425   86A7 20 14                  BRA     ADDEL
```

73 MPU-laren NR$ 1980

```
425  86A7 20 14              BRA    ADDEL
426  86A9 7C 00 09 DASHEL    INC    LETYPE    MAKE LAST ELEM TYPE=DASH
427  86AC 96 11              LDA A  KDTIM     GET KD INTERVAL
428  86AE 97 0E              STA A  LDATIM    STORE IN LAST DASH TIME
429  86B0 16                 TAB              SAVE IN ACCUM B
430  86B1 44                 LSR A            DIVIDE KD INTERVAL BY 2
431  86B2 97 0F              STA A  TQLDAT    SAVE 1/2 KD
432  86B4 44                 LSR A            DIVIDE 1/2 KD BY 2
433  86B5 9B 0F              ADD A  TQLDAT    ADD 1/2 TO 1/4 KD INTERVAL
434  86B7 97 0F              STA A  TQLDAT    STORE RESULT
435  86B9 58                 ASL B            MULTIPLY KD INTERVAL BY 2
436  86BA D7 10              STA B  TLDAT     STORE RESULT
437  86BC 0D                 SEC
438  86BD 79 00 0D ADDEL     ROL    RCHAR     ADD NEW ELEM TO CHARACTER
439  86C0 96 11              LDA A  KDTIM     GET KD INTERVAL
440  86C2 16                 TAB              SAVE IN ACCUM B
441  86C3 44                 LSR A            DIVIDE KD BY 2
442  86C4 97 0A              STA A  HLETIM    STORE 1/2 KD INTERVAL
443  86C6 58                 ASL B            MULTIPLY KD BY 2
444  86C7 D7 0B              STA B  TLETIM    STORE TWICE KD INTERVAL
445  86C9 39                 RTS

447                  ***SUBROUTINE TO PRINT RCVD CHAR***
448  86CA 7C 00 13 PRINT     INC    CHCTR     INCREMENT CHAR COUNTER
449  86CD D6 13              LDA B  CHCTR
450  86CF C1 3C              CMP B  #60       CHECK CHAR COUNT
451  86D1 25 13              BCS    OK        BRANCH IF LESS THAN 60
452  86D3 C1 47              CMP B  #71
453  86D5 25 0B              BCS    CKSP      BRANCH IF LESS THAN 71
454  86D7 36           ENDLIN PSH A            SAVE A
455  86D8 8D 66              BSR    CRLFR     PRINT CR AND LF
456  86DA 7F 00 13           CLR    CHCTR     RESET CHARACTER COUNTER
457  86DD 32                 PUL A            RESTORE A
458  86DE 81 20              CMP A  #$20      IS CHARACTER A SPACE?
459  86E0 27 06              BEQ    DONE      BRANCH IF YES
460  86E2 81 20        CKSP  CMP A  #$20
461  86E4 27 F1              BEQ    ENDLIN    END THE LINE IF SPACE
462  86E6 20 2A        OK    BRA    SDATA     STORE CHARACTER
463  86E8 39           DONE  RTS

465                  ***SUBR. TO GET ASCII CHAR FROM CODE TABLE***
466  86E9 0D           GAFT  SEC              CHANGE FORMAT OF RCHAR.
467  86EA 49                 ROL A
468  86EB 48           GAFT1 ASL A
469  86EC 24 FD              BCC    GAFT1
470  86EE CE 84 5D           LDX    #RESRT-1
471  86F1 A1 00        STAB1 CMP A  0,X
472  86F3 27 09              BEQ    TABM      FOUND MATCH
473  86F5 09                 DEX
474  86F6 8C 84 21           CPX    #RTAB+1   END OF TABLE?
475  86F9 26 F6              BNE    STAB1     NO!
476  86FB 86 5F              LDA A  #'_       RETURN "_" FOR NO MATCH
477  86FD 39                 RTS
478                  *
479  86FE DF 14        TABM  STX    RECX
480  8700 96 15              LDA A  RECX+1    X(LOW) TO ACCA.
481  8702 39                 RTS

483                  ***SUBROUTINE TO PRINT 1 ASCII CHAR***
484  8703 F6 80 08 PACH      LDA B  ACIAC
485  8706 C5 02              BIT B  #2
```

```
485  8706 C5 02              BIT B  #2
486  8708 27 07              BEQ    PACH1      TST FOR TX REG. EMPTY.
487  870A 8D 1D              BSR    GDATA
488  870C 24 03              BCC    PACH1      EMPTY
489  870E B7 80 09           STA A  ACIAD      PRINT CHARACTER
490  8711 39         PACH1   RTS

492                  ***SUBROUTINE TO STORE DATA IN REC BUFFER***
493  8712 DE 18      SDATA   LDX    P2
494  8714 9C 16              CPX    P1
495  8716 27 0F              BEQ    BFUL
496  8718 8C A0 49           CPX    #BBOT-1
497  871B 26 03              BNE    SDATA1
498  871D CE A0 53           LDX    #BTOP
499  8720 A7 00      SDATA1  STA A  0,X
500  8722 09                 DEX
501  8723 DF 18              STX    P2
502  8725 0C                 CLC                NOT FULL
503  8726 39                 RTS
504                  *
505  8727 0D         BFUL    SEC                BUFFER FULL
506  8728 39                 RTS

508                  ***SUBROUTINE TO GET DATA FROM REC BUFFER***
509  8729 DE 16      GDATA   LDX   P1
510  872B 09                 DEX
511  872C 9C 18              CPX    P2
512  872E 27 0E              BEQ    EMPTY
513  8730 8C A0 49           CPX    #BBOT-1
514  8733 26 03              BNE    GDATA1
515  8735 CE A0 53           LDX    #BTOP
516  8738 A6 00      GDATA1  LDA A  0,X
517  873A DF 16              STX    P1
518  873C 0D                 SEC                NOT EMPTY
519  873D 39                 RTS
520                  *
521  873E 0C         EMPTY   CLC                EMPTY
522  873F 39                 RTS

524                  ***SUBROUTINE TO SEND CR/LF TO REC BUFFER***
525  8740 CE 87 DA   CRLFR   LDX    #CRLF
526  8743 A6 00      CRR     LDA A  0,X
527  8745 81 04              CMP A  #4
528  8747 27 09              BEQ    CRR1
529  8749 DF 14              STX    RECX
530  874B 8D C5              BSR    SDATA
531  874D DE 14              LDX    RECX
532  874F 08                 INX
533  8750 20 F1              BRA    CRR
534  8752 39         CRR1    RTS

536                  *** CHARACTER STRINGS ***
537  8753 4D         MSG1    FCC    'MORSE CODE SEND/RECEIVE '
538  876B 0A                 FCB    $0A,$0D
539  876D 50                 FCC    'PROGRAM FOR MOTOROLA '
540  8782 36                 FCC    '6800 MICROPROCESSOR'
541  8795 0D                 FCB    $0D,$0A,0,0,0,0
542  879B 56                 FCC    'VER.2.5  ** W7KGU ** WA7VKZ **'
543  87BA 0A                 FCB    $0A,$0D
544  87BC 4D                 FCC    'MODIFIERAT AV *** SMOKDE ***'
545  87D8 0A                 FCB    $0A,$0D
```

```
545  87D8 0A                    FCB    $0A,$0D
546  87DA 0D          CRLF      FCB    $0D,$0A,0,0,0,0,4
547                   * FIXA STARTADRESS
548  87F0                       ORG    CODE+$3F0
549  87F0 7E 84 5E              JMP    RESRT
550                   *

552  87F8                       ORG    CODE+$3F8
553  87F8 84 EC                 FDB    POLL,RESRT,RESRT,RESRT VECTORS
554                             END    RESRT

NO ERROR(S) DETECTED

 SYMBOL TABLE:

ACIAC8008    ACIAD  8009    ADDEL  86BD    BADCH  8607    BBOT   A04A
BFUL 8727    BTOP   A053    BUFBOT 001A    BUFFUL 85FF    BUFLAG 0006
BUFTOP 007F  CES1   8518    CFCS   853E    CFDAH  851B    CFES   8526
CFWS 8533    CHCNT  0007    CHCTR  0013    CK1    854C    CKCNT  8544
CKFSP8688    CKHI   8654    CKSP   86E2    CMPTKD 865F    CODE   8400
COMMR85C0    COMMR1 856B    COMPKD 8693    COMPKU 8676    COSTA  0008
COUNT0004    CRLF   87DA    CRLFR  8740    CRR    8743    CRR1   8752
CVCX 0000    DASHEL 86A9    DELOOP 866A    DELOP2 8668    DONE   86E8
DOTEL86A3    EMPTY  873E    ENDLIN 86D7    ERROR  8603    EX1    84D2
EX2  84BF    EXEC   84B2    GAFT   86E9    GAFT1  86EB    GDATA  8729
GDATA1 8738  GNEL   856D    GNEL1  8595    GNELR  8587    HLETIM 000A
IRQVEC A000  KD     8635    KDLOOP 863B    KDTIM  0011    KEYDWN 8639
KEYUP8615    KU     8649    KUCONT 8628    KULOOP 8617    KUTIM  0012
L1   8462    LDATIM 000E    LETYPE 0009    MAXKD  8640    MOREL  8692
MOV1 85B3    MOVRT  85BF    MOVUP  85B0    MSG1   8753    NOINC  861E
OK   86E6    P1     0016    P2     0018    PACH   8703    PACH1  8711
PIAAC8025    PIAAD  8024    PIABC  8027    PIABD  8026    POLL   84EC
POLL184F5    POLL2  8506    PRINT  86CA    PRNT   85F0    RCHAR  000D
REC  860B    RECI   8503    RECM   84FF    RECX   0014    RESMSK 0005
RESRT845E    RETRN  854B    RTAB   8420    RTSR   84EB    SAVEX  0002
SCR  85F4    SCR1   84DE    SCRLF  84D5    SCS    84D8    SDATA  8712
SDATA1 8720  SODAH  859E    SOEL   85A0    SOONE  8562    SOZERO 8559
SPEEDK 000C  STAB1  86F1    STACK  A07F    STM    85DB    SWS    85A5
TABM 86FE    TIMER  8664    TLDAT  0010    TLETIM 000B    TQLDAT 000F
UNZERO 865B
```