

R

GODT NYTÅR

Ser man tilbage på det korte spand af år, EDB sproget COMAL har eksisteret - jvfr. omstående artikel af Børge Christensen - må man forbavses over den usædvanlig hurtige udbredelse, det har opnået.

Man kan endvidere konstatere, at det rent faktisk er blevet undervisningssprog på datamater i Danmark og er godt på vej til at blive det også i andre lande.

Arsagen dertil er jo imidlertid helt enkelt, at COMAL foruden at være let at lære også er let at overskue, let at rette og ændre, og desuden er det let at udvide med nye programdetaljer. Dertil må føjes, at COMAL er et så kraftigt sprog, at det kan anvendes langt uden for undervisningssektoren til bl. a. administrative formål og til styring af diverse teknik. Summa summarum: Det er ikke så mærkeligt, at COMAL "har haft vind i sejlene".

Men det skal dog medgives, at succesen i høj grad skyldes entusiasme hos COMAL pionererne. Uden den var resultaterne næppe nået i det tempo, hvori det faktisk er sket.

Den samme begejstring for COMAL ligger til grund for dannelsen af COMAL BRUGER GRUPPE. Et antal af de mange, der daglig anvender COMAL, er fundet sammen om den opgave at værne om COMAL.

Det lyder nok noget højtravende: "at værne om COMAL"; men ikke desto mindre må det siges at være et relevant udtryk, al den stund der findes afskrækkende eksempler på, hvordan det kan gå, hvis et EDB sprog får lov at udvikle sig aldeles frit.

Universitetsbiblioteket, 2. afd.

0108-4925

THE DANISH GAZETTE FOR COMAL LANGUAGE

INDHOLDSFORTEGNELSE

Leder: Godt nytår	1
Programmeringssprog	2
Små tips	4 og 8
Definitionsgruppen	5
Radioudsendelse om COMAL 80 ..	5
Løserbreve	5
Rekursive func og proc	6
Velkommen Comal 80	7
Spørgrebrevkassen	8
Armeldelser	9 og 10
Redaktionelt	10
Hold styr på disketterne	11 til 16

200002999373



COMAL BRUGER NYT

Værdien af et EDB sprog er blandt andet afhængig af mulighederne for dets benyttelse på forskellige datamaskiner.

Det ideelle ville være, om programmer fremstillet på en maskine var umiddelbart anvendelige på enhver anden.

Desværre må tanken om et sådan idealsprog nok kaldes utopisk, fordi der er så store forskelle i de maskinbestemte krav til sproget; men er man tvunget til at acceptere dette, så burde også dermed ulighederne være udtømt. Sprogets syntaks og terminologi burde være hævet over maskinfabrikaterne og være ens for alle.

Med de tiltag, der allerede er sket i retning af, at de forskellige COMAL udbydere skal tilnærme sig samme COMAL kerne og fremtidig aftale de enkelte skridt i sprogets fortsatte udvikling, er der gjort en god begyndelse, som alle brugere må slutte tæt op om. Vi må støtte disse bestræbelser og påtale, hvor vi finder, at man ikke følger intentionerne. Vi vil også kunne udtrykke ønsker om forbedringer og tilføjelser ved kommende revisioner af COMAL.

Sådanne opgaver er simpelthen COMAL BRUGER GRUPPENS vigtigste.

Vi appellerer til alle medlemmerne om at være med til at løse disse opgaver.

Vor medlemsflok er sammensat af folk, der er eksperter i programmering og i anvendelse af COMAL, af folk, der er helt nybegyndere, og af folk af enhver grad af indsigt i COMAL mellem disse to yderpunkter. Alle er de lige vigtige, når talen er om at medvirke til, at forholdene vedrørende COMAL bliver så gode som muligt.

COMAL BRUGER NYT henvender sig til alle, der beskæftiger sig med COMAL.

For mange bliver vi forhåbentlig en støtte ved at bringe stof, der udvider den enkeltes viden og kunnen om COMAL; men en vigtig forudsætning herfor er, at brugerne kommer frem med deres ønsker. Derfor har vi en læserbrevkasse og en spørgebrevkasse. Derudover vil enhver henvendelse til redaktionen med ønsker og forslag være særdeles velkomne.

For andre vil behandlingen af de mere enkle problemer næppe have samme interesse. Vi satser imidlertid på, at der også skal blive stof af værdi for sådanne læsere; men her må vi igen bede om, at man tilkendegiver, hvad man gerne ser behandlet i CBN. Og så håber og satser vi jo i høj grad på, at mange læsere selv vil være i stand til og føle sig inspireret til direkte at yde bidrag til bladets spalter.

Med dannelsen af COMAL BRUGER GRUPPE og udgivelsen af COMAL BRUGER NYT er der skabt noget, som vi følte, at mange savnede og kunne bruge, når først det var der.

Nu er det der. Brug det. Lad det vokse og udvikle sig. Lad det også for COMAL i enhver henseende blive et

G O D T N Y T A R I

S. Chr. H.

C-B-N

COMAL BRUGER NYT
ISSN 0108-4925

Udgiver:
C B G
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

PROGRAMMERINGSSPROG OG DEN GLOBALE UDFORDRING

Første gang, jeg hørte ordet "basic", var i 1946. Jeg traf en af de mange sømand, som var kommet hjem fra England efter den tyske besættelse af Danmark, og som i mellemtiden havde gjort tjeneste i "The Royal Navy". På det tidspunkt, hvor jeg mødte ham, havde jeg lært engelsk i om trent et år, og jeg var ivrig efter at få at vide, hvordan det var sådan at komme til England uden videre. Hvordan med sproget? Kunne han tale engelsk, da han kom over? Eller hvordan fik han det lært? Han svarede på mine mange spørgsmål ved at fortælle mig, at de alle sammen fik tilbudt et kursus i noget, der hed "Basic English". Da jeg spurgte ham, hvordan det var, gav han mig et svar, som jeg ikke rigtigt forstod: "Det var en luder af et sprog. Det kunne bruges til at dække et øjeblikkeligt behov, men der var ikke nogen virkelig glæde ved det".

Kort efter kom ordet ind i mit engelske ordforråd i sin sædvanlige betydning, og det fortsatte derefter som et hvilket som helst andet almindeligt, brugbart og anstændigt engelsk ord indtil 1971. På det tidspunkt blev en minidatamat installeret i matematikafdelingen på seminariet. En RC-7000. Et nyt mirakel. Vi var alle meget stolte af den og det blev endog nævnt i "Vestkysten" på lokalsiderne. Jeg havde taget et Algol-kursus på universitet (i Århus sømand.) og blev derfor betragtet som den lokale ekspert på området - der var heldigvis ingen der vidste, at det havde været et meget kort kursus, og at jeg næsten intet kunne huske om hverken Algol eller datamater -. Men jeg blev altså bedt om at tage mig af den "nye indretning", og det blev betonet, at det ikke ville være af vejen, hvis nogle af de studerende skulle få lært at bruge den. Og så var det, jeg mødte ordet "BASIC" i en helt ny sammenhæng. Og det blev stavet med STORE BOGSTAVER.

Efter et år med flere "kilometer" BASIC-programmer blev jeg klar over, at der var noget galt med dette programmeringssprog. De studerende, der overhovedet kom i nærheden af minien, lavede masser af fejl i programmerne, og det var ofte meget svært at finde disse fejl og det hele taget finde ud af, hvad der forgik i programmet. I

begyndelsen var jeg tilbøjelig til at mene, at hele miseren skyldtes min mangel på grundig viden, men efterhånden vovede jeg at antage, at dette nye vidunder af et sprog - som oven i købet kom fra STATERNE - måske havde visse mangler. Min mistanke blev bekræftet, da jeg begyndte at diskutere sagen med nogle af mine kolleger på Datalogisk Institut ved Århus Universitet. De sagde lige ud, at BASIC var en skandale, og at dets popularitet var aldeles ufortjent og kun skyldtes den kendsgerning, at man i almindelighed ikke havde andre sprog til rådighed på datamater, der blev benyttet ved elementær undervisning.

En af forskerne ved universitetet, Benedict Løfstedt, fattede interesse for mit problem - det er ofte meget vanskeligt for en lærer at vække interesse på et universitet for undervisningsproblemer - og rådede mig til at læse en bog, som netop var udkommet. Bogens titel er "Systematic Programming", og den er skrevet af en vis hr. Niklaus Wirth.

Jeg købte bogen og begyndte at læse den, og efter en måneds tid vidste jeg, at Århus-folkene havde ret. BASIC er en skandale og havde for en stor del skylden for, at der blev skrevet så dårlige programmer, ikke blot hos os, men også på de fleste andre uddannelsessteder indenfor området. Men hvad skulle vi gøre? Pascal - det sprog, der bliver fremstillet i "Systematic Programming" - var kun implementeret på nogle få store datamater og blev næsten udelukkende anvendt i akademiske kredse. Dengang var der ikke ret mange, som kunne forudse den status, Pascal har i dag. Og når alt kommer til alt er det ikke blot sproget, det kommer an på. Sprogets såkaldte "bruger-omgivelser" er også af afgørende betydning. Det, man nutildags kalder "its Human Interface". Det BASIC-system, som blev brugt på vor minidatamat var helt interaktivt og blev støttet af et udmærket flerbrugersystem; det indbefattede desuden et meget anvendeligt fil-håndterings-system, og vi kunne ikke klare os uden alle disse ting.

Efter at have overvejet alt dette vendte jeg tilbage til Benedict Løfstedt, og han forslog nu, at vi skulle forsøge at beskrive nogle få, men slagkraftige udvidelser til det sprog og system, vi havde, og at vi skulle bruge så mange af ideerne fra Pascal som muligt. Det var indlysende,

COMAL BRUGER NYT

hvad der var mest påtrængende: LANGE variabelnavne, en global IF-THEN-ELSE struktur (ikke den tåbelige een-linies IF-THEN-ELSE, som findes i enhver fjollet BASIC nutildags), REPEAT- og WHILE-løkke, en flerforgrenings-CASE struktur og navngivne underprogrammer. I løbet af de følgende seks måneder blev beskrivelsen gjort færdig. Jeg opfandt ordet COMAL (COMMon Algorithmic Language) som navn for det nye sprog. I juni 1974 begyndte Per Christensen at implementere de udvidelser, vi havde beskrevet, på RC7000-datamaten her i Tønder. Senere kom Knud Christensen til, og i februar 1975 var vi i stand til at skrive programmer som fx følgende:

```
IF FORSØG<3 THEN
PRINT "NEJ, PRØV IGEN."
ELSE
PRINT "NEJ, DET RIGTIGE SVAR ER";
RESULTAT
PRINT "SKRIV DET."
ENDIF
```

Jeg har valgt dette eksempel, fordi den viste IF-THEN-ELSE struktur var den første, der hurtigt blev populær blandt brugerne. De første versioner af COMAL tillod ganske vist kun, at IF-THEN-ELSE forgreninger blev indlejret til en dybde af fire, men det er meget sjældent, man har brug for mere, og det er meget lettere at gøre det på COMAL-måden end ved at bruge GOTO-sætninger. Man behøver ikke gå længere end til en dybde af to i BASIC-programmer for at få noget, som er svært at læse, og de fleste begyndere går vild i den labyrint af BASIC-sætninger, som man har brug for ved mere end tre forgreninger inden i hinanden. Jeg skylder at nævne, at den allerførste COMAL-version selv udførte de indrykninger, som er vist i eksemplet.

COMAL fik større betydning, end vi havde troet muligt. De studerende skriver meget bedre programmer i COMAL, end de plejede at gøre i BASIC, og - hvad der ikke er mindst vigtigt - deres programmer er blevet læselige. I begyndelsen blev vore ideer afvist af mange lærere, og det var i almindelighed studerende og skoleelever, som var hurtigst til at fatte, at de havde fået et godt værktøj i hånden. Det skete ikke så sjældent, når jeg var ude at holde foredrag om COMAL, at jeg fik det indtryk, at vi havde startet en slags religionskrig, og ikke blot opfundet et sæt nyttige udvidelser til et programmeringsprog, som man kunne diskutere under begreberne nytte

og effektivitet. Efterhånden ændrede holdningen sig dog til fordel for vore ideer, og siden 1977/78 er det blevet meget vanskeligt at sælge en datamat til undervisningsbrug, hvis den kun tilbyder BASIC.

I 1979 beskrev jeg en ny version af COMAL, beregnet til at blive implementeret på en ny mikrodatamat, som samtidig blev konstrueret af akademiingeniør Mogens Pelle. Denne version - COMAL 80 - er senere blevet yderligere udbygget og indbefatter nu sådanne faciliteter som parameteroverføring (både værdiparametre og henvisningsparametre), lokale variable og rekursive procedurer. COMAL 80 findes nu på flere mikrodatamater, fx Comet 3000, RC Piccolo, DDEs SPC/1 og samtlige Commodore mikrodatamater. På sidstnævnte type er COMAL 80 for nylig blevet udvidet med en række grafiske kommandoer, som udvider brugen af sproget i betydeligt omfang, og det kan sikkert ventes, at lignende udvidelser vil blive implementeret på andre COMAL-maskiner. Det ser endvidere ud til, at der nu kommer COMAL-versioner fra udenlandske firmaer, og en helt ny COMAL 80, som for tiden bliver skrevet til Commodore 64, skulle sammen med de bestræbelser, som firmaet Metanic udfolder for at sælge sproget, gøre det kendt for en stor kreds af udenlandske brugere.

Børge R. Christensen

C-B-N

SMÅ TIPS

Stop og start af "listning"

Når LIST'en ruller hen over skærmen, er det rart at kunne standse rulningen. Det kan man selvfølgelig med STOP tangenten. Så kan man jo imidlertid ikke fortsætte ud den at indtaste ordren LIST igen.

Det kan man kan slippe for ved at trykke på SPACE tangenten. Så standser listningen, og den kan nu sættes igang igen ved et nyt tryk på SPACE.

Hvis man efter standsningen med SPACE vil gå ind i en programlinje for at udføre en rettelser, må man dog først trykke på STOP.

En påfølgende ny listning er som ovenfor nævnt kun mulig ved indtastning af LIST.

COMAL BRUGER NYT

DEFINITIONSGRUPPEN FOR COMAL 80

Der har været afholdt møde 10. maj på Tønder Statsseminarium vedrørende fastlæggelse af den endelige Comalsyntax. Deltagerne var Børge Christensen (grundlæggeren af Comal), Coleen Kitchen og Kevin Ryan (begge fra Dublin Universitet, Irland), Len Lensay (Comal User Group, Madison, Wisconsin, USA), Jørgen Olsen og Erik Jepsen (begge fra a/s Regnecentralen, København), Arne Christensen og Mogens Pelle (begge fra Metanic Aps, København), Mogens Kjær, Lars Laursen og Jens Erik Jensen (tre repræsentanter fra den gruppe, der har implementeret Comal til Commodore Computere).

Man vedtog blandt andet at ændre syntaxen (regler for sprogets sætningsopbygning) for funktioner i Comal, så det også bliver muligt at implementere funktioner, der kan returnere en tekststreng, det vil sige, at det nu i Comal er muligt at definere reele, heltals- og strengfunktioner. Desuden vedtog man, at ændre syntaxen for udtagning af delstreng. Tidligere brugte man fx. `b$:= a$(fra:længde)`. Fremover vil det hedde `b$:= a$(fra:til)`. Det blev ligeledes bestemt at GLOBAL skal ændre navn til IMPORT.

Man udarbejdede også et forslag, der beskriver syntaxen for fremtidige udvidelser.

Næste møde i definitionsgruppen afholdes tidligt dette år, hvor man bl.a. vil diskutere, hvorledes et fornuftigt pakkebegreb skal implementeres i Comal (ved pakkebegreb forstås man en programdel, der bliver liggende i maskinens lager, selv om hovedprogrammet udskiftes. En af egenskaberne ved pakker er, at de har egne variable, procedurer og funktioner).

Man vil også diskutere indførelse af en error handler struktur. Denne facilitet muliggør, at man kan fange runtime fejl - altså fejl der opstår under kørslen. Eksempelvis kan nævnes den kendte situation, hvor divisor i en division antager værdien 0. Som det er nu, meddeler systemet, at division med 0 er opstået og derefter stoppes programudførelsen. Med en error handler får man mulighed for at reagere passende på en sådan fejl.

RADIOUDSENDELSE OM COMAL 80

Engang i marts eller april sendes en 25 minutters udsendelse over program I, Danmarks Radio. Det endelige sendetidspunkt er endnu ikke fastlagt.

Børge R. Christensen, Tønder Statsseminarium, gør rede for en række interessante forhold mennesker og computer imellem. Blandt andet taler han om et nyt begreb: computeren som menneskets sprogforlænger.

Han forudsiger en udvikling, der vil nødvendiggøre en stærk kritisk holdning til computere.

En kritisk holdning forudsætter imidlertid, at menneskene kender disse maskiners muligheder, og Børge Christensen slår derfor til lyd for en omfattende undervisning af skoleelever på et tidligt alderstrin.

Han understreger, at EDB sprog skal udformes, så de bliver lettere at forstå mellem mennesker indbyrdes end mellem mennesker og maskiner.

Dette fører naturligt over i omtalen af COMAL 80, der netop opfylder et sådant krav.

Udsendelsen, der er tilrettelagt af Frank Hejndorf, hedder "Mennesker og EDB".

fh



LÆSERBREV

Denne rubrik står åben for indlæg af enhver art fra læserne.

Specielt er vi interesseret i læsernes mening om bladets indhold og i forslag til forbedringer og nye emner.

COMAL BRUGER NYT

REKURSIVE FUNKTIONER

OG PROCEDURER

En funktion eller en procedure kan i COMAL 80 kalde sig selv direkte. Det vil f. eks. sige, at proceduren A inde i sig selv har et kald til A.

En funktion eller en procedure kan også kalde sig selv indirekte via andre funktioner eller procedurer. F. eks. kan A kalde B, og B kan kalde A.

Mange problemer kan løses elegant ved anvendelse af rekursion; men ofte vil iteration kunne anvendes med fordel - herom senere.

I det følgende vil to små demonstrationsprogrammer blive anvendt for at vise brugen af rekursion.

Eks. 1.

```
0100 DIM A$ OF 20
0110 INTEGER LENGTH // ikke på alle
systemer
0120 //
0130 PROC DUMMY(REF B$)
0140 IF B$(LENGTH)=" " THEN
0150 LENGTH:=LENGTH-1
0160 EXEC DUMMY(B$)
0170 ENDIF
0180 B$:=B$(1:LENGTH)
0190 ENDPROC DUMMY
0200 A$:="REKURSION "
0210 PRINT"TEKSTSTRENGEN ";A$;" er.
";LEN(A$);" karakterer lang"
0220 PRINT"før vi går til proceduren
DUMMY"
0230 PRINT
0240 LENGTH:=LEN(A$)
0250 EXEC DUMMY(A$)
0260 PRINT"Tekststrengen ";A$;" er nu
";LENGTH;" karakterer lang"
0270 END
```

Dette lille eksempel på, hvorledes rekursion kan anvendes, skal ikke tages for andet end netop et eksempel. Hvis man vil fjerne efterstillede blanktegn fra en tekststreng, har man enklere metoder til sin rådighed.

Programmet fjerner altså blanktegn, der måtte være efterstillet en given tekststreng, og det sker i proceduren DUMMY ved anvendelse af rekursive kald.

Som det kan ses, vil DUMMY kalde sig selv, så længe betingelsen i IF-sætningen er sand.

Linierne 200 - 260 giver en visuel fremstilling af, hvad der sker.

Som nævnt vil mange situationer, hvor rekursion frister til anvendelse, kunne klares ved brug af iteration, hvilket må anbefales.

Man kan meget nemt løbe ind i problemer, hvis man forsøger at oversætte rekursive Pascal-programmer til COMAL 80.

I det foregående lille eksempel ville det være enklere at anvende enten en REPEAT- eller en WHILE-sætning.

F. eks.

```
PROC DUMMY(REF B$,RELENGTH)
WHILE B$(LENGTH)=" " DO
LENGTH:=LENGTH-1
B$:=B$(1:LENGTH)
ENDPROC DUMMY
```

Det næste eksempel viser et rekursivt kald af en procedure, der befinder sig inden i en anden procedure.

Eks. 2.

```
0100 INTERGER I // ikke nødvendig i alle
systemer
0110 //
0120 PROC A
0130 FOR I:=1 TO 94 DO
0140 PRINT CHR$(I+32);" ";
0150 EXEC B
0160 NEXT I // ENDFOR i CBM-COMAL 80
0170 PROC B
0180 IF I MOD 10 OR I=94 THEN PRINT
0190 ENDPROC B
0200 PRINT"FÆRDIG."
0210 ENDPROC A
0220 //
0230 EXEC A
0240 END
```

Hver gang FOR/NEXT-sløjfen har kørt et step, sendes I til test i procedure B. Hvis I er blevet forøget med 10, vil B foretage et linieskift, og når I har nået sin slutværdi, skiftes der ligeledes linie.

Også i dette tilfælde kunne problemet være løst uden procedure B. IF-sætningen kunne ganske enkelt placeres efter linie 140; men som nævnt er dette kun eksempler, og pladsen tillader ikke større programafsnit.

VELKOMMEN COMAL 80

Der er al mulig grund til at sige velkommen til COMAL 80.

Ganske vist har COMAL 80 været tilgængelig på nogle enkelte maskiner i et par år; men det er tilsyneladende nu, at en større udbredelse af sproget vil komme.

Alene den omstændighed, at COMAL BRUGER GRUPPEN er dannet, viser noget om den stigende interesse.

COMAL er imidlertid ikke noget ganske nyt. Sproget kan snart fejre sin 10 års fødselsdag, idet historien fortæller, at det var i 1973, at Benedict Løfsted og Børge Christensen konstruerede COMMON Algorithmic Language.

I følge Børge Christensen lavede man COMAL, fordi eleverne, der skulle lære PASCAL, gav sig til at programmere i BASIC, så snart læreren vendte ryggen til dem.

Grunden til denne adfærd skal sikkert søges i to forhold.

Det ene drejer sig om PASCAL's store krav til disciplin samt den noget besværlige vej, der er fra kildetekst til kørende program.

Det andet er BASIC's store tilgængelighed.

Ved konstruktion af COMAL blev det tilstræbt at implementere så mange af PASCAL's gode sider som muligt, samtidig med at man holdt sig for øje at få en tilgængelighed så god som BASIC's.

Det må siges, at resultatet blev vellykket, og allerede fra starten var der da også en vis interesse for sproget. Det blev anvendt på en del læreanstalter, og i hvert fald et par danske fabrikker kunne levere maskiner, der kørte COMAL. Den mest kendte - og vist nok også den første - var Dansk Data Elektronik's ID-7000, der senere udvikledes til den kendte SPC/1.

Imidlertid udviklede COMAL sig noget ukontrolleret med det til følge, at man nærmest kunne tale om, at der opstod COMAL-dialekter.

For at få skrevet en standard for COMAL, der nu skulle hedde COMAL 80, besluttede en kreds af interesserede fabrikanter, leverandører, undervisere og andre i oktober 1979 at nedsætte en arbejdsgruppe, hvis opgave blev at give en beskrivelse af sprogets kerne, komme med evt. forslag til udvidelser samt forslag til et filsystem.

Allerede i foråret 1980 kunne gruppen skrive følgende tre rapporter:

1. COMAL 80, Programming Language-proposal. 25.03.1980

2. COMAL 80, Recommendation for Extensions. 10.04.1980

3. COMAL 80, File System Extension. 04.03.1980

Der var nu grundlag for en standard; men åbenbart må noget have kikset, for allerede i maj 1982 samledes man igen, og nu var interessegruppen blevet udvidet med bl. a. folk fra Irland og USA. En ny beskrivelse - eller snarere en udvidelse af den første - blev resultatet.

Ved en sammenligning mellem de tre ovennævnte rapporter og appendix H i Len Linsay's udmærkede COMAL 80 Handbook, Reston Publishing, vil man se, at COMAL 80 har fået en i mange henseender udmærket afpudsning.

Man kan nu håbe, at leverandører og fabrikanter vil bestræbe sig på i størst mulig udstrækning at følge den beskrivelse af sproget, som de selv har været med til at lave.

Foreløbig har jeg kun haft lejlighed til at se nærmere på CBM COMAL 80, og dette studium har givet mig visse bange anelser. Der er nemlig noget, der tyder på, at man går videre, end beskrivelsen siger. Selv om afvigelserne fra kernebeskrivelsen ikke er de helt store og betydningsfulde, er det dog, som om der kan spores en tendens til, at leverandøren gerne vil lave noget, der kendetegner hans produkt. Måske gør jeg ham uret. Måske er de konstaterede afvigelser blot noget, der bliver standard??

COMAL BRUGER NYT

Man kan selvfølgelig spørge, om det gør noget. Er det ikke kun godt, at der arbejdes videre med sproget?

Jo, ubetinget! Men risikerer man ikke, at den oprindelige hensigt med at lave en standard forflygtiges? Hvad siger de, der skal købe og bruge COMAL 80? ja, hvad siger COMAL 80-brugeren?

Så længe man kun skal kommunikere med sin egen maskine, kan det vel være lige-gyldigt, om der er mindre dialektforskelle; men når man skal kommunikere med andre - bl: a. gennem dette blad, vil en høj grad af standard være ønskelig. Også når der undervises isproget, vil det være rart, om dialekter kan undgås.

COMAL BRUGER GRUPPEN må have en stor interesse i at være medvirkende til at påvirke sprogets udvikling, og man må håbe på, at CBG bliver inviteret med, næste gang en revision forestår.

Der skal sammen med den velkomst, der her gives COMAL 80, udtrykkes ønsket om, at sproget vil arte sig vel og ikke til-lægge sig alle de uvaner, som BASIC har gjort.

Velkommen COMAL 80.

Frank Hammenshøj

C-B-N

SMÅ TIPS

SAVE et program oven i et eksisterende

Når man har rettet i et program, er man sjældent interesseret i at beholde den gamle version på disketten, men ønsker at "save" oven i den.

Det gøres ved hjælp af tegnet "at": @ (også kaldet "snabel-a" og andre mindre høviske ting - men det hedder altså "at"):

SAVE "@0: programnavn"

0 angiver diskstationens port 0. Hvis programmet skal saves på en diskette i port 1, skal der skrives:

SAVE "@1: programnavn"

Hvis programmet skal gemmes på ASCII form, testes:

LIST "@0: programnavn"
eller

LIST "@1: programnavn"

SPØRGEKASSE

Spørgsmål:

Jeg savner en god ide til en procedure, der markerer, hvad der blev valgt, når programmet giver flere muligheder.

EN

Svar:

Her er et program, der viser en mulig løsning af problemet.

I linie 150 skal der vælges og indtastes et bogstav.

Derefter vises alle bogstaverne i alfabetet på skærmen (linie 180), og en pil (linie 190) peger på det valgte bogstav.

Derudover er der tilføjet en else-sætning, der protesterer, hvis der vælges og indtastes noget, der ligger uden for de opstillede muligheder - i dette tilfælde alfabetet - hvorved betingelsen i linie 160 ikke opfyldes.

Loop'en (linie 140) er kun medtaget for at lette afprøvningen af dette program ved gentagelse.

```
0010 // SAVE "@:23.ALFABETPROGRAM"
0020 // CBM COMAL 80 REV 01.02 DISK IN/23
0030 // VERIFY "23.ALFABETPROGRAM"
0040 POKE 59468,12
0050
0060 DIM A$ OF 29, B$ OF 1
0070 A$=""
0080 PRINT CHR$(147)
0090
0100 FOR I := 65 TO (65+29) DO
0110   A$=A$+CHR$(I)
0120   ENDFOR I
0130
0140 LOOP
0150   INPUT "SKRIV ET BOGSTAV: ": B$
0160   IF B$ IN A$ THEN
0170     POS:=B$ IN A$
0180     PRINT A$
0190     PRINT TAB(POS), "↑"
0200     PRINT
0210   ELSE
0220     PRINT B$;"ER IKKE ET BOGSTAV"
0230     PRINT
0240   ENDIF
0150 ENDLOOP
```

C-B-N

COMAL BRUGER NYT

B. Aagaard, E. Christensen, K. Grosen,
L. Pehrsson
DATALÆRE MED MIKRODATAMAT
ISBN 87-7351-027-0
151 sider. Format A 4. Kr. 67,10
forlaget systime a/s, Herning

Dette er en udpræget lærebog for begyndere. Der forudsættes intet kendskab til datamater, og der stilles meget små fordringer til elevernes forkundskaber i øvrigt.

Den er hovedsagelig baseret på anvendelse af en RC 702 PICCOLO, men gennemgår også COMET/DITAMAT og COMMODORE CBM. Baggest i bogen anføres et sammenligningssekema ang. de mest benyttede maskinbestemte ordrer. Man fristes til at sige, at alene denne tabel er bogens pris værd.

Undervisningen går meget langsomt og grundigt til værks, og datamaten tages i anvendelse bogstavelig talt fra første færd - en fremgangsmåde, som eleverne sikkert vil sætte stor pris på frem for at få mange siders teoretisk snak og pairnusseri.

Der er ikke sparet på luft mellem programmerne og forklaringerne, og der er anvendt en hensigtsmæssig varierende skrift til de forskellige dele, hvoraf lektionerne er opbygget. De anvendte nøgleord er meget klart fremhævede i sidernes margen; men endnu engang ser man, at trykkeren har glemt, at margen på en højreside er til højre for teksten og ikke til venstre. Man må altså åbne bogen fuldt for at få øje på de fremhævede vejledende ord, som man burde kunne se blot ved en simpel gennembladning.

Hvert kapitel slutter med en oversigt over det lærte - en god idel og med et antal velgennemtænkte og lettilgængelige opgaver.

Der gøres meget ud af rutediagrammer, og det vises, hvorledes eleverne i virkeligheden er vel kendt med andre instruktioner, der også kan analyseres og opstilles i diagram (f. eks. brugsanvisninger).

Forfatterne lægger vægt på, at faget datalære også bør indeholde en gennemgang af den sociale og samfundsmæssige side af anvendelsen af datamater. Der er eksempler på gode og dårlige anvendelser af registre og faren ved de sidste.

Bogen er uden tvivl en god støtte for læreren, og den appellerer til elevens selvstændige arbejde med stoffet uden at stille urimelige krav.

S. Chr. H.

C-B-N

Østergaard, Poul
Programmering i Comal 80
ISBN 87-571-0701-7
248 sider. Kr. 149,50
Teknisk Forlag A/S, København 1982

Man mærker straks, at forfatteren kan mere end sit modersmål. Han forstår at delagtiggøre sine medmennesker i Comal - det EDB-sprog, der mere end noget andet har vundet fodfæste i de senere år - ikke alene til undervisning, men nu også til brug for administration og processtyring.

Bogen er egnet både til undervisning og til selvstudium. Den er seriøs og henvender sig også til folk, der ønsker et nærmere kendskab til struktureret programmering. Det sørger et utal af eksempler for. Gør man sig den ulejlighed at gennemgå disse eksempler stykke for stykke, kan man komme ret langt i forståelsen. Man vil tillige blive opdraget til at "hakke" programmer op i små bidder - de såkaldte proc og func.

Bogen omhandler også de mere raffinerede ting som parameteroverføring, og forfatteren berører ganske let det, man kan kalde Comal for viderekomne: recursive proc. og func. Her er tale om en vanskelig afdeling, som gør Comal spændende, længe efter at den grundlæggende syntax er lært.

Apropos proc - som jo er en forkortelse af procedurer: forfatteren har opbygget sin bog som procedurer, og et stikordsregister på over 500 ord er nøglen til hurtige opslag gennem 248 sider. Forfatteren skriver ganske vist et sted, at bogen ikke skal betragtes som en manual; men hvis jeg her og nu vil vide noget om nøgleordet mod, henviser registret til side 23 og 27, og vupti: 2 minutter efter ved jeg en hel del om den sag.

Bogen er blevet opdateret i foråret 1982, idet der er indlagt et blad med tilføjelser og ændringer.

Frank Hejndorf

COMAL BRUGER NYT

Arne Christensen, Mogens Pelle, Susanne Sønderup
 COMAL 80 HANDBOG
 ISBN 87-88059-03-3
 147 sider format A 5
 kr. 175,00 + moms og forsendelse
 Bogika Data Systemer, Ikast

Dette er en virkelig god håndbog i
 COMAL 80!

Det drejer sig om et løsbladesystem med sider af kraftigt lysebrunt papir med mørkebrunt tryk indsat i et tiltalende ringbind. Der er simpelthen kølet for udstyret, så det er en øjenfryd at arbejde med bogen.

I introduktionen nævnes det, at værket ikke er tænkt som nogen lærebog i COMAL 80, men som en håndbog for de specielle faciliteter, som METANIC COMAL 80 indeholder. Ikke desto mindre vil brugere af andre COMAL maskiner kunne have rigt udbytte af at læse og anvende bogen, og her kommer dens opbygning med løsblade specielt til nytte, idet brugeren jo kan indsætte blade med noter om sin egen maskine og sine egne erfaringer.

Bogen er opdelt i to hovedafsnit: Først en generel omtale af sproget, terminologien og hvordan det fungerer. Dernæst den altovervejende del - en gennemgang af samtlige nøgleord taget i alfabetisk rækkefølge, så det er let at finde frem til netop det emne, man står og savner oplysning om. Og denne gennemgang er fortrinlig. Den er systematisk - alle sider er inddelt efter en fast gennemgående plan, og det anvendte sprog er holdt i almindelige, letforståelige vendinger.

Bogen slutes med 4 tillæg: fejlmeldinger, demonstrationsprogrammer, biblioteksrutiner og ASCII karakter koderne.

En beskeden indvending: Nøgleordene, som man kigger efter, når man søger i bogen, er trykt inders inde på arkene, altså helt inde ved bogryggen, medens man som bekendt ved en hurtig gennembladning ser den yderste del af arkene, længst borte fra ryggen.

Alt i alt en virkelig god håndbog, som kan anbefales på det bedste.

S. Chr. H.

Birgit Huss Nielsen, Finn Lunde Nielsen
 COMAL-80
 ISBN 87-88059-04-9
 88 sider, format A 4
 Kr. 45,00 + moms og forsendelse
 Bogika Data Systemer, Ikast

Til brug ved gymnasiets EDB undervisning er denne bog skrevet på basis af Metanics version af COMAL 80 til anvendelse på MPS-3000.

Vejledende løsninger til øvelserne kan fås i form af et hæfte, og programmerne kan erhverves på to disketter.

Lærebogen er præget af en klar inddeling og et overskueligt layout. Alle anvisninger gives i et velformet og distinkt sprog uden overflødige dikkedarer. Derved overkommes en utrolig stor stofmængde på en relativ beskeden plads.

Men der er skub i foretagendet! Der gås hurtigt frem, og der forudsættes kendskab til matematikken og dens specielle terminologi. Til gengæld når man at analysere selv ret indviklede kurver og figurer; med der gøres ikke meget ud af, at datamater faktisk kan andet end beskæftige sig med tal.

Det er uden tvivl en fortrinlig bog til sit specielle formål.

S. Chr. H.

C B N

REDAKTIONELT

Næste nummer af COMAL BRUGER NYT udkommer i marts 1983.

Indlæg til bladet modtages meget gerne, og må være redaktionen i hænde senest 28. februar.

Bladet indhold vil i meget høj grad blive bestemt af læserne, der alle bedes betragte sig som direkte medarbejdere, ved f. eks. at levere stof til bladet, eller ringe til redaktionen med specielle ønsker om artikler eller programmer.

CBN vil gøre sit bedste for at imødekomme alle anmodninger. Um Ny abonnenter optages ved telefonisk eller skriftlig henvendelse til redaktionen og ved indbetaling af abonnementsprisen for et år: kr. 90,00.

Gironummeret er 1 75 23 75.

COMAL BRUGER NYT

HOLD STYR PÅ DISKETTERNE

I dette og næste nummer af CBN bringes en programudskrift af et diskettestyringsprogram skrevet i CBM COMAL 80 rev. 1.02. Det udnytter faciliteter, der er specifikke for COMMODORES computere og diskettestationer, men det vil sandsynligvis ikke være særlig svært at omskrive programmet til en anden COMAL-computer, da de fleste specielle faciliteter er pakket inde i procedurer for sig (d.v.s. læsning af kataloger på en diskette, indtastning af et tegn uden at det udskrives, en sikker inputroutine, o.s.v.).

For at programmet skal kunne benyttes, er det nødvendigt at man har adgang til en dobbelt diskettestation, en 40- eller 80-tegns computer med mindst 25 Kb frit lager samt en printer. Programmet i sig selv fylder ca. 9 Kb (570 programlinier), men behøver op til 15 Kb til data (d.v.s. globale og lokale variable).

I diskettestationens drive 0 skal diskettestyringsdisketten anbringes. På disketten skal diskettestyringsprogrammet være samt alle data om de styrede disketter. For hver diskette, der skal styres, opføres en fil med diskettens navn og af type 'usr' (speciel COMMODORE filtype), som indeholder oplysninger om antal filer (programmer), hvilke filer og disses størrelse og type. Da antallet af filnavne, der må være på et diskettekatalog, er begrænset, kan en diskettestyringsdiskette højst styre 144 andre disketter.

Programmet har følgende menupunkter:

1. Hold styr på ny diskette,
2. Disketteudskrift,
3. Titeldskrift,
4. Navne på alle disketter,
5. Fjern diskette,
6. Slut.

ad 1. Tryk 'h'. Den nye diskette anbringes i diskettestationens drive 1. Alle oplysninger om disketten læses og gemmes på

styringsdisketten. Hvis der allerede blev holdt styr på en diskette af det navn, spørges om de gamle data skal overskrives.

ad 2. Tryk 'd'. Indtast et diskette-navn. Diskettekataloget vil herefter blive udskrevet på enten skærm eller printer. Diskettenavnet må gerne angives ufuldendt; d.v.s. hvis diskettenavnet er DEMO DISKETTE, må det gerne forkortes til DEMO DISK*, DEMO* eller DEM*, hvis det er entydigt. Stjernen '*' angiver, at der må være 0, 1 eller flere variable tegn efter navnet.

ad 3. Tryk 't'. Udskriver samtlige filnavne på de styrede disketter i alfabetisk orden med henvisning til, på hvilken diskette filen kan findes. Denne funktion kan godt tage ret lang tid. Hvis der er f. eks. 20 diskettebeskrivelser, vil der gå ca. 10 minutter med sortering før udskriften kan begynder.

ad 4. Tryk 'n'. Navnene på alle de styrede disketter udskrives.

ad 5. Tryk 'f'. Indtast et diskettenavn. Alle data om denne diskette vil herefter være glemt. Ved angivelsen af diskettenavnet må der benyttes en forkortelse på samme måde som ved punkt 2. Men pas på. Hvis DEMO* angives, betyder det at alle styrede disketter, der har DEMO som de første bogstaver af navnet, vil blive slettet.

ad 6. Tryk 's'. Programmet slutter.

PS. Hvis man har adgang til en ASCII printer, kan COMAL'en konvertere fra PETASCII til standard ASCII, når kommandoen POKE 588,1 gives.

LL

COMAL BRUGER NYT

```

0010 //SAVE"@0:HOLDSTYR"
0020 //
0030 // *****
0040 // ** **
0050 // ** HOLDSTYR **
0060 // ** **
0070 // ** AF . **
0080 // ** **
0090 // ** LARS **
0100 // ** LAURSEN **
0110 // ** **
0120 // *****
0130 //
0140 LOOP
0150 // UDSKRIV MENU, OG VÆLG FUNKTION.
0160
0170 CLEAR
0180 OVERSKRIFT("HOLD STYR PÅ DISKETTERNE !!")
0190 PRINT ""
0200 PRINT "VÆLG FUNKTION:"
0210 PRINT
0220 MENU("HOLD STYR PÅ NY DISKETTE")
0230 MENU("DISKETTE UDSKRIFT")
0240 MENU("TITELUDSKRIFT")
0250 MENU("NAVNE PÅ ALLE DISKETTER")
0260 MENU("FJERN DISKETTE")
0270 MENU("SLUT")
0280 PRINT "FUNKTION: ",
0290
0300 CASE TAST( "hldDtTnNfFsS" ) OF
0310 WHEN 1,2
0320 FUNKTION("HOLDSTYR PÅ NY DISKETTE")
0330 HOLDSTYR_PÅ_NY_DISKETTE
0340 WHEN 3,4
0350 FUNKTION("DISKETTE UDSKRIFT")
0360 KATALOG UDSKRIFT
0370 WHEN 5,6
0380 FUNKTION("TITELUDSKRIFT")
0390 TITELUDSKRIFT
0400 WHEN 7,8
0410 FUNKTION("NAVNE PÅ ALLE DISKETTER")
0420 NAVNE_PÅ_KATALOGER
0430 WHEN 9,10
0440 FUNKTION("FJERN DISKETTE")
0450 FJERN KATALOG
0460 WHEN 11,12
0470 PRINT "SLUT"
0480 EXIT
0490 ENDCASE
0500
0510 ENDLLOOP
0520
0530 END "SLUT PÅ HOLDSTYR"
0540
0550 PROC CLEAR
0560 PRINT CHR$(14), // IKKE SEMI-GRAFIK
0570 PRINT CHR$(147), // SLET SKÆRMEN
0580 ENDPROC CLEAR
0590

```

HOLD STYR PÅ DISKETTERNE

OBS!

De fleste skrivemaskiner kan ikke skrive #. I stedet skrives £.

Det er f.eks. også tilføjet med dette program hvor man skal erstatte £ med #.

Her er det aktuelt fra og med linie 2490.

COMAL BRUGER NYT

```

0600 PROC OVERSKRIFT(TEKST$) CLOSED
0610   PRINT TEKST$
0620   // UNDERSTREG:
0630   FOR I:=1 TO LEN(TEKST$) DO PRINT CHR$(247),
0640   PRINT
0650 ENDPROC OVERSKRIFT
0660
0670 PROC MENU(TEKST$) CLOSED
0680   // SÆT STREG UNDER FØRSTE TEGN:
0690   PRINT " ",TEKST$
0700   PRINT " ",CHR$(227)
0710 ENDPROC MENU
0720
0730 FUNC TAST(LOVLIGE_TEGN$) CLOSED
0740   IMPORT TASTATUR
0750
0760   // BLINK MED CURSOR INDTIL ET LOVLIGT
0770   // LOVLIGT ER INDTASTET.
0780   // RETURNER POSITIONEN AF TEGNET I
0790   // STRENGEN LOVLIGE_TEGN$.
0800
0810   REPEAT
0820     POS:=CHR$(TASTATUR) IN LOVLIGE_TEGN$
0830   UNTIL POS>0
0840   RETURN POS
0850
0860 ENDFUNC TAST
0870
0880 FUNC TASTATUR CLOSED
0890
0900   // BLINK MED CURSOREN INDTIL
0910   // DER ER INDTASTET ET TEGN.
0920   // DEREFTER RETURN ASCII-VÆRDIEN.
0930
0940   TID:=TIME
0950   REPEAT
0960     IF (TIME-TID) MOD 50<25 THEN
0970       PRINT CHR$(164)+CHR$(157),
0980     ELSE
0990       PRINT " "+CHR$(157),
1000     ENDIF
1010     TEGN:=-ORD(KEY$)
1020     UNTIL TEGN<>0
1030     PRINT " "+CHR$(157),
1040     RETURN TEGN
1050 ENDFUNC TASTATUR
1060
1070 PROC FUNKTION(TEKST$)
1080   // UDSKRIV SVARET PÅ FUNKTION:
1090   PRINT TEKST$(1)
1100   PAUSE(.5)
1110   CLEAR
1120   OVERSKRIFT(TEKST$)
1130 ENDPROC FUNKTION
1140
1150 PROC HOLDSTYR PÅ NY DISKETTE CLOSED
1160   IMPORT DIR,DISKETTE_ISAT
1170   IMPORT PAUSE,FINDES_TAST
1180   IMPORT FJERN_MELLEMRUM
1190

```

COMAL BRUGER NYT

```

1200 DIM KATALOG$(1:144) OF 27
1210 DIM DISKETTE$ OF 18
1220 ANTLIN:=0
1230 REPEAT
1240 PRINT
1250 PRINT "HVIS DISKETTEN ER I DRIVE 1,"
1260 PRINT "SA TRYK "+CHR$(18)+" SPACE "+CHR$(146)+" ": ",
1270 BLIND:=TAST(" ")
1280 PRINT
1290 PRINT
1300 OKAY:=DISKETTE ISAT
1310 IF NOT OKAY THEN
1320 PRINT CHR$(7)+"DER ER ",
1330 PRINT CHR$(18)+" INGEN DISKETTE "+CHR$(146),
1340 PRINT " I DRIVE 1 !!"
1350 ENDIF
1360 UNTIL OKAY
1370 DIR("1:*",TRUE,ANTLIN,KATALOG$)
1380 PRINT
1390 DISKETTE$:=KATALOG$(1)(5:20)
1400 FJERN MELLEMRUM(DISKETTE$)
1410 DISKETTE$:=""+"0:"+DISKETTE$
1420 GEMMES:=TRUE
1430 IF FINDES(DISKETTE$) THEN
1440 // ER FILEN AF TYPE 'USR' ?
1450 IF FINDES(DISKETTE$+"",USR") THEN
1460 PRINT "DENNE DISKETTE KENDES ALLEREDE. SKAL"
1470 PRINT "DEN GAMLE DISKETTE OVERSKRIVES ? ",
1480 IF TAST("JNN")<3 THEN // D.V.S. 'JA'
1490 PRINT "JA"
1500 DELETE DISKETTE$
1510 PRINT
1520 ELSE
1530 PRINT "NEJ"
1540 PRINT
1550 PRINT "SA GEMMES DISKETTEN IKKE."
1560 GEMMES:=FALSE
1570 ENDIF
1580 ELSE // FIL AF FORKERT TYPE
1590 GEMMES:=FALSE
1600 PRINT "KAN DESVÆRRE IKKE GEMME EN DISKETTE"
1610 PRINT "AF NAVNET '",DISKETTE$(3:LEN(DISKETTE$))','"
1620 PRINT
1630 PAUSE(3)
1640 ENDIF
1650 ENDIF
1660 IF GEMMES THEN
1670 PRINT "NU GEMMES DISKETTEN..."
1680 PRINT
1690 OPEN FILE 2,DISKETTE$+"",USR,WRITE
1700 WRITE FILE 2: ANTLIN-2 // ANTAL TITLER I KATALOGET
1710 FOR I:=1 TO ANTLIN DO
1720 WRITE FILE 2: KATALOG$(I)
1730 ENDFOR I
1740 CLOSE FILE 2
1750 PRINT "DISKETTEN ER GEMT."
1760 ENDIF
1770 PAUSE(2)
1780 ENDPROC HOLDSTYR_PÅ_NY_DISKETTE
1790

```

COMAL BRUGER NYT

```

1800 PROC KATALOG UDSKRIFT CLOSED
1810   IMPORT INDFAST,PAUSE,FINDES
1820   IMPORT TAST,SPACE,VÆLG_OUTPUT
1830   IMPORT OUTPUT_FÆRDIG
1840   DIM KATALOG$ OF 22
1850   DIM LINIE$ OF 40
1860   PRINT
1870   PRINT "UDSKRIFT AF HVILKEN DISKETTE ?"
1880   INDFAST("DISKETTE: ",16,KATALOG$)
1890   PRINT
1900   PRINT
1910   KATALOG$:="0:"+KATALOG$+"USR"
1920   IF NOT FINDES(KATALOG$) THEN
1930     PRINT "DEN DISKETTE FINDES IKKE."
1940     PAUSE(2)
1950   ELSE
1960     SKÆRM:=VÆLG_OUTPUT(7)
1970     OPEN FILE 2,KATALOG$,READ
1980     READ FILE 2: ANTLIN
1990     FOR I:=1 TO ANTLIN+2 DO
2000       READ FILE 2: LINIE$
2010       PRINT " ",LINIE$
2020       IF SKÆRM THEN SPACE
2030     ENDFOR I
2040     CLOSE FILE 2
2050     OUTPUT_FÆRDIG(SKÆRM)
2060   ENDIF
2070 ENDPROC KATALOG_UDSKRIFT
2080
2090 PROC TITELUDSKRIFT CLOSED
2100   IMPORT DIR,VÆLG_OUTPUT,OUTPUT_FÆRDIG
2110   IMPORT SPACE,CLEAR
2120   PROC QUICKSORT(FRA,TIL)
2130     S:=1; STAKV(1):=FRA; STAKH(1):=TIL
2140     REPEAT
2150       VENSTRE:=STAKV(S); HØJRE:=STAKH(S); S:-1
2160     REPEAT
2170       I:=VENSTRE; J:=HØJRE; PIVOT$:=TITEL$(INDEX$(I+J) DIV 2)
2180     REPEAT
2190       WHILE TITEL$(INDEX$(I))<PIVOT$ DO I:+1
2200       WHILE PIVOT$<TITEL$(INDEX$(J)) DO J:-1
2210       IF I<=J THEN
2220         SK:=INDEX$(I); INDEX$(I):=INDEX$(J); INDEX$(J):=SK
2230         I:+1; J:-1
2240       ENDIF
2250     UNTIL I>J
2260     IF J-VENSTRE<HØJRE-I THEN
2270       IF I<HØJRE THEN
2280         S:+1; STAKV(S):=I; STAKH(S):=HØJRE
2290       ENDIF
2300     HØJRE:=J
2310   ELSE
2320     IF VENSTRE<J THEN
2330       S:+1; STAKV(S):=VENSTRE; STAKH(S):=J
2340     ENDIF
2350     VENSTRE:=I
2360   ENDIF
2370   UNTIL VENSTRE>=HØJRE
2380   UNTIL S=0
2390 ENDPROC QUICKSORT
2400

```


COMAL BRUGER NYT

```

2410 PROC FLET(N2)
2420   RENAME("SORT1","SORT2")
2430   PIL1:=1; PIL2:=1
2440   OPEN FILE 1,"0:SORT2",READ
2450   OPEN FILE 2,"0:SORT1",WRITE
2460   READ FILE 1: N1
2470   WRITE FILE 2: N1+N2
2480   IF N1>0 THEN
2490     READ FILE 1: TITEL1$,INFO1$,KAT1$
2500     WHILE PIL1<=N1 AND PIL2<=N2 DO
2510       IF TITEL1$<TITEL$(INDEX$(PIL2)) THEN
2520         WRITE FILE 2: TITEL1$,INFO1$,KAT1$
2530         PIL1:+1
2540         IF PIL1<=N1 THEN READ FILE 1: TITEL1$,INFO1$,KAT1$
2550       ELSE
2560         I:=INDEX$(PIL2)
2570         WRITE FILE 2: TITEL$(I),INFO$(I),KAT$(I)
2580         PIL2:+1
2590       ENDIF
2600     ENDWHILE
2610   WHILE PIL1<=N1 DO
2620     WRITE FILE 2: TITEL1$,INFO1$,KAT1$
2630     PIL1:+1
2640     IF PIL1<=N1 THEN READ FILE 1: TITEL1$,INFO1$,KAT1$
2650   ENDWHILE
2660 ENDIF
2670 WHILE PIL2<=N2 DO
2680   I:=INDEX$(PIL2)
2690   WRITE FILE 2: TITEL$(I),INFO$(I),KAT$(I)
2700   PIL2:+1
2710 ENDWHILE
2720 CLOSE FILE 2
2730 CLOSE FILE 1
2740 DELETE "0:SORT2"
2750 ENDPROC FLET
2760
2770 PROC RENAME(OLDFILE$,NEWFILE$)
2780   PASS "RENAME:"+"NEWFILE$+"+"OLDFILE$"
2790 ENDPROC RENAME
2800
2810 MAX:=300
2820 DIM TITEL$(1:MAX) OF 16, INFO$(1:MAX) OF 7
2830 DIM INDEX$(1:MAX), KAT$(1:MAX)
2840
2850 // TITEL$(): TITLER, DER FLETTES MED ALLEREDE
2860 //          SORTEREDE TITLER I FILEN 'SORT1'.
2870 // INFO$(): INFORMATION OM TITLEN (STØRRELSE,
2880 //          FILTYPE).
2890 // KAT$( ) : DE KORRESPONDERENDE KATALOGNUMRE
2900 //          TIL TITEL$().
2910 // INDEX$( ): INDEKS-VEKTOR. VED SORTERINGEN
2920 //          BYTTES KUN OM PÅ INDICES, IKKE
2930 //          POSTER.
2940
2950 DIM TITEL1$ OF 27, INFO1$ OF 7, TEMP$ OF 40
2960 DIM PIVOT$ OF 16, STAKV(15), STAKH(15)
2970 DIM STARTTEGN$ OF 1
2980

```