

4

3. ÅRGANG

SEPTEMBER 1979

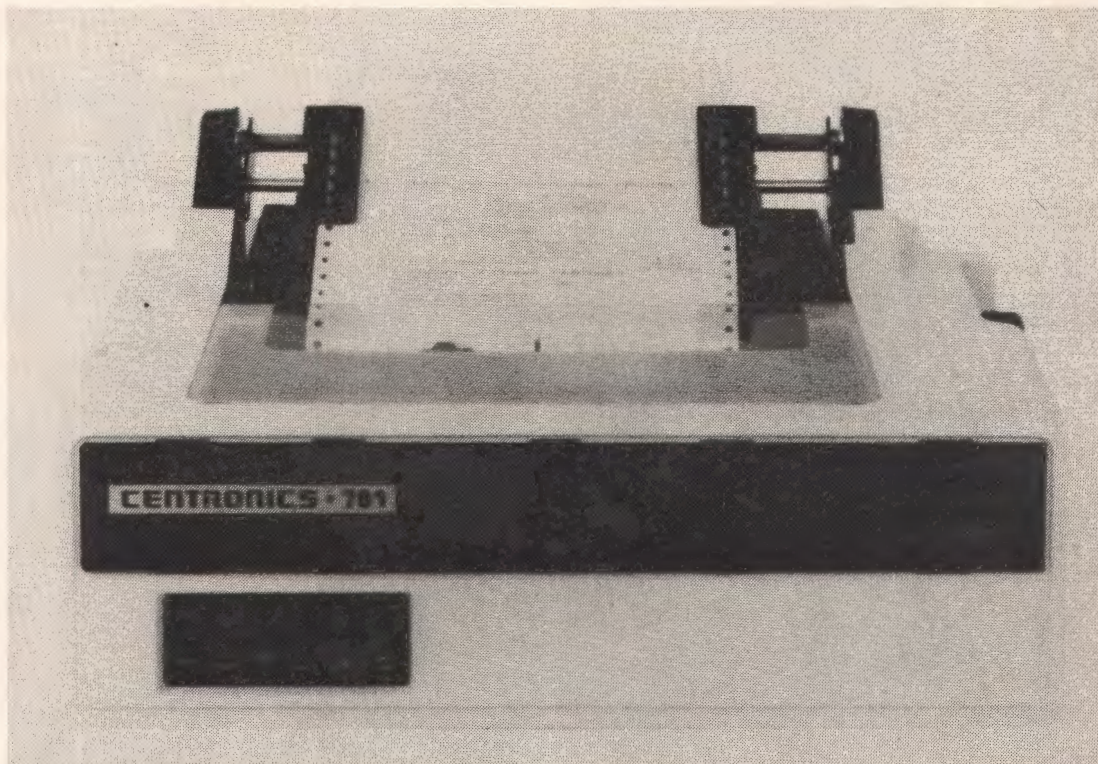
data lære

INDHOLD

En eksperimentel sammenligning af to datamaskiner
Anvendelse af lineskriver til tegning af billeder
Quicksort
Sortering som Call-routine
Besøg hos SAS

Udgivet af

FORENINGEN FOR DATALÆRE OG ANVENDELSE AF EDB I UNDERVISNINGEN



CENTRONICS

781

60 TEGN PR. SEKUND

BI-DIRECTIONAL PRINT

LOGISK SØGNING

80 SKRIVEPOSITIONER

5 x 7 MATRIX

DOBBELT TRACTOR FEED

CENTRONICS PROGRAMMET OMFATTER MATRIX- OG
KÆDEPRINTERE FRA 30 TEGN PR. SEKUND TIL 600
LINIER PR. MINUT.

INTERMEDIUM EDB

TEKNIK OG SERVICE APS
SYDVESTVEJ 53
2600 GLOSTRUP
TLF.: (02) 45 82 33

Jeg vil gerne vide mere
om CENTRONICS printere
og Interface muligheder

navn _____
adresse _____
postnr./by _____



En eksperimentel sammenligning af to datamaskiner

af H. B. Hansen

Indledning

I foråret 1979 havde vi i datalogiafdelingen på Roskilde Universitetscenter lejlighed til at lege med en af de nymodens mikrodatamater, nemlig en SPC/1. Denne anledning benyttede jeg til at forsøge at skabe mig et indtryk af hvad disse små, prisbillige datamaskiner egentlig kan præstere. Jeg lavede nogle småprogrammer ved hjælp af hvilke jeg kunne måle forskellige ting, og sammenligne med vores RC 7000 minidatamat. Jeg gik med vilje ikke ud fra fabrikanternes tekniske specifikationer for deres maskiner, men benyttede mig af at begge maskiner har en Comal-fortolker.

Det følgende er en rapport over forsøget og måleresultaterne. På grund af den ret korte tid vi havde mikrodatamaten er undersøgelsen på ingen måde fuldstændig, men måske kan den alligevel have en vis interesse også for andre.

Tidsmålinger

Det første jeg målte var tiden for at udføre forskellige beregninger. Her stødte jeg på den vanskelighed at mikrodatamaten tilsyneladende ikke havde noget programmeringsmæssigt tilgængeligt ur. Jeg besluttede derfor at måle tider på begge maskiner ved hjælp af et stopur. En anden vanskelighed var at mikrodatamaten var en enkelt-bruger-maskine, mens minidatamaten var forsynet med fem multiprogrammerede terminaler. Tiderne på minidatamaten blev derfor målt udenfor normal arbejdstid, hvor jeg var eneste bruger af systemet.

Princippet i tidsmålingerne bygger på at måle tiden af følgende algoritme:

```
100 INPUT N
110 FOR I=1 TO N
120   REM HER INDSÆTTES DET DER
130 NEXT I           SKAL MÅLES
140 END
```

Først slettes linie 120 helt, hvorved tiden for en tom løkke kan måles. Det er vigtigt at denne løkketid er ret nøjagtigt bestemt, og derfor målte jeg den for forskellige værdier af N. I linie 100 beder maskinen om et N, og når man har tastet det startes stopuret idet man trykker på RETURN; uret stoppes igen når maskinen trykker sin slutmeddelelse i linie 140. Målepunkterne afsatte jeg på millimeterpapir som vist på fig. 1. Hældninger af

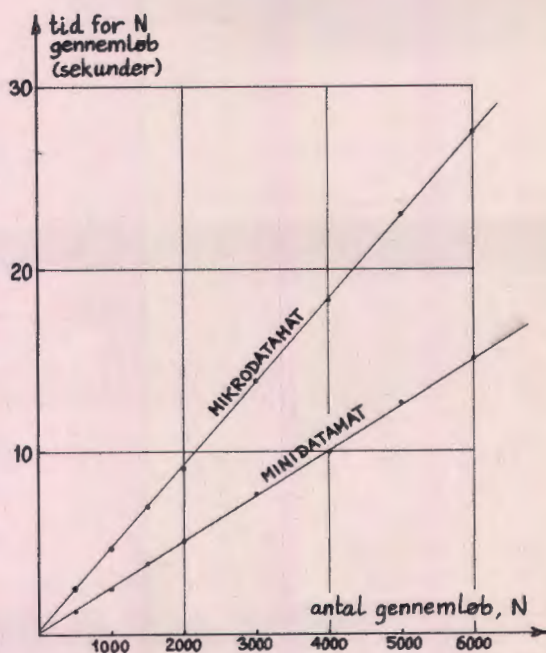


Fig. 1. Resultat af tidsmålinger for en tom løkke. De to hældninger af de rette linier er henholdsvis 4.55 msec/gennemløb for mikrodatamaten, og 2.48 msec/gennemløb for minidatamaten. Minidatamaten udfører således en løkke ca. 1.8 gange hurtigere end mikrodatamaten.

den rette linie man får frem er tiden pr. gennemløb af den tomme løkke.

Nu kan man sætte de sætninger man ønsker at måle tid på, ind i løkken i linie 120, måle tiden for et passende N, samt korrigere for løkketiden. Jeg brugte N=3000 på mikrodatamaten og N=4000 på minidatamaten. I nogle tilfælde må man initialisere nogle størrelser inden løkken (det skal så ske inden linie 100), og i andre må der anbringes nogle sætninger mellem linie 130 og 140 (f. eks. en tom procedure), men disse detaljer fremgår af fig. 2, der er en tabel over de bearbejdede måleresultater.

Til nærmere forklaring af tabellens tal gennemgås nu beregningerne for et enkelt eksempel, nemlig tredje række i tabellen, en addition af to variable.

For mikrodatamaten brugtes N=3000, og tiden for programmet målt tre gange. Middeltallet af målingerne var 23.3 sekunder. Af fig. 1 aflæses

Modifikationer af tidtagningsprogrammet	N gennemløb (sekunder)		Tid pr. gennemløb (millisekunder)		Mini:Mikro
	Mikro	Mini	Mikro	Mini	
120 LET X=3.14159	18.0	19.3	1.4	2.3	1.64
120 LET X=3.14159+4.6452	22.5	22.9	2.9	3.2	1.10
50 LET X=3.14159; Y=4.6452 120 LET Z=X+Y	23.3	26.1	3.1	4.0	1.29
50 LET X=3.14159; Y=4.6452 120 LET Z=X*Y	23.3	27.1	3.1	4.3	1.39
120 GOTO 130	14.5	14.5	0.2	1.1	5.50
120 EXEC P 135 PROC P 137 ENDPROC	17.1	18.5	1.1	2.1	1.91
120 GOSUB 135 135 RETURN	16.2	19.1	0.8	2.3	2.88
50 LET X=3.14159 120 LET Z=LN(X) for mikro 120 LET Z=LOG(X) for mini	27.7	45.0	4.6	8.8	1.91
50 LET X=3.14159 120 LET Z=EXP(X)	27.7	43.2	4.6	8.3	1.80
50 LET X=0.23456 120 LET Z=COS(X)	26.5	40.3	4.2	7.6	1.81
50 LET X=0.23456 120 LET Z=ATAN(X) for mikro 120 LET Z=ATN(X) for mini	28.1	43.8	4.7	8.5	1.81
50 LET X=0.23456 120 LET Z=TAN(X)	28.1	47.1	4.7	9.3	1.98
50 DIM A(10) 55 LET X=5 120 LET Z=A(X)	24.1	27.3	3.4	4.3	1.26
50 DIM A(10,10) 55 LET X=5 120 LET Z=A(X,X)	25.9	16.7	21.2	5.8	0.27

Fig. 2. Tidsmålinger for forskellige beregninger. Der blev benyttet $N=3000$ for mikrodatamaten, og $N=4000$ for minidatamaten, med undtagelse af den sidste måling, indiceret variabel med to indices, hvor $N=1000$ for mikrodatamaten, og $N=2000$ for minidatamaten.

at løkketiden for N=3000 er 13.9 sekunder, således at tiden pr. operation bliver:

$$\frac{23.3 - 13.9}{3000} \text{ sek} = 3.1 \text{ msek}$$

For minidatamaten var N=4000, og programtiden (gennemsnit af tre målinger) målt til 26.1 sekunder. Af fig. 1 aflæses løkketiden til 10.0 sekunder, så her finder vi:

$$\frac{26.1 - 10.0}{4000} \text{ sek} = 4.0 \text{ msek}$$

Det er interessant at se, at skønt mikrodatamaten er ca. 1.8 gange langsommere end minidatamaten hvad angår FOR/NEXT, så viser det sig at mikrodatamaten er hurtigere end minidatamaten til de fleste beregninger med LET-sætninger. En undtagelse er beregninger med indicerede variable med to indices, hvor mikrodatamaten er meget ineffektiv. Det er overraskende at mikrodatamaten udfører GOTO-sætninger ca. 5.5 gange hurtigere end minidatamaten. Det skyldes sikkert at Comalfortolkeren i mikrodatamaten er optimaliseret på dette punkt, f. eks. ved at lageradressen i hver GOTO-sætning huskes efter første gennemløb af sætningen, i stedet for at søge i programteksten efter den rette linie hver gang. Mikrodatamatens overlegenhed ved beregninger af standardfunktioner skyldes sikkert at hele standardfunktionsbiblioteket befinder sig på en enkelt integreret kreds, mens minidatamatens funktionsberegninger er baseret på maskinprogrammer i lageret. Hvis det er sådan betyder det også, at standardprogrammerne ikke optager plads i mikrodatamatens lager.

Jeg kender som sagt ikke de tekniske specifikationer for de to maskiner, f. eks. deres cyklustid, men det der må interessere den almindelige bruger er vel også mest effektiviteten af de kørende programmer. Afvigelse mellem de to maskiner skyldes utvivlsomt for en stor dels vedkommende, at der er tale om to helt forskellige Comalfortolkere. F. eks. vil jeg tro at når mikrodatamaten er lidt langsommere til EXEC end til GOSUB, mens det er omvendt for minidatamaten, så skyldes det at man kan have formelle parametre til procedurer i mikrodatamaten, men ikke i minidatamaten.

Nu kan man gå videre og måle andre udførelsestider. F. eks. målte jeg tiden for REPEAT/UNTIL og WHILE/ENDWHILE på følgende måde:

Først indførtes i løkkeprogrammet følgende ændringer:

```
50 LET X=0
120 LET X=X+1
```

Tiden for LET X=X+1 målt herefter som lige beskrevet. Jeg fandt:

Mikrodatamat: N=3000, T= 22.7 - 13.9= 8.8
 Minidatamat: N=4000, T= 24.8 - 10.0= 14.8

Nu målt udførelsestiden for følgende programmer:

```
50 LET X=0
100 INPUT N
110 REPEAT
120 LET X=X+1
130 UNTIL X=N
140 END
```

og:

```
50 LET X=1
100 INPUT N
110 WHILE X<N DO
120 LET X=X+1
130 ENDWHILE
140 END
```

Jeg brugte som sædvanlig N=3000 på mikrodatamaten, og N=4000 på minidatamaten. Resultatet af tidsmålingerne blev:

	Mikro	Mini
REPEAT/UNTIL	18.1	28.7
WHILE/ENDWHILE	18.1	28.9

Korrigeres med tiden for linie 120 i programmerne, og divideres med N fås:

	Mikro	Mini
REPEAT/UNTIL	3.1	3.5
WHILE/ENDWHILE	3.1	3.5

hvor måleenheden er millisekunder pr. gennemløb. Her er de to maskiner altså praktisk taget lige hurtige.

Nøjagtighedsmålinger

Det viste sig altså at hvad angår beregningshastighed kunne mikrodatamaten nok klare sig i sammenligning med minidatamaten. Dette kunne skyldes at mikrodatamaten regner mere unøjagtigt end minidatamaten. For at undersøge disse forhold gennemførte jeg nogle kørsler der viser noget om maskinernes beregningsfejl.

En interessant størrelse i denne forbindelse er maskinernes præcision, dvs. antallet af cifre i deres talrepræsentation. Atter her afstod jeg fra at slå op i håndbøger. I stedet benyttede jeg at præcisionen må kunne udtrykkes som det største tal der netop er så lille at det ikke har nogen virkning, når det lægges til tallet 1. Jeg kørte derfor følgende program på de to maskiner:

```

100 LET P=1
110 REPEAT
120 LET P=P/2
130 UNTIL 1+P=1
140 PRINT P
150 END

```

Resultatet blev 5.96046E-08 for mikrodata-
maten, og 2.38419E-07 for minidatamaten. Det
ser således ud til at mikrodatamaten arbejder
med 2 bit mere i tallene end minidatamaten
(forholdet mellem de to resultater er 4, dvs. 2²).

Herefter fortsatte jeg med at måle fejlene på
systemernes standardfunktioner. Jeg brugte det
princip at beregne de to sider af en kendt mate-
matisk identitet, som jeg derefter fandt afvigelsen
imellem. F. eks. ved man jo at:

$$\cos(2x) = 2\cos^2(x) - 1.$$

Ved at beregne størrelsen $\cos(2x) - 2\cos^2(x) + 1$
for en masse forskellige værdier af x , og holde
rede på minimum, maksimum og gennemsnit,
får man et billede af maskinens nøjagtighed ved
beregning af cosinusfunktionen.

Som illustration af metoden ses på fig. 3 det
program der blev benyttet ved ovennævnte be-
regning. Med dette program som skabelon blev
der gennemført de målinger der fremgår af fig. 4.

Før at afprøve logaritmefunktionen blev pro-
grammet udvidet til at bestå af to løkker inden i
hinanden, hvor den ydre gennemløb intervallet
 $1 \leq I \leq N$, og den indre gennemløb intervallet
 $I \leq J \leq N$. I den ydre løkke beregnedes et X i
intervallet $XMIN \leq X \leq XMAX$ som vist på fig. 3,
linie 130, og i den indre løkke blev et Y i inter-
vallet $XMIN \leq Y \leq XMAX$ beregnet efter et lig-

```

100 LET PI=3.14159265; MIN=1; MAX=-1; SUM=0
110 LET N=100; XMIN=0; XMAX=2*PI
120 FOR I=1 TO N
130 LET Q=(I-1)/(N-1); X=Q*XMAX + (1-Q)*XMIN
140 LET FEJL=COS(2*X) - 2*COS(X)*COS(X) + 1
150 LET SUM=SUM + ABS(FEJL)
160 IF FEJL<MIN THEN LET MIN=FEJL
170 IF FEJL>MAX THEN LET MAX=FEJL
180 NEXT I
190 PRINT MIN,MAX,SUM/N
200 END

```

Fig. 3. Program til nøjagtighedsmåling. Program-
met beregner FEJL for N ækvidistante punkter
mellem $XMIN$ og $XMAX$, og trykker minimums-
værdi, maksimumsværdi og gennemsnit af de
beregnete FEJL-værdier. Andre funktioner kan
måles ved at ændre linie 110 og 140 passende.

nende princip. Identiteten i linie 140 blev ændret
til:

```
140 LET FEJL=LN(X) + LN(Y) - LN(X*Y)
```

og ved gennemsnitsberegningen blev der taget
hensyn til at antallet af (X,Y) -par er $N(N+1)/2$.

Med det samme program blev multiplikation
afprøvet, idet linie 140 ændredes til:

```
140 LET FEJL=X*X - Y*Y - (X+Y)*(X-Y)
```

Resultatet fremgår af fig. 5. Det ser jo helt kata-
strofalt ud for multiplikation - kan maskinerne
ikke multiplicere?

Det jeg har fundet ved disse beregninger er
den absolutte fejl, men teorien for flydende tal
siger at det er den relative fejl der er nogenlunde
konstant. Derfor modificerede jeg multiplika-
tionsprogrammet ved at dividere FEJL med
 $(X+Y) - (X-Y)$, idet jeg samtidig sørgede for at
undgå division med nul.

Identitet	N	XMIN	XMAX	MIN	MAX	SUM/N
$x = 1/(1/x)$	100	1	1000	-1.22070E-04 -6.10352E-05	6.10351E-05 9.15527E-05	2.98405E-05 6.25610E-06
$x = \sqrt{x} * \sqrt{x}$	100	1	1000	-6.10351E-05 -7.32422E-04	1.83105E-04 2.44141E-04	5.11074E-05 1.65262E-04
$e^x = 1/e^{-x}$	100	0	10	-9.76562E-04 -7.81250E-03	1.95312E-03 3.90625E-03	1.65139E-04 5.48240E-04
$\cos(2x) = 2\cos^2(x) - 1$	100	0	2π	-1.31130E-06 -9.53670E-07	1.43051E-06 1.37091E-06	3.58820E-07 3.49879E-07
$\text{tg}(2x) = 2\text{tg}(x)/(1-\text{tg}^2(x))$	100	0	π/2.01	-6.59890E-04 -1.08004E-04	9.15527E-05 1.22161E-02	1.41194E-05 1.30665E-04

Fig. 4. Resultat af kørsler med varianter af programmet fra fig. 3. De identiteter der er anvendt frem-
går af tabellens første kolonne. For hver identitet er resultatet for mikrodatamaten anført i første linie,
og resultatet for minidatamaten i anden linie.

Identitet	N	XMIN	XMAX	MIN	MAX	SUM/N/(N+1)*2
$\log(xy) = \log(x) + \log(y)$	15	1	1000	-1.90735E-06 -9.53674E-06	9.53674E-07 7.62939E-06	3.97876E-07 2.91864E-06
$(x+y)(x-y) = x^2 - y^2$	15	1	1000	-1.25000E-01 -2.73438E-02	6.25000E-02 6.25000E-02	2.73926E-02 4.55729E-03

Fig. 5. Resultat af kørsler med et program hvor to variable, x og y, indgår i den identitet der er anvendt. For hver identitet er resultatet for mikrodatamaten anført i første linie, og resultatet for minidatamaten i anden linie. Antallet af testede talpar var i begge tilfælde $15 \cdot 16/2 = 120$.

Resultatet blev:

	MIN	MAX	SUM/N/(N+1)*2
Mikrodatamat:	-6.12583E-07	5.33026E-07	8.65905E-08
Minidatamat:	-7.64779E-07	7.64097E-07	4.38893E-08

altså af samme størrelsesorden som maskinernes præcision. Der er derfor ikke noget mærkeligt ved de store absolutte fejl på fig. 5, snarere kan man måske undre sig over hvor små de absolutte fejl kan holdes visse steder, f. eks. ved beregning af eksponentialfunktionen (se fig. 4), hvor man kommer op på at beregne så store tal som e^{10} .

Det fremgår altså at mikrodatamaten heller ikke falder igennem overfor minidatamaten hvad angår beregningsøjagtighed. I visse tilfælde er den endda overlegen, f. eks. ved beregning af tangensfunktionen. Jeg vil tro at den lille værdi af MAX i forhold til minidatamatens ret store værdi skyldes, at mikrodatamaten "giver op" på et senere tilspunkt i nærheden af diskontinuitetspunktet $\pi/2$ på grund af de to ekstra bits i tallene.

Andre målinger

En vigtig egenskab ved et programmelsystem er hvor store datasæt man kan arbejde med. Dette prøvede jeg på de to maskiner ved at udføre et program der kun bestod af en DIM-sætning, hvis størrelsesangivelse jeg varierede. Det viste sig at på mikrodatamaten var den maksimale størrelse der ikke gav fejludskrift afhængig af hvor mange gange DIM-sætningen var blevet rettet. Dette tyder på at rettede programlinier bliver stående i maskinens lager og tager plads op, sandsynligvis indtil man gemmer programmet på baggrundslager. På minidatamaten var den største dimension uafhængig af antal rettelser. De største dimensioner var henholdsvis 3415 talvariable for mikrodatamaten, og 4034 talvariable for minidatamaten.

Den tilladte kompleksitet af beregningsudtryk hænger nøje sammen med hvor mange mellemresultater der kan gemmes i løbet af evalueringen af et aritmetisk udtryk. Et mål herfor er dybden af parentesniveauer. Denne dybde

målte jeg ved at taste følgende "program":

```
100 LET X=(((((((
```

og blive ved med "(" indtil maskinen gjorde vrøvl. Jeg fandt 9 for mikrodatamaten og 7 for minidatamaten.

En anden interessant dybde er hvor mange procedurer man kan kalde inden i hinanden. Dette undersøgte jeg med følgende rekursive pit

```
100 LET N=0
110 EXEC P
120 STOP
130 PROC P
140 LET N=N+1
150 PRINT N
160 EXEC P
170 ENDPROC
180 END
```

Det største tal der blev trykt inden fejludskrift var 44 på mikrodatamaten og 7 på minidatamaten. Ingen af resultaterne blev påvirket af at der blev reserveret plads med DIM-sætning i begyndelsen af programmet. Jeg fik præcis samme resultat med et analogt program hvor jeg brugte GOSUB i stedet for EXEC.

Dybden af FOR-, REPEAT- og WHILE-sætninger målte jeg med programmer hvis idé er at Comal-fortolkeren ikke opdager at man hopper ud af en løkke. Til illustration vises programmet for FOR-sætningen:

```
100 LET K=0
110 FOR I=1 TO 7
120 LET K=K+1
130 PRINT K
140 GOTO 110
150 NEXT I
160 END
```

Dette program nåede til K=22 på mikrodatamaten, men på minidatamaten kørte det uendeligt. Derimod gav analoge programmer for REPEAT- og WHILE-løkker fejludskrift efter K=29 på mikrodatamaten, og efter K=7 på minidatamaten.

Til yderligere undersøgelse af FOR-sætningen kørte jeg to små programmer der viser noget om hvad der sker når man har aritmetiske udtryk efter TO og STEP. Programmerne så således ud:

```

100 LET N=10
110 FOR I=1 TO N-I
120 PRINT I
130 NEXT I
140 END

```

og:

```

100 LET N=2000
110 FOR I=1 TO N STEP I
120 PRINT I
130 NEXT I
140 END

```

Mikrodatamaten trykte talrækken 1, 2, 3, 4, 5 når man kørte første program, og potenserne af 2, dvs. 1, 2, 4, 8, . . . indtil 1024, når man kørte andet program. Dette viser at udtrykkene efter TO og STEP evalueres ved hvert nyt gennemløb af FOR-løkken på mikrodatamaten. På minidatamaten fik jeg talrækken fra 1 til 9 med første program, og talrækken fra 1 til 2000 med andet program. Her evalueres udtrykkene efter TO og STEP åbenbart kun ved første gennemløb af FOR-løkken. Det er muligvis derfor minidatamaten er hurtigst til løkkeregning.

Undersøgelse af de tilfældige tal

De sidste undersøgelser jeg nåede at foretage inden mikrodatamaten blev afhentet igen, vedrørte de tilfældige tal der produceres af RND-funktionen. Det begyndte med at jeg kørte et program der beregnede sumfunktionen svarende til de tilfældige tal. Programmet ses på fig. 6. Både på mikrodatamaten og minidatamaten fik jeg en smuk ret linie ud af det, jo smukkere desto flere tilfældige tal jeg brugte.

Nogle studerende der også gik og legede med mikrodatamaten påstod imidlertid hårdnakket at der var knas med de tilfældige tal. Vores operatør, Peter Jacobsen, fandt hullet under nogle eksperimenter med maskinens CURSOR-sætning, ved hjælp af hvilken man kan flytte markøren rundt på skærmen. Et af hans programmer gik ud på at generere tilfældige tal i par, som han derefter opfattede som koordinater til et punkt på skærmen, hvor han så trykte en stjerne. Man skulle tro at stjernerne fordelte sig jævnt over hele skærmen, men efter lang tids kørsel med programmet var der stadig nogle skrå diagonaler på skærmen, hvor der slet ikke kom nogle stjerner.

Jeg lavede en version af hans program, der opsamlede information i et todimensionalt skærm-

billede, som jeg derefter trykte på matrixskriveren. Programmet ses på fig. 7. Dette udelukker samtidig den mulighed at det er CURSOR-sætningen der er noget i vejen med. Resultatet af en kørsel ses på fig. 8. Jeg prøvede også at starte programmet med en RANDOMIZE-sætning. Stjernerne flyttede sig lidt, men der kom stadig ingen stjerner i korridorerne på fig 8.

På minidatamaten var der ikke noget problem - her blev skærmen jævnt belagt med stjerner.

Programmet er lidt sværere at programmere fordi der ikke findes nogen CURSOR-sætning på minidatamaten, men ved brug af de anvisninger der er offentliggjort i RC 7000-åren (DATA-LÆRE nr. 2, februar 1979, side 24), lod det sig gøre.

Eksperimentet viser at de tal der genereres af RND-funktionen på mikrodatamaten ikke er så tilfældige når man betragter rækkefølger af tallene. Hvad grunden kan være har jeg ikke undersøgt nærmere, det var der ikke tid til.

```

100 INPUT N,M
110 DIM A(M)
120 FOR I=1 TO M
130 LET A(I)=0
140 NEXT I
150 FOR I=1 TO N
160 LET X=RND(0)
170 FOR J=1 TO M-1
180 IF X<J/M THEN LET A(J)=A(J)+1
190 NEXT J
200 LET A(M)=A(M)+1
210 NEXT I
220 PRINT N;M
230 FOR I=1 TO M
240 PRINT I/M,TAB(20+50*A(I)/N); "*"
250 NEXT I
260 END

```

Fig. 6. Program til test af tilfældige tal. De M tælleværker med navnet A svarer til en inddeling af intervallet fra 0 til 1 i M lige store grupper. Der genereres N tilfældige tal, X, og for hvert X tælles 1 op i alle de tælleværker der svarer til talværdier større end X. For hvert tælleværk trykkes sluttelig dels dets øvre grænse, og dels en stjerne hvis afstand fra venstre kant af papiret er proportional med tælleværkets indhold. Stjernerne danner tilsammen et billede af sumfunktionen for de tilfældige tal.

Det sidste jeg undersøgte var RANDOMIZE-sætningen. Hvis man på mikrodatamaten kører følgende program:

```
100 INPUT N
110 FOR I=1 TO N
120 PRINT-RND(0)
130 NEXT I
140 END
```

så får man trykt en søjle af N tal der fordeler sig tilfældigt mellem 0 og 1. Det samme sker på minidatamaten. Indfører man nu ændringen:

```
115 RANDOMIZE
```

så sker der det at der fremkommer lange rækker af ens "tilfældige tal" på skærmen. Slår man matrixskriveren til, så tallene i stedet trykkes på den, bliver de ens i grupper på to eller tre, og hver gang de ændrer sig er det kun meget lidt. Denne mærkelige opførsel skyldes efter al sandsynlighed at RANDOMIZE bruger maskinens interner ur som sæd for de tilfældige tal, men uret går desværre så langsomt at man kan nå mange gange rundt i løkken (afhængigt af hvor hurtigt trykningen sker) inden uret ændrer sig. Endvidere vil en lille ændring af uret give en tilsvarende lille ændring af de tilfældige tal.

På minidatamaten var tallene stadig forskellige hvad enten man brugte RANDOMIZE eller ej, eller rettere sagt: man kunne ikke konstatere nogen regelmæssighed i tallene ved en umiddelbar betragtning.

Afslutning

Prisbilligheden på den nye mikrodatamat er ikke opnået på bekostning af en dårligere præstation af centralenheden, når man kører i et højere sprog som Comal. Til enkeltbrugere der har brug for talberegninger står mikrodatamaten på højde med minidatamaten. Undersøgelsen viser intet om de to maskiners indbyrdes forhold med hensyn til arbejde med store datamængder, ligesom deres operativsystemer ikke er blevet undersøgt. Der er også kun gennemført målinger med en enkelt bruger, så deres indbyrdes præstationer ved multiprogrammering fremgår ikke.

Denne undersøgelse kan derfor ikke alene danne grundlag for en endelig dom over de to maskiners egnethed til f. eks. skolebrug, og man kan naturligvis slet ikke slutte noget om mikro- og minidatamater i al almindelighed, på grundlag af målinger udført på to konkrete fabrikater. Alligevel synes jeg at resultatet af undersøgelsen er tankevækkende.

Jeg har forelagt resultatet af undersøgelsen for de to fabrikanter af maskinerne. Ingen af dem bekvilede rigtigheden af de opnåede måleresultater. Fabrikanten af minidatamaten bemærkede dog, at jeg havde benyttet et anlæg med plade-lager, og ikke med diskette. Det betyder at anlægget arbejder virtuelt, således at man kan køre store programmer i et relativt lille lager. Dette belastet systemet med lidt overhead, selv når man har plads nok, og et diskettesystem kører derfor lidt hurtigere for assignment, løkker og GOTO. Tidsgevinsten er størst for kald af underprogrammer, hvor fabrikanten rapporterer om op til en halvering af tiden i forhold til fig. 2. Tiden for beregning af standardfunktioner er uændret. Mikrodatamatfabrikanten nævnte at de naturligvis løbende arbejder med forbedringer af systemet, og erkendte at de tilfældige tal synes at trænge til en "ansigtsløftning".

SAS - edb

en gentagelse af mødet d. 7. maj.

Tirsdag d. 25. september kl. 19.00 præcis.
Hvordan løses billetbestilling og fragtbestilling via edb?
Hvordan holder SAS styr på fly og personale via edb?
Hvordan styres lastning og reparation af fly via edb?

Disse spørgsmål samt spørgsmål ang. hardware, software, systemer, applikationer sikkerhed, leverandører osv. vil blive besvaret/belyst gennem foredrag, filmforevisning og rundvisning af datachef Arne Hansen og hans medarbejdere Erik Quist Sørensen og Ole Skaarup. Mødet finder sted hos SAS i Kastrup på Engvej 165, blok 2.
Begrænset deltagerantal

Tilmelding senest torsdag d. 20. september til Gerd Belhage, (02) 97 10 46.

Københavns - fraktionen



FORLAGET BOGIKA ApS

Akacieparken 38

7430 Ikast

Tlf. (07) 15 31 55

paul bjørnum

Den mest anvendte bog i faget datalære ved basisuddannelsen, fordi den er:

1. pædagogisk opbygget
2. letlæselig for eleverne
3. et godt hjælpemiddel ved eksamen
4. der er udarbejdet vejledende løsninger til alle opgaver og øvelser.

På grund af det store oplag kan prisen stadig holdes på kr. 40,- (excl. moms).

DATA- LÆRE

Grundbog med opgaver

Det bemærkes, at der maksimalt skal trykkes seks linier oven i hinanden for at få en given billedrække frem. Dvs., lineskriverpapiret skal være af en rimelig kvalitet for at kunne tåle sådanne overskrivninger.

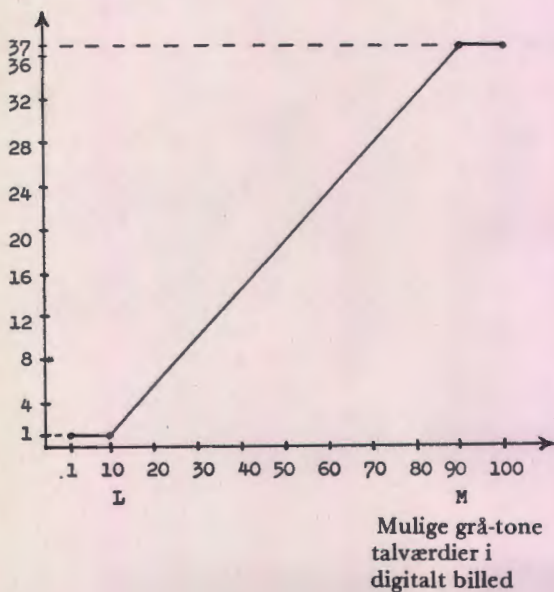
Yderligere bemærkes, at et givet tegn trykt oven i sig selv giver et mørkere tegn end tegnet selv.

Transformation til grå-tone skala

Den ovenfor angivne grå-tone skala (figur 2) indeholdt 37 grå-tone værdier, nummereret fra 1 til 37. Men det er næppe sandsynligt, at et givet digitalt billede indeholder heltallige grå-tone talværdier i intervallet 1 til 37. F. eks. kan et digitalt billede være givet ved heltallige grå-tone talværdier i intervallet 0 til 99 eller ved reelle grå-tone talværdier i intervallet fra 0 til 1.

Før et givet digitalt billede kan udtegnes på lineskriveren, er det således nødvendigt at transformere grå-tone talværdierne i det givne digitale billede til hele tal i intervallet 0 til 37.

Grå-tone nr. i 37-trins grå-tone skala



Figur 3

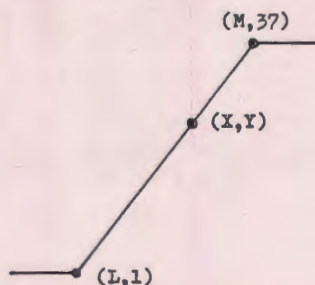
Eksempel på transformation fra grå-tone talværdier i digitalt billede til grå-tone numre i 37-trins grå-tone skala.

Denne transformation kræver to transformationsparametre, L (lys) og M (mørk). Lad os tage et eksempel. Antag, at et digitalt billede er givet ved heltallige grå-tone talværdier i intervallet 1 til 99. Grå-tone talværdier mindre end eller lig med 10

ønskes udtegnet som blanke (grå-tone nr. 1 i figur 2). Derfor sættes L til 10. Grå-tone talværdier større end eller lig med 90 udtegnes mørkest muligt (grå-tone nr. 37 i figur 2). Derfor sættes M lig med 90. Grå-tone talværdier i intervallet 11 til 89 transformeres til grå-tone numre (figur 2) ved lineær interpolation mellem punkterne (10,1) og (90,37) hvor abscissen altså angiver en grå-tone talværdi i det givne digitale billede og ordinaten et grå-tone nr. i skalaen fra figur 2. Figur 3 viser transformationen grafisk.

Fremgangsmåden skitseret ovenfor kan, for et vilkårligt digitalt billede, generaliseres således:

1. Bestem transformationsparametrene L og M. Disse kan bestemmes på mange måder. F. eks. kan L og M fastholdes for alle digitale billeder udtrykt ved grå-tone talværdier i et bestemt interval. Dette var tilfældet i eksemplet ovenfor, hvor L og M blev fastholdt på værdierne 10 og 90 for alle billeder med heltallige grå-tone talværdier i intervallet 1 til 99. Parametrene L og M kan også bestemmes for hvert billede. F. eks. ved at sætte L og M til hhv. mindst og størst forekomne grå-tone talværdi i det givne billede.
2. Transformer fra grå-tone talværdi (X) i det digitale billede til gråtone nr. (Y) i skalaen, figur 2.



Figur 4

Transformation af given grå-tone talværdi (X) til grå-tone nr. (Y).

Af figur 4 ses, at for X beliggende mellem L og M gælder:

$$\frac{X-L}{M-L} = \frac{Y-1}{37-1}$$

hvoraf fås

$$Y = \frac{36}{M-L} (X-L) + 1$$

eller, ved at sætte $A = 36/(M-L)$ fås:

$$Y = A(X-L) + 1$$

Transformationen kan da angives ved:

(Fortsættes side 16)

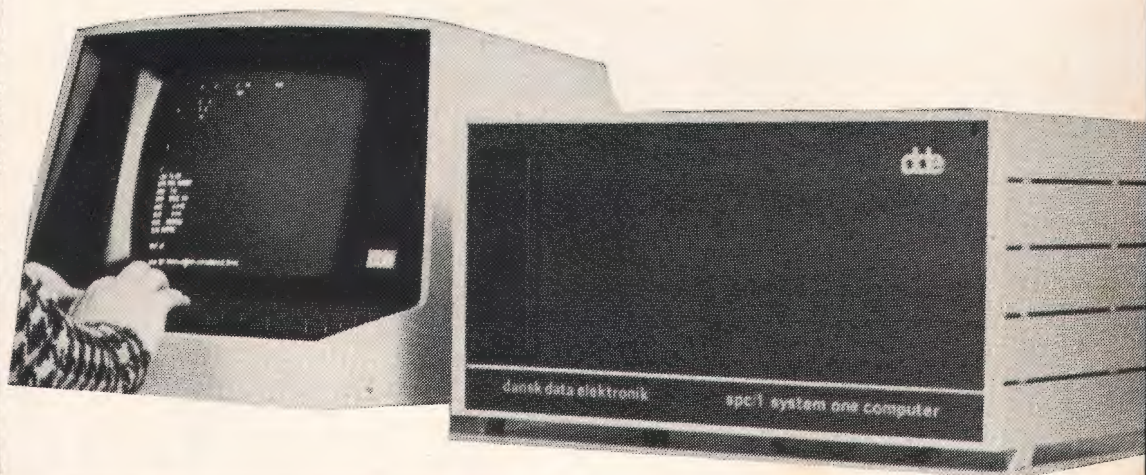
Hvorfor gøre det svært??

**vælg den lille maskine
til skolemarkedet
med udbygningmuligheder**

SPC/1

**fra 1 til 8 brugere
på et prislæg**

GRUNDVERSION 19.500kr
mikrodatamat + dataskærm ex. moms



— Når det er så nemt!!

— med de store muligheder

— for store og små

— på alle niveauer

fra 


— med omfattende programmel

— hvor ALLE kan deltage

dansk data elektronik aps

Herlev Hovedgade 207

2730 Herlev +02 84 5011

 er et dansk firma; vi udvikler, producerer og sælger mikrodatamatudstyr til mange formål.

(Fortsat fra side 13)

$$Y = \begin{cases} 1 & \text{hvis } X \leq L \\ A(X-L)+1 \text{ afrundet} & \text{hvis } L < X < M \\ 37 & \text{hvis } X \geq M \end{cases} \quad M = \left\lfloor \frac{A \cdot 36}{M-L} \right\rfloor \quad (1)$$

Med grå-tone talværdier (X-værdier) i intervallet 1 til 99, $L=10$ og $M=90$ (som i eksemplet figur 3), transformeres f. eks. X-værdierne 7, 47 og 97 til Y-værdierne (grå-tone numrene) 1, 18 og 37.

Bemærk, at intensiteten af et givet billede kan ændres ved at parallelforskyde transformations-grafen parallelt med første-aksen, dvs. ved identiske ændringer i parametrene L og M. Ligeledes kan kontrasten af et billede ændres ved at ændre hældningen af det skrå stykke på transformations-grafen, dvs. ved forskellige ændringer i parametrene L og M.

Såfremt transformations-grafen ikke indeholder vandrette stykker, dvs. hvis man sætter $L=1$ og $M=99$ i eksemplet ovenfor, kan "negativet" af et billede dannes ved at bytte om på værdierne af L og M, dvs. ved at sætte $L=99$ og $M=1$ i eksemplet ovenfor. Endelig bemærkes, at også ikke-lineære transformationer kan anvendes, f. eks. parabelstykker.

Udtegning

Ved udtegning af et givet billede behandles en linie ad gangen. Hertil anvendes tre en-dimensionale talsæt (s = antal søjler i billede):

BL(1:s), utransformerede grå-tone talværdier for den givne linie i det givne billede (BL = billedelinie).

SL(1:s), Transformerede grå-tone numre for denne linie, (SL=skalalinie).

UL(1:s), indeholder de tegn der skal udskrives, (UL=udskriftlinie). Fyldes op maksimalt seks gange for hver skalalinie (se figur 2).

Det bemærkes, at BL(1:s) ikke behøves erklæret særskilt, idet det er en række i det to-dimensionale talsæt, der indeholder det givne digitale billede.

Desuden anvendes ved udtegningen en to-dimensionel tabel, SKALA(1:37,1:6), der indeholder den 37-trins grå-tone skala (se figur 2). Denne tabel er antydnet i figur 5, hvor de angivne tegn i et program skal erstattes af tegnets værdi. Blanke celler angiver tegnet "mellemslag" (blanktegn). Denne tabel skal initialiseres først i programmet. Som tidligere bemærket skal der maksimalt trykkes seks linier oven i hinanden for at få en given billedrække frem. Ofte kan færre end seks gøre det, thi af skalaen (figur 2) ses, at kun grå-tone numre større end eller lig med 28 giver anledning til, at der trykkes tre eller flere linier oven i hinanden. Derfor indføres en heltalsvari-

abel MAX, hvis værdi er det største grå-tone nummer i talsættet SL(1:s). Idet det da vedtages, at der altid trykkes mindst to linier oven i hinanden, kan ANTAL (heltalsvariabel der for en given billedrække angiver, hvor mange linier der skal trykkes oven i hinanden) da bestemmes ved:

$$\text{ANTAL} = \begin{cases} 2 & \text{hvis } \text{MAX} \leq 27 \\ 3 & \text{hvis } 28 \leq \text{MAX} \leq 32 \\ 4 & \text{hvis } 33 \leq \text{MAX} \leq 34 \\ 5 & \text{hvis } 35 \leq \text{MAX} \leq 36 \\ 6 & \text{hvis } \text{MAX} = 37 \end{cases}$$

i \ j	1	2	3	4	5	6
1						
2	'					
3	-					
.						
.						
.						
36	M	W	&	O	Ø	
37	M	W	&	O	Ø	+

Figur 5
Tabel (SKALA(i,j), $i=1, \dots, 37, j=1, \dots, 6$) der indeholder den 37-trins grå-tone skala. Se figur 2 angående udfyldelse af resten af tabellen.

Efter at have indført ovenstående variable kan fremgangsmåden ved trykningen af en billedrække illustreres ved et simpelt eksempel. Antag, at billedrækken BL(1:s) er givet ved (kun tre grå-tone værdier medtages, smlgn. med eksemplet sidst i forrige afsnit):

..... 7 47 97
BL(4) BL(12) BL(25)

Såfremt $L=10$ og $M=90$ bliver de transformerede grå-tone numre, SL(1:s), som følger:

..... 1 18 37
SL(4) SL(12) SL(25)

Da $\text{MAX}=37$ bliver $\text{ANTAL}=6$. Der skal altså, for at tegne denne billedrække, trykkes seks linier oven i hinanden. Dvs. UL(1:s) skal fyldes op med tegn (hentet fra SKALA (1:37, 1:6)) seks gange

skalasøjle (figur 5). Værdierne af disse variable opfylder ulighederne $1 \leq RK \leq R$, $1 \leq SJ \leq S$ og $1 \leq SKSJ \leq \text{ANTAL} \leq 6$.

Notationen "M" anvendes for tegnværdien af tegnet mellem apostrofferne, her M. Således betyder

f. eks. " " tegnværdien for mellemslag (blanktegn). Skitsen er skrevet i et tillempet programmeringssprog, hvor hvis-så-ellers er medtaget for at undgå for lange konstruktioner.

```
// Initialisering af skalatabel //
// Tabellen fyldes først med blanktegn //
for I:=1 til 37
  for J:=1 til 6
    SKALA(I,J):=" "
  slut for
slut for;
// Dernæst udfyldes de celler i skalatabellen der ikke //
// skal være blanke, sammenlign evt. med figur 5 //
SKALA(2,1):=" ' ";
SKALA(3,1):=" - ";
.
.
.
SKALA(36,1):="M"; . . . . . SKALA(36,5):="Ø";
SKALA(37,1):="M"; . . . . . SKALA(37,6):="+";

// Indlæs og check transformationsparametre L og M //
indlæs(L,M);
hvis L=M så fejl;

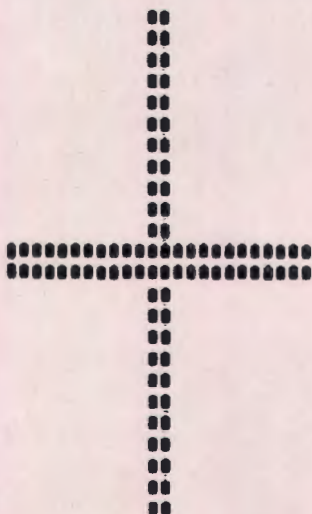
// Beregn konstanten A til formel (1) //
A:=36/(M-L);

// Udskriv sideskift og eventuel overskrift //
udskriv(sideskift, "OVERSKRIFT");

// Er nu klar til at behandle næste billedrække //
for RK:=1 til R
  // Transformation fra grå-tone talværdier til skala- //
  // numre, formel (1). Værdien af MAX bestemmes samtidig //
  MAX:=1;
  for SJ:=1 til S
    X:=BL(SJ);
    hvis X ≤ L så Y:=1 ellers
    hvis X ≥ M så Y:=37 ellers Y:=afrund(A(X-L)+1);
    SL(SJ):=Y;
    hvis Y > MAX så MAX:=Y;
  slut for;
  // Bestem værdien af ANTAL //
  hvis MAX > 36 så ANTAL:=6 ellers
  hvis MAX > 34 så ANTAL:=5 ellers
  hvis MAX > 32 så ANTAL:=4 ellers
  hvis MAX > 27 så ANTAL:=3 ellers ANTAL:=2;
  // Udskriv linieskift //
  udskriv(linieskift);
  // Dan og udskriv overtrykkslinier for denne billedrække //
  for SKSJ:=1 til ANTAL
    for SJ:=1 til S
      UL(SJ):=SKALA(SL(SJ),SKSJ)
    slut for;
    udskriv(tegn i UL(1:s) forfra på denne linie);
  slut for;
slut for;
```

Eksempler

Figur 6, 7 og 8 viser nogle eksempler på tegning på lineskriver. Figur 6 og 7 er ikke så spændende, men giver dog en fornemmelse af mulighederne. Derimod viser figur 8, delfinerne, et mere detaljeret og smukt eksempel; (Hold billedet ud i strakt arm og drej det til lyset falder rigtigt). Desværre kan grå-tone værdierne til delfin-eksemplet ikke oplyses, da jeg ikke kender dem. Hvis nogen får lyst til at prøve at tegne ved brug af lineskriveren, kræver det et eller flere digitale billeder at prøve med. Da det kræver et større arbejde at udarbejde et nogenlunde detaljeret prøvebillede af en vis størrelse (f. eks. 32 x 32 eller 64 x 64), er det nok lettest at starte med simple billeder, f. eks. i stilen som vist i figur 6 og 7. Får man blod på tanden, kan man altid gå i gang med et større digitaliseringsarbejde.



Figur 6

En nem metode til at fremstille rimeligt detaljerede billeder er i øvrigt at generere grå-tone talværdier som funktion af indices i det to-dimensionale talsæt, der indeholder det digitale billede. F. eks. kan man sætte $BL(i,j)=i^2+j^2$, $BL(i,j)=i*j$ eller hvad man nu lyster. Fjernsynets prøvebillede kan måske også inspirere.

Som tidligere nævnt håber jeg, at læsere der eventuelt får lyst til at forsøge sig ud i kunsten at tegne på en lineskriver, vil offentliggøre resultatet i bladet.

Foreløbig ønskes god fornøjelse.

Figur 7



Figur 8
Delfiner



METRIC ABC 80 – 80'ernes »PRIVAMAT«

METRIC ABC 80 – Datamaten som kan erhverves af alle. Nyeste teknik – Stor kapacitet – Lav pris.

METRIC ABC 80 – Nøglen til indførelse af EDB i hverdagen. METRIC ABC 80 klarer eksempelvis bogholderi, lagerstyring, kunderegistrering, teknisk/videnskabelige beregninger, industriel kontrol, styring, registrering, undervisning m.v.

METRIC ABC 80 kan udbygges med en lang række ydre enheder efterhånden som behovet stiger. F.eks. printere, floppy diske, plotter, digitizer m.fl.

METRIC ABC 80 – Skandinavisk fremstillet – Udførlig og let læselig dokumentation – Stadigt voksende programbibliotek.

PRIS inkl. kassetteenhed kun kr. 8.990,- ekskl. moms.

SC **METRIC** AIS

Dataafdelingen, Skodsborgvej 305, 2850 Nærum, tlf.: (02) 80 42 00.

QUICKSORT

brugt til extern sortering

af Børge Christensen

I den sidste artikel gennemgik jeg quicksort (datalære nr. 2. feb. 79, p. 21 - 30), og jeg viste ved et eksempel, hvordan en række tal blev sorteret ved denne algoritme.

Quicksort kan også bruges til sortering af poster i en fil, som står skrevet på en diskette eller en disk, men processen skal modificeres en smule, før den kan anvendes til dette formål, og der skal tages en hjælpefil i brug til støtte for sorteringen af hovedfilen.

For at forstå princippet i sorteringen vil vi først betragte følgende lille eksempel:

Fem udmærkede gentlemen skal løbe om kap. Det drejer sig om: Emil, Teddy, Torsten, Torben og Erling. De kondi-glade herrer stiller naturligvis op med hvert sit nummer på ryggen, så også fans i det fjerne kan følge løbet og komme med opmuntrende tilråb til deres idoler. Der er tale om følgende nummerering:

1	TEDDY
2	EMIL
3	TORSTEN
4	ERLING
5	TORBEN

Løbet starter ved en kulkran i Odense havn, og de fem løbere forsvinder i en sky af støv og begejstrede tilråb i retning af Ejby Overdrev. Da de igen når Odense havn og kommer i mål én for én, viser det sig sælsomt nok, at de netop kommer i alfabetisk orden. På resultatlisten læser man derpå - skrevet med kridt - følgende:

1	2
2	4
3	1
4	5
5	3

Denne tavle skal læses således: Løbet blev vundet af Emil (nr. 2), og dernæst fulgte Erling (nr. 4). Efter Erling kom Teddy (nr. 1) i mål fulgt af Torben (nr. 5). Den sidste, der kom ind, var Torsten (nr. 3). Man lægger mærke til, at navnene på deltagerne i løbet ikke optræder på tavlen. I stedet optræder løbernes numre. Man kan sige, at løberne er repræsenteret ved deres numre.

Lad os nu i stedet tænke os, at vi har en datafil, som indeholder oplysninger om de samme fem personer. I hver post i filen står foruden personens fornavn også vedkommendes efternavn,

adresse, personnummer, m.v. En enkelt post kan afbildes således:

|fornavn|efternavn|adresse|postdistrikt|cpmr|...

Vi kan endvidere tænke os, at de fem personer står opført i filen i den samme rækkefølge som i løbet. Post nr. 1 indeholder altså oplysninger om Teddy, post nr. 2 indeholder oplysninger om Emil, osv. Vi ønsker nu at få sorteret de fem filer i rækkefølge efter fornavnene. Vi vil kort sagt benytte fornavnene som nøgler for sorteringen. Vi kan naturligvis sætte datamaten til at bytte rundt på indholdet af posterne, så posten med oplysninger om Emil kommer til at stå først, efterfulgt af posten med oplysninger om Erling, osv., og det vil ikke være nogen større sag, så længe det kun drejer sig om de nævnte fem sæt af persondata. Hvis filen derimod er på flere tusinde poster, er det klart, at selv en hurtig datamat får noget at se til. Som vi har set i de foregående artikler, bygger enhver sortering på en mængde ombytninger af de involverede elementer, og en ombytning af to poster i en fil er et ganske omstændeligt foretagende. Det totale indhold af begge poster skal kopieres i arbejds-lageret og derpå skrives op i pladeleret igen. Vi vil derfor vælge en anden fremgangsmåde, der bygger på samme idé som resultatlisten fra væddeløbet: I stedet for at flytte omkring med posterne, flytter vi omkring på deres numre! Når sorteringen er tilendebragt, har vi fået opstillet en tabel, der ser ud nøjagtig som resultatlisten, og af hvilken vi kan læse, hvilken post, der er den første i rækken, hvilken der er den næste, osv. Det betyder naturligvis, at vi foruden filen selv, må opretholde en sådan tabel, i hvilken vi til enhver tid kan slå op og finde den post, vi søger. Denne tabel er vi nødt til at skrive i en særlig fil, indeksfilen, i det ydre lager. Det kunne umiddelbart se ud, som om vi har gjort det hele mere omstændeligt ved at oprette endnu en fil, foruden den vi havde i forvejen, men en kort overvejelse skulle gøre det klart for læseren, at det er langt lettere at manipulere med indeksfilens elementer end med hovedfilens. Dertil kommer, at man, inden arbejdet med hovedfilen begynder, læser en kopi af tabellen ned i arbejds-lageret. Det kræver selvfølgelig, at der skal være plads til den, og netop denne omstændighed sætter en vis grænse for metodens anvendelighed. Det er imidlertid klart, at tabellen i almindelighed vil fylde mange gange mindre i arbejds-lageret end hovedfilen selv. Som bilag til artiklen har jeg skrevet et program, som kan foretage en sortering af en fil under benyttelse af quicksort med indekstabel. I linie 70 - 90 oprettes en indekstabel i arbejds-lageret. Jeg har kaldt den for PIL, idet jeg opfatter dens komponenter som pegepinde eller pile til de en-

kelte poster i hovedfilen. Derpå åbnes hovedfilen og sorteringen påbegyndes (100 - 110).
 Proceduren SORTERING er fremkommet ved, at jeg har rettet i den tilsvarende procedure i artiklen om quicksort. Den første afgørende forskel på de to procedurer finder man i linje 250, som bør sammenlignes med den sidste tildeling i linje 70 af den oprindelige procedure. I linje 250 står der:

```
READ FILE(0,PIL((V+H) DIV 2)), PIVOT$
```

I stedet for at tildele pivot'en den værdi, som komponent nr. (V+H) DIV 2 har, læser vi det første felt i den post, som komponent nr. (V+H) DIV 2 peger på. Lad os tænke os, at (V+H) DIV 2 har værdien 15, og at PIL (15) er lig med 7. Sætningen ovenfor vil da bevirke, at værdien af det første felt i post nr. 7 bliver tildelt den variable PIVOT\$. Som tidligere nævnt regner vi med, at dette felt indeholder fornavnet. Læg i den anledning mærke til, at det er det eneste felt i posten, vi har brug for. Resten læser vi ikke!

Der næst er der på afgørende måde ændret i den oprindelige procedure i linjerne 270, 300, 320 og 350. Her finder man sætninger af typen:

```
READ FILE (0,PIL(I)),KEY$
```

I stedet for - som oprindeligt - at sammenligne pivot'en med den I'te komponent i en vektor, sammenligner vi den med nøglen - i dette tilfælde indholdet af første felt - i den post, som udpeges af PIL(I).

Man kan sige, at vi bruger PIL-tabellen på samme måde, som visse poster i et orienteringsløb: Man kigger efter i meldingen ved den pågældende post for at finde adressen på det næste sted, man skal søge.

Den helt afgørende ændring finder man i linje 380:

```
LET BUF=PIL(I); PIL(I)=PIL(J); PIL(J)=BUF
```

I stedet for at ombytte indholdet af posterne, bytter vi om på pegepindene til posterne, og angiver dermed, at disse har fået nye relative placeringer i "løbet". Vi sorterer altså indekstabelens elementer på grundlag af de data, de repræsenterer. Indekstabellens tal er i sig selv aldeles uinteressante. Det, de peger på, er sagen.

Hermed er beskrivelsen af denne sorteringsmetode sådan set tilendebragt. Tilbage står dog nogle "oprydningsbemærkninger". For det første: Hvordan skal man sætte indekstabelen op, inden sorteringen påbegyndes? Det simpleste er, at man lader værdien af hver enkelt komponent i PIL være lig med værdien af indekset. Hertil kan bruge en programstump som denne:

```
FOR I=1 TO MAX
  PIL(I) = I
NEXT I
```

22

PIL(1) peger altså til at begynde med på post nr. 1, PIL(2) på post nr. 2, osv.

Et par ændringer af indekstabelen er sikkert heller ikke af vejen. Man kan fx. bruge den, hvis man ønsker en udskrift af filens poster i den orden, nøglerne angiver. Hertil kan man bruge:

```
FOR I=1 TO MAX
  READ FILE(0,PIL(I)), <liste af feltvariable>
  "Her udskrives værdierne af de variable"
NEXT I
```

Man kan også, hvis man har plads i det ydre lager, skrive en ny kopi af filen, idet posterne nu skrives i den rækkefølge, indekstabelen angiver. Når det er gjort, er filen i realiteten blevet sorteret i sig selv, og man kan evt. slette indeksfilen. Hertil kunne man bruge:

```
FOR I=1 TO MAX
  READ FILE (0,PIL(I)), <liste af feltvariable>
  WRITE FILE(1,I), < liste af feltvariable>
NEXT I
```

Den gamle fil er åbnet med kanal 0 og den nye med kanal 1. Ved søgning i filen kan man benytte den tidligere omtalte binære søgning, idet man udpeger elementer til sammenligning med søgegrundlaget ved hjælp af indeksfilen. Læserne opfordres til at modificere den tidligere angivne algoritme, så metoden med indekstabel kan bruges ved søgningen (se datalære nr. 3, 1978 p.4-8). I den næste artikel vil jeg begynde gennemgangen af fletning af filer.

```
0030 INPUT "ANGIV ANTAL POSTER I FILEN: ",MAXANTAL
0040 DIM PIL(MAXANTAL),HOVEDFIL$(10)
0060 INPUT "HVAD HEDDER HOVEDFILEN? ",HOVEDFIL$
0070 FOR I=1 TO MAXANTAL
0080 LET PIL(I)=I
0090 NEXT I
0100 OPEN FILE (0,0),HOVEDFIL$
0110 EXEC SORTERING
0120 CLOSE
0130 END OF MAIN
0150 PROC SORTERING
0160 DIM STAK(20,2),KEY$(30),PIVOT$(30)
0170 LET SP=1; STAK(SP,1)=1; STAK(SP,2)=MAXANTAL
0180 REPEAT ** LES STAKTOP **
0220 LET V=STAK(SP,1); W=STAK(SP,2); SP=SP-1
0230 REPEAT ** OPDELING AF INDEKSTABEL **
0240 LET I=V; J=W
0250 READ FILE (0,PIL((V+H) DIV 2)),PIVOT$
0260 REPEAT
0270 READ FILE (0,PIL(I)),KEY$
0280 UNTIL KEY$<PIVOT$ DO
0290 LET I=I+1
0300 READ FILE (0,PIL(I)),KEY$
0310 ENDUNTIL
0320 READ FILE (0,PIL(J)),KEY$
0330 UNTIL PIVOT$<KEY$ DO
0340 LET J=J-1
0350 READ FILE (0,PIL(J)),KEY$
0360 ENDUNTIL
0370 IF I<=J THEN
0380 LET BUF=PIL(I); PIL(I)=PIL(J); PIL(J)=BUF
0390 LET I=I+1; J=J-1
0400 ENDIF ** PILE OMBYTTET OG NYE GJORT KLAR **
0410 UNTIL I>J
0420 IF I<H THEN
0430 REN ** PUSH PILE TIL HØJRE DEL-TABEL **
0440 LET SP=SP+1; STAK(SP,1)=I; STAK(SP,2)=H
0450 ENDIF
0460 LET H=J
0470 UNTIL V>H
0480 UNTIL SP=0
0490 ENDPROC SORTERING
```

Sortering som CALL-rutine

En ny og samtidig meget interessant facilitet i COMAL er muligheden for kald af procedurer skrevet i assembler (CALL-rutiner), dvs. maskinkode, direkte fra COMAL programmer. Dels for at vise nogle af mulighederne med CALL-rutiner, dels fordi man ofte har brug for en hurtig sortering af numeriske størrelser, har jeg prøvet at udvikle et sorteringsprogram skrevet i maskinkode.

Kald af proceduren sker vha. COMAL sætningen
 XXXX CALL "SORT", <VEC>,
 <REAL>, <BOOLEAN>
 hvor <VEC> er den vektor der skal sorteres, <REAL> er en variabel eller et udtryk der fortæller hvor mange elementer i <VEC> der skal sorteres, og <BOOLEAN> er en boolsk variabel (ell.udtryk), der styrer sorteringsrækkefølgen. Er værdien 0 sorteres i voksende orden, ellers i aftagende. Dette muliggøres, ikke ved flere "IF" sætninger efter hinanden, eller ved at skrive 2 forskellige sorteringsprogrammer, men ved at lade rutinen modificere sig selv, på den måde, at en central spørgesætning ændres, afhængig af værdien af <BOOLEAN> ; en mulighed der jo desværre ikke findes i COMAL.

Sorteringsmetoden er helt banal; jeg har anvendt den såkaldte "bubblesort", som ses på fig.1. Det er klart at "Worst case" for denne algoritme er den situation, hvor tallene "står omvendt" i forhold til den rækkefølge, de ønskes sorteret i, og dette tilfælde er da også anvendt ved test af rutinen. (fig. 2). Alligevel er hastigheden forbavsende. Med én bruger på systemet (et almindeligt RC 7000 anlæg med DOMUS), tager det for 10, 100, 200 og 400 tal hhv. under 1, 6, 24 og 98 sek. for den assemblerkodede sortering at ordne disse, mens bubblesort i COMAL bruger ca. 2, 273, 118 og 4415 sek. henholdsvis for de samme tal, altså en meget væsentlig forbedring.

Det fremgår af dette eksempel klart at der er meget væsentlige hastighedsmæssige fordele ved CALL-rutiner, og ved at benytte en effektiv

sorteringsmetode, kan der helt givet opnås endnu bedre resultater.

Jesper Träff, 3Fz
 Nakskov Gymnasium

Efterskrift: Dokumentation af den omtalte procedure i form af ASCII strimmel + udskrift kan evt. erhverves ved indsendelse af en adresseret og frankeret prøvekuvert til:

W. Kjellberg Christensen
 Strandpromenaden 32
 4900 Nakskov

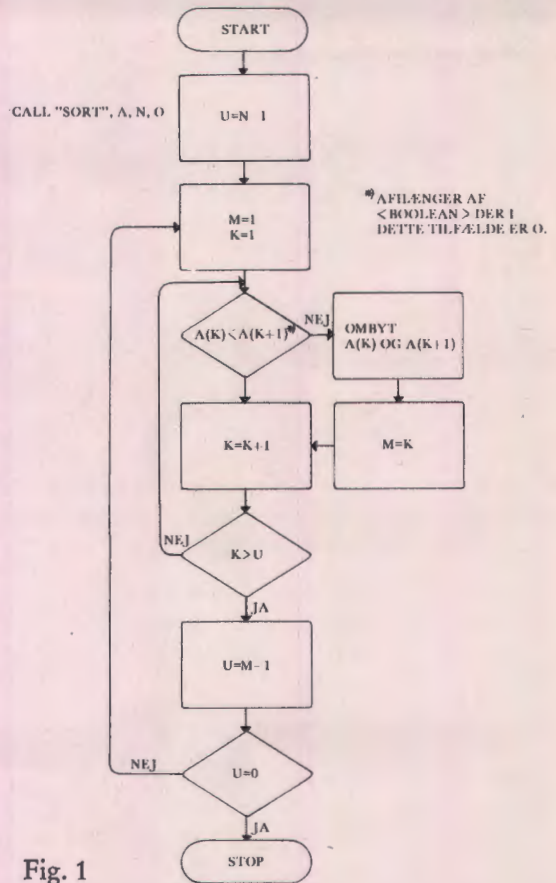


Fig. 1

```

0010 REM (TEST AF CALL-ROUTINE "SORT".)
0020 INPUT "ANTAL TAL: ",ANT
0030 INPUT "VOKSENDE ELLER AFTAGENDE: (V=0,A=1) ",RAEKKEF
0040 DIM A(ANT)
0050 FOR I=1 TO ANT
0060 LET A(I)=I
0070 IF RAEKKEF=0 THEN LET A(I)=ANT-I
0080 PRINT A(I),
0090 NEXT I
0100 PRINT "SORTERING NU."
0110 CALL "SORT",A,I,RAEKKEF
0120 PRINT "FAERDIG"
0130 FOR I=1 TO ANT
0140 PRINT A(I),
0150 NEXT I
0160 END
  
```

Fig. 2

LOKAL DATAMAT

**HUSK altid
METRIC**
før De handler-
vi har
kvalitetsprodukter
til fornuftige priser



TRÆT AF TERMINALLØSNINGER? -ANSKAF LOKAL DATAMAT!

Metric tilbyder et usædvanligt bredt program af datamater, som hver for sig kan konfigureres efter netop Deres behov. Det store udvalg af interfaces sikrer Dem anvendelse af allerede indkøbt terminaludstyr.

Beskriv Deres behov, og vi giver Dem mulighederne for valg af lokal datamat til den rigtige pris.

Hvad f.eks. med en ny fuldblods »MINI« (LSI 4) fra Computer Automation med 64K ord, 2x5 Mb Cartridge Disk, 8-kanals DIOS (interface for 8 brugere), Multiuser Basic, Fortran, Pascal, ASM, RTX. Monteret i 19" rack opstillet og indkøbt med 6 mdrs. fri service inkl. rejseomkostninger for kun kr.

ekskl. moms **130.000,-**

Kun en arbejdsplads! – Javel, en ideel opgave for en »MICRO«.

Zilog MCZ 1/05 med 60K bytes RAM, 3K PROM, 2x315K bytes Dual Floppy, seriel og parallel I/O, Basic, Cobol, PLZ, Fortran (medio 79), ASM. Opstillet og indkøbt med 6 mdrs. fri service inkl. rejseomkostninger for kun kr.

ekskl. moms **44.620,-**

Disse og mange flere er mulighederne når Metric er med i billedet. Ring til os og lad os diskutere hvilke løsninger De har brug for.

DATAUDSTYR FRA SC **METRIC** ANS

DATAAFDELINGEN, SKODSBORGVEJ 305, 2850 NÆRUM, TLF. (02) 80 42 00

BESØG HOS SAS

Et arrangement, der var en gentagelse værdig - og den kommer!

Et spoleret kardanophæng havde nær forhindret min deltagelse i foreningens besøg hos SAS mandag den 7. maj; men jeg fortryder bestemt ikke, at jeg løb risikoen for et "mekanisk sammenbrud" og luskede til Kastrup den aften.

Det lykkedes lige akkurat for mig at nå med ind gennem sikkerhedssystemerne inden forestillingen tog sin begyndelse. Umiddelbart ved min indtræden i samlingslokalet, der var godt besat, fik jeg en velgørende fornemmelse af "atmosfære", en fornemmelse der ganske stille tiltog efterhånden som besøget blev afviklet.

De dygtige SAS-folk, der med datachef Arne Hansen i spidsen bidrog til gennemførelsen af arrangementet, fortjener virkelig ros for deres meget fine evne til at spore sig ind på forsamlingen.

Programmet indeholdt først en indtroduktion om SASCO-SAS Computer System-, dernæst en rundvisning i datacentret og som afrunding af besøget en spørgerunde hyggeligt kombineret med et traktament.

Det vil i denne sammenhæng ikke være rimeligt at fremdrage detaljer fra indtroduktionen, der spændte fra fra SAS-historie til fremtidsvisioner

via beskrivelser af de tekniske, økonomiske og menneskelige forhold, der betinger og udgør SASCO.

De mange nødvendige oplysninger blev under anvendelse af flere forskellige medier og en særdeles varieret fremstillingsform gjort til en fordøjelig affære.

Hvis man ikke havde fået fat i dimensionerne for SASCO under indtroduktionen, så fik man endnu en god chance gennem den anskuelsesundervisning som rundturen udgjorde.

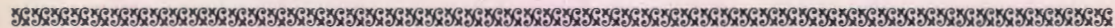
Eksempel: Et on-line-system med en bagatel af ca. 5000 terminaler fordelt over hele kloden!

Den afsluttende runde var henlagt til EDB-centrets kantine, hvor de sidste spørgsmål blev besvaret under et kaffebord i en for arrangementet karakteristisk hyggelig atmosfære.

Som en sidste gestus overfor forsamlingen fremlagde SAS-folkene ideer til et nyt arrangement, som de dermed tilbød foreningen til afholdelse i efterårssæsonen.

Den overtegnede tilmeldingsliste fortæller noget om forhåndsinteressen for dette besøg - og det var godt!

Drejer



EDB-kapacitetsudvalgets skoleudvalg har udsendt et notat om EDB-kapacitet ved undervisningen i 1. - 12. uddannelsesår.

Notatet beskæftiger sig med de nye muligheder, som MIKRODATAMATER giver og vil blive et nyttigt værktøj ved planlægning af den kommende eksplosive udvikling.

Eksemplarer kan fås ved henvendelse til:

Undervisningsministeriets Bibliotek
Frederiksholms Kanal 25 D
1220 København K
Tlf. (01) 13 52 82

AV, en tur til London

Dansk Audio-Visuelt Selskab (DAVS), arrangerer i dagene 11. - 18. november en studietur til London, hvor man vil se på undervisningsteknologi. Der er udarbejdet et udførligt program for turen, og nærmere oplysninger vil kunne fås ved henvendelse til forstander Poul Mogensen, Byggeteknisk Højskole, TLF.: (01) 81 22 90 lokal 250, eller på dennes privatnummer: (02) 98 90 73.

Har du husket



Der afholdes seminar vedr. brug af microprocesorbaserede smådatamater i undervisningssektoren på Sydjysk universitetscenter i Esbjerg i dagene 28.-29. september 1979.



Der afholdes generalforsamling lørdag, den 24. november 1979 i København.
Reserver allerede nu dagen. Nærmere om tid og sted samt om de faglige arrangementer i forbindelse med generalforsamlingen vil fremkomme senere.



DATALÆRE i Aalborg

Datalæreforsøget i Aalborg har været en succes, og fortsætter nu i forstærket omfang i dette skoleår. Interessen blandt eleverne har været meget stor, og det har sat sit præg på valgholdene på de fire skoler, som deltager i forsøget. F. eks. har man på en af skolerne haft en vækst fra 1 hold sidste år til seks hold i indeværende år. Ialt er der mere end en snes datalærehold på de fire skoler tilsammen.

Vilkårene har også været gode for datalæreundervisningen, bl. a. har man mødt stor forståelse hos skolernes ledere og ikke mindst skoleforvaltningen. Samarbejdet med Aalborg Seminarium, hvorfra man henter datakraften, fungerer også udmærket og har hidtil været til begge parter fordel, så det fortsætter fremover.

Et interessant aspekt i tilladelsen fra ministeriet til fortsættelse af forsøget var, at man direkte bad om at få inddraget erfaringer med datamaskinstøttet undervisning i rapporteringen i så vid udstrækning som muligt.

Rapportskrivningen for det forgangne år foregår for tiden, og det varer ikke længe før rapporten foreligger. Hvis man er meget interesseret, kan man henvende sig til skolekonsulent Jens Damberg, Skoleforvaltningen, Hasserisvej 174, 9000 Aalborg. (08) 12 61 44. Her vil man kunne høre nærmere om mulighederne for at få rapporten.

Københavnsmødet om økonomiske modeller og statiske analyser.

En god og givtig aften fik de folk, der i vinter troppede op til foreningens møde på Akademisk Studenterkursus. Peter Ferdinand gennemgik hvordan man i samfundsfagsundervisningen kan have glæde af datalæreundervisningen og omvendt. Man havde i samfundsfag arbejdet med anvendelse af økonomisk simulationsmodeller, simulation af sociale systemer, virksomhedsspil m.v. I datalære havde man fabrikeret spørgeskema til en velfærdsundersøgelse, samt kørt et kønsrolleprojekt, hvor man havde set på sammenhængen mellem to variable: Køn og de problemer vælgerne fandt mest interessante ved folketingsvalget 1971.

I øvrigt mente Peter Ferdinand, at det helt klart for faget datalære er en nødvendighed med samarbejde med andre fag for at få nok af stof, der kan gøre arbejdet med maskinerne meningsfyldt for eleverne.

Efter gennemgangen af årets arbejde fik vi mulighed for at afprøve nogle af de anvendte modeller, og inden vi drog af, fik de, der havde interesse derfor, strimler og arbejds papirer til flere af modellerne - en meget fin service ovenpå en udmærket aften.

BASIC - PASCAL ??? COMAL!

Programmeringssprogenes forskellige fortrin frem for hinanden har altid været et godt diskussions-emne. Bl. a. diskuteres der tit ud fra mangler ved BASIC.

Også i USA går denne diskussion, og i en artikel i Northwest Computer News blev det forsøgt at vise PASCAL's overlegenhed over for BASIC ved nedenstående eksempler:

```
100 IF X = 3 THEN 500      IF X = 3
200 LET Y = Y - 1          THEN BEGIN
300 LET Z = Z - 1          Y := Y + 1;
400 GO TO 700              Z := Z + 1;
500 LET Y = Y + 1          END
600 LET Z = Z + 1          ELSE BEGIN
700 REM                    Y := Y - 1;
                           Z := Z - 1;
                           END;
```

Den samme programstump ville jo nok se sådan ud i COMAL:

```
100 IF X = 3 THEN
200   LET Y = Y + 1 ; Z = Z + 1
300 ELSE
400   LET Y = Y - 1 ; Z = Z - 1
500 ENDIF
```

Man kunne jo også, og det gælder ikke bare for COMAL, men også for flere BASIC versioner, klare det på to linier:

```
100 IF X = 3 THEN LET Y = Y + 1 ; Z = Z + 1
200 IF X <> 3 THEN LET Y = Y - 1 ; Z = Z - 1
```

Og hvad er så bedst?

Man kan da selv vælge. . .

●●●

Fra Køge kommunale skolevæsen, der har været aktive på datalærefronten i flere år, er der nu kommet et nyt skrift:

DATALÆRE 1, Elevhæfte.

Hæftet, der er udarbejdet af Jarl Friis, Kirsten Vej Petersen og Jette Ryberg, er på 191 sider, og det behandler de forskellige, vel snart klassiske, emner som kommunikation, data, problem-løsning, algoritmer, databehandlingens historie osv. Et specielt træk ved hæftet er, at der er medtaget en facitliste til øvelserne.

➔ **OBS! OBS!**

Stof til næste nummer af bladet skal være redaktionen i hænde senest mandag, den 22. oktober 1979

METRIC'S "RULLENDE" UNDERVISNINGSSYSTEM



PÅ VEJ TIL DEM?

Systemet består af:

- 1 Alpha LSI computer med 24 K ord lager
- 3 BEEHIVE dataskærme
- 1 MANNESMANN matrix printer
- 1 TRUE DATA stregmarkeringskortlæser
- 1 Dual flexible disk system. "Floppy disk"
- 1 GNT papirstrimmelæser

Kontakt venligst:

L. Graff-Nielsen, Tlf. 02/80 4200 lok.32

DATAUDSTYR FRA SC **METRIC** AIS

DATAAFDELINGEN, SKODSBORGVEJ 305, 2850 NÆRUM, TLF. (02) 80 42 00

Almindelige oplysninger om foreningen

Bestyrelsens sammensætning:

- Formand: ERLING SCHMIDT
Revlingebakken 40, II, 9000 Ålborg, tlf. (08) 18 53 66.
- Næstformand: WILLY KJELLBERG CHRISTENSEN
Strandpromenaden 32, 4900 Nakskov, tlf. (03) 92 30 34.
- Sekretær: FRITZ G. KNUDSEN
Kollerupvej 17, 8900 Randers, tlf. (06) 43 49 04.
- Kasserer: TORBEN HØIRUP
Karl Withsvej 2, 5000 Odense C, tlf. (09) 14 33 53.
- HUGO JØRGENSEN
Olivenvvej 11, Helsted, 8900 Randers, tlf. (06) 42 37 91.
- GERD BELHAGE
Slettebjergvej 7, 2750 Ballerup, tlf. (02) 97 10 46.
- TORSTEN ALF JENSEN
Langemarken 27, 5762 Vester Skerninge, tlf. (09) 24 22 35.

Henvendelser til foreningen:

Indmeldelser, adresseændringer o.l. til kassereren:

FORENINGEN FOR DATALÆRE OG ANVENDELSE AF EDB I
UNDERVISNINGEN
Rismarksvej 80, 5200 Odense V, tlf. (09) 16 86 50.

eller til privatadressen.

Årskontingent: 90 kr. incl. blad. Studerende 45 kr.

Øvrige henvendelser til formanden.

BLADET:

Ansvarshavende redaktør:

TEDDY LANG PETERSEN
Holstedvej 7, 5200 Odense, tlf. (09) 16 90 56.

Henvendelser vedr. annoncer/stof:

Til redaktøren.