

PROGRAMVARA FÖR TERMINAL MED ZOOM

BENGT OHLSSON

INSTITUTIONEN FÖR REGLERTEKNIK  
LUNDS TEKNISKA HÖGSKOLA  
JULI 1981

<b>LUND INSTITUTE OF TECHNOLOGY</b> DEPARTMENT OF AUTOMATIC CONTROL Box 725 S 220 07 Lund 7 Sweden	Document name	
	MASTER THESIS	
	Date of issue	
	July 1981	
Author(s) Bengt Ohlsson	Document number	
	CODEN:LUTFD2/(TFRT-5258)/1-130/(1981)	
	Supervisor	
	Hilding Elmqvist	
	Sponsoring organization	
	Swedish Board for Technical Development	
	Contract: STU 80-3962	
Title and subtitle		
Programvara för terminal med ZOOM		
(Software for Terminal with ZOOM)		
Abstract		
<p>Software for an alphanumeric terminal with zoom is described. The terminal contains a microcomputer and a video unit capable of presenting characters of different size.</p> <p>It is possible to zoom and scroll the picture by using a joy stick. About one hundred lines can be shown on the screen. This means that information such as indentations and new sections can be used for example when locating a piece of text. When the text is found it can be zoomed to any size.</p> <p>The terminal also allows information zooming. It is based on the fact that the text lines are often structured as a tree. When for example editing a Pascal program, the tree structure is represented as indentations. Information zooming means that only the lines above a certain level in the tree are shown on the screen.</p>		
Key words		
Classification system and/or index terms (if any)		
Supplementary bibliographical information		
ISSN and key title		ISBN
Language	Number of pages	Recipient's notes
Swedish	130	
Security classification		

DOKUMENTATABLAD RT 3/81

Distribution: The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

Programvara för terminal med ZOOM

Bengt Ohlsson

Examensarbete

Handledare: Hilding Elmqvist

Institutionen för Reglerteknik

Tekniska Högskolan i Lund

,

INNEHÅLLSFÖRTECKNING

Kap 1: Inledning	2
Kap 2: Hårdvarubeskrivning	6
Kap 3: Inledande programbeskrivning	7
Kap 4: Bildzoomning	13
Kap 5: Informationszoomning	17
Kap 6: Terminal kommando	25
Referenser	36
Appendix A: Programlistning	A1
Appendix B: Listning av alla fonter	B1
Appendix C: Hjälpprogram vid teckengenereringen C1	

## 1 INLEDNING

Denna rapport behandlar mjukvaran till en mikroprocessor i en dator terminal. Hårdvaruenheten har byggts vid ett tidigare examensarbete, referens 1.

Terminalen är främst tänkt att användas vid programutveckling och har för detta speciella hjälpmedel som bildzoomning och informationszoomning.

### Bildzoomning

Bildzoomning innebär att man med hjälp av olika stora teckenuppsättningar, fonter, kan zooma alfanumerisk text. Detta kan användas vid sökning av ett visst textavsnitt i en stor textmassa. Tecknen förminskas och man får plats med fler rader på skärmen. För minsta fonten är höjden för en rad endast fyra linjer (se appendix B), och då får ca 100 rader plats på skärmen. Med så många rader kan man inte läsa orden i texten. Iden är att man med hjälp av textens struktur skall kunna lokalisera ett önskat textavsnitt. Detta kan sedan förstöras till önskad storlek. Vid editering av program kan man t.ex. ha nytta av radernas indentering som återspeglar begreppen inom strukturerad programmering.

Man kan även specificera en minsta fontstorlek för en viss textrad. Detta kan användas för att göra sökningen beskriven ovan lättare. Vid förminskningen av texten låts vissa karakteristiska rader förbli stora. Dessa kommer därför att framträda mycket tydligt och det blir lätt för användaren att orientera sig i texten. Se fig 1.1-1.3.

### Informationszoomning

Informationszoomning är baserat på att textraderna är strukturerade som ett träd. För programspråk som Pascal där strukturerad programmering används kan trädstrukturen direkt fås genom indenteringen. Rader som är lika indenterade ligger på samma nivå i trädet. Sedan innebär informationszoomning att man specificerar hur djupt i trädet som informationen ska visas på skärmen. Detta medför att man t.ex. i ett stort program kan visa endast procedurhuvudena på skärmen. När man sedan hittat rätt procedur kan hela proceduren fås fram på skärmen genom att informationszooma till ett större djup i trädet. Se fig 1.4-1.6.

### Kontinuitet

En viktig princip vid båda formerna av zoomning är att zoomningen görs kontinuerligt. Härmed blir det naturligt att använda styrspakar för att zooma texten på skärmen. För bildzoomningen fås kontinuiteten genom att succesivt lägga in extra blanka linjer mellan raderna. Härmed kan det först göras plats innan en ny font sätts in och övergången blir mjuk.

I informationszoomningen löses kontinuiteten genom att en ny nivå succesivt sätts in på skärmen. Se fig 1.4-1.6. Först görs det plats för de nya raderna genom att höjden för deras listhuvud (streckade raderna i fig 1.4-1.6) ökas. Raderna sätts sedan in med minsta fontstorleken. Fortsätter man att zooma kommer sedan fontstorleken för dessa rader att öka tills de är lika stora som omgivningen. Sedan kan proceduren upprepas för nästa nivå.

### Bakgrund

Bildzoomningen bygger på att videogeneratoren kan hantera olika stora fonter. Informationszoomningen däremot är helt mjukvaruimplementerad och bygger på att raderna ligger lagrade som en länkad lista i textminnet. Detta medför att rader snabbt kan sättas in eller tas bort. (Se systembeskrivning i referens 1).

### Programutveckling

All programmering har gjorts i Pascal. Högnivåspråk har använts för att få ett bättre strukturerat program. Programutvecklingen har gjorts på ett microNOVA system vid avdelningen för Digital och Datorteknik, Lunds Tekniska Högskola.

```

(* Program TEVE, Module TVBODY *)
Procedure Insertafter(listent,this:lineptr);
begin
  with this do
  begin
    next:=listent^.next;
    previous:=next^.previous;
    next^.previous:=this;
  end (with);
  listent^.next:=this;
end (Insertafter);

Procedure Takeout(this:lineptr);
begin
  with this do
  begin
    next:=next^.next;
    previous:=previous^.previous;
  end (with);
end (Takeout);

Procedure Printall;
begin
  Printall:=listent;
  while Printall <> nil do
  begin
    Print(Printall);
    Printall:=Printall^.next;
  end (while);
end (Printall);

```

Fig 1.1 Minsta fontstorleken. Bilden innehåller ca 100 rader. Märk att procedurhuvudena, som är specificerade för relativ minfont, är större än övriga texten.

```

(* Program TEVE, Module TVBODY *)
Procedure Insertafter(listent,this:lineptr);
begin
  with this do
  begin
    next:=listent^.next;
    previous:=next^.previous;
    next^.previous:=this;
  end (with);
  listent^.next:=this;
end (Insertafter);

Procedure Takeout(this:lineptr);
begin
  with this do
  begin
    next:=next^.next;
    previous:=previous^.previous;
  end (with);
end (Takeout);

Procedure Printall;
begin
  Printall:=listent;
  while Printall <> nil do
  begin
    Print(Printall);
    Printall:=Printall^.next;
  end (while);
end (Printall);

```

Fig 1.2 Här har texten zoomats upp. Vanliga texten har fontstorlek sju.

```

(* Program TEVE, Module TVBODY *)
Procedure Insertafter(listent,this:li);
begin
  with this do
  begin
    next:=listent^.next;
    previous:=next^.previous;
    next^.previous:=this;
  end (with);
  listent^.next:=this;
end (Insertafter);

Procedure Takeout(this:lineptr);
begin

```

Fig 1.3 Nu har procedurhuvudena uppnått största fontstorleken.

```

(* Program TEVE, Module TVBODY *)
Procedure Insertafter(listent,this:li
Procedure Takeout(this:lineptr);
Procedure Marklast;
Procedure Emptyline;
Procedure Print;
Procedure Printlines;

```

Fig 1.4 På denna bild syns endast första nivån i trädstrukturen.

```

(* Program TEVE, Module TVBODY *)
Procedure Insertafter(listent,this:li
  begin
  with this do
  -----
  listent.next:=this
  end (Insertafter);
Procedure Takeout(this:lineptr);
  begin
  with this do
  -----
  end (Takeout);
Procedure Marklast;

```

Fig 1.5 Här syns två nivåer. Listhuvudena, streckade linjerna i bilden, har gjorts högre för att bereda plats för nästa nivå.

```

(* Program TEVE, Module TVBODY *)
Procedure Insertafter(listent,this:li
  begin
  with this do
  -----
  listent.next:=this
  end (Insertafter);
Procedure Takeout(this:lineptr);
  begin
  with this do
  -----
  end (Takeout);
Procedure Marklast;

```

Fig 1.6 Nu har sista nivån kommit med. Denna är specialbehandlad och har här den minsta fontstorleken.



## 2 HÄRDVARUBESKRIVNING

Terminalen innehåller CPU-enhet, videogenerator, ett för dessa gemensamt textminne, samt ett teckenminne. Det senare läses av videogeneratoren och innehåller information om alla fonterna. Se fig 2.1.

Via värddatorn och/eller styrspaker för zoomning och scrolling kan operatören ge kommando till CPU-enheten. Styrspakarna, tre stycken är kopplade till var sin analog till digital omvandlare.

I textminnet lagras en textrad som en nod i en dubbelt länkad lista. För varje nod finns ett nodhuvud med information till bl.a. videogeneratoren. Denna information beräknas av CPU:n, som även ska sköta kommunikationen med värddator, tangentbord och styrspakar.

Det finns även olika register för att ange första rad på skärmen och position för cursorn. Dessa register laddas av CPU:n.

För vidare information om hårdvaran hänvisas till referens 1.

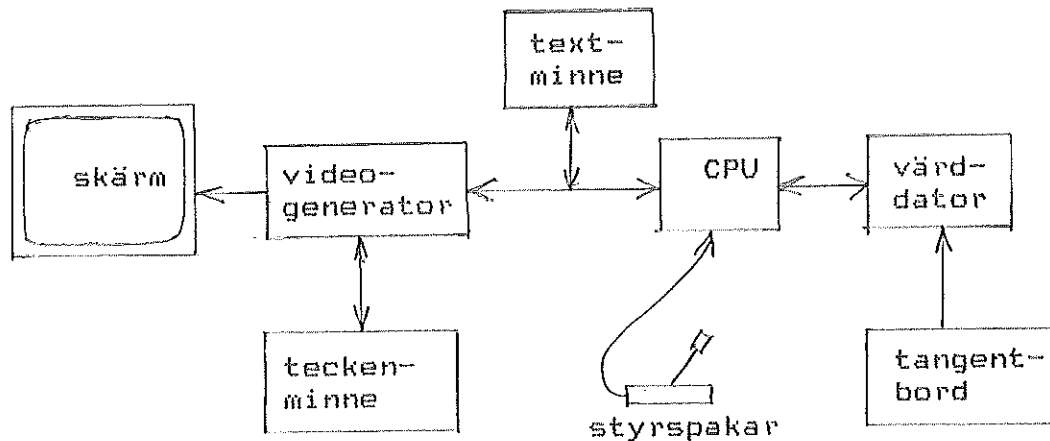


Fig 2.1 Systemutformning.

### 3 INLEDANDE PROGRAMBESKRIVNING

Programmet är skrivet i Pascal. Högnivåspråk har använts för att få ett strukturerat program.

Som nämnts i hårdvarubeskrivningen så är textminnet organiserat som en dubbelt länkad lista, där var nod utgörs av en textrad och tillhörande information som t.ex. fontstorlek för raden. Detta gör det lämpligt att i pascal definiera en rad som en record.

#### Raddefinition

```
Headtype= (antal,font,minfont,slask,
           totheight,firstch,firstkol);

rowpointer= ^row;

row= record
  vfp: integer;
  status: integer;
  rowhead: array[headtype] of char;
  string: array[1..maxchar] of char;
  rownr, level: integer;
  subfp,subbp,headfp,headbp,hover: rowpointer;
  fp,bp,fpo,bpo: rowpointer;
end;
```

För att komma åt enskilda bytes som videogeneratorn läser var det nödvändigt att deklarera delar av nodhuvudet som en teckensträng. Detta bortser vi ifrån i fortsättningen och refererar direkt till de olika värdena för headtype. Dvs vi skriver t.ex. endast font när rowhead[font] menas.

#### Beskrivning av variabler tillhörande en rad

Nedan anges betydelsen för de olika variablerna tillhörande en rad. Variabler som läses av videogeneratorn är understrukna.

vfp: framåtpekare till nästa rad. Denna är identisk med fp (se nedan) men måste finnas då videogeneratorn och mikroprocessorn läser två bytes (= en pekare) i olika ordning.

status: anger typ av rad. Följande värden används:

- 0: normal rad.
- 1: märkta tecken (se string) i raden ska strykas under.
- 2: absolut minfont.
- 3: absolut minfont och understrykning (se 1!).
- 4: relativ minfont.
- 5: relativ minfont och understrykning (se 1!).
- 6: raderna textstart, textend eller slut.
- 7: listhuvud, används i trädstrukturen (se kap 5).
- 10-17: specialmarkering för rader i understa synliga nivå. Denna markering innebär att statusvärdet har ökats med tio, dvs information om gamla värdet finns kvar. Användningen beskrivs i kap 5.

antal: antal tecken som videogeneratorn ska visa.

font: aktuell fontstorlek för raden. Åttonde biten anger för videogeneratorn om märkta tecknen i raden ska strykas under eller visas med dubbel intensitet.

minfont: kan vara absolut eller relativ vilket anges av status. Absolut betyder att font för raden aldrig får bli mindre än minfont. Relativ betyder att font för raden ska vara minfont större än globalt.

slask: användes ej.

totheight: totala höjden för raden. Ska vara större än eller lika med fonthöjden. För översta raden i bilden för totheight vara mindre än fonthöjden och då visas undre delen av tecknen på raden.

firstch: anger första tecken som ska visas på raden. Ändras ej av programmet. Raderna visas alltid från början.

firstkol: anger antal kolumner som ska visas av firstch. Ändras ej av programmet. Hela första tecknet visas. (För programmering av firstkol se referens 1).

string: en vektor med tecknen som ska visas. Den åttonde biten för ett tecken anger om tecknet ska visas med dubbel intensitet respektive understruket. Se font och status ovan.

rownr, level, subfp, subbp, headfp, headbp, hover, fpo, bpo: användningen av dessa variabler beskrivs i kapitlet om informationszoomning.

fp: framåtpekare till nästa rad.

bp: bakåtpekare till föregående rad.

## Speciella variabler

Två variabler är deklarerade som en variant record. Shift användes i funktionen pswap för att beräkna vfp till videogeneratoren. Dummy användes för att kunna använda samma variabel både som pekare och heltal vid initialiseringen av textminnet.

## Procedurer

Procedurerna i programmet kan grovt indelas i fem grupper

- 1: externa assemblerskrivna procedurer. Användes huvudsakligen för laddning av olika register till videogeneratoren. Se appendix A.
- 2: cursorförflyttning.
- 3: editering, t.ex sätta in eller ta bort rader.
- 4: procedurer som användes vid bildzoomning.
- 5: procedurer som användes vid informationszoomning.

Vidare finns även några procedurer för initialisering av variabler vid start av programmet och för poolhanteringen. Poolen fungerar som en stack av blanka rader. Nya rader som ska sättas in hämtas från poolen och rader som tas bort läggs i poolen.

## Huvudprogram

Huvudprogrammet består av en initialisering vid start av programmet och sedan följer en oändlig loop. Här sker kommunikationen med omgivningen; avläsning av spakar för zoomning och scrollning samt kommandoavkodning. Detta tillgår så att tecken från tangentbordet endast mottages och sänds vidare till värddatorn. För tecken som kommer från värddatorn kontrolleras om det är något kommando; annars skrivs de ut på skärmen på cursorns plats och cursorn flyttas ett steg till höger. Cursorn stannar dock vid högermarginalen. Ett inkommet kommando avkodas och utförs om det känns igen av programmet; annars ignoreras det

Två logiska funktioner används vid kommunikationen; host och keyboard. Dessa får värdet true så länge det finns tecken att läsa i inbuffertarna från värddator respektive tangentbord. Mottagningen av enskilda tecken görs av assemblerskrivna avbrottsrutiner och tecknen läggs då i respektive buffert för värddatorn och tangentbordet. Härigenom missas inte några tecken. Sedan används host och keyboard för att testa om det finns några tecken i buffertarna. Se fig 3.1.

Normalt läses först alla tecken som finns i inbufferten från värddatorn och sedan alla tecken från tangentbordet. Efter detta avläses zoomnings och scrollnings spakarna; samt kontrolleras om någon zoomning eller scrollning ska göras.

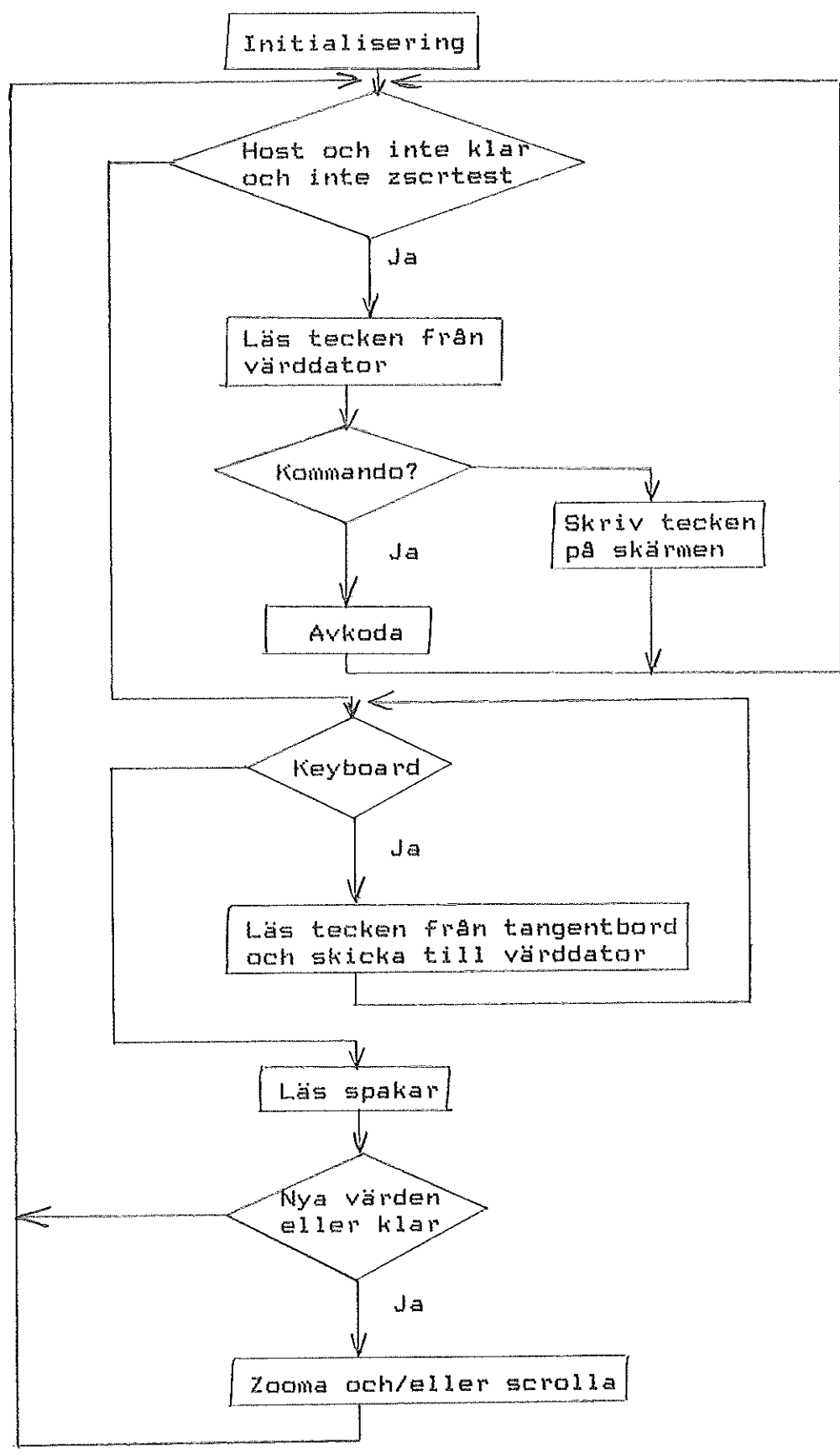


Fig 3.1 Flödesschema för huvudprogrammet.

De logiska variablerna klar och zscrtest används för att kunna lämna loopen med kommandoavkodningen innan inbufferten från värddatorn är tom. Se fig 3.1. Klar har den högre prioritet. Om zscrtest får värdet true avläses spakarna och det kontrolleras om någon zoomning eller scrollning ska göras. Om klar får värdet true kommer alltid procedurerna för zoomningen och scrollningen att anropas. Variablerna klar och zscrtest sätts efter vissa kommando om man vill att bilden direkt ska räknas om; t.ex. när rader har satts in eller tagits bort i texten.

Spakarna som det refereras till är som tidigare nämnts tre stycken. Dessa används för scrollning, bildzoomning och informationszoomning. Värdet från en spak finns tillgängligt i en inport som avläses med funktionen getport.

### Programstart

Vid start av programmet ges en del variabler startvärden, däribland de vektorer som anger bredd och höjd för de olika fonterna. Här initialiseras även poolen och de rader som användes för att markera början och slutet på texten, (textstart och textend). Det går ej att sätta cursorn på dessa rader. Från början finns en rad mellan textstart och textend. Sedan hämtas efterhand nya rader från poolen då värddatorn skickar text till terminalen. För att inte behöva fylla skärmen med blanka rader användes en blank rad (slut) efter textend som pekar på sig själv. Se fig 3.2.

### Fonterna

Det finns sexton olika fonter. I appendix B finns en tabell på hur stora de är samt en listning hur de ser ut. I tabellen ser vi att höjden för font åtta är mindre än för font sju. Detta beror på att font sju visas med interlace, se kap 4, och font åtta visas på normalt sätt, dvs endast varannan linje på skärmen används. Detta medför att totala antalet linjer på skärmen som font åtta upptar är fjorton jämfört med elva för font sju.

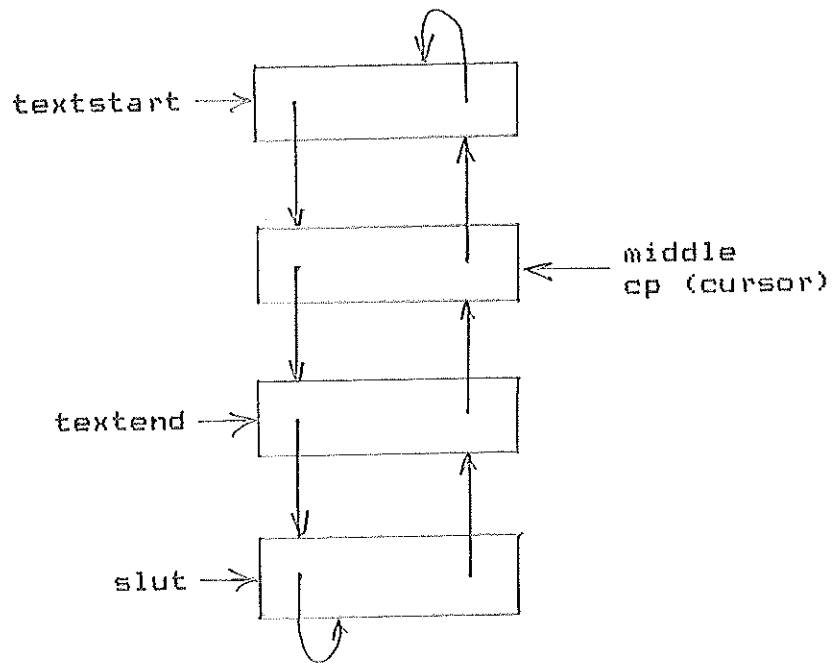


Fig 3.2 Struktur efter start av programmet.

#### 4 BILDZOOMNING

I förra kapitlet såg vi att videogeneratoren läser information om fontstorlek och totalhöjd för varje rad. Det är alltså dessa två variabler som måste ändras när man vill zooma texten på skärmen. Det finns också ett register, kontroll-reg, som kan användas för att skifta från och till interlace mod för videogeneratoren. I programmet användes en motsvarande variabel ctrlreg vars värde laddas i kontroll registret.

#### Interlace

Normalt användes endast varannan linje på bildskärmen, vilket ger att bilden ändras 50 ggr per sekund. För att få bättre upplösning kan man använda interlace. Detta innebär att man använder alla linjerna men visar jämna varannan gång och udda varannan gång. I interlace kommer bilden att ändras endast 25 ggr per sekund, vilket gör att texten flimrar lite. Flimret är priset man får betala för den bättre upplösningen.

För font noll till sju användes interlace och för font åtta till femton visas bilden på vanligt sätt.

#### Kontinuerlig zoomning

Om man vid zoomningen ändrar font och totheight till samma värde för alla raderna så kommer bildens informationsinnehåll att ändras språngvis. Men man vill ha en kontinuerlig ändring av bilden. Detta kan lösas genom att succesivt lägga in extra blanka linjer mellan raderna. Dessa extra linjer fås enkelt genom att för de berörda raderna ladda totheight med fonthöjden för närmaste större font. När man sedan t.ex. zoomar upp så kommer inte font att ändras förrän den nya storleken får plats och man kommer att få en mjuk övergång.

Zoomningen utgår från mitten. Dvs raden mitt på skärmen står stilla. Extra blanka linjer läggs in från mitten och ut mot över- och underkanten. Vidare så testas inte om cursorn vid zoomning eller scrollning försvinner från skärmen. Men om man skriver någonting och cursorn inte syns så kommer bilden att scrollas så att raden med cursorn kommer i mitten.

#### Minsta fontstorlek för en rad

Eftersom videogeneratoren läser informationen om font och totheight för varje rad kan en viss rad göras större än omgivningen. Detta görs genom att variabeln status för raden laddas med värdet motsvarande absolut eller relativ minfont



och variabeln minfont laddas med önskat värde. Vid absolut minfont kommer vid zoomning font att laddas med det större av nya fontstorleken eller variabeln minfont.

```
font:= max(nya fontstorleken, minfont)
```

Vid relativ minfont kommer vid zoomning font att laddas med nya font storleken plus variabeln minfont, dock måste font vara mindre än eller lika med femton.

```
font:= min(nya fontstorleken + minfont, 15)
```

Minfont kan t.ex. användas för att alltid visa procedurhuvud med stor text.

Användningen av interlace ger problem när en rad specificeras för någon minfont. Antag att bilden visas med globala fontstorleken sju, se tabell appendix B, och en viss rad har specificerats för minfont åtta. Resultatet blir här att den utvalda raden blir mindre än omgivningen. För att lösa detta måste man ha en speciell uppsättning fonter för storlekarna åtta till och med elva. Dessa ska då användas för rader som ska visas med fontstorlekarna åtta till och med elva när bilden visas med interlace, dvs den globala fontstorleken är mindre än eller lika med sju.

### Implementering

Det är sju procedurer som har hand om bildzoomningen. I huvudprogrammet avläses den spak som anger zoomningsgrad. Detta värde läggs i variabeln zoomlev. I proceduren newzvar beräknas sedan nya värden för de variabler som används vid zoomningen, speciellt då newfont som anger nya globala fontstorleken och blanklines som anger på hur stor del av skärmen som blanka linjer ska sättas in. Blanklines fungerar så att värdet anger avståndet från mitten och ut mot kanterna där extera blanka linjer ska sättas in, dvs bestämmer ett område i antal linjer på skärmen. Se fig 4.1.

### Ny startadress

Sedan beräknas en ny startadress i proceduren newstartadr. Härvid utgår man ifrån mitten. En pekare, middle, pekar alltid på raden mitt i skärmen. Variabeln midlines anger hur många linjer hos middle som ligger under mitten. Proceduren goback används sedan för att från middle gå upp rad för rad. För varje rad summeras antalet linjer som denna rad kommer att uppta och när summan överstiger antalet linjer för halva skärmen har man hittat startraden. Totheight justeras sedan för denna rad så att summan av antalet linjer blir rätt.

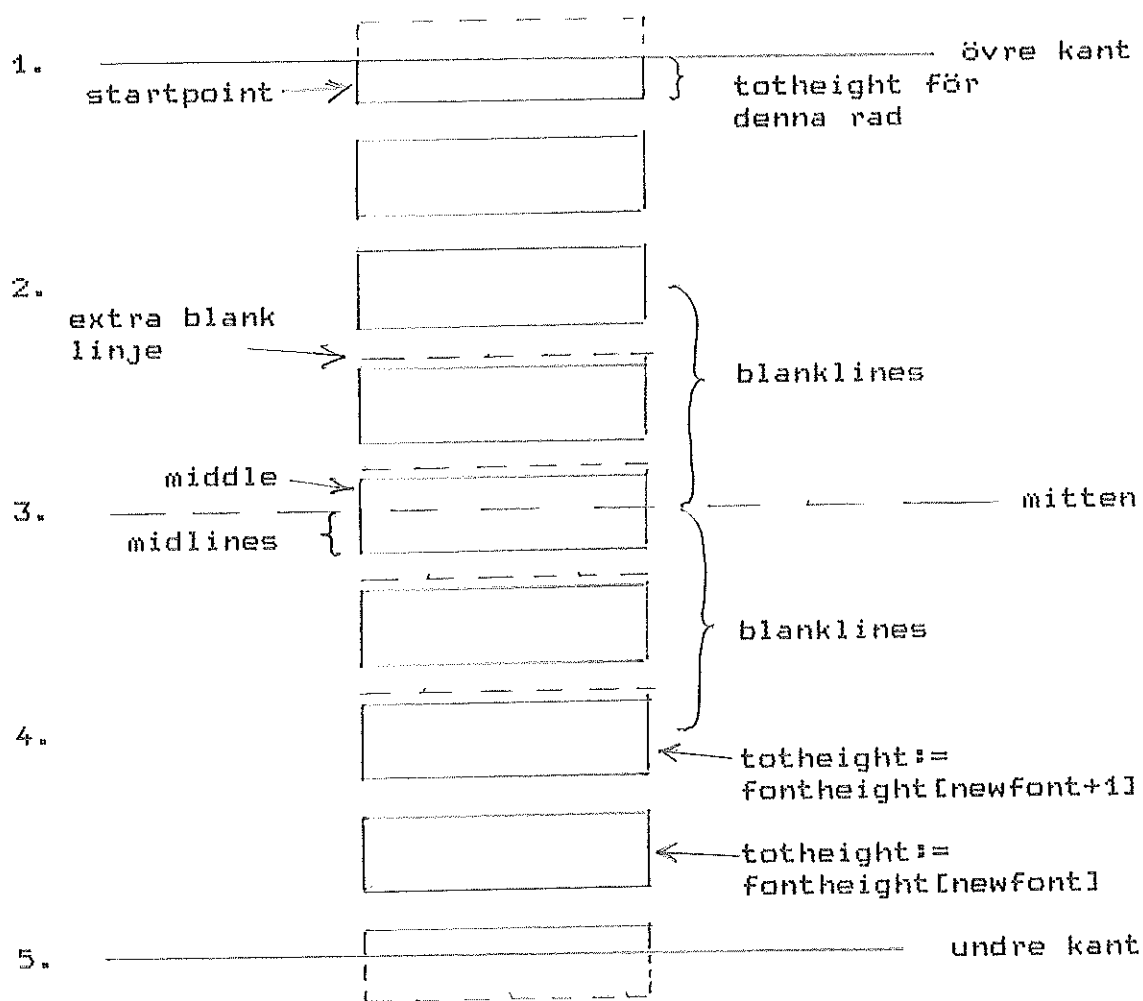


Fig 4.1 Beskrivning av variabler vid bildzoomning.

gången används fonthöjden för fontstorleken newfont, motsvarande området mellan ett och två i fig 4.1.

### Scrollning

Om man vill scrolla bilden så görs detta samtidigt med zoomningen. När zoomningsspaken läses av så läses även spaken för scrollning av. Värdet skalas och lagras i variabeln vertscr. Denna anger hur många linjer som ska scrollas upp eller ner. I newstartadr ändras sedan midlines med värdet hos vertscr. Detta gör att man får en mjuk scrollning med varierbar hastighet.

### Zoomning av bilden

När den nya startadressen är klar återstår det att ändra font och totheight för alla rader som kommer att synas på skärmen. Detta görs av proceduren changefont som börjar med att ladda startadress och kontrollregistren med de nya värdena. Sedan anropas proceduren newhead som för varje rad ändrar font och totheight samt summerar antalet linjer som raden upptar. Det sista görs för att endast behöva gå igenom de rader som syns på skärmen.

Newhead anropas tre gånger med fonthöjden (inchgt och height) motsvarande newfont, newfont + 1 och newfont. Första gången motsvarar området mellan ett och två i fig 4.1. Andra gången läggs det in extra blanka linjer och detta motsvarar området mellan två och fyra. Sista gången motsvarar området mellan fyra och fem. Sedan är nya bilden klar.

I goback och newhead används en case sats för att testa på status för varje rad. Detta ger ett enkelt och effektivt sätt att kontrollera om t. ex. newfont ska testas mot minfont. Alla olika slag av rader har ett unikt statusvärde så att endast ett test behöver göras.

### Behandling av lägsta nivån på skärmen

Procedurerna subgoback och subnewhead används för den lägsta nivån i trädstrukturen som syns på skärmen. Deras uppgift är densamma som för goback respektive newhead. Deras användning kommer ytterligare att förklaras i nästa kapitel. Variabeln testklar används för att kontrollera om man befinner sig i den lägsta synliga nivån när goback eller newhead anropas. I så fall försätter anropet till subgoback eller subnewhead.

### Zoomningsnivåer

När spaken för bildzoomning avläses fås ett värde mellan 0 och 255 inklusive gränserna. Detta värde dividerat med sexton ger sedan den nya fontstorleken. Blanklines beräknas sedan utgående från resten vid divisionen med sexton. Blanklines kan alltså anta sexton olika värden för varje fontstorlek.

## 5 INFORMATIONSZOOMNING

Som nämnts i inledningen innebär informationszoomning att vissa rader inte alltid visas på skärmen. Detta löses genom att bygga upp texten i en trädstruktur. Sedan kan man specificera att rader som ligger under ett visst djup i trädet inte ska visas.

### Listhuvud

Ett underträd definieras av ett listhuvud. Ett nytt listhuvud kan sättas in överallt i texten. Från ett listhuvud utgår sedan speciella pekare till första och sista raden i dess lista (den närmast underordnade nivån i underträdet). Även dessa rader kan i sin tur vara nya listhuvuden. Ett listhuvud markeras genom statusvärdet sju.

### Beskrivning av radvariabler för trädstrukturen

I kap 3 beskrevs de olika variablerna tillhörande en rad utom de för informationszoomningen. Nedan följer en beskrivning för dessa variabler. Ingen av dem läses av videogeneratoren. Se raddefinitionen kap 3.

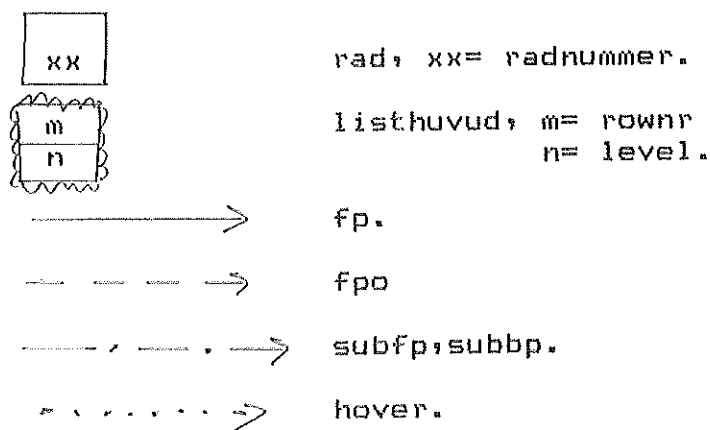
- rownr: endast listhuvud, anger antalet rader inklusive listhuvud i aktuella listhuvudets lista.
- level: endast listhuvud, anger vilken nivå som listhuvudet ligger på
- subfp: endast listhuvud, pekare som pekar på första raden i listhuvudets lista.
- subbp: endast listhuvud, pekare som pekar på sista raden i listhuvudets lista.
- headfp: endast listhuvud, pekare som pekar på nästföljande listhuvud.
- headbp: endast listhuvud, pekare som pekar på föregående listhuvud.
- hover: en pekare från alla rader inkl listhuvud till listhuvudet för den aktuella nivån.
- fpo: framåtpekare inom en lista. Pekar på slut för sista raden i listan.
- bpo: bakåtpekare inom en lista. Pekar på slut för första raden i listan.

Slutligen ska påpekas att pekarna fp och bp ger den struktur som syns på skärmen.

Pekarna headfp och headbp gör att alla listhuvud bildar en egen lista. På samma sätt som textstart och textend används för att indikera början och slut för texten används här en speciell rad headstart, som utgör början och slut för listan av listhuvud. Headstart tjänar även som listhuvud för översta nivån. Detta gör att specialfall undviks.

### Exempel

För att något klargöra användningen av alla pekare följer nu ett exempel som visar hur strukturen blir för ett kort program. Bp och bpo ritas ej ut då de följer samma struktur som fp och fpo. Följande symboler används för att rita strukturen.



Vi utgår från följande korta program.

```

11  program exempel;
12  var i: integer; ch: char;
13  begin
14  for i:= 1 to 10 do
21    begin
31      repeat
32        read(ch);
33        until ch <> ' ';
22        write(ch);
23      end;
15  end.

```



Om endast översta nivån ska visas fås följande struktur.

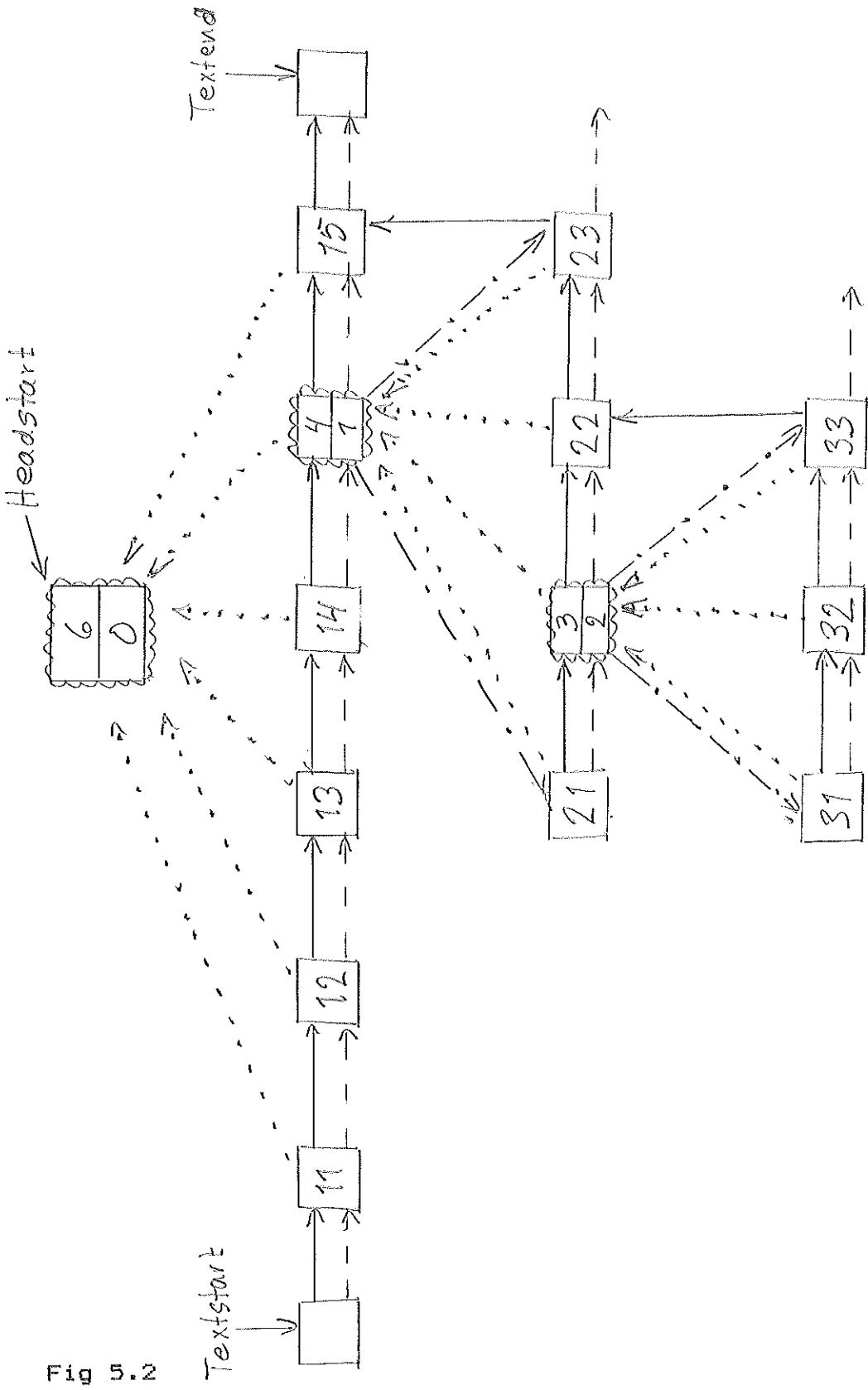


Fig 5.2

Slutligen fås följande struktur för listhuvudena.  
(headfp och headbp).

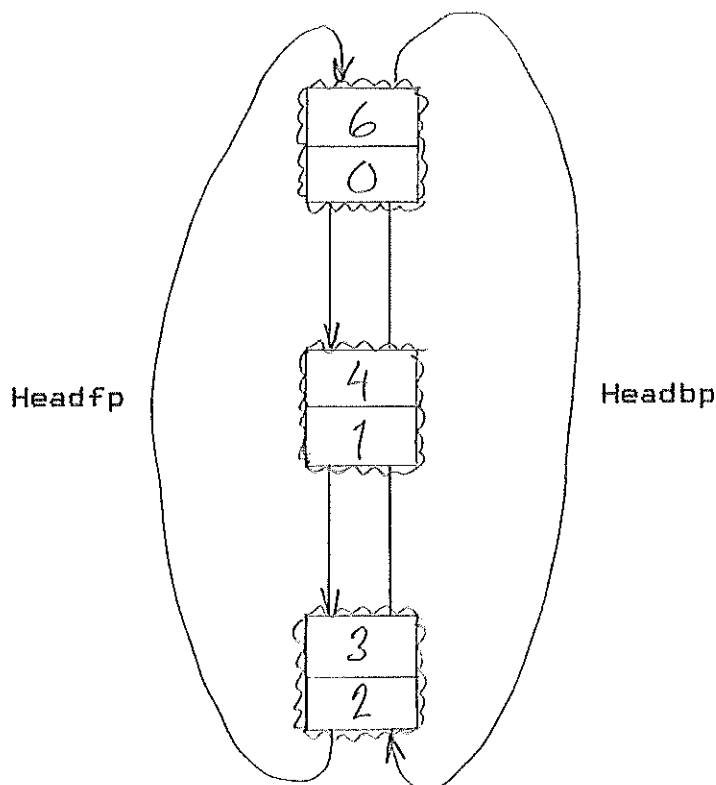


Fig 5.3

Som framgår av exemplet är det alltså endast fp (vfp) och bp som ska ändras när zoomningsdjupet ändras. Då alla listhuvud bildar en egen lista behöver endast dessa gås igenom vid en ändring av zoomningsdjupet. För varje listhuvud sätts sedan fp och bp för de rader som fpo, bpo, subfp, subbp pekar på beroende på värdet hos variabeln level.

Ett listhuvud syns på skärmen som en streckad linje. Detta är till för att indikera att det finns mer information på lägre nivåer. Se fig 1.5.

#### Kontinuerlig zoomning

För att få en mjukare övergång vid t.ex. en ökning av zoomningsdjupet så plockas inte en ny nivå in direkt på skärmen. I stället får först höjden för listhuvudena till nya nivån succesivt öka. Den nya nivån kommer in på skärmen när dess rader får plats med minsta fontstorleken. Sedan ökar man fontstorleken för dessa rader med en enhet i taget. När fontstorleken för de nya raderna är lika med fontstorleken för omgivningen behandlas dessa på vanligt sätt. Vid fortsatt zoomning kommer hela proceduren att



upprepas. Se fig 1.4-1.6.

### Införande av ny lista

För att införa en ny lista sätter man först in en tom rad på det önskade stället. Detta måste användaren själv ge kommando om. Denna rad omvandlas sedan till ett listhuvud. Samtidigt sätts den första raden in i den nya listan.

Ett listhuvud kan inte tas bort direkt. Däremot när sista raden i en lista tas bort så att rownr blir noll för aktuella listhuvudet, då kommer automatiskt även detta att tas bort.

### Implementering

Det är sex procedurer som hanterar trädet och informationszoomningen. Proceduren makelisthead används för att omvandla en vanlig rad till ett listhuvud. Detta görs när man inför en ny nivå.

För att gå upp en nivå används proceduren uplevel. Detta behöver endast göras vid inskrivning av ny text, dvs när raden efter cursorn är densamma som textend.

### Beräkning av nya variabelvärden

I huvudprogrammet avläses spaken som reglerar informationszoomningen. Det avlästa värdet skalas och läggs till variabeln infolev. Om en ändring har skett kommer proceduren newtreevar att anropas. Här beräknas nya värden för informationsdjupet, variabeln infodeep, och för variabeln subzoom. Om subzoom blir större än tre så minskas denna med fyra och anger sedan fontstorleken för lägsta nivån på skärmen. Denna fontstorlek kan dock aldrig bli större än globala fontstorleken. Annars anger subzoom\*rownr höjden för synliga listhuvud (dvs listhuvud i lägsta nivån på skärmen) på skärmen, det bereds plats för en ny nivå. Vilket av fallen som gäller ges av variabeln setmark som ges värdet true i första fallet. Sambandet mellan dessa variabler ges också i fig 5.4.

### Ändring av informationsdjupet

Proceduren checklevel används sedan för att sätta fp och bp så att informationsdjupet i trädet blir det önskade infodeep. Om setmark är lika med true så kommer första och sista raderna i listorna på understa synliga nivå att bli speciellt markerade. Denna markering görs genom att status för dessa rader ökas med tio. Härmed har man fortfarande kvar informationen om det ursprungliga statusvärdet.

Infolev	Infodeep	Subzoom	Setmark
0	0	0	
1	0	1	
2	0	2	
3	0	3	
4	1	0	x
5	1	1	x
6	1	2	x
.	.	.	.
.	.	.	.
18	1	14	x
19	1	15	x
20	1	0	
21	1	1	
22	1	2	
23	1	3	
24	2	0	x
25	2	1	x
26	2	2	x
.	.	.	.
.	.	.	.
38	2	14	x
39	2	15	x
40	2	0	
41	2	1	
42	2	2	
43	2	3	
44	3	0	x
45	3	1	x
osv			

Fig 5.4 Samband mellan infolev, infodeep, subzoom och setmark. De tre senare variablerna beräknas ur värdet på den första, om setmark är satt anger subzoom fontstorleken för lägsta synliga nivå, annars anger subzoom\*rownr höjden för synliga listhuvud.

Eftersom status ändå kontrolleras för varje rad i goback och newhead är det enklast att använda status variabeln även för denna markering. Genom dessa markeringar kan nu den aktuella nivån specialbehandlas. Se fig 5.7-5.9

#### Specialbehandling av understa synliga nivå

De två sista procedurerna är bekanta sedan tidigare, nämligen subgoback och subnewhead. En informationszoomning följs alltid av en bildzoomning. Detta för att få rätt fontstorlek för den understa nivån på skärmen. När goback eller newhead träffar på en markerad rad anropas subgoback eller subnewhead. Sedan återgår man när den aktuella lista tar slut (fpo eller bpo lika med slut). På det här sättet blir den understa nivån på skärmen specialbehandlad och informationszoomningen kan härmed göras relativt kontinuerlig.

## 6 TERMINALKOMMANDO

Det finns tre olika slag av kommandon, kontroll tecken, escape sekvenser och kontroll sekvenser.

Vid hänvisning till ascii tabellen (se tabell 6.1) används representationen kolumn/rad. T.ex skrivs då tecknet 'K' som 4/11.

Kontroll tecken är enkla tecken som ligger i de två första kolumnerna i ascii tabellen. Programmet reagerar endast på de kontroll tecken som visas i tabell 6.2. Övriga kontroll tecken ignoreras.

Escape sekvenser är av formen ESC F. Här är ESC kontroll tecknet 1/11 i ascii tabellen och F ett godtyckligt tecken. Sekvensen känns igen av programmet om F är något av tecknen som visas i tabell 6.3. Annars ignoreras hela sekvensen.

Kontroll sekvenser är av formen ESC [ Pn F. ESC är samma som ovan och [ är tecknet 5/11 i ascii tabellen. pn är en parameter sekvens i området 3/0 till 3/9 inklusive gränserna. Endast en parameter kan användas. F slutligen kan vara ett godtyckligt tecken som inte tolkas som en parameter. Sekvensen känns igen om F är något av tecknen i tabell 6.4. Annars ignoreras hela sekvensen.

Ett parametervärde skickas med en siffra i taget. T.ex. värdet 123 skickas som tecken sekvensen 3/1, 3/2, 3/3.

Slutligen tolkas även tecknet DEL (7/15) som ett kommando.

## översikt över alla kommandon

## Cursor kommando:

CR	Vagnretur	CR
LF	Radframmatning	LF
BS	Bakåtsteg	BS
CUU	Cursor upp Pn= antalet rader	ESC [ Pn A
CUN	Cursor ner Pn= antalet rader	ESC [ Pn B
CUF	Cursor framåt Pn= antal positioner	ESC [ Pn C
CUB	Cursor bakåt Pn= antal positioner	ESC [ Pn D
CRN	Cursor en rad ner	ESC D
CRU	Cursor en rad upp	ESC M
CHA	Cursor horisontellt absolut Pn= kolumn nummer	ESC [ Pn G
CUH	Cursor hem	ESC [ H
CVA	Cursor vertikalt absolut Pn= rad nummer	ESC [ Pn d
SCP	Skicka cursor position	ESC J

## Editeringskommando:

SIT	Sätt in tecken Pn= antal tecken	ESC [ Pn @
TBT	Tag bort tecken Pn= antal tecken	ESC [ Pn P
NRF	Nya rader före Pn= antal rader	ESC [ Pn L
NRE	Nya rader efter Pn= antal rader	ESC [ Pn Y
TBR	Tag bort rader Pn= antal rader	ESC [ Pn M

## Bildkontroller:

UDP	Understrykning/dubbel intensitet på	ESC A
UDA	Understrykning/dubbel intensitet av	ESC B
UDT	Understrykning/dubbel intensitet för ett tecken	ESC C
US	Understrykning	ESC @
DI	Dubbel intensitet	ESC R
SCU	Scrolla upp Pn= antal linjer	ESC [ Pn S
SCN	Scrolla ner Pn= antal linjer	ESC [ Pn T
BZU	Bildzooma upp Pn= antal zoomningsnivåer	ESC [ Pn U
BZN	Bildzooma ner Pn= antal zoomningsnivåer	ESC [ Pn V
BZA	Bildzooma absolut Pn= zoomningsnivå	ESC [ Pn ^
AMF	Absolut minfont Pn= minfont	ESC [ Pn [
RMF	Relativ minfont Pn= minfont	ESC [ Pn ]
SS	Sätt status Pn= status	ESC [ Pn \

## Trädkontroller:

NLH	Nytt listhuvud	ESC N
UEN	Upp en nivå	ESC O
SAI	Skicka aktuellt informationsdjup	ESC T
IZA	Informationszooma absolut Pn= informationsdjupet	ESC [ Pn _

övriga kommando:

IT	Initialisera textminnet	ESC U
LHR	Ladda hzoom registret Pn= nytt värde för hzoom	ESC [ Pn ?
SR	Sudda i raden 0= från cursor till slut 1= från början till cursor 2= hela raden	ESC [ Pn K

## Kod tabeller

kol	0	1	2	3	4	5	6	7
rad								
0	NUL	DLE	SP	0	@	p	°	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
10	LF	SUB	*	:	J	Z	j	z
11	VT	ESC	+	;	K	[	k	{
12	FF	FS	,	<	L	\	l	
13	CR	GS	-	=	M	]	m	}
14	SO	RS	.	>	N	^	n	~
15	SI	US	/	?	O	_	o	DEL

Tabell 6.1 Ascii tabellen.

kol	0	1	2	3	4	5	6	7
rad								
0								
1								
2								
3								
4								
5								
6								
7								
8	BS							
9								
10	LF							
11		ESC						
12								
13	CR							
14								
15								

Tabell 6.2 Kontroll tecken.



	kol	0	1	2	3	4	5	6	7
rad									
0									
1						UDP	US		
2						UDA	DI		
3						UDT			
4						CRN	SAI		
5							IT		
6									
7									
8									
9									
10						SCP			
11									
12									
13						CRU			
14						NLH			
15						UEN			

Tabell 6.3 Escape sekvenser.

	kol	0	1	2	3	4	5	6	7
rad									
0						SIT	TBT		
1						CUU			
2						CUN			
3						CUF	SCU		
4						CUB	SCN	CVA	
5							BZU		
6							BZN		
7						CHA			
8						CUH			
9							NRE		
10									
11						SR	AMF		
12						NRF	SS		
13						TBR	RMF		
14							BZA		
15					LHR		IZA		

Tabell 6.4 Kontroll sekvenser.

## Kommando beskrivning

Nedan följer en beskrivning av alla kommandon som programmet känner igen.

AMF Absolut minfont.

ESC [ Pn [

Ändrar variablerna status och minfont för den aktuella raden. Status tilldelas värdet två motsvarande absolut minfont. Minfont tilldelas parametervärdet.

BS Bakåtsteg.

BS

Flyttar cursorn ett steg bakåt. Tecknet på denna plats ändras ej. Cursorn stannar vid vänster kant.

BZA Bildzooma absolut.

ESC [ Pn ^

Sätter bildzoomningsnivån till parametervärdet. Om värdet är skilt ifrån noll kan zoomningen inte påverkas av zoomningsspaken längre. För att återgå till normal användning av spaken måste zoomningsnivån sättas till noll genom kommando från värddatorn.

BZN Bildzooma ner.

ESC [ Pn V

Minskar zoomningsnivån med parametervärdet. Här utgås ifrån den nivå som tidigare blivit satt genom kommando från värddatorn. Dvs om spaken använts normalt innan så utgås ifrån värdet noll. Sedan påverkas funktionen hos zoomningsspaken på samma sätt som för BZA.

BZU Bildzooma upp.

ESC [ Pn U

Ökar zoomningsnivån med parametervärdet. Zoomningsnivån kan dock högst bli lika med 255. Se vidare BZN.

CHA Cursor horisontellt absolut.

ESC [ Pn G

Flyttar cursorn på den aktuella raden till positionen som anges av parametervärdet. Parametervärdet N flyttar cursorn till position N + 1. Cursorn flyttas ej förbi kanterna.

CR vagnretur.

CR

Cursorn flyttas till första positionen för raden.

CRN Cursor en rad ner.

ESC D

Se beskrivning för LF.

CRU Cursor en rad upp.  
ESC M

Cursorn flyttas uppåt till närmast föregående rad som inte är ett listhuvud. Cursorn stannar på första raden i texten.

CUB Cursor bakåt.  
ESC [ Pn D

Flyttar cursorn värdet av Pn positioner bakåt. Cursorn stannar vid vänstra kanten av raden.

CUH Cursor hem.  
ESC [ H

Flyttar cursorn till första positionen hos mittersta raden på skärmen.

CUF Cursor framåt.  
ESC [ Pn C

Flyttar cursorn värdet av Pn positioner framåt. Cursorn stannar vid högra kanten av raden.

CUN Cursor ner.  
ESC [ Pn B

Flyttar cursorn värdet av Pn rader neråt på skärmen. Cursorn stannar vid sista raden på skärmen eller i texten om någon av dessa påträffas.

CUU Cursor upp.  
ESC [ Pn A

Flyttar cursorn värdet av Pn rader uppåt på skärmen. Cursorn stannar vid första raden på skärmen eller i texten.

CVA Cursor vertikalt absolut.  
ESC [ Pn d

Flyttar cursorn till den rad som anges av parametervärdet. Värdet noll eller ett flyttar cursorn till första raden i texten.

DEL Radera tecken.  
DEL

Om det finns tecken efter cursorn så raderas dessa från och med cursorn. Annars flyttas cursorn ett steg bakåt och denna position raderas.

DI Dubbel intensitet.  
ESC R

Sätter status variabeln för aktuella raden så att markerade tecken på raden visas med dubbel intensitet.

IT Initialisera textminnet.  
ESC U

Initialiserar hela textminnet på nytt. Enklaste sättet att radera alla raderna i textminnet.

IZA Informationszooma absolut.

ESC [ Pn \_

Informationsdjupet på skärmen sätts till parametervärdet.

LF Radframmatning.

LF

Flyttar cursorn framåt till nästa rad som inte är ett listhuvud. Om cursorn från början befann sig på sista raden i texten sätts en ny rad in som cursorn sedan flyttas till. Den nya raden hämtas från poolen. Skulle poolen vara tom tas den första raden i texten i stället. Om cursorn kommit ner till underkanten av skärmen avläses kontroll spakarna för att kontrollera om bilden t.ex. ska scrollas.

LRH Ladda hzoom registret.

ESC [ Pn ?

Laddar hzoom registret med parametervärdet, dock högst 63. Ett värde skilt från noll medför att extra blanka punkter läggs in på skärmen i horisontell ledd.

NLH Nytt listhuvud.

ESC N

Omvandlar en vanlig rad till ett listhuvud. Samtidigt sätts den första raden in i nya listan. Cursorn flyttas sedan till den nya raden.

NRE Nya rader efter.

ESC [ Pn Y

Sätter in en ny blank rad efter den aktuella raden. Den nya raden kommer alltid på samma nivå i trädstrukturen som den aktuella. Cursorn flyttas sedan till den nya raden. Detta upprepas sedan tills värdet av Pn rader satts in.

NRF Nya rader före.

ESC [ Pn L

Samma som för NRE förutom att de nye raderna här sätts in för i stället för efter den aktuella raden.

RMF Relativ minfont.

ESC [ Pn ]

Ändrar variablerna status och minfont för aktuella raden. Status sätts till fyra motsvarande relativ minfont och variabeln minfont laddas med parametervärdet.

SAI Skicka aktuellt informationsdjup.

ESC T

Terminalen svara på detta kommando med att skicka det aktuella informationsdjupet på skärmen till värddatorn. Sekvensen som skickas är ESC [ Pn a där Pn är något av tecknen noll till nio motsvarande informationsdjupet.

SCN Scrolla ner.

ESC [ Pn T

Bilden scrollas ner värdet av Pn linjer.

SCP Skicka cursor position.

ESC J

Terminalen svara på detta kommando med att skicka sekvensen ESC [ Pn1 ; Pn2 R. Pn1 motsvarar cursorns vertikala position och Pn2 motsvarar cursorns horisontella position. Pn1 och Pn2 är här vanliga parametersekvenser, (Se början av kapitlet).

SCU Scrolla upp.

ESC [ Pn S

Bilden scrollas upp värdet av Pn linjer.

SIT Sätt in tecken.

ESC [ Pn @

Sätter in värdet av Pn blanka tecken från och med cursorns position och framåt. De ursprungliga tecknen på dessa platserna flyttas fram Pn positioner. Eventuellt förloras dessa tecken om raden inte räcker till.

SR Sudda i raden.

ESC [ Pn K

Suddar några eller alla tecken på aktuella raden beroende på parametervärdet.

0: Från cursorn till slutet av raden.

1: Från början av raden till och med cursorn.

2: Hela raden.

SS Sätt status.

ESC [ Pn \

Om parametervärdet ligger mellan noll och sju inklusive gränserna så laddas variabeln status för aktuella raden med detta värde.

TBR Tag bort rader.

ESC [ Pn M

Tar bort aktuella raden. Om listan där denna rad fanns nu blivit tom tas även listhuvudet för denna lista bort. Cursorn flyttas sedan till nästa rad. Detta upprepas tills värdet av Pn rader tagits bort.

TBT Tag bort tecken.

ESC [ Pn P

Tar bort värdet av Pn tecken från och med cursorn på aktuella raden. Dock tas inte fler tecken bort när det inte finns fler tecken på cursor positionen eller höger därom.

UDA Understrykning/dubbel intensitet av.

ESC B

Stänger av moden för att markera tecken så att de visas understrykna eller med dubbel intensitet.

UDP Understrykning/dubbel intensitet på.

ESC A

Sätter på moden beskriven i UDA.

UDT Understrykning/dubbel intensitet för ett tecken.

ESC C

Markerar aktuella tecknet så det visas understryket eller med dubbel intensitet. Markeringen försvinner om man skriver ett nytt tecken i denna position.

UEN Upp en nivå.

ESC O

Om den aktuella raden är den sista raden i texten så kommer denna rad att flyttas upp en nivå i trädstrukturen. Annars görs ingenting. Kommandot används för att avsluta en lista när textminnet fylls med ny text.

US Understrykning.

ESC @

Sätter variabeln status för aktuella raden så att markerade tecken visas understrykna.

REFERENSER

- 1: Lindvall Bertil, Lynner Bertil: Videogenerator för zoomning av alfanumerisk text, Examensarbete IV5 8101, mars 1981, Avdelningen för Digital och Datorteknik, Lunds Tekniska Högskola.

APPENDIX\_A

Nedan följer en listning på hela programmet.



```
1 (*****  
2 (*  
3  
4  
5  
6         terminal with zoom  
7  
8  
9     this program is written for an intel 8085 microprocessor,  
10    wich is controlling an alphanumeric computer terminal.  
11  
12    with the hardware provides the program facilities as  
13    zooming of alphanumeric text and information zooming.  
14  
15    the program is done as an examen work at the department of  
16    automatic control, lund institute of technology.  
17  
18                                     written by bengt ohlsson  
19                                     supervisor hilding elmqvist  
20  
21  
22  
23                                                     *)  
24 (*****  
25  
26  
27  
28 program terminal(input,output,dator);  
29  
30 const maxchar= 80;  
31     headnr= 34;  
32     rowbytes= 114;    (* maxchar + headnr *)  
33     skaln= 16;  
34     blank= ' '  
35     listh= '-'  
36     nrlines= 140;  
37     (* orte adr *)   port8c= 140;  
38     (* orte *)      port8d= 141;  
39     (* resetorte *) port8e= 142;  
40     (* ctrl reg *)  port8f= 143;  
41     (* ad1 *)       port50= 80;  
42     (* ad2 *)       port60= 96;  
43     (* ad3 *)       port70= 112;  
44  
45 type text= file of char;  
46  
47     headtype= (antal,font,minfont,slask,  
48               totheight,firstch,firstkol);  
49  
50     rowpointer= ^row;  
51     row= record  
52         vip: integer;  
53         status: integer;  
54         rowhead: array[headtype] of char;  
55         string: array[1..maxchar] of char;  
56         rownr,level: integer;  
57         subip,subbp,headip,headbp,hover: rowpointer;  
58         ip,bp,ipo,bpo: rowpointer  
59     end;  
60
```

```
61     sht= record
62         case tag: integer of
63             1: (sp: rowpointer);
64             2: (int: integer);
65             3: (cha: array[1..2] of char);
66         end;
67
68     dum= record
69         case tag: boolean of
70             true: (dpoint: rowpointer);
71             false: (dint: integer);
72         end;
73
74 var dator: text;
75     dummy: dum;
76     shift: sht;
77     indx: headtype;
78     ch, ch1, ch2, ch3, ch4, ch5, chnoll: char;
79     i, i1, i2, i3, i4, i5: integer;
80     p, p1, p2, p3, p4, p5: rowpointer;
81
82     (* common variables *)
83     klar, testklar: boolean;
84     del: char;
85     val, par, cpkol, ctrlreg: integer;
86     textstart, textend, cp, scratch, pool: rowpointer;
87
88     (* zoom variables *)
89     zscr, zscrttest: boolean;
90     intfont, zoomlev, newzlev, vertscr, blanklines: integer;
91     inchgt, inchgt1, inchgt2, startinc: integer;
92     newfont, ulfont, startfont: char;
93     height, height1, height2, starthgt: char;
94     startpoint, endpoint, point: rowpointer;
95
96     fontheight: array[-1..15] of char;
97     intheight: array[-1..15] of integer;
98     fontwidth: array[0..15] of char;
99
100     (* tree variables *)
101     infoz, setmark: boolean;
102     subfont, subheight: char;
103     infolev, infochange, subinthgt: integer;
104     currentlev, maxdeep, infodeep, subzoom: integer;
105     headstart, headover, midover, headrow, slut: rowpointer;
106
107     (* screen variables *)
108     uldi: boolean;
109     maxlines, lines, midlines, endlines: integer;
110     middle: rowpointer;
111
112     (* external procedures *)
113
114 procedure setstart(p: rowpointer); external;
115     (* sets the start address for the videogenerator *)
116
117 procedure setcursor(i: integer); external;
118     (* sets the row address for the cursor *)
119
120 procedure cursorkol(i: integer); external;
```

```
121      (* sets the horizontal position for the cursor *)
122
123 procedure hzoom(i: integer); external;
124      (* sets additional points in the horizontal video scan *)
125
126 procedure bwait; external;
127      (* to synchronize the program with the videogenerator *)
128
129 function keyboard: boolean; external;
130      (* to check if anything is coming from the keyboard *)
131
132 function host: boolean; external;
133      (* to check if anything is coming from the host computer *)
134
135 procedure putport(data,adr: integer); external;
136      (* to put a new value on an output *)
137
138 function getport(adr: integer): integer; external;
139      (* to get a value from an input *)
140
141      (* end external procedures *)
142
143
144 function pswap(p: rowpointer): integer;
145      (* swaps two bytes. used to generate the video forwardpointer *)
146 begin
147     with shift do
148         begin
149             sp:= p;
150             ch5:= cha[1];
151             cha[1]:= cha[2];
152             cha[2]:= ch5;
153             pswap:= int;
154         end;
155     end;
156
157 procedure pushpool(p: rowpointer);
158      (* pushes a row on the pool *)
159 begin
160     p^:= scratch^;
161     p^.fp:= pool;
162     pool:= p;
163 end;
164
165 procedure delete(p: rowpointer);
166      (* deletes a row from the screen *)
167 begin
168     if startpoint = p then
169         begin
170             startpoint:= p^.bp;
171             setstart(startpoint);
172         end;
173     with p^ do
174         begin
175             bp^.vip:= vip;
176             bp^.fp:= fp;
177             fp^.bp:= bp;
178             if bpo = slut then
179                 headover^.subfp:= fpo
180             else
```

```
181     bpo^.fpo:= fpo;
182     if fpo = slut then
183         headover^.subbp:= bpo
184     else
185         fpo^.bpo:= bpo;
186         if status = 7 then
187             begin
188                 headfp^.headbp:= headbp;
189                 headbp^.headfp:= headfp;
190             end;
191         end;
192     with headover^ do rownr:= rownr - 1;
193 end;
194
195 function getrow: rowpointer;
196 (* gives a pointer to a new row. if the pool is empty the
197 procedure deletes the first row *)
198 begin
199     if pool = nil then
200         begin
201             p5:= textstart^.fpo;
202             while p5^.rownr > 0 do p5:= p5^.subfp;
203             delete(p5);
204             p5:= scratch^;
205             getrow:= p5;
206         end
207     else
208         begin
209             getrow:= pool;
210             pool:= pool^.fp;
211         end;
212 end;
213
214 procedure inafter(p,q: rowpointer);
215 (* inserts a new row p after q *)
216 begin
217     p5:= p;
218     with p5^ do
219         begin
220             fp:= q^.fp;
221             vfp:= q^.vfp;
222             bp:= q;
223             q^.fp:= p;
224             q^.vfp:= pswap(p);
225             fp^.bp:= p;
226             fpo:= q^.fpo;
227             bpo:= q;
228             q^.fpo:= p;
229             hover:= headover;
230             if fpo = slut then
231                 headover^.subbp:= p
232             else
233                 begin
234                     fpo^.bpo:= p;
235                     if fpo^.status = 7 then fpo^.bp:= p;
236                 end
237             end;
238         with headover^ do rownr:= rownr + 1;
239     end;
240
```

```
241 procedure inbefore(p,q: rowpointer);
242 (* inserts a new row p before q *)
243 begin
244   p5:= p;
245   with p5^ do
246     begin
247       fp:= q;
248       vfp:= q^.bp^.vfp;
249       bp:= q^.bp;
250       bp^.fp:= p;
251       bp^.vfp:= pswap(p);
252       q^.bp:= p;
253       fpo:= q;
254       bpo:= q^.bpo;
255       q^.bpo:= p;
256       hover:= headover;
257       if bpo = slut then
258         headover^.subfp:= p
259       else
260         begin
261           bpo^.fpo:= p;
262           if bpo^.status = 7 then
263             begin
264               bpo^.fp:= p;
265               bpo^.vfp:= pswap(p);
266             end;
267           end;
268         end;
269       with headover^ do rownr:= rownr + 1;
270     end;
271 end;
272 function opout: boolean;
273 (* checks if the cursor is on the screen, if so the
274    function value becomes true *)
275 begin
276   p:= endpoint;
277   opout:= true;
278   while (p <> startpoint) and (p <> op) do
279     p:= p^.bp;
280   if p = op then opout:= false;
281 end;
282
283 procedure resetdisp;
284 (* moves the row with the cursor to th middle of the screen.
285    if necessary changes the information deep so the
286    cursor is seen on the screen *)
287 begin
288   middle:= op;
289   midlines:= 0;
290   midover:= headover;
291   i5:= headover^.level;
292   if infodeep < i5 then
293     begin
294       infodeep:= i5;
295       subzoom:= 0;
296       infolev:= infodeep * 20;
297       setmark:= false;
298       infoz:= true;
299     end;
300   klar:= true;
```

```
301 end;
302
303 procedure ophome;
304 (* moves the cursor to the row in the middle of the screen *)
305 begin
306   op:= middle;
307   opkol:= 0;
308   headover:= midover;
309   zscr:= false;
310 end;
311
312 procedure linefeed;
313 (* moves the cursor down to the next row wich is not a listhead.
314   inserts a new row if textend is reached *)
315 begin
316   repeat
317     if op^.fp = textend then
318       begin
319         point:= getrow;
320         point^.rowhead[totheight]:= height1;
321         point^.rowhead[font]:= newfont;
322         inafter(point,op);
323         op:= point;
324         endlines:= endlines - inchgt1;
325         if endlines < 10 then zsertest:= true;
326       end
327     else
328       op:= op^.fp;
329     until op^.status <> 7;
330   headover:= op^.hover;
331 end;
332
333 procedure reverself;
334 (* moves the cursor up to the next row wich is not a listhead *)
335 begin
336   p:= op^.bp;
337   while p^.status = 7 do p:= p^.bp;
338   if p <> textstart then op:= p;
339   if op = startpoint^.bp then zscr:= true;
340   headover:= op^.hover;
341 end;
342
343 procedure opleft(ant: integer);
344 (* moves the cursor ant steps to the left, stops at the left margin *)
345 begin
346   opkol:= opkol - ant;
347   if opkol < 0 then opkol:= 0;
348 end;
349
350 procedure opright(ant: integer);
351 (* moves the cursor ant steps to the right, stops at the right margin *)
352 begin
353   opkol:= opkol + ant;
354   if opkol > maxchar then opkol:= maxchar;
355 end;
356
357 procedure opup(ant: integer);
358 (* moves the cursor up ant rows but not out of the screen *)
359 begin
360   while (ant > 0) and (op <> startpoint) and (op <> textstart^.fp) do
```

```
361     begin
362     cp:= cp^.bp;
363     ant:= ant - 1;
364     end;
365     headover:= cp^.hover;
366     end;
367
368 procedure cpldown(ant: integer);
369 (* moves the cursor down ant rows but not out of the screen *)
370 begin
371 while (ant > 0) and (cp <> endpoint) and (cp <> textend^.bp) do
372     begin
373     cp:= cp^.fp;
374     ant:= ant - 1;
375     end;
376     headover:= cp^.hover;
377     end;
378
379 procedure cpvertabsolute(ant: integer);
380 (* positions the cursor to the ant:th row from the beginning *)
381 begin
382 if ant = 0 then ant:= 1;
383 cp:= textstart;
384 while ant > 0 do
385     begin
386     ant:= ant - 1;
387     cp:= cp^.fp;
388     while cp^.rownr > 0 do cp:= cp^.subfp;
389     end;
390 if cp^.status = 6 then cp:= textend^.bp;
391     end;
392
393 procedure getvertpos(var ant:integer);
394 (* to get the cursors vertical position from the textstart *)
395 begin
396 p:= cp;
397 ii:= 0;
398 while p <> textstart do
399     begin
400     ii:= ii + 1;
401     p:= p^.bp;
402     while p^.rownr > 0 do p:= p^.subbp;
403     end;
404 ant:= ii;
405     end;
406
407 procedure writechar;
408 (* writes a char on the screen at the cursor position
409 and moves the cursor one step to the right *)
410 begin
411 with cp^ do
412     begin
413     if uldi then string[opkol]:= chr(i + 128)
414     else string[opkol]:= ch;
415     ch5:= chr(opkol + 1);
416     if rowhead[antall] < ch5 then rowhead[antall]:= ch5;
417     end;
418 if opkol < maxchar then opkol:= opkol + 1;
419     end;
420
```

```

421 procedure insertchar(ant: integer);
422 (* inserts ant blank chars after the cursor *)
423 begin
424   with cp^ do
425     begin
426       i2:= ord(rowhead[ant]);
427       if opkol < i2 then
428         begin
429           i1:= maxchar - opkol + 1;
430           if ant > i1 then ant:= i1;
431           i2:= i2 + ant - 1;
432           if i2 > maxchar then i2:= maxchar;
433           i3:= ant + opkol;
434           for i1:= i2 downto i3 do
435             string[i1]:= string[i1 - ant];
436           for i1:= opkol to i3 - 1 do
437             string[i1]:= blank;
438           rowhead[ant]:= chr(i2 + 1);
439         end;
440       end;
441     end;
442
443 procedure delchar(d: integer);
444 (* deletes d chars after the cursor *)
445 var ant: integer;
446 begin
447   with cp^ do
448     begin
449       ant:= ord(rowhead[ant]);
450       if ant > opkol then
451         begin
452           i2:= ant - opkol;
453           if d > i2 then d:= i2;
454           for i1:= opkol to ant - d - 1 do
455             string[i1]:= string[i1 + d];
456           for i1:= ant - d to ant - 1 do
457             string[i1]:= blank;
458           rowhead[ant]:= chr(ant - d);
459         end;
460       end;
461     end;
462
463 procedure makelisthead(p: rowpointer);
464 (* converts a row to a listhead and puts a new row in the new list.
465   the cursor is moved to the new row *)
466 begin
467   p5:= getrow;
468   with p^ do
469     begin
470       subfp:= p5;
471       subbp:= p5;
472       headfp:= headover^.headfp;
473       headbp:= headover;
474       rownr:= 1;
475       status:= 7;
476       i5:= headover^.level + 1;
477       level:= i5;
478       infolev:= i5 * 20;
479       if maxdeep < i5 then maxdeep:= i5;
480       bp^.fp:= p5;

```



```
481     bp^.vip:= pswap(p5);
482     ip^.bp:= p5;
483     i4:= headover^.level;
484     i4:= i4 + i4;
485     i5:= i4 + 5;
486     if i5 > maxchar then i5:= maxchar;
487     for ii:= i4 to i5 do
488         string[ii]:= listh;
489     rowhead[antal]:= chr(i5 + 1);
490     end;
491     headover^.headfp:= p;
492     p^.headfp^.headbp:= p;
493     headover:= p;
494     with p5^ do
495         begin
496             hover:= headover;
497             bpo:= slut;
498             fpo:= slut;
499             ip:= p^.ip;
500             vip:= p^.vip;
501             bp:= p^.bp;
502         end;
503     cp:= p5;
504     klar:= true;
505     cpkol:= 0;
506     end;
507
508 procedure uplevel;
509 (* if the cursor is at textend the current row is moved
510    up one level in the tree *)
511 begin
512     if (cp^.fp = textend) and (headover^.level > 0) then
513         begin
514             with headover^ do
515                 begin
516                     fp:= cp;
517                     vip:= cp^.bp^.vip;
518                     fpo:= cp;
519                     rownr:= rownr - 1;
520                 end;
521             with cp^ do
522                 begin
523                     headover^.subbp:= bpo;
524                     bpo^.fpo:= slut;
525                     bpo:= headover;
526                     headover:= headover^.hover;
527                     hover:= headover;
528                 end;
529             with headover^ do rownr:= rownr + 1;
530         end
531     end;
532
533 procedure newtreevar;
534 (* computes new values for the tree variables *)
535 begin
536     i4:= maxdeep * 20;
537     if (infolev = 0) or (infolev = i4) then
538         infoz:= false;
539     infolev:= infolev + infochange;
540     if infolev < 0 then
```

```
541   infolev:= 0
542   else if infolev > i4 then
543     infolev:= i4;
544   infodeep:= (infolev + 16) div 20;
545   subzoom:= infolev mod 20;
546   if subzoom > 3 then
547     begin
548       subzoom:= subzoom - 4;
549       setmark:= true;
550     end
551   else
552     setmark:= false;
553   if subzoom > intfont then
554     subzoom:= intfont;
555   end;
556
557 procedure checklevel;
558 (* uses the values from newtreevar and changes the structure
559    on the screen, fp,vfp,bp is changed *)
560 begin
561   subfont:= chr(subzoom);
562   subheight:= fontheight[subzoom - 1];
563   subinhtgt:= intheight[subzoom - 1];
564   headrow:= headstart^.headfp;
565   i3:= infodeep + 1;
566   while headrow <> headstart do
567     begin
568       p5:= headrow;
569       with p5^ do
570         begin
571           if (level < i3) and (rownr > 0) then
572             begin
573               bp^.fp:= subfp;
574               bp^.vfp:= pswap(subfp);
575               fp^.bp:= subbp;
576               subfp^.bp:= bp;
577               subbp^.fp:= fp;
578               subbp^.vfp:= vfp;
579               if (level = infodeep) and setmark then
580                 begin
581                   with subfp^ do
582                     if status < 8 then status:= status + 10;
583                   with subbp^ do
584                     if status < 8 then status:= status + 10;
585                 end
586               else
587                 begin
588                   with subfp^ do
589                     if status > 7 then status:= status - 10;
590                   with subbp^ do
591                     if status > 7 then status:= status - 10;
592                 end;
593             end
594           else if level = i3 then
595             begin
596               bp^.fp:= headrow;
597               bp^.vfp:= pswap(headrow);
598               fp^.bp:= headrow;
599             end;
600         end;
599     end;
600   end;
```

```
001     headrow:= headrow^.headip;
002     end;
003 end;
004
005 procedure initialize;
006 (* initializes some variables at start of the program *)
007 begin
008     chnoll:= chr(0);
009     del:= chr(127);
010     val:= 1;
011     zscr:= false;
012     newzlev:= 0;
013     vertscr:= 0;
014     maxdeep:= 0;
015     infolev:= 0;
016     uldi:= false;
017     for i:= 0 to 5 do fontwidth[i]:= chr(6);
018     for i:= 6 to 8 do fontwidth[i]:= chr(7);
019     for i:= 9 to 13 do fontwidth[i]:= chr(i - 1);
020     fontwidth[14]:= chr(14);
021     fontwidth[15]:= chr(16);
022     for i:= -1 to 3 do intheight[i]:= i + 5;
023     for i:= 4 to 6 do intheight[i]:= i + i + 2;
024     intheight[7]:= 8;
025     for i:= 8 to 13 do intheight[i]:= i + i - 7;
026     intheight[14]:= 22;
027     intheight[15]:= 26;
028     for i:= -1 to 15 do
029         fontheight[i]:= chr(intheight[i]);
030     end;
031
032 procedure setpool;
033 (* clears the hole textmemory and reinitializes the control rows.
034    can be done by command from the host computer *)
035 var rowant: integer;
036 begin
037     pool:= nil;
038     maxdeep:= 0;
039     with dummy do
040     begin
041         dint:= -16384; (* c000 *)
042         rowant:= -dint div rowbytes;
043         scratch:= dpoint;
044         with scratch^ do
045         begin
046             status:= 0;
047             for indx:= antal to firstkol do
048                 rowhead[indx]:= chnoll;
049             rowhead[totheight]:= chr(4);
050             for i5:= 0 to maxchar + headnr - 16 do
051                 string[i5]:= chnoll;
052             end;
053             for i:= 1 to rowant - 1 do
054             begin
055                 dint:= dint + rowbytes;
056                 pushpool(dpoint);
057             end;
058             point:= getrow;
059             textstart:= point;
060             textstart^.bp:= point;
```

```
661   for i:= 1 to 2 do
662     begin
663       p:= point;
664       point:= getrow;
665       point^.bp:= p;
666       point^.bpo:= p;
667       p^.fp:= point;
668       p^.fpo:= point;
669       p^.vip:= pswap(point);
670     end;
671   point^.ip:= point;
672   point^.vfp:= pswap(point);
673   point^.fpo:= point;
674   textend:= p;
675   textstart^.status:= 6;
676   textend^.status:= 6;
677   point^.status:= 6;
678   slut:= point;
679   headstart:= getrow;
680   headover:= headstart;
681   with headover^ do
682     begin
683       headfp:= headstart;
684       headbp:= headstart;
685       hover:= headstart;
686       level:= 0;
687       status:= 7;
688     end;
689   midover:= headstart;
690   middle:= getrow;
691   inafter(middle,textstart);
692   midlines:= 0;
693   opkol:= 0;
694   op:= middle;
695   startpoint:= middle;
696   dpoint:= middle;
697   setstart(startpoint);
698   setcursor(dint + 4);
699   cursorkol(1);
700   end;
701 end;
702
703 procedure subgoback;
704 (* used by goback for going back at the lowest level on screen *)
705 begin
706   testklar:= false;
707   while (lines > 0) and not testklar do
708     begin
709       point:= point^.bp;
710       lines:= lines - subinthgt;
711       if point^.bpo = slut then testklar:= true;
712     end;
713 end;
714
715 procedure subnewhead;
716 (* used by newhead to set the fontinformation for the rows
717    at the lowest level on screen *)
718 begin
719   testklar:= false;
720   while (lines > 0) and not testklar do
```

```
721 begin
722 point:= point^.fp;
723 with point^ do
724   begin
725     rowhead[font]:= subfont;
726     rowhead[toheight]:= subheight;
727     lines:= lines - subinhtgt;
728     if fpo = slut then testklar:= true;
729   end;
730 end;
731 end;
732
733 procedure goback;
734 (* used by newstartadr to get the new start address *)
735 begin
736 if not testklar then subgoback;
737 while lines > 0 do
738   begin
739     point:= point^.bp;
740     case point^.status of
741       0,1: lines:= lines - inchgt;
742       2,3: begin
743         ch2:= point^.rowhead[font];
744         if ch2 > newfont then
745           lines:= lines - intheight[ord(ch2)];
746         else
747           lines:= lines - inchgt;
748         end;
749       4,5: begin
750         i2:= intfont + ord(point^.rowhead[font]);
751         if i2 > 15 then i2:= 15;
752         lines:= lines - intheight[i2];
753       end;
754       6: lines:= lines - 255;
755       7: with point^ do
756         begin
757           i5:= rownr * subzoom;
758           if i5 < 4 then i5:= 4;
759           lines:= lines - i5;
760         end;
761       10,11,12,13,14,15,16,17:
762         begin
763           point:= point^.fp;
764           subgoback;
765         end;
766     end;
767   end;
768 end;
769
770 procedure newhead;
771 (* used by changefont to change the fontinformation *)
772 begin
773 if not testklar then subnewhead;
774 while lines > 0 do
775   begin
776     point:= point^.fp;
777     with point^ do
778       begin
779         rowhead[font]:= newfont;
780         rowhead[toheight]:= height;
```

```
781     lines:= lines - inchgt;
782     case status of
783     0:   ;
784     1:   rowhead[font]:= ulfont;
785     2:   begin
786         ch2:= rowhead[font];
787         if ch2 > newfont then
788             begin
789                 rowhead[font]:= ch2;
790                 i2:= ord(ch2);
791                 rowhead[totheight]:= fontheight[i2];
792                 lines:= lines - intheight[i2] + inchgt;
793             end;
794         end;
795     3:   begin
796         ch2:= rowhead[font];
797         if ch2 > newfont then
798             begin
799                 i2:= ord(ch2);
800                 rowhead[font]:= chr(i2 + 128);
801                 rowhead[totheight]:= fontheight[i2];
802                 lines:= lines - intheight[i2] + inchgt;
803             end;
804         end;
805     4:   begin
806         i2:= intfont + ord(rowhead[font]);
807         if i2 > 15 then i2:= 15;
808         rowhead[font]:= chr(i2);
809         rowhead[totheight]:= fontheight[i2];
810         lines:= lines - intheight[i2] + inchgt;
811     end;
812     5:   begin
813         i2:= intfont + ord(rowhead[font]);
814         if i2 > 15 then i2:= 15;
815         rowhead[font]:= chr(i2 + 128);
816         rowhead[totheight]:= fontheight[i2];
817         lines:= lines - intheight[i2] + inchgt;
818     end;
819     6:   begin
820         if point = textend then
821             endlines:= endlines + inchgt;
822         lines:= lines - 255 + inchgt;
823     end;
824     7:   begin
825         i5:= rownr * subzoom;
826         if i5 < 4 then i5:= 4;
827         rowhead[font]:= chr(2);
828         rowhead[totheight]:= chr(i5);
829         lines:= lines - i5 + inchgt;
830     end;
831     10,11,12,13,14,15,16,17:
832         begin
833             point:= point^.bp;
834             lines:= lines + inchgt;
835             subnewhead;
836         end;
837     end;
838 end;
839 end;
840 end;
```

```
841
842 procedure newzvar;
843 (* computes new values for the variables used for
844    screen zooming *)
845 begin
846   intfont:= zoomlev div skaln;
847   if intfont < 8 then
848     begin
849       ctrlreg:= 16;
850       maxlines:= nrlines + nrlines;
851     end
852   else
853     begin
854       ctrlreg:= 0;
855       maxlines:= nrlines;
856     end;
857   newfont:= chr(intfont);
858   ulfont:= chr(intfont + 128);
859   blanklines:= (maxlines * (zoomlev mod skaln)) div skaln;
860   height1:= fontheight[intfont - 1];
861   height2:= fontheight[intfont];
862   inchgt1:= intheight[intfont - 1];
863   inchgt2:= intheight[intfont];
864   if intfont = 7 then
865     begin
866       inchgt2:= 16;
867       height2:= chr(16);
868     end;
869   end;
870
871 procedure newstartadr;
872 (* computes the new start address for the videogenerator when
873    zooming or/and scrolling *)
874 begin
875   height:= height1;
876   inchgt:= inchgt2;
877   if subzoom > intfont then
878     begin
879       subzoom:= intfont;
880       subfont:= chr(subzoom);
881       subheight:= fontheight[subzoom - 1];
882       subinchgt:= intheight[subzoom - 1];
883     end;
884   while midover^.level > infodeep do
885     begin
886       middle:= midover;
887       midover:= midover^.hover;
888     end;
889   while (midover^.level < infodeep) and (middle^.rownr > 0) do
890     begin
891       midover:= middle;
892       middle:= middle^.subip;
893     end;
894   if vertscr <> 0 then
895     begin
896       midlines:= midlines - vertscr;
897       i2:= ord(middle^.rowhead[totheight]);
898       while (midlines >= i2) and (middle <> textstart^.ip) do
899         begin
900           midlines:= midlines - i2;
```

```
901     middle:= middle^.bp;
902     i2:= ord(middle^.rowhead[totheight]);
903     end;
904     while (midlines < 0) and (middle <> textend^.bp) do
905     begin
906         middle:= middle^.fp;
907         midlines:= midlines + ord(middle^.rowhead[totheight]);
908     end;
909 end;
910 midover:= middle^.hover;
911 if midlines > 30 then midlines:= 30;
912 if midlines < 0 then midlines:= 0;
913 point:= middle^.fp;
914 lines:= midlines + blanklines - inchgt;
915 if (midover^.level = infodeep) and setmark then
916     testklar:= false
917 else
918     testklar:= true;
919 goback;
920 lines:= lines + maxlines - blanklines + inchgt;
921 inchgt:= inchgti;
922 goback;
923 startpoint:= point;
924 if (point^.hover^.level = infodeep) and setmark then
925     testklar:= false
926 else
927     testklar:= true;
928 with startpoint^ do
929     begin
930     startfont:= newfont;
931     startinc:= inchgt;
932     case status of
933     0:   ;
934     1:   startfont:= ulfont;
935     2:   begin
936         ch2:= rowhead[minfont];
937         if ch2 > newfont then
938             begin
939                 startfont:= ch2;
940                 startinc:= intheight[ord(ch2)];
941             end;
942         end;
943     3:   begin
944         ch2:= rowhead[minfont];
945         if ch2 > newfont then
946             begin
947                 i2:= ord(ch2);
948                 startfont:= chr(i2 + 128);
949                 startinc:= intheight[i2];
950             end;
951         end;
952     4:   begin
953         i2:= intfont + ord(rowhead[minfont]);
954         if i2 > 15 then i2:= 15;
955         startfont:= chr(i2);
956         startinc:= intheight[i2];
957     end;
958     5:   begin
959         i2:= intfont + ord(rowhead[minfont]);
960         if i2 > 15 then i2:= 15;
```



```
961         startfont:= chr(i2);
962         startinc:= intheight[i2];
963         end;
964         6:   startinc:= 255;
965         7:   begin
966             startinc:= rownr * subzoom;
967             if startinc < 4 then startinc:= 4;
968             end;
969         10,11,12,13,14,15,16,17:
970             begin
971                 startinc:= subinhtgt;
972                 startfont:= subfont;
973             end;
974         end;
975     end;
976     if startinc = 255 then startinc:= inohgt1;
977     if not testklar then
978     begin
979         startfont:= subfont;
980         startinc:= subinhtgt;
981     end;
982     lines:= lines + startinc;
983     if lines > maxlines then lines:= maxlines;
984     starthgt:= chr(lines);
985 end;
986
987 procedure changefont;
988 (* changes the fontinformation for the rows that will be seen on
989    the screen after the change *)
990 begin
991     endlines:= 0;
992     lines:= maxlines - blanklines - lines;
993     point:= startpoint;
994     (* bwait *)
995     setstart(startpoint);
996     putport(ctrlreg,port&f);
997     with point^ do
998         begin
999             rowhead[font]:= startfont;
1000            rowhead[totheight]:= starthgt;
1001            if ipo = slut then testklar:= true;
1002            end;
1003        newhead;
1004        lines:= lines + blanklines + blanklines;
1005        height:= height2;
1006        inchgt:= inchgt2;
1007        newhead;
1008        lines:= lines + maxlines - blanklines;
1009        height:= height1;
1010        inchgt:= inchgt1;
1011        newhead;
1012        endpoint:= point;
1013    end;
1014
1015 begin
1016 (* writes in the monitor to the microprocessor *)
1017 writeln('init');
1018 initialize;
1019 setpool;
1020 zoomlev:= 5;
```

```
021 writeln('ready');
022
023 while true do
024   begin
025
026     (* from hostcomputer *)
027
028     zsortest:= false;
029     klar:= false;
030     (* if zsortest becomes true tests if zooming or
031        scrolling will be done.
032        if klar becomes true do always a zoom or scroll *)
033     while host and not klar and not zsortest do
034       begin
035
036         (* reads characters coming from the hostcomputer *)
037         read(dator,ch);
038         i:= ord(ch);
039
040         case val of
041         i: begin
042           (* tests if it is a control character
043              otherwise write it on the screen *)
044           if (i < 32) or (i = 127) then
045             begin
046               if ch = del then
047                 begin
048                   (*del*) i2:= ord(cp^.rowhead[antall]) - 1;
049                   if (cpkol > 0) and (i2 < cpkol) then
050                     cpkol:= cpkol - 1;
051                   for i:= cpkol to i2 do
052                     cp^.string[i]:= blank;
053                   cp^.rowhead[antall]:= chr(cpkol);
054                   end
055                   else if (i > 7) and (i < 28) then
056                     case i of
057                     (*bs*) 8:   if cpkol > 0 then cpkol:= cpkol - 1;
058                     (*lf*) 10:  linefeed;
059                     (*cr*) 13:  begin
060                               cpkol:= 0;
061                               end;
062                     (*esc*) 27:  val:= 2;
063                               9,11,12,14,15,16,17,18,19,20,21,22,23,24,25,26: ;
064                               end;
065                   end
066                 else
067                   writechar;
068                 (* tests if the cursor is on the screen *)
069                 if zsor and (val <> 2) then
070                   begin
071                     if cpout then resetdisp;
072                     zsor:= false;
073                   end;
074                 end;
075
076             2: begin
077               val:= 1;
078               (* checks if it is a proper escape sequense *)
079               if (i > 64) and (i < 92) then
080                 case i of
```

```

081      (*l*) 91: begin
082            val:= 3;
083            par:= 0;
084            end;
085      (*a*) 65: uldi:= true;
086      (*b*) 66: uldi:= false;
087      (*c*) 67: begin
088            i5:= ord(cp^.string[cpkol]);
089            if i5 < 128 then i5:= i5 + 128;
090            cp^.string[cpkol]:= chr(i5);
091            end;
092      (*d*) 68: linefeed;
093      (*m*) 77: reverseli;
094      (*j*) 74: begin
095            (* sends cursor position to host computer *)
096            getvertpos(par);
097            write(dator,chr(27));      (*esc*)
098            write(dator,chr(91));      (* [ *)
099            write(dator,chr(par div 100 + 48));      (*par*)
100            par:= par mod 100;
101            write(dator,chr(par div 10 + 48));
102            write(dator,chr(par mod 10 + 48));
103            write(dator,chr(59));      (* ; *)
104            par:= cpkol;
105            write(dator,chr(48));
106            write(dator,chr(par div 10 + 48));      (*kol*)
107            write(dator,chr(par mod 10 + 48));
108            write(dator,chr(82));      (* r *)
109            end;
110      (*n*) 78: if cp^.rownr = 0 then makelisthead(cp);
111      (*o*) 79: uplevel;
112      (*q*) 81: with cp^ do
113            (* sets underline mod *)
114            begin
115              i5:= status;
116              if (i5=0) or (i5=2) or (i5=4) then status:= i5 + 1;
117              rowhead[font]:= chr(ord(rowhead[font]) + 128);
118            end;
119      (*r*) 82: with cp^ do
120            (* resets underline mod *)
121            begin
122              i5:= status;
123              if (i5=1) or (i5=3) or (i5=5) then status:= i5 - 1;
124              rowhead[font]:= chr(ord(rowhead[font]) + 128);
125            end;
126      (*t*) 84: begin
127            (* sends current information deep to the host computer *)
128            write(dator,chr(27));      (*esc*)
129            write(dator,chr(91));      (* [ *)
130            write(dator,chr(infodeep + 48));      (*par*)
131            write(dator,chr(97));      (* a *)
132            end;
133      (*u*) 85: setpool;
134            69,70,71,72,73,75,76,80,83,86,87,88,89,90: ;
135            end;
136      end;
137
138      3: begin
139            (* checks if it is a parameter *)
140            if (47 < i) and (i < 58) then

```

```

141     par:= 10 * par + i - 48
142     (* checks if it is a proper control sequense *)
143     else if (i > 62) and (i < 101) then
144         begin
145             val:= 1;
146             if par > 255 then par:= 255;
147             if par < 0 then par:= 0;
148             case i of
149 (*7*) 63: begin
150             if par > 63 then par:= 63;
151             hzoom(par);
152             end;
153 (*d*) 100: opvertabsolute(par);
154 (*a*) 65: opup(par);
155 (*b*) 66: opdown(par);
156 (*c*) 67: opright(par);
157 (*d*) 68: opleft(par);
158 (*g*) 71: begin
159     (* cursor horizontal absolute *)
160     opkol:= par;
161     if opkol > maxchar then opkol:= maxchar;
162     end;
163 (*h*) 72: ophome;
164 (*k*) 75: with op^ do
165     (* erase in line *)
166     begin
167     if par = 0 then
168     for ii:= opkol to maxchar do string[ii]:= blank
169     else if par = 1 then
170     for ii:= 0 to opkol - 1 do string[ii]:= blank
171     else if par = 2 then
172     for ii:= 0 to maxchar do string[ii]:= blank;
173     end;
174 (*a*) 64: insertchar(par);
175 (*p*) 80: delchar(par);
176 (*l*) 76: begin
177     (* inserts new rows before the cursor and
178     sets the cursor on the new row *)
179     while par > 0 do
180     begin
181     par:= par - 1;
182     p:= getrow;
183     inbefore(p,op);
184     reverself;
185     end;
186     klar:= true;
187     info:= true;
188     end;
189 (*y*) 89: begin
190     (* inserts new rows after the cursor and
191     sets the cursor on the new row *)
192     while par > 0 do
193     begin
194     par:= par - 1;
195     p:= getrow;
196     inafter(p,op);
197     linefeed;
198     end;
199     klar:= true;
200     info:= true;

```

```

201         end;
202 (*m*) 77: begin
203         (* deletes the cursor row and sets the cursor
204            on the next row. if the list becomes empty
205            the listhead is deleted *)
206         for i2:= 1 to par do
207             if cp^.rownr = 0 then
208                 begin
209                     pi:= cp;
210                     pi:= headover;
211                     linefeed;
212                     if middle = p then middle:= cp;
213                     delete(p);
214                     pushpool(p);
215                     while pi^.rownr = 0 do
216                         begin
217                             pi:= pi;
218                             pi:= pi^.hover;
219                             delete(p);
220                             pushpool(p);
221                         end;
222                     end;
223                 headover:= cp^.hover;
224                 klar:= true;
225                 infoz:= true;
226                 end;
227         (* vertical scroll *)
228 (*s*) 83: vertscr:= vertscr + par;
229 (*t*) 84: vertscr:= vertscr - par;
230         (* sets new zoom level *)
231 (*u*) 85: begin
232         newzlev:= newzlev + par;
233         if newzlev > 255 then newzlev:= 255;
234         end;
235 (*v*) 86: begin
236         newzlev:= newzlev - par;
237         if newzlev < 0 then newzlev:= 0;
238         end;
239 (*^*) 94: newzlev:= par;
240 (*[*] 91: if par < 16 then
241         (* sets absolute minfont *)
242         begin
243             cp^.rowhead[minfont]:= chr(par);
244             cp^.status:= 2;
245             klar:= true;
246         end;
247 (*]*) 93: begin
248         (* sets relative minfont *)
249         cp^.rowhead[minfont]:= chr(par);
250         cp^.status:= 4;
251         klar:= true;
252         end;
253 (*\*) 92: if par < 7 then
254         cp^.status:= par;
255 (*_*) 95: begin
256         infodeep:= par;
257         subzoom:= 0;
258         infolev:= infodeep * 20;
259         setmark:= false;
260         infoz:= true;

```

```
261         klar:= true;
262         end;
263         69,70,73,74,78,79,81,82,87,88,90,96,97,98,99: val:= 1;
264         end;
265         end
266         else
267         val:= 1;
268         end;
269         end; (* end case *)
270         cursorkol(cpkol);
271         with dummy do
272         begin
273         dpoint:= cp;
274         setcursor(dint + 4);
275         end;
276         end; (* end host while *)
277
278 (* from keyboard *)
279
280 while keyboard do
281     begin
282     (* reads characters from keyboard *)
283     read(ch);
284     write(dator,ch);
285     end;
286
287 (* zoom scroll *)
288
289 i1:= getport(port50) - 128;
290 i4:= abs(i1);
291 vertscr:= vertscr + (i4 * i1) div 512;
292
293 i2:= getport(port60);
294 if newzlev <> 0 then i2:= newzlev
295 else if i2 < 2 then i2:= zoomlev;
296
297 i3:= getport(port70) - 128;
298 if i3 < -125 then i3:= 0;
299 infochange:= i3 div 32;
300 if infochange <> 0 then
301     begin
302     newtreevar;
303     infoz:= true;
304     klar:= true;
305     end;
306
307 i4:= abs(i2 - zoomlev);
308 if (i4 > 2) then
309     begin
310     zoomlev:= i2;
311     newzvar;
312     klar:= true;
313     end;
314 if (vertscr <> 0) or klar then
315     begin
316     if infoz then
317         begin
318         checklevel;
319         infoz:= false;
320         end;
```

```
321     newstartadr;
322     changefont;
323     zscr:= true;
324     vertscr:= 0;
325     end;
326
327     with dummy do
328     begin
329     dpoint:= op;
330     setcursor(dint + 4);
331     cursorkol(opkol);
332     end;
333     end;
334 end.
335
```

## APPENDIX B

Nedan följer en tabell med information om alla fonterna, samt en listning av dem. För betydelsen av block och blockadress hänvisas till referens 1 avsnitt teckenminne. Fonterna finns lagrade på en floppydisk för LSI-11 dator.



## FONT TABELL

font	matrix		font		antal	block
	width	height	width	height	block	adress
0	5	3	6	4	4	2
1	5	4	6	5	5	6
2	5	5	6	6	6	11
3	5	6	6	7	7	17
4	5	7	6	8	8	24
5	5	8	6	10	10	32
6	5	9	7	12	12	42
7	5	11	7	14	14	54
8	5	7	7	8	8	68
9	6	8	8	9	9	76
10	7	9	9	11	11	85
11	8	11	10	13	13+1	96
12	9	13	11	16	15*2	110
13	10	15	12	19	17*2	140
14	11	17	14	22	19*2	174
15	13	20	15	26	22*2	212

FONT0

matrixwidth= 5  
matrixheight= 3

!	"	#	\$	%	&	'	(	)
x x x	x x	xx xxxx xx	xx xxxx xx	x x x x x	xx xxxx xxx	x	x x x	x x x
*	+	,	-	.	/	0	1	2
x x xx x x	x xxx x	x	xxx	x	x x	xx xx x xx	x x x	xxx xx xxxx
3	4	5	6	7	8	9	:	;
xxxx xx xxxx	x x xxxx x	xxxx xxxx xxx	x xxx xxx	xxxx x x	xx xxxx xxxx	xxx xxx x	x x	x x
<	=	>	?	@	A	B	C	D
x x x	xxxx	x x	xxx xx x	xx x xx xxx	xx xxxx x x	xxxx xxx xxxx	xxx x xxx	xxx x x xxx
E	F	G	H	I	J	K	L	M
xxxx xxx xxxx	xxxx xxx x	xxx x xx xxx	x x xxxx x x	x x x	x x x	x xx xx x xx	x x xxxx	x x xx xx x x x
N	O	P	Q	R	S	T	U	V
xx x x xx x x	xx x x xx	xxx xxxx x	xx x xx xxx	xxx xxxx x x	xx xxxx xx	xxx x x	x x x x xxxx	x x xx x
W	X	Y	Z	[	\	]	^	_
x x x xx xx x x	x x xx x x	x x x x	xxxx xx xxxx	xxx x xxx	x x x	xxx x xxx	x x x	xxx
o	a	b	c	d	e	f	g	h
x	xx xxxx xxx	x xxxx xxx	xxx x xxx	x xxxx xxx	xx xxxx xxx	x x x	xxx x xx xxx	x xxx x x
i	j	k	l	m	n	o	p	q
x x x	x x x	x xx xx x xx	x x x	xx x x x x x x x	xxx x x x x	xx x x xx	xxx xxxx x	xxx xxxx x

r	s	t	u	v	w	x	y	z
xx	xx	x	x x	x x	x x	x x	x x	xxxx
x	xxxx	x	x x	xx	x x x	xx	xxx	xx
x	xx	x	xxxx	x	x x	x x	xx	xxxx
{		}						
xx	x	xx						
xx	x	xx						
xx	x	xx						

FONT1

matrixwidth= 5  
matrixheight= 4

!	"	#	\$	%	&	'	(	)
x x	x x x x	xx xxxx xxxx xx	xx xxx x xx	x x x x x x	xx xx x x x xx x	x x	x x x x	x x x x
*	+	,	-	.	/	0	1	2
x x xx xx x x	x x xxx x	x x	xxx	x	x x x x	xx x xx xx x xx	x x x x	xxx x xx xxxx
3	4	5	6	7	8	9	:	;
xxxx xx x xxx	x x x x xxxx x	xxxx xxx x xxx	x x xxx xxxx	xxxx x x x	xx xx x x xx	xxxx xxx x x	x x x x	x x x x
<	=	>	?	@	A	B	C	D
x x x x	xxxx xxxx	x x x	xxx x xx x	xx x x x xx xxxx	x xx xxxx x x	xxxx xxx x x xxx	xxx x x xxx	xxx x x x x xxx
E	F	G	H	I	J	K	L	M
xxxx xxx x xxxx	xxxx x xxx x	xxx x x xx xxx	x x xxxx x x x x	x x x x	x x x xx	x x x x xxx x x	x x x xxxx	x x xx xx x x x x x
N	O	P	Q	R	S	T	U	V
x x xx x x xx x x	xx x x x x xx	xxx x x xxx x	xx x x x xx xxx	xxx x x xxx x x	xx xxx x xx	xxx x x x	x x x x x x xxxx	x x x x xx xx
W	X	Y	Z	[	\	]	^	_
x x x x x xx xx x x	x x xx xx x x	x x xx x x	xxxx x x xxxx	xxx x x xxx	x x x x	xxx x x xxx	x x x x x	xxxx
°	a	b	c	d	e	f	g	h
x x	xx xxx x x xxx	x xxx x x xxx	xxx x x xxx	x xxx x x xxx	xx xxxx x xx	x x x x	xxx x x xx xxx	x xxx x x x x

i	j	k	l	m	n	o	p	q
x	x	x	x	xx x	xxx	xx	xxx	xxx
x	x	x xx	x	x x x	x x	x x	x x	x x
x	x	xx	x	x x x	x x	x x	xxx	xxx
x	x	x xx	x	x x x	x x	xx	x	x
r	s	t	u	v	w	x	y	z
x x	xx	x	x x	x x	x x	x x	x x	xxxx
xx	xxx	xx	x x	x x	x x x	xx	x x	x
x	x	x	x x	xx	x x x	xx	xxx	x
x	xx	x	xxx	xx	x x	x x	xx	xxxx
{		}						
xx	x	xx						
xx	x	xx						
x	x	x						
xx	x	xx						

FONT2

matrixwidth= 5  
matrixheight= 5

!	"	#	\$	%	&	'	(	)
x	x x	x x	xxx	xx x	x	x	x	x
x	x x	xxxxx	x x	x x	x x	x	x	x
x		x x	xxx	x	x x		x	x
		xxxxx	x x	x x	x x		x	x
x		x x	xxx	x xx	xxx x			
*	+	,	-	.	/	0	1	2
x	x				x	xxx	x	xxx
x x x	x				x	x xx	xx	x x
xxx	xxxxx		xxxxx		x	x x x	x	xx
x x x	x	x			x	xx x	x	x
x	x	x		x	x	xxx	x	xxxxx
3	4	5	6	7	8	9	:	;
xxxxx	x x	xxxxx	x	xxxxx	xxx	xxx		
x	x x	x	x	x	x x	x x	x	x
xx	xxxxx	xxxx	xxx	x	xxx	xxx		
x	x	x	x x	x	x x	x		x
xxxx	x	xxxx	xxx	x	xxx	x	x	x
<	=	>	?	@	A	B	C	D
x		x	xxx	xxx	x	xxxx	xxxx	xxxx
x	xxxxx	x	x x	x x	x x	x x	x	x x
x		x	x	x xxx	x x	xxxx	x	x x
x		x	x	x	xxxxx	x x	x	x x
x	xxxxx	x	x	xxxx	x x	xxxx	xxxx	xxxx
E	F	G	H	I	J	K	L	M
xxxxx	xxxxx	xxxx	x x	xxx	x	x xx	x	x x
x	x	x	x x	x	x	x x	x	xx xx
xxxx	xxxx	x xx	xxxxx	x	x	xx	x	x x x
x	x	x x	x x	x	x x	x x	x	x x
xxxxx	x	xxxx	x x	xxx	xxx	x xx	xxxxx	x x
N	O	P	Q	R	S	T	U	V
x x	xxx	xxxx	xxx	xxxx	xxx	xxxxx	x x	x x
xx x	x x	x x	x x	x x	x	x	x x	x x
x x x	x x	xxxx	x x x	xxxx	xxx	x	x x	x x
x xx	x x	x	x x	x x	x	x	x x	x x
x x	xxx	x	xx x	x x	xxx	x	xxx	x
W	X	Y	Z	[	\	]	^	_
x x	x x	x x	xxxxx	xxxx	x	xxxx		
x x	x x	x x	x	x	x	x	x	
x x x	x	x	x	x	x	x	x x	
xx xx	x x	x	x	x	x	x	x x	
x x	x x	x	xxxxx	xxxx	x	xxxx		xxxxx

°	a	b	c	d	e	f	g	h
x	xxx	x	xxxx	x	xxx	x	xxxx	x
x	x	xxxx	x	xxxx	x x	x	x	x
	xxxx	x x	x	x x	xxxxx	xxx	x xx	xxxx
	x x	x x	x	x x	x	x	x x	x x
	xxxx	xxxx	xxxx	xxxx	xxx	x	xxxx	x x
i	J	k	l	m	n	o	p	q
x	x	x xx	x	xx x	x xx	xxx	xxxx	xxxx
	x	x x	x	x x x	xx x	x x	x x	x x
x	x	xx	x	x x x	x x	x x	xxxx	xxxx
x	x	x x	x	x x x	x x	x x	x	x
x	x	x xx	x	x x x	x x	xxx	x	x
r	s	t	u	v	w	x	y	z
x x	xxx	x	x x	x x	x x	x x	x x	xxxxx
xx	x	xxx	x x	x x	x x	x x	x x	x
x	xxx	x	x x	x x	x x x	x	xxxx	x
x	x	x	x x	x x	x x x	x x	x	x
x	xxx	x	xxxx	x	x x	x x	xxx	xxxxx
{	!	}						
xx	x	xx						
x	x	x						
x		x						
x	x	x						
xx	x	xx						

FONT3

matrixwidth= 5  
matrixheight= 6

!	"	#	\$	%	&	'	(	)
x	x x	x x	x	xx x	x	x	x	x
x	x x	x x	xxx	x x	x x	x	x	x
x		xxxxx	x x	x	x		x	x
x		x x	xxx	x	x x x		x	x
		xxxxx	x x	x x	x x		x	x
x		x x	xxx	x xx	xxx x		x	x
*	+	,	-	.	/	0	1	2
x					x	xxx	x	xxx
x x x	x				x	x x	xx	x x
xxx	x				x	x xx	x	x
xxx	xxxxx		xxxxx		x	x x x	x	xx
x x x	x	x			x	xx x	x	x
x	x	x		x	x	xxx	xxx	xxxxx
3	4	5	6	7	8	9	:	;
xxxxx	x x	xxxxx	x	xxxxx	xxx	xxx		
x	x x	x	x	x	x x	x x	x	
xx	x x	xxxx	xxx	x	xxx	x x		x
x	xxxxx	x	x x	x	x x	xxx		
x x	x	x	x x	x	x x	x	x	x
xxx	x	xxxx	xxx	x	xxx	x		x
<	=	>	?	@	A	B	C	D
x		x	xxx	xxx	x	xxxx	xxxx	xxxx
x	xxxxx	x	x x	x x	x x	x x	x	x x
x		x	x	x x x	x x	xxxx	x	x x
x	xxxxx	x	x	x xxx	xxxxx	x x	x	x x
x		x	x	xxxxx	x x	xxxx	xxxx	xxxx
E	F	G	H	I	J	K	L	M
xxxxx	xxxxx	xxxx	x x	xxx	x	x x	x	x x
x	x	x	x x	x	x	x x	x	xx xx
xxxx	x	x	xxxxx	x	x	x x	x	x x x
x	xxxx	x xx	x x	x	x	xxx	x	x x x
x	x	x x	x x	x	x x	x x	x	x x
xxxxx	x	xxxx	x x	xxx	xx	x x	xxxxx	x x
N	O	P	Q	R	S	T	U	V
x x	xxx	xxxx	xxx	xxxx	xxx	xxxxx	x x	x x
xx x	x x	x x	x x	x x	x	x	x x	x x
x x x	x x	x x	x x	x x	xxx	x	x x	x x
x x x	x x	xxxx	x x x	xxxx	x	x	x x	x x
x xx	x x	x	x x	x x	x	x	x x	x
x x	xxx	x	xx x	x x	xxx	x	xxx	x



W	X	Y	Z	[	\	]	^	_
x x	x x	x x	xxxxx	xxxx	x	xxxx		
x x	x x	x x	x	x	x	x		
x x x	x	x x	x	x	x	x	x	
x x x	x	x	x	x	x	x	x x	
xx xx	x x	x	x	x	x	x	x x	
x x	x x	x	xxxxx	xxxx		xxxx		xxxxx
°	a	b	c	d	e	f	g	h
x		x		x		x		x
x	xxx	x	xxxx	x	xxx	x	xxxx	x
	x	xxxx	x	xxxx	x x	x	x	xxxx
	xxxx	x x	x	x x	xxxxx	xxx	x xx	x x
	x x	x x	x	x x	x	x	x x	x x
	xxxx	xxxx	xxxx	xxxx	xxx	x	xxxx	x x
i	j	k	l	m	n	o	p	q
x	x	x	xx					
		x xx	x	xx x	x xx	xxx	xxxx	xxxx
xx	x	x x	x	x x x	xx x	x x	x x	x x
x	x	xx	x	x x x	x x	x x	xxxx	xxxx
x	x	x x	x	x x x	x x	x x	x	x
xxx	xx	x xx	xxx	x x x	x x	xxx	x	x
r	s	t	u	v	w	x	y	z
		x						
x x	xxx	xxx	x x	x x	x x	x x	x x	xxxxx
xx	x	x	x x	x x	x x	x x	x x	x
x	xxx	x	x x	x x	x x x	x	xxxx	x
x	x	x	x x	x x	x x x	x x	x	x
x	xxx	x	xxxx	x	x x	x x	xxx	xxxxx
€	!	}						
xx	x	xx						
x	x	x						
x		x						
x	x	x						
xx	x	xx						

FONT4

matrixwidth= 5  
matrixheight= 7

!	"	#	\$	%	&	'	(	)
X	X X	X X	X	XX X	X	X	X	X
X	X X	X X	XXX	XX X	X X	X	X	X
X	X X	XXXXX	X X	X	X X	X	X	X
X		X X	XXX	X	X X		X	X
		XXXXX	X X	X	X X X		X	X
		X X	XXX	X XX	X X		X	X
X		X X	X	X XX	XX X		X	X
*	+	,	-	.	/	0	1	2
X					X	XXX	X	XXX
X X X	X				X	X X	XX	X X
XXX	X				X	X XX	X	X
X	XXXXX		XXXXX		X	X X X	X	X
XXX	X	X			X	XX X	X	X
X X X	X	X			X	X X	X	X
X		X		X	X	XXX	XXX	XXXXX
3	4	5	6	7	8	9	:	;
XXXXX	X X	XXXXX	XX	XXXXX	XXX	XXX		
X	X X	X	X	X	X X	X X		
X	X X	XXXX	X	X	X X	X X	X	X
XX	XXXXX	X	XXXX	X	XXX	XXXX		
X	X	X	X X	X	X X	X		X
X X	X	X X	X X	X	X X	X	X	X
XXX	X	XXX	XXX	X	XXX	XX		X
<	=	>	?	@	A	B	C	D
X		X	XXX	XXX	X	XXXX	XXX	XXXX
X		X	X X	X X	X X	X X	X X	X X
X	XXXXX	X	X	X X X	X X	X X	X	X X
X		X	X	X XXX	X X	XXXX	X	X X
X	XXXXX	X	X	X	XXXXX	X X	X X	X X
X		X	X	X	X X	X X	X X	X X
X		X	X	XXXX	X X	XXXX	XXX	XXXX
E	F	G	H	I	J	K	L	M
XXXXX	XXXXX	XXXX	X X	XXX	X	X X	X	X X
X	X	X	X X	X	X	X X	X	XX XX
X	X	X	X X	X	X	X X	X	X X X
XXXX	XXXX	X XX	XXXXX	X	X	XX	X	X X X
X	X	X X	X X	X	X	X X	X	X X
X	X	X X	X X	X	X X	X X	X	X X
XXXXX	X	XXXX	X X	XXX	XX	X X	XXXXX	X X

N	O	P	Q	R	S	T	U	V
x x	xxx	xxxxx	xxx	xxxx	xxx	xxxxxx	x x	x x
xx x	x x	x x	x x	x x	x x	x	x x	x x
x x x	x x	x x	x x	x x	x	x	x x	x x
x x x	x x	xxxxx	x x	xxxx	xxx	x	x x	x x
x xx	x x	x	x x x	x x	x	x	x x	x x
x x	x x	x	x x	x x	x x	x	x x	x
x x	xxx	x	xx x	x x	xxx	x	xxx	x

W	X	Y	Z	[	\	]	^	_
x x	x x	x x	xxxxxx	xxxx	x	xxxx		
x x	x x	x x	x	x	x	x		
x x	x x	x x	x	x	x	x	x	
x x x	x	x	x	x	x	x	x x	
x x x	x x	x	x	x	x	x	x x	
xx xx	x x	x	x	x	x	x		
x x	x x	x	xxxxxx	xxxx	x	xxxx		xxxxxx

°	a	b	c	d	e	f	g	h
x		x		x		x		x
x	xxx	x	xxxxx	x	xxx	x	xxxx	x
x	x	xxxxx	x	xxxxx	x x	x	x	xxxxx
	xxxxx	x x	x	x x	xxxxxx	xxx	x	x x
	x x	x x	x	x x	x	x	x xx	x x
	x x	x x	x	x x	x	x	x x	x x
	xxxxx	xxxxx	xxxxx	xxxxx	xxx	x	xxxxx	x x

i	j	k	l	m	n	o	p	q
x	x	x	xx					
		x xx	x	xx x	x xx	xxx	xxxxx	xxxxx
xx	x	x x	x	x x x	xx x	x x	x x	x x
x	x	xx	x	x x x	x x	x x	x x	x x
x	x	x x	x	x x x	x x	x x	xxxxx	xxxxx
xxx	xx	x x	xxx	x x x	x x	xxx	x	x

r	s	t	u	v	w	x	y	z
		x						
x xx	xxx	xxx	x x	x x	x x	x x	x x	xxxxxx
xx	x	x	x x	x x	x x	x x	x x	x
x	xxx	x	x x	x x	x x	x	xxxxx	x
x	x	x	x x	x x	x x x	x	x	x
x	x	x	x x	x	x x x	x x	x	x
x	xxx	x	xxxxx	x	x x	x x	xxx	xxxxxx

{		}
x	x	x
x	x	x
x	x	x
x		x
x	x	x
x	x	x
x	x	x

FONT5

matrixwidth= 5  
matrixheight= 8

!	"	#	\$	%	&	'	(	)
x	x x	x x	x	xx	x	x	x	x
x	x x	x x	xxx	xx x	x x	x	x	x
x	x x	xxxxx	x x	x	x x	x	x	x
x		x x	xxx	x	x		x	x
x		x x	x x	x	x x x		x	x
		xxxxx	x x	x	x x		x	x
		x x	xxx	x xx	x x		x	x
x		x x	x	x xx	xx x		x	x
-----								
*	+	,	-	.	/	0	1	2
x					x	xxx	x	xxx
x	x				x	x x	xx	x x
x x x	x				x	x xx	x	x
xxx	x				x	x x x	x	x
x	xxxxx		xxxxx		x	x x x	x	x
xxx	x	x			x	xx x	x	x
x x x	x	x			x	x x	x	x
x	x	x		x	x	xxx	xxx	xxxxx
-----								
3	4	5	6	7	8	9	:	;
xxxxx	x x	xxxxx	xx	xxxxx	xxx	xxx		
x	x x	x	x	x	x x	x x		
x	x x	x	x	x	x x	x x	x	x
xx	x x	xxxxx	xxxxx	x	xxx	x x		
x	xxxxx	x	x x	x	x x	xxxxx		
x	x	x	x x	x	x x	x	x	x
x x	x	x x	x x	x	x x	x		x
xxx	x	xxx	xxx	x	xxx	xx		x
-----								
<	=	>	?	@	A	B	C	D
x		x	xxx	xxx	x	xxxx	xxx	xxxx
x		x	x x	x x	x x	x x	x x	x x
x	xxxxx	x	x	x x	x x	x x	x	x x
x		x	x	x xxx	x x	xxxx	x	x x
x	xxxxx	x	x	x xxx	xxxxx	x x	x	x x
x		x	x	x	x x	x x	x	x x
x		x	x	x	x x	x x	x x	x x
x		x	x	xxxx	x x	xxxx	xxx	xxxx
-----								
E	F	G	H	I	J	K	L	M
xxxxx	xxxxx	xxxx	x x	xxx	x	x x	x	x x
x	x	x	x x	x	x	x x	x	xx xx
x	x	x	x x	x	x	x x	x	x x x
xxxx	xxxx	x	xxxxx	x	x	x x	x	x x x
x	x	x xx	x x	x	x	xx	x	x x
x	x	x x	x x	x	x	x x	x	x x
x	x	x x	x x	x	x x	x x	x	x x
xxxxx	x	xxxx	x x	xxx	xx	x x	xxxxx	x x
-----								

N	O	P	Q	R	S	T	U	V
x x	xxx	xxxx	xxx	xxxx	xxx	xxxxx	x x	x x
x x	x x	x x	x x	x x	x x	x	x x	x x
xx x	x x	x x	x x	x x	x	x	x x	x x
x x x	x x	xxxx	x x x	xxxx	xxx	x	x x	x x
x xx	x x	x	x x	x x	x	x	x x	x x
x x	x x	x	x x	x x	x x	x	x x	x
x x	xxx	x	xx x	x x	xxx	x	xxx	x
W	X	Y	Z	[	\	]	^	_
x x	x x	x x	xxxxx	xxxx	x	xxxx		
x x	x x	x x	x	x	x	x		
x x	x x	x x	x	x	x	x	x	
x x	x	x x	x	x	x	x	x x	
x x x	x	x	x	x	x	x	x x	
x x x	x x	x	x	x	x	x	x x	
xx xx	x x	x	x	x	x	x		
x x	x x	x	xxxxx	xxxx	x	xxxx		xxxxx
°	a	b	c	d	e	f	g	h
x		x		x		x		x
x		x		x		x		x
x	xxx	x	xxxx	x	xxx	x	xxxx	x
	x	xxxx	x	xxxx	x x	x	x	xxxx
	xxxx	x x	x	x x	xxxxx	xxx	x	x x
	x x	x x	x	x x	x	x	x x	x x
	x x	x x	x	x x	x	x	x x	x x
	xxxx	xxxx	xxxx	xxxx	xxx	x	xxxx	x x
i	j	k	l	m	n	o	p	q
x	x	x	xx					
		x	x					
		x xx	x	xx x	x xx	xxx	xxxx	xxxx
xx	x	x x	x	x x x	xx x	x x	x x	x x
x	x	xx	x	x x x	x x	x x	x x	x x
x	x	x x	x	x x x	x x	x x	xxxx	xxxx
x	x	x x	x	x x x	x x	x x	x	x
xxx	xx	x x	xxx	x x x	x x	xxx	x	x
r	s	t	u	v	w	x	y	z
		x						
x xx	xxx	xxx	x x	x x	x x	x x	x x	xxxxx
xx	x	x	x x	x x	x x	x x	x x	x
x	xxx	x	x x	x x	x x	x	xxxx	x
x	x	x	x x	x x	x x x	x	x	x
x	x	x	x x	x	x x x	x x	x	x
x	xxx	x	xxxx	x	x x	x x	xxx	xxxxx

<u>l</u>	<u>i</u>	<u>j</u>
x	x	x
x	x	x
x	x	x
x		x
x		x
x	x	x
x	x	x
x	x	x

FONT6

matrixwidth= 5  
matrixheight= 9

!	"	#	\$	%	&	'	(	)
x	x x	x x	x	xx	x	x	x	x
x	x x	x x	xxx	x	x x	x	x	x
x	x x	xxxxxx	x x	xx x	x x	x	x	x
x		x x	x x	x	x x		x	x
x		x x	xxx	x	x		x	x
		x x	x x	x	x x x		x	x
		xxxxxx	x x	x xx	x x		x	x
		x x	xxx	x	x x		x	x
x		x x	x	xx	xx x		x	x
*	+	,	-	.	/	0	1	2
x					x	xxx	x	xxx
x	x				x	x x	x	x x
x x x	x				x	x xx	xx	x x
xxx	x				x	x x x	x	x
x	xxxxxx		xxxxxx		x	x x x	x	x
xxx	x				x	x x x	x	x
x x x	x	x			x	xx x	x	x
x	x	x			x	x x	x	x
x		x		x	x	xxx	xxx	xxxxxx
3	4	5	6	7	8	9	:	;
xxxxxx	xx	xxxxxx	xx	xxxxxx	xxx	xxx		
x	x x	x	x	x	x x	x x		
x	x x	x	x	x	x x	x x	x	x
x	x x	xxxx	x	x	x x	x x		
xx	x x	x	xxxx	x	xxx	xxxx		
x	xxxxxx	x	x x	x	x x	x		
x	x	x	x x	x	x x	x	x	x
x x	x	x x	x x	x	x x	x		x
xxx	x	xxx	xxx	x	xxx	xx		x
<	=	>	?	@	A	B	C	D
x		x	xxx	xxx	x	xxxx	xxx	xxx
x		x	x x	x x	x x	x x	x x	x x
x	xxxxxx	x	x	x x	x x	x x	x	x x
x		x	x	x x x	x x	xxxx	x	x x
x		x	x	x xx	xxxxxx	x x	x	x x
x	xxxxxx	x		x	x x	x x	x	x x
x		x		x	x x	x x	x x	x x
x		x	x	xxxx	x x	xxxx	xxx	xxx

E	F	G	H	I	J	K	L	M
XXXXXX X X X XXXX X X X XXXXXX	XXXXXX X X X XXXX X X X X	XXX X X X XX X X X XXX	X X X X X X XXXXX X X X X X X X X	XXX X X X X X X XXX	X X X X X X X X X	X X X X X X XX X X X X X X X X	X X X X X X XXXXX	X X XX XX X X X X X X X X X X X X X
N	O	P	Q	R	S	T	U	V
X X X X XX X X X X X X X X XX X X X X	XXX X X X X X X X X X X X X XXX	XXXX X X X X X X XXXX X X X	XXX X X X X X X X X X X XX X	XXXX X X X X X X XXXX X X X X X X	XXX X X X X XXX X X XXX	XXXXXX X X X X X X X	X X X X X X X X X X X X X X XXX	X X X X X X X X X X X X X X
W	X	Y	Z	[	\	]	^	_
X X X X X X X X X X X X X X XX XX X X	X X X X X X X X X X X X X X X	X X X X X X X X X X X X X	XXXXX X X X X X X XXXXX	XXXX X X X X X X XXXX	X X X X X X X X	XXXX X X X X X X XXXX	X X X X X X X X	XXXXX
°	a	b	c	d	e	f	g	h
X X X	XXX X XXXX X X X X XXXX	X X XXXX X X X X XXXX	X XXXX X XXXX	X XXXX X X X X X X XXXX	X XXXX X X XXXX X XXXX	X X X XXXX X X	X XXXX X X X X X XXXX	X XXXX X X X X X XXXX
i	j	k	l	m	n	o	p	q
X  XX X X X XXX	X  X X X X XX	X X X X X X X X X X	XX X X X X XXX	XX X X X X X X X X X X X X X	X XX XX X X X X X X X	XXX X X X X X X X X XXX	XXXX X X X X X X X X X	XXXX X X XXXX XXXX X





FONT7

matrixwidth= 5  
matrixheight= 11

!	"	#	\$	%	&	'	(	)
x	x x	x x	x	xx	x	x	x	x
x	x x	x x	xxx		x x	x	x	x
x	x x	x x	x x	xx x	x x	x	x	x
x	x x	xxxxx	x x		x x	x	x	x
x		x x	x x		x		x	x
x		x x	xxx		x		x	x
x		x x	x x		x x x		x	x
		xxxxx	x x	x	x x x		x	x
		x x	x x	x xx	x x		x	x
		x x	xxx		x x		x	x
x		x x	x	xx	xx x		x	x
*	+	,	-	.	/	0	1	2
x						xxx	x	xxx
x					x	x x	x	x x
x x x	x				x	x x	xx	x x
x x x	x				x	x xx	x	x
xxx	x				x	x x x	x	x
x	xxxxxx		xxxxxx		x	x x x	x	x
xxx	x				x	x x x	x	x
x x x	x	x			x	xx x	x	x
x x x	x	x			x	x x	x	x
x		x			x	x x	x	x
x		x		x		xxx	xxx	xxxxx
3	4	5	6	7	8	9	:	;
xxxxx	x	xxxxx	xx	xxxxx	xxx	xxx		
x	xx	x	x	x	x x	x x		
x	x x	x	x	x	x x	x x		
x	x x	x	x	x	x x	x x	x	x
xx	x x	xxxx	x	x	x x	x x		
x	xxxxxx	x	xxxx	x	xxx	xxxx		
x	x	x	x x	x	x x	x	x	x
x	x	x	x x	x	x x	x		
x	x	x	x x	x	x x	x		
x x	x	x x	x x	x	x x	x		
xxx	x	xxx	xxx	x	xxx	xx		x
<	=	>	?	@	A	B	C	D
x		x	xxx	xxx	x	xxxx	xxx	xxx
x		x	x x	x x	x x	x x	x x	x x
x		x	x x	x x	x x	x x	x x	x x
x	xxxxxx	x	x	x x	x x	x x	x	x x
x		x	x	x xxx	x x	xxxx	x	x x
x		x	x	x x x	xxxxxx	x x	x	x x
x	xxxxxx	x	x	x xx	x x	x x	x	x x
x		x		x	x x	x x	x x	x x
x		x		x	x x	x x	x x	x x
x		x	x	xxxx	x x	xxxx	xxx	xxx

E	F	G	H	I	J	K	L	M
XXXXXX X X X XXXXX X X X XXXXXX	XXXXXX X X X XXXXX X X X X X	XXX X X X X XX X X X XXX	X X X X X X X X XXXXXX X X X X X X X X X X	XXX X X X X X X XXX	X X X X X X X XX	X X X X X X X X XX X X X X X X X X	X X X X X X XXXXXX	X X X X XX XX X X X X X X X X X X X X X
N	O	P	Q	R	S	T	U	V
X X X X X X XX X X X X X X X X XX X X X X	XXX X X X X X X X X X X X X XXX	XXXX X X X X X X XXXX X X X	XXX X X X X X X X X X X XX X	XXXX X X X X X X XXXX X X X X X X	XXX X X X X X X XXX	XXXXXX X X X X X X X	X X X X X X X X X X X X XXX	X X X X X X X X X X X X X
W	X	Y	Z	[	\	]	^	_
X X X X X X X X X X X X X X XX XX X X X X	X X X X X X X X X X X X X X X	X X X X X X X X X X X X	XXXXXX X X X X X XXXXXX	XXXXXX X X X X X X X XXXXXX	X X X X X X X	XXXXXX X X X X X X XXXXXX	X X X X X X X X X	XXXXXX
o	a	b	c	d	e	f	g	h
X X X X	XXX X XXXX X X X X X X XXXX	X XXXX X X X X X X X X XXXX	XXXXX X X X X XXXX	X XXXX X X X X X X X X XXXX	X XXXX X X XXXXX X XXXX	X X X X XXXX X	X XXXX X X X X X X XXXX	X X X X XXXX X X X X

i	j	k	l	m	n	o	p	q
x	x	x	xx					
		x	x					
		x	x					
		x x	x	xx x	x xx	xxx	xxxx	xxxx
xx	x	x x	x	x x x	xx x	x x	x x	x x
x	x	x x	x	x x x	x x	x x	x x	x x
x	x	xx	x	x x x	x x	x x	x x	x x
x	x	x x	x	x x x	x x	x x	xxxx	xxxx
x	x	x x	x	x x x	x x	x x	x	x
x	x	x x	x	x x x	x x	x x	x	x
xxx	x	x x	xxx	x x x	x x	xxx	x	x

r	s	t	u	v	w	x	y	z
		x						
		x						
x xx	xxx	xxx	x x	x x	x x	x x	x x	xxxxx
x	x	x	x x	x x	x x	x x	x x	x
xx	x	x	x x	x x	x x	x x	x x	x
x	xxx	x	x x	x x	x x	x	xxxx	x
x	x	x	x x	x x	x x x	x	x	x
x	x	x	x x	x x	x x x	x x	x	x
x	x	x	x x	x	x x	x x	x	x
x	xxx	x	xxxx	x	x x	x x	xxx	xxxxx

{	i	}
x	x	x
x	x	x
x	x	x
x		x
x		x
x	x	x
x	x	x
x	x	x
x	x	x

FONT8

matrixwidth= 5  
matrixheight= 7

!	"	#	\$	%	&	'	(	)
X X X X X	X X X X X X	X X X X XXXXX X X X X	X XXX X X XXX X	XX XX X X X X XX XX	X X X X X X X X X XX X	X X X X X	X X X X X	X X X X X
*	+	,	-	.	/	0	1	2
X X X X XXX X XXX X X X X	X X XXXXX X X	X X	XXXXX	X	X	XXX X X X XX X X X X X XXX	X XX X X X XXX	XXX X X X X X XXXXX
3	4	5	6	7	8	9	:	;
XXXXX X X XX X X X XXX	X XX X X XXXXX X	XXXXX X XXXX X X XXX	XX X XXXX X X XXX	XXXXX X X X X	XXX X X X X XXX X X XXX	XXX X X X X X XX	X X X X XXX	X X X X X
<	=	>	?	@	A	B	C	D
X X X X X	XXXXX	X X X X	XXX X X X X	XXX X X X XXX X XX X XXXX	X X X X X XXXXX X X X X	XXXX X X X X X X X X XXXX	XXX X X X X X XXX	XXXX X X X X X X X X XXXX
E	F	G	H	I	J	K	L	M
XXXXX X X XXXX X X XXXXX	XXXXX X X XXXX X X	XXXX X X X XX X X X X XXXX	X X X X X X XXXXX X X X X	XXX X X X X XXX	X X X X X X XXX	X X X X X X X X X X X	X X X X X XXXXX	X X XX XX X X X X X X X X X X



FONT9

matrixwidth= 6  
matrixheight= 8

!	"	#	\$	%	&	'	(
x x x x x x x	x x x x x x	x x x x xxxxx x x x x xxxxx x x	x xxxx x x xxxx x x x x x	xx xx x x x x x xx xx	xx x x x x x x x x x x x x x xxx x	x x x	x x x x x x
)	*	+	,	-	=	/	0
x x x x x x	x x x x xxx x xxx x x x x	x xxxxx x x	x x x	xxxxxxx		x x x x	xxxx x x x xx x x x xx x x x xxxx
1	2	3	4	5	6	7	8
x xx x x x x xxx	xxxx x x x xx x x xxxxxx	xxxxxxx x x xxx x x x xxxx	x xx x x xxxxxxx x x x	xxxxxxx x x xxxxxx x x x xxxx	xx x x xxxx x x xxxx	xxxxxxx x x x x x x	xxxx x x x x xxxx x x x x xxxx
9	:	;	<	=	>	?	@
xxxx x x x x xxxxx x x xx	x x x x x xxxx	x x x x x	x x x x x	xxxxxxx xxxxxxx	x x x x x	xxxx x x x x x	xxxx x x x x xxxx x x x x xxxx
A	B	C	D	E	F	G	H
xx x x x x x x xxxxxx x x x x x x	xxxxx x x x x xxxxx x x x x xxxx	xxx x x x x x x xxx	xxxx x x x x x x x x xxxx	xxxxxxx x x xxxx x x xxxxxxx	xxxxxxx x x xxxx x x x	xxx x x x x xxx x x x x xxxx	x x x x x x xxxxxx x x x x x x

<b>I</b>	<b>J</b>	<b>K</b>	<b>L</b>	<b>M</b>	<b>N</b>	<b>O</b>	<b>P</b>
xxx x x x x x xxx	xxx x x x x x x xxx	x x x x x x xxx x x x x x x	x x x x x xxxxxx	x x xx xx x xx x x xx x x x x x x x	x x x x xx x x x x x xx x x x x	xxxx x x x x x x x x x x xxxx	xxxxx x x x x x x xxxxx x x
<b>Q</b>	<b>R</b>	<b>S</b>	<b>T</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>
xxxx x x x x x x x x x x x xxx x	xxxxx x x x x xxxxx x x x x x x	xxxx x x xxxx x x x x x xxxx	xxxxx x x x x x x	x x x x x x x x x x x x xxxx	x x x x x x x x x x xx xx	x x x x x x x xx x x xx x xx xx x x	x x x x x x xx xx x x x x
<b>Y</b>	<b>Z</b>	<b>[</b>	<b>\</b>	<b>]</b>	<b>^</b>	<b>_</b>	<b>°</b>
x x x x x x x x xx x x x	xxxxxxx x x x x x xxxxxxx	xxxxxxx xx xx xx xx xx xxxxxxx	 x x x x x x	xxxxxxx xx xx xx xx xx xxxxxxx	   xx x x x x	      xxxxxxx	 x x x
<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>h</b>
   xxxx x xxxxx x x xxxxx	 x x x xxxxx x x x x xxxxx	   xxxxx x x x xxxxx	   xxxxx x x x x x x xxxxx	   xxxx x x xxxxxx x xxxx	   xx x x x xxxx x	   xxxxx x x xxx x x xxxxx	 x x x xxxxx x x x x x x
<b>i</b>	<b>j</b>	<b>k</b>	<b>l</b>	<b>m</b>	<b>n</b>	<b>o</b>	<b>p</b>
   xx x x x xxx	 x x x x xx	 x x x xx x x xxx x x x xx	 xx x x x xxx	   xxx x x x x x x x x x x x x x	   x xxx xx x x x x x x x	   xxxx x x x x x x xxxx	   xxxxx x x xxxxx x x





FONT10

matrixwidth= 7  
matrixheight= 9

!	"	#	\$	%	&	'
X	X X		X	X X	XX	X
X	X X	X X	XXXXX	X X X	X X	X
X	X X	X X	X X X	X	X X	X
X		XXXXXXXX	X X	X	X X	
X		X X	XXXXX	X	X	
X		X X	X X	X	X X X	
		XXXXXXXX	X X X	X	X X X	
		X X	XXXXX	X X X	X XX	
X		X X	X	X X	XXX X	
(	)	*	+	,	-	.
X	X	X				
X	X	X X X	X			
X	X	X X X	X			
X	X	XXX	X			
X	X	X	XXXXXXXX		XXXXXXXX	
X	X	XXX	X			
X	X	X X X	X	X		
X	X	X X X	X	X		
X	X	X		X		X
/	0	1	2	3	4	5
	XXXXX	X	XXXXX	XXXXXXXX	X	XXXXXXXX
X	X X	XX	X X	X	XX	X
X	X XX	XXX	X	X	X X	X
X	X X X	X	X	XX	X X	XXXXX
X	X X X	X	XX	X	X X	X
X	X X X	X	X	X	XXXXXXXX	X
X	XX X	X	X	X	X	X
X	X X	X	X	X X	X	X X
	XXXXX	XXXXX	XXXXXXXX	XXXXX	X	XXXXX
6	7	8	9	:	;	<
XXX	XXXXXXXX	XXXXX	XXXXX			X
X	X	X X	X X			X
X	X	X X	X X			X
X	X	X X	X X	X	X	X
XXXXXXXX	X	XXXXX	XXXXXXXX			X
X X	X	X X	X			X
X X	X	X X	X	X	X	X
X X	X	X X	X		X	X
XXXXX	X	XXXXX	XXX		X	X

```

=          >          ?          @          A          B          C
-----
          X          XXXXX          XXXXX          X          XXXXXX          XXXX
          X          X      X          X      X          X      X          X      X
          X          X          X          X      X          X      X          X
XXXXXXXXX          X          X          X      XXX X          X      X          X
          X          X          X      X      X      X          XXXXXX          X
          X          X          X          X      XX XX          XXXXXXXX          X
XXXXXXXXX          X          X          X          X          XXXXXXXX          X
          X          X          X          X          X      X          X      X          X
          X          X          X          XXXXXX          X      X          XXXXXX          XXXX
-----
D          E          F          G          H          I          J
-----
XXXXXX          XXXXXXXX          XXXXXXXX          XXXX          X      X          XXX          XXX
X      X          X          X          X          X      X          X          X          X
X      X          X          X          X          X          X          X          X          X
X      X          XXXXXX          XXXXXX          X          XXXXXXXXX          X          X
X      X          X          X          X      XXX          X          X          X          X
X      X          X          X          X          X      X          X          X          X
X      X          X          X          X          X      X          X          X          X
XXXXXX          XXXXXXXX          X          XXXX          X      X          XXX          XXX
-----
K          L          M          N          O          P          Q
-----
X      X          X          X          X      X          XXXX          XXXXXXXX          XXXX
X      X          X          XX XX          X      X          X      X          X      X          X
X      X          X          X      X      X          XX          X          X          X          X
X      X          X          X      X      X          X      X          X          X          X
XXXX          X          X      X      X          X      X          X          X          XXXXXXXX          X
X      X          X          X          X          X          X          X          X          X          X
X      X          X          X          X          X          X          X          X          X      X
X      X          X          X          X          X          X      X          X          X          X
X          X          XXXXXXXX          X          X          XXXX          X          XXXX X
-----
R          S          T          U          V          W          X
-----
XXXXXXXXX          XXXX          XXXXXXXXX          X      X          X          X          X          X          X
X          X          X      X          X          X          X          X          X          X          X
X          X          X          X          X          X          X          X          X          X          X
X          X          XXXXXX          XXXXXX          X          X          X          X          X          X          X
XXXXXXXXX          X          X          X          X          X          X          X          X          X          X
X      X          X          X          X          X          X          X          X          X          X
X          X          X          X          X          X          X      X          X          X          X
X          X          X          X          X          X          X      X          XX XX          X          X
X          X          XXXX          X          XXXXXX          X          X          X          X          X
-----
Y          Z          [          \          ]          ^          _
-----
X          X          XXXXXXXX          XXXXXXXXX          XXXXXXXXX          XXXXXXXXX
X          X          X          XX          X          XX          X          XX          X          XX
X          X          X          XX          X          XX          X          XX          X          XX
X      X          X          XX          X          XX          X          XX          X          XX
X          X          X          XX          X          XX          X          XX          X          XX
X          X          X          XX          X          XX          X          XX          X          XX
X          X          XXXXXXXX          XXXXXXXXX          XXXXXXXXX          XXXXXXXXX
-----

```

o	a	b	c	d	e	f
<pre> x   x     x       xxxxxx         x       xxxxxx x      x x      x   xxxxxx           </pre>	<pre> x   x     x       xxxxxx         x       xxxxxx x      x x      x   xxxxxx           </pre>	<pre> x   x     x       xxxxxx         x       xxxxxx x      x x      x   xxxxxx           </pre>	<pre>       xxxxxx x x x   xxxxxx           </pre>	<pre>       xxxxxx x x x   xxxxxx           </pre>	<pre>       xxxxxx x x x   xxxxxx           </pre>	<pre>       xx x x x   xxxxxx           </pre>
g	h	i	j	k	l	m
<pre>       xxxxxx x x x   xxxxxx           </pre>	<pre> x   x     x       xxxxxx         x       xxxxxx x      x x      x x      x   xxxxxx           </pre>	<pre>       xx x x x   xxxxxx           </pre>	<pre>       x x x   xxxxxx           </pre>	<pre> x   xx x   xxxxxx x   xxxxxx           </pre>	<pre>       xxx x x x   xxxxxx           </pre>	<pre>       xxx xx x x x   xxxxxx           </pre>
n	o	p	q	r	s	t
<pre> x xxxxx xx  x x   x x   x x   x x   x           </pre>	<pre>   xxx x  x x   x x   x   xxx           </pre>	<pre> xxxxxxx x   x x   x xxxxxxx x           </pre>	<pre> xxxxxxx x   x x   x xxxxxxx x           </pre>	<pre> x xxx xx x x x           </pre>	<pre> xxxxxx x xxxxxx x xxxxxx           </pre>	<pre>       x xxxxx x x x   xxxxxx           </pre>
u	v	w	x	y	z	{
<pre> x   x x   x x   x x   x x   x xxxxxxx           </pre>	<pre> x   x x   x x   x x   x   xx           </pre>	<pre> x   x x   x x   x x   x x   x           </pre>	<pre> x   x x   x   xx   xxxxxx x   x           </pre>	<pre> x   x x   x x   x xxxxxxx xxxxxxx           </pre>	<pre> xxxxxxx x xxxxxxx           </pre>	<pre>       xx x x x   xxxxxx           </pre>
!	}					
<pre> x   x     x           </pre>	<pre>   xx     x       x         x       xx     x       x         x       xx           </pre>					



<p>9</p> <pre> XXXXXXXXX X      X X      X X      X XXXXXXXXX       X       X      XX     XX   </pre>	<p>:</p> <pre>       XX       XX       XX       XX       XX       XX       XX       XX       XX   </pre>	<p>!</p> <pre>       XX       XX       XX       XX       XX       XX       XX       XX       XX   </pre>	<p>&lt;</p> <pre>       XX       X       X       X       X       X       XX   </pre>	<p>=</p> <pre> XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX   </pre>	<p>&gt;</p> <pre>       XX       X       X       X       X       X       XX   </pre>
<p>?</p> <pre> XXXXXXXXX X      X X      X       X       X      XX     XX   </pre>	<p>@</p> <pre> XXXXXXXXXX X      X X      X X      X X XX XX X X X X X XX X X XXXXXXXXXX   </pre>	<p>A</p> <pre>       XX       X X       X X       X X       X X XXXXXXXXXXXXX       X X       X X       X X       X X   </pre>	<p>B</p> <pre> XXXXXXXXXX X      X X      X X      X XXXXXXXXXX X      X X      X X      X XXXXXXXXXX   </pre>	<p>C</p> <pre>       XX       X X       X X       X X       X X       X X       X X       X X       XX   </pre>	<p>D</p> <pre> XXXXXXXXXX X      X X      X X      X X      X X      X X      X X      X XXXXXXXXXX   </pre>
<p>E</p> <pre> XXXXXXXXXX X X X X XXXXXXXXXX X X X X XXXXXXXXXX   </pre>	<p>F</p> <pre> XXXXXXXXXX X X X X XXXXXXXXXX X X X X   </pre>	<p>G</p> <pre>       XX       X X       X X       X X       X       XXX       X       X       X       XXX   </pre>	<p>H</p> <pre>       X X       X X       X X       X X XXXXXXXXXXXXX X      X X      X X      X X      X X      X   </pre>	<p>I</p> <pre>       XX       X       X       X       X       X       X       XXX   </pre>	<p>J</p> <pre>       XX       X       X       X       X       X       X       X       XXX   </pre>
<p>K</p> <pre>       X X       X X       X X       X X       XXX       X X       X X       X X       X X       X X   </pre>	<p>L</p> <pre>       X       X       X       X       X       X       X       X       X       XXXXXXXXX   </pre>	<p>M</p> <pre>       X X       XX XX       X XX X       X XX X       X       X       X       X       X       X   </pre>	<p>N</p> <pre>       X X       X X       XX X       X X X       X X X       X X X       X XX       X X       X X       X X   </pre>	<p>O</p> <pre>       XXXX       X X       X X       X X       X X       X X       X X       X X       X X       XXXX   </pre>	<p>P</p> <pre> XXXXXXXXXX X      X X      X X      X X      X XXXXXXXXXX X X X X X   </pre>



i	j	k	l	m	n
-----	-----	-----	-----	-----	-----
xx	xx	x	xxx		
		x	x		
		x	x		
		x	x		
xxx	x	x	x	xxxx xx	x xxxx
x	x	x	x	x x x	xx x
x	x	x	x	x x x	x x
x	x	xx	x	x x x	x x
x	x	x	x	x x x	x x
xxxxx	xx	x	xxxxx	x x x	x x

o	p	q	r	s	t
-----	-----	-----	-----	-----	-----

xxxx	xxxxxxxx	xxxxxxxx	x xxx	xxxxxx	x
x x	x x	x x	xx	x x	xxxx
x x	x x	x x	x	x	x
x x	xxxxxxxx	xxxxxxxx	x	xxxxxx	x
x x	x	x	x	x	x
x x	x	x	x	x x	x
xxxx	x	x	x	xxxxxx	xx

u	v	w	x	y	z
-----	-----	-----	-----	-----	-----

x x	x x	x x	x x	x x	xxxxxxxx
x x	x x	x x	x x	x x	x
x x	x x	x x	x x	x x	x
x x	x x	x x	xx	xxxxxx	xx
x x	x x	x xx x	x x	x	x
x x	x x	x x x x	x x	x	x
xxxxxx	xx	x x	x x	xxxxxx	xxxxxxxx

{		}
-----	-----	-----

xx	x	xx
x	x	x
x	x	x
x		x
xx		xx
x		x
x	x	x
x	x	x
xx	x	xx



FONT12

matrixwidth= 9  
matrixheight= 13

!	"	#	\$	%	&
XX XX XX XX XX XX XX XX XX	X X X X X X X X X X	X X X X X X XXXXXXXXXX X X X X X X XXXXXXXXXX X X X X X X	X XXXXXX X X X X X X X X X X XXXXXX X X X X X X X X X X X X X X X XXXXXX X	XX X X X XX X X X X X X X X X X X X	XXX XXX X
XX					
,	(	)	*	+	,
X X X X X	X X X X X X X X X X	X X X X X X X X X X	X X X X X X X X X X XXX X XXX X X X X X X X X X X	X X X X XXXXXXXXXXXX X X X X	XX XX X X
-	.	/	0	1	2
XXXXXXXXXXXX	XX	X X X X X X X X X X X	XXXXXX X X X X X X X X X X X X X X XX X X X XXXXXX	X XX XXX X X X X X X X XXXXXX	XXXXXX X X X X X X X X X X XX X X X XXXXXXXXXXXX

3

```

XXXXXXXXXX
      X
        X
      X
    XXXX
      X
      X
      X
X      X
X      X
  XXXXX

```

4

```

XXXXXX
X      X
X      X
X      X
X      X
  XXXXXXXX
      X
      X
      XX
    XX

```

5

```

XXXXXX
X      X
X      X
      X
      X
      X
      X
      X
      X
      X
      XX

```

6

```

XXXXXXXXXX
X
X
X
X
X
XXXXXXXXXX
X
X
X
X
X
X
X
XXXXXXXXXX

```

7

```

      X
      XX
      X X
      X X
      X X
      X X
      X X
XXXXXXXXXXXX
      X
      X
      X
      X
      X
      X
      X

```

8

```

      XX
      X
      X
      X
      X
      XX
      XX
      X
      X

```

9

```

XXXXXX
X      X
X      X
X      X
X      X
  XXXXXXXX
      X
      X
      XX
    XX

```

0

```

XXXXXXXXXX
X
X
X
X
X
XXXXXXXXXX
X
X
X
X
X
X
X
XXXXXXXXXX

```

1

```

XXXXXXXXXX
X
X
X
X
XXXXXXXXXX
      X
      X
      X
      X
      X
      X
      X
      X
      X
      XXXXX

```

2

```

      XX
      X
      X
      X
      X
      XX
      XX
      X
      X

```

3

```

XXXXXX
X      X
X      X
      X
      X
      X
      X
      X
      X
      X
      XX

```

4

```

XXXXXXXXXX
X
X
X
X
X
XXXXXXXXXX
X
X
X
X
X
X
X
XXXXXXXXXX

```

5

```

      XX
      XX
      X
      X
      X
      X
      X
      X
      X
      XXXXXXXX
      X
      X
      X
      X
      X
      X
      X
      X
      X
      XXXXX

```

6

```

      XX
      X
      X
      X
      X
      XX
      XX
      X
      X

```

7

```

XXXXXXXXXX
X      X
X      X
X      X
X      X
  XXXXXXXX
X      X
X      X
X      X
X      X
X      X
X      X
XXXXXXXXXX

```

8

```

XXXXXX
X
X
X
X
X
XXXXXXXXXX
X
X
X
X
X
X
X
XXXXXXXXXX

```

9

```

XXXXXXXXXX
      X
      X
      X
      X
      X
      X
      X
      X
      X
      XXXXXXXX
      X
      X
      X
      X
      X
      X
      X
      X
      X
      XXXXX

```

0

```

XXXXXXXXXXXX
      X
      X
      X
      X
      XX
      XX
      X
      X

```

1

```

XXXXXX
X      X
X      X
X      X
X      X
  XXXXXXXX
X      X
X      X
X      X
X      X
X      X
X      X
XXXXXXXXXX

```

2

```

XXXXXXXXXX
X
X
X
X
X
XXXXXXXXXX
X
X
X
X
X
X
X
XXXXXXXXXX

```

3

```

XXXXXX
X      X
X      X
X      X
X      X
  XXXXXXXX
      X
      X
      X
      X
      X
      X
      X
      X
      X
      XXXXX

```

4

```

      XX
      X
      X
      X
      X
      XX
      XX
      X
      X

```

5

```

XXXXXXXXXX
X      X
X      X
X      X
X      X
  XXXXXXXX
X      X
X      X
X      X
X      X
X      X
X      X
XXXXXXXXXX

```

6

```

XXXXXXXXXX
X
X
X
X
X
XXXXXXXXXX
X
X
X
X
X
X
X
XXXXXXXXXX

```

K

```

-----
x       x
x       x
x       x
x      x
x     x
x    x
xxx
x  x
x   x
x   x
x   x
x   x
x   x
x   x
x   x
x   x
-----

```

L

```

-----
x
x
x
x
x
x
x
x
x
x
x
xxxxxxx
-----

```

M

```

-----
x       x
xx      xx
x x     x x
x  x   x  x
x   x  x
x    x  x
x     x  x
x      x  x
x       x  x
x        x  x
x         x  x
x          x  x
x           x  x
x            x  x
x             x  x
-----

```

N

```

-----
x       x
x       x
x       x
xx      x
x  x    x
x   x   x
x    x  x
x     x  x
x      x  x
x       x  x
x        x  x
x         x  x
x          x  x
x           x  x
x            x  x
x             x  x
-----

```

O

```

-----
xxxxxx
x       x
x       x
x       x
x       x
x       x
x       x
x       x
x       x
x       x
x       x
x       x
x       x
x       x
x       x
x       x
xxxxxx
-----

```

P

```

-----
xxxxxxxx
x       x
x       x
x       x
x       x
x       x
xxxxxxxx
x
x
x
x
x
x
x
x
-----

```

Q

```

-----
xxxxxx
x       x
x       x
x       x
x       x
x       x
x      x x
x     x  x
x    x   x
x   x    x
x  x     x
x x      x
xxxxxx x
-----

```

R

```

-----
xxxxxxxx
x       x
x       x
x       x
x      x
xxxxxxx
x  x
x   x
x    x
x     x
x      x
x       x
-----

```

S

```

-----
xxxxxx
x       x
x       x
x       x
x      x
xxxxxx
x     x
x    x
x   x
x  x
x x
xxxxxx
-----

```

T

```

-----
xxxxxxxxxxx
x
x
x
x
x
x
x
x
x
x
x
-----

```

U

```

-----
x       x
x       x
x       x
x       x
x       x
x       x
x       x
x       x
x       x
x       x
xxxxxxx
-----

```

V

```

-----
x       x
x       x
x       x
x       x
x      x
x     x
x    x
x   x
x  x
x x
x
-----

```

W

```

-----
x       x
x       x
x       x
x       x
x       x
x      x x
x     x  x
x    x   x
x   x    x
x  x     x
x x      x
xx       xx
x       x
-----

```

X

```

-----
x       x
x       x
x      x
x     x
x    x
x   x
x  x
x x
x x
x x
x x
x x
x x
x x
x x
-----

```

Y

```

-----
x       x
x       x
x       x
x      x
x     x
x    x
x   x
x  x
x x
x
x
x
x
-----

```

Z

```

-----
xxxxxxxxxxx
x
x
x
x
x
x
x
x
x
x
xxxxxxxxxxx
-----

```

[

```

-----
xxxxxxxxxxx
xx
xx
xx
xx
xx
xx
xx
xx
xx
xx
xxxxxxxxxxx
-----

```

\

```

-----
x
x
x
x
x
x
x
x
x
x
-----

```

]

```

-----
xxxxxxxxxxx
xx
xx
xx
xx
xx
xx
xx
xx
xx
xx
xx
xx
xx
xx
xx
xxxxxxx
-----

```

^

```

-----
x
x
x
x
x
x
x
x
x
x
-----

```

\_

```

-----
xxxxxxxxxxx
-----

```

o

```

-----
x
x
x
x
-----

```

a

```

-----
xxxxxx
x
xxxxxxx
x
x
x
x
xxxxxxx
-----

```

b

```

-----
x
x
x
x
x
xxxxxxx
x
x
x
x
xxxxxxx
-----

```

c	d	e	f	g	h
<pre> XXXXXXXXX X         X X X X X X         X XXXXXXXXX </pre>	<pre> X X X X X XXXXXXXXXX X         X X         X X         X X         X XXXXXXXXXX </pre>	<pre> XXXXXXXXX X         X X         X XXXXXXXXXXXXX X X         X XXXXXXXXX </pre>	<pre> XXX X X X X X XXXXXXXXX X X X X X </pre>	<pre> XXXXXXXXX X         X X X X X X         X XXXXXXXXXX </pre>	<pre> X X X X X XXXXXXXXXX X         X X         X X         X X         X </pre>

i	j	k	l	m	n
<pre> XX </pre>	<pre> XX </pre>	<pre> X X X X X X         XX X         XX X X X         XX X         XX </pre>	<pre> XXX X X X X X X XXXXXXXXX </pre>	<pre> XXXX XXXX X X X X X X X X X X X X X X X X X X </pre>	<pre> XXXXXX XX X X X X X X X X X X X </pre>

o	p	q	r	s	t
<pre> XXXXXX X         X X         X X         X X         X X         X XXXXXXXXX </pre>	<pre> XXXXXXXXXX X         X X         X XXXXXXXXXXXXX X X X </pre>	<pre> XXXXXXXXXX X         X X         X XXXXXXXXXXXXX X X X </pre>	<pre> X XXXX X X XX X X X X </pre>	<pre> XXXXXXXXXX X         X XXXXXXXXXXXXX X XXXXXXXXXXXXX </pre>	<pre> X X XXXXXX X X X XX </pre>

u	v	w	x	y	z
<pre> XXXXXX X         X X         X X         X X         X X         X XXXXXXXXX </pre>	<pre> XXXXXXXXXX X         X X         X XXXXXXXXXXXXX X X X </pre>	<pre> XXXXXXXXXX X         X X         X XXXXXXXXXXXXX X X X </pre>	<pre> X XXXX X X XX X X X X </pre>	<pre> XXXXXXXXXX X         X XXXXXXXXXXXXX X XXXXXXXXXXXXX </pre>	<pre> X X XXXXXX X X X XX </pre>

<pre> X         X X         X X         X X         X X         X X         X XXXXXXXXXX </pre>	<pre> X         X X         X X         X X         X X         X X         X XXXXXXXXXX </pre>	<pre> X         X X         X X         X X         X X         X X         X XXXXXXXXXX </pre>	<pre> X         X X         X X         X X         X X         X X         X XXXXXXXXXX </pre>	<pre> X         X X         X X         X X         X X         X X         X XXXXXXXXXX </pre>	<pre> XXXXXXXXXXXXX X X X X X X XXXXXXXXXXXXX </pre>
---	---	---	---	---	--

{		}
xx	x	xx
x	x	x
x	x	x
x	x	x
x		x
xx		xx
x		x
x	x	x
x	x	x
x	x	x
xx	x	xx

FONT13

matrixwidth= 10  
matrixheight= 15

!	"	#	\$	%
-----	-----	-----	-----	-----
XX	X  X		X	XXX
XX	X  X	X  X	X	X  X
XX	X  X	X  X	XXXXXXXX	XXX  X
XX	X  X	X  X	X  X  X	X
XX	X  X	X  X	X  X  X	X
XX	X  X	XXXXXXXXXXXX	X  X	X
XX		X  X	X  X	X
XX		X  X	XXXXXXXX	X
XX		X  X	X  X	X
		XXXXXXXXXXXX	X  X	X
		X  X	X  X  X	X
		X  X	X  X  X	X
		X  X	XXXXXXXX	X  XXX
XX		X  X	X	X  X  X
XX			X	XXX
-----	-----	-----	-----	-----

&	'	(	)	*
-----	-----	-----	-----	-----
XX	X	X	X	
X  X	X	X	X	
X  X	X	X	X	X
X  X	X	X	X	X  X  X
XX		X	X	X  X  X
XX		X	X	XXX
X  X		X	X	X
X  X  X		X	X	XXX
X  X  X		X	X	X  X  X
XX		X	X	X  X  X
XX		X	X	X
X  X  X		X	X	
XXXX  X		X	X	
-----	-----	-----	-----	-----

+	,	-	.	/
-----	-----	-----	-----	-----
X				X
X				X
X				X
X				X
XXXXXXXXXXXX		XXXXXXXXXXXX		X
X				X
X	XX			X
X	XX			X
X	XX			X
	X		XX	X
	X		XX	
-----	-----	-----	-----	-----







N

```

-----
X      X
X      X
X      X
XX     X
X X    X
X  X   X
X   X  X
X    X X
X     XX
X      X
X      X
X      X
X      X
X      X

```

O

```

-----
      XXXX
    X   X
  X     X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X

```

P

```

-----
XXXXXXXXXX
X          X
X          X
X          X
X          X
X          X
X          X
XXXXXXXXXX
X
X
X
X
X
X
X
X

```

Q

```

-----
      XXXX
    X   X
  X     X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X

```

R

```

-----
XXXXXXXXXX
X          X
X          X
X          X
X          X
X          X
XXXXXXXXXX
X   X
X  X
X  X
X   X
X   X
X   X
X   X

```

S

```

-----
      XXXXX
    X   X
  X     X
X      X
X      X
X      X
XXXXXXX
      X
      X
      X
X      X
X      X
XXXXXXX

```

T

```

-----
XXXXXXXXXXXX
      X
      X
      X
      X
      X
      X
      X
      X
      X
      X
      X
      X
      X

```

U

```

-----
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X

```

V

```

-----
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X
X      X

```

W

```

-----
X      X
X      X
X      X
X      X
X      X
X      X
X      X
XX     X
XX     X
X  X  X
X  X  X
XX     XX
X      X

```

X

```

-----
X      X
X      X
X      X
X      X
X      X
X      X
XX
X  X
X  X
X  X
X      X
X      X
X      X
X      X
X      X
X      X
X      X

```

Y

```

-----
X      X
X      X
X      X
X      X
X      X
X      X
XX
X
X
X
X
X
X
X

```

Z

```

-----
XXXXXXXXXXXX
          X
          X
          X
          X
          X
          X
          X
          X
          X
          X
          X
          X
          X

```

£

```

-----
XXXXXXXXXXXX
XX
XX
XX
XX
XX
XX
XX
XX
XX
XX
XX
XX
XX
XX

```

\

```

-----
X
X
X
X
X
X
X
X
X
X
X
X
X
X

```



l	m	n	o	p
<pre>       xxx       x       x       x       x       x       x xxx xxx       xx  x  x       x   x  x       x   x  x       x   x  x       x   x  x       x   x  x       x   x  x       x   x  x       xxxxx </pre>	<pre>       x xxx xxx       xx  x  x       x   x  x       x   x  x       x   x  x       x   x  x       x   x  x       x   x  x       x   x  x </pre>	<pre>       x  xxxxxx       x x      x       xx       x       x        x       x        x       x        x       x        x       x        x       x        x </pre>	<pre>       xxxxx       x   x       x   x       x   x       x   x       x   x       x   x       xxxxx </pre>	<pre>       xxxxxxxxx       x           x       x           x       x           x       xxxxxxxxx       x       x       x       x       x </pre>

q	r	s	t	u
<pre>       xxxxxxxxx       x           x       x           x       x           x       xxxxxxxxx       x       x       x       x </pre>	<pre>       xx  xxx       x x       xx       x       x       x       x       x </pre>	<pre>       xxxxxxxxx       x           x       x           x       x           x       xxxxxxxxx       x       x       x       xxxxxxxxx </pre>	<pre>       x       x       x       xxxxxx       x       x       x       x       x       x       x       xxx </pre>	<pre>       x           x       x           x       x           x       x           x       x           x       x           x       x           x       xxxxxxxxx </pre>

v	w	x	y	z
<pre>       x           x       x           x       x           x       x           x       x           x       x           x       x           x       x           x       x           x       x           x </pre>	<pre>       x           x       x           x       x           x       x           x       x           x       x           x       x           x       x           x       x           x       x           x </pre>	<pre>       x           x       x           x       x           x       x           x       xx       x           x       x           x       x           x       x           x       x           x </pre>	<pre>       x           x       x           x       x           x       x           x       xxxxxxxxx       x           x       x           x       xxxxxxxxx </pre>	<pre>       xxxxxxxxx       x           x       x           x       x           x       xx       x       x       xxxxxxxxx </pre>

€	l	ƒ
xxx		xxx
x	x	x
x	x	x
x	x	x
x	x	x
x		x
xx		xx
x		x
x	x	x
x	x	x
x	x	x
x	x	x
x	x	x
xxx		xxx



+  
-----

```
      x
      x
      x
      x
XXXXXXXXXXXXXXXXX
      x
      x
      x
      x
```

9  
-----

```
      xx
      xx
      xx
      x
      x
```

-  
-----

```
XXXXXXXXXXXXXXXXX
```

.  
-----

```
      xx
      xx
```

/  
-----

```
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
```

0  
-----

```
XXXXXXXXXX
 x          x
x          x
x          x
x          xx
x          x x
x          x x
x          x x
x          x x
x          x x
x          x x
x          x x
x          x x
x          x x
x          x
x          x
xx         x
x          x
x          x
      xxxxxx
```

1  
-----

```
      x
      xx
      xxx
x   x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      xxxxxx
```

2  
-----

```
      xxxxxx
x          x
x          x
x          x
      x
      x
      x
      x
      xx
      x
      x
      x
      x
      x
      xxxxxx
```

3  
-----

```
XXXXXXXXXXXXXXXXX
      x
      x
      x
      x
      x
      xxxxx
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      xxxxxx
```

4  
-----

```
      x
      xx
      x x
      x x
      x x
      x x
      x x
      x x
      x x
      x x
      xxxxxxxxxxxxxx
      x
      x
      x
      x
      x
      x
      x
```

5  
-----

```
XXXXXXXXXXXXXXXXX
x
x
x
x
XXXXXXXXXXXXX
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      xxxxxx
```

6  
-----

```
      xx
      xx
      x
      x
      x
      x
      xxxxxxxxxxxxxx
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      xxxxxx
```

7  
-----

```
XXXXXXXXXXXXXXXXX
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
      x
```

8  
-----

```
      xxxxxx
x          x
x          x
x          x
x          x
x          x
x          x
x          x
x          x
x          x
x          x
x          x
x          x
x          x
x          x
x          x
x          x
      xxxxxx
```

9  
-----

```
      xxxxxx
x          x
x          x
x          x
x          x
x          x
x          x
      xxxxxxxxxxxxxx
      x
      x
      x
      x
      x
      xx
      xx
```

= < = >

xx  
xx

xx  
xx

xx  
x  
x  
x  
x  
x  
x  
x  
x  
x  
x  
x  
xx

xxxxxxxxxxxx

xx  
x  
x  
x  
x  
x  
x  
x  
x  
x  
xx

xx  
xx

xx  
xx  
xx  
x  
x

xxxxxxxxxxxx

?

@

A

B

C

xxxxx  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
xx  
xx

xxxxx  
x x  
x x  
x x  
x xxx x x  
x x x x x  
x x x x x  
x xxx xx  
x x  
x x  
x x  
x x  
x x  
xxxxxx

x  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
xxxxxxxxxxxx  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
xxxxxx

xxxxxxxxx  
x x  
x x  
x x  
x x  
x x  
xxxxxxxxx  
x x  
x x  
x x  
x x  
x x  
xxxxxx

xxxxx  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
xxxxxx

D

E

F

G

H

xxxxxxxxx  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
x x  
xxxxxxxxx

xxxxxxxxxxxx  
x x  
x x  
x x  
x x  
xxxxxxxxx  
x x  
x x  
x x  
x x  
xxxxxxxxx

xxxxxxxxxxxx  
x x  
x x  
x x  
x x  
xxxxxxxxx  
x x  
x x  
x x  
x x  
x x  
x x

xxxxx  
x x  
x x  
x x  
x x  
x x  
xxxxx  
x x  
x x  
x x  
x x  
xxxxxx

x x  
x x  
x x  
x x  
x x  
xxxxxxxxxxxx  
x x  
x x  
x x  
x x  
x x  
x x







<p><u>g</u></p> <pre> XXXXXXXXXX X           X X X           X X           X X           X X           X X           X X           X X           X X           X X           X XXXXXXXXXX X           X X           X X           X X           X X           X X           X XXXXXXXXXX         </pre>	<p><u>h</u></p> <pre> X X X           XX X           XX X X X XXXXXXXXXXXX X           X X           X X           X X           X X           X X           X X           X X           X X           X         </pre>	<p><u>i</u></p> <pre> XXXXXXXXXX X           X X           X X           X X           X X           X X           X XXXXXXXXXXXX         </pre>	<p><u>j</u></p> <pre> XXXXXXXXXX X           X X           X X           X X           X X           X X           X X           X X           X X           X X           X X           X XXXXXXXXXX         </pre>	<p><u>k</u></p> <pre> X X X           XX X           XX X X X           XX X           XX X           XX X           XX X           XX X           XX X           XX X           XX         </pre>
---	---	--	--	--

<p><u>l</u></p> <pre> XXXXX XXXXXX         </pre>	<p><u>m</u></p> <pre> XXXXXXXXXXXX XX  X  XXXX X   X  X   X X   X  X   X X   X  X   X X   X  X   X X   X  X   X X   X  X   X X   X  X   X X   X  X   X X   X  X   X         </pre>	<p><u>n</u></p> <pre> XXXXXXXXXX X  XXXXXXXX X X           X XX          X X           X X           X X           X X           X X           X X           X X           X X           X         </pre>	<p><u>o</u></p> <pre> XXXXXXXXXX X           X X           X X           X X           X X           X X           X X           X X           X X           X X           X XXXXXXXXXX         </pre>	<p><u>p</u></p> <pre> XXXXXXXXXXXX X           X X           X X           X X           X XXXXXXXXXXXX X           X X           X X           X X           X         </pre>
---	--	---	--	--

<p><u>q</u></p> <pre> XXXXXXXXXXXX X           X X           X X           X X           X XXXXXXXXXXXX X X X X X         </pre>	<p><u>r</u></p> <pre> XX  XXXX X  X XX X X X X X X         </pre>	<p><u>s</u></p> <pre> XXXXXXXXXX X           X X           X XXXXXXXXXXXX X           X X           X XXXXXXXXXXXX         </pre>	<p><u>t</u></p> <pre> X X X XXXXXXXXXXXX X X X X X X X X X XXXXXXXXXX         </pre>	<p><u>u</u></p> <pre> X           X X           X X           X X           X X           X X           X X           X X           X X           X XXXXXXXXXX X         </pre>
--	---	---	--	---

v \_\_\_\_\_

w \_\_\_\_\_

x \_\_\_\_\_

y \_\_\_\_\_

z \_\_\_\_\_

```

x      x
x      x
x      x
x      x
  x    x
  x    x
    x  x
    x  x
      x

```

```

x      x
x      x
x      x
x      x
  x    x
  x    x
    x  x
    x  x
      x  x
      x  x

```

```

x      x
  x    x
    x  x
      x x
      xxx
    x  x
      x  x
        x  x

```

```

x      x
x      x
x      x
x      x
  x    x
    xxxxxx

```

```

xxxxxxxxxxxxx
               x
                x
                 x
                  x
                   xx
                    x
                     x
                      x
                       xxxxxxxxxx

```

{ \_\_\_\_\_

| \_\_\_\_\_

} \_\_\_\_\_

```

      xxx
      x
      x
      x
      x
      x
xxx
x
  x
  x
  x
  x
  x
      x
      xxx

```

```

      x
      x
      x
      x
      x
      x
      x
      x
      x

```

```

      xxx
      x
      x
      x
      x
      x
      x
      x
      xxx
      x
      x
      x
      x
      x
      x
      xxx

```









A

---

```

      X
     XXX
    XX XX
   XX  XX
  XX   XX
 XX    XX
XX     XX
XX      XX
XX       XX
XX        XX
XXXXXXXXXXXXXXXX
XX         XX
XX          XX
XX           XX
XX            XX
XX             XX
XXXXXXXXXXXXXXXX
XX              XX
XX               XX
XX                XX
XX                 XX
XX                  XX
XXXXXXXXXXXXXXXX

```

---

B

---

```

XXXXXXXXXXXXXXXX
XX          XX
XX           XX
XX            XX
XX             XX
XX              XX
XX               XX
XXXXXXXXXXXXXXXX
XX                XX
XX                 XX
XX                  XX
XX                   XX
XX                    XX
XXXXXXXXXXXXXXXX
XX                     XX
XX                      XX
XXXXXXXXXXXXXXXX

```

---

C

---

```

      XXXXXX
     XX   XX
    XX    XX
   XX     XX
  XX      XX
 XX       XX
XX        XX
XX         XX
XX          XX
XX           XX
XXXXXXXXXXXXXXXX
XX            XX
XX             XX
XX              XX
XX               XX
XX                XX
XXXXXXXXXXXXXXXX
XX                 XX
XX                  XX
XX                   XX
XXXXXXXXXXXXXXXX

```

---

D

---

```

XXXXXXXXXXXXXXXX
XX          XX
XX           XX
XX            XX
XX             XX
XX              XX
XX               XX
XXXXXXXXXXXXXXXX
XX                XX
XX                 XX
XX                  XX
XX                   XX
XX                    XX
XXXXXXXXXXXXXXXX
XX                     XX
XX                      XX
XXXXXXXXXXXXXXXX

```

---

E

---

```

XXXXXXXXXXXXXXXX
XX
XX
XX
XX
XX
XX
XX
XX
XXXXXXXXXXXXXXXX
XX
XX
XX
XX
XX
XX
XX
XX
XXXXXXXXXXXXXXXX

```

---

F

---

```

XXXXXXXXXXXXXXXX
XX
XX
XX
XX
XX
XX
XX
XX
XXXXXXXXXXXXXXXX
XX
XX
XX
XX
XX
XX
XX
XX

```

---

G

---

```

      XXXXXX
     XX   XX
    XX    XX
   XX     XX
  XX      XX
 XX       XX
XX        XX
XX         XX
XX          XX
XXXXXXXXXXXXXXXX
XX            XX
XX             XX
XX              XX
XX               XX
XX                XX
XXXXXXXXXXXXXXXX
XX                 XX
XX                  XX
XX                   XX
XXXXXXXXXXXXXXXX

```

---

H

---

```

XX          XX
XX           XX
XX            XX
XX             XX
XX              XX
XX               XX
XXXXXXXXXXXXXXXX
XX                XX
XX                 XX
XX                  XX
XX                   XX
XX                    XX
XXXXXXXXXXXXXXXX
XX                     XX
XX                      XX
XXXXXXXXXXXXXXXX

```

---









a

```

-----
XXXXXXXXX
      XX
        XX
          XX
            XX
XXXXXXXXXXXXXXXXX
  XX      XX
XX      XX
XX      XX
  XX      XX
XXXXXXXXXXXXXXXXX
-----

```

b

```

-----
XX
XX
XX
XX
XX
XX
XX
XX
XX
XXXXXXXXXXXXX
XX      XX
XX      XX
XX      XX
XX      XX
XX      XX
XX      XX
XX      XX
XX      XX
XXXXXXXXXXXXX
-----

```

c

```

-----
XXXXXXXXXX
      XX      XX
        XX      XX
          XX
XXXXXXXXXX
XX      XX
XX      XX
XX      XX
XX      XX
XX      XX
XX      XX
XX      XX
XXXXXXXXXX
-----

```

d

```

-----
XX
XX
XX
XX
XX
XX
XX
XX
XXXXXXXXXXXXX
  XX      XX
    XX      XX
      XX      XX
        XX      XX
          XX      XX
XXXXXXXXXXXXX
  XX      XX
    XX      XX
      XX      XX
        XX      XX
          XX      XX
XXXXXXXXXXXXX
-----

```

e

```

-----
XXXXXXXXXX
  XX      XX
XX      XX
XX      XX
XXXXXXXXXXXXXXXXX
XX
XX
XX
  XX      XX
  XX      XX
XXXXXXXXXX
-----

```

f

```

-----
      XXXX
      XX
      XX
      XX
      XX
      XX
      XX
      XX
      XX
XXXXXXXXXXXXX
XX
XX
XX
XX
XX
XX
XX
XX
XX
-----

```

g

```

-----
XXXXXXXXXX
  XX      XX
XX      XX
XX
XX
XX
  XXXXX
XX      XX
  XX      XX
XXXXXXXXXX
-----

```

h

```

-----
XX
XX
XX
XX
XX
XX
XX
XXXXXXXXXXXXX
  XX      XX
XX      XX
XX      XX
XX      XX
XX      XX
XX      XX
XX      XX
XX      XX
XX
-----

```







## APPENDIX C

Nedan följer listningarna för några program som använts för att omvandla informationen från fonterna i appendix B så att den kunde lagras i prom för teckenminnet.



```
1 (*****  
2 (*  
3     programmet genererar alla block till teckenminnet.  
4     fontfile skall vara en fil med alla tecknen for  
5     en font efter varandra och i sin tur de olika fonterna  
6     efter varandra.  
7     de tva forsta blocken med fontinformationen laeses i borjan  
8     av programmet fran tangentbordet  
9     blockfile innehaller sedan alla blocken efter varandra. *)  
10 (*****  
11  
12  
13  
14 program fonter(fontfile,blockfile,input,output);  
15  
16 type blocktype= array[0..127] of integer;  
17     string= array[1..8] of char;  
18     text= file of char;  
19  
20 var fontfile: text;  
21     blockfile: file of integer;  
22     font: array[33..125,1..20,1..2] of integer;  
23     fi: array[0..15,1..4] of integer;  
24     block,nollblock: blocktype;  
25     bit: array[1..8] of integer;  
26     t: array[2..4] of string;  
27  
28     i,k,l,fontsize,dw: integer;  
29     width,height,blockant: integer;  
30     ch: char;  
31  
32 procedure readfont;  
33     var i,k,sum,dw,kol: integer;  
34     ch: char;  
35     begin  
36     for i:= 1 to 3 do readln(fontfile);  
37     readln(fontfile,width,height);  
38     for i:= 1 to 4 do readln(fontfile);  
39     for i:= 33 to 125 do  
40         for k:= 1 to 20 do  
41             for kol:= 1 to 2 do  
42                 font[i,k,kol]:= 0;  
43     for i:= 33 to 125 do  
44         begin  
45         for k:= 1 to 2 do  
46             readln(fontfile);  
47         for k:= 1 to height do  
48             begin  
49                 dw:= 0;  
50                 repeat  
51                     dw:= dw + 1;  
52                     sum:= 0;  
53                     kol:= 0;  
54                     while (kol < 8) and (not eoln(fontfile)) do  
55                         begin  
56                             kol:= kol + 1;  
57                             read(fontfile,ch);  
58                             if ch <> ' ' then sum:= sum + bit[kol];  
59                             end;  
60                     font[i,k,dw]:= sum;
```

```
61         until eoln(fontfile) or (dw = 2);
62         readln(fontfile);
63     end;
64     readln(fontfile);
65     end;
66 end;
67
68 procedure storeblock;
69 var i: integer;
70 begin
71     for i:= 0 to 127 do
72     begin
73         blockfile:= block[i];
74         put(blockfile);
75     end;
76     block:= nollblock;
77     blockant:= blockant + 1;
78 end;
79
80 begin
81 k:= 256;
82 rewrite(blockfile);
83 reset(fontfile);
84 t[2]:= 'startadr';
85 t[3]:= 'fonthojd';
86 t[4]:= 'dwfbredd';
87 k:= 256;
88 for i:= 1 to 8 do
89     begin
90         k:= k div 2;
91         bit[i]:= k;
92     end;
93 for i:= 0 to 127 do
94     nollblock[i]:= 0;
95 blockant:= 0;
96 for i:= 0 to 15 do
97     begin
98         writeln('font',i:1);
99         fi[i,1]:= 0;
100        for k:= 2 to 4 do write(t[k]);
101        writeln;
102        read(fi[i,2],fi[i,3],fi[i,4]);
103    end;
104 for i:= 0 to 3 do
105     for k:= 0 to 15 do
106         begin
107             block[i * 32 + k + k]:= i;
108             block[i * 32 + k + k + 1]:= fi[k,3];
109         end;
110 storeblock;
111 for i:= 0 to 3 do
112     for k:= 0 to 15 do
113         begin
114             block[i*32+k+k]:= fi[k,2];
115             block[i*32+k+k+1]:= fi[k,4];
116         end;
117 storeblock;
118
119 for fontsize:= 0 to 15 do
120     begin
```

```
121 readfont;
122 if fontsize = 12 then storeblock;
123 dw:= 1;
124 if width > 8 then dw:= 2;
125 if (fi[fontsize,3] - height) > 1 then
126   for i:= 1 to dw do storeblock;
127 for k:= 1 to height do
128   for i:= 1 to dw do
129     begin
130       for l:= 33 to 125 do
131         block[l]:= font[l,k,i];
132       storeblock;
133     end;
134 for i:= 1 to dw do storeblock;
135 if (fontsize = 6) or (fontsize = 7) then storeblock;
136 writeln(fontsize,width,height,blockant);
137 end;
138 end.
```

```
1 (*****  
2 (*  
3     programmet sorterar filen bfile till p0..p15, dvs den  
4     information som skall prommas i kapsel 0 till 15.  
5     filerna p0 till p15 blir pa 2048 bytes  
6     information till programmet ar hur  
7     manga par av kapslar som skall hoppas over,  
8     dvs naer p6 och p7 genereras skall 3 laesas  
9     in efter utskriften prom0  
10                                          *)  
11 (*****  
12  
13  
14 program sort(bfile,p0,p1,input,output);  
15 type blocktype= array[0..63] of integer;  
16     intfile= file of integer;  
17 var bfile,p0,p1: intfile;  
18     i,i1,i2: integer;  
19     block: blocktype;  
20  
21 procedure loadblock;  
22     var i: integer;  
23     begin  
24         for i:= 0 to 63 do  
25             begin  
26                 block[i]:= bfile^;  
27                 get(bfile);  
28             end;  
29     end;  
30  
31 procedure storeblock(var pfile: intfile);  
32     var i: integer;  
33     begin  
34         for i:= 0 to 63 do  
35             begin  
36                 pfile^:= block[i];  
37                 put(pfile);  
38             end;  
39     end;  
40  
41 begin  
42 reset(bfile);  
43 get(bfile);  
44 rewrite(p0);  
45 rewrite(p1);  
46 writeln('prom0');  
47 read(i2);  
48  
49 for i1:= 1 to i2 do  
50 for i:= 0 to 31 do  
51     loadblock;  
52  
53 for i:= 0 to 15 do  
54     begin  
55         loadblock;  
56         storeblock(p0);  
57         loadblock;  
58         storeblock(p1);  
59     end;  
60 end.
```

```
1 (*****  
2 (*  
3     aendrar infil fran 2048 bytes till 4096 bytes for  
4     utfil genom att for varannat byte saetta in ett extra.  
5     innehallet andras inte annars.  
6                                             *)  
7 (*****  
8  
9  
10 program shift(infil,utfil);  
11 var infil,utfil: file of integer;  
12     sign,k,i,i1,i2: integer;  
13 begin  
14     reset(infil);  
15     get(infil);  
16     rewrite(utfil);  
17     for k:= 0 to 1023 do  
18         begin  
19             sign:= 0;  
20             i:= infil^;  
21             get(infil);  
22             if i<0 then begin sign:= 128; i:= i - 32768; end;  
23             i1:= i div 256 + sign;  
24             i2:= i mod 256;  
25             utfil^:= i1;  
26             put(utfil);  
27             utfil^:= i2;  
28             put(utfil);  
29         end;  
30 end.
```