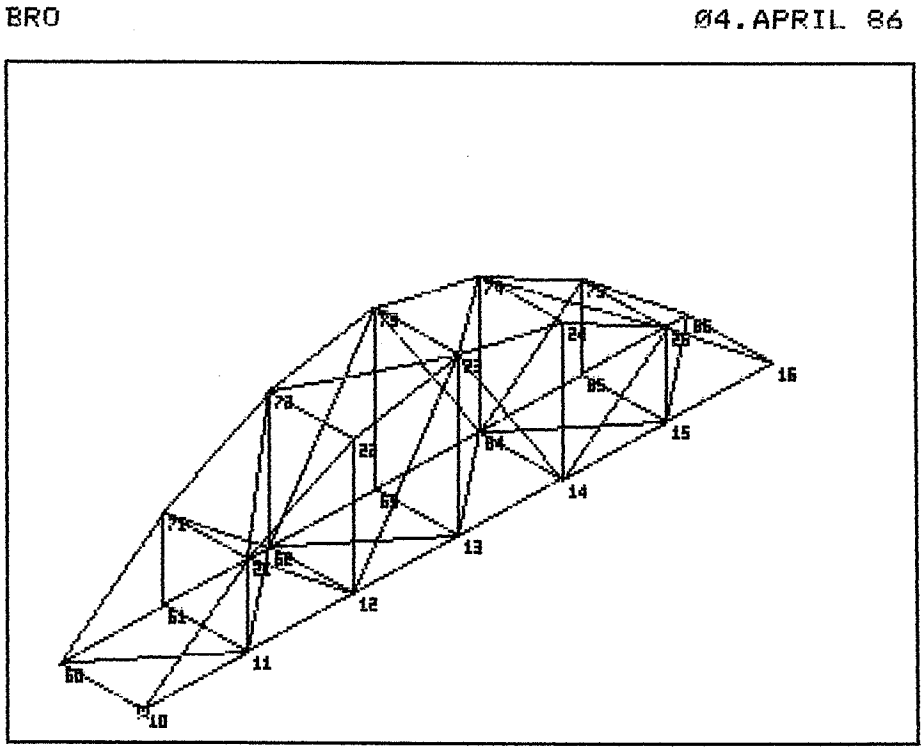
INSTITUTTET FOR HUSBYGNING

Rapport nr. **177**



JOHS. F. MUNCH - PETERSEN

RUMGITTER PÅ QL

EN INTRODUKTION TIL GRAFIK PÅ QL - DATAMATEN

Den polytekniske Læreanstalt, Danmarks tekniske Højskole

Lyngby **1986**

I N D H O L D S F O R T E G N E L S E

FORORD 5

INSTRUKTION 7

INDLEDNING 21
- Tegneprogrammer på PC'ere 21
- IFH og tegneprogrammer 23
- IFH's "programudbud" primo 1986 24

BOOT 27
- Struktur- og procedureoversigt 30

HOVEDPROGRAM 33
- isometri_tegning 34
- LOAD, SAVE 42
- INPUT-løkken 44
- Hjælpefunktioner (DIVERSE) 48
- opret_linie 54
- talput 58
- plotter (SERVOGOR 281) 60
- fejlkontrol 64
- c80routiner 73

EKSEMPEL BRO 75

TILLÆG 81
- ROLAND DXY880 plotter 82
- tput (forbedret talput) 85
- nymem (genopfrisk memory) 88
- SKÆRMEN 89
  - Koordinatsystemer 90
  - Vinduer, kanaler, farve, BORDER 91
  - Pixler 92
  - Grafik, SCALE, LINE 93
  - CURSOR 94
  - Karakterudskrift 95
  - OVER 0, OVER 1 95
  - OVER -1 96
- DANSKE OG EUROPÆISKE BOGSTAVER 97
- isometri_vindue 98
- vvsfount (vvstegn) 100
- Skærmbilledet, 32768 bytes 102
- SAVE, LOAD, LIST, DIR, STAT, WSTAT 103
- ALTERNATIVT LOADEPROGRAM 104
- PRINCIPIELT LOADEPROGRAM 105
- PRINTEREN 106
  - Specialtegn 107
- ERROR 108

FORORD

Dette notat, RUMGITTER PÅ QL, beskriver programmet "isometri_tegning".

Det omfatter fire afsnit; et til den almindelige bruger, tre til "programmører".

INSTRUKTION, der beskriver programmets brug. Benyttes programmet, som det foreligger, eller er det allerede omstillet til et bestemt formål, behøver brugeren kun at læse dette afsnit.

INDLEDNING, til hovedprogrammet, der omtaler "tegneprogrammer" til små datamaters PC'ere,

GENERELT i relation til anvendelse på bygningsingeniør-opgaver,
SPECIELT i relation til INSTITUTTET FOR HUSBYGNING (IFH, DTH).

Det fremgår, at jeg mener, at KOMBINATIONEN af beregninger og grafik er vigtigere for bygningsingeniører end "rene" tegneprogrammer.

HOVEDPROGRAM, herunder BOOT og EKSEMPEL BRO, der beskriver programmets PROCedurer, normalt med programudskrifter på venstre side, og kommentarer på højre side i hvert opslag.

Der forudsættes kendskab til SUPERBASIC, til QL-datamatens BRUGERVEJLEDNING og en vis forudgående øvelse i at programmere QL'en.

TILLÆG, der uddyber BRUGERVEJLEDNINGENS beskrivelse af de grafiske PROCedurer. Mange læsere vil antageligt have mere udbytte af dette notat ved først at læse TILLÆG, sideløbende med studier af VEJLEDNINGENS "NØGLEORD" og "BEGREBER".

Notatet kan også ses som en INTRODUKTION TIL QL's GRAFISKE POTENTIEL,

DELS ud fra programmet og kommentarerne hertil,
DELS ud fra TILLÆGGETS eksempler på anvendelse af grafikken.

PROGRAMMET "isometri_tegning" er kun et eksempel på en begyndelse til et program, der lever op til de af mig i INDLEDNING rejste krav.

Som det ser ud nu, kan de "fysiske legemer", gitterstængerne/rørene, tillægges egenskaber (farve/form, materiale, dimension). På samme måde har "rumligere" legemer, fx bygningskomponenter, ikke blot koordinater til hjørnerne, men også egenskaber, der senere skal indgå i beregninger af fx styrke, produktion, leverance, pris.

Programmet er også relativt brugervenligt, inden for de - i dag - givne muligheder for kontrol med såvel INPUT som tilknyttet HARDWARE.

KONFIGURATION: QL-512 K, SHINWA CPA80S-printer og/eller QL-printer, SERVOGOR 281 (BBC) plotter (ROLAND DXY-880, se TILLÆG), CST-INTERFACE til floppy-disk, CARE-QJUMP-TOOLKIT2. Printer og plotter til SER1 hhv SER2. Herudover BUVEX DATA TEAM APS's skærm-til-printer-rutine, SCOPY/QCOPY.

Notatet er tung kost. Mange "små" kommandoer har deres betydning ! Fx er brugen af BORDER med/uden farve, OVER 0,1,-1 lumske. Meget kunne være gjort mere elegant, se kommentarerne.

15. april 1986.

Johs. F. Munch-Petersen

INSTRUKTION

Isometri_tegning

Dette program har en begrænset værdi i sig selv. Det er tænkt som et grundlag, der skal udvides, justeres til specifikke formål.

Det kan eksempelvis benyttes til visuel kontrol af, at tredimensionale koordinatsæt for rumgitre, rørsystemer etc. er rimeligt rigtige, før data benyttes til beregninger af dimensioner, spændinger etc.

Det kan tænkes anvendt som dagligt input-medie for data, der senere skal indgå i en større sammenhæng på en større datamat.

Det kan også bruges til hurtig udtegning af/revision af rumlige afbildninger af tre-dimensionale sammenhænge, visuel analyse/illustration af forsøgsresultater etc.

Programmet tegner i en 3×120^0-isometri, men kan ændres til at tegne i andre afbildningsformer.

Drejning af figurer er en anden facilitet, der kunne indbygges.

INPUT er punkternes numre og deres koordinater, absolutte og relative. Punktnumre kan vælges udskrevet på skærmen.

Til et punktnummer hører dettes tre-dimensionale koordinater, 2-dimensionale skærmkoordinater, 2-dimensionale plotter-koordinater og antallet af forbindelser (linier) til punktet. Flere oplysninger kunne tilføjes.

Til en linie hører liniens FRA-nummer, TIL-nummer, Δx, Δy, Δz, farve (symbol for anvendelsen, visuelt og/eller praktisk), materiale, dimension. De tre sidste kan ændres og udvides i antal ved simple ændringer.

Alt input kontrolleres i størrelse, indenfor fastlagte grænser.

Linier kan oprettes, revideres, slettes.

Skjulte fejl kan fejlkontrolleres.

OUTPUT er printer-kopi af skærmbilledet, 5-farve plot af tegningen (med/uden punktnumre og dimensioner), printer-udskrift af data, data-SAVE m.v.

Brugeren kan (hvis noget synes at gå galt, hvis han er i tvivl, etc.) til enhver tid stoppe med CTRL + mellemrum.

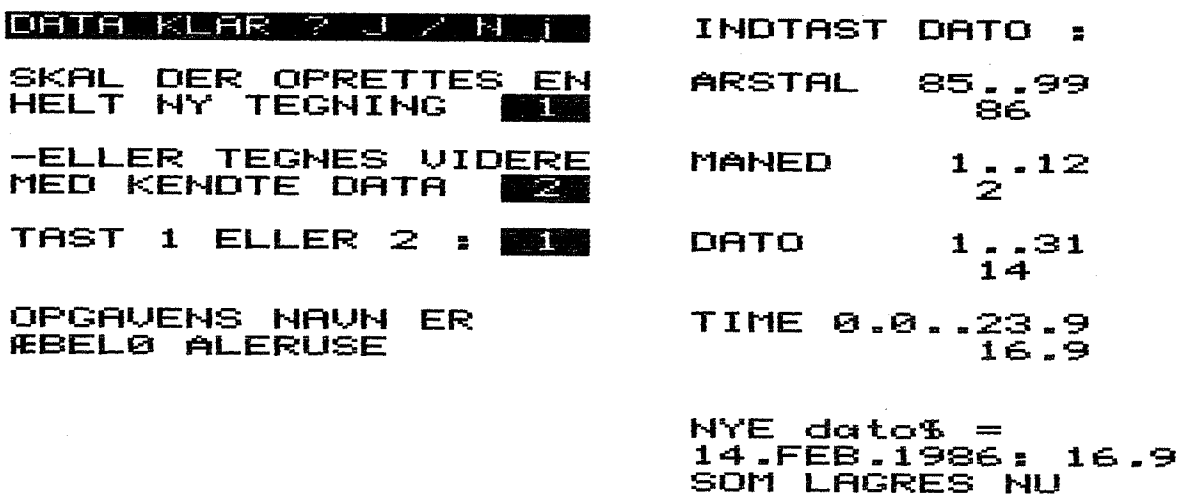
Han kan genstarte med "RUN" - eller (hurtigere) med "indtast 1", hvis skærmbilledet ikke skal retableres.

Brugeren kan, ved at taste "i", når INPUT af tal forlanges, få en række instruktions-skærmbilleder med simpel retur til programmet.

Disse vises og kommenteres på de følgende sider.

I N S T R U K T I O N S - R E S U M E , SKÆRMBILLEDER

SKÆRMBILLEDE VED OPRETTELSE AF NY TEGNING



SKÆRMBILLEDE VED LOADING AF EXISTERENDE FIL



PROGRAMSTART.

Programmet består af to dele, et indlæsningsprogram, "isometri_boot" og et hovedprogram, "isometri_tegning" plus en række maskinkoderutiner.

Program såvel som DATA-filer benytter samme floppy-disk-drev, normalt FLP1. Ønskes FLP2, kan de to BASIC-programmer omstilles i hver sin indledende programlinie (flp$, DATA_USE, PROG_USE, DEST_USE).

1. Programmet startes ved LOAD+RUN: kommando LRUN FLP1_isometri_boot. Hvis "isometri_boot" omdøbes til "boot"(RENAME isometri_boot, boot), vil programmet starte selv. Disketten skal indsættes umiddelbart efter, at QL'en er tændt (prøvebillede ala græsmark). Indlæsningen starter, når der tastes "F1" (monitor). Programmet kan ikke benyttes på TV.

2. Under boot meddeles, at der indlæses (LBYTES) en række maskinkoderutiner, "qltegn" sikres pænere Æ,Ø,Å end QL's egne tegn. "vvsfount" har de små tal, der benyttes på tegningen. "RAM" og "ptpatch" er beskrevet i kommentarerne til programmet. "SCOPY" ("QCOPY") har BUVEX DATA APS copyright på.

3. Til slut spørges, om brugeren vil gå direkte til programmet eller vil se instruktionen først. TAST "i"/"p".

4. Vælges "instruktion", skal der tastes mellemrum (SPACE) for hvert billede.

5. Nu indlæses hovedprogrammet på ca. 2 minutter. Et digital-ur viser tiden.

6. Selve programmet indleder med at spørge, om der benyttes CPA80S- eller QL-printer. TAST "c"/"q". På IFH har vi begge printere. Spørgsmålet kan fjernes, og styrevariablerne sættes en gang for alle.

7. Programmet undersøger, om der faktisk sidder en diskette i FLP1. Brugeren skal jo på dette tidspunkt eventuelt udskifte programdisketten med DATA-disketten (DATA KLAR J/N). Se hosstående skærmbillede. Fil-navnet skal angives. Programmet afviser at oprette såvel en ny tegning under et navn, der eksisterer på FLP1, som at indlæse en (kendt) fil under et ikke-eksisterende navn.

NB. VED FEJL, fx som angivet under 7, vil programmet afvise INPUT og returnere til (nyt) INPUT. Under 7 udskrives fejl/fundet fil. Dette program udskriver ikke altid bruger-fejl, men afviser altid INPUT, der ikke er korrekt, forsåvidt fejlen kan opdages af programmet, fx INPUT af tal uden for fastlagte grænser. Programmet kan derimod ikke checke, om der er sat papir og penne i plotteren, når brugeren svarer ja til "PLOTTER KLAR?".

8. Sluttelig indtastes DATO: år (to eller fire cifre), måned (1..12), dato (1..31), men 31. FEBRUAR giver ikke fejl og klokkeslet, normalt i timer, men om ønsket med een decimal (à 6 minutter). Hele dato-indtastningen skal godkendes/kasseres (nyt INPUT).

```
isometri_tegning  MED LODRET y-AKSE OG HELTALS-KOORDINATER.
                    UDTEGNING PA SKÆRM, PRINTER & PLOTTER.

NY TEGNING       ( PROGRAMMET SÆTTER TEGNINGERNES skalaer ).
MIN. & MAX. KOORDINATERNE   SAMT STARTPUNKTETS KOORDINATER.

EN EXISTERENDE TEGNING TEGNES AUTOMATISK UD FRA DATASÆTTET.

LINIER ANGIVES VED "FRA-PUNKTETS" KOORDINATER   ( HELTAL ).
                OG "TIL-PUNKTETS" KOORDINATER,
             ELLER "TIL-PUNKTETS" AFSTANDE  ( xx, yy, zz ).

PUNKTER  NUMRENE 1...99 ACCEPTERES ( MAX: 8 FORBINDELSER ).
TASTES ET PUNKT-NUMMER, UDSKRIVES PUNKTETS KOORDINATER / 0.

ET PUNKT KAN VÆRE ET NYT PUNKT: INDTAST  x,y,z   +   ENTER
         ELLER ET EXIST. PUNKT: ACCEPTER x,y,z : KUN ENTER
         ET "TIL-PKT." ( IKKE ET "FRA-PKT" ) KAN REVIDERES.


INSTRUKTIONS-RESUME 1
```




```
NY LINIE eller
REVISION
        < SPACE >

SLETNING
          < ESC >

INPUT
 tal + < ENTER >

REVISION
 tal + < ENTER >

UÆNDREDE DATA
 KUN   < ENTER >

"DIVERSE"
OMFATTER:

SAVE / UDSKRIFT
AF AKTUELLE DATA

PRINT / PLOT
AF SKÆRMBILLEDET
```

INSTRUKTIONS-RESUME 1

beskriver oprettelse/revision af punkter og linier
Se også instruktions-resume 2 og de følgende skærmbilleder.

Indtastningen indledes ved valg af ny linie eller revision af eksisterende ved tastning af "SPACE" (mellemrum).

Dette og de følgende skærmbilleder er indeholdt i "instruktion", som kan kaldes under "boot", men også fra selve programmet. Når der i hovedprogrammet spørges om et tal (dog ikke under Dato), vil indtastning af "i" medføre, at skærmbilledsekvensen fremvises. Der returneres til hovedprogrammet, med uændrede DATA.

OPRETTELSE/REVISION AF LINIER

0. Denne instruktion resumeres pag. 16-17 med tilhørende skærmbilleder.

1. Der benyttes isometri med 120^0 (60^0) mellem akserne. Jeg har benyttet lodret y-akse i overensstemmelse med en række populære EDB-programmer. Som en advarsel til ingeniører udtegnes x-y-z-logo'et på tegningerne på skærmen (ikke på printer og plotter).

2. En ny tegning kræver angivelse af minimums- og maksimumskoordinaterne, således at programmet kan skalere til maximal udnyttelse af tegningsfeltet (se også pag. 14). Plotter-tegninger vil ligeledes blive skaleret til maximal udnyttelse af plotter-arealet. Det er i en række tilfælde værd at overveje, at lade programmet vælge mellem kendte, mulige skalaer (der bør udskrives) på plotteren. Plotter-rutinerne skal alligevel ofte tilpasses brugerens behov/den aktuelle plotter-type.

3. Linier angives ved koordinaterne til liniens endepunkter, og ved egenskaber ved linien (se de følgende sider, bl.a. pag. 16). Programmet accepterer kun heltalskoordinater mellem min. og max. værdierne.

   Minimum og maximum skal ligge i intervallet -20000 til +20000, og forskellen mellem minimum og maksimum skal være mindst 100.

4. Punktnumrene 1..99 accepteres, udskrives 2-cifrede (01..99). Der kan højst gå 8 linier fra/til et punkt.

5. Linier oprettes ved indtastning af FRA-punktets nummer og (absolutte) koordinater. Alternativt kan TIL-punktet defineres ud fra FRA-punktet ved de relative koordinater (afstande, med fortegn!).

6. Linier oprettes/revideres ved
   at taste SPACE (mellemrum)
   at indtaste FRA-punktets nummer (+ENTER)

7. Eksisterende FRA-punkt: Koordinaterne udskrives, med grønt.
   Nyt FRA-punkt: Der udskrives "-". Indtast koordinaterne.

8. Der fortsættes til TIL-punkt.
   Tast SPACE samt TIL-punktets nummer (+ENTER).
   Se pag. 16-17, hvor valg af absolutte/relative koordinater beskrives.

9. Eksisterende TIL-punkt: Koordinaterne udskrives, med grønt.
   a. Accepter det udskrevne med ENTER.
   b. Revision: Tast ny koordinat +ENTER.
   Nyt TIL-punkt: Der udskrives "0".
   Tast koordinater +ENTER.

NB Det er således kun TIL-punkter, der kan revideres.

10 ff. Se næste opslag.

REVISION AF LINIERS KOORDINATER, FARVE, MATERIALE OSV:
INDTAST NYE DATA + ENTER. ACCEPTER EXIST. DATA: KUN ENTER.
REVISION AF FLERE LINIER SKAL FØRES HELT TIL BUNDS. CHECK.
DEN NYE LINIE TEGNES, OG DEN GAMLE FJERNES VED RENTEGNING.

SKÆRMEN: NYE NUMRE SKRIVES. EXIST.NUMRE KAN VÆLGES SKREVET.
DER KAN VÆLGES TRE STREGFARVER PA SKÆRMEN. NUMRE ER HVIDE.

PA PLOTTER KAN VÆLGES EN FJERDE FARVE TIL DIMENS. OG RAMME.
- LINIEFARVER TIL PLOTTER VÆLGES UAFHÆNGIGT AF SKÆRMFARVER.
- VÆLG UDSKRIFT AF NUMMER / DIMENSION / BADE - OG / INTET.

SLETLINIE FJERNER EN LINIE OG EVT. ET PUNKT < ESC >

DIVERSE OMFATTER: SORT-HVID PRINTER-TEGNING,
4-FARVET PLOTTER-TEGNING,
UDSKRIFT OG LAGRING AF DATA,
NYT DATASÆT I DETTE PROGRAM.

SE OGSA PROCEDURERNE c80dir, c80list,
c80program.

INSTRUKTIONS-RESUME 2



NY LINIE eller
REVISION
< SPACE >

SLETNING
< ESC >

INPUT
tal + < ENTER >

REVISION
tal + < ENTER >

UÆNDREDE DATA
KUN < ENTER >

"DIVERSE"
OMFATTER:

SAVE / UDSKRIFT
AF AKTUELLE DATA

PRINT / PLOT
AF SKÆRMBILLEDET

INSTRUKTIONS-RESUME 2

Fortsat fra forrige opslag.

10. Til en linie hører, udover endepunkternes koordinater, også liniens egenskaber. I det foreliggende program tre egenskaber, der indtastes umiddelbart efter TIL-punktets koordinater.

11. FARVE angiver liniens farve: 2=rød, 4=grøn, 7=hvid, med sort skærm. Farven kan benyttes som symbol for en egenskab, fx til at angive, om den stang, som linien repræsenterer på skærmen, er et L-profil eller et I-profil - eller som symbol for liniens anvendelse. (Fremløb/returløb i et centralvarmeanlæg, se vvs_tegning).

12. MATERIALE og DIMENSION har minimums- og maximumsværdier, der her er 0...4, hhv. 5...1000. Tastes "0" for MATERIALE, overspringes DIMENSION (der også sættes til 0).

13. Ved revision af en linie, udtegnes den nye, men den gamle bliver stående, således at revisionens forløb kan bedømmes. Af og til bør der rentegnes: Flyt med "↓" (pil nedad) CURSOR'en til "DIVERSE" og tast "ENTER". I "menuen" for "diverse" kan vælges RENTEGNING (Tast 2, smlgn. pag. 18), med eller uden udtegning af punktnumre.

14. Rentegning "uden numre" kan give en pænere printerkopi - eller gøre yderligere revision af tætliggende punkter/linier lettere. Bortset herfra er det mere praktisk at have numrene udtegnet på skærmen. Nye linier får deres numre udskrevet, selv om udgangspunktet er en tegning uden numre.

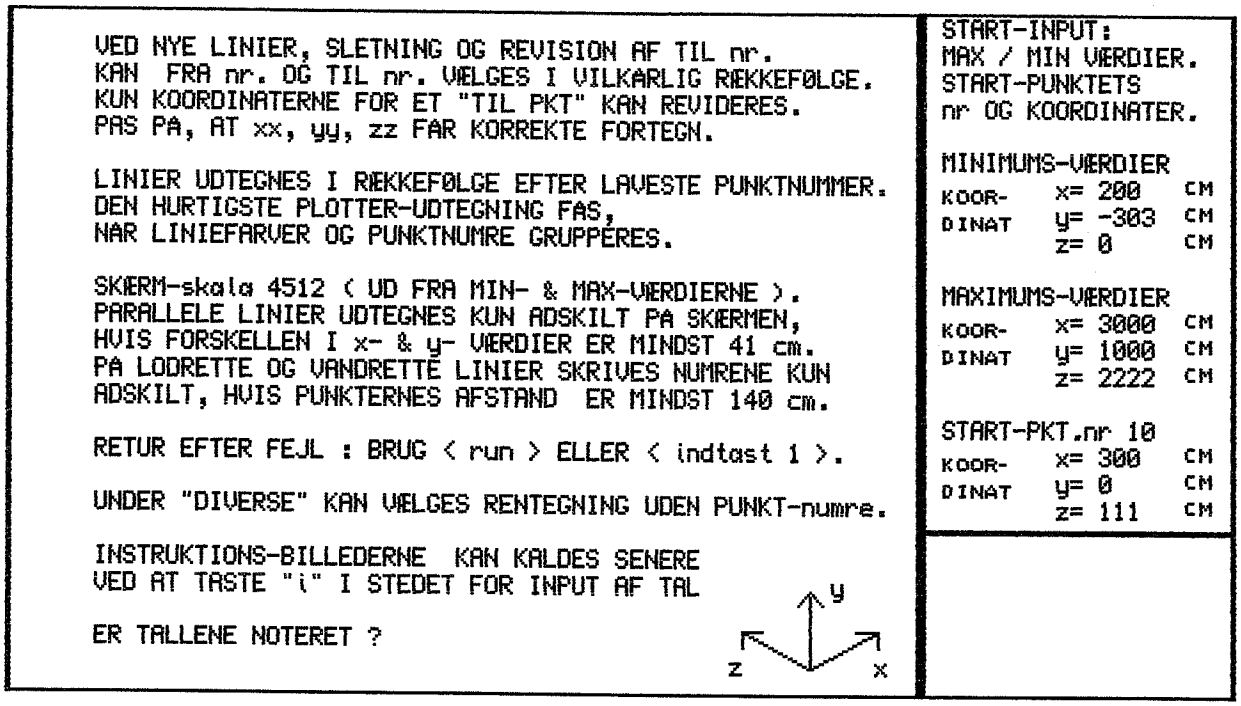
15. Der henvises til pag. 18 ff vedrørende SLET LINIE, DIVERSE, PRINTER, PLOTTER, DATA-UDSKRIFT og DATA-LAGRING.

16. I tvivlstilfælde: Tast "i" i stedet for et tal og se instruktionsbillederne igen.

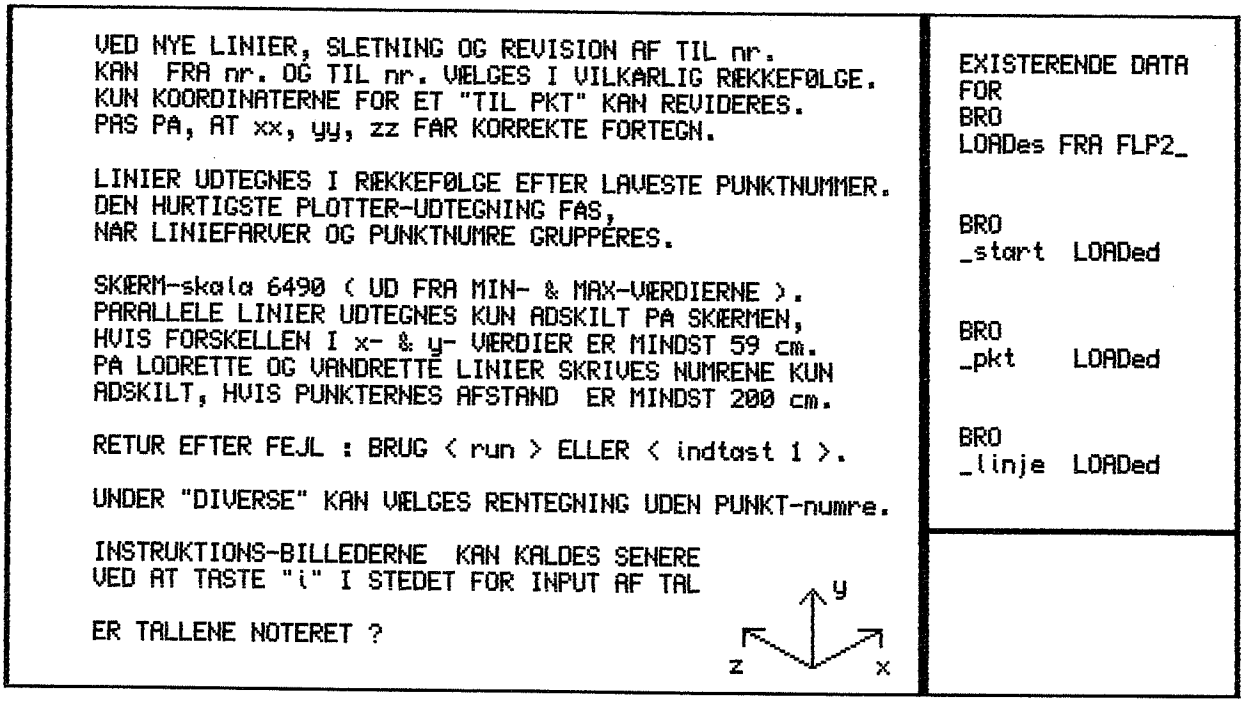
17. Går noget galt, kan programmet standses med "CTRL" og "SPACE". Indtastfeltet gen-startes med "indtast 1" (+ENTER).

18. Skærm og indtastfelt gen-startes med "RUN" (+ENTER).

19. Hverken 17 eller 18 medfører tab af DATA. Standses programmet med CTRL+SPACE efter indtastning af alle seks koordinater, er linien oprettet, med DEFAULT-værdierne 2,0,0 for farve, materiale og dimension/tidligere definerede værdier for denne linie. Stoppes midt i en koordinat-indtastning, annulleres linien.



Indtastning af minimums- og maximumskoordinater til en ny tegning. Skala og minimumsafstande for tætliggende punkter og linier udskrives.



Tilsvarende skærmbillede efter indlæsning af data-sættet for "BRO". Tallet 59 (xxmin) gentages ved linie-INPUT, se pag. 17.

PROGRAMMETS START-INFORMATION.

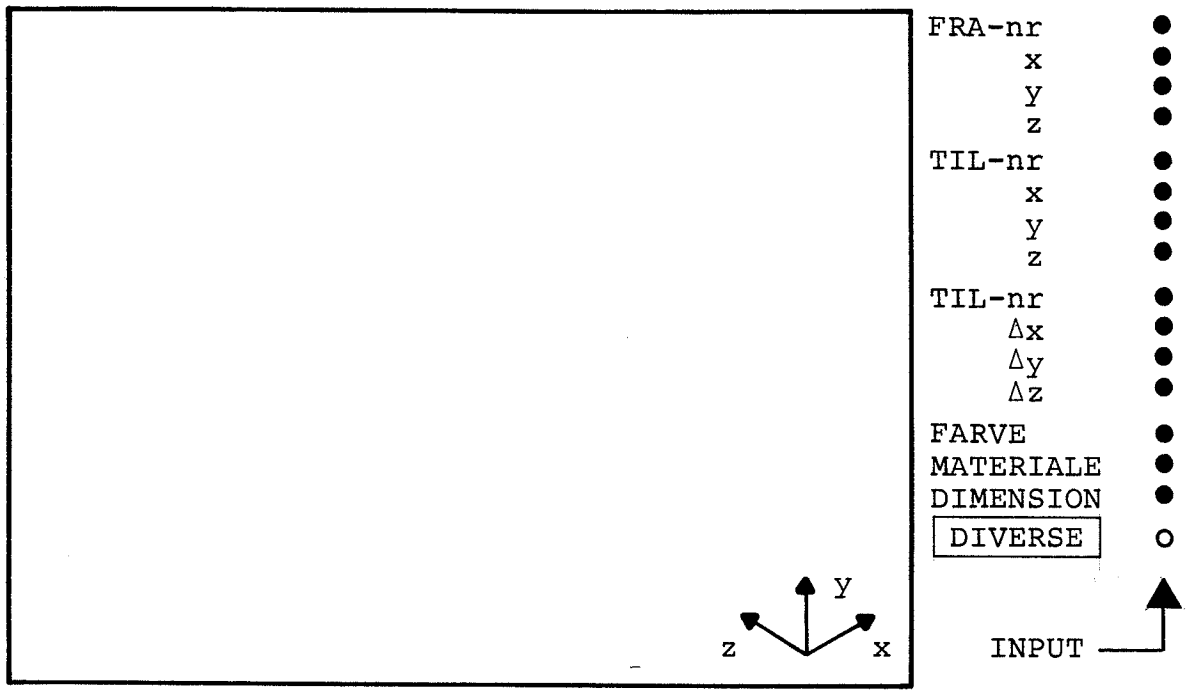
Disse skærmbilleder er en del af "instruktion", men vises altid ved begyndelsen af et program. Det øverste ved oprettelse af en ny tegning, det nederste billede efter indlæsning af et eksisterende DATA-sæt.

DATA-indlæsning tager iøvrigt ca. 1½ min., så hav tålmodighed. Bemærk, at der her er benyttet et FLP2-program. Det kunne også have været et WINCHESTER-program.

Skærmbilledet resumerer de væsentligste instruktioner og oplyser desuden

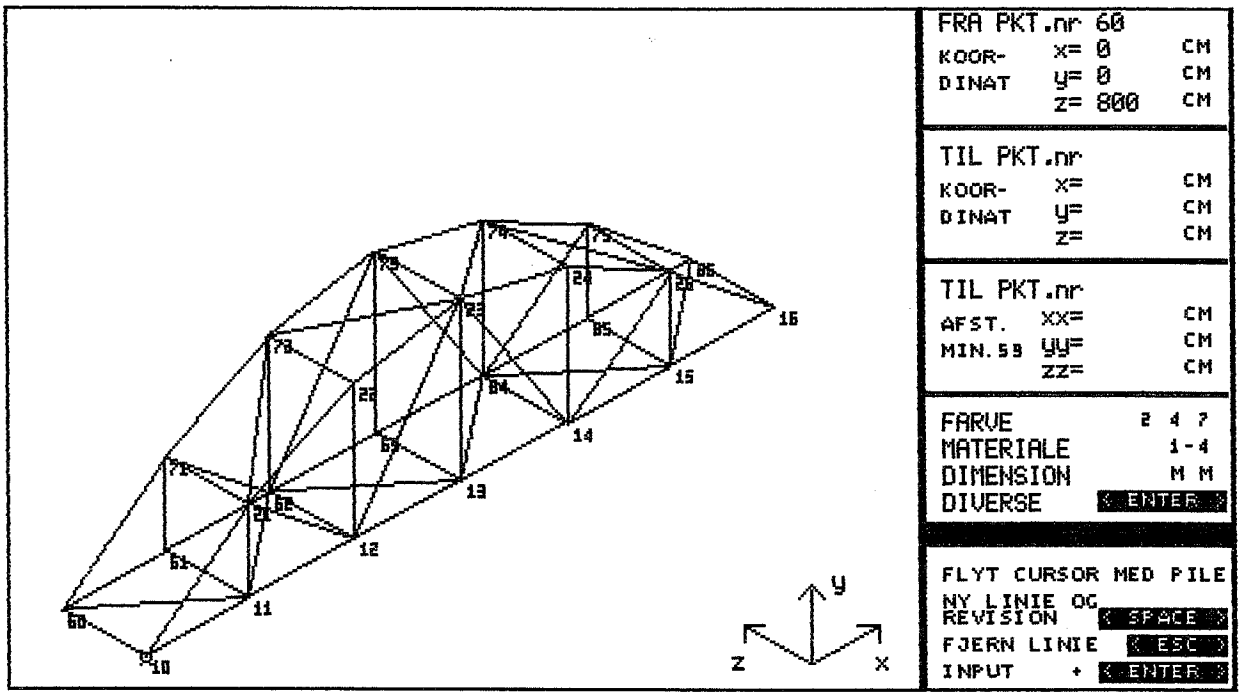
at man bør overveje numre og farver nøje, hvis en hurtig PLOTTER-udtegning ønskes (programmet har ikke PLOTTER-optimering)

at skærmens skala i dette tilfælde er 4512 (lodret), hvilket medfører, at parallelle linier ikke udtegnes synligt adskilt på skærmen (men nok på plotteren), hvis koordinaterne til endepunkterne ikke har forskelle, der er mindst 41 cm. Endvidere at punktnumre skrives "oveni" hinanden for punkter med afstande under 140 cm. Tallene er gennemsnitlige erfaringstal, der beregnes ud fra de givne minimums- og maximumsværdier og den deraf følgende skala.



NORMAL INPUT-SEKVENS

Numre	Mellem 1 og 99. Udskrives tocifrede (01..99). Trecifrede numre er der næppe plads til. Evt. kunne tallene 01..99 udskrives med hvidt, 101..199 med rødt, uden første 1-tal.
Koordinater	Heltal. For at få en rimelig skala er mindste forskel mellem min- og max-værdierne 100. Største koordinat er i programmet begrænset til 200 m, hvilket er rimeligt, skærmens opløsning taget i betragtning. Forsåvidt kan man da iøvrigt se bort fra, at programmet spørger efter cm - og i stedet benytte mm eller km ....
Farver	2,4,7.
Materialer	0,1,2,3,4. Kan ændres efter behov.
Dimension	5...1000. Kan ændres efter behov.

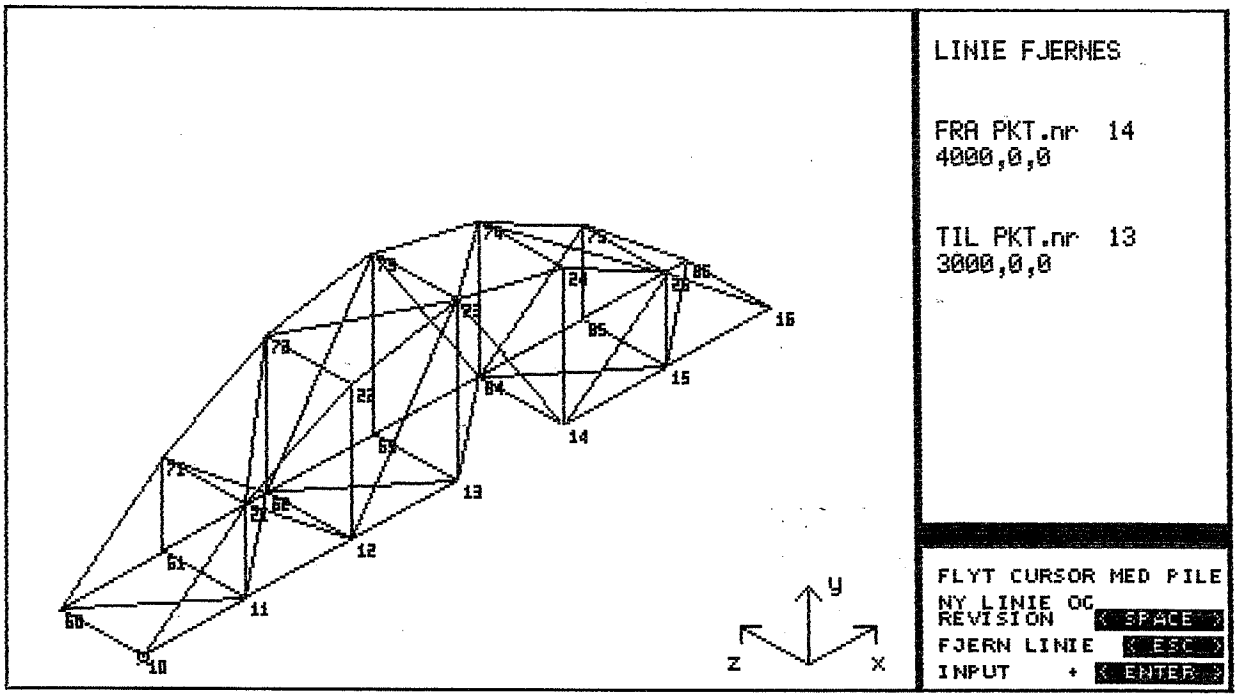


INDTAST AF NYE LINIER/REVISION.

0. Se den mere detaljerede beskrivelse pag. 10-13.
1. TAST "SPACE" (mellemrumstangenten).
2. TAST FRA-nummer +ENTER.
3. KOORDINATER: 3a. Nyt nummer: "-" udskrives. Tast koordinat +ENTER.
   3b. Eksist. nummer: Eksist. koordinater udskrives.
4. TAST "SPACE", hvis der skal benyttes absolutte koordinater, ... ellers '↓' til felterne for tre relative koordinater.
5. TIL-nummer med absolutte koordinater, x, y, z.
   5a. Nyt nummer:"0" udskrives. Tast koordinat +ENTER.
   5b. Eksist. nummer: Eksist. koordinater udskrives.
   Tast ENTER for accept af eksisterende koordinater.
   Tast revideret koordinat +ENTER.
6. TIL-nummer med relative koordinater, afstandene xx,yy,zz.
   Husk fortegnene !
   Nyt,accept,revision som 5a og 5b.

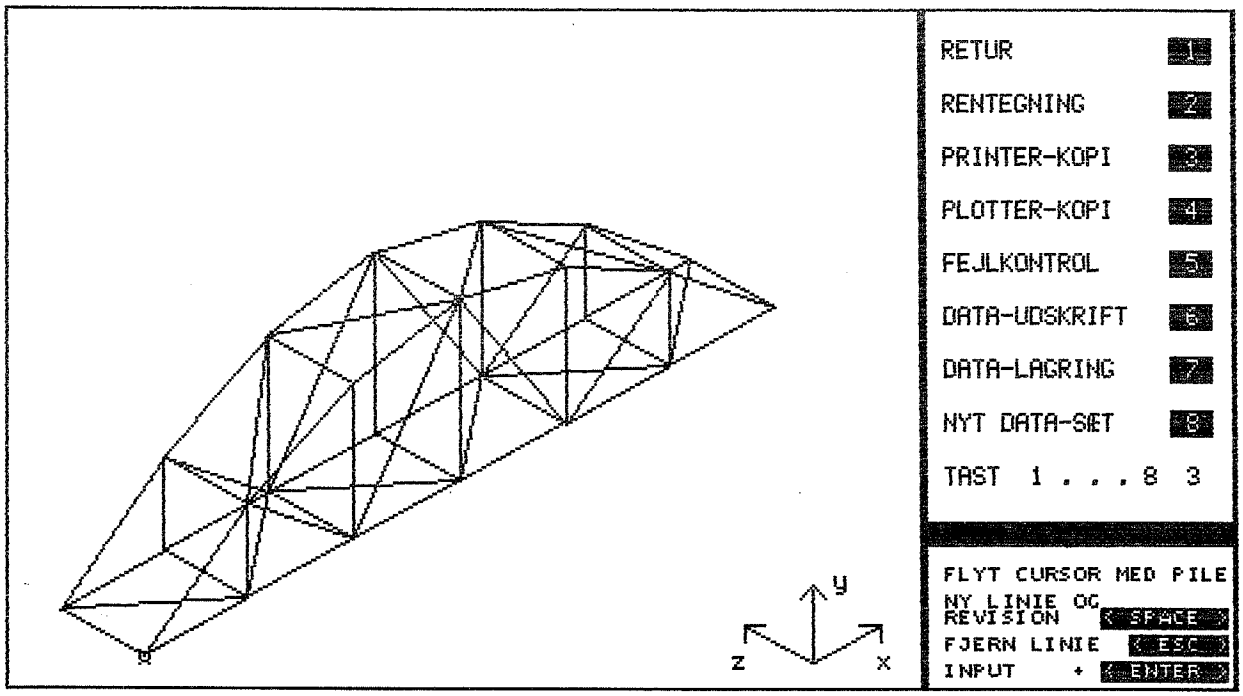
NB Under 5 og 6 anerkendes INPUT kun inden for de opgivne grænser.

7. INPUT af FARVE: "2" er DEFAULT, tast ENTER - eller vælg 4/7 +ENTER.
8. INPUT af MATERIALE: "0" medfører, at DIMENSION overspringes og også sættes til "0". 1...4 anerkendes som DATA.
9. INPUT af DIMENSION: 5...1000 anerkendes som DATA.
10. ALTERNATIV: TAST "↓" indtil CURSOR står ved "DIVERSE".
    TAST "ENTER" for at benytte hjælpefunktionerne.
11. ALTERNATIV: TAST "ESC" for at slette en linie.



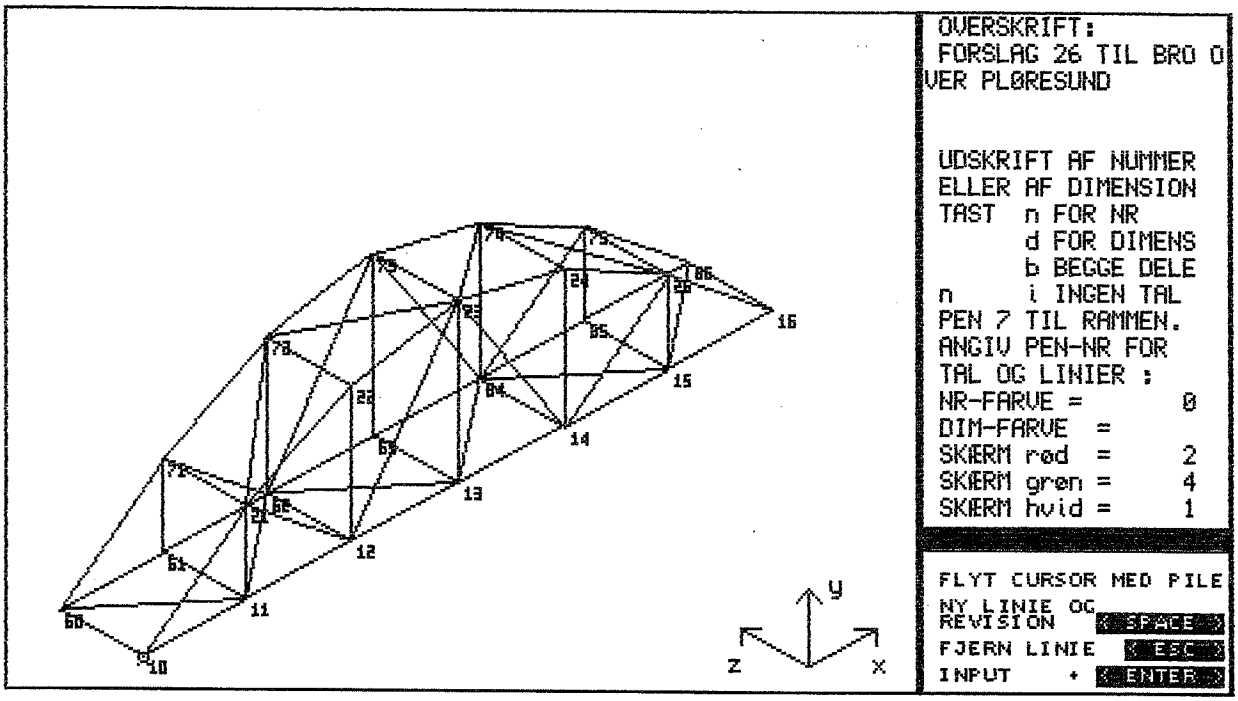
SLETNING

TAST "ESC" og angiv fra nr. og til nr.
Punkternes koordinater udskrives, linien fjernes med farve 0.
Punktnumre fjernes kun, hvis punktet ikke benyttes til andre linier (antal forbindelser reduceret til 0).
En ikke-eksisterende linie udskrives som fejl.



DIVERSE

1. RETUR til fortsat data-udtegning
2. NY UDTEGNING ud fra gældende DATA
3. PRINTER-KOPI af skærmbilledet
4. PLOTTER-udtegning
5. FEJLKONTROL (skæring mellem linier, punkter på linier osv.)
6. DATA-UDSKRIFT på printer (se EKSEMPEL BRO)
7. SAVE af eksisterende DATA på FLP
8. NYT DATASÆT



PLOTTER-instruktion, efter "diverse", 4

Denne INPUT-sekvens for plotter er den samme for SERVOGOR (i programmet) og ROLAND DXY 880 (TILLÆG) og kan antageligt bruges til de fleste flerfarve-plottere.

OVERSKRIFT kan være en ganske lang streng, over flere linier i menu-vinduet.

DATO udskrives automatisk, se nedfotograferet eksempel pag. 72.

NUMMER og/eller DIMENSION kan udskrives i valgte farver. Valg af begge dele frarådes på komplicerede tegninger. Lav hellere to tegninger. På grund af, at vinklerne er 33^0 på skærmen, 30^0 på plotteren for linier parallelle med x- og z-akserne, kan man ikke påregne, at tal, adskilt på skærmen, også er det på plotteren, et vice versa.

Der er forudsat 8 mulige penne (og pen-numre), nummeret fra 0 til 7, dvs svarende til QL's farvekoder.

Pen 7 (~hvid) benyttes til rammen, ofte med en kraftigere streg.

Principielt kan numrene anvendes til forskellige stregtykkelser, af samme farve, fx sort.

Der skal angives farve (pen-nummer) for alle tre på skærmen mulige farver, uanset om de alle er benyttet, idet programmet, som det foreligger, ikke holder regnskab med de benyttede linie-farver.

INDLEDNING

Tegneprogrammer på PC'ere, i bygningsingeniør-sammenhæng

For et projekterende eller udførende ingeniørfirma synes tegneprogrammer, der kun er tegneprogrammer at have begrænset værdi, i hvert fald med den software, der findes på markedet i dag. Derimod har kombinationen af grafik og beregninger efter min opfattelse betydelig interesse.

Perspektiv-tegneprogrammer kan dog have værdi til illustration af et projekt over for bygherre, offentlighed osv. Perspektivtegningen er mere "naturtro" end de gængse ingeniørtegninger.

Specielt i sammenhænge, hvor det visuelle indtryk spiller en væsentlig rolle, og hvor der ønskes flere (mange) afbildninger af det samme objekt, er den EDB-styrede perspektiv-tegning berettiget.

Her bliver der et rimeligt forhold mellem arbejdsindsatsen og resultatet.

Resultatet af indtastningsprocessen kan blive en omfattende analyse, fx. til bedømmelse af en konstruktions æstetiske virkning fra mange øjepunkter (retninger, højder, afstande) eller til bedømmelse af en motorvejs forløb i landskabet, for at tage to kendte exempler.

Ofte er rene tegneprogrammer af tvivlsom værdi. Dette kan skyldes

a) at den tilgængelige software er "lukket", så tillempning til danske forhold eller det enkelte firma's behov er umulig, eller i hvert fald vanskelig,

b) at arbejdet med indtastning (og kontrol) af data er stort og langvarigt,

c) at plotter-tegningen ofte ikke kan stå mål med professionelt arbejde. Nogle plottere har i dag et eller flere tegnesæt, der er acceptable, og i så fald - men prøv det først - er taludskrifter, især ved mange, små tal, bedre eller i hvert fald på højde med professionelt arbejde. Streger synes derimod at være substandard, med mindre en professionel hele tiden overvåger udtegningen, udskifter penne, kasserer uheld etc.

   Specielt er det svært at opnå rimelige talplaceringer for målrækker langs mållinier, fordi datamater nu engang ikke - med dagens teknik - kan flytte et tal væk fra mållinien og anbringe dette på en "fri" plads, rimeligt nær, med en S-formet henføringslinie til den formelle plads. Det er i hvert fald håbløst i isometrier o.lign. at undgå liniesammenfald, tal oven på linier, tal oven på tal osv.

   Her er det, at den professionelle tegner "snyder" og løser problemet, fx. ved at indføre lidt forkerte mål med en streg under.

d) at ingeniørfirmaer ikke har råd til selv at udvikle de ad hoc-programmer, deres firma, den aktuelle opgave, har behov for.

Når jeg alligevel tror på, at det grafiske har en stor plads i de næste års EDB-udvikling, skyldes det,

e) at grafikken på en skærm kan udnyttes til visuel kontrol af data, der senere skal indgå i komplicerede beregninger, uden at belaste beregningsprogrammet med langvarige, meningsløse beregninger ud fra datasæt, der "kun" indeholder 0.1‰ gale tal,

f) at skærmen kan udnyttes til at illustrere/analysere problemer, svare på tilbagevendende spørgsmål.

   En skærmanalyse af forsøgsdata er vi vant til i det små, fx. oscillografen. Skærmen kan udnyttes meget mere effektivt, både til generel illustration af forsøgets/konstruktionens/ideens udvikling og til speciel påpegning af "afvigelser", mulige fejl osv.

   Visuel kontrol af examensopgaver har vist sig hurtigere end den gamle metode på visse IFH-kurser, hvor grove placeringsfejl for bygningskomponenter blev afsløret, således at de (i øvrigt (!)) rigtige beregninger ville være meningsløse i praksis.

g) at sådanne skærmanalyser kan dumpes på printeren, giver den fordel, at man får et groft billede - og en huskeseddel - af noget, der senere bør dokumenteres nærmere.

h) at sådanne skærmanalyser i mange tilfælde kan føre til acceptable plotter-tegninger, der kan bruges til rapporter osv.

i) at der er et potentiel i grafik plus teknik, se senere.

Bare der dog var nogen, der kunne få lært de datamater + plottere at få skriftens udseende og placering til at være anstændig.

I dag bliver videnskabelige, tekniske og faglige rapporter ofte præget af hæslige EDB-udskrevne opstillinger, der klart viser EDB-programmernes begrænsninger.

At der er personer, der finder på at tale om soejler og bjaelker, er jo kun en fornaermelse mod rapportens modtager. Det kan løses i dag på selv en 1000 kr.s plotter.

Det er lige så galt, at man ofte ser overdreven brug af æstetisk inkommensurable skriftbilleder og størrelser, vandrette bogstaver og tal langs lodrette linier og lignende, der viser, at nogle teknikere i dag opgiver at få tid til den æstetiske side af sagen.

Teknikere er jo ellers vant til at samarbejde med grafikere. EDB-teknikere, brugere såvel som leverandører, synes at være temmelig ligeglade.

DANSK TEKST-TV er nok det styggeste eksempel, for her er der ressourcer nok til at løse problemet, og dansk TV har tradition for at satse på kvalitet. Men Å er forvandlet til en hæslig dværg.
Se TILLÆG, pag. 97.

IFH og TEGNEPROGRAMMER

Når IFH satser kraftigt på EDB, og specielt den grafiske side, har det er klart formål.

Formålet er en del af det "initiativområde" CAD - som er oprettet under Bygningsingeniør-fagområdet, hvor den første af 4 VIP ansættes ved IFH.

Formålet er at medvirke til den EDB-udvikling, danske rådgivende og udførende firmaer må gennemløbe i disse år.

IFH kan medvirke til forskning på feltet, gerne til programmeludvikling, men også til generel analyse af mulighederne.

IFH kan medvirke ved undervisning, så unge ingeniører bedre kan analysere markedets software - og derefter udvikle noget, firmaet kan bruge.

IFH kan ikke undervise i data-teknologi på avanceret maskinkodesprog eller i metoder til at udvide "lukket" software til praktisk brug. Det kunne andre DTH-institutter måske.

IFH kan satse i forskning og udvikling på udviklingen af samspillet mellem grafik og ingeniørmæssig design, primært: beregninger og overførsel af instruktioner (data) mellem rådgivende/udførende/bygherre etc.

Dette er min personlige mening, som jeg pt. handler ud fra.

Jeg tror, at mange vil være enige, men der kan da hugges mange pinde på formuleringen, spåner = detaljer.

Da IFH har både arkitekt- og ingeniørmedarbejdere, er det æstetisk-teknisk-grafiske et af afdelingens naturlige arbejdsområder.

IFH kunne også medvirke til at udvikle mere anstændige danske karakterer, jfr. fx det i dette program benyttede alternative tegnsæt, hvor fx Å ligner et dansk Å mere end den dværg, QL benytter, se TILLÆG.

Anstændige bogstaver på alle PC'ere vil antageligt betyde, at der må etableres et fælles-europæisk samarbejde. Alle sprog, bortset fra engelsk, har nemlig brug for en højere matrice til udskrift af store bogstaver med bolle, accent etc. oven over grundbogstavet.

I skrivende stund vides det ikke, hvilke datamater B-sektoren vil anskaffe til undervisning eller til forskning. Afdelingen har for egne midler anskaffet 5 stk. QL-512 datamater, $3\frac{1}{2}$" og $5\frac{1}{4}$" floppydiske, printere, plotter m.v. Hertil kommer tilslutningen til NEUCC.

Det er godt nok til øvelser til den dag, vi - og gerne i samråd med fx FRI - når til et "system".

QL'ernes sprog er SUPER-BASIC, en avanceret form for BASIC, med (rekursive) procedurer osv.osv. uden de fristende unoder (GOTO, GOSUB), som gammeldags BASIC har, til skade for logisk programmering. Dermed være ikke sagt, at mit program er klart, forståeligt, godt. Det er første forsøg.

Alt kan gøres bedre. Meget bedre. Se fx den ½ år yngre "tput"-procedure i relation til "talput" i "isometri_tegning". "tput" kan meget mere, på samme plads.

At QL ikke er kompatibel med en række andre PC'ere, lægger jeg ikke så meget vægt på. De skal bruges til at indøve programudvikling inden for det teknisk-grafiske. Her er QL, i forhold til prisen, en "kraftig" PC, velegnet til eksamensprojekter.

I øvrigt tror jeg, at "kompabilitet" er et "naivt" ord. Selv fabrikat xx er ikke kompatibelt med xx, version 2, endsige xx plus 3 år.

Alt er "kompatibelt", hvis det blot drejer sig om overførsel af data fra ingeniør til fabrikant, for så er "kompatibiliteten" ikke et hardware, men et software problem, nemlig en simpel aftale mellem en række firmaer om data-strukturen.

"Standarder" har vi tradition for i byggebranchen. Vi får også "data-standarder" meget snart. Telefonmodemet eksisterer.

I Norge er der udgivet standarder.

IFH's "program-udbud" primo 1986

IFH har ikke brugsklare professionelle programmer, men vi har da opbygget startgrundlaget for det i form af

a) Basisrutiner: En lang række, afprøvede grafiske rutiner, karaktersæt til skærmudskrifter, plotter-rutiner, input-kontrolrutiner (tal/streng/alternative procedurekald, max, min (decimaltal/streng) osv. Alt det trivielle. Det kan gøres bedre.

b) Tegneprogram for udtegning (skærm/plotter) af definerede konstruktioner, opbygget af definerede komponenter, sammenbygget på defineret måde, i retvinklede projektioner eller (drejeligt) perspektiv. Her er "komponenter" med to parallelle flader, typisk dæk- og vægelementer, med huller, udsporinger, konsoller, døre, vinduer osv. (Chr. B. Lund's examensprojekt, afleveret januar 1986).

c) Tegneprogram for isometrisk afbildning (45+45^0, dvs. vvs-afbildning og 120+120^0, dvs. "normal" isometri er færdige, andre isometrier osv. kræver kun små ændringer, evt. indførsel af "drejningsmatricer" etc.). Her er komponenterne dele af et gittersystem, fx. et rørsystem eller et statisk rumgitter.

Programmerne b) og c) har det tilfælles, at de begge kan bruges grafisk i praxis, som de er, b) fx. til mangfoldige perspektiver af et givet præfabrikeret hus, c) fx. til udtegning af VVS-tegninger med radiatorer, eller af gittersystemer i isometri med udskrevne dimensioner og/eller punktnumre.

Programmerne b) og c) har imidlertid noget væsentligere tilfælles:
Tegneprogrammerne er generelt ikke udviklet som rene tegneprogrammer.

De er begge udviklet med det formål, at tegningerne kun er en del af programmet, måske den mindst væsentlige.

Programmerne er udviklet som basis for det, jeg tror, grafik på bygningsorienteret EDB skal bruges til: At danne visuel baggrund for ingeniørmæssig beregning etc.

Komponenterne, hvad enten de er "legemer" a la elementer eller "gitterstænger" a la rør/bærende træk/trykstænger, har ikke kun koordinater, længder osv. Komponenterne har også egenskaber, fx. dimensioner, detailform, anvendelse, farve, for den sags skyld lugt og smag.

Det egentlige formål er beregning, prissætning, udførelse.

Programudviklingen har også som formål at lære de studerende, at gøre deres programmer brugervenlige:

Man kunne fx forlange, at et program skulle kunne betjenes af en teknisk praktikant efter max. 2 timers instruktion (for så vidt angår indtastningen, konklusionerne må den sags-ansvarlige drage).

Et program skal loades automatisk (boot). Det bør så vidt muligt ikke kunne gå i stå. Men det skal kunne stoppes vilkårligt (CTRL + SPACE), når brugeren er i tvivl. Det skal kunne gen-startes ved en simpel kommando. fx. RUN. Der må ikke gå data tabt ved stop eller start. Der må ikke være indlagt data i "databasen" før såvel brugeren som fejlkontrol-procedurerne er enige. Brugeren skal "til enhver tid" kunne kalde en slags hjælp/instruktion. (Her: tast "i", når der ønskes INPUT af et tal). Data-udskrifter, skærmdump, plotter-tegninger skal automatisk fremkomme på ønskede tidspunkter (evt. ved simple kommandoer, dvs. een karakter).

Jeg har dog endnu ikke fået lært datamaten at kontrollere, at det er korrekt, når brugeren siger, at plotteren er tændt og har papir og penne parat. Brugeren kan dog taste CTRL+SPACE, gøre plotteren klar, og taste "RUN" (samt ny plotterordre).

Nogle printere signalerer - men akustisk ! - , hvis de ikke har mere papir at skrive på. En oplagt opgave: at udvikle en software, der standser udskriften, til papiret er isat, ved en simpel tilbagemelding til computeren og/eller modtagerens telefon. Printeren står måske i et kontor 100 m, måske 100 km væk, og den udskriver måske fortrinsvis om natten, mens brugeren sover.

Jeg har lært datamaten at spørge om (og derefter justere sin printerdriver), hvilken printer, der skal benyttes. Mange kommercielle programmer forstår ikke, at en arbejdende ingeniør må tage den af måske 4-5 forskellige printere og plottere, der er ledig, og at han derfor med en enkel, en-karakters kommando, må kunne omstille printer og plotter-driverne. QL'ens fire programmer er håbløse i denne forbindelse. I øvrigt kan (vil) QL ikke checke, om oplysningerne om printer-type er korrekte.

Eksamensprojekter kan omfatte alt, med hovedvægten på grafik, teknik, snurrepiberier eller maskinkoder, men principielt især med tilknytning til erhvervslivets problemer. Kommunikation og udskrift skal ske på dansk, inden for de grænser, vore bevillinger (desværre) sætter.

BOOT

```
100 REMark ****  LOADE-PROGRAM isometri_boot  ****
102   flp$= 'FLP2_': DATA_USE flp2_: PROG_USE flp2_: DEST_USE flp2_
104   PAPER#2;2: INK#2;7: CLS#2: CSIZE#2;2,0:CSIZE#1;2,0
106   AT 2,1: PRINT'PRINTER QL / CP80A'\' TAST    Q  / C'
108   REPeat pr: AT 3,18: INPUT  pri$: IF pri$(1) INSTR 'qc': EXIT pr

110   ramf= RESPR (4988): LBYTES ram_Ver1,ramf: CALL ramf
112   CLS: TK2_EXT: CSIZE 2,0: CSIZE#2; 2,0: PRINT\\'   RAM     LOADed'
114   IF pri$(1)=='c': printer$= 'c': kop$= 'scopy_bin'
116   IF pri$(1)=='q': printer$= 'q': kop$= 'qcopy_bin'
118   KOPI=     ramf + 1780: LBYTES kop$, KOPI: CALL KOPI
120     PRINT \'   SCOPY   LOADed'
122   dansk=    ramf + 2980: LBYTES qltegn   , dansk
124   minitegn= ramf + 3568: LBYTES vvsfount , minitegn
126   prik=     ramf + 4444: LBYTES nyptpatch, prik: CALL prik
128     PRINT \'   qltegn  LOADed'
130     PRINT \'   vvstegn LOADed'\\'   ptpatch LOADed'

132   AT#2; 2,1: PRINT#2;'INSTRUKTIONEN FØRST'
134   AT#2; 5,1: PRINT#2;'DIREKTE TIL PROGRAM': STRIP#2;7: INK#2;2
136     PRINT \'   © 1986 JMP mfl':              AT#2;3,17: PRINT#2;' i '
138   AT#2; 8,1: PRINT#2;'TAST i ELLER p     ': AT#2;6,17: PRINT#2;' p '
140   AT#2;12,1: PRINT#2; flp$(1 TO 4) & ' SKAL INDEHOLDE'
142   AT#2;13,1: PRINT#2;'PROGRAM - FLOPPYEN.'
144   AT#2;15,1: PRINT#2;'SENERE SKAL BRUGES '
146   AT#2;16,1: PRINT#2;'DATASÆT - FLOPPYEN.'
148   AT#0;2,2: PRINT#0;'INSTRUKTIONEN KAN KALDES IGEN SENERE ';
150   PRINT#0;'VED AT TASTE "i" I STEDET FOR INPUT AF ET TAL.'
152   REPeat loop
154     AT#2;8,17: INPUT#2;' ';ip$;' ': IF ip$ INSTR 'ip': EXIT loop
156     AT#2;8,17: PRINT#2;'    '
158   END REPeat loop

160  IF ip$=='i' THEN
162   REMark Qsoft dump assembler routine.Version 002-AA 7.aug.1985
164   STRIP 7: INK 2: AT 14,2: PRINT' TAST MELLEMRUM ': CSIZE#2; 0,0
166   AT 15,2: PRINT' FOR  HVERT     ': AT 16,2: PRINT' SKÆRMBILLEDE.  '
168   REPeat vent: IF INKEY$(-1)=' ': EXIT vent
170   adr= ALCHP(32768): mc= ALCHP(100): LBYTES download_bin, mc
172   RESTORE 182: DIM na$(8,22): FOR n= 1 TO 8: READ na$(n)
174   FOR n= 1 TO 8
176     LBYTES na$(n), adr: CALL mc,adr,131072
178     REPeat vent : IF INKEY$(-1)=' ': EXIT vent
180   END FOR n
182   DATA 'isometri_dato','isometri_instr1','isometri_nytegning'
184   DATA 'isometri_instr3','isometri_nrtegning'
186   DATA 'isometri_instr2','isometri_linietegning','isometri_afslut'
188   AT#2;2,2: PAPER#2;0: INK#2;7: PRINT#2;'EN GANG TIL? J / N'
190   REPeat engangtil
192     AT#2;2,22: PRINT#2;'  ': AT#2;2,22: INPUT#2;y$: y$= y$(1)
194     IF y$=='j': GO TO 174
196     IF y$=='n': RECHP adr: RECHP mc: EXIT engangtil
198   END REPeat engangtil
200  END IF

202   STRIP 7: INK 2: AT 14,3: PRINT'  PROGRAMMET  '
204   AT 15,3: PRINT'  LOADes  PÅ  ': AT 16,3: PRINT'  2 MINUTTER  '
206   PAUSE 100: CLS#0: INK#0;0: FORMAT#0; ram2_4: INK#0;4
208   SDATE 1986,1,1,0,0,0: CLOCK#2; ' %m:%s '

210   OPEN_NEW#5;ram2_: PUT#5; 'ny', printer$, minitegn, dansk: CLOSE#5
212   LRUN isometri_tegning
```

Programmet LOADes i to omgange (flp$ mv. kan ændres af brugeren)

isometri_boot LOADer maskinkoder, giver mulighed for at se instruktionen og LRUNer (LOAD + RUN) derefter hovedprogrammet.

LRUN medfører en NEW, der fjerner hele boot-programmet (linie 210)

SORT SKÆRM UNDER LOAD UNDGÅS, SOM VIST PAG. 105.

Maskinkoder skal LOADES FØR PROCEDURer defineres, så "boot" og hovedprogrammet kan ikke slås sammen.

At "boot" er fjernet, når hovedprogrammet kører, medfører
a) at hovedprogrammet kan gen-startes med RUN, uden at maskinkoderne gen-LOADes med systemsammenbrud til følge.
b) at INPUT af koordinater kun overføres til de to arrays, når det er sikkert, at alle tal er klar og godkendte, se linie 7700-7840.
c) at adresserne på de maskinkoder, der ikke allerede er kaldt (CALL) i "boot" må overføres til hovedprogrammet (går tabt efter en NEW). Derfor linie 208, hvor adresserne "minitegn" (vvsfount) og "dansk" (qltegn) gemmes i RAM (FORMAT med usynlig udskrift, se linie 204). Samtidigt sættes load$="ny" i hovedprogrammet, se linie 410, og værdien af printer$ overføres.

MASKINKODER

ram-Ver1 (ramf) medfører, at CARE-QJUMP-udvidelsen kan benytte RAM1...Ram16 som (midlertidige) disketter, dimensioneret enkeltvis, op til 900 sektorer. I linie 204 dimensioneres RAM2 til at overføre variable til hovedprogrammet. TK2_EXT sikrer, at TOOLKIT2 ellimine-rer, hvad fx INTERFACE måtte have af procedurer med samme navn.

SCOPY (kopi) er en rutine til udprintning af skærmbilledet på CP80A, QCOPY (kopi) tilsvarende på QL-printeren.

qltegn er de normale tegn, dog med mere anstændige Æ og Å.
vvstegn er små tal og bogstaver, der kan anvendes som eksponenter og indices, her til numre og dimensioner. Se bl.a. linie 9000-9190.

nyptpatch (prik) er en rutine til at undgå den i den danske ROM indbyggede fejl, hvor POINT (+ korte linier) medfører to pixler (prikker) i stedet for een.

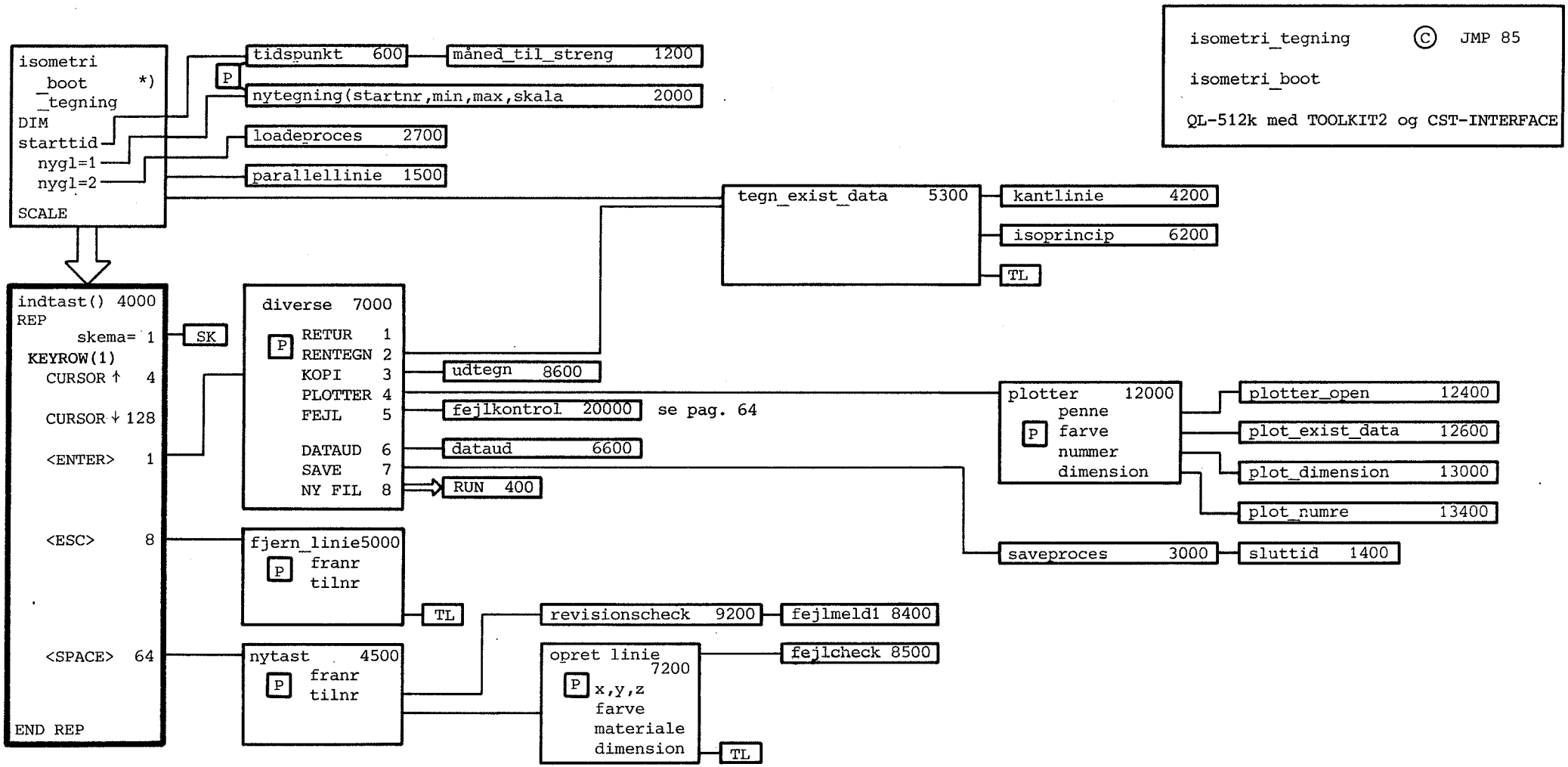
Maskinkoderne er fastlagt ved addition af længderne for at spare plads i hukommelsen. Enkeltvis brug af RESPR allokerer n×512 bytes. Summen er 1780+1436+588+876+544=5224, idet qltegn og vvsfount rundes op til lige tal (588,876).

I instruktionsdelen (PROC i hovedprogrammet) benyttes en skærmbilledrutine. "download_bin", der lægges i adressen ALCHP (32768) = længden af skærmbilledet. ALCHP er en midlertidig adresse, der kan fjernes med RECHP, linie 194 (TOOLKIT-vejledningen). En sådan allokering kan godt finde sted, selv om PROCedurer er definerede forud.

linie 134 COPYRIGHT-tegnet udskrives ikke af CPA80-printeren.

linie 150 REPEAT loop. INPUT bør altid ske i form af en streng (xx$), af hensyn til kontrol, smlgn. "talput" ("tput", tillægget). Her sikrer loop, at kun INPUT af "i"/"p" lader programmet fortsætte. Bemærk, at == betyder "cirka lig med", dvs 10E-7 for tal og STORE = små bogstaver for strenge (linie 192)

linie 206 Her sættes en dato, der sikrer, at uret i kanal 2 starter med 00:00. Smlgn. hovedprogrammets linie 430, hvor CLOSE#2 mv standser uret. Kanalen må åbnes igen, som con (!)



isometri_PROCEDURER SK = skematext 3400 | TL = tegn_linie 5500 | strengnr 9900

iso vindue 8900 | vvstegn 9000 | qltegn 9100

STANDARD-PROCEDURER P = talput 9500 | c80routiner 30000 (4 stk.) | c80program 6500

*) I boot og talput valg af: instruktion (200)

GENSTART EFTER STOP: EXCL. TEGNING: indtast 1 (0) INCL. TEGNING: RUN (se linie 400-590)

PROCEDUREOVERSIGT

```
  200: SELVE STARTEN - EFTER boot
  600: tidspunkt
 1200: måned_til_streng
 1400: sluttid
 1500: parallellinie
 2000: nytegning
 2700: loadeproces
 3000: saveproces
 3400: skematext
 4000: indtast (skema)
 4200: kantlinie
 4500: nytast
 5000: fjern_linie
 5500: tegn_linie (franr,pos,nrfarve)
 5900: tegn_exist_data
 6200: isoprincip
 6500: c80program
 6600: dataud
 7000: diverse
 7200: opret_linie
 8400: fejlmeld1
 8500: fejlcheck
 8600: udtegn
 8900: iso_vindue
 9000: vvstegn
 9100: qltegn
 9200: revisionscheck
 9500: talput (kanal,lin,tab,min,max,antal)
 9900: strengnr (nrs,xs,ys,xnr,ynr)
11600: nymem. Se tillægget.
12000: plotter
12400: plotter-open
12600: plot_exist_data
13000: plot_dimension
13400: plot_numre
20000: fejlkontrol
20400: punktkontrol
20700: fejludskrift1
21000: liniekontrol
21200: determinant
21300: skæringspunkt
21700: printerstart
30000: max_vindue
30200: dobb_vindue
30400: c80list (linie1,linie2)
30500: c80dir
```

Efter LRUN FLP1_isometri_boot indlæses programmets hoveddel.
Det består af en indtast-løkke "indtast 1"(0), der fortsætter, indtil programmet afbrydes.
Ud fra løkken kan oprettes/revideres linier, fjernes linier, benyttes hjælpefunktioner, idet alt returnerer til indtast-løkken.

Bemærk, at programmet også kan benytte FLP2 eller en WINCHESTER disk, hvis der ændres i LOADeprogrammets linie 102 og hovedprogrammets linie 402.
Benyttes FLP1 kan starten reduceres til LRUN isometri_boot, eventuelt kan programmet gøres selvstartende ved at ændre navnet "isometri_boot" til "boot" (RENAME), idet filnavnet "boot" på FLP1 medfører, at QL automatisk indlæser "boot". (SE TILLÆG PAG. 105-106)

HOVEDPROGRAMMET

ISOMETRI-TEGNING

```
200 REMark **********************
          *  isometri_tegning  *
          **********************
202 DEFine PROCedure instruktion
204   REMark Qsoft dump assembler routine.Version 002-AA 7.aug.1985
206   STRIP 7: INK 2: AT 14,2: PRINT' TAST MELLEMRUM ': CSIZE#2; 0,0
208   AT 15,2: PRINT' FOR  HVERT     ': AT 16,2: PRINT' SKÆRMBILLEDE.   '
210   REPeat vent: IF INKEY$(-1)=' ': EXIT vent
212   adr= ALCHP(32768): mc= ALCHP(100): LBYTES download_bin,mc
214   RESTORE 224: DIM na$(8,22): FOR n= 1 TO 8: READ na$(n)
216   FOR n= 1 TO 8
218     LBYTES na$(n), adr: CALL mc,adr,131072
220     REPeat vent : IF INKEY$(-1)=' ': EXIT vent
222   END FOR n
224   DATA 'isometri_dato','isometri_instr1','isometri_nytegning'
226   DATA 'isometri_instr3','isometri_nrtegning'
228   DATA 'isometri_instr2','isometri_linietegning','isometri_afslut'
230   AT#2;2,2: PAPER#2;0: INK#2;7: PRINT#2;'EN GANG TIL? J / N'
232   REPeat engangtil
234     AT#2;2,22: PRINT#2;'  ': AT#2;2,22: INPUT#2;y$: y$= y$(1)
236     IF y$=='j': GO TO 216
238     IF y$=='n': RECHP adr: RECHP mc: EXIT engangtil
240   END REPeat engangtil
242 END DEFine instruktion

400 REMark ********** HER BEGYNDER HOVEDPROGRAMMET **********************
402 flp$= 'FLP1_'
404 OPEN#5;ram2_: GET#5; load$, printer$, minitegn, dansk: CLOSE#5


410 IF load$== 'ny' THEN
412   DIM start%(12),pkt%(99,9),linje%(98,57),pf%(4),pt%(4),la%(5)
414   DIM dato$(2,17),ktal(5),p%(99): trat= 0: trap= 0: baudtal= 0
416   nytastnr=0: nygl=0: antallinier=0: xcurs=0: ycurs=0: skema=0
418   filnavn$=FILL$(' ',50): starttid= 0: skala= 0: opad= 0: nedad= 0
420   højre= 0: venstre= 0: x0= 0: y0= 0: xp0%= 0: yp0%= 0: xxmin= 0
422   xmin= 0: ymin= 0: zmin= 0: xmax= 0: ymax= 0: zmax= 0
424   franr= 0: tilnr= 0: fran= 0: w= 0: pos= 0: rev= 0: prnr$=''
430   CLS#0: CLOSE#2:OPEN#2;con_: dobb_vindue:qltegn:MODE 512: tidspunkt
440   IF printer$== 'q': trap= 0: trat= 0: baudtal= 9600
450   IF printer$== 'c': trap= 1: trat= 0: baudtal= 4800
460 END IF : REMark NY RUN >>>> linier 480,490,580


470   CSIZE#2;0,0: CSIZE 0,0: TRA trap: REMark TRA trat ved SCOPY
480   iso_vindue: qltegn: CLS#0: BORDER#0;1


490 IF load$== 'ny' THEN
500   CLS: skala= 1000: isoprincip: BORDER 1,2: BORDER 0
510   IF nygl= 2: loadeproces
520   IF nygl= 1: nytegning  : nygl= 2
530   SCALE  skala, -skala*11/344, -skala*11/254: opad = skala
540   nedad= skala*11/254: venstre= skala*11/344: højre= skala*1.098
550   CLS#0: OVER 1: parallellinie: OVER 0: load$= 'gl'
560   OPEN#5;ram2_: PUT#5 \ 0, 'gl', printer$, minitegn, dansk: CLOSE#5
570 END IF

580 tegn_exist_data : indtast 1
```

PROCeduren "instruktion" er magen til "boot"s linie 158-198

PROGRAMSTART, linie 400-490

De i "boot" definerede variabler indlæses fra RAM.

Tre arrays alle med heltal (%) oprettes til permanente data:

start%(12):	startpunktets nr	(1)
	minimumsværdier for x,y,z	(2,3,4)
	maximumsværdier for x,y,z	(5,6,7)
	x0,y0 på skærmen	(0,9)
	skala	(10)
	antal linier	(11)
	min. afstand ml. parallelle linier	(12)

pkt%(99,9) punktnr, x,y,z, xs,ys,xp,yp, antal forbindelser, hvor xs,ys og xp, yp er de to-dimensionale koordinater på hhv. skærm og plotter.

linie%(98,57) franr (1)
tilnr (2,9,16,23,30,37,44,51)(max. 8 numre)
Δx,Δy,Δz (tilnr+1,2,3)
farve (tilnr+4)
materiale (tilnr+5)
dimension (tilnr+6)

Når xs,ys,xp,yp,Δx,Δy,Δz udregnes ved INPUT, og ikke når disse data skal bruges, skyldes det, at brugeren har tid - mellem flere INPUT - men ikke, når hele tegningen skal tegnes på skærm/plotter:

ktal(1) benyttes talrige steder via talput (linie 9500), til indtastning. DIM kan benyttes når som helst. Definerer og nulstiller.

Den aktuelle dato indtastes under "tidspunkt" (linie 600).

Afhængigt af den aktuelle printer sættes BAUD og TRA i linie 440-450.

Under "tidspunkt" er opgavens navn angivet, og om den er ny eller eksisterende (nygl=1/2)

Skala m.v. i iso_vindue er omtalt i indledningen.

I linie 580 forlades de indledende rutiner, og programmet fortsætter i REPEAT-løkken under "indtast (0/1)" (linie 4000 ff)

linie 414: ktal(5) er principielt overflødig, se talput (linie 9500) talput kunne omstilles til en serie INPUT 8antal fx 5), hvis der tilføjes en løkke til indtastsekvensen.

linie 430: Uret standses. Kanal må åbnes igen, som con (ikke scr)

linie 500: Midlertidig skala, før isoprincip.

linie 510: Eksisterende fil loades.

linie 520: Ny fil oprettes, nygl er nu 2.
Herefter sættes skala mv, kantlinie tegnes og relevant brugeroplysning udskrives.
"Parallellinie", se skærmbilledet pag. 14.

linie 560: load$='gl'. Tegneprogrammet begynder.

```
600 DEFine PROCedure tidspunkt
610   LOCal mdvklar$, fildato$, fundet, tid$, måned, dtal$, tim$
620   CLS: CSIZE 2,0: CSIZE#2;2,0: STRIP 7: INK 2
630   AT 1,1: PRINT'DATA KLAR ? J / N   '
640   REPeat spmloop1
650     AT 1,19: INPUT mdvklar$
660     IF mdvklar$=='j' AND FOP_DIR(#3,flp$)= 0 : EXIT spmloop1
670     AT 1,18: PRINT'    ': AT 20,1: PRINT flp$(1 TO 4) & '  MANGLER.'
680   END REPeat spmloop1: CLOSE#3: PAPER 2: INK 7

690     AT 3,1: PRINT'SKAL DER OPRETTES EN'\' HELT NY TEGNING'
700     AT 6,1: PRINT'-ELLER TEGNES VIDERE'\' MED KENDTE DATA'
710     AT 9,1: PRINT'TAST 1 ELLER 2 :': STRIP 7: INK 2
720     AT 4,18: PRINT' 1 ': AT 7,18: PRINT' 2 ': AT 9,18: PRINT '    '
730     talput 1,9,19,1,2,1: nygl= INT(ktal(1)): PAPER 2: INK 7
740   REPeat spmloop3
750     AT 12,1: PRINT'OPGAVENS NAVN ER'
760     AT 13,1: INPUT filnavn$: fildato$= flp$ & filnavn$ &'_dato'
770     fundet= 0: IF FOP_IN(#3,fildato$)= 0: fundet= 1: CLOSE#3
780     IF nygl + fundet== 1: tids$= FILL$(' ',17): CLS#2: EXIT spmloop3
790     IF nygl + fundet== 3 THEN
800       OPEN_IN#6;fildato$: INPUT#6;dato$(2): CLOSE#6
810       AT 16,1: PRINT'KENDTE dato$ ='\' '; dato$(2)
820       tids$=  FILL$(' ',17): CLS#2: EXIT spmloop3
830     END IF
840     IF nygl= 2: AT 20,1: PRINT'NAVNET FINDES IKKE.'
850     IF nygl= 1: AT 20,1: PRINT'NAVNET ER BRUGT FØR.'
860     PAUSE 200
870   END REPeat spmloop3

880   IF nygl=2: STRIP 7: INK 2: AT 20,1: PRINT'NAVNET FUNDET.      '
890   REPeat datoloop
900     REMark tids$= 02.MAJ 1985: 14.1 / 12.MARTS 85: 21.0   SE 1200 ff
910     AT#2;1,1: PRINT#2;'INDTAST DATO : '
920     AT#2;3,1: PRINT#2;'ÅRSTAL  85..99 '
930     AT#2;6,1: PRINT#2;'MÅNED    1..12 '
940     AT#2;9,1: PRINT#2;'DATO     1..31 '
950     AT#2;12,1: PRINT#2;'TIME 0.0..23.9 '
960     talput 2,4,11,85,99,1: tids$(10 TO 12)= INT(ktal(1))&':'
970     talput 2,7,11, 1,12,1: måned= INT(ktal(1)): måned_til_streng
980     talput 2,10,11,1,31,1: dtal$= INT(ktal(1))
990     IF ktal(1)<10 THEN dtal$='0'& INT(ktal(1))
1000     tids$(1 TO 3)= dtal$&'.'
1010     talput 2,13,11,0,23.9,1: tim$= FILL$(' ',4)
1020    tim$(1 TO 3)= FILL$('0',(ktal(1)<10)) & INT(ktal(1)) & '.'
1030    tim$(4)= 10*ktal(1) MOD 10
1040    tids$(14 TO 17)= tim$(1 TO 4): dato$(1)= tids$
1050    AT#2;16,1: PRINT#2;'NYE dato$ ='\' ';tids$
1060    AT#2;18,1: PRINT#2;'KORREKT  J / N '
1070    REPeat okdato
1080      AT#2;18,16: INPUT#2;y$: IF y$=='j': EXIT datoloop
1090      IF y$=='n': CLS#2: EXIT okdato
1100      AT#2;18,16: PRINT#2;'  '
1110    END REPeat okdato
1120  END REPeat datoloop: AT#2;18,1: PRINT#2;'SOM LAGRES NU    '

1130      SELect ON nygl= 2 : DELETE fildato$
1140      OPEN_NEW#6; fildato$: PRINT#6;dato$(1): CLOSE#6
1150  SDATE 1900+tids$(10 TO 11),måned,dtal$,tim$(1 TO 2),tim$(4)*6,0
1160  starttid= DATE
1170 END DEFine tidspunkt
```

spmloop1 — sikrer, at der er en diskette i FLP1.

spmloop3 — checker filens navn.
Er nygl=2 skal navnet kunne findes.
Er nygl=1 må navnet ikke eksistere.
- der kan/skal evt. benyttes forskellige disketter til program og DATA.

Er filen kendt, udskrives dens tidligere dato.
Der udskrives fejl, med nyt navne-INPUT.

linie 770 — Bemærk FOP_DIR og FOP_IN, se TOOLKIT2.
Hvis filen findes, er FOP_IN=0, og CLOSE#3 er nødvendig. Findes filen ikke, er kanal 3 lukket.

datoloop

datoens format: — se "måned_til_streng", linie 1200 ff
se også tids$, linie 730

datoen indtastes. — Der benyttes tal, inden for grænser, der giver en vis kontrol med fejl - men ikke fuldstændig.

talput — (linie 9500) accepterer decimaltal, men INT fjerner decimaler på år, måned, dato.

timetallet — Kan angives med een decimal (á 6 min).
Der udskrives .0, hvis ingen decimal angives.

Til slut gives der brugeren mulighed for at taste om.

Den nye dato lagres, og QL's interne ur sættes (SDATE)

Dato mv. strengbehandles så omstændeligt for at sikre, at printer- og plottertegningernes dato-angivelse ser "dansk" ud, se næste opslag.

```
1200 DEFine PROCedure måned_til_streng
1210   SELect ON måned
1220     =1: tids$(4 TO 9)= 'JAN.19'
1230     =2: tids$(4 TO 9)= 'FEB.19'
1240     =3: tids$(4 TO 9)= 'MARTS '
1250     =4: tids$(4 TO 9)= 'APRIL '
1260     =5: tids$(4 TO 9)= 'MAJ 19'
1270     =6: tids$(4 TO 9)= 'JUNI  '
1280     =7: tids$(4 TO 9)= 'JULI  '
1290     =8: tids$(4 TO 9)= 'AUG.19'
1300     =9: tids$(4 TO 9)= 'SEPT. '
1310     =10:tids$(4 TO 9)= 'OKT.19'
1320     =11:tids$(4 TO 9)= 'NOV.19'
1330     =12:tids$(4 TO 9)= 'DEC.19'
1340   END SELect
1350   REMark GL.DAGS DATO. TIMER MED 1 DECIMAL
1360 END DEFine måned_til_streng


1400 DEFine PROCedure sluttid
1410   LOCal timetal, time$, fildato$
1420   timetal= dato$(1)(14 TO 17) + INT((DATE-starttid)/360 +.5)/10
1430   time$= timetal: IF timetal < 10: time$='0'&time$
1440   IF LEN(time$)= 2: time$= time$&'.0'
1450   dato$(2)= dato$(1)(1 TO 12)&' '&time$
1460   fildato$= flp$ & filnavn$ &'_dato':  DELETE fildato$
1470   OPEN_NEW#6;fildato$: PRINT#6;dato$(2): CLOSE#6
1480 END DEFine sluttid


1500 DEFine PROCedure parallellinie
1510 LOCal t: t= 6: INK 4: AT 1,t: PRINT'VED NYE LINIER, SLETNING OG ';
1520 PRINT 'REVISION AF TIL nr.'
1530 INK 7: AT 2,t: PRINT 'KAN  FRA nr. OG TIL nr. ';
1540 PRINT 'VÆLGES I VILKÅRLIG RÆKKEFØLGE.      '
1550 AT 3,t: PRINT 'KUN KOORDINATERNE FOR ET "TIL PKT" KAN REVIDERES.'
1570 AT 4,t: PRINT 'PAS PÅ, AT xx, yy, zz FÅR KORREKTE FORTEGN.'
1580 AT 6,t: INK 4: PRINT 'LINIER UDTEGNES I RÆKKEFØLGE';: INK 7
1590 PRINT ' EFTER LAVESTE PUNKTNUMMER.'
1600 PRINT TO t; 'DEN HURTIGSTE PLOTTER-UDTEGNING FÅS,'
1610 PRINT TO t; 'NÅR LINIEFARVER OG PUNKTNUMRE GRUPPERES.'
1620 AT 10,t: INK 4: PRINT 'SKÆRM-skala ';start%(10);: INK 7
1630 PRINT ' ( UD FRA MIN- & MAX-VÆRDIERNE ).'
1640 AT 11,t : PRINT 'PARALLELE LINIER UDTEGNES KUN ADSKILT PÅ SKÆRMEN,'
1650 PRINT TO t; 'HVIS FORSKELLEN I x- & y- VÆRDIER ER MINDST ';
1660 INK 4: PRINT start%(12);' cm.': INK 7: AT 13,t
1670 PRINT 'PÅ LODRETTE OG VANDRETTE LINIER SKRIVES NUMRENE KUN'
1680 PRINT TO t;'ADSKILT, HVIS PUNKTERNES AFSTAND  ER MINDST ';
1690 INK 4: PRINT 10 *INT(.35 * start%(12));' cm.': AT 16,t
1692 PRINT'RETUR EFTER FEJL :';:INK 7: PRINT' BRUG < run > ELLER ';
1694 PRINT'< indtast 1 >.': INK 4: AT 18,t: PRINT 'UNDER "DIVERSE" ';
1700 INK 7:PRINT'KAN VÆLGES RENTEGNING UDEN PUNKT-numre.': INK 4:AT 20,t
1710 AT 20,t: PRINT' INSTRUKTIONS-BILLEDERNE ';:INK 7
1720 PRINT ' KAN KALDES SENERE'
1730 AT 21,t: PRINT'VED AT TASTE "i" I STEDET FOR INPUT AF TAL'
1740 INK 2: AT 23,t: PRINT;'ER TALLENE NOTERET ? ';
1750 INPUT'  (KUN xx GENTAGES) '; y$: INK 7
1760 END DEFine parallellinie
```

dato$ og tids$

er en streng på 17 karakterer
3 karakterer til dato og . (1- 3)
6 karakterer til måned og evt. 19 (4- 9)
3 karakterer til (årstal-1900) samt: (10-12)
mellemrum (13)
4 karakterer til timetal med 1 decimal (14-17)

sluttid

proceduren sikrer automatisk korrekt dato, når dato saves. Se "saveproces", linie 3300.

parallellinie

udskriver - udover nogen instruktion, de for den givne tegning relevante oplysninger om skala og minimumsafstande mellem linier og numre, hvis disse ønskes adskilt på skærmen. På plotteren er væsentligt mindre afstande adskilte.

Bemærk brugen af ;

```
PRINT 'aaaa';
PRINT 'xxxx'     GIVER aaaa xxxx

PRINT 'aaaa'
PRINT 'xxxx'     GIVER aaaa
                       xxxx
```

se også brugen af \, som CPA80S printeren udskriver som ■ : linieskift. Der kan benyttes flere \ til at opnå liniemellemrum.

Husk også ; efter PRINTSÆTNINGER i nederste linie. hvis SCOPY skal bruges (se fx. linie 6000),

På en række steder i programmet optræder BORDER

BORDER er 0, hvis BORDER ikke angives.
Hele kanalen udnyttes, i samme farve, fx. hvid.

BORDER n;2,4 giver i kanal n en grøn (4) ramme, der er 2 pixler tyk langs vandrette kanter, 4 pixler langs lodrette kanter(!)

En ny skærmfarve+CLS ændrer kun det, der er inden for BORDER.

BORDER "fratrækkes" i QL's placering af CURSOR osv. Ens karakterer, udskrevet på samme (linie, tab), er derfor ikke sammenfaldende før og efter en ændret BORDER (!)

USYNLIG BORDER (BORDER#n;2)dvs. uden farveangivelse har betydning for karakterplaceringen og for skift af PAPER farve. Kun den del, der er inden for den usynlige BORDER, skifter farve (!)

```
2000 DEFine PROCedure nytegning
2010   LOCal nr, lmax, hmax: REPeat vent01: IF KEYROW(1)= 0: EXIT vent01

2020   qltegn: CLS#4: BORDER#4;1
2050   AT#4; 5,1: PRINT#4;'MINIMUMS-VÆRDIER'
2060   AT#4;10,1: PRINT#4;'MAXIMUMS-VÆRDIER'
2070   AT#4;15,1: PRINT#4;'START-PKT.nr'
2080   AT#4; 0,1: PRINT#4;'START-INPUT:': INK#4;4
2090   AT#4; 1,1: PRINT#4;'MAX / MIN VÆRDIER.'
2100   AT#4; 2,1: PRINT#4;'START-PUNKTETS'
2110   AT#4; 3,1: PRINT#4;'nr OG KOORDINATER.': INK#4;7
2120   FOR p= 0 TO 2: AT#4;6+5*p,9: PRINT#4;'x='
2130   FOR p= 0 TO 2: AT#4;7+5*p,9: PRINT#4;'y='
2140   FOR p= 0 TO 2: AT#4;8+5*p,9: PRINT#4;'z='
2150   vvstegn: INK#4;4
2160   FOR p= 0 TO 2: AT#4;6+5*p,1: PRINT#4;'koor-'
2170   FOR p= 0 TO 2: AT#4;7+5*p,1: PRINT#4;'dinat'
2180   FOR n= 0 TO 2
2190     FOR p= 0 TO 2: AT#4;6+5*n+p,18: PRINT#4;'CM'
2200   END FOR n


2210   qltegn: INK#4;7
2220   talput 4, 6,12, - 20000,19900,1: start%(2)=ktal(1):xmin=start%(2)
2230   talput 4, 7,12, - 20000,19900,1: start%(3)=ktal(1):ymin=start%(3)
2240   talput 4, 8,12, - 20000,19900,1: start%(4)=ktal(1):zmin=start%(4)
2250   talput 4,11,12,xmin+100,20000,1: start%(5)=ktal(1):xmax=start%(5)
2260   talput 4,12,12,ymin+100,20000,1: start%(6)=ktal(1):ymax=start%(6)
2270   talput 4,13,12,zmin+100,20000,1: start%(7)=ktal(1):zmax=start%(7)
2280   talput 4,15,14,1,99,1:           start%(1)= ktal(1)
2290   nr= ktal(1):                  pkt%(nr,1)= ktal(1)
2300   talput 4,16,12,xmin,xmax,1: pkt%(nr,2)= ktal(1)
2310   talput 4,17,12,ymin,ymax,1: pkt%(nr,3)= ktal(1)
2320   talput 4,18,12,zmin,zmax,1: pkt%(nr,4)= ktal(1)
2330   REPeat nyloop
2340     AT#0; 1,1: PRINT#0; 'TALLENE OK ?'
2350     AT#0; 2,1: INPUT#0; 'j / n      '; y$
2360     IF y$== 'n': nytegning
2370     IF y$== 'j': EXIT nyloop
2380   END REPeat nyloop


2390   start%(8)=         (xmin-zmax) * .828: x0= start%(8)
2400   start%(9)= ymin + (xmin+zmin) * .561: y0= start%(9)
2410   xp0%=              (xmin-zmax) * .866
2420   yp0%=       ymin + (xmin+zmin)/2
2430   pkt%(nr,5)= (pkt%(nr,2)-pkt%(nr,4))*.828 -x0
2440   pkt%(nr,6)= (pkt%(nr,2)+pkt%(nr,4))*.561 +pkt%(nr,3) -y0
2450   pkt%(nr,7)= (pkt%(nr,2)-pkt%(nr,4))*.866 -xp0%
2460   pkt%(nr,8)= (pkt%(nr,2)+pkt%(nr,4))/2     +pkt%(nr,3) -yp0%
2470   pkt%(nr,9)= 1000: REMark STARTnr KAN IKKE BARE FJERNES

2480   lmax= (xmax-xmin+zmax-zmin)*.828: start%(10)= lmax * .972
2490   hmax= (xmax-xmin+zmax-zmin)*.561+ymax-ymin
2500   IF hmax * 1.095 >  start%(10): start%(10) = hmax * 1.095
2510   skala= start%(10): start%(12)= start%(10)/110: REMark xxmin
2520 END DEFine nytegning
```

KEYROW(1)= Ø sikrer, at programmet ikke smutter ned til INPUT-linierne, førend den "foregående" tast er sluppet. Overflødigt her, hvor talput ikke lader sig narre (men husk, at talput med sidste parameter > 100 opfatter ENTER som en accept af data, linie 9610). Se vvs, hvor 'r' ændrer linie til radiatorskitse. Se også skærmbilledet pag. 14.

min -20000 (200m) (begge mindre end heltalsgrænsen 32767)
min +19900 (199m)
er indført, fordi tegninger med skala < 100 kan få absurd "skæve" linier som følge af afrundingsfejl. For små tegninger må cm erstattes af mm (eller mindre), således at heltallene bliver rimeligt store, gerne 1000 og opefter.

Der er indlagt en "fortrydelsesløkke", REPeat nyloop, linie 2330.

Bemærk:
pkt%(nr,7 og 8) har de korrekte faktorer for 120^{0}isometri.
Plotter kan tegne ordentligt under enhver vinkel.

Når der på skærmen, pkt%(nr,5 og 6), benyttes en lidt stejlere vinkel, opnås, at linier parallelle med x og y aksen tegnes med tangens=1.355/2, hvor linierne ser regelmæssige ud med to pixler vandret for hvert lodret trin, se TILLÆG, pixler, pag. 92.

tangens=1.355 følger pixelmønstret og kan illudere 45^0 (reelt 53^0) (vvs_tegning).

linie 2390 ff	smlgn. 7700 ff. og linie 12230.
linie 2360	Her udnyttes, at PROCedurer er rekursive.
linie 2510	xxmin=skala/110 for linier og 3½ gang så meget for numre (linie 1690) bygger på en lang række forsøg.

```
2700 DEFine PROCedure loadeproces
2710   LOCal starte$,punkt$,linjer$
2720   qltegn: CLS#4: BORDER#4;4: starte$= flp$ & filnavn$ &'_start'
2730   punkt$=flp$ & filnavn$ &'_pkt': linjer$=flp$ & filnavn$ &'_linje'

2740   AT#4;1,1: PRINT#4;'EXISTERENDE DATA' \ ' FOR'
2750   PRINT#4;' ';filnavn$: PRINT#4;' LOADes FRA '& flp$
2760   OPEN_IN#6; starte$: FOR st= 1 TO 12: INPUT#6; start%(st)
2770   CLOSE#6: AT#4;7,1: PRINT#4;filnavn$; \ ' _start  LOADed';
2780   x0= start%(8): xmin= start%(2): ymin= start%(3): zmin= start%(4)
2790   y0= start%(9): xmax= start%(5): ymax= start%(6): zmax= start%(7)
2800   xp0%= (xmin-zmax) * .866: yp0%= ymin + (xmin+zmin) /2
2810   skala= start%(10): antallinier= start%(11): xxmin= start%(12)

2820   OPEN_IN#6; punkt$
2830   FOR pktnr= 1 TO 99
2840     FOR pos= 1 TO 9: INPUT#6; pkt%(pktnr,pos)
2850   END FOR pktnr
2860   CLOSE#6: AT#4;11,1: PRINT#4;filnavn$: PRINT#4;' _pkt    LOADed'
2870   OPEN_IN#6; linjer$
2880   FOR franr= 1 TO 98
2890    INPUT#6; linje%(franr,1)
2900     FOR p= 2 TO 51 STEP 7
2920      FOR pos= p TO p +6: INPUT#6; linje%(franr,pos)
2930   END FOR p: END FOR franr
2940   CLOSE#6: AT#4;15,1: PRINT#4;filnavn$: PRINT#4;' _linje  LOADed'
2950   BORDER#4;1
2960 END DEFine loadeproces

3000 DEFine PROCedure saveproces
3010   LOCal starte$,punkt$,linjer$,f$,x$,d$,st,pktnr: CLS#4: OVER#4;0
3020   AT#4;1,1: PRINT#4;'FILNAVNET ER NU': PRINT#4;' '; filnavn$
3030   PRINT#4;' TAST  ok  ELLER' \' nyt_filnavn': INPUT#4;' '; x$
3040   f$= flp$(1 TO 4): IF x$=='ok' OR x$=='j': x$= filnavn$

3050   filnavn$= x$: AT#4;7,1: PRINT#4;'ER '& f$ &'  KLAR ?'\' J / N'
3060   REPeat f12
3070    AT#4;8,14:INPUT#4;d$:IF d$=='j'AND FOP_DIR(#3,flp$)=0: EXIT f12
3080   END REPeat f12: CLOSE#3

3090   qltegn: starte$= flp$ & filnavn$ &'_start'
3100   punkt$=flp$ & filnavn$ &'_pkt': linjer$=flp$ & filnavn$ &'_linje'

3110   DELETE starte$: DELETE punkt$: DELETE linjer$: AT#4;10,1
3120   PRINT#4;'PÅ ' & f$ &'  LAGRES' \' NYE DATA FOR' \' ';filnavn$
3130   OPEN_NEW#6;starte$: FOR st= 1 TO 12: PRINT#6; start%(st)
3140   CLOSE#6: AT#4;13,1: PRINT#4; '_start   SAVEd'
3150   OPEN_NEW#6; punkt$
3160   FOR pktnr= 1 TO 99
3170     FOR pos= 1 TO 9: PRINT#6; pkt%(pktnr,pos)
3180   END FOR pktnr: CLOSE#6: AT#4;14,1: PRINT#4; '_pkt     SAVEd'
3190   OPEN_NEW#6; linjer$
3200   FOR franr= 1 TO 98
3210    PRINT#6;linje%(franr,1)
3220     FOR p= 2 TO 51 STEP 7
3230      FOR pos= p TO p +6: PRINT#6; linje%(franr,pos)
3240   END FOR p: END FOR franr
3250   CLOSE#6: AT#4;15,1: PRINT#4; '_linje   SAVEd'
3260   sluttid: AT#4;16,1: PRINT#4; '_dato    SAVEd'
3270   AT#4;17,1: PRINT#4;'dato$ ( slut )  =': PRINT#4;' '; dato$(2)
3280 END DEFine saveproces
```

SAVE/LOAD af et array, dimensioneret med DIM (a,b)

Arrayet har a+1 linier, b+1 søjler, idet 0 kan bruges. Dette er ikke udnyttet her, for overskuelighedens skyld. Exempelvis er det simplere at franr angiver linien for franr's koordinater (franr-1 er for mig "besværligt")

Et array (a,b) har (a+1)*(b+1) pladser. Det defineres (og nulstilles) ved DIM (a,b). Det må SAVEs og LOADes med alle sine pladser, uanset at mange er nuller. Linie% (99,57) har plads til 800 linier, men der er ofte kun 50-100 linier lagret.

SAVE og LOADe - tiden er derfor lang ved denne form for array. Det er imidlertid nemt at overskue og at benytte til søgeprocesser.

Når array'er holdes tæt pakket (dvs. "trunkeres", hvis en linie fjernes, se linie 5200 ff), kan der ved udtegning spares tid ved at overspringe nuller (se fx. linie 6060).

Når noget tilsvarende ikke er indført i linie 12680, er det forsåvidt en fejl. Linie 12680 kunne have været: IF linje %(franr,1)=0: EXIT pos (Der er ikke flere mulige linier for det franr) og linie 12800 kunne stryges. Med en plotter spiller forholdet ingen rolle: Tegnetiden overstiger langt QL's regnetid. Enhver kan prøve at revidere linie 12680,12800,13060 og 13180. Der spares ingen tid.

Bemærk linie 2720/2950: Hvis en procedure omstiller BORDER, skal den selv stille tilbage (tilsv. med TRA, INK, PAPER, OVER osv.)

I saveproces kontrolleres, at flp$ er klar. Gamle data slettes. (Findes der ingen gamle data, fortsætter QL uden fejlmelding).

Datasættet er LOADed under det oprindelige navn. Før SAVE kan navnet ændres - og er så ændret permanent ved nye udskrifter af fx data, skærmkopier, plottertegninger.

Dette er måske/måske ikke en fordel.

Det er en fordel, at SAVE under et nyt navn. Så er risikoen ved fejl under lagringen mindre. Den gamle fil eksisterer da i det mindste. Filer kan altid slettes eller omdøbes (DELETE,RENAME).

ETHVERT DATASÆT SKAL SAVES PÅ TO DISKETTER.

SE I ØVRIGT TILLÆG, pag. 103.

```
3400 DEFine PROCedure skematext
3410   qltegn: CLS#0: CLS#4: OVER#4;0: BORDER#4;1: INK#4;7
3420   AT#4; 0,1: PRINT#4;'FRA PKT.nr'
3430   AT#4; 5,1: PRINT#4;'TIL PKT.nr'
3440   AT#4;10,1: PRINT#4;'TIL PKT.nr'
3450   FOR p= 0 TO 2: AT#4;1+p*5,9: PRINT#4;'x='
3460   FOR p= 0 TO 2: AT#4;2+p*5,9: PRINT#4;'y='
3470   FOR p= 0 TO 2: AT#4;3+p*5,9: PRINT#4;'z='
3480   FOR r= 1 TO 3: AT#4;10+r,8 : PRINT#4; CHR$(119+r)

3490   vvstegn: INK#4;4
3500   AT#4; 1,1: PRINT#4;'koor-': AT#4;15,15: PRINT#4;'# % ('
3510   AT#4; 6,1: PRINT#4;'koor-': AT#4;16,17: PRINT#4;'"<%'
3520   FOR n= 2,7: AT#4; n,1: PRINT#4;'dinat'
3530   AT#4;11,1: PRINT#4;'pkt. ': AT#4;17,17: PRINT#4;'M M'
3540   AT#4;12,1: PRINT#4;'afst.'
3550   FOR c= 0 TO 2
3560     FOR p= 0 TO 2: AT#4; 1 + c*5 + p,18: PRINT#4;'CM'
3570   END FOR c: STRIP#4;0: INK#4;2: UNDER#4;1
3580   AT#4;18,12: PRINT#4;'/ ENTER >': UNDER#4;0: LINE#4; 28,1 TO 49,1

3590   qltegn: PAPER#4;2: INK#4;7
3600   AT#4;15,1: PRINT#4;'FARVE'
3610   AT#4;16,1: PRINT#4;'MATERIALE'
3620   AT#4;17,1: PRINT#4;'DIMENSION'
3630   AT#4;18,1: PRINT#4;'DIVERSE'
3640   FOR n= 0 TO 2: BLOCK#4; 127,2,0,43 + n*50,0

3650   vvstegn: BORDER#0;1: BLOCK#0; 127,7,0,0,0
3660   AT#0;1,1: PRINT#0;'flyt cursor med pile'
3670   AT#0;2, 1: PRINT#0;'ny linie og' \ ' REVISION'
3680   AT#0;4, 1: PRINT#0;'FJERN LINIE' \ ' INPUT    :'
3690   STRIP#0;0: INK#0;2: UNDER#0;1
3700   AT#0;3,12: PRINT#0;'/ SPACE >'
3710   AT#0,4,14: PRINT#0;'/ ESC >'
3720   AT#0;5,12: PRINT#0;'/ ENTER >': UNDER#0;0
3730   FOR n=0,17,34: LINE#0; 90,n TO 160,n

3740   qltegn: PAPER#0;2: INK#0;7
3750   ycurs= 0: nytastnr= 0
3760 END DEFine skematext

4000 DEFine PROCedure indtast (skema)
4010   qltegn: REPeat venttal: IF KEYROW(1)= 0: EXIT venttal
4030   REPeat talloop
4040     IF skema= 1: skematext: skema= 0
4050     OVER#4;-1: CURSOR#4; 72,ycurs*10: PRINT#4;'?': PAUSE 5
4060     CURSOR#4; 72,ycurs*10: PRINT#4;'?': PAUSE 5
4070     pile= KEYROW(1)
4080     SELect ON pile
4090     =1 : IF ycurs= 18 : diverse : REMark ENTER
4100     =4 : ycurs= ycurs-5*(ycurs<14 AND ycurs>9): REMark PIL OPAD
4110     =8 : fjern_linie  : REMark ESC
4120     =64: nytast       : REMark SPACE
4130     =128:
4140          ycurs= ycurs + 5*(ycurs MOD 5=0): REMark PIL NEDAD
4150          IF ycurs= 15: ycurs= 18
4160     END SELect
4170   END REPeat talloop
4180 END DEFine indtast
```

"skematext" viser eksempler på brug af og placering af tekst, såvel almindelige qltegn som små vvstegn.

Hensigten er at tydeliggøre teksten.

Bemærk, se senere procedurer, at eksisterende data udskrives småt, grønt, mens accepterede/nye data udskrives normalt og i hvidt, som orientering for brugeren.

# % (= 2 4 7	vvstegn, oppe (linie 3500 ff)
" < % = 1 - 4	printerudskrift kan se kryptisk ud.
M M = mm	Husk, at en række CHR$-værdier nok kan bruges på skærmen - men med de underligste resultater på CP80-printeren. Se derfor printerens vejledning og karaktersættet for QL.

PROCEDUREN INDTAST (SKEMA)

Se struktur- og procedureoversigten pag. 30.

Brugen af CURSOR, OVER-1 bør studeres i praksis og sammenlignes med TILLÆGs tekst om disse kommandoer.

Det vigtigste er dog, at

indtast (skema) er programmets centrale del, den "løkke", hvorudfra alt foregår.

skema=1 : gå til skematekst først.
skema=0 : fortsæt uden videre.

Linie 4040 : Hvis programmet stoppes med CTRL + mellemrum, og sk rmbilledet er "uskadt", er "indtast 1", hurtigere end "RUN" til genstart af programmet.

```
4200 DEFine PROCedure kantlinie
4210   LINE -venstre, -nedad TO -venstre, opad-nedad
4220   LINE TO højre-venstre, opad-nedad
4230   LINE TO højre-venstre, -nedad TO -venstre, - nedad
4240 END DEFine kantlinie

4500 DEFine PROCedure nytast
4520   qltegn: OVER#4;0: REPeat venttal: IF KEYROW(1)= 0: EXIT venttal

4530   SELect ON ycurs

4540   = 0: REMark FRA PKTnr (nytastnr= 0), I FORELØBIGT ARRAY
4550     talput 4,0,12,1,99,1: franr= INT(ktal(1)): pf%(1)= franr
4560     IF pkt%(franr,1)= 0 THEN
4570       FOR n=1 TO 3: AT#4; n,12: PRINT#4;'-'
4580       talput 4,1,12,xmin,xmax,1: pf%(2)= ktal(1)
4590       talput 4,2,12,ymin,ymax,1: pf%(3)= ktal(1)
4600       talput 4,3,12,zmin,zmax,1: pf%(4)= ktal(1)
4610      ELSE
4620       FOR n=2 TO 4: pf%(n)=pkt%(franr,n):AT#4;n-1,12:PRINT#4;pf%(n)
4630     END IF : REMark NYT ELLER GL.PUNKT. INGEN REVISION

4640     vvstegn: INK#4;4
4650     AT#4;11,1: PRINT#4;'afst.'
4660     AT#4;12,1: PRINT#4;'min.';start%(12): PAPER#4;2: INK#4;7

4670     qltegn: ycurs= 5: nytastnr= 1


4700   = 5 TO 10: REMark TIL PKTnr (nytastnr= 1)
4710     IF nytastnr= 0: ycurs= 0: nytast

4720     talput 4,ycurs,12,1,99,1: tilnr= INT(ktal(1))
4730     IF tilnr = franr: AT#4;ycurs,12: PRINT#4;'NEJ': PAUSE 150
4740     IF tilnr <> franr THEN
4750       fran = franr: w = 1: IF franr > tilnr: fran = tilnr: w = -1
4760       revisionscheck: IF pos> 0: opret_linie
4770     END IF

4780     FOR p= 0 TO 10 STEP 5
4790       FOR n= 0 TO 3: AT#4; p + n, 12: PRINT#4;'       '
4800     END FOR p
4810       FOR n= 0 TO 2: AT#4; 15 + n, 12: PRINT#4;'     '
4820     ycurs= 0: nytastnr= 0

4900   = 18: diverse
4910   END SELect
4920 END DEFine nytast
```

kantlinie — er en indramning, der er lidt pænere end en BORDER.

nytast — franr er enten et nyt nummer, hvis koordinater skal indtastes, eller et existerende nummer, hvis koordinater ikke kan revideres.

Principielt kan revision af franr også indlægges, men så vil de færreste brugere kunne styre brugen.

REVISION KAN KUN GØRES UD FRA ET FAST FRANR TIL ET REVIDERET TILNR.

franr/tilnr/fran/w. Brugeren skal ikke, men programmet skal holde styr på liniens retning: fra et lavere til et højere nummer.

Internt defineres alle linier med franr som det mindste, tilnr som det største nummer.

Derved sikres a) at søgeprocesserne forenkles og b) at linier tegnes i samme retning ved oprettelse, revision, sletning. Exempelvis bliver en sletning (farve INK 0) ikke altid korrekt p.gr. af afrundingsfejl, hvis linien oprettes og fjernes i modsatte retninger.

Se bl.a. — array linje%(98,57), pag. 95
fjern_linie, linie 5000 ff
opret_linie, linie 7200 ff (7830-40 !)
revisionscheck, linie 9200 ff

linie 4760 — pos= 0 : linien kan ikke oprettes: mere end 8 forbindelser. (Se revisionscheck og tilhørende fejludskrift, linie 9200 ff) og linie 8400 ff.

linie 4780 — Eventuel gammel udskrift skal fjernes før næste INPUT-sekvens. Et princip, der ofte bliver glemt i skyndingen. "Livsfarligt", hvis programmet i visse situationer tillader accept af udskrevne DATA: Er det viste så udskrift fra denne eller forrige sekvens?

```
5000 DEFine PROCedure fjern_linie
5010   LOCal xt: REPeat ventese: IF KEYROW(1)= 0: EXIT ventese
5020   qltegn: BORDER#4;0: CLS#4: BORDER#4;1: OVER#4;0
5030   AT#4;1, 1: PRINT#4;'LINIE FJERNES': PRINT#4;\\' FRA PKT.nr'
5040   talput 4,4,13,1,99,1: franr= INT(ktal(1)): nr= franr
5060   AT#4;5,1: PRINT#4; pkt%(nr,2);','; pkt%(nr,3);','; pkt%(nr,4)
5070   AT#4;8,1: PRINT#4;'TIL PKT.nr'
5080   talput 4,8,13,1,99,1: tilnr= INT(ktal(1)): nr= tilnr
5100   AT#4;9,1: PRINT#4; pkt%(nr,2);','; pkt%(nr,3);','; pkt%(nr,4)
5110   IF franr > tilnr: bnr= tilnr: tilnr= franr: franr= bnr

5120   pos= 0
5130   FOR p= 2 TO 51 STEP 7: IF linje%(franr,p)= tilnr: pos= p: EXIT p
5140   IF pos= 0 THEN
5150     AT#4;13,1: PRINT#4;'EN LINIE ML. ': PRINT#4; ' ';franr;' OG ';
5160     PRINT#4; tilnr: PRINT#4;' FINDES IKKE.';\\;' PRØV IGEN.'
5170     PAUSE 200: skema= 1: RETurn
5180   END IF

5190   REMark LINIE, EVT. PUNKT FJERNES. antallinier -1. franr TRUNKERES
5200   vvstegn: rev= -1: tegn_linie franr,pos,0
5210   REPeat trunker
5220     FOR flyt= pos TO pos + 6
5230       IF pos = 51: linje%(franr,flyt)= 0
5240       IF pos < 51: linje%(franr,flyt)= linje%(franr,flyt + 7)
5250     END FOR flyt
5260     IF pos= 51: EXIT trunker
5270     IF linje%(franr,pos + 7)= 0: EXIT trunker
5280     pos= pos + 7
5290   END REPeat trunker

5300   antallinier= antallinier-1: start%(11)= antallinier
5310   pkt%(franr,9)= pkt%(franr,9) -1: pkt%(tilnr,9)= pkt%(tilnr,9) -1
5320   IF pkt%(tilnr,9)= 0: FOR p= 1 TO 8: pkt%(tilnr,p)= 0
5330   IF pkt%(franr,9) > 0 THEN
5340     xt= 0: FOR n= 2 TO 51 STEP 7: xt= xt + linje%(franr,n)
5350     IF xt= 0: linje%(franr,1)= 0: REMark franr er nu kun et tilnr
5360   END IF

5370   IF pkt%(franr,9)= 0  THEN
5380    linje%(franr,1)= 0: FOR p= 1 TO 8: pkt%(franr,p)= 0
5390   END IF

5400   skema= 1
5410 END DEFine fjern_linie

5500 DEFine PROCedure tegn_linie (franr,pos,nrfarve)
5510   LOCal farve, xs1, ys1, xsp, ysp
5520   vvstegn: OVER 1: farve= linje%(franr,pos+4)
5530   tilnr= linje%(franr,pos): INK farve: IF nrfarve= 0: INK 0
5540   xs1= pkt%(franr,5): ys1= pkt%(franr,6)
5550   xsp= pkt%(tilnr,5): ysp= pkt%(tilnr,6)

5560   LINE xs1,ys1 TO xsp,ysp: POINT xsp,ysp: INK nrfarve
5570   IF prnr$=='j' THEN
5580    p%(franr)= p%(franr) + rev: p%(tilnr)= p%(tilnr) + rev
5590    IF p%(franr) + (rev=-1)= 1 OR rev= 0: strengnr franr,xs1,ys1,2,0
5600    IF p%(tilnr) + (rev=-1)= 1 OR rev= 0: strengnr tilnr,xsp,ysp,2,0
5610   END IF
5620   qltegn: OVER 0
5630 END DEFine tegn_linie
```

linie 5040 INT kun nødvendig for at hindre pjat i form af punktnumre med decimaler.
(Der er altid nogen, der bevidst driller).

linie 5110 franr/tilnr ombyttes eventuelt.

linie 5130 tilnr og pos fundet.

linie 5140 der er ingen pos med det tastede tilnr.

linie 5200 linien fjernes med sort liniefarve.

linie 5210 der er - måske - nu et "hul" i linje% (franr,---). "Hullet" udfyldes ved at flytte eventuelle tilnr på højere pos_værdier (incl. tilhørende data) en plads (-7) til venstre. Det tager lidt tid, som brugeren næppe opfatter, og gør udtegning senere hurtigere.

linie 5300 antallinier og antal forbindelser reduceres.
SÅVEL VED FJERN SOM OPRET_LINIE ER
Σ antal forbindelser = $\frac{1}{2}$ (antallinier)
(÷1000, se startpunkt, linie 2470)

linie 5320 pkt% (tilnr,9)= 0 : punktet bruges ikke mere.

linie 5330 punktet bruges stadigvæk (Een forbindelse), men måske ikke som franr, kun som tilnr ?

linie 5400 skema-området genetableres (brugt i linie 5020 ff).

linie 5520 nrfarve bruges til at vise, om linien, evt. punktet skal oprettes/nedlægges.

linie 5590 udskrift af punktnumre sker kun ved nye numre - eller numre, der skal fjernes. Ikke for eksisterende numre.

Mon denne procedure ikke tager mere tid end et (overflødigt) genoptryk af nummeret ? *)

"strengnr" se linie 9900

*) Uanset hvad man (med rette) kan sige om afrundingsfejls uæstetiske indflydelse på linier og kurvers udseende, så afrunder QL altid på samme måde. 50 numre til samme punktkoordinator falder altid (korrekt) oven i hinanden.

```
5900 DEFine PROCedure tegn_exist_data
5910   LOCal nrs, xs, ys
5920   qltegn: BORDER 0: CLS: INK 2: CLS#0: CLS#4: BORDER#4;1

5930   kantlinie: isoprincip: OVER 1: BORDER 1: INK 7

5940     AT#0;2,2: PRINT#0;'SKAL PUNKT nr'\'  UDSKRIVES'\'    J / N ?';
5950   REPeat skrivpkt
5960     AT#0;4,14: INPUT#0; prnr$
5970     IF prnr$ =='j' OR prnr$=='n': EXIT skrivpkt
5980   END REPeat skrivpkt

5990   nrs= start%(1): xs= pkt%(nrs,5): ys= pkt%(nrs,6)
6000   CURSOR xs,ys,-3,-5: PRINT'¤';: REMark CTRL SHIFT W og ;

6010   vvstegn: IF prnr$=='j': strengnr nrs,xs,ys,2,0

6020   rev= 1: FOR n= 1 TO 99: p%(n)= 0
6030   FOR franr=1 TO 98
6040    IF linje%(franr,1) >0 THEN
6050     FOR pos= 2 TO 51 STEP 7
6060       IF linje%(franr,pos)= 0: EXIT pos
6070       tegn_linie franr,pos,7
6080     END FOR pos
6090    END IF
6100   END FOR franr
6105   qltegn: OVER 0: INK 7: prnr$='j': REMark NYE numre
6110 END DEFine tegn_exist_data

6200 DEFine PROCedure isoprincip
6210   BLOCK 79,41,294,210,2
6220   SCALE 345.5,0,0: INK 7: PAPER 2
6228   LINE 306,33 TO 334,14: LINE 334.5,14 TO 362.5,33
6230   LINE 334,14 TO 334,52
6240   LINE 329,45.5 TO 334,52: LINE 334,52.5 TO 340,45
6250   LINE 306,25 TO 306,33 TO 315,33
6260   LINE 362,25 TO 362,33 TO 352,33
6270   AT 24,50: PRINT 'z';: AT 24,60: PRINT 'x';: AT 21,57: PRINT 'y';
6280   PAPER 0: SCALE skala, -skala*11/344, -skala*11/254
6290   REMark SKALA GENOPRETTET. I linie 6270 BRUGES ; AF HENS.T. SCOPY
6300 END DEFine isoprincip

6500 DEFine PROCedure c80program
6502   LOCal begynd, slut
6510   begynd= 1: RESTORE 6560: REMark TRA og BAUD i c80list
6520   FOR n= 1 TO 25
6530     READ slut: IF slut= 0: PRINT#0;' STOPPET ': STOP
6540     c80list  begynd TO slut: begynd= slut
6550   END FOR n

6560   DATA  590,1190,1990,2690,3390,4190,4990,5890,6590,7190,7690
6570   DATA 8390,9190,9990,12590,13590,20890,21690,30990,  0
6580 END DEFine c80program
```

linie 5930 "isoprincip" (linie 6200) vises hver gang af hensyn til den uøvede bruger. Teoretisk kan man definere x,y,z så akavet, at tegningen når ind i området (der fjernes, inclusive eventuelle linier, før printertegning udføres, se linie 8640).
("kantlinie", se linie 4200).

linie 6000 startpunktet har en særstatus med et specielt symbol (o) - og antallet af forbindelser er sat til det reelle tal + 1000, således at det ikke kan fjernes. Denne "fidus" kan fjernes, hvis det ønskes, i linie 2470. Bemærk ; efter ¤ . Startpunktet er oftest nederst på skærmen. Smlgn. kommentar til linie 6270.

linie 6020 p%(99) til optælling af brug/sletning af punktnummer.

linie 6060 her udnyttes, at linje% er "trunkeret".

linie 6100 selv om der er valgt udtegning uden numre, vil nye numre blive udskrevet. Jeg finder det praktisk, men det kan fjernes.

Synspunktet er, at udtegning uden numre (af kendte data) giver et klarere billede, især hvis der er mange linier/punkter ("tegn_linie", linie 5500).

Begynder man at udvide, revidere, bør numrene udskrives som en (nødvendig ?) hjælp.

linie 6200 De små unøjagtigheder i koordinaterne er bevidst indført for at modvirke afrundingsfejl, der skæmmede "iso_logoet".

BEMÆRK linie 6270 + REMARK i linie 6290.
Er der PRINTet karakterer nederst på skærmen, vil SCOPY rykke alt en linie op, hvis der ikke er plads til en karakter på den nederste linie. ; (og ingen udskrift yderst tilhøjre i nederste linie) modvirker denne "fejl". PAUSE 1000 virker også (måske).

C80program er en PROCedure, der ikke kan kaldes uden forudgående STOP. Den giver en pænere udskrift af det samlede program, idet PROCedurer holdes samlet på de enkelte sider.
Til notatet er der benyttet en speciel variant.

```
6600 DEFine PROCedure dataud
6610   LOCal st$(12,6):TRA trap: OPEN#3,ser1: BAUD baudtal
6620   PRINT#3; CHR$(27);"1"; CHR$(5) ; CHR$(27);"Q"; CHR$(77);
6630   PRINT#3; CHR$(27);"C"; CHR$(72); CHR$(27);"N";CHR$(10);
6640   PRINT#3;filnavn$;FILL$(' ',60-LEN(filnavn$));dato$(1)(1 TO 11);\\
6650   PRINT#3;'start%( 12 ) =' \ '(1)     STARTPUNKTnr = '; start%(1)
6652   FOR n=2 TO 9: st$(n)= FDEC$(start%(n),6,0)
6660   PRINT#3;'(2,3,4) x, y, z  min =  '; st$(2),st$(3),st$(4)
6670   PRINT#3;'(5,6,7) x, y, z  max =  '; st$(5),st$(6),st$(7)
6680   PRINT#3;'(8,9)   x0, y0       =  '; st$(8),st$(9)
6690   PRINT#3;'(10)    SKALA = ';start%(10);\'(11)    ANTAL LINIER = ';
6700   PRINT#3; start%(11);\'(12)    MIN.AFSTAND xx = '; start%(12)
6710   PRINT#3;\\\'pkt% ( 99, 9 ) =\'    nr       x       y       z';
6720   PRINT#3;'      xs      ys      xp      yp  forbind'
6730   FOR n= 1 TO 99
6738    IF pkt%(n,1) THEN
6740      PRINT#3: FOR pos=1 TO 9: PRINT#3; FDEC$( pkt%(n,pos), 6,0),
6750   END IF : END FOR n: PRINT#3; CHR$(12);
6760   PRINT#3;filnavn$;FILL$(' ',57-LEN(filnavn$));dato$(1)(1 TO 11);\\
6770   PRINT#3;'linje% ( 98, 57 ) ='
6780   PRINT#3;'franr, tilnr        xx      yy      zz   FARVE, ';
6790   PRINT#3;'MATERIALE, DIMENSION' \\
6800   FOR franr= 1 TO 98
6810    IF linje%(franr,1) THEN
6820     PRINT#3; FDEC$(franr,2,0);' til';
6830     FOR p= 2 TO 51 STEP 7
6840      IF linje%(franr,p)= 0: EXIT p
6848      FOR pos= p TO p + 6
6850        PRINT#3;' ',FDEC$( linje%(franr,pos), 6-4*(pos=p),0);
6860      END FOR pos: PRINT#3
6910   END FOR p: END IF : END FOR franr: PRINT#3; CHR$(12);: CLOSE#3
6920 END DEFine dataud

7000 DEFine PROCedure diverse
7010   LOCal divsvar:  REPeat entervent: IF KEYROW(1)= 0: EXIT entervent
7020   qltegn: BORDER#4;0: CLS#4: BORDER#4;1: OVER#4;0: CLS#0:BORDER#0;0
7030   AT#4;1,1:PRINT#4;'RETUR': AT#4;3,1:PRINT#4;'RENTEGNING'
7032   AT#4;5,1:PRINT#4;'PRINTER-KOPI': AT#4;7,1:PRINT#4;'PLOTTER-KOPI'
7034   AT#4;9,1:PRINT#4;'FEJLKONTROL': AT#4;11,1:PRINT#4;'DATA-UDSKRIFT'
7036   AT#4;13,1:PRINT#4;'DATA-LAGRING':AT#4;15,1:PRINT#4;'NYT DATA-SÆT'
7038   STRIP#4;7: INK#4;2: FOR p=1 TO 8: AT#4;p*2-1,17:PRINT#4;' ';p;' '
7040   PAPER#4;2: INK#4;7: AT#4;17,1: PRINT#4;'TAST  1 . . . 8'
7042   INK#0;0: LINE#0;8,100 TO 156,100: INK#0;7: BORDER#0;1
7044   PRINT#0;' STOP: "CTRL"+"SPACE"'\' PRINTER    "CLOSE#3"'
7046   PRINT#0;' PLOTTER    "CLOSE#5"'\' AFBRYD   SER1 / SER2'
7048   PRINT#0;' TAST    "SAVEPROCES"'\' "RUN" // "INDTAST 1"'

7060   talput 4,17,18,1,8,1: divsvar= INT(ktal(1))
7070   SELect ON divsvar
7080   = 1: BORDER#0;0: skema= 1
7090   = 2: tegn_exist_data: diverse
7100   = 3: udtegn         : diverse
7110   = 4: plotter        : diverse

7120   = 5: fejlkontrol    : diverse
7130   = 6: dataud         : diverse
7140   = 7: saveproces: PAUSE 250: diverse
7150   = 8:
7154        OPEN#5;ram2_:PUT#5 \ 0,'ny', printer$, minitegn, dansk
7156        dobb_vindue: RUN
7160   END SELect
7170 END DEFine diverse
```

linie 6610 QL-printer: TRA=0 altid, undtagen ved plotter.
CP80-printer: TRA 0 til SCOPY, ellers TRA 1.

linie 6620 'l' venstre margin = 5
'Q' højre margin = 77
spaltebredde 72, svarende til max_vindue, kanal 0.
'C' 72 linier på hvert ark
'N' 10 linier overspringes (over perforering)

se den korte BPUT udgave i linie 30450, pag. 73.

linie 6710 tre blanke linier

linie 6740 ; giver ingen mellemrum, mens , giver ΔTAB= 8.

linie 6910 CHR$(12): ny side (FORMFEED)

linie 7156 Smlgn. boot. RUN medfører NEW, og maskinkoderne skal bevares. Filen er en 'ny' fil.

divsvar

=1:	returnerer til INPUT-løkken,	linie 4000
=2:	ny tegning med/uden numre,	linie 5900
=3:	printerkopi af skærm,	linie 8600
=4:	plotter, farvevalg etc.,	linie 12000
=5:	kontrol af (skjulte) fejl,skæring,	linie 20000
=6:	udskrift af DATA på printer,	linie 6600
=7:	SAVE af eksisterende DATA på FLP1	linie 3000
=8:	RUN med load$='ny'	linie 400

Bemærk, at divsvar 2-7 returnerer til "diverse".

```
7200 DEFine PROCedure opret_linie
7210   LOCal fjl, farve: INK#4;4: vvstegn
7220   pt%(1)= tilnr: REMark PAS PA la%(3..5)'s FORTEGN !

7230   SELect ON ycurs
7240   = 5 TO 8
7250   FOR n=1 TO 3: CURSOR#4;0,100,73,10*n+47: PRINT#4; pkt%(tilnr,n+1)

7260   INK#4;7: qltegn
7270   talput 4,6,12,xmin,xmax,101: REMark antal >100
7280   IF ktal(1) <xmin THEN
7290      ktal(1)= pkt%(tilnr,2): AT#4;6,12: PRINT#4; ktal(1)
7300   END IF
7310   pt%(2)= ktal(1): la%(3)= pt%(2)- pf%(2)

7320   talput 4,7,12,ymin,ymax,101
7330   IF ktal(1) <ymin THEN
7340      ktal(1)= pkt%(tilnr,3): AT#4;7,12: PRINT#4; ktal(1)
7350   END IF
7360   pt%(3)= ktal(1): la%(4)= pt%(3)- pf%(3)

7370   talput 4,8,12,zmin,zmax,101
7380   IF ktal(1) <zmin THEN
7390      ktal(1)= pkt%(tilnr,4): AT#4;8,12: PRINT#4; ktal(1)
7400   END IF
7410   pt%(4)= ktal(1): la%(5)= pt%(4)- pf%(4)

7420   = 10 TO 13
7430   FOR x= 1 TO 3
7440     pt%(x+1)= pkt%(tilnr,x+1)
7450     la%(x+2)= ( pt%(x+1) - pf%(x+1) ) * ( pkt%(tilnr,1) >0 )
7460     CURSOR#4; 0,100,73,10*x+97: PRINT#4; la%(x+2)
7470   END FOR x

7480   INK#4;7: qltegn
7490   talput 4,11,12,xmin-pf%(2),xmax-pf%(2),101
7500   IF ktal(1) < xmin-pf%(2) THEN
7510      ktal(1)= la%(3): AT#4;11,12: PRINT#4; ktal(1)
7520   END IF
7530   pt%(2)= pf%(2)+ ktal(1): la%(3)= ktal(1)

7540   talput 4,12,12,ymin-pf%(3),ymax-pf%(3),101
7550   IF ktal(1) < ymin-pf%(3) THEN
7560      ktal(1)= la%(4): AT#4;12,12: PRINT#4; ktal(1)
7570   END IF
7580   pt%(3)= pf%(3)+ ktal(1): la%(4)= ktal(1)

7590   talput 4,13,12,zmin-pf%(4),zmax-pf%(4),101
7600   IF ktal(1) < zmin-pf%(4) THEN
7610      ktal(1)= la%(5): AT#4;13,12: PRINT#4; ktal(1)
7620   END IF
7630   pt%(4)= pf%(4)+ ktal(1): la%(5)= ktal(1)
7640   END SELect

7650   fjl= 0: fejlcheck: IF fjl= 1: RETurn : REMark linielængde > 0
7690 REMark fortsættes
```

forneden pag. 55

OPRET_LINIE

Kendte data i grønt med vvstegn. Kan accepteres, og ændres da til hvidt.

Nye/reviderede data i hvidt med qltegn. Nyt INPUT, hele tallet.

talput med 6.parameter = 101 : ENTER alene betyder accept af eksisterende data (se "talput" linie 9500)

ycurs= 5 TO 8 : absolutte koordinater

pt%(4) og la%(5) er foreløbige værdier for pkt% og linje%

ycurs= 10 TO 13 : relative koordinater

"tput" (TILLÆG, pag. 85 ff) benytter en afvigende teknik til at adskille det udskrevne, tidligere gældende fra det nye/accepterede: Udskrift i FDEC$ feltet i kontrastfarve, der ændres til normalt PAPER+INK efter "ENTER". Tallene udskrives også med korrekt højremargen. FDEC$, se TOOLKIT2. I "talput" skal hele tallet revideres. I "tput" kan man nøjes med at revidere et enkelt ciffer!

linie 7660 Her kan indføres kontrol af x,y,z,Δx,Δy,Δz etc., hvis programmet gør det ønskeligt. Alle værdier findes i de to foreløbige arrays. Smlgn. vvs_tegning, hvor alle linier skal være akseparallelle, dvs. een og kun een af Δx,Δy,Δz skal være <> 0. Nu kan de foreløbige koordinater indlægges i de endelige arrays, se linie 7700 ff nedenfor.

```
7700   FOR n= 1 TO 4: pkt%(franr,n)= pf%(n): pkt%(tilnr,n)= pt%(n)
7710   pkt%(franr,5)= (pf%(2)-pf%(4))*.828 -x0
7720   pkt%(franr,6)= (pf%(2)+pf%(4))*.561 + pf%(3) -y0
7730   pkt%(franr,7)= (pf%(2)-pf%(4))*.866 -xp0%
7740   pkt%(franr,8)= (pf%(2)+pf%(4))/2    + pf%(3) -yp0%
7750   pkt%(tilnr,5)= (pt%(2)-pt%(4))*.828 -x0
7760   pkt%(tilnr,6)= (pt%(2)+pt%(4))*.561 + pt%(3) -y0
7770   pkt%(tilnr,7)= (pt%(2)-pt%(4))*.866 -xp0%
7780   pkt%(tilnr,8)= (pt%(2)+pt%(4))/2    + pt%(3) -yp0%
7790   pkt%(franr,9)= pkt%(franr,9) + rev: REMark SE revisionscheck
7800   pkt%(tilnr,9)= pkt%(tilnr,9) + rev: REMark SMLGN. slet_linie
7810   antallinier  = antallinier   + rev: start%(11)= antallinier
7820   linje%(fran,1)= fran
7830   linje%(fran,pos)= tilnr: IF w < 0 : linje%(fran,pos)= franr
7840   FOR n= 1 TO 3: linje%(fran,pos+n)= la%(n+2) * w
```

```
7850 REMark opret_farve etc : linien existerer
7860   IF linje%(fran,pos+4)= 0: linje%(fran,pos+4)= 2
7870   REMark UNDGÅ SORTE LINIER VED SNYDERI / PROGRAMAFBRYDELSE

7880   INK#4;4: vvstegn: FOR x=4 TO 6
7890     CURSOR#4;0,100,73,10*x+107: PRINT#4; linje%(fran,pos+x)
7900   END FOR x: INK#4;7: qltegn

7910   REPeat farveloop
7920     talput 4,15,12,2,7,101: ktal(1)= INT(ktal(1))
7930     IF ktal(1)= 1: ktal(1)= linje%(fran,pos+4)
7940     IF ktal(1) >1: EXIT farveloop
7950   END REPeat farveloop
7960   farve= INT(ktal(1))
7970   SELect ON farve= 2 TO 3: linje%(fran,pos+4)= 2
7980   SELect ON farve= 4 TO 5: linje%(fran,pos+4)= 4
7990   SELect ON farve= 6 TO 7: linje%(fran,pos+4)= 7
8000   AT#4;15,12: PRINT#4; linje%(fran,pos+4)

8010   REPeat materialeloop
8020     talput 4,16,12,0,4,101: ktal(1)= INT(ktal(1)): REMark min = 0 !
8030     IF ktal(1)= -1: ktal(1)= linje%(fran,pos+5)
8040     IF ktal(1)>-1: EXIT materialeloop
8050   END REPeat materialeloop: linje%(fran,pos+5)= ktal(1)
8060   AT#4;16,12: PRINT#4; linje%(fran,pos+5)

8070   REPeat dimensloop
8080     IF linje%(fran,pos+5)=0: ktal(1)=0: EXIT dimensloop
8090     talput 4,17,12,5,1000,101: ktal(1)= INT(ktal(1))
8100     IF ktal(1)= 4: ktal(1)= linje%(fran,pos+6)
8110     IF ktal(1) >4: EXIT dimensloop
8120   END REPeat dimensloop: linje%(fran,pos+6)= ktal(1)
8130   AT#4;17,12: PRINT#4; linje%(fran,pos+6)

8140   tegn_linie  fran,pos,7
8150 END DEFine opret_linie
```

Nu er de foreløbige værdier indlagt i de endelige arrays. For at hindre snyd/fejl, sættes liniefarven til rød. Teoretisk kan denne sikkerhedsforanstaltning bryde sammen ved strømsvigt eller bevidst STOP, netop som linie 7700-7840 gennemløbes.

Farve kan benyttes til at skelne mellem primære og sekundære forbindelser, mellem forgrund og baggrund. Den kan benyttes til at indikere funktion, materiale, tværsnitsform.

Materiale og dimension kan benyttes tilsvarende. Mere end tre parametre kræver ændring af menuens skærmlayout.

Her kan tastes 0 for materiale, hvorefter dimension overspringes.

Husk problemerne ved accept med ENTER. Se materialeloop!

```
8400 DEFine PROCedure fejlmeld1
8410   qltegn: BORDER#4;0: CLS#4: BORDER#4;1
8420   AT#4;2,1: PRINT#4;'DER MÅ IKKE OPRETTES'
8430     PRINT#4;' MERE END OTTE LINIER'
8440     PRINT#4;' FRA ET PUNKT.': PAUSE 200
8450 END DEFine fejlmeld1

8500 DEFine PROCedure fejlcheck
8510   IF ABS (la%(3)) + ABS (la%(4)) + ABS (la%(5)) = 0 THEN
8520     qltegn: BORDER#4;0: CLS#4: BORDER#4;1
8530     AT#4;2,1: PRINT#4; 'LINIELÆNGDE NUL' \ ' ACCEPTERES IKKE'
8540     PAUSE 200: fjl= 1: skema= 1
8550   END IF
8560 END DEFine fejlcheck

8600 DEFine PROCedure udtegn
8610   OPEN#3,ser1: BAUD baudtal: TRA trap: BPUT#3, 27,108,10;
8620   PRINT#3; filnavn$; FILL$(' ',36-LEN(filnavn$)); dato$(1)(1 TO 11)
8630   CLOSE#3
8640   BORDER 0: BLOCK 79,41,294,210,0: TRA trat: SCOPY#1: isoprincip
8650   BORDER 1: OPEN#3,ser1: PRINT#3; CHR$(12);: CLOSE#3: TRA trap
8660 END DEFine udtegn

8900 DEFine PROCedure iso_vindue
8910   OPEN#4; con_132x194a380x0_128
8920   WINDOW#1; 512,256,   0,   0: PAPER#1;0: CLS#1
8930   WINDOW#0; 132, 62,380,194: PAPER#0;2: INK#0;7
8940   WINDOW#1; 380,256,   0,   0: PAPER#1;0: INK#1;7
8950   WINDOW#2; 380,256,   0,   0: PAPER#2;4: INK#2;0: BORDER#2; 1,2
8960   WINDOW#4; 132,194,380,   0: PAPER#4;2: INK#4;7
8970 END DEFine iso_vindue

9000 DEFine PROCedure vvstegn
9010   FOR n= 0,1,2,4: CHAR_USE#n; minitegn,0
9020 END DEFine vvstegn

9100 DEFine PROCedure qltegn
9110   FOR n= 0,1,2,4: CHAR_USE#n; 0,dansk
9120 END DEFine qltegn
```

"fejlmeld1"	hører til "revisionscheck", linie 9200.
"fejlcheck"	hører til "opret_linie, linie 7650
linie 8610	venstre margen 10 .
linie 8640	BLOCK fjerner isoprincip før SCOPY (printer-kopi).
linie 8940	PAPER 0= sort, idet dette giver den tydeligste tre-farve stregtegning (rød,grøn,hvid)
linie 9010	CHAR_USE fra TOOLKIT2. Tegnene defineret ved UDC (User Defined Characters), et separat BASIC-program.
iso_vindue	benyttes til tegning, se også de to vinduer i linie 30000 ff.

```
9200 DEFine PROCedure revisionscheck
9202   LOCal fra: fra= pkt%(franr,9) - 1000 *(pkt%(franr,9) > 100)
9210   pos= 0: rev= 0

9220   FOR p= 2 TO 51 STEP 7
9230     IF w = 1 : IF linje%(franr,p)= tilnr: pos=p: rev= 0: EXIT p
9240     IF w = -1: IF linje%(tilnr,p)= franr: pos=p: rev= 0: EXIT p
9250   END FOR p

9260   FOR p=2 TO 51 STEP 7
9270     IF pos >0: EXIT p
9280     IF linje%(fran,p)= 0: pos= p: rev= 1: EXIT p
9290   END FOR p

9300   IF pos= 0 OR fra + rev > 8: pos= 0: skema= 1: fejlmeld1
9310   REMark rev=0 =REVISION.    rev=1 =NY LINIE.    rev=-1 =FJERN LINIE
9320 END DEFine revisionscheck

9500 DEFine PROCedure  talput ( kanal,lin,tab,min,max,antal )
9510  LOCal e$,k,p,lenn,n,tal,tal$
9530  e$= '': IF antal >100: e$='ent'

9550   REPeat tastind
9560    AT# kanal; lin,tab: INPUT#kanal; tal$

9570    IF tal$ & load$ == 'igl': instruktion: RUN

9580    k= 0: p= 0: lenn= LEN (tal$)
9590    REPeat kontrol
9600     IF lenn= 0 THEN
9610       IF KEYROW(1)=1 AND e$='ent': ktal(1)= min -1: EXIT tastind
9620       EXIT kontrol
9630     END IF

9640     IF tal$(1)= '-': k= 1
9650     FOR n=1+k TO lenn
9660       IF tal$(n) INSTR '0123456789': k= k+1
9670       IF tal$(n) INSTR '.': p= p+1 : k= k+1
9680     END FOR n

9690     IF k < lenn OR p > 1: EXIT kontrol: ELSE tal= tal$
9700     SELect ON tal= min TO max: ktal(1)= tal: EXIT tastind
9710     EXIT kontrol
9720    END REPeat kontrol

9730      AT#kanal; lin,tab: PRINT#kanal; FILL$(' ',lenn)
9740   END REPeat tastind
9770 END DEFine talput


9900 DEFine PROCedure strengnr ( nrs,xs,ys,xnr,ynr )
9910   nr$= nrs: IF nrs <10: nr$= '0'&nr$
9920   FOR n= 1 TO 2
9930     CURSOR xs,ys, xnr+(n-1)*4, ynr: PRINT CHR$( CODE( nr$(n)) -15 )
9960   END FOR n
9970 END DEFine strengnr
```

"revisionscheck"

smlgn. linie 4600 og "fejlmeld1", linie 8400 ff.
linie 9220-50: Eksisterer linien ?
linie 9260-90: Er der en ledig pos ?

"talput" - SPARER EN UTROLIG MASSE LINIER,
HVER GANG NOGET SKAL INPUT'tes.

INPUT af streng, der kontrolleres for sine karakterer.

I vvs_tegning er det indbygget, at nr = R42 opfattes som en radiator med franr = 42. Indtastes "i", fås "instruktion", linie 200 ff.

Det kontrolleres, at strengen er et tal, eventuelt negativt, mellem minimums- og maximumsværdierne.

Heltal såvel som decimaltal godkendes.

"talput" svarer herudover til en række kommandoer som AT#n,linie, tab: INPUT#n,y$, eventuelt i en REP-løkke.

"talput" har 6 parametre:

kanal	
lin,tab	linie og TAB for INPUT
min,max	for INPUT
antal	kunne udnyttes til at etablere en indtastløkke. Her er der kun eet INPUT, og antal =101 angiver, at eksisterende DATA skal accepteres (e$='ent'). Ktal(1) bliver midlertidigt sat til min-1. Se fx talput i opret_linie, linie 7280.
k og p	bruges til at optælle, at antallet af korrekte karakterer stemmer med tallets længde. (- se linie 9690):

"talput" giver fornyet INPUT-mulighed, hvis der indtastes noget forkert (andet end tal, eller tal uden for grænseværdierne).

Smlgn. "tput" (TILLÆG, pag. 85 ff) en fornyet - og bedre - version, en blanding af Chr. B. Lund's procedure og min "talput". Se også teksten til "opret_linie" pag. 55.

strengnr — se anvendelsen af CURSOR med 4 parametre til korrekt placering af tal, se TILLÆG pag. 94.

I vvs_tegning anbringes tallene over/under og til venstre/højre for punktet, afhængigt af liniens farve.

```
12000 DEFine PROCedure plotter: REMark isometri,SERVOGOR
12010   LOCal u$, pfarve(5), rgh, lpmax, hpmax, pskala

12030   qltegn: BORDER#4;0: CLS#4: BORDER#4;1: INK#4;7
12040   INK#4;4: CLS#4: AT#4;0,1: PRINT#4;'OVERSKRIFT:'
12050   INK#4;7: INPUT#4;' '; over$
12060   INK#4;4: AT#4;5,1: PRINT#4;'UDSKRIFT AF NUMMER'
12070   PRINT#4;' ELLER AF DIMENSION'
12080   INK#4;7: PRINT#4;' TAST  n FOR NR'\'        d FOR DIMENS'
12090   PRINT#4;'        b BEGGE DELE'\'        i INGEN TAL'

12100   REPeat loop1
12110    AT#4;10,1: INPUT#4; u$
12120    IF u$=='n' OR u$=='d' OR u$=='b' OR u$=='i': EXIT loop1
12130   END REPeat loop1

12140   INK#4;4: AT#4;11,1: PRINT#4;'PEN 7 TIL RAMMEN.'
12150   PRINT#4;' ANGIV PEN-NR FOR'\' TAL OG LINIER :'
12160   INK#4;7: AT#4;14,1: PRINT#4;'NR-FARVE =  '\' DIM-FARVE  ='
12170   PRINT#4;' SKÆRM rød  ='\' SKÆRM grøn ='\' SKÆRM hvid ='

12180   FOR rgh=1 TO 5
12190    IF (u$=='d' OR u$=='i') AND rgh= 1: NEXT rgh
12200    IF (u$=='n' OR u$=='i') AND rgh= 2: NEXT rgh
12210    talput 4,13+rgh,18,0,7,1: pfarve(rgh)= INT(ktal(1) + 1)
12220   END FOR rgh

12230   lpmax= (xmax -xmin +zmax -zmin)*.866
12240   hpmax= (xmax -xmin +zmax -zmin)/2 + ymax - ymin
12250   pskala= hpmax/2500: IF lpmax/3600 >pskala: pskala= lpmax/3600
12260   pskala= hpmax/2600: IF lpmax/3180 >pskala: pskala= lpmax/3180

12270   plotter_open: plot_exist_data
12280   IF u$=='n' OR u$=='b': plot_numre
12290   IF u$=='d' OR u$=='b': plot_dimension

12300   PRINT#5; 'F0H0/0K'
12310   CLS#4: AT#4;2,1: PRINT#4;'ER PENNENE FJERNET ?'
12320   REPeat loop2: AT#4;3,20: INPUT#4; y$: IF y$=='j': EXIT loop2
12330   PRINT#5; CHR$(3): CLOSE#5
12340 END DEFine plotter

12400 DEFine PROCedure plotter_open
12410   OPEN#5,ser2: BAUD 1200: PRINT#5; CHR$(1); 'P'
12420   PRINT#5;CHR$(35);4
12430  FOR n=1 TO 2
12440   PRINT#5;'W';0,INT(200.5+lpmax/pskala),0,INT(200.5+hpmax/pskala)
12450   PRINT#5;'F 8';'V'
12460  END FOR n
12470   PRINT#5;'H';30;'/';INT(hpmax/pskala +140); 'K'
12480   PRINT#5;'Z';40,0,40

12490   TRA 1: PRINT#5;'B';over$; CHR$(13): TRA trap
12500   PRINT#5;'H';INT(lpmax/pskala-275);'/';INT(hpmax/pskala+140); 'K'
12510   PRINT#5;'B';'cop JMP 86'; CHR$(13)
12520   PRINT#5;'H';INT(lpmax/pskala-180);'/';30; 'K'
12530   PRINT#5;'Z';30,0,30
12540   PRINT#5;'B';dato$(1)(1 TO 11); CHR$(13)
12550 END DEFine plotter_open
```

SERVOGOR 281 (BBC) PLOTTER

plotter (-start i kanal 5)

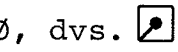
pfarve: Der kan anvendes fem stregfarver, flere slags penne.

linie 12170 ff	Der skal angives farve for nummer og dimension (hvis de ønskes udskrevet) Der SKAL angives pennefarver for alle tre skærmfarver, selvom der kun er brugt 1 eller 2 farver. DETTE BURDE ÆNDRES. Det er akavet at angive farve på noget, man ikke bruger.
linie 12230 ff	smlgn. analoge tal i line 2390 ff.
linie 12300	FO=pen retur. H=PEN UP. 0/0= 0,0. K= flyt til x,y
linie 12330	CHR$(3)= PLOTTER OFF
linie 12410	CHR$(1);'p'= PLOTTER ON Plotter via ser2. BAUD 1200 (SERVOGOR PLOTTER)
linie 12420	CHRS(35);4= karaktersæt 4, DANSK.
linie 12440	W= DEFINE WINDOW, xmin,xmax,ymin,ymax. Mål i 1/10 mm. Her etableres 20 mm afstand til ramme. F= PEN med pfarve(1). V= tegnramme. To gange, for at rammen ikke skal have "svage" steder.
linie 12470	Hæv pen, flyt til 30, INT(---).
linie 12480	skrift 40 mm høj, hældning 0, bredde 40.
linie 12490	overskrift. CHR$(13)= CR= stop karakterudskrift. TRA 1 for at få Æ,Ø,Å udskrevet på plotter.

```
12600 DEFine PROCedure plot_exist_data
12610   LOCal snr, tilnr, sfarve: snr= start%(1)
12620   PRINT#5;'H';INT(pkt%(snr,7)/pskala+100.5);
12630   PRINT#5;'/';INT(pkt%(snr,8)/pskala+100.5);'K'
12640   PRINT#5;'Z';20,0,20: PRINT#5;'M';0
12650   FOR franr=1 TO 98
12660    IF linje%(franr,1) >0 THEN
12670     FOR pos= 2 TO 51 STEP 7
12680      IF linje%(franr,pos) >0 THEN
12690       tilnr= linje%(franr,pos): sfarve= linje%(franr,pos+4)
12710       IF sfarve= 2: PRINT#5;'F';pfarve(3)
12720       IF sfarve= 4: PRINT#5;'F';pfarve(4)
12730       IF sfarve >6: PRINT#5;'F';pfarve(5)
12760       PRINT#5;'H';INT(pkt%(franr,7)/pskala+100.5);
12770       PRINT#5;'/';INT(pkt%(franr,8)/pskala+100.5);'K'
12780       PRINT#5;'I';INT(pkt%(tilnr,7)/pskala+100.5);
12790       PRINT#5;'/';INT(pkt%(tilnr,8)/pskala+100.5);'K'
12800      END IF
12810     END FOR pos
12820    END IF
12830   END FOR franr
12840 END DEFine plot_exist_data

13000 DEFine PROCedure plot_dimension
13010  LOCal dimens, korrx, korry, xxpl, yypl, xb%, yb%
13020  PRINT#5;'F'; pfarve(2): REMark 2 mm SKRIFT
13030  FOR franr=1 TO 98
13040   IF linje%(franr,1) >0 THEN
13050    FOR pos= 2 TO 51 STEP 7
13060     IF linje%(franr,pos) >0 THEN
13070      tilnr= linje%(franr,pos)
13080      dimens = linje%(franr,pos+6)
13090      korrx = 10: korry= -10
13100       xxpl= ABS (pkt%(tilnr,7) - pkt%(franr,7))
13110       yypl= ABS (pkt%(tilnr,8) - pkt%(franr,8))
13120      IF 1.8*ABS(yypl)<=ABS(xxpl): korrx=-9* LEN(dimens): korry=-30
13130      xb%= (pkt%(franr,7)+pkt%(tilnr,7)) / 2 / pskala +100 +korrx
13140      yb%= (pkt%(franr,8)+pkt%(tilnr,8)) / 2 / pskala +100 +korry
13150      PRINT#5; 'H'; xb%; '/'; yb%; 'K'
13160      PRINT#5; 'Z'; 20,0,20
13170      PRINT#5; 'B'; dimens; CHR$(13)
13180     END IF
13190    END FOR pos
13200   END IF
13210  END FOR franr
13220 END DEFine plot_dimension

13400 DEFine PROCedure plot_numre
13410   LOCal pnr, nr$, xb%, yb%
13420   PRINT#5;'F'; pfarve(1)
13430   FOR punk= 1 TO 99
13440    IF pkt%(punk,1) > 0 THEN
13450     pnr= pkt%(punk,1): nr$= pnr: IF pnr< 10: nr$= '0'&nr$
13460     xb%= pkt%(punk,7)/pskala +109
13470     yb%= pkt%(punk,8)/pskala + 72
13480     PRINT#5; 'H'; xb%; '/'; yb%; 'K'
13490     PRINT#5; 'Z'; 20,0,20;
13500     PRINT#5; 'B'; nr$; CHR$(13)
13510    END IF
13520   END FOR punk
13530 END DEFine plot_numre
```

linie 12640 startpunkt får symbol MØ, dvs. 

linie 12700 penfarve vælges.

linie 12760 linier tegnes.
H= PEN UP. Flyt (K) til næste x,y
I= PEN NED. Flyt til næste x,y

plot-dimension og plot_numre

er mere kompliceret i vvs_tegning med skrå skrift i radiatorer i z-axens retning, og dimensioner/numre over/under, til venstre/højre for røde/grønne linier.

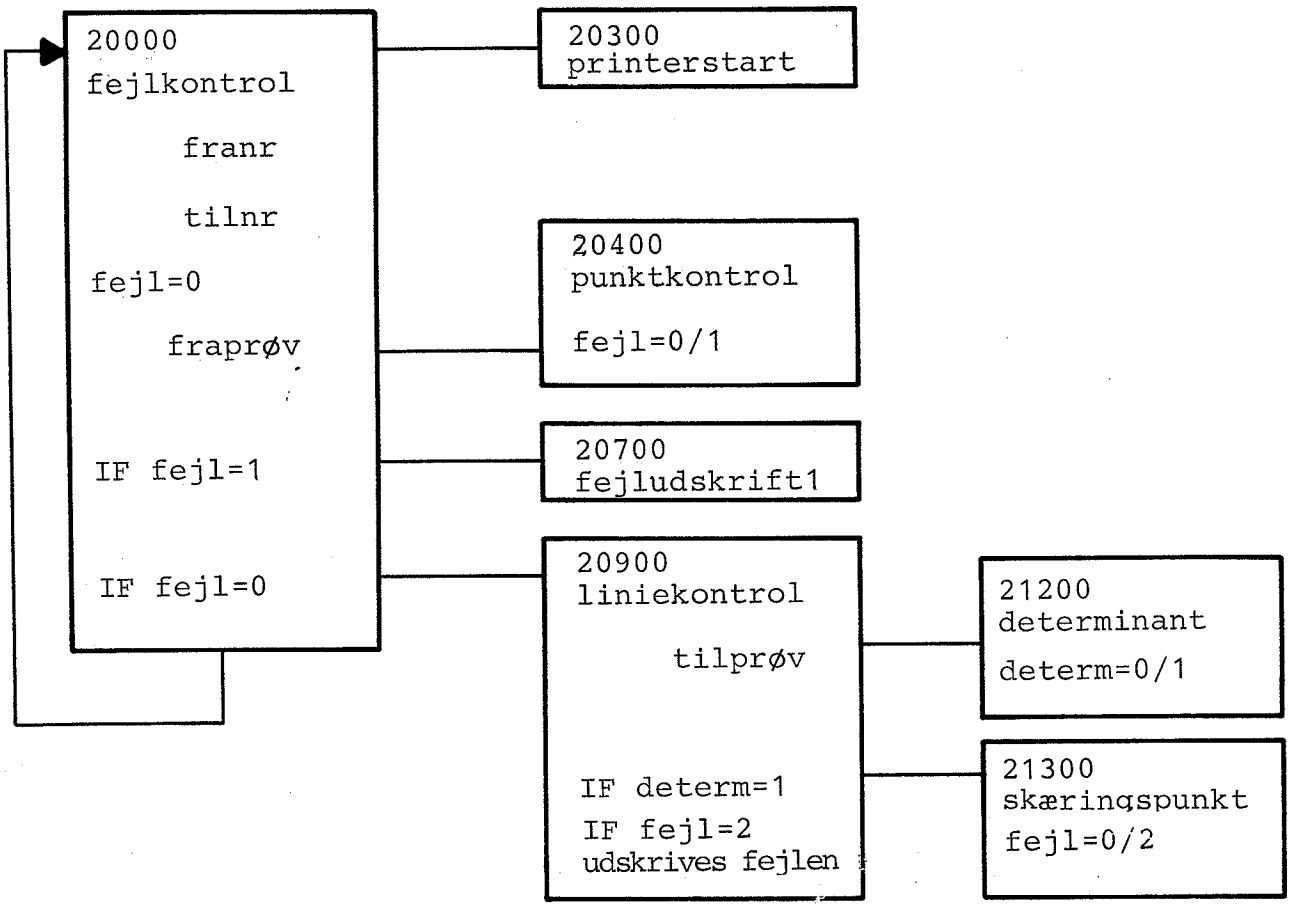
Her kan vi nøjes med korrx og korry (0,8 og 1 mm korrektion) i forhold til punktet (og i forhold til vandrette/lodrette linier gennem punktet). Bemærk, at plotterens karakterer har nulpunkt i nederste, venstre hjørne (printeren i øverste, venstre hjørne).

xxpl og yypl: liniens hældning får indflydelse på, om karaktererne (tallene) skal flyttes lidt i forhold til liniens midtpunkt: linie 13120 ændrer eventuelt korrx og korry.

20 mm skrift.

linie 13460-70 Tallene 109 og 72 placerer nummeret til højre for, neden for punktet, hvilket 'ser' helt korrekt ud, hvis der kun går en lodret og en vandret linie gennem punktet.

FEJLKONTROL



LINIERNE BESKRIVES VED PARAMETERLIGNINGER, IDET DER BENYTTES DE DATA, DER I FORVEJEN FINDES I TO ARRAYS:

pkt%(99,9) pkt%(nr,1) = pktnr
pkt%(nr,2---4) = x--y--z

linje%(99,57) linje%(franr,1) =franr
linje%(franr,pos) = tilnr
linje%(franr,pos+1---3) = Δx--Δy--Δz
hvor pos = 2,9,16--51 (max 8 linier pr. franr)

Parameter f: f<0 eller f>1 : udenfor linien (figur pag. 67)
f=0 eller f=1 : endepunkter
0< f < 1 : på linien

Alle numre checkes mod andre punkter (koordinater)

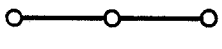
Alle linier checkes mod alle punkter (franr-tilnr // fraprøv)

Alle linier checkes mod alle linier (franr-tilnr // fraprøv-tilprøv)
(startnr 2 > startnr 1)

FEJLMULIGHEDER

- som er skjulte (især hvis punktnumre ikke udskrives)
- som er skjulte for den uopmærksomme
- som er skjulte, hvis fx. et 8-tal skrives over et 3-tal
- eller som i hvert fald er lumske og bør checkes:
- som skal udskrives, ikke rettes, da de kan være ok.

◉ to punktnumre med samme x,y,z-koordinater

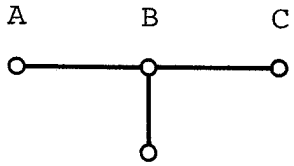


A-B og A-C er definerede
- ikke A-B og B-C

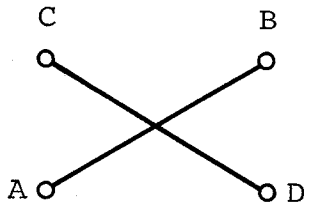
(flere varianter da linier defineres fra lavt til højt nr. internt, uanset hvad brugeren taster)



kun A-D og B-C er definerede (!)
(variant af tilfældet ovenfor)
(ok for bjælke B-C på mur A-D)



kun A-C og B-D er definerede
(ok for B som bevægelig langs AC)



skæringen eksisterer, men skæringspunktet er ikke defineret
(ok for bardun-afstivning, kryds mellem slappe rør osv.)

Alle disse exempler kan være fejl, men også bevidste og ok.

Hertil skal føjes, at også punkter "tæt ved" en linie bør undersøges for eventuelle fejl.

Fejlmulighederne udskrives på printeren, men det er op til brugeren at afgøre, om der er fejl.

Proceduren varer lang tid, op til 30 minutter for 90 linier, og bør derfor forbedres.

Det var tilstræbt, at lade "punktkontrol" klare så meget som muligt - og lade de "billige (hurtige) IF'er sortere hurtigt. Alt ud fra simple, to-dimensionale overvejelser.

```
20000 DEFine PROCedure fejlkontrol
20010 LOCal fra(3),til(3),fratil(3),prøv(3),pft(3),det(3),f,fejl,tilprøv

20020 LOCal tnr: printerstart: REMark iso-UDGAVE: kontrol MED 2 * xxmin
20040  FOR franr= 1 TO 98
20050  IF linje%(franr,1) > 0 THEN
20060    FOR n=1 TO 3: fra(n)= pkt%(franr,n+1)
20070    FOR pos= 2 TO 51 STEP 7
20080     IF linje%(franr,pos) = 0 : EXIT pos
20090     tilnr= linje%(franr,pos)
20100     tnr= tilnr +(tilnr< 99): FOR n=1 TO 3: til(n)= pkt%(tilnr,n+1)
20110     FOR n=1 TO 3: fratil(n)= linje%(franr,pos+n)
20120     FOR fraprøv= 2 TO franr-1, franr+1 TO tilnr-1, tnr TO 99
20130      IF tilnr= 99: IF fraprøv= 99: NEXT pos
20140      IF pkt%(fraprøv,1) > 0 THEN
20150       FOR n=1 TO 3: prøv(n)= pkt%(fraprøv,n+1)

20160       fejl= 0    : punktkontrol: IF fejl=1: fejludskrift1
20170       IF fejl= 0:IF fraprøv > franr: IF fraprøv < 99: liniekontrol

20180      END IF
20190     END FOR fraprøv
20200    END FOR pos
20210  END IF : END FOR franr: PRINT#3; CHR$(12);: CLOSE#3
20220 END DEFine fejlkontrol

20300 DEFine PROCedure printerstart
20310   AT#4;17,1: PRINT#4;'PRINTER KLAR  J / N '
20320   REPeat pstart: AT#4;18,19: INPUT#4;y$: IF y$=='j': EXIT pstart
20330   AT#4;17,1:PRINT#4;'DET KAN TAGE 30 MIN.'\' MOTION ER SUNDT !    '
20340   OPEN#3,ser1: BAUD baudtal: TRA trap
20350   BPUT#3; 27,108,5,27,81,77,27,67,72,27,78,10
20360   PRINT#3;'FØLGENDE PUNKTER / LINIER BØR CHECKES FOR EVT. FEJL:'\\
20370 END DEFine printerstart

20400 DEFine PROCedure punktkontrol
20410   LOCal n, m, a, b, diff
20420   IF ABS(fratil(1)) >= ABS(fratil(2)) THEN
20430     IF ABS(fratil(1)) >= ABS(fratil(3)): n= 1: ELSE : n= 3
20440    ELSE
20450     IF ABS(fratil(2)) >= ABS(fratil(3)): n= 2: ELSE : n= 3
20460   END IF
20465   diff= 2 * xxmin: IF fratil(n) < 0: diff= -diff
20470   m= ( prøv(n)- fra(n) + diff ) * ( til(n)- prøv(n) + diff )
20480   IF m < 0: RETurn
20500   f=(prøv(n)-fra(n)) / fratil(n): a= n+1 -3*(n=3): b= n-1 +3*(n=1)
20510   IF ABS(fra(a) + fratil(a) * f - prøv(a)) > 2 * xxmin: RETurn
20520   IF ABS(fra(b) + fratil(b) * f - prøv(b)) > 2 * xxmin: RETurn
20530   fejl= 1
20540 END DEFine punktkontrol

20700 DEFine PROCedure fejludskrift1
20710   IF f== 0 OR f== 1 THEN
20720     PRINT#3; 'PÅ LINIE '; franr;'-'; tilnr;' ER';
20730     IF f== 0: PRINT#3; TO 18; 'PUNKT '; franr;
20740     IF f== 1: PRINT#3; TO 18; 'PUNKT '; tilnr;
20750     PRINT#3;' MÅSKE = PUNKT '; fraprøv: RETurn
20760   END IF
20770   PRINT#3;'PUNKT ';fraprøv ;
20780   PRINT#3; TO 18; 'LIGGER PÅ / NÆR VED LINIE '; franr;'-'; tilnr
20790 END DEFine fejludskrift1
```

FEJLKONTROL

PROCedurerne bygger på to-dimensionale analyser, som letter forklaring og figurer, men er i øvrigt indført, fordi jeg tror, at simple algoritmer og mange IF'er giver et hurtigere regneprogram end tre-dimensionale analyser. Påstanden bør undersøges, og det endelige programafsnit kompileres til et EX-program.

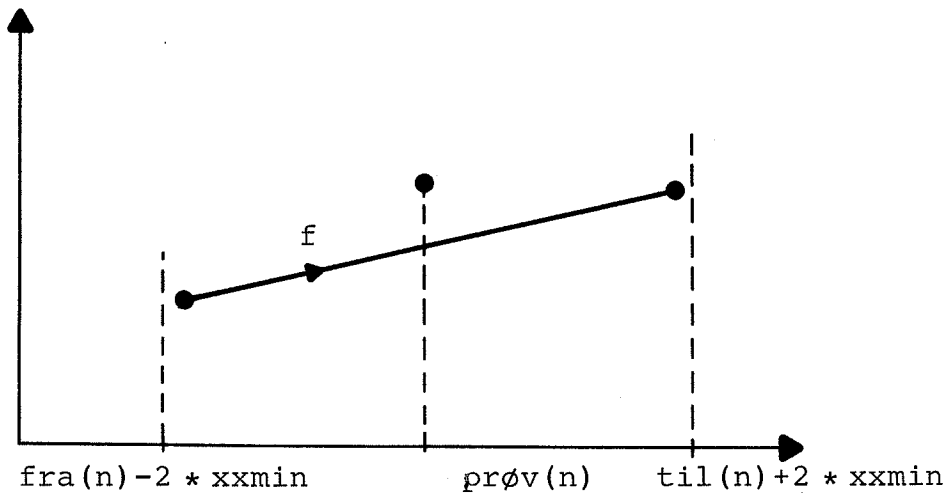
Linien franr-tilnr sammenlignes med punkt fraprøv og med linien fraprøv-tilprøv, hvis disse eksisterer.

linie 20080	Der findes intet tilnr, ejheller på de efterfølgende pos+7, idet arrayet trunkeres i linie 5210, når et tilnr fjernes.
linie 20100	tnr, se linie 20180(---tnr TO 99) ff.
linie 20160	punktkontrol
linie 20720	punktsammenfald
linie 20780	punkt på linie
linie 20170	liniekontrol, se i øvrigt linie 21000.

NB linie 20130 og 20170: Formen kan ændres til
IF (fejl=0)*(fraprøv<99)*(fraprøv>franr), som
ikke er hurtigere, men egnet til kompilering.

PUNKTKONTROL f findes fx af x = prøv(1), CHECK y og z koordinater

linie 20400 vælger mellem x,y og z som udgangspunkt, idet der vælges den akse, linien er "mest parallel" med.



2 * xxmin: altid synlig, oftest klart synligt gap (isometri_tegning)
1 * xxmin: netop synlig i 95% af tilfældene (vvs_tegning)

xxmin=skala/110, se linie 2510. Divisor valgt ved mange forsøg.

```
21000 DEFine PROCedure liniekontrol
21010  FOR psn= 2 TO 51 STEP 7
21020   IF linje%(fraprøv,psn)= 0: RETurn
21030   tilprøv= linje%(fraprøv,psn)
21040   IF tilprøv= franr OR tilprøv= tilnr THEN
21050     IF psn= 51: EXIT psn
21060     NEXT psn
21070   END IF
21080   FOR n= 1 TO 3: pft(n)= linje%(fraprøv,psn+n)

21090   fejl= 0: determinant: IF determ= 1: skæringspunkt

21100   IF fejl= 2 THEN
21110     PRINT#3;'LINIE '; franr;'-';tilnr;
21120     PRINT#3; TO 18; 'SKÆRER MASKE LINIE '; fraprøv;'-'; tilprøv
21130   END IF
21150  END FOR psn
21160 END DEFine liniekontrol

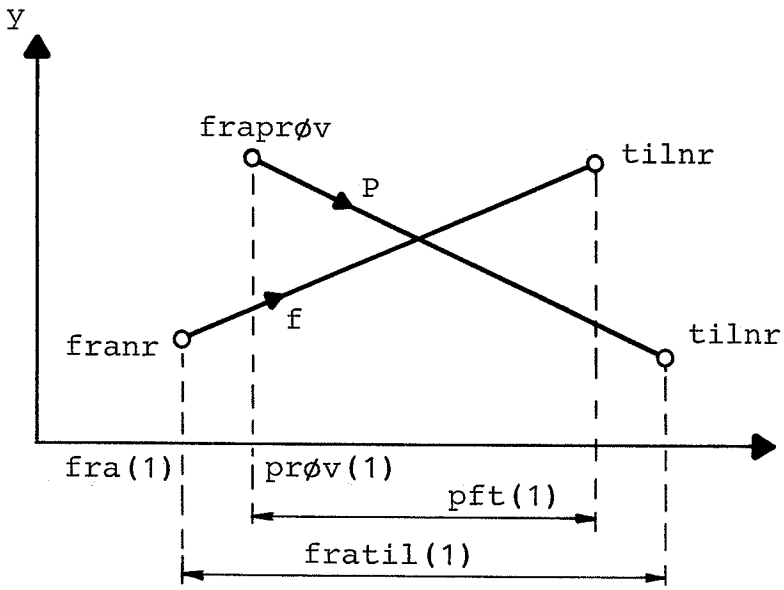
21200 DEFine PROCedure determinant
21210   det(3)= fratil(1) * pft(2) - fratil(2) * pft(1)
21220   det(2)= fratil(3) * pft(1) - fratil(1) * pft(3)
21230   det(1)= fratil(2) * pft(3) - fratil(3) * pft(2)
21240   determ= 1 - ( det(1)==0 ) * ( det(2)==0 ) * ( det(3)==0 )
21250   REMark determ= 0 HVIS ALLE det(n)= 0
21260 END DEFine determinant

21300 DEFine PROCedure skæringspunkt
21310  LOCal diff
21320  IF ( det(3)==0 )= 0 THEN
21330   p=((prøv(1)-fra(1))*fratil(2)-(prøv(2)-fra(2))*fratil(1))/det(3)
21340   f=((prøv(1)-fra(1))*   pft(2)-(prøv(2)-fra(2))*   pft(1))/det(3)
21350   IF p * (p-1) > 0 OR f * (f-1) > 0: RETurn
21360     REMark p og f = 0/1 ER FUNDET. SKÆRING UDEN FOR >>> RETURN
21370   diff= fra(3) + fratil(3) * f - prøv(3) - pft(3) * p
21380   IF ABS ( diff ) < 2 * xxmin: fejl=2
21390   RETurn
21400  END IF

21410  IF ( det(2)==0 )= 0 THEN
21420   p=((prøv(3)-fra(3))*fratil(1)-(prøv(1)-fra(1))*fratil(3))/det(2)
21430   f=((prøv(3)-fra(3))*   pft(1)-(prøv(1)-fra(1))*   pft(3))/det(2)
21440   IF p * (p-1) > 0 OR f * (f-1) > 0: RETurn
21450   diff= fra(2) + fratil(2) * f - prøv(2) - pft(2) * p
21460   IF ABS ( diff ) < 2 * xxmin: fejl=2
21470   RETurn
21480  END IF
21490  IF ( det(1)==0 )= 0 THEN

21500   p=((prøv(2)-fra(2))*fratil(3)-(prøv(3)-fra(3))*fratil(2))/det(1)
21510   f=((prøv(2)-fra(2))*   pft(3)-(prøv(3)-fra(3))*   pft(2))/det(1)
21520   IF p * (p-1) > 0 OR f * (f-1) > 0: RETurn
21530   diff= fra(1) + fratil(1) * f - prøv(1) - pft(1) * p
21540   IF ABS ( diff ) < 2 * xxmin: fejl=2: RETurn
21550  END IF
21560 END DEFine skæringspunkt
```

LINIEKONTROL Determinanterne for løsning i xy, yz og zx-planerne er udgangspunktet. Her vises tilfældet, hvor det(3)<>0.



x= fra(1)+fratil(1)*f = prøv(1)+pft(1)*p

y= fra(2)+fratil(2)*f = prøv(2)+pft(2)*p

f og p findes ved udtryk af typen f * (determinant)=konst

Hvis determinanten, fx det(3)= fratil(1)* pft(2)-fratil(2)* pft(1) er 0, kan ligningen ikke løses i planen 1-2. Eksempler er:

Parallelle linier: $\frac{\text{fratil(2)}}{\text{fratil(1)}} = \frac{\text{pft(2)}}{\text{pft(1)}}$

specielt: fratil(1)=pft(1)=0 (parallel med akse 2)
fratil(2)=pft(2)=0 (parallel med akse 3)

En eller begge linier vinkelret på plan 1-2, pft(1)=pft(2)=0

Hvis alle tre determinanter er 0, er linierne parallelle. Specielt er de sammenfaldende, men punktkontrol har taget sig af, om de aktuelle dele af linierne har fælles punkter, dele (fejl=1).

Hvis mindst een determinant er <> 0,
skærer linierne hinanden på en løsbar måde i den (de) planer, hvori determinanten er > 0.

CHECK KOORDINATER I DEN PLAN, HVOR DETERMINANTEN ER > 0
CHECK OM SKÆRINGEN FINDES PÅ DE AKTUELLE DELE AF LINIERNE

Eksempel: Linierne er parallelle med y- og z-akserne. det(3)=det(2)=0. det(1)<>0. determ=1. Linierne 21320-21480 overspringes. Kontrol i linierne 21490-21560.

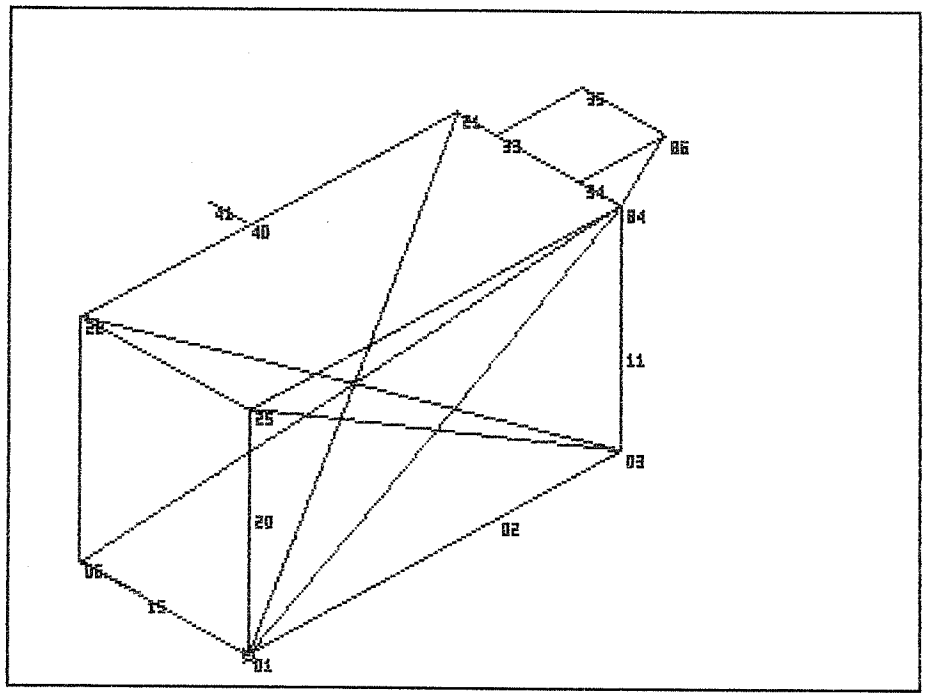
fejlforsøg 02.APRIL 86

```
start%( 12 ) =
(1)      STARTPUNKTnr = 1
(2,3,4)  x, y, z  min =        0        0        0
(5,6,7)  x, y, z  max =     1100      600      500
(8,9)    x0, y0       =     -414        0
(10)     SKALA = 1640
(11)     ANTAL LINIER = 25
(12)     MIN.AFSTAND xx = 15
```

pkt% (99, 9) =

nr	x	y	z	xs	ys	xp	yp	forbind
1	0	0	0	414	0	433	0	1008
2	600	0	0	911	337	953	300	1
3	900	0	0	1159	505	1212	450	4
4	900	600	0	1159	1105	1212	1050	6
6	0	0	400	83	224	87	200	4
11	900	250	0	1159	755	1212	700	1
15	0	0	250	207	140	217	125	2
20	0	350	0	414	350	433	350	1
21	900	600	400	828	1329	866	1250	3
22	0	600	400	83	824	87	800	4
25	0	600	0	414	600	433	600	4
33	900	600	300	911	1273	953	1200	2
34	900	600	100	1076	1161	1126	1100	2
35	1100	600	300	1076	1385	1126	1300	2
36	1100	600	100	1242	1273	1299	1200	2
40	400	600	410	406	1054	424	1005	1
41	400	600	495	335	1102	351	1048	1
84	900	600	0	1159	1105	1212	1050	1
86	1100	600	100	1242	1273	1299	1200	1

fejlforsøg 10.JAN.1986



fejlforsøg 02.APRIL 86

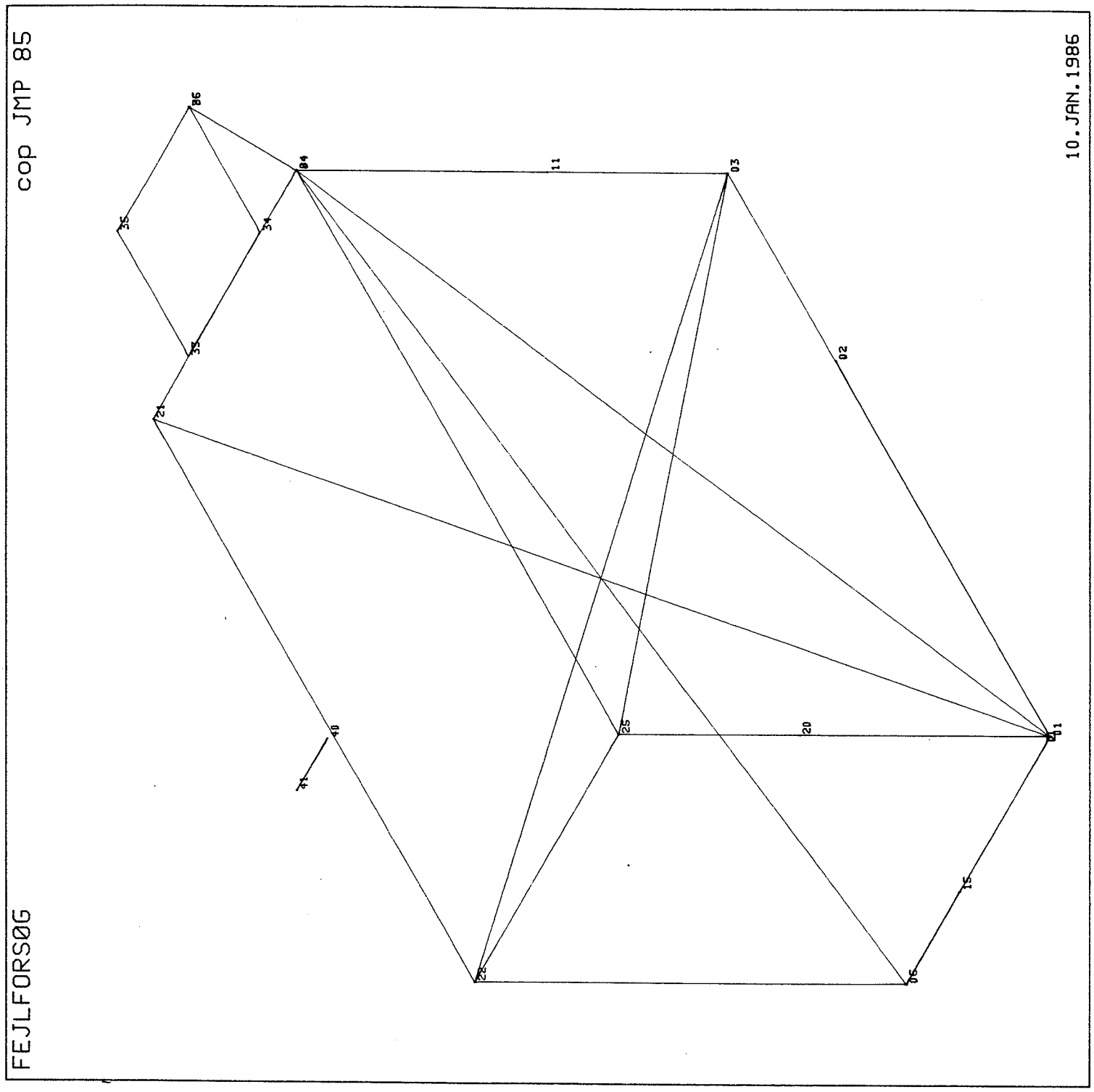
linje% (98, 57) =

franr,	tilnr	xx	yy	zz	FARVE,	MATERIALE,	DIMENSION
1 til	2	600	0	0	2	0	0
	3	900	0	0	2	0	0
	20	0	350	0	2	0	0
	25	0	600	0	2	0	0
	6	0	0	400	2	0	0
	15	0	0	250	2	0	0
	4	900	600	0	2	0	0
	21	900	600	400	2	0	0
3 til	4	0	600	0	2	0	0
	25	-900	600	0	2	0	0
	22	-900	600	400	2	0	0
4 til	11	0	-350	0	2	0	0
	25	-900	0	0	2	0	0
	6	-900	-600	400	2	0	0
	21	0	0	400	2	0	0
6 til	15	0	0	-150	2	0	0
	22	0	600	0	2	0	0
21 til	22	-900	0	0	2	0	0
22 til	25	0	0	-400	2	0	0
33 til	34	0	0	-200	2	0	0
	35	200	0	0	2	0	0
34 til	36	200	0	0	2	0	0
35 til	36	0	0	-200	2	0	0
40 til	41	0	0	85	2	0	0
84 til	86	200	0	100	2	0	0

FØLGENDE PUNKTER / LINIER BØR CHECKES FOR EVT. FEJL:

```
PUNKT 2              LIGGER PÅ / NÆR VED LINIE 1-3
PUNKT 20             LIGGER PÅ / NÆR VED LINIE 1-25
PUNKT 15             LIGGER PÅ / NÆR VED LINIE 1-6
LINIE 1-4            SKÆRER MÅSKE LINIE 3-25
PÅ LINIE 1-4 ER      PUNKT 4 MÅSKE = PUNKT 84
LINIE 1-21           SKÆRER MÅSKE LINIE 3-22
LINIE 1-21           SKÆRER MÅSKE LINIE 4-6
PUNKT 11             LIGGER PÅ / NÆR VED LINIE 3-4
PÅ LINIE 3-4 ER      PUNKT 4 MÅSKE = PUNKT 84
LINIE 3-22           SKÆRER MÅSKE LINIE 4-6
PÅ LINIE 4-11 ER     PUNKT 4 MÅSKE = PUNKT 84
PÅ LINIE 4-25 ER     PUNKT 4 MÅSKE = PUNKT 84
PÅ LINIE 4-6 ER      PUNKT 4 MÅSKE = PUNKT 84
PUNKT 33             LIGGER PÅ / NÆR VED LINIE 4-21
PUNKT 34             LIGGER PÅ / NÆR VED LINIE 4-21
PÅ LINIE 4-21 ER     PUNKT 4 MÅSKE = PUNKT 84
PUNKT 40             LIGGER PÅ / NÆR VED LINIE 21-22
LINIE 34-36          SKÆRER MÅSKE LINIE 84-86
PÅ LINIE 34-36 ER    PUNKT 36 MÅSKE = PUNKT 86
LINIE 35-36          SKÆRER MÅSKE LINIE 84-86
PÅ LINIE 35-36 ER    PUNKT 36 MÅSKE = PUNKT 86
PÅ LINIE 84-86 ER    PUNKT 84 MÅSKE = PUNKT 4
PÅ LINIE 84-86 ER    PUNKT 86 MÅSKE = PUNKT 36
```

På plotteren er punkt 40's beliggenhed uden for linien synlig
(70 % nedfotografering fra A3).



C80ROUTINER

```
30000 REMark ***  c80routiner ***
30100 DEFine PROCedure max_vindue
30110   OPEN#4; con_
30120   WINDOW#1;512,256,   0,   0:                  PAPER#1;0: CLS#1
30130   WINDOW#0;436, 33,   0,223: BORDER#0;1,0: PAPER#0;2: INK#0;7
30140   WINDOW#1;512,223,   0,   0: BORDER#1;1,2: PAPER#1;6: INK#1;0
30150   WINDOW#2;512,223,   0,   0: BORDER#2;1,2: PAPER#2;5: INK#2;0
30160   WINDOW#4; 76, 33,436,223: BORDER#4;1,0: PAPER#4;2: INK#4;7
30170 END DEFine max_vindue

30200 DEFine PROCedure dobb_vindue
30210   OPEN#4; con_
30220   WINDOW#1;512,256,   0,   0:                  PAPER#1;0: CLS#1
30230   WINDOW#0;436, 33,   0,223: BORDER#0;1,0: PAPER#0;2: INK#0;7
30240   WINDOW#1;280,223,   0,   0: BORDER#1;1,0: PAPER#1;2: INK#1;7
30250   WINDOW#2;232,223,280,   0: BORDER#2;1,0: PAPER#2;2: INK#2;7
30260   WINDOW#4; 76, 33,436,223: BORDER#4;1,0: PAPER#4;2: INK#4;7
30270 END DEFine dobb_vindue

30400 DEFine PROCedure c80list (linie1,linie2)
30410   TRA trap: OPEN#3,ser1: BAUD baudtal
30450   BPUT#3,27,108,5,27,81,77,27,67,72,27,78,10
30460   LIST#3, linie1 TO linie2:     REMark xx,yy eller xx TO yy
30470   BPUT#3,12: CLOSE#3
30480 END DEFine c80list

30500 DEFine PROCedure  c80dir (dirmdv$)
30510   TRA trap: OPEN#3,ser1: BAUD baudtal
30520   BPUT#3,27,10: PRINT#3; \\\
30530   DIR#3, dirmdv$
30540   BPUT#3,12: CLOSE#3
30550 END DEFine c80dir
```

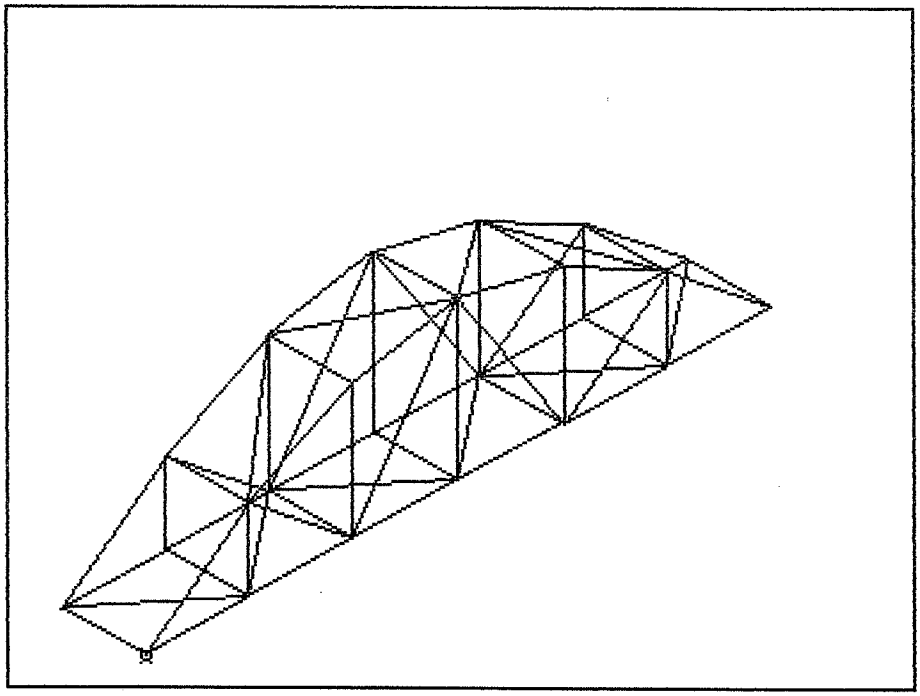
De to vinduer har tre linier à 72 karakterer i kanal 0.
72 karakterer, som printeren i "c80list".

"c80list" TRA og BAUD, se linie 440 - 450.
BPUT#3, se TOOLKIT2-vejledningen.
Smlgn. linie 6620 - 6630, ala printer-vejledning.
Udskriver programafsnit, linie 30460.
Benyttes af "c80program", linie 6500 ff.
BPUT#3;12 = FORM FEED, ny side.

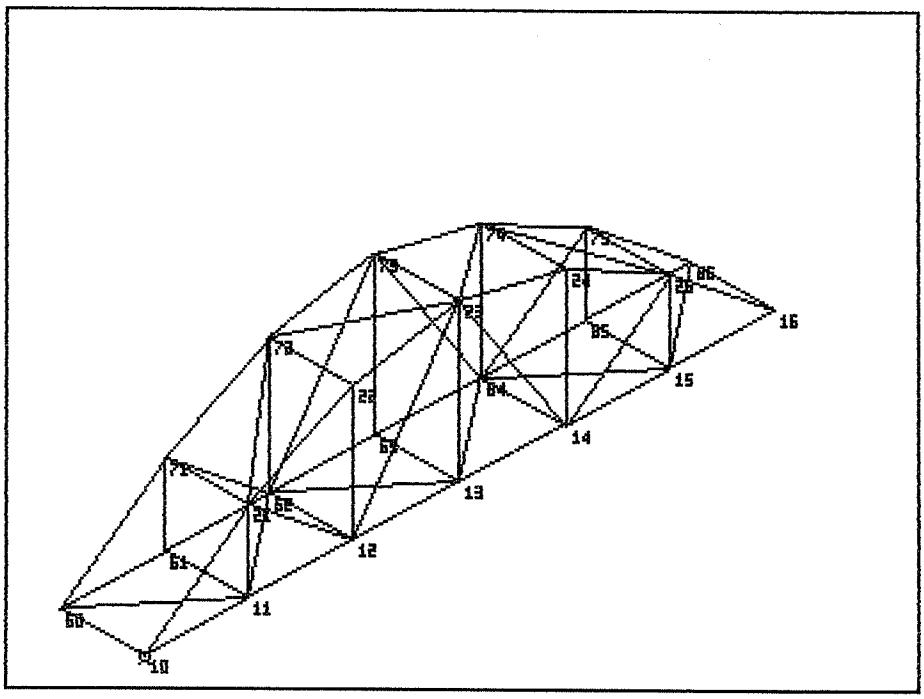
"c80dir" c80dir "ram2_" giver RAM2_s directory på printeren.

EKSEMPEL BRO

BRO Ø2.DEC.1985



BRO Ø2.DEC.1985



BRO 02. APRIL 86

```
start%( 12 ) =
(1)      STARTPUNKTnr = 10
(2,3,4)  x, y, z  min =        0        0        0
(5,6,7)  x, y, z  max =     6000     2000     1000
(8,9)    x0, y0       =     -828        0
(10)     SKALA = 6490
(11)     ANTAL LINIER = 64
(12)     MIN. AFSTAND xx = 59
```

pkt% (99, 9) =

nr	x	y	z	xs	ys	xp	yp	forbind
10	0	0	0	828	0	866	0	1003
11	1000	0	0	1656	561	1732	500	6
12	2000	0	0	2484	1122	2598	1000	6
13	3000	0	0	3312	1683	3464	1500	6
14	4000	0	0	4140	2244	4330	2000	6
15	5000	0	0	4968	2805	5196	2500	6
16	6000	0	0	5796	3366	6062	3000	3
21	1000	900	0	1656	1461	1732	1400	6
22	2000	1500	0	2484	2622	2598	2500	4
23	3000	1750	0	3312	3433	3464	3250	8
24	4000	1500	0	4140	3744	4330	3500	4
25	5000	900	0	4968	3705	5196	3400	6
60	0	0	800	166	449	173	400	4
61	1000	0	800	994	1010	1039	900	4
62	2000	0	800	1822	1571	1905	1400	8
63	3000	0	800	2650	2132	2771	1900	4
64	4000	0	800	3478	2693	3637	2400	8
65	5000	0	800	4306	3254	4503	2900	4
66	6000	0	800	5134	3815	5369	3400	4
71	1000	900	800	994	1910	1039	1800	5
72	2000	1500	800	1822	3071	1905	2900	6
73	3000	1750	800	2650	3882	2771	3650	6
74	4000	1500	800	3478	4193	3637	3900	6
75	5000	900	800	4306	4154	4503	3800	5

BRO 02.APRIL 86

linje% (98, 57) =

franr,	tilnr	xx	yy	zz	FARVE,	MATERIALE,	DIMENSION
10 til	11	1000	0	0	7	0	0
	60	0	0	800	2	0	0
	21	1000	900	0	7	0	0
11 til	12	1000	0	0	7	0	0
	21	0	900	0	7	0	0
	61	0	0	800	2	0	0
	60	-1000	0	800	2	0	0
	62	1000	0	800	2	0	0
12 til	13	1000	0	0	7	0	0
	22	0	1500	0	7	0	0
	62	0	0	800	2	0	0
	21	-1000	1000	0	7	0	0
	23	1000	1750	0	7	0	0
13 til	14	1000	0	0	7	0	0
	23	0	1750	0	7	0	0
	63	0	0	800	2	0	0
	62	-1000	0	800	2	0	0
	64	1000	0	800	2	0	0
14 til	15	1000	0	0	7	0	0
	24	0	1500	0	7	0	0
	64	0	0	800	2	0	0
	23	-1000	1750	0	7	0	0
	25	1000	1000	0	7	0	0
15 til	16	1000	0	0	7	0	0
	25	0	900	0	7	0	0
	65	0	0	800	2	0	0
	64	-1000	0	800	2	0	0
	66	1000	0	800	2	0	0
16 til	66	0	0	800	2	0	0
	25	-1000	900	0	7	0	0
21 til	22	1000	600	0	7	0	0
	71	0	0	800	7	0	0
	72	1000	600	800	7	0	0
22 til	23	1000	250	0	7	0	0
	72	0	0	800	7	0	0
23 til	24	1000	-250	0	7	0	0
	73	0	0	800	7	0	0
	74	1000	-250	800	7	0	0
	72	-1000	-250	800	7	0	0
24 til	25	1000	-600	0	7	0	0
	74	0	0	800	7	0	0
25 til	75	0	0	800	7	0	0
	74	-1000	600	800	7	0	0
60 til	61	1000	0	0	4	0	0
	71	1000	900	0	7	0	0
61 til	62	1000	0	0	4	0	0
	71	0	900	0	4	0	0
62 til	63	1000	0	0	4	0	0
	72	0	1500	0	4	0	0
	71	-1000	1000	0	4	0	0
	73	1000	1750	0	4	0	0
63 til	64	1000	0	0	4	0	0
	73	0	1750	0	4	0	0
64 til	65	1000	0	0	4	0	0
	74	0	1500	0	4	0	0
	73	-1000	1750	0	4	0	0
	75	1000	1000	0	4	0	0

65 til	66	1000	0	0	4	0	0
	75	0	900	0	4	0	0
66 til	75	-1000	900	0	7	0	0
71 til	72	1000	600	0	7	0	0
72 til	73	1000	250	0	7	0	0
73 til	74	1000	-250	0	7	0	0
74 til	75	1000	-600	0	7	0	0

FØLGENDE PUNKTER / LINIER BØR CHECKES FOR EVT. FEJL:

FEJLFRI. Ingen udskrift under "fejlkontrol".

TILLÆG

ROLAND dxy880 plotter

tput, en forbedret talput

SKÆRMEN & dens brug

SAVE, LOAD, LIST, DIR, WSTAT

ALTERNATIVT & PRINCIPIELT LOADEPROGRAM

PRINTEREN

ERROR

STRØTANKE:

OSCILLOGRAFEN HAR BRUGT SKÆRMEN I 50 ÅR!
MEN IDAG ANSER SERIØSE EDB-INGENIØRER TIT
BEVÆGELIG GRAFIK FOR LIGEGYLDIGT LEGETØJ.
JEG BEKLAGER, AT DETTE NOTAT OGSÅ OMTALER
PROBLEMERNE MED AT BEVÆGE KURVER, FIGURER
HEN OVER ANDRE FIGURER, KOORDINATSYSTEMER
UDEN AT EFTERLADE SIG SPOR, OG I FARVER !

ROLAND DXY880 PLOTTER

```
12000 DEFine PROCedure plotter: REMark ROLAND DXY, isometri
12010   LOCal u$, pfarve(5), rgh, lpmax, hpmax, pskala
12020   qltegn: BORDER#4;0: CLS#4: BORDER#4;1: INK#4;7
12030   AT#4;1,1: PRINT#4;'SLUK PLOTTEREN OG' \' NULSTIL PEN-ARMEN.'
12040   AT#4;4,1: PRINT#4;'ILÆG PAPIR.  TÆND' \\' KLAR  J / N  ?'
12050   REPeat klarpl: AT#4;6,17: INPUT#4; y$: IF y$=='j': EXIT klarpl
12060   INK#4;4: CLS#4: AT#4;0,1: PRINT#4;'OVERSKRIFT:'
12070   INK#4;7: INPUT#4;' '; over$
12080   INK#4;4: AT#4;5,1: PRINT#4;'UDSKRIFT AF NUMMER'
12090   PRINT#4;' ELLER AF DIMENSION'
12100   INK#4;7: PRINT#4;' TAST  n FOR NR'\'        d FOR DIMENS'
12110   PRINT#4;'        b BEGGE DELE'\'        i INGEN TAL'
12120   REPeat loop1
12130    AT#4;10,1: PRINT#4;'  ': AT#4;10,1: INPUT#4; u$: u$= u$(1)
12140    IF u$ INSTR 'ndbi': EXIT loop1
12150   END REPeat loop1
12160   INK#4;4: AT#4;11,1: PRINT#4;'PEN 7 TIL RAMMEN.'
12170   PRINT#4;' ANGIV PEN-NR FOR'\' TAL OG LINIER :'
12180   INK#4;7: AT#4;14,1: PRINT#4;'NR-FARVE =  '\' DIM-FARVE ='
12190   PRINT#4;' SKÆRM rød  ='\' SKÆRM grøn ='\' SKÆRM hvid ='
12200   FOR rgh=1 TO 5
12210    IF (u$=='d' OR u$=='i') AND rgh= 1: NEXT rgh
12220    IF (u$=='n' OR u$=='i') AND rgh= 2: NEXT rgh
12230    talput 4,13+rgh,18,0,7,1: pfarve(rgh)= INT(ktal(1) + 1 )
12240   END FOR rgh
12250   lpmax= (xmax -xmin +zmax -zmin)*.866
12260   hpmax= (xmax -xmin +zmax -zmin)/2 + ymax - ymin
12270   pskala= hpmax/2500: IF lpmax/3600 >pskala: pskala= lpmax/3600
12280   plotter_open: PRINT#5;'S 2': plot_exist_data
12290   IF u$=='n' OR u$=='b': PRINT#5;'^SI 0.12,0.16;': plot_numre
12300   IF u$=='d' OR u$=='b': PRINT#5;'^SI 0.12,0.16;': plot_dimension
12310   PRINT#5;'^ SI;'
12320   PRINT#5;'J 0': CLOSE#5: CLS#4: PRINT#4;\\' ER PENNENE FJERNET ?'
12330   REPeat loop2: AT#4;3,20: INPUT#4; y$: IF y$=='j': EXIT loop2
12340 END DEFine plotter

12400 DEFine PROCedure plotter_open
12410   LOCal xr, yr: OPEN#5,ser2: BAUD 1200
12420   REMark SWITCH 1,1-4 sparer PRINT#5;'CA 7'
12430   PRINT#5;'J 8': PRINT#5;'M 0,0'
12440   xr= INT(200.5+lpmax/pskala): yr= INT(200.5+hpmax/pskala)
12450  FOR n=1 TO 2
12460   PRINT#5;'D';xr;',';0;',';xr;',';yr;',';0;',';yr;',';0;',';0
12470  END FOR n
12480   PRINT#5;'M';30;',';INT(hpmax/pskala +140)
12490   TRA 1: PRINT#5;'S 4';'P'; over$: TRA trap: REMark A.H.T. Æ,Ø,Å
12500   PRINT#5;'M';INT(lpmax/pskala-135);',';INT(hpmax/pskala+140)
12510   PRINT#5;'P';'cop JMP 86'
12520   PRINT#5;'M';INT(lpmax/pskala-100);',';30
12530   PRINT#5;'S 3';'P';dato$(1)(1 TO 11)
12540 END DEFine plotter_open
```

```
12600 DEFine PROCedure plot_exist_data
12610   LOCal snr, tilnr, sfarve: snr= start%(1)
12620   PRINT#5;'M';INT(pkt%(snr,7)/pskala+100.5);
12630   PRINT#5;',';INT(pkt%(snr,8)/pskala+100.5)
12640   PRINT#5;'N 1'
12650   FOR franr=1 TO 98
12660    IF linje%(franr,1) >0 THEN
12670     FOR pos= 2 TO 51 STEP 7
12680      IF linje%(franr,pos) >0 THEN
12690       tilnr= linje%(franr,pos): sfarve= linje%(franr,pos+4)
12710       IF sfarve= 2: PRINT#5;'J';pfarve(3)
12720       IF sfarve= 4: PRINT#5;'J';pfarve(4)
12730       IF sfarve >6: PRINT#5;'J';pfarve(5)
12760       PRINT#5;'M';INT(pkt%(franr,7)/pskala+100.5);
12770       PRINT#5;',';INT(pkt%(franr,8)/pskala+100.5)
12780       PRINT#5;'D';INT(pkt%(tilnr,7)/pskala+100.5);
12790       PRINT#5;',';INT(pkt%(tilnr,8)/pskala+100.5)
12800      END IF
12810     END FOR pos
12820    END IF
12830   END FOR franr
12840 END DEFine plot_exist_data

13000 DEFine PROCedure plot_dimension
13010  LOCal tilnr, dimens, korrx, korry, xxpl, yypl, xb%, yb%
13020  PRINT#5;'J'; pfarve(2): REMark 1.6 mm SKRIFT LINIE 13170
13030  FOR franr=1 TO 98
13040   IF linje%(franr,1) >0 THEN
13050    FOR pos= 2 TO 51 STEP 7
13060     IF linje%(franr,pos) >0 THEN
13070      tilnr = linje%(franr,pos)
13080      dimens= linje%(franr,pos+6)
13090      korrx = 10: korry = -10
13100       xxpl = ABS (pkt%(tilnr,7) - pkt%(franr,7))
13110       yypl = ABS (pkt%(tilnr,8) - pkt%(franr,8))
13120      IF 1.8*ABS(yypl)<=ABS(xxpl): korrx=-9* LEN(dimens): korry=-30
13130      xb%= (pkt%(franr,7)+pkt%(tilnr,7)) / 2 / pskala +100 +korrx
13140      yb%= (pkt%(franr,8)+pkt%(tilnr,8)) / 2 / pskala +100 +korry
13150      PRINT#5; 'M'; xb%; ','; yb%
13170      PRINT#5; '^LB'; dimens; CHR$(3)
13180     END IF
13190    END FOR pos
13200   END IF
13210  END FOR franr
13220 END DEFine plot_dimension

13400 DEFine PROCedure plot_numre
13410   LOCal pnr, nr$, xb%, yb%
13420   PRINT#5;'J'; pfarve(1)
13430   FOR punk= 1 TO 99
13440    IF pkt%(punk,1) > 0 THEN
13450     pnr= pkt%(punk,1): nr$= pnr: IF pnr< 10: nr$= '0'&nr$
13460     xb%= pkt%(punk,7)/pskala +109
13470     yb%= pkt%(punk,8)/pskala + 72
13480     PRINT#5; 'M'; xb%; ','; yb%
13490     PRINT#5; '^LB'; nr$; CHR$(3)
13500    END IF
13510   END FOR punk
13520 END DEFine plot_numre
```

```
100 PAPER 2:INK 7:PAPER#2;7:INK#2;2:WMON 4:text=1
110 PRINT#2;'DEMO AF':REMark KONTROL AF INPUT AF TAL OG STRENGE
120 STRIP#2;2:INK#2;7:alt$='abcde':funk$='a1a2b1b2hjælp'
130 PRINT#2;'tput(ind$,typ$,kn,pa,bl,li,ta,læ,dc,mi,ma)'
140 BLOCK#2;252,40,0,30,4::AT#2;3,1:STRIP#2;4:INK#2;0
150 ind$='-12.88':kn=1:IF text=0:ind$='TAST ↑'
160 PRINT#2;'(';ind$;', ,1, , , , , , , , , )'
170 AT#2;7,0:STRIP#2;2:INK#2;7
180 PRINT#2;'ind$     UDSKRIFT AF EXIST.DATA // FORSLAG'
190 PRINT#2;'         "ENTER" >>>> ACCEPT AF UDSKRIFTEN'
200 PRINT#2;'typ$  $  LANG STRENG, ALLE KARAKTERER ETC.'
210 PRINT#2;'      #  KORT STRENG, ind$ INSTR "funk$"'
220 PRINT#2;'      %  DECIMALTAL SAMT ind$ INSTR "alt$"'
230 PRINT#2;'kn,pa,bl KANAL-NR., PAPER, INK '
240 PRINT#2;'li,ta    LINIE, TAB FOR  INPUT '
250 PRINT#2;'læ,dc    STRENG-LÆNGDE,  DECIMALER I FDEC$'
260 PRINT#2;'mi,ma    MIN. OG MAX. FOR DECIMALTAL-INPUT'
270 PRINT#2;'alt$     1 BOGST.-PROCEDUREKALD, TAL-INPUT'
280 PRINT#2;'funk$    PROCEDUREKALD, OP TIL "læ" BOGST.'
290 PAPER#2;7:INK#2;2
300 PRINT#2;'FOR LANGE TAL UDSKRIVES FLYTTET,TILFØJET *'
310 PRINT#2;'I DEMO: alt$="abcde",funk$="a1a2b1b2hjælp"'
320 PAPER#2;4:INK#2;0
330 AT#2;6,1:PRINT#2;'KANAL ';
340 IF text=1:PRINT#2;'OG FORSØGS-STRENG ';
350 PRINT#2;'FASTLAGT I DEMO'\\
360 AT 11,2:PRINT'I DEMO SKAL PAPER OG INK'\'  VÆLGES FOR HVERT FORSØG.'
370 AT 14,2:PRINT'tput SKAL KUN HAVE PAPER'\'  OG INK ANGIVET AF HENSYN'
380 AT 16,2:PRINT'TIL UDSKRIFT-FELTETS FARVE'
390 IF text=0:AT#2;2,1:INPUT#2;'ind$= ';ind$
400 AT#2;4, 1:PRINT#2;'VÆLG: '
410 AT#2;4, 7:INPUT#2;'typ$ ';typ$:AT#2;3, 9:PRINT#2;typ$
420 AT#2;4,16:INPUT#2;'pa ';pa:AT#2;3,13:PRINT#2;pa
430 AT#2;4,22:INPUT#2;'bl ';bl:AT#2;3,15:PRINT#2;bl
440 PAPER pa:INK bl:CLS
450 AT#2;4,28:INPUT#2;'li ';li:AT#2;3,17:PRINT#2;li
460 AT#2;4,35:INPUT#2;'ta ';ta:AT#2;3,20:PRINT#2;ta
470 AT#2;5, 7:INPUT#2;'læ ';læ:AT#2;3,23:PRINT#2;læ
480 AT#2;5,14:INPUT#2;'dc ';dc:AT#2;3,26:PRINT#2;dc
490 AT#2;5,20:INPUT#2;'mi ';mi:AT#2;3,28:PRINT#2;mi
500 AT#2;5,32:INPUT#2;'ma ';ma:AT#2;3,35:PRINT#2;ma
510 linie=2+14*(li<10):AT linie,2
520 PRINT'TAST < ENTER >, NÅR'\'  RETTELSERNE ER FORETAGET'
530 tput ind$,typ$,kn,pa,bl,li,ta,læ,dc,mi,ma
540 REPeat igen
550   AT linie+2,2:INPUT'FLERE FORSØG J / N ';y$
560   IF y$=='n' OR y$=='j':EXIT igen
570 END REPeat igen
580 IF y$=='j':text=0:PAPER#2;7:INK#2;2:CLS#2:CLS:RUN 110
590 STOP: REMark **************************************************
```

"tput" er en generel indtast-PROCEDURE for tal og strenge, med kontrol af længde, talstørrelse osv. Ved tal-INPUT kan funktioner kaldes ved indtastning af en karakter, der ikke er et tal.

Der kan indføres udskrift af tidligere data for den pågældende (nye) variabel.

Tal udskrives med FDEC$ i ordentlige kolonner, korrekt antal decimaler (også nuller bagved).

"tput" kræver at TOOLKIT loades (exten_bin eller normalt exten_pbin) eller at TOOLKIT2 RAM/ROM anvendes.

JM-P's anstændige danske tegn i stedet for QL's SWAHILI-tegn kræver, at "qltegn" loades.

DEMO_tput illustrerer PROCEDUREn (linie 100-590)

tput" selv findes i linie 18000-18590. Fjernes DEMO-delen, kan resten normalt MERGES ind i programmer med det selvstændige i linier op til 15000 og 20000 ff til andre, kendte hjælpeprocedurer.

NB linie 18050 må omskrives, hvis man vil arbejde med tal > 32767 (+/-)

ind$ er en streng. Den kan være neutral (" eller ' '), eller et forslag til sandsynligt svar. Oftest anvendes ind$ til udskrift af tidligere data, som skal fx. kontrolleres, revideres.

typ$ angiver, hvad input (svaret på ind$) skal anvendes til.

typ$= '$' angiver, at der blot skal svares med en streng, der kan indeholde tegn mellem CHR$(32)=mellemrum og CHR$(170)=Æ, herunder altså også fx. franske og tyske tegn.

typ$= '#' angiver, at der skal indtastes en streng af given form, fx. et procedurekald, et navn på et datasæt (byggeelement etc.). Svaret skal være indeholdt i den på forhånd givne streng "funk$", der ligeledes kan indeholde alle tegn op til Æ. I DEMO er "a","a1","2b","2hjælp" exempelvis korrekte svar (forudsat at læ tillader de længere strenge).

typ$= '%' angiver, at der skal indtastes en decimalvariabel (uanset at % formelt betyder heltal), se mi og ma nedenfor.

ka,pa,bl angiver kanalen (skærm-nr.) og dennes normale PAPER- og INK-farver.

li,ta angiver, hvor input-cursoren begynder (linie og tab).

læ angiver strengens maximale længde for typ$='$' eller '#'. For lange strenge trunkeres, så de sidste karakterer udelades.

læ,dc angiver strengens maximale længde og antal decimaler for decimalvariabler, udskrevet med FDEC$. Er talstrengen i ind$ for lang, reduceres tab, så tallet udskrives korrekt. tab (ta) er altid >=0(linie 18060), men programmet sikrer ikke, at der ikke udskrives "forbi" (ned på næste linie), hvis tab er nær kanalens højere grænse. (dc kan sættes lig 0 eller " for typ$='$' eller '#'). Hvis det reviderede tal kan udskrives korrekt, justeres tab tilbage til det ønskede.

mi,ma sættes til 0 eller " for typ$= '$' eller '#'. mi og ma er minimums- og maximumsværdier for den indtastede nye/reviderede decimalvariabel.

```
18000 DEFine PROCedure tput(ind$,typ$,ka,pa,bl,li,ta,læ,dc,mi,ma)
18010  LOCal find$,find%,curs,ok,pt,glen,glta,pil,x,flyt,a$,x$
18020  STRIP#ka;2:INK#ka;7:IF pa=2:STRIP#ka;7:INK#ka;2
18030  find$=ind$:ok=0:pt=0:glen=læ:glta=ta:x$=''

18040  IF typ$='%'THEN
18050   find%=find$:IF LEN(find%)+1+dc>læ:glen=LEN(find%)+1+dc:x$='*'
18060   find$=FDEC$(find$,glen,dc):ta=ta+læ-glen:IF ta<0:ta=0
18070  ELSE
18080   IF LEN(find$)>læ:find$=find$(1 TO læ)
18090   find$=find$&FILL$(' ',læ-LEN(find$))
18100  END IF

18110  FOR n=1 TO glen:IF find$(n)=' ':find$(n)='°'
18120  curs=ta:OVER#ka;0:AT#ka;li,curs:PRINT#ka;find$&x$

18130  REPeat tast2
18140   REPeat vent:IF INKEY$(#ka)='':EXIT vent
18150   OVER#ka;-1:AT#ka;li,curs:PRINT#ka;find$(curs-ta+1)
18160   pil=KEYROW(1):PAUSE 2
18170   IF pil=2 OR pil=16:AT#ka;li,curs:PRINT#ka;find$(curs-ta+1)
18180   IF pil=2:curs=curs-1*(curs>ta):NEXT tast2
18190   IF pil=16:curs=curs+1*(curs<ta+glen-1):NEXT tast2
18200   IF pil=1:AT#ka;li,curs:checkstreng
18210   IF ok:EXIT tast2
18220   BGET#ka;x:flyt=0
18230   IF x>31 AND x<171 THEN
18240    x=x+154*(x=32):a$=CHR$(x):find$(curs-ta+1)=a$
18250    OVER#ka;0:AT#ka;li,curs:PRINT#ka;' '
18260    AT#ka;li,curs:PRINT#ka;a$:flyt=1*(curs<ta+glen-1)
18270   END IF
18280   AT#ka;li,curs:PRINT#ka;find$(curs-ta+1):curs=curs+flyt
18290  END REPeat tast2

18300  PAPER#ka;pa:INK#ka;bl:OVER#ka;0
18310  AT#ka;li,ta:PRINT#ka;FILL$(' ',glen)
18320  IF x$='*'AND ok=2:AT#ka;li,ta+glen:PRINT#ka;' ':ta=glta:x$=''
18330  AT#ka;li,ta:PRINT#ka;ind$&x$
18340 END DEFine tput

18400 DEFine PROCedure checkstreng
18410   LOCal check$,start,slut,a,b,tal
18420   check$=find$:start=1:slut=glen
18430   FOR a=1 TO glen:IF check$(a)='°':check$(a)=' '
18440   FOR a=1 TO glen:IF check$(a)=' ':start=start+1:ELSE EXIT a
18450   FOR b=glen TO 1 STEP -1:IF check$(b)=' ':slut=slut-1:ELSE EXIT b
18460   check$=check$(start TO slut)

18470   IF typ$='$':ind$=check$:ok=1:RETurn
18480   IF typ$='#'AND check$ INSTR funk$:ind$=check$:ok=1:RETurn
18490   IF typ$='%'THEN
18500    IF check$(1) INSTR alt$:ind$=check$(1):typ$='#':ok=2:RETurn
18510    nlen=LEN(check$):IF check$(1)='-':ok=1
18520    FOR a=1+ok TO nlen
18530      IF check$(a) INSTR '0123456789':ok=ok+1
18540      IF check$(a) INSTR '.': pt=pt+1:ok=ok+1
18550    END FOR a
18560    IF ok<nlen OR pt>1:ok=0:RETurn
18570    tal=check$:IF NOT(tal>=mi)*(tal<=ma):ok=0:RETurn
18580    IF nlen>læ:ok=1:ind$=FDEC$(tal,glen,dc):RETurn
18590    ok=2:ind$=FDEC$(tal,læ,dc):RETurn
18600   END IF
18610 END DEFine checkstreng
```

Bemærkninger til selve "tput", refererende til linie-nr.

18020 ind$ udskrives i kontrast til skærmen, rød/hvid eller hvid/rød.
18030 De oprindelige værdier af ta og læ "gemmes".
x$ registrerer for lange strenge (18050). ok registrerer, om strengen er korrekt.
18050 find%=længden af heltallet. læ kontrolleres.
18080 For typ$='$' eller '#' reduceres for lange strenge, henholdsvis udbygges korte strenge med tomme pladser (bagfra).
18110 Tomme pladser (' ') gøres synlige med 'o'(CHR$(186))
18120 Strengen ind$ udskrives. For lange, flyttede decimalvariabler tilføjes '*'

18130-18290 er indtast-løkken

18150 OVER -1 giver sorte tal på CURSORs plads.
18160 KEYROW(1) benyttes til at flytte CURSOR(2 og 16).
KEYROW(1)=1 er ENTER, der angiver, at brugeren har afsluttet sin indtastning/revision.
18200 Strengen kontrolleres i PROCEDUREn "checkstreng".
En korrekt streng (ok=1 eller 2) fører til 18300.
18220 Den indtastede karakter "aflæses". Kun karakterer med kode 32 til 170 accepteres, og 'o'(CODE 186) indføres for mellemrum (CODE 32).
Den indtastede karakter udskrives.
18260 Bemærk, at CURSOR flyttes mod højre ved indtastning, som naturligt er, men at CURSOR i øvrigt skal flyttes med pilene, og ikke kan flyttes uden for grænserne (18180-90).
Højre flytning med mellemrumstangenten medfører et mellemrum (som EDB-brugere er vant til).
18300 En korrekt streng udskrives med OVER Ø.
Den gamle ind$ fjernes, den nye udskrives, evt. fjernes '*', hvis den nye streng (kan placeres korrekt og) er indenfor de for typ$='%' fastlagte grænser (ok=2 i stedet for 1).
Blot betingelserne mi og ma er opfyldt, er strengen jo korrekt, omend brugeren har udskrevet den forkert (og det kan ødelægge et ellers pænt skærm-lay-out).

18400-580 er kontrol-proceduren

18430 'o' ændres til ' ' i checkS (ikke i ind$, der eventuelt skal bruges, hvis kontrollen ikke passeres)
18440 Alle mellemrum ' ' foran (a) og bag (b) fjernes
18470 '$'-strenge anerkendes altid (brugere kan indlægge kontrol her om ønskeligt)
18480 '#'-strenge skal være indeholdt i "funk$"
18490 '%'-strenge kan have første karakter som et kald (hjælp, instruktion, funktion etc.)
18500 Også for funkkald ud fra for lange talstrenge må ok sættes til 2(18320)
18510 Talstrenge skal have alle karakterer = et tal (eller et minustegn som første karakter). Eet punktum er også gyldigt.
18530 RETURN. En karakter var ikke et tal, to punktummer etc.
18540 Tallet skal overholde minimum og maximum.
18550 Tal-strengen er stadig for lang. ok=1
18560 Tal-strengen er nu revideret til den opgivne længde

GENOPFRISK MEMORY, nymem

Et ordentligt program opbruger ikke sin frie hukommelse. Mange uafsluttede GOTO og GOSUB'er kan medføre det. Problemet opstår snarere ved ufærdige programmer på experimentstadiet, ved nye DIM osv. - og opstår næppe på en 512K-QL.

Udskrift af FREE_MEM engang imellem er udmærket.

Har man en (indtast-)løkke, der ofte passeres, kan funktionen "nymem" kaldes, hvis FREE_MEM er blevet for lille.

Den medfører CLEAR, men SAVEr via RAM først de variabler og data, man ikke kan tåle at miste.

Her vises den som om, der i linie 4032 stod
"IF FREE_MEM<8000:nymem".

I den viste procedure mangler adressen dansk til "qltegn" mv.

De fire variable, der skal gemmes til efter CLEAR, er: filens navn, minitegn (adresse), tidspunkt og dato$ (samt dansk).

ROM-RAM-udvidelsen FORMATteres og anvendes til at lagre variaberne. DATA SAVEs på FLOPPY med "saveproces". RAM kunne være brugt. CLEAR. DIM. Reload af variabler. Tidspunkt fra start genoprettes. Loadeproces.

```
11600 DEFine PROCedure nymem
11610   LOCal reload$(4,20): reload$(1)= filnavn$: reload$(2)= minitegn
11620   reload$(3)= INT((DATE-starttid)/360 +.5)/10: reload$(4)=dato$(1)
11630   AT#4;2,1: PRINT#4;'MEMORY FULL.'\' DATA SAVEs'
11640   AT#4;4,1: PRINT#4;'MIDLERTIDIGT'
11650   AT#4;7,1: INPUT#4;'KLAR ? '; y$: FORMAT#4; ram1_2
11660   OPEN_NEW#6,ram1_nyload: FOR ql=1 TO 4: PRINT#6; reload$(ql)
11670   CLOSE#6: saveproces: CLEAR: DIM reload$(4,20)
11710   OPEN_IN#6, ram1_nyload: FOR ql=1 TO 4: INPUT#6; reload$(ql)
11720   filnavn$= reload$(1): minitegn= reload$(2): dato$(1)= reload$(4)
11730   load$='ny': load_iso= 0: nygl= 2: starttid= DATE - reload$(3)
11740   skema= 1: RUN 400
11750 END DEFine nymem
```

SKÆRMEN

DENS BRUG GENERELT

DENS ANVENDELSE i isometri_tegning

KARAKTERUDSKRIFT

SAVE,LOAD,LIST,DIR,STAT,WSTAT

SKÆRMEN, koordinatsystemer

Nedenstående forudsætter, at der benyttes MODE 512, hvor skærmen har:

512 "individuelle" pixler vandret x 256 pixler lodret og

4 farver: 0 = sort 2 = rød 4 = grøn 7 = hvid.

MODE 0 = MODE 4 = MODE 512. Farvekommando 1,3,5,6 giver hhv. 0,2,4,7.
MODE 8 = MODE 256 har 8 farver - men kun 256 (dobbelte) x 256 pixler.

Skærmen kan inddeles i vinduer (WINDOW) med hver sit kanalnummer, # ,
PAPER-farve, INK-farve, BORDER-bredde,farve (vindues-kantlinie).

Skærmen har tre KOORDINATSYSTEMER:

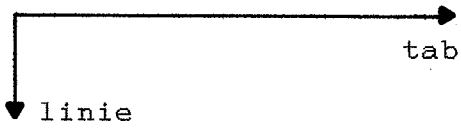
PIXEL-KOORDINATSYSTEMET,
der benyttes til placering af WINDOW's
på skærmen og til interne placeringer
i det enkelte vindue, fx af CURSOR.



GRAFIK-KOORDINATSYSTEMET,
der benyttes internt i vinduerne til
grafik, fx linier, punkter, cirkler og
til skala, CURSOR mv.



PRINT-KOORDINATSYSTEMET,
der benyttes til udskrift af bogstaver
og tal samt selvdefinerede symboler og
tegn.



Alle koordinatsystemer har første pixel / tegn i 0, 0.
Hele skærmen har derfor: xpix= 0...511 og ypix= 0...255.
ygraf= 0...skala og xgraf= skala * 1.48 (SCALE skala,0,0).
Endvidere linie= 0...24 og tab= 0...84, idet en karakter optager
6 x 10 pixler, dvs 25 linier à 85 bogstaver. Grafik- og pixel-koordi-
naterne har en vis sammenhæng, med faktorerne 6 og 10, men med "x, y"
ombyttede, idet karakterer placeres med linie,tab (!).

SCALE, se senere. SCALE 100,0,0 er default-værdien.

pixel-koordinater benyttes af
BLOCK, CURSOR, PAN, SCROLL mv.

grafik-koordinater benyttes af
ARC, CIRCLE, CURSOR, LINE, POINT, SCALE mv.

relative koordinater for tegne-kommandoer begynder med
placering af grafik-markøren (POINT, LINE x,y)

CURSOR (text-markøren) placeres ved PRINT, AT, TO
og ved grafik- og pixel-koordinater.
CURSOR styrer også PAN, SCROLL mv.

PRINT-koordinater benyttes af PRINT, AT, TO.

SKÆRMEN, vinduer, kanaler.

Skærmen har fra starten tre vinduer, forbundet til:

kanal 0 (#0) til kommandoer, fejludskrift mv.
kanal 2 (#2) hvori programmet udskrives, medmindre man udtrykkeligt har bedt om noget andet (LIST#n).
kanal 1 (som benyttes, uanset om man har skrevet #1 eller ej). Her udskrives alt andet, som ikke er sendt til #n.

Kanal 3 er den kanal, der normalt benyttes til printeren. (#3, ser1).
Alle kanaler udover 0, 1, 2 skal åbnes før brug. De bør lukkes igen, så man ikke får fejludskrift, hvis de senere åbnes (til andet brug).
Luk aldrig flere kanaler op end nødvendigt. Brug lave numre.
Kanal 0 kan (heldigvis) ikke lukkes.
Kanal 1 og 2 kan lukkes, men må så åbnes igen.
Enhver kanal kan åbnes til et vindue, der defineres samtidigt (eller senere). Som scr til udskrift. Som con_ , hvis den også skal bruges til INPUT. Se fx linie 30000 ff.

QL's åbningsskærm, med kanal 1 til højre (!!) retableres med WMON 4.
Den indeholder:

```
WINDOW#0; 512,256,  0,  0: PAPER#0; 0: CLS#0
WINDOW#0; 512, 50,  0,206: PAPER#0; 0: INK#0; 4
WINDOW#1; 256,202,256,  0: PAPER#1; 2: INK#1; 7: BORDER#1; 1,7,0
WINDOW#2; 256,202,  0,  0: PAPER#2; 7: INK#2; 2: BORDER#2; 1,7,0
```

Vinduer defineres i pixel-koordinatsystemet ved længde, højde, x, y, (x, y i øverste, venstre hjørne). Det ses at ypixel 202....205 ikke benyttes. Prøv: PAPER#0; 4: CLS#0.

Det er klogt at starte med den helt sorte skærm, kanal 0, førend man definerer sine vinduer. Derved hindrer man, at evt ubenyttede områder lader tidligere vinduers indhold blive stående urørt. Det er hæsligt, men ses, selv i professionel software.

SKÆRMEN, farve, BORDER.

Baggrundsfarven fastlægges med PAPER#n; farve.
Skrift- og linie-farven fastlægges med INK#n; farve (farve= 0,2,4,7).

Til fler-farvegrafik anbefales PAPER 0 og liniefarverne 2, 4, 7, idet andre kombinationer giver ringere kontrast mellem to af de benyttede farver.

BORDER, dvs indramningslinien er noget kunstig.
GENEREL KOMMANDO: BORDER#n; bredde, farve: BORDER 1,2 giver i #1 en rød ramme, 1 pixel bred langs de vandrette kanter, men 2 pixler langs de lodrette kanter. Det er lumsk, især da BORDER reducerer det brugelige areal og dermed har indflydelse på skalaen. Er et vindue defineret som 202 højt, er det kun 200 pixler højt efter BORDER 1,2, dvs at skalaen ændres fra 100 = 202 pixler til 100 = 200 pixler. Pas på med INPUT, PRINT mv. Bogstaver flyttes til højre, nedad, når Border øges. Dette betyder også, at vinduers højde bør være n x 10 + 2 pixler, da der så er plads til BORDER 1, samtidigt med at der er n linier til at udskrive karakterer.

Her skal det straks tilføjes, at når QL's normalvindue angiver BORDER med tre parametre, betyder det kanal, farve, stipple: Et sort - hvidt skakbrædtmønster, der ikke kan anbefales som PAPER-farve, da det i de fleste tilfælde er umuligt at se linier eller læse bogstaver på mønstrede baggrunde.
I seriøse programmer benyttes normalt MODE 512 og rene farver.

BORDER med een parameter er tilladt og giver en usynlig BORDER, der kan benyttes til flere formål.

BORDER 10 vil give en 10/20 pixler bred, usynlig BORDER, der medfører
a) at man ved farveskift får bevaret den tidligere PAPER-farve langs randen af vinduet,
b) at al ny udskrivning nu foregår med linie og tab forøget med 1 hhv 20/6, en fremgangsmåde der fx flytter text "ind på skærmen", se fx "delsave" under c80routiner,
c) at det, der findes under den usynlige BORDER, ikke fjernes af CLS.
I PROCeduren "kantlinie" tegnes en een pixel bred kantlinie - pænere end en BORDER - og denne dækkes derefter med en usynlig BORDER.

Der er generelt anbefalelsværdigt at benytte BORDER 0 før CLS, for så at retablere en ønsket kantlinie / BORDER.

SKÆRMEN, pixler (printer, plotter).

En pixel er ikke en cirkulær eller kvadratisk "prik".
Det er en udflydende klat på et uskarpt gitter, men dog noget, der er brugeligt til fremstille linier, tegn osv.
Den bør opfattes som et rektangel, hvis højde er 1.355 gange bredden.
QL's grafiske system, grafik-koordinater, er indrettet, så fx cirkler bliver til cirkler på skærmen. LINE 0,0 to 50,50 giver en linie under 45 grader, men "den hakker i det", og det gør den også på printeren, uanset om printer-driveren giver forvrængning eller ej.

SCOPY giver en vis forvrængning af skærmbilledet på CPA80-printeren, men det er i mit tilfælde en fordel: I mit 120 + 120 grader isometri-program, bliver x- og z-akse - parallele linier - dem er der mange af i et byggeri - udtegnet med tangens = 1.355 / 2, ca 34 grader. Skærm-linierne er derfor regelmæssige. 2 pixler hen, 1 op, osv. De ser også anstændige ud på printeren, og vinklen er reduceret til ca 30 grader.
I "vvs_tegning" benyttes den i branchen normale 45+45 grader isometri med tan = 1.355, ca 54 grader på skærmen, på printeren reduceret til ca 45 grader, i begge billeder med helt regelmæssige linier. Program-met accepterer kun akse-parallele linier.

Mine to isometri-programmer har derfor også flere sæt koordinater til de enkelte punkter: Rumlige x,y,z. Skærm x,y. Plotter x,y. Se linie 2390 - 2470 (med bemærkninger) og 7700 - 7780.

De enkelte printere kan justeres mere eller mindre godt.
SCOPY / QCOPY er rimeligt gode til CPA80S- / QL-printerne.

Generelt bør man forlange, at hverken skærmbillede eller printerkopi udviser forvrængning, da disse afbildninger nok er uacceptable i fx rapporter og lign., men dog gode nok til personlige analyser, orientering, huskeseddel, mellemresultat, arbejdskopi. Skærmbilledet skal også være en orientering om, hvad der kan forventes på plotteren. Her kan mine programmer, p.gr.af de to koordinatsæt godt narre en smule.

SKÆRMEN, SCALE.

Med skalaen, SCALE , kan man få en figur udtegnet i ønsket størrelse, eller man kan sikre sig, at alle figurer, inden for fastlagte grænser altid kan afbildes på skærmen. QL opgiver ikke at tegne en "for stor" figur, men tegner det, der ligger inden for skærmens grænser. Ved at ændre SCALE kan man således fx gøre en figur større, så kun en del af figuren udtegnes.
Ved at vælge en passende skala kan man også sikre, at linier let kan udtegnes uden at skære bogstaver og tal. SCALE kan også benyttes til at skabe en simpel sammenhæng mellem linieafstande, mål og pixelkoordinater.
Benyttes fx hele skærmen, vil skalaen 255 medføre, at ygraf = ypix, at 10 ygraf = bogstavernes linieafstand, og at skærmen grafisk dækker formatet 377 x 255. Skala 345.5 vil give xgraf = xpix, 13.55 ygraf = bogstavernes linieafstand, og formatet 345.5 x 511 (grafisk).
Skala-valg må træffes i relation til den valgte skærm og dens BORDER og afprøves. Det sidste ciffer / decimal er vanskelig at overskue, se beregningerne til isometri_-vindue, pag. 98-99.

Skalaen angives ved
SCALE#n; skala, xgrafisk, ygrafisk.
x og y sættes ofte til 0,0 , dvs med det grafiske nulpunkt i vinduets nederste, venstre hjørne. Nulpunktet kan vælges inden eller uden for vinduet. SCALE 150, 200, 200 medfører, at der kun tegnes for ygrafisk i intervallet 200 til 350. Brugervejledningen forekommer mig i øvrigt ulogisk på dette punkt. Jeg ville have begyndt med 150, -200, -200 og fået et uønsket resultat. Tilsvarende gælder PAN og SCROLL.
Prøv SCALE 150,200,200: LINE 201,201 TO 201,349 TO 338,349 TO 338,201 TO 201,201. Resultatet er en hvid linie langs kanten af QL's skærm 1.

Ved brug af varierende skala og gentagen figurudtegning kan tegninger forstørres, formindskes, flyttes. Ønskes ZOOM-effekt, er OVER -1 velegnet, se nedenfor. Skala og x,y må afpasses, fx så et bestemt punkt eller figurens centrum fastholdes.
Flytning af tegninger kan også opnås med PAN, SCROLL eller med POINT med varierende koordinater plus LINE_R til figuren. Se nedenfor.

SKÆRMEN, LINE (og andre streg-ordrer).

Der benyttes grafiske koordinater. LINE a,b TO c,d giver linien fra a,b til c,d minus den sidste pixel. Hvis der ikke skal tegnes videre fra a,b, bør der sluttes med POINT c,d. Grafikmarkøren, POINT x,y / LINE x,y kan benyttes som udgangspunkt for relative koordinater, fx LINE_R TO a,0 TO 0,b TO -a,0 TO 0,-b, dvs et rektangel på a x b.

SKÆRMEN, PAN, SCROLL, CLS.

Se bruger-vejledningen og pas på fortegnene for pixlerne.
CURSOR (markørpositionen) er givet ud fra forudgående PRINT-kommandoer, eller styres som angivet nedenfor.

SKÆRMEN, CURSOR.

Printmarkøren, CURSOR , styres af koordinater til markørens øverste, venstre hjørne, men kun næsten. Prøv dette korte program:

```
LINE 0,10 TO 20,10: LINE 10,0 TO 10,20
CURSOR 10,10,0,0: PRINT CHR$(27)
```

CHR$(27) tegner et skakbrædt på 5x9 pixler i kvadrant 4, MEN fjerner den lodrette linie, ikke den vandrette linie (!).
Se nedenfor om udskrift af karakterer og om vvsfount, pag. 100

CURSOR kan have to eller fire parametre.
CURSOR#n; xpix, ypix benytter pixel-koordinater.
CURSOR#n; xgraf, ygraf, xpix, ypix benytter det grafiske koordinatsystem, absolut, i vinduet, og pixel-koordinatsystemet relativt til den grafiske markør. Flytningen - i forhold til det grafiske nulpunkt - er derfor: mod højre: xgraf + xpix, og opad: ygraf - ypix.

DET ANBEFALES KRAFTIGT, at man altid sikrer sig en klar og erkendt omsætningsfaktor mellem grafik- og pixel-koordinater. SCALE indgår i regnestykket, der er lidt besværligt og må checkes. Se indledningens beregninger af isometri vindue. Se også bemærkningerne om SCALE og xxmin i linie 2500 ff og i linie 20410 ff (punktkontrol, figur side 67). Skala / 110 og pixel-formatet 1.355 er "hellige" tal på QL.

SKÆRMEN, udskrift af karakterer (bogstaver, tal, tegn, symboler).

MODE 512 fører automatisk til CSIZE 0,0 i alle kanaler, dvs normale 6 x 10 pixler store karakterfelter, teksten pag.94 , vvsfount. De andre størrelser af CSIZE kan defineres, i en kanal, i MODE 512.

I vandret retning udskrives karakterer pr. 10 pixelrækker = 1 linie, når den anden parameter i CSIZE (højden) er 0.
Pixelrække 0 : Benyttes ikke. Fjerner, hvad der måtte være tegnet i denne række (undtagen ved OVER 1 eller OVER -1).
række 1-2: Benyttes af bl.a. tal og STORE bogstaver.
række 3-7: Benyttes af alle bogstaver og tal.
række 8-9: Benyttes af små bogstaver med nedstreger, fx g og j, og af kommaer mv.
Understregning (UNDER 1) benytter række 8.
I vandret retning udskrives karakterer således (TAB = ialt):

Pixelantal:	Blank	Karakter	Blank	Ialt		Karakterer / linie
CSIZE 0, h	1	5	0	6	>	max 85
CSIZE 1, h	1	5	2	8	>	max 64
CSIZE 2, h	2	10	0	12	>	max 42
CSIZE 3, h	2	10	4	16	>	max 32

CSIZE 0,0 benytter således et 6 x 10 pixelfelt, medens CSIZE 3,2 benytter et 16 x 20 pixelfelt, idet h = 1 sætter højden = 20 pixler.

ALLE KARAKTERER er opbygget over en 5 x 9 matrix, som i CSIZE 0,0. CSIZE 2,h gør alle karakterer "dobbelt-brede" ved vandret gentagelse af pixlerne. CSIZE 1,h og 3,h spatierer karaktererne ved at tilføje 2 hhv 4 blanke pixler mellem karakterne.
CSIZE b,1 øger højden til 20 pixler ved fordobling i lodret retning.

Karakterer kan placeres med PRINT, AT, TO (#n;), se vejledningen. Mere nøjagtigt i pixelmønstret med CURSOR og fire parametre.

```
10 PRINT 'PER'
20 PRINT \ 'OLSEN' \\ 'JENS'
30 PRINT TO 4; 'LISE';
40 PRINT 'ANNA'
```

```
PER

OLSEN

JENS
    LISEANNA
```

giver følgende udskrift, idet

linie 10 er "normal".
linie 20 begynder med et extra linieskift og har et dobbelt skift mellem OLSEN og JENS.
linie 30 starter i TAB 4, dvs på 5.plads og slutter med ; der giver fortsat udskrift !

SKÆRMEN, STRIP#n; farve.

STRIP#n; farve ændrer baggrunds-farven, uafhængigt af den aktuelle PAPER-farve. Der skal i reglen tilføjes en ændret INK-kommando. Husk at stille PAPER og INK tilbage. Husk: ALT udskrives med STRIP, dvs at fx PRINT TO t; 'xx' vil give t blanke i STRIP's PAPER-farve !

SKÆRMEN, BLOCK#n; lpix, hpix, xpix, ypix.

Kan benyttes til - på een gang - at etablere et felt for udskrift med en speciel farve. Se fx "tput" (tillægget), linie 140.

SKÆRMEN, karakterudskrift ved punkter, linier

CURSOR med 4 parametre benyttes til nøjagtig placering af fx numre og dimensioner langs fx linier. Se bl.a. isometri_tegning og (især) de i vvs_tegning benyttede xpix- og ypix-korrektioner, afhængigt af, om der skal udskrives tal over / under, til højre / venstre for en linie med en eller anden hældning, med en så stor afstand, i pixler, at der ikke opstår ulæselige tal. Opgaven er (næsten) uløselig, hvis alle linie-hældninger er tilladt. Se følgende simple program:

```
100 MODE 512: CLS
110 LINE 0,50 TO 100,50: LINE 50,0 TO 50,100
120 CURSOR 50,50, 2,-12: PRINT CHR$(27);1
130 CURSOR 50,50,-8,-12: PRINT CHR$(27)
140 CURSOR 50,50,-8,  0: PRINT CHR$(27)
150 CURSOR 50,50, 2,  0: PRINT CHR$(27);4
```

De to linier skærer i 50, 50 (grafisk). CURSOR dirigeres til dette punkt og korrigeres derefter med pixel-koordinater til en placering, der (netop) giver en adskillelse mellem linie og tal.
OVER 1 bør benyttes, hvis der ønskes udskrift tættere på, eller hvis der benyttes selvdefinerede, små tal. Se vvsfount i indledningen.

SKÆRMEN, OVER 0, 1.

OVER#n; 0 er default, det normale.
OVER#n; 1 medfører, at der skrives "oven i" det existerende. Fx vil bogstavfeltets blanke pixler ikke fjerne linier (eller andre tegn). Dette kan udnyttes til at skrive karakterer tættere sammen, se bl.a. isometri_tegnings udskrift af tal, linie 9900 ff med kun 4 pixler ml. de selvdefinerede, 3 pixler brede vvstegn (1 blank + 3 + 2 blanke). Herved kan selvdefinerede symboler (UDC / TOOLKIT) på 5 x 9 pixler kombineres uden lodrette og vandrette blanke pixelrækker. Se CHAR_INC og CHAR_USE i TOOLKIT. CHAR_INC (increment) sættes til 5,9 i stedet for det normale, 6,10. OVER -1: Se næste side.

SKÆRMEN, OVER -1.

OVER#n; -1 er mere subtil (En XOR-funktion). Udskrift af karakterer sker i an anden farve. Skrives der oveni noget existerende, vil der i skærings-pixlerne mellem det existerende og det nye (linier / tegn) opstå en tredje farve. Gentages udskriften, på samme sted, er alt det existerende retableret, det nye forsvundet !
OVER -1 kan generelt benyttes til udskrift af noget, der skal fjernes pæny. Man kan bevæge en figur, med gentagen udtegning og "vandrende" koordinater, fx relative til POINT, henover en anden figur eller et koordinat-net, uden at der opstår "huller" i linier, tegn, figurer.

OVER -1 kræver omhyggelig analyse, hvis der ikke skal opstå en uantagelig farveforvirring. Se dette korte program:

```
100 MODE 512: PAPER 2: INK 7: CLS
110 AT 13,1 : PRINT 'ABCDEFGHIJKLMNOPQRSTUVWXYZÆØÅ'
120 OVER -1 : INK 4
130 FOR n = 1 TO 66
140   LINE 2 + n, 20: LINE_R TO 0,11 TO 2,0 TO 0,3 TO -2,0 TO 0,-3
150   IF n= 66: EXIT n
160   LINE 2 + n, 20: LINE_R TO 0,11 TO 2,0 TO 0,3 TO -2,0 TO 0,-3
170 END FOR n
180 PAPER 2: INK 7
```

I linie 110 udskrives alfabetet i hvidt på rød baggrund.
Derefter udskrives, fra 2 + n, 20 til 2 + n, 31 en lodret pegepind med et 2 x 3 stort rektangel på toppen. Pegepinden vandrer mod højre mens rektanglet udskrives over bogstaverne. Når pinden standser, til højre for alfabetet, er bogstaverne intakte. Alt foregår i hvidt og rødt.
Ændres linie 120 til ...INK 7, er pegepind og rektangel grønne, dog er pixler, fælles for rektangel og bogstaver, sorte. Benyttes INK 2, fås en sort pegepind med grønne fællespunkter. INK 0 har ingen synlig indflydelse. OVER -1 med INK 4 er altså pænest på PAPER 2 og 7.

Forklaringen kan findes ved at skrive farverne binært og addere dem ud fra OVER -1's XOR-funktion: 1 + 1 = 0, 0 + 1 = 1, 0 + 0 = 0.

PAPER 2 = 010 og INK 7 = 111 samt OVER -1 og INK 4 = 100 giver da:
Baggrundsfarve + pegepind: 010 + 100 giver 110, dvs farve 6, MODE 512, dvs hvid. Fællespixlerne får 111 + 100, dvs 010, farve 2, rød. Herved afsløres det også, hvad øjet knap opfattede, at rektanglet ikke bliver fuldt udtegnet, førend pegepinden standser, efter alfabetet.

PAPER 2 = 010 og INK 7 = 111 samt OVER -1 og INK 7 = 111 ville give et lidt mere forvirret billede. Pegepind : 010 + 111 = 101, farve 5, MODE 512, dvs grøn. Fællespixler : 111 + 111 = 000, dvs farve 0, sort.

PAPER 2, INK 7, OVER -1, INK 2 giver 000, sort pegepind og 101, grønne fællespunkter.

At OVER -1, INK 0 = 000 ikke giver noget resultat, er da vist klart.

LINIER KAN KUN TEGNES OG FJERNES
- MED OVER -1 eller MED INK = PAPER -
HVIS DE TEGNES OG FJERNES I SAMMME RETNING.
Se også teksten til "nytast" pag. 47.

I "tput" benyttes OVER -1 på en anden måde:
Her udskrives gældende DATA, til revision eller accept, i et felt, der er rødt / hvidt eller hvidt / rødt, valgt i kontrast til det aktuelle vindues PAPER-farve, fx 2, der vil give: STRIP 7, INK 2 (linie 18020).

I indtastløkken, REPeat tast2 (linie 18130 - 18290), benyttes OVER -1, men INK 2 ændres ikke. I det felt, hvor CURSOR befinder sig, udskrives den gældende karakter påny, og da 010 + 010 = 000, ses CURSOR's placering af, at karakteren ikke er rød, men sort (linie 18150). CURSOR kan flyttes med pilene (linie 18170 f), og det nye felt får sort karakter, mens det forladte felt igen har en rød karakter (linie 18170).

Tastes ENTER, er indtastningen slut, hvis resultatet kan godkendes af PROCeduren checkstreng (linie 18200 - 18210 og 18400 - 18580).

Det endelige resultat udskrives med skærmens oprindelige PAPER og INK, se linie 18300 ff.

DANSKE OG EUROPÆISKE BOGSTAVER

Bogstaver udskrives på PC'ere med begrænset opløsningsevne (pixelantal) og på matrixprintere i en lille matrix, fx 5×9 pixler, der er udviklet til angelsaksiske sprog. De fleste europæiske sprog benytter accenter, boller etc. over en række bogstaver. Alle nationer må også acceptere andres stavemåder i korrespondance.

En række bogstaver kan derfor ikke udskrives på billige datamater, og på almindelige PC'ere kun i forvrænget form.

Når der i Danmark ses ae, oe og aa, er det måske nødvendigt i programmer, hvor ikke alle datamater (som QL) accepterer æ, ø og å i variabler, men det er REN DOVENSKAB i udskriften.

ENHVER PLOTTER OG MATRIXPRINTER KAN SKRIVE "DANSK", med større eller mindre succes. Grimme, danske bogstaver skyldes dels fabrikanternes manglende forståelse, dels at matricen er for lille: for lav til et anstændigt Å.

Pag. 90-96 er udskrevet af QL-printeren i NLQ (near Letter Quality), og det er vel stort set læseligt. Dette er udskrevet med IBM-skrivemaskine. Hvilken forbedring er det ikke !

Figuren viser dels QL's Æ og Ø, dels mit forslag, "qltegn", til forbedring. Endvidere to prøveudskrifter med CPA80 og QL.

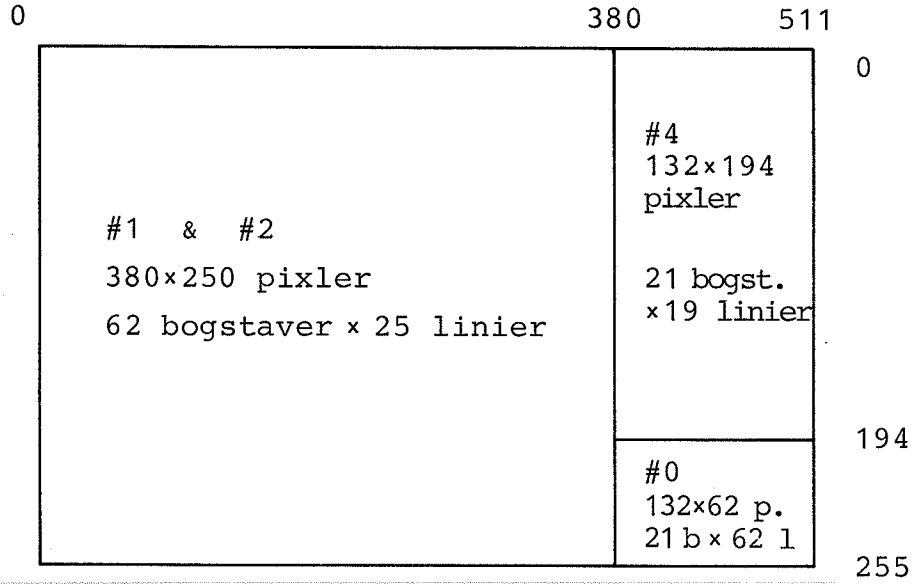
Pag. 100-101 viser 3×5 pixler store tegn, "vvsfount" til tegninger.

CPA80S-printeren forsyner bl.a. Å med en sær klat, se disse exempler:
MÅL, ÅA MÜLLER, ÜU ESPAÑA, ÑN

QL-printeren gør til gengæld Å til en sær dværg - i Near Letter Quality:
MÅL, ÅA MÜLLER, ÜU ESPAÑA, ÑN ELYSÉE, ÉE

SÅ LET KAN Æ OG Å GØRES BEDRE: 

Isometri-vindue PRINCIP, KANAL 0,1,2,4




```
window#0; 132,62,380,194
window#1; 380,256,0,0
window#2; 380,256,0,0
window#4; 132,194,380,0
```

Her er valgt et nærmest kvadratisk vindue, da det normalt er bedst egnet til 120^0-isometri. Sammenlign vvs-tegnings aflange vindue.

Skala= 1,095 * hmax
Skala= 0,972 * lmax
} Den største vælges

SCALE Skala, $-\frac{11}{344}$ * skala, $-\frac{11}{254}$ skala

Opad = skala

Nedad = skala * $\frac{11}{254}$

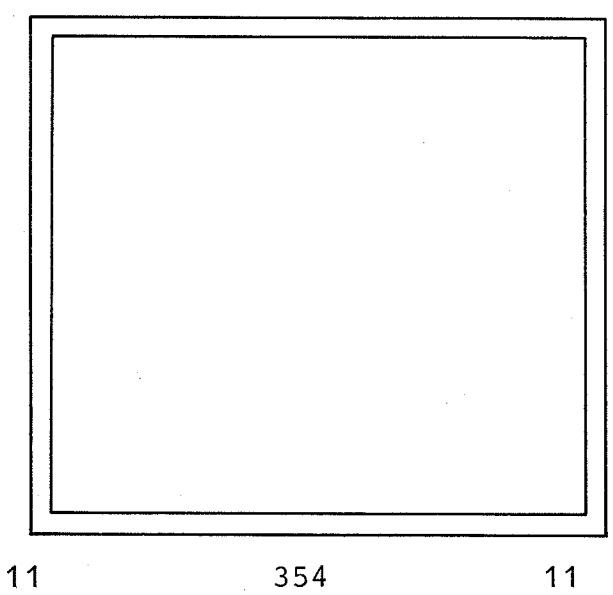
Venstre = skala * $\frac{11}{344}$

Højre = skala * 1,098

PIXEL-FORMAT : h=1.355×b

SE SKÆRMOVERSIGT næste side

Isometri-vindue SKÆRMOVERSIGT. KANAL 1 (2), 380 x 256 pixler



BORDER 1

11=plads til tal
(1 i højderetningen)
(2 i vandret retning)
(tal er vvs-tegn)

254=256 - 2 * BORDER 1
376=380 - 2 * BORDER 1 (som er 2 pixler vandret)

```
lodret   = 254 pixler = skala           * skala
            11 pixler = 11/254          * skala = hmax
           232 pixler = 232/254         * skala
             1 pixel  = 1/254           * skala
vandret      1 pixel  = 1/254/1,355     * skala
                      = 1/344,17        * skala (pixelformat)
            11 pixler = 11/344          * skala
           354 pixler = 354/254/1.355   * skala = lmax
```

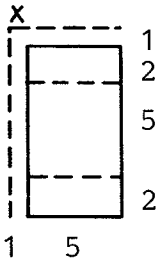
Heraf fås:

skala = hmax * 254/232 = 1.095 hmax
skala = lmax * 254/354 * 1.355 = 0.972 hmax

NB. I praxis bør tilsvarende "beregninger" efterprøves grundigt, især i vandret retning.

vvstegn (vvsfount, 875 (876) BYTES)

små tal og bogstaver.



Normale karakterer (i CSIZE 0,0) dannes indenfor en 5×9 matrix (45 pixler). De øverste 2 rækker benyttes af STORE bogstaver og små, høje bogstaver (fx "h"). De nederste 2 rækker benyttes af nedragende, små bogstaver (fx "j"). Afstand 6 pixler). Hertil kommer en pixelrække over og en pixelrække til venstre for selve karaktermatricen. Det felt, der udskrives, er således 6×10 pixler, jfr. pag. 94.

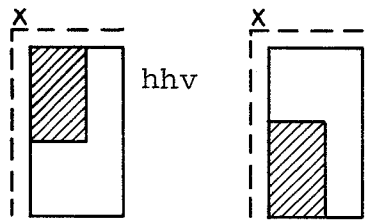
Med OVER 0 fjernes alt tidligere skrevet og tegnet inden for dette felt.

Med OVER 1 sker dette ikke, og CHAR_INC kan reduceres fra det normale 6,10 til 5,9 eller 4,8.

CURSOR (4 parametre) placerer karaktererne eksakt, ud fra den med x markerede pixel, der ligger uden for feltet (!)

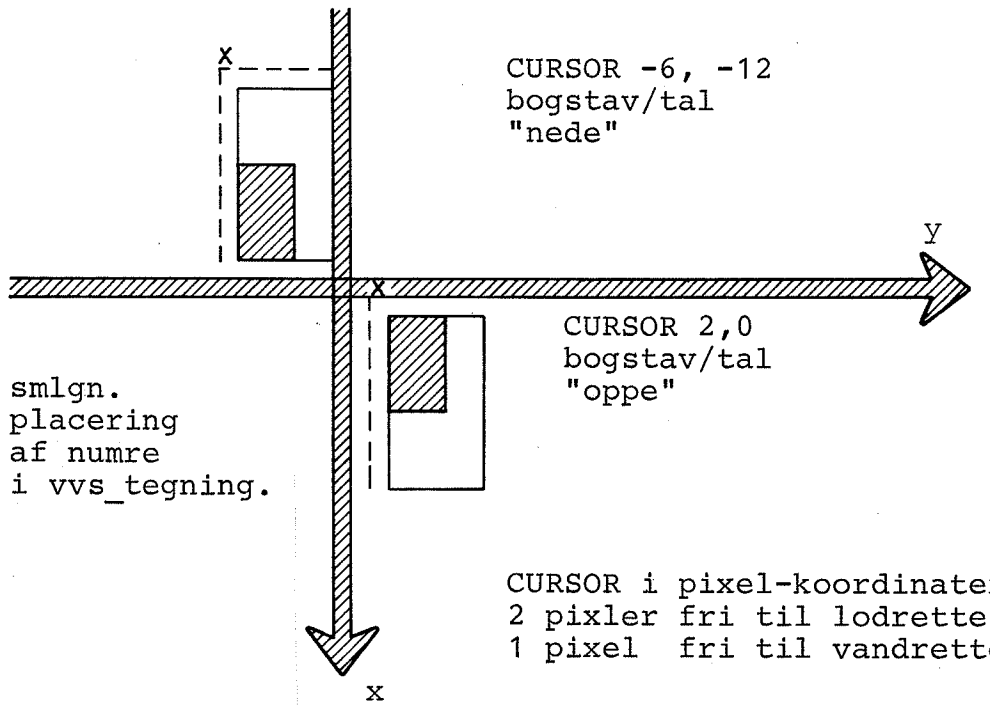
UDC (BASIC-program, USER DEFINED CHARACTERS) benytter den samme 5 9 (6 10) matrice. Små karakterer må have reduceret CHAR_INC, vandret til 4. De må skrives i "rigtig" rækkefølge, eller OVER 1 må benyttes.

"vvsfount" har bogstaver og tal "nede" og "oppe", alt som store bogstaver:



x = CURSOR

vandret bogstavsafstand 4 pixler (CHAR_INC)



CURSOR -6, -12
bogstav/tal
"nede"

CURSOR 2,0
bogstav/tal
"oppe"

smlgn.
placering
af numre
i vvs_tegning.

CURSOR i pixel-koordinater.
2 pixler fri til lodrette linier.
1 pixel fri til vandrette linier.

TAST	CHR$	vvsfount nede	TAST	CHR$	vvsfount oppe
0	48	0	SHIFT 1 (!)	33	0
1	49	1	SHIFT ' (")	34	1
2	50	2	SHIFT 3 (#)	35	2
3	51	3	SHIFT 4 ($)	36	3
4	52	4	SHIFT 5 (%)	37	4
5	53	5	SHIFT 7 (&)	38	5
6	54	6	'	39	6
7	55	7	SHIFT 9 (()	40	7
8	56	8	SHIFT 0 ())	41	8
9	57	9	SHIFT 8 (*)	42	9
+	43	+	SHIFT . (:)	58	+
'	44	'	SHIFT , (;)	59	'
-	45	-	SHIFT = (<)	60	-
.	46	.	=	61	.
			/	47	<
			SHIFT + (>)	62	>
a	97	A	SHIFT a (A)	65	A
'	'	'	'	'	'
'	'	'	'	'	'
'	'	'	'	'	'
'	'	'	'	'	'
z	122	Z	SHIFT z (Z)	90	Z
CTRL= ({)	123	Æ	CTRL 9 ([)	91	Æ
CTRL DEL (!)	124	Ø	SHIFT DEL (\)	92	Ø
CTRL+ (})	125	Å	CTRL 0 (])	93	Å

-15 (CHR$ 37 → 38)

+15 (CHR$ 58 → 59)

-32 (CHR$ 65 → 90)

-32 (CHR$ 91 → 93)

I cirklen er angivet forskellen i CHR$-værdier

SKÆRMEN, 32768 bytes.

Skærmens samlede indhold kan lagres med SBYTES_FLP1_navn,131072,32768. Med LBYTES FLP1_navn,131072 kan det lagrede skærmbillede fremvises, men pas på, der ikke dukker en CURSOR op i kanal 0, der forstyrrer det nye skærmbillede. Se boot, linie 222, hvor INKEY$(-1) løser problemet. I boot, såvel som i isometri_tegning, linie 202 - 246, er der i øvrigt benyttet en meget hurtigere indlæsningsprocedure. Prøv med SBYTES plus LBYTES og se den ejendommelige måde, et skærmbillede opbygges på.

Skærmens opbygning ses af figuren, dog uden nogen egentlig forklaring på, hvordan QL håndterer et skærmbillede.

Med POKE_W og PEEK_W kan information for 8 pixler (2 bytes, 16 bit) sendes / opnås. Det tager lang tid for selv en mindre del af skærmen, og bør kun bruges til fx at anbringe en lille streg på et bestemt sted på skærmen, uafhængigt af den aktuelle vindues-fordeling. Et logo kan hurtigere indsættes med en PRINT / LINE -PROCedure, som imidlertid vil være bundet til et vindue.

På figuren står SBYTES navn,131072,32768. Bemærk den korte notation, der benyttes i megen software. Navn skal ofte - som her - forstås som fx FLP1_navn eller MDV2_navn_tillægsnavn. I andre tilfælde er mediets navn givet, fx i form af flp$, således at navn kun er et navn, smlgn. boot, linie 102 og isometri_tegning, linie 404, der simpelt kan rettes efter brugerens behov. SAVE sker da med SAVE flp$ & 'navn'. Metoden er ikke uden risici, da denne streng-udformning af medienavn plus filnavn medfører, at QL accepterer stort set alt som filnavn, også strenge med mystiske tegn, flere ord osv. Disse filer kan kun LOADes ved en analog anvendelse af strenge. Heldigvis er DIR ligeglad.

SAVE, LOAD, LIST.

Filer SAVEs og LOADes ved angivelse af mediet_navnet, jfr. bemærkningerne i slutningen af forrige afsnit.

Mediet kan være FLP1_, FLP2_, MDV1_, MDV2_ og RAM1_....RAM8_.
Mediet skal være FORMATteret først, for FLP og MDV med et medienavn.
MIKRODREV skal FORMATteres, fejlfrit, med mindst 200 sektorer, mindst 2 gange. En enkelt mislykket FORMATtering kan vel accepteres, hvis den efterfølges af mindst 4 korrekte FORMATteringer.
FLOPPIES skal FORMATtere, første gang, til 1440 / 1440 sektorer. Hvis dette ikke er tilfældet SKAL FLOPPIEN RETURNERES.

RAM FORMATteres uden navn med angivelse af antal sektorer (512 bytes), 2 til 900, afhængigt af behov og FREE_MEM. RAM bruges som en FLP, men RAM slettes af RESET, ikke af NEW, LOAD etc. Exempel: FORMAT RAM5_40.

Bemærk: et medies _navn må ikke ende på Æ, Ø, Å. Det må en PROC og en variabel godt. Under brugen skelnes ikke mellem STORE / små bogstaver. MEN QL er stædig: Den første kombination af STORE / små vil være fastlåst i resten af programmet.

Filer kan omdøbes (Mediet dog ikke): RENAME FLP1_Ohlsen,OLESEN.

Software indeholder ofte vejledning under navne som instruktion, vejledning, læs_mig_først etc. Disse filer er ofte ASCII-filer, oprettet fx med skærmeditoren. De læses med COPY_N FLP1_læs_mig_først TO scr eller ..TO #3,ser1 til printeren (kanal 3 skal åbnes først).

Et medies indhold af filer kan udskrives med DIR#n; FLP1_ (n = 1-16). Der udskrives mediets navn, benyttede / totale antal sektorer og alle filers navne.

En mere oplysende udskrift fås med:
STAT#n; FLP1_ mediets navn og benyttede / totale antal sektorer.
WSTAT#N; FLP1_ filernes navne, længde (bytes), oprettelsesdato.

Enhver start af QL bør indledes med indtastning af SDATE..., så denne datofacilitet bliver udnyttet. Programmer burde derfor have en SDATE som første handling. Så bliver det heller ikke glemt.

WSTAT kan i øvrigt nøjes med at udskrive de filer, der begynder eller ender med en bestemt streng, fx WSTAT FLP1_ole_ / WSTAT#3; MDV2__bin

Endelig må jeg henvise til TOOLKIT bl.a. vedrørende CLOCK, GET, PUT, BGET, BPUT, WDEL, WDIR, UDC, CHAR_USE, CHAR_INC etc, etc.
UDC, User Defined Characters, er i BASIC og ret så lærerigt. Prøv fx AT 10,20: BPUT 108. Nu står der l midt på skærm 1. Prøv så PUT 108 !

ALTERNATIVT LOADEPROGRAM (kan ikke kompileres)

Dette er program "boot".
Det LOADer "isometri_tegning" med "startiso" som mellemled.
Derved opnås, at der ikke er en sort skærm, mens hovedprogrammet LOADes. Skærmbilledet "isostart" med ur fremvises, i realiteten det samme billede som benyttes i "instruktion" under INPUT-sekvensen.
Program "boot" er helt magen til "isometri_boot", bortset fra starten og slutnigen og en renummerering :

```
100 REMark ****  LOADE-PROGRAM boot  ****
102   flp$= 'FLP1_': DATA_USE FLP1_: PROG_USE FLP1_: DEST_USE FLP1_
104   TK2_EXT: FOR n=1 TO 2: PAPER#n;2: INK#n;7: BORDER#n;0
106   MODE 8: LBYTES dthifh_bin,131072: PAUSE 150: MODE 4
108   LBYTES scrlogo_bin,131072: FOR n=1 TO 2: CSIZE#n;2,0
110   AT#2;19,1:PRINT#2;'PRINTER QL / CP80A':AT 19,1:PRINT'TAST  Q / C'
114   REPeat p
116   AT 19,18:PRINT'  ':AT 19,18:INPUT pri$: IF pri$(1)INSTR'qc':EXIT p
118   END REPeat p

208   END REPeat engangtil
212  END IF : STRIP 7: INK 2: AT 14,5: PRINT'  PROGRAMMET  '
214   AT 15,5: PRINT'  LOADes  PÅ  ': AT 16,5: PRINT'  2 MINUTTER  '
216   PAUSE 100: PAPER#0;2:INK#0;2: FORMAT#0; ram2_4: INK#0;4:PAPER#0;0
218   SDATE 1986,1,1,0,0,0:BORDER#2;0:PAPER#2;2:INK#2;7:CLOCK#2;' %m:%s'
220   OPEN_NEW#5;ram2_: PUT#5; 'ny', printer$, minitegn, dansk: CLOSE#5
222   BORDER 0: LRUN startiso
```

Dette er program "startiso":

```
200 LBYTES flp1_isostart,131072: MRUN flp1_isometri_tegning
```

PRINCIPIELT LOADEPROGRAM (BOOT)

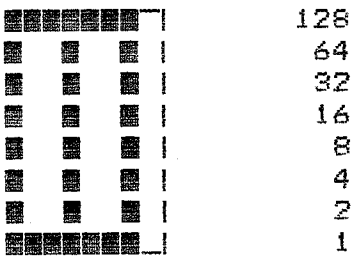
```
ram_Ver1
      1780  1986 Apr 02 19:16:49
qcopy_bin
      1435  1986 Apr 07 22:50:39
scopy_bin
      1434  1986 Apr 07 22:50:35
qltegn
       587  1986 Apr 02 19:15:52
vvsfount
       875  1986 Apr 02 19:15:55
nyptpatch
       544  1986 Apr 02 19:15:59
```

```
10 REMark ****  PRINCIPIELT LOADE-PROGRAM   ****
11   flp$= 'FLP1_': DATA_USE FLP1_: PROG_USE FLP1_: DEST_USE FLP1_
12 REMark flp$ sikrer, at et program let kan omstilles til et andet
13 REMark medie, hvis der overalt i programmet bruges flp$ & 'filnavn'.
14 REMark DATA_,PROG_,DEST_USE betyder, at fx SAVE fil bruger FLP1_,
15 REMark undtagen hvor et andet medie direkte angives.
16   PAPER#2;2: INK#2;7: CLS#2: AT 2,1: PRINT'PRINTER QL / CP80  Q / C'
17   REPeat pr: AT 2,26: INPUT  pri$: IF pri$(1) INSTR 'qc': EXIT pr
18   ramf= RESPR (5224): LBYTES ram_Ver1,ramf: CALL ramf
19 REMark RESPR 5224 er summen af ram_Ver1= 1780, scopy_bin= 1436,
20 REMark qltegn= 588, vvsfount= 876, nyptpatch= 544
21 REMark RAM kan nu formatteres som en FLP og bruges til fx SAVE, LOAD
22 REMark Husk, at RAM forsvinder efter RESET, ikke efter NEW, LRUN.
23   CLS: TK2_EXT: CSIZE 2,0: CSIZE#2; 2,0: PRINT\\'   RAM     LOADed'
24 REMark TK2_ext medfører, at TOOLKIT2 overlejrer INTERFACE-kommandoer
25   IF pri$(1)=='c': printer$= 'c': kop$= 'scopy_bin': BAUD 4800
26   IF pri$(1)=='q': printer$= 'q': kop$= 'qcopy_bin': BAUD 9600
27 REMark qcopy_bin til QL-printeren har samme kommandoer som scopy_bin,
28 REMark ( SCOPY #n osv ). Begge stiller TRA til 0  og tilbage til det,
29 REMark der gælder i programmet. BAUD skal dog stilles til 4800 / 9600
30   KOPI=     ramf + 1780: LBYTES kop$, KOPI: CALL KOPI
31     PRINT \'   SCOPY   LOADed'
32   dansk=    ramf + 3216: LBYTES qltegn   , dansk
33 REMark qltegn, dansk giver pænere æ ø å Æ Ø Å på skærmen.
34   minitegn= ramf + 3804: LBYTES vvsfount , minitegn
35 REMark vvsfount, minitegn er små tal.
36   prik=     ramf + 4680: LBYTES nyptpatch, prik: CALL prik
37 REMark nyptpatch, prik giver korrekt POINT mv ( Kun een pixel )
38     PRINT \'   qltegn  LOADed'
39     PRINT \'   vvstegn LOADed'\\'   ptpatch LOADed'
40   STRIP 7: INK 2: AT 14,3: PRINT'  PROGRAMMET  '
41   AT 15,3: PRINT'  LOADes  PÅ  ': AT 16,3: PRINT'  2 MINUTTER  '
42   PAUSE 100: CLS#0: INK#0;0: FORMAT#0; ram2_2: INK#0;4
43 REMark FORMAT RAM2_2 medfører udskrift af 2/2 sektorer.
44 REMark Her benyttes INK #0; 0 til at gøre udskriften usynlig.
45   SDATE 1986,1,1,0,0,0: CLOCK#2; ' %m:%s '
46 REMark Her startes et ur i kanal 2, der viser LOADe-tiden.
47 REMark Det stoppes i hovedprogrammet med CLOSE#2: OPEN#2; con_
48   OPEN_NEW#5;ram2_: PUT#5; 'ny', printer$, minitegn, dansk: CLOSE#5
49 REMark De fire variabler lagres midlertidigt i RAM. De ville ellers
50 REMark gå tabt ved LRUN, der medfører en NEW for LOADe-programmet.
51 REMark LOADes i hovedprogrammet med OPEN#5;RAM2_:GET#5; osv :CLOSE#5
52   LRUN hovedprogram: REMark Nu bruges DEFAULT-mediet
```

PRINTEREN

PRINTERENS indstillling omfatter:
TRA, der styrer udskrift af Æ, Ø, Å.
QL-printer 0, CPA80-printer 1 ved PRINT.
QL-printer og CPA80-printer 0 ved skærmkopi.
PLOTTER altid 1.
BAUD QL-printer 9600, CPA80-printer 4800, PLOTTER 1200.
Default-værdier : 0 og 9600 (QL-printer) PAS PÅ PLOTTEREN !
BPUT#3, 27,108,5, 27,81,77, 27,67,72, 27,78,10.
Højre margen 5, venstre margen 77, dvs 72 karakterer / linie.
72 linier pr papirark, 10 linier overspringes ved perforering.
De fleste af mine programmer har derfor netop 72 linier i #0.
Ved programmering er følgende vinduer en hjælp:
FOR n= 1 to 2: WINDOW#n; 436, 202, 0, 0.
Se endvidere "c80program", linie 6500, til organiseret udskrift af et program uden sideskift midt i en PROCedure.

BINÆRE VÆRDIER — CPA80S-SPECIALTEGN =



```
                128
                 64
                 32
                 16
                  8
                  4
                  2
                  1

L  L  L     3 x 255      SUM FOR KOLONNER
 LL LL      4 x 129
       L    1 x   0      SE linie 160
```

```
100 REMark ** SELVDEFINEREDE CPA80S-TEGN ********************************
110 OPEN#3,ser1: BAUD 4800: TRA 1
120 BPUT#3;27,108,5: REMark venstre margen 5
130 PRINT#3;'    BINÆRE VÆRDIER';
140 PRINT#3;FILL$(' ',19);'CPA80S-SPECIALTEGN =   ';
150 BPUT#3;27,65,8,27,75,96,0;:REMark 8/72 inch/linie,12x8 pixelkolonner
160  FOR n= 1 TO 12: BPUT#3;255,129,129,255,129,129,255,0
170   REMark 12 karme med to sidehængte rammer
180 BPUT#3;10,10,27,65,12:REMark 2 x LINEFEED,retur til 12/72 inch/linie
190 REMark ** LIDT FORKLARING TIL FIGURUDSKRIFTEN ***********************
200 FOR n= 1 TO 7: BPUT#3;174;
210 BPUT#3;180,229,32,32,32,32,32,32,49,50,56,10
220 BPUT#3;174,32,32,174,32,32,174,32,229,32,32,32,32,32,32,32,54,52,10
230 BPUT#3;174,32,32,174,32,32,174,32,229,32,32,32,32,32,32,32,51,50,10
240 BPUT#3;174,32,32,174,32,32,174,32,229,32,32,32,32,32,32,32,49,54,10
250 BPUT#3;174,32,32,174,32,32,174,32,229,32,32,32,32,32,32,32,32,56,10
260 BPUT#3;174,32,32,174,32,32,174,32,229,32,32,32,32,32,32,32,32,52,10
270 BPUT#3;174,32,32,174,32,32,174,32,229,32,32,32,32,32,32,32,32,50,10
280 FOR n= 1 TO 7: BPUT#3;174
290 BPUT#3;160,229,32,32,32,32,32,32,32,32,32,49,10,10
300 BPUT#3;186,32,32,186,32,32,186,32,32,32,32,51,32,120,32,50,53,53
310 PRINT#3;'       SUM FOR KOLONNER': REMark SIDSTE 10 = LINEFEED
320 BPUT#3;32,186,186,32,186,186,32,32,32,32,32,52,32,120,32,49,50,57,10
330 BPUT#3;32,32,32,32,32,32,32,186,32,32,32,49,32,120,32,32,32,48
340 PRINT#3;'       SE linie 160'
350 BPUT#3;10: LIST#3: BPUT#3;12: CLOSE#3: REMark LINEFEED,LIST,FORMFEED
```

TRE EKSEMPLER PÅ SPECIALTEGN.

1. HER FORUDSÆTTES, AT CPA80S IKKE HAR sigma OG 2. potens:

SPECIALTEGN DEFINERES OG UDSKRIVES VIA PROCedurer.
BEMÆRK, AT PRINTEREN EFTER SPECIALTEGNENE RETURNERER TIL NORMAL PRINT.

TRÆKSPÆNDING, σ= 22 MPa/m² FOR FORSØG NR. 1.

```
100 OPEN#3,ser1: BAUD 4800: TRA 1: BPUT#3;27,108,5
110 PRINT#3;'TRÆKSPÆNDING, ';: sigma :PRINT#3;'= 22 MPa/m';:pot2
120 PRINT#3;' FOR FORSØG NR. 1.'
130 PRINT#3;FILL$(CHR$(174),44)
140 BPUT#3;10,10,10: LIST#3: BPUT#3;12: CLOSE#3
150 DEFine PROCedure sigma
160   BPUT#3;27,75,8,0,28,42,42,42,78,0,0,0
170 END DEFine sigma
180 DEFine PROCedure pot2
190   BPUT#3;27,75,8,0,0,184,168,232,0,0,0,0
200 END DEFine pot2
```

2. HER UDNYTTES, AT CPA80S FAKTISK HAR sigma OG superscript:

sigma UDSKRIVES MED:
SKIFT TIL GRÆSK TEGNSÆT, CHR$(142) = sigma, RETUR TIL DANSK TEGNSÆT.

2. potens UDSKRIVES MED superscript:
BRUG superscript, CHR$(50) = 2, BRUG NORMALSKRIFT.

TRÆKSPÆNDING, σ= 22 MPa/m² FOR FORSØG NR. 1.

```
100 OPEN#3,ser1: BAUD 4800: TRA 1: BPUT#3;27,108,5
110 PRINT#3;'TRÆKSPÆNDING, ';: sigma :PRINT#3;'= 22 MPa/m';:pot2
120 PRINT#3;' FOR FORSØG NR. 1.'
130 PRINT#3;FILL$(CHR$(174),44)
140 BPUT#3;10,10,10: LIST#3: BPUT#3;12: CLOSE#3
150 DEFine PROCedure sigma
160   BPUT#3;27,82,21,242,27,82,14
170 END DEFine sigma
180 DEFine PROCedure pot2
190   BPUT#3;27,83,0,50,27,84
200 END DEFine pot2
```

SÅVIDT JEG KAN SE, KAN METODE 2 BRUGES TIL "QUILL", VED HJÆLP AF "TRANSLATE" I "PRINTER-DRIVEREN", MENS METODE 1 IKKE UMIDDELBART ER BRUGBAR, FORDI DER ER FOR MANGE KODER I DEFINITIONERNE (?). HVIS QL IKKE HAR TEGNET, MÅ ET ANDET TEGN VIKARIERE !

3. QL-PRINTEREN HAR IKKE sigma, MEN superscript.
HER DEFINERES BEGGE TEGN SOM I EKSEMPEL 1, MEN I
NEAR LETTER QUALITY. DERFOR 8 x 2 PIXELKOLONNER.

TRÆKSPÆNDING, ≦= 22 MPa/m² FOR FORSØG NR. 1.
ΘΘΘ

WHEN ERROR initialiseres her ved PROceduren "errorcheck" fra linie 100.

Demonstrationen her er INPUT af et tal. Ethvert tal, heltal eller decimaltal anerkendes, også i exponentiel notation. "talput" og "tput" (pag. 54 og 80) anerkender ikke '+' eller exponentionel notation.

I linie 130 findes INPUT (en streng), og variablen "tal" sættes lig strengen. Her burde der altid komme en fejl, hvis strengen ikke kan omdannes til et tal.

DESVÆRRE HAR QDOS EN GROV FEJL: BARE STRENGENS FØRSTE KARAKTERER KAN OMDANNES TIL ET TAL, SKER DETTE, MENS RESTEN AF STRENGEN BLOT SMIDES VÆK !

"135xy" bliver til 135. "xy135" er derimode en fejl. "123." er lig med "123" og "123.0" og alle korrekte. "123.0.8" er en fejl.

Fejl medfører at WHEN ERROR træder i funktion.

IF ERLIN er nødvendig, da en fejl jo også kan skyldes en direkte kommando, der naturligvis ikke har en fejl-linie.

IF ERNUM kan da vist udelades. En fejl uden fejlkode er en af de væmmelige fejl, hvor QL forsvinder ud i det blå...

a$ er blot en streng, der fjerner det tidligere udskrevne. Her er eet af de praktiske problemer, der bliver yderligere fremhævet i linie 250: Der er ingen given sammenhæng mellem fejllinien og de linier, hvori sletning / RETRY skal foretages.

Teoretisk kunne man indføre SELECT ON ERLIN og indføre en stribe betingelser. Dette er kun muligt i et absolut færdigt program, da RETRY ikke RENUMmeres. I øvrigt skal det meste tal-INPUT checkes, fx for minimum og maximum, så hvad godt gør WHEN ERROR uden en hel del betingelser ? At 123xy som nævnt er et tal, gør ikke det hele ret meget nemmere. WHEN ERROR bør analyseres nærmere !

Endvidere: ERROR-flag RESETtes ikke. TOOLKIT2 påstår, at det gør en MRUN. Jeg har prøvet med en i øvrigt ligegyldig linie, uden at det havde nogen effekt !

BEMÆRK, at når WHEN ERROR først er startet, er den aktiv overalt: Alle fejl fanges, program - kommandoer såvel som direkte kommandoer. Der må i hvert fald indføres en ERLIN = og måske en ERNUM = REMAINDER med fejl-udskrift og STOP.

KONKLUSION: "tput" er sikrere, indtil WHEN ERROR er blevet analyseret nærmere. "tput" kan dog kun, ret så omstændelgt, udbygges til at dække exponentiel notation.

Error messages and codes

In almost all the QDOS trap and utility routines the success of the operation is reflected in register D0 and the status register. A range of codes between 0 and -21 can be returned. Only zero means success. The other error codes are shown below.

-1 ERR.NC means not complete–the routine could not do what it was asked for some reason. For example, an attempt to allocate resident procedure space will return with this error code if the transient program area is not empty.

-2 ERR.NJ means that the job ID was invalid or a job did not exist.

-3 ERR.OM means out of memory. This can occur during a channel open as each channel requires a block of memory. If there is no room left then this error will occur. It will also occur in SuperBasic if you attempt to open a channel with too high a channel number–#2581 for instance. The reason for this is that each Basic channel takes up $28 bytes starting from a particular address, and the channel number must be multiplied by $28 to point to the correct channel block. Obviously the use of large channel numbers uses up more memory.

-4 ERR.OR means out of range. This error occurs when an attempt is made to print (using AT or cursor) outside a window.

-5 ERR.BO means buffer overflow. When reading input from a device a buffer is allocated to hold characters received. If this fills up before an operation is complete then this error will occur. Try it from Basic by typing 'INPUT var$' and holding down a key for a while after pressing ENTER.

-6 ERR.NO means not open. An attempt has been made to read from, write to or close a channel that does not exist.

-7 ERR.NF means not found. This can occur if you try to load a non-existent file or if an unrecognised device name is used in an open channel call.

-8 ERR.EX means file exists. If an attempt is made to open a new exclusive file that already exists this error results.

-9 ERR.IU means in use. If one attempts to open a file or device that has already been opened exclusively then you'll get this error.

-10 ERR.EF means end of file. Obvious enough.

-11 ERR.DF means drive full. Again, rather obvious.

-12 ERR.BN means bad name. This crops up all the time. It can happen if you use a device name that isn't quite correct, such as CON__100x200q50x40__10 (the q is incorrect), or if you call a non-existent procedure or function. Calling a procedure as if it were a function and vice versa can also cause this error.

-13 ERR.TE means transmission error. This error occurs if something goes wrong in an RS-232 operation.

-14 ERR.FF means format failed. Either something is wrong with a microdrive cartridge or your QL's up the creek.

-15 ERR.BP means bad parameter. This error crops up more than any other and can stem from such things as missing out parameters to procedure and function calls, giving invalid parameters or attempting an operation which is not applicable, such as trying to set a file pointer on a network device.

-16 ERR.FE means file error. Probably the first error message you ever saw on the QL–Bad or changed medium. A pain with seemingly no solution.

-17 ERR.XP means error in expression. Using an expression containing a non-existent variable or calls to non-existent functions.

-18 ERR.OV means overflow. This happens whenever the QL cannot cope with the numbers. Division by zero, integers over 32767 or under -32768, that sort of thing.

-19 ERR.NI means not implemented. The story of the QL! This error often crops up if you do things in procedures that could corrupt the stack such as file merges.

-20 ERR.RO means read only. An attempt has been made to write to a read only file.

-21 ERR.BL means bad line. Essentially a syntax error. This is the error that causes a line being entered or edited to be thrown back at you.