



INSTITUT FOR INFORMATIK
Danmarks Lærerhøjskole

COMAL 10

PRÆSENTATION AF ET STRUKTURERET PROGRAMMERINGSSPROG
COMAL-80 I RC-VERSIONEN

af

Erik Nielsen og Erik Norup

Noter no. 4
København - 1985

INSTITUT FOR INFORMATIK
Danmarks Lærerhøjskole
Emdrupvej 115B
2400 København NV
telefon 01 696633

©1985 Institut for Informatik og forfatter(e)
Copenhagen
1. Udgave, 2. Oplag

Noteserien er en af flere serier, hvormed instituttets orienteringspligt søges opfyldt. Serien udgives for at orientere om igangværende kursusudvikling, og for at gøre undervisningsmateriale af mere almen karakter tilgængelig for provinsafdelingerne ved Danmarks Lærerhøjskole. Publikationerne rekvireres skriftligt så længe lager haves.

ISBN: 87 88289 26 5

COMAL 10 er et produkt af instituttets kursusudvikling i læseåret 1984/85.

Med instituttets oprettelse føltes det påtrængende at give kurset "Grundlæggende begreber i informatikken" en ny profil. Det måtte understreges, at kurset ikke kun var en del af et forløb til fremskaffelse af undervisere til faget datalære, men også skulle betragtes som et kursus for alle lærere, uanset hvilken faggruppe de tilhører.

I "Beskrivelse af informatikkurserne 85/86" blev det formuleret at:

"Alle lærere bør have et så solidt kendskab til informatikområdet, at de på kvalificeret vis kan tage stilling til anvendelse af informationsteknologi i folkeskolen. Herudover skal de være i besiddelse af en brugererfaring, der sætter dem i stand til at anvende teknologien i skolens enkelte fag".

For at opnå disse kvalifikationer, mener vi, der må sættes ind på følgende områder:

- (1) Generel viden om datamaskinen, dens anvendelsesmuligheder og begrænsninger.
- (2) Generel viden om samfundets brug af EDB-teknologi.
- (3) Generel viden om de konsekvenser en udstrakt brug af informationsteknologi kan medføre, såvel for samfundet som for den enkelte. Herunder især konsekvenserne for skolesektoren.

COMAL 10 er en del af et forløb, der retter sig mod punkt (1).

I dette forløb er der lagt vægt på at betragte datamaskinen som en åben konstruktion, der, forsynet med forskellige fortolkere, er i stand til at løse forskellige opgaver.

Forskellige fortolkere medfører, at der må anvendes forskellige "sprog", men det væsentlige:

- at dataskinen fungerer ved hjælp af et program,
- at det sprog, programmet er skrevet i, giver begrænsninger i anvendelsesområdet,
- at problemløsning kræver en gradvis nedbrydning i mindre problemkomplekser,
- at ikke alle problemer egner sig for databehandling,

skulle gerne være den viden, der udkrystalliseredes ved arbejdet.

Det er vor faste overbevisning, at denne viden kun kan opnås, hvis de studerende selv arbejder med programmering og problemløsning ved hjælp af datamaskinen.

Vi har arbejdet med 3 forskellige fortolkere.
For at undgå problemer ved overgangene, har vi kun valgt fortolkere, der tillader struktureret programmering.

I første del af forløbet blev der arbejdet med en MYRESNAK-fortolker.

Myresnak er et sprog, der kan anvendes i mange pædagogiske sammenhænge.

I denne forbindelse udnyttes udelukkende, at der er tale om et simpelt struktureret sprog, som inden for et begrænset univers med danske kommandoer og nøgleord, blandt andet giver de studerende erfaringer med problemløsning, programmering, indtastning, fejlretning og brug af ydre enheder i arbejdet med grafik på dataskærmen.

Det karakteristiske ved programmering i Myresnak er brugen af procedurer og parametre.

Som grundbog blev anvendt Viggo Sadolin : Myresnak (Teknisk Forlag).

Erfaringerne var gode. De studerende gav efter forløbet udtryk for,

- at tastaturskrækken var væk,
- at de fandt arbejdet spændende,
- at produkterne var tilfredsstillende.

Forløbet strakte sig over 6 kursusgange a 3 lektioner.

I den næste del af forløbet anvendtes en SKOMAL-fortolker.

Skomal giver mulighed for at arbejde med løsning af problemer inden for skolens pensum i regning/matematik. Fortolkeren er opbygget på en sådan måde, at grænsefladen ligner Myresnaks mest muligt, og det giver plads til at koncentrere sig om den lagermodel af datamaskinen, der introduceres.

Under afvikling af programmerne kan bevægelserne i lageret følges på en skematisk opstilling på datamaskinens skærm.

Til forløbet brugte vi noter fra Matematisk Institut, men da noterne i mellemtiden er blevet udgivet, vil vi fremover anvende Viggo Sadolin : Skomal (Teknisk Forlag).

Det er vores indtryk, at arbejdet med den dynamiske lagermodel gav en forståelse, der lettede overgangen til den næste fase.

Forløbet strakte sig over 2 kursusgange a 3 lektioner.

I den sidste del af forløbet præsenteredes en COMAL 80-
fortolker.

Comal 80 er et højere struktureret programmeringssprog, med
mangfoldige anvendelsesområder og med en rimelig simpel
grænseflade.

Formålet med præsentation af Comal 80 er, at give de studerende

- overblik over karakteristiske anvendelsesområder for,
- færdighed i at læse, forstå og tilrette programmer,
skrevet i,
- mulighed for at erkende datamaskinens begrænsninger,
ved anvendelse af,
- mulighed for at opnå en vis programmeringsfærdighed i,

et højere programmeringssprog.

COMAL 10 er vores bud på, hvordan disse mål kan opfyldes inden for
en rimelig tid.

COMAL 80 præsenteres struktureret. Det betyder at et program
altid består af et hovedprogram, der afspejler den første
problemnedbrydning, samt en række procedurer.
I hovedprogrammet indgår styrestrukturer og procedurekald, men
ingen tildelinger.

Vi er klar over at det kan støde "erfarne programmører" at se
procedurer indeholdende en enkelt programlinje, men vi mener
at fordelene ved den type programmers læselighed og mulighed
for udvidelser/ændringer retfærdiggør det lidt større
skrivearbejde.

For at kunne give indtryk af anvendelsesområder, starter alle
notesæt med et færdigt program. I de to første sæt skal
programmet indtastes, men i de følgende sæt hentes programmet
på disketten.

De studerendes opgave bliver så at læse, forstå, ændre og udvide
dette program.

Det er blevet præciseret at det "obligatoriske" pensum er nået,
når første opgave 1 sættet er løst. At de to følgende opgaver er
at betragte som henholdsvis bredde og dybdeopgaver, og at
arbejdet med dem er et tilbud til de studerende, der har ønske
om at opnå en større programmeringsrutine, og som har mulighed
for at arbejde med en datamaskine hjemme eller på skolen.

Noterne har været afprøvet i samarbejde med ca 100 studerende.
I praksis viste det sig at omkring halvdelen arbejdede videre
med opgave 2 og en fjerdedel med opgave 3.

Vi har i noterne udnyttet, at de studerende i forvejen har arbejdet med Myresnak og Skomal, og derfor er fortrolige med nedbrydning af problemer, opstilling af procedurer, afvikling af programmer og brug af ydre enheder.

For at vænne de studerende til på egen hånd at sætte sig ind i et programmeringssprogs finesser, har vi i starten af hvert notesæt blot skrevet hvilke kommandoer, nøgleord og strukturer, der kan anvendes ved opgaveløsningen. Det kræver naturligvis at de har en manual eller opslagsbog til rådighed. På vore kurser anvendte vi Mogens Lyster Knudsen: Struktureret Basic, Comal 80 versionen (Gad).

Noterne er anvendt i et undervisningsforløb, hvor der i den første lektion blev demonstreret nedbrydningen af et problem, under anvendelse af de strukturer og teknikker, der skulle arbejdes med i de sidste 2 lektioner. Herefter har de studerende haft adgang til i hold a 2 at arbejde ved datamaskinen.

Der har været arbejdet med Metanic-comal, og noterne er derfor udviklet hertil.

For at imødekomme et udbredt ønske fra provinsafdelingerne, har vi dog samtidigt omskrevet noterne til anvendelse på maskiner, der betjener sig af RC-comal og CBM-comal, den sidste både indlæst fra cartridge og bånd/ diskette til Commodore 64.

Programmerne til alle 4 versioner vil kunne fås på diskette.

Det er vores håb, at vi med disse noter kan bidrage til en afklaring af den standende diskussion om omfanget af programmering på kurser, der henvender sig til samtlige lærergrupper i skolen.

Erik Nielsen Erik Norup.

COMAL 1.

I denne øvelse behandles:

Kommandoer : LIST, RUN, EDIT, NEW, LOAD, SAVE, DEL, DIR,
LIST"PRINTER"

Nøgleord : INPUT, PRINT, PRINT CHR\$(12), //, PROC,
ENDPROC, TAB, SELECT OUTPUT"PRINTER",
SELECT OUTPUT"CONSOLE", EXEC, MARGIN,
DIM(af en tekstvariabel).

(1) . Indtast følgende program.

```
10 // VISITKORT
20 DIM NAVN$ OF 20, ADR$ OF 20, BY$ OF 20
30 //
40 // HOVEDPROGRAM
50 PRINT CHR$(12)
60 EXEC INDLÆS
70 EXEC UDSKRIV
80 //
90 //
100 // PROCEDURER
110 PROC INDLÆS
120   NAVN$:="Erik"
130   ADR$:="Viben 61"
140   BY$:="6270 Tønder"
150 ENDPROC INDLÆS
160 //
170 PROC UDSKRIV
180   PRINT"*****"
190   PRINT
200   PRINT NAVN$
210   PRINT ADR$
220   PRINT BY$
230   PRINT
240   PRINT"*****"
250 ENDPROC UDSKRIV
```


- (1.1). Ret PROC UDSKRIV, så visitkortet bliver helt indrammet af stjerner, og udskrives midt på skærmen.
- (1.2). Skift Eriks data i PROC INDLÆS ud med dine egne. Kør det nye program.
- (1.3). Udvid PROC UDSKRIV, så udskriften kommer på printeren.
- (1.4). Ret PROC INDLÆS, så "maskinen" selv beder om data.
- (1.5). Gem programmet på disketten.

(2). Faktura.

Programmet skal udskrive en faktura, med angivelse af beløb, moms og samlet pris.

- (2.1). Opstil følgende procedurer:
 - INDLÆS, der beder om antal og stykpris.
 - BEREGN, der udregner pris, moms og samlet pris.
 - UDSKRIV, der udskriver en faktura.
- (2.2). Indtast hovedprogram og procedurer. Afprøv programmet.
- (2.3). Udbyg PROC INDLÆS, så der spørges om dato og kundens navn. Oplysningerne skal udskrives på fakturaen.
- (2.4). Udvid PROC UDSKRIV, så fakturaen påføres et ekspeditionsgebyr på 10 kr.
- (2.5). Udvid PROC UDSKRIV, så fakturaen udskrives på printeren.
- (2.6). Gem programmet på disketten.

(3). Momsafregning.

En forretning har brug for et program, der udskiller dagens salg i henholdsvis indgået moms og indgået indtægt.

Opstil og indtast et program, der beder om dato og dagens salg. Herefter udskrives en blanket til bogholderiet indeholdende de udskilte beløb.

COMAL 2.

I denne øvelse behandles:

Kommandoer : AUTO
Nøgleord : INPUT"<tekst>"
Strukturer : IF <udsagn> THEN
 <programdel>
 ENDIF

IF <udsagn> THEN
 <programdel>
ELSE
 <programdel>
ENDIF

(1) TIPSGEVINSTER.

Ved tipsgevinster er de første 200 kr skattefri. Af resten trækkes 15% fra til skattevæsenet.

(1.1) Indtast følgende program:

```
10 // TIPSGEVINST
20 //
30 //HOVEDPROGRAM
40 EXEC INDLÆS
50 EXEC BEREKN
60 EXEC UDSKRIV
70 //
80 //
90 // PROCEDURER
100 PROC INDLÆS
110   PRINT CHR$(12)
120   INPUT"Tipsgevinstens størrelse : ": GEV
130 ENDPROC INDLÆS
140 //
150 PROC BEREKN
160   IF GEV>=200 THEN
170     SK:=(GEV-200)*0.15
180   ELSE
190     SK:=0
200   ENDIF
210   UDB:=GEV-SK
220 ENDPROC BEREKN
230//
240 PROC UDSKRIV
250   PRINT
260   PRINT "Tipsgevinst   : ";GEV;" kr."
270   PRINT "Skat         : ";SK;" Kr."
280   PRINT "Udbetaling    : ";UDB;" Kr."
290 ENDPROC UDSKRIV
```

- (1.2) Afprøv programmet for forskellige gevinster.
- (1.3) Indsæt TAB sætninger i PROC UDSKRIV, så udskriften kommer midt på skærmen.
- (1.4) Udvid PROC UDSKRIV, så der anbringes en streg mellem skat og udbetaling samt dobbeltstreg under udbetaling.
- (1.5) Skatteministeren overvejer at sætte skatteprocenten op. Lav en ændring i PROC INDLÆS, så der spørges om skatteprocent. (Husk at rette sætning 170).
- (1.6) De samme overvejelser gælder den skattefri bundgrænse. Lad PROC INDLÆS spørge om denne bundgrænse.
- (1.7) Udvid PROC UDSKRIV, så udskriften kommer på printerens.

(2) OVERSKYDENE SKAT.

En skatteyder, der har betalt for meget i skat, får kun den overskydende skat udbetalt, såfremt den er 25 kr eller derover. Hvis den overskydende skat udbetales, får skatteyderen desuden udbetalt et tillæg på 5% af den overskydende skat.

- (2.1) Opstil følgende procedurer:
 - INDLÆS, der spørger om størrelsen af den overskydende skat
 - BEREGN, der indeholder en forgrening, og som tillægger 5%, hvis den overskydende skat er på mere end 25 kr.
 - UDSKRIV, der udskriver en passende meddelelse til skatteyderen.
- (2.2) Indtast hovedprogram og procedurer.
- (2.3) Udvid PROC UDSKRIV, så udskriften kommer på printerens.

(3) RESTSKAT.

- (3.1) Udarbejd et restskatteprogram, hvor inddata udgør det beløb, som er betalt for lidt i skat, og hvor uddata er en passende meddelelse til skatteyderen om restskattens størrelse, samt om hvordan denne gæld skal afvikles.
Der tillægges 8% i rente af det skyldige beløb. Er beløbet over 5000 kr, skal gælden betales i 3 rater.
- (3.2) Brug programmet til at eksperimentere dig frem til det beløb, du højst må have betalt for lidt i skat, hvis du skal undgå at betale de 3 rater i januar, februar og marts.

COMAL 3.

I denne øvelse behandles:

Kommandoer : RENUMBER
Nøgleord : PRINT AT, AND, OR, NOT, PRINT USING,
INPUT AT

(1) Statsskat.

Statsskatten beregnes for 1984 på baggrund af den skattepligtige indkomst således:

af de første 108200 kr betales 19.75%.

fra 108200 kr til 179000 kr betales 21369.50 kr plus 34.15% af det beløb, der overstiger 108200 kr.

over 179000 kr betales 45547.70 kr plus 44.95% af det beløb, der overstiger 179000 kr.

- (1.1) Som bilag er der anbragt et program, der beregner statsskatten. Læs programmet igennem og find ud af, hvordan det virker.
- (1.2) Programmet ligger på disketten under navnet ENSKAT. Hent det ind i arbejdslageret og prøv det med forskellige inddata.
- (1.3) Af den skattepligtige indkomst skal der også betales kommune, amts og kirkeskat.
Lav en ændring af PROC INDLÆS, så der spørges om kommune-, amts- og kirke-skatteprocenter.
- Tilføj dernæst tre procedurer, KOMMUNESKAT, AMTSSKAT og KIRKESKAT, der udregner skattebeløbene.
(Bemærk at disse skatteformer ikke er progressive).
- Udvid hovedprogrammet, så disse procedurer udføres.
- Udvid endelig PROC UDSKRIV, så stats, kommune, amts og kirkeskat udskrives sammen med den samlede skat.
- (1.4) Udvid PROC UDSKRIV, så udskriften kommer på printeren.

Der er ikke taget hensyn til skatteværdi af personfradrag.

(2) Brevporto.

Taksterne for brevporto fremgår af nedenstående tabel:

indtil 20g	:	2.70 kr
20 - 100g	:	3.70 kr
100 - 250g	:	5.50 kr
250 - 500g	:	9.00 kr
500 - 1000g	:	13.00 kr
over 1000g	:	kan ikke sendes som brev.

(2.1) Opstil følgende procedurer:

INDLÆS, der beder om brevets vægt

BEREGN, der udregner porto

UDSKRIV, der udskriver portobeløbet

(2.2) Indtast hovedprogram og procedurer. Afprøv programmet.

(2.3) Ved anbefalede breve opkræves 13 kr ekstra.

Dimensioner en variabel, SVAR\$ og lad PROC INDLÆS spørge om brevet er anbefalet.(ja/nej).

Udvid PROC BEREGN, så der lægges 13 kr til portoen, hvis det er et anbefalet brev.

(3) Rabat.

Et firma giver rabat ved store ordrer.

10% hvis antallet overstiger 25,

25% hvis antallet overstiger 100.

Opstil og indtast et program, der beder om antal og stykpris, hvorefter der udskrives en faktura som nedenstående:

25/11-84

200 stk af 10 kr	:	2000 kr
Rabat 25%	:	500 kr

Samlet pris	:	1500 kr
		=====

COMAL 4.

I denne øvelse behandles:

Nøgleord : RANDOMIZE, RND, FALSE
Strukturer : REPEAT
 <programdel>
 UNTIL <udsagn>

(1). Additionstræning.

- (1.1) Som bilag er der anbragt et program, der stiller opgaver i addition indenfor talområdet 1 til 20.
Læs programmet igennem og find ud af, hvordan det virker.
- (1.2) Programmet ligger på disketten under navnet ENPLUS. Hent det ind i arbejdslageret og prøv det nogle gange.
- (1.3) Ret de forskellige procedurer, så der stilles opgaver i multiplikation i stedet.
- (1.4) Udvid PROC OPGAVE med en tæller. Lav en ændring så der højst kan gættes 3 gange.
Udvid PROC KONTROL, så det rigtige svar skrives på skærmen, hvis der har været svaret forkert 3 gange i træk.
Svaret skrives sammen med en passende meddelelse efter at skærmen er blevet ryddet.
- (1.5) Udvid PROC KONTROL, så der udover RIGTIGT/FORKERT skrives hvilken gang der er blevet forsøgt. Meddelelsen skrives midt på skærmens øverste linie.
- (1.6) Afslut PROC OPGAVE med et spørgsmål, om der ønskes flere opgaver. Erstat derefter nøgleordet FALSE i hovedprogrammet med et udsagn, så programmet sluttes, hvis svaret er benægtende.

(2). tabeltræning.

Et program skal bruges til at træne 7 tabellen.

(2.1) Opstil følgende procedurer:

VÆLG, der finder et tal mellem 1 og 9.

FACIT, der udregner facit som (tallet * 7).

OPGAVE, der stiller opgaven, og beder om elevens svar.
Elev skal have tre forsøg.

KONTROL, der kaldes fra PROC OPGAVE, og kontrollerer
elevens svar. Proceduren udskriver en passende
meddelelse.

(2.2) Indtast hovedprogram og procedurer. Afprøv programmet.
(husk RANDOMIZE)

(2.3) Udvid PROC OPGAVE med et spørgsmål om der ønskes flere
opgaver. Anbring en REPEAT struktur i hovedprogrammet,
så programmet afsluttes, hvis svaret er nej.

(2.4) Udvid PROC VÆLG, så der spørges efter, hvilken tabel der
skal trænes. Ret de øvrige procedurer i overensstemmelse
med dette.

(3). Kontrol af addition.

Et program skal anvendes af elever til at kontrollere deres
additionsopgaver.

Maskinen skal bede om de 2 tal samt elevens facit. Hvis facit
er forkert skal det rigtige svar ikke udskrives. Det meddeles
blot at facit var forkert. Hvis facit er rigtigt meddeles dette.

Efter hvert svar, skal der spørges om der er flere opgaver.
Når der svares nej skal der udskrives hvor mange stykker, der
er kontrolleret og hvor mange af disse, der var korrekte.

COMAL 5.

(1). Afbetaling med fast løbetid.

Når man køber på afbetaling, eller når anden gæld skal afvikles, er forholdet som regel det, at ydelsen er konstant, således at renten falder og afdragene stiger hen gennem den fastlagte løbetid.

Følgende formel kan anvendes ved beregning af ydelsen:

$$\text{ydelse} := \text{gæld} * r / 100 / (1 - 1 / (1 + r / 100)^n)$$

hvor r er renten pr termin i %, og n er antallet af terminer.

- (1.1) Som bilag er anbragt et program, der beregner ydelsen, og udskriver en tabel over afviklingen af restgælden. Læs programmet igennem og find ud af, hvordan det virker.
- (1.2) Programmet ligger på disketten under navnet ENAFBETA. Hent programmet ind i arbejdslageret, og afprøv det med forskellige inddata.
- (1.3) Ved en afbetaling skal der som regel erlægges en vis procentdel i udbetaling. Udvid PROC INDLÆS med et spørgsmål om udbetalingsprocenten, og udvid PROC YDELSE, så udbetalingen beregnes og fratrækkes gælden inden ydelsen beregnes.
- (1.4) Udvid PROC BEREKN så den samlede rente beregnes.
- (1.5) Tilføj en procedure, der udskriver den samlede rente. Udvid hovedprogrammet, så denne procedure udføres.
- (1.6) Udvid hovedprogrammet, så udskriften kommer på printeren.

(2). opsparing.

Ved køb af et hus, skal der bruges 100000 kr. til udbetaling, flytning og handelsomkostninger. Hvert år sættes et fast beløb i banken. Hvornår kan huset købes?

(2.1) Et program, der skal løse ovenstående opgave kan indeholde følgende procedurer:

INDLÆS, hvor der spørges om navn, årlig opsparing og renten i % pr år.

BEREGN, hvor rentetilskrivning og ny saldo beregnes.

UDSKRIV, der kaldes fra PROC BEREGN og udskriver navn samt en tabel over termin, indbetaling, rente og saldo.

(2.2) Udvid hovedprogrammet, så udskriften kommer på printeren.

(3). Afbetaling med fast ydelse.

I nogle tilfælde vælger man selv, med hvilken ydelse man vil afvikle en gæld.

Opstil et program, der efter indlæsning af gæld, ydelse og renten i procent pr termin følger afviklingen af gælden.

Dette program kan f.eks anvendes til afvikling af pantebrevsgæld.

COMAL 6.

I denne øvelse behandles:

Nøgleord : MOD, DIV
Strukturer : FOR <variabel> := <tal> TO <tal> DO
 <programdel>
 NEXT <variabel>

(1). Simulering af møntkast.

- (1.1) Som bilag er anbragt et program, der simulerer kast med en mønt.
Læs programmet igennem og find ud af, hvordan det virker.
- (1.2) Programmet ligger på disketten under navnet ENMØNT.
Hent programmet ind i arbejdslageret og prøv det nogle gange.
- (1.3) Tilføj en ny procedure BEREKN, der beregner frekvensen for plat og krone, og udvid PROC UDSKRIV, så frekvenserne udskrives.
Udvid hovedprogrammet, så proceduren udføres.
- (1.4) Tilføj en procedure UDVALG, der spørger om udskriften skal på printer eller skærm, og som alt efter svaret vælger printer eller skærm som uddatamedie.
Udvid hovedprogrammet, så proceduren udføres
- (1.5) Udvid PROC UDFØR, så maskinen udskriver, hvor mange møntkast, der er udført. Brug PRINT AT til at få skrevet tallet midt på skærmen.
- (1.6) Lav en ændring af PROC UDFØR, så antallet af udførte møntkast kun udskrives hver gang der er udført 100 kast.

(2). Kast med en terning.

Der skal konstrueres et program, der simulerer kast med en terning , og som beregner frekvensen af seksere.

(2.1) Opstil følgende procedurer:

INDLÆS, der spørger om antallet af kast.

UDFØR, der kaster terningen det ønskede antal gange, opsummerer antallet af seksere og beregner frekvensen af seksere.

UDVALG, der spørger om udskriften skal på skærm eller printer, og som udfører valget.

UDSKRIV, der udskriver hyppighed og frekvens.

(2.2) Indtast hovedprogram og procedurer, husk RANDOMIZE. Afprøv programmet.

(2.3) Udvid PROC INDLÆS, så der spørges om hvilket øjental, der ønskes undersøgt. Udskift tallet 6 med det vilkårlige øjental i de andre procedurer.

(3). Kast med 2 terninger.

Et program skal bruges til at undersøge frekvensen af en given sum ved kast med 2 terninger.

Maskinen skal bede om antal kast og om hvilken sum, der ønskes undersøgt.

Der skal udskrives antal udførte kast med de 2 terninger, samt hyppighed og frekvens for den ønskede sum.

COMAL 7.

I denne øvelse behandles :

Nøgleord : READ, DATA, RESTORE, EOD, SQR
DIM(af en taltabel)

(1). Statistisk analyse 1.

- (1.1) Som bilag er anbragt et program, der foretager en statistisk analyse af et observationssæt. Læs programmet igennem og find ud af, hvordan det virker.
- (1.2) Programmet ligger på disketten under navnet ENSTAT. Hent det ind i arbejdslageret og prøv det.
- (1.3) Tilføj nogle flere datasætninger, og prøv det igen.
- (1.4) Tilføj en procedure, der bestemmer den mindste observation. Ret hovedprogram og PROC UDSKRIV2, så oplysningen udskrives.
- (1.5) Tilføj en procedure, der bestemmer variationsbredden. (Forskellen mellem største- og mindsteværdi). Ret hovedprogram og PROC UDSKRIV2, så oplysningen udskrives.
- (1.6) Tilføj en procedure, der beregner variansen. (Hver enkelt observation fratrækkes gennemsnittet og kvadreres. De herved fundne tal lægges sammen og divideres med antallet af observationer. Teknikken fra PROC SUM kan anvendes). Ret hovedprogram og PROC UDSKRIV2, så oplysningen udskrives.
- (1.7) Tilføj en procedure, der beregner standardafvigelsen. (Kvadratroden af variansen). Ret hovedprogram og PROC UDSKRIV2, så oplysningen udskrives.
- (1.8) Gem programmet på disketten.

(2). Statistisk analyse 2.

En række statistiske mærketal findes lettest, når observationerne er sorteret efter størrelse.

- (2.1) Tilføj følgende procedure til programmet fra (1) med passende linienumre:

```
PROC SORTER
  FOR I:=1 TO ANTAL-1 DO
    FOR J:=I+1 TO ANTAL DO
      IF A(I)>A(J) THEN
        GEM:=A(I)
        A(I):=A(J)
        A(J):=GEM
      ENDIF
    NEXT J
  NEXT I
ENDPROC SORTER
```

- (2.2) Opstil og indtast følgende procedure :

MEDIAN, der bestemmer den midterste observation.
Hvis der ikke findes en midterste observation,
er medianen den observation, der står til "højre
for midten".

- (2.3) Udvid PROC UDSKRIV2, så medianen udskrives.

- (2.4) Tilføj EXEC SORTER og EXEC MEDIAN til hovedprogrammet.
Tilføj også en ekstra EXEC UDSKRIV1, så både det usorterede
og det sorterede observationssæt udskrives inden de
mange mærketal.

- (2.5) Gem programmet på disketten.

(3). Statistisk analyse 3.

Størsteværdi, mindsteværdi og variationsbredde findes lettest,
når observationssættet er sorteret.

Lav en række ændringer af programmet fra(2), så ovenstående 3 mærketal
findes på letteste måde.
Det er muligt at hovedprogrammets EXEC-ordrer skal stå i en
ændret rækkefølge.

COMAL 8.

I denne øvelse behandles :

Nøgleord : IN, CHR\$, ORD, DIM(af en teksttabel)

Strukturer : CASE <variabel> OF
WHEN <liste>
 <programdel>
OTHERWISE
 <programdel>
ENDCASE

(1).Søgning 1.

- (1.1) Som bilag er der anbragt et program, der enten kan søge efter et telefonnummer ved indlæsning af et navn/dele af et navn, eller kan udskrive en telefonliste. Læs programmet igennem og find ud af, hvordan det virker.
- (1.2) Programmet ligger på disketten under navnet ENTLEFNR. Hent programmet ind i arbejdslageret, og prøv det.
- (1.3) Tilføj nogle flere datasætninger og prøv det igen.
- (1.4) Tilføj en procedure SØGNAVN, der ved indtastning af et telefonnummer eller dele deraf, udskriver den/de personer, hvorom oplysningen er korrekt. Husk at udvide PROC MENU.
- (1.5) Udvid PROC LISTE, så telefonlisten udskrives på printeren.
- (1.6) Gem programmet på disketten.

(2).Søgning 2.

- (2.1) Hent programmet fra (1) ind i arbejdslageret igen og tilføj følgende procedure med passende linienumre:

```
PROC SORTER
  FOR I=1 TO ANTAL-1 DO
    FOR J=I+1 TO ANTAL DO
      IF NAVN$(I)>NAVN$(J) THEN
        GEM1$=NAVN$(I);GEM2$=TLF$(I)
        NAVN$(I)=NAVN$(J);TLF$(I)=TLF$(J)
        NAVN$(J)=GEM1$;TLF$(J)=GEM2$
      ENDIF
    NEXT J
  NEXT I
ENDPROC SORTER
```

- (2.2) Udvid PROC LISTE med linien EXEC SORTER
Udvid dimensioneringerne med GEM1\$ OF 30 og
GEM2\$ of 10.
Afprøv programmet.

Programmet, der nu er fremstillet, er et generelt program til søgning 1, og sortering af teksttabeller. Det kan f. eks. også bruges til en vareliste, hvor NAVN\$ indeholder varens navn, og TLF\$ indeholder varens pris.

- (2.3) Udskift datasætningerne, så de indeholder varer og priser.
(EKS.: DATA "Vaskepulver", "29.95").
Afprøv programmet.
- (2.4) Ret i programmet, så der overalt kommer passende udskrifter, der passer til den nye situation.
Ret også navnene, så de bliver mere oplysende.
(EKS.: NAVN\$ til VARE\$ og PROC SØGNAVN til
PROC SØGVARE).
- (2.5) Gem programmet på disketten.

(3). Søgning 3.

I øvelsen ovenfor var priserne indeholdt i en teksttabel. Dette er ikke hensigtsmæssigt, da der er tale om tal.

Lav ændringer i programmet, så tekstvariable for priser ændres til talvariable, og teksttabellen for priserne ændres til en taltabel.

Lav en ændring af PROC SØGVARE, så der kan søges indenfor prisgrupper.

COMAL 9.

I denne øvelse behandles :

Nøgleord : INT
Strukturer : WHILE <udsagn> DO
 <programdel>
 ENDWHILE

(1). Butik 1.

- (1.1) Som bilag er der anbragt et program, der modtager bestillinger på varer, kontrollerer at varerne er på lager, registrerer købet og udskriver en regning. Programmet startes om morgenen og afsluttes om aftenen. Læs programmet igennem og find ud af, hvordan det virker.
- (1.2) Programmet ligger på disketten under navnet ENBUTIK. Hent det ind i arbejdslageret og prøv det.
- (1.3) Tilføj nogle flere datasætninger, og prøv det igen.
- (1.4) Udvid PROC REGNING, så regningen udskrives på printeren, og stil derefter over til skærmen igen.
- (1.5) Tilføj en procedure BETALING, der spørger efter det beløb, kunden lægger på bordet. Så længe det indbetalte beløb er for lille skal der udskrives en passende meddelelse, hvorefter der spørges igen. Udvid PROC SALG, så proceduren udføres.
- (1.6) Tilføj en procedure RETURBELØB, der udregner og udskriver det beløb, kunden skal have tilbage. Udvid PROC SALG, så proceduren udføres.
- (1.7) Gem programmet på disketten.

(2). butik 2.

Når dagen er slut, skal lageret tælles op, og der skal bestilles nye varer.

- (2.1) Hent dit program fra (1) ind i arbejdslageret, og tilføj følgende procedurer:

LAGER , der gennem søger lageret, og udskriver varenummer, varenavn, varepris og antallet af resterende enheder.

(Det er denne liste, der anvendes når varens nummer indtastes af ekspedienten ved køb. Herudover anvendes listen til indtastning af data næste morgen).

VÆRDI , der gennem søger lageret og udskriver lagerets værdi.

BESTIL , der gennem søger lageret, og udskriver varenavn, hvis antallet af enheder er mindre end 5.

- (2.2) Udvid hovedprogrammet så procedurerne udføres, når dagen er slut.

- (2.3) Udvid programmet, så lagerlisten, værdien og bestillingslisten udskrives på printeren

- (2.4) Gem programmet på disketten.

(3). BUTIK 3.

For at lette ekspedientens arbejde, skal maskinen udskrive, hvordan returbeløbet udbetales på nemmest måde.

Hent programmet fra (2) ind i arbejdslageret, og udvid PROC RETURBELØB, så der udskrives hvor mange stykker af de forskellige møntenheder, der skal udbetales.
(Denne procedure vil kunne tilsluttes en automat).

COMAL 10.

I denne øvelse behandles :

Nøgleord : OPEN FILE, CLOSE FILE, CLOSE, READ FILE
WRITE FILE, CHAIN, RENAME, DELETE

(1). Database 1.

En faglig forening er ved at oprette en database, indeholdende medlemmernes cpr. nummer, navn og løn.

(1.1) Som bilag er anbragt tre programmer :

ENOPRET , der opretter en medlemsfil med de
ønskede oplysninger.

ENUDSKRIV, der udskriver indholdet af filen.

ENSØG1 , der ved indtastning af navn/dele
af navn udskriver oplysninger om
det søgte medlem.

Læs programmerne igennem og find ud af, hvordan de
virker.

- (1.2) Programmerne ligger på disketten.
Hent ENOPRET ind i arbejdslageret, og opret en database
med 10 medlemmer.
- (1.3) Hent ENUDSKRIV ind i arbejdslageret, og kontroller de
indlæste data.
- (1.4) Hent ENSØG1 ind i arbejdslageret, og prøv at udføre nogle
søgninger.
- (1.5) Lav ændringer af ENSØG1, så der ved indtastning af
cpr.nummer eller dele heraf udskrives oplysninger om
medlemmet.
Gem programmet på disketten under navnet SØG2.
- (1.6) Lav ændringer af ENSØG1, så der ved indtastning af løn
udskrives oplysninger om det eller de medlemmer, der
oppebærer den pågældende løn.
Gem programmet på disketten under navnet SØG3.
- (1.7) Lav ændringer af ENSØG1, så der ved indtastning af
minimumsløn og maximumsløn udskrives oplysninger om de
medlemmer, der oppebærer en løn i det indtastede
løninterval.
Gem programmet på disketten under navnet SØG4.

(2). Database 2.

Når der kommer nye medlemmer, skal der kunne føjes nye oplysninger til databasen.

- (2.1) Som bilag er der anbragt et program, der forlænger filen. Læs programmet igennem og find ud af, hvordan det virker.
- (2.2) Programmet ligger på disketten under navnet ENTILFØJ. Hent det ind i arbejdslageret, og tilføj 5 medlemmer. Hent ENUDSKRIV ind i arbejdslageret, og kontroller, at de nye medlemmer er registrerede.
- (2.3) Lav et program, der kan slette et medlem, og prøv at slette et par medlemmer.
(Et medlem kan f. eks. slettes ved at oplysningerne ikke føres over i "bufferen").
Kontroller at medlemmerne ikke længere står i filen ved hjælp af programmet ENUDSKRIV.
Gem programmet på disketten under navnet SLET.
- (2.4) Lav et program, der kan ændre et medlems løn. Prøv at udføre et par ændringer og kontroller at ændringerne er foretaget i filen.
Gem programmet på disketten under navnet NYLØN.

(3). Database 3.

For at lette arbejdet med databasen, laves som regel et menuprogram, der spørger efter hvilket program, der skal anvendes, og derefter "chainer" til det ønskede program.

- (3.1) Konstruer et sådant menuprogram, hvorfra alle de 9 indtil nu konstruerede programmer kan kaldes.
- (3.2) Lav ændringer i alle de konstruerede programmer, så der automatisk "chaines" tilbage til menuprogrammet.

```

0010 // BILAG TIL COMAL 3
0020 // ENSKAT
0030 DIM navn$ OF 20
0040 //
0050 EXEC indlæs
0060 EXEC statsskat
0070 EXEC udskriv
0080 //
0090 //
0100 PROC indlæs
0110   PRINT CHR$(12)
0120   PRINT AT(1,10);
0130   INPUT "Hvad er dit navn      : ": navn$
0140   PRINT
0150   INPUT "Skattepligtig indkomst : ": indk
0160 ENDPROC indlæs
0170 //
0180 PROC statsskat
0190   IF indk<=0 THEN
0200     skat:= 0
0210   ELSE
0220     IF indk<=108200 THEN
0230       skat:= indk*19.75/100
0240     ELSE
0250       IF indk<179000 THEN
0260         skat:= 21369.5+(indk-108200)*34.15/100
0270       ELSE
0280         skat:= 45547.7+(indk-179000)*44.95/100
0290       ENDIF
0300     ENDIF
0310   ENDIF
0320 ENDPROC statsskat
0330 //
0340 PROC udskriv
0350   PRINT CHR$(12)
0360   PRINT AT(1,10);
0370   PRINT "Statsskat for : ",navn$
0380   PRINT
0390   PRINT USING "Skattepligtig indkomst : ##### kr": indk
0400   PRINT
0410   PRINT USING "Statsskat                : ##### kr": skat
0420   PRINT TAB(27);"*****"
0430 ENDPROC udskriv

```

```

0010 // BILAG TIL COMAL 4
0020 // ENPLUS
0030 RANDOMIZE
0040 //
0050 REPEAT
0060     EXEC valg
0070     EXEC facit
0080     EXEC opgave
0090     EXEC pause(10)
0100 UNTIL FALSE
0110 //
0120 //
0130 PROC valg
0140     tal1:= RND(1,10)
0150     tal2:= RND(1,10)
0160 ENDPROC valg
0170 //
0180 PROC facit
0190     fac:= tal1+tal2
0200 ENDPROC facit
0210 //
0220 PROC opgave
0230     PRINT CHR$(12)
0240     REPEAT
0250         PRINT AT(1,10);
0260         PRINT USING "SS + SS = ": tal1,tal2
0270         PRINT AT(11,10);
0280         PRINT " "
0290         PRINT AT(11,10);
0300         INPUT "": svar
0310         EXEC kontrol
0320     UNTIL svar=fac
0330 ENDPROC opgave
0340 //
0350 PROC kontrol
0360     PRINT AT(30,23);
0370     IF svar=fac THEN
0380         PRINT "R I G T I G T"
0390     ELSE
0400         PRINT "F O R K E R T"
0410     ENDIF
0420 ENDPROC kontrol
0430 //
0440 PROC pause(n)
0450     i:= 0
0460     REPEAT
0470         i:= i+1
0480     UNTIL i>n*25
0490 ENDPROC pause

```

```

0010 // BILAG TIL COMAL 5
0020 // ENAFBETA
0030 //
0040 EXEC indlæs
0050 EXEC ydelse
0060 EXEC overskrift
0070 EXEC beregn
0080 //
0090 //
0100 PROC indlæs
0110   PRINT CHR$(12)
0120   PRINT AT(1,10);
0130   INPUT "Gældens størrelse           : ": gald
0140   INPUT "Rentens størrelse i %      : ": r
0150   INPUT "Antal terminer              : ": n
0160 ENDPROC indlæs
0170 //
0180 PROC ydelse
0190   ydel:= gald*r/100/(1-1/(1+r/100)^n)
0200 ENDPROC ydelse
0210 //
0220 PROC overskrift
0230   PRINT CHR$(12)
0240   PRINT USING "Ydelsen er på 55555.55 kr": ydel
0250   PRINT
0260   PRINT "      Gald           Rente           Afdrag           Ny restgald"
0270 ENDPROC overskrift
0280 //
0290 PROC beregn
0300   nygald:= gald
0310   REPEAT
0320     gald:= nygald
0330     rente:= gald*r/100
0340     afdrag:= ydel-rente
0350     nygald:= gald-afdrag
0360     EXEC udskriv
0370   UNTIL nygald<1
0380 ENDPROC beregn
0390 //
0400 PROC udskriv
0410   PRINT USING "555555.55      55555.55      ": gald,rente;
0420   PRINT USING "55555.55      555555.55": afdrag,nygald
0430 ENDPROC udskriv

```

```

0010 // BILAG TIL COMAL 6
0020 // ENMØNT
0030 DIM svar$ OF 3
0040 RANDOMIZE
0050 //
0060 EXEC indlæs
0070 EXEC udfør
0080 EXEC udskriv
0090 //
0100 //
0110 PROC indlæs
0120   PRINT CHR$(12)
0130   PRINT AT(1,10)
0140   INPUT "Indtast antal møntkast : ": n
0150 ENDPROC indlæs
0160 //
0170 PROC udfør
0180   p:= 0
0190   k:= 0
0200   FOR i:= 1 TO n DO
0210     kast:= RND(1,2)
0220     IF kast=1 THEN
0230       p:= p+1
0240     ELSE
0250       k:= k+1
0260     ENDIF
0270   NEXT i
0280 ENDPROC udfør
0290 //
0300 PROC udskriv
0310   PRINT CHR$(12)
0320   PRINT AT(1,10);
0330   PRINT USING "Der blev udført ##### møntkast ": n
0340   PRINT
0350   PRINT USING "Det blev plat ##### gange ": p
0360   PRINT USING "Det blev krone ##### gange ": k
0370 ENDPROC udskriv

```

```

0010 // BILAG TIL COMAL 7
0020 // ENSTAT
0030 //
0040 EXEC størrelse
0050 EXEC indlæs
0060 EXEC udskriv1
0070 EXEC summering
0080 EXEC gennemsnit
0090 EXEC størstevardi
0100 EXEC udskriv2
0110 //
0120 //
0130 PROC størrelse
0140     antal:= 0
0150     REPEAT
0160         READ x
0170         antal:= antal+1
0180     UNTIL EOD
0190 ENDPROC størrelse
0200 //
0210 PROC indlæs
0220     DIM a(antal)
0230     RESTORE
0240     FOR i:= 1 TO antal DO
0250         READ a(i)
0260     NEXT i
0270 ENDPROC indlæs
0280 //
0290 PROC summering
0300     sum:= 0
0310     FOR i:= 1 TO antal DO
0320         sum:= sum+a(i)
0330     NEXT i
0340 ENDPROC summering
0350 //
0360 PROC gennemsnit
0370     gsnit:= sum/antal
0380 ENDPROC gennemsnit
0390 //
0400 PROC størstevardi
0410     max:= a(1)
0420     FOR i:= 2 TO antal DO
0430         IF max<a(i) THEN
0440             max:= a(i)
0450         ENDIF
0460     NEXT i
0470 ENDPROC størstevardi
0480 //
0490 PROC udskriv1
0500     PRINT CHR$(12)
0510     PRINT AT(1,10);
0520     FOR i:= 1 TO antal DO
0530         PRINT a(i);
0540     NEXT i
0550     PRINT
0560     PRINT
0570 ENDPROC udskriv1
0580 //
0590 PROC udskriv2
0600     PRINT USING "Antal"           :5555": antal
0610     PRINT USING "Summen"         :5555": sum
0620     PRINT USING "Gennemsnit"    : 555.5": gsnit
0630     PRINT USING "Størstevardi"  : 555": max
0640 ENDPROC udskriv2
0650 //
0660 DATA 12,34,25,67,39,89,23,54,76,67
0670 DATA 45,17,39,70,23,78,65,47,73,98

```



```

0010 // BILAG TIL COMAL 8
0020 // ENTLFNR
0030 MARGIN 80
0040 DIM n$ OF 30,t$ OF 10,svar$ OF 30
0050 //
0060 EXEC størrelse
0070 EXEC indlæs
0080 EXEC menu
0090 //
0100 PROC størrelse
0110     antal:= 0
0120     REPEAT
0130         READ n$,t$
0140         antal:= antal+1
0150     UNTIL EOD
0160 ENDPROC størrelse
0170 //
0180 PROC indlæs
0190     DIM navn$(antal) OF 30,tlf$(antal) OF 10
0200     RESTORE
0210     FOR i:= 1 TO antal DO
0220         READ navn$(i),tlf$(i)
0230     NEXT i
0240 ENDPROC indlæs
0250 //
0260 PROC menu
0270     REPEAT
0280         PRINT CHR$(12)
0290         PRINT AT(1,10);
0300         PRINT "1. Søg telefonnummer."
0310         PRINT
0320         PRINT "2. Udskriv telefonliste."
0330         PRINT
0340         INPUT "Indtast det ønskede nr: ": valg
0350     UNTIL valg=1 OR valg=2
0360     CASE valg OF
0370         WHEN 1
0380             EXEC søg
0390         OTHERWISE
0400             EXEC liste
0410     ENDCASE
0420 ENDPROC menu
0430 //
0440 PROC søg
0450     PRINT CHR$(12)
0460     PRINT AT(1,10);
0470     INPUT "Indtast navn : ": svar$
0480     PRINT
0490     FOR i:= 1 TO antal DO
0500         IF svar$ IN navn$(i) THEN
0510             PRINT navn$(i);TAB(30);" : ";tlf$(i)
0520         ENDIF
0530     NEXT i
0540 ENDPROC søg
0550 //
0560 PROC liste
0570     PRINT CHR$(12)
0580     PRINT "TELEFONLISTE : "
0590     PRINT
0600     FOR i:= 1 TO antal DO
0610         PRINT navn$(i);TAB(30);" : ";tlf$(i)
0620     NEXT i
0630 ENDPROC liste
0640 //
0650 DATA "NORUP ERIK","(02)736460"
0660 DATA "NIELSEN ERIK","(04)725070"

```

```

0010 // BILAG TIL COMAL 9
0020 // ENBUTIK
0030 MARGIN 80
0040 DIM v$ OF 15, svar$ OF 1
0050 //
0060 EXEC størrelse
0070 EXEC indlæs
0080 REPEAT
0090     EXEC salg
0100 UNTIL svar$="J" OR svar$="j"
0110 PRINT CHR$(12)
0120 //
0130 //
0140 PROC størrelse
0150     antal:= 0
0160     REPEAT
0170         READ v$,a,b
0180         antal:= antal+1
0190     UNTIL EOD
0200 ENDPROC størrelse
0210 //
0220 PROC indlæs
0230     DIM varer$(antal) OF 15, pris(antal), styk(antal), salg(antal)
0240     RESTORE
0250     FOR i:= 1 TO antal DO
0260         READ varer$(i), pris(i), styk(i)
0270     NEXT i
0280 ENDPROC indlæs
0290 //
0300 PROC salg
0310     EXEC nulsalg
0320     REPEAT
0330         EXEC varekontrol
0340         EXEC stykkontrol
0350         EXEC køb
0360     UNTIL svar$="J" OR svar$="j" // KØBET SLUT
0370     EXEC regning
0380     PRINT AT(15,23);
0390     INPUT " ER DAGEN SLUT (J/N) ": svar$
0400 ENDPROC salg
0410 //
0420 PROC nulsalg
0430     FOR i:= 1 TO antal DO
0440         salg(i):= 0
0450     NEXT i
0460 ENDPROC nulsalg
0470 //
0480 PROC varekontrol
0490     PRINT CHR$(12)
0500     PRINT AT(1,10);
0510     INPUT "VARENR.   : ": nr
0520     WHILE nr>antal DO
0530         PRINT AT(15,8);
0540         PRINT "VARENUMMER ER FOR STORT"
0550         PRINT AT(1,10);
0560         PRINT "
0570         PRINT AT(1,10);
0580         INPUT "VARENR.   : ": nr
0590     ENDWHILE
0600 ENDPROC varekontrol

```

```

0620 PROC stykkontrol
0630   PRINT CHR$(12)
0640   PRINT AT(1,10);
0650   PRINT varer$(nr)
0660   INPUT "HVOR MANGE ? ": stk
0670   WHILE stk>stykk(nr) DO
0680     PRINT AT(15,8);
0690     PRINT "VI HAR KUN ";stykk(nr);" PA LAGER "
0700     PRINT AT(1,11);
0710     PRINT "
0720     PRINT AT(1,11);
0730     INPUT "HVOR MANGE ? ": stk
0740   ENDWHILE
0750 ENDPROC stykkontrol
0760 //
0770 PROC køb
0780   salg(nr):= salg(nr)+stk
0790   styk(nr):= styk(nr)-stk
0800   PRINT AT(15,23);
0810   INPUT "ER KØBET SLUT (J/N) ": svar$
0820 ENDPROC køb
0830 //
0840 PROC regning
0850   PRINT CHR$(12)
0860   sum:= 0
0870   PRINT AT(1,4);
0880   FOR i:= 1 TO antal DO
0890     IF salg(i)<>0 THEN
0900       spris:= pris(i)*salg(i)
0910       sum:= sum+spris
0920       PRINT varer$(i);
0930       PRINT TAB(15);" : ";
0940       PRINT USING "$$$ STK AF $$$.$$ KR :$$$$$. $$ KR": salg(i),pris(i),spr
is
0950     ENDIF
0960   NEXT i
0970   PRINT "
0980   PRINT USING "SAMLET PRIS
0990   PRINT
1000   asum:= INT(sum*20+0.5)/20
1010   PRINT USING "AFRUNDET PRIS
1020   PRINT "
1030 ENDPROC regning
1040 //
1050 DATA "HOF",4.19,200
1060 DATA "TUBORG",4.21,300
1070 DATA "DANSKVAND",1.97,38
1080 DATA "GL. DANSK",102.98,100

```

```

-----"
:$$$$$. $$ KR ": sum
:$$$$$. $$ KR ": asum
=====

```

```

0010 // BILAG TIL COMAL 10
0020 // ENOPRET
0030 DIM cpr$ OF 11,navn$ OF 20,svar$ OF 1
0040 //
0050 EXEC filåbner
0060 EXEC indlæsning
0070 EXEC fillukker
0080 //
0090 //
0100 PROC filåbner
0110   OPEN FILE 1,"REGISTER", WRITE
0120   OPEN FILE 3,"FILLGD", WRITE
0130 ENDPROC filåbner
0140 //
0150 PROC indlæsning
0160   antal:= 0
0170   REPEAT
0180     PRINT CHR$(12)
0190     INPUT "ANGIV CPR. NUMMER : ": cpr$
0200     INPUT "ANGIV NAVN      : ": navn$
0210     INPUT "ANGIV MANEDSLØN  : ": løn
0220     antal:= antal+1
0230     WRITE FILE 1: cpr$,navn$,løn
0240     PRINT
0250     INPUT "FLERE INDLÆSNINGER ? (J/N) ": svar$
0260     UNTIL svar$="n" OR svar$="N"
0270     WRITE FILE 3: antal
0280     PRINT CHR$(12)
0290 ENDPROC indlæsning
0300 //
0310 PROC fillukker
0320   CLOSE
0330 ENDPROC fillukker

```

```

0010 // BILAG TIL COMAL 10
0020 // ENUDSKRI
0030 DIM cpr$ OF 11,navn$ OF 20
0040 MARGIN 80
0050 //
0060 EXEC filåbner
0070 EXEC udlæsning
0080 EXEC fillukker
0090 //
0100 //
0110 PROC filåbner
0120   OPEN FILE 1,"REGISTER", READ
0130   OPEN FILE 3,"FILLGD", READ
0140 ENDPROC filåbner
0150 //
0160 PROC udlæsning
0170   READ FILE 3: antal
0180   PRINT CHR$(12)
0190   FOR i:= 1 TO antal DO
0200     READ FILE 1: cpr$,navn$,løn
0210     PRINT cpr$;" ";navn$;TAB(35);
0220     PRINT USING "SSSSS KR": løn
0230   NEXT i
0240 ENDPROC udlæsning
0250 //
0260 PROC fillukker
0270   CLOSE
0280 ENDPROC fillukker

```

```

0010 // BILAG TIL COMAL 10
0020 // ENSØGI
0030 MARGIN 80
0040 DIM cpr$ OF 11,navn$ OF 20,søg$ OF 20
0050 //
0060 EXEC filåbner
0070 EXEC søge
0080 EXEC fillukker
0090 //
0100 //
0110 PROC filåbner
0120   OPEN FILE 1,"REGISTER", READ
0130   OPEN FILE 3,"FILLGD", READ
0140 ENDPROC filåbner
0150 //
0160 PROC søge
0170   PRINT CHR$(12)
0180   PRINT AT(10,10);
0190   INPUT "ANGIV NAVN : ": søg$
0200   PRINT CHR$(12)
0210   READ FILE 3: antal
0220   FOR i:= 1 TO antal DO
0230     READ FILE 1: cpr$,navn$,løn
0240     IF søg$ IN navn$ THEN
0250       PRINT cpr$;" ";navn$;TAB(35);
0260       PRINT USING "SSSS$ KR": løn
0270     ENDIF
0280   NEXT i
0290   PRINT
0300   PRINT "KARTOTEKET ER GENNEMSØGT"
0310 ENDPROC søge
0320 //
0330 PROC fillukker
0340   CLOSE
0350 ENDPROC fillukker

```

```

0010 // BILAG TIL COMAL 10
0020 // ENTILFØJ
0030 DIM cpr$ OF 11,navn$ OF 20,svar$ OF 1
0040 //
0050 EXEC filåbner
0060 EXEC kopier
0070 EXEC forlæng
0080 EXEC fillukker
0090 EXEC omdøb
0100 //
0110 //
0120 PROC filåbner
0130   OPEN FILE 1,"REGISTER", READ
0140   OPEN FILE 2,"BUFFER", WRITE
0150   OPEN FILE 3,"FILLGD", READ
0160   OPEN FILE 4,"FILBUF", WRITE
0170 ENDPROC filåbner
0180 //
0190 PROC kopier
0200   READ FILE 3: antal
0210   FOR i:= 1 TO antal DO
0220     READ FILE 1: cpr$,navn$,løn
0230     WRITE FILE 2: cpr$,navn$,løn
0240   NEXT i
0250 ENDPROC kopier
0260 //
0270 PROC forlæng
0280   REPEAT
0290     PRINT CHR$(12)
0300     PRINT AT(1,7);
0310     INPUT "ANGIV CPR. NUMMER : ": cpr$
0320     INPUT "ANGIV NAVN      : ": navn$
0330     INPUT "ANGIV MANEDSLØN : ": løn
0340     WRITE FILE 2: cpr$,navn$,løn
0350     antal:= antal+1
0360     PRINT
0370     INPUT "FLERE INDLÆSNINGER ? (J/N) ": svar$
0380     UNTIL svar$="n" OR svar$="N"
0390     WRITE FILE 4: antal
0400 ENDPROC forlæng
0410 //
0420 PROC fillukker
0430   CLOSE
0440 ENDPROC fillukker
0450 //
0460 PROC omdøb
0470   DELETE "REGISTER"
0480   RENAME "BUFFER","REGISTER"
0490   DELETE "FILLGD"
0500   RENAME "FILBUF","FILLGD"
0510 ENDPROC omdøb

```