

DATAMÆRE MED MINI DAT

Bjarne Aagaard
Ester M. Christensen
Knud Grosen
Leif Pehrsson

Klassesæt

1981

systeme



Datalære med mikrodatamat

Bjarne Aagaard
Ester M. Christensen
Knud Grosen
Leif Pehrsson

Datalære med mikrodatamat

sysime

**Datalære med
mikrodatamat**

© 1982 Bjarne Aagaard, Ester M. Christensen, Knud Grosen,
Leif Pehrsson og forlaget systime a/s, Herning.

Omslag af Hans Møller
Trykt hos: Werks Offset, Århus

Fotografisk, mekanisk eller anden gengivelse eller mangfoldiggørelse
af denne bog er ikke tilladt ifølge gældende dansk lov om ophavsret.

Printed in Denmark 1982
ISBN 87-7351-027-0



forlaget systime a/s
Klokkebakken 20, Gjellerup
7400 Herning
07-11 90 11

Forord

Denne bog er udarbejdet med henblik på at give begynderen vejledning i

- at arbejde med en mikrodatamat
- at udarbejde programmer i programmeringssproget COMAL

Desuden giver den læseren en orientering om edb's anvendelse i samfundet.

En liste, der viser de små variationer mellem de forskellige udgaver af COMAL, gør bogen lige velegnet til, f.eks.

Regnecentralens mikrodatamat
Ditamat/Comet
Commodore

Alle de programmer, der henvises til i bogen, findes på en diskette til hver af de nævnte maskintyper samt til SPC/1.

Bogen er først og fremmest beregnet til undervisningsbrug, men den er også særdeles velegnet til selvstudium. Der kræves ingen forkundskaber i datalære af den, der benytter bogen, og den er let tilgængelig for alle.

Herning, juni 1982

Forfatterne

INDHOLDSFORTEGNELSE

KAPITEL 1 INDLEDNING	9
KAPITEL 2 MIKRODATAMATEN	11
Tastatur	13
Dataskærm	15
Centralenheden	17
Diskettestation	18
Disketter	19
Båndstation	20
Printer	21
Symboler	22
Opgaver	23
KAPITEL 3 PROGRAMKØRSEL	27
Resume	30
Opgaver	31
Testopgave	32
KAPITEL 4 PROGRAMMER PÅ DISKETTE	33
Eksempel på programkørsel	33
Opgaver	36
KAPITEL 5 ALGORITMER	39
Divisions-algoritmen	40
Algoritme - Instruktioner	41
Beskrivelse af algoritmer	45
Oversigt	46
Opgaver	47
KAPITEL 6 RUTEDIAGRAMMER	49
Forgreninger	56
Oversigt	59
Opgaver	60
Testopgave	63
KAPITEL 7 PROGRAMMERING I	65
Programmeringssproget comal	65
Programeksempel 1	65
Variable	67
Beregninger	68
Programeksempel 2	69
Inddata	70
Beregning	71
Uddata	71
Oversigt	71
Opgaver	72
Oversigt	74
Opgaver	75
KAPITEL 8 PROGRAMMERING 2	79
Ubetingede hop-ordrer	79
Programeksempel	80
Betingede hop-ordrer I	80
Programeksempel	80
Sandhedsværdi	81
Oversigt	82
Opgaver	82
Betingede hop-ordrer II	83
Programeksempel	83
Opgaver	85
Standardfunktioner	88
Oversigt	90
Opgaver	90

KAPITEL 9 PROGRAMMERING 3	93
Løkker	93
For-next-løkker	93
Oversigt	94
Opgaver	96
While-endwhile-løkker	98
Programeksempel 1	98
Programeksempel 2	99
Opgaver	100
Repeat-until-løkker	102
Programeksempel 1	102
Programeksempel 2	102
Opgaver	104
Programmer med større datamængder	107
Programeksempel	107
Oversigt	108
Opgave	109
KAPITEL 10 PROGRAMMERING 4	111
Procedurer	111
Programeksempel X-by varmeværk	111
Opgaver	114
KAPITEL 11 PROGRAMMERING 5	117
Tekstvariable	117
Oversigt	118
Opgaver	118
Testopgave	120
KAPITEL 12 STØRRE DATA-ANLÆG	121
Maskinkonfiguration	121
Opgaver	122
Eksempel på anvendelse af et større dataanlæg	123
Edb-service-bureau	123
Eksempel 1 Tidstro database-handling og datatransmission	124
Eksempel 2 Tidstro database-handling og datatransmission	125
Opgaver	125
KAPITEL 13 EDB OG SAMFUND	127
Mikroprocessor	127
Eksempler på anvendelse af edb	128
Eksempel 1 Styring i landbruget	128
Eksempel 2 Industrirobotten	129
Eksempel 3 Kasseapparater	130
Eksempel 4 Teledata	130
Eksempel 5 Datamater og våben	131
Eksempel 6 Edb i undervisning	132
Eksempel 7 Edb-kriminalitet	133
Opgaver	134
KAPITEL 14 REGISTRE	135
Eksempel på spørgeskema til edb-register	136
CPR-registret	137
Personnummeret	137
Kontrolciffer	138
Opgaver	138
Database	138
Registerloven	139
KAPITEL 15 OPGAVESAMLING	141
Opgaver til kapitel 4	141
Opgaver til kapitel 7	141
Opgaver til kapitel 8	142
Opgaver til kapitel 9	143
Opgaver til kapitel 10	146
APPENDIKS	147
Stikordregister	151

KAPITEL 1 INDLEDNING

Du har sikkert allerede hørt om eller måske endda set en mikrodatamat. Men du har måske ikke prøvet at arbejde med den. I begyndelsen var det kun banker og forsikringsselskabet samt større handels- og industrivirksomheder, der havde mulighed for at anskaffe og udnytte datamaterne.

Med den udvikling, der er sket inden for elektronikken de sidste år, vil det blive mere og mere almindeligt, at hver familie får sin egen datamat. Det vil gå på samme måde, som det er gået med telefonen, radioen og fjernsynet.

Mikrodatamaten vil i den nærmeste fremtid komme til at ændre din dagligdag.

Du behøver blot at tænke på de muligheder, denne maskine giver.

Den vil kunne hjælpe dig med dine hjemmeopgaver. Den kan give dig mange oplysninger, som du ellers skal søge i et leksikon. Den vil kunne vise dig billeder. Den vil kunne lære dig et fremmedsprog. Den kan måske endda uddanne dig i et fag.

Den vil også kunne give dig oplysninger om, hvor mange penge du har i banken. Du vil være i stand til at overføre penge til købmanden, så du ikke selv behøver at gå i banken. Den kan hjælpe dig med selvangivelsen og beregne din skat - eller måske endda vise, hvordan du kan nedsætte din skat.

Får du lyst til et spil skak, vil den gerne spille med dig, men du får nok svært ved at vinde over den.

Disse og sikkert mange andre muligheder kan du udnytte på mikrodatamaten.

Derfor vil vi i denne bog lære dig, hvordan mikrodatamaten er sammensat, hvordan den arbejder, og hvordan du selv kan lave programmer.

KAPITEL 2 MIKRODATAMATEN



Mikrodatamaten RC 702 PICCOLO

Mikrodatamaten er en lille EDB-maskine, der består af

- 1) Tastatur.
- 2) Dataskærm.
- 3) Centralenhed med diskettestation.
- 4) Printer.

Disse fire enheder er koblet sammen, således at de tilsammen udgør en mikrodatamat.

Vi skal på de næste sider se nærmere på de forskellige enheder.

Der findes mange forskellige mikrodatamater. Vi har her vist tre fabrikater:



Mikrodatamaten COMET/DITAMAT



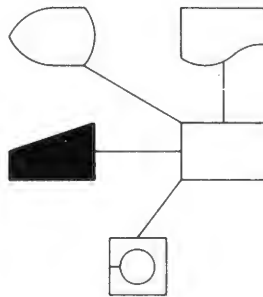
Mikrodatamaten COMMODORE

TASTATUR

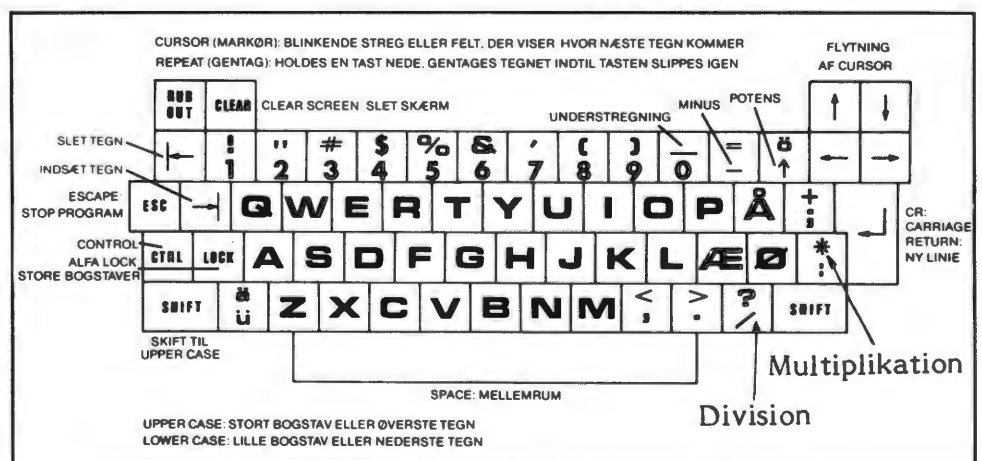
Arbejdspladsen ved en datamat består af tastatur og dataskærm.



Tastatur af en RC 702



Som du kan se af ovenstående diagram, er tastaturet symboliseret ved den sorte figur.



Skitse af tastatur fra RC 702

I appendix A bag i bogen findes et billede af et tastatur til Comet/Ditamat og Commodore.

Tastatur

Tastaturet indeholder stort set de samme bogstaver og tal som en almindelig skrivemaskine - plus nogle specielle tegn.

Læg mærke til, at tastaturet har **et l-tal og et nul**, som er forskellige fra bogstaverne l og o!

Hvis en tast trykkes ned i mere end 0,7 sek., bliver maskinen ved med at skrive tegnet (10 gange i sekundet), indtil tasten slippes.

Specielle taster



RETURN Den store tast i højre side af tastaturet kaldes "RETURN". Den bruges til ny linie.



SHIFT Ved hjælp af tasten "SHIFT" kan man skifte fra små til store bogstaver, og til øverste tegn ved tallene.



LOCK Alfa lock benyttes kun til store bogstaver



CURSOR Cursor frem. (Sletter ikke tegnene)



CURSOR Cursor tilbage. (Sletter ikke tegnene.)



ESCAPE Escape. Afbryder et programforløb.



SPACE Mellemrum mellem tegnene.

Upper & lower case

UPPER CASE: Stort bogstav eller øverste tegn.

LOWER CASE: Lille bogstav eller nederste tegn.

Cursor

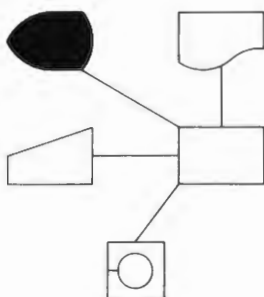
CURSOR: Markør, d.v.s. blinkende streg eller felt, der viser, hvor næste tegn kommer på skærmen.

Tastaturerne på de forskellige mikrodataloger er ikke helt ens; men ovenstående taster findes på dem alle.

DATASKÆRM



Dataskærm RC 702



På ovenstående diagram kan du se dataskærmen symboliseret ved den sorte figur.

Dataskærm

Dataskærmen er opbygget som et lille TV apparat. Betjeningsknapperne har samme funktion som på et almindeligt TV:

ON - OFF: Tændt-slukket (rød lampe lyser).

BRIGHT: Lysstyrke.

CONT: Kontrast.

Som regel er der på en dataskærm plads til 80 tegn i 24 rækker. Når du trykker på en af tasterne på tastaturet, kan du straks se resultatet på skærmen.

Øvelse A

Undersøg et tastatur, og besvar følgende spørgsmål:

Hvordan foregår lineskift? _____

Hvilket tegn udskrives, når man trykker på SHIFT og 8? _____

Hvilket tegn udskrives, når man bruger LOCK-tasten og trykker på 8?

Hvilket tegn bruger man til addition _____

subtraktion _____

multiplikation _____

division _____

potensopløftning _____

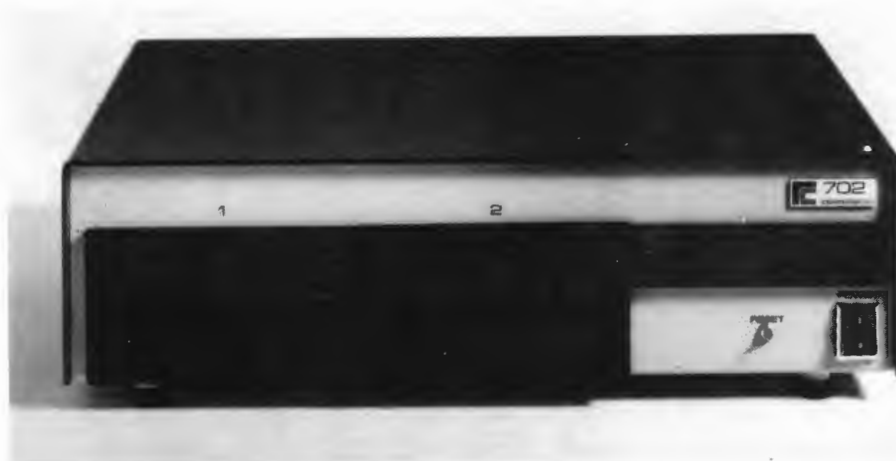
Øvelse B

Hvordan vil du indtaste 34 kroner og 25 øre som decimaltal, når du skal anvende samme princip som ved regnemaskinen?

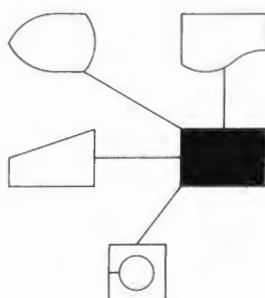
Følgende fotografi viser en datamat, hvor dataskærm, central enhed og tastatur er sammenbygget.



CENTRALENHEDEN



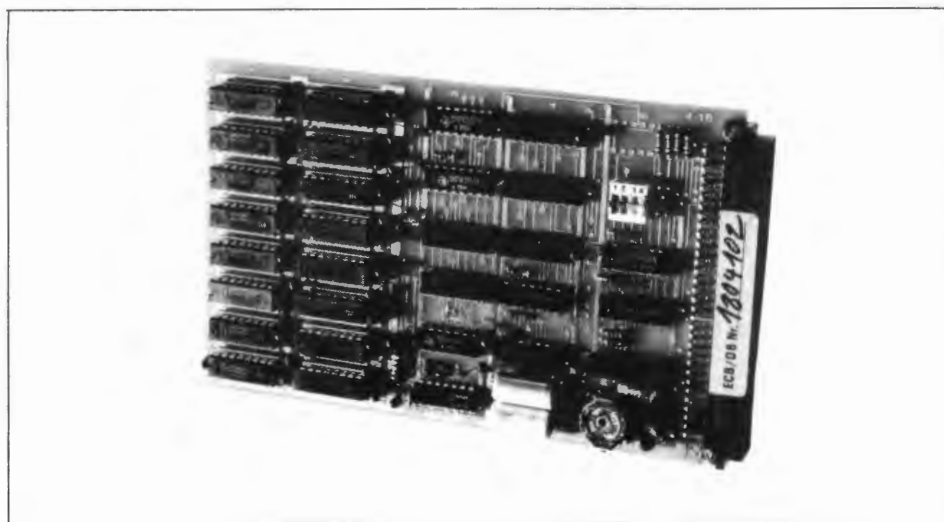
Centralenheden med indbygget diskettestation på en RC 702



På diagrammet kan du se centralenheden symboliseret ved den sorte figur.

Centralenhed

Centralenheden er datamatsens kerne. Den ligger gemt i en kasse. Ofte er den bygget sammen med diskettestationen eller en anden af datamatsens dele. Den består af printplader med en mængde integrerede kredse.



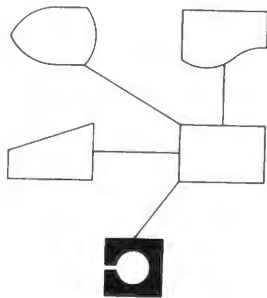
Printplade

DISKETTESTATION



Enkeltstående diskettestation uden centralenhed

Diskettestation



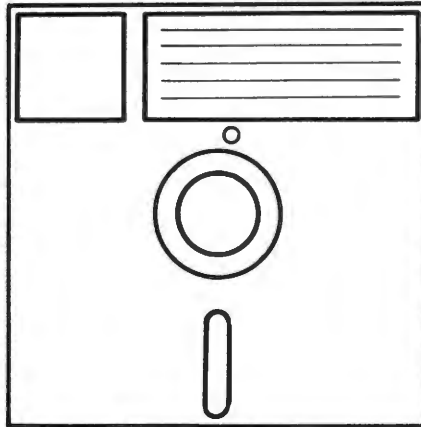
I diagrammet er diskettestationen symboliseret ved den sorte figur.

Diskettestationen er ofte bygget sammen med centralenheden. I enkelte tilfælde kan den dog fremstå som en enhed for sig selv. Dette gælder især, hvor det drejer sig om en diskettestation, der er beregnet til disketter på 8". (Se næste side). Normalt anvendes nu disketter på 5", der ikke fylder så meget. Diskettestationen virker som en udvidelse af datamaskinens egen "hukommelse".



Nok kan datamaten klare store og indviklede problemer, men der er dog en vis begrænsning.

DISKETTER



Disketter

Disketten er en tynd, bøjelig plastskive med en magnetiserbar belægning. Den ligner en grammofonplade.

Fra tastaturet kan man overføre data gennem centralenheden til disketten. Overførsel fra centralenhed til diskette sker med meget stor hastighed, ca. 30.000 tegn pr. sekund.

Disketter til mikrodatabasen har plads til 240.000 - 400.000 tegn, alt efter om det drejer sig om en 5" diskette eller en 8" diskette.

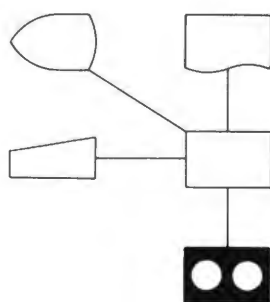


Billedet viser tre disketter, der indeholder hvert sit programmeringssprog.

BÅNDSTATION



Dette billede viser en mikrodatamat med en båndstation. Båndstationen ses til venstre.



På diagrammet kan du se båndstationen symboliseret ved den sorte figur.

Båndstation

En båndstation er normalt en kassettebåndoptager med et ganske almindeligt kassettebånd.

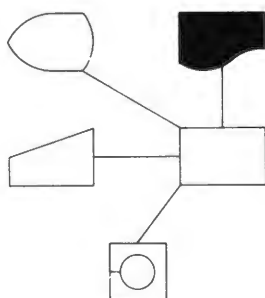
Dette kan anvendes som "medium" til lagring af såvel data som programmer. Men det er ikke særlig praktisk til formålet, fordi overførselen fra bånd til centralenhed sker langsomt. Det kan også være ret besværligt at finde frem til det sted på båndet, hvor programmet står.

Fordelen ved kassettebåndet er, at det er noget billigere end disketten.

PRINTER



Mikroline



Den sorte figur i diagrammet viser printerens plads i edb-anlægget.

Printer

Printeren er en elektrisk skrivemaskine. Den er koblet til centralenheden, så der kan udskrives på papir i stedet for på skærm.

Printeren har foruden de almindelige bogstaver og tegn mulighed for at lave grafik, så den både kan skrive og tegne.

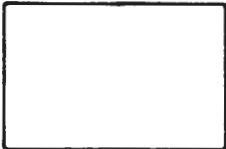
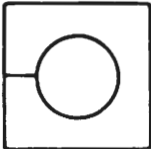
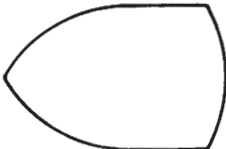
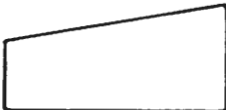
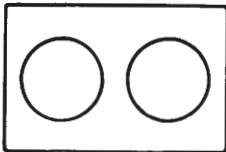
Printeren er meget hurtig. Den kan udskrive 30-200 tegn pr. sekund. Til sammenligning kan en rutineret kontordame skrive 3-4 tegn pr. sekund.

SYMBOLER

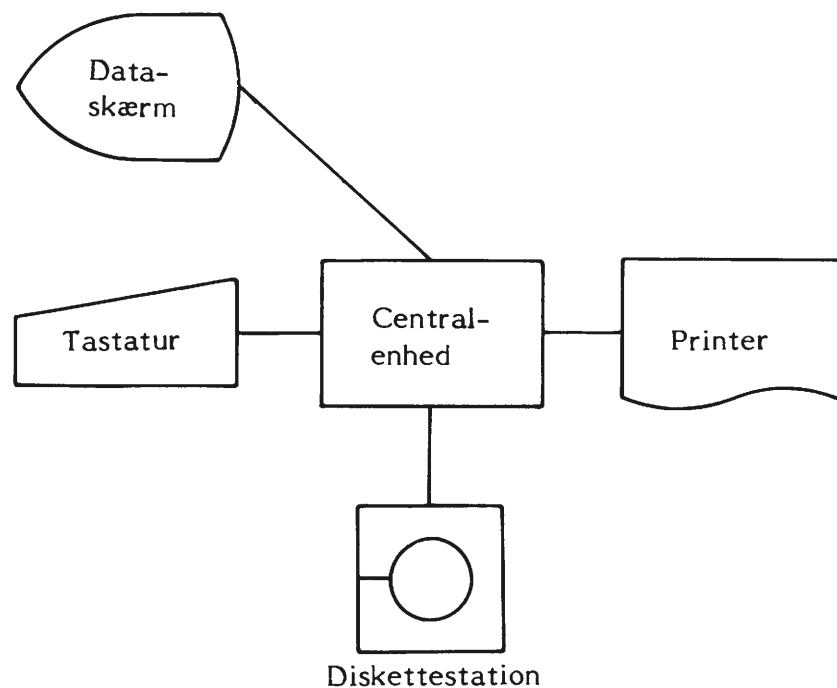
For at gøre det nemmere at arbejde med EDB anvendes bestemte symboler for datamaterns forskellige enheder.

Vi vil i begyndelsen holde os til de symboler, som hører til vores mikrodatamat.

Symboler

Symbol	Symbolnavn
	Centralenhed
	Diskettestation (oftest bygget sammen med centralenheden).
	Dataskærm
	Tastatur
	Printer
	Kassettebåndstation

En datamat som den, du skal lære at bruge, kan med symboler vises således:

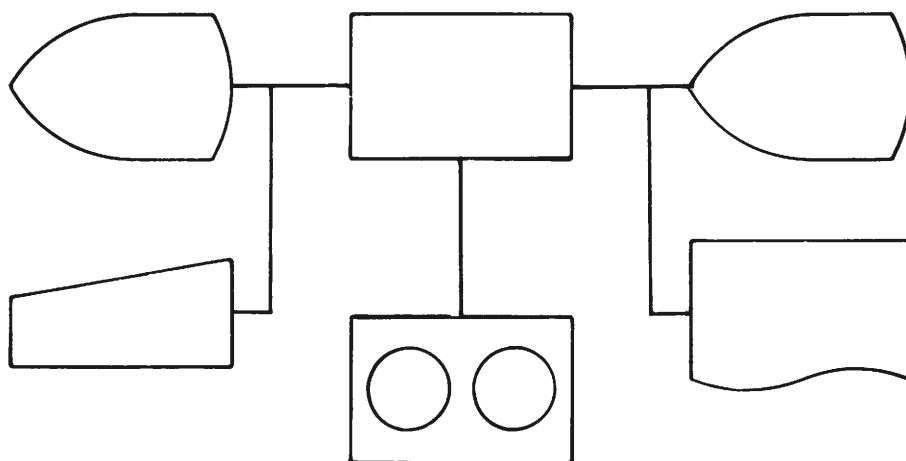


Dette kaldes en MASKINKONFIGURATION. Tastatur, skærm og evt. diskettestation kan være sammenbygget.

OPGAVER

Opgave 2.1

Skriv inde i symbolerne navnene på de enheder, som denne mikrokrodatamat består af.



Opgave 2.2

Kig på billedet og tegn dataanlægget med symboler.



Opgave 2.3

Tegn en model af et dataanlæg med 1 centralenhed, 2 skærme og 1 printer.

Opgave 2.4

Tegn en model af jeres eget anlæg.

Opgave 2.5

Find nogle brochurer, klip ud og lav en planche, som viser et anvendeligt EDB-anlæg.

KAPITEL 3 PROGRAMKØRSEL

Når du har undersøgt, hvad de forskellige betjeningsknapper og taster benyttes til, skal du prøve at skrive på datamaten. Du skal indtaste nogle programmer. Hvordan disse programmer er sammensat, vil blive gennemgået i kap. 7-11. Du skal blot skrive nøjagtigt af. Vær omhyggelig med indtastningen, da selv en lille fejl er nok til, at programmet ikke virker.

NEW

Inden du begynder at indtaste et nyt program, skal du indtaste ordet NEW. Herved opnås, at et allerede indtastet program bliver slettet, således at det gamle program ikke bliver blandet med det nye program.

NEW

Systemkommando, som sletter et allerede indtastet program.

En SYSTEMKOMMANDO er en besked til datamaten om at udføre en handling.

Øvelse A

Indtast NEW
Tryk RETURN

Du skal nu prøve, om du kan få maskinen til at opfatte, hvad du hedder.

Skriv dit navn og tryk RETURN

Hvad sker? _____

Det ser ud til, at maskinen skal have din besked på en bestemt måde.

Nu får du lidt hjælp. Skriv således:

0010 PRINT "JEG HEDDER OLE"

Tryk RETURN
Skriv RUN
Tryk RETURN

Hvis maskinen får beskeden: PRINT (SKRIV), forstår den det åbenbart.

RUN

Du skal altså henvende dig til maskinen i et særligt sprog. Dette sprog indeholder en række ordrer og systemkommandoer. PRINT er en ordre. NEW og RUN er systemkommandoer.

For at få programmet til at virke skal du bruge systemkommandoen RUN.

RUN	Systemkommando, som starter udførelsen af et program.
------------	---

Øvelse B

Indtast systemkommandoen NEW.

Indtast følgende program. Husk at trykke RETURN ved linjeskift.

```
0010 PRINT "          VELKOMMEN TIL          "  
0020 PRINT "          DATALÆRE          "  
0030 PRINT "X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X"
```

Kontroller, at programmet er skrevet rigtigt af.

Indtast derefter RUN og tryk på RETURN.

Hvis du ikke har lavet fejl, fremkommer følgende billede på skærmen:

```
          VELKOMMEN TIL  
          DATALÆRE  
X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X
```

Øvelse C

Indtast disse 4 linjer. Tryk RETURN efter hver linje.

```
0010 PRINT "NU KOM DEN NÆSEKOLDE TID,  "  
0040 PRINT "LUN LUFT PÅ FROSTBLÅ NÆSER. "  
0030 PRINT "I SMUG GÅR RUNDT OG BLÆSER  "  
0020 PRINT "HVOR ALLE, SOM HAR UNDERBID,"
```

Skriv LIST

Tryk RETURN

Hvilken linie kommer sidst? _____

LIST

LIST	Systemkommando, som giver en udskrift af programmet.
-------------	--

Læg mærke til, at de 4 linjer udskrives i NUMMERORDEN.

Øvelse D

Du skal indtaste følgende 3 linjer. Husk at skrive NEW først.

```
0010 PRINT "DET VILLE VÆRE RART AT HAVE EN DATAMAT TIL "  
0020 PRINT "LÆREREN OG TIL ALLE OG EN- "  
0030 PRINT "HVER AF ELEVERNE I KLASSEN. "
```

Skriv derefter RUN

Tryk RETURN

Hvad udskrives?

Skriv derefter

DEL 20

Tryk RETURN

Skriv RUN

Tryk RETURN

Hvilken udskrift fås nu?

DEL

DEL

Systemkommando, der sletter en linje.
Det sker, når man skriver DEL linjenum-
mer og derefter trykker på RETURN.

Øvelse E

Indtast følgende linjer:

```
0010 PRINT "DET VAR EN LØRDAG AFTEN, "  
0020 PRINT "JEG SAD OG VENTED DIG. "  
0030 PRINT "DU GAV OS DE BLØMSTER, "  
0040 PRINT "SOM LYSTEN IMOD OS "  
0050 PRINT "MED ALFELET SKØNHED, "  
0060 PRINT "DENGANG VI VAR SMÅ. "  
0070 PRINT "DU LOVED MIG AT KOMME VIST, "  
0080 PRINT "MEN DU KOM EJ TIL MIG. "
```

Hvor mange linjer har programmet?_____

Skriv derefter DEL 30, 60

Tryk RETURN

Skriv LIST og tryk igen på RETURN

Hvor mange linjer har programmet nu?_____

Man kan slette programafsnit ved at indtaste DEL efterfulgt af første og sidste linienummer (adskilt af komma) i det afsnit, man ønsker at udelade, og derefter trykke RETURN.

Øvelse F

Formålet med det næste program er at lave en navneseddel til databogen.

Indtast følgende program, idet du indsætter dit navn og din adresse i den åbne rude.

```
0010 PRINT "*****"
0020 PRINT "***:***"
0030 PRINT "***:***"
0040 PRINT "***:***"
0050 PRINT "***:***"
0060 PRINT "***:***"
0070 PRINT "*****"
```

Undersøg, om der forekommer fejl i teksten, og om dit navn og din adresse er placeret midt i ruden.

Brug cursoren, hvis der er noget, der skal rettes.

Indtast RUN

Tryk RETURN

Skærbilledet viser den færdige navneseddel.

SELECT OUTPUT Med systemkommandoen SELECT OUTPUT "printer" kan man give datamaten besked om at udskrive navnesedlen på printeren.

Skriv SELECT OUTPUT "printer"

Tryk RETURN

Skriv RUN

Tryk RETURN

Når man i programmet skal tilbage til skærmen, skrives SELECT OUTPUT "console".

**SELECT
OUTPUT**

Systemkommando, som dirigerer udskriften til printer eller skærm.

RESUME

Du har nu lært at anvende anlægget. Husk, at hvis maskinen skal opfatte det, der skrives, skal man afslutte med at trykke RETURN.

Vi har lært nogle systemkommandoer. Dem vil vi repetere.

NEW

NEW

Systemkommando, som sletter et allerede indtastet program.

RUN

RUN

Systemkommando, som starter udførelsen af et program.

LIST

LIST

Systemkommando, som giver en udskrift af programmet.

DEL

DEL

Systemkommando, som sletter en programlinie eller et programafsnit.

SELECT OUTPUT

**SELECT
OUTPUT**

Systemkommando, som dirigerer udskriften til en printer eller en skærm.

OPGAVER

Opgave 3.1

Du skal nu selv prøve at indføre en tekst i programmet (f.eks. åbningstider, vil ikke forstyrres o.lign.).

```
0010 PRINT "*****"
0020 PRINT "*****"
0030 PRINT "*****"
0040 PRINT "*****"
0050 PRINT "*****"
0060 PRINT "*****"
0070 PRINT "*****"
```

Skriv først programmet på de åbne linier her i bogen.

Når du har noteret din tekst, udgør linierne et program.

Indtast programmet.

Skriv RUN

Tryk RETURN

Hvilken udskrift fremkommer på dataskærmen:

Hvis din tekst skal benyttes til noget, kan du få den udskrevet på printeren ved at bruge systemkommandoen SELECT OUTPUT. Se foregående øvelse.

Opgave 3.2

Lav et program, som giver følgende udskrift:

EEEEEEEEEE	DDDDDD	BBBBBBB	
EEEEEEEEEE	DDDDDDDD	BBBBBBBBBB	
EEEEEEEEEE	DDDDDDDD	BBBBBBBBBBBB	E
EEEEEEEEEE	DDDDDDDD	BBBBBBBBBBBB	
EEEE	DDDD DDDDD	BBBB BBBB	R
EEEE	DDDD DDDDD	BBBB BBBB	
EEEE	DDDD DDDDD	BBBB BBBB	
EEEE	DDDD DDDDD	BBBB BBBB	
EEEEEE	DDDD DDDDD	BBBBBBBBBB	S
EEEEEE	DDDD DDDDD	BBBBBBBBBB	
EEEEEE	DDDD DDDDD	BBBBBBBBBB	J
EEEEEE	DDDD DDDDD	BBBBBBBBBB	
EEEE	DDDD DDDDD	BBBB BBBB	D
EEEE	DDDD DDDDD	BBBB BBBB	
EEEE	DDDD DDDDD	BBBB BBBB	V
EEEE	DDDD DDDDD	BBBB BBBB	
EEEEEEEEEE	DDDDDDDD	BBBBBBBBBBBB	T
EEEEEEEEEE	DDDDDDDD	BBBBBBBBBBBB	
EEEEEEEEEE	DDDDDD	BBBBBBBBBB	..
EEEEEEEEEE	DDDDDD	BBBBBBBBBB	..

Opgave 3.3

Testopgave.

Programmet "TEST1" ligger på en diskette.

Sæt disketten i diskettestationen.

Skriv LOAD "TEST1"

Tryk RETURN

Programmet læses nu ind i centralenheden.

Vent, indtil datamaten er klar igen.

Skriv RUN, og husk RETURN.

Du får nu en række spørgsmål. Til hvert spørgsmål er der tre muligheder for svar. Indtast det svar, som du mener er rigtigt.

KAPITEL 4 PROGRAMMER PÅ DISKETTE

I dette afsnit vil vi arbejde med programmer, der ligger lagrede på disketter.

Vi har allerede lært:

Ordren **PRINT**

Systemkommandoerne **NEW, RUN, LIST, DEL og SELECT OUTPUT**

Nu skal vi beskæftige os med to nye systemkommandoer: **LOAD og EDIT.**

Et program, der ligger lagret på en diskette, har et navn, som benyttes, når programmet skal findes igen.

LOAD

Når disketten er anbragt i diskettestationen, kan programmet hentes ind i centralenheden ved hjælp af **LOAD** efterfulgt af programmets navn i anførselstegn.

LOAD

Systemkommando, som overfører et program fra diskette til centralenhed.

EDIT

Hvis man skal rette en eller flere linier i et program, som allerede er indtastet, kan man ved hjælp af **EDIT** få de pågældende linier frem på skærmen.

EDIT

Systemkommando, der benyttes til at rette i en eller flere allerede indtastede programlinier.

Eksempel på programkørsel

Øvelse A

Indsæt disketten med programmer.
Indtast på tastaturet: **LOAD "HANS"**
Tryk **RETURN**

Programmet med programnavnet **HANS** hentes nu fra disketten

og lagres i centralenhedens arbejdslager.
Indtast LIST
Tryk RETURN

Følgende udskrift af programmet fås:

```
0010 PRINT "KÆRE HANS"
0020 PRINT
0030 PRINT "JEG ER NU BEGYNDT PÅ ET EDB-KURSUS, SOM JEG FINDER INTERESSANT."
0040 PRINT "JEG HAR NU LÆRT AT HENTE ET PROGRAM FRA DISKETTEN OG LÆGGE DET "
0050 PRINT "OVER I CENTRALENHEDENS ARBEJDSLAGER."
0060 PRINT "JEG SKAL FORTÆLLE DIG MERE OM EDB NÆSTE GANG, VI MØDES."
0070 PRINT
0080 PRINT "
0090 PRINT "
MED VENLIG HILSEN
PETER.
```

I linierne 20 og 70 står kun ordet PRINT, hvilket bevirker linieskift. (Mellemrum mellem linierne).

Skriv RUN
Tryk RETURN

Du får så følgende udskrift:

```
KÆRE HANS
JEG ER NU BEGYNDT PÅ ET EDB-KURSUS, SOM JEG FINDER INTERESSANT.
JEG HAR NU LÆRT AT HENTE ET PROGRAM FRA DISKETTEN OG LÆGGE DET
OVER I CENTRALENHEDENS ARBEJDSLAGER.
JEG SKAL FORTÆLLE DIG MERE OM EDB NÆSTE GANG, VI MØDES.
MED VENLIG HILSEN
PETER.
```

Derefter skal du skrive EDIT 10 og trykke på RETURN.
Linie 10 står nu på skærmen med cursor (markør) på P's plads i PRINT, således at vi straks kan rette linien.

Vi indtaster dernæst:

```
0010 PRINT "KÆRE SUSANNE" (eller dit eget navn)
```

og RETURN

Systemet udskriver nu samme brev som før, men med overskriften KÆRE - det indtastede navn - i stedet for KÆRE HANS.

Øvelse B

Du skal skrive LOAD "GRØNNE" og trykke på RETURN.
Skriv derefter LIST, således at du får en udskrift af det program, som har navnet GRØNNE.

Programmet kommer ud på skærmen i samme øjeblik, som du har trykket på RETURN-tasten.

Hvor mange linier indeholder programmet?_____

Indtast dernæst

DEL 20	RETURN
DEL 40	RETURN
DEL 60	RETURN
DEL 80	RETURN

Skriv RUN
Tryk RETURN
Hvilke 4 linier udskrives?

Øvelse C

Vi vil nu prøve at tage et aftryk af det samme program som før ved at skrive LOAD "GRØNNE". Du skal derefter skrive LIST og trykke på RETURN.

Du får følgende udskrift:

```
0010 PRINT "JEG ELSKER DE GRØNNE LUNDE      "  
0020 PRINT "I ALLE DE RIGER OG LANDE,      "  
0030 PRINT "MED TONERNES VUGGENE FALD.      "  
0040 PRINT "HVORHEN JEG I VERDEN FOR.       "  
0050 PRINT "JEG ELSKER DE BLANKE SUNDE,     "  
0060 PRINT "JEG FÆGTED MED ÅBEN PÅNDE,     "  
0070 PRINT "MED SEJLERNES TUSINDTAL.       "  
0080 PRINT "OG PÅ, HVAD JEG FOR ALVØR TRØR."
```

Du skal slette fire af program-linierne for at få det andet vers udskrevet. Hvordan lyder det andet vers?

Øvelse D

Du skal skrive LOAD "VISIT" og trykke på RETURN.
Tag derefter et udskrift af programmet ved at skrive LIST.

I hvilken linie står HANS HANSEN? _____

I hvilken linie står STOREGADE 29? _____

I hvilken linie står 8310 TRANBJERG J.? _____

Skriv derefter RUN.

Du får nu følgende udskrift:

```
*****  
*****  
***** HANS HANSEN *****  
***** STOREGADE 29, *****  
***** 8310 TRANBJERG J. *****  
*****  
*****
```

Du skal derefter skrive

EDIT 30, 50

og trykke på RETURN.

Linie 30 står på skærmen, klar til at blive rettet. Du skal nu i linie 30 skrive dit eget navn i stedet for HANS HANSEN.

Når du trykker på RETURN, står linie 40 på skærmen, klar til at blive rettet. Du skal atter trykke på RETURN-tasten, og du har nu linie 50 på skærmen, klar til at blive rettet. I stedet for 8310 TRANBJERG J. skal du indtaste dit eget postnummer og din by.

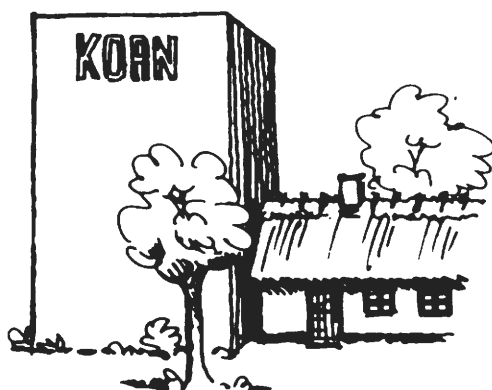
Når du er færdig, skal du tage en udskrift ved at skrive RUN.

Hvis du er tilfreds med resultatet, skriver du **SELECT OUTPUT "printer"**, trykker på RETURN, skriver RUN og trykker på RETURN igen.

Dit visitkort bliver nu udskrevet på printerens.

OPGAVER

Opgave 4.1



Ovenstående kornsilo har en grundflade på 15x15 m og en højde på 18 m.

Vi vil nu finde rumfanget af siloen ved at benytte et rumfangsprogram, som vi i forvejen har lagt på disketten.

Du skal skrive **LOAD "RUMFANG"** og derefter følge brugsanvisningen, når du har indtastet RUN.

Siloen har form som en kasse, og vi finder dens rumfang ved at indtaste **LÆNGDE**, **BREDDE** og **HØJDE**.

Når du er færdig, skal du besvare følgende spørgsmål:

Hvor mange m^3 korn indeholder siloen, når den er helt fyldt op?

Derefter skal du ved at fortsætte programmet finde de manglende måltal for rumfang i skemaet nedenfor.

(Husk ved indtastningen, at LÆNGDE, BREDDE og HØJDE skal have samme benævnelse).

LÆNGDE	BREDDE	HØJDE	RUMFANG
22,0 cm	33,0 cm	12,0 cm	cm ³
12,5 dm	3,7 dm	2,1 dm	dm ³
20,0 m	10,0 m	3,0 m	m ³

Opgave 4.2

Iflg. KNUD LUNDBERG's bog "Slankhedens ABZ" skal mænd og kvinder af middel bygning have en vis idealvægt, der er afhængig af højden.

Skriv: LOAD "IDEAL" og besvar følgende spørgsmål.

Hvis du indtaster højden 172 (underforstået cm), hvor meget skal man så veje, hvis man

1) er en mand? _____

2) er en kvinde? _____

(Hvis du tager en LIST af programmet, vil du opdage, at der er en del ordrer, som du endnu ikke har lært. Dem vil du lære, når du senere skal til den egentlige programmering).

Følg programmet nøje og udfyld de 2 nedenstående skemaer:

Skema over mænds idealvægt i relation til højden

156 cm . . . __kg	168 cm . . . __kg	180 cm . . . __kg
158 cm . . . __kg	170 cm . . . __kg	182 cm . . . __kg
160 cm . . . __kg	172 cm . . . __kg	184 cm . . . __kg
162 cm . . . __kg	174 cm . . . __kg	186 cm . . . __kg
164 cm . . . __kg	176 cm . . . __kg	188 cm . . . __kg
166 cm . . . __kg	178 cm . . . __kg	190 cm . . . __kg

Skema over kvinders idealvægt i relation til højden

156 cm . . . __kg	168 cm . . . __kg	180 cm . . . __kg
158 cm . . . __kg	170 cm . . . __kg	182 cm . . . __kg
160 cm . . . __kg	172 cm . . . __kg	184 cm . . . __kg
162 cm . . . __kg	174 cm . . . __kg	186 cm . . . __kg
164 cm . . . __kg	176 cm . . . __kg	188 cm . . . __kg
166 cm . . . __kg	178 cm . . . __kg	190 cm . . . __kg



KAPITEL 5 ALGORITMER

Man får ofte det indtryk, at alle problemer kan løses ved hjælp af EDB. Det er ikke rigtigt.

Problemer som:

Hvem vinder i lotteriet?
Bliver det regnvejr i morgen?
Vil der ske en ulykke med mig?



- kan ikke løses af en datamat, da de ligger uden for det område, som mennesker har indflydelse på.

Når man skal løse et problem ved hjælp af en datamat, har man brug for en plan, som nøjagtigt viser, hvordan man når frem til løsningen. En sådan plan kaldes en ALGORITME.

ALGORITME

En plan, der viser, hvordan et problem løses.

Øvelse A

Du kan få et indtryk af, hvad en algoritme er, ved at løse en opgave. Du skal blot følge den viste plan og kun gøre det, der står.

Løs opgaven og indsæt værdierne.

- | | | | |
|---------------------------|---|---|-----|
| 1. opgave | $5(x + y)$ | $+ 3(x - y)$ | $=$ |
| 2. gang ind i parenteser | $(\underline{\quad} + \underline{\quad})$ | $+ (\underline{\quad} - \underline{\quad})$ | $=$ |
| 3. hæv parenteserne | $\underline{\quad} + \underline{\quad}$ | $+ \underline{\quad} - \underline{\quad}$ | $=$ |
| 4. reducer | $\underline{\quad}$ | $+ \underline{\quad}$ | $=$ |
| 5. indsæt: $x=2$ og $y=4$ | $\underline{\quad}$ | $+ \underline{\quad}$ | $=$ |
| 6. gang tallene | $\underline{\quad}$ | $+ \underline{\quad}$ | $=$ |
| 7. læg sammen | $\underline{\quad}$ | $+ \underline{\quad}$ | $=$ |
| 8. sum | | $\underline{\quad}$ | |

Øvelsen viser, hvordan en datamat arbejder.

Linie 1 angiver starttilstanden (opgaven). Linie 2 - 3 - 4 - 5 - 6 - 7 angiver de trin og den rækkefølge, man skal benytte. Linie 8 angiver slutttilstanden (resultatet).

I kan se, at øvelsen er opdelt i trin og løst i en bestemt rækkefølge. Sådanne problemer er velegnede til at blive behandlet af en datamat.

Man beskriver en algoritme ved nøje at angive:

1. Starttilstanden
2. Hvert trin i opgaveløsningen
3. Rækkefølgen, de enkelte trin skal løses i
4. Slutttilstanden

Divisionsalgoritmen

Fra matematik kender du sikkert en hel del algoritmer. For eksempel benyttes en særlig metode til at dividere med tocifrede tal. Denne metode er en algoritme.

Divisionsalgoritmen

①	1 2 5 6 : 4 8 = 2	Der bliver $125 : 48$ tiere. Skøn: $120 : 50 = 12 : 5 \approx 2$
②	1 2 5 6 : 4 8 = 2 9 6 2 9 6	Vi »ganger tilbage« og finder resten i tiere: $2 \cdot 48 = 96$; $125 - 96 = 29$. Når 6-tallet »trækkes ned«, har vi netop den samlede rest: 296
③	1 2 5 6 : 4 8 = 2 6 9 6 2 9 6	Divisionen må så få $296 : 48$ enere. Skøn: $300 : 50 = 30 : 5 \approx 6$
④	1 2 5 6 : 4 8 = 2 6 9 6 2 9 6 2 8 8 8	Vi »ganger tilbage« og finder resten i enerne: $6 \cdot 48 = 288$; $296 - 288 = 8$
⑤	1 2 5 6 : 4 8 = 2 6 $\frac{8}{48}$ 9 6 2 9 6 2 8 8 8 2 6 $\frac{1}{6}$	Vi får altså 8 enere til rest, som også skal divideres med 48 $\frac{8}{48}$ kan forkortes med 8. Det blandede tal $26\frac{1}{6}$ er <i>nøjagtigt</i> facit.

Kilde: Carl Johansen

Øvelse B

Løs følgende divisionsopgave:

$$\begin{array}{r} 1479 : 12 = \underline{\hspace{2cm}} \\ - \hspace{1.5cm} \underline{\hspace{2cm}} \\ - \hspace{1.5cm} \underline{\hspace{2cm}} \\ - \hspace{1.5cm} \underline{\hspace{2cm}} \end{array}$$

Øvelse C

Du skal gange $4\frac{1}{2}$ med 3. Skriv den algoritme, der bruges. Der er tre muligheder.

Metode A	Metode B	Metode C

Betingelsen for, at en plan kan betragtes som en algoritme, er:

- at man kommer til samme resultat, hver gang algoritmen følges,
- at resultatet bliver det samme, lige meget hvem der udfører algoritmen.

ALGORITMER - INSTRUKTIONER

Det er ikke kun inden for matematikken, man benytter algoritmer. Brugsanvisninger og opskrifter til madlavning, håndarbejde, hobbyarbejde o.s.v. er også algoritmer.

Fremgangsmåden, som man skal følge, er beskrevet trin for trin i den rigtige rækkefølge. En god instruktion må ikke kunne misforstås og skal altid føre til det ønskede resultat. En god instruktion er altså en algoritme.

Øvelse D

Vejledningen i at ringe fra en mønttelefon er en velkendt algoritme.

Gennemgå algoritmen og diskuter, om I kommer til det ønskede resultat.

1. Løft røret og vent, til klartonen lyder.
2. Indkast 2 x 25 øre.
3. Drej det ønskede nummer.
4. Vent, til ringetonen har lydt.

Øvelse E

En strikkeopskrift kan godt opfattes som en algoritme. For at følge den kræves det, at du forstår "sproget". Her ser du en strikkeopskrift til vanter. Prøv at følge den.

Vanter: slå 30 (32) 34 (34) m op med enkelt tråd på p nr. 3½ og strik retst. Efter 5 (6) 7 (7) riller skiftes til 2 tråde garn, og der tages 8 m ind jævnt fordelt over første p. Skift til p nr. 5. Strik 5 (6) 7 (7) cm lige op i retst. **Næste p:** tag 4 m ind jævnt fordelt over p. 1 p uden indt. **Næste p:** tag 4 m ind jævnt fordelt. Strik 1 p uden indt, og træk enden af garnet igennem m og hæft; sy vanten sm langs siden med små kasting fra vrangen. Strik en vante magen til. Hækl et bånd af 1m ca. 25 cm langt til hver vante. Træk båndet igennem vanten lige over første p med dobbelt garn, bind en sløjfe i ydersiden af vanterne.

Øvelse F

Gennemgå følgende fire eksempler og afgør, i hvilke tilfælde der er tale om en algoritme.

Eksempel 1

Gluemaster



Gluemaster flisepasta er en »klar til brug« pasta på elastisk akrylbasis til indendørs opsætning og lægning af keramiske fliser.

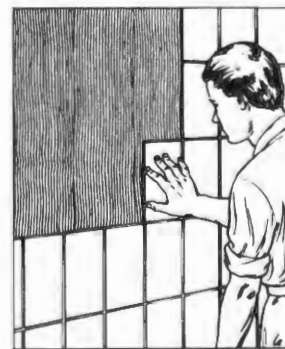
EVERS



Underlaget renses og klargøres



Pasta påføres med tandspartel



Fliser opsættes



Udfugning ved hjælp af gummispartel



Rengøring med svamp



Elastisk fuger i hjørner og omkring gulv

Eksempel 2

Broget hverdagsgryde



- 2 dl finhakket løg
- 2 fed finhakket hvidløg
- 3 spsk olie
- 400-500 g hakket oksekød
- 3 dl langkornede ris
- 1-2 tsk paprika
- 1/2 tsk allehånde
- evt. 1/4 tsk cayennepeber
- 7-8 dl bouillon
- 1 pose dybfrosne majs, evt. med peberfrugt
- 1 pose dybfrosne, fine ærter
- 1 dåse henkogte tomater (1/2 kg ds)
- salt, peber

PYNT: 1 peberfrugt.

Svits de finhakkede løg og hvidløg i olie, indtil løgene er klare, men kun let gyldne. Tilsæt det hakkede kød og rør, til kødet har delt sig.

Tilsæt ris og krydderier og bland det hele godt. Hæld bouillon og væden* fra tomatene ved retten og dæk gryden med låg. Lad retten småsnurre under låg 20-30 min. Spæd evt. mere væske til. Tilsæt majs, ærter og ituskårne dåsetomater. Kog retten videre ca. 10 min. uden låg. Smag retten til.

SERVERING: Retten serveres i gryden. Tynde ringe af peberfrugt lægges over.

VARIATION AF RETTEN:

NINAS MILLIONBØF: 10 skalotteløg halveres og svitses i 3 spsk olie. 1/2 kg hakket flæskeskød tilsættes, og der røres, til kødet har delt sig. 1 dåse henkogte tomater med væde, 1 pakke dybfrosne bønner og lidt vand tilsættes. Retten krydres med salt, peber og lidt salvie og småkoger under låg 20-30 min. Serveres med løse ris.

Eksempel 3

Ved indstilling af sekunderne rykker uret 1 minut frem, hvis sekunderne er mellem '30' og '59'.

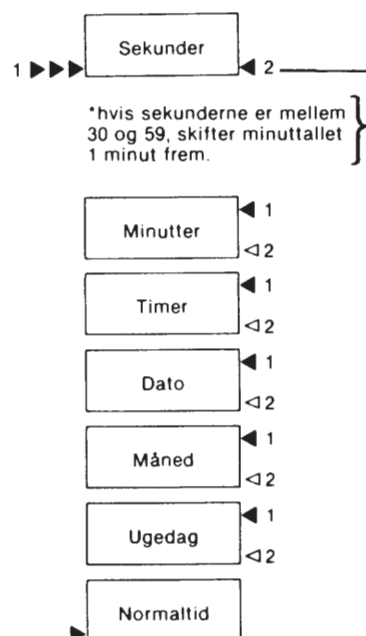
Ved visning af måned og dato forsvinder disse tal af sig selv efter et par sekunder.

A: tiden mellem kl. 24.00 og 12.00
P: tiden mellem kl. 12.00 og 24.00

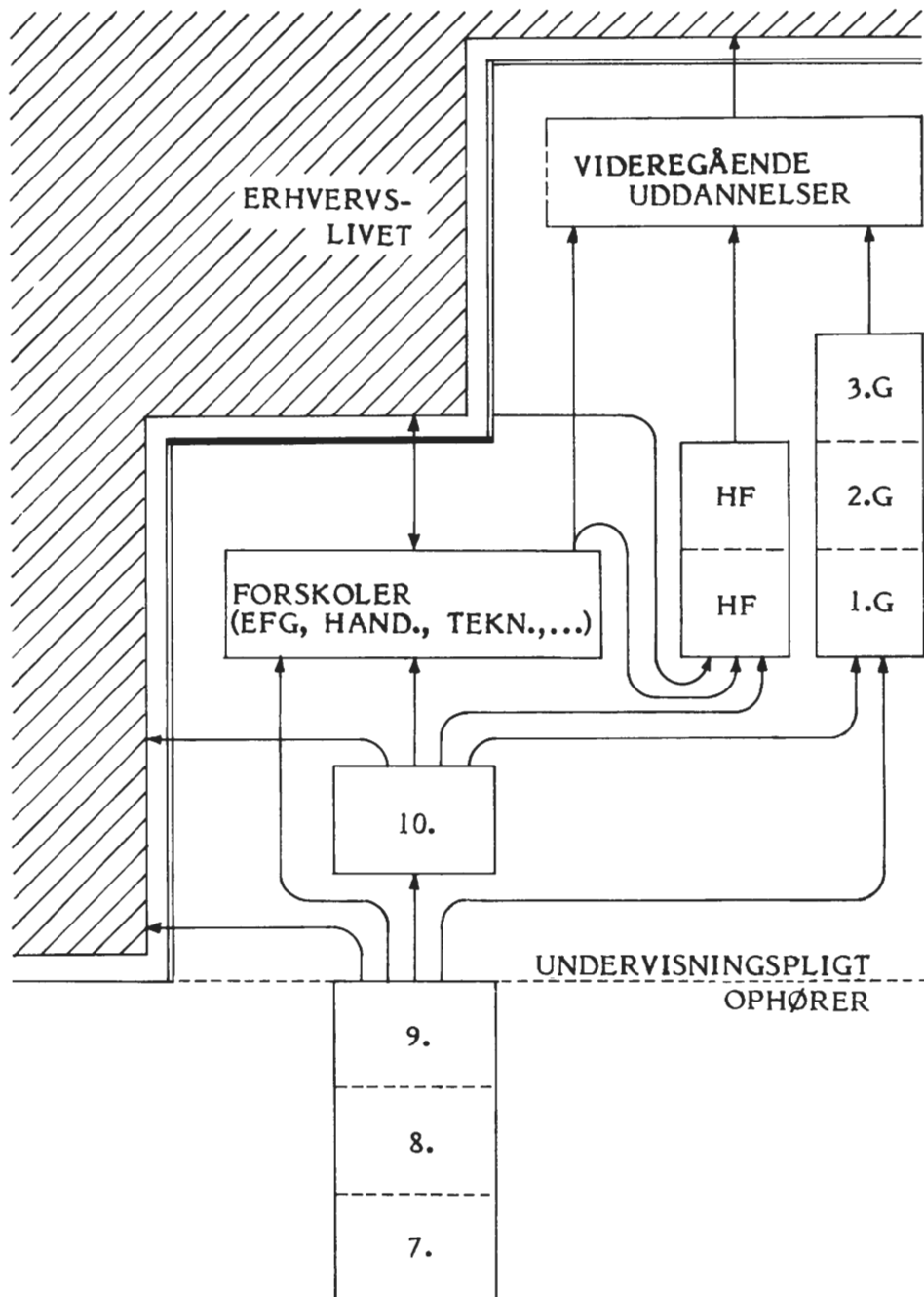
► = 1 tryk ▷ = gentagne tryk
►► = 2 tryk
►►► = 3 tryk



Indstilling af normaltid



Eksempel 4



Nævn nogle eksempler på algoritmer fra din hverdag.

BESKRIVELSE AF ALGORITMER

I det foregående har du set forskellige eksempler på, hvordan algoritmer kan beskrives. Beskrivelsen kan ske ved en billedserie, en tekst eller begge dele.

Du skal prøve at beskrive en algoritme med dine egne ord. Udgangspunktet er en billedserie.

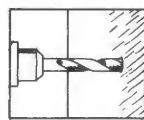
Øvelse G

Følg billedserien.

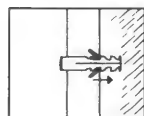
Hvad sker?

Beskriv med egne ord algoritmen.

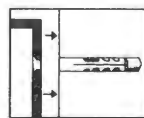
(Husk: trinvis, rækkefølge)



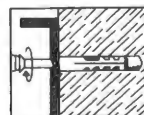
1) _____



2) _____



3) _____



4) _____

Øvelse H

Følg billedserien.

Hvad sker?

Beskriv med egne ord algoritmen.

(Husk: trinvis, rækkefølge)



①



②

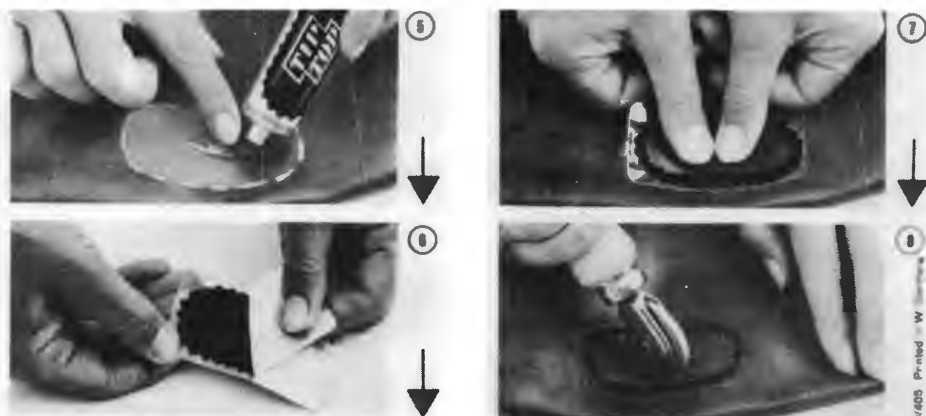


③



④





OVERSIGT

Du har nu lært, hvordan man opstiller og arbejder med algoritmer.

Ved opstilling af en algoritme skal man nøje angive:

1. Starttilstanden.
2. Hvert trin i problemløsningen.
3. Rækkefølgen, de enkelte trin skal løses i.
4. Sluttilstanden.

OPGAVER

Opgave 5.1

- a.
Undersøg dit eget ur og opstil en algoritme over normaltids.
- b.
Opstil en algoritme over procentberegning på en lommeregner.
- c.
Opstil en algoritme over, hvordan man skifter ventilgummi på en cykel.
- d.
Opstil en algoritme over, hvordan man skifter tændrør på en knallert.
- e.
Opstil en algoritme over, hvordan man afspiller et kassettebånd.

Opgave 5.2

Algoritmespil:
Opstil en algoritme over en bestemt opgave (f.eks. fremstilling af en papirflyver). Byt med en kammerat og undersøg, om I kommer til det ønskede resultat.

KAPITEL 6 RUTEDIAGRAMMER

Når man skal beskrive en algoritme, kan man illustrere dens trin ved hjælp af symboler. Disse symboler udgør tilsammen et RUTEDIAGRAM.

Et rutediagram viser tydeligt:

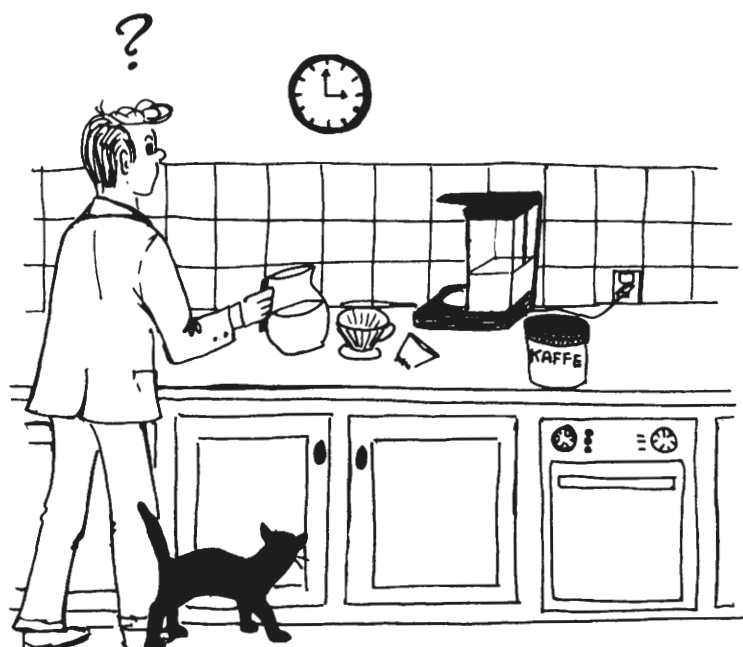
Starttilstanden	hvor processen begynder.
Hvert trin	hvorledes opgaven løses.
Rækkefølgen	hvorledes trinene følger efter hinanden.
Sluttilstanden	hvor processen ender.

Som det første eksempel på et rutediagram vil vi bruge en kendt situation fra hverdagen

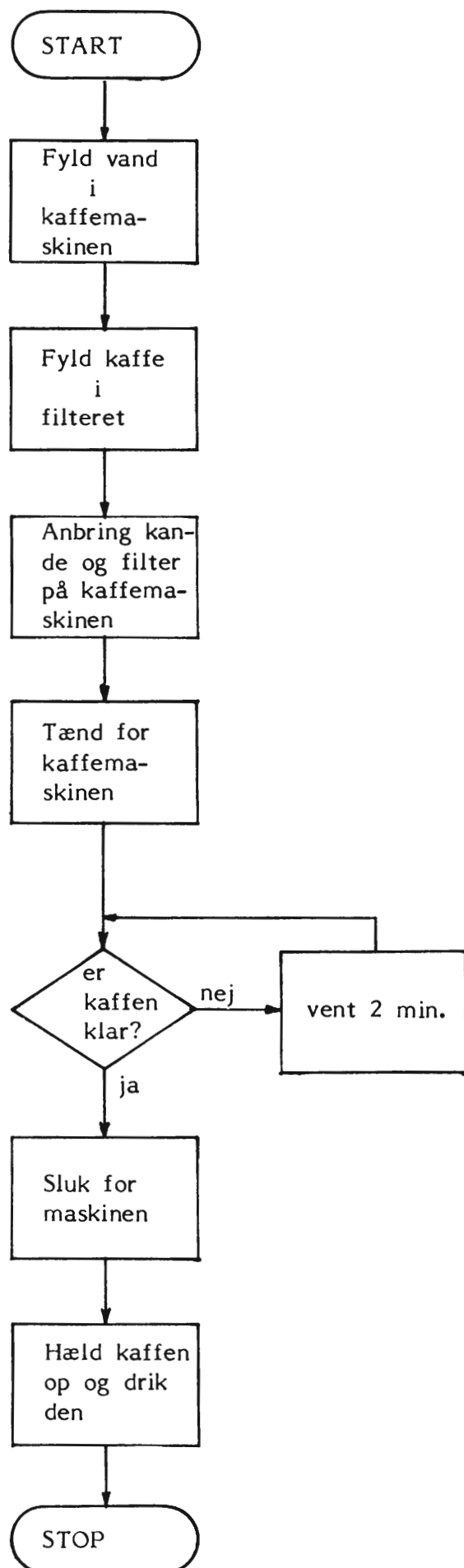
Ole skal til at lave kaffe.

Han er i besiddelse af: kaffemaskine, vand, filterposer og kaffebønner.

Den fremgangsmåde, han følger trin for trin, er beskrevet i rutediagrammet på næste side.



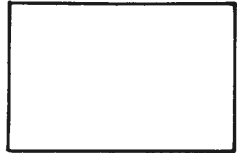
Rutediagram



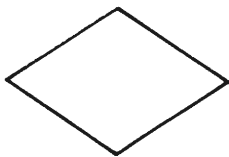
I rutediagrammet har vi brugt følgende symboler:



START eller STOP, der viser, hvor rutediagrammet begynder eller slutter.



BEHANDLING, der bruges, hvor der skal foretages en handling - f.eks. "Fyld kaffe i filteret" eller "Sæt $x = 4.5$ "



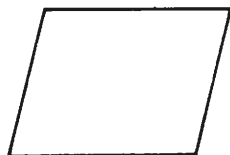
SPØRGEKASSE eller FORGRENING. Heri anbringes det, der i matematikken kaldes et **udsagn** - altså et spørgsmål, som kun kan besvares med JA eller NEJ - f.eks. "Lyser den grønne lampe?" eller "Er x større end 4,5?".



PILE, der forbinder "kasserne", angiver rækkefølgen, RUTEN.

FRA en spørgekasse (Forgrening) udgår der mindst **2 ruter**. Fra de andre kasser udgår der kun **én rute**.

Ud over disse symboler, som vi allerede har set anvendt, vil du få brug for endnu et:

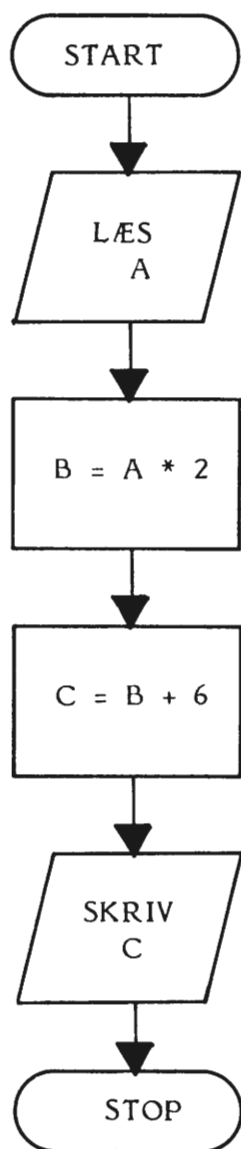


LÆSE- eller SKRIVEKASSE, som bruges, både når der skal indlæses data, og når resultater skal udskrives.

Herefter vil vi bruge rutediagrammerne til at beskrive, hvordan problemer kan løses ved hjælp af en datamat. De fleste af opgaverne ligner dem, du kender fra regning og matematik.

Øvelse A

Gennemløb rutediagrammet med værdierne for A, og udfyld højre side af skemaet. Første eksempel, hvor A er 4, er løst.



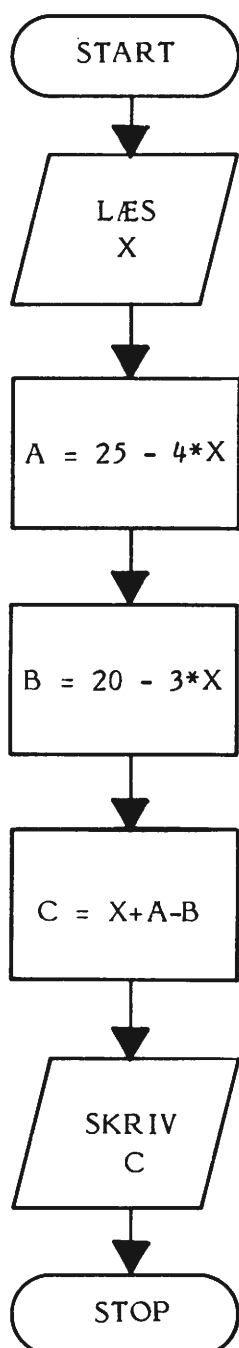
IND (A)	UD (C)
4	14
6	
8	
13	
15	
100	
200	
500	



Øvelse B

Gennemløb rutediagrammet med de værdier, som er angivet i søjlen under X. Find evt. selv nogle værdier for X, som du kan bruge.

Der er plads til, at du kan skrive værdierne for A, B og C i skemaet.

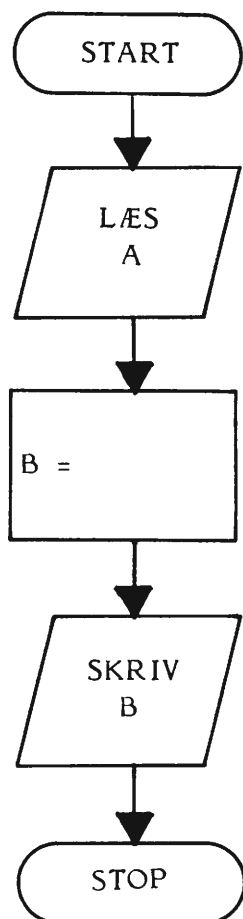


X	A	B	C
1			
2			
3			
4			
5			
6			
0			
0,25			
0,5			

Øvelse C

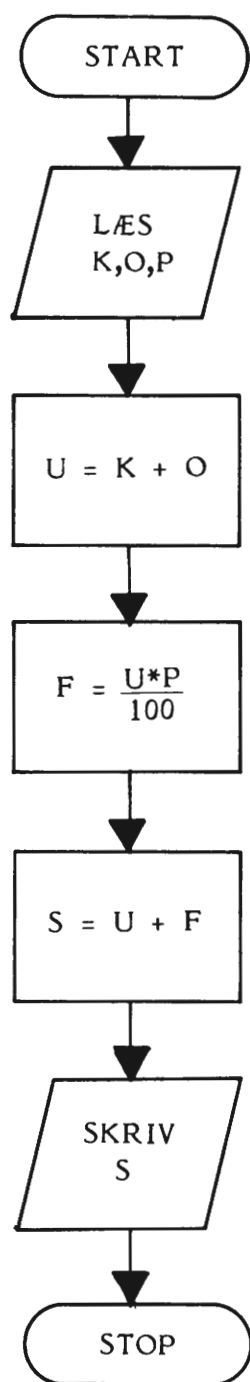
Dette rutediagram er ikke færdigt.

Skriv det, der skal stå i behandlings-kassen, for at det kan passe med følgende ind- og uddata. Udfyld skemaet.



A	B
1	4
2	7
3	10
4	13
5	16
6	
7	
8	
9	

Øvelse D



K	O	P	U	F	S
500	50	8			
1200	300	20			
25000	1000	25			

Prøvekør rutediagrammet med de data, som er vist i skemaet ovenfor. Find eventuelt selv på flere data.

Hvad udfører algoritmen? _____

FORGRENINGER

Valg

I mange af hverdagens situationer står du over for at skulle vælge mellem forskellige muligheder. De beslutninger, du træffer, kan være påvirkede af tilfældige ydre omstændigheder eller af det humør, du er i den dag.

Når datamaten skal foretage et valg, må det naturligvis ikke være afhængigt af tilfældigheder. Den skal komme til det samme resultat hver gang, programmet udføres.

Aritmetiske udsagn

Hvis datamaten er i gang med et program, hvor X har en bestemt værdi, kan den med sikkerhed afgøre, om udsagnet

$X = 4$ er sandt eller falsk.

I udsagn af denne type, hvor talstørrelser skal sammenlignes, anvendes forskellige symboler, som er kendt fra matematikken:

Symboler for sammenligning

=	lig med
>	større end
>=	større end eller lig med
<	mindre end
<=	mindre end eller lig med
<>	forskellig fra

Øvelse E

Vælg det rigtige symbol:

34 56

56 56

56 54

Øvelse F

Vælg det symbol, som angiver den modsatte betingelse:

$A = 4$ $A \neq 4$

$A > 9$ $A \leq 9$

$A < 78$ $A \geq 78$

$A \geq 0$ $A < 0$

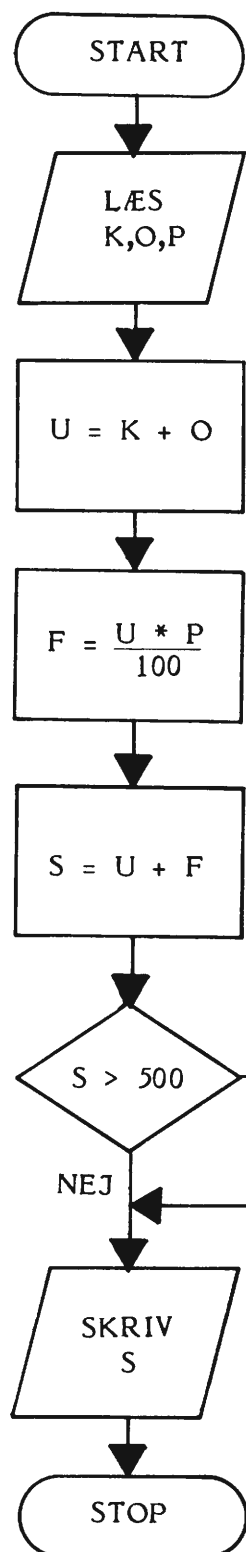
Øvelse G

Rutediagrammet i øvelse D beskrev, hvorledes man beregner salgsprisen for en vare.

Lad os give opgaven denne tilføjelse:

Hvis salgsprisen er større end 500 kr., gives der 5% rabat.

Rutediagrammet ser således ud:

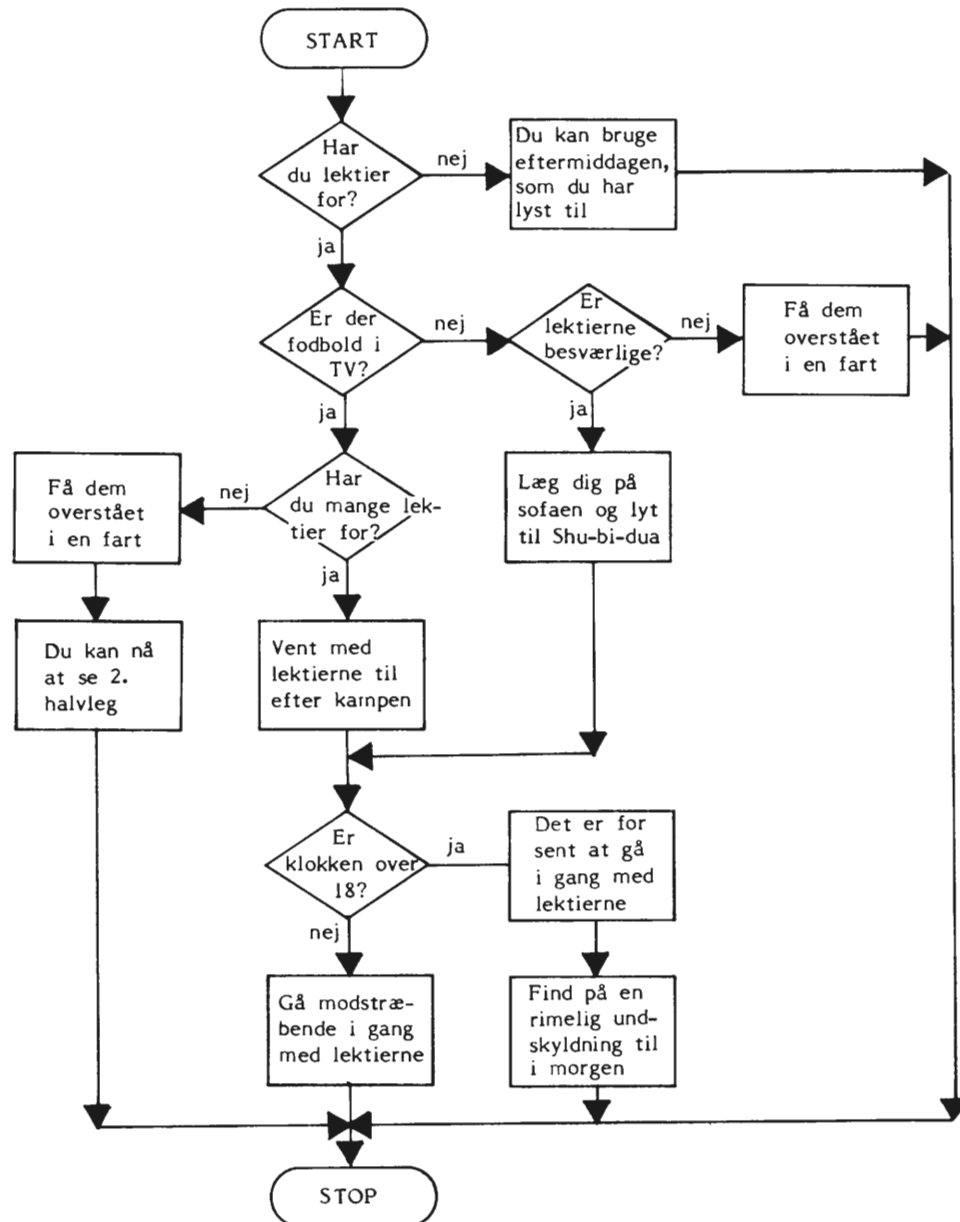


K	200	400	275
O	50	100	125
P	8	20	25
U			
F			
S			

Når du kommer til spørgekassen i øvelse G, skal du besvare spørgsmålet: Er S større end 500? Hvis svaret er JA, følger du pilen til højre. Er svaret NEJ, følger du pilen nedad.

Øvelse H

Her kommer til sidst et rutediagram, som du kan benytte efter skoletid.



OVERSIGT

Nu har du lært, at et program til datamaten skal bygges op som en algoritme, og at algoritmen kan gengives ved hjælp af et rutediagram. Du mangler blot at lære, hvordan rutediagrammet skal oversættes til et sprog, datamaten forstår.

Men først må vi lige gøre os klart, hvad man overhovedet kan få en datamat til at udføre:

Den kan:

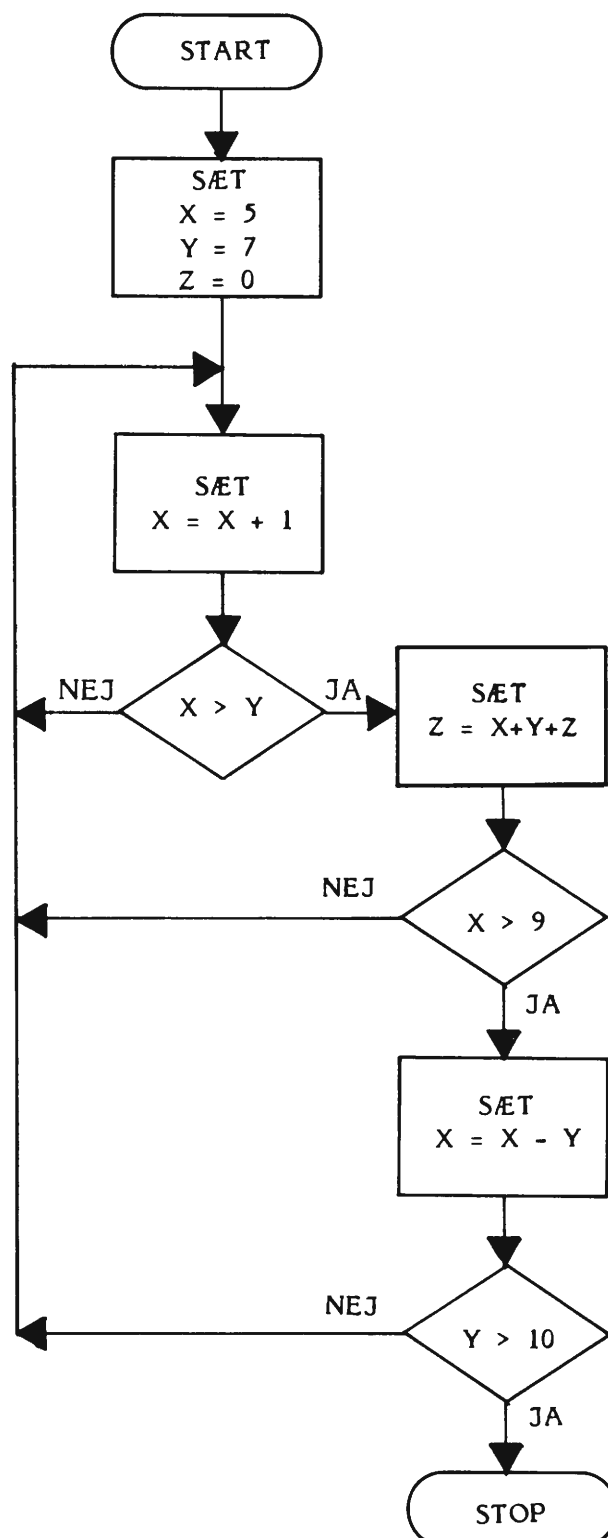
LÆSE	fra skærm, diskette eller bånd
SKRIVE	på skærm, diskette, bånd eller papir
HUSKE	de oplysninger, den har modtaget
REGNE	i lighed med en lommeregner
SAMMENLIGNE	tal og afgøre deres indbyrdes størrelsesforhold, bogstaver og afgøre deres indbyrdes placering i alfabetet

OPGAVER

Opgave 6.1

Hvilken værdi har Z, når nedenstående diagram når STOP?

- A. 27
- B. 31
- C. 17
- D. 46
- E. Fejl i diagram.



Opgave 6.2

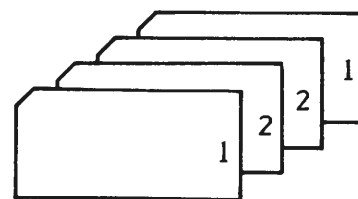
Du skal følge rutediagrammet på næste side.

Når du kommer til

LÆS
ET
KORT

anvender du i rækkefølge de

data, som er angivet på disse kort:



altså tallene 1 - 2 - 2 - 1

Du skal samtidig holde styr på 2 kontakter X og Y. De er begge slukkede ved opgavens start; det er angivet i skemaet nedenunder med bogstavet S.

Når du får ordre til at tænde for kontakt X eller Y, streger du bogstavet S og skriver i stedet bogstav T for **tændt**.

Det er vigtigt, at du holder godt øje med, om kontakterne X og Y er Slukkede eller Tændte, idet du med jævne mellemrum bliver spurgt, om kontakterne er tændte eller slukkede.

Når du kommer til en udskriftkasse, f.eks.

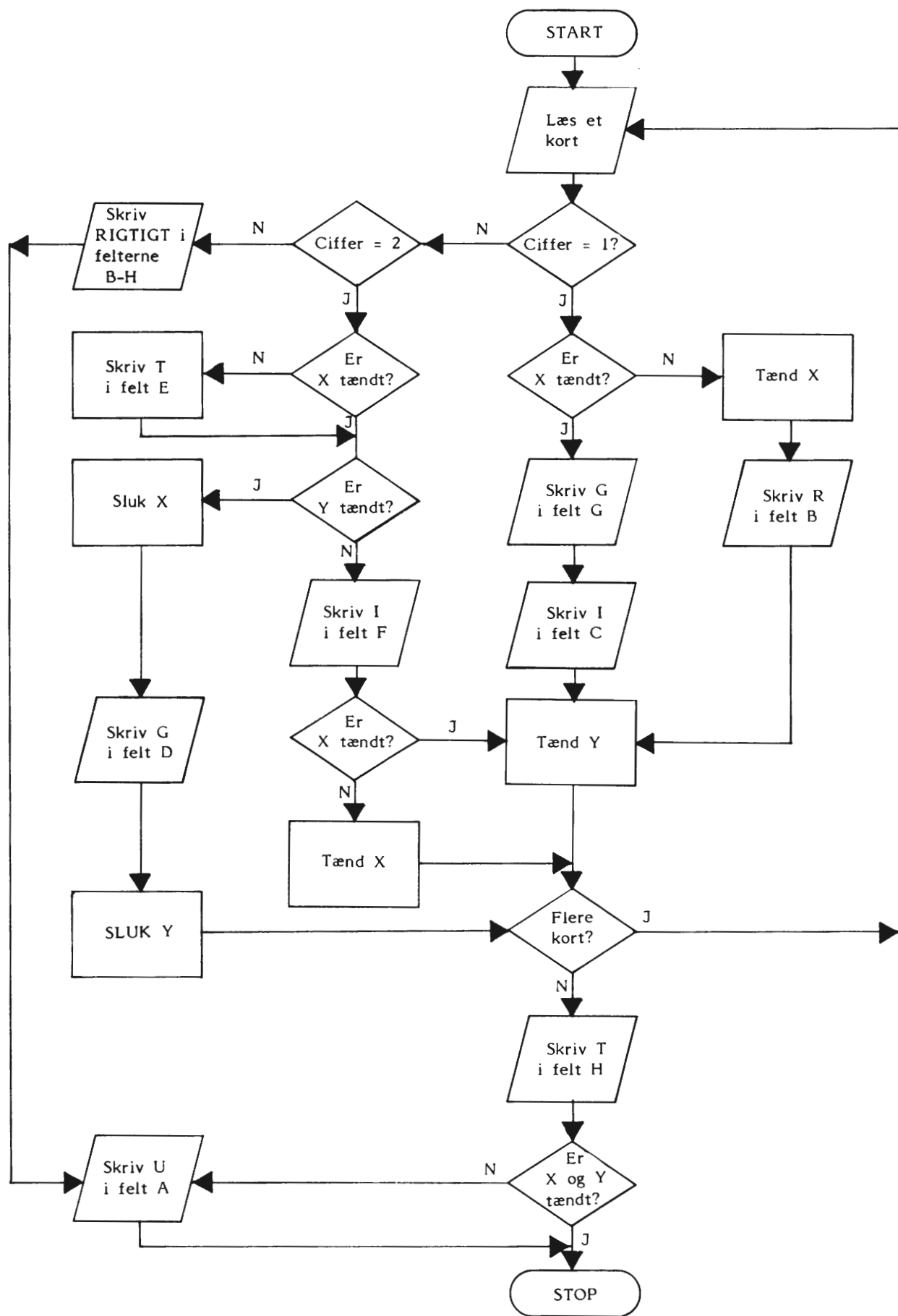
SKRIV
"T" I
FELT H

så skriver du i skemaet nederst på denne side bogstavet T i feltet under bogstav H.

Når du kommer til stop i rutediagrammet, kan du i den nederste tabel se, om har løst opgaven korrekt.

KONTAKT	X	S								
KONTAKT	Y	S								

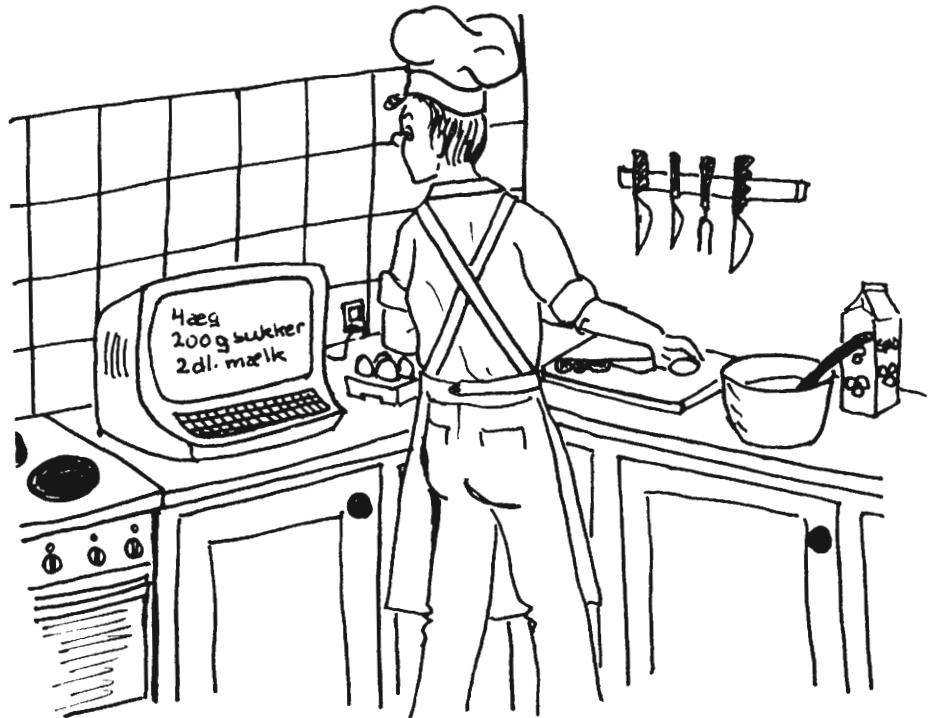
[illegible]



Testopgave

Opgave 6.3

Programmet TEST ligger på disketten. LOAD "TEST2" og svar på spørgsmålene.



KAPITEL 7 PROGRAMMERING I

Når vi skal løse problemer ved hjælp af en datamat, skal vi først opstille en **algoritme**.

Algoritmen vil vi til at begynde med beskrive ved hjælp af et **rutediagram**.

Rutediagrammet skal nu blot "oversættes" til et sprog, som maskinen kan forstå.

Comal

Programmeringssproget COMAL.

Vi vil i denne bog skrive programmer ved hjælp af programmeringssproget **COMAL**.

Et COMAL PROGRAM består af en række comal-sætninger - en på hver linie.

Til hver sætning hører et LINIENUMMER. Som linienummer må benyttes tal mellem 1 og 9999.

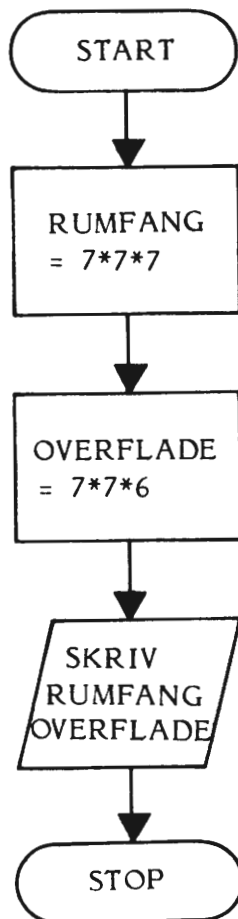
Da vi tit får brug for at rette eller tilføje sætninger, er det en god ide at anvende linjenumrene 10 - 20 - 30 o.s.v.

Programeksempel 1

Vi vil beregne rumfang og overflade af en terning med siden 7 cm.

Rutediagrammet kan se således ud:

Programmet kan se således ud:



0010 // TERNING

0020 rumfang:= 7*7*7

0030 overflade:= 7*7*6

0040 PRINT rumfang,overflade

Forklaring

0010 // TERNING

I en // sætning kan man indsætte bemærkninger, der f.eks. fortæller, hvad programmet udfører.

0020 rumfang:= 7*7*7

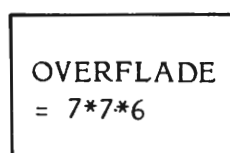
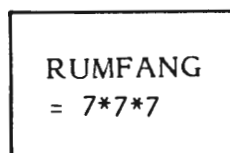
Her foretages en beregning. Den variable RUMFANG tildeles værdien af udtrykket 7*7*7.

0030 overflade:= 7*7*6

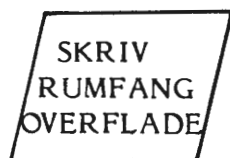
Den variable OVERFLADE tildeles værdien 7*7*6.

//

Beregning

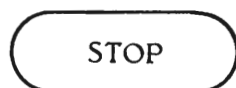


PRINT



```
0040 PRINT rumfang,overflade
```

Her udskrives resultatet.



Behøver ikke noget linjenummer, da maskinen af sig selv stopper, når den har udført linjen med det højeste linjenummer.

Dynamisk lighedstegn

Dynamisk lighedstegn := betyder "får værdien af"
--

Eks: A := 4 betyder, at A får værdien 4
B := A + 7 betyder, at B får værdien af A + 7

Programmet kan altså se således ud:

```
0010 // TERNING  
0020 rumfang:= 7*7*7  
0030 overflade:= 7*7*6  
0040 PRINT rumfang,overflade
```

Øvelse 7.1

Indtast programmet

Skriv RUN

Tryk RETURN

Hvad bliver rumfang og overflade? _____

VARIABLE

Sætninger af typen:

```
0010 ANTAL := 4 * 7 + 3  
0025 pris:= 12.8*1.22  
0050 sum:= sum+1
```

Variable

indeholder **på venstre side af lighedstegnet** en VARIABLE. Som variabel kan bruges et bogstav, en samling bogstaver eller en samling af bogstaver og tal. Ialt må variabelnavne indeholde højst 16 tegn. Det første tegn **skal** være et bogstav; de næste må være bogstaver eller tal.

Det er praktisk at bruge variabelnavne, som har forbindelse med den opgave, som skal løses, f.eks. PRIS, HØJDE, SALGSPRIS o.s.v. Derved bliver programmet let at læse både for den, som har udarbejdet programmet, og for den, som skal bruge det.

Her er nogle eksempler på anvendelige variabelnavne:

KØBSPRIS
FORTJENESTE
PRIS1
PRIS2
AREAL

På højre side af lighedstegnet står værdien af den variable. Det kan være et tal eller et aritmetisk udtryk.

Når maskinen under udførelsen af et program når en sætning som

```
0030 moms:= pris*0.22
```

sker der det, at en "celle" i maskinens lager får navnet MOMS. I denne "celle" placeres nu det tal, som fremkommer ved at man ganger PRIS med 0.22. Da PRIS er en variabel, hentes værdien fra "cellen" med navnet PRIS frem og ganges med 0.22, hvorefter resultatet placeres i "cellen" MOMS.

Hver gang maskinen i et program møder et nyt variabelnavn, navngives en ny "celle".

Hvis man senere i programmet tildeler den variable MOMS en ny værdi, vil denne nye værdi **erstatte** den gamle værdi. Der kan altså kun stå **en** værdi for hver variabel i en "celle". Hvis vi i et program forsøger at finde en variabel, som ikke er oprettet, vil maskinen stoppe, og vi vil få en fejludskrift.

BEREGNINGER

Til de forskellige beregninger inden for Comal bruges følgende regnetegn:

Regnetegn

SYMBOL	BETYDNING	EKSEMPEL
+	ADDITION	12 + 3
-	SUBTRAKTION	15 - 6
*	MULTIPLIKATION	32 * 96
/	DIVISION	64/24
↑	POTENS	2 ↑ 10
0.5	DECIMALKOMMA	5.56
SQR(A)	KVADRATROD	SQR(16)

Når udtryk skal beregnes, gælder følgende rækkefølge:

- 1) Potensopløftninger udføres først
- 2) Multiplikationer og divisioner udføres derefter
- 3) Additioner og subtraktioner udføres sidst

Øvelse A

Hvordan omsættes følgende matematiske udtryk til udtryk i Comal?

$$a^2$$

$$\underline{a \uparrow 2}$$

$$a(b-c)$$

$$\frac{a^2}{b}$$

$$b^2 - 4ac$$

$$k(l + r)^n$$

$$\frac{k \cdot p \cdot d}{100 \cdot 360}$$

$$3,6 \cdot 7^2$$

$$\frac{4}{3} \cdot \pi \cdot r^3$$

$$\sqrt{a^2 + b^2}$$

Øvelse B

Udregn værdien af følgende udtryk:

$$A = 2 + 3 \cdot 5 \uparrow 2$$

$$A = \underline{\hspace{2cm}}$$

$$B = 8 \cdot (256/8 + 32)$$

$$B = \underline{\hspace{2cm}}$$

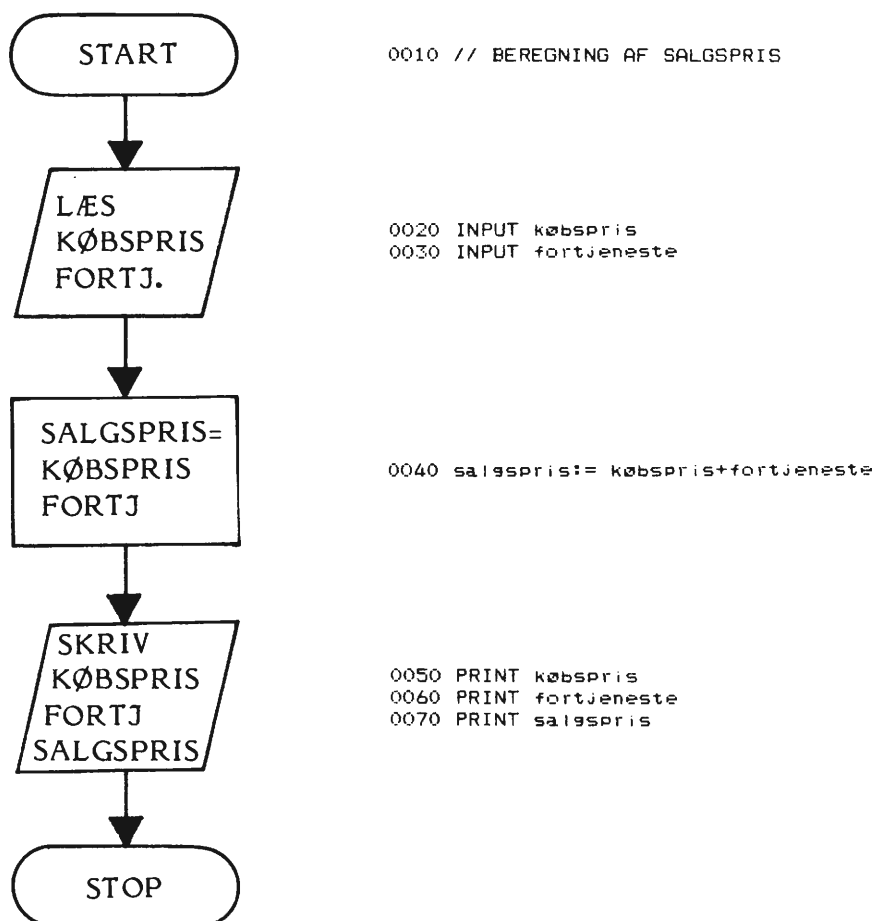
$$C = 4 \uparrow 2 + 3 \cdot (5 + 2 \uparrow 3)$$

$$C = \underline{\hspace{2cm}}$$

Programeksempel 2

Vi vil nu vise, hvorledes man udregner salgsprisen, når man indtaster købspris og fortjeneste.

INDDATA	BEREGNING	UDDATA
Købspris Fortjeneste	Salgspris = Købspris + Fortjeneste	Købspris Fortjeneste Salgspris

**Forklaring**

0010 // Beregning af salgspris

// bruges til at fortælle programmets indhold til brugeren, når han tager "en LIST" af programmet. En // sætning har ingen indvirkning på programmets udførelse.

INDDATA

0020 INPUT købspris
0030 INPUT fortjeneste

I linierne 0020 og 0030 står Comal-ordren INPUT. Hvis vi skriver RUN, og programmet kommer til linie 0020, vil systemet udskrive enten et : eller et ?.

Det betyder, at systemet venter på, at vi indtaster en konstant i den variable, som vi har kaldt KØBSPRIS.

Vi indtaster tallet 100 og trykker RETURN. Tallet 100 er nu blevet lagt ind i den variabel, som vi har kaldt KØBSPRIS.

Systemet er nu nået til linie 0030 og udskriver igen et : eller et ?. Det vil sige, at vi skal indtaste en ny konstant. Vi indtaster tallet 25, som bliver lagt ind i cellen FORTJENESTE, når vi trykker på RETURN.

BEREGNING

```
0040 salgspri:= købspri+fortjeneste
```

Når programmet når til linie 0040, vil SALGSPRIS blive udregnet på grundlag af de to konstanter, som vi har lagt ind i cellerne KØBSPRIS og FORTJENESTE, og tallet 125 vil blive lagt ind i den variable, som vi har kaldt SALGSPRIS.

UDDATA

```
0050 PRINT købspri
0060 PRINT fortjeneste
0070 PRINT salgspri
```

I linierne 0050, 0060 og 0070 bliver de konstanter, som er indlagt i de tre celler, som vi har kaldt KØBSPRIS, FORTJENESTE og SALGSPRIS udskrevet:

100
25
125

d.v.s.,	KØBSPRIS	= 100
	FORTJENESTE	= 25
	SALGSPRIS	= 125

OVERSIGT

//	En // sætning består af tegnet // + kommentar
----	---

PRINT	PRINT-ordren benyttes bl.a. til udskrivning af tal
-------	--

VARIABEL En variabel kan bestå af 16 tegn, hvoraf det første skal være et bogstav

INPUT INPUT-ordre, der bruges til at indtaste en talværdi (konstant) i en variabel.

OPGAVER

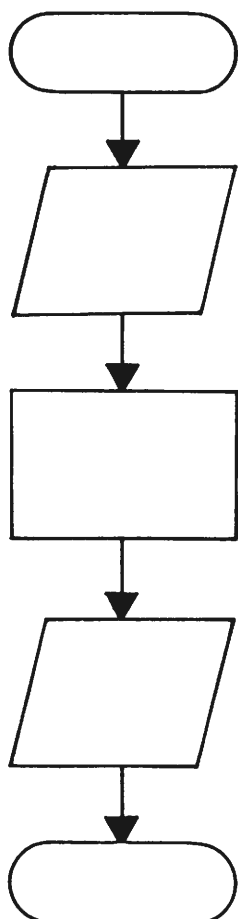
Opgave 7.1

Lav et program, der udskriver arealet af et rektangel, når man indtaster rektanglets længde og bredde.

INDDATA	BEREGNING	UDDATA
Bredde Længde	$\text{Areal} = \text{Bredde} * \text{Længde}$	Bredde Længde Areal

Udfyld rutediagrammet og færdiggør programmet i Comal:

RUTEDIAGRAM



PROGRAM i COMAL

0010 // AREALBEREGNING AF REKTANGEL

0020 INPUT bredde

0030 INPUT længde

0040

0050

0060

0070

Skriv derefter det tilsvarende program i Comal.

Til sidst skriver du LOAD "AREAL" og LIST, så du nu kan prøvekøre programmet nogle gange, idet du indtaster nedenstående tal for BREDDE og LÆNGDE og udfylder de manglende rubrikker.

BREDDE	12	56	27	38
LÆNGDE	5	99	33	28
AREAL				

Opgave 7.2

Du skal nu indtaste følgende program:

```
0010 // ADRESSEUDSKRIVNING
0020 PRINT " HANS HANSEN "
0030 PRINT " VIBORGVEJ 27 "
0040 PRINT " 1111 SØBY.  "
```

Skriv derefter RUN og tryk på RETURN:

Hvilke tre linier udskrives?

Opgave 7.3

Du skal nu skrive et meddelelsesark til 8. b's forældre med følgende udskrift:

TIL HANNES FORÆLDRE,
VI KAN MEDDELE DEM, AT HANNE HAR FÅET FØLGENDE
KARAKTERER I MATEMATIK:

1. SKR. MATEMATIK:	8
2. MDT. MATEMATIK:	9
3. ORDEN:	9

MED VENLIG HILSEN,
K. V. ROED,
MATEMATIKLÆRER.

Man kan skrive de tre karakterer ind i brevet ved hjælp af tre input-sætninger.

På næste side kan du se, hvorledes du kan begynde programmet:

Begynd således:

Skriv AUTO og tryk på RETURN:

Ved hjælp af systemkommandoen AUTO udskriver systemet selv linienummer, hver gang du trykker på RETURN.

```
0010 PRINT CHR$(12)
0020 INPUT "SKR. MATEMATIK:
0030 INPUT "MDT. MATEMATIK:
0040 INPUT "ORDEN:
0045 PRINT CHR$(12)
0050 PRINT "TIL HANNE'S FORÆLDRE,
0060 PRINT
0070 PRINT "VI KAN MEDDELE DEM, AT HANNE HAR FÆT FØLGENDE "
0080 PRINT "KARAKTERER I MATEMATIK:
0090 PRINT
0100 PRINT "1. SKR. MATEMATIK:
": skr
": mdt
": orden
"
"
"
```

etc.

Gør selv programmet færdigt, og prøvekør det nogle gange med forskellige tal for de tre karakterer.

Skriv derefter LOAD "HANNE", og sammenlign programmerne.

●OVERSIGT

PRINT

PRINT-ordren bruges bl.a. til udskrivning af tekster. Alle tegn samt mellemrum, der står mellem anførselstegnene, bliver udskrevet på nøjagtig samme måde, som de er indtastet

PRINT CHR\$(12)

bevirker, at vi kan begynde brevet på en "ny" skærm

INPUT

Ved INPUT-sætninger bliver alt, der står i anførselstegn, udskrevet efter de samme regler, som gælder ved PRINT-sætninger

AUTO

Systemkommandoen AUTO udskriver automatisk linienumre som multipla af 10

OPGAVER

Opgave 7.4

Indtast følgende program:

```
0010 INPUT højde
```

```
0020 PRINT "DERES HØJDE ER MÅLT TIL: ", højde
```

I linie 20 finder man såvel en **tekst** som en **variabel**.

Teksten lyder: _____

Variablen hedder: _____

Tast RUN og skriv derefter 180, når systemet har skrevet tegnet ":".

Hvad udskrives? _____

Indtast derefter:

```
0020 PRINT "DERES HØJDE ER MÅLT TIL: ", højde, " CM. "
```

Hvad udskrives nu? _____

Opgave 7.5

Skriv følgende program:

```
0010 PRINT "INDTAST DATOEN FOR SOMMERFERIENS BEGYNDELSE: "
```

```
0020 INPUT dato
```

```
0030 PRINT "SOMMERFERIEN BEGYNDER: ", dato
```

Hvilken udskrift fås? _____

HUSK! Det tal, som du indtaster i den variable DATO, skal være en ren talværdi som f.eks. 200682 eller 820620.

Opgave 7.6

Skriv et program, der udskriver rumfanget af en kasse, når man indtaster BREDDE, LÆNGDE og HØJDE.

Udskriften kunne f.eks. se således ud:

BREDDE:	10	CM.
LÆNGDE:	20	CM.
HØJDE:	30	CM.
RUMFANG:	6000	CM ³ .

Begynd således:

AUTO, derefter RETURN.

```
0010 PRINT CHR$(12)  (RETURN)
```

```
0020 INPUT "INDTAST BREDDE: " : bredde
```

```
0030 INPUT "INDTAST LÆNGDE: " : længde
```

```
0040 INPUT "INDTAST HØJDE: " : højde
```

```
0050 rumfang:= bredde*længde*højde  
etc.
```

Fortsæt selv resten af programmet, således at du får en udskrift som den ovenfor skitserede.

Prøvekør dit program nogle gange. Skriv derefter LOAD "RUMF" og sammenlign programmerne.

Opgave 7.7

Skriv et program, der omsætter Celsius til Réamur og Fahrenheit.

Benyt formlerne:

$$\begin{aligned}\text{Réamur} &= \text{Celsius} * 0.8 \\ \text{Fahrenheit} &= \text{Celsius} * 1.8 + 32\end{aligned}$$

Udskriften kunne være således:

CELSIUS:	100
REAMUR:	80
FAHRENHEIT:	212

Begynd således:

```
0010 PRINT CHR$(12)
0020 INPUT "INDTAST CELSIUS: " ; celsius
0030 reamur:= celsius*0.8
etc.
```

Gør selv programmet færdigt, og prøvekør det derefter nogle gange.

Skriv derefter SELECT OUTPUT "printer". Du vil nu få en kopi af dit eget program på printeren.

Til sidst skriver du LOAD "CELSIUS" og LIST, så at du kan sammenligne programmerne.

Opgave 7.8

Skriv et program, der udskriver renten for et bestemt antal dage.

Man kan benytte formlen:

$$\text{RENTEN} = \text{KAPITAL} * \text{RENTEFOD} * \text{DAGE}/100/360$$

NB! RENTEDAGE kan højst blive 360.

INDDATA	BEREGNING	UDDATA
Kapital Rentefod Rentedage	$\text{Rente} = \text{Kapital} * \text{Rentefod} * \text{Rentedage} / 100 / 360$	Kapital Rentefod Rentedage Rente

Begynd således:

```

0010 PRINT CHR$(12)
0020 INPUT "INDTAST KAPITAL:           ": kapital
0030 INPUT "INDTAST RENTEFOED:         ": rentefod
0040 INPUT "INDTAST RENTEDAGE:         ": rentedage
0050 rente:= kapital*rentefod*rentedage/100/360
etc.

```

Udskriften kunne f.eks. være:

KAPITAL:	RENTEFOED:	RENTEDAGE:	RENTE:
1000 KR.	10 %	180 DAGE.	50 KR.

Prøvekør dit program et par gange. Tag derefter en udskrift af dit eget program på printeren ved at skrive SELECT OUTPUT "printer" og LIST.

Til sidst kan du skrive LOAD"RENTE" og sammenligne programmerne.

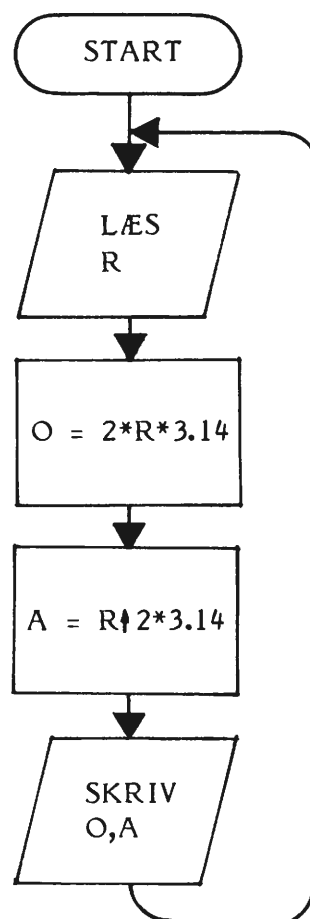


KAPITEL 8 PROGRAMMERING 2

UBETINGEDE HOP-ORDRER

Normalt vil et program altid blive udført i stigende linjenummerorden. Denne rækkefølge kan dog ændres ved forskellige instruktioner:

Programeksempel



Programmet kan se således ud:

```
0005 fortsæt: Label fortsæt

0010 INPUT "INDLÆS RADIUS: " : radius

0020 overflade:= 2*radius*3.14

0030 areal:= radius^2*3.14

0040 PRINT "OVERFLADE: " ,overflade
0050 PRINT "AREAL: " ,areal

0060 GOTO fortsæt
```

Ordren i linje 60 kaldes en "ubetinget hopordre". Den anvendes meget i et sprog, som er beslægtet med COMAL - nemlig sproget BASIC. I COMAL er ordren unødvendig, og da overdreven brug af GOTO-sætninger gør programmer vanskelige at læse, vil vi kun omtale den kort.

GOTO

Øvelse A

Indtast programmet. Prøvekør programmet.

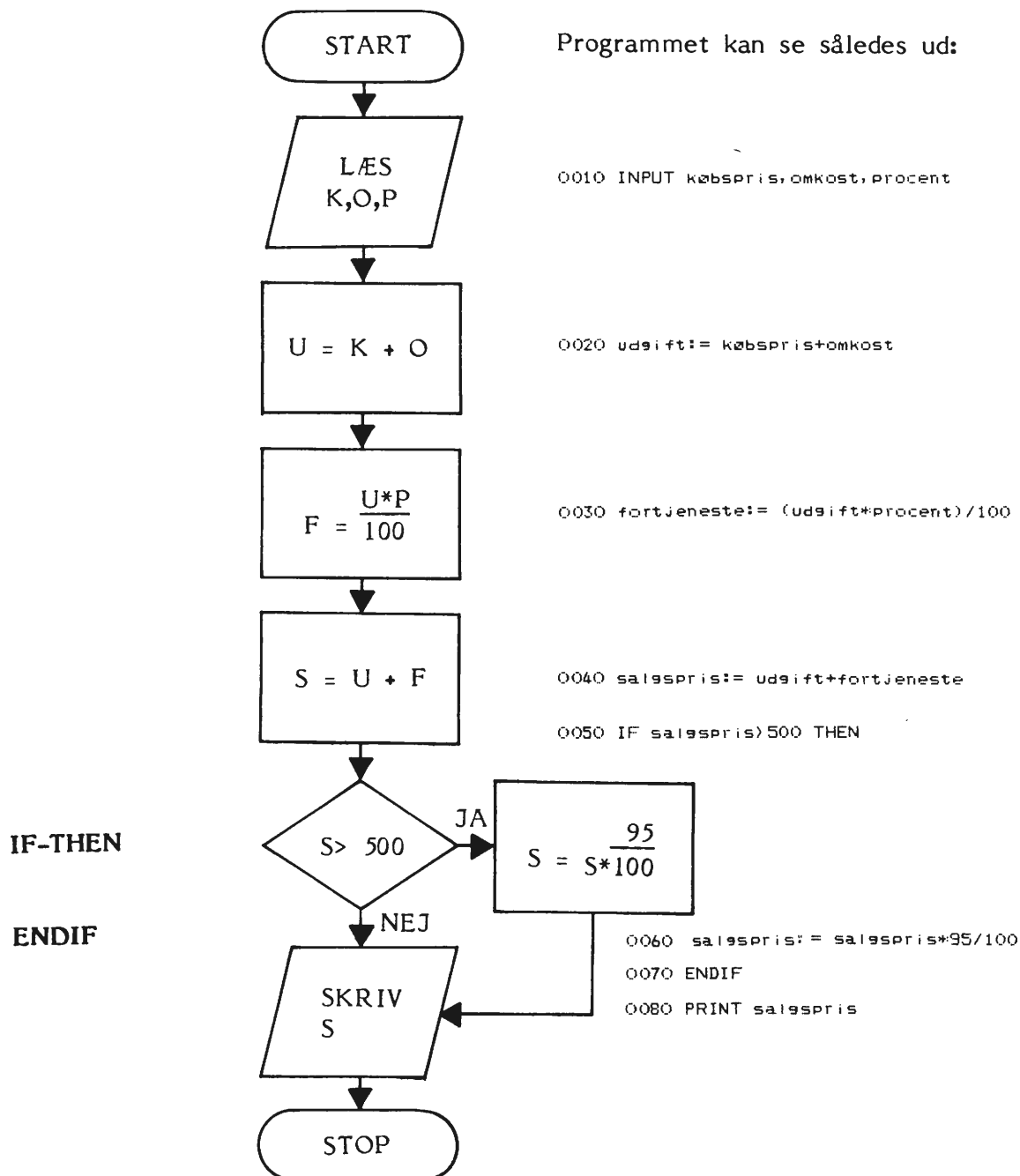
GOTO er en instruktion, der styrer programudførelsen. Når programmet er nået til linje 60, springes der til linje 5, hvor etiketten FORTSÆT står.

Som vi kan se på rutediagrammet, er der ingen slutning på programmet. Det er faktisk muligt at skrive programmer, som bliver ved i det uendelige. Det er jo meningsløst, og bl.a. derfor anvendes GOTO ikke i særlig udstrakt grad. I et senere kapitel vil vi vise, hvordan et problem som ovenstående kan løses meget mere elegant.

BETINGEDE HOP-ORDRER I

Vi ønsker at skrive et program, som kan beregne salgsprisen for en bestemt vare. Såfremt salgsprisen er større end 500 kr., gives 5% rabat. Datamaten skal altså foretage et valg.

Program eksempel

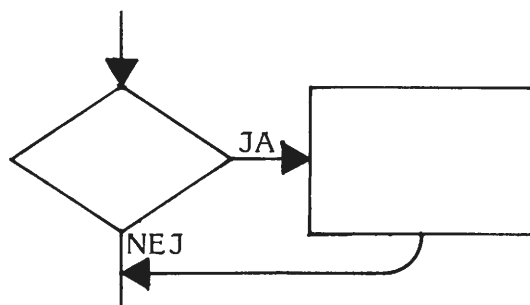


Øvelse B

Indtast og prøvekør programmet.

Som vi har set, kan datamaten afgøre, om et udsagn som f.eks.:

SALGSPRISEN STØRRE END 500 er sandt eller falsk.



En struktur som den her viste kan i programmeringssproget COMAL klares med denne programblok:

```
IF . . . THEN
. . . . .
ENDIF
```

Hvis udsagnet i IF-sætningen er sandt, udføres de ordrer, som følger umiddelbart efter. Hvis udsagnet er falsk, springes de over, og der fortsættes i programmet med det linjenummer, som kommer efter ENDIF.

Mellem IF . . . THEN og ENDIF kan der indsættes flere linjer. Programmet på foregående side er som øvelse G i kap. 6.

Programmet løser følgende opgave:

Hvis salgsprisen er større end 500 kr, gives 5% rabat. Det svarer til, at salgsprisen bliver 95% af den oprindelige salgspris.

Sandhedsværdi

Sandhedsværdi: (Boolsk udtryk)

Inden for programmering kaldes et åbent udsagn et "Boolsk udtryk". Datamaskinen kan afgøre, om udsagnet er sandt eller falsk.

Når to størrelser skal sammenlignes, bruges følgende logiske symboler:

Symbol	Betydning	Eksempel
=	lig med	a = 7
>	større end	b > 8
<	mindre end	c < 9
>=	større end eller lig med	d >= 10
<=	mindre end eller lig med	e <= 11
<>	forskellig fra	f <> 12

NB. Indenfor matematik skriver man som regel $d \geq 8$, men ved programmering bliver man altid nødt til at skrive det som to tegn: $d > = 8$.

Øvelse C

Vælg det rigtige tegn:

Tekst	Betingelse	Modsat betingelse
a større end 10 b lig med 8 k forskellig fra 10 m større end eller lig med 11 n mindre end - 3 o mindre end eller lig med 230		

Den modsatte betingelse af $x > 5$ er $x \leq 5$

Øvelse D

Skriv den modsatte betingelse:

Betingelse	Modsat betingelse
m > 3 t > -8 d >= 10 k <= 13 t = 7 p <> 8	

OVERSIGT

IF-sætning	IF åbent udsagn, efterfulgt af THEN
ENDIF	Kun ordet ENDIF

OPGAVER

Opgave 8.1

Skriv et program til følgende opgave: Et postordrefirma tillægger salgsprisen et gebyr på 15 kr, såfremt den samlede ordre er under 200 kr.

Brug IF . . . THEN ENDIF.

Opgave 8.2

En billet til Zoo koster normalt 16 kr. Hvis man er mere end 20 personer sammen, kan man komme ind for 3/4 af den nor-

male pris.

Skriv et program, der udskriver den samlede billetpris. Antal personer er inddata.

Opgave 8.3

Folk, som er fyldt 65 år, kan som regel rejse med DSB for halv pris.

Skriv et program, der med den rejsendes alder som INDDATA udregner billetprisen. Antag, at normalprisen for rejsen er 280 kr.

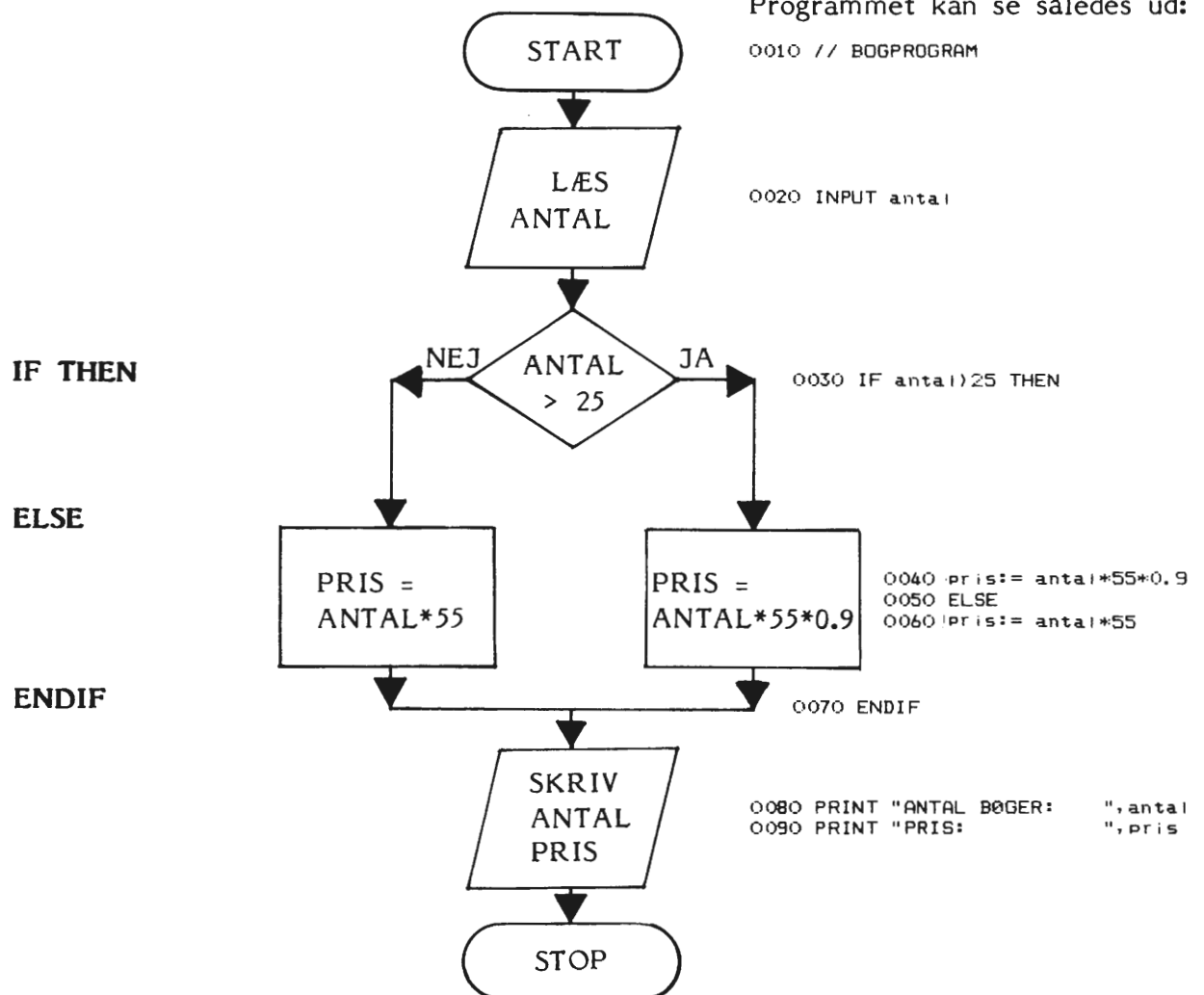
Du kan finde de 3 programmer på disketten under navnene: POST, ZOO og DSB.

BETINGEDE HOP-ORDRER II

Programeksempel

En skole bestilte et vist antal bøger hos et forlag. Prisen var 55 kr. pr eksemplar. Hvis der købes flere end 25 eksemplarer, giver bogforlaget 10% rabat.

Programmet kan se således ud:

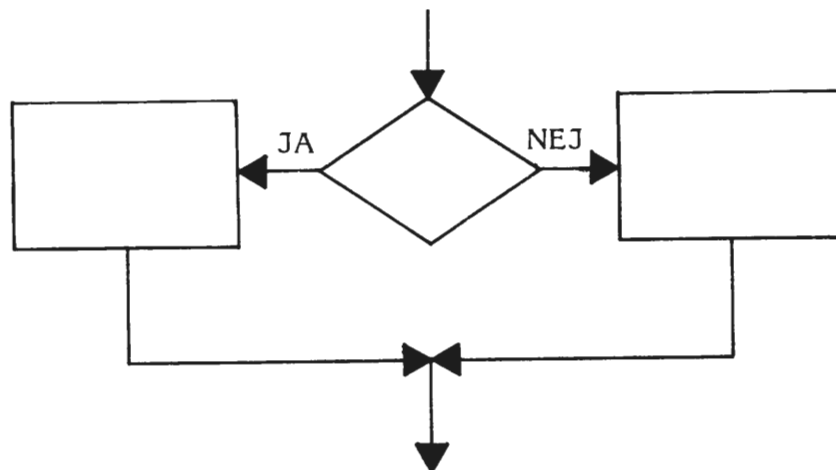


0030 I denne linie spørges, om antallet er større end 25.
Hvis JA, går man videre til linie 0040, hvor prisen med
10% rabat bliver udregnet. Programmet hopper derefter
ned til linie 80 og 90, hvor antal og pris bliver udskre-
vet.

Hvis det åbne udsagn i linie 30 derimod ikke er sandt, hopper
programmet ned til linie 60, hvor prisen uden rabat bliver ud-
regnet. Derefter bliver antal og pris udskrevet i linierne 80 og
90.

Øvelse E

Indtast programmet og prøvekør det.



En struktur som den her viste kan i COMAL klares med denne
programblok:

```
IF . . . THEN
. . . . .
. . . . .
ELSE
. . . . .
. . . . .
ENDIF
```

IF - THEN - ELSE - ENDIF

IF åbent udsagn (Boolsk udtryk) THEN

Denne del af programmet bliver **kun** udført, hvis det åbne
udsagn er sandt

ELSE

Denne del af programmet bliver **kun** udført, hvis det åbne
udsagn er falsk

ENDIF

OPGAVER

Opgave 8.4

Udskriv ved hjælp af en IF - THEN - ELSE - ENDIF -sætning et program, der giver:

1. Kassens rumfang
2. Kassens overflade,

når man indtaster kassens bredde, længde og højde.

Begynd således:

```
0010 PRINT CHR$(12)
0020 INPUT "INDTAST KASSENS BREDDER: " : bredde
0030 INPUT "INDTAST KASSENS LÆNGDE: " : længde
0040 INPUT "INDTAST KASSENS HØJDE: " : højde
0050 rumfang:= bredde*længde*højde
0060 overflade:= (længde*bredde+længde*højde+bredde*højde)*2
```

Nu skal du selv lave programmet færdigt, så at du kan få udskrevet enten kassens rumfang eller kassens overflade.

Når dit program virker tilfredsstillende, skal du skrive LIST "printer", så dit program bliver udskrevet på printeren.

Til sidst skriver du LOAD "KASSE" og sammenligner programmet, som du har fået fra disketten, med dit eget program.

Opgave 8.5

Rejsebureauet "Den gyldne sol" sælger ferierejser til Syden.

Enkeltbillet:	2000 kr
Gruppebillet:	1500 kr

Lav et program, der har følgende udskrift:

```
DE SKAL BETALE 2000 KR FOR DERES BILLET.
eller
HVER PERSON I GRUPPEN SKAL BETALE 1500 KR PR BILLET.
```

Brug en sætning som IF - THEN - ELSE - ENDIF

Begynd således:

```
0010 // SOLOPGAVE
0020 INPUT "INDTAST ANTALLET AF PERSONER: " : antal
0030 IF antal=1 THEN
```

Du kan nu selv fortsætte med programmet, indtil det virker tilfredsstillende. Til sidst kan du skrive LOAD "SOL" og sammenligne programmerne.

Opgave 8.6

I andengradsligningen:

$$Ax^2 + Bx + C = 0$$

er diskriminanten $= B^2 - 4 * A * C$.

Som du kan se nedenfor, er der tre muligheder for diskriminansens fortegn:

1. Hvis diskriminanten < 0 , er der ingen reelle rødder.

2. Hvis diskriminanten $= 0$, er der dobbeltroden: $\frac{-B}{2A}$

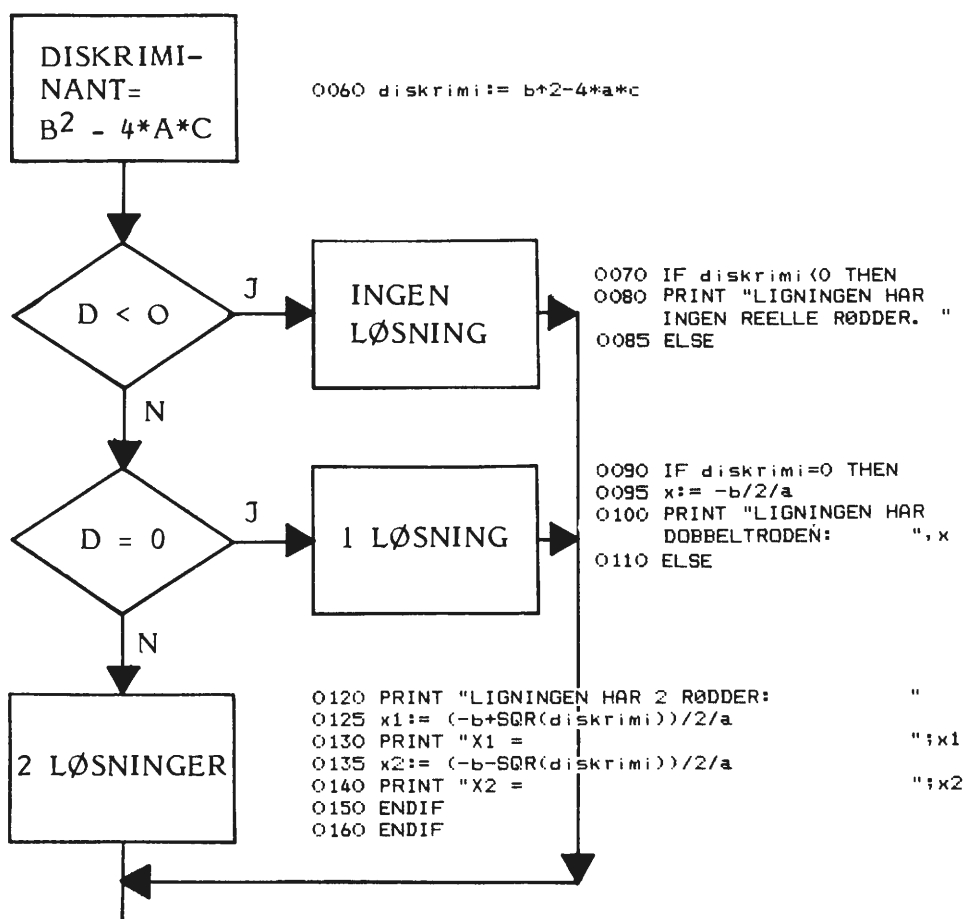
3. Hvis diskriminanten > 0 , er der to rødder:

$$x_1 = \frac{-B + \sqrt{D}}{2A} \quad \text{og} \quad x_2 = \frac{-B - \sqrt{D}}{2A}$$

Kvadratroden af f.eks. 16 skrives som $\text{SQR}(16)$. Se side 68

Lav et program, der giver løsningen, når man indtaster A, B og C, men se først nedenunder.

Her ser du den del af programmet, hvor man undersøger diskriminansens fortegn.



Lav nu selv programmet færdigt, og når det virker tilfredsstillende, kan du tage en udskrift af det på printeren. Derefter skriver du LOAD "ANDENGR" og sammenligner programmerne.

Opgave 8.7

Når du har lært at løse en andengradsligning, har du sikkert også lært at finde en parabels toppunkt ud fra ligningen:

$$y = Ax^2 + Bx + C.$$

$$\text{Toppunktets koordinater: } (x,y) = \left(\frac{-B}{2A}, \frac{-D}{4A} \right)$$

Du skal nu lave et program, der udskriver en parabels toppunkt, når man indtaster A, B & C.

Benyt samme fremgangsmåde som vist i den foregående opgave. Når du er færdig med dit program, og det virker, som det skal, kan du tage en udskrift af det på printeren.

Skriv derefter LOAD "TOPPKT", og sammenlign programmerne.

Opgave 8.8

Udskriv et program, der beregner rumfanget af:

1. Kasse
2. Pyramide
3. Pyramidestub
4. Cylinder
5. Kegle
6. Keglestub
7. Kugle

Man kan bruge IF - THEN - ELSE - ENDIF, idet if-sætningerne kan ligge inde i hinanden.

Skriv LOAD "RUMFANG2", derefter LIST og sammenlign programmerne. Prøvekør begge programmer indenfor hvert rumfang.

Opgave 8.9

Udskriv et program, der beregner den totale overflade af følgende legemer:

1. Kasse
2. Cylinder
3. Kegle
4. Keglestub
5. Kugle

Når du har prøvekørt dit eget program, skriv LOAD "OVERFL" og LIST. Sammenlign programmerne.

Også i næste øvelse har du brug for at sætte flere IF . . . THEN - ELSE - ENDIF-sætninger inden i hinanden.

Opgave 8.10

Posttakster.

Posttakster til indlandet gældende fra 17.maj 1982.

TIL INDLANDET

Breve	indtil	20 g	2,00 kr.
	over 20 g indtil	100 g	2,70 kr.
	over 100 g indtil	250 g	4,00 kr.
	over 250 g indtil	500 g	7,00 kr.
	over 500 g indtil	1 kg	10,00 kr.

Skriv et program, der som inddata bruger brevets vægt, og som udskriver portoen i øre.

Såfremt der gælder andre posttakster, - og det er jo meget tænkeligt -, så hent en ny folder på det nærmeste postkontor. (Et program kan hentes fra disketten: LOAD "POSTTAKST").

STANDARDFUNKTIONER

Standard funktioner

Til ethvert programmeringssprog hører nogle små programmer, som man kan kalde fra sine egne programmer. De kaldes **standardfunktioner**.

Vi vil i dette kapitel se på nogle af dem.

SQR

Med ordren SQR(A) beregnes en tilnærmet værdi for $\sqrt{A}$. En programlinie kan f.eks. se således ud:

```
0030 b:= SQR(a)
```

B tildeles nu værdien af kvadratroden af A. I almindeligt sprog ville vi skrive:

$$B = \sqrt{A}$$

SQR kommer af det engelske ord SquareRoot, som betyder kvadratrode.

Øvelse F

Skriv et kort program, som beregner kvadratroden af et tal. Brug INPUT.

INT

INT kommer af ordet INTeger, som betyder heltalsdel. INT-funktionen runder ned til nærmeste hele tal.

Øvelse G

Indtast og afprøv følgende program:

```
0010 b:= 50
0020 INPUT a
0030 c:= b/a
0040 d:= INT(b/a)
0050 PRINT c
0060 PRINT d
```

Prøv programmet med forskellige værdier af A. Hvad er forskellen på værdierne C og D?_____

Øvelse H

Prøv om du kan finde ud af, om INT-funktionen kan bruges til at afrunde decimaltal til heltal, således at f.eks. 4.6 bliver afrundet til 5, og ikke til 4.

Øvelse I

Det kan være praktisk at bruge INT-funktionen, hvis man skal undersøge, om to tal går op i hinanden.

Det kan gøres således:

```
0010 INPUT a,b
0020 c:= b/a
0030 d:= INT(b/a)
0040 IF c=d THEN
0050   PRINT a;" GÅR OP I "b
0060 ELSE
0070   PRINT a;" GÅR IKKE OP I "b
0080 ENDIF
```

I linie 40 spørges, om C er lig med D. D er heltalsdelen af resultatet af divisionen B/A. Hvis der **ikke** er nogle decimaler at smide væk, så er C og D lige store. Såfremt A går op i B, gælder altså: C = D.

Programlinierne 20, 30 og 40 kan faktisk erstattes af 1 linie, således:

```
0020 IF b/a=INT(b/a) THEN
```

INT-funktionen kan anvendes til at foretage en korrekt afrunding til f.eks. 2 decimaler.

Øvelse J

Prøv følgende lille program:

```
0010 INPUT a
0020 b:= (INT(a*100+0.5))/100
0030 PRINT a;" AFRUNDET TIL 2 DECIMALER: "b
```

Afprøv programmet med tal som	56.6789
	0.2348
	6.655

Kig en gang på linie 20.
Hvorfor adderes 0.5?

RND

RND er hentet fra det engelske ord RaNDom, som betyder tilfældig. Da man i mange tilfælde anvender datamaten til at "simulere" eller efterligne virkeligheden, har man brug for tilfældige tal, -f.eks. når man skal regne på sandsynligheder.

Vi vil prøve at få maskinen til at slå plat eller krone. Det kan

gøres på den måde, at maskinen vælger et tilfældigt tal mellem 0 og 1.

Hvis tallet er mindre end 0.5, får vi plat.

Hvis tallet er større end eller lig med 0.5 får vi krone.

Et program kan se således ud:

```
0010 RANDOMIZE // DENNE SÆTNING ER NØDVENDIG
0020 t:= RND
0030 IF t<0.5 THEN
0040   PRINT "PLAT"
0050 ELSE
0060   PRINT "KRONE"
0070 ENDIF
```

I linie 20

```
0020 t:= RND
```

vælger maskinen tilfældigt et tal mellem 0 og 1.

Hvis vi skal bruge et tilfældigt tal mellem 1 og 100 gør vi sådan:

```
0020 t:= INT(RND*100)+1
```

Hvis vi skal bruge et tilfældigt tal mellem 20 og 50 gøres det sådan:

```
0020 t:= INT(RND*31)+20
```

OVERSIGT

SQR(a)	udregner kvadratroden af tallet A
---------------	-----------------------------------

INT(a)	giver os værdien af tallet A uden decimaler
---------------	---

RND	giver et tilfældigt tal mellem 0 og 1
------------	---------------------------------------

RANDOMIZE	begynder ved et tilfældigt tal i en tilfældig række af tal
------------------	--

OPGAVER

Løs opgaverne ved at anvende SQR - INT - RND

Opgave 8.11

En formel, som kan anvendes til beregning af trekantens areal, når trekantens sider er kendt, ser således ud:

Herons formel. $T = \sqrt{s(s-a)(s-b)(s-c)}$ $s = \frac{1}{2}*(a + b + c)$

s er halvdelen af trekantens omkreds.

Skriv et program, der beregner arealet af en trekant med siderne a, b og c.

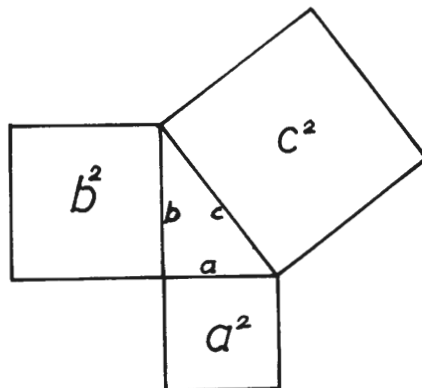
Et program kan hentes fra disketten: LOAD "HERON".

Opgave 8.12

Phytagoras sætningen.

For en retvinklet trekant, hvor c er hypotenusen, og a og b er kateter, gælder følgende:

$$c = \sqrt{a^2 + b^2}$$



Skriv et program, der med a og b som inddata beregner c.
Skriv også et program, der med b og c som inddata beregner a.
Færdige programmer til disse to opgaver kan findes på disketten under navnene "HYP" og "KATETE".

Opgave 8.13

Et år er lig med 365.2425 døgn, hvert på 24 timer. Derfor er man nødt til at regulere antallet af dage fra år til år. Det kan man gøre ved at indføre SKUDÅR hvert fjerde år. Altså har hvert fjerde år 366 dage og resten 365 dage. Det passer imidlertid ikke helt; derfor er man blevet enige om, at når årstallet er deleligt med 100, skal det alligevel ikke være skudår til trods for, at 4 jo går op i 100. Det passer imidlertid heller ikke helt; derfor er man nødt til at gøre årstal, som er delelige med 400, til skudår alligevel.

Prøv nu at skrive et program, som over en periode på 400 år beregner, om det hele stemmer. Prøv derefter at lave et program, som kan afgøre, om et givet år er et skudår.

Programmet kan hentes fra disketten: LOAD "SKUDÅR".

Opgave 8.14

Skriv et program, som kan simulere (efterligne) terningkast.
LOAD "TERNING", hvis du vil sammenligne med dit program.

KAPITEL 9 PROGRAMMERING 3

LØKKER

Program m.

Vi har indtil nu beskæftiget os med programmer med IF - THEN - ELSE - ENDIF, hvor programmet kun bliver gennemløbet én gang. Vi vil på de næste sider beskæftige os med programmer, der bliver gennemløbet mere end en gang:

1. FOR - NEXT
2. WHILE - ENDWHILE
3. REPEAT - UNTIL

FOR-NEXT

FOR - NEXT - LØKKER

FOR - NEXT - løkker anvendes, når en programdel skal gennemløbes et bestemt antal gange.

Øvelse A

Skriv følgende program og prøvekør det.

```
0010 FOR a:= 0 TO 200 STEP 2 DO
0020   PRINT a;
0030 NEXT a
```

a kaldes tælleren
0 kaldes begyndelsesværdien
200 kaldes slutværdien
2 kaldes step-længden

1. Først sættes $a = 0$, og dette tal udskrives på skærmen (linie 0020). Derefter går programmet ned til linie 0030, hvor det får besked om at gå videre til det næste A.
2. Dernæst sættes $a = 2$, idet programmet i linie 0010 har fået besked om at springe 2 trin af gangen. Dernæst bliver værdien 2 udskrevet i linie 0020, og programmet går videre til 0030, hvor det får besked om at returnere til 0010.
3. Dernæst sættes $a = 4$, og sådan fortsætter programmet, indtil slutværdien på 200 er udskrevet.

En FOR - NEXT - sætning består af:

FOR-NEXT	FOR	tæller	:=	begyndelsesværdi	TO
		slutværdi	STEP	trinværdi	DO
		sætningsliste			
		NEXT	tæller		

Øvelse B

Skriv nu det samme program som i øvelse A, idet du erstatter linie 0020 med:

```
0020 PRINT a;
```

Skriv derefter RUN og tryk på RETURN.

Hvordan bliver tallene nu udskrevet? _____

Hvad bevirker ordren PRINT a; ? _____

Øvelse C

Skriv det samme program som i foregående øvelse, idet du nu erstatter linie 0020 med:

```
0020 PRINT a,
```

Skriv derefter RUN og tryk på RETURN.

Hvordan bliver udskrivningen nu? _____

Hvad bevirker ordren PRINT a, ? _____

Øvelse D

Tilføj denne linie til ovenstående program:

```
0015 ZONE=10
```

Skriv RUN og Tryk RETURN.

Hvad bevirker ordren ZONE 10? _____

OVERSIGT

PRINT a	Tallene bliver udskrevet med et tal på hver linie.
----------------	--

PRINT a;	Tallene bliver udskrevet med et enkelt blanktegn imellem.
-----------------	---

PRINT a,	Tallene bliver udskrevet uden mellemrum. Idet ZONE automatisk antager værdien 0 ved start.
-----------------	--

ZONE N	Tallene udskrives kolonnevis. N angiver, hvor mange positioner, der er i hver kolonne.
---------------	--

Øvelse E

Undersøg, hvad der sker, hvis STEP-værdien sættes til 0, ved at køre følgende program:

```
0010 FOR i:= 1 TO 10 STEP 0 DO
0020   PRINT "DET VAR "i;" GANG."
0030 NEXT i
```

I næste eksempel anvendes negative tal i FOR sætningen.

Øvelse F

```
0010 sum:= 0
0020 FOR tal:= 5 TO -5 STEP -1 DO
0030   sum:= sum+tal
0040   PRINT sum:
0050 NEXT tal
```

Tast programmet ind og prøv det.

Det skulle gerne give følgende udskrift:

5 9 12 14 15 15 14 12 9 5 0

Startværdi, slutværdi og stepværdi kan angives ved variable.

Prøv:

```
0005 sum:= 0
0010 INPUT start,slut,værdi
0020 FOR tal:= start TO slut STEP værdi DO
0030   sum:= sum+tal
0040   PRINT sum:
0050 NEXT tal
```

Anvend følgende værdier for start, slut og værdi: 15, -5, -2.

Skriv udskriften her: _____

Øvelse G

Skriv 5-tabellen til 1000.

Begynd således:

```
0010 // 5-tabel
0020 FOR a:= 0 TO 1000 STEP 5 DO
etc.
```

Hvor intet er angivet efter slutværdien, er stepværdien 1. d.v.s.

```
0020 FOR a:= 0 TO 1000 DO
0020 FOR a:= 0 TO 1000 STEP 1 DO
```

har samme værdi.

STEP

I en FOR - NEXT - sætning kan man efter slutværdien skrive STEP trinværdi, d.v.s. hvor mange skridt, man skal gå hver gang, efterfulgt af DO

OPGAVER

Opgave 9.1

Skriv et program, som beregner gennemsnit af f.eks. 10 tal. Brug INPUT samt FOR ... NEXT.

Et eksempel på, hvordan programmet kan se ud, findes på disketten: LOAD "GNSNIT".

Opgave 9.2

Jensen køber en bil på afbetaling. Restbeløbet (f.eks. 35.000 kr.) betales over f.eks. 48 måneder til en rente på f.eks. 22% p.a. Lånet betales på den måde, at der hver måned betales LIGE STORE AFDRAG + 1 måneds rente af restbeløbet. Det betyder, at den samlede ydelse - afdrag + rente - bliver mindre og mindre for hver måned.

Jensen skriver nu et program, som kan fortælle ham, hvor meget han skal betale måned for måned. Ud fra følgende plan skal der skrives et program, som kan løse problemet.

1. INDLÆS LÅNEBELØB, PROCENT OG ANTAL MDR.
2. BEREGN AFDRAGET
3. UDFØR FØLGENDE SÅ MANGE GANGE, SOM ANTALLET AF MDR. ANGIVER:
 - BEREGN RENTE AF RESTGÆLDEN
 - SÆT YDELSEN = AFDRAG + RENTE
 - BEREGN NY RESTGÆLD
 - SKRIV YDELSE, RENTE, AFDRAG OG RESTGÆLD

Prøvekør derefter programmet, og få evt. en udskrift på printeren.

Udbyg derefter programmet, så der desuden beregnes den samlede rente i kr. for hele afbetalingshandelen.

Programeksempel på disketten: LOAD "AFBETALING".

Opgave 9.3

Skriv et program, som simulerer, at vi slår plat eller krone 100 gange. Programmet skal som udskrift give antallet af "plat" og antallet af "krone".

Programeksempel på disketten: LOAD "PLAT".

Opgave 9.4

Skriv et program, som simulerer, at en terning kastes 600 gange. Programmet skal undersøge og udskrive, hvor mange af de 600 kast der er 6-ere.

Programeksempel på disketten: LOAD "TKAST".

Opgave 9.5

En kapital på f.eks. 1000 kr. indsættes i en bank til 15% i 10 terminer. Hvor meget er kapitalen vokset til i løbet af de 10 terminer?

Programmet kunne se således ud:

```
0005 PRINT CHR$(12)
0010 // OPSPARING
0020 INPUT "INDTAST KAPITAL: " : kapital
0030 PRINT
0040 INPUT "RENTEFOD: " : rentefod
0050 PRINT
0060 INPUT "INDTAST ANTALLET AF TERMINER: " : termin
0070 PRINT
0080 sum:= INT(kapital*((1+rentefod/100)^termin)*100)/100
0090 PRINT
0100 PRINT "KAPITALEN ER: " : kapital
0110 PRINT "RENTEFODEN ER: " : rentefod
0120 PRINT "ANTALLET AF TERMINER: " : termin
0130 PRINT "KAPITALEN ER VOKSET TIL: " : sum
```

Erstat nu linierne 70, 80 og 90 med en FOR - NEXT-sætning, der udregner den opsparede kapital. Ellers er programmet nøjagtigt det samme.

Når du er færdig, skriv RUN, og gennemkør programmet et par gange. Skriv derefter LIST "printer", så at du får dit eget program udskrevet på printeren.

Hvis du vil se det samme program med FOR - NEXT, skal du skrive LOAD "SPAROP", LIST og sammenligne programmerne.

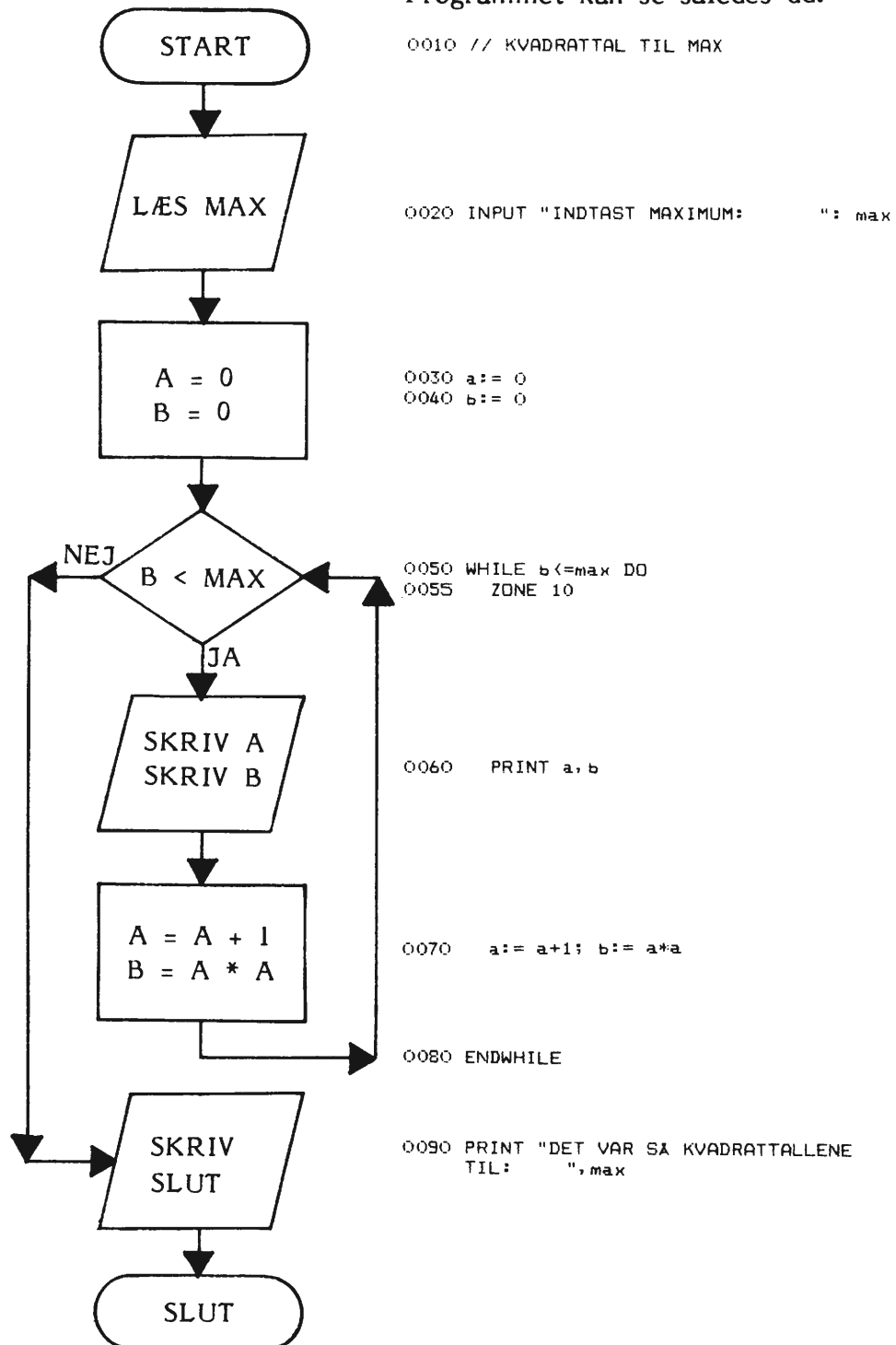
WHILE - ENDWHILE - løkker

Medens vi i de foregående opgaver havde en løkke, der skulle gentages et bestemt antal gange, har vi i denne sætning en løkke, hvor der i begyndelsen er indlagt en betingelse.

Lad os se nærmere på nedenstående opgave:

Programeksempel 1

Programmet kan se således ud:



Ovenstående opgave findes på disketten under navnet KVAD-

RAT. Skriv LOAD "KVADRAT", derefter RUN, og udfyld nedenstående skema:

Eks. MAXIMUM = 200

A	0	1	2								
B	0	1	4								

Programeksempel 2

Indlæs 10 tal, og find deres middeltal.

WHILE . . . DO
.
.
.
ENDWHILE

```
0005 sum:= 0
0010 tæller:= 1
0020 WHILE tæller<=10 DO
0030   INPUT tal
0040   sum:= sum+tal
0050   tæller:= tæller+1
0060 ENDWHILE
0070 middeltal:= sum/(tæller-1)
0080 PRINT middeltal
```

Anvend følgende data: 483, 817, 56, 345, 87, 23, 78, 346, 80, 234

I linie 0020 WHILE tæller<=10 DO sker følgende:

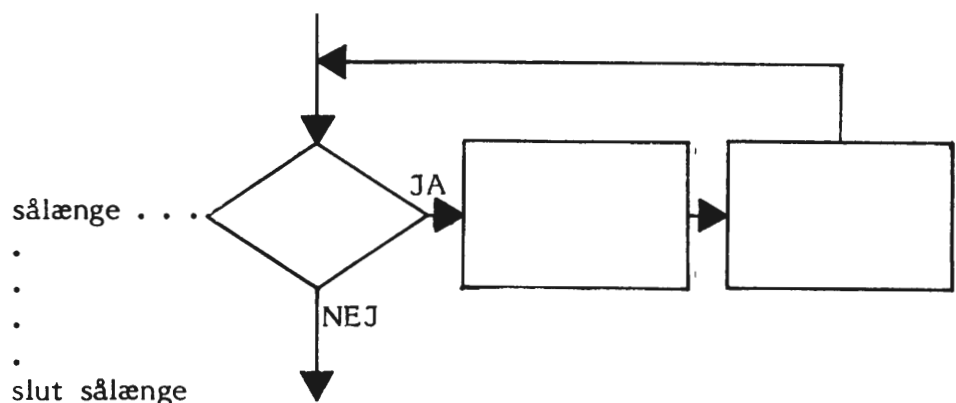
Så længe tælleren er mindre end eller lig med 10, udføres linierne 30-50.

Når tælleren overstiger 10, springes til den linie, som følger umiddelbart efter ENDWHILE.

Indtast programmet, og efterprøv det. Anvend eventuelt andre DATA.

WHILE . . DO WHILE - ENDWHILE-sætninger benyttes, når en række sætninger skal udføres, så længe en betingelse er opfyldt

Rutediagrammet for denne del af opgaven kan se således ud:



En struktur som den her viste kan i COMAL klares med:

```
WHILE . . . DO
.
.
.
ENDWHILE
```

Opbygning af en WHILE - ENDWHILE-sætning.

```
WHILE   logisk udtryk   DO
sætningsliste
ENDWHILE
```

Betingelsen for udførelsen af WHILE - ENDWHILE - løkken ligger i begyndelsen af programblokken.

OPGAVER

Opgave 9.6

Skriv et program, som beregner kubiktallene til og med 1.000.000 (100^3). Altså de første 100 kubiktal.

Programeksempel på disketten: LOAD "KUBIKT".

Opgave 9.7

Skriv 7-tabellen til et vist maximum ved hjælp af en WHILE -ENDWHILE-sætning.

Begynd således:

```
0010 // 7- TABEL
0020 a:= 0
0030 INPUT "INDTAST MAXIMUM:      ": max
0040 WHILE a<=max DO
etc.
```

Gør selv programmet færdigt, indtil det virker tilfredsstillende. Skriv derefter LOAD "TABEL7", og sammenlign programmerne.

Opgave 9.8

Bakterien COMALUS er sygdomsfremkaldende. Sygdommen, som hedder DATAMITIS, kan bekæmpes med en bestemt medicin. Anvendes denne medicin i normal dosis, halveres antallet af bakterier i blodet på et døgn. Så længe antallet af bakterier er større end 1 pr. cm^3 blod, er man syg. Først når den daglige blodprøve viser, at antallet af bakterier pr. cm^3 er 1 eller

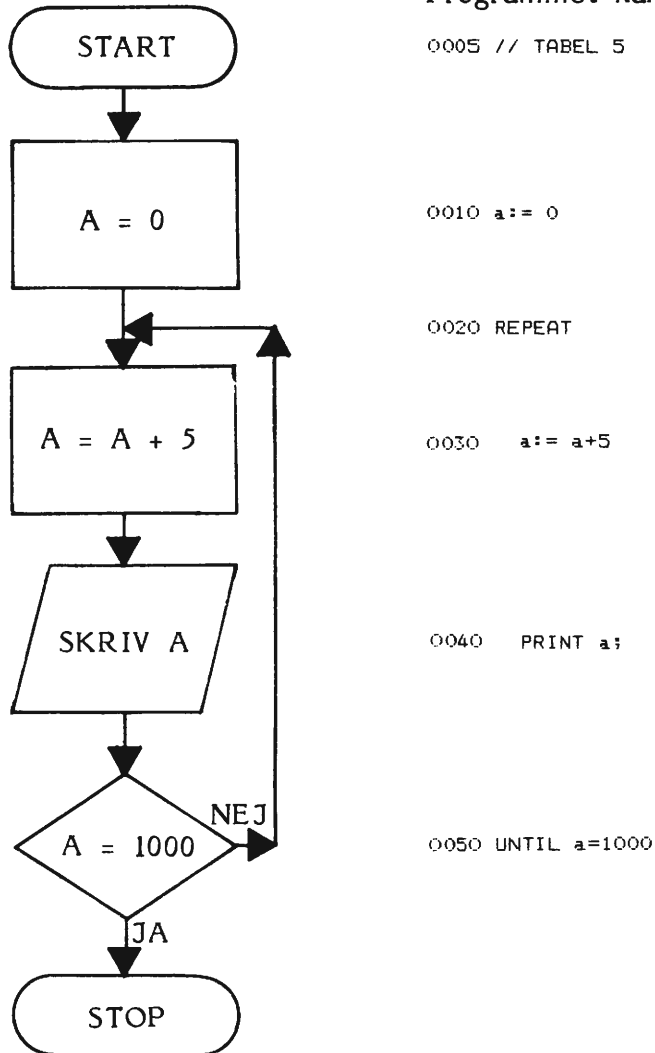
REPEAT - UNTIL løkker

Vi skal nu lave et program ved hjælp af en ny COMAL-ordre: REPEAT - UNTIL.

Program eksemp 1

Programmet skal udskrive 5-tabellen indtil 1000.

Programmet kan se således ud:



Prøvekør programmet.

Program eksemp 2

Vi vil undersøge, hvor mange divisorer et givet tal har. Vi kan først prøve alle de tal, som ligger mellem tallet 1 og det tal, som vi vil undersøge.

Et program kan se således ud:

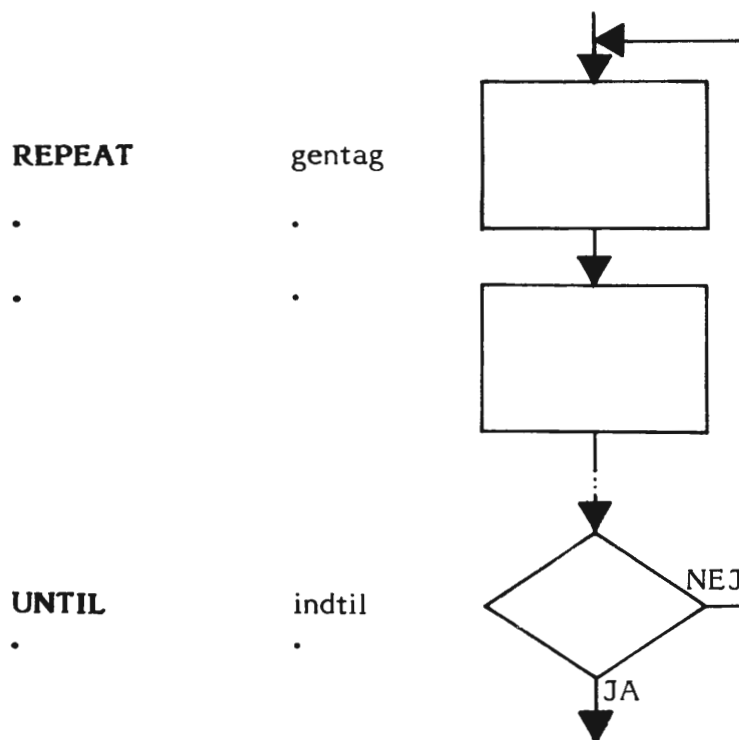
```
0010 INPUT tal
0020 divisor:= 1
0030 REPEAT
0040 IF tal/divisor=INT(tal/divisor) THEN
0050 PRINT divisor
0060 ENDIF
0070 divisor:= divisor+1
0080 UNTIL divisor>tal
```

Linierne 30 til 80 gentages, indtil betingelsen "divisor er større end tal" er opfyldt.

Når data skal behandles, kan vi komme ud for, at en bestemt programdel skal gentages, indtil en betingelse er opfyldt.

REPEAT - UNTIL - REPEAT - UNTIL-sætninger benyttes, når en række sætninger skal udføres, indtil en betingelse er opfyldt

Rutediagrammet for denne del af opgaven kan se således ud:



En struktur som den her viste kan i COMAL klares med følgende programblok:

REPEAT
sætningsliste
UNTIL logisk udtryk

Betingelsen for udførelsen af REPEAT - UNTIL-løkken ligger i slutningen af programblokken.

OPGAVER

Opgave 9.9

I programeksempel 2 (om divisorer) prøvede vi alle tal mellem 1 og det tal, som skulle undersøges. Det er ikke nødvendigt.

Skriv et program, som er hurtigere end programeksempel 2.

Et færdigt program på disketten: LOAD "DIVISOR".

Opgave 9.10

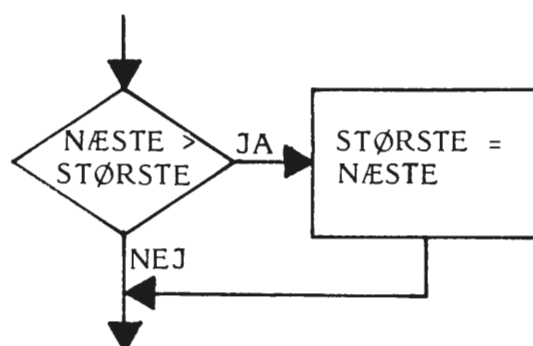
Datamater anvendes i udstrakt grad til løsning af opgaver, hvor data skal **sorteres**.

Vi vil skrive et program, som kan få datamaten til at udskrive det største blandt f.eks. 10 tal. Maskinen skal altså gemme det største tal. Hver gang et nyt tal indtastes, skal maskinen sammenligne det med det foregående. Hvis det nye tal er størst, skal det huskes og det gamle slettes.

Der skal ske følgende:

```
INDLÆS FØRSTE TAL  
SÆT STØRSTE = FØRSTE  
GENTAG  
  INDLÆS NÆSTE TAL  
  HVIS "NÆSTE" TAL ER STØRRE END "STØRSTE" TAL  
  SÆT "STØRSTE" = "NÆSTE"  
  INDTLIL ALLE TAL ER UNDERSØGT
```

En vigtig detalje er tildelingen af "største" tal. Den kan beskrives ved følgende lille rutediagramstump:



Hvis "næste" tal er større end "største", flytter værdien af "næste" over og erstatter den gamle værdi i den celle, som vi har kaldt "største".

På denne måde vil det største tal hele tiden være placeret i den celle, som vi kalder "største".

Et program kan se således ud:

```
0010 INPUT "ANTAL:                ": antal
0020 INPUT "FØRSTE TAL:           ": første
0030 største:= første
0040 tæller:= 1
0050 REPEAT
0060     INPUT "NÆSTE TAL:         ": næste
0070     IF næste>største THEN
0080         største:= næste
0090     ENDIF
0100     tæller:= tæller+1
0110 UNTIL tæller=antal
0120 PRINT "AF DE "antal;" TAL ER "største;" DET STØRSTE."
```

Indtast programmet, og tryk RUN.

- Opgave 9.11** Prøv at ændre programmet, så det i stedet for at finde det største blandt tallene finder det mindste.
- De to programmer findes på disketten under navnene "STØRST" og "MINDST".
- Opgave 9.12** Skriv 25-tabellen ved hjælp af et Comal-program med en REPEAT - UNTIL-sætning.
- Når du er færdig, og dit program virker, kan du skrive LOAD "IGEN25", LIST, og sammenligne programmerne.
- Opgave 9.13** Skriv et program med REPEAT - UNTIL-sætning, der viser, hvor længe et indskud i en bank er om at vokse til et vist slutbeløb ved en fast årlig rente på 12%.
- Begynd således:
- ```
0010 PRINT CHR$(12)
0020 termin:= 0
0030 INPUT "INDTAST INDSKUD: ": indskud
0040 INPUT "INDTAST SLUTBELØB: ": slutbeløb
0050 REPEAT
etc.
```
- I REPEAT - UNTIL-sætningen skal der være en linie, der opdaterer antallet af terminer, f.eks. TERMIN = TERMIN + 1
- Gør selv programmet færdigt.
- Hvis du har problemer med at skrive programmet, kan du få hjælp ved at hente det tilsvarende program frem på skærmen.
- Du skriver LOAD "VOKSE", LIST. Du kan nu skrive dit eget program færdigt ved at sammenligne. Du bør til sidst afprøve programmet på maskinen.
- Opgave 9.14** Et moderne kasseapparat kan tilsluttes en datamat, som kan holde styr på, hvilke varer der sælges og i hvor stort omfang.

Kasseapparatet kan også give de rigtige penge tilbage. Det sidste vil vi se på i et lille program.

INDDATA er et beløb i øre. Som UDDATA skrives det pågældende beløb i kr. og øre.

```
0005 kroner:= 0
0010 INPUT "INDTAST BELØB I ØRE: ": beløb
0020 b:= beløb
0030 REPEAT
0040 b:= b-100
0050 kroner:= kroner+1
0060 UNTIL b<100
0070 PRINT
0080 PRINT beløb;"ØRE = ";kroner;"KR. OG ";b;"ØRE."
```

I løkken fra 30 til 60 trækkes 100 fra BELØBet, indtil beløbet er mindre end 100. Det antal gange, 100 kan trækkes fra beløbet i øre, er lig med det antal kroner, beløbet kan veksles om til. Derfor er der indbygget en tæller, som lægger 1 KRONE til for hvert gennemløb.

Indtast programmet og prøvekør det.

Hvad sker der, hvis antallet af ører er mindre end 100? Prøv med f.eks. 78 øre

Skift programmet ud med følgende program, og prøv igen

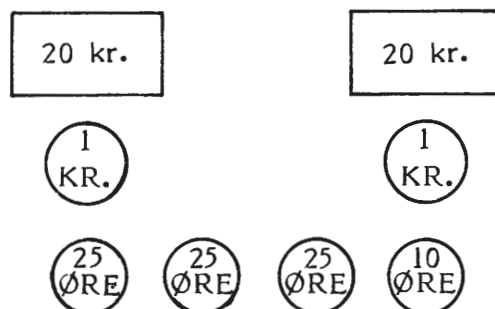
```
0010 kroner:= 0
0020 INPUT "INDTAST BELØB I ØRE: ": beløb
0030 b:= beløb
0040 WHILE b>=100 DO
0050 b:= b-100
0060 kroner:= kroner+1
0065 PRINT
0070 ENDWHILE
0075 PRINT
0080 PRINT beløb;"ØRE = ";kroner;"KR. OG ";b;"ØRE."
```

Disse eksempler viser forskellen på anvendelse af REPEAT - UNTIL og WHILE - ENDWHILE.

### Opgave 9.15

Dette program kan udbygges til at simulere (efterligne) et kasseapparat, som giver penge tilbage. Det kan gøres således, at et beløb betales i så få sedler og mønter som muligt.

F.eks. 4287 øre:



Skriv et program, som betaler beløb op til 100 kr. (10000 øre) med så få mønter og sedler som muligt.

Et færdigt program findes på diskettes: LOAD "PENGE".

## PROGRAMMER MED STØRRE DATAMÆNGDER

Indtil nu har vi beskæftiget os med INPUT-sætningerne, hvor mængden af data er ret begrænset.

### READ-DATA

Hvor det drejer sig om mange inddata, vil en READ - DATA -sætning være ganske bekvem til indlæsning af data.

#### Programeksempel

I nedenstående opgave skal vi på grundlag af

INDDATA:

1. Arbejdersnummer
2. Ugetimetal
3. Timeløn

have udskrevet:

1. Arbejdersnummer
2. Ugetimetal
3. Timeløn
4. Ugeløn

| INDDATA                                  | BEREGNING                      | UDDATA                                             |
|------------------------------------------|--------------------------------|----------------------------------------------------|
| Arbejdersnummer<br>Ugetimetal<br>Timeløn | Ugeløn =<br>Ugetimetal*Timeløn | Arbejdersnummer<br>Ugetimetal<br>Timeløn<br>Ugeløn |

```

0010 // DETTE ER ET READ - DATA PROGRAM
0020 // DER ER OPGIVET ARBEJDERNUMMER, UGETIMETAL & TIMELØN
0030 PRINT CHR$(12) "LØN"
0040 FOR i:= 1 TO 3 DO
0050 READ nummer, antal, timeløn
0060 PRINT "ARBEJDERNUMMER: ";nummer
0070 PRINT "ANTAL TIMER I UGEN: ";antal
0080 PRINT "TIMELØN: ";timeløn
0090 ugeløn:= antal*timeløn
0100 PRINT "UGELØN: ";ugeløn
0110 PRINT
0120 NEXT i
0130 PRINT "SÅ ER DER IKKE FLERE, DER SKAL HAVE LØN."
0140 DATA 1,40,45
0150 DATA 2,40,56
0160 DATA 3,38,68

```

## Øvelse H

Se nøje på programmet på forrige side og besvar følgende spørgsmål:

I hvilken linie står READ-sætningen? \_\_\_\_\_

I hvilken linie står de tre første tal, som indlæses i de variable størrelser NUMMER, ANTAL, TIMELØN? \_\_\_\_\_

I hvilken linie vender programmet tilbage til READ-sætningen efter at have læst de 3 første tal? \_\_\_\_\_

Skriv LOAD "ARB" og derefter RUN.

## Øvelse I

Hvordan bliver uddata for arbejder nummer 3:

---

---

---

---

Når et program, der indeholder READ - og DATA-sætninger, skal udføres, vil datamaten danne et DATA-register, hvori alle DATA-instruktionernes indhold stilles op på en lang række. Når programmet starter, vil der blive placeret en "pil" ved den første værdi i DATA-rækken.

Når programmet når en READ-sætning, vil den første værdi blive læst, og "pilen" flyttes nu automatisk til næste værdi. Hver værdi bliver læst én og kun én gang.

## OVERSIGT

### READ

Ved ordren READ gives der besked om at indlæse det tal i DATA-sætningen, som "pilen" peger på. Ordren består af: READ med en liste af variable, adskilt af komma.

### DATA

DATA-sætningen kan stå et vilkårligt sted i programmet. Hvis der er flere DATA-sætninger, bliver de indlæst i linienummerorden. Ordren består af: DATA med en liste af tal, adskilt af komma.

## OPGAVER

### Opgave 9.16

Skriv et program, med READ - DATA, der giver følgende udskrift:

| Elevnummer | Skr. matematik | Mdt. Matematik | Orden |
|------------|----------------|----------------|-------|
| 1          | 8              | 9              | 8     |
| 2          | 10             | 10             | 9     |
| 3          | 6              | 5              | 7     |
| 4          | 13             | 10             | 10    |

Begynd således:

```
0010 PRINT CHR$(12)
0020 PRINT "ELEVNUMMER:";TAB(20);"SKR. MATEMATIK:";
0030 PRINT TAB(40);"MDT. MATEMATIK:";TAB(60);"ORDEN:"
0040 FOR a:= 1 TO 4 DO
0050 READ nummer,skr,mdt,orden
```

Gør selv programmet færdigt, idet du benytter de forskellige tal fra ovenstående skema i Data-sætningerne.

Skriv derefter LOAD "ELEV4", og LIST. Sammenlign programmerne.

|                |                                                                         |
|----------------|-------------------------------------------------------------------------|
| <b>TAB(20)</b> | bevirker, at datamaskinen begynder at skrive fra position 20 på skærmen |
|----------------|-------------------------------------------------------------------------|

Skærmen har 24 linier á 80 positioner i hver linie.

### Opgave 9.17

Skriv et program, der beregner kursværdi, når man har opgivet pålydende værdi og kurs.

| INDDATA           | BEREGNING                           | UDDATA                         |
|-------------------|-------------------------------------|--------------------------------|
| Pålydende<br>Kurs | Kursværdi =<br>Pålydende * Kurs/100 | Pålydende<br>Kurs<br>Kursværdi |

Benyt følgende DATA - sætninger:

```
0100 DATA 2000,114
0110 DATA 5000,430
0120 DATA 600,87.5
```

Begynd således:

```
0010 // PROGRAMMET UDSKRIVER KURSVÆRDI PÅ GRUNDLAG AF
0020 // PÅLYDENDE VÆRDI $ KURS.
0030 PRINT CHR$(12)
0040 FOR i:= 1 TO 3 DO
0050 READ pålydende,kurs
```

Skriv derefter LOAD "KURSVÆR" og RUN. Sammenlign programmerne.

## KAPITEL 10 PROGRAMMERING 4

### PROCEDURER

Det er vigtigt, at især store programmer skrives på en sådan måde, at de er overskuelige. Det skal helst være umiddelbart indlysende, hvordan programmet fungerer.

Hvordan kan man gøre et program på flere hundrede linier overskueligt? Det kan man ved at dele det op i mindre programblokke, som så styres fra et hovedprogram. Man laver en oversigt over, i hvilken rækkefølge de enkelte programafsnit skal udføres. Derefter udarbejder man så de enkelte programafsnit.

Til hjælp med dette arbejde anvendes **procedurer**.

#### Programeksempel X-by varmekær:

Betalingen for varme er delt op i en fast og en variabel afgift. Den faste afgift er afhængig af det antal rummeter ( $m^3$ ), som skal opvarmes. Den variable afgift er afhængig af det antal  $m^3$  varmt vand, som løber igennem systemet i løbet af et år.

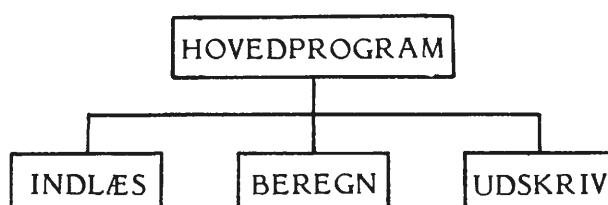
|                           |           |
|---------------------------|-----------|
| 1 $m^3$ rumindhold koster | kr. 2,80  |
| 1 $m^3$ varmt vand koster | kr. 18,50 |
| Fast målerafgift          | kr. 75,00 |
| Moms 22%                  |           |

Varmekærens programmør skal skrive et program, som kan udregne, hvor meget den enkelte forbruger skal betale. For hver forbruger indtastes husets rumindhold i  $m^3$  samt forbruget af vand i  $m^3$ .

Opgaven deles i tre afsnit:

1. Indlæsning
2. Beregning
3. Udskrift

Princippet kan vises med denne struktur:



Til hver af disse dele udarbejdes en programblok. Hver af disse programblokke ser således ud:

PROC

ENDPROC

```

PROC navn
.
.
.
ENDPROC navn

```

PROC er en forkortelse af det engelske ord PROCEDURE, som betyder "fremgangsmåde".

Første programafsnit kaldes

PROC indlæs

Proceduren ser således ud:

```

0050 PROC indlæs
0060 INPUT "INDTAST HUSETS RUMINDHOLD I M3: ": rum
0070 INPUT "INDTAST FORBRUG AF VARME I M3: ": varme
0080 ENDPROC indlæs

```

Andet programafsnit kaldes

PROC beregn

Proceduren ser således ud:

```

0100 PROC beregn
0110 fast:= rum*2.8
0120 variabel:= varme*18.5
0130 ialt:= fast+variabel+75
0140 moms:= ialt*0.22
0150 udgift:= ialt+moms
0160 ENDPROC beregn

```

Tredie programafsnit kaldes

PROC udskriv

Proceduren ser således ud:

```

0180 PROC udskriv
0190 PRINT
0200 PRINT "RUMINDHOLD: ";rum;"M3. - FORBRUG: ";varme;" M3."
0210 PRINT
0220 PRINT "FAST AFGIFT: ", fast
0230 PRINT "VARIABLE AFGIFT: ", variabel
0240 PRINT "IALT - U/MOMS: ", ialt
0250 PRINT "MOMS: ", moms
0260 PRINT "AT BETALE IALT: ";udgift
0270 ENDPROC udskriv

```

For at angive, i hvilken rækkefølge disse tre programmer skal udføres, indsættes i begyndelsen af programmet 3 linier:

```
0010 EXEC indlæs
0020 EXEC beregn
0030 EXEC udskriv
```

## EXEC

EXEC er en forkortelse af ordet EXECUTE, som kan oversættes med UDFØR.

Når datamaten under programudførelsen når til linie,

```
0010 EXEC indlæs
```

udføres programblok **PROC indlæs**, altså linierne 50-80. Efter ordren i linie 80 **ENDPROC indlæs** springes tilbage til den linie, som følger umiddelbart efter **EXEC indlæs**.

I denne linie får maskinen ordren

```
0020 EXEC beregn
```

som betyder, at programblok **PROC beregn** skal udføres. Efter linie 160 **ENDPROC beregn** springes til den linie, som følger umiddelbart efter **EXEC beregn**.

I denne linie får maskinen ordren

```
0030 EXEC udskriv
```

hvorefter programblok **PROC udskriv** udføres. Efter linie 270 **ENDPROC udskriv** stopper programmet.

## Øvelse A

Programmet findes på disketten under navnet "VARME1".

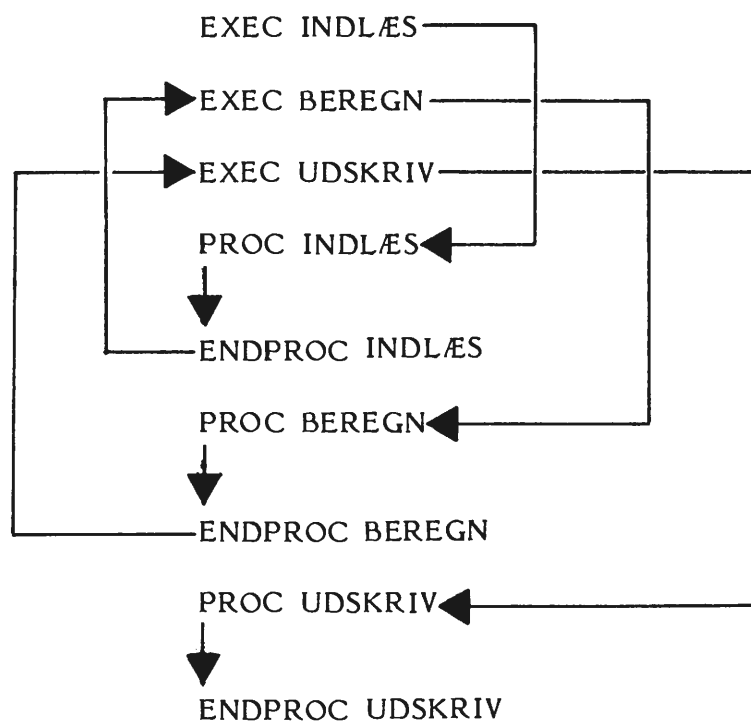
Indlæs programmet fra disketten, og prøv det med forskellige inddata.

Her følger det samlede program.

```
0010 EXEC indlæs
0020 EXEC beregn
0030 EXEC udskriv
0040 PRINT
0050 PROC indlæs
0060 INPUT "INDTAST HUSETS RUMINDHOLD I M3: " : rum
0070 INPUT "INDTAST FORBRUG AF VARME I M3: " : varme
0080 ENDPROC indlæs
0090 PRINT
0100 PROC beregn
0110 fast:= rum*2.8
0120 variabel:= varme*18.5
0130 ialt:= fast+variabel+75
0140 moms:= ialt*0.22
0150 udgift:= ialt+moms
0160 ENDPROC beregn
0170 PRINT
0180 PROC udskriv
0190 PRINT
0200 PRINT "RUMINDHOLD: ";rum;"M3. - FORBRUG: ";varme;" M3."
0210 PRINT
0220 PRINT "FAST AFGIFT: " , fast
0230 PRINT "VARIABLE AFGIFT: " , variabel
0240 PRINT "IALT - U/MOMS: " , ialt
0250 PRINT "MOMS: " , moms
0260 PRINT "AT BETALE IALT: " , udgift
0270 ENDPROC udskriv
```

| IND      |         | UD   |          |      |      |        |
|----------|---------|------|----------|------|------|--------|
| RUMINDH. | FORBRUG | FAST | VARIABEL | IALT | MOMS | UDGIFT |
| 290      | 400     |      |          |      |      |        |
| 220      | 300     |      |          |      |      |        |
| 400      | 700     |      |          |      |      |        |
|          |         |      |          |      |      |        |

Programstrukturen kan anskueliggøres på følgende måde:



## OPGAVER

### Opgave 10.1

Bestyrelsen foreslår, at den faste afgift for huse med stort rumindhold skal nedsættes en del. Det skal ske ved at afgiften pr.  $\text{m}^3$  rum, som er 2.80 kr., nedsættes for den del af rumindholdet, som overstiger 200  $\text{m}^3$ . Herefter skal der betales fast afgift efter følgende regler:

|                              |                           |
|------------------------------|---------------------------|
| De første 200 $\text{m}^3$ : | 2.80 kr. pr. $\text{m}^3$ |
| Resten                       | 0.90 kr. pr. $\text{m}^3$ |

Indbyg denne ændring i programmet fra før. Beregn, hvor stor forskellen i udgift er for et hus med et rumindhold på 300  $\text{m}^3$ .

Programmet findes på disketten under navnet "VARME2".

### Opgave 10.2

Fiskeskipper Olsens mandskab deler overskuddet efter dagens fangst. Fordelingen af overskuddet kan f.eks. se således ud:

|            |      |
|------------|------|
| Skipper    | 40%  |
| Styrmand   | 28%  |
| Kok        | 22%  |
| Skibsdreng | 10%  |
| Ialt       | 100% |

Skriv et program, hvor du anvender PROCEDURER.

Programmet skal bruges til beregning af, hvor meget hver af besætningsmedlemmerne skal have af overskuddet.

Del opgaven op i 3 dele.

1. Indlæs: Som inddata skal angives, hvilken procentsats hver af besætningsmedlemmerne skal have. (Find selv nogle procentsatser og kontroller at summen af dem bliver 100). Desuden skal indlæses overskuddets størrelse.
2. Beregn: Udregn, hvor meget hver af fiskerne skal have.
3. Udskriv: Udskriv, hvor meget hver af fiskerne skal have udbetalt. Udskriv eventuelt også overskuddets størrelse.

Et færdigt program findes på disketten under navnet "FISK".

## KAPITEL 11 PROGRAMMERING 5

### TEKSTVARIABLE

Hidtil har vi kun anvendt tal som variable. Der er ikke noget i vejen for, at vi kan behandle variable, hvori der skal indsættes bogstaver eller andre tegn.

Vi kalder sådanne variable for

|                      |                                                    |
|----------------------|----------------------------------------------------|
| <b>Tekstvariable</b> | En tekstvariabel kan være et bogstav eller et ord. |
|----------------------|----------------------------------------------------|

Tekstvariabelnavne afsluttes med \$-tegnet for at adskille dem fra numeriske variable.

Eksempel på variabelnavne af denne type:

```
navn$
adresse$
svar$
```

Når tekstvariable skal anvendes, skal der i modsætning til alm. variable bestilles plads til hver enkelt. Det skal ske i begyndelsen af programmet.

Hvis vi i et program skal anvende de tre ovenstående tekstvariable, skal vi nu afgøre, hvor mange tegn hver enkelt højst kan bestå af. Det skal fremgå af en DIM sætning.

Eksempel:

**DIM**

```
DIM navn$ OF 15
```

betyder, at hvert navn højst må bestå af 15 bogstaver eller tegn. Eventuelle mellemrum skal tælles med.

```
DIM adresse$ OF 25
```

Hvis adressen fylder mere end 25 tegn, bliver de overskydende tegn ikke opfattet af datamaten.

Hvis vi i SVAR skal have plads til svaret JAVEL, skal vi altså reservere plads til mindst 5 tegn.

Således:

```
DIM svar$ OF 5
```

## Øvelse A

Prøv dette program:

```
0010 DIM navn$ OF 15
0020 INPUT "HVAD HEDDER DU? ": navn$
0030 PRINT "NÅ, SÅ DU HEDDER: ": navn$
```

Prøv at eksperimentere lidt med, hvilke bogstaver og tegn programmet accepterer.

## Øvelse B

Endnu et lille program.

```
0010 DIM svar$ OF 5
0020 INPUT "SYNES DU GODT OM DATALÆRE? ": svar$
0030 IF svar$="JA" THEN
0040 PRINT " DET TÆNKTE VI NOK. "
0050 ELSE
0060 PRINT "DU ER ET UTAKNEMMELIGT SKARN. "
0070 ENDIF
```

## OVERSIGT

Et \$ tegn (dollartegn) efter en variabel, giver datamaten besked på at opfatte variabelen som en tekstvariabel

Med ordren DIM a\$ OF n reserveres plads til n tegn i tekstvariabelen a\$

## OPGAVER

### Opgave 11.1

I kemi benytter man indikatoren lakmus for at finde ud af, om en væske er en syre eller en base. Hvis lakmus farver væsken rød, er det en syre, hvis væsken farves blå, er det en base. Programmet ser således ud:

```
0005 DIM svar$ OF 15
0010 // PAVISNING AF SYREBASE
0020 PRINT "TILSÆT LAKMUS: "
0030 PRINT "HVILKEN FARVE FREMKOMMER? "
0040 INPUT "SKRIV FARVEN: ": farve$
0050 IF farve$="RØD" THEN
0060 PRINT "VÆSKEN ER EN SYRE. "
0070 ELSE
0080 PRINT "VÆSKEN ER EN BASE. "
0090 ENDIF
```

Indtast programmet og afprøv det.

Man kan også komme ud for, at resultatet af lakmusprøven bliver farveløs. Væsken er så hverken en syre eller en base, men en neutral væske.

Du skal nu ændre programmet, så der er mulighed for endnu en udskrift, nemlig:

VÆSKEN ER HVERKEN EN SYRE ELLER EN BASE.

Et færdigt program findes på disketten under navnet "LAKMUS"

### Opgave 11.2

Som du måske kan huske, gennemløbes programmer med REPEAT-UNTIL, indtil en bestemt betingelse er opfyldt.

Eksempel: REPEAT

.  
.  
.

INPUT "ØNSKER DU AT FORTSÆTTE", svar\$  
UNTIL svar\$ = "NEJ"

Denne del af programmet vil gennemløbes, indtil svaret på spørgsmålet er NEJ.

Skriv et program, som efterligner kast med en terning.  
Programmet skal afsluttes med spørgsmålet:

ØNSKER DU AT FORTSÆTTE?

Programmet skal gennemløbes, indtil der svares NEJ.

Husk at bestille plads til svaret. F.eks. DIM svar\$ OF 3

Et færdigt program findes på disketten under navnet "TERN3".

### Opgave 11.3

Prøv følgende program, som findes på disketten.

Skriv LOAD "JUL", og skriv derefter LIST.

I linie 10 reserveres der plads til:

Modtagerens navn, 15 pladser

Afsenderens navn, 15 pladser

Ved strengvariable skal bruges dollartegnet, som man kan se i linierne: 10, 20, 30, 40 og 120.

Skriv derefter RUN, og prøvekør programmet nogle gange med forskellige navne.

### Opgave 11.4

Der ønskes et program med følgende udskrift:

Kære (NAVN)  
(ADR)  
(POSTD)

(DDATO)

De bedes venligst betale (BELØB) inden (DATO), idet vi i modsat tilfælde bliver nødt til at hente vor (VARIABLE) retur.

Med venlig hilsen  
(AFSNAVN)

(AFSADR)  
(AFSPOST)

Følgende forkortelser dækker:

|          |                              |
|----------|------------------------------|
| DDATO    | Dags dato                    |
| NAVN     | Modtagerens navn             |
| ADR      | Modtagerens adresse          |
| POSTD    | Modtagerens postnummer og by |
| BELØB    | Det skyldige beløb           |
| DATO     | Sidste betalingsdato         |
| VARIABEL | Varens art                   |
| AFSNAVN  | Afsenderens navn             |
| AFSADR   | Afsenderens adresse          |
| AFSPOST  | Afsenderens postnummer og by |

Alle ord i parentes skal indtastes via en input-sætning. **Husk** at reservere plads til de forskellige tekstvariable, f.eks.

DIM navn\$ OF 30

Skriv derefter LOAD "RYKKER" og LIST og sammenlign programmerne.

### Opgave 11.5

#### Testopgave

Programmet "TEST3" ligger på disketten. LOAD programmet og svar på spørgsmålene.



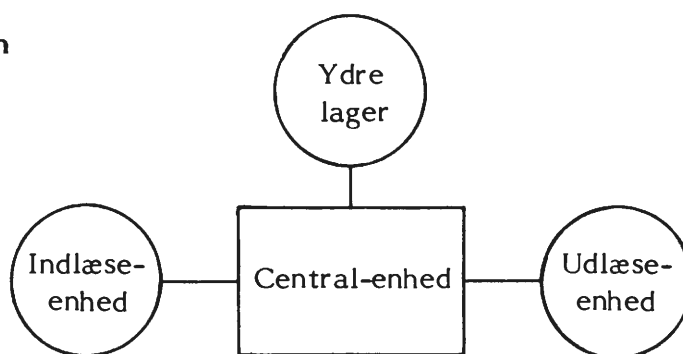
## KAPITEL 12 STØRRE DATA-ANLÆG

Hidtil har vi kun beskæftiget os med mikrodatamater, da det sandsynligvis er dem, du vil træffe på i undervisningen. Men hvis du kommer på besøg i en virksomhed - enten i det private erhvervsliv eller i den offentlige administration - vil du få langt større dataanlæg at se.

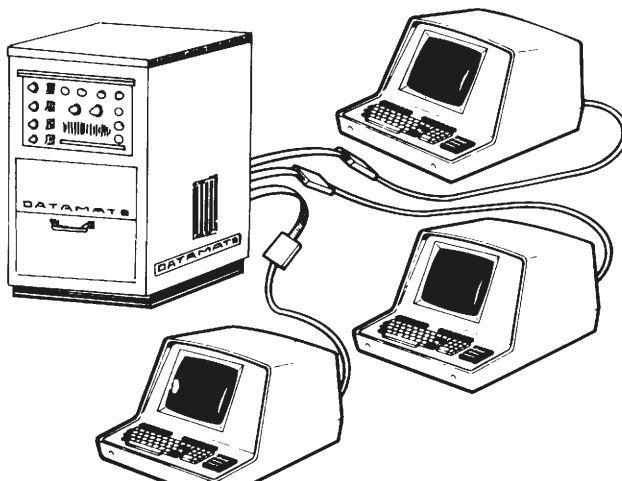
I princippet er de store dataanlæg bygget op på samme måde som mikrodatamaten. De består også af:

- en centralenhed
- en eller flere indlæseenheder
- en eller flere lagerenheder
- en eller flere udlæseenheder

### Maskinkonfiguration



I et større dataanlæg er centralenhedens kapacitet mange gange større end mikrodatamaten. Derfor kan den arbejde med meget omfattende programmer, og den kan betjene flere ydre enheder samtidig.

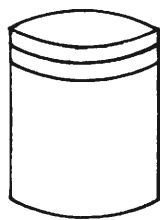


Til en større centralenhed kan der f.eks. være adgang fra adskillige dataskærme, som er anbragt i firmaets forskellige afdelinger.

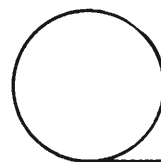
## Magnetplade Magnetbånd

I forbindelse med mikrodatamaten har vi anvendt disketter eller kassettebånd som ydre lager. De store datamaskiners ydre lager består som regel af magnetplader og/eller magnetbånd. Hver af disse kan rumme flere millioner tegn.

Symboler for



Magnetplade-lager

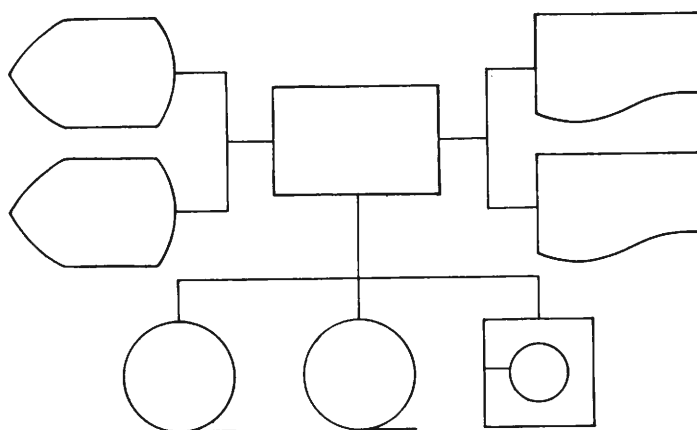


Magnetbånds-lager

### OPGAVER

#### Opgave 12.1

Skriv navne på enhederne i denne maskin-konfiguration. Markér med I, U eller YL, om det er en indlæse-, udlæse- eller ydre lagerenhed.



#### Opgave 12.2

Tegn maskin-konfigurationen for et anlæg, der består af: Tre dataskærme, to magnetpladestationer og en printer - foruden centralenheden.

## Eksempel på anvendelse af et større dataanlæg

### Grammofonplade klub

Maskin-konfigurationen i opgave 12.2 kunne være dataanlægget i den grammofonplade-klub, hvor du er medlem. Den første skærm er anbragt i ordre-modtagelsen, den anden på varelageret og den tredje i bogholderiet. Det ene pladelager indeholder kunde-registeret det andet vare-registeret.

### Øvelse A

Tilføj disse betegnelser på din tegning.

Når din **bestilling** på en grammofonplade er modtaget, indtastes den på **skærm 1**, og den registreres i centralenheden. Herfra kontaktes **vare-registeret**. Hvis pladen er på lager, **reserveres** den til dig, og der **trækkes 1** fra antallet af den pågældende plade.

Følgende figur viser et bestillingskort til pladeklubben.

| EKSTRA BESTILLINGSKORT |                         | SEND MIG SNAREST: |           |
|------------------------|-------------------------|-------------------|-----------|
| MEDLEMSNUMMER          | ← SKAL UDFYLDES!        | VARE NR.:         | VARE NR.: |
| _____                  |                         | _____             | _____     |
| NAVN                   | SKRIV MED BLOKBOGSTAVER | VARE NR.:         | VARE NR.: |
| _____                  |                         | _____             | _____     |
| ADR.                   |                         | VARE NR.:         | VARE NR.: |
| _____                  |                         | _____             | _____     |
| POSTNR.                | BY                      |                   |           |
| _____                  | _____                   |                   |           |

På **skærm 2** udskrives nu besked om, at pladen skal findes frem, og **printer** skriver det giro-indbetalingskort, der skal sendes med til dig.

Samtidig **bogføres** pladens pris på din konto i **kunde-registeret**. Når du har **betalt** pladen, indtastes beløbet på **skærm 3**, så din konto udlignes.

### Øvelse B

Prøv ved hjælp af pile at vise denne "handlingsgang" på din tegning fra opgave 12.2.

### Edb-service-bureau

#### Edb-service-bureau

For mange - især mindre - virksomheder kan det ikke betale sig selv at anskaffe edb-udstyr. De kan i stedet lade et edb-service-bureau ordne f.eks. bogholderi og lønregnskab. Sådanne bureauer findes overalt i landet. De fleste er private, men nogle af de største ejes af det offentlige: **Kommunedata**, der løser edb-opgaver for amter og kommuner, **Datacentralen**, der bl.a. fungerer som service-bureau for de forskellige ministerier.

### Eget dataanlæg

I løbet af de sidste 4-5 år er der kommet mange mindre og

billigere, men meget effektive datamaskiner på markedet. Derfor bliver der stadig flere og flere virksomheder, der anskaffer eget dataanlæg og "går over til edb".

### Eksempel 1 Tidstro databehandling og data-transmission

#### Pladsreservation

Hvis du ringer til DSB, SAS eller et af de store rejsebureauer for at bestille en billet, kan du i løbet af et øjeblik få at vide, om der er reserveret plads til dig.

Den, der modtager din bestilling, taster den ind på sin skærm. Og hvad enten det sker i København, Ålborg, Rønne eller et andet sted i landet, transmitteres din forespørgsel til det centrale plads-register. Herfra kommer der på få sekunder besked tilbage i form af en udskrift på skærmen. Hvis der er plads på den rejse, du ønsker, udskrives en billet til dig, og du noteres for beløbet - akkurat som beskrevet i forbindelse med grammo-fonpladeklubben.

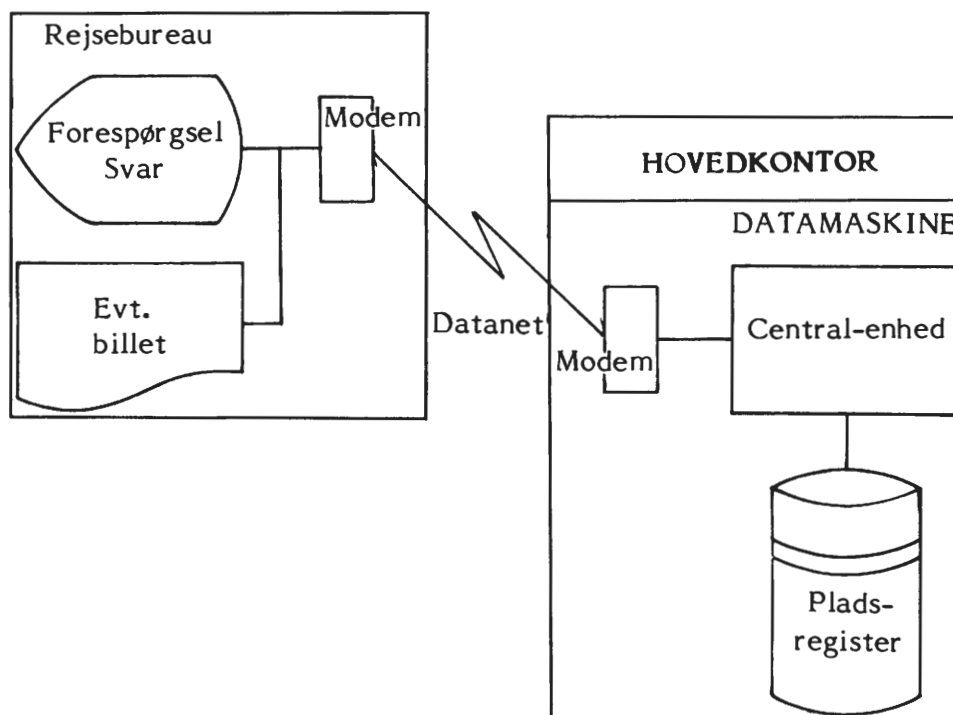
Med det specielle ved denne her metode er

#### Tids-tro databehandling

1. at det hele ordnes, mens man endnu holder telefonforbindelsen. Der forløber altså kun et øjeblik, fra bestillingen afgives, til reservationen har fundet sted. Dette kaldes **tidstro** databehandling.

#### Data-transmission

2. Centralenheden og det tilhørende ydre lager (pladsregisteret) befinder sig et helt andet sted i landet end dataskærmen. Dataskærmen kaldes nu også en **terminal**. Datasignalerne må derfor sendes gennem telefonnettet, eller et specielt datanet. Dette kaldes **data-transmission**. Da telefonnettet jo er beregnet til at transportere lydsignaler, må signalerne til og fra datamaten omformes i et apparat, som kaldes Modem



## Eksempel 2 Tidstro databehandling og data-transmission

Mange pengeinstitutter er tilsluttet et stort, centralt edb-anlæg f.eks. SDC, Sparekassernes Datacenter. Herfra er der forbindelse med over 3000 skærmterminaler, fordelt i sparekasse-afdelinger ud over hele landet.

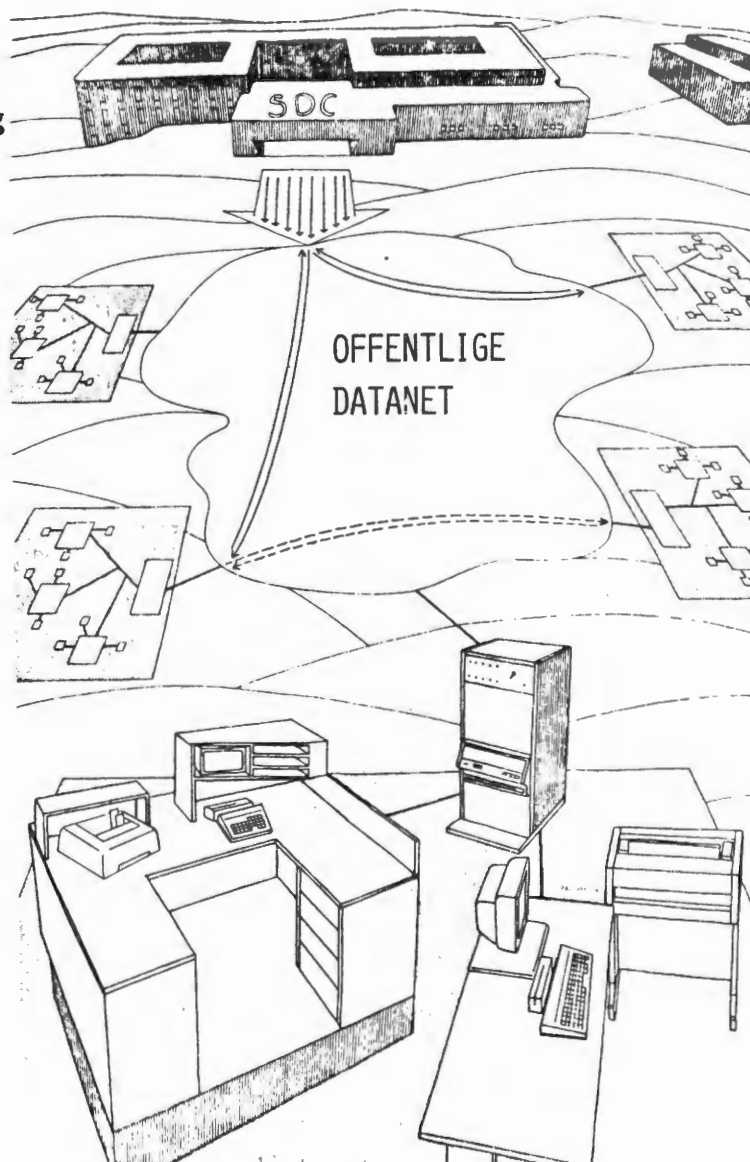
Centrale registre

Centralt edb-anlæg

Data-transmission

Lokal datamat

Skærm-terminal



Hvis du henvender dig i den lokale sparekasse f.eks. for at få at vide, hvad der i øjeblikket står på din konto, indtastes din forespørgsel på skærm-terminalen. Ved hjælp af dit konto-nummer, findes din konto i det centrale register. På få sekunder har du svaret udskrevet på en lille printer.

### OPGAVER

#### Opgave 12.3

Et par hundrede af de største boghandlere er tilknyttet et edb-system, der kan besvare forespørgsler angående 30-40.000 bøger fra ca. 400 forskellige forlag.

Hvis din boghandler er tilknyttet edb-systemet, kan du få svar på følgende:

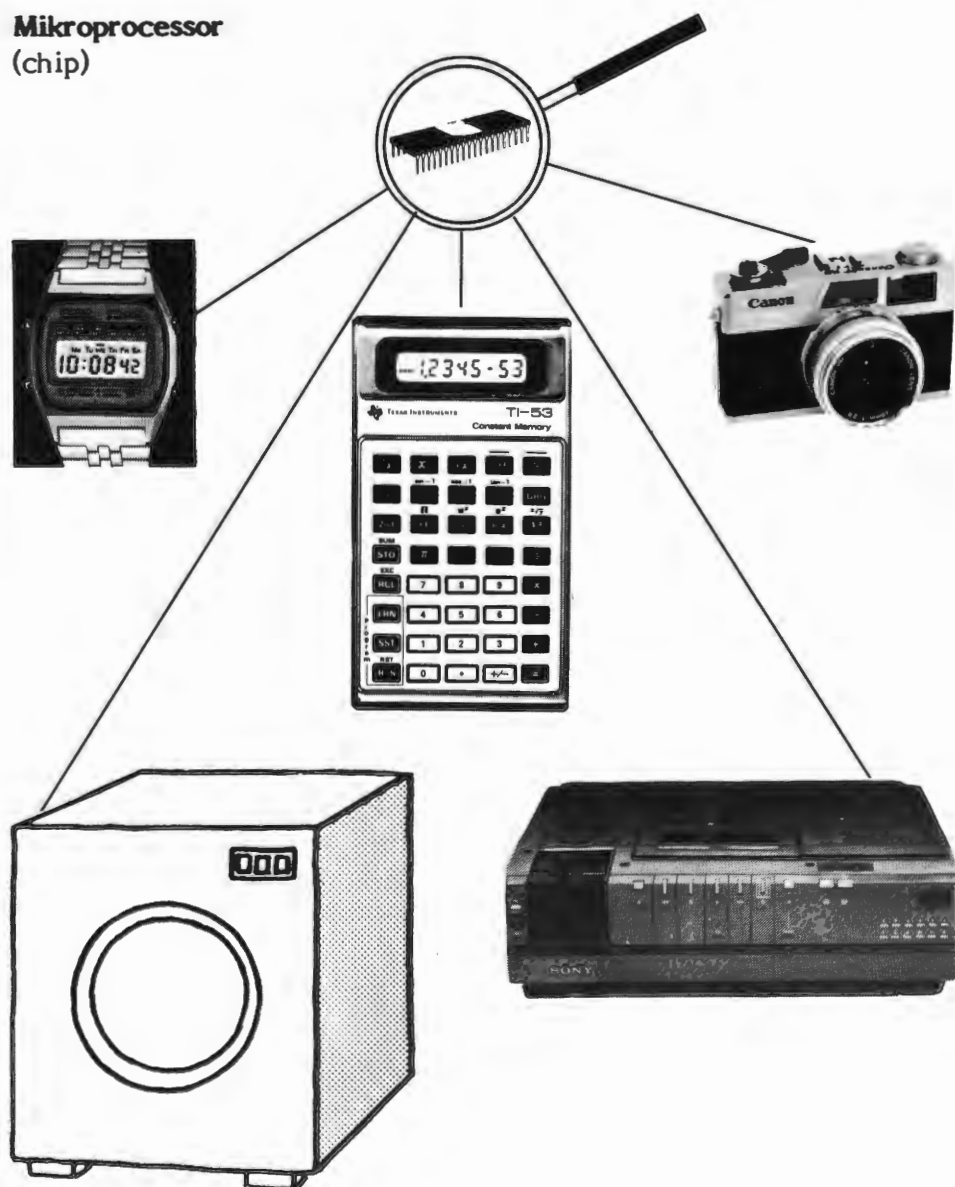
1. Hvad hedder den bog, der har dette nummer:  
ISBN 87-7351-0440-0?
2. Hvilke bøger er skrevet af A. Egebjerg Johansen?
3. Hvilke danske bøger har "data" i titlen?
4. Hvilke danske bøger har "edb" i titlen?

## KAPITEL 13 EDB OG SAMFUND

Efter at have arbejdet med mikrodatamaten har du nu fået et indtryk af, hvordan en datamat virker, og hvilke opgaver den kan løse. Der tales meget om, at edb og specielt mikroprocessoren vil få stor betydning for vort samfund. Det må derfor være naturligt at beskæftige sig med edb og samfundet.

De små elektriske kredse, som en mikrodatamat er opbygget af, kaldes chips. De er kun nogle få mm store. En af disse chips kaldes en mikro-processor. Det er mikroprocessoren, der styrer datamaten. I dag bruges mikroprocessorer til styring af de mest almindelige apparater.

### Mikroprocessor (chip)



## Hvorfor edb

Et af de første spørgsmål, man må stille, er: Hvorfor bruger vi EDB?

For at besvare dette spørgsmål, må vi se på, hvad datamaterne kan.

De kan arbejde meget hurtigt.  
De kan arbejde meget sikkert.  
De kan behandle store mængder oplysninger.

Det er bl.a. disse egenskaber ved datamater, der udnyttes i vort samfund.

## Hvor anvendes edb

Anvendelsen af datamater er allerede så udbredt, at det ville blive en uoverskuelig opgave at lave en liste over de erhverv, der anvender EDB.

For at få et overblik kan man opdele anvendelsen af edb i nogle hovedområder:

Administration  
Produktion  
Undervisning og videnskab

## EKSEMPLER PÅ ANVENDELSE AF EDB

På de følgende sider kan du læse om eksempler på anvendelsen af EDB.

### Eksempel 1 Styring i landbruget

Følgende artikel fortæller, hvordan man kan styre fodring af svin ved hjælp af mikrodatamater.

# Dansk EDB-opfindelse sætter mere sul på bacon-svinene

Anvendelsen af EDB-teknik og mikroprocessorer i landbruget kan blandt andet give grise bedre spisevaner og derved forbedre udbyttet i svinestaldene.

Det viser et anlæg, som proprietær og ingeniør Steen Blicher, Dalkildegaard ved Faaborg, har udviklet.

Med en computer som styringsenhed fodres og kontrolleres en hel svinebesætning på 750 svin. Det klares af en enkelt mand, uden at han teoretisk behøver at komme i forbindelse med grisene under hele deres opvækst.

Ingeniør Steen Blichers opfindelse er en biomat, der

fodrer grisene efter en helt ny fodringsfilosofi.

Biomaten giver foderet i mundfulde til hver enkelt gris, og hver mundfuld er nøje vejjet af.

Tidsrummet mellem hver enkelt mundfuld, der serveres for grisen, er desuden nøjagtigt afmålt.

Det hele styres af en computer.

Systemet indlærer automatisk grisene gode spisevaner. Hver enkelt gris har sin egen foderskål, som den går hen til, når den hører, at fodringsanlægget sætter i gang.

Når den kun får serveret én mundfuld ad gangen — oven i købet tilpasset nøjagtigt efter den portion, grisen

kan æde — beholder grisen sin plads ved ædeskålen.

Dyret opdager nemlig hurtigt, at det ikke kan betale sig at skubbe en svagere gris bort fra truget for også at æde foderet fra denne gris.

Det er normalt ellers dyrets adfærd, og det er derfor, at der normalt er stor forskel på de enkelte grise i samme sti i en svinestald.

Nu styres imidlertid grisenes adfærd af EDB-teknikken. Det er mikroprocessorerne, der gør, at vi nu kan stille de betingelser, så vi opnår et optimalt adfærdsmønster, siger ingeniør Steen Blicher.

## Eksempel 2 Industrirobotten

En mængde arbejde i industrien består i at indstille og regulere de maskiner, som udfører selve arbejdet. Gennem de sidste 15 år har man automatiseret maskinerne mere og mere. I dag kan det gøres ved at programmere en mikrodatamat, der er sluttet til maskinerne.

Hos FIAT i Italien og VOLVO i Sverige bliver bilkarosserierne samlet af industrirobotter.

Der findes enkelte fabrikker, hvor hele produktionen er fuldautomatiseret.

Følgende artikel fortæller, hvor og hvorfor man anvender industrirobotter.

### Monotont arbejde kan afskaffes:

# Industri-robotter ind på de miljøfarlige job

– En stor udvikling kan ventes på industrirobotområdet. Robotterne kan afskaffe monotont arbejde og bruges i kampen mod miljøfarlige arbejdspladser, og robotterne vil ikke nedlægge den samlede sum af arbejdspladser, tværtimod...

Det siger lederen af et dansk firma, der producerer industri-robotter, direktør Sven Erik Blaagaard, firmaet Scandinavian Phoenix. Firmaet har hovedsæde i Rødovre ved København og en afdeling på Frederikshavnsvej i Hjørring.

Sven Erik Blaagaard siger, at industri-robotter er en naturlig videreudvikling af værktøjsmaskiner og automatikudstyr, og at der for tiden i Skandinavien kun er cirka 130 industrirobotter, heraf størstedel i Sverige, og de er fortrinsvis i bilindustrien. I Danmark er der kun meget få egentlige industrirobotter.

– De områder, som vi vil satse på, er der, hvor der er tale om monotont arbejde eller miljøgener, dvs. røg, støj og møg. Endelig er der også områder, hvor en robot faktisk er bedre end mennesket, siger Sven Erik Blaagaard.

Det kan være til at håndtere store enheder eller f. eks. placere ting med en nøjagtighed på én hundrededel millimeter, hvor det menneskelige øje »kun« kan gøre det samme med ca. en millimeters nøjagtighed.

I forventningen om det voksende marked for industrirobotter, har Scandinavian Phoenix for nylig været med til at stifte foreningen »Dansk Robotforening«. Foreningens formål er bl.a. at udveksle erfaringer medlemmerne imellem og fremme en relevant udvikling og anvendelse af industrirobotter.

Foreningen vil være med til at placere industrirobotter fornuftigt i offentlighedens bevidsthed.

– Der findes mange fordomme omkring industrirobotter. Mange tror, at det blot drejer sig om at nedlægge arbejdspladser. Imidlertid er der i mange tilfælde tale om det modsatte, nemlig oprettelse af flere job. Industrirobotter er med til en forøget produktion, der kan medføre øget beskæftigelse, siger direktør Sven Erik Blaagaard.



### Eksempel 3 Kasseapparater

Kassedamerne i de store varehuse er også blevet EDB-medarbejdere. Deres kasseapparater er små datamater. Den kassebon, som kunden modtager, giver mange oplysninger.

Gennemgå kassebonnen, og find ud af, hvilke oplysninger den giver.

|                     |        |      |       |
|---------------------|--------|------|-------|
| HAVECENTRET         |        |      |       |
| HAR ÅBEN            |        |      |       |
| SØNDAG              |        |      |       |
| FRA KL 10 TIL KL 16 |        |      |       |
| BILKA               | 24.04  | 1982 |       |
| HERRETEXTIL         | 89,50  |      |       |
| SPORT               | 139,00 |      |       |
| TOTAL               | 228,50 |      |       |
| CHECK               | 250,00 |      |       |
| BYTTEPENGE          | 21,50  |      |       |
| 0130                | 144    | 13   | 12:59 |

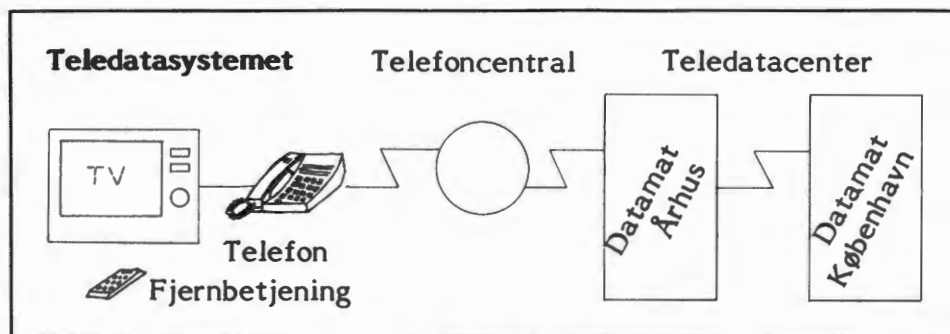
Når datamaten har udskrevet kassebonnen, bliver alle oplysningerne ved salget samlet i et EDB-anlæg i varehuset. Derved kan man føre kontrol med salget og varelageret. Når en bestemt vare er ved at være udsolgt, giver anlægget besked om at bestille en ny sending.

Ved hjælp af oplysningerne i EDB-anlægget kan man f.eks. også se, hvor hurtigt kassedamen arbejder, og hvor meget hun har solgt for.

### Eksempel 4 Teledata

Et forsøg her i landet går ud på at undersøge, om der er interesse for et teledatasystem. Teledata er betegnelsen for et elektronisk oplysningssystem. Ved hjælp af en telefon og et fjernsynsapparat kan man komme i forbindelse med en datamat.

Teledatasystemer er i disse år ved at dukke op i mange lande f.eks. England, Norge, Sverige, Vesttyskland, Canada og Japan. I Danmark er det telefonselskaberne, der sammen med Post- og Telegrafvæsenet står for forsøget.



## Teledata i 100 danske hjem

■ I 100 danske hjem kan familiemedlemmerne i dag spørge deres TV-apparater om madopskrifter, udenlandsnyheder, togafgange eller priser på forsikringer. Hjemmene er nemlig de første deltagere i det danske Teledata-forsøg, der indledtes i slutningen af marts. Bag forsøget står de danske tele-administrationer, som har fået Christian Rosing A/S til at opstille to datacentre i København og Århus, som styre Teledata-systemet. Modtagelsen sker på specielle farve-TV, fremstillet af B&O. Forsøget skal vare i to år, og i den tid flyttes de 100 terminaler fire gange. For øjeblikket står de hos 60 familier i Silkeborg og Skanderborg og hos 40 i Lyngby. Desuden findes 75 terminaler hos erhvervsvirksomheder, der leve-



rer stof til Teledata. Folk, der ikke selv deltager i forsøget, kan studere mulighederne på 25 skærme, som er opstillet på bl.a. biblioteker. Fra starten af omfatter Tele-

data i alt 20.000 sider tekst. Informationsleverandører er bl.a. DSB, SA', dag- og fagblade, kommuner, banker, forsikringselskaber og forskellige organisationer.□

Den enkelte bruger kalder op til datamaten på sin telefon. Datacentret melder klar med et "velkomstbillede" på TV-skærmen. Brugeren kan søge oplysninger på tre måder: trinvis søgning, direkte sidevalg eller alfanumerisk søgning.

## Eksempel 5 Datamater og våben

Her kan du læse om to eksempler på, hvordan et computer-system kan udnyttes både under en krig og til fredelige formål.

# Atter kamp på Falkland

Søndagens kamphandlinger var de første omkring Falklandsøerne siden tirsdag, da et argentinsk fly med Exocet-raket udskedeliggjorde den britiske destroyer »Sheffield«.

»Sheffield«s forkullede vrøg blev i går taget på slæb og trukket ud af krigszonen til sikkert farvand.



Når den franske Exocet-raket - her i affyringsøjeblikket - kunne uskadeliggøre destroyeren »Sheffield«, skyldes det, at raketten er udstyret med et effektivt målsøgningsanlæg. Ved hjælp af radar og en datamat programmeres raketten til at styre direkte mod målet. En »træffer« er således undgæelig.

# Krigsteknik skal sikre gasskib mod kollision

A. P. Møller-rederiets nyeste skib, gastankeren »Svend Mærsk« har et sikkerhedssystem, der hviler på samme principper som det argentinske Exocet-missil. Missilet, der i en fuldtræffer gjorde det af med en britisk destroyer i krigen om Falklandsøerne.

Men mens missilet via en computer kan fodres med oplysninger, så det styrer direkte mod sit mål, har man på »Svend Mærsk« vendt om på principperne – så man undgår kollision.

Det betyder, at computeren, når man inden for en radius af 24 sømil – ca. 45 kilometer – opfanger et radarspor, lynhurtigt kan beregne, om det er på kollisionskurs – og hvordan man undviger, hvis svaret er bekræftende.

En såkaldt antikollisionsradar.



## Eksempel 6 Edb i undervisningen

I undervisningen kan edb udnyttes til opbevaring af oplysninger på samme måde, som vi bruger et leksikon. Man kan med fordel også anvende datamater i selve undervisningen, når det drejer sig om at indlære en bestemt rutine. Det kan for eksempel være indlæring af glosser, tabeller eller kemiske formler.

# Skoleelever elsker at lege med datamaskiner

Nikala, Jette og Gitte løser regneopgaver på en datamaskine i Bella Centret. Faget regning ville nok være sjovere, hvis man havde sådan en maskine i klassen, mener de. Men det har lange udsigter, for kommunerne har ikke penge til det.

Af Lili Ochsner

I USA lærer mange børn i skolen at stave og regne ved hjælp af en EDB-maskine. Staver de et ord forkert, siger maskinen buuuuuhhh –

staver de rigtigt, synger den en lille sang.

I Danmark er man endnu ikke blevet enig om, hvorvidt skoleeleverne skal have dataanlæg som selvstændigt obligatorisk fag allerede på fjerde eller femte klassetrin, om EDB-lære skal puttes ind i andre fag som matematik eller samfundslære, eller om datalære skal være et valg fra 10. klasse.

Det kommer under alle omstændigheder til at vare nogle år, før de danske skoleelever får mulighed for at sidde i klasseværelset og trykke på EDB-maskinens knapper.

Kommunerne mangler penge i disse tider, og det vil koste cirka 100.000 kroner til en skole at anskaffe det udstyr, der er nødven-

digt for en ordentlig undervisning.

## De nye analfabeter

Eksperterne inden for EDB-området mener, at det er uhyre farligt, at danske børn ikke er kommet i gang med at lære EDB.

— Hvis ikke alle skoleelever får mulighed for at få et grundigt kendskab til, hvordan man arbejder med EDB, vil de som voksne blive delt op i to grupper, de teknologisk kloge og de teknologiske analfabeter. I det nærmeste fremtidssamfund vil det være lige så afgørende, at man har forstand på at operere en computer, som det i dag er helt afgørende, at man kan læse og skrive, mener de.

### Legetøj

I København i Bella Centret har der lige været en udstilling om små datamater. Mange skoleelever havde taget bussen derud, og de voksne udstillere lod gerne eleverne sidde og trykke på apparaterne. Ja, de var endda glade for at se, at børn og unge til forskel fra mange voksne ikke er spor bange for EDB, men tværtimod betragter datamaterne som legetøj. Der var særlig tre EDB-programmer, der tiltrak børnenes opmærksomhed: et, hvor man kunne tegne grafik på en TV-skærm, et, hvor man kunne prøve sine regne- og stavefærdigheder, og et, hvor man kunne prøve at være statsminister for fantasilandet Sumaria.

### Sjov regning

Nikala, Jette og Gitte, som alle tre går i anden klasse,

syntes, det var dødsjovt at regne på maskinen. Hver gang de havde et stykke, for eksempel 114 plus syv, fik de at vide, om svaret var rigtigt eller forkert. Var det rigtigt, kom der en masse små røde hjarter frem på skærmen. Var det forkert, sagde maskinen *søøvvvv*, og de fik den samme opgave en gang til.

Efter fem opgaver kunne de på skærmen læse, hvor mange stykker, de havde løst forkert eller rigtigt. De tre piger var enige om, at faget regning nok ville være sjovere, hvis de også af og til kunne få lov at regne på en datamaskine.

### Landet Sumaria

For de ældre elever, som har samfundslære i skolen, var EDB-programmet »Leder af landet Sumaria for 10 år« den helt store oplevelse. På EDB-skærmen fik børne-

ne følgende oplysninger: Hvor mange indbyggere, der er i Sumaria, hvor mange hektar agerland og hvor meget korn, der er på lager. Opgaven gik så ud på, at de skulle beslutte for et år ad gangen, hvor meget jord de ville opdyrke, hvor meget korn de ville sælge ud af landet eller bruge til mad til landets egne borgere. Når børnene havde taget beslutningen, fik de omgående konsekvenserne at vide på skærmen, og det kunne være barske konsekvenser som, på et år har du sultet halvdel af Sumarias befolkning ihjel, du er en tyrann — befolkningen hader dig og et oprør er på vej, du har solgt mere jord, end der er i Sumaria, eller mere positivt, du har styret landet godt, Anker Jørgensen, Henning Christophersen og Poul Schlüter kunne ikke have gjort det bedre.

### Øvelse A

Hvilke fordele vil der være ved at blive undervist af en datamat?

---

---

Hvilke ulemper vil der være ved at blive undervist af en datamat?

---

---

I hvilke fag kunne du tænke dig at blive undervist af en datamat?

---

---

### Eksempel 7 Edb-kriminalitet

Den stigende anvendelse af edb har medført en helt ny form for kriminalitet. Det drejer sig om tyverier over edb-anlæg. Det er særligt banker og forsikringsselskaber, der bliver bestjålet. I USA, hvor anvendelsen af edb er længst fremme, regner man med, at datatyverierne løber op i 75 mia. dollars årligt.

Fremgangsmåden for tyverierne er mange, men to metoder er oftest benyttet.

### **Salami-systemet**

En klassisk computerforbrydelse er Salami operationen. Det kan være en bankansat, som piller ved udbetalingerne af renter. Fidusen er at skære en meget tynd skive - men mange steder. Hvis en rentetilskrivning beregnes til for eksempel 87,234 kroner vil maskinen runde ned til 87,23 kroner og de 0,4 øre statistisk set gå lige op med en tilsvarende oprunding. Salami operatøren overfører i stedet alle de fritstående brøkdele af ører til sin egen konto. Og i en bank med mange millioner transaktioner kan mange regndråber godt blive til en stor flod.

**EDB ER BLEVET  
ET ELDORADO  
FOR TYVE**

### **Trojansk hest**

Den avancerede computerforbrydelse er som en trojansk hest. Den bygges ind i systemet og aktiviseres først, når det rette sæt af omstændigheder fremkommer. Måske er de designede til kun at virke en gang for derefter at blive glemt i maskinens hjerne. Data-detektiverne siger til output Magazine, at som oftest bliver de ikke opdaget.

Det er derfor kontrollen med ansatte anses for så vigtig. Folks adgang til maskinernes hukommelse skal begrænses. En ansat bør ikke have lejlighed til at sidde ved maskinerne om natten og teste, om de vil forkaste hans svindelprogram. Han behøver nemlig ikke på forhånd at vide, hvilke sikkerhedsmekanismer, der allerede er bygget ind i computeren.

**DATATYVERIER  
LØBER OP I  
MILLIARDER**

## **VIRKSOMHEDER SIKRER SIG**

## **FOR LIDT MOD EDB-FUSK**

### **OPGAVER**

- Opgave 13.1** Find artikler og eksempler på andre områder, hvor EDB anvendes. Det kan være artikler og eksempler, som du enten har set, læst eller hørt om. Du kan eventuelt lave en collage, rapport o.lign.
- Opgave 13.2** Undersøg hvor mange datamater, der findes i din kommune eller nærmere afgrænset område. Lav en liste over, hvilke erhvervsgrene det drejer sig om. Udvalg en eller flere af disse og planlæg et besøg.
- Opgave 13.3** Opstil en liste over de positive og de negative sider ved EDB.
- Opgave 13.4** Diskuter om EDB er ven eller fjende.

## KAPITEL 14    REGISTRE

Der bliver skrevet mange forskellige oplysninger om os i løbet af vores levetid. Mange af disse oplysninger bliver samlet i registre. En borger kan for eksempel være optaget i følgende registre:

|                         |                                                                         |
|-------------------------|-------------------------------------------------------------------------|
| Folkeregisteret         | Navn og adresse<br>Forældres navn og adresse                            |
| Skatteregisteret        | Indkomst<br>Fradrag<br>Trækprocent                                      |
| Sygehusregisteret       | Blodtype<br>Vaccinationer<br>Indlæggelser                               |
| Undervisningsregisteret | Uddannelser<br>Uddannelsessted og år                                    |
| Militærregisteret       | Uddannelse<br>Militær rang                                              |
| Strafferegisteret       | Bøder<br>Fængselstraffe<br>Betingede domme                              |
| Socialregisteret        | Socialindkomst<br>Socialhjælp                                           |
| Kirkeregisteret         | Ægteskabelig stand<br>Børn                                              |
| Motorregisteret         | Køretøjets art<br>Mærke og årgang<br>Vægtafgift<br>Indregistreringsdato |

De nævnte eksempler kaldes offentlige registre, fordi de er oprettet af og bruges af stat og kommune (offentlige kontorer).

Også mange private virksomheder har registre. Banker og sparekasser, stormagasiner, forsikringsselskaber osv. registrerer kunderne i kunderegistre.

Man ved ikke helt bestemt, hvor mange registre der findes. Men man regner med at gennemsnits-danskeren er registreret i ca. 100 forskellige registre.

Langt den største del af oplysningerne er afgivet enten af os selv eller vores kommune. Enhver skatteyder kan læse skattevæsenets oplysninger på sit skattekort. Banker og sparekassers oplysninger kender vi fra kontoudtog.

Ønsker man en udskrift af oplysningerne i et bestemt register, sker det ved henvendelse til kommunen eller den virksomhed, hvor man er registreret. En undtagelse er dog politiets og hospitalernes journaler. I det sidste tilfælde skal en læge forklare, hvad registeret indeholder. Det er bl.a. for at undgå misforståelser.

### Eksempel på spørgeskema til edb-register

Et politisk parti her i landet er gået over til edb-registrering. Her kan man se det spørgeskema, der blev udsendt til parti-medlemmerne.

Skemaerne blev udfyldt frivilligt.

|                                                            |                                                 |
|------------------------------------------------------------|-------------------------------------------------|
| Medlem af partiforening.....                               | indmeldt år.....                                |
| Navn.....                                                  |                                                 |
| Vej/gade.....                                              |                                                 |
| Postnummer/postdistrikt.....                               |                                                 |
| Kommune.....                                               |                                                 |
| Telefon.....                                               | fødselsår.....                                  |
| Stilling.....                                              |                                                 |
| Arbejdsplads.....                                          | telefon.....                                    |
| Uddannelsesinstitution (hvis du er under uddannelse).....  |                                                 |
| Fagforening/fagforbund.....                                | afdeling.....                                   |
| Uddannelsesorganisation (hvis du er under uddannelse)..... |                                                 |
| Medlem af og /eller aktiv i:                               |                                                 |
| .....                                                      | Folkebevægelsen mod EF                          |
| .....                                                      | OOA eller OVE eller anden energigruppe, hvilken |
| .....                                                      | .....                                           |
| .....                                                      | bolig/lejerbevægelse, hvilken.....              |
| .....                                                      | kvindebevægelse, hvilken.....                   |
| .....                                                      | forbrugergrupper, hvilken.....                  |
| .....                                                      | NOAH eller anden miljøbevægelsen, hvilken.....  |
| .....                                                      | fredsbevægelse, hvilken.....                    |
| .....                                                      | International solidaritetsarbejde, hvilken..... |
| .....                                                      | anden bevægelsesarbejde, hvilken.....           |
| .....                                                      | " " ".....                                      |
| .....                                                      | andre organisationer, hvilke.....               |
| Tillidshverv i fagforeninger, hvilke.....                  |                                                 |
| Tillidshverv i bevægelserne, hvilke.....                   |                                                 |
| Tillidshverv i organisationer iøvrigt.....                 |                                                 |
| Medlem af kommunalbestyrelsen i .....kommune               |                                                 |
| Medlem af skolekommissionen i .....kommune                 |                                                 |
| Medlem af skolenævnet ved.....skole.....kommune            |                                                 |
| Vurderingsmand.....amtsrådsmedlem.....                     |                                                 |
| Dato.....                                                  |                                                 |
| Underskrift.....                                           |                                                 |

## Øvelse A

Diskuter følgende spørgsmål:

Hvilke oplysninger kan have almindelig interesse for partiet?

Hvilke oplysninger kan have særlig interesse for partiet?

Hvordan kan oplysningerne gavne partiet?

Kan oplysningerne misbruges af uvedkommende personer?

Kommentér, om du frivilligt vil give de samme oplysninger.

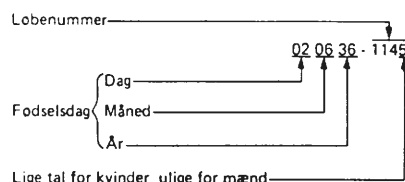
## CPR-REGISTRET

I et moderne samfund har det offentlige et ansvar for den enkelte borger. Derfor er det nødvendigt for staten og kommunerne at indsamle og opbevare oplysninger om os. For 10-15 år siden blev disse persondata opbevaret på kartotekskort. Men med udviklingen af datamaskiner var det naturligt at benytte edb-registre. På den måde blev det nemmere at holde orden på de mange oplysninger og finde frem til dem, når de skulle bruges.

Staten oprettede i slutningen af 1960'erne det Centrale Personregister (CPR-registeret) for at samle de mest almindelige persondata fra de forskellige registre i et stort edb-register. Her bliver 5 mill. danskers persondata lagret på ca. 30 magnetplader og magnetbånd. For at sikre, at oplysningerne på en bestemt person ikke bliver lagret under et forkert navn, har man tildelt hver dansker et **personnummer**.

## Personnummeret

PERSONNUMMERET er opbygget således:



Eksemplet vedrører en mand, født den 2. juni 1936. Løbenummeret (de 4 sidste cifre) vil være højere end 5000 for personer, født før den 1. januar 1900.

### VEJLEDNING:

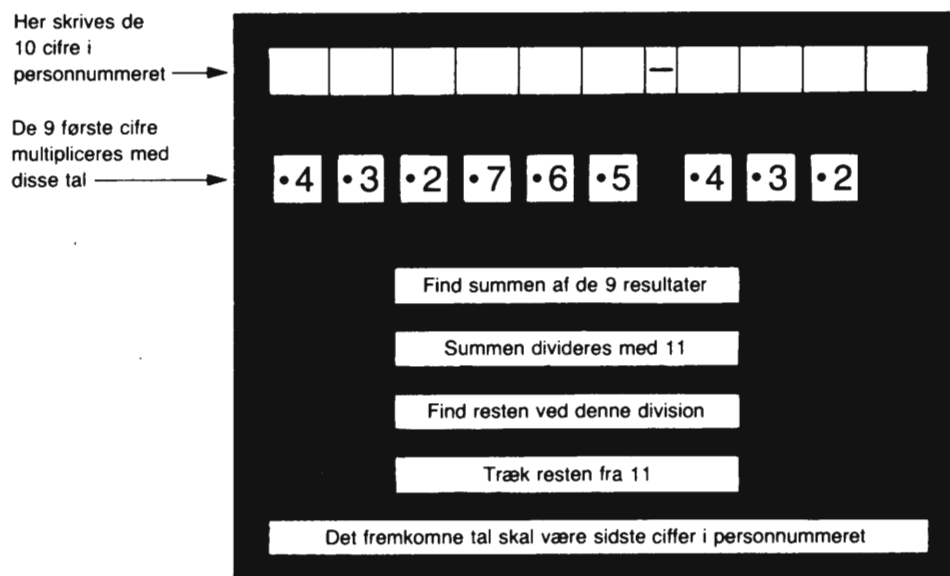
- Oplysningerne i personnummerbeviset er udskrivet på grundlag af de oplysninger, som det centrale personregister (CPR) havde på udstedelsesdagen. Det anførte navn vil af tekniske grunde i nogle tilfælde være forkortet, ligesom udenlandske bogstaver i visse tilfælde er erstattet af danske.
- Ved henvendelse til folkeregistret angående fejl ved personnummeret eller det anførte navn må **dokumentation** indsendes eller forevises (normalt døbs-, fødsels- eller vielsesattest).
- Hvis et tidligere tildelt personnummer er erstattet af et nyt, er kun det nye personnummer gyldigt.
- For at modvirke misbrug bør beviset **underskrives**, dog ikke førend indehaveren er fyldt 15 år.

For at undgå fejlkodninger er det sidste ciffer i løbenummeret et kontrolciffer. Ved indkodning af data foretager datamaskinen automatisk en kontrolberegning. Hvis beregningen slutter med et andet tal end kontrolcifferet, forkastes CPR-nummeret. På denne måde findes 99,5 procent af alle fejl ved indtastning af personnumre.

## Kontrolciffer

Personnumre er opbygget på en sådan måde, at det er muligt at kontrollere dem.

Kontrolmetoden fremgår af dette skema:



## OPGAVER

### Opgave 14.1

Undersøg om CPR-nummeret 20 03 74 - 3708 er rigtigt.

Undersøg om CPR-nummeret 20 03 74 - 3709 er rigtigt.

### Opgave 14.2

Undersøg dit eget CPR-nummer.

Prøv at bytte om på to cifre i dit nummer og undersøg, hvad der sker ved kontrolberegningen.

## DATABASE

En database er et register, der kan indeholde meget store mængder af oplysninger. Kapaciteten kan være på milliarder af tegn.

Formålet med en database er at opbevare oplysninger i et stort samlet register frem for at sprede dem på flere mindre registre.

### Opgave 14.3

Du skal nu igen arbejde med datamaten. Indsæt disketten med færdige programmer og LOAD "DATABASE". Undersøg, hvilke oplysninger denne DATABASE indeholder.

Vi gør opmærksom på, at det er et "tænkt" eksempel.

## REGISTERLOVEN

Et af de emner, der er stor debat om i dag, er problemerne omkring EDB-registre. Der er mange forskellige opfattelser. Nogle mener, at samfundet ikke kan fungere uden registre, og andre mener, at al registrering skal afskaffes.

Problemerne hænger sammen med to af vort samfunds grundlæggende principper: nemlig den enkelte borgers ret til privatlivets fred og samfundets behov for persondata, for at det kan opfylde sit ansvar.

For at undgå misbrug vedtog Folketinget i 1978 en lov om EDB-registre. Den indeholder regler for, hvem der må oprette registre - hvilke oplysninger, der må registreres - hvad registrenes indhold må bruges til.

Uddrag af loven om EDB-registre:

### *Registrering og opbevaring af oplysninger.*

**§ 9.** Der må kun registreres oplysninger, der klart er af betydning for varetagelsen af vedkommende myndigheds opgaver. Der må endvidere registreres oplysninger, der klart er af betydning for varetagelsen af en anden myndigheds opgaver, såfremt registret indrettes således, at oplysningerne kun kan benyttes af den anden myndighed.

**Stk. 2.** Der må ikke registreres oplysninger om politiske forhold, som ikke er offentligt tilgængelige. Andre oplysninger, der vedrører rent personlige forhold, herunder oplysninger om race, religion og hudfarve, om seksuelle og straffbare forhold samt oplysninger om helbredsforhold og misbrug af nydelsesmidler og lignende, må ikke registreres, medmindre dette er nødvendigt for varetagelse af registerets opgaver.

**Stk. 3.** Registrerede oplysninger, der på grund af deres alder eller af andre grunde har mistet deres betydning for varetagelsen af registerets opgaver, skal slettes. Registre, der anvendes løbende, skal indrettes således, at der foretages fornøden ajourføring af oplysningerne.

**Stk. 4.** Justitsministeren kan efter indhentet udtalelse fra registertilsynet tillade, at registereksemplarer overføres til opbevaring i arkiv på nærmere angivne vilkår.

**§ 10.** Det kan i forskrifterne for et register fastsættes, at der skal gives en regi-

streret meddelelse om optagelse i registeret. Meddelelsen skal indeholde oplysning om den registreredes adgang til at få oplysninger fra registeret efter bestemmelserne i §§ 13-15.

**§ 11.** Der skal foretages fornøden kontrol til sikring af, at der ikke registreres urigtige eller vildledende oplysninger. Oplysninger, der viser sig urigtige eller vildledende, skal snarest muligt slettes eller berigtiges.

**§ 12.** Der skal træffes de fornødne sikkerhedsforanstaltninger mod, at oplysningerne misbruges eller kommer til uvedkommendes kendskab.

**Stk. 2.** Sikkerhedsforanstaltningerne skal om fornødent omfatte kopier og udskrifter af registeret, herunder kopier og udskrifter, der er videregivet til andre myndigheder.

**Stk. 3.** For registre, der indeholder oplysninger af særlig interesse for fremmede magter, skal der træffes foranstaltninger, der muliggør bortskaffelse eller tilintetgørelse i tilfælde af krig eller lignende forhold.

### *Registrerede personers adgang til oplysninger om sig selv.*

**§ 13.** Fremsætter en registreret person begæring herom, skal den registeransvarlige myndighed snarest muligt give den registrerede meddelelse om de oplysninger, der er registreret om ham.

## Opgave 14.4

Hvilke oplysninger må registre indeholde? \_\_\_\_\_

Hvilke oplysninger må ikke registreres? \_\_\_\_\_

I hvilke tilfælde skal oplysninger i registreret slettes?

I hvilke tilfælde træder sikkerhedsforanstaltningerne i gang?

Henvend dig til Folkeregisteret på kommunekontoret. Bed dem om en udskrift af oplysningerne på dit CPR-nummer med henvisning til § 13.



## KAPITEL 15 OPGAVESAMLING

Opgaverne kan udføres, efter at man har læst de tilsvarende kapitler.

### Opgaver til kapitel 4

**Opgave 15.1** På disketten vil du finde 6 små programmer:

1. TÆNKETAL
2. TREKANTER
3. BATNUM
4. TRICK
5. TALGÆT
6. PERSONNUMMER

Skriv f.eks. LOAD "TÆNKETAL", og når programmet er indlæst, skriv da RUN og følg anvisningerne i programmet.

### Opgaver til kapitel 7

**Opgave 15.2** Omskriv følgende udtryk til "COMAL":

$$A + (B : 4 + C) * A \quad \underline{\hspace{10cm}}$$
$$A * B + C * D - \frac{D - A}{B} \quad \underline{\hspace{10cm}}$$

**Opgave 15.3** Udregn værdien af følgende udtryk:

$$B = 3 - ((340 + 56)/9 - 4) \quad B =$$

$$C = (41 - (8 - 32/8) - 5) \quad C =$$

**Opgave 15.4** Udregn følgende udtryk:

$$D = (64 + 12) - (18 + 5*16/2\uparrow 3*5) \quad D =$$

$$E = (64 + 12 - 18 + 5) * 16/2\uparrow 3*5 \quad E =$$

**Opgave 15.5**

Skriv et program, som beregner omkreds og areal af et rektangel med siderne 234 og 45.75. Prøv at eksperimentere med PRINT-sætningerne.

**Opgave 15.6**

Skriv et program, som kan udregne rumfanget af en tønde.

Eksempel:

$$h = 15 \text{ dm}$$

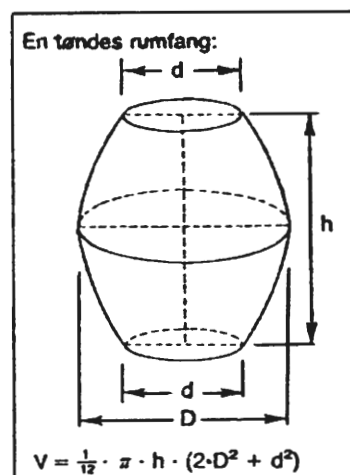
$$d = 5 \text{ dm}$$

$$D = 10 \text{ dm}$$

$$= 3,14$$

Hvad bliver rumfanget af denne

tønde: \_\_\_\_\_

**Opgave 15.7**

En Købmand køber et parti kaffemaskiner til 235 kr. pr. stk. Hans omkostninger er 16% af købsprisen, hans fortjeneste er 20% af de samlede udgifter, og momsen er 22% af de samlede udgifter plus fortjenesten. Prøv at skrive et program, som beregner

Omkostninger

Samlede udgifter

Fortjeneste

Salgspris uden moms

Salgspris plus moms

Brug godt med PRINT-sætninger.

**Opgaver til kapitel 8****Opgave 15.8**

En radioforretning har tilbud på kassettebånd. Prisen pr. bånd er 17,85 kr. Hvis man køber 5 bånd eller flere, er prisen pr. stk. 14,40 kr. Lav et program, der udskriver beløbet, når antallet af bånd indtastes.

**Opgave 15.9**

I firmaet A. Olsens Maskinfabrik er timelønnen for de første 40 timer om ugen 56,80 kr.

For de næste 5 timer er timelønnen 56,80 kr. + 50%.

For de næste 5 timer er timelønnen 56,80 kr. + 75%.

Skriv et program, der med det ugentlige antal timer som inddata udskriver lønseddel for den pågældende arbejder.

**Opgave 15.10**

Restskat betales i princippet på to måder. Hvis det samlede beløb (restskatten + et rentetillæg på f.eks. 8%) er mindre end 5000 kr., indregnes restskatten i næste års skat. I modsat fald skal restskatten betales i 3 rater i begyndelsen af det nye år.

Prøv at lave et program, der udskriver enten

RATE

eller

ENGANGSSKAT

samt

SAMLEDE RESTSKAT

**Opgaver til kapitel 9****Opgave 15.11**

På en vejstation målte man temperaturen 8 gange på et døgn. Man fik følgende målinger: -7, -2, 0, 5, 6, 3, -3 grader. Lav et program, der beregner middeltemperaturen.

Brug evt. READ - DATA.

**Opgave 15.12**

Ved en lægeundersøgelse blev 40 elevers højde målt:

|                       |   |       |
|-----------------------|---|-------|
| 5 elever målte 165 cm |   |       |
| 15 -                  | - | 164 - |
| 5 -                   | - | 163 - |
| 4 -                   | - | 162 - |
| 2 -                   | - | 161 - |
| 9 -                   | - | 160 - |

Lav et program, der beregner elevernes middelhøjde.

**Opgave 15.13**

I en 8. klasse målte man i fysik smeltepunktet for naftalin. De seks hold fik følgende forsøgsresultater:

75, 79, 80, 81, 82, 89 grader.

Lav et program, der beregner gennemsnitstemperaturen.

**Opgave 15.14**

Skriv et program, der udskriver summen af ulige tal, der er mindre end 162.

**Opgave 15.15**

Skriv et program, der udskriver gennemsnittet af de lige tal, der er mindre end 109.

- Opgave 15.16**      Skriv et program, der udregner gennemsnittet af 5 indtastede tal. Hvis der ønskes fortsat indtastning, skal der indtastes 1, ellers 0.
- Opgave 15.17**      Skriv et program, der kun udskriver summen af ulige tal, når der indtastes 7 forskellige tal. (NB: Nogle af de indtastede tal kan godt være lige).
- Opgave 15.18**      Skriv et program, der udskriver de lige tal, der er mindre end 100.
- Opgave 15.19**      I et land lever for øjeblikket 4.500.000 mennesker. Befolkningstilvæksten er 1.4 promille. Landet kan ikke føde flere end 10 millioner mennesker. Hvor lang tid varer det, inden landet er overbefolket?
- Formlen for vækst:  $k * (1 + x)^n$
- k er 4.500.000, x er 1.4 : 1000, n er antallet af år.
- Som inddata angives altså antal indbyggere og tilvæksten i promille. Som uddata forventes antallet af år, inden landet er overbefolket. Indsæt evt. andre promiller end 1.4.
- Opgave 15.20**      Et tal, som kun har 2 divisorer, kaldes et primtal. Et tal, som har mere end 2 divisorer, kaldes et sammensat tal.
- Skriv et program, der kan afgøre, om et tal er et primtal eller et sammensat tal.
- Opgave 15.21**      Olsen har regnet ud, at hans bil gennemsnitlig mister 10% i værdi pr. år. Næste års værdi er altså dette års værdi gange 0.9.
- Skriv et program, der kan beregne, hvor meget Olsens bil er værd efter 1 år, 2 år, 3 år o.s.v. Vælg selv købspris.
- Opgave 15.22**      Ved den årlige generalforsamling i DE HØJE PIGERS KLUB skal pigerne måles, for at de - forhåbentlig - kan opfylde mindstekravet på 180 cms højde.
- Der skal skrives et READ - DATA program, hvor alle højderne indskrives i en datasætning. Programmet udskriver de forskellige pigers højde samt pigernes antal.

**Opgave 15.23**

I denne opgave vil man gerne have samme udskrift som i opgave 15.22 med en ekstra linie:

GENNEMSNITSHØJDEN I DE HØJE PIGERS KLUB ER: CM

Brug i begyndelsen af programmet SUM = 0 og senere SUM = SUM + HØJDE.

**Opgave 15.24**

I O. Hansens isenkram er alle varer forsynet med en pris, som er incl. moms, det vil sige, at momsen (22%) er lagt til. Af hensyn til momsregnskabet skal ekspedienten indtaste prisen excl. moms (altså prisen uden moms) på kasseapparatet.

Skriv et program, som kan udskrive en tabel, som Hansens ekspedienter kan bruge, når de skal indtaste priserne. F.eks. således:

| INCL MOMS | EXCL MOMS |
|-----------|-----------|
| 1 kr      | 0.82 kr   |
| 2 kr      | 1.64 kr   |
| 3 kr      | 2.46 kr   |
| .         | .         |
| .         | .         |
| 100 kr    | 81.97 kr  |

Afrund tallene til kr. med to decimaler. Brug INT.

**Opgave 15.25**

Skriv et program, der kan omregne hastigheder fra meter/sek. til km/t. Lav evt. en tabel og forær den til din fysiklærer.

**Opgave 15.26**

Skriv et program, der kan simulere terningekast, og som udskriver:

Antallet af enere:  
Antallet af toere:  
Antallet af treere:  
Antallet af firere:  
Antallet af femmere:  
Antallet af seksere:  
ved 10.000 kast.

**Opgave 15.27**

Galileis faldlov ser således ud:

Hvis faldtiden er t sek., og tyngdeacc. er 9.8 m/sek<sup>2</sup>, er faldvejen s bestemt ved følgende formel:

$$s = 1/2 * 9,8 * t^2$$

Skriv et program, der kan beregne faldvejen, når faldtiden (t) i sek. er givet.

Skriv endvidere et program, som kan beregne, hvor længe et legeme ved frit fald er om at falde 100 m, 200 m, 300 m o.s.v. Udarbejd igen evt. en tabel.

#### Opgave 15.28

Ole skal på sin sommerferie med Interrail passere både Tyskland, Holland, Belgien og Frankrig. Han får brug for en tabel, så han umiddelbart kan sammenligne de forskellige landes valutaer. Den kan f.eks. se således ud:

| Danske kr. | Tyske mark | Hollandske gylde | Belgiske fr. | Franske fr. |
|------------|------------|------------------|--------------|-------------|
| 1          | 0.29       | 0 .35            | 4            | 0.41        |
| 2          |            |                  |              |             |
| 5          |            |                  |              |             |
| 25         |            |                  |              |             |
| 50         |            |                  |              |             |
| 75         |            |                  |              |             |
| 100        |            |                  |              |             |

Find de aktuelle valutakurser og fremstil en sådan tabel.

#### Opgaver til kapitel 10

#### Opgave 15.29

I forbindelse med en ændring af momsen fra 20.25% til 22% ønsker et stormagasin udarbejdet et program, der på grundlag af indtastning af varenummer og gammel pris (incl. 20.25%) kan udskrive varenummer og ny pris (incl. 22% moms). Programmet skal stoppe, når der indtastes 0,0 (nul,nul) for henholdsvis varenummer og gammel pris.

Beregning af ny pris sker efter følgende forskrift:

$$\text{Ny pris} = \frac{\text{gammel pris} \times 122}{120,25}$$

Udarbejd et comal-program til denne proces.

#### Opgave 15.30

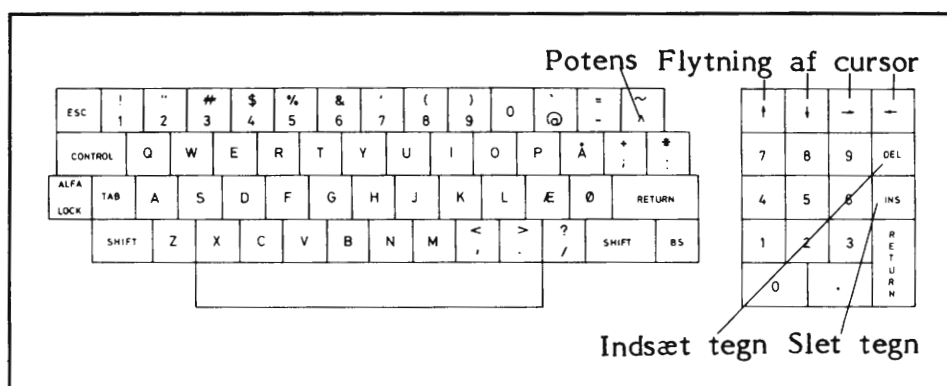
Et edb-firmas salgschef aflønnes pr. måned i procent af månedens salg. Den del af det samlede månedlige salg (sum af de enkelte salgsbeløb), der er mindre end eller lig med kr. 100.000, aflønnes med 10%. Den del af salget, der er større end kr. 100.000, men mindre end eller lig med kr. 150.000, aflønnes med 15%. Månedligt salg større end kr. 150.000 aflønnes med 20%.

INDDATA til databehandlingen er de enkelte salgsbeløb.

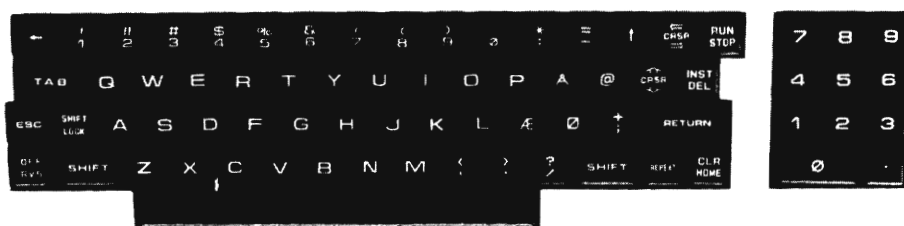
UDDATA: salgschefens månedsløn i kr.

## APPENDIX

Følgende skitser viser tastatur til henholdsvis Comet/DIamat og Commodore



### TASTATUR COMET/DIamat



### TASTATUR COMMODORE

Følgende oversigt viser forskellen i Comal til Piccolo 702, Commodore og Comet/DItamat. Vi har kun medtaget de Comal-ordrer, der er omtalt i denne bog.

|                                                                 | PICCOLO 702                | COMMODORE               | DITAMAT/COMET                               |
|-----------------------------------------------------------------|----------------------------|-------------------------|---------------------------------------------|
| Uddata på printer                                               | SELECT OUTPUT<br>"printer" | SELECT OUTPUT<br>"LP"   | SELECT OUTPUT "LP2:"<br>SELECT OUTPUT "LP:" |
| Uddata på skærm                                                 | SELECT OUTPUT<br>"console" | SELECT OUTPUT<br>"DS"   | SELECT OUTPUT "DS:"                         |
| Programhop                                                      | GOTO start                 | GOTO start              | GOTO 10/GOTO START                          |
| Tilfældigt tal                                                  | RND<br>mellem 0 og 1       | RND(1,6)<br>fra 1 til 6 | RND(1.6)<br>fra 1 til 6                     |
| Udskrivning af<br>programmer som<br>f.eks. lager på<br>diskette | DIR                        | CAT                     | CAT/DIR                                     |
| Kolonnebredde                                                   | ZONE 10                    | ZONE 10                 | TAB = 10                                    |
| Renser skærmen                                                  | PRINT CHR\$(12)            | PRINT "CLR"             | CLEAR                                       |
| Overførsel af pro-<br>gram fra diskette<br>til centralenhed     | LOAD "NAVN"                | LOAD "NAVN"             | LOAD NAVN                                   |
| BLINK                                                           | PRINT CHR\$(130)           | -                       | -                                           |
| RANDOM(IZE)                                                     | RANDOMIZE                  | -                       | RANDOM(IZE)                                 |
| Bemærkninger til<br>program                                     | //                         | REM el.//               | REM el. //                                  |
| Omnummerering                                                   | RENUMBER                   | RENUM                   | RENUM                                       |
| Lagring af<br>programmer                                        | SAVE "NAVN"                | SAVE "NAVN"             | SAVE NAVN                                   |

BENYT PROGRAMMET INFO, SOM LIGGER PÅ DISKETTEN.

[illegible]

## KAPITEL 9.

```

OPG. 9.1: GNSNIT
OPG. 9.2: AFBETALING
OPG. 9.3: PLAT
OPG. 9.4: TKAST
OPG. 9.5: SPAROP
EKSEMPEL: KVADRAT
OPG. 9.6: KUBIKT
OPG. 9.7: TABEL7
OPG. 9.8: COMALUS
OPG. 9.9: DIVISOR
OPG. 9.10: STØRST
OPG. 9.11: MINDST
OPG. 9.12: IGEN25
OPG. 9.13: VOKSE
OPG. 9.15: PENGE
EKSEMPEL: ARB
OPG. 9.16: ELEV4
OPG. 9.17: KURSVÆR

```

## K A P I T E L 10.

```
ØVELSE A: VARME1
OPG. 10.1: VARME2
OPG. 10.3: FISK
```

## K A P I T E L 11.

```
OPG. 11.1: LAKMUS
OPG. 11.2: TERN3
OPG. 11.3: JUL
OPG. 11.4: RYKKER
OPG. 11.5: TEST3
```

S P I L.

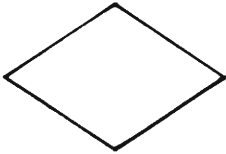
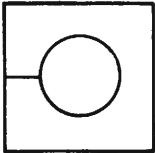
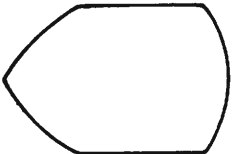
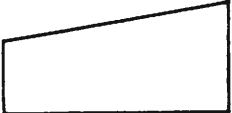
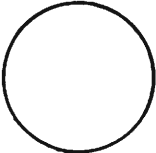
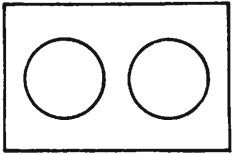
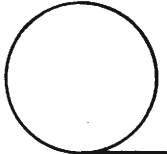
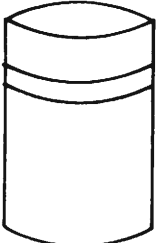
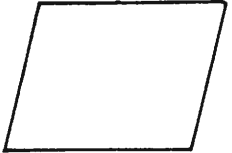
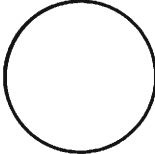
1. TÆNKETAL
2. TREKANTER
3. BATNUM
4. TRICK
5. TALGÆT
6. PERSONNUMMER

S L U T

OPG. 8.1: POST  
OPG. 8.2: ZOO  
OPG. 8.3: DSB  
OPG. 8.4: KASSE  
OPG. 8.5: SOL  
OPG. 8.6: ANDENGR  
OPG. 8.7: TOPPKT  
OPG. 8.8: RUMFANG2  
OPG. 8.9: OVERFL  
OPG. 8.10: POSTTAKST  
OPG. 8.11: HERON  
OPG. 8.12A: HYP  
OPG. 8.12B: KATETE  
OPG. 8.13: SKUDAR  
OPG. 8.14: TERNING

149

Følgende symboler er anvendt i denne bog:

| SYMBOL                                                                              | SYMBOLNAVN                    | SYMBOL                                                                              | SYMBOLNAVN              |
|-------------------------------------------------------------------------------------|-------------------------------|-------------------------------------------------------------------------------------|-------------------------|
|    | CENTRALENHED                  |    | SPØRGEKASSE             |
|    | DISKETTESTATION               |    | BEHANDLING              |
|    | DATASKÆRM                     |    | PILE, der for-          |
|   | TASTATUR                      |   | binder kasserne         |
|  | PRINTER                       |  | YDRE LAGER              |
|  | KASSETTE<br>BÅNDSTATION       |  | MAGNETBÅND-<br>STATION  |
|  | START eller<br>STOP           |  | MAGNETPLADE-<br>STATION |
|  | LÆSE- eller<br>SKRIVEKASSE    |                                                                                     |                         |
|  | INDLÆSE- eller<br>UDLÆSEENHED |                                                                                     |                         |

## STIKORDSREGISTER

- Algoritme 39, 41, 45
- AUTO 74
- Beregning 66, 68
- Betingede hopordrer 80
- Boolsk udtryk 81
- Båndstation 20
- CPR-registret 137
- Centrale registre 125
- Centralenhed 17
- CHR\$ 74
- Comal 65
- Comet 12
- Comodore 12
- Cursor 14
- DATA 107-108
- Database 138
- Datamat 9, 125
- Dataskærm 15
- Datatransmission 124
- DEL 29
- DIM 117
- DIR 148
- Diskette 19
- Diskettestation 18
- Ditamat 12
- Divisionsalgoritme 40
- DO 99
- Dollartegn 117
- Dynamisk lighedstegn 67
- EDIT 33
- ELSE 83-84
- ENDIF 80-82
- ENDPROC 112-113
- ENDWHILE 98-100
- ESC 14
- Etikette 79
- EXEC 113
- FOR 93
- Forgrening 56-58
- GOTO 79
- IF 80-82
- Inddata 70
- INPUT 72, 74
- INT 88
- Kassettebånd 20
- Kontrolciffer 138
- Linienummer 65
- LIST 28
- LOAD 33
- Lock-tast 14
- Lower case 14
- Løkker 93
- Magnetbånd 122
- Magnetplade 122
- Markør 14
- Maskinkonfiguration 23, 121
- Mikrodatamat 11
- Mikroprocessor 127
- NEW 27
- NEXT 93
- OUTPUT, se SELECT OUTPUT
- Personnummer 137
- PICCOLO 11
- PRINT 67, 74, 94
- Printer 21
- Printplade 17
- PROC 112-113
- Procedurer 111
- Programmer på diskette 33
- Programkørsel 27
- Programmering 65 f
- RANDOM(IZE) 90
- READ 107-108
- Regnetegn 68
- Registerloven 139
- Registre 135
- REPEAT 102-103
- RETURN 14
- Return-tast 14
- RND 89
- RUN 28
- Rutediagram 49-57
- SAVE 148
- SELECT OUTPUT 30
- Shift-tasten 14
- Små bogstaver 14
- Sortere 104
- Space 14
- SQR 88
- Standard funktioner 88-90
- STEP 95
- Store bogstaver 14
- Symboler 22
- Symbolliste 22
- Sammenligning 56
- Systemkommandoer 27, 31, 33
- Sætninger, COMAL- 65
- Tab 109
- Tastatur 13-14
- Tekstvariable 117
- Teledata 130
- Terminal 124
- TEST1 32
- TEST2 63
- TEST3 120
- THEN 80-82
- Tidstro Databehandling 124
- Ubetingsede hopordrer 79
- Uddata 71
- UNTIL 102-103
- Upper case 14
- Valg 56
- Variabel 67-68, 72
- WHILE 98-100
- ZONE 94

## **DATALÆRE MED MIKRODATAMAT**

indeholder en beskrivelse af programmeringssproget Comal samt en kort beskrivelse af mikrodatamaten og edb's anvendelse i samfundet.

Til bogen er knyttet en diskette til forskellige mikrodatamater. Disketterne indeholder programmer til bogens øvelser og opgaver.

Bogen er først og fremmest beregnet til undervisningsbrug, men den er også velegnet til selvstudium.