

R

C B N**ORGAN FOR COMAL BRUGER GRUPPE DANMARK**

1983, nr. 3

COMAL på flere måder

Der er en rivende udvikling i mikrocomputermarkedet. Ingen, der følger med i inden- og udenlandske tidsskrifter om emnet, kan være i tvivl. Der går næppe en måned, hvor der ikke dukker en ny maskine op.

Naturligvis kommer der flest nye af den type, der er karakteriseret ved at have alt samlet i et kabinet, der set udefra nærmest er et løsrevet tastatur; men som på en utrolig lille plads rummer alt, hvad der fordres for at udgøre en hel computer, når der blot tilsluttes en eller anden form for skærm - ja, det nyeste er, at ikke så få af disse datamater har et LCD-display med 2 - 8 liniers tekst eller tal.

Vore læsere spejder selvfølgelig efter, om de nye maskiner fødes med eller kan udbygges med COMAL, og her er oplysningerne noget usikre.

NEW BRAIN har i hvert fald reklameret med COMAL, og også i forbindelse med COMMODORE 64 har COMAL været omtalt. Men så vidt vides, er ingen af maskinerne endnu fremme på markedet med COMAL.

Anderledes stiller det sig med METANIC's seneste frembringelse: RUTLER, der ganske vist hører til den mere avancerede type, der bedst kan sammenlignes med COMEL'en, PICCOLO'en og CBN 8032 SK'en. Den skulle efter de udsendte oplysninger allerede være på markedet.

Efterhånden som mere konkrete efterretninger foreligger, vil vi berette om disse maskiner og deres COMAL.

Indholdsfortegnelse

Leder: COMAL på flere datamater	1
Procedurer - Sortering	2
Små tips: Blinkende bogstav	5
Nyt fra forlag og firmaer	6
Små tips: LOAD, RUN eller DELETE	7
Spørgrebrevkassen	8
Små tips: Programhoved	10
Hold styr på disketterne	11
En sikker inputrutine i COMAL	11

200002999314



Universitetsbiblioteket, 2. afd.

0108-4925

THE DANISH GAZETTE FOR COMAL LANGUAGE

Procedurer - sortering

Nedenstående programmer skal vise anvendelsen af stående procedurer og desuden tjene som grundlag for gennemgang af begrebet parameter-overførsel.

Som eksempel er valgt et program, der tænkes brugt til at afgøre, hvem der er valgt ind i en bestyrelse. Det indledes med, at maskinen skriver efter tur "KANDIDAT nr.", hvorefter man så indtaster den pågældendes stemmetal.

Der er her tale om så enkelt et program som muligt - og selve programmet vil næppe have interesse for nogen; men den videre behandling af tallene vil formentlig vise nogle fremgangsmåder, som begynderen kan have glæde af.

```
0010 // save "1.valg"
0020
0030 alle:=4
0040 DIM sttal(alle)
0050 FOR nr:=1 TO alle DO
0060   PRINT "kandidat";nr,". -";
0070   INPUT "skriv stemmetal: ":
       sttal(nr)
0080 ENDFOR nr
```

Når program "1.valg" køres, kommer følgende udskrift på skærmen (med de her valgte stemmetal):

```
KANDIDAT 1. - SKRIV STEMMETAL: 23
KANDIDAT 2. - SKRIV STEMMETAL: 8
KANDIDAT 3. - SKRIV STEMMETAL: 12
KANDIDAT 4. - SKRIV STEMMETAL: 56
```

Nu drejer det sig om at få sorteret tallene efter faldende størrelse.

På disketten har vi en stående procedure, der hedder "L2.Sortering".

Det er den helt enkle boblesortering - den simpleste af alle.

Den har et L stående først i sit navn, hvilket betyder, at den ikke er "SAVE"et over på disken, men "LIST"et, og det vil igen sige, at vi ikke kan kalde den med "LOAD", men med "ENTER".

I virkeligheden vil vi gøre noget ganske andet, fordi vi gerne vil have sorteringsproceduren hægtet på vort 1. program.

Så skal vi bruge kommandoen "MERGE", der ligesom "ENTER" kan anvendes ved kald af en "LIST"-fil.

Men her er nu først proceduren "Sortering":

```
0010 // list "12.boblesortering"
0020
0030 PROC sortering
0040   FOR i:=1 TO slut DO
0050     FOR x:=(i+1) TO slut DO
0060       IF tal(x)<tal(i) THEN
0070         a:=tal(i)
0080         tal(i):=tal(x)
0090         tal(x):=a
0100       ENDIF
0110     ENDFOR x
0120   ENDFOR i
0130 ENDPROC sortering
```

Så er vi klar!

I computeren har vi (eller "LOAD"er vi) program "1.Valg".

Derefter taster vi MERGE "12.boblesortering", og så har vi et program, vi kan tilrette, så det kommer til at se sådan ud:

COMAL BRUGER NYT

```
0010 // save "3.vnlg"
0020
0030 alle:=4
0040 DIM sttal(alle)
0050 FOR nr:=1 TO alle DO
0060   PRINT "kandidat";nr,". -";
0070   INPUT "skriv stemmetal: ":
      sttal(nr)
0080 ENDFOR nr
```

```
0090
0100 sortering
0110
0120 // list "12.boblesortering"
0130
0140 PROC sortering
0150   FOR i:=1 TO alle DO
0160     FOR x:=(i+1) TO alle DO
0170       IF sttal(x)<sttal(i)
          THEN
0180         a:=sttal(i)
0190         sttal(i):=sttal(x)
0200         sttal(x):=a
0210       ENDIF
0220     ENDFOR x
0230   ENDFOR i
0240 ENDPROC sortering
0250
0260 PRINT
0270 FOR i:=alle TO 1 STEP -1 DO
0280   PRINT sttal(i) .
0290 ENDFOR i
```

i linie 100 kalder vi proceduren sortering.

Fra linie 120 til linie 240 har vi proceduren, som vi "MERGE"de fra disketten. men vi har ændret variabelen tal til sttal, fordi det er dette variabelnavn, vi har brugt i begyndelsen af programmet.

Derudover har vi tilføjet linierne 260 - 290, der udskriver stemmetallene, efter at de er ordnet.

Bemærk, at selve sorteringsproceduren ordner stemmetallene, så det største kommer først; men udskrivningsalgoritmen tager tallene bagfra (fra alle til 1).

Man kunne lige så godt have ladet sorteringsproceduren ordne tallene fra det største mod det mindste - blot ved at vende ulighedstegnet om i linie 170, så det blev til > i stedet for <. Men så skulle linie 270 naturligvis gå fra 1 til alle (og "step -1" skulle udelades).

Her har vi altså anvendt vor procedure uden parameteroverførsel ved simpelthen at bruge samme variabel (sttal) gennem hele programmet. Men det medfører altså, at man skal rette i den stående procedure, der altid er parat på disketten.

C B N
COMAL BRUGER NYT
ISSN 0108-4925



Udgiver:
C B N
Comal Bruger Gruppe
Mindegade 42
8700 Horsens

Tlf. 05 - 62 15 67
Giro 1 75 23 75

Ansvarshavende
redaktør:
S. Chr. Hansen

Layout:
Frank Hejndorf

Tryk:
tekst & tryk
Horsens

Abonnement:
Mindegade 42
8700 Horsens

COMAL BRUGER NYT

Det må nok siges at være snildere at anvende proceduren uændret, som den kommer fra disketten, men så indføre parameteroverførsel.

Det gør vi i næste eksempel:

```
0010 // save 4.valg
0020
0030 alle:=4
0040 DIM sttal(alle)
0050 FOR nr:=1 TO alle DO
0060   PRINT "kandidat";nr,". -";
0070   INPUT "skriv stemmetal: ":
      sttal(nr)
0080 ENDFOR nr
0090
0100 sortering(sttal,alle)
0110
0120 // list 12.boblesortering
0130
0140 PROC sortering(REF
      tal(),slut)
0150   FOR i:=1 TO slut DO
0160     FOR x:=(i+1) TO slut DO
0170       IF tal(x)<tal(i) THEN
0180         a:=tal(i)
0190         tal(i):=tal(x)
0200         tal(x):=a
0210       ENDIF
0220     ENDFOR x
0230   ENDFOR i
0240 ENDPROC sortering
0250
0260 PRINT
0270 FOR i:=alle TO 1 STEP -1 DO
0280   PRINT sttal(i)
0290 ENDFOR i
```

I linie 100 kalder vi ligesom sidst proceduren sortering; men denne gang tilføjer vi de parametre, vi ønsker overført. Det er variablerne sttal og alle.

Men vi vil ikke nøjes med at overføre værdien af disse variabler til sorteringproceduren, vi vil også have proceduren til - når sorteringen er færdig - at aflevere resultatet til de oprindelige variabler, så sttal fortsat kan bruges som variabel, men nu med nye værdier.

Det klarer vi ved i procedurehovedets formelle parametre at anføre ordet REF foran den variabel, der skal modtage de nuværende og senere afgive de nye værdier.

Men i parenteser for de formelle parametre skal vi tillige have en ekstra (tom) parentes, der fortæller, at vor variabel: sttal i virkeligheden er et array.

I selve proceduren er der overhovedet ikke foretaget rettelser.

Altså: alt, hvad man skal gøre er at anføre de aktuelle variabler i parameterparentesen ved procedurekaldet og de formelle variabler i procedurehovedets parameterparentes.

Efter endt sortering kan vi så udskrive resultatet ved anvendelse af de oprindelige variabelnavne: sttal og alle.

Som nævnt i indledningen er dette program ikke meget anvendelig - man kunne stort set lige så godt ordne tallene ved håndkraft. Specielt er det en svaghed, at man i udskriften af valgresultatet ikke engang får anført, hvilke kandidater, der har fået de forskellige stemmetal.

Det kan man nu imidlertid hurtigt råde bod på, og så kan man samtidig se, hvordan man får kandidaterne sorteret ind i samme rækkefølge som stemmetallene.

```
0010 // save 5.valg
0020
0030 PRINT CHR$(147)
0040
0050 alle:=4
0060 DIM sttal(alle), kand(alle)
0070 FOR nr:=1 TO alle DO
0080   PRINT "kandidat";nr,". -";
0090   INPUT "skriv stemmetal: ":
      sttal(nr)
0100   kand(nr):=nr
0110 ENDFOR nr
```

COMAL BRUGER NYT

```

0120
0130 sortering(stattal,kand,alle)
0140
0150 // list 12.boblesortering
0160
0170 PROC sortering(REF tal(),REF
      pers(),slut)
0180   FOR i:=1 TO slut DO
0190     FOR x:=(i+1) TO slut DO
0200       IF tal(x)<tal(i) THEN
0210         a:=tal(i); b:=pers(i)
0220         tal(i):=tal(x);
           pers(i):=pers(x)
0230         tal(x):=a; pers(x):=b
0240       ENDIF
0250     ENDFOR x
0260   ENDFOR i
0270 ENDPROC sortering
0280
0290 PRINT
0300 FOR i:=ølle TO 1 STEP -1 DO
0310   PRINT "kandidat nr. ";
           kand(i); "fik ";stattal(i);
           "stemmer"
0320 ENDFOR i

```

Programmet indledes med, at indholdet af skærmen slettes (CLEAR SCREEN = chr\$(147)).

Programlinierne er derudover de samme som i de foregående udgaver. Her er dog tilføjet en linie 100, der betegner kandidaten ved sit nr.

I linie 130 har vi procedurekaldet, hvor der nu er tilføjet kand, fordi vi ønsker, at kandidaterne - karakteriseret ved deres nummer - skal sorteres med.

I procedurehovedet - linie 170 - er der indføjet en ny referenceparameter: pers, der skal forstås som person, og den tomme parentes betyder også her, at der er tale om et array.

I sorteringsproceduren ses det, hvorledes vi får sorteret kandidaten med på samme måde som stemmetallene sorteres: simpelthen ved i

ombytningerne (linie 210 - 230) at medtage pers med samme index som tal. Tal og pers skal blot adskilles med et semikolon. Og så skal tal og pers naturligvis have hver sin midlertidige variabel (a og b) i linie 210 og 230.

I udskriften bruges ligesom sidst de oprindelige variable: alle, kand og stattal.

C-B-N

Små tips

Blinkende bogstav

Her er et lille program, hvor et blinkende bogstav bruges som blikfang, så man ikke kan undgå at få øje på, hvad man skal foretage sig.

```

CHR$(145) = cursor up
CHR$(147) = clear screen

```

```

0010 // save "Blinkende bogstav"
0020
0030 PRINT CHR$(147)
0040
0050 LOOP
0060   PRINT CHR$(145),TAB(0),"B" =
           Beqynd!"
0070   pause(.2)
0080
0090   CASE KEY$ OF
0100     WHEN "b,"B"
0110       EXIT
0120     OTHERWISE
0130       PRINT CHR$(145),TAB(0)," "
0140       pause(.2)
0150     ENDCASE
0160   ENDOLOOP
0170
0180 PRINT
0190 PRINT "Klar til at beqynde."
0200
0210 PROC pause(sec)
0220   TIME 0
0230   WHILE TIME<60*sec DO NULL
0240 ENDPROC pause

```

COMAL BRUGER NYT

Fra forlag og firmaer

Corstensen, Jens
 LIGNINGER OG INFINITESIMALREGNING.
 ISBN 87-7351-094-7
 PRINTAL, DIVISORER OG TALFØLGER.
 ISBN 87-7351-092-0
 SANDSYNLIGHEDSREGNING, SIMULATION
 OG SPIL.
 ISBN 87-7351-061-0
 Ca. 62 sider. Format A 5.
 Kr. 39,00.
 forlaget systime a/s, Herning

Datamater er af natur regnemaskiner, og at de kan regne, er man grundigt overbevist om, når man har læst Jens Corstensen's tre bøger.

Bøgerne er først og fremmest skrevet til anvendelse i forbindelse med gymnasiets matematikundervisning, og emnerne set under et strækker sig over alle tre gymnasieklasser - lige fra to ligninger med to ubekendte til de mere sofistikerede behandlinger af avanceret talteori.

Ind imellem findes programmer, der er mere alment anvendelige som f. eks. amortisation.

Af særlig interesse for EDB folk er behandlingen af forskellige metoder til såvel alfabetisk som nummerisk sortering.

Som titlen angiver, handler den ene bog blandt andet om spil, og den giver i direkte modsætning til de alle vegne tilgængelige ukommenterede tidsfordriv særdeles gode forklaringer på det enkelte spils ide og opbygning. Hvor det er muligt, afsløres tilmed en vinderstrategi, og først derefter udvikles og forklares programmet.

Alle tre bøger er velskrevet i et letforståeligt sprog med gode forklaringer og illustrationer.

Programmeringssproget er COMAL, og forfatteren har i vid udstrækning undgået at anvende maskinspecifikke fremgangsmåder, så det skulle være en let sag at indtaste programmerne på enhver COMAL datamat.

De tre bøger slutter med 20 gode opgaver (ens for alle bøgerne).

Programdisketten, der er fælles for bøgerne, er skrevet for COMET - MPS 3000.

S. Chr. H.



Børge R. Christensen
 COMAL
 PROGRAMLØSNING OG PROGRAMMERING
 ISBN 87-88059-05-7
 176 sider. Format A 4.
 Kr. 80,00.
 PROGRAMHÆFTE
 ISBN 87-88059-08-1
 PROGRAMDISK KR. 200,00
 Priser er excl. moms og forsendelse.
 Bogika Data Systemer, Ikast.

Børge Christensen's lærebøger i COMAL er ved at udkomme i ny udgave.

Hvor der tidligere til hver del hørte to bøger - en grundbog og et arbejdshæfte - er der nu kun en bog, der rummer hele staffet.

Det synes at være en god ide.

Foreløbig er første del færdig, og anden del må formodes at komme snart.

Bogen præges af en meget klar opdeling understøttet af en afvekslende og konsekvent typografi, der letter tilegnelsen af indholdet og forhøjer bogens værdi som opslagsbog.

COMAL BRUGER NYT

Som det sig hør og bør i en begynderbog, gives der en grundig indføring i de mest elementære begreber og fremgangsmåder.

Der gås langsomt til værks, og hyppige gentagelser bidrager til, at stoffet sætter sig hos brugeren.

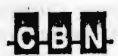
Programkæmpler kan tages enten fra diskette eller fra programhæfte, der begge købes separat.

Der lægges overalt op til elevernes selvstændige fremstilling af programmer. En god hjælp er det, at en bog med vejledende løsninger hertil kan fås.

Endelig er der selvprøver knyttet til hvert afsnit, og de er udført med sådan, at den læser, der kan "bestå" dem, må siges at have til-egnet sig indholdet af den pågældende stof.

Bogen er præget af forfatterens sikre pædagogiske håndlag, systematik og klarhed i udtryk og fremstilling.

S. Chr. H.



Små tips

LOAD, RUN eller DELETE fra CAT

En snild måde at flytte et program fra diskstation til computer er følgende:

Ist CAT.
Når diskettens indholdsfortegnelse står på skærmen, flyttes cursoren op på den linie, hvor det ønskede program står anført.

Ist LOAD hen over angivelsen af antal blokke, der står lige foran programnavnet. Der er god plads til de fire bogstaver.

Tryk nu RETURN.
Cursoren springer hen til bogstaverne PRG, og der kommer en fejlmelding: SYNTAX ERROR.

Den neqligerer du og trykker tre gange på SPACE, så bogstaverne PRG slettes.

Tryk nu igen på RETURN, så bliver programmet loadet og er klar til at blive udført med RUN.

Du kan også gå direkte til udførelsen ved at taste RUN, hvor du for tastede LOAD. Men her er du nødt til at se efter, at alle cifrene i blokantallet er dækket af bogstaverne RUN (evt. + SPACE), for er de ikke det, så virker "fidusen" ikke.

Fremgangsmåden er i øvrigt helt den samme, og du er straks igang med programmet, der dog på skærmen muligvis bliver skrevet oven i resten af kataloget, hvis programmet ikke indledes med et "slet indholdet af skærmen": PRINT CHR\$(147).

På ganske tilsvarende måde kan man i kataloget slette et program af disketten:

Ud for navnet på det program, der skal slettes, taster man DELETE; men dette ord fylder så meget, at man er nødt til at skaffe plads til det ved først at taste SHIFT/INSERT nogle gange.

Foruden DELETE skal der også foran programnavnet tages mellemrum (SPACE) og "0":
"0: skal der testes, hvis disketten befinder sig i diskstationens port 0. Ellers skal der testes "1:

Derefter skal der som sædvanlig trykkes på RETURN, og PRG skal slettes med SPACE.

Efter fornyet tryk på RETURN, kører disketten, og programmet slettes.

COMAL BRUGER NYT

Spørgrebrevkassen

Når man skal angive et kronebeløb over 1000 kr. (jvfr. bankkonto-udskrifter), bruger man ofte både komma og punktum.

Eks.: 123.456,78 kr.

Hvordan programmerer man sig ud af det problem, at et format ikke kan indeholde både . og , ?

ET

Svar:

Forskellige har givet deres løsning på problemet.

Her er foreløbig to forskellige svar.

A)

```
0010 tal:=123.456.78
```

```
0020
```

```
0030 PRINT USING "###.###," : INT(tal)/1000;
```

```
0040 PRINT USING "£": ((INT(tal*10)/10)-INT(tal))*10;
```

```
0050 PRINT USING "£": (tal*10-INT(tal*10))*10
```

Bemærk: Printereren skriver fejlagtigt £ i stedet for nummertegnet. I slutningen af linierne 30 og 40 skal ";" i visse COMAL versioner erstattes med "," for at undgå afstand mellem cifrene for ørebeløbet.

B)

Nedenstående program kan f. eks. benyttes til på en linie at skrive

1) en tekst (se linie 1000)

2) et tal opdelt i treciffergrupper og placeret i den kolonne, man ønsker det - Det er de to sidste cifre (dem efter kommaet) i tallet i linie 1010, der bestemmer placeringen af ørebeløbets sidste ciffer.

3) en tekst efter tallet (se linie 1020)

Bemærk:

1) Kan udelades.

2) Tallet kan altså stå hvor som helst på linien, blot der er plads.

3) Kan udelades.

COMAL BRUGER NYT

```

0010 // skriv trecifret
0020 //
0030 PROC skriv(beløb,højre) CLOSED
0040 //
0050 // Udskrivning af pengebeløb,
0060 // afrundet til nærmeste fem-øre,
0070 // med højrejustering. Der kan
0080 // skrives videre hen ad linien
0090 // fra position 'højre'.
0100 //
0110 // beløb : Pengebeløb, der
0120 // skal udskrives.
0130 // højre : højremargen.
0140 //
0150 // Eksempel: PRINT "Beløb:"
0160 // EXEC skriv(34047.33,20)
0170 // PRINT "kr"
0180 //
0190 // Resultat: "Beløb: 34.047,35 kr"
0200 // kolonne: 1234567890123456789012
0210 // højre=20:
0220 //
0230 DIM res$ OF 20
0240 a:=ABS(beløb+.025) // afrund
0250 kroner:=INT(a)
0260 ører:=INT((a-INT(a))*20)*5
0270 REPEAT
0280 tusind:=kroner MOD 1000
0290 kroner:=kroner DIV 1000
0300 IF kroner=0 THEN
0310 res$:=STR$(tusind)+res$
0320 ELSE
0330 res$:=STR$(tusind+1000)+res$
0340 res$(1:1):="."
0350 ENDIF
0360 UNTIL kroner=0
0370 IF beløb<0 THEN res$:="-"+res$
0380 PRINT TAB(højre-LEN(res$)-3),res$,
0390 res$:=STR$(ører+100)
0400 PRINT "+res$(2:3),
0410 ENDPROC skriv
0420
1000 PRINT "Indsat:"
1010 skriv(12345.67,47)
1020 PRINT "kr."
1030 PRINT "indsat:"
1040 skriv(789123.45,47)
1050 PRINT "kr."

```

Indmeldelsesblanket på bagsiden. Må gerne kopieres!

COMAL BRUGER NYT

Små tips

PROGRAMHOVED

I arbejdet med datamater drejer det sig om at tilrettelægge programmerne sådan, at de giver størst mulig lettelse i arbejdet og størst mulig sikkerhed i anvendelsen.

En hjælp i den retning kan det være at indlede ethvert program med 4 linier, som man "standardiserer" efter sit eget hoved.

Her er et forslag:

```
0010 // SAVE "$0:25.VENTE"
0020 // CBM COMAL 80 REV 01.02
      DISK 1N/24
0030 // VERIFY "25.VENTE"
```

I linie 10 anvendes ideen omtalt i et tidligere tip; men der er føjet et nummer til programnavnet.

Fordelen derved er, at man nu ganske let kan kalde sit program:

```
LOAD "25 *
RETURN
```

Så kører disketten, og programmet læses ind i maskinens hukommelse klar til benyttelse med

```
RUN
```

Ja, man kan endog slippe endnu nemmere om ved det:

```
RUN "25 *
RETURN
```

Så begynder programudførelsen straks efter indlæsningen.

Ingen ventetid, ingen pauser!

Men det forudsætter naturligvis, at hvert program på den pågældende diskette har sit eget nummer, ellers vælger maskinen det program med et bestemt nummer, der står først i kataloget.

Stjernen (asterik'en = *) efter programnummeret erstatter al den øvrige tekst, der karakteriserer programnavnet.

I linie 20 anføres helt præcist hvilket sprog der er anvendt, og hvilken diskette programmet står på.

Det første har betydning navnlig for andre, der eventuelt kommer til at benytte programmet.

Diskettebetegnelsen og programnummeret har interesse for ejeren, der en anden gang skal finde programmet frem ud fra en printerudskrift af list'en eller ud fra et kartotek over programmer.

Linie 30 benyttes, når man ønsker, at maskinen skal undersøge, om der er overensstemmelse mellem det program, der ligger på disken, og det program af samme navn, der er i maskinen.

Bemærk, at linierne 10 - 30 er såkaldte kommentarsætninger, der som bekendt slet ikke kommer med i udførelsen af programmet.

De anvendes således:

Medens list'en - eller i hvert fald de 3 første linier - står på skærmen, flyttes cursoren op på linie 10.

Med SPACE slettes 0010 //, så der nu står:

```
SAVE "$0:25.vente"
```

Tryk RETURN.

På disketten i port 0 gemmes nu hele programmet uanset, om det står der allerede under samme programbetegnelse (se tidligere tip herom).

Efter endt "save'ing" går cursoren automatisk 2 linier ned og havner altså på linie 30.

COMAL BRUGER NYT

VERIFY "25.vente"

Tryk RETURN.

Disketten kører, og indspilningen sammenlignes med indholdet af maskinens hukommelse.

Hvis de er ens, kommer cursoren frem 2 linier længere nede og står og venter på næste ordre.

Hvis der er uoverensstemmelse mellem versionerne på disketten og i maskinen, kommer der en fejlmeddelelse på skærmen.

Hvis man nu vil sikre sig, at indspilningen er i orden, bruger man SPACE til at slette 0030 //, hvorefter der står:

C-B-N

C-B-N

Hold styr på disketterne

Vi er blevet spurgt om, hvordan man forholder sig, hvis man vil lave mere end en samling (bibliotek) af disketter - f. eks. med forskelligt emne.

Svaret er, at man må overspille programmet på en ny diskette og indspille sit andet bibliotek på denne.

Forholdet er jo det, at man har programmet og "biblioteket" liggende på disketten i port 0, medens man indspiller ved i port 1 efter tur at anbringe de forskellige disketter, hvis katalog skal overspilles.

En sikker input-rutine skrevet i COMAL

Når man skal foretage input fra tastaturet, benytter man som regel en INPUT-sætning. Det er udmærket så længe kun programmøren bruger programmet med INPUT-sætningen. I det øjeblik andre skal bruge programmet, bør man forlange at man ikke kan indtaste ulovlige tegn (f.eks. trykke på clear-tasten, således at skærbilledet slettes). INPUT-sætningen i COMAL indeholder ikke mulighed for at angive at specielle tegn skal ignoreres, eller skal virke på andre måder end de plejer, f.eks. at clear-tasten kun skal slette det felt af skærmen hvori inputtet udskrives.

For at komme ud over INPUT-sætningens begrænsninger, må man selv skrive procedurer, der læser et tegn ind ad gangen, undersøger hvilket tegn det er, og endelig reagerer på dette tegn (ved f.eks. at udskrive det på skærmen). Jeg har forsøgt at lave tre procedurer, der opfylder nogle af de krav, man kan stille til tastaturinput.

Procedurerne har følgende faciliteter:

Det indtastede udskrives i et felt af en ganske bestemt længde;

COMAL BRUGER NYT

man kan ikke på nogen måde bevæge sig udenfor dette felt;

man kan på ethvert tidspunkt få skærbilledet udskrevet på printer;

man kan editere i inputfeltet.

Hvis man bruger INPUT-sætninger af følgende form:

```
INPUT "Indtast dit navn: ": navn$
INPUT "Hvad er din alder ? ": alder
```

skal disse nu skrives således:

```
EXEC indstr("Indtast dit navn: ",navn$,20)
EXEC indtal("Hvad er din alder ? ",alder)
```

Når man skal indtaste en streng, skal proceduren `indstr` anvendes:

```
PROC indstr(ledetekst$,REF streng$,længde) CLOSED
// ledetekst$: Tekst, der skal udskrives
//           før input begynder.
// streng$   : Den variabel, som den
//           indtastede tekst skal
//           anbringes i.
// længde    : Maksimal længde af den
//           streng, der skal indtastes.
//
// Alle store og små bogstaver, samt cifre
// og specialtegn kan indtastes.
```

Når man skal indtaste et tal, skal proceduren `indtal` anvendes:

```
PROC indtal(ledetekst$,REF tal) CLOSED
// ledetekst$: Tekst, der skal udskrives
//           før input begynder.
// tal       : Variabel, hvori tallet
//           skal anbringes.
//
// Kun cifre (maksimalt 6 cifre) kan indtastes.
```

Man har flg. editeringsmuligheder:

Inst	Funktion
cursor til venstre	Flyt cursoren en plads mod højre.
cursor til højre	Flyt cursoren en plads mod venstre.
slet tegn	Slet tegn bagved cursor og ryk teksten efter cursoren et tegn mod venstre.

COMAL BRUGER NYT

indsæt tegn	Skub teksten foran cursoren en plads mod højre (hvis der er plads) og skriv et mellemrum på cursorens plads.
return	Afslut input.
home	Gå til begyndelsen af inputfeltet.
clear	Slet indholdet af inputfeltet og gå til begyndelsen af inputfeltet.
nummertegn	Udskriv skærbilledet på printeren

Disse procedurer er skrevet til CBM COMAL 80 og benytter styretegn med værdier specielt for Commodore maskiner. For at kunne flytte procedurerne til en anden COMAL, er alle styretegnene navngivet. Den angivne procedure til udskrift af skærbilledet på printer er også speciel for CBM COMAL 80. Den er muligvis den sværeste procedure at flytte, da RC COMAL80 og Metanic COMAL-80 mig bekendt ikke har faciliteter i COMAL'en til at læse skærmen.

```

0010 PROC INDSTR(LEDETEKST$,REF STRENG$,LÆNGDE) CLOSED
0020   IMPORT LÆSFELT
0030   PRINT LEDETEKST$,
0040   LÆSFELT(STRENG$,LÆNGDE)
0050   PRINT
0060   FUNC PRINTBAR(TEGN)
0070     IF TEGN=34 THEN // ANFØRELSESTEGN
0080       RETURN FALSE
0090     ELIF 32<=TEGN AND TEGN<=95 THEN // LOWER CASE
0100       RETURN TRUE
0110     ELIF 193<=TEGN AND TEGN<=221 THEN // UPPER CASE
0120       RETURN TRUE.
0130     ELSE
0140       RETURN FALSE
0150     ENDIF
0160   ENDFUNC PRINTBAR
0170 ENDPROC INDSTR
0180
0190 PROC INDTAL(LEDETEKST$,REF TAL) CLOSED
0200   IMPORT LÆSFELT
0210   DIM FELT$ OF 6
0220   PRINT LEDETEKST$,
0230   LÆSFELT(FELT$,6)
0240   TAL:=VAL(FELT$)
0250   PRINT
0260   FUNC PRINTBAR(TEGN)
0270     RETURN 48<=TEGN AND TEGN<=57
0280   ENDFUNC PRINTBAR
0290 ENDPROC INDTAL
0300
0310 PROC LÆSFELT(REF TEKST$,LANGDE) CLOSED
0320   IMPORT PRINTBAR
0330
0340   CURSORHØJRE:=29
0350   CURSORVENSTRE:=157
0360   SLETTEGN:=20

```

COMAL BRUGER NYT

```

0370  INDSÆTTEGN:=148
0380  CR:=13
0390  HOME:=19
0400  CLEAR:=147
0410  NUMMERTEGN:=35
0420  MARKØR:=164
0430
0440  DIM MELLEMRUM$ OF LÆNGDE
0450  MELLEMRUM$(1:LÆNGDE):=""
0460  TEKST$:=MELLEMRUM$
0470  I:=1
0480  REPEAT
0490    LÆSTEGN
0500    CASE TEGN OF
0510      WHEN CURSORHØJRE
0520        IF I<LÆNGDE THEN
0530          PRINT CHR$(CURSORHØJRE),
0540          I:=I+1
0550        ENDIF
0560      WHEN CURSORVENSTRE
0570        IF I>1 THEN
0580          I:=I-1
0590          PRINT CHR$(CURSORVENSTRE),
0600        ENDIF
0610      WHEN SLETTEGN
0620        IF I=1 THEN
0630          PRINT TEKST$(I+1:LÆNGDE);
0640        ELSE
0650          PRINT CHR$(CURSORVENSTRE),TEKST$(I:LÆNGDE);
0660          I:=I-1
0670        ENDIF
0680      FOR J:=I TO LÆNGDE DO PRINT CHR$(CURSORVENSTRE),
0690      TEKST$(I:LÆNGDE):=TEKST$(I+1:LÆNGDE)
0700    WHEN INDSÆTTEGN
0710      IF TEKST$(LÆNGDE:LÆNGDE)=" " THEN
0720        TEKST$(I:LÆNGDE):=" "+TEKST$(I:LÆNGDE)
0730        PRINT TEKST$(I:LÆNGDE),
0740        FOR J:=I TO LÆNGDE DO PRINT CHR$(CURSORVENSTRE),
0750      ENDIF
0760    WHEN CR
0770      IF TEKST$=MELLEMRUM$ THEN
0780        FOR J:=2 TO I DO PRINT CHR$(CURSORVENSTRE),
0790        I:=1
0800      ELSE
0810        I:=LÆNGDE
0820        WHILE TEKST$(I:1)=" " DO I:=I-1
0830        TEKST$:=TEKST$(1:I)
0840        RETURN // INDTASTNING FÆRDIG.
0850      ENDIF
0860    WHEN HOME
0870      FOR J:=2 TO I DO PRINT CHR$(CURSORVENSTRE),
0880      I:=1
0890    WHEN CLEAR
0900      FOR J:=2 TO I DO PRINT CHR$(CURSORVENSTRE),
0910      PRINT MELLEMRUM$,
0920      FOR J:=1 TO LÆNGDE DO PRINT CHR$(CURSORVENSTRE),
0930      I:=1; TEKST$:=MELLEMRUM$

```

COMAL BRUGER NYT

```

0940 WHEN NUMMERTEGN
0950     HARDCOPY
0960 OTHERWISE
0970     IF PRINTBAR(TEGN) THEN
0980         PRINT CHR$(TEGN),
0990         TEKST$(1:1):=CHR$(TEGN)
1000         IF I<LÆNGDE THEN
1010             I:=I+1
1020         ELSE
1030             PRINT CHR$(CURSURVENSTRE),
1040             ENDIF
1050         ENDIF
1060     ENDCASE
1070 UNTIL FALSE
1080
1090 PROC LÆSTEGN
1100     TIMER:=0
1110     REPEAT // BLINK MED CURSOREN
1120         IF TIMER=0 THEN
1130             PRINT CHR$(MARKØR)+CHR$(CURSORVENSTRE),
1140             ELIF TIMER=50 THEN
1150                 PRINT TEKST$(1:1)+CHR$(CURSURVENSTRE),
1160                 ELIF TIMER=100 THEN
1170                     TIMER:=-1
1180                 ENDIF
1190                 TIMER:=TIMER+1
1200                 TEGN:=ORD(KEY$)
1210             UNTIL TEGN<>0
1220             PRINT TEKST$(1:1)+CHR$(CURSORVENSTRE),
1230         ENDPROC LÆSTEGN
1240
1250 PROC HARDCOPY GLOED
1260     IMPORT CR,CLEAR
1270     SKÆRMVIDDE:=40 // 80 FOR 80-TEGNS SKÆRM
1280     DIM LINIE$ OF SKÆRMVIDDE
1290     LINIENR:=PEEK(216)+1 // GEM CURSOR POSITION.
1300     SØJLENR:=PEEK(198)+1
1310     CURSOR 1,1
1320     OPEN FILE 2,"",UNIT 3 // ÅBEN SKÆRMEN SOM FIL.
1330     SELECT OUTPUT "LP" // SKRIV PÅ LINIESKRIVER.
1340     FOR I:=1 TO 24 DO // LÆS & SKRIV SKÆRMEN LINIEVIS.
1350         LINIE$:=GET$(2,SKÆRMVIDDE)
1360         IF LINIE$(SKÆRMVIDDE:SKÆRMVIDDE)<>CHR$(CR) THEN
1370             PRINT LINIE$
1380         ELSE
1390             PRINT LINIE$(1:SKÆRMVIDDE-1)
1400         ENDIF
1410     ENDFØR I
1420     PRINT GET$(2,SKÆRMVIDDE-1)
1430     PRINT CHR$(CLEAR) // SKIP TIL NÆSTE SIDE.
1440     SELECT OUTPUT "DS"
1450     CLOSE FILE 2
1460     CURSOR LINIENR,SØJLENR // SAT CURSOREN IGEN.
1470     ENDPROC HARDCOPY
1480
1490 ENDPROC LÆSFELT

```

COMAL BRUGER NYT

C B G
COMAL BRUGER GRUPPE
MINDEGADE 42
8700 HORSSENS

tlf 05 - 62 15 67
giro 1 75 23 75

I N D M E L D E L S E I C B G

Indmeldelse foretages ved udfyldelse af nedenstående blanket, der sendes til ovenstående adresse.

Samtidig indbetales kr. 90,00 - prisen for 1 års abonnement på "COMAL BRUGER NYT", der agtes udgivet med 6 numre pr. år.

Indbetaling kan ske pr. check eller ved overførsel til:

Giro nr. 1 75 23 75, COMAL BRUGER GRUPPE, Mindegade 42, 8700 Horsens.

```

*****
*
*   Undertegnede
*
*   Navn:.....
*
*   Adresse:
*   Gade:.....
*
*
*   Postnummer:.....   By:.....
*
*   Indmelder sig i COMAL BRUGER GRUPPE.
*
*   Kr. 90,00 vedlægges i check ....
*
*           indbetales pr. giro ....
*
*
*****

```

Sæt venligst X

Venlig hilsen

COMAL BRUGER GRUPPE

S. Chr. Hansen

