

© CREATIVISION **BASIC**

VERSION
1:0

INTERPRETER



REFERENCE MANUAL



Innehållsförteckning

1. Förord
2. Bruksanvisning
3. Att programmera i BASIC
4. Grafik och ljud
5. Användning av kassettminne
6. Övningar
7. Sammanfattning av kommandon och instruktioner
8. Teckenkoder
9. Felkoder

1 FÖRORD

Denna handbok är avsedd att ge en introduktion till programmering i BASIC på CreatiVision Computer System. Den tar upp de mest elementära grunderna och går steg för steg igenom hur man skapar program och använder CreatiVision Computer System. Den beskriver alla funktioner och kommandon som behövs för grundläggande BASIC-programmering och kräver normalt inga speciella förkunskaper om BASIC eller om programmering.

När Du läser boken bör Du pröva att utföra alla de exempel och övningar som finns beskrivna, genom att skriva in och köra programmen. Du bör också försöka experimentera med att förändra exemplen, förutsäga effekterna av förändringen och observera resultatet. Därigenom kommer Du snart att kunna skriva egna program för att lösa enkla uppgifter. Endast genom att öva och praktisera kommer Du att lära och förstå möjligheterna att använda datorn till praktisk nytta.

Det går inte att genom det Du skriver in från tangentbordet förstöra Ditt CreatiVision Computer System. Känn Dig därför hela tiden fri att experimentera vid alla tillfällen.

Kapitlen i handboken är upplagda på ett sådant sätt att den bör läsas i den ordning de är skrivna, med ett undantag; Kapitel 5 som handlar om lagring av program på kasett-bandspelaren, kan läsas så snart Du vill spara ett program för senare användning. Boken är visserligen skriven som en introduktion till BASIC speciellt för CreatiVision Computer System, men det mesta kan tillämpas även vid programmering i BASIC på andra datorer. Påpekas bör dock att BASIC-språket har många dialekter och att det oftast finns smärre skillnader mellan olika system.

Det är vår förhoppning att denna handbok skall hjälpa Dig att förstå grunderna i programmering i BASIC. För mer avancerad användning av BASIC hänvisar vi till det stora utbud av allmänna böcker om BASIC och programmering som finns i bokhandeln.

LYCKA TILL!

Vad är en dator?

En dator är en enhet som kan fås att utföra varierande operationer beroende av instruktioner som den som använder datorn har angivit.

Allmänt sett består varje dator av följande delar;

- En centralenhet (CPU – Central Processing Unit) som kan fås att utföra operationer som specificerats i form av instruktioner, ex vis beräkningar eller jämförelser. Denna del är datorns 'hjärna'.
- Primärminne. I detta lagras instruktioner och data som datorn själv beräknat eller som matats in av användaren. Primärminnet finns inuti själva datorn och står i direkt kontakt med centralenheten.
- Sekundärminne. Denna enhet används för lagring av större mängder av instruktioner och data. Sekundärminnet är oftast en separat enhet och är mycket långsammare än primärminnet men kan å andra sidan lagra stora mängder instruktioner och data till låg kostnad. Exempel på sekundärminnen är kasettminne och diskettminne. Data och instruktioner måste överföras till primärminnet innan centralenheten kan få tillgång till dem.
- Inmatningsenheter. Enheter som användaren använder för att mata in instruktioner och data till datorn. Ex. Tangenbord, joy-sticks.
- Utmatningsenheter. Enheter för utmatning från centralenheten till användaren. Ex. skrivare, högtalare, bild-monitor.

Tillsammans möjliggör in- och utmatningsenheterna en 'dialog' mellan användare och datorsystemet.

Trots att stora skillnader finns i storlek och prestanda så består alla datorer i grund och botten av de uppräknade enheterna.

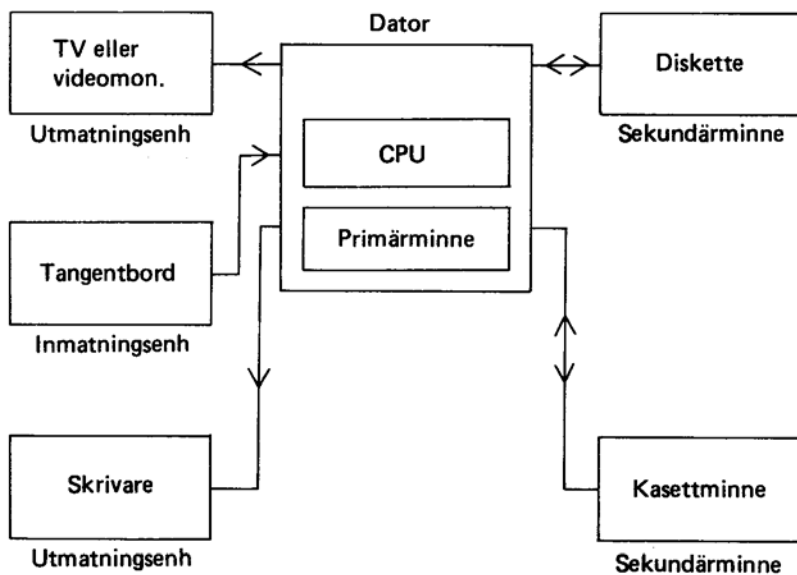
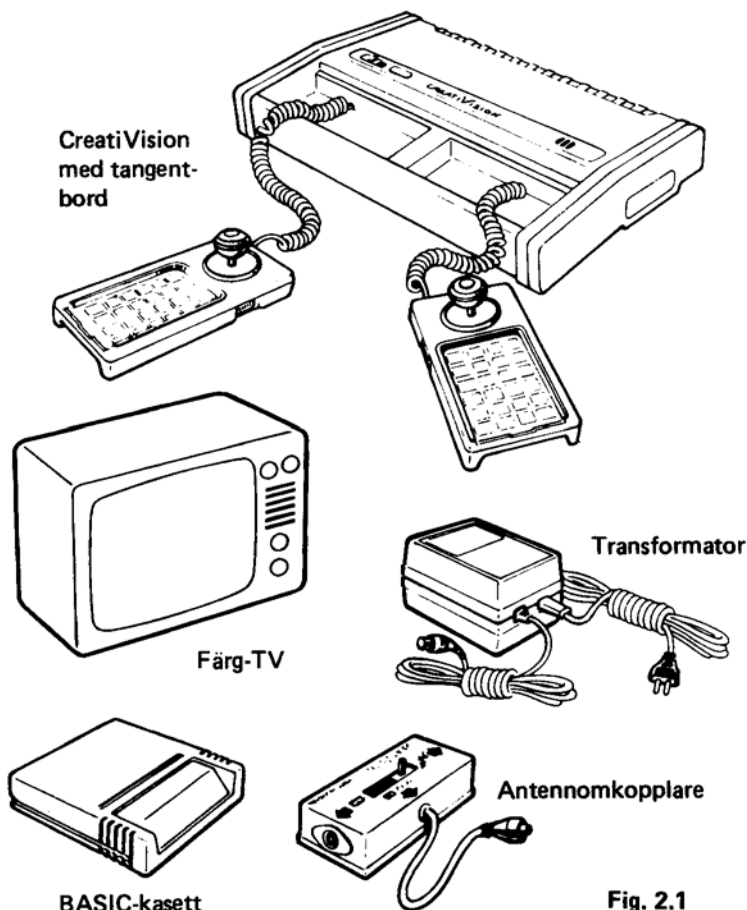


Fig. 1.1 Översikt dator

2 BRUKSANVISNING

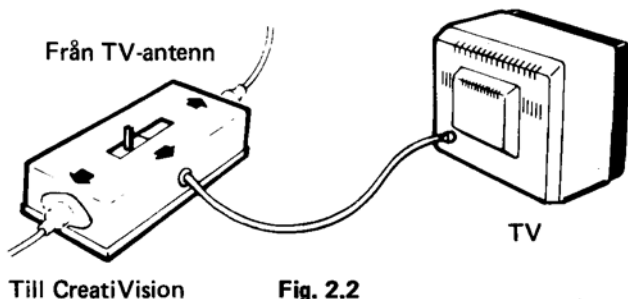
För att starta ditt CreatiVision Computer System behöver du följande:



Antennomkopplaren

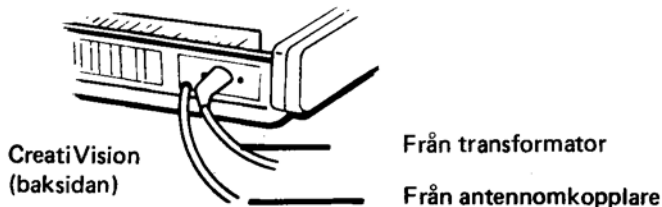
Med hjälp av antennomkopplaren kan du på ett enkelt sätt koppla om Din TV mellan mottagning och CreatiVision.

- Koppla loss antennkabeln från Din TV och anslut den till antennomkopplaren.
- Anslut kabeln från antennomkopplaren till antenningången på TV'n.
- Anslut kabeln från CreatiVision till antennomkopplaren.



Igångsättning

1. Kontrollera att strömbrytaren på CreatiVision står i läge "OFF".
2. Anslut kontakten från transformatorn till uttaget på baksidan av CreatiVision.
3. Anslut transformatorn till vägguttaget.



4. Ställ antennomkopplaren i läge "GAMES".
5. Stoppa in BASIC-kassetten i uttaget på höger sida av CreatiVision.

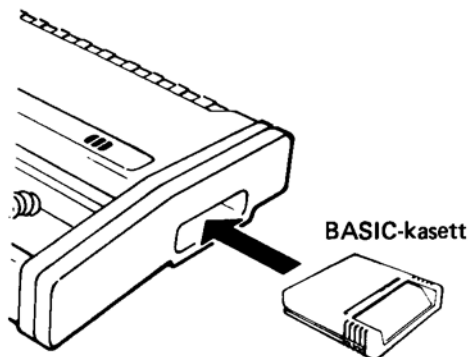


Fig. 2.4

6. Slå på TV'n och ställ in mottagningen på VHF kanal 3.
7. Slå på strömmen på CreatiVision. En bild med texten "CREATI-VISION BASIC..." skall då komma fram på TV-rutan. Finjustera vid behov kanalinställningen. OBS! Om Din TV är utrustad med automatisk fininställning skall denna stängas av medan Du finjusterar bilden. Se bruksanvisningen för Din TV.

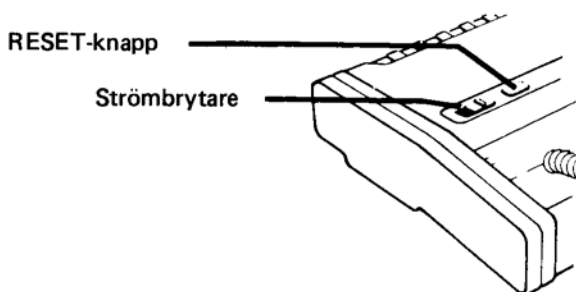


Fig. 2.5

8. CreatiVision BASIC är klar att användas.

Att iakttaga

- Utrustningen är känslig för fukt. Använd inte vätskor i närheten.
- Undvik att utsätta kassetter eller utrustning för värme. Förvara dem i ventilerat utrymme.
- Slå av strömmen och tag ut BASIC-kassetten då utrustningen inte används.
- Utrustningen kan skadas om den tappas. Hantera den varsamt.
- Sätt in BASIC-kassetten varsamt utan att använda våld. Stäng alltid av strömmen då kassetten sätts i.
- Stick inte in fingrar i den öppna ändan av BASIC-kassetten. Statisk elektricitet kan skada den känsliga elektroniken inuti kassetten.

Nedkoppling

1. Slå av strömmen på CreatiVision och tag ut BASIC-kassetten.
2. Slå om antennomkopplaren i läge "TV ANTENNA".
3. TV'n är nu klar för normal användning.

Sammanfattning

1. Stoppa in BASIC-kassetten ordentligt i uttaget på höger sida av CreatiVision.
2. Kontrollera att transformatorn är ansluten till vägguttag och CreatiVision.
3. Kontrollera att antennomkopplaren står i läge "GAME".
4. Kontrollera att alla antennkablar är anslutna.
5. Kontrollera att strömmen är påslagen på CreatiVision och TV.
6. Kontrollera att rätt kanal är inställd på TV'n.

Utbyggnadsmöjligheter

CreatiVision Computer System kan lätt byggas ut till en komplett hemdator. Här följer en kort presentation av de tillbehör som finns på marknaden redan nu, samt även de tillbehör som beräknas komma senare.

1. Kasettminne

Med CreatiVision kasettminne kan Du spara program på vanliga ljudbandskassetter och hämta fram dem igen. Bandspelaren styrs direkt från BASIC-programmet (se kapitel 5). Överföringshastighet 600 tecken/sekund.

2. Parallell och serieinterface

Parallell och serieinterface gör det möjligt att ansluta olika in- och utmatningsenheter. Den parallella utgången gör det möjligt att ansluta en skrivare. Utgången är en standard "Centronic"-utgång till vilken man kan ansluta alla skrivare med Centronic-anslutning exempelvis EPSON MX 80, SEIKOSHA GP-80, CENTRONICS 779 m fl.

De seriella utgångarna är avsedda för anslutning av diskett-minne respektive för telefon-modem.

3. Minnesexpansion

Det interna minnet på CreatiVision Computer System kan byggas ut genom minnesexpansionsmoduler från 16K (16000 tecken) till 64K. Expansionsmodulerna innehåller 16K eller 32K.

4. 'Riktigt' tangentbord

Som extra tillbehör finns också ett 'riktigt' tangentbord med vanliga tangenter. Tangentbordet underlättar skrivandet avsevärt.

3 Att programmera i BASIC

Ett program är en serie instruktioner, som talar om för en dator hur något skall göras.

Ett BASIC-program använder sig för detta ändamål av BASIC-språket, som får datorn att förstå hur vi vill att olika data skall behandlas, hur och vilka beräkningar som skall göras, osv.

Ett BASIC-program innehåller därför en lista av exakta instruktioner, som skall utföras av datorn. Listan består av ett antal rader med instruktioner eller satser.

Börja programmera

Datorn kan utföra satser på två olika sätt.

Direkt

Satser, som inte börjar med ett radnummer, utförs direkt efter att man tryck på **RET'N**-knappen. (**RET'N** står för **RETURN** = "vagnretur").

Exempel **PRINT 2+4 RET'N**
 6
 PRINT 2+4, 3/6 RET'N
 6 0.5
 PRINT "RESULTAT", (2+4)/(3/6) RET'N
 RESULTAT 12

Direkt utförande gör att datorn fungerar som en kraftfull kalkylator. Den kan beräkna mer än ett aritmetriskt uttryck i taget och kan också behandla andra tecken än siffror.

För att direkt utförande skall bli meningsfullt, måste Du använda **PRINT**-instruktionen. Annars får Du nämligen inte ut resultatet på TV-skärmen.

Program

Satser, som börjar med ett radnummer, utförs inte förrän man ger kommandot **RUN** och trycker på **RET'N**.

Exempel > 10 PRINT 2+4 RET'N
 > RUN RET'N
 6

Kommandot **RUN** gör att alla satser, som för tillfället finns i datorn, utföres i nummerordning oavsett i vilken ordning de matats in.

Kommandot **NEW** raderar ut eventuella gamla programrader i datorns minne. Därför bör Du ge kommandot **NEW** före varje exempel nedan.

I fortsättningen är det underförstått, att Du trycker **RET'N** efter varje rad på skärmen.

Enkla in- och utmatningar

Beräkna och skriv.

Exempel > **10 PRINT 24/6**
 > **20 END**

Programmet består av två satser. Den första satsen, **10 PRINT 24/6**, ger datorn order om att beräkna 24/6 samt skriva ut resultatet. Den andra satsen, **20 END**, markerar att programmet är slut.

Varje sats skall förses med ett satsnummer. Upp till fyra (4) siffrors längd är tillåtet. Datorn utför satserna i satsnummerordning oavsett i vilken ordning inmatningen har skett. Därför är det lämpligt att Du numrerar glest, t.ex. **10**, **20**, **30** osv. Då kan Du lätt i efterhand lägga till nya satser emellan de gamla.

Exempel > **NEW**
 > **10 PRINT 24/6**
 > **20 END**
 > **RUN**
 4

Variabler

Vill Du utföra samma beräkning, men kunna variera talen, kan Du använda s.k. variabler. Inmatningen av tal sker då med instruktionen **INPUT**.

Exempel > **10 INPUT A,B**
 > **20 PRINT A/B**
 > **30 END**
 > **RUN**
 ? 24
 ? 6
 4

När datorn träffar på instruktionen **INPUT**, skriver den ut ett frågetecken (?) och inväntar värdet på den första variabeln (A).

När värdet på variabeln A är inmatat, skriver datorn ett nytt frågetecken (?) och inväntar värdet på variabeln B. När värdet på variabeln B är inmatat, fortsätter datorn till nästa instruktion, **20 PRINT A/B**, och utför denna.

Som variabelnamn i CreatiVision BASIC använder Du en bokstav i det engelska alfabetet A – Z.

Tal

Ett tal, som Du t.ex. matar in till en **INPUT**-instruktion, får innehålla högst sex (6) signifikanta siffror. Talen skriver Du på följande sätt:

- som heltal +2, 34, -3741
- som decimaltal med decimalpunkt 3.1416, 0.047
- med 10 -potenser
45000 kan skrivas $4.5E4$ där $E4$ står för 10^4
0.0045 kan skrivas $4.5E-3$ där $E-3$ står för 10^{-3}

ARITMETIK

Algebraiska uttryck

Skrivs i BASIC på nästan samma sätt som i den vanliga algebran.

Operator	Exempel	Innebörd
+	$A+B$	Adderar B till A
-	$A-B$	Subtraherar B från A
*	$A*B$	Multiplicerar A med B
/	A/B	Dividerar A med B
**	$A**B$	Upphöjer A med B (A^B)

Glöm inte, att när man i vanlig algebra skriver t.ex. **3A**, underförstår man multiplikationstecknet. I BASIC måste man skriva **3*A**.

Ett BASIC-uttryck får också innehålla parenteser. Beräkningarna utföres i normal ordning, dvs enligt följande prioritetsregler:

1. Uttryck inom parentes.
2. Upphöjt till (Exponentiering).
3. Multiplikation och division (lika prioritet).
4. Addition och subtraktion (lika prioritet).

Exempel $3*3**2+8/2$ beräknas $3*9+8/2=45+4=49$
 $3*3**(2+8/2)$ beräknas $3*3(2+4)=3*3^6=3*729=2187$
 $3*3**((2+8)/2)$ beräknas $3*3(10/2)=3*3^5=3*243=729$

Vid lika prioritet, sker beräkningarna från vänster till höger. Om Du är osäker på beräkningsgången, kan Du sätta ut extra parenteser.

Textutskrifter

Datorer kan naturligtvis mer än att utföra beräkningar. De kan också lagra, bearbeta och skriva ut texter.

PRINT

Skriver värdet av variabler eller värdet av ett uttryck på TV-skärmen.

```
Exempel  > 10 PRINT "CREATIVISION BASIC"  
          > 20 END  
          RUN  
          CREATIVISION BASIC
```

Texten inom citationstecken i **PRINT**-satsen, skrivs ut precis som den är.

```
Exempel  > 10 PRINT "MATA IN A OCH B"  
          > 20 INPUT A,B  
          > 30 PRINT "A*B="; A*B  
          > 40 END  
  
          > RUN  
          MATA IN A OCH B  
          ? 2  
          ? 3  
          A*B=6
```

Texten i sats 10 kan betraktas som ledtext till de frågetecken (?), som **INPUT**-satsen ger.

"A*B=" ger utskriften A*B=, medan A*B ger utskriften 6, som är uttryckets värde. Ett semikolon (;) mellan de båda utskriftsposterna gör att de kommer tätt intill varandra.

Om Du ändrar satsen 30 så, att det blir ett kommatecken (,) istället för semikolon (;).

30 PRINT "A*B=", A*B

Fås utskriften

A*B= 6

Kommatecknet ger en glesare utskrift.

Om Du avslutar sats 10 med ett semikolon (;)

10 PRINT "MATA IN A OCH B";

Får Du utskriften

MATA IN A OCH B?

och inmatning av värdet på A kan skrivas på samma rad. Du kan nu göra om programmet och få en bättre dialog.

Exempel > **10 PRINT "VÄRDET PÅ A";**
 > **20 INPUT A**
 > **30 PRINT "VÄRDET PÅ B";**
 > **40 INPUT B**
 > **50 PRINT "A*B="; A*B**
 > **60 END**

 > **RUN**

 VÄRDET PÅ A?2
 VÄRDET PÅ B?3

 A*B=6

TAB(n)

Används i en **PRINT**-sats för att flytta in texten åt höger. Antalet steg, som texten skall flyttas med, skrivs inom parentes.

Exempel > **PRINT TAB(5); "A"; TAB(5); "B"**
 A B

LPRINT

Fungerar precis som **PRINT**, men ger utskrift på skrivare istället för på TV-skärmen.

Hur variabler tilldelas värden

Det finns tre olika sätt att tilldela variabler värden.

INPUT

Används när Du vill föra en dialog med datorn. Datorn frågar, Du matar in värden, datorn bearbetar och skriver ut resultatet.

Före varje **INPUT**-sats lägger Du vanligen en **PRINT**-sats, som kort förklarar vilken inmatning som skall följa på **INPUT**-satsens frågetecken.

Exempel *Beräkna ytan på en rektangel.*

```
> 10 PRINT "MATA IN BREDDEN";  
> 20 INPUT A  
> 30 PRINT "MATA IN LÄNGDEN";  
> 40 INPUT B  
> 50 PRINT "YTAN = ";A*B;" KVM"  
> 60 END
```

READ + DATA

Vid stora datamängder blir dialogformen ofta för långsam. Ibland har Du variabelvärden, som inte ändras under programmets gång. I dessa fall kan Du använda inmatning med **READ + DATA**.

Exempel

```
> 10 DATA 1.45, 2.14, 34  
> 20 READ A, B, C
```

DATA måste stå före **READ**, lämpligen i början av BASIC-programmet.

Du kan ha en hel följd av **DATA**-satser. De betraktas då som en enda följd av värden.

READ-satsen behöver inte läsa alla datavärden på en gång. En intern pekare håller reda på vilket värde som skall läsas nästa gång.

Exempel > 10 DATA 1, 2
> 20 DATA 3
> 30 READ A
> 40 PRINT A
> 50 READ A
> 60 PRINT A
> 70 READ A
> 80 PRINT A
> 90 END

> RUN

1
2
3

RESTORE

Återställer pekaren till första **DATA**-värdet och gör det möjligt att med **READ** komma åt **DATA**-värdena igen.

Exempel > 10 DATA 1
> 20 READ A
> 30 PRINT A
> 40 RESTORE
> 50 READ B
> 60 PRINT B
> 70 END

> RUN

1
1

LET

LET-satsen tilldelar värdet av uttrycket till höger om "=" till variabeln till vänster om "=". **LET**-satser används inne i program när delresultat behöver sparas.

Exempel > 10 LET A=5
 > 20 LET B=10
 > 30 LET C=A+B
 > 40 PRINT C
 > 50 END
 > RUN
 15

Creativision BASIC tillåter att Du utesluter ordet "LET".

Exempel > 10 A=5
 > 20 B=10
 > 30 C=A+B
 > 40 PRINT C
 > 50 END
 > RUN
 15

Ofta gäller det att öka eller minska värdet av en viss variabel. LET-satserna används då på följande sätt.

Exempel > 10 LET A=5
 > 20 LET A=A+2
 > 30 PRINT A
 > 40 END
 > RUN
 7

Observera att sats 20 inte skall tolkas som en likhet. När datorn kommer till sats 20, beräknar den först värdet av uttrycket till höger om likhetstecknet. Därefter tilldelas A detta värde. Man kan säga att innebörden av likhetstecknet i LET-satsen närmast är "tilldelas" eller "ersätts med".

Om Du gör fel

Anta att Du vill mata in satsen

20 INPUT A,B,C

och skriver

20 INPUT AB,

Du kan då korrigera sitt misstag på två sätt:

1. Tryck på **RET'N** och skriv om hela satsen. Den sist inskrivna satsen med nummer **20** gäller.
2. Radera ut tecknen bakåt med hjälp av tangenten ←. När Du Raderat tillräckligt många tecken, kan Du skriva dit de rätta.

Vill Du ta bort en hel sats är det bara att skriva satsnumret och trycka på **RET'N**.

LIST

När Du skrivit in ett program med många rättelser, vill Du gärna se programmet i rättad form. Kommandot **LIST** ger en utskrift av programmet som det har uppfattats av datorn.

Om Du vill se en speciell rad, låter Du **LIST**-kommandot åtföljas av radnummer. Vill Du se två rader, skriver man båda radnumren med kommatecken emellan.

LLIST

Har samma funktion som **LIST**, men ger utskrift på skrivare i stället för på **TV**-skärm.

REM

Det är ofta önskvärt, att skriva kommentarer i sitt program. T.ex. information om namn, vad programmet gör, hur man kör det osv. Kommandot **REM** (remark = anmärkning) kan skrivas var som helst i programmet och påverkar inte övriga funktioner i programmet.

Exempel > *10 REM DETTA ÄR EN ANMÄRKNING*
 > *20 REM BASIC-PROGRAMMERING*
 > *30 A=3*
 > *40 PRINT A*
 > *50 END*

 > *RUN*

 3

Hopp och slingor

Normalt utförs satserna i ett program i nummerordning. Med särskilda hoppinstruktioner kan denna ordning brytas.

GOTO

Hopp utan villkor.

```
Exempel  > 10 PRINT "KALLE"  
          > 20 GOTO 10  
          > 30 END  
  
          > RUN  
  
          KALLE  
          KALLE  
          KALLE  
  
          .  
          .  
          .
```

Du måste bryta detta program genom att trycka på **RESET**-tangentsen. Satsen, 20 GOTO 10, gör att programmet hoppar till sats 10.

IF-THEN

Hopp med villkor.

Om Du vill hoppa eller utföra någon sats i programmet bara då ett visst villkor är uppfyllt, används satsen

IF villkor **THEN** satsnummer eller sats

Betrakta satsen, 30 IF A=3 THEN 80. den ger datorn följande order:

Om A=3, dvs om villkoret är uppfyllt, så sker hopp till sats 80. I annat fall fortsätter programmet med nästa sats i turordning, dvs 40.

Följande symboler får användas i villkorssatserna i BASIC.

Symbol	Betydelse	Exempel
=	Lika med	$3 * x = 100$
<>	Ej lika med	$A < > 24$
<	Mindre än	$3 + B < A$
>	Större än	$A > B$
<= el. =K	Mindre eller lika med	$C < = A * B$
>= el. =<	Större eller lika med	$C > = 4$

Exempel *Vi läser in en följd av tal A. Om ett inläst tal $A=0$, så avbryter vi. Är talet > 25 , så skall det skrivas ut.*

```

> 10 DATA 11,26,34,8,23,56,0,43
> 20 READ A
> 30 IF A=0 THEN 80
> 40 IF A>25 THEN 60
> 50 GOTO 20
> 60 PRINT A
> 70 GOTO 20
> 80 END

> RUN

26
34
56

```

Sammansatta villkor

Villkorsdelen i en **IF-THEN**-sats kan bestå av flera olika villkor. De sammansätts då av de logiska operatorerna **AND** (=och) eller **OR** (=eller).

AND innebär att både villkoret till vänster om **AND** och villkoret till höger om **AND** skall vara uppfyllt, för att **THEN**-delen skall utföras.

OR innebär att något av villkoren till vänster eller höger om **OR** skall vara uppfyllt, för att **THEN**-delen skall utföras.

Ett villkor kan negeras genom att sätta in ett **NOT** före villkoret, som då skall stå inom parentes.

Exempel > 10 INPUT A,B,C
 > 20 IF A>B AND B>C THEN PRINT "A>B>C"
 > 30 IF NOT (A>B) OR NOT (B>C) THEN PRINT "A>B>C
 ÄR FEL"
 > 40 END
 > RUN
 ?10
 ?5
 ?7
 A>B>C ÄR FEL

GOSUB – RETURN

Hopp med återgång.

Om Du behöver utföra samma beräkning flera gånger, kan det vara lämpligt att använda en subrutin. En subrutin kan betraktas som ett självständigt programavsnitt, som vid lämpliga tillfällen anropas av ett annat program, huvudprogrammet.

Huvudprogram	Subprogram
100 GOSUB 400	400.....
110.....
.....
.....
200 GOSUB 400	450 RETURN
210.....	
.....	
.....	

Figuren visar ett huvudprogram, som två gånger anropar en subrutin. Med satsen, **100 GOSUB**, hoppar programmet ut till subrutinen. När beräkningen i subrutinen är klar, sker genom satsen, **450 RETURN**, återhopp till den sats i huvudprogrammet, som följer närmast efter sats **100**. I detta fall återgår vi till sats **110**. Samma förlopp upprepas vid det andra anropet.

Exempel *Du vill flera gånger i ett program beräkna och skriva ut kvadraten på variabeln A. Då kan Du använda en subrutin på följande sätt.*

```
> 10 LET A=2
> 20 GOSUB 100
> 30 LET A=A+3
> 40 GOSUB 100
> 50 LET A=A+7
> 60 GOSUB 100
> 70 GOTO 200
> 100 PRINT
> 110 PRINT "A I KVADRAT="; A **2
> 120 PRINT
> 130 RETURN
> 200 END

> RUN

A I KVADRAT=4
A I KVADRAT=25
A I KVADRAT=144
```

FOR-NEXT

Repetitionssatser.

Det är när samma arbetsmoment skall upprepas flera gånger som det lönar sig att använda en dator. Styrkan i ett program ligger därför ofta i att några satser genomlöps gång på gång. Den del av programmet, som utförs upprepade gånger brukar kallas en slinga eller en loop. Samma typ av beräkningar utförs varje gång i slingan, endast utgångsvärdena ändras. Slingan bildas med en speciell instruktion **FOR – NEXT**.

Exempel > 10 FOR N=1 TO 5 STEP 1
 > 20 PRINT N,N**2
 > 30 NEXT N
 > 40 END
 > RUN

1	1
2	4
3	9
4	16
5	25

Ovanstående exempel upprepar sats 20 fem gånger. Varje gång ökas värdet av variabeln N med ett. Steglängden kan ha ett annat värde än ett. Vi gör om exemplet så att vi beräknar kvadraten på alla udda tal mellan ett och tio.

```
> 10 FOR N=1 TO 10 STEP 2
> 20 PRINT N,N**2
> 30 NEXT N
> 40 END
> RUN
```

1	1
3	9
5	25
7	49
9	81

Exempel Beräkna summan av alla udda tal mellan ett och nittionio.

```
> 10 LET S=0
> 20 FOR X=1 TO 99 STEP 2
> 30 LET S=S+X
> 40 NEXT X
> 50 PRINT S
> 60 END

> RUN

2500
```

Om steglängden är ett (1), kan **STEP** utelämnas.

Alla **FOR**-slingor måste avslutas med en **NEXT**-sats. **NEXT**-satsen bör innehålla samma variabel som användes i startuttrycket i **FOR**-satsen.

Funktioner

Nedanstående s.k. biblioteksfunktioner finns som färdiga förprogrammerade rutiner i Creativision BASIC. De kan Du använda direkt i olika numeriska uttryck.

ABS(X)

Absolutvärdet av X. T. ex. $ABS(-34.67) = 34.67$

```
Exempel  > 10 PRINT ABS(3+4-6*5)
          > 20 END
          > RUN
          23
```

SGN(X)

Tecknet för X. Funktionen ger värdet +1 om X är ett positivt värde, Värdet 0 om X är 0 och - 1 om X är ett negativt värde.

```
Exempel  > 10 LET A=12
          > 20 PRINT SGN(A), SNG(A-A)
          > 30 END
          > RUN
          -1      0
```

RND(O)

Ger ett slumptal mellan 0 och 1.

Exempel > 10 FOR A=1 TO 5
 > 20 PRINT RND(O)
 > 30 NEXT A
 > 40 END
 > RUN
 0.6583
 0.63328
 0.53318
 0.13275
 0.53105

RND(N)

Ger ett slumpstal mellan 0 och N. N skall vara ett positivt tal.

Exempel > 10 FOR A=1 TO 5
 > 20 PRINT RND(10)
 > 30 NEXT A
 > 40 END
 > RUN
 7
 8
 4
 8
 3

SQR(X)

Ger kvadratroten ur X.

Exempel > 10 A=64
 > 20 PRINT SQR(A)
 > 30 END
 > RUN
 8

På samma sätt fungerar också följande förprogrammerade funktioner.
 Prova själv med egna exempel.

INT(X)

Ger heltalsdelen av X, det största heltalet mindre än eller lika med X.
 $\text{INT}(3.9) = 3$, $\text{INT}(-1.3) = -2$.

SIN(X)

Ger sinus för X. X anges i radianer.

COS(X)

Ger cosinus för X. X anges i radianer.

TAN(X)

Ger tangenten för X. X anges i radianer.

LOG(X)

Ger naturliga logaritmen av X.

EXP(X)

Exponentialfunktionen E_x . där $E = 2.71828$.

För $\text{SIN}(X)$, $\text{COS}(X)$ och $\text{TAN}(X)$ skall X vara $-1000 < X < 1000$.

Tabeller

En tabell är en uppsättning variabler, som alla har samma namn och endast åtskiljs av ett tal skrivet inom parentes efter variabelnamnet. Det är på detta sätt Du bygger upp tabeller i en dator. $A(I)$ är det I -te elementet i tabellen A .

```
Exempel  > 10 DIM A(6)
          > 20 FOR I=1 TO 6
          > 30 PRINT "GE BELOPP";
          > 40 INPUT A(I)
          > 50 LET S=S+A(I)
          > 60 NEXT I
          > 70 PRINT "SUMMA = ";S
          > 80 END
```

DIM (dimension) reserverar utrymme för tabeller. Du kan ha en eller två dimensioner på en tabell.

$A(6)$ är en endimensionell tabell med 6 element. $A(6,6)$ är en tvådimensionell tabell med $6 \times 6 = 36$ element. En tvådimensionell tabell kräver två tal inom parentes för att specificera varje element, precis som kolumner och rader i en vanlig tabell.

Om Du t.ex skall skapa en tabell med dimensionerna 3 och 5, använder Du **DIM**-satsen.

DIM(3,5)

Den ger $3 \times 5 = 15$ element

$A(1,1), A(1,2) \dots A(1,5)$

$A(2,1), A(2,2) \dots A(2,5)$

$A(3,1), A(3,2) \dots A(3,5)$

Exempel > 10 DIM A(3,3)
> 20 FOR I=1 TO 3
> 30 FOR J=1 TO 3
> 40 LET A(I,J)=1+J
> 50 PRINT A(I,J);
> 60 NEXT J
> 70 PRINT
> 80 NEXT I
> 90 END

> RUN

234

345

456

Teckensträngar

En följd av tecken omgivna av citationstecken, t.ex. "ABCDEF", kallas för en teckensträng. Vill Du efter **INPUT**-satsens frågetecken mata in text, måste speciella strängvariabler användas. En strängvariabels namn består av en bokstav följt av ett dollartecken, T.ex A\$.

```
Exempel  > 10 PRINT "VAD HETER DU"?  
          > 20 INPUT A$  
          > 30 PRINT "HEJ"; A$; "!!"  
          > 40 END  
  
          > RUN  
  
          VAD HETER DU:KALLE  
          HEJ KALLE!!
```

Maximala antalet tecken i en strängvariabel är 32 (trettio två).

Strängbehandling.

Creativision BASIC tillåter, att Du behandlar teckensträngar med operanden "+". "+" har då betydelsen sammanlänkning.

```
Exempel  > 10 LET A$="XYZ123"  
          > 20 LET P$=A$+"PQR"  
          > 30 PRINT P$  
          > 40 END  
  
          > RUN  
  
          XYZ123 PQR
```

```

Exempel  > 10 LET A$=""
          > 20 FOR I=1 TO 5
          > 30 LET A$=A$+"B"
          > 40 PRINT A$
          > 50 NEXT I
          > 60 END

          > RUN

          B
          BB
          BBB
          BBBB
          BBBB

```

LEFT\$

(stränguttryck, numeriskt uttryck), t.ex. **LEFT\$(A\$, N)**. Ger en delmängd av en sträng från och med det första tecknet från vänster till och med det N-te tecknet.

```

Exempel  > 10 LET A$="ABCDE"
          > 20 LET B$=LEFT$(A$, 3)
          > 30 PRINT B$
          > 40 END

          > RUN

          ABC

```

RIGHT\$

(stränguttryck, numeriskt uttryck), t.ex. **RIGHT\$(A\$, N)**. Ger en delmängd av en sträng bestående av de sista N tecknen.

```

Exempel  > 10 LET A$="ABCDEF"
          > 20 LET B$=RIGHT$(A$, 3)
          > 30 PRINT B$
          > 40 END

          > RUN

          DEF

```

MID\$

(stränguttryck, numeriskt uttryck, numeriskt uttryck), **MID\$(N,N)**

Ger en delmängd av en sträng från och med det M-te tecknet, N stycken tecken.

Exempel > 10 LET A\$="ABCDEFGH"
 > 20 LET B\$=MID\$(A\$, 3,4)
 > 30 PRINT B\$
 > 40 END

 > RUN

 CDEF

CHR\$

(numeriskt uttryck), **CHR\$(N)**

Ger ett tecken med **ASCII**-värdet (se kap 8) av N. Till exempel är **CHR\$(65)** lika med "A".

Exempel > 10 FOR I=65 TO 70
 > 20 PRINT CHR\$(I)
 > 30 NEXT I
 > 40 END

 > RUN

 A
 B
 C
 D
 E
 F

STR\$

(numeriskt uttryck), **STR\$(N)**.

Ger en sträng av ett numeriskt uttryck.

Exempel > 10 LET A\$=STR\$(0.125+0.5)
 > 20 LET B\$=A\$+"K"
 > 30 PRINT B\$
 > 40 END

 > RUN

 0.625K

LEN

(stränguttryck), t.ex. LEN(A\$).

Ger antalet tecken inklusive blanktecken i stränguttrycket.

Exempel > 10 LET A\$="ABCDEF"
 > 20 LET X=LEN (A\$)
 > 30 PRINT X
 > 40 END

 > RUN

 6

VAL

(stränguttryck), t.ex. val(A\$).

Ger det numeriska värdet av stränguttrycket.

Exempel > 10 LET A\$="124+1"
 > 20 LET X=VAL(A\$+"-100")
 > 30 PRINT
 > 40 END

 > RUN

 24

ASC

(stränguttryck), t.ex. ASC(A\$).

Ger **ASCII**-koden för det första tecknet i stränguttrycket.

Exempel > 10 LET X=ASC("AXY")
 > 20 PRINT X
 > 30 END
 > RUN
 65

Strängjämförelser

Relationsoperatorer kan Du använda på stränguttryck på samma sätt som på numeriska uttryck.

Jämförelsen sker genom att jämföra tecknens **ASCII**-värden (se kap 8).

Exempel > 10 A\$="AA"
 > 20 B\$="BA"
 > 30 IF A\$<B\$ THEN PRINT 20
 > 40 END
 > RUN
 20

DÅ ASCII-VÄRDET av A är mindre än **ASCII**-värdet av B ($65 < 66$), är A\$ mindre än B\$.

Jämförelsen sker från vänster till höger, vid först påträffade olikhet avbryts jämförelsen.

Användning Av Systemminnet

Följande två kommandon ger möjlighet att låta BASIC-program lagra och hämta data från systemets interna minne

PEEK

PEEK-kommandot hämtar det värde (mellan 0 och 255) som finns lagrat i en viss minnesadress.

Syntax: **PEEK** (adress)

Exempel: > *10 PRINT "ADRESS" "LAGRAD DATA"*
 > *20 FOR K=0 to 20*
 > *30 PRINT K, PEEK(K)*
 > *40 NEXT K*

POKE

Medan **PEEK**-kommandot användes för att läsa innehållet i en minnesadress, så gör **POKE** motsatsen, dvs skriver ett värde i en viss minnesadress.

Syntax: **POKE** adress, värde

Exempel: > *POKE 300, 48*

Denna instruktion kommer att skriva värdet 48 i minnesadress 300. Prova att hämta tillbaka värdet med

 > *PRINT PEEK (300)*

En praktisk användning av **PEEK** och **POKE**-kommandona har man då data skall överföras mellan två olika BASIC-program. Det första programmet kan då spara ex vis resultatet av en beräkning genom **POKE**-kommandot. Då nästa program laddats in finns värdena kvar och kan hämtas med **PEEK**-kommandot. Om Du har tillgång till kasettminne prova med följande exempel.

Exempel: > 1 REM ** PROGRAM A **
> 10 INPUT A\$
K 20 L= LEN(A\$)
> 30 POKE 300, L
> 40 FOR I=1 TO L
> 50 X\$=MID\$(A\$, I, 1)
> 60 A=300+I
> 70 POKE A, ASC (X\$)
> 80 NEXT I
> 90 PRINT "PROGRAM A KLART"
> 100 CRUN

> CSAVE (Spara programmet på kasettminnet)

> NEW (Rensa programminnet)

> 1 REM ** PROGRAM B **
> 10 L=300+PEEK(300)
> 20 FOR I=301 TO L
> 40 V=PEEK(I)
> 50 PRINT CHR\$(V);
> 60 NEXT I

> CSAVE (Spara programmet efter program A)

Backa tillbaka kasettminnet till det första programmet ge kommandot >**CRUN** och tryck på **PLAY** på kasettminnet.

Det första programmet laddas då in och kommer att begära en inmatning. Mata in en text och följ vad som händer.

Programkontroll

För att ge möjlighet att styra körningen av ett program, finns det ett antal instruktioner och kommandon. Instruktionerna är **STOP** och **END** och kommandona är **CONT** och **CNTL/C**. Du kan också avbryta ett program med **RESET**-knappen.

STOP

STOP-instruktionen avbryter ett program och ger kontrollen tillbaka till operatören. Du kan ha flera **STOP**-instruktioner i ett och samma program. När **STOP**-instruktionen utförts skrivs texten.

STOP AT sats nummer

```
Exempel  > 10 PRINT "TEST AV STOP"
          > 20 PRINT 123
          > 30 STOP
          > 40 END
          > RUN
          TEST AV STOP
          123
          STOP AT 30
```

CONT

Kommandot **CONT** (continue = fortsatt) använder Du när programmet har stoppats med **STOP**-instruktionen. Programmet fortsätter med satsen som följer direkt efter **STOP**-satsen.

STOP-och **CONT** är mycket användbara vid felsökning i program. Du lägger in **STOP**-satser på lämpliga ställen och när programmet är avbrutet, kan Du undersöka värdet i olika variabler med direktmetoden (dvs instruktioner utan radnummer). Du kan också ändra värden på variabler. När Du är klar, startar Du programmet med **CONT**.

CNTL/C

Du kan också avbryta programmet genom att trycka ner tangenterna **CNTL** och **C** samtidigt (håll nere **CNTL** och tryck på **C**). Du kan starta programmet igen med **CONT**-kommandot.

END

END-instruktionen används för att avsluta körningen av ett program.

Det måste vara den sista satsen i ett program. Ett program som avbrutits av **END** kan inte startas igen med **CONT**-kommandot. **END**-satsen kan uteslutas, men det är god programmeringssed att ha med den.

RESET

RESET-knappen sitter på konsolen intill **POWER-OFF-ON**-knappen. När Du trycker på **RESET**, avbryts programmet. Dessutom återställs färg och grafik till normalläge.

4 SPECIALFUNKTIONER

GRAFIK

Med grafik-kommandona kan Du använda bildytan på Din TV uppdelad i 24 rader om vardera 32 kolumner. De 28 tecken som får plats på varje rad då Du skriver ett BASIC program upptar kolumn 3 – 30. På inställningen på olika TV – apparater syns ibland inte de två första resp. sista kolumnerna på raden. Vi rekommenderar Dig därför att endast använda kol 3 – 30 då du använder de grafiska kommandona.

De 4 grafiska kommandona är:

- CLS – (Clear Screen) Rensar TV-bilden
- COLOR – Anger färger
- CHAR – Definera valfria tecken
- PLOT – Skriver tecken på valfri plats på bilden

LJUD

Med ljud-kommandot **SOUND** kan Du använda BASIC att alstra ljud som styrs från Ditt program. Det finns 3 ljudkanaler som kan styras oberoende av varandra.

JOY-STICKS

Din CreatiVision är i grundutförande utrustat med två stycken styrpinnar, s k Joy-sticks. Läget på styrpinnarna och de tillhörande avfyrningsknapparna kan läsas in till BASIC-programmet med **JOY**-kommandot.

CLS

Clear Screen. Blankar ut bilden

CLS kommandot använder Du då hela bilden skall rensas. Då **CLS** kommandot utförs kommer hela bilden att fyllas med blanktecken (space).

Exempel > **10 CLS**
 > **20 PRINT "CLEAR SCREEN"**
 > **RUN**

När programmet körs rensas hela bildytan ut och texten "**CLEAR SCREEN**" skrivs längst ner på TV-bilden.

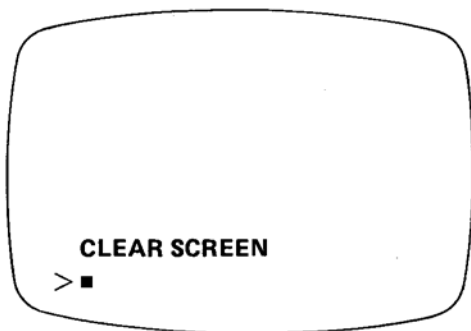


Fig. 4.1

COLOR

Med **COLOR** kommandot kan du själv ange färg på tecknen som skrivs på bilden.

Syntax: **COLOR** tecken-group, förgrundsärg-kod, bakgrundsfärgkod

Exempel **COLOR 2, 10, 14**

Varje tecken som skrivs på bilden har egentligen två färger. Den färg som själva tecknet har kallas förgrundfärg, medan den färg som omger tecknet kallas bakgrundsfärg. Din Creativision ger Dig möjlighet att välja mellan 16 färger som kan användas som både förgrunds – och bakgrundsfärg. Följande färger finns:

Färg-kod	Färg
1	Transparent (genomskinlig)
2	Svart
3	Grön
4	Ljusgrön
5	Mörkblå
6	Ljusblå
7	Mörkröd
8	Cyan (himmelsblå)
9	Röd
10	Ljusröd
11	Mörkgul
12	Ljulgul
13	Mörkgrön
14	Violett
15	Grå
16	Vit

För att använda **COLOR** kommandot måste Du också ange vilken teckengrupp som tecknet du vill färglägga tillhör. Listan över vilka teckenkoder som tecknen på tangentbordet har finns i Kapitel 8. Tecknen är indelade i följande teckengrupper:

Teckengrupp		Teckengrupp	
— nr	tecken kod	— nr	tecken kod
1	0 — 7	17	128 — 135
2	8 — 15	18	136 — 143
3	16 — 32	19	144 — 151
4	24 — 31	20	152 — 159
5	32 — 39	21	160 — 167
6	40 — 47	22	168 — 175
7	48 — 55	23	176 — 183
8	56 — 63	24	184 — 191
9	64 — 71	25	192 — 199
10	72 — 79	26	200 — 207
11	80 — 87	27	208 — 215
12	88 — 95	28	216 — 223
13	96 — 103	29	224 — 231
14	104 — 111	30	232 — 239
15	112 — 119	31	240 — 247
16	120 — 127	32	248 — 255

Observera att alla positioner som är tomma på bilden, i själva verket innehåller blank-tecken (space). Om du anger teckengrupp 5 i ett **COLOR** kommando kommer Du att färglägga teckenkoderna 32 — 39 dvs också blank-tecknet (teckenkod 32). Alla tomma positioner på bilden kommer att få den bakgrundfärg som du angivit.

```
Exempel  > 100 REM ** COLOR KOMMANDOT **
          > 200 CLS
          > 300 COLOR 5, 6, 6
          > 400 GO TO 400
          > RUN
```

Bakgrundsfärgen på bilden kommer att ändras till mörkblå. Avbryt programmet genom att trycka **CTL — C**.

CHAR

Med **CHAR** kommadot kan Du definiera Dina egna tecken och grafiska symboler. Du kan skapa helt nya tecken ex. Å, Ä och Ö.

Syntax: **CHAR** tecken-kod, tecken-beskrivning

Tecken-koden anger den kod som du vill använda för det nya tecknet. Koden skall vara en siffra mellan 1 – 255. I kapitel 8 kan du se vilka tecken-koder som används för tecknen på tangentbordet.

Tecken-beskrivningen är 16 koder (Ø-F) som beskriver det mönster som skall bilda det nya tecknet. De 16 koderna representerar de 64 punkter som alla tecken på bilden är uppbyggda av.

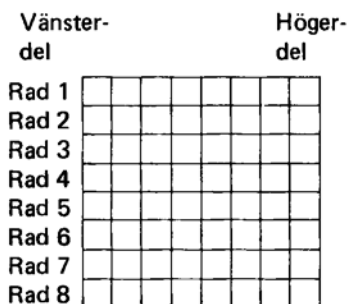


Fig. 4.2

Varje rad i tecknet är uppdelat i en vänster- och en högerdel

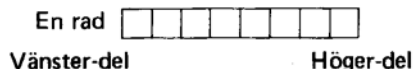


Fig. 4.3

Varje kod i tecken-beskrivningen beskriver vilka punkter i vänster resp höger-delen av en rad som skall vara fyllda. Du anger koderna i ordningen vänster-del, höger-del för varje rad i tecknet, uppifrån och ned. Dvs de första 2 koderna anger mönstret i tecknets första rad, nästa 2 koder beskriver rad 2 osv.

Tecknet byggs upp genom att låta vissa punkter vara 'på' och andra 'av'. För att beskriva det nya tecknet måste Du ange för datorn vilka punkter som skall vara på resp av i de 16 höger resp vänsterdelar som tecknet är uppbyggt av.

Om Du anger mindre än 16 koder kommer datorn att sätta de kvarvarande till 0. Om Du anger fler än 16 koder kommer datorn att ignorera de övertaliga.

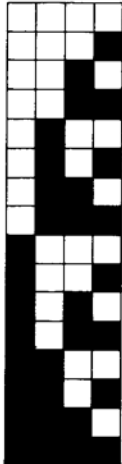
Block	Kod	0=av, 1=på
	0	0000
	1	0001
	2	0010
	3	0011
	4	0100
	5	0101
	6	0110
	7	0111
	8	1000
	9	1001
	A	1010
	B	1011
	C	1100
	D	1101
	E	1110
	F	1111

Fig. 4.4

Exempel *CHAR 40, 18 18 00 18 3C 66 7E 42*

		KOD
Rad 1		18
Rad 2		18
Rad 3		00
Rad 4		18
Rad 5		3C
Rad 6		66
Rad 7		7E
Rad 8		42

Fig. 4.5

Du har nu definierat tecknet "" vänsterparentes till att bli ett Å. Varje gång du trycker på ""-knappen kommer tecknet "Å" att skrivas på bilden. Genom att trycka på **RESET**-knappen kommer tangentbordet att återställas. På motsvarande sätt kan du byta ut två andra tecken att bli Ä resp Ö.

CHAR 41.00 24 00 18 3C 66 7E 42

Ä på ")"-tangenten

CHAR 42, 00 24 00 3C 66 42 66 3C

Ö på "*" -tangenten

PÅ motsvarande sätt kan man bilda

CHAR 41,00 3E 68 48 7E 48 48 4E

Danskt Ä

Självklart kan Du på detta sätt bygga upp tecken av vilket utseende Du vill. Använder Du tecken-koder som inte mostvaras av någon tangent på tangent-bordet måste Du använda **PLOT** kommandot för att skriva tecknet på bilden. Kom bara ihåg att om Du med **CHAR**-kommandot ändrar ett tecken som redan finns skrivet på bilden, så kommer tecknet att ändras på alla ställen där det finns.

PLOT

Med **PLOT** kommandot kan Du skriva ett tecken på vilken plats som helst på bilden. Bilden är uppdelad i 24 rader om vardera 31 kolumner.

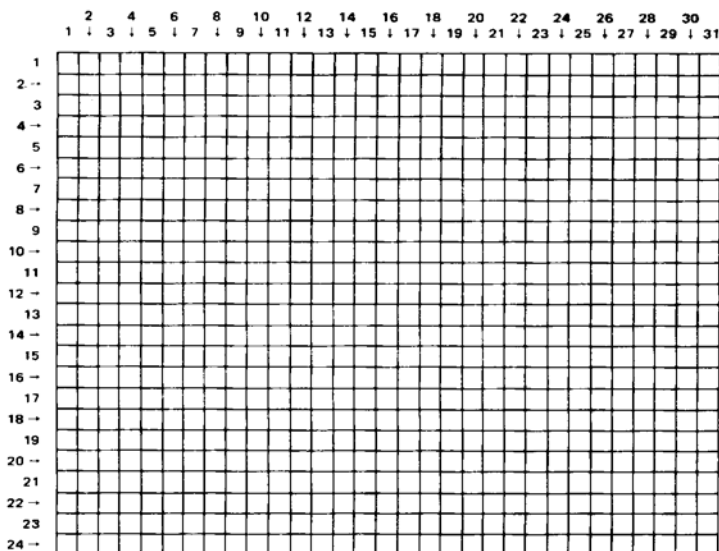


Fig. 4.6

Kolumn 1 är den första kolumnen i varje rad.

Rad 1 är den första raden i varje kolumn.

Raderna och kolumnerna bildar tillsammans ett 'nät' som bilden visar. Med **PLOT** kommandot kan Du skriva ett tecken i vilken ruta som helst i "nätet".

Syntax: **PLOT** kolumn-nr, rad-nr, tecken-kod

Kolumn-nr anger vilken kolumn (1-31) som tecknet skall skrivas i. Rad-nr anger på vilken rad (1-24) som tecknet skall skrivas på. Tecken-koden anger vilket tecken som skall skrivas (0-255). Tecken-koderna för de vanliga tecknen finner Du i kapitel 8. Du kan också skriva tecken som definierats med **CHAR** kommandot.

Exempel > 10 CLS
 > 20 COLOR 5, 4, 6
 > 30 CHAR 33, 18 18 FF 3C 7E FF 14 26
 > 40 FOR K=1 TO 32
 > 50 PLOT K, 10, 33
 > 60 NEXT K
 > 70 GOTO 70
 > RUN

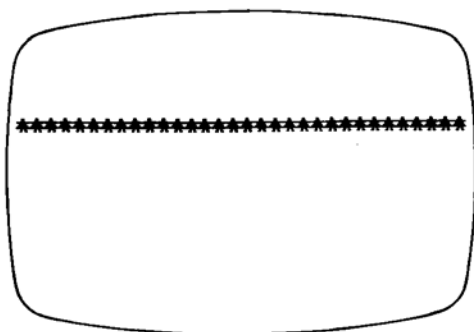


Fig. 4.7

Programmet skriver en rad med tecken på rad 10.
Bryt programmet med **CTL — C**.

Exempel > 05 CLS
> 10 FOR A=1 TO 16
> 20 COLOR A+16, A, A
> 30 NEXT A
> 40 FOR X=1 TO 32
> 50 FOR Y=1 TO 20
> 60 PLOT X,Y,X*4+128
> 70 NEXT Y
> 80 NEXT X
> 90 GO TO 90
> RUN

Programmet kommer att skriva ränder med olika färger över bilden.

SOUND

Med CreatiVision **BASIC** kan Du också skriva program som alstrar toner genom **TV**-apparatens högtalare. Det finns 3 musikkanaler som Du kan programmera oberoende av varandra.

Syntax: **SOUND** frekvens-kod; tonlängds-kod, frekvens-kod; tonlängds-kod, . . .

Frekvens-koden anges i form av en tal mellan 1 och 32 och styr vilken ton som skall ljuda. Tonlängds-koden är ett tal mellan 0 och 8 och anger hur länge tonen skall ljuda.

Första paret av frekvens- och tonlängdskod kommer att styra den första musikkanalen, andra paret styr andra kanalen och tredje paret styr den tredje kanalen.

Frekvens-kod	ton	Frekvenskod	ton
1	ingen ton	17	D [#] , Eb
2	C	18	E
3	C [#] , Db	19	F
4	D	20	F [#] , Gb
5	D [#] , Eb	21	G
6	E	22	G [#] , Ab
7	F	23	A
8	F [#] , Gb	24	A [#] , B
9	G	25	H
10	G [#] , Ab	26	C
11	A	27	C [#] , Db
12	A [#] , B	28	D
13	H	29	D [#] , Eb
14	C	30	E
15	C [#] , Db	31	F
16	D	32	Ingen ton

Tonlängds-kod	varaktighet	
0	16-del	
1	8-del	
2	punkterad 8-del	
3	4-del	
4	punkterad 4-del	
5	1/2-not	
6	punkterad 1/2-not	
7	helnot	

För att lägga in pauser i musiken kan du använda frekvenskod 1 eller 32 så kommer tonlängdskoden att ange pausens längd.

Exempel *SOUND 4;5, 6;7, 7;7*

Kommer att spela en treklang

Exempel: > 10 REM ** BIG BEN **
 > 20 SOUND 26;3, 21;3, 14;3
 > 30 SOUND 30;3, 25;3, 18;3
 > 40 SOUND 28;3, 23;3, 16;3
 > 50 SOUND 21;5, 16;5, 9;5
 > 60 SOUND 26;3, 21;3, 14;3
 > 70 SOUND 28;3, 23;3, 16;3
 > 80 SOUND 30;3, 25;3, 18;3
 > 90 SOUND 26;7, 21;7, 14;7
 > 100 GOTO 20

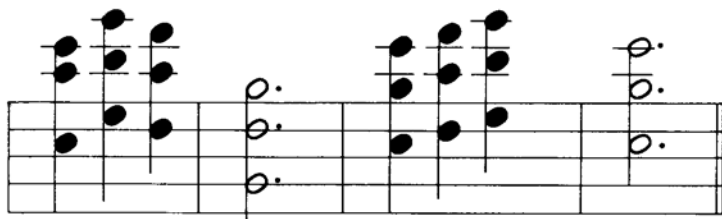


Fig. 4.8

> RUN

Datorn kommer att spela ovanstående melodi.

JOY

Med **JOY** kommandot kan **BASIC** programmet läsa in information om läget på de två styr-pinnarna (s-k Joy-sticks). Kommandot är särskilt användbart då Du vill skriva egna spel i **BASIC**.

Tangentbordet på din Creativision är delat i en vänster- och en högerdel. Varje del har en styr-pinne och två avfyrnings-knappar.

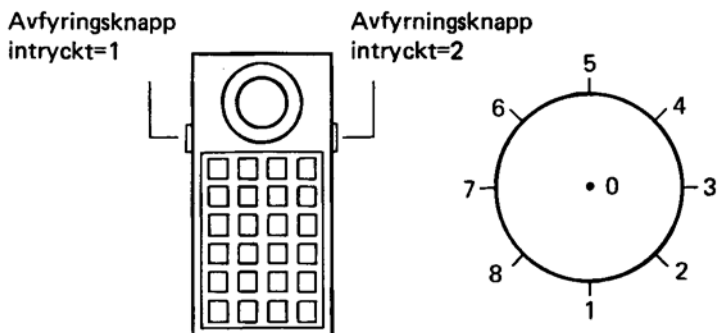


Fig. 4.9 Visar vilka värden som erhålls beroende på styr-pinnens resp avfyrningsknapparnas läge.

Syntax: **JOY(N)** Där N anger vilken styrpinne enl följande

JOY(1) Läget för vänster styrpinne

JOY(2) Läget för höger styrpinne

JOY(3) Vänster avfyrningsknappar

JOY(4) Höger avfyrningsknappar

Exempel > 10 PRINT JOY(1);
 > 20 GOTO 10
 > RUN

Programmet kommer att skriva ut siffror mellan 0 och 8 beroende på åt vilket håll du för vänster styrpinne.

Exempel > 05 CLS
 > 10 LET A=JOY(1)
 > 20 IF A=3 THEN GOTO 100
 > 30 IF A=7 THEN GOTO 200
 > 40 GOTO 10
 > 100 PLOT 4,10,32
 > 110 PLOT 28,10,65
 > 120 GOTO 10
 > 200 PLOT 28,10,32
 > 210 PLOT 4,10,65
 > 220 GOTO 10
 > RUN

Programmet kommer att skriva ut ett "A" på vänster eller höger sida av bilden, beroende på åt vilket håll du för vänster styrpinne.

5 Användning av kassetminne

Man kan bygga ut sitt Creativision Datorsystem genom att ansluta en kassettenhet. Creativision **BASIC** ger möjlighet att lagra resp. ladda program med hjälp av kassetband.

Anslut kassettenheten enligt instruktionen, som medföljer kassettenheten.

CSAVE

Lagra program på kassett.

1. Anslut kassettenheten.
2. Sätt i en kassett av god kvalitet.
3. Tryck ner **PLAY-** och **RECORD-**tangenterna. Notera, att kassettenheten inte startar. Datorn kontrollerar kassettenhetens motor. Därför startar inte kassettenhetens motor förrän datorn beordrar den att starta.
4. Notera räkneverkets ställning. Det har Du användning för när Du skall läsa in programmet i datorn igen.
5. Ge kommandot **CSAVE** och tryck på **RET'N**. Kassettenheten börjar nu spela in programmet samtidigt listas programmet på skärmen.
6. När programmet är färdiginspelat, skrivs ett **>**-tecken på skärmen. Tryck då på **STOP**-tangenten på kassettenheten.

CLOAD

Läs in ett program till datorn från kassettenheten.

1. Anslut kassettenheten.
2. Lägg i kassetten och spola fram till rätt position enligt räkneverket.
3. Tryck ner **PLAY**-tangenten.
4. Ge kommandot **CLOAD** och tryck på **RET'N**. Programmet förs nu över till datorn samtidigt som det listas på skärmen.

5. När programmet är inladdat skrivs ett >-tecken på skärmen. Tryck på **STOP**-tangentsen på kassettenheten.

Anm. Om kassetten är en Creativisionkassett, kommer programinläsningen att åtföljas av tal- och ljudillustrationer över **TV**-apparaten.

CRUN

Samma funktion som **CLOAD + RUN**. Programmet läses in i datorn och startas automatiskt.

Följ proceduren för **CLOAD**.

OBS !

Kommandona **CSAVE**, **CLOAD** och **CRUN** ger alla automatiskt en listning av programmet på skärmen. Om man vill undvika detta, gör man tillägget **(,N)** efter kommandot. T.ex **CSAVE,N**.

6 Övningar

- Övn. 1. Skriv ett program som beräknar volymen av ett klot.
Indata: Diametern D
Formel: Volymen $V = \frac{\pi D^3}{6}$
Utdata: Volymen V
- Övn. 2. Skriv ett program som beräknar bromssträckan för en bil. För en torr väg gäller att bromssträckan S i meter vid hastigheten V i km/h kan beräknas med formeln $S = 0.2V + 0.006V^2$.
Låt programmet skriva ut en tabell över V och S . Hastigheten V kan få anta värdena 20, 30, 40, ..., 140 km/h.
- Övn. 3. Skriv ett program som undersöker hur många termer det behövs för summan
- Övn. 4. $1^2 + 2^2 + 3^2 + 4^2 + \dots$
ska överskrida 5000.
- Övn. 4. Skriv ett program som beräknar hypotenusan i en rätvinklig triangel med hjälp av Pythagoras sats.
Indata: Kateterna A och B
Formel: Hypotenusan $C = A^2 + B^2$
Utdata: Hypotenusan C

Övn. 5. Skriv ett program som beräknar radien i en cirkel med given area.

Indata: Areal A

Formel: Radien $R = \sqrt{A/\pi}$

Utdata: Radien R

Övn. 6. Hur stort blir ett kapital som med ränta på ränta får tillväxa under ett visst antal år.

Indata: Antalet år N
Startkapital K
Räntesats P

Utdata: Antalet år N
Slutkapitalet

Lösningsförslag

- Övn. 1. > 10 PRINT "DIAMETERN";
 > 20 INPUT D
 > 30 LET V=3.1416*D**3/6
 > 40 PRINT "VOLYMEN=";V
 > 50 END
- Övn. 2. > 10 FOR V=20 TO 140 STEP 10
 > 20 LET S=0.2*V+0.006*V**2
 > 30 PRINT V,S
 > 40 NEXT V
 > 50 END
- Övn. 3. > 10 LET S=0
 > 20 FOR X=1 TO 100
 > 30 LET S=S+X**2
 > 40 IF S>5000 THEN 60
 > 50 NEXT X
 > 60 PRINT "X=";X;"GER S=";S
 > 70 END
- Övn. 4. > 10 REM PYTHAGORAS SATS
 > 20 PRINT "ANGE KATETERNA A OCH B";
 > 30 INPUT A,B
 > 40 LET C=SQR(A**2+B**2)
 > 50 PRINT "HYPOTENUSAN=";C
 > 60 END

- Övn. 5.
- ```
> 10 REM CIRKELNS RADIE
> 20 PRINT "AREAN=";
> 30 INPUT A
> 40 LET R=SQR (A/3.1416)
> 50 PRINT "RADIEN=";R
> 60 END
```
- Övn. 6.
- ```
> 10 PRINT "MATA IN N,K OCH P";
> 20 INPUT N,K,P
> 30 FOR T=1 TO N
> 40 LET K=K*(1+P/100)
> 50 NEXT T
> 60 PRINT "EFTER";N;"ÅR"
> 70 PRINT "ÄR KAPITALET="; K
> 80 END
```

7 Sammanfattning Av Kommandon Och Instruktioner

KOMMANDON

Kommando	Beskrivning	Sida
NEW	Raderar programmet från minnet	3 – 2
LIST	Listar hela eller del av programmet LIST, 100 Listar till sats 100 LIST 50, 80 Listar satser mellan 50 och 80 LIST Listar hela programmet	3 – 12
LLIST	Som LIST men skriver på skrivaren	3 – 12
RUN	Kör programmet	3 – 2
CLOAD	Laddar program från kasettminne CLOAD, N Laddar utan att visa programmet	5 – 1
CSAVE	Sparar programmet på kasettminnet CSAVE, N Sparar utan att visa programmet	5 – 1
CRUN	Som CLOAD , men kör dessutom programmet	5 – 2
CONT	Continue, Återstartar ett program som har avbrutits med STOP -instruktionen.	3 – 32
CNTL/C	Stannar programmet	3 – 32

INSTRUKTIONER

Instruktion	Beskrivning	Sida
DATA	Lagrar värden i programmet för läsning av READ -instruktion	3 – 9
DIM	Reserverar utrymme för tabeller 10 DIM A\$(5) Tabell för 5 strängvariabler 50 DIM I (20) Tabell för 20 värden	3 – 23
END	Avslutar programmet.	3 – 32
FOR .. TO ..	Utför ett programavsnitt det antal gånger som bestäms av loop-variabeln.	3 – 17
GOSUB	Hopp till subrutin. 100 GOSUB 400	3 – 16
GOTO	Hopp till radnummer 10 GOTO 55	3 – 14
IF .. THEN ..	Villkorlig instruktion 10 IF A>10 THEN GOTO 25	3 – 14
INPUT	Läser data från tangentbordet till variabel 20 INPUT A\$	3 – 9
LET	Tilldelning av värde till variabel 10 LET A\$="CREATIVISION"	3 – 10
LPRINT	Som PRINT , men skriver på skrivaren. 10 LPRINT "CREATIVISION"	3 – 8
NEXT	Avslutar FOR -slinga 150 NEXT I	3 – 17

Instruktion	Beskrivning	Sida
PRINT	Skriver på TV -skärmen 10 PRINT "MERA CREATIVISION"	3 – 6
READ	Läser data från DATA -sats 15 READ A\$, I	3 – 9
REM	Remarks. Kommentarer till programmet 1 REM ** PROGRAM A ***	3 – 12
RESTORE	Återställer pekaren så att nästa READ kommer att börja läsa från första DATA -sats.	3 – 10
RETURN	Avslutar subrutin.	3 – 16
STOP	Stannar programmet. Kan återstartas med CONT	3 – 32
TAB(N)	Styr efterföljande utskrift N kolumner till höger 10 PRINT "AAAAA"; TAB (5); "BBBBB"	3 – 7

STRANGINSTRUKTIONER

Instruktion	Bestrivning	Sida
ASC(A\$)	Ger ASCII -koden för första tecknet i A\$	3 – 28
CHR(I)	Ger tecknet som har ASCII -koden I	3 – 27
LEFTS(A\$, I)	Ger de första I tecknen i strängen A\$	3 – 26
LEN(A\$)	Ger antal tecken i strängen A\$	3 – 28
MIDS(A\$,I,J)	Ger J tecken från och med position I från A\$	3 – 27
RIGHTS(A\$,I)	Ger alla tecken i A\$ från och med position I	3 – 26
STR\$(I)	Gör om värdet i I till teckensträng	3 – 27
VAL(A\$)	Ger värdet av A\$ tolkat som ett tal	3 – 28

MATEMATISKA FUNKTIONER

Funktion	Beskrivning	Sida
ABS(X)	Ger absolutvärdet av X	3 – 20
COS(X)	Ger cosinus av X. X är vinkel i radianer	3 – 22
EXP(X)	Ger exponential funktionen E_x	3 – 22
INT(X)	Ger naturliga logaritmen för X	3 – 22
LOG(X)	Ger naturliga logaritmen för X	3 – 22
RND(O)	Ger slumpal mellan 0 och 1	3 – 20
RND(N)	Ger slumpal mellan $\emptyset - N$	3 – 21
SGN(X)	Ger tecknet för X. 1=positivt, -1=negativt, 0=0	3 – 20
SIN(X)	Ger sinus för X radianer	3 – 21
SQR(X)	Ger roten av X	3 – 21
TAN(X)	Ger tangens för X radianer	3 – 22

SPECIALFUNKTIONER

Instruktion	Beskrivning	Sida
CLS	Blankar ut TV-bilden	4 – 2
CHAR	Definierar tecken CHAR 41,000 24 00 18 3C 66 TE 42	4 – 5
COLOR	Anger färg för tecken COLOR 2, 10, 14	4 – 3
JOY	Ger värden som anger position på joy-sticks och avfyrningsknappar. 10 PRINT JOY(1)	4 – 14
PLOT	Skriver ett tecken på valfri plats på TV-bilden PLOT K,R, 33	4 – 9
SOUND	Skapar ljud i TV-högtalaren Sound 2, 5; 3, 6; 4, 7	4 – 11

8 Teckenkoder

Tecknen som skrivs in i datorn lagras enligt en internationell standard, den s k **ASCII**-koden. Varje tecken (bokstav, siffra eller specialtecken) har tilldelats en speciell kod. Nedan följer en uppställning av koderna för de tecken som går att skriva in via tangentbordet. Observera dock att med **CHAR**-kommandot kan man i CreatiVision definera nya tecken och då utnyttja alla koder mellan 0 och 255.

Teckenkod	Tecken	Teckenkod	Tecken
32	blank	64	@ a'— tecken
33	!	65	A
34	"	66	B
35	#	67	C
36	\$	68	D
37	%	69	E
38	&	70	F
39	'	71	G
40	(72	H
41)	73	I
42	*	74	J
43	+	75	K
44	,	76	L
45	—	77	M
46	.	78	N
47	/	79	O
48	0	80	P
49	1	81	Q
50	2	82	R
51	3	83	S
52	4	84	T
53	5	85	U
54	6	86	V
55	7	87	W

forts . . .

Teckenkod	Tecken	Teckenkod	Tecken
56	8	88	X
57	9	89	Y
58	:	90	Z
59	;		
60	<		
61	=		
62	>		
63	?		

Observera att de s k nationella tecknen, ex Å, Ä och Ö inte finns i **ASCII**-koden. Dessa kan dock lätt skapas i CreatiVision, se kapitel 4.

9 Felmeddelanden

Om Creativision BASIC upptäcker ett fel som får programmet att stanna, skrivs en felkod ut tillsammans med den felaktiga satsen. Felkoderna förklaras här nedan.

0. Fel vid inladdning av program från kassett.
1. **NEXT** utan tillhörande **FOR**.
2. **RETURN** utan **GOSUB**.
3. Hopp till radnummer som inte finns.
4. Operand saknas.
5. Syntaxfel (en instruktion är felskriven).
6. Mer än sex signifikanta siffror.
7. Felaktigt gjorda **FOR-NEXT**-slingor.
8. Felaktigt gjorda **GOSUB**-slingor.
9. Systemfel.
10. Stack overflow.
11. Felaktigt skriven operand.
12. Felaktig **IF**-sats.
13. Antalet höger-och vänsterparenteser ej lika.
14. Parentesnivåfel.
15. Sträng saknas.
16. Fel vid utvärdering av en sträng.
17. Division med noll (ej tillåtet).
18. För många **READ** (**DATA**-värdena räcker ej).
19. **DATA** för stor.
20. **DIM** felskriven.
21. Felaktig stränglängd.

©CREATIVISION
BASIC
INTERPRETER



© 1982 VTL
MADE IN HONG KONG
91-0092-06