

R

...DET ER GANSKE VIST !

CBN 2/7

2. SAMLING - NR. 1.

[1985]

## COMAL VAR EN LANGSOM STARTER

Intet under - opstået som sproget er blandt undervisere i Danmark, har det fået en langsom udbredelse - hovedsagelig gennem skolerne, der jo har haft store problemer med at få lov at lære eleverne at bruge edb-maskiner.

Men nu er der "gået hul på bylden": de højere pædagogiske magter har lyst barnet delvis lære i kuld og køn, og landet over købes der nu ind af maskiner, så man aldrig har set mage, og det drejer sig udelukkende (så vidt vi ved) om COMAL-maskiner. Alt andet er utænkeligt.

Det er en fastslået kendsgerning, at i Danmark er undervisningssproget COMAL.

Når vi først har konstateret denne succes, kan vi tilføje, at der er klare tegn i sol og måne på, at den hurtigt vil brede sig også uden for vore grænser.

Det er vist en kendt sag, at COMAL i adskillige andre lande, i og uden for Europa, allerede i længere tid har været kendt og benyttet. I flæng kan nævnes USA, England, Irland, Tyskland, Sverige o. m. a.

Men noget egentlig verdenssprog kan vi næppe tillade os at sige, at COMAL hidtil har været.

Men wait and see: Nu sker der en ændring!

Det er vel ingen hemmelighed, at et firma som IBM har en eventyrlig indflydelse på, hvordan fremtiden vedrørende EDB skal formes. Og da nu IBM har truffet aftale om at få COMAL implementeret på sine PC'er, tør man nok spå, at det vil give sproget et helt usædvanligt skub fremad.

En anden maskine med kolossal udbredelse tilbyder nu for alvor COMAL: Appel. Det

vil utvivlsomt få en lignende virkning, når det nu via IBM er fastslået, at COMAL duer og er kommet for at blive.

Af de mere hjemlige tiltag skal nævnes Regnecentralens "Piccoline", der har været meget nær ved at blive kanoniseret som undervisningssektorens "industristandard".

Om den megen blæst, der rejste sig mod, at et enkelt fabrikat efter sigende af myndighederne blev favoriseret, har medført, at programudbuddet til skolerne kommer til at dække flere maskiner, end der var lagt op til - ja, det får den nærmeste fremtid vise.

I de mange hjem er det dog nok de mindre maskiner, der er afgørende for, om COMAL skal slå igennem, og det vil uægtelig påvirke situationen, at børnene lærer dette sprog i skolerne. Så de vil jo helt naturligt brokke sig slemt derhjemme, hvis de der skal skifte over til en "BASIC-dåse". (Det var ikke pænt sagt!)

Det er den helt naturlige forklaring på annoncer med indholdet: Lær det sprog, du lærer i skolen (COMAL 80), på din egen ZX-Spectrum 48 K ved hjælp af ZX-COMAL.

P.S. Læs inde i bladet om CBM COMAL 80 version 0.14.

Universitetsbiblioteket, 2. afd.

0108-4925

INDHOLDSFORTEGNELSE

200003451649



# COMAL BRUGER NYT

## INDHOLDSFORTEGNELSE

|                                 |    |
|---------------------------------|----|
| Leder: Det er ganske vist ..... | 1  |
| Redaktionelt .....              | 2  |
| Streng i revers .....           | 2  |
| Anmeldelse .....                | 3  |
| Rettelse .....                  | 3  |
| Urtid .....                     | 4  |
| Anmeldelse .....                | 5  |
| Om parameteroverførsel .....    | 6  |
| Bland tal .....                 | 10 |
| Banko .....                     | 11 |
| LOAD fra CAT .....              | 13 |
| Rentedage .....                 | 14 |
| Længsomme bogstaver .....       | 15 |

## REDAKTIONELT

Med dette nummer af CBN (2/7), der er 1. nummer i 2. samling, indleder vi et nyt afsnit af vor tilværelse.

Vi havde planlagt, at 1. samling skulle udkomme i løbet af et år; men vi må sæde, at al begyndelse er vanskelig.

I den forbindelse må vi prise os lykkelige for vore læsers tålmodighed, som vi efter bedste evne vil forsøge at sætte på en mindre krævende prøve for fremtiden.

Hvad redaktionen kunne ønske sig, er en nærmere kontakt med brugergruppens medlemmer. Vi gør vort bedste for at gætte læsernes ønsker; men det blev jo mindre gætteeri, hvis man skrev til bladet eller ringede.

En ting er selvfølgelig givet: vi ser meget gerne, at der indsendes artikler, programmer, spørgsmål, forslag og kommentarer. Et tidsskrift, der er talerør for en brugergruppe står og falder med opslutningen om formålet: at bidrage til at udbrede kendskab til COMAL og at bistå hinanden i stedet at få udvidet sin indsigt i sproget og dets mange muligheder.

Efterhånden har vi bragt så mange brugbare programmer, at der kan være grund til at overveje udgivelsen af en diskette med disse. I den forbindelse efterlyser vi folk, der vil være med til at transponere programmerne over til andre maskiner, end dem de direkte er skrevet for.

Til oplysning for nye læsere kan vi meddele, at de tidligere numre af bladet: 1. samling (nr. 1 - 6) stadig kan fås ved henvendelse til redaktionen.

red.

## STRENGE I REVERS

Hvis man ønsker streng sat i revers på skærmen (sorte bogstaver på lysende baggrund), kan dette lille program anvendes:

```
0010 DIM navn$ OF 20
0020 INPUT "navn > ": navn$
0030 neon(navn$)
0040
0050
0060 PROC neon(tekst$)
0070   PRINT CHR$(18);tekst$,
      CHR$(32)
      // revers - space
0080 ENDPROC neon
```

Linierne 0010 - 0050 er til afprøvning af proceduren. I tilfældet her er der tale om parameteroverføring.

Bemærk, at kun navn\$ er dimensioneret og ikke tekst\$. Det er unødvendigt, da indholdet i navn\$ overføres til tekst\$. Det har den praktiske betydning, at proceduren kan kaldes med en hvilken som helst strengvariabel.

Man kunne altså lige så godt have kaldt strengvariabelen i linie 0020 for a\$, vel at mærke hvis strengvariabelen i linie 0010 også blev kaldt a\$.

Frank Hejndorf

## COMAL BRUGER NYT

### ANMELDELSE

Børge R. Christensen  
COMAL 80 på 64'eren  
forlaget systime a/s  
Format A5  
Kr. 154,00 incl. moms.

Sagt med det samme:  
Her er den lærebog, du har ventet på.

Alle tekniske og pædagogiske sider er tilgodeset.

Opdelingen, opsætningen og typografien er logisk. Det betyder en stor lettelse ved tilgængelsen og ved senere opslag. (Stikordsregistret måtte for min skyld gerne være endnu mere fyldigt, end det faktisk er).

Selvfulgelig er en bog af Børge Christensen skrevet i en nøje afvejet, korrekt stigende sprogbrug og med hensyntagen til en lettelig indføring dels i maskinens og dels i COMAL's mange muligheder.

Bogen er konsekvent opdelt i forklarende afsnit, i øvelser og i opgaver. Tilmed indledes hvert kapitel med en oversigt over, hvad der skal gennemgås, og det afsluttes med en kort, men fyndig repetition af hvad man (forhåbentlig) fik lært. Det sidste kan anses en og anden til at læse kapitlet igen, hvis vedkommende finder ud af, at han ikke har fået det hele med.

Bogen er baseret på COMAL-80, sådan som sproget er implementeret i COMMODORE 64 i versionerne 0.12 og 0.14, idet der dog også findes henvisninger til versionen 2.00.

Forskellen mellem versionerne ligger teknisk i, at 0.12 og 0.14 er bånd- eller disketteudgaver, medens 2.00 er en ROM-udgave (placeret i en cartridge = kapsel) til at anbringe i modulporten.

Medens version 2.00 giver brugeren rådighed over 30.714 bytes, må han nøjes med 10 Kbytes, når de softload'ede udgaver anvendes.

Det siger sig selv, at kapseludgaven rummer flere muligheder end de øvrige. Ikke desto mindre er alle de vigtigste kommandoer medtaget, og end ikke en komfortabel højopløsningsgrafik (inclusive turtlegrafik) eller spriteshåndtering savnes, og Børge Christensen giver begge dele en rimelig behandling i bogen, der selvsagt ikke giver sig ud for at være udtømmende. Den sluttes imidlertid med en komplet oversigt over alle nøgleord og alle fejlkoder i CBM-COMAL-80 rev. 0.14.

Bogen kan på det varmeste anbefales til den, der for en beskedent sum vil igang med at lære COMAL.

S. Chr. H.

### RETTELSE

I CBN nr. 6 havde der indsneget sig en fejl på side 14 (sidenumrene manglede desværre også!) i programmet TIPSKUPUN.

I hver af linierne 240, 260 og 280 manglede der to anførelses tegn. Vi gengiver derfor hele proceduren, hvori disse linier forekommer.

Og denne gang er de rigtige!

red.

```
0190 PROC valg'tegn
0200   FOR x:=1 TO 9 DO
0210     FOR y:=1 TO 13 DO
0220       v:=RND(1,100)
0230       IF v<hjem(y) THEN
0240         tegn$(y):=tegn$(y)+" 1 "
0250       ELSEIF v<hjem(y)+uafg(y) THEN
0260         tegn$(y):=tegn$(y)+" x "
0270       ELSE
0280         tegn$(y):=tegn$(y)+" 2 "
0290       ENDIF
0300     ENDFOR y
0310   ENDFOR x
0320 ENDPROC valg'tegn
```

## COMAL BRUGER NYT

### URTID

Her er et program, der er skrevet for COMAL diskversionen 0.12 på COMMODORE 64.

64'eren har ikke funktionen TIME, ud fra hvilken man ellers kan "konstruere" en løbende digital udlæsning af klokkeslettet.

---

NB: £-tegnet skal på computeren erstattes med nummertegnet: #

---

Men man kan alligevel lave et lignende arrangement sådan:

```
0010 // list "l.urtid" // commodore 64
0020 // cbm comal 80 rev 0.12 disk 05
0030 // verify "l.urtid"
0040 a:=60
0050 PRINT CHR$(147);
0060 stiltid
0070 REPEAT
0080   PRINT CHR$(19) // cursor home
0090   FOR n:=1 TO 8 DO PRINT
0100     sekunder:=tid
0110     tim:=sekunder DIV (60*60)
0120     rest:=sekunder MOD (60*60)
0130     mi:=rest DIV 60
0140     se:=rest MOD 60
0150     PRINT USING          ££ ££ ££: tim,mi,se
0160 UNTIL FALSE
0170 FUNC tid
0180   ti:=INT((PEEK(160)*256*256+PEEK(161)*256+PEEK(162))/a)
0190   RETURN ti
0200 ENDFUNC tid
0210 PROC stiltid
0220   INPUT "skriv timer ": timer
0230   INPUT "skriv minutter ": minut
0240   INPUT "skriv sekunder ": sekund
0250   jiffy:=timer*60*60+a+minut*60+a+sekund*a
0260   høj:=jiffy DIV (256*256)
0270   jiffy:=jiffy MOD (256*256)
0280   mellem:=jiffy DIV 256
0290   lav:=jiffy MOD 256
0300   POKE 160,høj
0310   POKE 161,mellem
0320   POKE 162,lav
0330 ENDPROC stiltid
```

Programmet forlanger i tre input opgivet det øjeblikkelige klokkeslet i timer, minutter og sekunder.

Når de tre tal er indtastet, går uret igang, i samme øjeblik der trykkes på RETURN (efter sekund-inputtet).

## COMAL BRUGER NYT

De indtastede tal omsættes i linie 250 til jiffies (= tresindstyvendele af sekunder), som derefter i linie 260 - 290 opdeles i tre afsnit: høj, mellem og lav ved hjælp af funktionerne DIV (heltalsdivision) og MOD (modulus).

I linie 300 - 320 lægges disse tre talafrsnit ind i maskinens hukommelse i de tre celler, hvor den husker antallet af jiffies, der er gået siden maskinens start.

De samme tre hukommelsesceller aflæses derefter løbende i linie 100 - 140 og udskrives i linie 150.

Bemærk funktionen i linie 170 - 200, der "kigger" ind i de tre nævnte hukommelsesceller og returnerer det "sæte" i sekunder til linie 100.

Den, der har lyst at afprøve programmet på en CBM, må ændre linierne 300 - 320 til:

```
0300   POKE 141,høj
0310   POKE 142,mellem
0320   POKE 143,lav
```

S.J.

**CBN**

Skriv til CBN.

Brug spørgebrevkassen.

Benyt læsebrevkassen.

Indmeldelsesblanketten på bagsiden må gerne kopieres!

CBN hører gerne din mening.

Hvad kan CBN gøre for dig?

## ANMELDELSE

COMAL USER GROUP USA:  
COMAL-80 version 0.14  
på diskette og tape  
for COMMODORE 64

Som omtalt andetsteds i bladet har Børge Christensen lige skrevet "COMAL-80 på 64'eren", en lærebog, der i detaljer gennemgår COMAL, som det findes på bånd eller diskette beregnet for COMMODORE 64 og implementeret af UNICOMAL DENMARK APS.

Denne udgave af COMAL-80 (version 0.14) er helt speciel derved, at disketten (eller båndet) kan kopieres uden afgift, hvilket vil sige, at man kun betaler disketten (eller båndet) og ulejligheden ved overspilningen.

Der er ingen tvivl om, at dette tiltag i høj grad vil befordre udbredelsen af COMAL.

Forhandlere vil måske mene, at det vil hindre salget af COMAL kassen; men en sådan frygt er formodentlig ganske ubegrundet. Man kunne med lige så stor ret påstå, at det snarere ville øge behovet for "den rigtige version" af COMAL - den, der er parat, når man tænder for maskinen, og som rummer adskillige flere kommandoer end de softload'ede versioner, og som først og fremmest har de meget værdifulde "pækker".

Men om 0.14 udgaven skal tilføjes: Den er forsynet med en hurtiglader, der gør, at hele sproget indlæses på 15 sekunder fra disketten og på 2 minutter fra bånd.

Kom så ikke og sig, at 1541'eren og datasetten er langsomme maskiner.

Og så har 0.14 danske fejlmeddelelser. Det er der nok mange, der vil sætte pris på.

Endelig rummer den en særlig lækkerbid- sken for den, der ejer en printer MPS 801 - nemlig en "hcopy", ved hjælp af hvilken printeren laver en tro kopi af tegningen på skærmen.

Nu er der da ikke længere nogen undskyldning for at blive i BASIC!

S. Chr. H.

## COMAL BRUGER NYT

### OM PARAMETEROVERFØRSEL

Muligheden for anvendelse af procedurer er som bekendt en af de store fordele ved COMAL.

I almindelighed frembyder procedurer ikke større problemer - i hvert fald ikke, når man undlader overførsel af parametre; men netop denne udvidelse af procedurebegrebet er i mange situationer særdeles nyttig, hvorfor det nok var en god ide at ofre lidt tid på denne detalje.

Hvad forstår man ved parametre?

Jå, i denne artikel vil vi holde os til talstørrelser, der hentes et sted og bearbejdes et andet.

Den generelle ophugning af en procedure med parameteroverførsel er følgende:

```
0010 // eksempel 1
0020
0030 PRINT "pris excl. moms: kr. 200"
0040
0050 beregn'pris'incl'moms(200)
0060
0070 PROC beregn'pris'incl'moms(pris'uden'moms)
0080   pris'med'moms:=pris'uden'moms*1.22
0090 ENDPROC beregn'pris'incl'moms
0100
0110 PRINT
0120 PRINT "pris incl. moms: kr. ",pris'med'moms
```

Programkørslen giver følgende udskrift på skærmen:

PRIS EXCL. MOMS: KR. 200

PRIS INCL. MOMS: KR. 244

I linie 30 lader vi maskinen udskrive en pris excl. moms (kr. 200).

I linie 50 har vi lagt dette tal ind som en parameter i et procedurekald. Det drejer sig om en reel parameter, som vi ønsker behandlet i proceduren.

De ønskede parametre skal anføres i såvel procedurekaldet som i procedurehovedet.

I procedurekaldet anføres de reelle talstørrelser eller deres variabelnavne, som skal gælde for dette ene kald. I procedurehovedet anføres derimod formelle variabelnavne, der kan "modtage" alle tænkelige reelle udformninger af tal eller variabler - blot disse svarer til de formelle (af de reelle og formelle parametre skal der altså anføres lige mange og samme slags hvert sted).

Ved parameteroverførsel opnås, at en og samme procedure på forskellige steder af programmet kan behandle forskellige talsæt.

Vi begynder med det enkleste tænkelige eksempel:

I linie 70 følger så proceduren, hvor hovedet indeholder parameteren "pris'uden'moms". Til denne formelle parameter overføres værdien af den reelle parameter.

I linie 80 bliver det overførte tal: 200 multipliceret med 1.22, og resultatet gemmes i variabelen "pris'med'moms", der bliver udskrevet i linie 120.

Ved hjælp af et par ændringer kan vi gøre programmet meget mere generelt:

## COMAL BRUGER NYT

```

0010 // eksempel 2
0020
0030 INPUT "pris excl. moms: kr. ": beløb
0040
0050 beregn'pris'incl'moms(beløb)
0060
0070 PROC beregn'pris'incl'moms(pris'uden'moms)
0080   pris'med'moms:=pris'uden'moms*1.22
0090 ENDPROC beregn'pris'incl'moms
0100
0110 PRINT
0120 PRINT "pris incl. moms: kr. ",pris'med'moms

```

Kører vi programmet, ser vi på skærmen:

PRIS EXCL. MOMS: KR. 200

PRIS INCL. MOMS: kr. 244

Linie 30 har vi gjort til en input-sætning, så vi kan få behandlet ethvert tal, som vi gemmer i variabelen "beløb".

Denne variabel bruger vi så som reel parameter i linie 50, hvorefter programmet udføres ganske som i forrige eksempel.

Resultatet af vort regnestykke (som proceduren foretager for os) er nu gemt i en ny variabel: "pris'med'moms".

Den oprindelige variabel: "beløb" har ikke ændret indhold.

Tilføjer vi:

```
0130 PRINT beløb
```

Så får vi udskriften 200, som var det oprindelige tal.

Den form for parametre, vi her har anvendt til overførsel, kaldes værdiparametre.

Udtrykket skal forstås således, at vi overfører en værdi (i vore eksempler selve talet eller værdien af variabelen "beløb") til proceduren, der så bearbejder det overførte tal.

Variablen "beløb" beholder sin oprindelige værdi, når overførslen er sket med en værdiparameter.

Men vi kan ændre parameteroverførslen, så variabelen for den reelle parameter får en ny værdi, når proceduren har behandlet den.

Så skal vi i stedet for værdiparametre anvende referenceparametre ved overførslen:

```

0010 // eksempel 3
0020 indestående:=500; rt:=6
0030
0040 rentetilskrivning(indestående,rt)
0050
0060 PROC rentetilskrivning(REF sum,REF procent)
0070   rente:=sum*procent/100
0080   sum:=sum+rente
0090   PRINT sum
0100 ENDPROC rentetilskrivning
0110
0120 PRINT indestående

```

## COMAL BRUGER NYT

På skærmen ser vi efter programkørslen:

530  
530

I eksempel 3 benytter vi en procedure, der foretager en rentetilskrivning baseret på de tal, der overføres fra variablene i de reelle parametre.

Vi har valgt de fæste tal:  
500 (kr.) og 6 (%).

Det er der ikke noget nyt i.

Det nye kommer i procedurehovedet, hvor vi foran hver af de formelle parametre har anført REF.

Hermed er vi gået over til at benytte referenceparametre (der også kaldes henvisningsparametre).

Som vi ser af programkørslen, har vor oprindelige variabel: "indestående" efter procedurens bearbejdning fået en ny værdi. "Indestående" har nemlig fået samme værdi som "sum".

Det skete ikke i eksempel 2, hvor den oprindelige variabel: "beløb" bevarede sin værdi. Men det er netop forskellen på, hvad man får ud af at anvende værdiparametre sammenlignet med, hvad anvendelsen af referenceparametre giver som resultat.

Med andre ord:

Anvendes overførsel med værdiparametre (by value), så ændres størrelsen af de reelle parametre ikke.

Anvendes overførsel med referenceparametre (by reference), så ændres størrelsen af de reelle parametre ved bearbejdningen i proceduren.

Børge Christensen forklarer det således i sin bog "Comal problemløsning og programmering 2. del" side 77 (idet dog variabelnavnene er ændret, så de passer i vort eksempel):

"Måske kan læseren have fornøjelse af følgende lille billede: En dreng, som hedder Karl, kaldes af familien Kalle. Den person, som resten af verden omtaler som Karl, hedder altså Kalle, så længe han færdes inden for hjemmets fire vægge. Det betyder f. eks., at når Kalle spiser for meget derhjemme, ser resten af verden Karl blive for tyk.

Vender vi tilbage til proceduren ovenfor, er det sådan, at den værdi, vi giver "sum" i proceduren, bliver værdien for "indestående" i programmet uden for proceduren, - - -".

Bemærk i øvrigt, at denne overførsel af en ny størrelse til den oprindelige parameter naturligvis kun er mulig, når man anvender det rette variabelnavn inde i proceduren, der hvor resultatet skal refereres tilbage til parameteren.

I linie 80 skal variabelen altså hedde "sum", fordi den formelle parameter i procedurehovedet er blevet kaldt "sum".

Til slut et eksempel på anvendelsen af denne enkle form for referenceparametre:

...fortsættes næste side

**C-B-N**



## COMAL BRUGER NYT

```

0010 // eksempel 4
0020 n:=0
0030 INPUT "indtast indestående: ": indestående
0040 PRINT
0050 INPUT "indtast procenten: ": rt
0060 PRINT
0070 INPUT "indtast ønsket slutbeløb: ": slut
0080 REPEAT
0090   n:=+1
0100   PRINT n, ". år:"
0110   PRINT "indestående før rentetilskrivning ", indestående
0120   rentetilskrivning(indestående, rt)
0130   PRINT "indestående efter rentetilskrivning ", indestående
0140   PRINT
0150 UNTIL indestående >= slut
0160
0170 PROC rentetilskrivning(REF sum, REF procent)
0180   rente:=sum*procent/100
0190   sum:=sum+rente
0200 ENDPROC rentetilskrivning

```

Programkørslen giver følgende udskrift på skærmen, når de anførte tal indtastes:

INDTAST INDESTÅENDE: 200

INDTAST PROCENTEN: 12

INDTAST ØNSKET SLUTBELØB: 300

1. ÅR:  
 INDESTÅENDE FØR RENTETILSKRIVNING  
 200  
 INDESTÅENDE EFTER RENTETILSKRIVNING 224

2. ÅR:  
 INDESTÅENDE FØR RENTETILSKRIVNING  
 224  
 INDESTÅENDE EFTER RENTETILSKRIVNING 250.88

3. ÅR:  
 INDESTÅENDE FØR RENTETILSKRIVNING  
 250.88  
 INDESTÅENDE EFTER RENTETILSKRIVNING 280.9856

4. ÅR:  
 INDESTÅENDE FØR RENTETILSKRIVNING  
 280.9856  
 INDESTÅENDE EFTER RENTETILSKRIVNING 314.703872

I dette eksempel benyttes repetition af en procedure, der ligner den i eksempel 3, til at udskrive, hvordan et beløb, der er sat til forrentning, vokser fra år til år indtil en forud fastsat størrelse.

PS. Proceduren må ikke stå inden for repeat-løkken.

PPS. Denne løsning af den stillede opgave er ikke den eneste mulige - endelige den fiksentel Den tjener kun til illustration af, hvorledes parameteroverførsel kan anvendes.

S. Chr. H.

**C.B.N.**

## COMAL BRUGER NYT

### BLAND TAL

Ikke så sjældent har man brug for et antal tal i vilkårlig rækkefølge, men uden gentagelser.

Eksempel: bankospil.

90 forskellige numre skal efter tur udtrækkes; men der må ikke være gentagelser.

Ofte laves programmer til dette formål efter følgende recept:

1. Vælg et random tal mellem 1 og 90 (begge inclusive).

2. Undersøg, om det senest fundne tal er lig et tidligere.

Hvis det er det, så er det ugyldigt og kasseres.

Hvis tallet derimod ikke er udtrukket før, er det gyldigt.

3. Vælg et nyt tal og undersøg det.

4. Fortsæt, indtil der er 90 gyldige tal.

Punkt 2 vil imidlertid tage længere og længere tid, efterhånden som antallet af gyldige tal stiger, for så bliver der jo færre og færre resterende muligheder at vælge imellem, og valget vil derfor meget ofte ramme et tidligere valgt tal, hvorefter det bliver kasseret.

Her er et nyt program, der ikke lider af denne skavank:

```
0010 // list "1.bland tal"
0020
0030 INPUT ""147""17""17""17""17""17""17""17""17"antal tal" : antal
0040 DIM a(antal)
0050
0060 FOR x:=1 TO antal DO a(x):=x
0070
0080 FOR x:=1 TO antal DO
0090   indexnummer:=RND(1,antal-x+1)
0100   PRINT USING #####: a(indexnummer);
0110   a(indexnummer):=a(antal-x+1)
0120 ENDFOR x
0130
0140 END ""
```

NB: £-tegnet skal på computeren erstattes med nummertegnet: #

Det nye program her opretter i linie 60 et array af x tal under variablen a.

Lad os for overskuelighedens skyld sætte antallet (variabel: antal) til 10.

Så ser arrayet sådan ud:

1 2 3 4 5 6 7 8 9 10.

I linie 90 genereres et tilfældigt tal (variabel: indexnummer) - i første gennemløb blandt tallene fra 1 til (10 - 1 + 1) d. v. s. fra 1 til 10 - i andet gennemløb fra 1 til (10 - 2 + 1) d. v. s. fra 1 til 9 - i tredje gennemløb fra 1 til (10 - 3 + 1) d. v. s. fra 1 til 8 o. s. v.

## COMAL BRUGER NYT

I linie 100 udskrives det tal i det første array, der har det indexnummer, der lige er blevet genereret.

Hvis det først genererede indexnummer er 7, udskrives tallet 7 som det første.

I linie 110 erstattes tallet a(7) med tallet a(antal - x + 1) dvs a(10), som jo er tallet 10.

Altså:

I linie 110 flytter tallet 10 ind på tallet 7's plads, så hvis der ved næste udtrækning tilfældigvis igen vælges indextallet 7, så er det tallet 10, der denne gang er udtrukket, og tallet 9 vil derefter indtage plads nr. 7.

På denne måde undgås gentagelser, og tempoet vil være ganske ens, hvad enten der skal udtrækkes 10 eller 1000 tal.

**G-B-N**

### BANKO

Den, der har forsøgt sig som opråber ved et bankospil, ved, hvor megen koncentration det kræver at vende brikken, så tallet kan læses, og derefter at anbringe den det rette sted på kontrolkortet.

Og så er der jo altid en, der spørger: "Hvad var det sidste nummer, der blev råbt op?"

Men på det tidspunkt er man jo dybt engageret i det næste og har aldeles glemt det foregående.

Fat mod!

Alle disse kvaler er nu fortid!

Her er programmet, der klærer det hele:

1. Det vælger et nummer mellem 1 og 90.
2. Det gentager (selvfølgelig) ikke et nummer.
3. Det placerer det udtrukne nummer i et skema (10 x 10 matrix) så man til enhver tid kan se, hvilke numre der er udtrukket, og hvilke der mangler.
4. Det noterer på skærmens næstsidste linie, hvilket løbenummer man lige har trukket.
5. Det noterer på skærmens sidste linie det til enhver tid senest udtrukne nummer.
6. Det hele foregår ved at trykke på en vilkårlig tast (lettest: mellemrumstangenten).
7. Trykker man s, så standser hele sagen, og man bliver spurgt, om man vil fortsætte.

```
0010 // list "1.banko"
0020
0030 DIM s$ OF 1, ned$ OF 23, ind$ OF 40
0040 DIM a(90), b(90)
0050 ned$:=CHR$(19) // cursor home
0060 ind$:=CHR$(0) // tom ved begyndelsen
0070
0080 FOR x:=1 TO 22 DO ned$:=ned$+CHR$(17) // cursor ned
0090
0100 FOR x:=1 TO 38 DO ind$:=ind$+CHR$(29) // cursor til højre
0110
0120 REPEAT
0130 PRINT CHR$(147)
0140 PRINT "Tryk en tast for nyt tal. Tryk s for stop"
0150 PRINT ned$(1:19)," Valg nummer      :"
0160 PRINT ned$(1:21)," Sidat valgte tal:"
```

## COMAL BRUGER NYT

```

0170  tal'valg
0180  tal:=0
0190  REPEAT
0200    tal:=1
0210    skriv'tal
0220    skema
0230  REPEAT
0240    s$:=KEY$
0250    UNTIL s$<>CHR$(0)
0260  UNTIL s$="s" OR tal=90
0270    fortsæt
0280 UNTIL slut
0290
0300 PRINT CHR$(147),ned$(1:10),"Ik for denne gang"
0310 END "På gensyn"
0320
0330 PROC tal'valg
0340   FOR x:=1 TO 90 DO a(x):=x
0350   FOR x:=1 TO 90 DO
0360     valg:=RND(1,90-x+1)
0370     b(x):=a(valg)
0380     a(valg):=a(90-x+1)
0390   ENDFOR x
0400 ENDPROC tal'valg
0410
0420 PROC skriv'tal
0430   PRINT ned$(1:19),ind$(1:30),tal
0440   PRINT ned$(1:21),ind$(1:30),"  "
0450   PRINT ned$(1:21),ind$(1:30),b(tal)
0460 ENDPROC skriv'tal
0470
0480 PROC skema
0490   y:=b(tal)
0500   ytiere:=y DIV 10
0510   yenere:=y MOD 10
0520   PRINT ned$(1:ytier+4),ind$(1:3*yenere+4),y
0530 ENDPROC skema
0540
0550 PROC fortsæt
0560   PRINT ned$(1:23),"Ønsker du at fortsætte j/n"
0570   REPEAT
0580     s$:=KEY$
0590     UNTIL s$ IN "jn"
0600     IF s$="j" THEN slut:=0
0610     IF s$="n" THEN slut:=1
0620 ENDPROC fortsæt

```

I linie 50 - 100 lægges fundamen-  
tet for brugen af en fremgangsmå-  
de, der svarer til "print at".

Se f. eks. anvendelsen i linie 430  
- 450.

I proceduren "skema" udnyttes sam-  
me ide til at placere de udtrukne  
numre i række og geled efter den  
plads, tallene anviser dem.

Ved at dividere med 10 findes,  
hvor langt tallet skal ned i ske-  
mæt, og divisionsresten angiver,  
hvor langt ind i rækken det skal  
placeres.

## COMAL BRUGER NYT

### LOAD FRA CAT

Et spørgsmål til redaktionen lød:

"Når jeg sidder og gennemser CAT'en for en diskette, ville det være meget nemmere, hvis jeg ikke behøvede at taste LOAD "PROGRAMNAVN" o. s. v., når jeg har fundet det program, jeg vil arbejde med. Kan I ikke lave sådan et program?"

Her er det!

Når programmet RUN'es, kommer de første 20 programnavne på CAT'en til syne på skærmen.

Til venstre for program nr. 11 ses en pil. Den kan flyttes opad med tasten O eller nedad med tasten N. Begge taster er repeterende.

Kører man pilen neden for rækken af programnavne, udskiftes de øverste 10 med dem, der før stod nederst, og i bunden rykker nye navne ind.

Når man har fået pilen placeret ud for det program, man skal bruge trykker man på RETURN-tasten.

Straks kommer ordet RUN i stedet for pilen, og datamaskinen LOAD'er og RUN'er programmet.

Der er nogle "regler", der skal overholdes:

Bruger man et dobbelt diskdrive som f. eks. CBM 4040, skal disketten ligge i port 0 (højre).

Kun programmer, der i CAT'en er anført med PRG kan LOAD'es og RUN'es på denne måde.

Programmet er udformet sådan, at det kan anvendes på såvel COMMODORE 64 som på CBM maskinerne 4032, 8032 o. s. v.

Den ene af linierne 450 - 460 skal gøres til en REMARK v. hj. af de to skråstreger (//).

I eksemplet er det linie 460, der fungerer, hvilket vil sige, at programmet da er beregnet for en CBM maskine.

Den, der vil udnytte dette program til nem indlæsning og start af programmer, bør anbringe det som første program på sine disketter og starte det med SHIFT RUN/STOP.

S. J.

```
0010 // list "1.loader c 2.00"
0020
0030 DIM a$(0:144) OF 40, q$ OF 1
0040 OPEN FILE 2,"u8:$/s0",READ
0050 x:=0
0060
0070 REPEAT
0080   i$:=GET$(2,32)
0090   IF NOT "blocks free." IN i$ THEN
0100     pil:=CHR$(34) IN i$; pil:-1
0110     a$(x):=" "+i$(pil:pil+18)+" ! "+i$(pil+20:32)
0120     x:+1
0130   ENDIF
0140 UNTIL "blocks free." IN i$
0150
0160 inde:=x-1
0170 CLOSE FILE 2
0180
0190 REPEAT
0200   a$(v):=SPC$(32)
0210   x:+1
0220 UNTIL x MOD 20=0
0230
0240 PAGE
0250 PRINT "o = op      n = ned      <return> = indlæs"
0260 først:=1; sidst:=20
0270 IF inde<20 THEN sidst:=inde
```

## COMAL BRUGER NYT

```

0280
0290 REPEAT
0300   udskriv(først,sidst)
0310   flyt
0320 UNTIL FALSE
0330
0340 PROC udskriv(f,s)
0350   FOR x:=f TO s DO
0360     PRINT AT x+3-f.0: s$(x)
0370   ENDFOR x
0380 ENDPROC udskriv
0390
0400 PROC flyt
0410   s:=13
0420   IF inde<20 THEN s:=INT(inde/2)+2
0430   REPEAT
0440     PRINT AT s,0: "  ->"
0450     // IF PEEK(203)=64 THEN // C 64
0460     IF PEEK(166)=255 THEN // CBM
0470       REPEAT
0480         g$:=KEY$
0490         UNTIL g$ IN "on" OR g$=CHR$(13)
0500       ENDIF
0510       FOR x:=1 TO 150 DO NULL
0520       IF g$=CHR$(13) THEN indlæs
0530       PRINT AT s,0: "    "
0540       IF g$="n" AND s+først-3<inde THEN s:=+1
0550       IF g$="o" AND s+først-4>0 THEN s:=-1
0560     UNTIL s=2 OR s=23
0570     IF s=23 THEN
0580       først:=+10
0590       sidst:=først+19
0600     // IF sidst>inde THEN sidst:=inde

```

```

0610   ENDIF
0620   IF s=2 THEN
0630     først:=-10
0640     IF først<1 THEN først:=1
0650     sidst:=først+19
0660   ENDIF
0670 ENDPROC flyt
0680
0690 PROC indlæs
0700   PRINT AT s,0: "run "
0710   læs$:=s$(først+s-3)
0720   pil:=CHR$(34) IN læs$(8:26)
0730   læs$:=læs$(7:pil+6)
0740   CHAIN læs$
0750 ENDPROC indlæs

```

## RENTEDAGE

Pengeinstitutternes specielle måde at beregne antal rentedage kan nok volde visse kvaler

Her er et program, der tager hensyn til, at alle rentemåneder er på 30 dage (se linierne 80 og 130). og at et rentear har 360 dage, uanset hvad kalenderen siger.

Hvis der vælges datoer, så et årsskifte overskrides finder programmet selv ud af det og udregner det rigtige dageantal (se linie 160).

Datoerne skal skrives på gammeldags vis med dag før måned og adskilt af skråstreg, som eksemplerne i linie 50 og 100 viser.

## COMAL BRUGER NYT

```

0010 // list "1.rentedage"
0020
0030 REPEAT
0040   PAGE
0050   INPUT AT 8,0: "beg. dato (5/12) ": begdato$
0060   skel:="/" IN begdato$
0070   begdag:=VAL(begdato$(1:skel-1))
0080   IF begdag=31 THEN begdag:=30
0090   begmåned:=VAL(begdato$(skel+1:LEN(begdato$)))
0100   INPUT AT 10,0: "slut dato (5/12) ": slutdato$
0110   skel:="/" IN slutdato$
0120   slutdag:=VAL(slutdato$(1:skel-1))
0130   IF slutdag=31 THEN slutdag:=30
0140   slutmåned:=VAL(slutdato$(skel+1:LEN(slutdato$)))
0150   dage:=(slutmåned-begmåned)*30+(slutdag-begdag)
0160   IF dage<0 THEN dage:=+360
0170   PRINT AT 12,0: "antal rentedage      ",dage
0180   pause
0190 UNTIL FALSE
0200
0210 PROC pause
0220   PRINT AT 24,10: "tryk en tast for fortsæt"
0230   WHILE KFY$="" DO NULL
0240 ENDPROC pause

```

## LANGSOMME BOGSTAVER

Ved demonstrationer af forskellig art kan man "krydre" teksten på forskellig vis f. eks. ved at skrive den langsomt hen over skærmen.

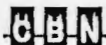
Her er et meget enkelt program, der klarer den sag.

Teksten indskrives i variabelen tekst\$ i linie 40, og skrivetempoet bestemmes i linie 80.

```

0010 // save "langsomme bogst"
0020 // COMAL 2.00
0030 page
0040 tekst$:="Langsomme bogstaver"
0050 PRINT TAB(4)
0060 FOR n:=1 TO LEN(tekst$) DO
0070   PRINT tekst$(n),
0080   FOR p:=1 TO 200 DO NULL
0090 ENDFOR n
0100 END ""

```



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

Tryk:  
tekst & tryk  
Horsens

Abonnement:  
Mindegade 42  
8700 Horsens

# COMAL BRUGER NYT

C B G  
COMAL BRUGER GRUPPE  
Mindegade 42  
8700 Horsens  
Tlf. 05 - 62 15 67  
Giro 1 75 23 75

## INDMELDELSE I C B G

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

Samtidig indbetales kr. 90,00 - prisen for den løbende samling = 6 numre af "COMAL BRUGER NYT".

Når beløbet er indgået til bladet, sendes omgående de hidtil udkomne numre af denne samling.

Ønskes de tidligere 6 numre (1. samling) af CBN, afkrydses dette på blanketten, og der indbetales yderligere kr. 90,00.

Indbetalingen kan ske pr. check eller ved overførsel til:

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

## INDMELDELSESBLANKET

|       |                                                        |            |
|-------|--------------------------------------------------------|------------|
| ***** |                                                        | *****      |
| *     |                                                        | *          |
| *     | Undertegnede                                           | *          |
| *     |                                                        | *          |
| *     | Navn: .....                                            | ... *      |
| *     |                                                        | *          |
| *     | Adresse:                                               | *          |
| *     | Gade: .....                                            | .... *     |
| *     |                                                        | *          |
| *     |                                                        | *          |
| *     | Postnummer: ..... By: .....                            | ..... *    |
| *     |                                                        | *          |
| *     |                                                        | *          |
| *     | Att.: .....                                            | *          |
| *     |                                                        | *          |
| *     | Indmelder sig i COMAL BRUGER GRUPPE (kr. 90,00) .....  | *          |
| *     |                                                        | Sæt *      |
| *     | Bestiller 1. saml. (nr 1 - 6) af CBN (kr. 90,00) ..... | venligst * |
| *     |                                                        | *          |
| *     | Beløbet: ..... er vedlagt i check .....                | X *        |
| *     | kr. ....                                               | *          |
| *     | er indbetalt pr. giro .....                            | *          |
| *     |                                                        | *          |
| ***** |                                                        | *****      |