



BESKRIVELSE		Side	1/1
System	COMET	Dato	1980-05-19
		Signatur	GDH/hh

Opdatering

Meddelelse nr. 6

Til indehaverne af mappen: COMET BRUGERVEJLEDNING.

Udarbejdelse af afsnit 5 til brugervejledningen er nu ved at være tilendebragt.

Afsnittet er blevet mere omfattende end oprindeligt antaget, hvorfor der er fremstillet en særlig indholdsfortegnelse til det.

Nærværende opdatering indeholder ca. 80% af afsnit 5. Resten vil følge indenfor de nærmeste uger sammen med en række andre informationer, som i øjeblikket indsamles.

Af hensyn til fremtidig opdatering er der endnu ikke påført sidenummering i afsnit 5. De enkelte delafsnit er dog nummereret og vil blive anvendt ved fremtidig reference.

Vi vil opfordre brugerne til skriftligt at dokumentere eventuelle fejl og mangler ved såvel COMAL-80 som denne brugervejledning og sende dokumentationer til:

ICL A/S
Bredgade 23
1260 København K

og mærke kuverten: COMET.

Afsnit 5 indsættes efter skilleblad 5 i brugervejledningen. Appendix A1 indsættes efter skilleblad 7.

Med venlig hilsen


Gunnar Dyekjær Hansen

Form BESKRIV (76 - 04)



Forblad 10

System / arkiv	Side
Dato	Sign.

C O M E T
B R U G E R V E J L E D N I N G

1. Indholdsfortegnelse

2. Start af COMET

Beskrivelse af de enkelte
3. enheder

4. Operativsystemet

5. COMAL-80 instruktionssættet

Vedligeholdelse, fejlmelding,
garantibestemmelser
6. og tilbehør

Appendix A1
7. Fejlmeddelelser

Appendix A2
8. Fejlfinding (h/w)

Appendix A3
9. Tekniske Data

Appendix A4
Styrekommandoer i midlertidigt
10 operativsystem

International Computers Limited a/s

Hovedkontor Bredgade 23 1260 København K Telefon 01 - 13 55 88
Jylland & Fyn Romancevej 9 8700 Horsens Telefon 05 - 62 75 88





1. Indholdsfortegnelse
2. Start af COMET
 - Sammenkobling
 - Kontrol af sammenkoblingen
3. Beskrivelse af de enkelte enheder
 - MIKRO-datamaten COMET
 - Kasettebåndstationen
 - Tastaturet
 - Dataskærmen
4. Operativsystemet
 - Systemkommandoer
5. COMAL-80 instruktionssættet
 - Alfabetisk oversigt og specifikation af instruktionerne
 - Editering
6. Vedligeholdelse, fejlmelding, garantibestemmelser og tilbehør
7. Appendix A1: Fejlmeddelelser
8. Appendix A2: Fejlfinding (h/w)
9. Appendix A3: Tekniske data
10. Appendix A4: Styrekommandoer i midlertidigt operativsystem.





Opstart af COMET Type 2001

Sammenkobling

COMET mikrodatamaten består af 3 enheder, nemlig:

1. Selve datamaten
2. Tastatur
3. Dataskærm

Sammenkoblingen er ganske enkel. Kablet fra tastaturet indsættes i øverste bøsning mærket "Tastatur" på datamatens bagside. Det medfølgende løse videokabel indsættes i bøsningen mærket "Videoudgang" - også på datamatens bagside - og i dataskærmens antenneindgang. Kablerne kan kun indsættes på én måde.

Dernæst sættes de to netledninger (én fra dataskærmen og én fra datamaten) i en stikkontakt (OBS! 220V vekselspænding).

Datamat og dataskærm tændes. Den røde lampe på COMET'ens forside skal straks lyse, og efter opvarmning skal der komme lys på monitorskærmen.

Der trykkes kortvarigt på "Reset" knappen og skærmen skal nu vise teksten:

COMET COS VO.O. Isæt system-bånd

COMET Type 2001 er en såkaldt "RAM-baseret datamat", hvilket betyder, at den kan arbejde med forskellige programmer, som indlæses via kassettebåndoptageren. Ovenstående udskrift betyder, at datamaten nu afventer, at et kassette-bånd indeholdende hovedprogrammet (også kaldet systemprogrammet) skal blive indsat i kassettebåndoptageren.

Sammen med COMET leveres 1 kassettebånd. Dette er et system-bånd, som indeholder COMAL-80 og operativsystemet. Båndet er indspillet i begge retninger, således at det kan indlæses, ligegyldigt hvordan kassetten vendes. Det går dog hurtigst, såfremt den fyldte spole er til venstre. Båndet indsættes og "Return"-tasten på tastaturet nedtrykkes. Båndoptageren vil nu begynde at indlæse COMAL-80. Det tager ca. 45 sekunder, hvorefter skærmen viser teksten:



Versionsnr. af COMAL-80

COMAL-80 VO.85

Implementeret af Arne Christensen.

COPYRIGHT METANIC APS.

*Ø

Datamaten er nu klar til at arbejde i COMAL-80 på normal måde.

Kontrol af sammenkoblingen

Hvis systemet ikke er sammenkoblet korrekt, indikeres og rettes fejlen på følgende måde:

- Den røde lampe i afbryderen lyser ikke

Kontrollér, at der er spænding på vægkontakten, samt at netstikket er sat ordentligt i. Se efter, om der er tændt for afbryderen på datamatens forside.

- Der kommer ikke lys på skærmen

Kontrollér, at den er tændt.

- Billedet er ustabil

Juster billedhold på skærmens bagside.





Beskrivelse af de enkelte enheder

COMET type 2001 består af 3 enheder, nemlig:

1. Selve datamaten
2. Tastatur
3. Dataskærm

Mikrodatamaten COMET type 2001

Mikrodatamaten er opbygget omkring den integrerede kreds zilog Z-80, som er markedets mest avancerede 8-bit CPU. Centralenheden er opbygget omkring en backplane med 8 pladser. COMET type 2001 anvender kun 4 af disse pladser, og der er således 4 frie pladser til senere udbygning. Denne udbygning kan f.eks. være tilslutning af diskette, linieskriver, hardware-matematik eller udbygning til koncentrator for en flerbrugergruppe.

Centralenheden er udstyret med 4K ROM-lager i standardudgave. Dette ROM-lager indeholder COS-software for styring af kassettebåndstationer. Herudover er der 32K RAM-lager til rådighed som arbejdslager. Dette lager kan udbygges i trin af 4K eller 16K til ialt 64K lager.

Opdatering og indlæsning af nye versioner af COMAL-80 og operativsystemer samt skift til andet programmeringssprog sker ved hjælp af den indbyggede kassettebåndoptager. Denne indlæsning tager for et 16K program ca. 25 sekunder.

Bagpladen er som standard forsynet med stik for dataskærm, for ASCII-kodet tastatur og for en serieudgang/indgang efter RS-232 standard.

Kassettebåndstation

COMET type 2001 er udstyret med en indbygget digital kassettebåndstation, og er fuldt forberedt for udbygning med endnu én.

Hver båndstation kan opbevare op til 128K bytes og overføre disse til arbejdslageret med en hastighed af 6000 bit pr. sekund.

Programmer og data lagres i sekventielle filer, som kaldes op i arbejdslageret under et navn. Hver kassette kan således indeholde flere filer (op til 15). For at beskytte allerede indlæste data, kan kassetterne skriveskyttes.



Skrivesikring

Kassettetapen kan skrivebeskyttes, således at der kun kan læses fra den. Det vil da ikke være muligt at anvende instruktionerne PUT og ERASE. Faciliteten anvendes til at beskytte indlæste programmer f.eks. COMAL-systemet og demonstrationsprogrammer.

Skrivebeskyttelse sker ved, at den sorte prop i hjørnet af kassetten prikkes ud. Side A beskyttes ved fjernelse af proppen ud for A→, og side B beskyttes tilsvarende ved fjernelse af proppen ved←B.

Siderne A og B identificeres ved henholdsvis et stort A og et stort B skrevet på siden af kassettebåndet. Der arbejdes på den side, som kan ses gennem glasruden i kassettebåndstationen, når kassettetapen er sat i.



Tastaturet

(Standard på COMET type 2001 leveret før 15 februar 1980).

Tastaturet er opbygget efter dansk skrivemaskinestandard med såvel små som store bogstaver. Desuden er det udstyret med et antal kontrolfunktioner og et numerisk tastatur for at lette indtastning af tal.

COMET type 2001 er en RAM-baseret datamat. Det betyder, at man nemt og hurtigt kan udskifte sine systemprogrammer.

Det betyder også, at de forskellige kontroltaster kan have forskellig betydning i de forskellige systemprogrammer. Den i det følgende nævnte betydning gælder, når man arbejder med COMAL-80 systemprogrammet. Arbejder man med andre systemprogrammer, skal kontroltasternes betydning findes i den tilhørende manual.

ESC Denne tast afbryder igangværende programkørsler eller listninger og bringer maskinen i en tilstand, hvor systemkommandoer kan indtastes.

CTRL Kontroltasten nedtrykkes samtidig med en anden tast og giver mulighed for at benytte bogstav- og taltasterne til forskellige kontrollerende funktioner.

Denne mulighed bruges ikke i COMAL-80, men benyttes i mange andre systemprogrammer.

SHIFT SHIFT-tasten har samme funktion som på en almindelig skrivemaskine, d.v.s. den anvendes, når man vil skrive store bogstaver eller det øverste tegn på dobbelttasterne. Tasten holdes nedtrykt samtidig med den anden tast.

RETURN "Vognretur"-tasten anvendes som på en normal skrivemaskine ved lineskift. Samtidig bevirker tasten, at COMAL-80 undersøger den indskrevne linie for at finde ud af, om det er en kommando, der skal udføres straks eller en programlinie, der skal gemmes.

REPT Repetitionstasten. Hvis den holdes nede sammen med en anden tast gentages tegnet eller funktionen ca. 2 gange i sekundet.



INS, DEL COMAL-80 er udstyret med kraftige editeringsfaciliteter. Til at understøtte disse er tastaturet udstyret med disse to knapper. INS åbner linien, hvor den blinkende cursor står, og muliggør indsætning af tegn i en allerede skrevet linie, uden at resten af linien skal omskrives. DEL fjerner på tilsvarende måde tegnet, hvor cursoren står.

ONLINE Denne tast får terminalen til at arbejde som terminal for et større EDB-anlæg. Alt, hvad her efter indtastes på tastaturet, sendes til serieudgangen, og på skærmen vises, hvad der ankommer til serieindgangen. Dette fortsætter, til tasten igen nedtrykkes sammen med "SHIFT" tasten, eller til "RESET"-knappen indtrykkes. Det er til denne anvendelse ikke nødvendigt at have indlæst noget program i arbejdslageret.

GRAP MODE COMET type 2001 er som standard udstyret med 128 tegn, som inkluderer dansk tegnsæt med store og små bogstaver samt nogle semigrafiske tegn. Herudover kan den som ekstrafacilitet udstyres med endnu 128 tegn. Dette kan være et udenlandsk tegnsæt eller flere semigrafiske tegn. Denne tast kan derefter, tegn for tegn, skifte mellem de 2 tegnsæt.

Herudover er tastaturet udstyret med nogle blanke taster, som er reserveret til senere udvidelser.

Nyt Tastatur

(Standard på COMET type 2002 leveret efter 15. februar 1980).

Tastaturet er opbygget efter dansk skrivemaskinestandard med såvel små som store bogstaver. Desuden er det udstyret med et antal kontrolfunktioner.

COMET type 2002 er en RAM-baseret datamat. Det betyder, at man nemt og hurtigt kan udskifte sine systemprogrammer.

Det betyder også, at de forskellige kontrolltaster kan have forskellig betydning i de forskellige systemprogrammer. Den i det følgende nævnte betydning gælder, når man arbejder med COMAL-80 systemprogrammet. Arbejder man med andre systemprogrammer, skal kontrolltasternes betydning findes i den tilhørende manual.



- ESC Denne tast afbryder igangværende programkørsler eller listninger og bringer maskinen i en tilstand, hvor systemkommandoer kan indtastes.
- CONTROL Kontroltasten nedtrykkes samtidig med en anden tast og giver mulighed for at benytte bogstav og taltasterne til forskellige kontrollerende funktioner.
- Denne mulighed bruges ikke ved programmering i COMAL-80, men benyttes i mange andre systemprogrammer.
- INS, DEL COMAL-80 er udstyret med kraftige editeringsfaciliteter. Til at understøtte disse er tastaturet udstyret med disse to knapper. INS åbner linien, hvor den blinkende cursor står og muliggør indsætning af tegn i en allerede skrevet linie, uden at resten af linien skal omkrives. DEL fjerner på tilsvarende måde tegnet, hvor cursoren står.
- BREAK/
GRAP MODE COMET type 2002 er som standard udstyret med 128 tegn, som inkluderer dansk tegnsæt med store og små bogstaver samt nogle semigrafiske tegn. Herudover kan den som ekstrafacilitet udstyres med endnu 128 tegn. Dette kan være et udenlandsk tegnsæt eller flere semigrafiske tegn. Denne tast kan derefter, tegn for tegn, skifte mellem de 2 tegnsæt.
- SHIFT SHIFT-tasten har samme funktion som på en almindelig skrivemaskine, dvs. den anvendes, når man vil skrive store bogstaver eller det øverste tegn på dobbelttasterne. Tasten holdes nedtrykket samtidig med den anden tast.
- LOCK Et tryk på denne tast låser tastaturet i samme tilstand, som man har, når SHIFT-tasten holdes nedtrykt. Funktionen ophører ved fornyet tryk.
- ALPHA
LOCK Et tryk på denne tast giver store bogstaver, men fortsat tal ved brug af tastaturet. Denne funktion udløses ved fornyet tryk.
- RETURN "Vognretur"-tasten anvendes som på en normal skrivemaskine ved lineskift. Samtidig bevirker tasten, at COMAL-80 undersøger den indtastede linie for at finde ud af, om det er en kommando, der skal udføres straks, eller en programlinie, der skal gemmes.

REPT	Tastaturet kan udbygges med repetitionsfunktion. I så fald skal denne tast anvendes.
NUL	Semigrafisk tegn. Optræder som mellemrum på skærmen.
BS/+ →	Cursormove-taster. Flytter cursoren til henholdsvis højre og venstre.
↑↓	Disse taster er reserveret til brug i en forstærket editor, som senere vil blive introduceret.

Controlkoder

Følgende funktioner opnås, når kontroltasten nedtrykkes sammen med en karakter:

CONTROL

+ G Denne kode får datamaten til at arbejde som teletypekompatibel terminal. Alt, hvad der skrives på tastaturet, sendes til serie udgangen. Alt, hvad der modtages på serieudgangen, vises på skærmen.

Dataskærm

Dataskærmen er på basissystemet et almindeligt kompakt fjernsyn udstyret med linieindgang. Skærmen kan vise 1024 tegn i 16 linier á hver 64 tegn.

NB: Fjernsynet er ombygget på en sådan måde, at det ikke kan arbejde som normalt fjernsyn.

Dataskærm og datamat arbejder efter normal CCIV-standard. Via en koblingsboks kan man derfor tilkoble normale monitorer som slaveskærme.



KAPITEL 5 : COMAL-80**Indholdsfortegnelse**

- 5.1 Indledning
- 5.2 Indtastning
 - 5.2.1 Programsætninger
 - 5.2.2 Kommandoer
 - 5.2.3 RETURN og ESC
- 5.3 Sprogets grundelementer
 - 5.3.1 Talkonstanter
 - 5.3.2 Strengkonstanter
 - 5.3.3 Identifikatorer
 - 5.3.4 Numeriske variable
 - 5.3.5 Strengvariable
 - 5.3.6 Udtryk (med operatorer)
- 5.4 COMAL-80 sætninger
 - 5.4.1 Indledning
 - 5.4.2 Kommentarer
 - 5.4.3 Erklæringer
 - 5.4.4 Tildelinger
 - 5.4.5 PRINT
 - 5.4.6 INPUT
 - 5.4.7 READ
 - 5.4.8 DATA
 - 5.4.9 RESTORE
 - 5.4.10 LABEL
 - 5.4.11 Betingede sætninger
 - 5.4.12 CASE
 - 5.4.13 Repeterende sætninger
 - 5.4.14 Procedureerklæring
 - 5.4.15 Procedurekald (EXEC)
 - 5.4.16 Funktionserklæring (DEF-ENDDEF)
 - 5.4.17 GOTO
 - 5.4.18 Skærmkontrol
 - 5.4.19 SELECT
 - 5.4.20 RANDOM

- 5.4.21 CHAIN
- 5.4.22 TRAP
- 5.4.23 STOP
- 5.4.24 END
- 5.4.25 Specielle (BASIC-) sætninger

5.5 Standardfunktioner og systemvariable

- 5.5.1 Aritmetiske standardfunktioner
- 5.5.2 Tegnorienterede standardfunktioner
- 5.5.3 Systemvariable:

- EOD
- ESC
- ERR
- EOF

5.6 Filsystem

- 5.6.1 INIT
- 5.6.2 RELEASE
- 5.6.3 OPEN
- 5.6.4 CLOSE
- 5.6.5 PRINT-file
- 5.6.6 INPUT-file
- 5.6.7 WRITE
- 5.6.8 READ
- 5.6.9 EOF

5.7 Systemkommandoer

- 5.7.1 AUTO
- 5.7.2 CAT
- 5.7.3 CLOSE
- 5.7.4 CON
- 5.7.5 DEL
- 5.7.6 DELETE
- 5.7.7 ENTER
- 5.7.8 FORMAT
- 5.7.9 GETUNIT
- 5.7.10 INIT
- 5.7.13 NEW
- 5.7.18 RUN
- 5.7.21 SIZE
- 5.7.23 UNIT



5.8 Editering

5.9 Tegn-koder



5.5 Standardfunktioner og systemvariable

I dette afsnit beskrives standardfunktioner og systemvariable i COMAL-80.
Det er valgt at dele beskrivelsen op i følgende 3 dele:

- aritmetiske funktioner
- tegnorienterede funktioner
- systemvariable



5.1 Indledning

I det følgende beskrives programmeringssproget COMAL-80 i den version, som er implementeret på mikrodatamaten COMET.

COMAL-80 er i lighed med det oprindelige COMAL-sprog defineret af Børge Christensen, Tønder Statsseminarium, som nu i en årrække har arbejdet med COMAL. Erfaringerne, som er blevet høstet gennem årene, har dannet grundlag for de udvidelser og ændringer, som, i forhold til den oprindelige COMAL, findes i COMAL-80.



5.2. Indtastning

Når COMAL-80 systemkassetten er blevet indlæst, udskrives en identifikation samt et versionsnummer på følgende måde:

COMAL-80 Vn.n

Af Arne Christensen
COPYRIGHT (C) 1980 METANIC ApS

Herefter udskrives en "*" for at tilkendegive, at systemet er parat til at modtage inddata.

Nu kan programsætninger og systemkommandoer indtastes.



5.2.1 Programsætninger

Et COMAL-80 program består af en række programsætninger, som alle er forsynet med et linienummer. Linienumrene må antage heltallige værdier mellem 1 og 9999, begge incl. En programsætnings linienummer bestemmer dens placering i programmet, idet programsætningerne automatisk ordnes i voksende linienummerorden, uanset indtastningsrækkefølge.

Umiddelbart efter indtastningen af en programsætning foretages en kontrol af sætningens opbygning (syntaksanalyse). Er sætningen ukorrekt opbygget, udskrives en fejlmeddelelse, og markøren (cursor) placeres ud for det tegn, hvor fejlen blev opdaget. Herefter kan sætningen rettes ved brug af systemets redigeringsfaciliteter, og en ny syntaksanalyse kan foretages. Godkendes programsætningens opbygning, udskrives blot en "*".

Indtastes en programsætning med samme liniemnummer som en allerede eksisterende sætning, vil den nye sætning erstatte den eksisterende.

Programsætninger er beskrevet i afsnit 5.4.



5.2.2 Kommandoer

Kommandoer er instrukser til COMAL-80 systemet. Kommandoer udføres øjeblikkeligt efter indtastningen og adskiller sig fra programsætninger ved ikke at begynde med et linienummer.

Kommandoer er beskrevet i afsnit 5.7.



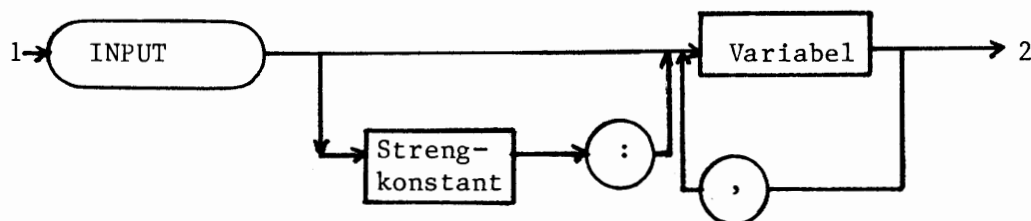
5.2.3 RETURN- og ESC-tastens funktion

Indtastning af en linie afsluttes ved at trykke på tastaturets RETURN-tast. Først når denne tast er aktiveret, begynder COMAL-80 systemet at behandle linien.

En påbegyndt indtastning kan afbrydes ved at trykke på tastaturets ESC-tast. Denne tast kan ligeledes anvendes til at afbryde en igangværende programkørsel, editering eller udskrivning.

5.3. Sprogets grundelementer

COMAL-80 sprogets syntaks beskrives i denne vejledning v.h.a. syntaksdiagrammer, som de kendes fra Wirth's PASCAL-rapport*. Et eksempel på tolkning af syntaksen for et af COMAL-80's sprogelementer, gennemgås herunder.



Diagrammet angiver en lidt forenklet syntaks for en INPUT-sætning. En vej gennem diagrammet, startende ved pilen, som peger ind i INPUT-rammen (1), og sluttende ved pilen i diagrammets højre side (2), beskriver en syntaktisk korrekt INPUT-sætning. Terminal-symboler, dvs. de symboler, som direkte skrives i et COMAL-80 program, er omgivet af afrundede rammer; symbolerne omgivet af kantede rammer angiver navne på andre syntaksdiagrammer.

Man kan således af diagrammet se, at en syntaktisk korrekt INPUT-sætning skal begynde med nøgleordet 'INPUT'. Efter 'INPUT' kan enten følge en række kommaadskilte variable, eller en tegnkonstant afsluttet af et ':', efterfulgt af en række kommaadskilte variable. Eksempler på legale INPUT-sætninger er vist herunder:

```
INPUT ANTAL#, STKPRIS
INPUT "NAVN":NAVN$
```

ANTAL#, STKPRIS og NAVN\$ er variable og "NAVN" er en streng-konstant.

* Kathleen Jensen and Niklaus Wirth: Pascal, User Manual and Report, Springer-verlag, 1978.

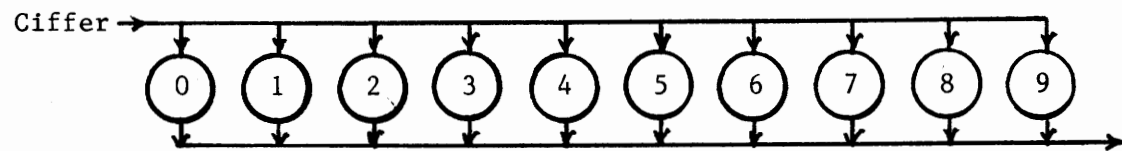
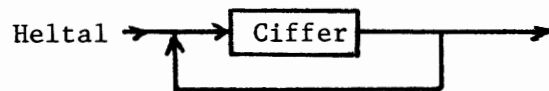
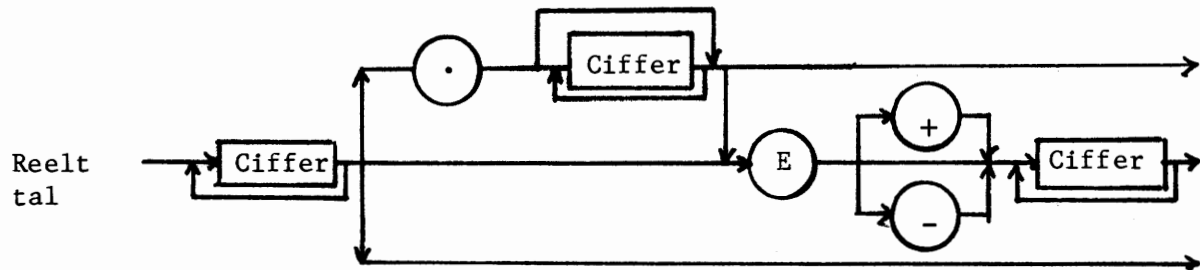


I beskrivelsen af syntaksen for visse sammensatte sprogkonstruktioner er linieskift illustreret ved at lade diagrammet forsætte på en ny linie. Se eksempelvis afsnit 5.4.11.4.

I det følgende kan blanktegn, med mindre andet udtrykkeligt er anført, forekomme overalt i en sætning uden at ændre dennes betydning.

5.3.1 Talkonstanter

Tal benyttes som konstanter i aritmetiske udtryk, i inddata samt i CASE- og DATA-sætninger. Der findes reelle tal samt heltal, og deres opbygning er vist herunder.



Blanktegn må ikke forekomme i talkonstanter.

Eksempler:

Heltal

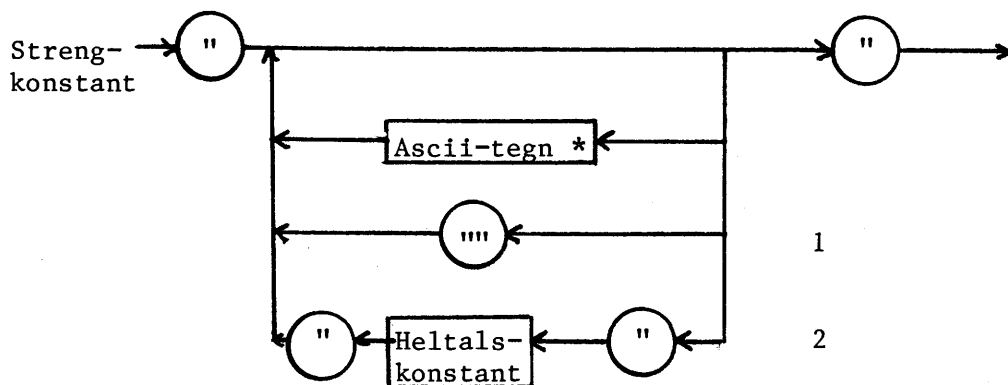
1 -3278

Reelle tal

348 2. 87.13E-14 -11.34

5.3.2 Strengkonstanter

Strengene benyttes som konstanter i strengudtryk, i inddata, i ledetekster ved brug af INPUT-sætningen samt i CASE- og DATA-sætninger. En strengkonstant har følgende opbygning:



* Her må et vilkårligt synligt ascii-tegn, undtagen anførselstegn ("), forekomme.

Som det ses af diagrammet, startes og afsluttes enhver strengkonstant med et anførselstegn. Ønsker man, at en strengkonstant skal indeholde et anførselstegn, angives dette med 2 anførselstegn umiddelbart efter hinanden (1). Ønsker man, at en strengkonstant skal indeholde tegn, som ikke kan indtastes fra tastaturet, kan man angive tegnets decimale ascii-kode omgivet af anførselstegn (2). Der må ikke forekomme blanktegn mellem tallet og anførselstegnene på venstre og højre side af det.

Eksempler:

"TASTES ""S"" STANDSER PROGRAMMET"



Ovenstående strengkonstant vil, hvis den eksempelvis udskrives på en dataskærm, se således ud:

```
TASTES "S" STANDSER PROGRAMMET
```

strengkonstanten

```
"ABC"13"
```

indeholder de 3 tegn ABC samt tegnet med den decimale ascii-kode 13 (CR).

5.3.3 Identifikatorer

Identifikatorer benyttes til navngivning af forskellige størrelser i et COMAL-80 program. En identifikator må maksimalt indeholde 16 tegn, hvoraf det første skal være et bogstav. De tilladte tegn i en identifikator er:

- små og store bogstaver
- cifre
- understregning

Små bogstaver omformes af COMAL-80 til store bogstaver, hvorfor identifikatorer, der eksempelvis indtastes som

Hansen hhv.
HANSEN

af COMAL-80 vil opfattes som værende identiske.

Eksempler på legale identifikatorer:

IND_LØN
F16
Anne

Eksempler på illegale identifikatorer:

1ABC (Første tegn er ikke et bogstav).

IND-LØN (Tegnet "-" må ikke forekomme i en identifikator).



5.3.4 Numeriske variable

I COMAL-80 findes 2 typer af numeriske variable, nemlig:

- heltalsvariable og
- reelle variable

Heltalsvariable kan tildeles heltallige værdier, og lagres i 2 oktetter (bytes). Reelle variable kan tildeles reelle værdier og lagres i 4 oktetter, 1 til eksponent og 3 til taldel.

Værdiområdet for heltal er således:

$$-32767 \leq X \leq 32767$$

og det absolutte værdiområde for reelle tal er:

$$2.9387359E-39 < X < 1.7014117E38$$

samt 0.

Systemets reaktion ved overskridelse af værdi-området er beskrevet i afsnit 3.6.5.

Numeriske variable kan optræde dels som simple variable og dels som talsæt.

Simple variable

En simpel variabel skal ikke erklæres, idet systemet afsætter plads til den, første gang den tildeles en værdi.

En simpel reel variabel refereres ved den tilhørende identifikator, mens identifikatoren for en simpel heltalsvariabel skal afsluttes af tegnet '#'. Således vil variablen:

LØN

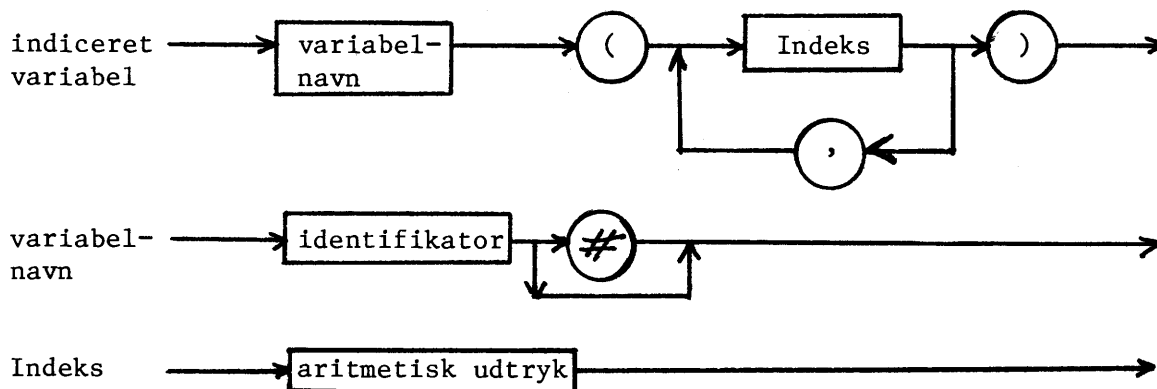
være en simpel reel variable, mens

POSTNUMMER#

vil være en simpel heltalsvariabel.

Talsæt

Et talsæt består af en samling indicerede variable organiseret som en vektor eller en matrix. En indiceret variabel har følgende opbygning:



'Identifikator' er talsættets navn, og '#' benyttes som for simple variable.

Et aritmetisk udtryk er et udtryk, som antager en aritmetisk værdi.

Et talsæt skal erklæres før det anvendes. Erklæringen foretages i en DIM-sætning, som ved udførelsen afsætter plads til talsættet og initialiserer talsættets elementer til 0.

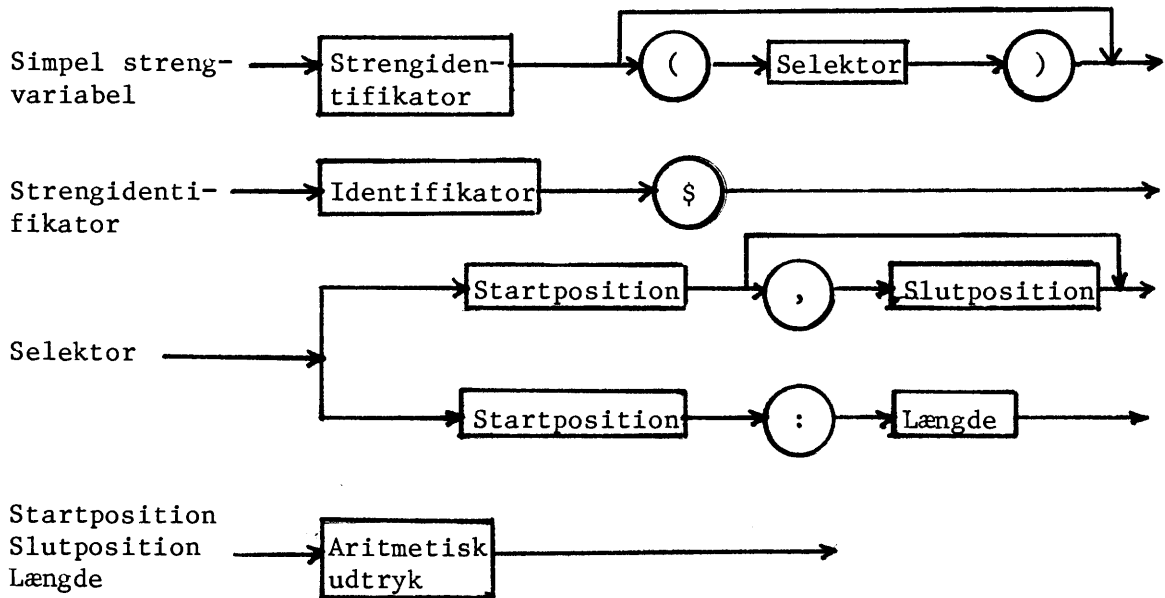
Ved alle referencer til talsæt foretages indekscheck, dvs. det undersøges, om antallet af indeks og deres værdier er tilladelige. Hvis resultatet af beregningen af et indeks giver en ikke-hel værdi, foretages en afrunding.

5.3.5 Strengvariable

Strengvariable kan i COMAL-80 optræde dels som simple variable og dels som indicerede variable.

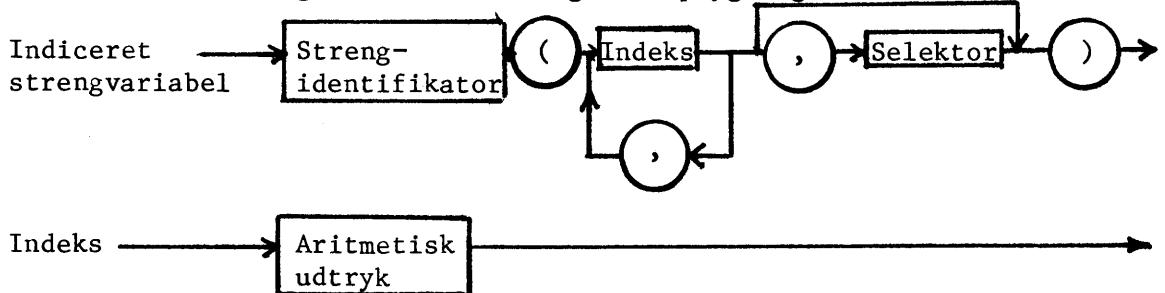
Simple variable

En simpel strengvariabel har følgende opbygning:



Indicerede variable

En indiceret strengvariabel har følgende opbygning:





Kommentarer

Som det ses af diagrammerne kendes strengvariable fra de numeriske variable ved, at der umiddelbart efter identifikatoren følger et "\$".

Forekommer en simpel strengvariabel uden selektor betyder dette, at der refereres til hele variabelens indhold. Er der angivet en selektor, betyder dette en delstreng startende med 'startposition' og sluttende med 'slutposition' eller ('startposition' + 'Længde'-1) afhængigt af hvilket af de 2 selektor-formater, der er benyttet. Angives kun 'startposition' i en selektor er 'slutposition' = 'startposition' underforstået.

Eksempler:

Antag at NAVN\$ er en simpel strengvariabel. Da er

NAVN\$(1,4) : Tegn nr. 1-4 i NAVN\$
NAVN\$(5) : Tegn nr. 5 i NAVN\$
NAVN\$(5:3) : Tegn nr. 5-7 i NAVN\$

For en indiceret strengvariabel skal indices angives for at specificere et ønsket element. Herud over kan der eventuelt være angivet en selector, hvis betydning er som beskrevet for simple strengvariable.

Eksempel:

Antag at NAVNE\$ er en streng-matrix med 2 rækker og 3 søjler. Rækker og søjler er nummereret fra 1 og fremefter. Da er

NAVNE\$(1,3) : Element nr. 3 i række 1.
NAVNE\$(1,2,3,4): Tegn nr. 3-4 i element nr. 2 i række nr. 1.

Strengvariable skal erklæres i en DIM-sætning inden de benyttes. Ved udførelsen af en sådan sætning afsættes lagerplads til strengvariablen. Den aktuelle længde af en tegnstreng er lig med antallet af tegn i strengen.

Ved referencer til indicerede tegnvariable samt tegnvariable med selektor-angivelse, foretages indekscheck. Ved indekscheck undersøges om antallet af indices og deres værdier samt værdierne i en eventuel selektorangivelse er tilladelige. Hvis resultatet af beregningen af et udtryk, som angiver et indeks eller indgår i en selektor, giver en ikke-hel værdi, foretages en afrunding.

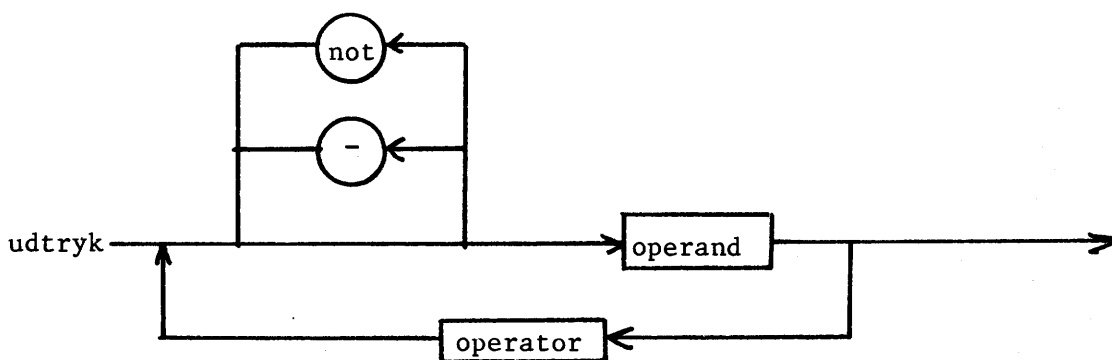
I denne vejledning vil 'variabel' blive anvendt som betegnelse for en vilkårlig simpel- eller indiceret variabel.

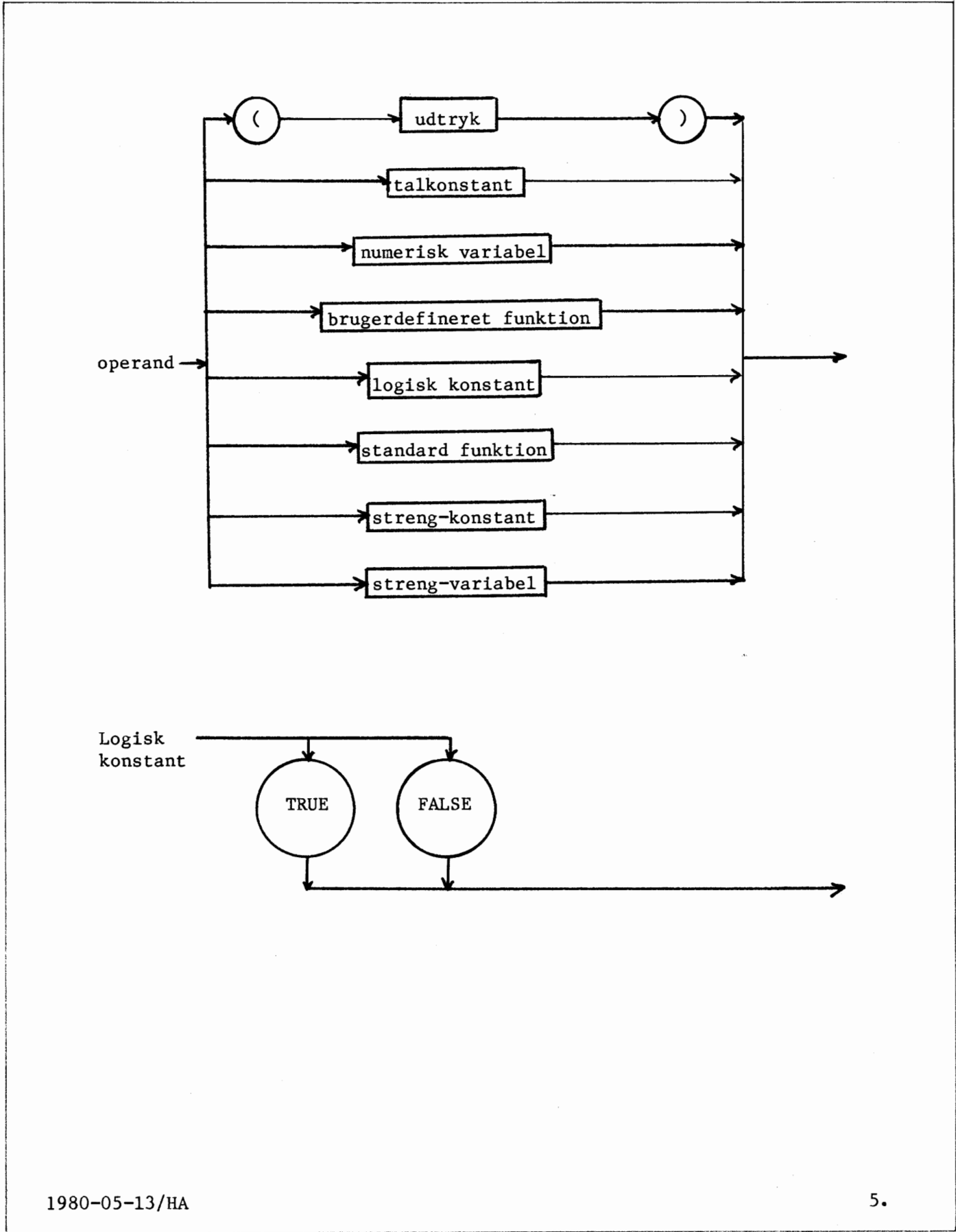
5.3.6 Udtryk

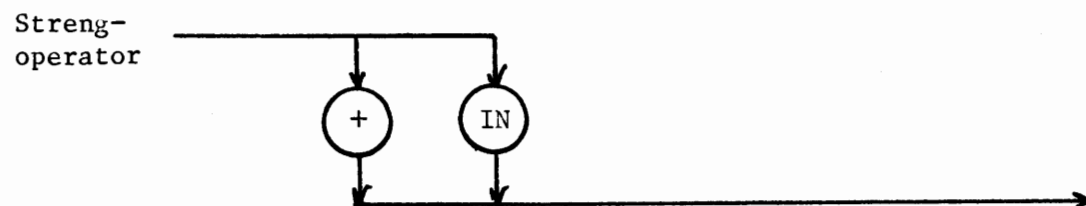
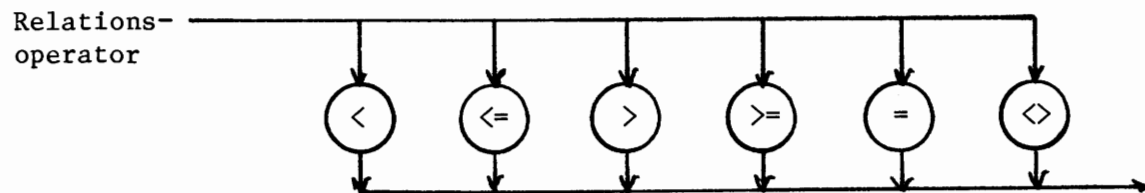
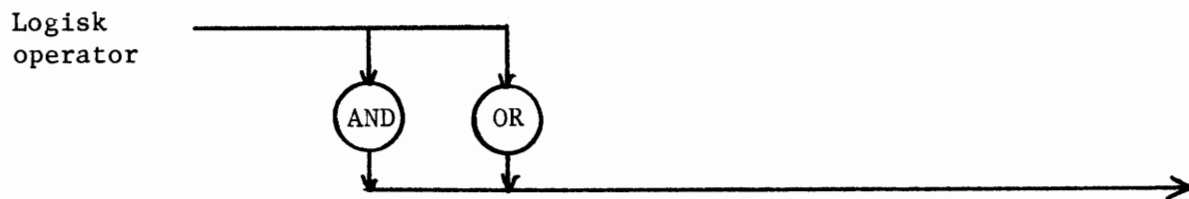
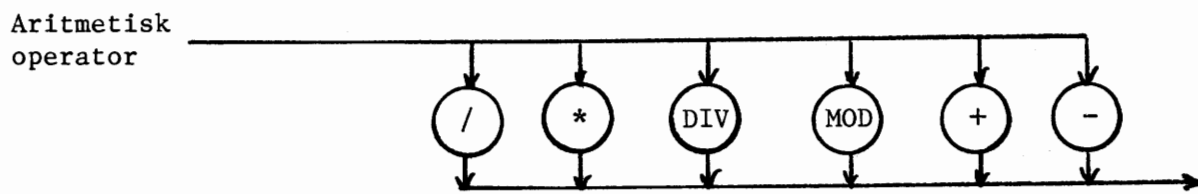
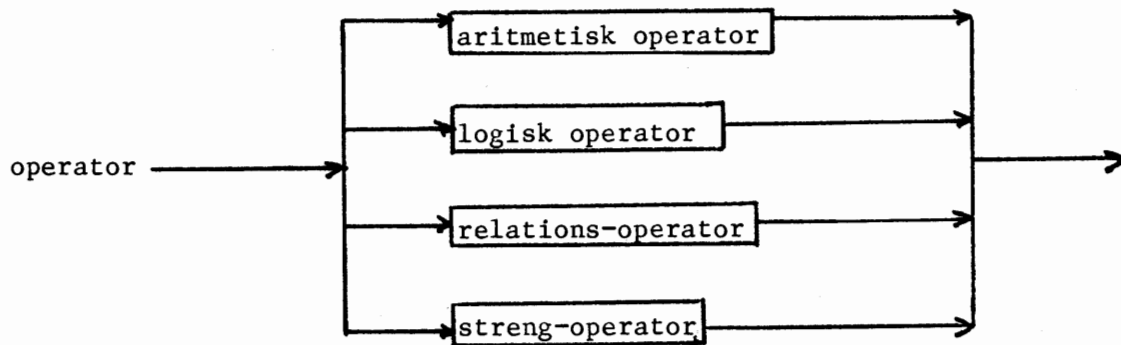
Udtryk indgår i en række forskellige sætninger. Resultatet af beregningen af et udtryk kan være:

- numerisk,
- logisk eller
- alfanumerisk

Et udtryk har følgende opbygning:







5.3.6.1 Aritmetiske operatorer

De dyadiske aritmetiske operatorer er følgende:

↑	:	potensopløftning
/	:	division
*	:	multiplikation
DIV	:	heltalsdivision
MOD	:	modulus
+	:	addition
-	:	subtraktion

De dyadiske aritmetiske operatorer må optræde mellem udtryk, som antager aritmetiske værdier.

Udover de dyadiske operatorer findes den monadiske aritmetiske operator '-' (foranstillet minus), der må optræde foran et udtryk, som antager en aritmetisk værdi.

Resultatet af anvendelse af en aritmetisk operator er en aritmetisk værdi.

Den aritmetiske værdi af et sandt logisk udtryk er 1. Den aritmetiske værdi af et falsk logisk udtryk er 0.

Kommentarer:

Værdien af $X \text{ DIV } Y$ er defineret som den heltallige værdi af N , som får udtrykket $(X - N * Y)$ til at antage sin mindste ikke-negative værdi.

Værdien af $X \text{ MOD } Y$ er defineret som den mindste ikke-negative værdi, som udtrykket $(X - N * Y)$ kan antage for heltallige N .

Eksempler:

11 DIV 7	= 1	11 MOD 7	= 4
-11 DIV 7	= -2	-11 MOD 7	= 3
-11 DIV (-7)	= 2	-11 MOD (-7)	= 3
11 DIV (-7)	= -1	11 MOD (-7)	= 4

5.3.6.2 Logiske operatorer

De dyadiske logiske operatorer er følgende:

AND: Logisk og
OR : Logisk eller

De dyadiske logiske operatorer må optræde mellem udtryk, som antager logiske værdier.

Udover de dyadiske operatorer findes den monadiske logiske operator "NOT" (negation), der må optræde foran et udtryk, som antager en logisk værdi.

Resultatet af anvendelse af en logisk operator er en logisk værdi.

Den logiske værdi af et aritmetisk udtryk, hvis værdi er nul, er 'falsk'.
Den logiske værdi af et aritmetisk udtryk, hvis værdi er forskellig fra nul, er 'sand'.



5.3.6.3 Relationsoperatorer

Relationsoperatorerne er følgende:

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

Relationsoperatorerne er alle dyadiske og må optræde mellem to udtryk, som begge antager en aritmetisk værdi, eller to udtryk, der begge antager en alfanumerisk værdi (tegnstreng).

Resultatet af anvendelsen af en relationsoperator er en logisk værdi. Hvis relationen er opfyldt er værdien 'sand', ellers 'falsk'.

Kommentarer

Sammenligningen af to tegnstreng foretages tegn for tegn fra venstre mod højre. Sammenligningen sker på grundlag af tegnenes ASCII-værdier (se afsnit 5.9).

Hvis to strenge af forskellig længde sammenlignes, og de første tegn i den længste streng er identiske med tegnene i den korteste, da vil den korteste streng være mindst. Eksempelvis er følgende relation sand:

"OLE" < "OLESEN"



5.3.6.4 Strengoperatorer

Strengoperatorerne er følgende:

+ : konkatenering (sammensætning)
IN : forekomst af en streng i en anden

Strengoperatorerne er begge dyadiske og må optræde mellem udtryk, der antager alfanumeriske værdier.

Resultatet af anvendelsen af '+' operatoren er en tegnstring, nemlig strengen på venstre side af operatoren forlænget med strengen på højre side af operatoren.

Eksempel:

"ABC" + "DE" = "ABCDE"

Resultatet af anvendelsen af 'IN'-operatoren er en logisk værdi. Hvis tegnstringen på venstre side af operatoren forekommer i tegnstringen på højre side af operatoren, er værdien 'sand', ellers falsk.

Eksempel:

"OMA" IN "AROMA" er 'sand'
"OMA" IN "MONA" er 'falsk'



5.3.6.5 Beregning af udtryk

Beregning af udtryk sker efter følgende regler:

1. Udtryk beregnes primært efter operatorernes prioritet og sekundært fra venstre mod højre. Operatorerne har følgende prioriteter:

Højest	- (monadisk)
	↑
	/ * DIV MOD
	+ - (dyadisk)
	< <= > >= = <> IN
	NOT
	AND
Lavest	OR

2. Hvis en operand er et udtryk omgivet af parenteser, beregnes dette udtryk til bunds før operatoren anvendes på operanden.

Eksempel:

1. $\frac{-3*2+3\uparrow("AB" \text{ IN } "ABC" \text{ AND } 3)}$
2. $\frac{-3*2+3\uparrow("AB" \text{ IN } "ABC" \text{ AND } 3)}$
3. $\frac{-3*2+3\uparrow(1 \text{ AND } 3)}$
4. $\frac{-3*2+3\uparrow 1}$
5. $\frac{-3*2+ 3}{-6 + 3}$
6. $\frac{-6 + 3}{-3}$

Understregningen viser hvilken beregning der udføres i det aktuelle trin.

Resulterende type ved beregning af udtryk

I skemaet herunder gives en oversigt over hvilke typer operanderne til de dyadiske operatorer må have, samt hvilken type resultatet får.

Operand type		↑	/	*	DIV MOD	+	-	IN	AND OR
venstre	højre								
streng	streng	-	-	-	-	streng*	heltal	-	-
heltal	heltal	reel	reel	heltal	heltal	heltal	-	heltal	heltal
heltal	reel	reel	reel	reel	reel	reel	-	heltal	heltal
reel	heltal	reel	reel	reel	reel	reel	-	heltal	heltal
reel	reel	reel	reel	reel	reel	reel	-	heltal	heltal

*) ikke '-'

Relationsoperatorerne <, <=, >, >=, =, <> må benyttes på vilkårlige par af operander, som antager streng-værdier og på vilkårlige par af operander, som antager aritmetiske værdier. Resultatet bliver et heltal 1 (sand) eller et heltal 0 (falsk).

Når der i et felt i skemaet ovenfor er skrevet et '-', betyder det, at den tilsvarende operator ikke må benyttes med det aktuelle par af operander.

Overskridelse af værdiområdet

Hvis den absolutte værdi af et udtryk, hvis resulterende type er heltallig, beregnes til en værdi, som er større end 32767, forekommer et overløb. Tilsvarende forekommer et overløb, hvis den absolutte værdi af et udtryk, hvis resulterende type er reel, er større end eller lig med 1.7014117E38.

I tilfælde af overløb, standses programudførelsen med en fejlmeddelelse, med mindre der er udført en TRAP_ERR - (afsnit 5.4.22).

Hvis den absolutte værdi af et udtryk, hvis resulterende type er reel, er mindre end 2.9387359E-39 uden at være 0, forekommer et underløb. I tilfælde af underløb sættes værdien af udtrykket til 0, hvorefter programudførelsen fortsætter.



Terminologi

I de følgende afsnit vil terminologien herunder blive anvendt:

Strengudtryk : et udtryk, hvis værdi er en tegnstreng
Aritmetisk udtryk: et udtryk, hvis værdi er et tal
Logisk udtryk : et udtryk, hvis værdi kan tolkes som værende 'falsk'
eller 'sand'.



5.4. COMAL-80 sætninger

5.4.1 Indledning

Et COMAL-80 program består af et antal programsætninger. Hver programsætning består af et linienummer og en COMAL-80 sætning. Linienummeret bestemmer programsætningens placering i programmet, og COMAL-80 sætningen indeholder en instruks til COMAL-80 systemet.

En COMAL-80 sætning indledes med et nøgleord, som identificerer sætningstypen.

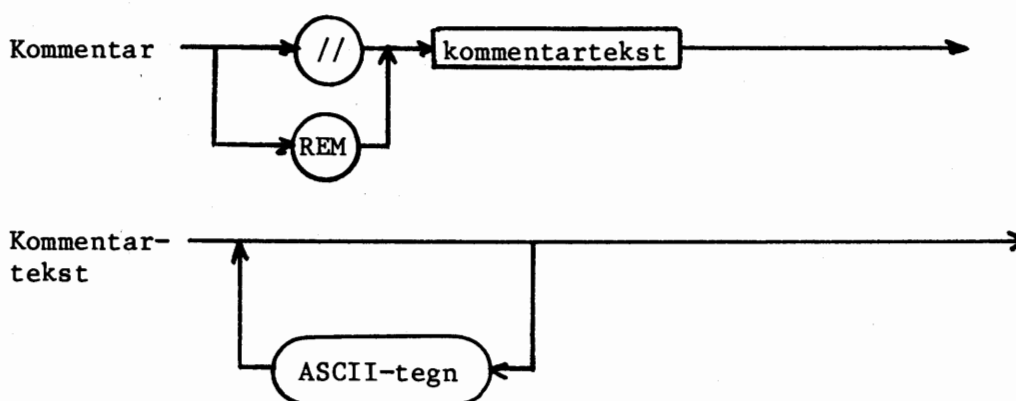
COMAL-80 sætningerne gennemgås i det følgende.

5.4.2 Kommentarer

Man har mulighed for at føje kommentarer ind i et COMAL-80 program. Kommentarer kan optræde som egentlige kommentarsætninger eller som kommentarer, der afslutter andre COMAL-80 sætninger.

Syntaks

En kommentar har følgende opbygning:



Udførelse

Kommentarer har ingen indflydelse på programudførelsen.

Kommentarer

I programudskriften konverteres 'REM' til '//'.

```

10 REM HOVEDPROGRAM
20 EXEC CPRNR // CPRNR INDLÆSES OG KONTROLLERES

```

Eksempler:

```

10 REM HOVEDPROGRAM
20 EXEC CPRNR // CPRNR INDLÆSES OG KONTROLLERES

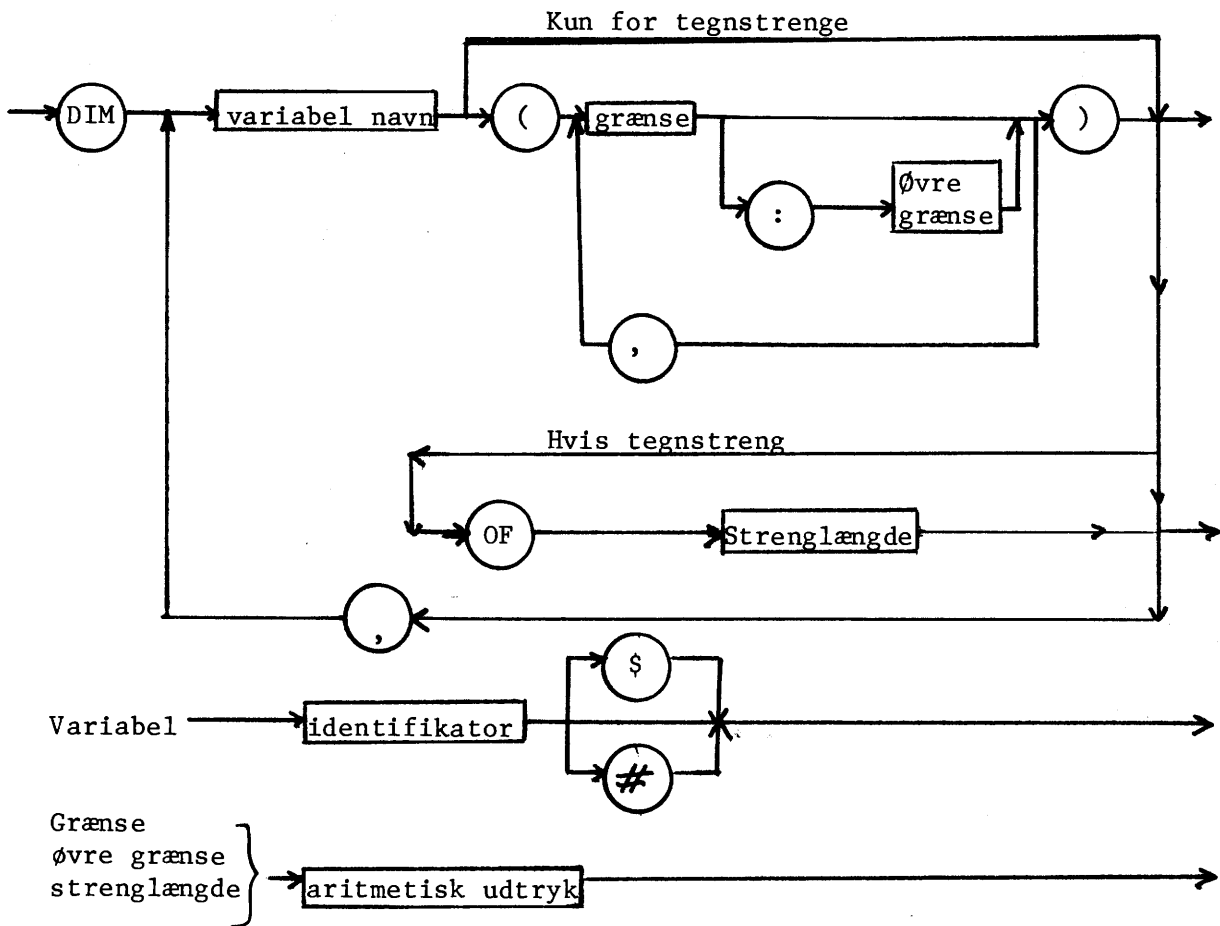
```

5.4.3 Erklæringer

Erklæring af talsæt, tegnstrege samt indicerede tegnstrege foretages ved hjælp af DIM-sætningen.

Syntaks

DIM-sætningen har følgende opbygning:



Udførelse

Ved udførelse af DIM-sætningen afsættes plads i dataområdet til de størrelser, der erklæres.

Kommentarer

1. Dimensionen af et variabel-sæt kan være vilkårligt stor og er kun begrænset af arbejdslagerets størrelse.
2. Indeksgrænser angives på et af følgende to formater:

1 : 'nedre grænse' : 'øvre grænse'
2 : 'øvre grænse'

Følgende ulighed skal være opfyldt:

'nedre grænse' \leq 'øvre grænse'

Hvis format nr. 2 anvendes, er en nedre grænse på 1 underforstået.

3. Hvis resultatet af beregningen af 'grænse', 'øvre grænse' eller 'strenglængde' ikke giver et helt tal, foretages en afrunding.
4. Erklæringer skal udføres, inden de erklærede størrelser første gang benyttes i programmet.
5. Alle elementer i et erklæret talsæt gives værdien 0, mens (elementerne i) en erklæret strengvariabel gives værdien 'tom-streng' ("").
6. En variabel må kun erklæres én gang.

Eksempler:

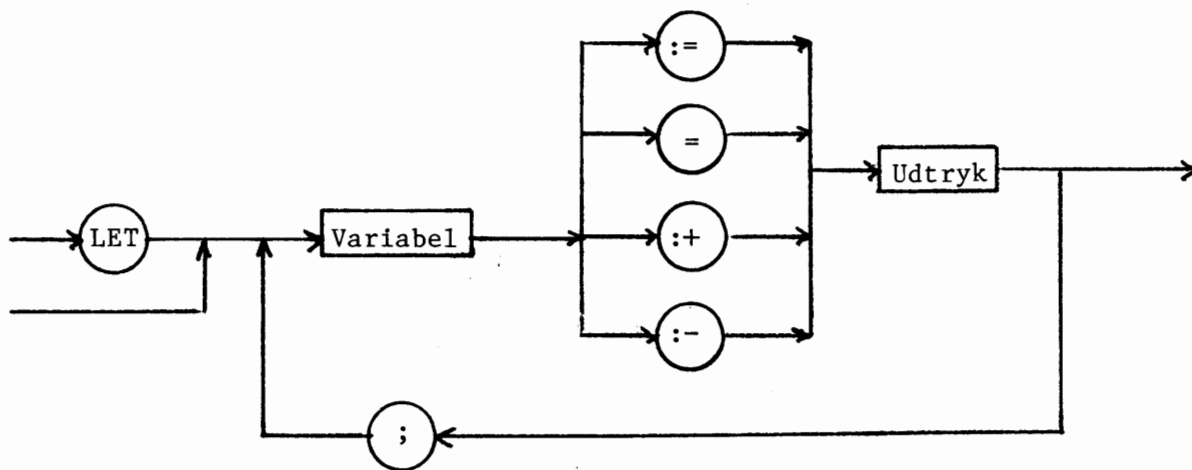
DIM NAVN\$ OF 20, HELTALSTABEL*(3,8)
DIM NAVNE\$(2:4,10) OF 20
DIM REEL_TABEL(10:20, 10:15)

5.4.4 TILDELINGER

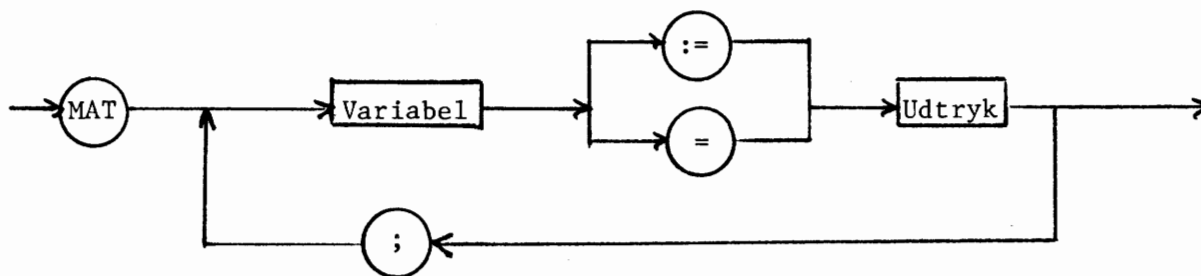
Tildeling af værdier til variable foretages i tildelingssætninger. Der findes 2 forskellige tildelingssætninger, nemlig LET- og MAT-sætningen.

Syntaks

LET-sætningen har følgende opbygning:



MAT-sætningen har følgende opbygning:



Udførelse

For hver tildeling i en LET-sætning udføres følgende:

- a. Adressen på variabelen på venstre side af ':= ' beregnes.
- b. Udtrykket beregnes.
- c. Variabelen på venstre side af ':= ' tildeles resultatet af trin b.

'=' og ':= ' er ækvivalente.

'Variabel' :+ 'udtryk'

er ækvivalent med

'variabel' := 'variabel' + 'udtryk'

og

'variabel' :- 'udtryk'

er ækvivalent med

'variabel' := 'variabel' - 'udtryk'

Eksempel:

Tildelingerne

```
TÆLLER := TÆLLER + 1
FORNAVN$ := FORNAVN$ + "-ERIK"
```

kan kort skrives

```
TÆLLER :+ 1
FORNAVN$ :+ "-ERIK"
```

I MAT-sætningen tildeles værdien af 'udtryk' til alle elementer i 'variabel', som skal være et talsæt eller et sæt af tegnstreng.

Også i MAT-sætningen er ':= ' og '=' ækvivalente.

Kommentarer

1. Typen for 'variabel' og 'udtryk' skal stemme overens. Hermed menes, at kun kombinationer, som i tabellen herunder er markeret med et '+', er tilladte.

Type for 'variabel' \ Type for 'udtryk'	Heltal	Reel	Streng
Heltal	+	-	-
Reel	+	+	-
Streng	-	-	+

2. For tildelinger til streng-variable gælder følgende regler:
 - a. Hvis 'udtryk' er længere end 'variabel', vil 'udtryk' blive afkortet, idet kun den første del bliver benyttet.
 - b. Hvis 'udtryk' er kortere end 'variabel' vil 'udtryk' blive venstrestillet i 'variabel'.
3. For tildelinger til delstreng gælder følgende regler:
 - a. 'delstreng' og 'udtryk' skal have samme længde.
 - b. Startpositionen for en delstreng skal være mindre end eller lig med strengens længde plus 1 ('startposition' \leq LEN ('strengvariabel')+1).

Er a og b ikke opfyldt, standser programudførelsen med en fejlmeddelelse.

Eksempler

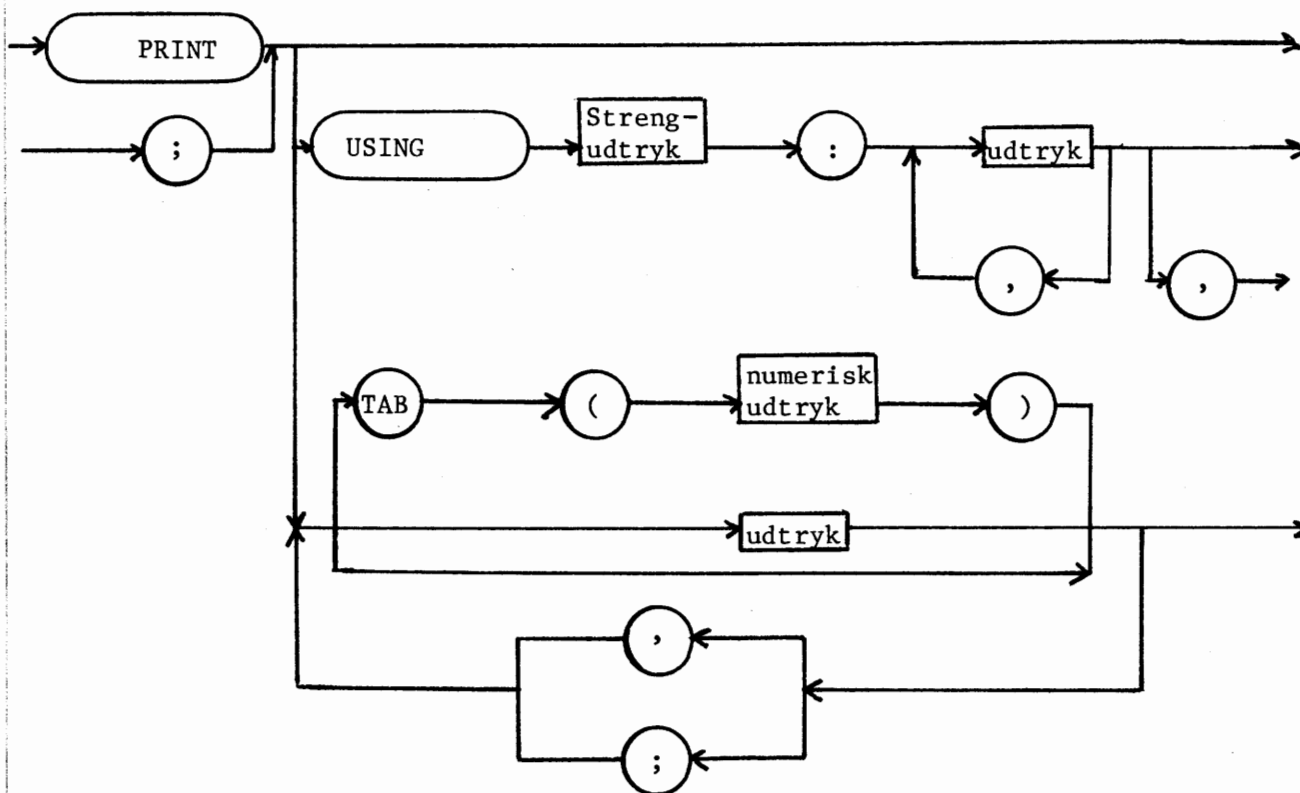
```
LET TÆLLER#:+1; SUM=0  
MAT TALTABEL :=0; NAVNE$ := ""
```

5.4.5 PRINT-sætning

PRINT-sætningen benyttes til udskrivning af resultater i form af tal eller tegnstreng på data skærm eller lineskriver (se afsnit 5.4.16).

Syntaks

PRINT-sætningen har følgende opbygning:



Udførelse

- a. Udføres en PRINT-sætning, som kun består af nøgleordet 'PRINT', medfører dette, at der laves en 'vognretur' samt et lineskift.
- b. Efterfølges 'PRINT' af nøgleordet 'USING', tilkendes herved, at et bruger-defineret udskrivningsformat skal benyttes. Dette udskrivningsformat er angivet i 'streng-udtryk'.

Tegnene i dette udtryk fortolkes på følgende måde:

- '#' : cifferposition og fortegn
- '.' : decimalpunkt (kun hvis omgivet af '#')
- '+' : foranstillet plus (kun hvis '#' følger umiddelbart efter)
- '-' : foranstillet minus (kun hvis '#' følger umiddelbart efter)

Alle andre tegn overføres direkte til udskriften. Hvis tallet er for stort til at være inden for det opgivne format, udfyldes formatet med '*'-er. Hvis tallet har flere decimaler end angivet i udskrivningsformatet, foretages en afrunding inden udskrivningen.

Begyndes et format med et '+', betyder det, at der afsættes en position til tallets fortegn, og at fortegn udskrives både for positive og negative tal.

Begyndes et format med et '-' betyder det, at der afsættes en position til tallets fortegn, og at fortegn kun udskrives for negative tal.

Et numerisk udskrivningsformat kan også benyttes til udskrivning af tegnstreng. I dette tilfælde angiver hvert tegn i formatet en position i tegnstrengens udskrivningsformat. Ved udskrivning venstrestilles tegnstrengen i formatet. Er tegnstrengen for lang, afskæres et antal tegn i strengens højre ende. Er tegnstrengen for kort til at udfylde formatet, efterstilles et antal blanktegn.

PRINT USING-sætningen udføres ved, at formatstrengen gennemløbes tegn for tegn fra venstre mod højre. Alle tegn, som ikke indgår i et udskrivningsformat, udskrives direkte på skærm eller linieskriver. Hver gang et udskrivningsformat mødes, konverteres og udskrives det næste 'udtryk' i udtrykslisten i overensstemmelse med det opgivne format. Hvis der ikke er flere 'udtryk' i udtrykslisten, standses udførelsen af PRINT USING-sætningen, dvs. resten af formatstrengen ignoreres. Hvis der er flere 'udtryk' i udtrykslisten, end der er formater i formatstrengen, begyndes der forfra på formatstrengen.

Afsluttes PRINT USING-sætningen af et ',', vil næste udskrivning fortsætte på den aktuelle linie. I modsat fald skiftes til ny linie inden næste udskrivning.



Eksempel:

```
10 PRINT USING "-###": -3,44.47,100,"ABC"  
20 PRINT USING "ETTAL ## OG EN TEGNSTRENG #":1,"ABC",  
30 PRINT " SLUT!"
```

Køres ovenstående program fås følgende udskrift:

```
- 3.0 44.5 ***.* ABC  
ETTAL 1 OG EN TEGNSTRENG AB SLUT!
```

Programudførelse afsluttet

- c. Efterfølges 'PRINT' af en række udtryk eller tabulatorangivelser adskilt af ',' eller ';', tilkendes gives herved, at standard udskrivningsformatet i COMAL-80 skal benyttes.

For reelle tal er standard-formatet følgende:

- tal i området

$$1 \leq |x| \leq 99999999$$

udskrives på decimal form uden eksponent og med maksimalt 7 cifre.

- decimaltal, hvor alle cifre efter decimalpunktet er nul, udskrives som heltal uden decimalpunkt.
- efterstillede nuller i decimaldel undertrykkes.
- eventuelt 0 foran decimalpunkt udskrives.
- tal i området

$$0 < |x| < 0.0000001 \text{ (1E-7) eller } |x| \geq 10000000 \text{ (1E7)}$$

udskrives med eksponent og 7 cifre i mantissen, heraf 1 ciffer foran decimalpunkt.

Skilletegnene ',' og ';' har indflydelse på udskriftens placering på linien.



Anvendes skilletegnet ';' mellem to elementer i udskrivningslisten, vil elementerne blive udskrevet i umiddelbar forlængelse af hinanden. ';' forandrer således ikke den aktuelle skriveposition.

Enhver udskrivningslinie er opdelt i et antal lige store zoner, hvis størrelse defineres af brugeren på følgende måde:

TAB := 'numerisk udtryk'

'Numerisk udtryk' angiver antallet af skrivepositioner i hver zone. COMAL-80 har på forhånd tildelt TAB værdien 0, hvilket giver ',' samme effekt som ';'. Denne værdi vil være gældende, indtil brugeren ændrer indholdet af TAB som vist ovenfor.

Anvendes skilletegnet ',' mellem to elementer i udskrivningslisten, vil udskrivning af elementet efter skilletegnet starte i næste zones første skriveposition.

Af syntaksdiagrammet kan ses, at det er tilladt at skrive flere skilletegn umiddelbart efter hinanden. Eksempelvis er

PRINT 1,,2;;3;,4

en tilladt konstruktion. For hvert ',', som mødes, rykkes den aktuelle skriveposition frem til første position i den efterfølgende zone. Overskydende ';' -er har ingen effekt.

Afsluttes en PRINT-sætning af et ',' eller ';' vil det første element i den næste PRINT-sætning blive udskrevet på samme linie i overensstemmelse med reglerne for skilletegn.

Afsluttes en PRINT-sætning ikke af et ',' eller ';' vil udskrivning af første element i den næste PRINT-sætning starte i første position på næste linie.

TAB er en tabulator-funktion, som bevirker, at næste element udskrives i den position, som argumentet angiver. Positionerne på linien nummereres fra 1 til og med 64. Er den aktuelle skriveposition større end argumentet for TAB-funktionen, standses programudførelsen med en fejlmeddelelse.

I stedet for nøgleordet 'PRINT' er det tilladt blot at skrive et ';'. I programudskrifter konverteres et sådant ';' til 'PRINT'.



Eksempel

```
0010 TAB :=5 // 5 positioner i hver zone
0020 PRINT"12345678901234567890"
0030 ; 3, "LU" ; "DO", 13.84
0040 PRINT, 123,
0050 PRINT "ABC", TAB(15); "SLUT!"
```

Køres ovenstående program fås følgende udskrift:

```
12345678901234567890
3   LUDO 13.84
    123ABC SLUT!
    123 ABC SLUT!
```

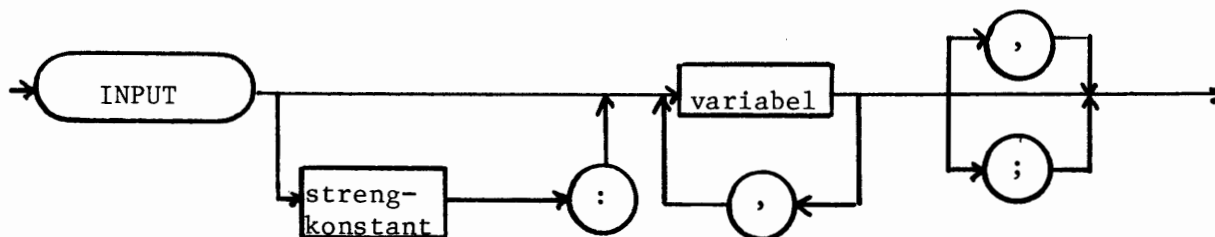
Programudførelse afsluttet

5.4.6 INPUT-SÆTNING

INPUT-sætningen benyttes til at indlæse værdier fra terminalen og tildele disse til variable.

Syntaks

INPUT-sætningen har følgende opbygning:



Udførelse

Hvis INPUT-sætningen indeholder en 'strengkonstant' udskrives denne på dataskærmen; ellers udskrives tegnet ':' for at indikere, at systemet forventer inddata.

Brugeren indtaster herefter et antal værdier, som variablerne i INPUT-sætningen skal tildeles. Indtastningen afsluttes ved, at man trykker på 'RETURN'-tasten.

Listen af variable i INPUT-sætningen gennemløbes fra venstre mod højre, og de tilsvarende indtastede værdier tildeles.

Hvis variabellisten afsluttes med et ',' eller et ';' gælder følgende:

',' : Næste skriveposition er første tegn i inddataliniens næste zone (se afsnit 5.4.5).

';' : Næste skriveposition er umiddelbart efter det sidst indtastede tegn i inddatalinien.

Kommentarer

1. Talkonstanter indtastes på den form, som er beskrevet i afsnit 5.3.1. Foranstillet '+' er dog tilladt. Flere talkonstanter adskilles med blanktegn.



2. En strengkonstant indtastes blot som en række af ASCII-tegn. Det er ikke muligt at indtaste værdier på inddatalinien efter en strengkonstant, idet resten af linien vil blive opfattet som værende en del af strengen. En strengkonstant, som indtastes efter en talkonstant, starter med det første tegn, som ikke kan indgå i tallet.
3. Indlæsning af værdier til strengvariable følger samme regler som tildelinger (se afsnit 5.4.4).

Eksempel

```
0010 INPUT "STYKTAL": STYKTAL#  
0020 INPUT "STYKPRIS": STYKPRIS  
0030 PRINT "TOTALPRIS:"; STYKTAL**STYKPRIS
```

Køres ovenstående program fås følgende skærmdialog, hvor brugersvar er understreget.

```
STYKTAL 10  
STYKPRIS 33.5  
TOTALPRIS 335
```

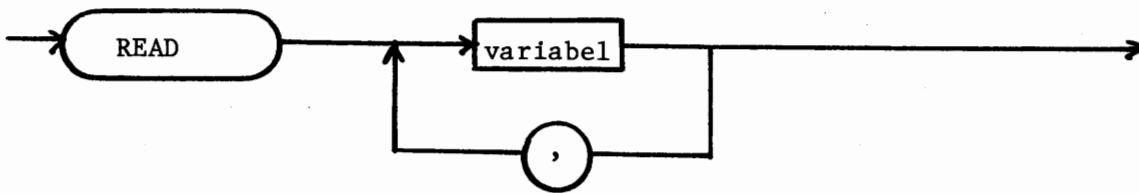
Programudførelse afsluttet

5.4.7 READ-sætning

READ-sætningen benyttes til at tildele startværdier til variable.

Syntaks

READ-sætningen har følgende opbygning:



Udførelse

Ved udførelsen af READ-sætningen vil variablerne i variabellisten få tildelt værdier fra de næste værdier i datalisten. Opbygningen af datalisten er beskrevet i afsnit 5.4.8.

Kommentarer

1. Hvis den læste værdis type ikke stemmer overens med den angivne variabels type, eller hvis datalisten er udtømt, stoppes programudførelsen med en fejludskrift.
2. Tildeling af værdier til strengvariable v.h.a. READ-sætningen følger samme regler, som beskrevet i afsnit 5.4.4.

Eksempel

```
0010 DIM FORNAVN$ OF 10, EFTERNAVN$ OF 10
0020 DATA "HANS", "OLSEN", 10
0030 READ FORNAVN$, EFTERNAVN$
0040 PRINT FORNAVN$+" "+EFTERNAVN$
0050 READ ALDER
0060 PRINT ALDER; "ÅR"
```



Køres ovenstående program fås følgende resultat:

HANS OLSEN
10 ÅR

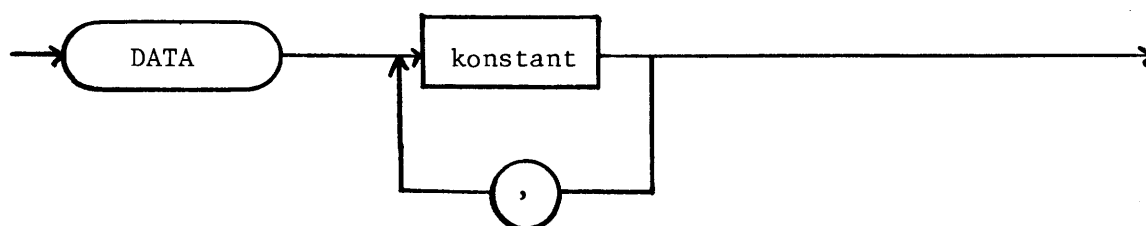
Programudførelse afsluttet

5.4.8 DATA-sætning

DATA-sætningen benyttes til at definere værdier i datalisten. Disse værdier kan indlæses v.h.a. READ-sætninger.

Syntaks

DATA-sætningen har følgende opbygning:



Udførelse

Ved starten af programmets udførelse, dvs. umiddelbart efter indtastning af RUN-kommandoen, gennemgås programmet i linienummerorden, og alle DATA-sætninger kædes sammen i en dataliste. Datalisten styres v.h.a. en "pegepind", som altid udpeger den næste værdi, som skal tildeles. Denne "pegepind" bliver sat til at udpege datalistens første værdi.

Under selve programudførelsen ignoreres DATA-sætningerne.

Kommentarer

1. Konstanter skal have den form, som er beskrevet i afsnit 5.3.1 og 5.3.2.

Eksempel

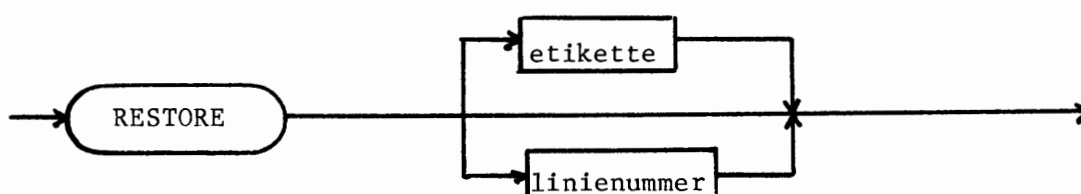
Se afsnit 5.4.7.

5.4.9 RESTORE-sætning

RESTORE-sætningen benyttes til at flytte "pegepinden" i datalisten, som er omtalt i afsnit 5.4.8.

Syntaks

RESTORE-sætningen har følgende opbygning:



Udførelse

Udførelse af RESTORE-sætningen uden etikette eller linienummerangivelse bevirker, at "pegepinden" flyttes til datalistens første værdi.

Efterfølges RESTORE af et linienummer skal den refererede linie indeholde en DATA-sætning. Udførelse af en sådan RESTORE-sætning bevirker, at "pegepinden" flyttes til den første værdi i den angivne DATA-sætning.

En etikette refererer til en linie, som er blevet navngivet i en LABEL-sætning (se afsnit 5.4.10). Efterfølges RESTORE af en etikette, skal linien, som følger umiddelbart efter den pågældende LABEL-sætning, være en DATA-sætning. Udførelse af en sådan RESTORE-sætning bevirker, at "pegepinden" flyttes til den første værdi i DATA-sætningen.



Eksempel

```
0005 DIM LINIE$ OF 10
0010 DATA "LINIE 10"
0020 LABEL LINIE_20
0030 DATA "LINIE 30"
0040 READ LINIE$
0050 PRINT LINIE$
0060 READ LINIE$
0070 PRINT LINIE$
0080 RESTORE LINIE_20
0090 READ LINIE$
0100 PRINT LINIE$
0110 RESTORE 10
0120 READ LINIE$
0130 PRINT LINIE$
0140 RESTORE
0150 READ LINIE$
0160 PRINT LINE$
```

Udføres ovenstående program fås følgende udskrift:

```
LINIE 10
LINIE 30
LINIE 30
LINIE 10
LINIE 10
```

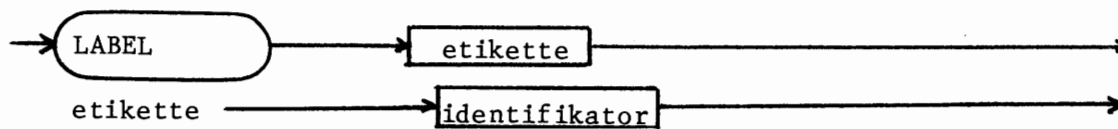
Programudførelse afsluttet

5.4.10 LABEL-sætning

LABEL-sætningen benyttes til at navngive en linie. Linien kan således refereres til ved sit navn i forbindelse med RESTORE- og GOTO-sætninger.

Syntaks

LABEL-sætningen har følgende opbygning:



Udførelse

LABEL-sætningen ignoreres under selve programudførelsen.

Eksempler

```

0010 LABEL FORFRA
      .
      .
0180 GOTO FORFRA
  
```

Se også eksemplet i afsnit 5.4.9.



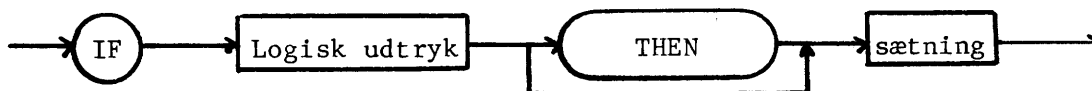
5.4.11 Betingede sætninger

COMAL-80 rummer muligheder for at gøre udførelsen af en række sætninger betinget af værdien af et logisk udtryk.

Der findes i COMAL-80 en række mulige konstruktioner for betingede sætninger; disse konstruktioner gennemgås enkeltvis i det følgende.

5.4.11.1 IF

Syntaks



Udførelse

Er 'logisk udtryk' sand udføres 'sætning', ellers ignoreres 'sætning'.

Kommentarer

1. 'Sætning' må ikke indeholde:

- en erklæring (DIM, PROC/ENDPROC, DEF/ENDEF, LABEL)
- en DATA-sætning
- en del af en betinget sætning (ELIF, ELSE, ENDIF)
- en del af en sammensat sætningskonstruktion (REPEAT/UNTIL, WHILE/ENDWHILE, CASE/WHEN/OTHERWISE/ENDCASE, FOR/NEXT)
- et END

'Sætning' må derimod gerne indeholde en ny IF-THEN-sætning.

Eksempler

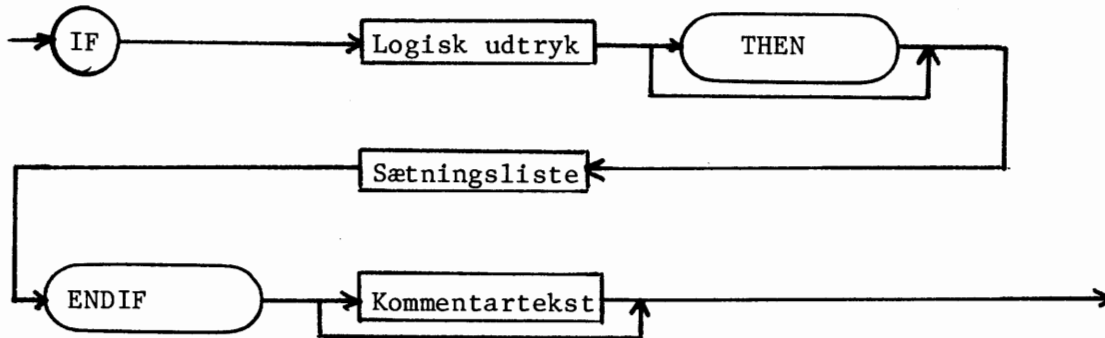
Følgende IF-THEN-sætninger er korrekte:

```

IF A <=100 THEN IF A >=90 PRINT "OK"
IF TAL THEN PRINT "TAL <> 0"
  
```

5.4.11.2 IF-ENDIF

Syntaks



'Sætningsliste' betegner et vilkårligt antal COMAL-80 sætninger.

Udførelse

Er 'logisk udtryk' sand udføres sætningerne i 'sætningsliste'. Er 'logisk udtryk' falsk fortsætter programudførelsen i sætningen umiddelbart efter ENDIF-sætningen.

Kommentarer

1. Hvis en sætningsliste indeholder en eller flere sammensatte sætninger, dvs. sætninger, som strækker sig over flere linier, skal disse sætninger afsluttes inden for sætningslisten.

Eksempler

```

0010 TAL1=3; TAL2=4
0020 IF TAL1 <> TAL2 THEN
0030   PRINT "TAL1 <> TAL2"
0040   PRINT "SUMMEN ER LIG MED "; TAL1+TAL2
0050 ENDIF UNDERSØGELSE SLUT
0060 END
    
```

Køres ovenstående program fås følgende udskrift:

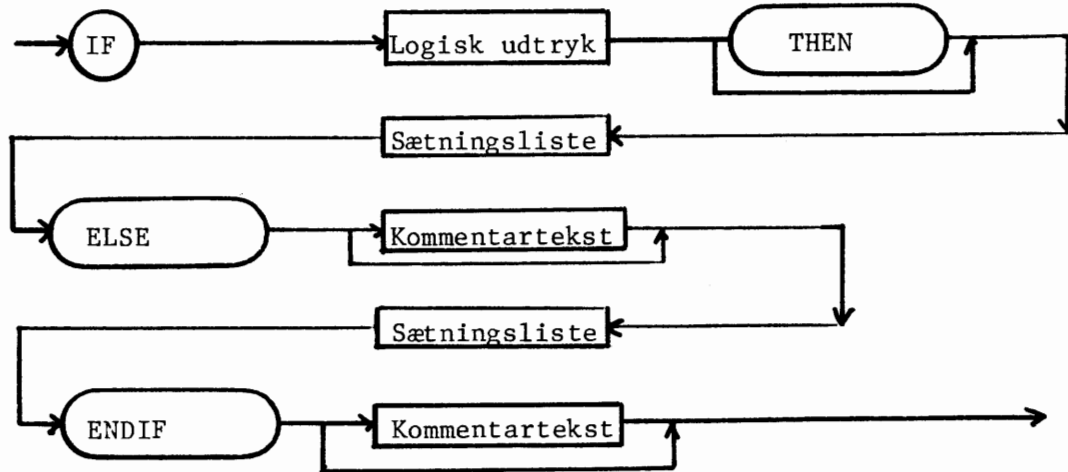
```

TAL1 <> TAL2
SUMMEN ER LIG MED 7
    
```

END i linie 60

5.4.11.3 IF-ELSE-ENDIF

Syntaks



Udførelse

Er 'logisk udtryk' sand, udføres sætningerne i sætningslisten efter THEN, hvorefter programudførelsen fortsætter med den første sætning efter den tilhørende ENDIF-sætning.

Er 'logisk udtryk' falsk, udføres sætningerne i sætningslisten efter ELSE.

Kommentarer

1. Hvis en sætningsliste indeholder en eller flere sammensatte sætninger, dvs. sætninger som strækker sig over flere linier, skal disse sætninger afsluttes inden for sætningslisten.

Eksempler

```
0010 INPUT "3+5 = " : SUM
0020 IF SUM = 8 THEN
0030   PRINT "RIGTIGT"
0040 ELSE
0050   PRINT "FORKERT"
0060   PRINT "RIGTIGT SVAR ER 8."
0070 ENDIF
```

Køres ovenstående program fås følgende skærmdialog, hvor brugersvar er understreget.

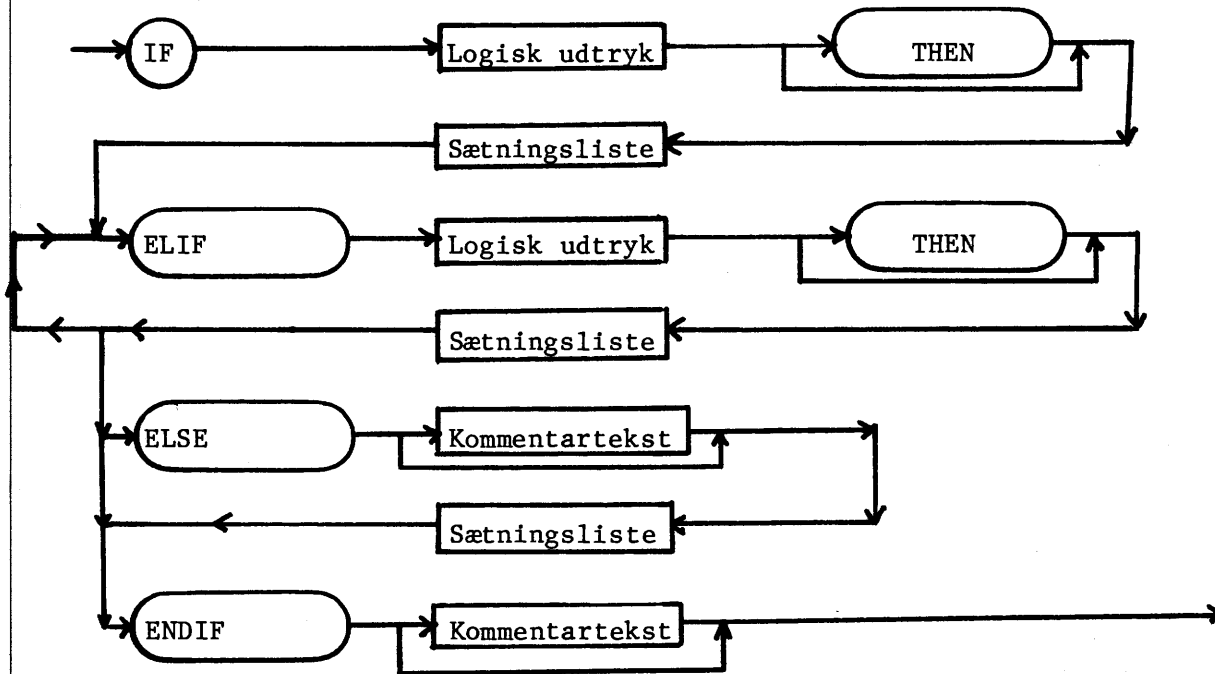
```
3+5 = 9
FORKERT
RIGTIGT SVAR ER 8.
```

Programudførelse afsluttet

5.4.11.4 IF-ELIF-(ELSE)-ENDIF

Nøgleordet ELIF er en forkortelse for ELSE IF.

Syntaks



Udførelse

Udførelsen skitseres v.h.a. 2 rutediagrammer herunder:

IF udtryk THEN



ELIF udtryk-1 THEN



ELIF udtryk-2 THEN

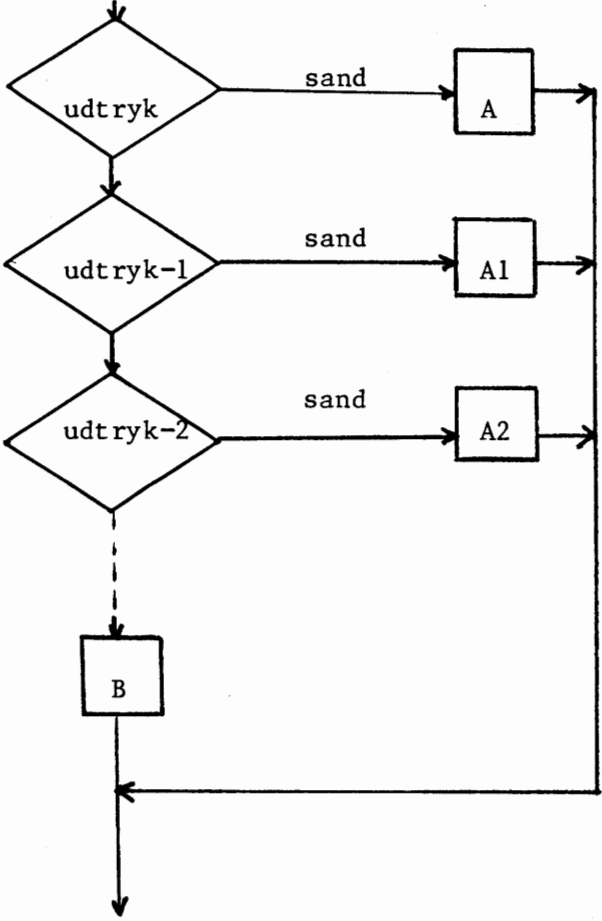


⋮

ELSE



ENDIF



IF udtryk THEN

A

ELIF udtryk-1 THEN

A1

ELIF udtryk-2 THEN

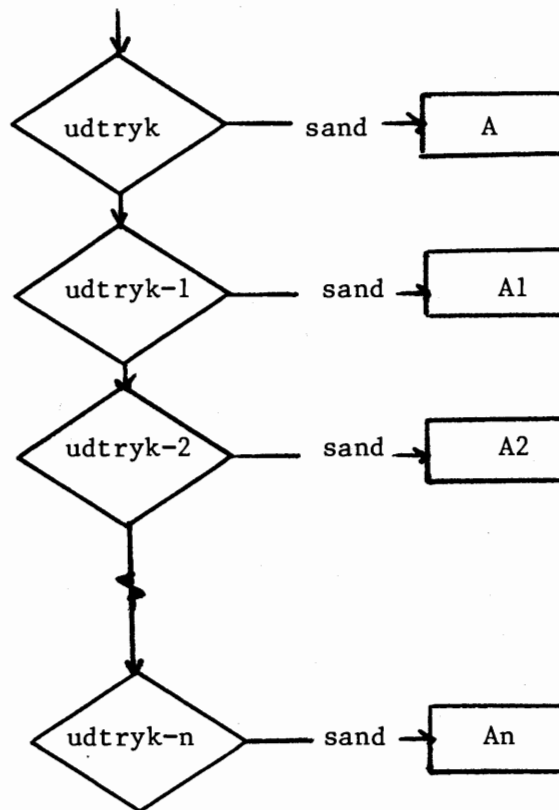
A2

⋮

ELIF udtryk-n THEN

An

ENDIF



A, A1, A2, An og B betegner sætningslister. Læg specielt mærke til, at selvom flere af udtrykkene udregnes til værdien sand, vil kun sætningslisten efter det første af disse udtryk blive udført.

Kommentarer

1. Hvis en sætningsliste indeholder en eller flere sammensatte sætninger, dvs. sætninger som strækker sig over flere linier, skal disse sætninger afsluttes inden for sætningslisten.

Eksempler

```
0010 INPUT "INDTAST ET TAL " : TAL
0020 IF TAL > 1000 THEN
0030   PRINT "TAL > 1000"
0040 ELIF TAL < 0 THEN
0050   PRINT "NEGATIVT TAL"
0060 ELIF TAL > 0 THEN
0070   PRINT "0 < TAL <= 1000"
0080 ELSE
0090   PRINT " TAL = 0"
0100 ENDIF
```

Køres ovenstående program fås følgende skærmdialog, hvor brugersvar er understreget.

```
INDTAST ET TAL 10
0<TAL<=1000
```

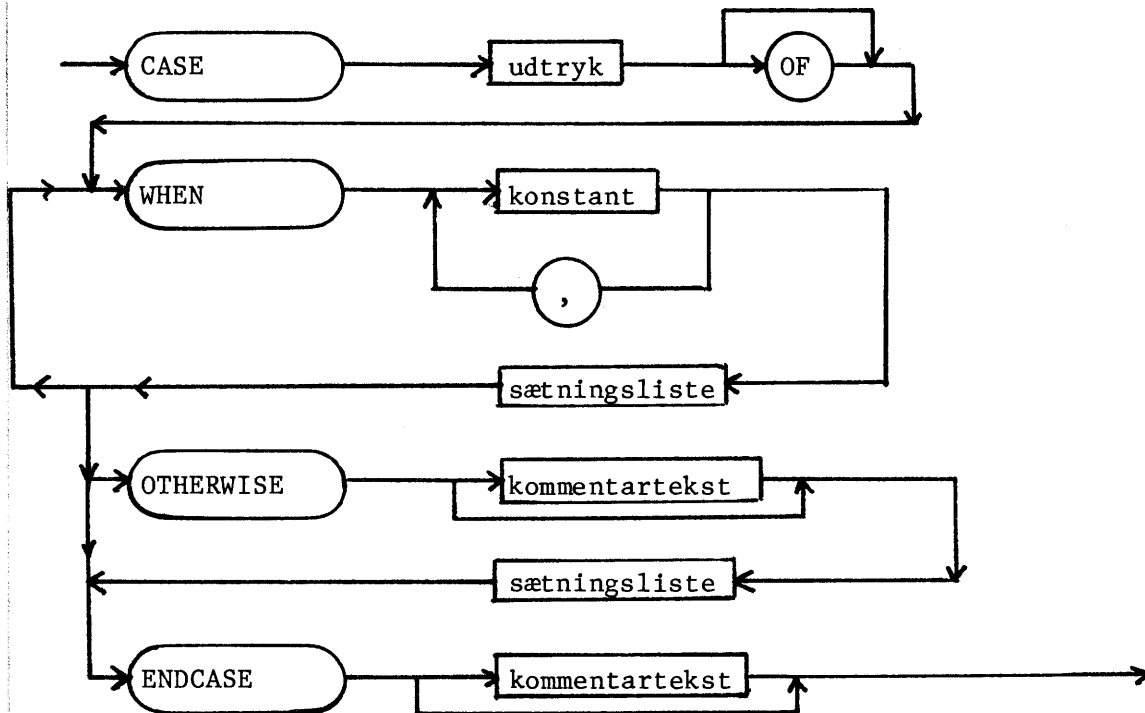
Programudførelse afsluttet

5.4.12 CASE-sætning

CASE-sætningen benyttes, når een af flere sætningslister skal udføres afhængigt af værdien af et udtryk.

Syntaks

CASE-sætningen har følgende opbygning:



'Sætningsliste' betegner et vilkårligt antal COMAL-80 sætninger.

Udførelse

'Udtryk' beregnes. Herefter gennemløbes konstanterne i WHEN-sætningerne, indtil der findes en værdi, som er identisk med værdien af det beregnede udtryk. Hvis en sådan identisk værdi er fundet, udføres sætningslisten, svarende til den pågældende WHEN-sætning, hvorefter udførelsen fortsætter efter ENDCASE.



Hvis der ikke blandt konstanterne i WHEN-sætningerne findes en værdi, som er identisk med det beregnede udtryks værdi, udføres den alternative sætningsliste efter OTHERWISE-sætningen. Indeholder CASE-sætningen ingen alternativ sætningsliste, vil programudførelsen i dette tilfælde standse med en fejlmeddelelse.

Kommentarer

1. Konstanterne i WHEN-sætningerne skal have samme type, som det beregnede udtryk.
2. Konstanter skal have den form, som er beskrevet i afsnit 5.3.1 og 5.3.2.
3. Hvis en sætningsliste indeholder en eller flere sammensatte sætninger, dvs. sætninger som strækker sig over flere linier, skal disse sætninger afsluttes inden for sætningslisten.

Eksempel:

```
0010 DIM TAL$ OF 1
0020 INPUT "INDTAST ET TAL " : TAL$
0030 CASE TAL$ OF
0040 WHEN "0"
0050   PRINT "DU HAR INDTASTET ET NUL"
0060 WHEN "1", "2", "3", "5", "7"
0070   PRINT "DU HAR INDTASTET ET PRIMTAL"
0080 WHEN "4", "8"
0090   PRINT "DU HAR INDTASTET EN POTENS AF TO"
0100 WHEN "6"
0110   PRINT "DET VAR ET SEKSTAL"
0120 WHEN "9"
0130   PRINT "DET VAR ET NITAL"
0140 OTHERWISE IKKE NOGET TAL
0150   PRINT "JEG BAD DIG OM AT INDTASTE ET TAL!"
0160 ENDCASE
0170 STOP
```



Herunder vises skærmdialogen ved 2 forskellige kørsler af ovenstående program. Brugersvar er understregede.

1. INDTAST ET TAL A
JEG BAD DIG OM ÅT INDTASTE ET TAL!
STOP i linie 0170
2. INDTAST ET TAL 4
DU HAR INDTASTET EN POTENS AF TO
STOP i linie 0170



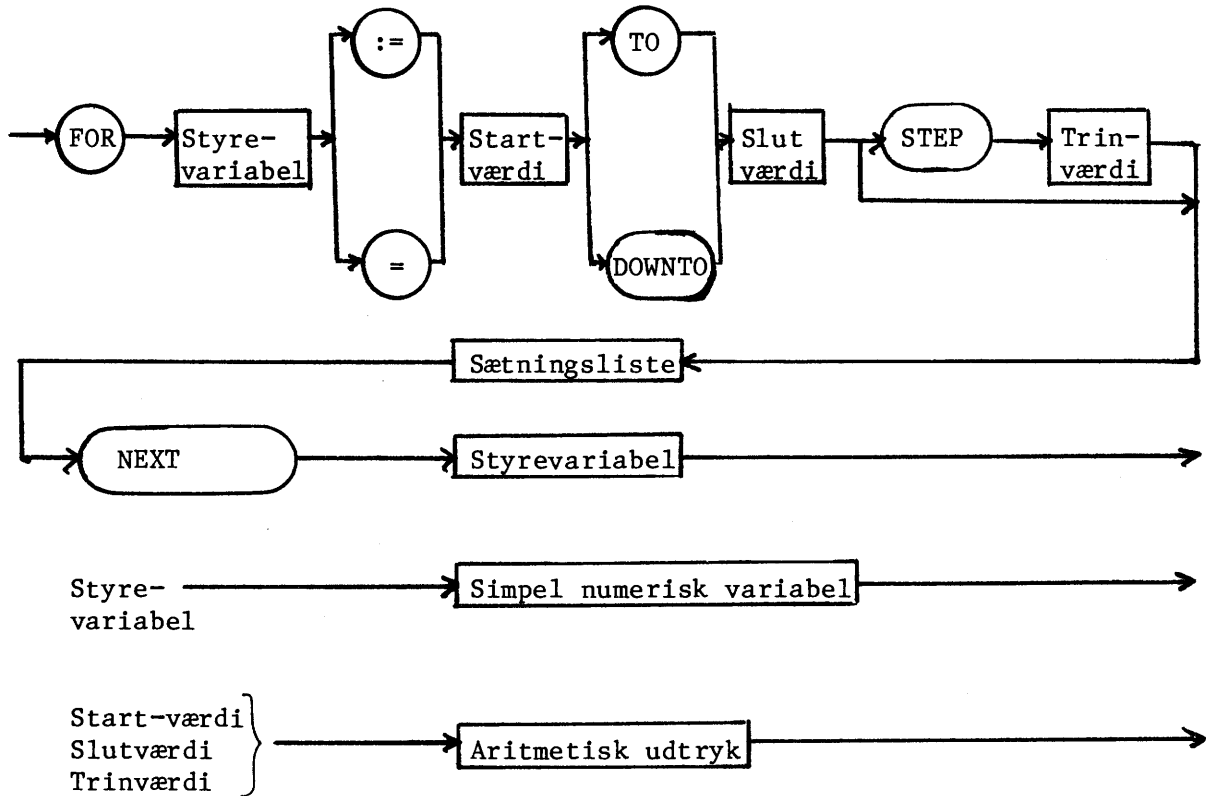
5.4.13 Repeterende sætninger

I COMAL-80 har man mulighed for at udføre en række sætninger et antal gange afhængigt af en stop-betingelse. Der findes 3 forskellige typer af repeterende sætninger, og hver af disse gennemgås i det følgende.

5.4.13.1 FOR-NEXT-sætning

Syntaks

FOR-NEXT-sætningen har følgende opbygning::



'Sætningsliste' betegner et vilkårligt antal COMAL-80 sætninger.

Udførelse

- a. Først beskrives udførelsen af en FOR-NEXT-sætning, hvor nøgleordet 'TO' er benyttet.

Er trinværdien ikke angivet benyttes værdien 1 som trinværdi.



Ved udførelsen af en FOR-sætning vil styrevariablen få tildelt startværdien. Herefter undersøges om de opgivne start-, slut- og trinværdier har nogen mening, dvs. om

$$(\text{slutværdi} - \text{startværdi}) * \text{SGN}(\text{trinværdi}) \geq 0$$

Er dette ikke tilfældet overspringes FOR-NEXT-sætningen.

Ved udførelsen af den tilsvarende NEXT-sætning bliver trinværdien adderet til værdien af styrevariablen. Såfremt styrevariablen nye værdi ligger i området mellem start- og slutværdien, fortsættes programudførelsen i den første sætning efter FOR-sætningen. I modsat tilfælde fortsættes med sætningen efter NEXT-sætningen.

- b. Er nøgleordet 'DOWNTO' benyttet i FOR-sætningen, udføres FOR-NEXT-sætningen fuldkommen som beskrevet i punkt a., men med en negeret trinværdi. Dvs.

benyttet trinværdi = -trinværdi

Kommentarer

1. Der skal være overensstemmelse mellem styrevariablen type og typen af start-, slut- og trinværdi på følgende måde:

<u>Styrevariablen</u> <u>type</u>	<u>Tilladt type for</u> <u>start-, slut- og trinværdi</u>
heltal	heltal
reel	reel/heltal

2. Hvis sætningslisten indeholder en eller flere sammensatte sætninger, dvs. sætninger, som strækker sig over flere linier, skal disse sætninger afsluttes inden for sætningslisten.

Eksempler

```
0010 SUM := 0
0020 FOR TAL# := 1 TO 500
0030 SUM :=+ TAL#
0040 NEXT TAL#
0050 PRINT SUM
```



```
0010 SUM := 0
0020 FOR TAL#:= 500 DOWNT0 1
0030 SUM :+ TAL#
0040 NEXT TAL#
0050 PRINT SUM
```

Begge ovenstående programmer beregner og udskriver summen af de hele tal fra 1 til 500, begge incl.

```
0010 Y := 3
0020 FOR X = 1.2 TO 4.7 STEP Y/2
0030 PRINT X
0040 NEXT X
```

Udføres ovenstående program fås følgende udskrift:

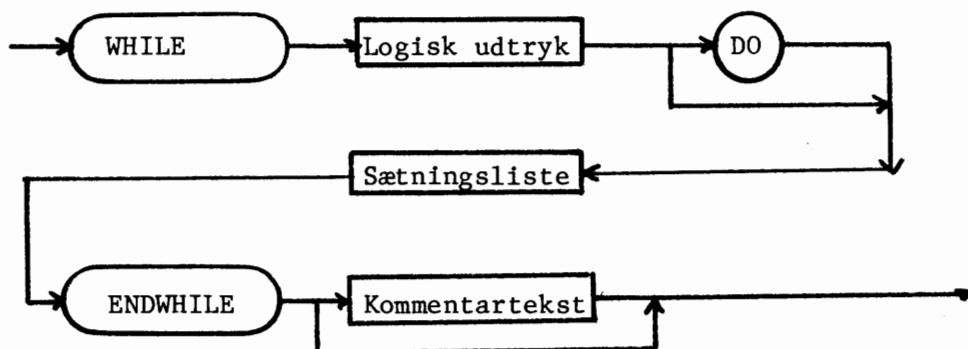
```
1.2
2.7
4.2
```

Programudførelse afsluttet

5.4.13.2 WHILE-ENDWHILE-sætning

Syntaks

WHILE-ENDWHILE-sætningen har følgende opbygning:



'Sætningsliste' betegner et vilkårligt antal COMAL-80 sætninger.

Udførelse

Når en WHILE-sætning udføres, udregnes værdien af det logiske udtryk efter WHILE. Er udtrykkets værdi sand, fortsættes programudførelsen i den efterfølgende sætning. Er udtrykkets værdi falsk, fortsættes programudførelsen i sætningen umiddelbart efter den tilhørende ENDWHILE-sætning.

Udførelsen af en ENDWHILE-sætning bevirker blot, at programudførelsen fortsættes i den tilhørende WHILE-sætning.

Kommentarer

1. Hvis sætningslisten indeholder en eller flere sammensatte sætninger, dvs. sætninger, som strækker sig over flere linier, skal disse sætninger afsluttes inden for sætningslisten.

Eksempel

```
0010 // BEREGNING AF KVADRATRØDDER
0020 X = 10
0030 DELTA:= X
0040 SQRT:= X/2
0050 // ITERATION
0060 WHILE ABS(DELTA) >0.001 DO
0070   DELTA:= (X/SQRT-SQRT)/2
0080   SQRT:+ DELTA
0090 ENDWHILE SLUT PÅ ITERATIONSLØKKE
0100 PRINT SQRT
```

Udføres ovenstående program fås følgende udskrift:

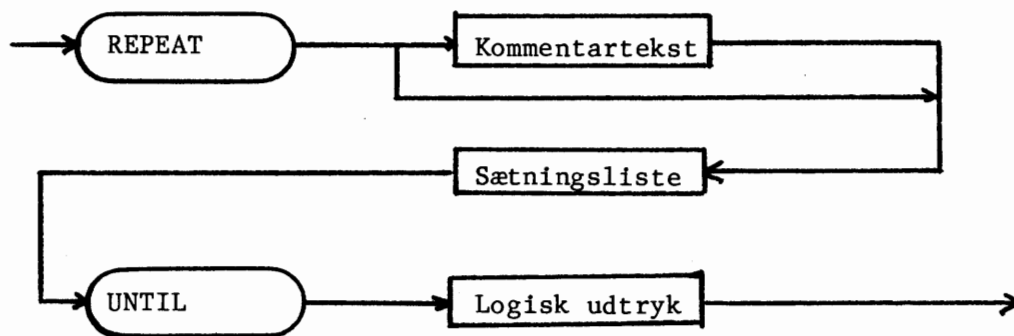
3.162278

Programudførelse afsluttet

5.4.13.3 REPEAT-UNTIL-sætning

Syntaks

REPEAT-UNTIL-sætningen har følgende opbygning:



'Sætningsliste' betegner et vilkårligt antal COMAL-80 sætninger.

Udførelse

REPEAT-sætningen markerer blot starten på en REPEAT-UNTIL-sætning, og udførelsen af den har ingen effekt.

Når en UNTIL-sætning udføres, udregnes det logiske udtryk efter UNTIL. Er udtrykkets værdi sand, fortsættes programudførelsen i sætningen umiddelbart efter UNTIL-sætningen. Er udtrykkets værdi falsk, fortsættes programudførelsen i sætningen umiddelbart efter den tilhørende REPEAT-sætning.

Kommentarer

1. Hvis sætningslisten indeholder en eller flere sammensatte sætninger, dvs. sætninger, som strækker sig over flere linier, skal disse sætninger afsluttes indenfor sætningslisten.

Eksempel

```
0010 // BEREGNING AF KVADRATRØDDER
0020 X := 10
0030 SQRT := X/2
0040 // ITERATION
0050 REPEAT
0060   DELTA := (X/SQRT-SQRT)/2
0070   SQRT :=+ DELTA
0080 UNTIL ABS(DELTA) <0.001
0090 PRINT SQRT
```

Udføres ovenstående program fås følgende udskrift:

3.162278

Programudførelse afsluttet

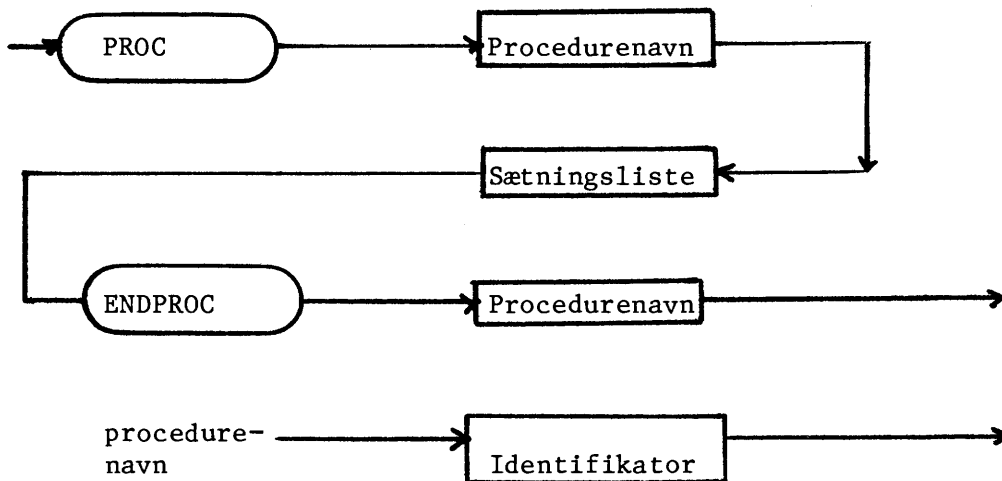
Sammenlign dette eksempel med eksemplet i afsnit 5.4.13.2.

5.4.14 Procedureerklæring (PROC-ENDPROC)

PROC- og ENDPROC sætningerne benyttes til at erklære en procedure.

Syntaks

En procedureerklæring har følgende opbygning:



'Sætningsliste' betegner et vilkårligt antal COMAL-80 sætninger.

Udførelse

Procedureerklæringen fastlægger, at den angivne sætningsliste skal opfattes som en procedure med navnet 'procedurenavn'.

Når en PROC-sætning mødes under programudførelsen, overspringes de følgende sætninger til og med den tilhørende ENDPROC-sætning, hvorefter programudførelsen fortsættes.

En procedure kaldes v.h.a. EXEC-sætningen. Returhop fra en procedure sker, når den tilhørende ENDPROC-sætning mødes.



Kommentarer

1. Hvis sætningslisten indeholder en eller flere sammensatte sætninger, dvs. sætninger, som strækker sig over flere linier, skal disse sætninger afsluttes inden for sætningslisten.
2. Procedureerklæringer må hverken forekomme inden i sammensatte sætninger eller inden i andre procedure- eller funktionserklæringer. Dvs. procedureerklæringer må ikke optræde i sætningslisterne for følgende konstruktioner:
 - Betingede sætninger
 - CASE-sætninger
 - Repeterende sætninger
 - Procedureerklæringer
 - Funktionserklæringer
3. Det er tilladt fra en procedure at kalde andre procedurer. Det er endog tilladt at lade en procedure kalde sig selv (rekursion).
4. Kommunikationen mellem en procedure og dens omgivelser sker via globale variable. Alle variable, som ikke optræder inden i en funktionserklæring, er globale.

Eksempel

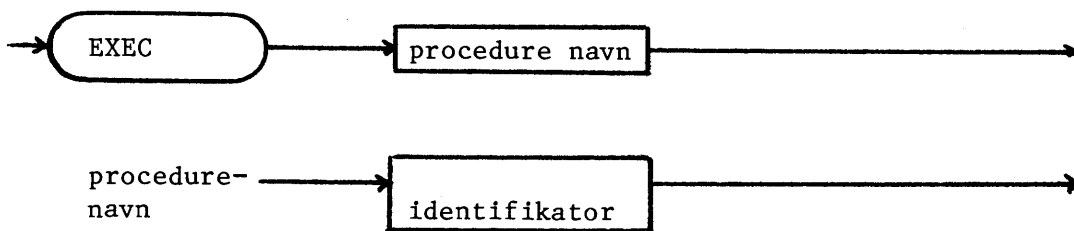
Se afsnit 5.4.15.

5.4.15 Procedurekald (EXEC-sætning)

Procedurekald benyttes til at få udført en procedure, som er defineret i en procedureerklæring.

Syntaks

Procedurekald har følgende opbygning:



Udførelse

Udførelse af EXEC-sætningen bevirker, at proceduren med navnet 'procedurenavn' kaldes. Programudførelsen starter i sætningen umiddelbart efter PROC-sætningen og fortsætter i proceduren, indtil ENDPROC-sætningen mødes. Herefter fortsætter programudførelsen i sætningen umiddelbart efter EXEC-sætningen.

Kommentarer

1. Det er tilladt, men ikke tilrådeligt, at afbryde udførelsen af en procedure v.h.a. en hop-sætning.

Eksempel

```

0010 PROC KVADRATSUM
0020   PRINT "X↑2 + Y↑2 =" ; X↑2 + Y↑2
0030 ENDPROC KVADRATSUM
0040 X := 3 ; Y := 5
0050 EXEC KVADRATSUM
0060 X := 8 ; Y := 11
0070 EXEC KVADRATSUM
  
```



Udføres ovenstående program, fås følgende udskrift:

$$X\uparrow 2 + Y\uparrow 2 = 34$$

$$X\uparrow 2 + Y\uparrow 2 = 185$$

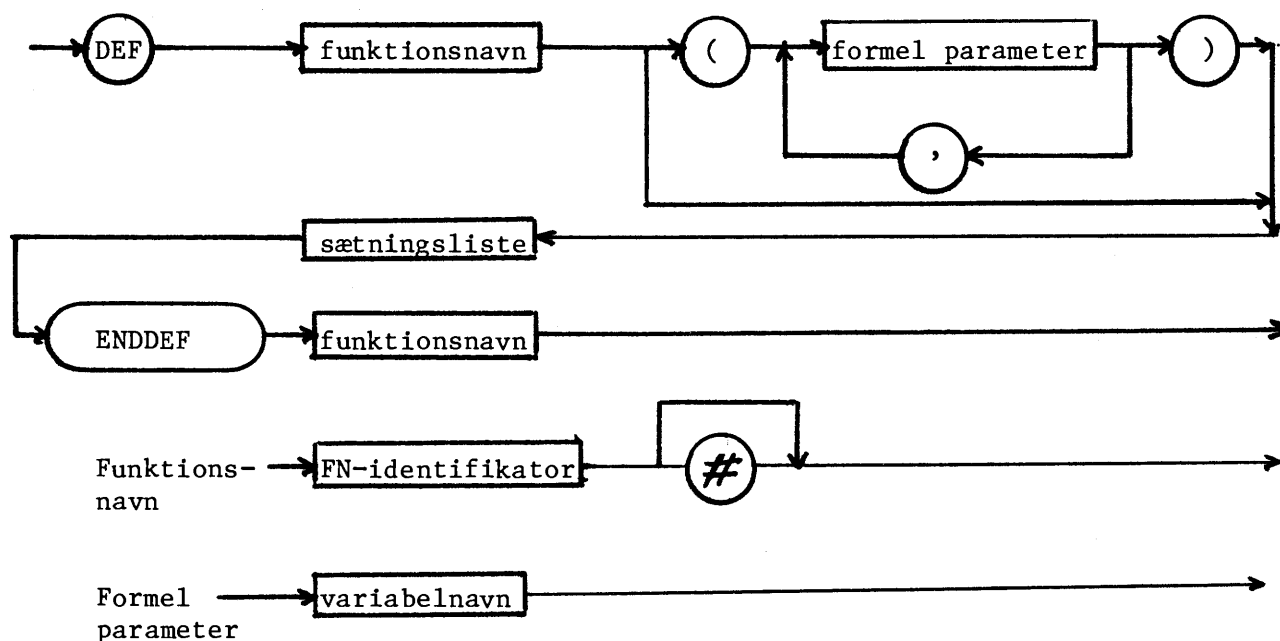
Programudførelse afsluttet

5.4.16 Funktionserklæring (DEF-ENDDEF)

DEF- og ENDDDEF-sætningerne benyttes til at erklære en brugerdefineret funktion.

Syntaks

En funktionserklæring har følgende opbygning:



'FN-identifikator' er en identifikator, hvis første 2 tegn er bogstaverne FN.

'Sætningsliste' betegner et vilkårligt antal COMAL-80 sætninger.

Udførelse

Funktionserklæringen fastlægger, at den angivne sætningsliste skal opfattes som en funktion med navnet 'funktionsnavn'.

Når en DEF-sætning mødes under programudførelsen, overspringes de følgende sætninger til og med den tilhørende ENDDDEF-sætning, hvorefter programudførelsen fortsættes.

En funktion kaldes, når funktionsnavnet, efterfulgt af en eventuel aktuel parameterliste, optræder i et udtryk. Kaldet bevirker, at de formelle parametre får tildelt værdierne af de tilsvarende aktuelle parametre, hvorefter sætningslisten udføres, indtil ENDDDEF-sætningen mødes. Funktionskaldet erstattes nu af den beregnede funktionsværdi, hvorefter beregning af udtrykket fortsætter.

Kommentarer

1. Hvis sætningslisten indeholder en eller flere sammensatte sætninger, dvs. sætninger, som strækker sig over flere linier, skal disse sætninger afsluttes for sætningslisten.
2. En funktions type kan enten være reel eller heltallig. Typen angives som for numeriske variable.
3. Funktionsværdien skal tildeles funktionsnavnet, inden ENDDDEF-sætningen mødes under programudførelsen; ellers er funktionsværdien undefineret.
4. Alle variable, som forekommer i en funktionserklæring, er lokale. Det vil sige, at selvom der i funktionserklæringen optræder variable, hvis navne er identiske med variabelnavne benyttet uden for funktionserklæringen, da er variablene ikke identiske. Den eneste mulighed for at overføre data fra det kaldende program til funktionen er således gennem parametrene.
5. M.h.t. hvor i programmet funktionserklæringer må forekomme, gælder samme regler som for procedureerklæringer. Der henvises til afsnit 5.4.14, kommentar 2.
6. Det er ikke tilladt fra en funktion at foretage en åbning eller lukning af en fil.
7. Det er tilladt fra en funktion at kalde andre funktioner. Det er tillige tilladt at lade en funktion kalde sig selv (rekursion).

Eksempel

```
0010 DEF FNFAKULTET#(TAL#)
0020   IF TAL# = 0 THEN
0030     FNFAKULTET#:=1
0040   ELSE
0050     FNFAKULTET#:= FNFAKULTET#(TAL#-1)*TAL#
0060   ENDIF
0070 ENDDF FNFAKULTET#
0080 PRINT FNFAKULTET#(6)
0090 PRINT FNFAKULTET#(4)
```

FNFAKULTET# beregner faktorieltfunktionen v.h.a. rekursion.

Udføres ovenstående program fås følgende udskrift:

```
720
24
```

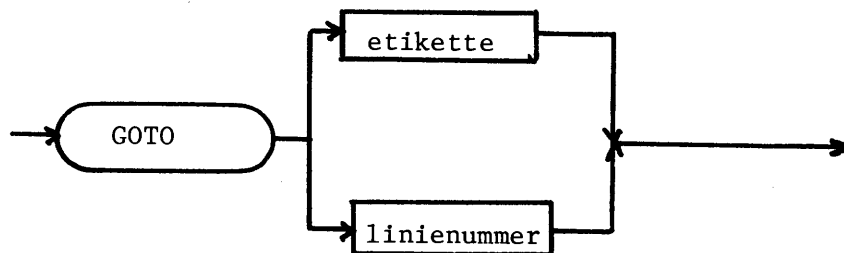
Programudførelse afsluttet

5.4.17 GOTO-sætning

GOTO-sætningen benyttes til at bryde den normale sekventielle programudførelse, for at fortsætte udførelsen i en angivet sætning.

Syntaks

GOTO-sætningen har følgende opbygning:



Udførelse

Udførelse af en GOTO-sætning medfører, at programudførelsen fortsættes i den angivne programlinie. 'Etikette' er navnet, som en linie er blevet tildelt i en LABEL-sætning (se afsnit 5.4.10).

Eksempel

```
0010 DIM BELØB(3)
0020 I:=1
0030 LABEL LÆS BELØB
0040 INPUT "INDTAST BELØB": BELØB(I)
0050 I:+1
0060 IF I < = 3 THEN GOTO LÆS BELØB
0070 PRINT "3 BELØB ER NU INDTASTET"
```

Udføres ovenstående program fås følgende skærmdialog, hvor brugersvar er understreget:

```
INDTAST BELØB 10.25
INDTAST BELØB 30
INDTAST BELØB 5.5
3 BELØB ER NU INDTASTET
```

Programudførelse afsluttet



5.4.18 Skærmkontrol

I dette afsnit beskrives 2 sætninger, som kan benyttes til henholdsvis at slette dataskærmens indhold og til at kontrollere markørens placering på skærmen.

5.4.18.1 CLEAR-sætning

CLEAR-sætningen benyttes til at slette dataskærmens aktuelle indhold.

Syntaks

CLEAR-sætningen har følgende opbygning:



Udførelse

Udførelse af CLEAR-sætningen bevirker, at dataskærmens indhold slettes, hvorefter markøren flyttes til øverste venstre hjørne af dataskærmen.

Eksempel

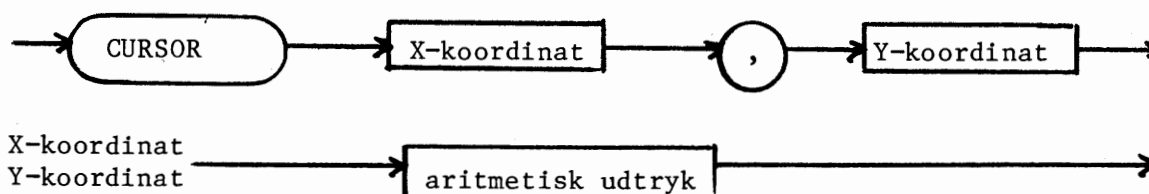
Se afsnit 5.4.18.2.

5.4.18.2 CURSOR-sætning

CURSOR-sætningen benyttes til at styre markørens placering på dataskærmen.

Syntaks

CURSOR-sætningen har følgende opbygning:



Udførelse

X-koordinat og Y-koordinat angiver koordinaterne for et punkt på dataskærmen. Ved udførelse af CURSOR-sætningen flyttes markøren til det angivne punkt.

X-aksen regnes positiv fra venstre mod højre, og dens koordinater må ligge i intervallet

$$1 \leq X \leq 64.$$

Y-aksen regnes positiv fra oven og nedefter, og dens koordinater må ligge i intervallet

$$1 \leq Y \leq 16.$$

Skærmens øverste venstre hjørne har således koordinaterne (1,1) og nederste højre hjørne har koordinaterne (64,16).

Kommentarer

1. Hvis beregningen af X- eller Y-koordinaten ikke giver et helt tal, foretages en afrunding.



Eksempel

```
0010 CLEAR  
0020 CURSOR 32, 8  
0030 PRINT "X"
```

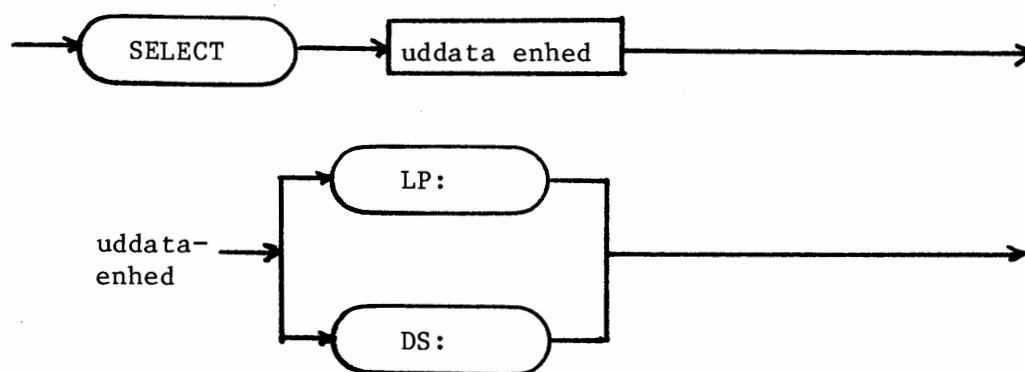
Udførelse af ovenstående program vil bevirke, at skærmens indhold slettes, hvorefter et X udskrives i position (32,8) på skærmen.

5.4.19 SELECT-sætning

SELECT-sætningen anvendes til at vælge enheden hvorpå uddata fra programmets PRINT-sætninger skal skrives.

Syntaks

SELECT-sætningen har følgende opbygning:



Udførelse

Udførelse af SELECT-sætningen bevirker, at uddata fra programmets PRINT-sætninger herefter skrives på den valgte uddataenhed. Uddataenheden benyttes under resten af programudførelsen, med mindre en anden uddataenhed vælges ved udførelse af en ny SELECT-sætning.

De 2 mulige uddataenheder er

LP:	Linieskriver
DS:	Dataskærm

Dataskærmen er standardenheden. Standardenheden benyttes som uddataenhed indtil en anden enhed vælges ved udførelse af en SELECT-sætning. Efter afslutningen af en programudførelse er standardenheden atter den gældende uddata-enhed.

Eksempel

```
0010 DIM TEKST$ OF 20
0020 PRINT "Demonstration af linieskriver"
0030 INPUT "TEKST " : TEKST$
0040 SELECT ,LP:
0050 PRINT TEKST$
0060 END
```

Udføres ovenstående program fås følgende skærmdialog, hvor brugersvar er understreget:

```
DEMONSTRATION AF LINIESKRIVER
TEKST Dette er en prøve
```

END i linie 60

Efter indtastning af teksten

Dette er en prøve

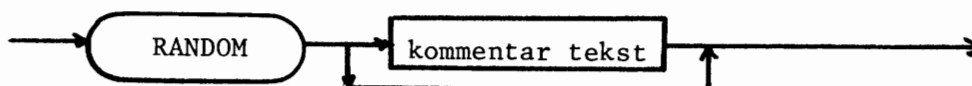
Udskrives denne på linieskriveren.

5.4.20 RANDOM-sætning

RANDOM-sætningen benyttes når det ønskes, at de tilfældige tal, genereret af RND-funktionen (se afsnit 5.5), skal starte et tilfældigt sted i sekvensen af (pseudo-) tilfældige tal.

Syntaks

RANDOM-sætningen har følgende opbygning:



Udførelse

Som omtalt ovenfor.

Kommentarer

1. Normalt vil RND-funktionen generere samme sekvens af tilfældige tal for hver gang en RUN-kommando udføres. Dette kan være nyttigt under programafprøvning.

Eksempel

```

0010 FOR I: = 1 TO 10
0020     PRINT RND(1,20);" ";
0030 NEXT I
  
```

Udføres ovenstående program et antal gange, vil det hver gang producere den samme sekvens af pseudo-tilfældige tal. Indsættes linien

```

0005 RANDOM
  
```

i programmet, vil gentagne kørsler ikke producere den samme sekvens af pseudo-tilfældige tal.



5.4.21 CHAIN-sætning

CHAIN-sætningen benyttes til at igangsætte indlæsning og udførelse af et program, som er gemt på binær form på baggrundslager.

Syntaks

CHAIN-sætningen har følgende opbygning:

CHAIN	filnavn
filnavn	strengudtryk

Udførelse

Udførelse af CHAIN-sætningen medfører, at den igangværende programudførelse standses, hvorefter det binære program 'filnavn' indlæses fra baggrundslager. Herefter startes udførelsen af det indlæste program i det laveste linienummer.

Syntaksen for 'filnavn' er beskrevet i kapitel 4.

Kommentarer

1. CHAIN-sætningen anvendes typisk til at opdele (segmentere) et stort program i et antal mindre programmer, eller til at udføre uafhængige programmer fra et styreprogram på basis af indtastede brugerkommandoer.
2. Som tidligere omtalt skal programmet 'filnavn' være binært, dvs. programmet skal være lagret ved brug af SAVE-kommandoen. Umiddelbart inden denne lagring, skal programmet have været udført.
3. Kommunikation mellem programmet, hvori CHAIN-sætningen forekommer, og programmet, som indlæses når CHAIN-sætningen udføres, kan kun ske via datafiler.

Eksempel

```
0005 // STYREPROGRAM
0010 DIM FAG$ OF 7
0020 REPEAT
0030     INPUT "FAG ": FAG$
0040 UNTIL FAG$= "ENGELSK" OR FAG$ = "TYSK"
0050 CASE FAG$ OF
0060 WHEN "ENGELSK"
0070     CHAIN "CSO:ENGELSK"
0080 WHEN "TYSK"
0090     CHAIN "CSO:TYSK"
0100 ENDCASE
0110 END
```

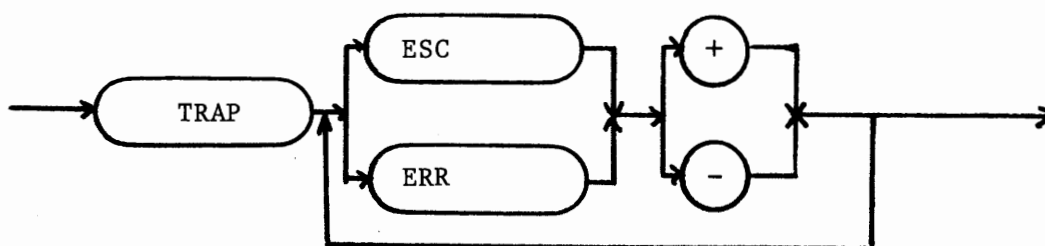
Ovenfor ses et styreprogram, der som inddata akcepterer tegnstrengen ENGELSK og TYSK. Afhængigt af den indtastede tegnstreng vælges et program, som læses ind fra enheden CSO:, for umiddelbart efter at blive udført.

5.4.22 TRAP-sætning

TRAP-sætningen benyttes til at skifte mellem programstyret reaktion- og standardreaktion på aktivering af ESC-tast samt på visse fejlsituationer.

Syntaks

TRAP-sætningen har følgende opbygning:



Udførelse

TRAP-sætningens funktion kan forklares på følgende måde:

Opfat TRAP_ESC og TRAP_ERR som navnene på to logiske systemvariable. Hver gang en programudførelse starter, tildeles TRAP_ESC og TRAP_ERR automatisk værdien sand. Har TRAP_ESC værdien sand, betyder det, at et tryk på tastaturets ESC-tast medfører en øjeblikkelig afbrydelse af programudførelsen. Har TRAP_ERR værdien sand, betyder det, at fejl i programudførelsen vil medføre, at programmet standser med en fejlmeddelelse.

Har TRAP_ESC værdien falsk, vil aktivering af tastaturets ESC_tast ikke medføre en programafbrydelse, men vil blot medføre, at den logiske systemvariabel ESC bliver tildelt værdien sand. Har TRAP_ERR værdien falsk, vil visse fejlsituationer ikke resultere i programafbrydelse, men blot medføre, at den logiske systemvariabel ERR bliver tildelt værdien sand.

Når TRAP_ESC og TRAP_ERR har værdien falsk, er det således muligt fra brugerprogrammet at konstatere, om ESC-tasten har været aktiveret, og om fejlsituationer har forekommet, ved at aflæse værdien af ESC og ERR.

ESC og ERR omtales nærmere i afsnit 5.5.3.



Efterfølges 'ESC' eller 'ERR' af et '+' i TRAP-sætningen, tildeles den tilsvarende af systemvariablene TRAP_ESC og TRAP_ERR værdien sand. Efterfølges ESC eller ERR af et '-' i TRAP-sætningen, tildeles den tilsvarende af systemvariablene TRAP_ESC og TRAP_ERR værdien falsk.

Kommentarer

1. De fejlsituationer, som ikke vil medføre programafbrydelse, når TRAP_ERR er falsk, er situationer, som COMAL-80 fortolkeren kan omgå på en "naturlig" og veldefineret måde. Sådanne fejl kaldes i det følgende ikke-fatale fejl.

Divisionen med 0 er et eksempel på en ikke-fatal fejl. Resultatet af en sådan division er defineret ved

$$X/0 = \text{SGN}(X) * \text{MAX_REEL}$$

hvor MAX_REEL er den største værdi, som et reelt udtryk kan antage.

En indeksfejl er et eksempel på en fatal fejl, idet den ikke kan omgås på en "naturlig" og veldefineret måde.

I oversigten over fejlmeddelelser, i appendix A1 i COMET BRUGERVEJLEDNING, er de fatale fejl markeret med et 'f'.

Eksempel

```
0010 TRAP ESC-
0020 REPEAT
0030   FOR I:= 1 TO 10
0040     PRINT I; " ";
0050   NEXT I
0060   PRINT
0070   IF ESC THEN STOP
0080 UNTIL FALSE
```

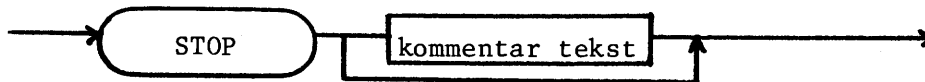
Ovenstående program udskriver en række linier, hver indeholdende tallene fra 1 til 10. Programmet kører i løkke og kan kun standses ved tryk på ESC-tasten. P.g.a. TRAP-sætningen i linie 10 er det imidlertid umuligt at standse programmet midt i udskrivningen af en linie. Programudførelsen kan kun standses i linie 70.

5.4.23 STOP-sætning

STOP-sætningen anvendes til at stoppe udførelsen af et program.

Syntaks

STOP-sætningen har følgende opbygning:



Udførelse

Udførelse af en STOP-sætning medfører, at programudførelsen standser med følgende udskrift på skærmen:

STOP i linie XXXX

hvor XXXX erstattes med STOP-sætningens linienummer.

Eksempel

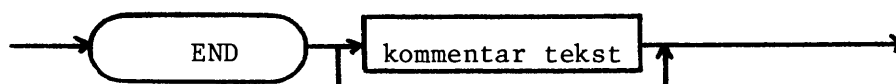
Se afsnit 5.4.24.

5.4.24 END-sætning

END-sætningen anvendes til at stoppe udførelsen af et program.

Syntaks

END-sætningen har følgende opbygning:



Udførelse

Udførelse af en END-sætning medfører, at programudførelsen standser med følgende udskrift på skærmen:

END i linie XXXX

hvor XXXX erstattes med END-sætningens linienummer.

Kommentarer

1. END-sætningen benyttes oftest til at angive den fysiske afslutning på et program, hvor den altså placeres i programmets sidste linie. Det er imidlertid ikke nødvendigt at angive programmets fysiske afslutning på denne måde, idet COMAL-80 fortolkeren selv placerer et for brugeren synligt END efter den sidste linie i et program. Når dette END mødes under programudførelsen, udskrives meddelelsen:

Programudførelse afsluttet

Eksempel

```

0010 FOR I:= 1 TO 2
0020   INPUT "SKRIV ET TAL": TAL
0030   IF TAL = 0 THEN STOP
0040 NEXT I
0050 END programmets fysiske afslutning
  
```

Herunder vises skærmdialogen for to forskellige udførelser af ovenstående program. Brugersvar er understreget.



1. Udførelse:

SKRIV ET TAL 0

STOP i linie 0030

2. Udførelse:

SKRIV ET TAL 1

SKRIV ET TAL 2

END i linie 0050



5.4.25 Specielle sætninger

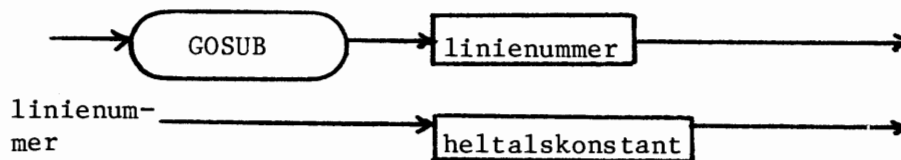
I dette afsnit beskrives nogle sætninger, som ikke falder naturligt ind i sprogstrukturen i COMAL-80. Sætningerne er "BASIC-levn", som kun er medtaget i COMAL-80 for ikke at umuliggøre kørsel med BASIC-programmer.

5.4.25.1 GOSUB-sætning

GOSUB-sætningen er et subrutinekald.

Syntaks

GOSUB-sætningen har følgende opbygning:



Udførelse

Udførelse af GOSUB-sætningen bevirker, at proceduren, som starter i linie 'linienummer', kaldes.

Programudførelsen starter i sætningen med det angivne linienummer og fortsætter herfra indtil en RETURN-sætning mødes. Herefter fortsætter programudførelsen i sætningen umiddelbart efter GOSUB-sætningen.

Kommentarer

1. Linienummeret, som angives i GOSUB-sætningen, bør ikke indgå i sætningslisten for nogen sammensat sætning, med mindre både den kaldende sætning samt hele subrutinen indgår i samme sætningsliste.

Eksempel

```

0010 X := 3; Y:= 5
0020 GOSUB 60
0030 X := 8; Y:= 11
0040 GOSUB 60
0050 STOP
0060 // KVADRATSUM
0070 PRINT "X↑2 + Y↑2 ="; X↑2 + Y↑2
0080 RETURN
    
```



Udføres ovenstående program fås følgende udskrift:

$$X^2 + Y^2 = 34$$

$$X^2 + Y^2 = 185$$

STOP i linie 0050

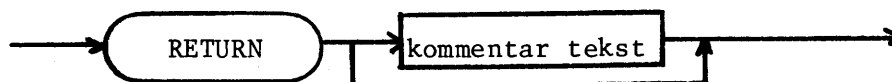
Sammenlign med eksemplet i afsnit 5.4.15.

5.4.25.2 RETURN-sætning

RETURN-sætningen benyttes til at angive, at returhop fra en subroutine skal foretages.

Syntaks

RETURN-sætningen har følgende opbygning:



Udførelse

Udførelse af en RETURN-sætning bevirker, at retur-hop fra den senest kaldte subroutine foretages.

Mødes en RETURN-sætning, når ingen subroutine er kaldt, standser programudførelsen med en fejlmeddelelse.

Eksempel

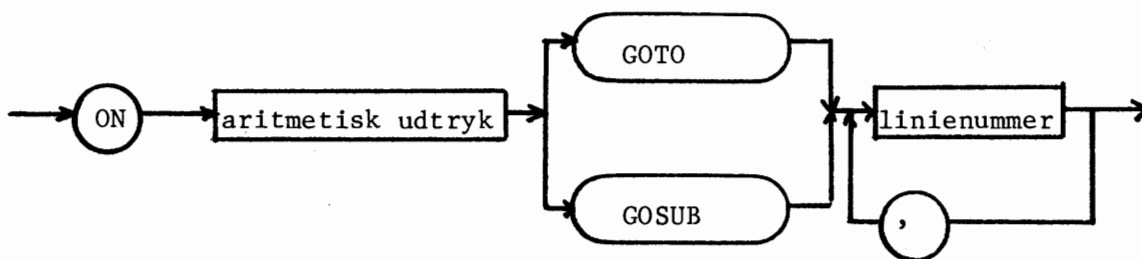
Se afsnit 5.4.25.1. Bemærk at udeladelse af STOP-sætningen i linie 50 ville resultere i en fejlmeddelelse i linie 80.

5.4.25.3 ON-GOTO/ON-GOSUB-sætninger

ON-GOTO- og ON-GOSUB-sætningerne benyttes til at vælge et af en række linienumre, hvori programudførelsen ønskes fortsat. Valget af linienummer foretages ud fra værdien af et aritmetisk udtryk.

Syntaks

Sætningerne har følgende opbygning:



Udførelse

Det aritmetiske udtryk beregnes. Lad os kalde værdien af udtrykket for n . Programudførelsen fortsætter herefter i linien, hvis linienummer forekommer som det n 'te i listen af linienumre.

Ved udførelse af en ON-GOSUB-sætning foretages returhop til sætningen umiddelbart efter denne, når et RETURN mødes.

Kommentarer

1. Hvis beregningen af det aritmetiske udtryk ikke giver et helt tal, foretages en afrunding.
2. Hvis n ikke opfylder kravet

$$1 \leq n \leq \text{antal linienumre i listen,}$$

ignoreres sætningen.



Eksempel

```
0010 TAL := 2
0020 ON TAL GOTO 30, 50, 70
0030 PRINT "TAL = 1"
0040 GOTO 80
0050 PRINT "TAL = 2"
0060 GOTO 80
0070 PRINT "TAL = 3"
0080 END
```

Udføres programmet fås følgende udskrift:

```
TAL = 2
```

END i linie 0080

Ændres linie 10 til

```
0010 TAL := -3
```

vil udførelsen af programmet give følgende udskrift:

```
TAL = 1
```

END i linie 0080

5.5.1 Aritmetiske standardfunktioner

Aritmetiske standardfunktioner kan indgå i aritmetiske udtryk.

Parametrene til de aritmetiske standard-funktioner må, hvis intet andet udtrykkeligt er nævnt, være udtryk af den angivne type. Til angivelse af parametrenes type er følgende navneregler anvendt:

r : reel
 h : heltallig
 x : reel eller heltallig
 s : tegnstreng

Typesøjlen angiver resultatets type, og her er følgende forkortelser anvendt:

H : heltallig
 R : reel

De aritmetiske standardfunktioner er følgende:

<u>Navn</u>	<u>Type</u>	<u>Resultat</u>
SIN(x)	R	sinus til x. x i radianer.
COS(x)	R	cosinus til x. x i radianer.
TAN(x)	R	tangens til x. x i radianer.
ATN(x)	R	arcus tangens til x, udtrykt i radianer.
LOG(x)	R	den naturlige logaritme af x.
EXP(x)	R	eksponentialfunktionen taget af x.
SQR(x)	R	kvadratroden af x. $x \geq 0$
ABS(x)	som x	den absolutte værdi af x.
SGN(x)	H	0 hvis $x = 0$, 1 hvis $x > 0$, -1 hvis $x < 0$
INT(r)	R	heltalsdelen af r, d.v.s. tallet som fremkommer ved bortkastning af decimalerne
FRAC(r)	R	$r - \text{INT}(r)$
RND	R	et (pseudo-) tilfældigt tal i intervallet $[0,1[$
RND(x,y)	H	et (pseudo-) tilfældigt tal i intervallet $[x,y]$. Er x og y ikke heltallige, foretages en afrunding.
POS(s1,s2)	H	Lad os kalde resultatet n. Hvis s1 forekommer i s2 er n lig med startpositionen for den første forekomst (regnet fra venstre mod højre) af s1 i s2. Hvis s1 er lig med den tomme streng er $n=1$. Hvis s1 ikke forekommer i s2 er $n=0$.



Eks. POS ("BC", "ABCD") = 2
POS ("", "BBB") = 1
POS ("BC", "BBB") = 0

LEN(S) H Aktuell længde af s. s skal være en strengvariabel uden selektor-angivelse.

ORD(S) H Den decimale ASCII-kode for første tegn i s.

IVAL(S) H Det hele tal, der forefindes som en tegnstreng i s.
Eks. IVAL ("357") = 357

RVAL(S) R Det reele tal, der forefindes som en tegnstreng i s.
Eks. RVAL ("35.7") = 35.7



5.5.2 Tegnorienterede standardfunktioner

Tegnorienterede standardfunktioner kan indgå i strengudtryk. Værdien af en sådan standardfunktion er en tegnstreng.

De tegnorienterede standardfunktioner er følgende:

<u>Navn</u>	<u>Resultat</u>
CHR\$(x)	tegnet som har den decimale ASCII-kode x. Er x ikke heltallig, foretages afrunding. x skal (efter en evt. afrunding) ligge i intervallet [0,255]
STR\$(x)	tallet x, som en tegnstreng. Eks. A\$:= STR\$(12.2*3) Efter ovenstående tildeling indeholder A\$ tegnstrengen "36.6".
SPC\$(x)	en tegnstreng bestående af x mellemrum (blanktegn). Er x ikke heltallig, foretages afrunding.

Parameteren til de ovenfor beskrevne standardfunktioner er et aritmetisk udtryk.



5.5.3 Systemvariable

Systemvariablene er variable, som under programudførelsen tildeles værdi af COMAL-80 systemet. Systemvariablene skal opfattes som logiske variable, der kan antage værdierne sand og falsk. Systemvariablenes type er heltallig.



5.5.3.1 EOD

Systemvariablen EOD anvendes i forbindelse med læsning af data fra DATA-sætninger.

EOD har værdien falsk, så længe "pegepinden" udpeger en værdi i datalisten (se afsnit 5.4.8). Når den sidste værdi er læst, tildeles EOD værdien sand. EOD er sand, hvis datalisten er tom.

Er datalisten ikke tom, kan EOD's værdi ændres fra sand til falsk ved udførelse af en RESTORE-sætning (se afsnit 5.4.9).

Eksempel:

```
0010 DIM STRENG$ OF 10, TEKST$ OF 10
0020 FOR I:= 1 TO 10
0030     WHILE NOT EOD
0040         READ STRENG$
0050         TEKST$:+ STRENG$
0060     ENDWHILE
0070     PRINT TEKST$
0080     RESTORE
0090     TEKST$:= ""
0100 NEXT I
0110 DATA "COMAL", "-", "80"
0120 END
```

Udføres ovenstående program, vil 10 linier med teksten

COMAL-80

blive udskrevet under hinanden.



5.5.3.2 ESC

Systemvariablen ESC anvendes til at undersøge, om tastaturets ESC-tast har været aktiveret under programudførelsen.

Ved programudførelsens start tildeles ESC værdien falsk. Når ESC-tasten nedtrykkes, tildeles ESC værdien sand. Herefter vil ESC være sand, indtil der spørges på dens værdi (eller den på anden måde indgår i et udtryk, som beregnes). Umiddelbart efter, at der er spurgt på værdien af ESC, tildeles den værdien falsk.

Forudsætningen for, at aktivering af ESC-tasten kan aflæses ved hjælp af systemvariablen ESC, er, at der forinden er udført en TRAP-sætning af formen

TRAP ESC-

For en detaljeret beskrivelse af TRAP-sætningen henvises til afsnit 5.4.22, hvor der også gives et eksempel på anvendelse af systemvariablen ESC.



5.5.3.3 ERR

Systemvariablen ERR benyttes til at undersøge, om visse fejlsituationer er forekommet under programudførelsen.

Ved programudførelsens start tildeles ERR værdien falsk. Når en fejlsituation opstår, tildeles ERR værdien sand, hvorefter programudførelsen fortsætter. Herefter vil ERR være sand, indtil der spørges på dens værdi.(eller den på anden måde indgår i et udtryk, som beregnes). Umiddelbart efter at der er spurgt på værdien af ERR, tildeles den værdien falsk.

Forudsætningen for, at fejlsituationer kan behandles som ovenfor omtalt, er, at der inden en sådan situation opstår, er udført en TRAP-sætning af formen

TRAP ERR-

For en detaljeret beskrivelse af TRAP-sætningen henvises til afsnit 5.4.22, hvori det også er beskrevet hvilke fejlsituationer, der kan behandles som ovenfor beskrevet.



5.5.3.4 EOF

Se afsnit 5.6.9.



5.6 Filsystem

COMAL-80 giver mulighed for, at overføre data til og fra sekundære lagre som kassettebånd og disketter. En sammenhørende mængde af data, som er lagret på et sekundært lager, vil i det følgende blive kaldt en fil.

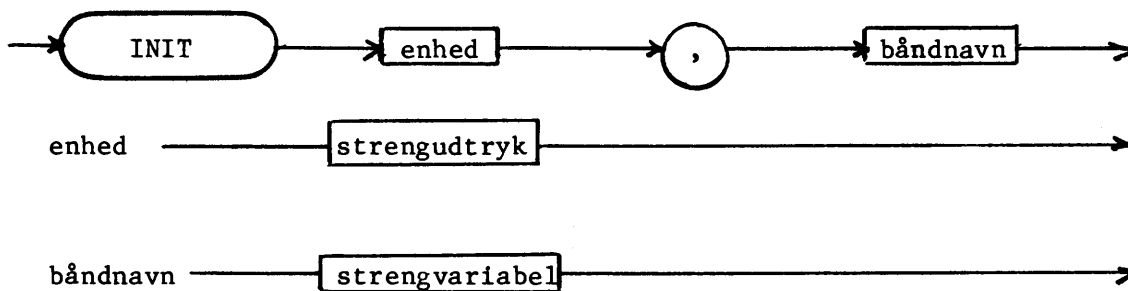
I dette kapitel beskrives de COMAL-80-sætninger, som har tilknytning til fil-systemet.

5.6.1 INIT-sætning

INIT-sætningen benyttes til at initialisere et bånd, som er placeret i (en af) datamatsens kassettebåndoptager(e). Når et kassettebånd initialiseres, indlæses dets katalog til et bestemt sted i datamatsens hovedlager. Først når et kassettebånd er initialiseret, kan man overføre data til/fra det.

Syntaks

INIT-sætningen har følgende opbygning:



Udførelse

Udførelse af INIT-sætningen medfører, at båndet i kassettebåndoptageren med enhedsbetegnelsen 'enhed' initialiseres. Båndsidens identifikation, dvs. navnet som båndsidens er blevet tildelt v.h.a. FORMAT-kommandoen, returneres i strengvariablen 'båndnavn'.

Kommentarer

1. Da et båndnavn må bestå af op til 50 tegn, bør strengvariablen 'båndnavn', hvis man ikke vil risikere at miste noget af navnet, erklæres med en maksimal længde på 50 tegn. Kan båndnavnet ikke rummes i strengvariablen, eller er båndnavnets længde mindre end strengvariablens maksimale længde, følges samme regler som for tildelinger til strengvariable (se afsnit 5.4.4.).
2. Med hensyn til enhedsbetegnelser henvises til kapitel 4.

Eksempel

Se afsnit 5.6.2.

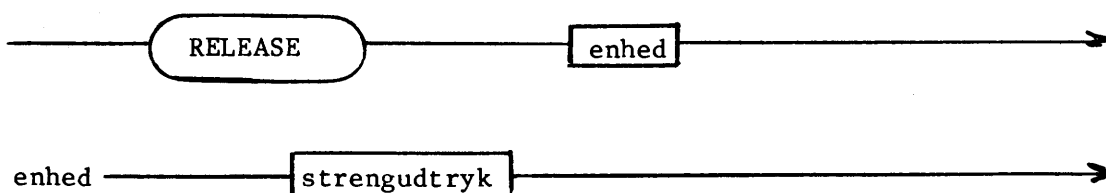


5.6.2 RELEASE-sætning

RELEASE-sætningen benyttes til at frigøre et kassettebånd, som er initialiseret. Når et bånd frigøres, ajourføres dets katalog. Først når båndet er frigjort, kan et andet bånd initialiseres på den pågældende kassettebåndoptager.

Syntaks

RELEASE-sætningen har følgende opbygning:



Udførelse

Båndet, som er initialiseret i kassettebåndoptageren med enhedsbetegnelsen 'enhed', frigøres.

Kommentarer

1. Med hensyn til enhedsbetegnelser henvises til kapitel 4.

Eksempel

```
0010 DIM SVAR$ OF 1, BÅNDNAVN$ OF 50
0020 REPEAT
0030     PRINT "Isæt båndet ENGELSK i båndoptageren."
0040     INPUT "Tryk på RETURN-tasten når det er gjort.":SVAR$
0050     INIT "CSO:", BÅNDNAVN$
0060     IF BÅNDNAVN$ <> "ENGELSK" THEN RELEASE "CSO:"
0070 UNTIL BÅNDNAVN$ = "ENGELSK"
0080 PRINT
0090 PRINT "om et øjeblik starter engelsk-kurset."
0100 CHAIN "CSO:ENGELSK1"
```



Udføres ovenstående program udskrives teksten:

Isæt båndet ENGELSK i båndoptageren.
Tryk på RETURN-tasten når det er gjort.

Herefter skal brugeren isætte det angivne bånd og trykke på RETURN-tasten.
Nu initialiseres det isatte bånd (linie 50). Har båndet ikke navnet
ENGELSK, frigøres det, hvorefter brugeren atter bedes om at isætte båndet
ENGELSK. Har båndet navnet ENGELSK, udskrives teksten:

Om et øjeblik starter engelsk-kurset.

Hvorefter programmet ENGELSK1 læses ind og udføres v.h.a. CHAIN-sætningen
i linie 100.

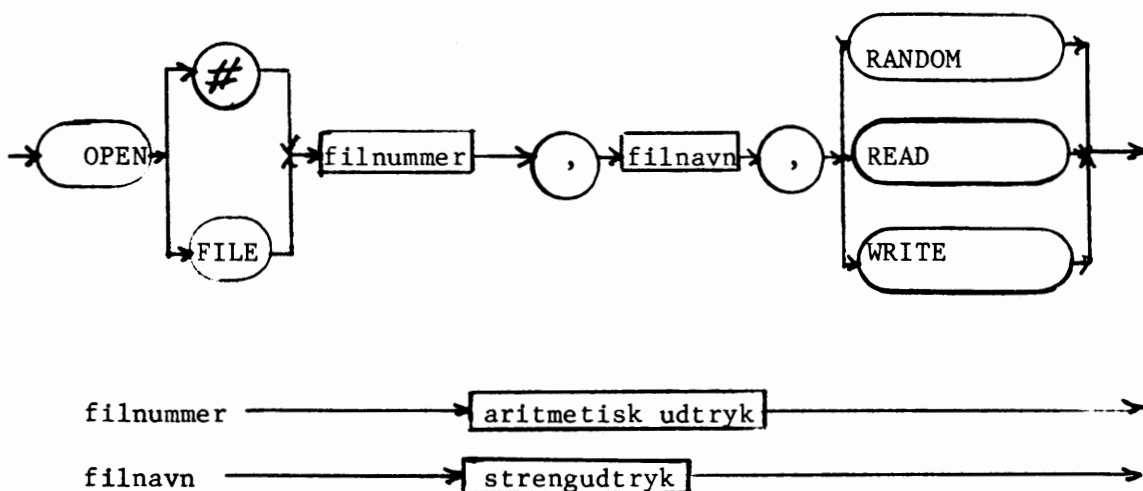


5.6.3 OPEN-sætning

Før en fil kan benyttes af et program, skal den åbnes. Ved åbningen informeres filsystemet om, at filen ønskes benyttet til at læse data fra og/eller skrive data på.

Syntaks

OPEN-sætningen har følgende opbygning:



Udførelse

I det følgende vil kun OPEN-sætninger, som afsluttes af et af nøgleordene 'READ' eller 'WRITE' blive beskrevet. Nøgleordet 'RANDOM' er knyttet til anvendelse af disketter, og vil blive beskrevet senere i en opdatering.

Ved udførelse af OPEN-sætningen åbnes filen 'filnavn' til læsning (READ) eller skrivning (WRITE). Filen får tildelt nummeret 'filnummer', som benyttes til at identificere filen i lukke-, læse- og skrivesætninger. Er 'filnummer' ikke heltallig, foretages en afrunding.

Åbnes filen til læsning, skal den i forvejen findes i båndets katalog. Åbnes filen til skrivning, skabes en ny fil bag den hidtil sidste fil på båndet, uanset om båndet i forvejen indeholder en fil med det angivne navn.



'#' og 'FILE' kan benyttes i flæng efter behag. I programudskrifter konverteres '#' til 'FILE'.

Kommentarer

1. De tilladte filnumre er 0, 1, 2 og 3.
2. På en enhed kan kun én fil være åben ad gangen.
3. Hvis filnavnet ikke indeholder nogen udvidelse (ext), vil udvidelsen DAT være underforstået. Der henvises til kapitel 4 for en beskrivelse af syntaksen for filnavne.
4. Når en fil er blevet åbnet, vil læsehovedet være ud for filens start.

Eksempel

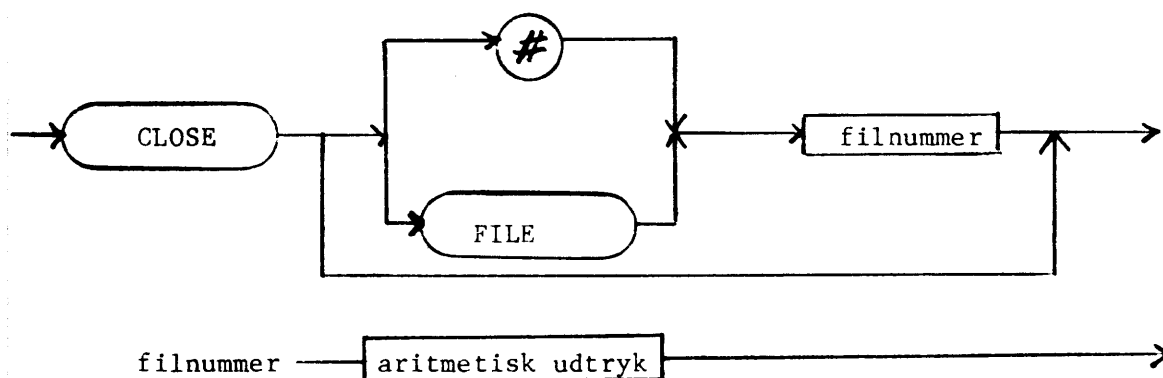
Se afsnit 5.6.5.

5.6.4 CLOSE-sætning

Når et program er færdig med at benytte en fil, skal den lukkes. Først når filen er lukket, kan en anden fil åbnes på samme enhed.

Syntaks

CLOSE-sætningen har følgende opbygning:



Udførelse

Består CLOSE-sætningen kun af nøgleordet 'CLOSE', bevirker udførelsen, at alle åbne filer lukkes. Ellers medfører udførelsen af en CLOSE-sætning, at filen, som ved åbningen er blevet tildelt nummeret 'filnummer', lukkes. Er filnummeret ikke heltalligt, foretages en afrunding.

'#' og 'FILE' kan benyttes i flæng efter behag. I programudskrifter konverteres '#' til 'FILE'.

Eksempel

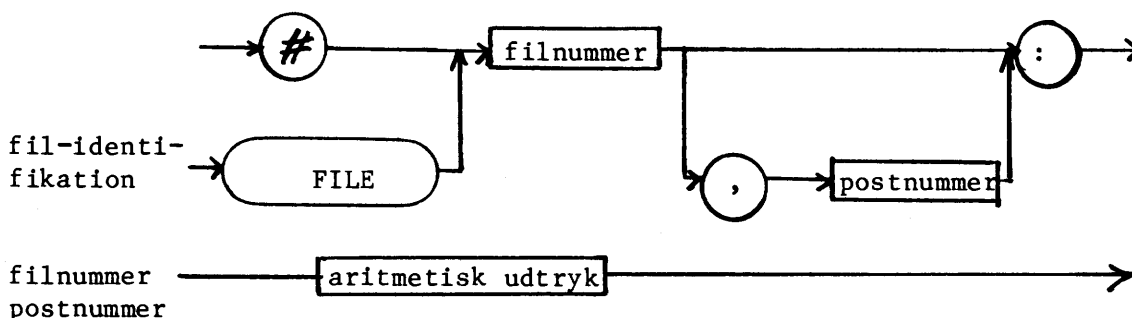
Se afsnit 5.6.5.

5.6.5 PRINT-file-sætning

PRINT-file-sætningen benyttes til at udskrive data på ASCII-form på en fil.

Syntaks

PRINT-file-sætningens opbygning er identisk med PRINT-sætningens, bortset fra at 'PRINT' skal efterfølges af en fil-identifikation med følgende opbygning:



Syntaksen for PRINT-sætningen kan ses i afsnit 5.4.5.

Udførelse

Angivelse af 'postnummer' har tilknytning til anvendelse af filer med direkte (random) tilgang, hvilket igen er knyttet til disketter. Anvendelse af filer med direkte tilgang vil blive beskrevet senere i en opdatering.

Udførelse af en PRINT-file-sætning er helt analog med udførelse af en PRINT-sætning, bortset fra at data udskrives på en fil i stedet for på skærm eller linieskriver. Data udskrives på filen, som ved åbningen er blevet tildelt nummeret 'filnummer'. Er 'filnummer' ikke heltallig foretages en afrunding.

'#' og 'FILE' kan benyttes i flæng efter behag. I programudskrifter konverteres '#' til 'FILE'.

Der henvises til afsnit 5.4.5, hvor PRINT-sætningen er beskrevet.

Kommentarer

1. Filen skal være åbnet til skrivning (WRITE).

Eksempel

```
0010 OPEN FILE 1, "CSO:DATAFIL", WRITE
0020 FOR I:= 1 TO 10
0030   FOR J:= 1 TO 5
0040     PRINT FILE 1: J;
0050   NEXT J
0060   PRINT FILE 1:
0070 NEXT I
0080 CLOSE FILE 1
```

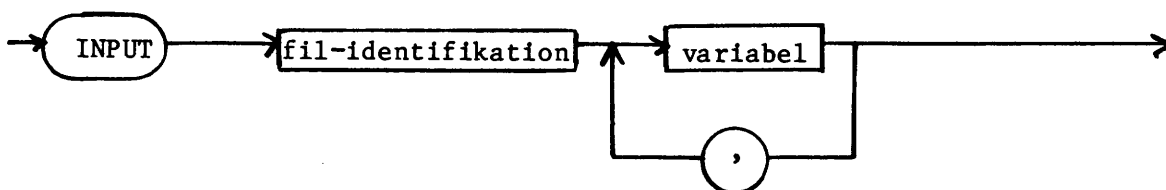
Udførelse af ovenstående program bevirker, at en fil med navnet DATAFIL dannes og åbnes til skrivning. Herefter udskrives 10 "linier", hver indeholdende cifrene 1, 2, 3, 4 og 5 skrevet umiddelbart efter hinanden.

5.6.6 INPUT-file-sætning

INPUT-file-sætningen benyttes til at indlæse data på ASCII-form fra en fil.

Syntaks

INPUT-file-sætningen har følgende opbygning:



fil-identifikation: se afsnit 5.6.5.

Udførelse

Med hensyn til betydningen af 'postnummer' i 'fil-identifikation' henvises til afsnit 5.6.5.

Udførelse af en INPUT-file-sætning er helt analog med udførelse af en INPUT-sætning, bortset fra at data læses fra en fil og ikke fra tastaturet. Data læses fra filen, som ved åbningen er blevet tildelt nummeret 'filnummer'. Er 'filnummer' ikke heltallig, foretages en afrunding.

Læsning af en fil foregår sekventielt "linie" for "linie" startende med filens første "linie". Når en INPUT-file-sætning mødes under programudførelsen, gennemløbes listen af variable i sætningen fra venstre mod højre, og de tilsvarende værdier fra filens aktuelle "linie" tildeles.

'#' og 'FILE' kan benyttes i flæng i fil-identifikationen. I programudskrifter konverteres '#' til 'FILE'.

Kommentarer

1. Filen skal være åbnet til læsning (READ).

Se kommentarerne til INPUT-sætningen i afsnit 5.4.6.

Eksempel

```
0010 OPEN FILE 1, "CSO:DATAFIL", READ
0020 FOR I:= 1 TO 10
0030     INPUT FILE 1: TAL
0040     PRINT TAL
0050 NEXT I
0060 CLOSE FILE 1
```

Udføres ovenstående program, efter at programmet i afsnit 5.6.5 er blevet udført, fås en udskrift bestående af 10 linier, hver indeholdende tallet 12345.



5.6.7 WRITE-sætning

WRITE -sætningen benyttes til at udskrive data på binær form på en fil.

Beskrivelse følger senere i en opdatering.



5.6.8 READ-sætning

READ-sætningen benyttes til at indlæse data på binær form fra en fil.

Beskrivelse følger senere i en opdatering.



5.6.9 Systemvariablen EOF

EOF kan opfattes som en logisk systemvariabel, der er organiseret som en vektor med 4 elementer EOF(0),...,EOF(3). EOF benyttes til at indikere, at sidste post i en fil er læst. Typen af EOF er heltallig.

Når en sekventiel fil åbnes for læsning, tildeles EOF(n), hvor n er filnummeret, værdien falsk. Når den sidste post på filen er læst, tildeles EOF(n) værdien sand.

Filnummeret n kan angives som et aritmetisk udtryk. Er udtrykkets værdi ikke heltallig, foretages en afrunding.

Eksempel

```
0010 OPEN FILE 1, "CSO:DATAFIL", READ
0020 WHILE NOT EOF(1)
0030     INPUT FILE 1: TAL
0040     PRINT TAL
0050 ENDWHILE
0060 CLOSE FILE 1
```

Udførelsen af ovenstående program giver samme resultat som udførelse af programmet i eksemplet i afsnit 5.6.6, under de beskrevne forhold.



Afsnit 5.7

Systemkommandoer

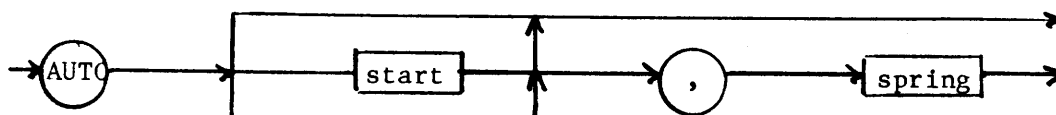
- 5.7.1 AUTO
- 5.7.2 CAT
- 5.7.3 CLOSE
- 5.7.4 CON
- 5.7.5 DEL
- 5.7.6 DELETE
- 5.7.7 ENTER
- 5.7.8 FORMAT
- 5.7.9 GETUNIT
- 5.7.10 INIT
- 5.7.13 NEW
- 5.7.18 RUN
- 5.7.21 SIZE
- 5.7.23 UNIT

5.7.1 AUTO

AUTO forsyner automatisk nye programlinier med linienummer.

Syntaks

AUTO-kommandoen har følgende opbygning:



Udførelse

Ved hver vognretur udregnes det nye linienummer som summen af det foregående linienummer og det angivne spring. Det fundne linienummer anbringes i I/O-bufferen og vises på skærmen som et fire-cifret tal. Markøren anbringes i position 6, klar til indtastning.

Eksempler

AUTO 15, 5
 AUTO 15
 AUTO ,5
 AUTO

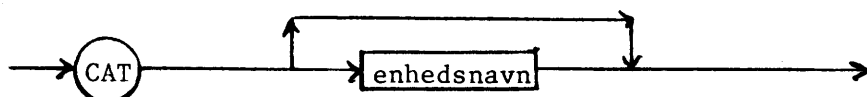
Kommentarer

1. Udelades værdien for "start" benyttes værdien 10.
2. Udelades værdien for "spring" benyttes værdien 10.
3. AUTO-mode kan til enhver tid standses ved tryk på "ESC"-tasten.

5.7.2 CAT

CAT udskriver katalogblokken for et initialiseret bånd.

Syntaks



Udførelse

Kommandoen henter fra arbejdslageret oplysninger om, hvilke filer der findes på det angivne bånd og viser dette på skærmen.

Eksempler

CAT
CAT CSO:

Kommentarer

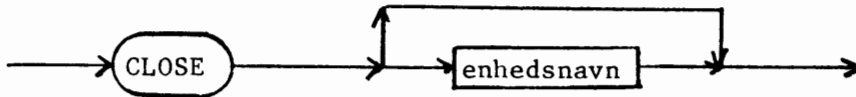
1. Udelades enhedsnavn benyttes det navn som sidst er angivet i en "UNIT"-kommando.

Indtil første "UNIT"-kommando benyttes som enhedsnavn "CSO:"

5.7.3 CLOSE

CLOSE lukker en datafil, som efter et programs afvikling stadig er åben.

Syntaks



Udførelse

For en datafil oprettet til skrivning udskriver kommandoen sidste blok efterfulgt af en END-OF-TAPE-blok. For en datafil oprettet til læsning angiver kommandoen, at filen ikke længere skal bruges.

Eksempler

CLOSE
CLOSE CSO:

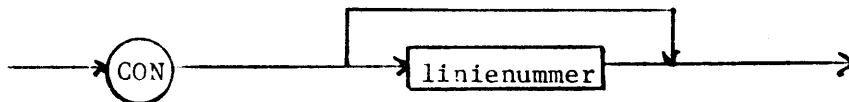
Kommentarer

1. Udelades enhedsnavn benyttes det navn som sidst er angivet i en "UNIT"-kommando.
Indtil første "UNIT"-kommando benyttes som enhedsnavn "CSO:".

5.7.4 CON

CON fortsætter programudførelsen efter stop forårsaget af en ikke fatal fejl eller tryk på ESC-tasten.

Syntaks



Udførelse

CON fortsætter programafviklingen uden at slette arbejdslagerets variabelområde.

Eksempler

CON
CON 100

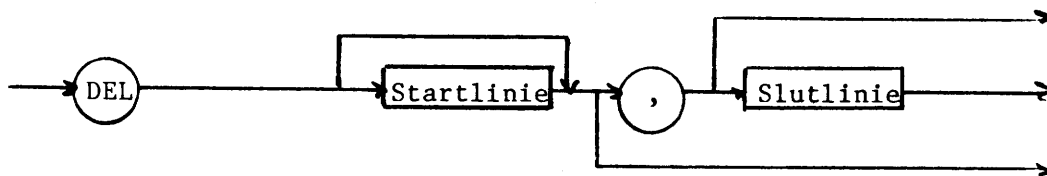
Kommentarer

1. Med LET-ordren kan variabelværdier ændres, inden programafviklingen genoptages.
2. Programafviklingen kan genoptages i en anden linie end den, der forårsagede stoppet.

5.7.5 DEL

DEL sletter dele af programmet.

Syntaks



Udførelse

De angivne linier fjernes fra arbejdslageret, og de efterfølgende dele af programmet rykkes op i hullet.

Eksempler

DEL 100
 DEL 100,160
 DEL 100,
 DEL ,100

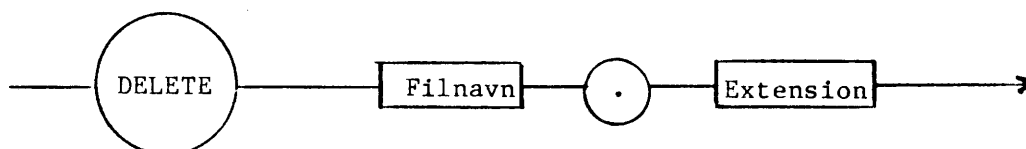
Kommentarer

1. Hvis kun "startlinie" angives, slettes kun linien med det angivne linienummer.
2. Hvis kun "startlinie" efterfulgt af komma angives, slettes denne linie og resten af programmet.
3. Hvis kun komma efterfulgt af "slutnummer" angives, slettes begyndelsen af programmet til og med den angivne linie.

5.7.6 DELETE

DELETE sletter filer på båndet.

Syntaks



Udførelse

Den angivne fil slettes på båndet og i katalogblokken i arbejdslageret.

Eksempler

DELETE FIL.DAT
DELETE TEST.CML

Kommentarer

1. Filen, som ønskes slettet, skal stå sidst på båndet.
2. Kommandoen skal - før datamaten slukkes - efterfølges af "RELEASE"-ordren for at få opdateret båndets katalogblok.

5.7.7 ENTER

Kommandoen henter en fil fra båndet og lægger den på plads i arbejdslageret.

Syntaks



Udførelse

Filen hentes fra den angivne båndstation som en streng af ASCII-karakterer. Efter hver vognretur syntakscontrolleres linien og lægges på plads i arbejdslageret.

Eksempler

```
ENTER LILLESKOLE  
ENTER CSO:PROGRAM  
ENTER TEST(2)
```

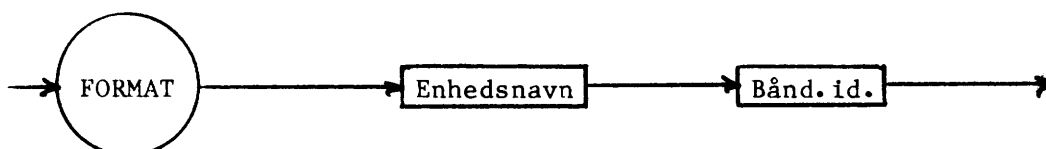
Kommentarer

1. ENTER-kommandoen kan kun læse filer, som er nedskrevet med LIST-kommandoen.

5.7.8 FORMAT

FORMAT klargør et nyt bånd til at arbejde under operativsystemet. Kommandoen kan også bruges til at rense et gammelt bånd for alle filer.

Syntaks



Udførelse

Det indsatte bånd spoles tilbage, og en katalogblok nedskrives.

Eksempler

FORMAT CSO:PROGRAMTAPE

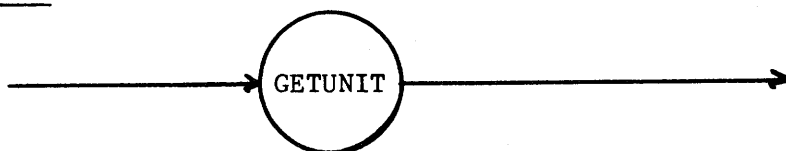
Kommentarer

1. Båndidentiteten kan være indtil 50 karakterer.
2. Hvis versionsnummer udelades, hentes den sidst nedskrevne version.
3. Udelades enhedsnavn benyttes det navn, som sidst er angivet i en "UNIT"-kommando. Indtil første "UNIT"-kommando benyttes som enhedsnavn "CSO:".
4. Hver oplæst linie syntakscontrolleres før den lægges på plads i arbejdslageret. Hvis der viser sig fejl, standses operationen, og den fejlbehæftede linie vises på skærmen. Fejlen kan nu rettes med systemets normale editeringsfaciliteter, og når der tasteres "RETURN", fortsætter operationen.

5.7.9 GETUNIT

Kommandoen viser hvilken båndstation som sidst er angivet i en "UNIT"-kommando.

Syntaks



Udførelse

Enhedsnavnet for den sidst angivne enhed vises.

Eksempel

GETUNIT

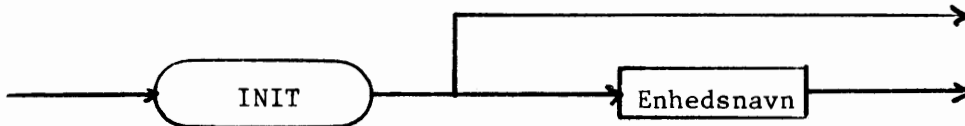
Kommentarer

1. Indtil første "UNIT"-kommando benyttes som enhedsnavn "CSO:".

5.7.10 INIT

Kommandoen klargør et tidligere formatteret bånd.

Syntaks



Udførelse

Båndet i den angivne båndstation tilbagespoles, og katalogblokken, som ligger først på båndet, indlæses i arbejdslageret. Båndets navn vises derefter på skærmen.

Eksempler

INIT
INIT CSO:

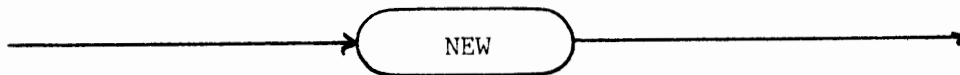
Kommentarer

1. Udelades enhedsnavn, benyttes det navn, som sidst er angivet i en "UNIT"-kommando.
Indtil første "UNIT"-kommando benyttes som enhedsnavn "CSO:".

5.7.13 NEW

Kommandoen fjerner gamle programmer fra arbejdslageret og klargør dette til indlæsning af nye.

Syntaks



Udførelse

Arbejdslageret, variabelområder og interne pile nulstilles.

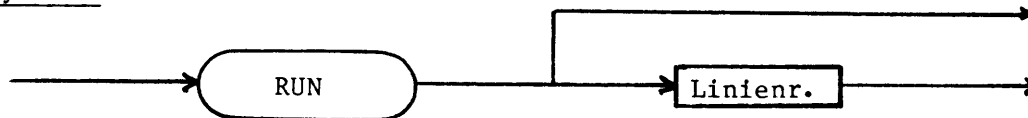
Eksempel

NEW

5.7.18 RUN

Kommandoen starter programmet i arbejdslageret.

Syntaks



Udførelse

COMAL-80 bringes i en veldefineret udgangstilstand, som blandt andet lukker åbne filer, og vælger skærmen som udskriftenhed. Derefter undersøges, i et særligt prepass, om programmet indeholder strukturer (FOR-NEXT, REPEAT-UNTIL osv.), eller henvisninger (PROC, LABEL, GOSUB osv.). Er dette tilfældet, udvides den interne repræsentation af disse linier med oplysninger, som forøger arbejdshastigheden. Herefter starter selve programafviklingen med det angivne linienummer og fortsætter, styret af programmet.

Eksempler

RUN
 RUN 100

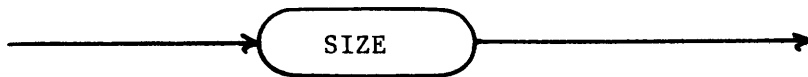
Kommentarer

1. Udelades "linienr.", begyndes med første linie i programmet.

5.7.21 SIZE

Kommandoen viser hvor stor en del af arbejdslageret, programmet optager, hvor meget fri plads, der er tilbage, samt hvor mange variable, man har brugt.

Syntaks



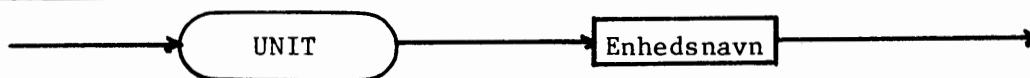
Eksempel

SIZE

5.7.23 UNIT

Kommandoen bruges til - i systemer med 2 båndstationer - at angive, hvilken enhed operativsystemet automatisk skal vælge, når ikke andet er angivet.

Syntaks



Eksempel

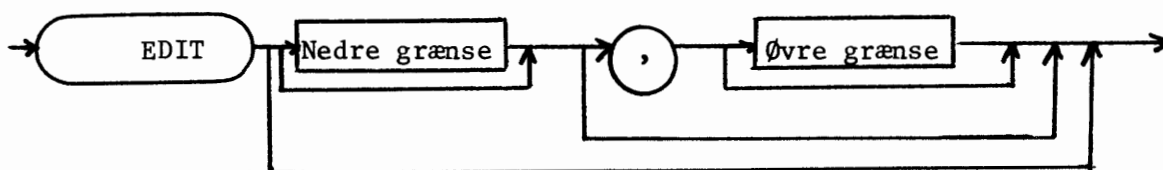
UNIT CS0:
UNIT CS1:

5.8. Editering

Rettelse af programsætninger, som allerede findes i datamatens arbejdslager, foretages ved hjælp af EDIT-kommandoen.

Syntaks:

EDIT-kommandoen har følgende opbygning:



Udførelse

Ved udførelse af EDIT-kommandoen hentes det angivne område af programmet - linie for linie - fra datamatens arbejdslager og udskrives på skærmen. Cursoren placeres umiddelbart efter sidste tegn og kan flyttes frem og tilbage over linien ved hjælp af de 2 kontroltaster mærket med henholdsvis højre- og venstrepil. Cursoren anbringes over det tegn, som skal ændres, det nye tegn indtastes, og cursoren rykker en plads til højre.

Når de ønskede rettelser er udført, trykkes på "RETURN", hvorefter linien syntaks-kontrolleres; og hvis den nu er korrekt, lægges den på plads i arbejdslageret. Næste linie vises, og hele forløbet gentages til "Øvre grænse" nås. Det er ikke nødvendigt at flytte cursoren hen bag sidste tegn.



Kommentarer

1. Hvis "nedre grænse" udelades, begyndes med programmets første linie.
2. Hvis "øvre grænse" udelades, fortsættes til programmets sidste linie.
3. Hvis begge grænser udelades, begyndes med programmets første linie og fortsættes til sidste linie.
4. Hvis kun "nedre grænse" angives (ikke efterfulgt af ",") editeres kun den pågældende linie.
5. Hvis tegnet under cursoren ønskes slettet, nedtrykkes "DEL"-tasten, hvorefter resten af linien rykker en plads til venstre.
6. Hvis der er brug for mere plads, åbnes linien umiddelbart foran cursoren ved tryk på "INS"-tasten.
7. Linienummeret kan også editeres, hvilket medfører, at linien lægges på den plads i arbejdslageret, som det nye linienummer angiver. Hvis der i forvejen findes en linie med dette nummer, slettes denne. Den originale linie fjernes i dette tilfælde ikke fra programmet, (brug kommandoen DEL xxx).
8. Når "RETURN" nedtrykkes, lægges linien på plads i arbejdslageret nøjagtigt som den står på skærmen og uden hensyn til, hvor cursoren står.
9. EDIT-kommandoen kan når som helst afbrydes ved tryk på "ESC"-tasten, men ændringer indføres kun i programmet, når "RETURN"-tasten nedtrykkes.



5.9 TEGN-koder

Decimal- og oktal-talværdier af ASCII-tegn

COMET mikrodatamaten overholder denne standard, men der er visse af koderne (bl.a. de specielle fra 0-31) som ikke er anvendt i øjeblikket.

Tegn	Dec.	Oktal	Tegn	Dec.	Oktal	Tegn	Dec.	Oktal	Tegn	Dec.	Oktal
NUL	0	0	blank	32	40	@	64	100	~	96	140
SOH	1	1	!	33	41	A	65	101	a	97	141
STX	2	2	"	34	42	B	66	102	b	98	142
EXT	3	3	#	35	43	C	67	103	c	99	143
EOT	4	4	\$	36	44	D	68	104	d	100	144
ENQ	5	5	%	37	45	E	69	105	e	101	145
ACK	6	6	&	38	46	F	70	106	f	102	146
BEL	7	7	'	39	47	G	71	107	g	103	147
BS	8	10	(40	50	H	72	110	h	104	150
HT	9	11)	41	51	I	73	111	i	105	151
NL	10	12	*	42	52	J	74	112	j	106	152
VT	11	13	+	43	53	K	75	113	k	107	153
FF	12	14	,	44	54	L	76	114	l	108	154
CR	13	15	-	45	55	M	77	115	m	109	155
MSP	14	16	.	46	56	N	78	116	n	110	156
MNL	15	17	^	47	57	O	79	117	o	111	157
DLE	16	20	0	48	60	P	80	120	p	112	160
DC1	17	21	1	49	61	Q	81	121	q	113	161
DC2	18	22	2	50	62	R	82	122	r	114	162
DC3	19	23	3	51	63	S	83	123	s	115	163
DC4	20	24	4	52	64	T	84	124	t	116	164
NAK	21	25	5	53	65	U	85	125	u	117	165
SYN	22	26	6	54	66	V	86	126	v	118	166
ETB	23	27	7	55	67	W	87	127	w	119	167
CAN	24	30	8	56	70	X	88	130	x	120	170
EM	25	31	9	57	71	Y	89	131	y	121	171
SUB	26	32	:	58	72	Z	90	132	z	122	172
ESC	27	33	;	59	73	Æ	91	133	æ	123	173
FS	28	34	<	60	74	Ø	92	134	ø	124	174
VP	29	35	=	61	75	Å	93	135	å	125	175
RS	30	36	>	62	76	↑	94	136	↷	126	176
US	31	37	?	63	77	-	95	137	Del	127	177

Operativsystem

COMET type 2001 er udstyret med et avanceret kassetteoperativsystem, som, i modsætning til systemprogrammerne, er indlagt i et permanent (ROM) lager. Når "RESET"-knappen indtrykkes, starter datamaten automatisk afviklingen af dette program, som på skærmen giver udskriften:

COMET Type 2001
Indsæt systembånd

NB: Der må først indsættes bånd i kassettebåndoptageren, når skærmen umiddelbart efter aktivering af "RESET"-knappen har vist ovennævnte udskrift.

Operativsystemet stiller nu følgende kommandoer til rådighed for brugeren:

<u>Kommando</u>	<u>Funktion</u>
CAT	Udskriver katalog
DELETE	Sletter filer
FORMAT	Initierer ubrugt kassettebånd
INIT	Initierer tidligere skrevet kassettebånd
GET UNIT	Udskriver default enhed
RELEASE	Spoler båndet tilbage
UNIT	Sætter default enhed

Operativsystemet er filorienteret, det vil sige data behandles som en navngivet sammenhørende mængde, der flyttes fra et sted til et andet.

Ovenstående kommandoer arbejder enten på enheder eller filer, hvorfor man ved anvendelsen af dem enten skal angive en enhedsspecifikation, en filspecifikation eller ingenting (GET UNIT). Generelt kan en enhedsspecifikation udtrykkes som:

[<enhedstype>[<enhedsnr>]:]

og en filspecifikation kan udtrykkes som:

[<enhedstype>[<enhedsnr>]:]<filnavn>[.<ext>][(<versionsnr>)]



Den anvendte notation skal forstås således, at elementer, som er omgivet af sammenhørende skarpe paranteser, eventuelt kan udelades. Man kan se, at enhedsspecifikationen eventuelt helt kan falde bort, i hvilket tilfælde systemets automatisk vil vælge defaultenheden.

De enkelte elementer har følgende betydning:

<enhedstype> angiver den ydre enhed, der skal samarbejdes med. Der er her mulighed for:

CS, som angiver kassettebåndoptager
LP, som angiver linieskriver

<enhedsnr.> Når der er flere enheder af samme type, angiver dette nr., hvilken aktuel enhed der skal anvendes. I en COMET udstyret med 2 båndstationer, er der mulighed for:

0
1

<filnavn> Op til 8 bogstaver eller tal samt understregning som angiver navnet på de data, der skal flyttes. Under skrivning af filnavn kan såvel små som store bogstaver anvendes, men eventuelle små bogstaver ændres til store. Følgende 2 filnavne er derfor identiske:

ABCD abcd

<ext> Er en 3-tegns kode, som angiver filens type. Som eksempler på denne kode kan nævnes:

SYS : systemfil

CML : comal kildetekst

DAT : datafil

Koden dannes automatisk af de systemer, som genererer pågældende filer, og brugeren behøver kun at angive koden ved sletning af filer.



<versions nr.> Operativsystemet tildeler automatisk et versions nr. til hver fil, som udskrives på bånd. I forbindelse med læsning fra båndet kan man derfor skelne mellem flere filer med identisk filnavn, ved at angive dette nummer omgivet af "()". Udelades versions nr., hentes automatisk den version af den givne fil, som har det højeste nr.

Eksempler på legale enhedsspecifikationer:

CS1:
LP:

Eksempler på legale filspekifikationer:

PROGRAM
CSO:HANS.CML(2)
CS:PASCAL.SYS



Tr-87-034





Vedligeholdelse

COMET MIKRO-datamaterne er konstrueret til at arbejde i mange år uden nogen form for vedligeholdelse.

Dog skal kassettebåndoptagernes læse/skrivehoveder renses for hver ca. 100 drifttimer. Dette sker ved at indsætte et specielt rensbånd med den fyldte spole til højre. Derefter skrives "INIT" efterfulgt af et tryk på "RETURN"-tasten. Båndet vil nu starte.

Båndet skal køres een gang hver vej. Det stopper automatisk. Dette tager ca. 10 sekunder. Hvis båndet ikke stopper trykkes der på "RESET", hvorefter rensbåndet stopper og kan udtages.

Rensbåndet kan bruges 10 gange og skal derefter kasseres. Markér rensbåndet for hver gang, det har været brugt.

BEMÆRK: Rensbåndet må ikke benyttes hyppigere end foreskrevet her. Ved overdreven anvendelse vil læse/skrivehovederne blive ødelagt. 100 drifttimer svarer til ca. 1 års brug af COMET'en.

Datamat og tastatur kan rengøres med normale husholdnings- og rengøringsmidler, men opløsningsmidler må ikke anvendes.

Fejlmelding

Ved fejl, som ikke kan afhjælpes af brugeren (se appendix A II), kontaktes producenten:

HH-Electronic Aps
Højvangen 6
3480 Fredensborg

telefonnr. 03 - 28 38 41



Garantibestemmelser

Producenten yder gennem ICL A/S, der har leveret udstyret, følgende 12 måneders garanti regnet fra leveringsdagen:

A Garantiens omfang

Garantien omfatter udstyret med samtlige dele og omfatter materialeprisen og arbejds løn. Garantien omfatter ikke skader og tab, som udstyret eller brugen af dette måtte give anledning til.

B Garantien bortfalder såfremt:

- o Andre end leverandøren eller fabrikanten har foretaget ændringer eller indgreb i udstyret.
- o Fabrikationsnummeret er fjernet eller ødelagt.
- o brugervejledningen ikke er fulgt, for eksempel ved at udstyret har været tilsluttet forkert spænding.
- o Udstyret har været udsat for fysisk overlast.

C Fremgangsmåden for at gøre garantien virksom

Defekte enheder eller dele fremsendes efter forudgående aftale for reparation eller udskiftning. I størst muligt omfang tilstræbes garanti og vedligeholdelse at ske efter udskiftningsprincippet baseret på en forudgående fejlsøgning efter de retningslinier, som er beskrevet i brugervejledningen.

Nærværende garanti indskrænker ikke på noget punkt de rettigheder, som tilkommer køber i henhold til købelovens bestemmelser.



Tilbehør til COMET

Følgende adresser på leverandører af tilbehør skal betragtes som orientering fra vor side. ICL A/S har ingen speciel tilknytning til disse firmaer.

NB: ICL A/S er interesseret i at udvide denne adresseliste, således at levering af f.eks. papir kan ske mest hensigtsmæssigt.

ICL A/S modtager derfor gerne information om lokale leverandører. Vi vil derefter medtage disse ved næste opdatering.

Kassettetape - Philips

Mini Digital Cassette LDB 4401/01
Type 8920 440 10101
Pakningsstørrelse 6 styks æske.
Pris 15.2.1980 pr. stk. kr. 33,00 exclusive moms.
Faktura tillægges porto.

Leverandør: COPAX A/S
Århusgade 88
2100 København Ø
Tlf.nr. 01-38 60 22.

(COPAX A/S er Philips' detail-selskab).

Rensetape - Philips

Cleaning Tape LFH 0009
Type 8935 000 90001
Pris 15.2.1980 pr. stk. kr. 33,00 exclusive moms.
Faktura tillægges porto.
Kan leveres som 1 stk.

Leverandør: COPAX A/S
(Som for kassettetape).

NB: Rensetape må kun anvendes for hver 100 timers brug af båndstationen - hvilket svarer til ca. een gang hvert år.

Rensetape skal kasseres efter 10 ganges gennemløb - markér kassetten med et x for hver gang, den anvendes.



Printerpapir i ruller

- o Hvis skolen har telex bestilles papir lettest hos Post & Telegrafvæsenet:

P & T Materialdepotet
Amagerfælledvej 52
2300 København S
Tlf.nr. 01-57 08 22 lokal 253

opgiv eget telex nr. samt tilbagesvar.

"Store ruller" kan anvendes.

Leveringstid 4-5 dage.

For yderlig information se telexhåndbogen.

Pris pr. rulle ca. 10,00 (uden kopi).

- o Hvis skolen ikke har egen telex må levering ske gennem de sædvanlige papirleverandører eller f.eks.

Tecnel A/S
Kronprinsessevej 4A
3480 Fredensborg
Tlf.nr. 03-28 23 53.

Vi imødeser gerne flere adresser!!

- o Printerpapir i Zig-Zag med huller i kanten samt selvklæbende etiketter, specialformularer m.m. kan leveres gennem de sædvanlige formularleverandører - se fagbogen under "Databehandling - Maskiner og art. f. databehandling".

Vær opmærksom på formatbredden - se specifikationerne i printerens instruktionsbog.

Vi imødeser gerne adresser - specielt på firmaer, som har A4 format i standardbredde liggende på lager.

Opdatering af systemprogrammel

Når nyt systemprogrammel foreligger, udsender ICL meddelelse herom til brugerne.

Der kan være tale om nye versioner af COMAL-80, forskellige andre programmeringssprog samt en række hjælperutiner, som vil blive udviklet i fremtiden.

Sammen med meddelelsen om nyt systemprogrammel vil ICL udsende en specifikation af, hvad det skal anvendes til, hvilke eventuelle fejl og mangler, der allerede er kendte, samt, hvis det er relevant, hvilke ændringer og forbedringer, der er foretaget i forhold til tidligere udsendte versioner.

Hvis man ønsker det nye systemprogrammel, skal man fremsende en blank eller ikke anvendt cassette-tape/diskette til:

HH-Electronic ApS
Højvangen 6
3480 Fredensborg

(telefon 03-283841)

Sammen med dette sendes oplysninger om, hvilket systemprogrammel, man ønsker, samt hvilken adresse (inclusive modtageren), det skal sendes til.

Det nye systemprogrammel vil i så fald blive returneret i løbet af ca. en uge.



Rapportering af principielle fejl

Det er i alles interesse, at hardware, COMAL-80, operativsystem m.m. fungerer som beskrevet i brugervejledningen, og at denne også iøvrigt er korrekt. Vi vil derfor opfordre brugerne til at rapportere eventuelle principielle fejl og mangler ved hjælp af omstående blanket (side 6.7) eller en kopi af denne.

Bemærk venligst følgende:

1. Fejlene skal være almene og principielle.

De skal altså kunne fremkaldes på en almindelig COMET med den seneste udgave af systemprogrammet. Fejl, der formodes at være specielle for brugerens egen maskine, skal blot rapporteres som angivet på side 6.1.

2. Fejlene skal være nyopdagede.

Dvs. der skal ikke rapporteres fejl, der optræder på listen over kendte fejl i appendix A4, eller fejl, der allerede vides at være dokumenteret overfor ICL.

Iøvrigt er forslag til forbedringer og nyudviklinger særdeles velkomne.



Blanket til rapportering af principielle fejl

Denne blanket (eller en kopi af den) kan indsendes sammen med evt. yderligere dokumentation til:

ICL A/S
Bredgade 23
1260 København K.

Kuverten bedes mærket "COMET".

Fejl i forbindelse med (sæt krydser):

- Hardware COMAL-80 OPERATIVSYSTEM
- BRUGERVEJLEDNING ANDET (Hvad?)

KORT BESKRIVELSE AF FEJLEN: _____

EVT. VEDLAGT NUMMERERET DOKUMENTATION (Sæt krydser)

- 1. DETAILLERET BESKRIVELSE AF FEJLEN
- 2. KARAKTERISTIKA VED KONFIGURATION
(Type og serienummer af hardware
Type og versionsnummer af programmel)
- 3. EVT UDSKRIFT PÅ PAPIR
- 4.
- 5.
- 6.

DERES NAVN:
ADRESSE:
.....

TELEFONNR: TRÆFFETID:

DATO:



Appendix A1Fejlmeldinger fra COMAL-80

Fejl nr.	1	Lagerplads opbrugt
- -	2	Syntaksfejl
- -	3	Overløb
- -	4	Ikke \$/# her
- -	5	Kun for strenge
- -	6	Ikke nyt navn her
- -	7	Ikke flere nye navne
- -	8	Ikke afsluttet streng
- -	9	Ulovligt tegn
- -	10	Ulovligt tegn
- -	11	Ulovligt linienummer
- -	12	For lang linie
- -	13	Variabel forventet
- -	14	') ' forventet
- -	15	Typekonflikt
- -	16	For kompliceret udtryk
- -	17	' (' forventet
- -	18	Typekonflikt i parameter
- -	19	Har ingen parametre
- -	20	Forkert type
- -	21	' , ' forventet
- -	22	TAB ikke tilladt her
- -	23	Operand forventet
- -	24	Konstant forventet
- -	25	' : ' forventet
- -	26	Funktion ikke tilladt her
- -	27	:=/:+/:-/= brugt forkert
- -	28	:=/:+/:- / forventet
- -	29	' ; ' ikke tilladt her
- -	30	' * ' forventet
- -	31	Linie-slut her?
- -	32	Ukendt device
- -	33	Et navn forventet
- -	34	Se manualen
- -	35	' OF ' forventet
- -	36	Ikke streng-funktion
- -	37	Linie-nummer forventet
- -	38	GOTO/GOSUB forventet
- -	39	Ikke efter ' THEN '
- -	40	Et navn forventet



Fejl nr.	41	Tabel ikke tilladt
- -	42	TO/DOWNTO forventet
- -	43	READ/WRITE forventet
- -	44	Fra > = til
- -	45	Linieslut forventet
- -	46	Programsætning forventet
- -	47	Kommando forventet
- -	48	Fejl i programstruktur
- -	49	Typekonflikt
- -	50	Fejl i programstruktur
- -	51	Dobbeltdefineret
- -	52	Funktions-navn forventet
- -	53	Navne-konflikt med PROC/DEF
- -	54	FOR-NEXT dybde
- -	55	Ukendt linienummer
- -	56	RESTORE: Kun til data-sætning
- -	57	Kontrolstruktur ikke afsluttet
- -	58	Kontrolstruktur ikke afsluttet
- -	59	Kontrolstruktur ikke afsluttet
- -	60	Kontrolstruktur ikke afsluttet
- -	61	Kontrolstruktur ikke afsluttet
- -	62	Kontrolstruktur ikke afsluttet
- -	63	Kontrolstruktur ikke afsluttet
- -	64	Ukendt PROC/DEF/LABEL
- -	65	For kompliceret program-struktur
- -	66	Indgreb i DATA-pil
- -	67	Index-fejl
- -	68	Ikke indført endnu
- -	69	Ikke delstreng her
- -	70	For få indices
- -	71	For mange indices
- -	72	Ikke flere data
- -	73	Fejl i tildeling til delstreng
- -	74	Kun for arrays
- -	75	Fejl i USING-strengen
- -	76	Ulovlig TAB-værdi
- -	77	Variablen findes allerede
- -	78	Se manualen
- -	79	Navne-konflikt med PROC/DEF
- -	80	CASE-værdi findes ikke
- -	81	STEP=0
- -	82	Systemfejl
- -	83	Systemfejl
- -	84	Ude af def.-område
- -	85	For lang
- -	86	Overløb
- -	87	Udefineret variabel eller funktions-værdi
- -	88	For lang
- -	89	Ikke nu
- -	90	Index-fejl



Fejl nr.	91	Typekonflikt i parameter
- -	92	For mange parametre
- -	93	For få parametre
- -	94	Division med 0
- -	95	Systemfejl
- -	96	Typekonflikt
- -	97	For lang linie
- -	98	Ikke nu
- -	99	Fejl i NEXT
- -	100	':' ikke tilladt her
- -	101	Ingen linier med angivet nr.
- -	102	Umuligt
- -	103	Umuligt
- -	104	Umuligt
- -	105	Auto overløb
- -	106	Intet program
- -	107	Programmet er ikke prepass'ed
- -	108	Systemfejl

Fejlmeldinger fra operativsystemet

Fejl nr.	201	Kanalen er allerede åben
- -	202	Kanalen er ikke åben
- -	203	Ulovligt kanalnummer
- -	204	Ukendt i/u enhed
- -	205	Ukendt i/u enhed
- -	206	Fejl i fil-navn
- -	207	Fejl i fil-type
- -	208	Fejl i versions-nr.
- -	209	Ingen fil-type angivet
- -	210	Fil-type ikke tilladt her
- -	211	Fejl m.h.t. privilegeret status
- -	212	Fejl m.h.t. fil-organisation
- -	213	Læse-beskyttet
- -	214	Kan ikke skrive
- -	215	Kan ikke læse
- -	216	Allerede åben i en anden mode (tilstand)
- -	217	Filen er i brug
- -	218	Systemfejl
- -	219	Kun en åben fil pr. bånd
- -	220	Filen findes ikke
- -	221	Versions-nr. ikke tilladt her
- -	222	Ikke INIT'et
- -	223	Allerede INIT'et
- -	224	Umuligt: en fil er åben der
- -	225	ID-COMAL bånd



**International
Computers
Limited a/s**

Hovedkontor

Jylland & Fyn

Bredgade 23
1260 København K
Telefon 01-13 55 88

Romancevej 9
8700 Horsens
Telefon 05-62 75 88

Reg.nr.
40572

Postgiro
5 45 30 11

Telegram
Computel

Telex 19724 iclcph dk
Telex 22467 iclsal dk

Vedr.: ICL's Uddannelseskoncept

Dato

Deres ref.

Vor ref. HA/hh 2-80 5206

Leveringen af den dansk udviklede og dansk producerede MIKRO-datamat COMET er nu i fuld gang. Produktionen er i faste rammer og produktet har vist sin stabilitet i daglig brug.

Samtidig foreligger beskrivelsen af ICL's uddannelseskoncept genoptrykt i revideret udgave. Vedlagte blå hæfte indeholder:

- o Status for projektet
- o Beskrivelse af ICL's uddannelseskoncept
- o Specifikationer for COMET
- o Specifikationer for COMAL-80

Det er os en glæde at kunne fremsende denne dokumentation, der beskriver resultatet af vort nære samarbejde med uddannelsessektoren. Dette resultat er dels markedets mest fleksible og kraftigste MIKRO-datamat og dels et fungerende hierarkisk netværk baseret på datakommunikation med store datamaskiner af typen ICL 2903, 2904 og ICL 2905.

Dette giver Dem mulighed for enten at etablere og igangsætte det totale løsningskoncept nu eller at starte med en enkelt COMET med sikkerhed for udbygningsmuligheder samt for at kunne vokse ind i et koncept, som er færdigudviklet.

Vi ser frem til en nærmere kontakt, f.eks. en demonstration, som vi er overbevist om, kan blive indledningen til et frugtbart samarbejde.

Med venlig hilsen
INTERNATIONAL COMPUTERS LIMITED A/S

Henning Adrian

bilag: ICL's Uddannelseskoncept - En totalløsning

OKI-printeren kan skrive med såvel små som
STORE bogstaver.

Endvidere kan den skrive med tre forskellige
skrifttyper: 16.5 tegn pr. tomme, hvilket
ser sådan ud

Nu skrives der med 16.5 tegn pr. tomme

Nu skrives der med 16.5 tegn pr. tomme

Nu skrives der med 16.5 tegn pr. tomme

Nu skrives der med 16.5 tegn pr. tomme

Nu skrives der med 16.5 tegn pr. tomme

eller med 10 tegn pr. tomme, hvilket ser
sådan ud

Nu skrives der med 10 tegn pr. tomme

Nu skrives der med 10 tegn pr. tomme

Nu skrives der med 10 tegn pr. tomme

Nu skrives der med 10 tegn pr. tomme

Nu skrives der med 10 tegn pr. tomme

eller med 5 tegn pr. tomme, og det ser sådan
ud

Nu	skrives	der	med	5	tegn	pr.	tomme
Nu	skrives	der	med	5	tegn	pr.	tomme
Nu	skrives	der	med	5	tegn	pr.	tomme
Nu	skrives	der	med	5	tegn	pr.	tomme
Nu	skrives	der	med	5	tegn	pr.	tomme

MAN KAN OGSÅ ÆNDRE PÅ AFSTANDEN MELLEMLINJERNE,
IDET MAN ENTEN KAN BRUGE

6 LINJER PR. TOMME

6 LINJER PR. TOMME

6 LINJER PR. TOMME

6 LINJER PR. TOMME

6 LINJER PR. TOMME

ELLER MAN KAN BRUGE

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

8 LINJER PR. TOMME

GRAFISKE TEGN PÅ PRINTER.

129 : ■	130 : ■			
131 : ■■	132 : ■	133 : ■■	134 : ■■	135 : ■■
136 : ■	137 : ■■	138 : ■■	139 : ■■	140 : ■■
141 : ■■	142 : ■■	143 : ■■	144 : ■	145 : ■■
146 : ■■	147 : ■■	148 : ■■	149 : ■■	150 : ■■
151 : ■■	152 : ■■	153 : ■■	154 : ■■	155 : ■■
156 : ■■	157 : ■■	158 : ■■	159 : ■■	160 : ■■
161 : ■■	162 : ■■	163 : ■■	164 : ■■	165 : ■■
166 : ■■	167 : ■■	168 : ■■	169 : ■■	170 : ■■
171 : ■■	172 : ■■	173 : ■■	174 : ■■	175 : ■■
176 : ■■	177 : ■■	178 : ■■	179 : ■■	180 : ■■
181 : ■■	182 : ■■	183 : ■■	184 : ■■	185 : ■■
186 : ■■	187 : ■■	188 : ■■	189 : ■■	190 : ■■
191 : ■■	192 : ■■	193 : ■■	194 : ■■	195 : ■■
196 : ■■	197 : ■■	198 : ■■	199 : ■■	200 : ■■
201 : ■■	202 : ■■	203 : ■■	204 : ■■	205 : ■■
206 : ■■	207 : ■■	208 : ■■	209 : ■■	210 : ■■
211 : ■■	212 : ■■	213 : ■■	214 : ■■	215 : ■■
216 : ■■	217 : ■■	218 : ■■	219 : ■■	220 : ■■
221 : ■■	222 : ■■	223 : ■■	224 : ■■	225 : ■■
226 : ■■	227 : ■■	228 : ■■	229 : ■■	230 : ■■
231 : ■■	232 : ■■	233 : ■■	234 : ■■	235 : ■■
236 : ■■	237 : ■■	238 : ■■	239 : ■■	240 : ■■
241 : ■■	242 : ■■	243 : ■■	244 : ■■	245 : ■■
246 : ■■	247 : ■■	248 : ■■	249 : ■■	250 : ■■
251 : ■■	252 : ■■	253 : ■■	254 : ■■	255 : ■■
256 :				



Fejl nr.	226	Simpel i/u enhed
-	227	Intet bånd
-	228	Blankt bånd
-	229	Skrive-beskyttet bånd
-	230	Fil-kataloget er fuldt
-	231	Båndet er fuldt
-	232	Filen er ikke sidst på båndet
-	233	Brugen af filen afviger fra, hvad der blev angivet ved åbning
-	234	End-Of-File
-	235	Command Overrun
-	236	End-Of-Tape
-	237	Forkert blok-længde
-	238	For meget bånd-støj
-	239	Læse-fejl

Herudover er de nuværende versioner af COMAL-80 udstyret med et stort antal tests, som forhindrer ødelæggelse af programmet, hvis der sker noget uventet. Disse tests giver følgende udskrift:

Fejl xxx Systemfejl

hvor xxx er et tal større end 239.

Under indlæsning af systemprogrammer kan kassetteoperativsystemet sende begrænsede fejlmeldinger. Disse melder sig således:

Fejl x

hvor x er et af bogstaverne A-I.

De enkelte fejlmeldinger har følgende betydning:

Fejl A	Forkert katalogsystem
- B	Anden fejl ved INIT
- C	Ingen filer på båndet
- D	Første fil på båndet har ikke ext..AUT
- E	Første fil har forkerte attributter
- F	Anden fejl ved åbning
- G	Filen er for lang/ for lidt lagerplads
- H	Læsefejl
- I	For lang blok i filen (>4096 bytes)



Tr 87-034





Tf 87-084

Appendix A4Alfabetisk oversigt over de vigtigste styre-kommandoer og COMAL-80-kommandoer til mikrodatamaten COMET

<u>NAVN</u>	<u>SYNTAKS</u>	<u>FUNKTION</u>
AUTO	AUTO eller AUTO startnr., spring	Genererer linienumre til identifikation af Comal- sætninger.
CAT	CAT	Viser indholdet af det aktu- elle båndes katalogblok, som ligger i datamatens lager.
CLEAR	CLEAR	Sletter skærmen.
CON	CON	Fortsætter udførelsen af et program, som er blevet standset, enten af Comal-fortolkeren p.g.a. programfejl, eller af brugeren ved tryk på ESC-tasten.
DEL	DEL linienr. eller DEL fra-nr., til-nr	Sletter en enkelt linie eller alle linier indenfor et inter- val af linienumre fra program- met i arbejdslageret.
DELETE	DELETE filnavn	Sletter filen, filnavn, på kassettebåndet, forudsat det er den sidst skrevne fil.
EDIT	EDIT eller EDIT linienr.	Kalder linieeditoren, ved hjælp af hvilken programlinier kan korrigeres. Kaldes editoren uden angivelse af linienr., får man adgang til alle pro- grammets linier en ad gangen i nummerorden. Kaldes editoren med angivelse af linienr., har man kun adgang til den specifi- cerede linie.
ENTER	ENTER filnavn	Henter filen (programmet) fra det aktuelle bånd ind i arbejdslag- eret. Programmet skal være gemt ved LIST-kommandoen (i ASCII- kode).



COMET BRUGERVEJLEDNING

<u>NAVN</u>	<u>SYNTAKS</u>	<u>FUNKTION</u>
FORMAT	FORMAT enhed:, båndnavn	Initialiserer katalogblokken på et nyt bånd. Båndet identificeres ved navnet båndnavn.
INIT	INIT	Læser katalogblokken fra det aktuelle bånd ind til et område i datamatens lager.
LET	LET variabelnavn = konstant	Tildeler variabelen, identificeret ved variabelnavn, værdien konstant.
LIST	a) LIST eller b) LIST linienr. eller c) LIST fra-nr., til-nr.	Viser hele programmet eller udvalgte linier fra programmet i arbejdslageret. Benyttes syntaks-regel a vises hele programmet, benyttes regel b vises den ved linienr. specificerede linie og benyttes regel c vises den del af programmet, hvor linienumrene falder indenfor intervallet givet ved fra-nr. og til-nr. Hvis det antal linier, der skal vises, fylder mere end én skærm, bladres videre frem ved enten at trykke på mellemrumstangenten (helt nyt skærbillede) eller på RETURN (en linie pr. tryk).
LIST	a) LIST LP: b) LIST linienr. LP: c) LIST fra-nr., til-nr. LP:	Samme funktioner som ovenfor, men programmet listes på printer i stedet for skærm.
LIST	LIST filnavn	Udskriver programmet, som ligger i arbejdslageret, i ASCII-kode på det aktuelle bånd og opdaterer båndets katalogblok i datamatens lager. Filen, som herefter indeholder programmet, identificeres ved filnavn.
LOAD	LOAD filnavn	Henter filen (programmet) fra det aktuelle bånd ind i arbejdslageret. Programmet skal være gemt ved SAVE-kommandoen (i binært format).



COMET BRUGERVEJLEDNING

<u>NAVN</u>	<u>SYNTAKS</u>	<u>FUNKTION</u>
PRINT	PRINT variabelnavn	Udskriver det aktuelle indhold af variabelen identificeret ved variabelnavn.
RELEASE	RELEASE	Skriver den opdaterede katalogblok, som ligger i datamatens lager, ud på det aktuelle bånd.
RENUMBER	RENUMBER	Renummerer linienumrene på programmet, som ligger i arbejds-lageret. Første programlinie får nummeret 10, og de følgende linier tildeles numre med spring på 10.
RUN	RUN	Eksekverer programmet, som ligger i arbejds-lageret.
SAVE	SAVE filnavn	Udskriver programmet, som ligger i arbejds-lageret, i binær form på det aktuelle bånd og opdaterer båndets katalogblok i datamatens lager. Filen, som herefter indeholder programmet, identificeres ved filnavn.

APPENDIX A4 (fortsat)

Lagring af programmer på kassettebånd

I det følgende beskrives dels hvordan et kassetbånd er opbygget og dels hvilke kommandoer, der skal benyttes i forbindelse med anvendelse af kassettebånd til lagring af programmer.

1. Båndets opbygning

En side på et kassettebånd er, set fra COMET's side, opdelt i en række mindre dele af fast længde, såkaldte blokke. Kassettebåndet ser derfor ud som vist på skitsen nedenfor:



Man kan af skitsen se, at der findes følgende 3 typer af blokke:

- Katalog-blok
- Data-blok
- Bånd-slut-blok

Katalog-blok

Forrest på båndet findes en indholdsfortegnelse, et katalog, over de programmer, der er lagret på båndet.

Data-blok

Efter katalogblokken følger en række datablokke, hvori programmerne er lagret. Der anvendes et helt antal data-blokke til lagring af et COMAL-program.

Bånd-slut-blok

Efter den sidste benyttede data-blok findes en bånd-slut-blok, som blot anvendes til at angive, at der ikke er lagret flere programmer på båndet.



2. Læsning og opdatering af katalog-blok

Inden et kassettebånd, hvorpå der allerede er lagret programmer, kan benyttes af COMAL-systemet, skal båndets katalog-blok indlæses til COMET'ens arbejdslager. Dette gøres ved at indtaste kommandoen:

INIT

efter at kassettebåndet er sat på plads i båndoptageren.

Når man er færdig med at benytte kassettebåndet, skal man, inden det fjernes fra båndoptageren, sørge for at kataloget, som ligger i arbejdslageret, bliver skrevet tilbage på båndet. Dette gøres ved at indtaste kommandoen:

RELEASE

Når denne kommando er udført, kan båndet fjernes fra båndoptageren.

Glemmer man at taste RELEASE inden båndet fjernes fra båndoptageren, vil båndets katalog ikke blive opdateret i overensstemmelse med de ændringer, som eventuelt er blevet påført båndet siden udførelsen af INIT-kommandoen.

Fjernelse af et bånd fra båndstationen uden forudgående udførelse af RELEASE-kommandoen vil iøvrigt medføre at advarslen

SÆT DEN KASSETTE I IGEN DU LIGE TOG UD. DEN VAR IKKE RELEASE'D.

vil blive udskrevet blinkende på skærmen. Advarslen fjernes fra skærmen så snart båndet igen er blevet placeret i båndstationen, og denne er blevet lukket.

3. Nyt bånd

Inden man kan anvende et helt nyt bånd til lagring af COMAL-programmer, skal båndet navngives, hvilket også medfører dannelse af en katalog-blok i arbejdslageret. Denne navngivning foretages ved at indtaste kommandoen:

FORMAT <enhed>:, <båndnavn>

hvor <båndnavn> er navnet, som man ønsker at identificere båndet med. Båndnavn må bestå af maksimalt 50 alfanumeriske tegn.

Eksempel: FORMAT CSO:, ENGELSK

Udførelsen af ovenstående kommando medfører, at kassettebåndet får navnet ENGELSK, og at en katalog-blok bliver dannet i arbejdslageret.

Når FORMAT-kommandoen er udført, kan man straks begynde at lagre programmer på båndet. Det er altså i dette tilfælde ikke nødvendigt at udføre kommandoen INIT, mens kommandoen RELEASE altid skal udføres efter brugen af et bånd.

4. Lagring af programmer

Ønsker man at lagre programmet, som findes i COMET'ens arbejdslager på kassettebånd, gøres dette ved indtastning af en af følgende 2 kommandoer:

```
LIST <programnavn>    eller
SAVE <programnavn>
```

hvor <programnavn> er navnet, hvorved man ønsker at identificere programmet. <Programnavn> må bestå af maksimalt 10 alfanumeriske tegn. Udførelsen af kommandoen medfører dels at programmet bliver lagret på båndet, og dels at katalogblokken i arbejdslageret bliver opdateret. Benyttes LIST-kommandoen, lagres programmet på ascii-form, og benyttes SAVE-kommandoen, lagres programmet på binær form.

Programmer, som ønskes udført fra andre programmer v.h.a. CHAIN-konstruktioner, skal være lagret binært ved benyttelse af SAVE-kommandoen. Det forlanges yderligere af sådanne programmer, at de skal have været udført inden lagringen.

Hvis der på båndet allerede er lagret et eller flere programmer med det specificerede navn, tildeles det aktuelle program et versionsnummer, som er lig med antallet af lagrede programmer med det specificerede navn.

Eksempel. LIST VERBER

Udførelsen af ovenstående kommando bevirker, at programmet i arbejdslageret bliver gemt på ascii-form på kassettebåndet under navnet VERBER. Findes der eksempelvis allerede 2 programmer på båndet, som er gemt under navnet VERBER, da tildeles det aktuelle program versionsnummeret 3.



5. Hentning af programmer

Ønsker man at hente et lagret program ind til arbejdslageret, gøres dette ved at indtaste en af følgende 2 kommandoer:

ENTER < programnavn> eller
LOAD < programnavn>

hvor programnavn er det navn, som programmet er blevet gemt under. ENTER skal anvendes til hentning af programmer, som er lagret på ascii-form (v.h.a. LIST), og LOAD skal anvendes til hentning af programmer, som er lagret på binær form (v.h.a. SAVE).

Eksempel: LOAD VERBER

Udførelsen af ovenstående kommando medfører, at programmet VERBER bliver hentet ind fra kassettebåndet til arbejdslageret. Findes der flere versioner af programmet, bliver den nyeste version, dvs. den med det største versionsnummer, hentet ind til arbejdslageret. Ønsker man ikke den nyeste version hentet, skal man efter programnavnet i parentes angive nummeret på den ønskede version.

Ønskes således version nummer 1 af programmet VERBER hentet, gøres dette ved indtastning af kommandoen:

LOAD VERBER(1)

6. Sletning af programmer

Man har mulighed for, at slette den sidst skrevne fil på en båndside. Kommandoen, som skal benyttes til dette formål, er:

DELETE <filnavn>.<EXT>

hvor <filnavn> er navnet på filen, som ønskes slettet, og <EXT> angiver filens type (se afsnit 7).

Eksempel: DELETE VERBER.CML

Udførelsen af ovenstående kommando medfører, at COMAL-programmet, VERBER, bliver slettet fra det aktuelle kassettebånd, forudsat at det er det sidst skrevne på båndside.



7. CAT-kommandoen

Man har mulighed for at få et båndes katalog udskrevet på skærmen ved at indtaste kommandoen:

CAT

CAT-kommandoen arbejder på kataloget, som ligger i arbejds-lageret, og kan derfor først udføres når båndet er blevet klargjort ved hjælp af INIT- eller eventuelt FORMAT-komman-doen. Udskriften, som produceres af CAT-kommandoen, ser ud som vist på eksemplet herunder:

BÅND-ID: BÅND1

16 BLOKKE BRUGT

NAVN	TYPE	VERS	ATTR	LGD
PROGRAM	.CML	1	..	5
DATARAM	.DAT	1	..	3
VERBER	.CML	1	..	2
VERBER	.CML	2	..	4

Udskriften ovenfor skal fortolkes således, at der på båndet BÅND1 er brugt 16 blokke, som er benyttet til lagring af følgende 4 filer:

- PROGRAM (COMAL-program)
- DATAFIL (ascii-datafil)
- VERBER (version nr. 1, COMAL-program)
- VERBER (version nr. 2, COMAL-program).

Tilsammen fylder disse 4 filer 14 blokke. De 2 resterende blokke udgøres af katalog- og båndslut-blok.

8. Kopiering af programmer

Ønsker man at kopiere et program fra ét kassettebånd over på et andet, kan det gøres som beskrevet herunder.



Kopiering af et program (PROGRAM) fra et bånd (BÅND1) til et andet (BÅND2) kan gøres på følgende måde:

- 1 Placer BÅND1 i båndoptageren
- 2 Klargør BÅND1 (INIT)
- 3 Hent PROGRAM til arbejdslageret (ENTER PROGRAM,
LOAD)
- 4 Frigør BÅND1 (RELEASE)
- 5 Fjern BÅND1 fra båndoptageren
- 6 Placer BÅND2 i båndoptageren
- 7 Klargør BÅND2 (INIT eller
FORMAT CSO:, BÅND2)
- 8 Gem PROGRAM på BÅND2 (LIST PROGRAM)
(SAVE)
- 9 Frigør BÅND2 (RELEASE)
- 10 Fjern BÅND2 fra båndoptageren

Nu er PROGRAM kopieret fra BÅND1 til BÅND2, og de ovenfor beskrevne 10 trin kan gentages, hvis flere programmer fra BÅND1 ønskes kopieret over på BÅND2.

9. Kortfattet oversigt over kommandoanvendelseA. Nyt bånd

FORMAT CSO:, <båndnavn>	Navngiv båndet
○	
○ (LIST, ENTER, CAT, DELETE) SAVE, LOAD,	Arbejd med båndet
○	
RELEASE	Frigør båndet

B. Tidligere benyttet bånd

INIT	Klargør båndet
○	
○ (LIST, ENTER, CAT, DELETE) SAVE, LOAD,	Arbejd med båndet
○	
RELEASE	Frigør båndet

C. Kopiering

INIT	Klargør bånd-1
ENTER LOAD <program>	Hent program
RELEASE	Frigør bånd-1
INIT (eller evt. FORMAT CSO:, <bånd-2>)	Klargør bånd-2
LIST SAVE <program>	Gem program
RELEASE	Frigør bånd-2

Appendix 4 fortsatAnvendelse af linjeskriver

Ønsker man at udskrive programmet, som ligger i arbejds-lageret, på en tilsluttet skriveenhed, skal det gøres v.h.a. kommandoen

LIST LP:

eventuelt forsynet med en linjenummerangivelse.

Eksempel.

LIST 30, 80 LP:

Udførelsen af ovenstående kommando medfører, at linjerne 30-80 i det aktuelle program bliver udskrevet på skriveenheden.

Ønsker man at styre uddata fra et programs PRINT-sætninger ud på linjeskriveren, skal man inden udførelsen af PRINT-sætningerne have udført en

SELECT LP:

sætning. Udførelsen af denne sætning vil medføre, at uddata fra alle efterfølgende PRINT-sætninger vil blive skrevet på linjeskriveren. Udskrivning på skærmen kan genskabes ved at udføre en

SELECT DS:

sætning.

Eksempel.

10 SELECT : LP

20 PRINT "PÅ LINJESKRIVER"

30 SELECT : DS

40 PRINT "PÅ SKÆRM"

Udførelse af ovenstående program vil medføre at teksten

PÅ LINJESKRIVER

udskrives på linjeskriveren, og at teksten

PÅ SKÆRM

udskrives på skærmen.

Appendix 4 (fortsat)Fejl og mangler ved COMAL-80

Den nu foreliggende version v. 0.92 af COMAL-80 er behæftet med nogle fejl og mangler.

1. Ikke igangsatte funktioner
2. Fejlbehæftede funktioner

Det bemærkes, at syntaksanalysen fungerer for alle funktioner, også ikke implementerede, og at det er eksekveringen, som ikke er i orden. Vi arbejder iøvrigt hårdt for at rette fejlene og det forventes, at den endelige version med versions nr. 1.00 vil foreligge pr. 1. august 1980.

1. Endnu ikke igangsatProgramsætninger

READ FILE	Binær indlæsning fra FIL.
WRITE FILE	Binær skrivning på FIL
PRINT USING	
DEF - END DEF	Brugerdefinerede funktioner.

Funktioner

DIV	Heltalsdivision
MOD	Modulusfunktionen

2. Fejlbehæftede funktioner

Programsætninger

Læsning af numeriske data efter alfanumeriske data, eks.:

```
INPUT ## 1:A$,I ##
```

Der anvendes indtil videre 2 linier
for læsning og skrivning.

Systemkommandoer

CON Efter fejl 87, 49 og 96 får systemet
til at gå ned.

COMAL-80 arbejder med en nøjagtighed på 7 cifre. Ca. 1. august 1980 vil en version med 13 cifres nøjagtighed være tilgængelig. Der er endnu ikke taget stilling til, om denne vil blive indført som standard eller om den vil komme som en specialversion. Dette afhænger blandt andet af, hvordan den mere komplicerede matematikpakke påvirker systemets effektivitet.



BESKRIVELSE		Side
System	Dato	1/1
COMET	1980-06-18	Signatur GDH/mp

Opdatering

Meddelelse nr. 7

Til indehaverne af mappen: COMET BRUGERVEJLEDNING.

Hermed følger de resterende ca. 20% af afsnit 5, som dermed er en komplet referencebeskrivelse af COMAL-80.

Derudover indeholder denne opdatering følgende:

- ny indholdsfortegnelse for afsnit 5
- oplysning om opdatering af systemprogrammel (afsnit 6)
- oplysning om og skema til fejlrapportering (afsnit 6)
- oversigt over kendte fejl i COMAL-80 version 92 (appendix A4)
- kopi af brev vedrørende oprettelse af brugerklub, som ICL A/S varmt anbefaler.

Det er hensigtsmæssigt, at COMET-brugerne gemmer opdateringsmeddelelserne som dokumentation for, hvilke opdateringer af COMET BRUGERVEJLEDNINGEN, der er foretaget. Disse meddelelser foreslås derfor indsat bagerst i afsnit 10.

<u>Udtag</u>	<u>Indsæt</u>	<u>Betegnelse</u>
Indholdsfortegnelse kapitel 5 (1 side)	Ny indholdsfortegnelse (3 sider)	Indholdsfortegnelse
	Nye afsnit 5.5. til 5.7 indsættes mellem afsnit 5.4 og 5.8	
	Nyt afsnit 5.9 indsættes efter 5.8	Tegnkoder
	Side 6.5	Opdatering af systemprogrammel
	Side 6.6 og 6.7	Fejlrapportering
Side A4.12 og A4.13 i appendix A4	Side A4.12 og A4.13	Kendte fejl i COMAL-80
	Meddelelse nr. 7 i afsnit 10	Denne meddelelse

Med venlig hilsen
INTERNATIONAL COMPUTERS LIMITED A/S

Gunnar Dyekjær Hansen
Gunnar Dyekjær Hansen

Form BESKRIV (76 - 04)

1980-06-11

Sognepræst
Hans. Chr. Grosbøll-Poulsen
Gl. Rye Præstegård
8680 Ry
Tlf.nr. 06 - 89 80 37

Lærer
Morten Grosbøl Poulsen
Aalsrodevej 49
8500 Grenaa
Giro: 4 32 02 63
Tlf.nr. 06 - 33 19 03

Kære Comet-bruger/MPS 2000-medbygger

Da vi ofte har savnet kontaktmuligheder med andre COMET/MPS brugere, ønsker vi hermed at tage initiativ til at få udgivet et brugerblad.

For at afklare behovet vil vi gerne vide, hvor mange der er interesse-rede i at være med i en brugergruppe. Vi har undersøgt, hvad det vil koste at lave et blad, og vi mener, det er muligt, at fremstille og udsende et for 5-6 kr. afhængig af oplagets størrelse. Bladets omfang svarer da til 8 A4 sider (nedfotograferet til A5 format), hvilket er minimum af hensyn til omkostningerne.

Det er tanken at udsendet bladet, ikke med faste mellemrum, men når der foreligger en tilstrækkelig mængde stof, eller når der er nyheder, vi finder bør ud hurtigt.

Vi håber selvfølgelig, at vi modtager så meget stof, at de 8 sider ikke slår til, men vi vil gerne gøre helt klart, at bladet er bruger-gruppens, og at det er dens medlemmer, der skal levere indholdet. Vi føler os således ikke forpligtet til selv at fylde bladet.

Indholdet vil naturligvis afhænge af jeres interesse: hardware, software (både maskinsprog og COMAL 80), anmeldelser køb, salg, annoncere-ring osv. Mogens Pelle har også lovet at bidrage.

Vi havde tænkt os, at brugergruppens medlemmer, i den udstrækning der er plads, skal have gratis adgang til at udbyde både hardware og software til salg. Denne ordning forudsætter dog at der ikke sker misbrug.

Hvis du er interesseret, bedes du indbetale kr. 25,00. Såfremt du ikke bruger vedlagte slip, men f.eks. giro, så sørg for at huske navn og adresse, så vi ved, hvor pengene kommer fra. Vi kommer med første nummer, såsnart vi kan se at brugergruppen er en realitet.

Med venlig hilsen

M. Poulsen

Jeg ønsker at være med i brugergruppen omkring COMET/MPS 2000.

NAVN: _____

ADRESSE: _____

IDEER OG BEMÆRKNINGER: _____



KAPITEL 5 : COMAL-80

Indholdsfortegnelse

- 5.1 Indledning
- 5.2 Indtastning
- 5.3 Sprogets grundelementer
- 5.4 COMAL-80 sætninger
- 5.5* Standardfunktioner
- 5.6* Filsystem
- 5.7* Systemkommandoer
- 5.8 Editering
- 5.9* ASCII-koder
- 5.10* Reserverede ord

De med * mærkede afsnit er pr. 1980-05-13 under udarbejdelse.

Appendix 4 fortsatFejl og mangler ved COMAL-80

Den foreliggende version 0.80 og 0.85 af COMAL-80 er 1. version, som naturligt er behæftet med nogle fejl og mangler.

Efter mange opfordringer fra brugerne har vi valgt at udsende den i den foreliggende tilstand. Vi beder brugerne i anvendelsen tage højde for følgende:

1. Ikke implementerede funktioner.
2. Fejlbehæftede funktioner.

Det bemærkes, at syntaksanalysen fungerer for alle funktioner, også ikke implementerede, og at det kun er eksekveringen, som ikke er i orden. Vi arbejder iøvrigt hårdt for at rette fejlene.

1. Endnu ikke implementeret:Programsætninger

READ FILE	Binær indlæsning fra FIL.
WRITE FILE	Binær skrivning på FIL.
PRINT USING	
FOR - NEXT	
INPUT ...,	

Funktioner

DIV	Heltalsdivision.
MOD	Modulusfunktionen.

Systemkomandoer

EDIT



2. Fejlbehæftede funktioner

Programsætninger

DEF - END DEF	Brugerdefinerede funktioner.
MAT	Fungerer ikke altid for reelle og alfanumeriske tabeller.
WRITE	Af delstreng giver fejl 49, typekonflikt.
TRAP ERR-	Fungerer med alfabetiske strenge, men ikke med numeriske.

Læsning af numeriske data efter alfanumeriske data, eks.:

```
READ#1:A$,I#
```

Funktion

ATN	Argus Tangens regner forkert.
RVAL	Konvertering fra ASCII til reelt tal. Får systemet til at gå ned.
IVAL	Konvertering fra ASCII til Heltal. Får systemet til at gå ned.
RND	Ændres til SIN i udskriften men fungerer som RND.

Systemkomandoer

RENUM	Omnummerering af linienumre i et program. RENUM kan ikke opdatere linienummerreferencer korrekt.
CON	Efter fejl 87, 49 og 96 får systemet til at gå ned.