

HOME COMPUTER



BRUGSANVISNING MED PROGRAMEKSEMPLER
FOR FAMILIE . FORRETNING . UNDERVISNING . UNDERHOLDNING

copyright: peter og o.c. andersen 1983

Alle rettigheder forbeholdes.
Gengivelse, mangfoldiggørelse
eller formidling på anden måde
ved brug af elektroniske eller
mekaniske midler af denne bog,
eller dele af den, er ikke tilladt
uden skriftlig tilladelse fra
SCAN-BIT-SOFTWARE

ADVARSEL!

16K RAM MÅ IKKE AF- OG PÅMONTERES
MED STRØM PÅ COMPUTOREN. DET VIL
KUNNE FORÅRSAGE BESKADIGELSE AF
16K RAM OG COMPUTOREN.
DENNE SKADE DÆKKES IKKE AF GARANTI-
EN.

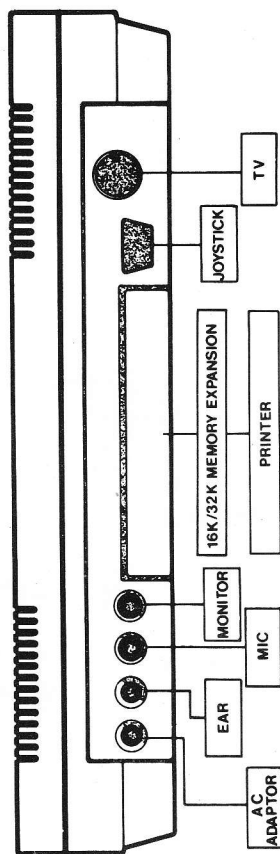
INDHOLDSFORTEGNELSE

	<u>SIDE</u>
INDLEDNING	
SEKTION 1 Opstilling af computoren	1 - 2
SEKTION 2 Opbygning af programlinier	3 - 10
SEKTION 3 Print med tekst	11 - 14
SEKTION 4 Computoren som regnemaskine	15 - 19
SEKTION 5 Basic	20 - 61
SEKTION 6 Grafik	62 - 67
SEKTION 7 Musik	68 - 72
SEKTION 8 Overføring af programmer	73 - 75
SEKTION 9 Matematiske funktioner	76 - 79
APPENDIX A Funktioner og kommandoer	80 - 88
APPENDIX B Fejl-/tilbagemelding	89
APPENDIX C Prioritetsrækkefølge	90
APPENDIX D Karakterkode	91 - 92
APPENDIX E Kontrolliste over fejl	92 - 93

KLAGØRING AF COMPUTOREN

Pak computoren ud af emballagen (gem denne). I kassen vil du finde hoveddelen (computortastatur), omformer, TV-kabel, et sæt båndoptager kabler, samt en brugsanvisning.

Fig 1-1 viser bagsiden af computoren.

Tilslutning

På bagsiden af computoren vil du finde stikkene TV og MONITOR. Skal et TV anvendes som skærm (display) for computoren, så forbind TV-stikket med TV-antenne stikket direkte med TV-kablet (HUSK, at fjerne antennekablet fra TV først). Indstil TV på UHF-båndet, kanal 36.

Båndoptager

Båndoptager kablet tilsluttes computoren ved "EAR" og "MIC" på bagsiden. Båndoptageren anvendes, når du skal indlæse (load) og, når du vil gemme (SAVE) noget.

Strømforsyning

Sæt svagstrømsstikkene i DC POWER på computoren. Anbring derefter netstikket i en stikkontakt og tænd. Nu vil den lille POWER lampe på tastaturet lyse, og der vil komme et "bib" fra computoren. Hvis fjernsynet er indstillet på den korrekte frekvens, vil skærmen blive sort og et lille hvidt kvadrat (CURSOR) vil blinke nede i venstre hjørne. Fylden TV-skærmen blot med støjstriber, skyldes det at frekvensen er forkert indstillet. Kan du ikke få skærbilledet indstillet korrekt, så kan du forsøge enten at holde ENTER-tasten nede et stykke tid og derefter trykke på RESET-tasten, eller du kan slukke for strømmen fra computoren og tænde igen.

Hvis alt er i orden, er computoren klar til programmering..

Monitor

Anvendes monitor istedet for TV er proceduren meget enkel. Du tænder for monitor, derefter tænder du for computoren. Der vil ske 3 ting, når du sætter computoren igang (ON). POWER-lampen lyser, der kommer et "bib" fra computoren og den blin-

kende kvadrat (CURSOR) kommer frem i nederst i venstre hjørne på skærmen.

Det er ikke nødvendigt med yderligere indstilling og billedet vil stå klart og stabilt på skærmen.

VELKOMMEN I COMPUTORALDEREN.

En computer består af to hoveddele, SOFTWARE - programmerne og HARDWARE, der er det udstyr, der skal udføre programmerne.

Med denne computer har du fået et HARDWARE, der åbner en ny verden for dig, dette uanset om du er 10 år eller 70 år.

Der er indbygget en oversætter i computeren, der oversætter alt hvad du beder den om, og som svarer tilbage til dig, i et sprog, der er så enkelt og logisk, at du nemt kan lære det.

Sproget der anvendes kaldes BASIC og består af 34 kommandoer, der er taget fra det engelske sprog og betydningen af ordene er oversat og vil stå i parentes.

Du vil hurtigt vænne dig til at tænke i det logiske BASIC sprog.
Så snart du ser en BASIC sætning som f.eks. LET A=5, vil dine logiske evner ikke kun opfatte A, men også at der foruden A er en talværdi nemlig A=5.

Hvis du i forvejen har kendskab til BASIC, kan du hurtigt gennemløse denne BIT GUIDE - opslagsbog.
Har du imidlertid ikke tidligere beskæftiget dig med at programmere i BASIC, vil vi anbefale dig, at du gennemgår denne opslagsbog sektion for sektion, således at du opnår sikkerhed i det grundlæggende ting.

Som du sikkert vil erfare, kræver computeren for at udføre de ting, du beder den om, at alle tegn, funktioner og kommandoer er rigtigt placeret og anvendt korrekt. Er der lavet en fejl vil computeren vise et blinkende E samtidig med at den anvender en af de 12 mulige fejlmeldinger, for dels at fortælle dig hvad du gør forkert og for dels at fortælle dig, hvor fejlen er.
Når fejlen er rettet og du kan gå videre i programmet vil computeren vise et OK.

Mange mennesker vil sikkert sige "Det finder jeg aldrig ud af", men hvis vi prøver at forestille os, at vi finder os 10 år fremme, så har vi helt sikkert vænnet os til den nye teknologi, og vil have svært ved at forestille os en hverdag uden minicomputoren.

Skulle du nu ikke være interesseret i at lære at programmere selv, kan du nøjes med at anvende de allerede færdige programmer. I dette tilfælde behøver du blot beskæftige dig med en sektion i BIT GUIDEN, nemlig den der giver en grundig beskrivelse af, hvordan man overfører programmerne til kasettebånd.
Disse færdige programmer leveres som tilbehør på kasettebånd og findes i alle mulige versioner, som f.eks. spil, undervisning, regneprogrammer o.s.v.

Et almindeligt TV med UHF, samt en kasettebåndoptager vil være nødvendigt for at anvende computeren, men det er klart at det vil være en stor fordel, hvis man er i besiddelse af et ekstra TV f.eks. 16", så man undgår at skulle af- og påmontere stik, og stille computeren op hver gang, man skal bruge den.

Der vil løbende blive udviklet nyt SOFTWARE - programmer til brug i hjem, undervisning og i erhvervslivet, og man vil løbende fremstille udstyr, der kan tilkøbes computeren.

Denne BIT GUIDE - opslagsbog - er udarbejdet til computeren i standardversion, men vil naturligvis løbende blive udvidet i takt med udviklingen.

C _ O _ D _ _ _ F _ O _ R _ N _ Ø _ J _ E _ L _ S _ E _ _ _

SEKTION 2

2-A-1	Cursor	2-E-1	Betjeningsstaster
2-A-2	Auto Line No	2-E-2	EDIT
2-A-3	Shift Auto Line No.	2-E-3	Højre pil
2-A-4	Enter (send ind)	2-E-4	DELETE (slet)
2-A-5	Shift Enter (skift til grafik)	2-E-5	Højre/venstre pile
2-A-6	E er SYNTAX ERROR	2-E-6	Op og Ned pile
2-A-7	RUN	2-E-7	Line No
2-A-8	Tilbagemeldinger	2-E-8	PRINT
2-A-9	Delete (slet)	2-E-9	EDIT
2-A-10	Op og Ned pile	2-E-10	DELETE
2-A-11	Højre og Venstre pile	2-E-11	Line No, PRINT
2-A-12	Nyt Line No.	2-E-12	Op og Ned pile/Enter
2-A-13	BREAK (bryde)	2-E-13	Højre/venstre pile/DELETE
2-A-14	NOBEEP	2-E-14	SYNTAX ERROR
2-A-15	Ø (nul)	2-E-15	Test
2-A-16	RESET		
2-A-17	NEW (ny)	2-D-1	Line No/PRINT
2-A-18	Afbryd strømmen	2-D-2	Linjestjerne
2-A-19	SHIFT (skift)	2-D-3	Ændring af linjestjernes placering
2-A-20	Cursor (hvid kvadrat)		
2-B-1	Shift/Line No.		
2-B-2	PRINT		
2-B-3	Øvelse		
2-B-4	Programlinie		
2-B-5	Enter		
2-B-6	Programliniens placering		
2-B-7	Cursor		
2-C-1	RUN		
2-C-2	ENTER		

2-A	INDLEDNING	RESULTAT
2-A-1	Den hvide kvadrat, der blinker nede i venstre hjørne kaldes CURSOR. Den markerer, hvor du befinder dig på skærmen.	
2-A-2 Auto LINE NO oplyser 10 hver gang	For at computoren skal kunne læse dine programlinier i den rigtige rækkefølge, behøver den et linienummer for hver programlinie.	10 20 30
2-A-3	Du har valget mellem selv at taste linienummer ind, med 10 eller 5 imellem hver programlinie, eller bruge SHIFT AUTO LINE NO.	
2-A-4 ENTER (send ind)	ENTER skal bruges, når du skal sende en programlinie ind i programmet, samt når du vil have udført en kommando, som du har indtastet uden linienummer.	
2-A-5 SHIFT ENTER (skift til GRAFIK)	SHIFT GRAFIK skal vi først bruge i SEKTION 6, men du skal lige vide, at hvis der fremkommer G i den hvide kvadrat CURSOR, så har du skiftet til GRAFIK. Tast SHIFT ENTER og G'et forsvinder.	G
2-A-6 E er SYNTAX ERROR (skriv og stav fejl)	Hvis der kommer en skarp brummen, samt fremkommer et blinkende E i CURSOR, når du vil sende din programlinie ind i dit program, betyder det, at der er SYNTAX ERROR.	E
2-A-7 RUN (udfører dit program)	RUN er den kommando du bruger, når du vil udføre dit program. RUN skal ligesom alle andre kommandoer uden linienummer efterfølges af ENTER for at sende det ind i dit program og udføre kommandoen. RUN sletter alle variabler.	
2-A-8 TILBAGEMELDIN- GER	TILBAGEMELDING sker hver gang du har bedt computoren om noget. En blinkende CURSOR betyder, at linie 10 er modtaget uden SYNTAX ERROR. Du kan også få 12 forskellige fejlmeldinger f.eks. UV IN 30 udefineret variabel i linie 30. Bagest i denne bit GUIDE vil du finde en fortegnelse over alle tilbagemeldinger.	UV IN 30
2-A-9 DELETE (slette)	Ved at holde SHIFT nede og taste DELETE vil CUSOREN bevæge sig til venstre, samtidig med at den sletter alt hvad den passerer hen over.	

2-A	INDLEDNING	RESULTAT
2-A-10 OP og NED pile	OP og NED pile kan bruges til at flytte den LINIESTJERNE, der står efter linienummer på den linie, der sidst er ændret ind i programmet.	
2-A-11 HØJRE og VENSTRE pile	De vandrette pile bruges også til at flytte CURSOR i nederste linie, det er meget brugt, når du vil rette fejl, da de kan flytte CURSOR til højre for det, du vil have slettet.	
2-A-12 Nyt LINE NO.	Hvis du skriver et nyt linienummer uden noget andet, vil den slette den linie, der havde dette nummer i programmet. Du kan også ændre linienummer på en linie, du har kaldt ned med EDIT, men så skal du huske, at slette den linie du har kopieret med EDIT (bemærk at når du kalder en linie ned med EDIT, så laver du en kopi på nederste linie, uden at slette den, der står i programmet).	
2-A-13 BREAK (bryde)	SHIFT BREAK er en kommando, der bruges hvis du vil bryde ind i et program eller, hvis der er fejl i programmet, så den ikke vil standse. Du kan starte med CONT ENTER. Den kan også bruges til at standse under SAVE og LOAD, der er de kommandoer, der bruges til at lave en båndkopi (saving) og til at sende et program fra båndoptageren ind i computerens RAM hukommelse. (LOADING).	
2-A-14 NOBEEP (ingen lyd)	Tast NOBEEP og send det ind med ENTER, lyden vil nu forsvinde fra tastaturet. Tast BEEP og ENTER og lyden kommer igen.	
2-A-15 Nul 0	Nul har på computoren en skræstreg 0, brug altid dette i linienummer og hvor det er tal, der har en talmassig værdi. 0 kan også bruges til tekstvariabler.	0
2-A-16 RESET (slette)	Et kort tryk på RESET vil slette den linie, der står uden for programmet på nederste linie, kun et kort tryk.	
2-A-17 NEW (ny)	Hvis du taster NEW og sender det ind i dit program med ENTER, vil alt hvad der er i computerens hukommelse straks være slettet. Denne kommando bruger man, når man starter på et nyt program.	

FAA	INDLEDNING	RESULTAT
FAA10 AFBRYD STRØMMEN	Nogle gange kan du ikke komme ud af dit program, i disse tilfælde er der kun et at gøre - afbryd strømmen - , det virker altid, men så er alt hvad du havde i hukommelsen (RAM) slettet.	
FAA10 SHIFT (skift)	For at skifte til de funktioner, der står skrevet på tastaturets plade, skal SHIFT holdes nede, mens du taster.	
FAA10 CURSOR (hvid kvadrat)	Nu er din computer startet op og CURSOR (i den hvide kvadrat) står og blinker nede i venstre hjørne, som tegn på at alt er klar til at skrive den første programlinje.	CURSOR blinker

2-B	INDTAST DEN FØRSTE PROGRAMLINJE	RESULTAT
2-B-1 SHIFT (skift) LINE NO (automatisk linienummer)	Hold SHIFT nede, mens du trykker LINE NO (som du ser skifter SHIFT tastens betydning til det, der står på pladen). Ved at anvende denne kommando vil linienummeret fremover automatisk springe 10 linier 10 frem. Dette er meget praktisk, idet der således altid vil være plads til nogle ekstra programlinier. Du kan også selv indtaste linienummeret med et hvilket som helst tal fra 0 til 9999	
Antal linienr. 0-9999		
2-B-2 PRINT (skriv)	Hold SHIFT nede imens du trykker PRINT, PRINT kan også indtastes bogstav for bogstav, ligesom du også selv kan indtaste alle andre kommandoer.	10PRINT
2-B-3	Nu indtaster du tallene 12345	10 PRINT 12345
2-B-4	Nu er din programlinje færdig, og du mangler kun at sende den ind i den hukommelse, der holder styr på alle dine programlinier.	
2-B-5 ENTER (send ind)	Dette gøres med tasten ENTER. Denne taste bruges, når du sender det, du har indtastet, ind i dit program.	
2-B-6	Nu står programlinjen foroven på skærmen. Læg mærke til at computoren selv sætter mellemrum (SPACE) i programlinjen.	10 PRINT 12345
2-B-7 CURSOR (hvid blinkende kvadrat) SYNTAX ERROR (skriv og stav ve fejl)	Nede i venstre hjørne blinker CURSOR. Dette betyder, at programlinjen er modtaget uden fejl. Hvis der var fejl SYNTAX ERROR ville CURSOR (den hvide blinkende kvadrat) blinke med et E, for at markere, hvor fejlen er i programlinjen.	

2-D	GÅT PROGRAMMET IGANG	RESULTAT
2-D-1 F001 RUN (startar dit program)	Tryk RUN uden linienummer og send kommandoen ved brug af tasten ENTER Du har nu kørt dit første program og fået resultatet printet ud på skærmen.	12345
2-D-2 ENTER	Tryk ENTER og dit program kommer tilbage i listeform klar til at bygge videre på.	LØPRINT12345

2-D	BYG VIDERE PÅ DIT PROGRAM	RESULTAT
2-D-1 LINE NO PRINT (skriv)	Hold SHIFT, tryk LINE NO (eller du kan selv skrive 20) Hold SHIFT, tryk PRINT, indtast 6789 og send det ind i programmet med ENTER.	LØ PRINT 12345 20xPRINT 6789
2-D-2	LINIESTJERNEN (x) står nu på den sidste indtastede linie, hvilket betyder, at linie 20 står ved indgangen til programmet.	
2-D-3	Hvis du har flyttet LINIESTJERNEN, og du benytter LINE NO, vil den næste linie springe 10 linier frem, fra det sted hvor LINIESTJERNEN står.	

2-E	BETJENING	RESULTAT
2-E-11 LINE NO, PRINT	Gentag øvelserne LINE NO og PRINT og husk at holde SHIFT nede hvergang, du skal bruge de funktioner, der er skrevet på næste pladen	
2-E-12 OP og NED pile	Gentag også øvelserne med OP og NED pile, der bruges for at flytte LINIESTJERNEN. Kald programlinien tilbage med EDIT og send den tilbage med ENTER VI VIL IKKE LÆNGERE GENTAGE, HVORNÅR DU SKAL BRUGE ENTER, DU SKULLE GERNE VÆRE KLAR OVER AT ENTER SKAL BRUGES HVERGANG COMPUTOREN SKAL UDFØRE DIN KOMMANDO (eller se pkt. 2-A-4 og 2-B-5.	
2-E-13 HØJRE og VENSTRE pile DELETE	Gentag øvelserne med HØJRE og VENSTRE pile og med DELETE, så du er helt fortrolig med at dirigere CURSOR derhen, hvor du vil indtaste DELETE (slette)	
2-E-14 SYNTAX ERROR	Skrive og stavfejl markeres ved hjælp af CURSOR med et E i, netop på det sted i linien, hvor fejlen er. Samtidig med den blinkende CURSOR med E i, fremkommer også en skarp brummen. Kør CURSOR derhen, hvor fejlen er (2-E-6 og 2-E-10) og ret fejlen.	
2-E-15 TEST	Tast altid NEW og ENTER inden du begynder på et nyt program. Indtast nedenstående program og foretag de foreslåede ændringer. LØPRINT24 Lav linie 10 til 200- slet sidste tal af 24 2ØPRINT56 Lav linie 20 til 500- slet sidste tal af 56 3ØPRINT48 Lav linie 30 til 400- slet sidste tal af 48 4ØPRINT39 Lav linie 40 til 300- slet sidste tal af 39 5ØPRINT10 Lav linie 50 til 100- slet sidste tal af 10 Slet linierne 10-20-30-40-50 Tast RUN og send kommandoen med ENTER. Resultat: 1 2 3 4 5 Bestod du testen, så gå videre med næste afsnit	

2-E	BETJENING	RESULTAT
2-E-1 Betjeningsstaster	Til betjening af computoren bruges funktioner: LINE NO, pile, EDIT, DELETE, BREAK, SHIFT, GRAPHICS, ENTER, RESET.	
2-E-2 EDIT (kald tilbage)	Tryk SHIFT EDIT og din programlinje trækkes ned. Nu har du kaldt linie 20 tilbage.	10 PRINT 12345 2ØPRINT6789
2-E-3 HØJRE PIL	Hold SHIFT og kør med HØJRE PIL indtil CURSOR står til højre for 6789.	2ØPRINT6789
2-E-4 DELETE (slet)	Hold SHIFT og tryk DELETE indtil kun 6'tallet er tilbage, tryk ENTER.	2ØPRINT6
2-E-5 HØJRE, VENSTRE PILENE	Venstre og højre pile bruges til at flytte CURSOR fra side til side, hen til det sted, hvor du vil rette eller indskrive noget mere i en programlinje. Du kan f.eks. holde SHIFT og køre med HØJRE og VENSTRE pile, så du selv kan bestemme hvor CURSOR skal stå uden at slette noget, tryk ENTER.	1ØPRINT1x2345
2-E-6 OP og NED pile	Hold SHIFT og kør skiftevis med OP og NED pile. Du kan nu se hvordan LINIESTJERNEN (x) flytter sig op og ned. På denne måde kan du flytte LINIESTJERNEN til den linie, du vil EDITERE (kald tilbage) tryk EDIT og linien vil blive kopieret i nederste linie, uden at slette den, der står i programmet, dette sker først, når du sender den tilbage igen.	LØxPRINT 12345 20 PRINT 6
2-E-7 LINE NO (se 2-B-1)	Hold SHIFT og tryk LINE NO. Du vil se at den springer 10 numre frem fra den linie, hvor LINIESTJERNEN, befinder sig.	10 PRINT 12345 20 PRINT 6 30
2-E-8 PRINT (aktiv)	Hold SHIFT og tryk PRINT eller skriv PRINT selv. Tryk ENTER.	3ØPRINT
2-E-9 EDIT (kald tilbage)	Hold SHIFT og kør med OP og NED pile for at placere LINIESTJERNEN ud for den linie, du vil kalde ud af programmet med SHIFT EDIT og ned på nederste linie	1Ø PRINT12345 20 PRINT6 3Ø PRINT
2-E-10 DELETE (slette)	Brug SHIFT HØJRE og VENSTRE pile til at flytte CURSOR (den hvide blinkende kvadrat) således, at den står til højre for det, du vil have slettet. Hold SHIFT og tryk DELETE.	1Ø PRINT 12345

(Du sletter et linienummer ved at sende samme linienummer ind uden kommando).
Eks.:
Du vil slette linie 10 - tast 10 og send ind med ENTER

PRINT MED TEKST

RESULTAT

3-A

3-A-1
 PRINT
 (tekststreng i
 anførselstegn)

Alle karakterer på tastaturet (ikke kommandoer, som f.eks. PIL, ENTER osv.) kan printes ind på skærmen ved brug af kommandoen PRINT.
 Drejer det sig om en tekst (også kaldet en tekststreng), der skal fremkomme på skærmen, når du kører dit program, skal dette indtastes i anførselstegn. Hold SHIFT og tast 5.
 Beregninger og variabler skal ikke i anførselstegn.

SEKTION 3

- 3-A-1 PRINT (tekststreng)
 3-A-2 TEKST
 3-A-3 Øvelse
 3-A-4 Print med semikolon
 3-A-5 Øvelse
 3-A-6 Komma
 3-A-7 Øvelse
 3-A-8 Punktum
 3-A-9 Print på samme linie
 3-B-1 Variabler/talvariabler
 3-B-2 LET
 3-B-3 Øvelse
 3-B-4 Udefineret variabel
 3-B-5 Øvelse
 3-B-6 Let talvariabel
 3-B-7 Definition
 3-C-1 Tekstvariabel
 3-C-2 Øvelse
 3-C-3 LET tekstvariabel øvelse
 3-C-4 Definition

3-A-2
 TEKST
 (tekststreng)
 SPACE
 (mellemlrum)

Den tekst, der skrives i anførselstegn har ingen indflydelse på dit program, men fremkommer kun som tekst.
 Hvis du ønsker mellemlrum i den tekststreng, der står mellem anførselstegnene, så skal du anvende mellemlrumstangenten (SPACE), mellemlrummene skal altid anføres indenfor anførselstegnene, hvis de skal give afstand i en tekststreng.

3-A-3
 ØVELSE
 (tekststreng)

Lav denne øvelse:
 Tast NEW - dette renser hukommelsen.
 LØPRINT"alt hvad der står på tastaturet kan du skrive imellem to anførselstegn"

3-A-4
 PRINT med
 SEMIKOLON

Semikolon bruges i en printsætning til at adskille tal, tekststreng og/eller variabel-er.
 SEMIKOLON ; virker ikke som en kommando, men kun som en besked om at skrive videre på samme linie.

3-A-5
 ØVELSE

LØPRINT " (semikolon) kan kun virke som PRINT efter en afbrudt PRINT kommando";2
 Hvis du ikke har ; før 2, så vil computeren melde E fejl.

3-A-6
 KOMMA
 (PRINT)

, (komma) kan give besked om at skrive efter en PRINT kommando. Dette bevirker at teksten flytter 16 karakterer frem, således at der bliver 2 kolonner.

3-A-7
 ØVELSE

Indtast følgende øvelse.
 Tast NEW
 LØPRINT1234,5678
 ØPRINT"abcd", "efgh"
 RUN
 ENTER

1234
abcd
5678
efgh

LØPRINT"E2

3-A PRINT MED TEKST RESULTAT

3-A-0 PUNKTUM. anvendes istedet for komma, når du skriver decimalbrøker, og indenfor anførselstegn " ".

3-A-9 PRINT på samme linie
PRINTkommandoen starter på en ny linie hver gang, men hvis du afslutter en PRINTsætning med et SEMIKOLON, vil næste PRINTkommando blive udført på denne programlinie.

3-B VARIABLER (tal- og tekstvariabler) RESULTAT

3-B-1 VARIABLER
TALVARIABLER (indeholder en værdi i tal)

Variabler bruger man til at opbevare en varierende værdi.
Talvariabler bruges til at opbevare en varierende talværdi.
En variabel består altid af et navn, der kan indeholde en værdi.
En talvariabel kan du selv bestemme navnet på. Det eneste der forlanges er, at den altid begynder med et bogstav, til gengæld kan den indeholde alle de tal og bogstaver, du har brug for.

3-B-2 LET
(lade noget være= noget andet)

En LET kommando, er i stand til at ændre værdier, når blot du giver den besked på det. Den gør det ved at flytte værdien, der står på højre side af = til det der står på venstre.

LET A=500

3-B-3 ØVELSE

LØLET A=500
2ØPRINT A
RUN
ENTER

500

OK IN 2Ø

3-B-4 UDEFINERET VARIABEL

Som du kan se er A efter PRINT ikke i "", det er fordi, det er en talvariabel, der ikke skal i anførselstegn.
Hvis du f.eks indtaster PRINT DETTE ER EN TEKST og undlader at bruge anførselstegn, vil computeren betragte det som en talvariabel, som du skal give en værdi ellers melder den udefineret variabel.

UV IN 1Ø

3-B-5 ØVELSE

1Ø LET DETTE ER EN TEKST=500ØØØ
2Ø PRINT DETTE ER EN TEKST
RUN

500ØØØ

OK IN 2Ø

3-B-6 LET TALVARIABLE
ØVELSE

En meget anvendelig måde, er at kombinere tekst- og talvariabler.

LØLET Aviser=200
2ØPRINT "SALGSPRISEN ER ";AVISER;" KR"
RUN

SALGSPRISEN ER 200KR.
OK IN 2Ø

3-B-7 DEFINITION

TALVARIABLER skal altid begynde med et bogstav og have sin værdi i tal.

LET AL2B=22222

TEKSTVARIABLER

RESULTAT

3-C	TEKSTVARIABLER	RESULTAT
3-C-1 TEKSTVARIABLEL (indeholder en værdi af tekst)	En tekstvariabel efterfølges altid af et \$ (dollar tegn). Computoren har 26 tekstvariable fra A\$ til Z\$. I forbindelse med en LET kommando, skal tekstvariablen altid have sin værdi i "" (anførselstegn)	
3-C-2 ØVELSE	LØLET A\$="et navn" 2ØPRINT A\$ RUN	et navn OK IN 2Ø
3-C-3 LET TEKSTVARIABLEL ØVELSE	En tekstvariabel kan også indsættes efter (semikolon) i en print sætning LØLET B\$="DYR" 2ØPRINT "SALGSPRISEN ER ";B\$ RUN Husk mellemrum før sidste anførselstegn i print sætningen.	SALGSPRISEN ER DYR
3-C-4 DEFINITION	TEKSTVARIABLER skal altid begynde med et bogstav efterfulgt af et \$ (dollar tegn) ligesom værdien i form af tekst er indsat i anførselstegn "".	

SEKTION 4

4-A-1	Aritmetriske beregninger
4-A-2	Parenteser
4-A-3	Punktum
4-A-4	Relationsoperatorer
4-A-5	Eksponentiel tegnsystem
4-A-6	Ciffer forhøjes/nedrunden
4-B-1	ALU
4-B-2	Beregninger
4-B-3	Beregninger med Print
4-B-4	Addition og print
4-B-5	Addition med LET
4-B-6	Space - Semikolon - Komma
4-B-7	Punktum ved beregninger
4-B-8	Subtraktion og Print
4-C-1	Prioriteter - Parenteser
4-C-2	Multiplikation/Division
4-C-3	Øvelse
4-C-4	Øvelse med parenteser
4-C-5	LET og PRINT med talvariabel
4-C-6	Øvelse med talvariable
4-C-7	LET og PRINT talvariable øvelse
4-C-8	Prioritetsrække

4-A	INDLEDNING	RESULTAT
4-A-1	<p>Inden du begynder at lave programlinier med beregninger, er der noget du skal være opmærksom på.</p> <p>I Basic program, der er tilpasset denne computer, anvender man bestemte tegn til aritmetriske beregninger, der udregnes i rækkefølge fra venstre mod højre. Først indenfor parenteserne og derefter det udenfor, i denne rækkefølge (prioritet).</p> <p>X X POTENSÖPLÖFTNING X MULTIPLIKATION og / DIVISION + ADDITION og - SUBTRAKTION</p>	
4-A-2	<p>Når du omskriver en formel fra matematisk tegnsystem til BASIC tegnsystem, bliver du nødt til at sætte det hele på en linie, og anvende parenteser.</p> <p>MATEMATISK BASIC</p> <p>$300 + \frac{4}{5}$ bliver til 300+4/5</p> <p>$\frac{300-5}{5}$ bliver til (300-5)/5</p> <p>$\frac{AB}{CD}$ bliver til (AXB)/(CXD) eller AXB/C/D</p>	<p>300.8</p> <p>59</p>
4-A-3 PUNKTUM	<p>Ved decimalbrøker skal der <u>altid</u> anvendes <u>punktum</u> istedet for komma.</p>	
4-A-4	<p>Relationsoperatorene</p> <p>> Større end først, og = lig med til sidst</p> <p>< Mindre end først, og = lig med til sidst</p>	<p>> =</p> <p>< =</p>
4-A-5 EKSPONENTIEL tegnsystem	<p>Du kan omskrive tal til titals potenser med anvendelse af bogstav E, der betyder gange 10 opløftet i potens. Computoren vil automatisk gå over til at anvende titals potenser, når den kommer til 14 cifre.</p> <p>1 0 000 000 000 000 = 1E+13 (dvs. 13 nuller efter etallet).</p> <p>Når tallet er større end nul sættes + foran</p> <p>Når tallet er mindre end nul sættes - foran</p> <p>f.eks.: 0.000 000 000 0001 = 1E-13, dvs. det 13ende ciffr efter punktum.</p>	
4-A	<p>INDLEDNING</p>	<p>RESULTAT</p>
4-A-6	<p>Computoren kan højst udskrive 8 cifre. Er der flere cifre, runder den af til nul. Det 9ende ciffr forhøjes til 10, hvis det er 5 eller derover, er 5 og derunder nedrundes til nul (0).</p> <p>B E M Æ R K !</p> <p>x tegn (MULTIPLIKATION) er vist som en stjerne på computeren (*).</p>	

SEKTION 4		SEKTION 4		side 18
COMPUTOREN SOM REGNEMASKINE		PRIORITETS RÆKKEFØLGE		RESULTAT
4-B-1 ALU regneenhed	Inde i computoren er der en meget skrap regneenhed (ALU=arithmetic logic unit), der indeholder de matematiske funktioner.	4-C-1 PRIORITETER	Når computoren udfører et program sker det på samme måde, som når du læser, fra venstre mod højre, oppe fra og ned, linie for linie, indtil der ikke er flere linier eller noget, der afbryder.	
4-B-2 BEREGNINGER	Der kan anvende 2 kommandoer til beregninger. 1. PRINT 100+200 eller 2. LET A = 100+200	PARENTESER	I nogle tilfælde springer computoren dog frem og tilbage på samme programlinie. Dette kaldes en PRIORITETS RÆKKEFØLGE dvs. at nogle funktioner og beregninger altid foretages i en bestemt rækkefølge, uanset hvor de er placeret, men kun indenfor samme programlinie. Kun ved brug af PARENTESER kan denne regel ophæves.	
4-B-3 BEREGNINGER med PRINT	På en programlinie med PRINT kommandoen, kan du få foretaget en beregning, samtidig med, at du får det printet ud.	4-C-2 MULTIPLIKATIONX DIVISION/	Ligesom i almindelig matematik udregnes alle multiplikations- (x-gange) og divisions (/=dele) beregninger før subtraktion (-trække fra) og addition (+lægge til)	
4-B-4 ADDITION+ og PRINT	10 PRINT 600+7000+550 20 PRINT 200+800 RUN	4-C-3 ØVELSE	10 PRINT 2X4/2+4X5 2X4 8/2 4 4X5 4 + 20 = 24	24 OK IN 10
4-B-5 ADDITION+ med LET	10 LET A1= 600+7000+550 20 LET A2= 200+800 30 PRINT A1 40 PRINT A2 RUN	4-C-4 ØVELSE med parenteser	10 PRINT 2X4/(2+4)X5 2+4 2X4 6 8/ 6 1.3333333X5 6.666	6.6666666 OK IN 10
4-B-6 SPACE(mellemrum) SEMIKOLON	Du kan også få talvariablerne A1 og A2 printet ud på en linie, men så er du nødt til at indsatte mellemrum mellem A1 og A2. 30 PRINT A1"SPACE";A2 40 tryk ENTER (derved slettes linie 40) RUN	4-C-5 LET og PRINT med TALVARIABLE	Som du kan se springer den over + og ganger først. Du kan se, bliver det i parentes udført først, som om det var en programlinie for sig selv, dvs. at samme prioritetrækkefølge også gælder indenfor parenteser.	
KOMMA	Du kan også anvende komma, derved vil du få tallene i 2 kolonner. 30 PRINT A1,A2	4-C-6 ØVELSE med TALVARIABLE	10 LET AB = 200 20 PRINT 4+8X100/AB RUN	8 OK IN 20
4-B-7 PUNKTUM ved beregninger	Når du skriver decimalbrøker i beregninger, må du <u>aldrig</u> bruge komma. <u>BRUG ALTID PUNKTUM</u> f.eks. 50.55		Som du kan bliv værdien 200 overført til AB i linie 10	
4-B-8 SUBTRAKTION og PRINT	På samme måde som du bruger ADDITION, kan du også anvende SUBTRAKTION. Tast NEW ENTER inden du starter. 10 PRINT 500-200-100 20 PRINT 100-200 RUN Som du kan se sætter computoren - foran tallet, hvis det er negativt.			

4-C	PRIORITETS RÆKKEFØLGE	RESULTAT
10 LET AB=5 20 LET CD=10 30 LET EF=20 40 LET A =EF+EF/CD*AB/(EF*XCD)+(EF*AB) 50 PRINT RUN	Talvariabler kan anvendes de steder, hvor der kan bruges tal, men husk at give variablen en værdi. Før den kommer til den programlinje, hvori variablen anvendes, eller vil følgende fremkomme på skærmen UV IN linienummer. (Udefineret variabel).	120.05 OK IN 50 UV IN linienr.
4-C-8 PRIORITETS RÆKKE	PRIORITETS RÆKKEN er den rækkefølge computoren udfører sine funktioner og beregninger i. Computoren gennemgår programlinjen fra venstre mod højre. Først udfører den alle funktionerne, hvor tekststrengfunktionerne har prioritet 12 og talfunktionerne prioritet 11, derefter kommer prioritetsrækkefølgen. Først de aritmetiske operationer Symbol Betydning Prioritet 10 xx Potens opløftning Prioritet 9 - Fortegnsminus Prioritet 8 +/ Multiplikation og division Prioritet 6 +- Addition og subtraktion Relationsoperatorer Relationsoperatorerne har alle prioritet 5. Symbol Betydning = Lig med > Større end < Mindre end > = Større end eller lig med < = Mindre end eller lig med < > Forskellig fra	LIST - ENTER - RUN STOP - BREAK CONT LET SLOW - FAST - PAUSE PAUSE Computorens arbejds måde Basic kommandoer Udførelse af programlinje ALGORITME - Definition Opgave Fra Rutediagram til Basicprogram Løsning af opgave Talvariabler Input Tekstvariabel LET Sammenkæde tekst Input i tekstvariabler Input Stop i Input PRINT Input CLS GOTO Bryde en GOTO løkke GOSUB RETURN SUBROUTINE
Logiske operatorer Symbol Betydning Prioritet 4 NOT(ikke) NOT relationen er sand, når relationen er falsk. NOT relationen er falsk, når relationen er sand. AND relationen er sand, når begge relationer er sande. Prioritet 2 OR(eller) OR relationen er sand når den ene af relationerne er sand.	Logiske operatorer Symbol Betydning Prioritet 4 NOT(ikke) NOT relationen er sand, når relationen er falsk. NOT relationen er falsk, når relationen er sand. AND relationen er sand, når begge relationer er sande. Prioritet 2 OR(eller) OR relationen er sand når den ene af relationerne er sand.	SEKTION 5 5-A-1 5-A-2 5-A-3 5-A-4 5-A-5 5-A-6 5-A-7 5-A-8 5-A-9 5-B-1 5-B-2 5-B-3 5-B-4 5-C-1 5-C-2 5-D-1 5-D-2 5-D-3 5-D-4 5-D-5 5-D-6 5-D-7 5-D-8 5-E-1 5-E-2 5-F-1 5-F-2

5-G-1	IF THEN		
5-G-2	Opbygning af en IF THEN sætning		
5-G-3	Udtryk		
5-G-4	Taludtryk		
5-G-5	Rutediagram		
5-G-6	IF med NOT, AND, OR		
5-G-7	IF THEN med talvariabler		
5-G-8	Tekstvariabler		
5-G-9	Eksempel	5-L-1	INT
		5-L-2	Eksempel
		5-M-1	RND
		5-M-2	RAND
		5-M-3	Øvelse
		5-M-4	Eksempel med RAND og RND
5-H-1	Opbygning af FOR sætning	5-N-1	Eksempel (del en tekststreng)
5-H-2	Stepværdi	5-N-2	SLICING
5-H-3	Kontrolvariablen		
5-H-4	Fortegns minus	5-O-1	CHR\$
5-H-5	FORNEXT	5-O-2	CODE
5-H-6	Rutediagram med 2 løkker	5-P-1	INKEY\$
		5-P-2	Øvelse
5-I-1	DIM ()	5-Q-1	CLEAR
5-I-2	DIM (dimensioner et antal)	5-Q-2	SCROLL
5-I-3	Øvelse	5-Q-3	Tekstbehandling
5-I-5	Indesererede variabler		
5-I-6	Øvelse		
5-I-7	Øvelse		
5-I-8	Øvelse		
5-I-9	Rutediagram for 3-dimensionel data		
5-I-10	DIM ()		
5-I-11	Indicerede tekstvariabler		
5-I-12	Øvelse		
5-I-13	Indicerede tekstvariabler i flere dimensioner		
5-J-1	TAB (tabulator		
5-J-2	PRINT AT		
5-J-3	Øvelse		
5-K-1	LEN til tekststreng		
5-K-2	Eksempel		
5-K-3	LEN til talkolonne - STR\$		
5-K-4	VAL		
5-K-5	Eksempel		

5-A	INDLEDNING	
5-A-1 LIST ENTER RUN	Hvis du taster LIST, kommer din programliste frem på skærmen og starter med første programlinje. Det samme resultat fremkommer, hvis du trykker ENTER, og endnu ikke har sat dit program igang med RUN. Dog skal du være opmærksom på, at i de tilfælde, hvor computoren venter på INPUT, kan ENTER ikke anvendes til at fremkalde programlisten, her skal du bruge BREAK efterfulgt af LIST eller ENTER.	Du vil kunne få skærmbilledet frem ved at indsætte en PAUSE i programmet, mens det er i FASET. Tast SLOW eller indsat SLOW i programmet, når du vil tilbage til normal hastighed. Også ved LYD kan du anvende FAST med en god effekt.
5-A-2 STOP (stands)	Hvis du vil fremkalde programlisten, begyndende med et bestemt linienummer, taster du LIST efterfulgt af linienummer.	PAUSE kommandoen standser programudførelsen i en tid, der svarer til det tal, der er indsat i PAUSE. f.eks. PAUSE 500 svarer til 10 sekunders pause
5-A-3 CONT (fortsat)	Vil du standse et program, kan du indsætte en STOP kommando i programmet. Herefter vil programafviklingen standse på dette sted og afvente en kommando fra dig, om at fortsætte. CONT ENTER	Denne PAUSE opnår den ved at tælle skærmbilleder, og da der kommer 50 skærmbilleder i sekundet, bliver PAUSE 50, det samme som et sekundt. Nu behøver du blot, at indtaste de antal sekunder, du ønsker PAUSE i med 50 efter kommandoen PAUSE i dit program
5-A-4 LET	Du kan også standse programmet, når computoren venter på INPUT, med BREAK. Når du har standset programmet med BREAK bliver computoren stående ved den programlinje, du brød ind i, imens kan du foretage ændringer i programmet, kommandere FAST og SLOW og stadigvæk fortsætte fra samme programlinje med CONT ENTER.	Den største pauselængde du kan programmere i en kommando er PAUSE 65535, der vil være ca. 22 minutter lang. Hvis du vil have længere pauser eller, hvis du vil lave et UR, kan du anvende FOR NEXT med PAUSE.
5-A-5 SLOW (langsom) FAST (hurtig)	Når du vil fortsætte efter en STOP kommando, kan du taste CONT og programafviklingen vil fortsætte fra det linienummer, hvor STOP kommandoen var indsat, samtidig vil CONT rydde skærmen. CONT kan anvendes, når S F (skærm flydt). CONT vil ligesom alle andre kommandoer rense skærmen, samtidigt med, at den vil få programmet til at fortsætte.	En computer er en binær maskine, dvs. at den arbejder med binære signaler, der kun består af 2 spændingsniveauer, der giver signal 0 ved ingen spænding og 1 ved spænding. Dette indebærer, at computoren kun forstår de 2 signaler 0 og 1.
5-A-6	I forbindelse med talvariabler (5 - C-1) anvendes LET kommandoen, for at give talvariablen sin værdi i form af tal. Du behøver ikke anvende LET. Computoren forstår udmærket en LET sætning uden LET, men i starten bør du nok anvende dette for overskuelighedens skyld. 10 10 LET B=222 er det samme som 10 B=222	I denne computer er der indbygget en BASIC oversætter, der oversætter til binære signaler (maskinsprog). Denne version af BASIC (basic kan afvige mere eller mindre fra computer til computer) er et meget enkelt sprog, der indeholder følgende: 34 Kommandoord 23 Funktioner 26 Bogstaver (a-z) 10 tal (1234567890) Specialtegn +(-), osv Grafiktegn Fejl og OK meldekoder FUNKTIONStaste ENTER SHIFT DELETE RESET EDIT BREAK

INDLEDNING

5-A-8 Disse BASIC kommandoer, funktioner osv. er din kontakt med computeren. Det er ud fra dette, du kan opbygge programmer, som computeren vil kunne arbejde efter.

Et program er som en lang remse, som computeren læser linie for linie. Den begynder at søge ved linie 0 og standser ved det første linienummer, hvor der er indprogrammeret noget.

5-A-9 Når computeren kommer til en programlinje, udfører den programlinjen fra venstre til højre, begyndende med den højeste prioritet (se sektion 4). Derefter søger computeren den næste programlinje, der er indskrevet noget på, og kun hvis den har fået en kommando som f.eks STOP GOTO GOSUB NEXT RETURN i linien foran, brydes linienumrenes rækkefølge, der går fra 0 til 9999.

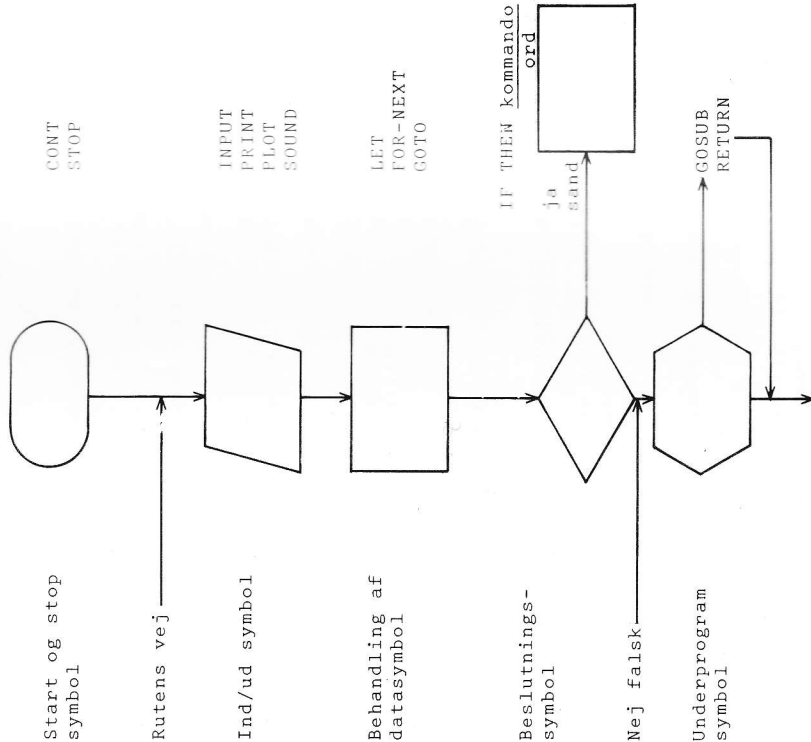
OPBYGNING AF ET PROGRAM

5-B-1 Når du vil løse en opgave eller et problem, ved hjælp af din computer, må du først gennemtænke en fremgangsmåde, der i den sidste ende vil føre til en løsning af opgaven eller problemet. Dette kaldes en ALGORITME, der er en utvetydig beskrivelse af en fremgangsmåde, der fører til en løsning.

5-B-2 Du har problemer med dine lomme penge, der slipper op sigst på måneden. Du får 3000.00kr om året, hvor meget kan du tillade dig at bruge hver dag.

ALGORITME I 1983 får du 3000.00kr, som du skal dele med det antal dage, året er på = 365 dage, så du har lige meget hver dag året rundt.

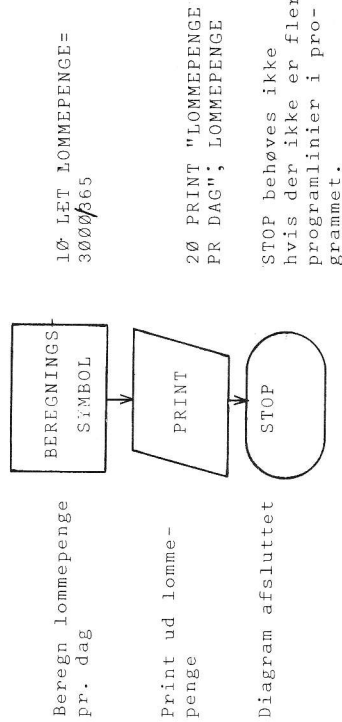
RUTEDIAGRAM En ALGORITME omskrives til et rutediagram, der er en grafisk fremstilling af en af problemet eller af opgaven. Til denne fremstilling anvendes nogle symboler, der er forbundet i den rækkefølge problemet eller opgaven løses.



5-B OPBYGNING AF PROGRAM

Med disse 5 symboler kan du ved hjælp af et Rutediagram få et bedre overblik over det problem eller opgave du vil løse. Vi vender tilbage til opgave 5-B-2 som vi omskrev til ALGORITME, det vil vi lave et RUTEDIAGRAM over, som vi vil omskrive til et BASICPROGRAM.

5-B-3 Fra RUTEDIAGRAM til BASICPROGRAM



Nu har vi løst vores opgave på en systematisk måde i følgende punkter.

DEFINITION:

Punkt 1. Du må klart og præcis vide hvilket problem eller opgave, hvilken data det skal indeholde, samt hvilket resultat du vil opnå.

ALGORITME:

Punkt 2. Du må gennemtenke en løsningsmodel som ligger inden for din Computers muligheder

RUTEDIAGRAM:

Punkt 3 Tegn et rutediagram der fra start til stop der gennemfører din ALGORITME.

PROGRAM:

Punkt 4. Omskriv dit Rutediagram til et Basicprogram

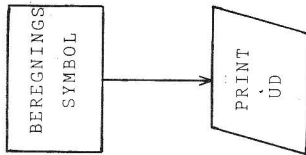
FEJLFINDING:

Punkt 5. Til fejlfinding bruger du dit Rutediagram hvor du følger rutensvej fra start til stop

5-C TALVARIABEL

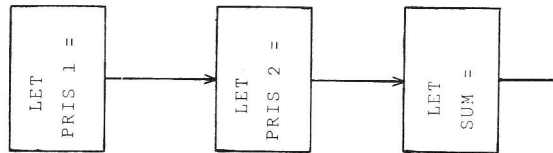
5-C-1

LET kan bruges i forbindelse med Talvariabler til at tildele Talvariablen sin værdi i form af tal. Dette symbol udfører i dette tilfælde først en beregning 3000/365 derefter overfører det værdien af beregningen til LOMMEPENGE der nu er blevet til en TALVARIABEL



Nu er printkommandoen meget enkel, først beder vi den printe en tekst ud, det gør vi ved at sætte teksten i anførelsestegn """, Derefter beder vi den om at printe på samme linie med semikolon ; og til sidst LOMMEPENGE uden anførelsestegn (som uden anførelsestegn er navnet på en Talvariabel) hvis værdi udskrives.

En TALVARIABEL kan indeholde en variable værdi, den består altid af et navn som eks. kan være de første bogstaver, f.eks. sum for summen, pris for prisen o.s.v., således at du let kan se hvad dens værdi står for. Derefter et lighedstegn = og så den værdi den skal have.



```

10 LET PRIS 1 = 500
20 LET PRIS 2 = 100
30 LET SUM = PRIS 1 + PRIS 2
40 PRINT "SUMMEN AF PRIS 1 OG
2 = "; SUM
  
```

Husk den første karakter i en TALVARIABEL skal altid være et bogstav. Se 3-B-1. Husk også at sætte mellemrum i Print-sætningerne inden for anførelsestegnene.

5-C

INPUT TALVARIABLEN

5-C-2

Hvis du undlader at tildele en af VARIABLENE en VÆRDI får du fejlmelding, UV IN og det linie nr, hvori den Udefineret Variabel befinder sig men det sker først under RUN når Computeren kommer til den Programlinie, så bare værdien er tildelt i Programlinien før er det OK.

Navne på talvariable, må ikke, indeholde sammensætning af bogstaver, der kan forveksles med kommando eller dunktionsord.

F. eks:
NAT= eller BORD

Nu er det ikke særlig praktisk at have et program som indeholder en fast værdi for lompepenge, for hvis nu alle kammeraterne kommer og de vil prøve programmet så skal du lave LET sætningen om hver gang, nej vi går systematisk frem og spørger efter hvor mange lompepenge får du om året, den næste kommando er så en INPUT sætning som står og venter på at du skal give den svaret.

```

10 PRINT "HVOR MANGE LOMPEPENGE FAAR
DU OM AARET"
20 INPUT LOMPEPENGE
30 LET LOMPEPENGE = LOMPEPENGE/3 65
40 PRINT "LOMPEPENGE PR DAG";
LOMPEPENGE

```

INPUT kommandoen indskrives i Programmet så du kan give Computeren data underprogramkørslen.

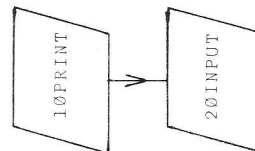
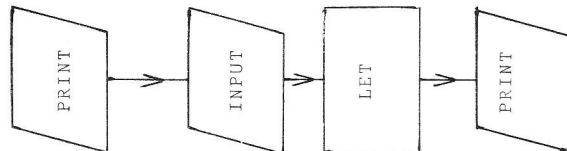
I INPUT sætning kan du kun indputte en Variabel, så hvis du har flere må du lave lige så mange INPUT sætninger.

Her er et eks. i linie 50 har vi en CLS kommando (Clear Screen) den fjerner det der står på skærmen, i dette program er det spørgsmålene i PRINT sætningen 10 og 30 dem har vi ikke mere at bruge til på skærmen når vi har besvaret INPUT sætningen.

```

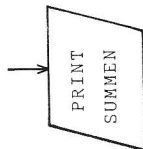
10 PRINT "HVLKET AARSTAL ER DU FØDT
I"
20 INPUT A
30 PRINT "HVLKET AARSTAL HAR VI I
AAR"
40 INPUT B
50 CLS
60 PRINT "DU ER FØDT I "; A
70 LET C = B - A
80 PRINT "DU ER/BLIVER "; C;" AAR
I AAR"
90 LET D = 100 - C
100 PRINT " DU VIL FYLDE 100 AAR
OM "; D;" AAR"

```



5-C

TALVARIABLE



5-C

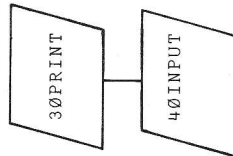
INPUT TALVARIABLEN

5-C-2

```

110 LET E = 2000 - A
120 PRINT "I AAR 2000 VIL DU FYLDE
";E;" AAR"

```



Som du kan se i programmet skal der en printsatning før en inputsætning for at indformere om hvad du skal indtaste. Også efter Input kan det være praktisk at have en PRINTsatning hvor den VARIAB-LES værdi som du lige har indtastet fremkommer.

Eks. med print på samme linie.

```

10 PRINT "HVILKET AAR ER DET I AAR. SVAR "
20 INPUT A
30 PRINT A

```

SEKTION 5

side 28a

5-D

TEKSTVARIABLER

5-D-1

TEKSTVARIABLEL

Computoren har 26 tekstvariable fra A# til Z#, der er navnet på variablen.

En tekstvariabels navn består altid af et bogstav efterfulgt af et dollartegn (\$) A# B# C# osv.,

En tekstvariabels værdi skal altid indskrives i anførselstegn ""

En tekstvariabel kan du tildele en tekstværdi i henhold til den værdi, der fremgår af karakterkoden i Appendix C, startende med kode 0 for SPACE værdien for tegn og tal, men det er især bogstaverne, der har betydning, hvis vi skal sammenligne tekst, da vi har værdien i alfabetisk rækkefølge med A som det mindste og Z som det bogstav, der har den største værdi.

ALFANUMERISK
VÆRDI

5-D-2

LET

LET kommandoen kan bruges i forbindelse med tekstvariable til at tildele tekstvariablen sin værdi i form af tekst.

```

10 LET A$ = "TEKST"
20 PRINT A$
RUN

```

5-D-3

SAMMANKÆDE

Du kan også kæde en tekst sammen med + men du kan ikke anvende - / x

```

10 LET A$ = "TEKST" + "STRENG"
20 PRINT A$
RUN

```

RUTEDIAGRAM

På samme måde som du brugte talvariable i forbindelse med rutediagram, kan du bruge tekstvariable.

INPUT TEKSTVARIABEL

5-E

GOTO

5-D-4
INPUT I Tekst
variabler

Udførelse af en tekst INPUT kommando bevirker at Computeren standser under programkørslen og står og blinker med CURSOR indsat i anførselstegn "" Dette betyder at den er standset ved en Tekstvariabel og venter på at du skal indtaste noget som Computeren selv sætter i anførselstegn og derved bliver det automatisk til tekst.

5-D-5
INPUT

Du kan bruge to INPUT sætninger til at sætte ord sammen, eks. sommer og fugl

```
10 PRINT "INPUT FØRSTE DEL AF ET ORD"
20 INPUT A$
22 PRINT "INPUT ANDEN DEL AF ET ORD"
  25 INPUT B$
30 PRINT A$ + B$
```

Du kan enten bruge + eller ; semikolon.

5-D-6
STOP I INPUT

Hvis du i INPUT sætning vil ud af programmet kan du bruge BREAK.
Start igen med CONT-ENTER.

5-D-7
PRINT INPUT

Ligesom med Talvariabler kan du også før en Tekstvariabel CLS (rensér skærm) efter INPUT og derefter en PRINT sætning der viser hvad du har indtastet.

```
10 PRINT "NAVNET PAA EN BY"
20 INPUT A$
30 CLS
40 PRINT A$
```

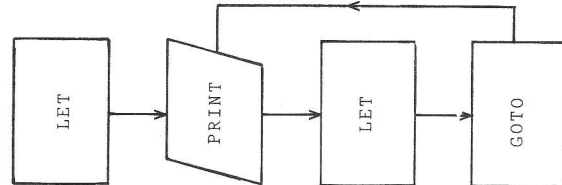
5-D-8
CLS

Som du kan se, så rensér CLS kun skærmen (det så kaldte skærmregister) uden at slette variablerne.

5-E-1

Nu har vi lavet programmer med PRINT LET og INPUT der følger linienummer ordene.
Nu vil vi bruge en GOTO (Gå til linie nr.) som eftergølgtes med det nr. som den skal springe op eller ned til.

Med denne kommando får du mange muligheder men samtidigt bliver du også nød til at holde kontrol med hvad der sker når du springer rundt i linienummerne, især de store spring kan let blive uoverskuelige. Hvis du vil lave et større program så vil Rutedia-grammet være en hjælp til at undgå problemer.



```
10 LET A=0
```

```
20 PRINT A;
```

```
30 LET A = A+1
```

```
40 GOTO 20
```

RUN

Som du sikkert har gættet så kører programmet i en GOTO løkke og hver gang den kommer til linie 30 for-højer den A + 1 og går tilbage til linie 20 Printer A hver gang.
Semikolonet efter PRINT A; bevirker at den printer på samme linie.

5-E-2

Du bliver sikkert træt af programmet inden den har printet skærmen fuld, så kan du bryde en GOTO løkke tast BREAK og med ENTER kommer list igen.

GOSUB RETURN

5-F-1
GOSUB-RETURN

Disse problemer vi havde med GOTO der uden kontrol kunne køre rundt i programmet uden vi var klar over hvad der skete, dem har vi ikke med GOSUB-RETURN (gå til et underprogram som afslutter med retur til næste linie efter GOSUB i hovedprogrammet) der sendte den i underprogrammet.

GOSUB bruges især til Gengrugsprogramstykker. Det er mest praktisk at give GOSUB et større linie nr. f.eks 1000 - 2000 for at komme væk fra hovedprogrammet, men husk at sætte STOP efter hovedprogrammet, ellers fortsætter programafviklingen ned i underprogrammerne, og dette vil normalt ikke være heldigt.

GOSUB stækker (en hukommelse der husker hvilket linie nr. der sendte GOSUB og lægger en til). Når programafviklingen møder en RETURN kommando, vil den vende tilbage til det næste linie nr. efter den GOSUB i hovedprogrammet som sendte den afsted.

5-F-2
SUBROUTINE

Et program med pause så du kan følge programafviklingen, de tomme PRINT er til at give afstand.

Du kan stoppe PAUSEN med ENTER

En praktisk ting at gøre er, at sætte anførselstegnet helt ud til højre med space, så kan du Printe teksten som den vil stå på skærmen når du kører RUN.

```

REM          10 REM DETTE ER ET PROGRAM MED
              SOUBROUTINE (UNDERPROGRAM) SAMT
              PAUSER
              20 PRINT
              "
              NU STARTER VI I HOVERPROGRAMMET "
              DEREFTER EN PAUSE 500 = IØ SEK
              PAUSE (1 SEK = PAUSE 50) OG MED GOSUB
              FORLADER VI HOVEDPROGRAMMET."
              30 PAUSE 500
              35 PRINT
              GOSUB 40 GOSUB 1000
              50 PRINT
              PRINT "
              NU ER VI TILBAGE I HOVEDPROGRAM
              ET. HUST ALTID AT SÆTTE STOP
              EFTER HOVEDPROGRAMMET."
              STOP 60 STOP
              100 PRINT
              "
              NU ER VI NEDE I SUBROUTINEM.
              INGEN PAUSE DEREFTER EN RETURN
              PRINT (RETURN) TIL HOVEDPROGRAMMET"
              PAUSE 1010 PAUSE 500
              1015 PRINT
              1020 RETURN
              RETURN

```

GOSUB RETURN

5-F-2

Normalt vil man have flere GOSUB til samme SUBROUTINE således at man anvender SUBROUTINE flere gange.

En anden anvendelse af SUBROUTINE er i forbindelse med IF sætningen den vender vi tilbage til under IF.

Du kan stoppe PAUSEN med ENTER.

En praktisk ting at gøre er at sætte anførselstegnet helt ud til højre med SPACE. så kan du Printe teksten, som den vil stå på skærmen, når du kører RUN.

IF THEN

5-0-1

En IF THEN kommando udfører den kommando der kommer efter THEN hvis betingelsen er opfyldt. (SAND) Hvis betingelsen ikke er opfyldt fortsætter den til næste programlinje.

Denne IF THEN kommando er som en sidevej der giver os mulighed for at komme væk fra programlinjen og udføre en kommando.

For at komme ud på den sidevej stiller vi et TAL eller TEKSTUDTRYK i forhold til et (i relation) et andet TAL eller TEKSTUDTRYK som betingelse for at udføre kommandoen efter THEN.

Til at stille betingelser bruger vi Relationsoperatorene.

- < Mindre end
- > Større end
- = Lig med
- >= Større end eller lig med
- <= Mindre end eller lig med
- <> Forskellig fra

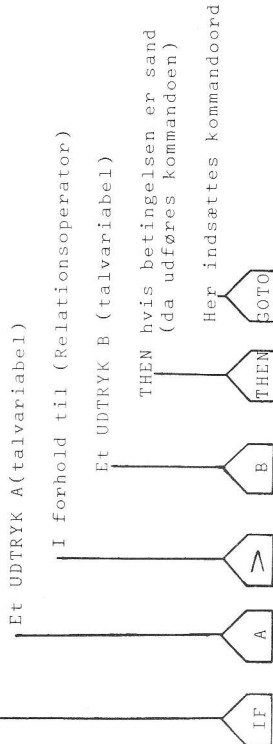
Husk altid før = og før =

Når du stiller tekst i forhold til en anden tekst, så har A den mindste og Z den største værdi. (alfanummeriskeværdi).

Du kan sammensætte en IF sætning af TAL, TEKST, VARIABLER, FUNKTIONER med beregninger i forhold (Relation) til andre TAL, TEKST, VARIABLER, FUNKTIONER med beregninger.

5-0-2
OPBYGNING
AF EN IF
THEN SÆT-
NING

IF Det ene udtryk i forhold til det andet UDTRYK



IF THEN

5-G-3
UDTRYK

1. TEKSTUDTRYK er enten Tekst eller Tekstvariabel A\$, B\$ o..v.) til denne tekst eller tekstvariabel er knyttet en værdi som stilles i forhold til en anden Tekst eller Tekstvariabels værdi. (alfanummeriskeværdi).

5-G-4
TALUDTRYK

TALUDTRYK'er enten et Tal eller Talvariabel, til denne Tal eller Talvariabel er knyttet en værdi som stilles i forhold til et andet Udtryk.

Værdien af et Udtryk kan også fremkomme af beregninger foretaget i IF sætningen på begge sider af relationsoperatoren.

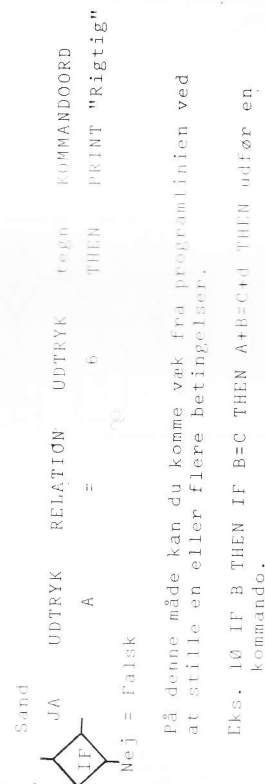
Eks. 50 IF AxB = C+2 THEN kommandoord.

Du kan også indsætte Matematiske Funktioner i beregningerne men lav en kontrol på dem inden du sætter dem ind i programmer, her til kan du lave et program der fortæller om Computeren vil akceptere opstillingen, hvis du i forvejen kender resultatet af Relationen.

```

10 REM PROGRAM TIL AT AFPRØVE
OM DINE IF SÆTNINGER ER SANDE
ELLER FALSK
20 LET A = 2
30 LET B = 6
40 LET C = 10
50 IF AxB = C + 2 THEN GOTO 80
60 PRINT "BETINGELSEN ER IKKE
OPFYLDT (FALSK)"
70 STOP
80 PRINT "BETINGELSEN ER OPFYLDT
(SAND)"
    
```

5-G-5
RUTEDIAGRAM
med IF THEN



På denne måde kan du komme væk fra programlinjen ved at stille en eller flere betingelser.

Eks. 10 IF B THEN IF B=C THEN A+B=C00 THEN udfør en kommando.

5-C

IF THEN med TEKSTVARIABLER

Du kan også i stedet for OR anvende
4 IF sætninger.

Som du sikkert kan se giver IF dig
mange muligheder.

SEKTION 5

5-H

FOR NEXT STEP

5-H-1

Opbygning af en FOR-sætning.
Som vi husker fra GOTO løkken i SEKTION 5-E-1 som
vi ikke kunne kontrollere hvor mange gange den kørte
rundt.

En FOR NEXT løkke er en løkke der ligner GOTO løkken
dog med den væsentlige forskel, at i FOR og NEXT
løkken har man indsat en tæller, samt givet den navn
i form af en kontrolvariabel.

Til kontrolvariablen må kun bruges 1 bogstav fra A
til Z.

Kontrolvariablen sættes til en start og en slut værdi
efterfulgt af en funktion STEP (trin) der fortæller
med hvor store trin den skal komme fra start værdi til
slut værdi, hvis du ikke sætter STEP efter værdien
vil den automatisk bruge 1 som STEP værdi.

Husk hvis du bruger samme bogstav til en kontrolvaria-
bel som du har anvendt til en talvariabel bliver tal-
variablen slettet.

En FOR NEXT løkke afsluttes altid med en NEXT som ef-
terfølges af KONTROLVARIABLENS bogstav.

Når kontrolvariablen har overskredet slutværdien
(grænseværdien) fortsætter den til næste linie efter
NEXT.

50 FOR = 1 TO 10 STEP 2

Linie nr.

FOR kommando.

Kontrolvariabel

= overfører værdien til kontrolvariablen

Startværdi (kan også være negativ)

TO (til)

Slutværdi (grænseværdi) kan også
være negativ

STEP (trinværdien)

tallet der giver trinværdie

50 FOR A = 1 TO 10 STEP 2

FOR NEXT STEP

FOR NEXT STEP

5-H

5-H

5-H-2
STEPVÆRDI

```
20 FOR A = 1 TO 10 STEP 2
25 PRINT A
30 NEXT A
```

Prøv at indtaste dette program. Læg mærke til at STEP plusser Startværdien med Stepværdien.

Du kan også få STEP til at trække fra ved at sætte minus - foran værdien af STEP, men så må startværdien være større end slutværdien.

5-H-3
Kontrol-
variablen

Kontrolvariablens værdi beregnes i.h.t. stepværdien hver gang FOR sætningen udføres og så snart startværdien er den samme som slutværdien (grænseværdien) fortsætter programudførelsen efter NEXT.

START

Istedet for de 3 værdier START - SLUT - STEP værdi kan der indstilles Beregninger (2+2) eller Talvariabler (A+B) eller (A+1).

SLUT

STEP

```
Eks. 10 LET D = 1
      20 LET B = 20
      30 LET C = 2
```

værdier

Talvariabel som start slut og step værdi

Men husk at du ikke kan bruge samme navn til en kontrolvariabel som du har anvendt til en Talvariabel, uden den ene ændre den anden.

```
40 FOR A = D TO B STEP C
50 PRINT A;";"
60 NEXT A
```

5-H-4
Fortegns
Minus

START - SLUT - STEP værdien kan også gøres negativ ved at sætte minus foran (-1 -20 -2)

```
10 FOR A = 10 TO 0 STEP -1
20 PRINT A
30 NEXT A
```

Også decimalbrøker kan bruges som værdier.

```
10 FOR A = 100 TO 110 STEP .5
20 PRINT A
30 NEXT A
```

5-H-5

Dobbelt FORNEXT løkke
FOR NEXT løkke i en FOR NEXT løkke.

Du kan indstille en FOR NEXT LØKKE i en FOR NEXT LØKKE i en FOR NEXT LØKKE o.s.v.

Denne form for løkker kan anvendes hvor du ønsker et stigende antal gennemløb for hver løkke.

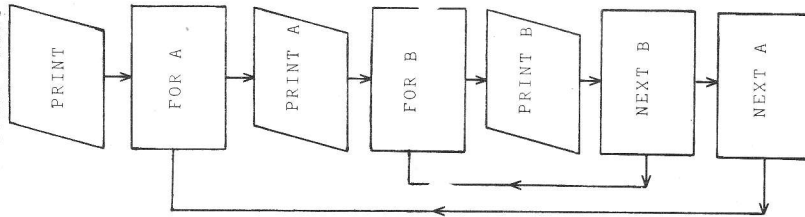
```
2 PRINT "GEMMENKØRSEL NR."
3 PRINT
5 PRINT "A-LØKKE", "B-LØKKE"
6 PRINT
10 FOR A = 1 TO 10
20 PRINT A;
30 FOR B = 0 TO 9
40 PRINT B;
50 NEXT B
70 NEXT A
```

FOR NEXT Rutediagram

5-H

Rutediagram med 2 løkker. Fra (5-H-5)

De tomme PRINT sætninger er ikke medtaget.



Undgå at hoppe ind i en løkke med GOTO, det vil give fejl.

Du kan også flette 2 løkker inden i hinanden prøv og følg denne sekvens.

```

1A 1B 2A 1B 3A 1B 4A 1B 5A 1B &A
1B 7A 1B 8A 1B 9A 1B 10A 1B 2B
3B 4B 5B 6B 7B 8B 9B 10B
  
```

FOR NEXT Rutediagram

5-H

```

10 FOR A = 1 TO 10
20 PRINT A; "A" ; "" ;
30 FOR B = 1 TO 10
40 PRINT B; "B" ; "" ;
50 NEXT A
60 NEXT B
70 PRINT
80 LIST
  
```


INDICEREDE TALVARIABLER i en dimension

5-I

5-I-1

DIM ()

Vi husker fra 5-C og 5-D at en almindelig variabel havde et navn som kan indeholde en værdi i form af tal for Talvariabler og Tekst for Tekstvariabler.

Du har en anden type Variabler som Computeren kan programmeres med, det er INDECEREDE variabler.

Du kan godt bruge samme bogstav til en Indeceret variabel som du i forvejen har anvendt til en alm. variabel uden den vil slette den alm. variabel.

Når du bruger flere variabler med samme navn bliver du nødt til at give den et nummer (indexnummer) som kan sammenlignes med en række huse i en Gade med nummer.

Hvis du kalder gaden for A bliver det første hus = A(1) det andet = A(2) o.s.v.

Hvis der er 25 huse i gaden, og vi vil lave en liste over hvor mange personer der bor i husene, begynder de med A(1) og slutter med A(25) skal du reservere Plads til 25 variabler i hukommelsen.

5-I-2

DIM
(dimensioner
et antal)

Dette gør du med kommandoen DIM (25) (gør plads til 25 variabler).

Til at indlæse disse 25 variabler kunne du bruge 25 INPUT sætninger INPUT A(1) INPUT A(2) o.s.v. indtil (25) men en FOR sætning med en kontrolvariabel B der tæller B værdi op til 25, er som skabt til dette brug.

Først en kommando der reserverer plads til 25 variabler
DIM (25)

Derefter en FOR NEXT løkke på 25 som du f.eks kalder B

```
FOR B = 1 TO 25
```

Første gang du foretager INPUT (B), har B værdien (1) næste gang (2) o.s.v. indtil løkken har nået 25 gange.

```
INPUT (B)
```

Nu mangler du kun at slutte af med NEXT B. samt at indsætte PRINT sætninger i løkken så du kan se hvor mange personer du har indputtet i hver hus.

fortsættes næste side

INDICEREDE TALVARIABLER i en DIMENSION

5-I

5-I-2

```
10 PRINT
"INDTAST DET ANTAL PERSONER
DER ER I
HUS"
20 DIM A(25)
30 FOR B = 1 TO 25
40 INPUT A(B)
50 PRINT "NR";B;" = ";A(B);" ";
60 NEXT B
```

Husk at placere alle tegnene rigtigt i Print-sætningerne.

5-I -3

Her er et program hvor du indtaster et beløb for hver måned.

Husk kommaet, der flytter beløbet 16 karakterer.

```
5 LET S = 0
10 DIM A (12)
20 PRINT "INDTAST BELØB"
25 PRINT
30 FOR M = 1 TO 12
40 INPUT A (M)
50 PRINT "MAANED NR. ";M,A(M)
60 LET S = S + A (M)
70 IF M = 12 THEN PRINT "AARET I
ALT";S
80 NEXT M
```

INDEGEREDE TALVARIABLER i flere dimensioner

5-I-5

Hvis du vil lave et program med Indesererede Variabler der indeholder flere dimensioner, skal du bruge lige så mange FOR NEXT løkker inden i hinanden (se 5-H-5 og 5-H-6) som du har dimensioner.

5-I-6
Opgave
DIM(3,3,3)

I en gade er der 3 blokke (blok 1) (blok 2) (blok 3)
I hver blok er der 3 opgange opg. 1 - opg. 2 - opg. 3
I hver opgange er der 3 lejligheder. Lejl. 1 - lejl. 2 - lejl. 3

Indlæs antal personer der bor i lejlighederne således at du kan spørge Computeren om netop den lejlighed du ønsker at kende antallet af personer på f.eks blok 2 opg. 3 Lejl. 2 og den giver dig svaret korrekt som det er indlæst.

5-I-7

Først reserverer du plads til 3x3x3 ialt 27 data med

```
DIM D(3,3,3)
```

Husk at bruge komma i DIM sætning imellem index nr. Derefter skulle du bruge lige så mange FOR NEXT løkker inden i hinanden som du har dimensioner, og med lige så stor værdi overført til kontrolvariablen, som der er Index i hver dimension.

```
100 DIM D (3,3,3)
110 FOR A = 1 TO 3
120 FOR B = 1 TO 3
130 FOR C = 1 TO 3
140 INPUT D (A,B,C)
150 PRINT A; B; C; " = "; D(A,B,C); " "
```

```
155 NEXT C
160 NEXT B
170 NEXT A
```

Prøv at indtaste dette program start det med RUN indlæs derefter tallene 1 til 27 så kan du følge dimensionernes Index fra D(1,1,1) til D(3,3,3)

Hvis du beder Computeren om at printe oplysninger om hvor mange personer der bor i D som var det navn vi gav den 3 dimensionelle variabel efterfulgt af f.eks. (2,1,2) og Computeren vil svare med 11
D(3,2,2) = 23. D(1,3,2) = 8 o.s.v.

INDEGEREDE TALVARIABLER i flere dimensioner

5-I-8

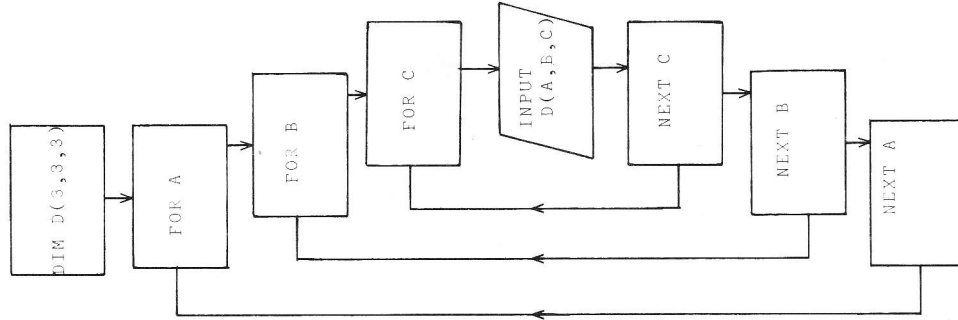
```
90 LET S = 0
100 DIM D (3,3,3)
110 FOR A = 1 TO 3
120 FOR B = 1 TO 3
130 FOR C = 1 TO 3
140 INPUT D (A,B,C)
145 LET S = S + I(A,B,C)
150 PRINT A;B;C; " = "; D(A,B,C); " "
```

```
160 NEXT C
170 NEXT B
180 NEXT A
185 CLS
190 PRINT "INDTAST 1 DIMENSION"
200 INPUT A
210 PRINT "INDTAST 2 DIMENSIONER"
220 INPUT B
230 PRINT "INDTAST 3 DIMENSIONER"
240 INPUT C
250 PRINT A; B; C; " = "; D (A,B,C)
260 PRINT "HVIS DU ØNSKER AT VI SE DEN SAMLET SUM TAST Q. HVIS D U ØNSKER FLERE DATA UD ET ANDET BOGSTAV"
265 INPUT A$
270 IF A$ = "Q" THEN GOTO 300
280 CLS
290 GOTO 190
300 PRINT "SUM="; S; " PERSONER"
310 INPUT B$
320 IF B$="Q" THEN GOTO 280
```

Et eks. på hvordan det efterfølgende program kan bygges på således at computeren stiller spørgsmål om hvilken adresse du ønsker oplysninger om, samt udskrive det samlede antal.

RUTEDIAGRAM FOR 3 DIMENSIONEL DATA

5-1-9
DIM(3,3,3)



Nu har du 3 kontrolvariabler der tæller op til tre hver.

Følg rutediagrammet LÆG mærke til at først får A så B og derefter får C sin værdi alle på 1 således at første data bliver indlæst på D(1,1,1)

Nu kører den i C løkken 3 gange og indlæser data på D (1,1,2) og D (1,1,3)

Derefter retunerer NEXT B den og tæller B en op.

Nu kører den igen 3 gange i C løkken og indlæser data i D (1,2,1) D (1,2,2) D (1,2,3)

Når B løkken er kørt 3 gange retunerer NEXT A den indtil A løkken har været 3 gange rundt.

Det sidste data der bliver indlæst i D (3,3,3) derefter fortsætter den til næste programlinje efter NEXT A

Du skal især lægge mærke til det sammenhæng der er mellem A og det første Index D (A, ,) mellem B og det andet Index D (, B,) mellem C og det tredje Index D (, , C)

Hvis du vender tilbage til opgaven i 5-1-5 kan du her overføre de 3 dimensioner A,B,C til :

- A = Blokkene
- B = Opgangene
- C = Lejlighederne

5-1

Indicerede tekstvariabler i en dimension.

5-1-10
DIM ()

5-1-11
INDICEREDE
TEKSTVARIABLER

Til at dimensionere indicerede Tekstvariabler anvender du samme fremgangsmåde som til indicerede Talvariabler, se sektion 5 kap. 5-1-1.

Istedet for en Talvariabel skal du nu bruge en indicerede Tekstvariabel. A\$(25,

Det første du skal gøre er at reservere plads til 25 A\$ indiserede tekstvariabler men samtidigt skal du reservere plads til tekstens maximum længde.

Det første index nr. er antallet af Variabler det sidste index er altid antal karakterer (bogstaver) uanset hvor mange deminsioner variabelen er på.

DIM A\$ (25,12)

Kommandoen DIM (reserverer plads til 25 variabler med en maximum længde på 12 karakterer) det betyder ikke noget, hvis ordet er kortere, men hvis det er længere end 12 karakterer vil de karakterer der er over 12 ikke blive optaget i programmet.

```

10 DIM A$(25,12)
20 FOR A = 1 TO 25
30 INPUT A$(A)
40 PRINT "NR" ; A ; " : " ; A$(A) ; " "
50 NEXT A
  
```

5-1

5-I-12
ØVELSE

Her er et program til at sortere navne.

```

5 REM SAVE "ORD"
8 CLS
10 PRINT "INDTAST ANTAL ORD"
20 INPUT O
25 LET U = O + 1
30 CLS
40 DIM D$(U,10)
50 FOR I = 1 TO O
60 INPUT D$(I)
85 NEXT I
86 PRINT "NU SORTERES ORDERNE"
87 PAUSE 50
88 FAST
89 CLS
90 FOR A = 1 TO O
100 LET H = A
110 FOR B = A + 1 TO U
120 IF D$(B) > D$(H) THEN GOTO 140
130 LET H = B
140 NEXT B
150 LET M$ = D$(H)
160 LET D$(H) = D$(A)
170 LET D$(A) = M$
180 NEXT A
183 SLOW
185 PRINT AT 20,0;
190 FOR C = 1 TO U
198 IF C >= 3 THEN SCROLL
200 PRINT D$(C)
202 IF C >= 2 THEN PAUSE 17
210 NEXT C
220 SCROLL
230 PRINT AT 21,0;"ØNSKER DU AT
SAVE INDHOLDET J/N"
240 INPUT X$
250 IF X$ <> "J" THEN RUN
260 SAVE "ORD"
270 GOTO 185

```

Indtast antal navne du ønsker at sortere i alfabetisk rækkefølge.

Indskriv navnene i vilkårlig rækkefølge (max 25 v. 2K. byte)
navnene bliver automatisk printet ud på skærmen når du har indskrevet det antal navne du har valgt.

5-I

INDISERENDE TEKSTVARIABLE I FLERE DIMENSIONER.

5-I-13

Til at dimensionere indlærende tekstvariable i flere dimensioner bruger du samme fremgangsmåde som til dimensionering af talvariable i flere dimensioner, kun med den forskel at det sidste led i en dimensioneres ikke som tekstvariable med konventionel plads til maximum karakterer som variable kan være på.

```
DIM A$(2,2,2,2,2,12)
```

A\$ i 5 dimensioner indeholdende tekst med maximum 12 karakterer.

Og nu skal du kun bruge lige så mange FOR NEXT løkker som der er dimensioner samt en INPUT sætning til at indlæse teksten.

```

10 DIM A$(2,2,2,2,2,12)
20 FOR A = 1 TO 2
30 FOR B = 1 TO 2
40 FOR C = 1 TO 2
70 FOR D = 1 TO 2
80 FOR E = 1 TO 2
90 INPUT A$(A,B,C,D,E)
100 PRINT A;B;C;D;E;" ";A$(A,B,
C,D,E); " ";
110 NEXT E
120 NEXT D
130 NEXT C
140 NEXT B
150 NEXT A

```

Læg især mærke til at index nr. som står for antal karakterer kun optræder i dimensioneringsætningen.

Når du har indlæst data kan du bede Computeren om at printe en hvilken som helst tekst imellem variabel nr. A\$(1,1,1,1,1) til nr. A\$(2,2,2,2,2) løst 2x2x2x2x2 = 24 variable.

5-J

PRINT TAB AT

5-J-1
TAB
(tabulator)

Denne Computer der giver dig muligheder for at printe på skærmen der hvor du beder den om det.

Til dette har den 2 print funktioner.

1. PRINT TAB virker som en tabulator på en skrivemaskine der inddeler de vandrette linier i 32 tabulatorpositioner, startende fra venstre med PRINT TAB Ø til den sidste PRINT TAB 31, ialt kan du printe 32 karakterer ind på en linie.

PRINT TAB Ø gør computeren ved alm. PRINT kommando Hvis du bruger et større tal end TAB 31 vil computeren dividere tallet med 32 og kun det tal der er til rest vil angive print positionen.

Eks. PRINT TAB 68 = 68/32 = 2 med 4 til rest således at PRINT TAB 68 er det samme som PRINT TAB 4.

Du kan også bruge semikolon foran ;TAB således du kan printe flere TAB positioner på en linie.

EKS. PRINT TAB 5;"A";TAB 10;"B";TAB 15;"C"

5-J-2
PRINT AT
(i)

2. PRINT AT skal efterfølges af to tal, det første angiver hvilken linie nummer, det andet angiver hvilken kolonnenummer.

De to tal linie, kolonne skal være adskilt med et komma samt efterfølges af semikolon ; for at genskabe den brudte print kommando på samme programlinie.

Eks. PRINT AT 10,15; "DETTE ER LINIE 10 DER BEGYNDER AT PRINTE I KOLONNE 15"

Du kan også indsætte talvariabler til at angive print AT positioner, herved kan du flytte print positionen ved at ændre talvariablens værdi.

Her er et eks. hvordan du under programudførelsen kan ændre printpositionerne, samtidigt giver programmet dig en fornemmelse af placeringen af de 21 linier og de 31 kolonner.

```
5 LET K = Ø
10 FOR L = Ø TO 21
20 LET K = K + 1
30 PRINT AT L, Ø; L; AT L,K; K
40 IF K = 22 THEN LET L = Ø
50 NEXT L
60 GOTO 10
```

5-J

PRINT AT

Når du har brugt PRINT AT f.eks. på linie 10 må du være klar over at næste alm. PRINT kommando vil PRINTE på næste linie, altså i dette tilfælde på linie 11.

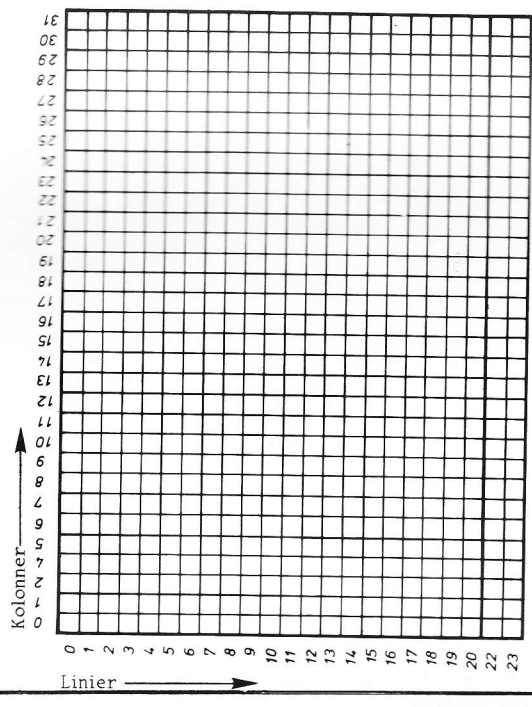
Hvis du vil PRINTE et andet sted f.eks. for oven af skærmen kan du give den kommandoen PRINT AT Ø,Ø og den vil fortsætte derfra på normalvis.

5-J-3

Hvis du kommanderer PRINT AT til en linie, kolonne der i forvejen er PRINTET på vil den slette det der står der.

Dette kan du anvende i forbindelse med SPACE til at slette bestemte ting, hvilket som helst sted på skærmen.

Ø kolonner eks. PRINT AT 8,10; "SPACE"



Du kan ikke programmere PRINT på de 2 nederste linier, de er forbeholdt til indtastninger af programlinier.

5-K

LEN VAL STR \$

5-K-1

LEN TIL
TEKSTSTRENG

LEN omskriver tekststreng til det antal karakterer der er i tekststrengen, og giver strengens længde i tal.

Eks. PRINT LEN "TEKSTSTRENG" = resultat II

På samme måde kan du også omskrive en Tekstvariabel til tal.

Eks. 10 LET A\$ = "TEKSTSTRENG "
20 PRINT LEN A\$

Hvis du vil have en tekst eller talkolonne til at slutte samme sted på skærmen (tabulatorposition) kan du trække længden af den streng eller talkolonne fra det sted (tabulatorposition) du vil have tal eller tekst til at slutte.

Eks. Du vil have en tekst til at slutte i kolonne 16 (det samme som tabulatorposition 16 = TAB 16)

PRINT TAB 16 - LEN "ROM"; "ROM"

Nu vil den trække længden af tekststrengen således at du altid slutter ved kolonne 16 (tabulatorposition 16)

5-K-2
Eksempel

Her er et eks. hvor du foretager indput af tekststreng af forskellige længde, prøv også at ændre tabulatorposition.

TAB-31 til et andet tal. se 5-I-1.

Eks. 10 INPUT A\$
20 PRINT TAB 31-LEN A\$; A\$
30 GOTO 10

5-K-3
LEN til tal-
kolonne

STR\$

Hvis du vil bruge LEN funktionen til tal eller talvariabler må du lave tal eller talvariablen om til den strengs længde det vil fylde i kolonner på skærmen.

Dertil bruger du STR\$ der har den funktion at den laver et tal eller talvariabel om til en streng.

Eks. 10 LET A = 555588
20 PRINT STR\$ A

Nu kan du bruge samme måde som du brugte til tekststreng.

Eks. 10 INPUT A
20 PRINT TAB 31-LEN STR\$ A; A
30 GOTO 10

5-K

LEN VAL STR\$

VAL
5-K-4

VAL funktionen virker modat BRT\$, således at den laver en tekststreng eller tekstvariabels indhold af talmæssige værdi om til tal.

Eks. 5 LET LONDON = 1234567
10 LET A\$ = "LONDON"
20 PRINT VAL A\$

Du kan også beregne med tekstens indhold af talmæssig værdi ved at lave tekststrengen om til tal med VAL.

Eks. 10 LET TI = 10
20 LET HUNDREDE = 100
30 LET TUSIND = 1000
40 PRINT VAL "2*TUSIND + 8*TI + 5*
HUNDREDE"

5-K-5

Her har du et eks. på hvordan du kan kombinere disse funktioner.

```
50 PRINT "PRISEN PAA SMØR"
60 INPUT SMØR
70 PRINT "PRISEN PAA KAFFE"
80 INPUT KAFFE
85 CLS
87 PRINT "INDTAST DEN VARE DU
VIL VIDE PRISEN PAA"
88 PRINT
90 INPUT A$
100 IF A$ = "SMØR" THEN PRINT "SM
ØR"; TAB 21-LEN STR$ SMØR; VAL A$;"
KR.PR KILO"
110 IF A$ = "KAFFE" THEN PRINT "K
AFFE"; TAB 21-LEN STR$ KAFFE; VAL
A$;" KR.PR KILO"
120 PRINT
130 GOTO 90
```

Det er vigtigt at du forstår hvordan du med VAL funktionen kan aflæse talværdien af en tekst eller tekstvariabel.

Du kan selv udvide antallet af varer ved at indsatte en PRINT en INPUT og en IF sætning for hver vare du tilføjer.

5-L

INT (integner)

5-L-1
INT
(helt tal)

Computeren har en funktion som du kan bruge til at runde ned til hele tal.

Eks. PRINT INT 3.4
og du får tallet 3 skrevet ud.

Du kan også få den til at runde opad ved at lægge 1 til tallet og sætte det i parentes ellers vil den udføre INT-funktionen før den lægger 1 til, da funktioner har højere prioritet (se 4-C-8)

Eks. PRINT INT (3.4+1)

5-L-2

Hvis du vil runde et tal af ved 100, 1000 o.s.v. kan du dividere tallet med det du vil afrunde det af til og anvende INT, derefter gange det op igen.

EKS. PRINT INT (34432/1000)*1000 = 34000.

Du kan også fjerne decimaler på den måde men her skal du først gange derefter dividere tallet.

EKS. PRINT INT (5.34432 x 100)/100 = 5.34

Hvis du vil runde op eller ned ved 0.5, kan du bruge +05.

EKS. PRINT INT (3.4 + 0.5)

5-M

RND RAND

5-M-1
RND

Computeren har en funktion der udsender en forud bestemt talrække på 65536 forskellige tal med en værdi mellem 0 og 1, men aldrig 1.

Denne talrække begynder samme sted hver gang du har brugt NEW eller haft strømmen afbrudt.

RND kan du bruge til at give en talvarlabel en skiftende værdi der tilsyneladende virker tilfældigt på grund af den lange række af tal.

Prøv dette eks.

```
10 PRINT RND
20 GOTO 10
```

Nu kan du se det første stykke af denne talrække der starter samme sted hver gang når strømmen har været afbrudt eller du har brugt NEW.

5-M-2
RAND

Hvis du ønsker et helt tilfældigt tal kan du indsatte kommandoen RAND i programmet på en programlinje der ligger før den du bruger RND 1.

RAND henter starttidspunktet til RND fra en tæller kredsløb i Computeren, så det tidspunkt RAND kommandoen bliver udført er afgørende for hvilket starttal RND får. Dette bevirker at du får en helt uforudsigelig talrække.

Du kan også selv bestemme starttidspunktet ved at sætte et tal efter RAND (max. 65535) eks. RAND 95

5-M-3

Med RND kan du få computeren til at udvælge tal mellem 0 og 1 men aldrig 1, og med RAND kan du gøre disse tal helt uforudsigelige.

Hvis du ønsker tilfældige tal imellem 0 og et andet tal eks. 10 ganger du RND med 10, så får du et tilfældigt tal fra 0 til 10, men aldrig 10.

Hvis du bruger INT. får du tal fra 0 til 9

```
10 INPUT A
20 LET B = INT(RND 10)
30 IF A = B THEN PRINT AT 10,10;
"RIGTIGT"
40 IF A<>B THEN PRINT AT 10,10
"FORKERT"
50 GOTO 10
```

5-M
RND RAND

Som du kan se er RAND kommandoen ikke brugt, det bevirker at den vil starte med samme tal hver gang du laver dette program, så hvis du vil undgå dette kan du indsætte en linie 5 RAND så vil den starte helt tilfældigt.

5-M-3

Som du kunne se i 5-M-1 er RND en decimalbrøk på op til 9 decimaler imellem 0 og 1, alle de decimaler har du almindelig vis ikke brug for så dem fjerner vi ved at sætte INT foran RND men husk hvis du vil have RND ganget op til et andet tal skal du indsætte gange stykket i parentes ellers vil du kun få nuller.

Eks. PRINT INT RNDx10 vil kun give 0.

Eks. PRINT INT (RNDx10) giver tal fra 0 til 9

5-M-4

Her har du et programeks. med RAND og RND.

```

1 REM GAET
5 RAND
10 LET Q = 0
20 CLS
30 LET A = INT (RND*100)
40 PRINT "FIND TALLET (0 TIL 1
00)"
50 INPUT B
60 LET Q = Q + 1
70 IF B = A THEN GOTO 130
80 IF B>A THEN GOTO 110
90 PRINT B; " FOR LAVT"
100 GOTO 50
110 PRINT B; " FOR HØJT"
120 GOTO 50
130 PRINT B; " FULSTÆNDIGT RIGT
IGT"
140 PRINT "DU BRUGTE ";Q;" FORS
0G"
150 PRINT "VIL DU PRØVE IGEN"
160 INPUT C$
170 IF C$ = "JA" THEN GOTO 10
175 IF C$ = "NEJ" THEN GOTO 200
180 FOR X = 0 TO 60
190 NEXT X

```

SEKTION 5

5-N
SLICING (dele en tekststreng)

5-N-1

En tekststreng kan deles ved at sætte tal og TO i parentes efter tekststrengen eller tekstvariablen.

EKS. PRINT "SKANDINAVIEN" (3 TO 7)

Du kan tælle dig frem til hvilken karakter du vil have printet ud.

Hvis du undlader det første tal (TO) tager den de første 7 karakterer og skærer resten væk.

Hvis du undlader det sidste tal eks. (3 TO) tager den ikke de 3 første karakterer.

5-N-2

SLICING

Du kan også indsætte en SLICING i en FOR NEXT løkke og bruge kontrolvariablen som tal for SLICING således at du lægger det antal karakterer du vil der skal være på skærmen til kontrolvariablen og sætter dette efter TO samt kontrolvariablen før TO

Eks. (A TO A+25)

Længden af FOR NEXT løkken bestemmes af længden af den tekststreng du indtaster minus længden af det antal karakterer der skal være på skærmen.

Husk at lave tekststrenger længere end det antal karakterer du vil have fremme på skærmen ellers kan FOR NEXT løkken ikke køre.

```

10 INPUT A$
20 FOR A = 1 TO LEN A$ - 25
30 PRINT AT 10,3; A$ (A TO A+25)
40 FOR B = 1 TO 3
60 NEXT B
70 NEXT A
80 GOTO 20

```


5-0

5-0-1
CHR\$
(karakter)

CHR\$ efterfulgt af et code nr. giver den karakter der har det code nr.

Computeren har et karaktersæt på 256 stk. karakterer du kan få hele karaktersættet, det samme som du har adgang til fra tastaturet printet ud på skærmen.

Det er mest praktisk at have nummeret med så du kan notere nummeret ned på de karakterer du vil bruge, men du bliver nød til at dele det i tre programmer

```
et hvor linie l0 er LET-A = 0
et hvor linie l0 er LET A = 80
et hvor linie l0 er LET A = 160
```

```
l0 LET A = 0
20 PRINT A; "="; CHR$ A; " ";
30 LET A = A + 1
40 GOTO 20
```

Med CHR\$ kan du ændre en karakters betydning ved at sætte den ind i en IF og en INKEY\$ sætning.

```
LET A$ = INKEY
IF A$ = CHR$ 19 THEN GOTO og udføre noget
```

5-0-2
CODE

Alle tegn, tal, bogstaver mm. repræsenteres af en CODE med et tal mellem 0 og 255

CODE giver talkoden af det første bogstav i en tekststreng, du kan sætte CODE foran tekst eller tekstvariabel og du får talkoden for det første bogstav i strengen.

Her er et eks. på hvordan du i et program kan begrænse brugen af tastaturet.

```
l0 INPUT A$
20 IF CODE A$ > 63 THEN GOTO l0
30 PRINT A$;
40 GOTO l0
```

5-P

5-P-1
INKEY\$

Funktionen INKEY\$ undersøger om der er en tast trykket ned idet øjeblik den programlinje bliver udført.

```
20 LET A$ = INKEY$
30 PRINT A$;
40 GOTO 20
```

Du kan indsætte en en PAUSE, hvis du vil nedsætte hastigheden i programmet.

5-P-2

Med INKEY\$ kan du ændre en tastens funktion. Brug en tekstvariabel til at overføre INKEY\$

Eks. LET A\$ = INKEY\$

Efter denne LET sætning kan du bruge IF sætninger til at undersøge om netop denne tast du vil ændre betydning på er tastet.

På denne måde kan du lave et tegne program som du kan styre med et JOYSTIK.

```
l0 LET X = 31
20 LET Y = 21
30 LET A$ = INKEY$
40 IF A$ = "F" THEN IF X < 63 THEN
LET X = X + 1
50 IF A$ = "4" THEN IF X > 0 THEN
LET X = X - 1
60 IF A$ = "R" THEN IF Y < 43 THEN
LET Y = Y + 1
70 IF A$ = "7" THEN IF Y > 0 THEN
LET Y = Y - 1
80 IF A$ = "U" THEN CLR
90 PLOT X,Y
l00 GOTO 30
```

Hver gang INKEY\$ bliver udført aflæser computeren tastaturet og registrerer om der er en tast der er aktiveret. Denne tasts karakterer tildeler INKEY\$.

5-Q

5-Q-1
CLEAR
(slettet)

CLEAR kan du bruge hvis du vil have slettet alle variable uden at starte programmet forfra som du vil gøre hvis du bruger RUN.
Start igen med CONT ENTER

5-Q-2
SCROLL
(rulle)

Med SCROLL kan du flytte skærbilledet 1 linie opad således at øverste linie forsvinder og der bliver plade til en ny linie foruden.

```
Eks 10 LET A = RND*31
      20 PRINT AT 21,A; ". "
      30 SCROLL
      40 GOTO 10
```

5-Q-3
TEKSTBEHAND-
LING

Her har du et færdigt program som du kan indtaste således at får et indtryk af hvordan du kan sætte kommandoer og funktioner til et færdigt program.

```
1 PRINT AT 21,0; " ";
2 LET A = 0
3 LET A$ = " "
5 IF A$ = INKEY$ THEN GOTO 5
7 LET A$ = INKEY$
14 IF A > 30 THEN GOTO 1010
16 IF A$ < " " THEN GOTO 5
17 IF A$ = "X" THEN GOTO 50
19 IF A$ = CHR$ 19 THEN GOTO 2000

33 LET A = A + 1
35 PRINT AT 21,A; A$;
40 GOTO 5
50 IF A = 30 THEN GOTO 1000
52 PRINT AT 21,A+2; "* ";
53 PRINT AT 21,A + 1; " ";
55 LET A = A + 1
60 GOTO 7
1000 PRINT AT 21,31; " ";
1010 SCROLL
1011 PRINT AT 21,0; " ";
1012 LET A = 0
1030 GOTO 7
2000 PRINT AT 21,A; " ";
2010 LET A = A - 1
2015 IF A < 0 THEN LET A = 0
2020 GOTO 7
```

Start programmet med RUN.
Slet og fremløb er lagt ind på tasterne < og X.

SEKTION 6

6-A-1 Grafik
6-A-2 INVERS

PLOT/UNPLOT

PLOT (afsæt et punkt)

Eksempel

Plottepunkter

UNPLOT (Fjern et punkt)

PLOTSKEMA

Søjlediagram eksempel

Eksempler med matematiske formler

Universalsprogram

GRAFIK MED PRINT

6-A-1
GRAFIK

På computeren findes der fordefineret GRAFIKTEGN, der er påtrykt tastene med grønt, disse anvendes på følgende måde:

Ovenover ENTER-tasten, står der GRAPHICS på pladen. Når du holder SHIFT og trykker ENTER fremkommer et blinkende G i CURSOR, som betyder at computeren er klar til at skrive i grafik.

Eksempel:

Skriv COMPUTOR og teksten vil fremkomme, som sort skrift på hvid baggrund.

VIGTIGT:

Disse grafiktegn kan ikke bruges som variabler eller til at skrive programmer med, de kan kun bruges til PRINT-sætninger.

Hvis du ønsker, at benytte de grafiske tegn på tastene skal du holde shift og taste det ønskede grafiktegn.

Grafiktegnene på tastene 1, 2 og 3 er det direkte adgang til, ved blot at holde SHIFT.

Når du ikke længere ønsker at anvende GRAFIK-tegnene, holder du SHIFT og trykker ENTER. G'et vil forsvinde fra CURSOR og computeren vil være tilbage til normalt tegnsæt.

Foruden alle tegn, tal og bogstaver i INVERS (omvendt) skrift, har du grafiktegnene, som du kan PRINTE med samme kommandoer som almindelig skrift.

Med denne grafik har du muligheder for at lave specielle effekter til tekst og spil m.m.

Eksempel:

```
10 LET A = RDN*29
20 PRINT AT 21,A,"MMM"
30 SCROLL
40 GOTO 10
```

De grafiktegn i PRINT AT sætningen bevirkker, at du må sætte RDN til gange 29, for at den ikke skal overskride skærmens 31 printpositioner.

6-A-2

GRAFIK MED PLOT

6-B-1

Computoren har 2 kommandoer, du kan anvende for at lave grafik på skærmen.

PLOT= der placerer et hvidt punkt (plottepunkt) på skærmen, der hvor du beder den om det, UNPLOT=fjerner det hvide punkt, der hvor du beder den om det.

6-B-2

PLOT
(afsæt et punkt)

I sektion 5-J-2 om PRINT TAB AT, Her inddelte vi skærmen i 22 linier med hver 32 kolonner ialt 22*32=704 printpositioner, hvor vi kunne have en karakter i hver position.

Når du anvender PLOT, inddeler du hver printposition i 4 PLOT-positioner, således at du nu har 4*704=2816 PLOT-positioner, som du kan indtegne med hvide punkter.

For at finde frem til hvert enkelt punkt, har du et koordinatsystem, der består af en vandret X-koordinat og en lodret Y-koordinat.

Prøv at taste nedenstående program ind, så du får en fornemmelse af, hvor mange punkter, du har til din rådighed.

```
10 FOR A= 1 TO 704
20 PRINT " ",
30 NEXT A
```

Du skal ikke lade dig forvirre af denne print sætning, det er kun for at lave et nemt program, ikal vi lave det samme program med PLOT, skal vi sikre os, at skærbilledet ikke bliver overskredet, og vi får stop med fejl (SF).

Du har også et koordinatsystem til PRINT AT efterfulgt af to tal, hvor det første tal var den lodrette række af linier.

I PLOT koordinatsystemet er det første tal efter PLOT det vandrette nummer på plottepunktet, altså lige modsat af PRINT AT.

I den lodrette Y koordinat begynder plottepunktet for neden i venstre hjørne, altså lige modsat PRINT AT, der begynder foroven.

Det er den vandrette X-koordinat og den lodrette Y-koordinat, du indsætter dine plotpunkter efter.

Vil du sætte et punkt nede i højre hjørne, ser du på plotteskemaet, og finder det punkt, du ønsker plottet på:

Eksempel:

```
PLOT 60,5      computoren sætter nu et hvidt punkt nede i
                højre hjørne.
```

GRAFIK MED PLOT

6-B-3
Eksempel

Du kan du finde frem til et hvilket som helst punkt på skemaet ved hjælp af X-koordinaten for det vandrette, (men husk tallet må ikke overstige 63) og med et komma efterfulgt af Y-koordinaten for det lodrette punkt. (Her må tallet ikke overstige 43)

For at holde tallene indenfor dette område, kan du bruge kontrolvariablen fra FOR NEXT, som i sig selv har en begrænsning eks. (1 TO 43), og som derudover har en stepværdi, således at du kan springe et vist antal over.

Eksempel:
10 FOR A=0 TO 43 STEP 2
20 PLOT 0,A
30 NEXT A

Læg mærke til, at den starter nederst i venstre hjørne med punktet 0,0.

Hvis du nu vil have den vandrette X-koordinat plottet kan du lave en FOR NEXT løkke, inden i den anden løkke, der begrænser sig til 63, og med en stepværdi på 2.

Eksempel:
10 FOR A=0 TO 43 STEP 2
15 FOR B=0 TO 63 STEP 2
20 PLOT B,A
25 NEXT B
30 NEXT A

På dette program kan du tydeligt se, hvor både X- og Y-koordinaten starter.

6-B-4
Plottepunkter

Du har 3 muligheder, når du skal angive plottepunkter.

1. Du kan angive direkte med tal
2. Du kan angive med talvariabler
3. Du kan indsatte beregninger direkte i PLOT-sætningen

Indgår der decimalbrøker i plottepunkterne vil computoren forhøje tallet til næste plote punkt, så snart tallet er over 5.

6-B-5
UNPLOT
(fjern et punkt)

Hvis du nu ønsker punktet fjernet igen, anvender du UNPLOT-kommandoen efter f.eks. tallet 60,5, og det hvide punkt forsvinder.

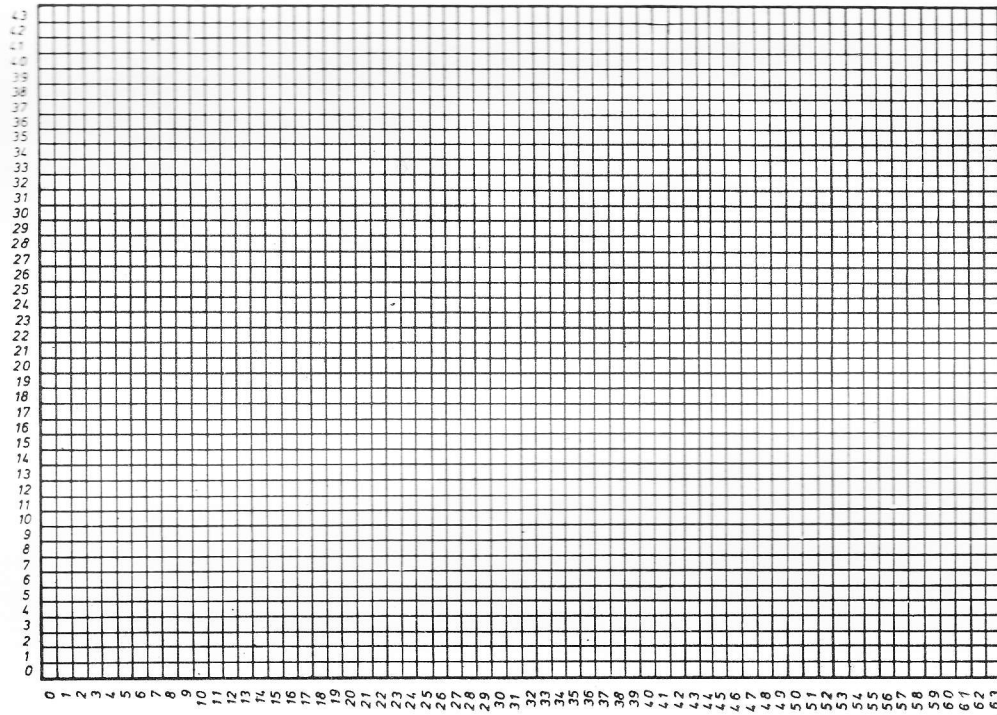
Du kan fjerne plottepunkterne efter samme koordinatsystem eks. PLOT 20,10 fjernes med UNPLOT 20,10.
Du kan også bruge dette til at få et punkt til at blinke eller køre henover skærmen, ved at anvende UNPLOT bagefter.

Eksempel:
10 for A=2 TO 43
20 FOR B=2 TO 63
30 PLOT B,A
40 UNPLOT B-1,A
50 NEXT B
60 NEXT A

GRAFIK MED PLOT

6-B-6
PLOT skema

Y- KOORDINATOR



X- KOORDINATOR

PLOT PROGRAM MED BEGRENINGER

I efterfølgende eksempel, kan du dreje cirklen så den ligner en planet.

```
10 FOR A=0 TO 2 STEP 2
20 FOR B=0 TO 120
30 PLOT 32+20*SIN (B/60*PI),22
+20xCOS (B/60*PI+A/4*PI)
40 NEXT B
50 NEXT A
```

De matematiske formler kan ændres, så du får andre figurer.

6-C-3

Her har du et univelsalprogram, hvor du skal INPUT tal for A,B,C,D. Du kan selv bestemme tallene, der vil give forskellige figurer.

```
1 REM INPUT TALLENE A,B,C,D
2 REM DE INSAETTES I FORMLER
3 REM OG FREMBRINGER GRAFIK=
4 REM KURVER
5 PRINT "INPUT A"
10 INPUT A
12 PRINT A
15 PRINT "INPUT B"
20 INPUT B
22 PRINT B
25 PRINT "INPUT C"
30 INPUT C
32 PRINT C
35 PRINT "INPUT D"
40 INPUT D
42 PRINT D
45 CLS
50 FOR T=0 TO 2 STEP .01
60 PLOT INT (.5+31*SIN ((A*T+B/6)*PI))+32,
INT (.5+21*SIN ((C*T+D/6)*PI))+22
90 NEXT T
```

PLOT PROGRAMMER MED BEGRENINGER

Her er et eksempel på et søjlediagram, der viser dagens valutakurser. I programmet er der brugt en INPUT-sætning der venter på, at du skal indtaste dagens valutakurser. Tælleren kan max. angive 10 forskellige valutaer:

```
Eksempel:
5 LET D=-1
10 PRINT AT 0,0,"INDTAST DAGEN
S VALUTAKURSER"
20 LET B = -2
30 INPUT A
40 LET B=B+2
50 LET A=A*(2.75/100)
60 FOR A=2 TO A
70 PLOT B,A
80 NEXT A
85 LET D=D+1
90 PRINT AT 21,D;D+1
100 GOTO 30
```

6-C-2

Eksempler med matematiske formler:

```
Eksempel 1:
10 FOR A=20 TO 0 STEP -5
20 FOR B=1 TO 3 STEP .05
30 PLOT AXSIN (PIXB)+30,20xCOS
(PIXB)+20
40 NEXT B
50 NEXT A

Eksempel 2:
10 FOR Z=1 TO 704
20 PRINT "■";
30 NEXT Z
40 FOR A=20 TO 0 STEP -5
50 FOR B=1 TO 3 STEP .0125
60 UNPLOT AXSIN (PIXB)+30,2+20
xCOS (PIXB)+20
70 NEXT B
80 NEXT A

Eksempel 3:
10 FOR A=0 TO 31
20 PRINT AT 11,A;"-"
30 NEXT A
40 FOR B=0 TO 6.3 STEP 0.1
50 PLOT Bx10,20+20xSIN B
60 NEXT B

Eksempel 4:
10 LET B=2
20 FOR T=0 TO 60
30 LET A=T/30*PI
40 LET X=21+BxSIN A
50 LET Y=22+BxCOS A
60 PLOT X,Y
70 NEXT T
80 LET B=B+2
90 GOTO 20
```

7-A
INDELEDNING

Denne computer er udstyret med en lydenhed, der giver dig mulighed for at anvende musik og lyd effekt i dine programmer.
Du kan anvende 3 kommandoer til at programmere lyd:

MUSIC: der giver toner i 3 oktaver

SOUND: der bruges som lyd effekt til programmer og andet du vil have lydeffekt på.

TEMPO: der bestemmer hastighederne på dine musikprogrammer.

Med musikkommandoen angives en tonehøjde, der defineres med bogstaver C D E F G A B (B er det samme som H, men her skal B anvendes).
Computoren adskiller de forskellige oktaver på den måde, at hvor der ikke er noget tegn efter bogstavet, så er det den mellemste tonehøjde.

Er der angivet > (større end), så er det højeste oktav:

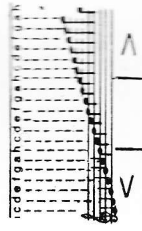
Er der angivet < (mindre end), så er det laveste oktav

Eks.:

"C<<D<E<C D E C>D>D>E"

Husk at sætte musikstrengene i anførselstegn.

Tone skala i 3 oktaver.



7-A-3
G-nøgle

7-A-4
Programlinje med MUSIK KOMMANDO skrives således at
F.eks.

10 MUSIC " node oktavtegn tonelængde "

10 MUSIC " C 10"

Tonelængden sætter du til et tal mellem 0 og 99
Prøv den dybeste tone MUSIC "C 10"
Og den mellemste tone MUSIC "F 10"
Og den højeste tone MUSIC "B 10"

SEKTION 7

7-A-1

MUSIC, SOUND, TEMPO

7-A-2

MUSIC

7-A-3

G-nøgle

7-A-4

Musik kommando

7-A-5

Øvelse

7-A-6

Tone længde

7-A-7

Nodebogstaver med grafikbogstaver

7-B-1

TEMPO

7-B-2

MUSIC

7-C-1

SOUND, SLOW, FAST

7-C-2

Opbygning af en SOUND sætning

7-C-3

Eksempel

7-C-4

RND

7-C-5

Eksempel med MUSIC, TEMPO, SOUND

7-A
INDLEDNING

Lav følgende øvelse:

```

5 TEMPO 20
10 A$="C<6 D<6 E<6 F<6 G<6 A<6
B<6"
20 B$="C<6 D<6 E<6 F<6 G<6 A<6
B<6"
30 C$="C6 D6 E6 F6 G6 A6 B6"
40 D$="C6 D6 E6 F6 G6 A6 B6"
50 E$="C>6 D>6 E>6 F>6 G>6 A>6
B<6"
60 F$="C>6 D>6 E>6 F>6 G>6 A>6
E<6"
70 MUSIC A$
80 MUSIC C$
90 MUSIC E$
100 MUSIC B$
110 MUSIC D$
120 MUSIC F$
130 MUSIC A$
140 MUSIC B$
150 MUSIC C$
160 MUSIC D$
170 MUSIC E$
180 MUSIC F$
    
```

7-A-6
TONELÆNGDE

Tonelængden i dette system kan du selv bestemme inden for tallene 0 til 99, men som et udgangspunkt kan du bruge denne tabel.

NOTE	TONELÆNGDE
Punkteret helnote	= 96
helnote	= 64
Punkteret halvnote	= 48
halvnote	= 32
Punkteret fjerdedelsnote	= 24
fjerdedelsnote	= 16
Punkteret ottendedelsnote	= 12
ottendedelsnote	= 8
Punkteret sekstendedelsnote	= 6
sekstendedelsnote	= 4
Punkteret toogtredivtedelsnote	= 3
toogtredivtedelsnote	= 2

7-A-7

Hvis du skriver nodebogstaverne med GRAFIKbogstaver for højer du tonen med $\frac{1}{2}$ node svarende til kryds.

Eks.
Kryds for C skrives som C i grafik.

7-B
TEMPO

7-B-1
1 til 255

Det tempo dine musikprogrammer afvikles med, bestemmer du selv indenfor tallene fra 1 til 255.

Til dette bruger du kommandoen TEMPO efterfulgt af det tal, du ønsker tempoet sat til.

I det efterfølgende eksempel har du et program, hvori der er indsat en talvariabel for tempo, du kan således selv prøve forskellige tempi. Dette eksempel er en strofe af OH SUSANNA.

```

Eks.
Start med det første input på ca. 16
10 PRINT "INDTAST ET TAL MELLEM 1 OG 255"
20 PRINT
30 INPUT A
40 TEMPO A
50 MUSIC "C6D2E3G3G6A2G3E3C6D2E3E3D3C3D12"
60 PRINT "TEMPO.":A,
70 GOTO 30
    
```

Når du har fundet det tempo, som du finder passende, kan du fjerne INPUT sætningen og erstatte variabelen A med tal f.eks. TEMPO 15.

7-B-2
MUSIC

Eksempel på musik program.
Nedenfor giver vi et eksempel på hvordan et færdigt musikprogram kan opbygges.

```

5 TEMPO 14
20 A$="C>2C2C>2G2C>2E2D>2G2E>2
C2E>2G2E>2E2F>2G2G>2C2G>2G2A>2E2
G>2G2E>2C2E>2G2E>2E2E>2G2"
30 B$="G>2C2G>2G2F>2E2E>2G2D>2
A<2D>2C2"
40 C$="C>2C2C>2G2C>2E2C>2G2"
50 D$="G>2C2G>2G2F>2E2E>2G2D>2
A<2D>2C2G2F2G2G2F>2A<2F>2G2E>2F2
D>2G2C>2C2C>2G2C>2E2C>2G2"
70 MUSIC B$+D$>2F2D>2G2"
80 MUSIC "F>2A<2F>2G2E>2F2D>2G
2"+C$
90 GOSUB 200
100 MUSIC D$
110 MUSIC D$
120 GOSUB 200
140 STOP
200 MUSIC A$
210 MUSIC B$+E>2F2D>2G2"
230 MUSIC "C>2C2G>2G2E>2E2G>2G2
C>8"
240 RETURN
    
```

7-C-1
SOUND

SOUND kommandoen anvender du i forbindelse med specielle lydeffekter i dine programmer.
Sound udfører computoren normalt i SLOW se 5-A-5, således at billedet står på skærmen, når SOUND udføres. Du kan også indsatte en FAST kommando før SOUND kommandoen. Dette forhøjer lydets frekvens 4 gange (på grund af FAST kommandoen), men til gengæld forsvinder skærm-billedet.

7-C-2
OPBYGNING AF EN
SOUND BÆTNING

SOUND kommandoen skrives som SOUND efterfulgt af to tal med et komma imellem.

Det første tal angiver tonehøjden, og skrives med tal fra 1 som den laveste tone og til 255, som den højeste tone.

Det næste tal angiver tonelængden og skrives med tal fra 1 som den korteste tone og til 65535, som den længste tone.

Eks.

10 SOUND 100,1200

Der kan også indsættes talvariabler efter SOUND

Eks.

10 SOUND A,B

Ved brug af ovennævnte kan du skifte værdi for både tonehøjde og tonelængde under programafviklingen.

7-C-3
EKSEMPEL

Her viser vi et eksempel med 3 forskellige toner med 3 forskellige længder.

10 SOUND 13,20000

20 SOUND 21,10000

30 SOUND 95,15000

7-C-4
RND

I nedenstående eksempel anvendes RND til at give variablen en værdi, der i dette tilfælde er indsat som tonehøjde

10 FAST

20 FOR Q=0 TO 200

30 LET A=RND*255

40 SOUND A,6000

50 NEXT Q

60 SLOW

7-C-5
EKSEMPEL MED
MUSIC, TEMPO,
SOUND

Vi samler nu alt hvad vi ved om MUSIC, TEMPO og SOUND kommandoerne i et færdigt program.

Som udgangspunkt anvendes følgende eksempel:

1 TEMPO 20

2 FOR W=0 TO 2

5 MUSIC "D>2D2D>2D2E>2E2E>2F2

C>2C2C>2C2C0<<2C2C<<2G2G<<2G2

G<<2G2G<<2

45 FAST

50 FOR A=0 TO 6

7-C
SOUND

7-C-4
fortsat

60 FOR B=18 TO 0 STEP -2
75 SOUND B+8,1500
80 NEXT B
100 NEXT A
105 NEXT W
110 SLOW
120 MUSIC "C<10"

8-A	OVERFØRING AF PROGRAMMER FRA BÅNDOPTAGER TIL COMPUTER
8-A-1 LOAD	<p>Computoren er udstyret med en indgang EAR, der er tilpasset EAR (høretelefon) udgangen på båndoptageren.</p> <p>Når du har opstillet computoren, som beskrevet i sektion 1, og CURSOR står og blinker, er du klar til at tilkoble en båndoptager.</p> <ol style="list-style-type: none"> 1. Med computoren fik du leveret et minl jock kabel, som du monterer fra båndoptagerens EAR udgang til computorens EAR indgang. 2. Læg et bånd i båndoptageren og spol båndet tilbage. 3. Indstil lydstyrken til ca. 3/4 del af fuld styrke. 4. Indstil tonekontrollen på HIGH (den højeste tone). 5. Skriv NEW og tast ENTER. Dette renser computorens hukommelse. 6. Skriv LOAD efterfulgt af programmets navn som indtastes i anførselstegn for eksempel LOAD "GAME" 7. Tast ENTER og start båndoptageren på PLAY. <p>Efter få sekunder vil der komme sort/hvide striber på skærmen. Dette er programsignaler, og de giver striber på skærmen. Det er ret afgørende, at disse striber er så regelmæssige som muligt.</p> <p>Bl. a. bånd, båndoptagerens kvalitet og tilstand har indflydelse på signalets styrke.</p> <p>Når båndet er overført til computoren melder den OK, kan du standse båndoptageren og taste RUN ENTER.</p>
8-A-2 LOAD UDEN NAVN	<p>Hvis du ikke kender navnet på programmet, kan du unlade at skrive dette, og blot skrive LOAD efterfulgt af 2 anførselstegn.</p> <p>LOAD""</p>
8-A-3	<p>Hvis computoren ikke modtager dit program, kan dette skyldes flere ting.</p> <ol style="list-style-type: none"> 1. Forkert styrke - prøv at indstille styrken, som beskrevet i sektion 8A-1 pkt.3 2. Tonehovedet er tilsnævset 3. Bånd eller båndoptager kan ikke gengive et tilstrækkeligt rent program, til at computoren vil modtage det.
8-A-4 STØJ	<p>Findes der støj i starten af båndet, kan det være en fordel, at lade båndet køre forbi støjen, indtil der kommer en svag susen. Stands båndoptageren og sæt EAR stikket i båndoptageren igen.</p> <p>Skriv LOAD"NAVN" og tast ENTER - start båndoptageren.</p>

SEKTION 8

- 8-A-1 LOAD
- 8-A-2 LOAD uden navn
- 8-A-3 Programmet modtages ikke
- 8-A-4 STØJ
- 8-B-1 SAVE
- 8-B-2 Flere programmer på et bånd
- 8-B-3 Start med GOTO
- 8-B-4 Program med automatisk SAVE

OVERFØRING AF PROGRAMMER FRA COMPUTOR TIL BÅNDOPTAGER

Du kan også lave programmer med automatisk SAVE, dette kan du anvende til f.eks finansprogrammer.

Eksempel:

```

10 PRINT "INDTAST NAVN"
20 INPUT A$
30 CLS
40 PRINT "INDTAST ADRESSE"
50 INPUT B$
60 CLS
70 PRINT "INDTAST PORTNUMMER"
80 INPUT A
90 CLS
100 PRINT "INDTAST BY"
110 INPUT C$
120 CLS
130 PRINT "MASKINEN ER NU KLAR
TIL AT SAVE DE INDTASTEDE DATA P
AA BAAND GØR DIN BÅNDOPTAGER
KLAR TIL AT OPTAGE COMPUTOREN S
AVER NAAR URET VISER LØØ"
140 PRINT AT 12,14; "-ØR-"
150 FOR B=10 TO 100
160 PRINT AT 13,15; B
170 NEXT B
180 FOR C=0 TO 25
190 PRINT AT 18,8; "START BAAANDE
T"
"
"
210 NEXT C
220 CLS
230 SAVE "DATALAGER"
240 PRINT TAB 10;"DATALAGER"
250 PRINT "NAVN"
260 PRINT A$
270 PRINT "ADRESSE"
280 PRINT B$
290 PRINT A; " "; C$
300 GOTO 10
    
```

Når alle kontonumre har fået input, vil den give dig signal, når du skal starte båndoptageren.

Hvis du vil udbygge programmet med flere variabler, kan du bruge indicerede variabler.

OVERFØRING AF PROGRAMMER FRA COMPUTOR TIL BÅNDOPTAGER

Når du har lavet et program, du gerne vil gemme, kan du anvende en almindelig kasettebåndoptager til at lave en kopi af det program, du har i computorens hukommelse.

I computoren findes en enhed, der når du giver den besked på det med kommandoen SAVE, udsender det program, der ligger i computorens hukommelse i en hastighed, der kan optages af den alm. kasettebåndoptager.

1. Når du vil overføre et program SAVE, forbinder du computorens MIC udgang til båndoptagerens MIC indgang.
2. Undgå at have EAR stik i båndoptageren under SAVE, da dette kan sende støj fra båndoptageren ind i computoren og igen give støj på programmet.
3. Sæt et bånd af god kvalitet (low-noise-ferro) i båndoptageren og spol det tilbage.
4. Skriv SAVE efterfulgt af det navn, du vil have programmet SAVED under i anførselstegn.
For eksempel SAVE "GAME"
5. Start båndoptageren med REC - PLAY og tast ENTER

Der vil nu fremkomme nogle sorte/hvide striber på skærmen. Når skærmen igen bliver grå, kan du standse båndoptageren og dit program er blevet overført.

Hvis du har tæller på din båndoptager, kan du godt indspille flere programmer på et bånd.
Hvis du ikke har tæller på din båndoptager, er det mest praktisk kun at have et program på båndet.

Save et program med automatisk start, der ikke sletter variablerne.
Når du indsætter SAVE kommandoen i programmet vil computoren udføre første linie efter SAVE, så snart programmet er overført.

Dette kan du anvende til automatisk start af dine programmer, ved at lade næste programlinie efter SAVE starte programmet enten med GOTO, hvis du vil undgå at slette variablerne i programmet, eller med RUN, men så sletter du den data du evt. har lagt ind i programmets variabler.

```

Eksempel:
500 PRINT " TAST CONT OG PROGRAMMET VIL SAVE MED AUTO
505 STOP MATISK START UDEN AT SLETTE DATA"
510 SAVE "PROGRAM NAVN"
520 GOTO 10
    
```

GOTO 10 er programmets start, hvis RUN indsættes istedet for GOTO, slettes variablerne (DATA).

Hvis programmet ikke står ved en STOP sætning, kan du starte SAVE med GOTO 510.

8-B

8-B-4

8-B-2

8-B-3

START MED GOTO

9-A

MATEMATISKE FUNKTIONER

9-A-1
FUNKTIONER

Din computer har en række matematiske funktioner, samt deres omvendte (INVERSE) funktioner.

FUNKTION	OMVENDT FUNKTION
SIN (sinus)	ASN
COS (cosinus)	ACS
TAN (tangenus)	ATN
LOG (naturlig logaritme)	EXP
SGN (fortegn)	
ABS (numerisk værdi)	
RND (random)(tilfældige tal)	
INT (integer)(heltal)	
SQR (square root) (kvadratrods)	

9-A-2
LOG
EXP

LOG betegner på dansk $\ln x$
Exp betegner på dansk e^x

Hvor $e = 2.71828182845904523536028747135266249775724709369995$
Den normale \ln -tals logaritme (Log) kan laves således:
LOG x / LOG 10

9-A-3
ØVELSE

10 LET x = 200
20 PRINT LOG x / LOG 10 Resultat = 2.30103

9-A-4
ABS (absolut-
te værdi)

ABS giver den positive værdi af et tal.

9-A-5
ØVELSE

10 PRINT " ABS 5="; 5
20 PRINT " ABS -5="; 5
30 PRINT " ABS 0="; 0
40 PRINT " ABS (2x3-7) ="; 1

9-A-6

Den omvendte funktion til \ln -tals logaritmen (antilog)
10 xx X

9-A-7
ØVELSE

10 PRINT 10 xx(LOG200/LOG 10) 200

9-A-8
PROGRAM

10 REM PROGRAM TIL OMREGNING MELLEM DIN OG ASA
20 PRINT "ASA"
30 INPUT ASA
40 PRINT ASA," OMREGNET TIL ",10x LOG ASA/LOG 10+1,"DIN"
50 PRINT "INDTAST DIN"
60 INPUT DIN
70 PRINT DIN," OMREGNET TIL ", 10xx((DIN-1)/10), " ASA"

SEKTION 9

9-A-1

FUNKTIONER

9-A-2

LOG - EXP

9-A-3

ØVELSE

9-A-4

ABS

9-A-5

Øvelse

9-A-6

Antilog

9-A-7

Øvelse

9-A-8

Program

9-A-9

SGN (fortegn)

9-A-10

Øvelse

9-A-11

INT

9-A-12

Øvelse

9-A-13

RND

9-A-14

Øvelse

9-A-15

SQR (kubikrødder, fjerderødder osv.)

9-A-16

PI

9-A-17

SIN, COS, TAN, ASN, ACS, ATN

9-A-18

Grader, Radianer

9-A-19

SIN, COS, TAN

9-A-20

Øvelse

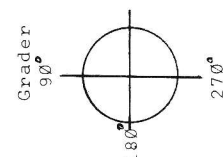
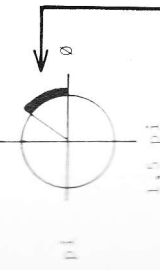
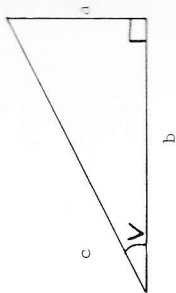
9-A-21

Beregning af højden på en flagstang

MATEMATISKE FUNKTIONER

9-A 0=A SIGN (Fortegn)	Angiver fortegnet af et tal Positivt 1 Negativt -1 Nul 0
9-A-10 ØVELSE	10 PRINT "SGN 5 = "; SGN 5 = 1 20 PRINT "SGN -5 = "; SGN -5 = -1 30 PRINT "SGN 0 = "; SGN 0 = 0 40 PRINT "SGN pi = "; SGN pi = 1
9-A-11 INT	Runder ned til det nærmeste hele tal.
9-A-12 ØVELSE	10 PRINT " INT 7 = "; INT 7 20 PRINT " INT 7.2 = "; INT 7.2 30 PRINT " INT 7.99999 = "; INT 7.99999 40 PRINT " INT -6.001 = "; INT -6.001 50 PRINT " INT (pi x LOG 3) = "; INT(pi x LOG 3)
9-A-13 RND	RND giver tilfældige tal mellem 0 og 1)fra og med 0, til 1, men aldrig 1)
9-A-14 ØVELSE	10 PRINT RND (tilfældige tal) 20 GOTO 10 10 INPUT A 20 PRINT INT (RND x A + 1); " "; 30 GOTO 10 Her får du et tilfældigt helt tal mellem 1 og det du giver i INPUT. Prøv f.eks 6 der giver terningkast .
9-A-15 XX SQR Kubikrødder Fjerderødder 0BV,	Omskrivning fra matematisk udtryk til BASIC udtryk 2^3 2xx3 SQR 7 eller 7**0,5 $\sqrt[3]{7}$ 8xx(1/3) $\sqrt[4]{8}$ 8xx(1/7) $\sqrt[5]{2}$ 2xx(5/7)
9-A-16 PI	Computoren gemmer 10 betydende cifre, men gengiver kun de 8 på skærmen PI 3.14159265358979

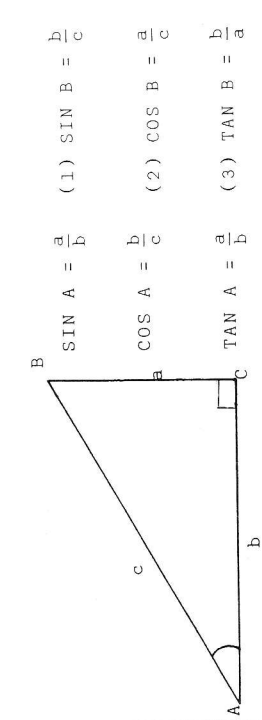
MATEMATISKE FUNKTIONER

9-A 9-A-17 SIN COS TAN ASN ACS ATN	De trigonometriske funktioner regnes i radianer. Vil man f.eks beregne sinus til 30 grader, gøres det på følgende måde: SIN (30/180 x pi) Vil man bestemme en vinkel, når cosinus til vinklen er 0,87 så giver ACS (0.87)/pi x 180 vinklen målt i radianer.
9-A-18 GRADER RADIANTER	Grader og radianer er to forskellige måder, at måle vinkler på.  Cirklen deles ind i 360 grader Cirklen har radius 1, radian tallet for vinklen V er længden af dette stykke 
9-A-19 SIN COS TAN	Der, gælder således: $180 \sim \pi$, $rad 60 \sim \frac{\pi}{3}$ i rad $360 \sim 2\pi$ radianer Funktioner defineres let for vinkler mellem 0 og 90 eller mellem 0 radianer og $\frac{1}{2} \pi$ radian ved hjælp af en retvinklet trekant.  COS V = $\frac{b}{c}$ Hosliggende katete divideret med hypotenusen SIN V = $\frac{a}{c}$ Hosstående katete divideret med hypotenusen TAN V = $\frac{a}{b}$ Modstående katete divideret med hosliggende katete.

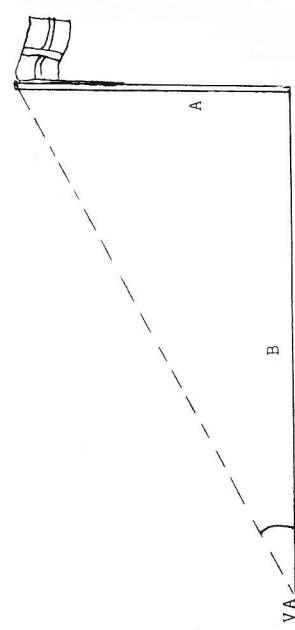
FUNKTIONER

ABS	Den absolutte værdi af et tal uden fortegn (teller-) og giver som resultat den positive værdi af et tal.
ACS	Arccosinus i radianer giver vinkel v 0 $\leq v \leq \pi$ AG fejl hvis operanden ikke er i området -1 til +1.
ASM	Arcsinus i radianer giver vinkel v $-\frac{\pi}{2} < v < \frac{\pi}{2}$ AG fejl hvis operanden ikke er i området -1 til +1.
ATN	Arctangens i radianer giver vinkel v $-\frac{\pi}{2} < v < \frac{\pi}{2}$
CHR\$	Den karakter hvis kode er et tal mellem 0 og 225 afrundet til det nærmeste hele tal IR fejl hvis tallet er uden for området.
COS	Giver cosinus til vinkel målt i Radianer.
EXP	Eksponentialfunktion e^x
INT	Giver heltal af det tal det står foran og runder nedad.
LOG	Betegnes på Dansk $\ln x$ og giver som resultat den naturlige logaritme af x Den normale \log tals logaritme (\log) kan laves således $\text{LOG } x / \text{LOG } 10$ AG fejl hvis x er < 0
PEEK	Værdien af en byte i hukommelsen indenfor adresseområdet 0 til 65535 (16 bits) Værdien fremkommer som et codedtal mellem 0 og 225 (8 bits data)

MATEMATISKE FUNKTIONER



Eksempel:
Beregn højden af en flagstang.



```

10 PRINT " INDTAST AFSTANDEN TIL FLAGSTANG"
20 INPUT B
30 PRINT "INDTAST SIGTEVINKLEN"
40 INPUT VA
50 LET A = B x TAN (VA x PI/180)
60 PRINT "FLAGSTANGENS HØJDE ER = "; A
    
```

9-A

9-A-21
ØVELSE

9-A-21
ØVELSE

SIN	Sinus giver som resultat: sinus af vinkel målt i radianer.
SQR	Giver som resultat kvadratroden af det tal som SQR sættes foran.
TAN	Giver som resultat tangens til vinkel målt i radianer.
USR	Maskincode rutine hvis start adresse sættes efter USR, runder af til det nærmeste hele tal IR fejl hvis adr. ikke er inden for området 0 til 65535.
SGN(X)	Fortegn hvor X kan være positiv, negativ, eller nul. Resultatet vil +1, hvis X er positiv -1, hvis X er negativ 0, hvis X er nul.

FUNKTIONER

CODE	Giver de nr. for den første karakter i en streng. 0 hvis det er en tom streng.
LEN	Angiver antal karakterer i en streng, 0 hvis det er en tom streng.
VAL	Opretter strengen som et algebraisk udtryk og beregner dets værdi IE fejl hvis strengen indeholder syntaks fejl.
STR\$	Giver som resultat en streng som er indtastet med et tal eller værdien af en talvariabel.
PI	pi(3.14159265)
INKEY\$	Læser tastaturet for at undersøge om der er en tast der er aktiveret og giver den karakteren som resultat. Hvis ingen tast er aktiveret giver det en tom streng.
RND	Giver et tal mellem 0 og 1 men aldrig 1

KOMMANDOER

BEEP	Anvendes til tastaturets lydtilbagemelding af forskellig frekvens alt efter hvilken tast du taster, kan bruges som kommando med eller uden linie nr.
CLEAR	Sletter alle variabler.
CLS	Renser skærmen.
CONT	Forsætter programmet hvis det har været stoppet af STOP eller BREAK.
COPY	Bruges i forbindelse med PRINTER og giver et kopi af skærbilledet
DIM A(index) max index 3077 DIM As(index)	Gør plads for det antal variabler som index angiver, i en eller flere dimensioner adskilt af komma. Fejl OM hvis RAM hukommelsen overskrives Gør plads til det antal variabler som index angiver i en eller flere dimensioner adskilt af komma, det sidste index angiver det antal af karakterer som tekststrengen maximum kan indeholde. Fejl OM hvis RAM hukommelsen overskrives.
FAST	Sætter databehandlingshastigheden 4 gange op samtidigt med at skærbilledet forsvinder. Du kan sætte FAST ind i programmet eller skifte til FAST imens programmet afvikles med BREAK-FAST CONT-
FOR	A = start TO slut STEPværdi
NEXT A	FOR efterfølges af en kontrolvariabel på et bogstav (som sletter en e.v.t. anden variabel med samme bogstav som navn) Værdien som kontrolvariabelen tildeles består af en startværdi, en slutværdi samt en stepværdi. Når forskellen mellem start og slutværdi er mindre Stepværdi springer den til den næste linie efter NEXT. Hvis ikke stepværdi er angivet stepper den med værdien 1. Når startværdien har overskredet slutværdien (Grenseværdien) springer den videre til næste linie efter NEXT.

GOSUB	Gosub sender programafviklingen ned i det underprogram som linienummeret efter Gosub angiver og returnerer det første gang RETURN optræder, til det første linie nr efter den samme Gosub.
GOTO	Hopper til det linie nr som efterfølger GOTO
IF THEN	IF(hvis) betingelsen er sant THEN (da) udfører den kommandoer der står efter THEN. Hvis betingelsen er Falsk går den videre til næste linie nr.
INPUT	Standser programafviklingen og venter på data hvis det er data i form af tekst, sætter den Cursorsoren i anførselstegn. INPUT må kun efterfølges af envariabel enten en alm. (INPUT A eller INPUT%) eller en indlæst variabel (INPUT A(index) eller INPUT A(index))
LET	Tildeler variabelen på venstre side af lighedstegnnet værdien fra højre side Let kan undlades da Computeren bruger let som udgangspunkt. Lighedstegnet i en let sætning som bruges til tildeleling er forskellig fra det lighedstegn der bruges til sammenligning i en IF sætning.
LIST	List uden linie nr. lister programmet startende med det første linie nr samtidigt flyttes liniesjernen hertil. LIST med linie nr. Lister programmet fra det linienummer der står efter List samtidigt flyttes liniesjernen hertil.
LOAD"NAVN"	Åbner adgang for signaler fra kasettebåndoptager med det navn som står efter Load.
LOAD""	Åbner adgang for signaler fra kasettebåndoptager uanset hvilket navn programmet er søvet under.

LBIST Bruges i forbindelse med Printer og Giver et kopi af listen.

LBPRINT Starter printeren hver gang kommandoen sendes ind med ENTER.

MUSIC fterfulgt af en Musik streng hvor bogstaverne C D E F G A B REPRESENTERER et oktav, efter et bogstav vil tegnet<satte tonen 1 oktav ned mens tegnet>vil have tonen 1 oktav. Efter et bogstav med evt tegn efter sættes et tal som angiver tonelængden (nodelængden) 0 til 99.

Karakter	Frekvens
c <	1310
[C] <	1383
D <	1471
[D] <	1555
E <	1650
[E] <	1747
F <	1847
[F] <	1958
G <	2070
[G] <	2196
A <	2338
[A] <	2492
B <	2671
[B] <	2871
C	2778
[C]	2955
D	3125
[D]	3318
E	3495
[E]	3693
F	3916
[F]	4167
G	4442
[G]	4740
A	5063
[A]	5424
B	5842
[B]	6333
C	6899
[C]	7500
D	8163
[D]	8899
E	9705
[E]	10590
F	11663
[F]	12888
G	14277
[G]	15844
A	17599
[A]	19568
B	21784
[B]	24384

Til pause indsættes et "." og længden af pausen angives på samme måde som med tonerne med tal fra 0 til 99. Musikstreng sættes som Tekststreng i anførselstegn MF fejl hvis musikstrengen ikke er korrekt.

NEW Sletter program og alle variabler op til Ramtop.

NEXT Efterfølges af kontrolvariablen bogstav hvis værdi den beregner iht. stepværdi for til at sende programafviklingen til næste linie når grænseværdien er overskredet.

PAUSE Standser programudførelsen i en bestemt tid 50 svarer til 1 sek. Hvis programmet er i FAST vil skærm billedet komme frem i den tid Pausen varer.

PLOT Afsætter er plottepunkt i overensstemmelse med den vandrette X (max 63) og lodrette Y (max 43) Korrelinat, der adskilles af et komma. Fejl IR hvis PLOT 63,43 ikke overholdes.

PRINT Giver adgang til at printe i skærmens 22 linier og 32 kolonner Print kommandoen kan genoprettes når den har været brudt, med semikolon eller komma, semikolon virker umiddelbart på samme linie mens komma kun begynder printning ved kolonne 0 eller 16.

Hvis numeriske udtryk er større end 10⁹ eller mindre end 10⁻⁹ udskrives det i eksponentiel notation, E + eller -

Print kommandoen udskrives de 8 første cifre og ændre resten til nuller, som ikke udskrives hvis det er efter punktum.

PRINT AT Print positionen sættes efter AT i form af to tal adskilt af et komma, det første tal angiver hvilken linie nr. (begyndende i øverste venstre hjørne med 0,) det andet angiver hvilken kolonne nr (begyndende i øverste venstre hjørne med 0 og slutter til højre med ,31) fejl IR hvis tallet 31 overskrides. Hvis du forsøger at printe i linie 22 og 23 (som er forbeholdt indskrivning) melder den fejl SF. Fejl IR hvis tallet er større end 23.

PRINT TAB
Efterfulgt af et tal der angiver tabulator positionen, hvis tallet er større en 31 divideres det med 32 således at kun det der er til rest angiver tabulatorposition.

RAND
Giver et tilfældigt tal som den henter i en tællerreds i Computeren. Dette øjeblikkelige tal bruges som starttidspunkt for RND på den måde bliver RND tilfældigt når kommandoen RAND sættes ind foran.

RAND
Efterfulgt af et tal f.eks RAND 10 vil du selv kunne bestemme starttidspunktet og vil dermed blive det samme åvrigt RND starter.

RETURN
Tager GOSUB sætningens linie nr. og lægger 1 til hvorved at programafviklingen forsætter ved den første linie efter GOSUB.
Fejl RG ved Return uden GOSUB-

RUN
Starter ved linie nr nul, sletter samtidigt alle variabler.

RUN
Efterfulgt af et linie nr, starter proafviklingen ved dette linie nr slettes også variabler.
Brug GOTO hvis undgå at slette variablerne.

SAVE
Overfører program og variabler til båndoptager under det navn der skrives i anførselstegn efter SAVE. Save kan sættes ind i programmet, dog ikke i underprogrammer.

SCROLL
Ruller skærm billedet 1 linie op således at der bliver en tom linie i bunden og glider en ud foroven.

SLOW
Sætter computeren til normal hastighed herved kommer skærm billedet igen. Tast BREAK SLOW ENTER CONT ENTER

SOUND

Frembringer en lyd ud fra en frekvens på 32500 Hz og med en svingningstid på 65 ms. Det første tal efter SOUND angiver tonehøjden og sættes inden for 8 bits tallet 0 til 255 adskilt af et komma angiver det andet tal tone-længden som sættes til et 16 bits tallet 0 til 65535 Fejl IR hvis tallene kommer uden for dette område.

STOP

Standser programafviklingen og melder ST IN linie nr. CONT vil den starte programmet samtidigt med den renser skærmen.

TEMPO

Sættes som et 8 bits tal 0 til 255 efter TEMPO, der giver en afvikling af Musikprogrammet svarende til tallet efter TEMPO x 3.94 ms.
Fejl IR hvis tallet er uden for området 0 til 255.
Hvis ikke TEMPO indsættes i programmet udføres Musikprogrammet i TEMPO 25 ca 98.5 ms.
RESET sætter computeren tilbage til normal tempo (TEMPO 25).

UNPLOT

Sletter et punkt på det koordinatpunkt som er angivet efter UNPLOT.

FJÆLLELDING/TILBAGEMELDING

	Betydning	Situationer
04	Afslutning O.K. Eller spring til linienummer større end noget eksisterende.	Alle
05	NEXT uden FOR NEXTkommando indskrevet uden tilhørende FOR-sætning.	NEXT
06	Udefineret Variabel Variabel er brugt uden forgående LET-sætning. Eller indiceret variabel brugt før den tilhørende DIM-sætning.	Alle
08	Indiceret variabel udenfor definitionsområde. Hvis tallet er større end 65535 fås fejlrapport IR. Hukommelsen i maskiner er fyldt op	Indicerede Variabler
09	Skærm overskredet. Tast CONT, det renser skærmen og fortsætter programmet.	Alle
0V	Tal for stort. eller variabel indeholdende tal større end 10	PRINT, LIST
0G	RETURN uden GOSUB RETURN uden forudgående GOSUB-sætning.	Al aritmetik
11	Illegal INPUT. INPUT brugt uden for program.	RETURN
0T	STOP-kommando udført	INPUT
0G	Ugyldig argument i funktionerne: Det anvendte heltal er uden for maskinens definitionsområde.	STOP
1R	Teksten i den tekststreng, hvorpå VAL argumentet bruges giver et ugyldigt numerisk udtryk.	SQR, LOG, ASN, ACS RUN, RAND, POKC, DIM, GOTO, GOSUB, LIST, LLIST, PAUSE, PLOT, UNPLOT, CHR\$, PEEK, USR.
1E	Program afbrudt af BREAK Program forsøgt savet uden navn Musikstreng formet ukorrekt	VAL BREAK SAVE MUSIC

PRIORITETSÆKKEFØLGE

PRIORITETS- RÆKKE	PRIORITETSÆKKEFØLGE
	PRIORITETSÆKKEFØLGEN er den rækkefølge computeren udfører sine funktioner og beregninger i.
	FUNKTIONERNE: Prioritet 12 - tekststrengfunktionerne - 11 - talfunktionerne
	ARIYMETRISKE OPERATORER: Symbol Betydning Prioritet 10 xx Potens opløftning - 9 - Fortegnsminus - 8 +/ Multiplikation og division - 6 +- Addition og subtraktion
	Relationsoperatorene Relationsoperatorene har alle prioritet 9. Symbol Betydning = \forall Lig med > \forall Større end < \forall Mindre end > \forall Større end eller lig med < \forall Mindre end eller lig med Forskellig fra
	Logiske operatorene Symbol Betydning Prioritet 4 NOT (ikke) NOT relationen er sand når relationen er falsk. Prioritet 3 AND (og) NOT relationen er falsk, når relationen er sand. Prioritet 2 OR (eller) AND relationen er sand, når begge relationer er sande. OR relationen er sand når den ene af relationerne er sand.

FORWARDED-CODE	.	=82	.	=83
"	.	=84	.	=85
"	.	=86	.	=87
"	.	=88	.	=89
"	.	=90	.	=91
"	.	=92	.	=93
"	.	=94	.	=95
"	.	=96	.	=97
"	.	=98	.	=99
"	.	=100	.	=101
"	.	=102	.	=103
"	.	=104	.	=105
"	.	=106	.	=107
"	.	=108	.	=109
"	.	=110	.	=111
"	.	=112	.	=113
"	.	=114	.	=115
"	.	=116	.	=117
"	.	=118	.	=119
"	.	=120	.	=121
"	.	=122	.	=123
"	.	=124	.	=125
"	.	=126	.	=127
"	.	=128	.	=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

A	=166	B	=167	AT	=212	TAB	=213
C	=168	D	=169	**	=214	OR	=215
E	=170	E	=171	AND	=216	<=	=217
F	=172	H	=173	>=	=218	<	=219
G	=174	I	=175	TEMPO	=220	MUSIC	=221
H	=176	J	=177	SOUND	=222	DEEP	=223
I	=178	K	=179	NOBEEP	=224	LPRINT	=225
J	=180	L	=181	LLIST	=226	STOP	=227
K	=182	M	=183	SLOW	=228	FAST	=229
L	=184	N	=185	NEW	=230	SCROLL	=231
M	=186	O	=187	COMT	=232	DIM	=233
N	=188	P	=189	REM	=234	FOR	=235
O	=190	Q	=191	GOTO	=236	GO SUB	=237
P	=192	R	=193	INPUT	=238	LOAD	=239
Q	=194	S	=195	LIST	=240	LET	=241
R	=196	T	=197	PAUSE	=242	NEXT	=243
S	=198	U	=199	POKE	=244	PRINT	=245
T	=200	V	=201	PLOT	=246	RUN	=247
U	=202	W	=203	SAVE	=248	RAND	=249
V	=204	X	=205	IF	=250	CLS	=251
W	=206	Y	=207	UNPLOT	=252	CLEAR	=253
X	=208	Z	=208	RETURN	=254	COPY	=255
Y	=210	CODE					
Z		LEN					
		COS					
		ASN					
		ATN					
		EXP					
		SQR					
		ABS					
		USR					
		CHR\$					
		VAL					
		SIN					
		TAN					
		ACS					
		LOG					
		INT					
		SGN					
		PEEK					
		STR\$					
		NOT					

APPENDIX E

Hvis computeren er ude af drift, og du overvejer at returnere den for reparation, så prøv først at checke følgende punkter:

Følg kontrol liste

Pkt.	Symptom	Årsag	Middel
1.	Strømforsyneren: LED lyser ikke, intet billede på skærmen, ingen 'Beep'	1. Strøm ikke tilsluttet 2. Forbindelse er ikke i orden.	1. Check strøm kilden. 2. Forbind ledningerne, og indsæt stikket i AC ordentligt.
2.	Der er intet billede, og der høres ikke noget 'Beep'	Computeren er ikke ordentligt opstillet.	1. Tryk på RESET 2. Hold ENTER knappen nede og tryk på RESET knappen. 3. Sluk computeren og tænd igen.
3.	Et 'Beep' høres, men der er ingen billede (CURSOR) på TV-skærmen	1. Den rigtige kable er ikke fundet 2. Antennestikket passer ikke 3. TV ledningen er sat i monitor	1. Juster TV-kablet. 2. Anvend en modstandsform for at få antennen indsat ordentligt. 3. Forbind TV-kablet med det korrekte stik
4.	Billedet på skærmen forstyrres.	TV modtageren er ikke indstillet korrekt	Juster indstilling vandret/lodret.
5.	Karakterer fra tastaturet kan ikke indsættes	1. En taster sidder fast. 2. Et fejlagtigt program er kommanderet RUN	1. Check alle taster. 2. Se pkt. 6
6.	Efter programmet er sendt ind (RUN) er computeren ude af kontrol (BREAK funktion virker ikke)	1. Et fejlagtigt program er sendt afsted.	1. Tryk RESET og check programmet 2. Hvis RESET ikke kan få computeren til at køre, så hold ENTER og tryk RESET.

7.	Efter indlæsning af programmet fra båndoptager til computer - kan computeren ikke køre	1. Et fejlprogram er indlæst	1. Se pkt. 6
8.	Et program kan ikke indlæses.	1. Kablerne mellem computeren og båndoptageren er forbundet rigtigt 2. Hukommelsen er ikke stor nok til at optage programmet.	1. Check at båndoptager stikkene er i orden og at de er forbundet korrekt. 2. Anvend en ekstra RAM pack for at udvide hukommelsen.
9.	Når RAM pack påmonteres computeren kan den ikke fungere	1. RAM pack er ikke forbundet korrekt 2. RAM pack virker ikke.	1. Check at RAM pack er påmonteret korrekt. 2. ADVARSEL: Hver gang du vil tilslutte/afkoble RAM pack - skal du slukke for computeren, hvis du ikke gør det ødelægger du bånd RAM pack og computer.
10.	Støjstriber på skærmen.	1. Transistorradio eller anden elektronisk enhed udsender støj.	1. Undgå at have elektroniske enheder i umiddelbar nærhed af computeren.
11.	Manglende funktion af tasterne	1. Joystick kan i særlige tilfælde begrænse tastaturet.	1. Afmonter Joystick efter brug.
12.	Fejl på Loadning	1. Båndoptageren i særlige tilfælde de give net støj.	1. Anvend batterier i båndoptageren.

