INTRODUKTION TILL
ZX81
BASIC

BECKMAN
Beckman Innovation AB
Telefon 08-39 04 00 Telex 10318
Stora Gungans väg 14 Box 7
S-122 21 ENSKEDE SWEDEN

ISBN 91 7382 508 5

BECKMAN
Beckman Innovation AB

Mikael Bonnier
tel: 046/293028

EN INTRODUKTION TILL ZX81 BASIC

INNEHÅLLSFÖRTECKNING

INLEDNING

ZX 81 - FOLKDATORN?

När ZX80 presenterades för facktidningarna 1980 talade man om "folkdatorn" som skulle ge alla möjlighet att få tillgång till datorkraft. Nu har ZX80 utvecklats till ZX81 och blivit både billigare och betydligt mer kraftfull än tidigare. Än större anledning att tala om folkdator? Javisst, men lika gärna utbildningsdator på grund av de utmärkta programmeringsegenskaperna. Kompletteras sedan datorn med skrivare och yttre minne finns också anledning att tänka i benämningar som "bruksdator".

För flera av läsarna till denna 'Introduktion till ZX81" är detta antagligen den första riktigt handgripliga kontakten med en teknik som är på väg att förändra hela vårt samhälle. Vi har svårt att se ett bättre val än det Du gjort! ZX81 är en lättillgänglig dator som är enkel att lära sig och ger stora möjligheter. Hur stora framgår inte helt av denna introduktion som till största delen är en översättning av den engelska manualen. Nu finns omfattande litteratur att tillgå framtagen av Studieförlaget i Uppsala. Med denna och datorn ZX81 har Du alla förutsättningar att få mycket stort utbyte av Ditt datorintresse. Inte bara i form av nöjet att arbeta med ZX81 utan än mer för den kunskap det ger om den nya tekniken - datortekniken!

LYCKA TILL!

Ulf Beckman
Beckman Innovation AB

Thomas Eriksson
Studieförlaget i Uppsala

KAPITEL 1

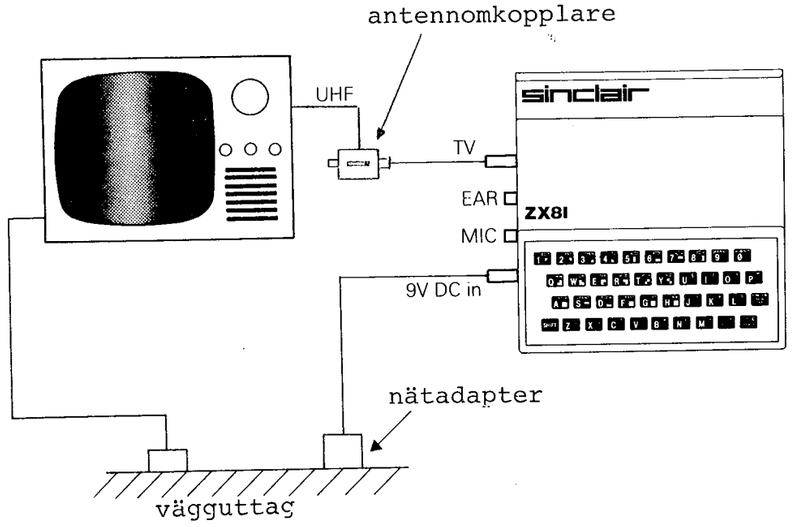
UPPKOPPLING AV DATORN ZX80/81

Datorn ZX81 (alternativt ZX80 med 8 kROM) levereras med:

1. Engelsk manual
2. Denna översättning av den engelska manualen
3. Nätadapter
4. En koaxialkabel för anslutning till TV:ns antennuttag
5. En dubbelledare för anslutning till bandspelare
6. Antennomkopplare (i vissa fall)

En vanlig hem-TV används som bildskärm där program och utdata presenteras. Det är nödvändigt att TV-apparaten har UHF-möjlighet (alla vanliga hem-TV har detta). En svart/vit TV, 10-12" ger utmärkt resultat (kostar ny under 500:- kronor) och rekommenderas om Du skall använda datorn mycket.

Uppkopplingen är mycket enkel. Det Du behöver för att få datorn att fungera är bara TV:n och nätadaptern samt eventuellt antennomkopplaren, som då ansluts mellan datorn och TV:ns antenningång.



Gör så här:

Anslut koaxialkabeln till datorns uttag märkt TV. Andra änden av kabeln anluts till TV:ns antenningång. Har Du antennomkopplare skall kabeln anslutas till omkopplarens uttag märkt GAME och omkopplarens sladd till antennuttaget på Din TV.

Anslut nätadaptern till ett vägguttag (220 Volt) och jackkontakten till datorns uttag märkt 9 VDC in.

Vrid TV:ns kanalväljare till UHF-läge. (Större TV-apparater har ofta tangenter för ett antal förinställda frekvenser. Om Du har en sådan skall Du förvissa Dig om att den tangent Du valt för datorn är inställd på UHF. Detta görs i regel med en liten omkopplare i anslutning till frekvensjusteringsratten för motsvarande tangent.) Slå till TV:n och vrid ner volymen till noll. Studera också TV:ns instruktionsbok!

Nu skall Du ställa in TV:ns frekvens så att den stämmer med datorns utsignal. Det är kanal 36 UHF Du skall söka efter. Vrid frekvensinställningen till dess att bilden klarnar och en liten fyrkantig markör dyker upp i nedre vänstra hörnet. Markören skall ha ett vitt infällt K i den svarta fyrkanten. Det kan hända att Du måste justera ljus och kontrast för att K:et skall synas ordentligt.

Vi väntar med att ansluta bandspelaren. Din dator är nu klar att tas i bruk...

KAPITEL 2

ZX81 SOM KALKYLATOR

När Du använder datorn måste den få instruktioner som den begriper och kan utföra. En sådan är PRINT. PRINT betyder för datorn att en utskrift skall göras på skärmen. Men PRINT innebär också att matematiska uttryck beräknas och resultatet erhålls på skärmen. Vill Du att datorn skall beräkna summan av 2+2 skriver Du bara:

PRINT 2 + 2

Tryck på P-tangenten! ZX81 är utrustad med en finess som underlättar programmeringsarbetet väsentligt. Det räcker att trycka på en tangent för att få alla instruktioner, kommandon och funktioner att skrivas ut. De är förprogrammerade! Dessutom håller datorn ordning på tangenterna så att när ett kommando (eller instruktion) förväntas tolkas en tangentnedtryckning som det kommando som står över tangenten. När Du tryckte på P skrev datorn därför ut hela ordet PRINT. Lägg märke till markören! I stället för ett K fick den nu ett L infällt. Markören K betyder att datorn väntar en instruktion eller kommando och L att enstaka tecken (bokstav eller siffra) väntas.

Tryck nu på 2. Tvåan skrevs ut och Du ser att datorn också ser till att åstadkomma mellanrum (både praktiskt och snyggt, eller hur?). Skriv + tecknet genom att hålla SHIFT nedtryckt när Du trycker på + (K-tangenten). Med SHIFT nedtryckt kommer Du åt alla rödmarkerade funktioner på tangenterna. Tryck på 2 igen! På skärmen står nu

PRINT 2 + 2

För att verkställa Dina order till datorn används alltid NEW LINE-tangenten. Då skall vi se om datorn kan räkna. Tryck på NEW LINE! Svaret 4 kom genast upp på skärmen i övre vänstra hörnet. Längst ner ser Du sifferkombinationen 0/0. Detta är ett s k felmeddelande till Dig. I detta fall betyder första nollan att operationen utförts utan fel. Lägg mörke till att datorn skriver noll med snedstreck Ø för att skilja den från bokstaven O. Här använder vi 0 (noll) respektive O!

Anm: Skulle Du skriva fel så använd SHIFT och RUBOUT (0-tangenten).

Du har nu tekniken för att få ZX81 att fungera som kalkylator:

1. Skriv PRINT
2. Skriv det Du vill ha uträknat t ex 2 + 2
3. Tryck på NEW LINE

Ändra felaktigheter i Dina uttryck går lätt. Skriv något tokigt t ex

PRINT + + 2

Försök få datorn att räkna ut detta genom att trycka på NEW LINE. Det går inte! Däremot fick Du fram en ny markör S. S-markören talar om att det Du skivit innehåller ett syntaxfel (= ej godkänt uttryck i ZX81 BASIC). S-markören placerar sig framför det ställe i raden där felet finns. För att rätta till detta kan Du naturligtvis sudda ut hela raden med SHIFT och RUBOUT men det finns bättre sätt. Du kan nämnligen flytta L-markören till felet och bara ta bort detta (eller skiva in det som fattas). Tangenterna 5 och 8 är försedda med två röda pilar riktade åt vänster respektive höger. Tryck på 5 samtidigt som SHIFT hålls nere. L-markören kan nu stegas tillbaka åt vänster. 8 (+ SHIFT)fungerar på motsvarande sätt. Prova gärna! Flytta markör L till detta läge:

PRINT + L + 2

och tryck på RUBOUT. Det vänstra + teckent försvann. Skriv in 2. Raden har nu utseendet

PRINT 2 L + 2

L-markörens läge har ingen betydelse när Du nu skall få uttrycket beräknat. Tryck på NEW LINE och det hela fungerar som tidigare.

Studera tangentbordet! Du ser att det är försett med en rad olika instruktioner och förkortningar m m. Upp till 5 olika symboler kan finnas på en och samma tangent. Du har nu lärt Dig hur Du erhåller enstaka tecken, instruktioner och kommandon över tangenterna samt de rödmarkerade tecknen.

Sammanfattning:
Kapitlet har behandlat

- hur datorn kan användas som kalkylator med PRINT
- K, L och S markörerna
- felmeddelande 0/0
- rättning av fel med RUBOUT och hur L-markören kan flyttas

KAPITEL 3

EN LEKTION I HISTORIA

ZX81 arbetar med ett datorspråk som kallas för BASIC (förkortning av engelskan Beginners´ All-purpose Symbolic Instruktion Code). För att bryta ner en instruktion i BASIC t ex PRINT till ett språk som datorns centralenhet använder sig av finns i datorn ett fast program som gör översättningen. Tack vara detta program kan vi programmera vår dator med engelska ord och vanliga siffror och bokstäver. Det in terna språket är mycket otympligt att använda och består av olika kombinationer av 0 och 1. I datateknikens barndom var det nödvändigt att skriva alla instruktioner med sådana koder.

BASIC utvecklades vid Dartmouth College i New Hampshire, USA, år 1964. BASIC har sedan blivit det förhärskande datorspråket för mindra datorer och används över hela världen. En av anledningarna till detta är språkets fördelar vid kommunikation direkt med datorn (uppgifter till datorn utförs och besvaras snabbt och enkelt. Det finns många sådana "högnivåspråk" t ex ALGOL och PASCAL, som båda har kraftfulla programmeringsmöjligheter, men få språk är lika bra som BASIC vid direktkommunikation med datorn (on-line-kommunikation).

Många datortidskrifter och böcker ge s ut med program skrivna i BASIC. Dessvärre är det så att knappast någon dator följer standard utan de har ofta egna dialekter av BASIC-språket. Det är därför sällan som det går att direkt applicera ett program skivet för en dator till en annan. Me n har Du lärt Dig en dialekt är det inte alltför svårt att göra de ändringar som behövs för att få ett sådant program att fungera.

KAPITEL 4

MER BERÄKNINGAR MED ZX81

I kapitel 2 såg Du hur ZX81 kan fungera som en kalkylator. Vi skall nu studera detta lite ytterligare.

Som Du säkert misstänker kan ZX81 inte bara addera utan också subtrahera, multiplicera, dividera m m.

+ , - , * (anges med stjärna , B-tangenten) och / kallar vi för operationer

talen vi kombinerar med operationerna ovan kallar vi operand

ZX81 kan också räkna exponentuttryck (SHIFT H) d v s "upphöjt till". Pröva t ex

PRINT 2 *3 betyder 2^3 eller 2*2*2

och Du får svaret 8. Du kan också kombinera flera operationer i samma uttryck. Prova med

PRINT 20 - 2 *3 **2 + 4/2 * 3

som också ger resultatet 8. Låt oss se hur datorn kommer fram till detta. Den har nämligen en bestämd turordning för att utföra de olika operationerna. Vi kallar detta prioriteringsordning.

20 - 2 *3**2 + 4/2 *3 först beräknas **vilket ger
20 - 2 *9 + 4/2 *3 sedan * och / vilket ger
20 - 18 + 6 sedan addition och subtraktion vilket ger
2 + 6 =8!

Prioriteringen anges med en siffra 1 till 16. Våra operationer ovan har prioriteringen:

**	10
* och /	8
+ och -	6

När - används för att negera något, t ex -1 har - prioritet 9. Prioriteringen kan sättas ur spel med paranteser t ex:

PRINT 2 3+2 ger 8 som resultat medan
PRINT 2 (3+2) ger 10 som resultat eftersom paranteser beräknas först. Träna nu själv med olika operationer och bilda egna matematiska uttryck!

Decimaltal kan anges med decimalkomma eller med hjälp av exponentuttryck. Samma sak gäller för stora tal där exponentuttryck också kan tillämpas. Exempel

0.0234	kan skrivas	2.34E-2 = 2.34 10 -2
2340		2.34E3 = 2.34 10 3
2.34		2.34E0 = 2.34 10 0

2 Positiva exponenter anger hur många steg kommat skall flyttas åt höger och negativa antal steg åt vänster. Nollan framför kommat skrivs inte alltid ut.

Det går bra att låta datorn skriva fler saker samtidigt på skärmen. Du använder då ; eller , för att separera de olika uttrycken. Försök med

PRINT 1; 2; 3; 4; 5
PRINT 1, 2, 3, 4, 5

Semikolon gör att utskriften görs utan mellanrum med kommatecken delar skärmen i två hälfter.

Sammanfattning:
Instruktioner: PRINT, med komma eller semikolon
Operationer: +, -, *, /, **
Exponentuttryck

Övningar:
1. Pröva PRINT 2.34 E0
PRINT 2.34 E1 osv till PRINT 2.34 E15
Gör samma sak med negativa exponenter dvs E-0 osv, vad händer?

2. Pröva PRINT 1,, 2,,,,3 förklara!
PRINT 1;; 2;;;;;3 förklara!

3. PRINT ger bara 8 signifikanta siffror medan talen lagras i minnet med 10 siffrors noggranhet. Beviset för detta ger
PRINT 4294967295, 4294967295 - 429E7

Det största tal (heltal) datorn tillåter med full noggrannhet är 4294967295.

KAPITEL 5

FUNKTIONER

Med en funktion menas en regel för att ge ett resultat bestämt av ett värde applicerat på funktionen, t ex den vanliga kvadratrotsfunktionen

PRINT SQR 9 ger resultat 3 (= roten ur 9)

SQR är funktionen, 9 är vårt värde (operanden). Hur får vi då fram SQR-funktionen? Alla funktioner är utskrivna under tangenterna och erhålls genom att sätta tangentbordet i "funktionsläge". Detta görs med SHIFT och NEW LINE-tangenten som har den röda texten FUNCTION påtryckt. Markören ändras till F och med H-tangenten erhålls nu SQR. Pröva SQR på några andra tal!

Alla funktioner har prioritetsordning 11 (utom NOT som har 4) vilket förklarar skillnaden mellan

PRINT SQR 2 * 2 och PRINT SQR (2 * 2)

För att ha full glädje av de olika funktionerna behöver Du kunna lite matematik. Då först får de mening. Nu kan ju datorn användas till så mycket annan än bara matematikproblem varför Du även utan sådana kunskaper kommer att få stort utbyte av Din ZX81. En av funktionerna skall vi dock se lite närmare på.

RND- funktionen ger ett slumptal mellan 0 (som kan antas) och 1 (som inte kan antas). RAND-instruktionen används tillsammans med RND för att styra slumpmässigheten. RND ger inte riktiga slumptal utan väljer i en lång lista av tal (65536 st) och ger intryck av slumpmässighet om RAND-instruktionen föregår slumptalsgenereingen. Den här lösningen med s k pseduoslumptal är vanlig bland smådatorer. Pröva med

RAND 1 och sedan
PRINT RND

flera gånger. Hela tiden erhålls 0.0022735596 - knappast slumpmässigt! Pröva i stället

RAND 0 (eller bara RAND) och det hela blir betydligt bättre!

Talen plockas nu ur listan från ett läge som bestäms av hur länge datorn varit uppkopplad mot TV:n och blir därför omöjliga att förutsäga.

För matematiker:

SGN	signum , ger +1 , 0 , -1 beroende på om operanden är positiv, noll eller negativ
ABS	absolutbeloppet
SIN	sinus
COS	cosinus
TAN	tangenten
ACS	arccos
ASN	arcsinus
ATN	arctan
LN	naturlig logaritm
EXP	exponentialfunktion e^x
SQR	kvadratrot
INT	avrundar nedåt till närmaste heltal
PI	pi (3.14....)

LOG erhålls med LN x/LN 10 där x är ett godtyckligt tal

Vinklar anges i radianer. Pi (3.14...) radianer = 180 grader och för att omvandla från grader till radianer divideras vinkeln med 180 och multipliceras med pi. Ex

PRINT TAN (45/180)* PI

vilket beräknar tangenten av vinkeln 45^o !

Sammanfattning:
Instruktionen RAND
Funktioner

KAPITEL 6

VARIABLER

Moderna fickkalkylatorer är vanligen försedda med ett litet minne där enstaka variabelvärden kan lagras för senare bruk. Din dator klarar att lagra hundratals sådana variabelvärden med hjälp av LET-instruktionen och dessutom ge varje värde sitt speciella namn. Med hjälp av detta namn kan den sedan plocka fram värdet närhelst det behövs.

Skriv LET OST=20 läs in med NEW LINE!

Datorn har nu lagrat värdet 20 i sitt minne och givit denna plats namnet OST. Låt oss anta att kilopriset på ost är 20 kronor. Vill vi få datorn att skriva ut kilopriset använder vi PRINT-instruktionen:

PRINT OST och 20 skrivs omgående ut på skärmen.

Pröva PRINT OST/2 priset på ett halvt kg ost!
och PRINT OST * 7 priset på 7 kg ost!

Skulle priset på ost stiga till 23:- är det bara att byta ut det gamla värdet med

LET OST=23

Skriv nu PRINT SOCKER

Resultatet blir felkoden 2/0. Felkoden 2 betyder att variabeln med namn SOCKER inte fanns i minnet. Eftersom vi inte infört någon sådan variabel är datorns reaktion naturlig. Med satsen

LET SOCKER=5 är ordningen återställd!

Namn på talvariabler som dessa får dock inte utformas hur som helst även om Du har stor frihet. Variabelnamn måste alltid börja med en bokstav sedan är det fritt fram med flera bokstäver och/eller siffror. Exempel på tillåtna namn:

ETT KG SOCKER
Z80
X
K2R990

Ej tillåtna namn är:

3 PKT	börjar med siffra
A?	? är varken bokstav eller siffra
DATA-1	- -"-

Inversa tecken (vit text mot svart botten) är inte heller tillåtna i namn.

Skriv nu CLEAR glöm inte NEW LINE!

och sedan PRINT OST

Resultatet blev felkoden 2/0 igen (=variabeln kan ej hittas i minnet). Effekten av CLEAR är att det utrymme i minnet som reserverats för variabler töms på sitt innehåll. Samma resultat erhålls om 9-Volts-kontakten tas ut med en väsentlig skillnad - allt lagrat i minnet försvinner! CLEAR rensar bara bort lagrade variabelvärden.

Sammanfattning:
Instruktioner: LET CLEAR
Variabler

Övningar:
1. Ange reglerna för namn på talvariabler!
2. Varför måste alla variabler ges namn?

KAPITEL 7

STRÄNGAR

Nu har inte fickkalkylatorn längre någon möjlighet att mäta sig med datorn. ZX81 kan, som alla andra BASIC-datorer, nämligen arbeta med text. Skriv

PRINT "HEJ, JAG HETER ZX81" " fås med SHIFT P

Texten mellan "-tecknen kallas för en sträng. En sträng kan bestå av vilka tecken som helst (utom ", vill Du använda detta tecken i en sträng används "" som erhålls med SHIFT Q), kravet är dock att Du börjar och avslutar med ". Skulle Du glömma detta vägrar datorn att ta emot strängen och markerar felet med den inbyggda syntaxkontrollen - ett inverst S visar att Du gjort fel. Strängar ger oss möjlighet att med text förklara och förtydliga utskrift. T ex

```
LET OST=20
PRINT "OST KOSTAR :"; OST;  " KR/KG"
```

Här är några saker Du kan använda och göra med strängar:

1. Ge dem namn och lagra dem i minnet med LET på samma sätt som talvariabler. Namn på sådana strängvariabler består alltid av endast en bokstav åtföljt av dollartecken t ex A$, Y$ etc

2. Addera strängar. Ex PRINT "OST" + "HYVEL" ger OSTHYVEL

3. Bearbeta strängarna med vissa funktioner:
   LEN ger längden (antal tecken, inklusive mellanslag, i strängen)
   ex: LEN "OST" ger resultatet 3.

   VAL beräknar det aritmetiska värdet av en sträng, ex VAL "2 + 7" ger resultat 9. Två regler gäller om VAL skall användas i längre uttryck:
   - VAL måste komma först i uttryck som LET X= 7+VAL"Y". Detta måste alltså ändras till LET X= VAL "Y" + 7
   - VAL får bara användas i första koordinaten i satserna PRINT AT, PLOT och UNPLOT. Ex PLOT 5, VAL"Y" måste ändras till två satser: LET Y=VAL "Y"
     PLOT 5, Y (vi återkommer längre fram)

   STR$ ger möjlighet att få en talvariabel att bearbetas som en sträng. T ex PRINT STR$ 3.5 ger samma resultat som PRINT "3.5"

4. Du kan kombinera strängfunktionerna ovan till stränguttryck. Ex
   VAL (STR$ LEN "123456" + "-4") som bearbetas på följande sätt:
   VAL (STR$ 6 + "-4")
   VAL ("6" +"-4 ")
   VAL (" 6 - 4 ")
   2

Sammanfattning:
   Funktioner: LEN VAL STR$
   Operation + för strängar
   Strängvariabler, textuttryck

Övningar:

1. Skriv LET A$ = "2 + 2" och sedan
   PRINT A$; "=" , VAL A$

   Pröva att ändra A$ till olika matematiska uttryck t ex
   LET A$ = "ATN 1 * 4" (svaret bör bli pi!)

2. En sträng "" dvs som helt saknar tecken kallas för nollsträng eller tom sträng. Den har längden 0. Blanda inte ihop den tomma strängen med ""-tecknet på Q-tangenten. Vad används den till?

3. Vill Du göra datorn generad så skriv

   PRINT " 2 + 2 = "; 2 + 1

Anm: Datorn saknar de svenska bokstäverna ÅÄÖ. De internationella beteckningarna för dessa bokstäver är:

Å - AA
Ä - AE
Ö - OE

KAPITEL 8

PROGRAMMERING

Nu skall vi äntligen börja skriva datorprogram för ZX81. Töm datorns minne och skriv

10 LET A= 75

Detta kallar vi för en rad eller sats! Tryck på NEW LINE och raden flyttas från skrivarean upp till skärmens första rad. En programrad skall alltid inledas med ett radnummer som anger den ordning datorn skall bearbeta de olika instruktionerna. (Det är praktiskt att välja radnummer som är jämna multiplar av 10. Det finns då alltid möjlighet att komplettera ett program med nya rader mellan de tidigare). Skriv

20 PRINT A

Du har nu ett litet program lagrat i datorns minne. När Du skall köra programmet (exekvera med ett finare ord) används kommandot

RUN åtföljt av NEW LINE. Gör det! På skärmen skrivs ut

75 och felkoden 0/20 talar om att programmet genomförts

utan fel och att sista raden var 20. Tryck på NEW LINE och programmet listas igen. Vi för in ytterligare en variabel. Skriv

15 LET B = 100

NEW LINE får raden att hamna på rätt plats i listan mellan rad 10 och 15. För att få utskrift av B vill vi ändra rad 20. Detta kan göras på följande sätt:
- genom att skriva en helt ny rad 20 PRINT A , B som då byter ut den gamla raden eller
- genom att editera rad 20! I programlistan ser Du en liten markör ▷. Den kan flyttas upp (med SHIFT 7) eller ner (med SHIFT 6). Flytta den till rad 20! Tryck sedan på EDIT (SHIFT 1) och hela rad 20 kopieras i skrivarean. Markören i skrivarean kan också flyttas. SHIFT 8 stegar markören åt höger och SHIFT 5 åt vänster. Flytta den till platsen omedelbart efter A och skriv , B ! Raden har nu önskat utseende och med NEW LINE för vi upp den i listan igen. Enklare kan det inte bli!

Kör programmet! Både A och B skrivs nu ut på skärmen. Tryck på NEW LINE för att få tillbaka listan. Skriv till raden

12 PRINT C

Att ta bort en rad är mycket lätt. Du skriver bara det aktuella radnumret och trycker på NEW LINE. Ta bort rad 12!

Nu försvann markören i programlistan. Naturligt eftersom den sist inlästa raden var 12 som ju saknade innehåll! Skriv

LIST 15 i fortsättningen får Du själv komma ihåg NEW LINE

Programmet listas nu från rad 15 och markören har återvänt. Rad 10 finns kvar men syns inte. Skriv

LIST 10 och hela programmet listas igen.

Sammanfattning:
Program
EDIT-funktioner
Instruktioner: RUN LIST

Övningar:
1. Ändra programmet så att också namnen A respektive B skrivs ut på skärmen.

2. Använd EDIT för att ändra värdet på A-variabeln

3. Kör programmet och skriv sedan PRINT A. Du ser att värdet är kvar även efter programkörningen.

4. Lägg märke till avd som händer när Du skriver
12 och NEW LINE
EDIT
Pröva också
30 och N/L
EDIT

5. Vad händer om LIST ingår i programmet. Pröva med 30 LIST!

KAPITEL 9

MER PROGRAMMERING

Skriv NEW

NEW rensar bort alla gamla variabler och programrader. Jämför med CLEAR som bara tar bort variabelvärden. Skriv sedan in följande lilla program:

```
10 REM DETTA PROGRAM BERAEKNAR KVADRATROTEN
20 INPUT A
30 PRINT A, SQR A
40 GOTO 20
```

(På rad 10 får Du själv läsa in mellanrum i texten med SPACE)
Kör nu programmet med RUN. Längst ner på skärmen ser Du markören L. Detta är tecken på att datorn väntar sig inläsning av en talvariabel. Datorn har avverkat programmet till rad 20 där INPUT säger att körningen skall avbrytas för indata. Skriv 4! Datorn tar emot talet och programmet fortsätter, roten beräknas och talet 4 samt roten 2 skrivs ut på skärmen. Sedan får datorn order att hoppa till rad 20 och väntar på nästa tal. Du kan nu fylla hela skärmen med olika tal och roten ur dessa. När skärmen är full avbryts körningen med felkod 5/30 som just betyder skärmen full! Med NEW LINE kommer programmet åter upp på skärmen. Kör programmet igen med RUN.

Om Du vill avbryta programkörningen innan skärmen blir full använder Du STOP (SHIFT A). Felkoden D/20 säger att en inputinstruktion avbrutits med STOP. Vill Du återvända till programkörningen efter avbrott används CONT. CONT kan Du också använda när körningen avbrutits på grund av "skärmen full".

Se på PRINT-satsen rad 30! Varje gång den utförs börjar utskriften på ny rad. Med semikolon (;) och komma (,) kan vi manipulera utskriften. Ändra programmet till (bara rad 30)

```
30 PRINT A,
```

Du ser att A-värdena som läses in skrivs ut i två kolumner. Skriv till raden

```
35 PRINT
```

Och Du får en radframmatning mellan varje utskrift.

Enbart PRINT används för radframmatning eller för att få blankrad. Skriv till följande rader:

```
100 REM STRAENGLAENGD                (AE står för Ä )
110 INPUT A$
120 PRINT A$ , LEN A$
130 GOTO 110
```

REM betyder inget för datorn. Det är Dina egna anteckningar om programmets innehåll. Du ser att det går bra med två separata program samtidigt i minnet. För att bara köra det sista skriver Du RUN 100. Gör det!

Nu erhölls " L " längst ner. Detta är tecken på att datorn väntar sig inläsning av text. Skriv Ditt namn. Vi kallar Dig Per!

"PER"

Datorn skriver ut namnet och anger samtidigt hur många bokstäver det innehöll. Vill Du stoppa programmet med STOP måste först det vänstra tecknet tas bort (eller markören flyttas till vänster om " " tecknen). Pröva! Du erhåller felkod D/110. Med NEW LINE får Du listan tillbaka.

För att starta programmet använde vi RUN 100. Det går också att starta med GOTO 100. Skillnaden är att RUN tar bort lagrade variabelvärden med GOTO inte påverkar dessa. GOTO kräver alltid radnummer medan enbart RUN startar programmet från första raden.

Ibland kan man av misstag hamna i en programslinga som aldrig avbryts. T ex raden

```
200 GOTO 200                         hoppar till sig själv!
```

Starta med GOTO 200 och Du är ohjälpligt fast. För att hejda programmet kan Du ta ur kontakten (= programmet förstörs) eller bättre använda BREAK (SPACE-tangenten). Programmet stannar då med felkod D/200. BREAK fungerar också vid laddning/lagring med bandspelare.

Sammanfattning:
Instruktioner: NEW PRINT GOTO CONT INPUT REM STOP BREAK

Övningar:
1. Ändra rad 40 i kvadratrotsprogrammet till 40 GOTO 15. Du finner att det inte har någon betydelse om raden inte existrar (rad 15). Samma sak gäller för RUN. (Enbart RUN betyder egentligen RUN 0)

2. Skriv ett program som läser in priset på en vara exkl moms och sedan skriver ut pris inkl moms. Ledning: multiplicera med 1.2346 .

3. Vilken effekt har CONT, CLEAR och NEW inne i ett program?

KAPITEL 10

DATORN FATTAR BESLUT . . .

Tidigare program har varit obevekliga! De har körts från första rad till sista och eventuellt börjat om på nytt. Nu skall vi ge våra program lite mera "liv". Skriv

```
10 INPUT A
20 IF A=10 THEN GOTO 50
30 PRINT A
40 GOTO 10
50 PRINT   "A=10"
60 PRINT "VILL DU FORTSAETTA"
70 INPUT A$
80 IF A$="JA" THEN GOTO 100
90 STOP
100 GOTO 10
```

Se först rad 90. Det går bra att använda STOP i ett program för att avbryta körningen (felkod 9/90 erhålls).

Vi har två IF-satser i vårt program. Den första testar om A är 10. Är så fallet hoppar programkörningen till rad 50. Om inte A är 10 fortsätter körningen som om inget hänt på nästa rad (30). Den andra IF-satsen testar en strängvariabel A$. Är denna JA hoppar datorn till rad 100 och sedan fortsätter programmet igen från början. Är A$ något annat avbryts körningen på rad 90.

IF . . . THEN testar ett villkor och om detta är uppfyllt utförs instruktionen efter THEN. Man kan säga att datorn "fattar beslut"! IF . . . THEN kan kombineras med nästan vilka villkor och instruktioner som helst. Vanliga operationer i villkor är

A = B	lika	A > B	större än
A < B	mindre än	A >= B	Större eller lika
A <= B	Mindre el lika	A <> B	ej lika

Ofta vill man testa flera villkor samtidigt. Detta görs med hjälp av de logiska operationerna AND, OR och NOT. Exempel

IF A=B OR B=C THEN . . . IF A> B AND NOT B=C THEN . . .

Stängar kan också testas på detta sätt. Vid jämförelse av strängar gäller:

att den alfabetiska ordningen testas med > och < . Ex
ADA<BEDA CECAR>BERTIL
att stränglängden jämförs med >= och <= . Ex
ADA<=BEDA

NOT har prioritet 4, AND har 3 och OR har 2. Som avrundning ett litet programexempel som läser in en följd av tal och skriver ut det hittills största av dessa.

```
10 PRINT "TAL", "STOERSTA TAL"
20 INPUT A
30 LET B=A
40 PRINT A, B
50 INPUT A
60 IF B< A THEN LET B=A
70 GOTO 40
```

Sammanfattning:
Instruktioner: IF...THEN STOP
Operationer AND OR NOT = = =

Övningar:

1. = är motsatsen till <> . Vad innebär NOT A<>B?

2. Vad är motsatsen till A>=B?

3. Hur utformas i BASIC "om A inte är lika med B eller C". Svaret är inte IF A<>B OR C inte heller IF A<>B OR A<>C !!!

4. Pröva programmet:
```
10 PRINT "X"
20 STOP
30 PRINT "Y"
```
Kör programmet. På vilken rad startar CONT körningen efter avbrottet? Rad 30 gissar Du säkert. Vad händer då om vi lägger till raden 21 STOP?

KAPITEL 11

TECKEN OCH KODER

Bokstäver, siffror, ja samtliga tecken inklusive tangentord som PRINT kallar vi för tecken och dessa uppfattas också av datorn som enstaka tecken med bestämda koder. Det finns 256 tecken totalt med kod mellan 0 till 255. För att omvandla från tecken till kod och tvärt om används de två funktionerna CODE och CHR$.

CODE X$ ger koden till första tecknet i strängen X$
CHR$ X ger det tecken vars kod är X

Detta program skriver ut hela teckenuppsättningen:

```
10 LET A=0
20 PRINT CHR$ A;
30 LET A=A + 1
40 IF A< 256 THEN GOTO 20
```

Högst upp på skärmen ser Du tecknen " £ $ osv till Z som alla återfinns på tangentbordet och kan skrivas ut när markören har utseendet L. Längre ner ser Du samma tecken igen men nu i invers form. Också dessa kan skrivas direkt på tangentbordet om detta försätts i "grafikläge" med SHIFT 9. Prova! När Du vill ut ur grafikläget används GRAPHICS eller NEWLINE. RUBOUT fungerar som vanligt. Markören indikerar grafikläge med ett G i invers form. Även de grafiska symbolerna som finns på vissa av tangenterna, t ex 1 till 8, kan tas fram i grafikläget med hjälp av SHIFT. Vissa funktioner saknar tecken t ex NEW LINE och markeras då med ?. De olika grafiksymbolerna ser Du på nästa sida.

Sammanfattning:
Funktioner: CODE CHR$

Övningar: Tänk dig det utrymme som upptas av ett tecken indelat i fyra
1. Tänk Dig det utrymme som upptas av ett tecken indelat i fyra fält. Dessa kan vara antingen vita eller svarta vilket ger 2 x 2 x 2 x 2 =16 kombinationer. Ta reda på alla dessa i teckenuppsättningen!

2. Tänk Dig varje tecken indelade i två horisontella fält som kan vara antingen vita, grå eller svarta vilket ger 3 x 3 =9 kombinationer. Finn dessa!

3. Prova detta program.

```
10 INPUT A
20 PRINT CHR$ A
30 GOTO 10
```

Läs in olika värden på A. Pröva också tal som 1.25 etc. Vad händer?

4. Detta program ritar skärmen full med slumpmässigt valda tecken:

```
10 LET A=INT (16 * RND)
20 IF A>=8 THEN LET A=A+120
30 PRINT CHR$ A;
40 GOTO 10
```

Hur fungerar detta program?

Symbol	Code	How obtained	Symbol	Code	How obtained
	0	K or L SPACE		128	G SPACE
	1	G shifted 1		129	G shifted Q
	2	G shifted 2		130	G shifted W
	3	G shifted 7		131	G shifted 6
	4	G shifted 4		132	G shifted R
	5	G shifted 5		133	G shifted 8
	6	G shifted T		134	G shifted Y
	7	G shifted E		135	G shifted 3
	8	G shifted A		136	G shifted H
	9	G shifted D		137	G shifted G
	10	G shifted S		138	G shifted F

ZX 81 grafiksymboler

KAPITEL 12

LOOP ELLER SNURRA

Anta att Du vill mata in fem tal och addera dessa. Detta kan göras med följande program:

```
10 LET S=0
20 LET N=1
30 INPUT A
40 LET S=S+A
50 LET N=N+1
60 IF N> 5 THEN GOTO 80
70 GOTO 30
80 PRINT "SUMMAN:"; S
```

I datorprogram är det mycket vanligt att vissa sekvenser (delar) skall genomlöpas flera gånger. För att kunna styra antalet varv som en sådan sekvens skall genomlöpas måste vi ha en kontrollvariabel som räknar dessa (här är det N som ökas med 1 för varje varv) och avbryter "rundgången" när kontrollvariabeln uppnått ett bestämt värde (görs ovan på rad 60). Nu finns en speciell instruktion för att åstadkomma sådana här "snurror" (eng. loop). Vi gör om programmet ovan:

```
10 LET S=0
20 FOR N=1 TO 5                  -kontrollvariabeln bestäms
30 INPUT A
40 LET S=S+A
50 NEXT N                        - räknar fram kontrollvariabeln
60 PRINT "SUMMAN:"; S              och ger hopp till rad 20 + 1
```

Snurran ovan kallas för FOR...NEXT-snurra. En intressant variant av snurran är kombination med STEP-funktionen. Se här:

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

Med STEP kan Du styra i vilka "steg" snurran skall genomlöpas. Pröva t ex STEP 0.5 , STEP 3/2 etc!

Du kan också ha flera snurror i varandra om följande regler iaktas:

Detta går bra	men inte detta
FOR N=...	FOR N=...
FOR I=...	FOR I=...
FOR J=...	FOR J=...
...	...
NEXT J	NEXT N
NEXT I	NEXT I
NEXT N	NEXT J

För att demonstrera detta kan Du skriva in detta program som åstadkommer en komplett uppsättning 6-punkters dominobrickor:

```
10 FOR M= 0 TO 6
20 FOR N=0 TO M
30 PRINT M;" : "; N;" ";
40 NEXT N
50 PRINT
60 NEXT M
```

En annan sak Du bör se upp med är att hoppa in i en snurra från annan plats i programmet eftersom snurrvariabeln sätts upp av FOR-satsen och därför inte är definierad (bestämd) när datorn träffar på NEXT-satsen. Resultatet blir avbrott med felmeddelande 1 eller 2!

Sammanfattning:

Instruktioner: FOR ...TO...NEXT, STEP

Övningar:

1. Skriv om programmet i kap 11 (som skrev ut tecknen) med hjälp av FOR...NEXT.

2. Skriv ett program som skriver ut talen 1 till 5 med FOR...NEXT.

3. När Du kört programmet i uppgift 2, skriv PRINT N (om N var namnet på din kontrollvariabel. Förklara resultetet!

4. Vad gör detta program?

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

5. Skriv ett program som läser in ett godtyckligt antal tal och adderar dessa. Ledning: Låt en INPUT-sats bestämma kontrollvariabelns slutvärde (INPUT X och sedan FOR I=1 TO X).

KAPITEL 13

TVÅ HASTIGHETER

ZX81 kan arbeta i två hastigheter, SLOW (långsamt) är det normala då bild visas samtidigt som datorn utför arbete och FAST (snabbt) som ger ca 4 ggr snabbare körning men då bild bara visas när programmet avbryts med INPUT eller annat uppehåll. I FAST-läge blinkar skärmen till varje gång en tangent trycks ner (bekant fenomen för ZX80 användare. Normalläge i modifierad ZX80 med 8 k ROM är FAST).
Skriv FAST! Läs sedan in programmet

```
10 FOR N=0 TO 255
20 PRINT CHR$ N;
30 NEXT N
```

När programmet körs i FAST-läge försvinner bilden under körningen och först när programmet är slut visas resultetet upp.
Pröva nu med samma program i SLOW läge. Du skriver SLOW och trycker som vanligt på NEW LINE.

FAST-läge är effektivt när
- ett stort antal beräkningar skall göras t ex i en lång snurra
- när långa program skall skrivas in

Det går bra att använda SLOW och FAST inne i ett program:

```
10 SLOW
20 FOR N=1 TO 64
30 PRINT "A";
40 IF N= 32  THEN FAST
50 NEXT N
60 GOTO 10
```

Sammanfattning:
Instruktioner: SLOW, FAST

KAPITEL 14

SUBRUTINER

Ibland återkommer samma typ av arbete på flera ställen i ett program. För att slippa skriva om sådana rutiner varje gång de skall användas används subrutiner. I stället skrivs bara

GOSUB åtföljt av det radnummer där rutinen (subrutinen) återfinns

Subrutinen avslutas alltid med RETURN-instruktionen för att åstadkomma hopp tillbaka till huvudprogrammet när rutinen utförts. Ex

```
10 PRINT "HUVUDPROGRAM"
20 GOSUB 1000
30 PRINT "FORTS PÅ HUVUDPROGRAMMET"
40 GOSUB 1000
50 PRINT "HUVUDPROGRAMMETS SISTA DEL"
60 STOP
1000 PRINT "DETTA AR SUBRUTINEN"
1010 RETURN
```

STOP-instruktionen rad 60 är viktig. Utan den skulle programmet fortsätta in i subrutinen.

I samband med bearbetning av kronor och ören kan man önska avrundning till 2 decimaler i samband med beräkningar eller utskrifter. För att varje gång när sådan avrundning behövs slippa föra in denna kan vi använda subrutinen

```
1000 LET X= INT (X * 100)/100
1010 RETURN
```

Sammanfattning:
Instruktioner: GOSUB , RETURN

Övningar:
1. Gör ett program som beräknar momspålägget på talen 1 till 20.

2. Använd subrutinen ovan för att få utskrift av priset inkl moms med angivande av två decimaler.

KAPITEL 15

LAGRING AV PROGRAM OCH DATA PÅ BAND

Med hjälp av en vanlig kassettbandspelare kan Du enkelt lagra program och tillhörande variabelvärden och sedan , när de skall användas nästa gång enkelt föra över dem till datorns minne igen. Vi skall se hur detta går till. Först skriver vi ett litet program:

```
10 REM  NAMN
20 PRINT "KALLE"
30 PRINT "KARIN"
40 PRINT "LOTTA"
```

Med kommandot SAVE skall vi nu lagra programmet på band. Koppla dubbelkontakten så att EAR-uttaget på datorn svarar mot EAR (extra hörtelefon) på bandspelaren. Samma sak med MIC-uttaget men till bandspelarens mikrofoningång. Gör sedan:

1. Spola bandet till ett ställe där Du vill ha programmet lagrat.
2. Använd mikrofonen för att tala in namnet på Ditt program (praktiskt när Du sedan vill söka rätt på det)
3. Skriv SAVE "NAMN" Vänta med NEW LINE!
4. Starta bandspelaren för inspelning
5. Tryck på NEW LINE
6. Studera skärmen. Först blir den grå, sedan kommer kraftiga störningar att synas för några sekunder. När skärmen blir blank och felkoden 0/0 visar sig är programmet lagrat.
7. Stoppa bandspelaren

Spola tillbaka bandet och lyssna på det nu inspelade programmet. Dra ner volymen!!! Intressant ljud, eller hur? Först ett knarrande sedan kommer själva programmet, ett högfrekvent oregelbundet ljud, därefter återkommer det knarrande lätet igen.

Vi skall nu föra över programmet till datorns minne. Skriv NEW för att få bort allt lagrat i RAM. (Programmet finns kvar efter SAVE i datorns minne).

1. Spola tillbaka bandet så att det hamnar framför det inspelade programmet.
2. Vrid upp volymen till max eller nästan max

3. Skriv LOAD "NAMN" Tryck inte på NEW LINE
4. Starta bandspelaren för uppspelning
5. Tryck på NEW LINE
6. Studera skärmen. Också nu uppträder kraftiga störningar på skärmen och efter några sekunder blir skärmen blank med felkod 0/0. Programmet är nu lagrat i datorns minne (om det tar onormalt lång tid kan fel uppstått. Pröva att variera ljudvolymen).
7. Stoppa bandspelaren.

Skriv LIST och Du ser programmet listas på skärmen igen.

Normalt skall det inte vara några problem med LOAD och SAVE. Skulle det inte fungera är någon av följande orsak mest trolig:
- kablarna felkopplade eller dåligt isatta
- volymen är för hög eller för låg
- bandspelaren/bandet ger kraftiga störljud (pga ålder kanske?).

Vissa bandspelare har också så grunda honkontakter för EAR och MIC att det kan vara nödvändigt att dra ut hankontakterna ett par m m.

Genom att ge våra program namn på detta sätt med SAVE kan datorn själv söka sig fram till rätt program på bandet. Namnet fungerar som igenkänningstecken varför flera program samtidigt kan vara inspelade, vart och ett med sitt eget namn - datorn hittar själv det rätta vid LOAD-operationen.

Vill Du att programmet skall starta sig själv efter LOAD gör Du så här:

Komplettera programmet med:

```
50  STOP
100 SAVE "NAMN"
110 GOTO 20
```

Lagra programmet med GOTO 100 i stället för med SAVE. I övrigt exakt som tidigare. När sedan programmet skall föras över till datorn kommer det att starta automatiskt. (Du märker att sista tecknet i NAMN har fått invers form. Detta har ingen betydelse för funktionen)

Eftersom även variabelvärden lagras på bandet är det viktigt att tänka på att om dessa skall användas att Du inte startar med RUN (som ju tar bort dessa). Använd istället GOTO plus önskat radnummer.

Sammanfattning:
LOAD, SAVE

KAPITEL 16

UTSKRIFTSTEKNIK

PRINT-instruktionen kan användas på en rad olika sätt. Du minns att med PRINT kan vi ge order att utföra en beräkning och få resultatet utskrivet eller bara för att skriva ut t ex text på skärmen. Enbart PRINT ger radframmatning och med , respektive ; kan vi styra hur utskriften skall göras på raden. Vi skall nu se på ytterigare några effektiva sätt att påverka utskriften. Pröva detta:

```
10 PRINT AT 0,0; "A"                    A              B
20 PRINT AT 0,31; "B"
30 PRINT AT 11,16; "C"      resultatet          C
40 PRINT AT 21,0; "D"       blir:
50 PRINT AT 21,31; "E"                  D              E
```

Med AT-funktionen bestäms utskriften av ett koordinatsystem där ena koordinaten anger radnummer (0 till 21) och den andra position på raden (0 till 31). Punkten 0,0 är alltså första radens första position.

Ta bort det gamla programmet med NEW. Skriv sedan:

```
10 PRINT TAB 5; "A"; TAB 10; "B"; TAB 15; "C"; TAB 20; "D"
```

Med TAB kan utskriften på en rad styras ungefär som motsvarande funktion (tabulering) på en skrivmaskin. Värdet efter TAB är normalt mellan 0 och 31. Större värden minskas med 32. Om resultatet då blir vänsterflyttning placeras utskriften på nästa rad. Både AT och erna (22 och 23) kan inte användas för utskrift med PRINT. De är reserverade för kommandon, inläsning av programrader etc.

Det finns två instruktioner till som är mycket användbara vid utskrift på skärmen. Det är CLS och SCROLL.

CLS tar bort allt skrivet på skärmen och SCROLL flyttar PRINT-positionen till nedersta raden så att textraderna hela tiden matas in på rad 21 och tidigare text flyttas upp ett steg. Pröva programmen:

```
20 INPUT A$                 10 SCROLL
30 CLS                      20 INPUT A$
40 PRINT A$                 40 PRINT A$
50 GOTO 20                  50 GOTO 10
```


(Vi har valt radnummer så att Du enkelt skall kunna ändra mellan programmen),

Om Du läser in tillräckligt många rader i SCROLL-exemplet så finner Du att översta raden försvinner när alla rader utnyttjats. Vi kan på detta sätt få texten att "rullas" upp på skärmen i all oändlighet.

Sammanfattning:
Instruktioner: PRINT AT, PRINT TAB, CLS, SCROLL

Övningar:

1. Prova programmet:

```
10 FOR I=1 TO 20
20 PRINT TAB 8 * I; I;
30 NEXT I
```

Av exemplet framgår två saker: Du kan låta TAB-värdet vara bestämt av ett uttryck (här 8 * I). Med TAB-värden större än 31 görs utskriften på nästa rad o s v.

KAPITEL 17

GRAFIK

Här skall vi studera en av de mer eleganta egenskaperna hos vår dator. Med PRINT är ju skärmen indelad i 22 rader om vardera 32 tecken. Med instruktionerna PLOT och UNPLOT arbetar vi med en indelning på hela 44 x 64 punkter. Skriv in programmet:

```
10 INPUT X                    läser in X-koordinaten
20 INPUT Y                    läser in Y-koordinaten
30 PLOT X,Y
40 GOTO 10
```

Läs in i tur och ordning koordinaterna (0,0), (63,0), (0.43),(63,43) så förstår Du hur koordinatsystemet fungerar. 0,0 finns i nedre vänstra hörnet och X-koordinaterna räknas från vänster till höger (max 63). Y-koordinaterna räknas nerifrån och upp till max 43.

Medan PLOT ger en liten svart fyrkant i en bestämd koordinat tar UNPLOT bort denna. Här ser Du hur detta fungerar:

```
10 FOR I= 1 TO 63
20 PLOT I,10
30 UNPLOT I-1,10
40 NEXT I
```

Här är ett annat bra exempel på användningen av PLOT. Programmet ritar en kurva över funktionen SIN.

```
10 FOR N=0 TO 63
20 PLOT N, 22+20 * SIN(n/32 * PI)
30 NEXT N
```

Detta program ritar SQR-funktionen:

```
10 FOR N=0 TO 63
20 PLOT N, 20 * SQR (N/16)
30 NEXT N
```


Sammanfattning:
Instruktioner: PLOT, UNPLOT

Övningar:

1. Det finns tre skillnader mellan koordinaterna i PRINT AT och PLOT. Vilka är dessa?

2. Ändra SIN-programmet så att det först ritar en X-axel med "-" och en Y-axel med "I".

3. Skriv ett program som ritar andra funktioner t ec COS.

4. Kör detta:

```
10 PLOT 21,21
20 PRINT "UTSKRIFT"
```

Vad händer?

5. Om Du har en ZX80-dator med 8 k ROM måste Du ändra programmet som demonstrerade UNPLOT för att se hur punkten rör sid över skärmen. Pröva med att lägga till raden 35 PAUS 30.

KAPITEL 18

TID OCH RÖRELSE

Tämligen ofta vill man ha med en tidsfördröjning i sina program. Detta görs enklast med PAUS-instruktionen:

PAUS n där n är ett tal mellan 0 och 32767.

Med n=50 görs ett uppehåll på en sekund och n=32767 ger en paus på nära 11 minuter. Större n-värden ger paus för "evigt". Vid slutet av pausen kommer skärmen att blinka till något. Här är ett program som ritar en urtavla och markerar en sekundvisare med en svart punkt:

```
10 FOR N=1TO 12
20 PRINT AT 10-10 * COS (N/6 * PI), 10+10 *SIN (N/6 * PI); N
30 NEXT N
40 FOR T=0 TO 10000
50 LET A=T/30 * PI
60 LET X=21+18 * SIN A
70 LET Y=22+18 * COS A
80 PLOT X,Y
90 PAUSE 42
100 POKE 16437,255
110 UNPLOT X, Y
120 NEXT T
```

Rad 40 gör att klockan kan gå i 2 3/4 timme (kan lätt ändras genom att ändra T-värdet.) och rad 90 ser till att klockan "tickar" en gång per sekund. Att n inte är 50 beror på att programmet i sig tar viss tid.

Om Du vill undvika att skärmen blinkar till vid pausens slut kan Du göra en fördröjning med en FOR - NEXT - snurra.(Se kap 12.)

Ändrà rad 90 till 90 FOR J=1 TO 25
och rad 100 till 100 NEXT J

En intressant funktion är INKEY$ som läser av tangentbordet utan att NEW LINE behövs. Tangentbordet tolkas som de tecken som erhålls i L-läge. Prova

```
10 IF INKEY$  "" THEN GOTO 10
20 IF INKEY$ =  "" THEN GOTO 20
30 PRINT INKEY$
40 GOTO 10
```

ZX81 fungerar nu som en skrivmaskin. Om DU har en modifierad ZX80 använder Du istället programmet:

```
10 PAUS 40000
20 POKE 16437,255
30 PRINT INKEY$
40 GOTO 10
```

Sammanfattning:
Instruktioner: PAUS
Funktioner: INKEY$

Övningar:

1. Vad händer om rad 10 tas bort i skrivmaskinsprogrammet?

2. I skrivmaskinsprogrammet fungerar inte SPACE. Ändra programmet så att också mellanrum kan erhållas. Ledning: Testa med IF A$ = CHR$ 115 THEN GOTO och lämpligt radnummer där mellanrum skrivs ut. 115 är koden för ⇨ (SHIFT 8) som då används för mellanrum.

3. Här är ett program som verkligen sätter tålamodet på prov. Du får ett nummer som skall skrivas in. I början har Du en sekund på Dig. Gör Du rätt minskas tiden och fel ökas den.

```
10 LET  T=50
20 SCROLL
30 LET A$= CHR$  INT (RND * 10 + CODE "0")
40 PRINT A$ ,
50 PAUS T
55 POKE 16437,255
60 LET B$ = INKEY$
70 IF B$ = "Q" THEN GOTO 200
80 IF A$ = B$ THEN GOTO 150
90 PRINT "INTE BRA"
100 LET T=T * 1.1
110 GOTO 20
150 PRINT "OK"
160 LET T=T * 0.9
170 GOTO 20
200 SCROLL
210 PRINT "SKICKLIGHETSGRAD:"; INT(500/T)
```

Använd Q för att få reda på hur duktig Du är!

KAPITEL 19

SUBSTRÄNGAR

Med substräng menas en del av en sträng. Substrängen måste bestå av ett eller flera tecken i följd. BOK är en substräng till BOKOMSLAG men B SLAG är det inte! Med funktionen n TO m kan vi ta ut en godtycklig substräng. Pröva:

```
PRINT "ABCDEF"(2 TO 5)          ger BCDE
PRINT "ABCDEF"( TO 5)           ger ABCDE
PRINT "ABCDEF"( 2 TO )          ger BCDEF
PRINT "ABCDEF"( TO )            ger ABCDEF
```

Om bara ett tecken önskas

```
PRINT "ABCDEF" (2 TO 2)         ger B
PRINT "ABCDEF"(2)               ger B  (enklare variant)
```

Med n TO m kan Du effektivt bearbeta text. Här är ett program som byter ut en del av en sträng:

```
10 LET A$ =" ERIK HAR EN BOK"
20 PRINT A$
30 LET A$ (1 TO 4)="KARL"
40 PRINT A$
```

Funktionen n TO m har prioritet 12

Sammanfattning:
Funktionen: n TO m

Övningar:

1. Vad blir resultatet av
```
PRINT "ABC"+"DEF"(1 TO 2)
PRINT ("ABC"+"DEF")(1 TO 2)
```

2. Skriv ett program som söker i en sträng och jämför varje tecken med bokstaven A. Varje gång A förekommer skall datorn skriva A på skärmen. Ledning: Använd en loop (snurra) för att underöka strängen.

KAPITEL 20

VEKTORER OCH MATRISER

Antag att Du vill arbeta med en lista av tal. Det kan t ex vara frågan om månatliga utgifter för hushållet. Dessa kan då lagras i datorn som oberoende variabler A, B , C osv men det finns ett mycket smidigare sätt. Med instruktionen DIM A(12) reserveras plats i minnet för 12 st variabler A(1), A(2), A(3) osv. 1, 2, 3 kallas index. och A(12) kallas en vektor med 12 element. Skriv in följande program:

```
10 DIM A(12)
20 FOR N=1 TO 12
30 PRINT A(N)
40 NEXT N
```

Programmet skriver ut de 12 elementens värde (de är alltid =0 i utgångsläget). Gör nu tillägget

```
25 INPUT A(N)
```

Programmet är nu utformat så att Du kan läsa in godtyckliga värden på de 12 elementen som ingår i vår vektor. Att arbeta med vektorer är inte bara ett smidigt sätt att skriva ut eller läsa in en serie av värden! Datorn ger också varje värde sitt speciella namn. Om Du t ex skriver PRINT A(5) som kommer Ditt femte inlästa tal att skrivas ut.

Antag nu att vi vill dela upp hushållskostnaderna i de tre posterna mat, kläder och hyra och redovisa dessa månad för månad. Också här tar vi hjälp av DIM-instruktionen. Med DIM(12,3) reserverar vi plats i datorns minne för en matris med 12* 3 = 36 element. Du kan tänka Dig de olika elementen på detta sätt:

mat	kläder	hyra	
A(1,1)	A(1,2)	A(1,3)	månad nr 1
A(2,1)	A(2,2)	A(2,3)	månad nr 2
etc etc till			
A(12,1)	A(12,2)	A(12,3)	månad nr 12

I programmet på nästa sida skall Du läsa in Dina kostnadsposter i tur och ordning: Först månad 1:s kostnader för mat, kläder och hyra, sedan på samma sätt månad 2:s kostnader o s v.

```
10 DIM A(12,3)
20 FOR N=1 TO 12
30 FOR I=1 TO 3
40 INPUT A(N,I)
50 PRINT AT N,(I-1) * 7;  A(N,I)
60 NEXT I
70 NEXT N
```

DIM tillåter inte bara att vi dimensionerar matrisen på detta sätt utan precis som var fallet med vektorer så ges varje element samtidigt ett eget namn: Skriver Du PRINT A(7,2) så skrivs kostnaden för kläder i juli månad ut! Det här var väl bra? Men det kan bli ännu bättre! Med DIM kan du sätta upp matriser av vilken dimension som helst (bara minnet räcker till!) s k multidimensionella matriser. Du kan därför bestämma att Du skall bokföra kostnaderna för mat, kläder och hyra, månad för månad i, säg, 5 år. Det gör Du med DIM A(5,12,3) Elementet A(2,7,2) är då kostnaden för kläder i juli månad år 2! Du kan på detta sätt bygga på i det oändliga.

Vektorer och matrisbegreppet kan också användas på strängvariabler. Även här kan Du använda hur många dimensioner Du vill. Ett enkelt programexempel ger vi här som läser in 5 textsträngar och skriver ut dem.

```
10 DIM A$(5,5)                5 strängar om max 5 tecken
20 FOR I=1 TO  5
30 INPUT A$(I)
40 PRINT A$(I)
50 NEXT I
```

Varje tecken i de 5 strängarna får också sitt namn. Med PRINT A$(2,3) skrivs tredje tecknet i andra strängen ut. Om andra strängen är ordet DATOR skrivs alltså T ! Matrisen ser ut så här:

```
A$(1):        A$(1,1),A$(1,2), ... A$(1,5)
A$(2):        A$(2,1),  o s v
...
A$(5):        A$(5,1)        ....   A$(5,5)
```

Med PRINT A$(1) skrivs hela sträng 1 ut. Med PRINT A$(5,5) skrivs 5:e tecknet i 5:e strängen ut etc.

Sammanfattning:
Vektorer, matriser, strängvektorer
Instruktioner: DIM

Övningar:
1. Sätt upp en strängvektor som kan användas för att skriva ut årets månader. Gör ett program för detta.

2. Gör om programmet så att Du kan läsa in ett månadsnr och få motsvarande månad utskiven.

KAPITEL 21

NÅGRA MER SPECIELLA INSTRUKTIONER

ZX81 printer

Några instruktioner är avsedda att användas för utskrift på skrivare.

LLIST	ger samma effekt som LIST men utskriften görs på skrivaren
LPRINT	ger utskrift på skrivare
COPY	ger en kopia av skärmen till skrivaren

ZX81 kan inte direkt anslutas till vilken skrivare som helst eftersom teckenkoderna är speciella för datorn. Nu har den en skrivare som är tillverkad just för ZX81 och då behövs naturligtvis inga anpassningar för att det hela skall fungera.

PEEK, POKE och USR

är tre intressanta instruktioner/funktioner. Nu kräver de tämligen ingående kunskaper om datorns uppbyggnad och funktion för att ge fullt utbyte.

Med PEEK kan vi studera innehållet i datorns minne, ta fram information direkt ur detta för olika ändamål. Med POKE kan vi placera information direkt i en minnescell. Pröva

PRINT PEEK 17300. Anteckna innehållet (antagligen 0) Skriv sedan POKE 17300,57. Instruktionen lagrar talet 57 i minnescell 17300. Kontrollera med
PRINT PEEK 17300

USR anropar en maskinspråksrutin på en bestämd adress och utför den. Maskinspråksrutiner är ett slags dataprogram som bara består av koder. För att kunna arbeta med USR är kännedom om datorns funktion ett måste.

KAPITEL 22

ATT PROGRAMMERA EN DATOR

Vid det här laget bör Du vara väl bekant med hur datorn ZX81 reagerar på de olika instruktioner den förstår att arbeta med. Du kanske t o m kan göra egna "riktiga" program! Program som kan användas för både nytta och nöje.

Att programmera en dator är roligt (vilket du säkert noterat) och datorn ZX81 är, det lilla formatet och låga priset till trots, en mycket "kunnig" dator. Den är fullt jämbördig med flera av sina betydligt dyrare kollegor och i vissa avseenden t o m bättre! Ta EDIT-funktionerna t ex. Eller exekveringshastigheten i FAST-läge, syntaxkontrollen, entangentsorden - alla är de egenskaper som underlättar och gör det trevligt att arbeta med datorn. Instruktionsrepertoaren och funktionerna för matematiska problem och stränghantering ger tillsammans med noggrannhet i talrepresentationen förutsättningar för mycket kvalificerade program.

För Dig som vill veta mera om hur ZX81 fungerar och kan användas har Studieförlaget i Uppsala utarbetat ett kursmaterial som består av grundbok (ca 175 sidor) och en studiehandledning. Här ges ingående beskrivningar om datorns funktion och programmering med rikliga exempel, betydligt mer än vad som ryms inom ramen för denna introduktion till ZX81 Basic. På grundboken, som heter I närkamp med mikrodatorn, bygger fortsättningsmaterialet Mer om Basic samt Maskinspråksprogrammering. Också dessa kommer från Studieförlaget.

Kunskap om datorer är viktigt. I allt fler tillämpningar tas datorn till hjälp. Zx81 tillsammans med den omfattande dokumentation som finns har Du stora möjligheter att lära Dig handskas med datorn. Kunskap om ZX81 och hur den programmeras och kan användas gäller inte bara ZX81. Datorer är i sig mycket lika och kan man en dator ordentligt är det lätt att snabbt lära sig handskas med andra, såväl stora som små!

KAPITEL 23

SAMMANSTÄLLNING ZX81 BASIC

Instruktion/ Funktion	Operand	Resultat
ABS	numerisk	absolutbelopp av ett tal
ACS	numerisk	arccosinus i radianer
AND	binär logik, högra operanden alltid num. vänstra kan vara en strängoperand	a AND b = a om b≠0 =0 om b=0 A$ AND b = A$ om b≠0 ="" om b=0
ASN	numerisk	arcsinus i radianer
ATN	numerisk	arctangens i radianer
CHR$ n	numerisk	det tecken vars kod är n
CODE	sträng	koden till första tecknet i en sträng
COS	numerisk (rad)	cosinus
EXP x	numerisk	e^x
INKEY$	ingen	läser tangentbordets grundtecken
INT	numerisk	heltalsvärdet avrundat nedåt
LEN	sträng	längden av en sträng (antal tecken)
LN	numerisk	naturliga logaritmen
NOT x	numerisk	0 om x≠0, 1 om x=0
OR	binär logik, numerisk	a OR b = 1 om b≠0 =a om b=0
PEEK x	numerisk	värdet av ett tal i minnesadress x
PI	ingen	3.14.... konstanten pi
RND	numerisk	ger slumptal 0 till o.99999999
SGN	numerisk	teckenbiten till ett tal (+1, -1 eller 0)

SIN	numerisk (rad)	sinus
SQR	numerisk	kvadratroten
STR$	numerisk	får ett tal att uppfattas som en sträng
TAN	numerisk (rad)	tangenten
USR	numerisk	anropar en maskinspråksrutin
VAL	sträng	tolkar strängens numeriska värde
-	numerisk	negation

Aritmetiska operationer

+	addition av tal eller strängar	prio 3
-	subtraktion	6
*	multiplikation	8
/	division	8
**	'upphöjt till' , exponentiering	10
=	lika med	5
>	större än	5
<	mindre än	5
>=	större än eller lika	5
<=	minre än eller lika	5
<>	inte lika med	5

Instruktioner

CLEAR	tar bort alla variab. och för dessa reserverat utrymme
CLS	tar bort allt skrivet på skärmen
CONT	fortsätter programkörning efter avbrott
COPY	kopierar skärmen till skrivare
DIM A(n)	reserverar plats för en vektor A med n element
DIM A(n, m)	reserverar plats för en matris med n x m element
DIM A$(n, m)	reserverar plats för en strängvektor men n strängar om m tecken
FAST	snabb programkörning utan bild
FOR TO	startar en snurra
STEP	bestämmer indelningen av en snurrvariabel
GOSUB	ger hopp till subrutin
GOTO	ger hopp till bestämd rad i programmet
IF ...THEN	testar ett vilkor och ger val av programdel beroende på utfall

INPUT	avbryter programkörning för inläsning av variabel
LET	tilldelar en variabel ett värde eller beräknar ett uttryck
LIST	listar programmet
LLIST	listar programmet på skrivare
LOAD	söker efter ett program och för över det till datorn (från band)
LPRINT	skriver ut på printer
NEW	tömmer minnet på allt lagrat
NEXT	avslutar en snurra
PAUSE	ger paus i programkörningen
PLOT	ritar en svart punkt i ett koordinatsystem
POKE	lagrar ett värde i en bestämd minnesadress
PRINT	skriver ut på skärmen, beräknar ett uttryck och skriver resultatet
RAND	styr slumptalsgenereringen
REM	ingen effekt, anv för anteckningar i programmet
REURN	anv för att avsluta en subrutin. ger hopp till huvudprogram
RUN	startar programmet efter att först tagit bort alla variabelvärden
SAVE	lagrar program och data på band
SCROLL	ger utskrift på rad 21. Skrämtexten "rullas" upp
SLOW	normalhastighet, bild visas under exekvering av program
STOP	avbryter programkörning
UNPLOT	ger en vit punkt i ett koordinatsystem

Felkoder

0	kommando eller program utfört utan fel
1	kontrollvariabel fel
2	variabel ej definierad
3	indexerad variabel fel
4	ingen plats kvar i minnet
5	ingen plats kvar på skärmen
6	aritmetiskt overflow
7	GOSUB saknas till RETURN
8	INPUT får inta anv som kommando
9	STOP utfört
A	felaktigt argument i vissa funktioner
B	heltal för stort
C	strängen kan inte beräknas med VAL
D	BREAK utfört
E	används inte
F	programnamn får ej vara tom sträng

APPENDIX

UTBYGGNADSMÖJLIGHETER, GARANTI, ADRESSER M.M

Utbyggnad av ZX81

För mer omfattande program och program som arbetar med stora datamängder är det interna minnet inte tillräckligt. Vidare kan man vilja använda datorn för att styra och reglera olika utrustningar t ex elektriska modelltåg, fysikexperiment, industriapparatur m m, vilket förutsätter en s k in/utenhet (I/O). I andra tillämpningar är skrivare ett värdefullt hjälpmedel för utskrift av program och arbetsresultat. Till datorn ZX81 och modifierad ZX80 finns därför extrautrustning för att göra datorn mer användbar i praktiska uppgifter:

- 16 k RAM minnesexpansionskort
- ZX81 printer (skrivare)
- I/O-kort

Observera: Det finns redan nu flera mindre företag och privatpersoner som säljer egna tillsatsutrustningar av olika slag till ZX80/81. Eftersom vissa elektroniska kretsar i datorn kan skadas av felaktigt dimensionerade tillsatser vill vi klargöra att fel som uppstått på detta sätt ej täcks av garantin. Ej heller gäller här fasta servicedebiteringar.

Litteratur

- I närkamp med mikrodatorn, grundbok, 2:a upplagan. En bok om teknik, programmering och användning av datorn ZX81. Rikligt med exempel och övningar. 175 sidor i A4 format. Studieförlaget.
- I närkamp med mikrodatorn, studiehandledning: Ger kommentarer och anvisningar, facit samt fördjupningsavsnitt till grundboken.
- Mer om BASIC: En bok i avancerad BASIC-programmering med datorn ZX81 (utkommer nov 81). Studieförlaget.
- Maskinspråksprogrammering: En bok om processorn z80 och dess olika instruktioner, utformning av maskinspråksrutiner, anslutning av datorn ZX81 till yttre enheter m m (nov 81). Studieförlaget.

Programvara

En rad olika program för datorn ZX80/81 finns framtagna och nya utvecklas både av Beckman Innovation AB och Studieförlaget. Ofta finns de inspelade på kassett.

Användarklubb ZX80/81

Om Du är intresserad av utbyte av idéer, programvara m m med likasinnade kan kanske medlemskap i den användarklubb som är under uppbyggnad vara till glädje. Kontakta Beckman Innovation AB!

Kurser i grundläggande mikrodatorkunskap

Flera företag och organisationer har använt föregångaren ZX80 i sin grundläggande datautbildning för personalen. ZX81 och andra upplagan av "I närkamp med mikrodatorn" är än mer lämpad för detta ändamål. Kontakta Studieförlaget för mer information.

Garanti

Sinclair ZX81 som köpts av Beckman Innovation AB eller våra återförsäljare omfattas av 1 års fabrikationsgaranti, täckande ingående delar samt funktion. Service utförs av Beckman Innovation AB, fritt serviceverkstaden - d v s eventuella postporton betalas av ZX81-ägaren. För byggsatser gäller att bygget skall vara utfört enligt medlevererad bygganvisning om garantin skall gälla. Vid reparation av fel utanför garantin eller om garantin upphört att gälla, tillämpas fasta priser (hösten 81 är normaldebitering för felsökning och reparation 100:- + moms och frakt). Inköpskvittot eller kopia måste alltid medsändas vid åberopande av garantin.

Adresser:

Generalagent
Beckman Innovation AB
Gamla Dalarövägen 2
Box 7
122 21 Enskede
Tel 08/39 04 00
Telex 10318 Beckman S

Studieförlaget
Box 386
751 06 Uppsala
Tel 018/15 53 90

Josty Kit AB
Malmö
tel 090/126718