

DATOR- UTBILDNING FÖR ALLA



TEXAS
INSTRUMENTS

DATORUTBILDNING FÖR ALLA

Detta häfte har framställts med TI-99/4A och
ordbehandlingsprogrammet TI Writer tillsammans
med en skönskrivare.

Häftet är en självständig svensk version av "Computer Awareness
Program for Adults".

(c) Texas Instruments Sweden 1983

Innehållsförteckning

Vad är en mikrodator?	5
Hur en dator fungerar	7
Vad består ett hemdatorsystem av?	8
En tangentguide	9
Tre sätt att lagra program och data	10
Programspråk	11
Instruktioner i TI BASIC	14
Färgkoder	15
Program i TI BASIC	16
Enkel färgdemonstration	16
Momsberäkningar	17
Å, Ä och Ö	18
Medelvärde	18
Sinuskurva	19
Syntetiskt tal	19
Ljud	19
Tärningar	20
Mer om TI LOGO	21
Användarföreningar	27
Tidskrifter	29
En översikt över datorns utveckling	30
Datorordlista	32

Vad är en mikrodator?

Datorer skapar för många av oss två mäktiga sinnesbilder. Den ena är som en scen hämtad ur en science fiction film i vilken kulisser fulla med lampor, knappar och spakar ställs in för en resa i rymden eller genom tiden. Vi tror inte på vad som händer och vi undrar om skådespelarna gör det.

Den andra är ett enormt rum fyllt med lådliknande utrustning - även denna fullspäckad med lampor, knappar och spakar. I dessa rum görs mängder av beräkningar, rymdflygningar övervakas, vetenskapsmän och matematiker forskar med sådant som tidigare var omöjligt.

Oavsett om man ser datorerna som en uppdiktad saga eller som teknisk verklighet så har många av oss svårt att tro vad vi ser. Föreställningen om datorer är något långt ifrån verkligheten och med säkerhet långt ifrån vårt dagliga liv.

Teknologin har emellertid med överraskande fart gjort datorn till en del av vårt liv. För fem år sedan skulle flertalet av er inte ens funderat över att läsa det här häftet, än mindre över att själv ha en dator hemma. Vad som hänt på senaste tiden är att teknologin har miniatyriserat datorn. Uträkningar som tidigare endast kunde genomföras på utrustning som krävde ett enormt rum och över fem ton luftkonditionering för att fungera kan nu utföras med en utrustning som lätt får plats på skrivbordet.

Det är nu möjligt att skaffa en hemdator och för varje år köper fler och fler av oss datorer. Datorer ger oss möjlighet att underlätta rutinmässig registerhantering i hemmet och i affärslivet, att öka utbildningsmöjligheterna för oss själva och för våra barn och tack vare utvecklingen av videospel - ha roligt.

Mikrodatorns popularitet har överöst oss med nya ord. Vi hör nu ord som CPU, RAM, ROM, bytes och bits, disks, chips, minne och mikrodator. För flertalet av oss är de nya orden lika obegripliga i våra försök att förstå hemdatorn som datorn i sig själv.

Ett av de enklaste sätten att förstå dessa nya termer är att jämföra dem med något som vi alla är vana vid - den mänskliga hjärnan. Vi får information om världen omkring oss genom våra sinnen. Datorer däremot är beroende av människor för sin information. Vi matar in information i en dator via ett tangentbord. För datorn är kretsarna som kopplats till tangentbordet dess ögon och öron.

Vi bearbetar den mottagna informationen och så gör även datorn. Vi fattar beslut baserade på tidigare erfarenhet. Vi resonerar. Datorn resonerar genom att använda sin centralenhet ("datorns hjärna").

Precis som vi själva behöver datorn "minne", eller lagringsplats, för att kunna behålla och bearbeta information. Människor har korttids- och långtidsminne. Vi kommer t ex ihåg hur man adderar i vårt långtidsminne, men vi kommer bara ihåg talen vi adderar tillräckligt länge för att lösa vår uppgift.

I datorn kallas dessa minnen för ROM (Read Only Memory = läsminne) och RAM (Random Access Memory = skriv/läsminne). ROM innehåller information som alltid behövs och RAM lagrar information tillfälligt - tills du raderar minnet eller slår av datorn. I både RAM och ROM lagras informationen i små minnesceller. Varje cell har en "adress" så att datorn kan hitta den information du vill ha och inte en röra av information. Informationen lagras i små enheter som kallas "bytes". Datorer har gränser för hur mycket information de kan lagra. Exempelvis har TI-99/4A 16 K RAM och kan lagra 16000 bytes.

Datorn liknar den mänskliga hjärnan i det avseendet att båda kan anpassa sig och följa nya instruktioner. Om vi får noggranna instruktioner kan vi använda en ny maskin, sätta ihop en leksak för våra barn eller laga en ny maträtt. Datorns motsvarighet till instruktioner är ett program. Genom att ge datorn ett nytt program lagrat på antingen kassettband, flexskiva eller programmodul kan du få datorn att följa nya instruktioner.

Datorn hjälper oss och hanterar information åt oss, men det arbetet skulle vara värdelöst om datorn inte kunde ge oss resultatet. Datorn presenterar resultatet på en TV eller monitor, eller i vissa fall på en skrivare. Vi kan jämföra det med att vi kommunicerar med andra genom att tala eller skriva.

Datorns grundläggande byggstenar beskrivna ovan ingick i 50-talets stora datorer. Många uppfinningar har gjort det möjligt att miniatyrisera dessa - men den viktigaste är den integrerade kretsen. I form av en integrerad krets kan en kiselkiva innehålla tusentals av datorns kretsar. Den integrerade kretsen har gjort mikrodatorn möjlig.

Teknologin som har gjort mikrodatorn möjlig har skapat nya utmaningar för att bli van vid nya villkor och ny utrustning. Som tur är har mikrodatorer konstruerats för att vara användarvänliga. Vi kan utnyttja mikrodatorer utan behöva bli datorexpert, precis som vi inte behöver bli bilmekaniker för att köra bil eller elektriker för att slå på ljuset.

Datorn är inte längre endast en kuliss i en science fiction film. Den har blivit en bekväm, spännande och underhållande del av det dagliga livet.

Hur en dator fungerar

De flesta datorer är uppbyggda av ungefär samma huvuddelar.

När du skriver på tangentbordet sänder du information till datorn. Detta flöde av information till datorn kallas "indata".

Indatan går till centralenheten där en bearbetning av informationen görs. Centralenheten kan göra beräkningar, jämföra och ordna information. Exakt vad den skall göra bestäms av ett program som är lagrat i minnet. Genom att byta program kan datorn utföra en helt annan uppgift.

Ett program för datorn är en serie instruktioner skrivna i ett lämpligt språk. Normalt förstår en centralenhet endast ett språk bestående av sifferkoder - "maskinspråk". Men med hjälp av ett översättningsprogram eller tolkprogram kan datorn förstå och utföra program skrivna i språk som är mer lika mänskliga språk. Ett sådant program finns ständigt lagrat i hemdatorn TI-99/4A och därför kan du programmera i det lättanvända datorspråket BASIC.

För att lagra program och information behöver datorn minne. Det finns två sorters minnen. Den ena är ROM (läsminne) som innehåller program eller information som alltid behövs. Det kan aldrig raderas eller ändras. Tolkprogrammet för BASIC ligger i ROM. Den andra typen är RAM (läs/skriv minne) som lagrar information tillfälligt. Innehållet försvinner när datorn slås av.

I både RAM och ROM är information uppdelad i små celler som kallas "bytes". Varje byte har en unik adress bestående av ett tal. Genom att ange adressen kan datorn komma åt rätt information. Man mäter minnets kapacitet på hur många bytes de innehåller. Hemdatorn har t ex 16K RAM vilket är ca 16000 bytes (1K=1024 bytes).

En byte kan lagra ett tecken (t ex en bokstav, en siffra eller någon annan symbol). Man brukar även dela in en byte i "bits" eller på svenska helt enkelt "bitar". En bit är den minsta enhet som datorn kan hantera. En bit kan ha två värden: 0 och 1. En byte består av 8 bitar.

När datorn har gjort beräkningar eller hanterat indatan på annat sätt måste datorn meddela resultat till omvärlden. Flödet av information från datorn kallas "utdata". Du ser utdatan på en skärm i form av ord eller bilder. Datorn kan också skriva information på en skrivare om en sådan är ansluten.

Vad består ett hemdatorsystem av?

1. **Hemdatorn TI-99/4A.** En skrivmaskinsliknande grundenhet som används för att mata in, lagra och behandla information.
2. **Färgmonitor.** En tiotums färgskärm som visar 24 rader med 32 tecken per rad. Den kan också återge ljud. En vanlig TV kan också användas.
3. **Utbyggnadsenhet.** Ett kompakt system konstruerat för att placera skivminne, RS232-gränssnitt, minneskort och andra tillbehör på ett ställe.
4. **Enhet för syntetiskt tal, "prator".** En "prator" härmar den mänskliga rösten på elektronisk väg.
5. **Programvara för hemdatorn.** Ett stort bibliotek med färdiga program på kassetter, flexskivor och programmoduler framställda för undervisning, hemekonomi eller kreativa spel.
6. **Styrspakar.** Ger ökade möjligheter att styra spel, grafik och ljud utan att behöva använda tangentbordet.
7. **Skivminne.** Används för att med hög hastighet lagra eller hämta data.



En tangentguide

1. ENTER. Används när du vill tala om för datorn att du har matat in en hel rad. Datorn tar hand om den inmatade raden när du trycker ENTER.

2. SHIFT. Tryck ned denna tangent tillsammans med en bokstavstangent för att få en stor versal. (Små bokstäver ser likdana ut som stora, förutom att de är mindre. Därför brukar man tala om stora och små versaler.) Om den används i kombination med en annan tangent fås symbolen som finns längst upp på tangenten. Exempelvis ger SHIFT 1 ett utropstecken (!).

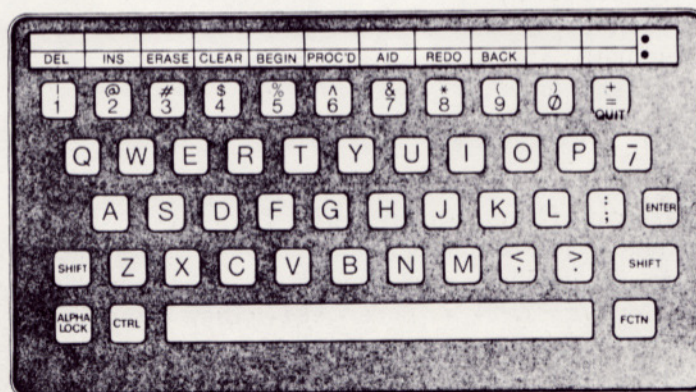
3. ALPHA LOCK. I nedfällt läge kan bara stora versaler matas in. I uppfällt läge ges små versaler (om du inte samtidigt trycker SHIFT). Sifvertangenter ger alltid siffror oberoende av tangentens läge.

4. MELLANSLAGSTANGENT. Ger ett mellanslag (blanktecken). Om man stegar över andra tecken med tangenten raderas dessa.

5. FCTN. Aktiverar vissa speciella funktioner genom att man trycker ned FCTN tillsammans med en sifvertangent. Funktionens namn hittas på instickssremsan ovanför sifvertangenterna. FCTN tillsammans med vissa tangenter som har tecken skrivna på kanten skriver det tecknet på skärmen. Exempelvis ger FCTN P citationstecken (") och FCTN I ger frågetecken. FCTN = (QUIT) stoppar de flesta program och raderar hela minnet.

6. CTRL. Används på liknande sätt som FCTN i vissa program.

OBS! Om man håller nere en tangent längre tid än en sekund repeteras denna tangent automatiskt på skärmen tills tangenten släpps.



Tre sätt att lagra program och data

Programmoduler

Följande utrustning behövs:

Grundenheten
Färgmonitor eller TV
En modul

Programmoduler är utmärkta för att köra färdiga program. De flesta modulerna kräver inga extra tillbehör eftersom de sätts in direkt i grundenheten. Normalt kan innehållet i en modul inte ändras. I den speciella "Mini Memory" modulen kan du dock lagra dina egna program och data.

Kassetter

Följande utrustning behövs:

Grundenheten
Färgmonitor eller TV
En kassettspelare
Kassettkabel
Kassetband av hög kvalitet

Kassetter är ett billigt sätt att lagra egna data eller program och för att köra färdiga program. Även om det ibland tar lite tid att hämta ett program, är kassetter lämpliga om man vill undvika den högre kostnaden för ett skivminne.

Flexskivor

Följande utrustning behövs:

Grundenheten
Färgmonitor eller TV
En skivstyrenhet
Utbyggnadsenhet
En till tre skivenheter
Flexskivor

Skivminne är den mest avancerade metoden för datalagring. Färdiga program finns också på flexskiva (ofta finns samma program både på kassett och flexskiva). Det kostar lite mer att använda skivminne, men pga av flexskivors högre snabbhet är det väl värt det. TI:s skivminne använder 5 1/4 tums enkelsidiga, "mjuksektorerade" flexskivor med enkel lagringstäthet. Ungefär 90000 tecken kan lagras på en flexskiva.

Programspråk

Ett programspråk är precis som alla andra språk avsett för att utbyta information. Språk som är begripliga för en dator skrivs enligt vissa regler. Utan programspråk skulle du inte ha någon möjlighet att ge datorn dina instruktioner. Många programspråk har under årens lopp utvecklats. Några som finns till hemdatorn beskrivs här.

TI BASIC

BASIC är en förkortning för "Beginners All-purpose Symbolic Instruction Code". BASIC bygger på engelska språket - ord som PRINT, GO TO, RUN och END har nästan samma betydelse i engelska och BASIC. Eftersom BASIC är väldigt lätt att lära är språket det populäraste för mikrodatorer.

Liksom svenska "talas" BASIC i olika dialekter. Dialekten för TI-99/4A kallas TI BASIC. Många ägare av hemdatorn tycker att detta språk är fullt tillräckligt för deras behov.

Nödvändiga tillbehör: Ingenting, men för att spara program behövs en kassettbandspelare eller skivminne.

TI Extended BASIC

Om ett mer kraftfullt programspråk behövs är TI Extended BASIC det första i en serie mer avancerade programspråk.

TI Extended BASIC används ofta för professionell programvara. Språket innehåller betydligt fler instruktioner än TI BASIC. Exempelvis har Extended BASIC instruktioner som styr rörlig, animerad grafik, vilket inte är möjligt i TI BASIC.

Extended BASIC ger bl a bättre möjligheter att kontrollera inmatning av data och utskrift av data. Detta språk kan också använda minnesutbyggnaden vilket möjliggör hantering av större informationsmängder. Om skivminne används kan nästan obegränsat stora program skrivas i Extended BASIC.

Nödvändiga tillbehör: Modulen TI Extended BASIC. För att spara program behövs också kassettbandspelare eller skivminne.

Assembler

Assembler är mycket likt maskinspråk. Varje instruktion översätts direkt till en maskinspråksinstruktion. Vid körningen kan datorn utföra programmet direkt vilket gör språket mycket snabbt. Man har också fullständig kontroll över datorn.

Assembler är å andra sidan betydligt svårare att använda än andra språk som t ex BASIC och är därför inte avsett för nybörjaren.

Nödvändiga tillbehör: Utbyggnadsenhet, skivminne, minneskort, modulen Editor/assembler.

UCSD Pascal

Pascal ger dig tillgång till en stor mängd professionella program. Pascalprogram är lätta att läsa och förstå. I motsats till BASIC har Pascal konstruerats för att kunna köras på olika sorters datorer.

Pascalprogram översätts till p-kod som skall gå att köra på alla Pascaldatorer. P-koden översätts för körning på hemdatorn med TI:s p-kodkort.

Nödvändiga tillbehör: P-kodkort, skivminne med minst två skivenheter, utbyggnadsenhet, minneskort, följande program på flexskivor: Pascal-kompilator, Editor/Filer.

TI LOGO

TI LOGO är ett extremt lätt att använda och gör inlärninng ROLIGT. Det är särskilt avsett att ge barn datorvana och samtidigt lära dem matematik och kommunikation.

I TI LOGO kan du få fram spännande grafik med enkla kommandon. Du kan börja med att rita bilder, fortsätta med animerad grafik till matematisk problemlösning.

Nödvändiga tillbehör: Modulen TI LOGO, utbyggnadsenhet, minneskort. För att spara program behövs antingen kassettbandspelare eller skivminne.

TI FORTH

FORTH delar flera av fördelarna med de andra språken beskrivna här. Det är nästan lika snabbt som assembler och ger nästan samma kontroll över datorn. Samtidigt är det betydligt lättare att använda. Ordentligt skrivna FORTH-program är också lätta att läsa och förstå.

Programmering i FORTH går till så att man utökar språket med nya instruktioner. På det sättet kan man "skräddarsy" sitt eget språk.

Nödvändiga tillbehör: Flexskiva med FORTH, utbyggnadsenhet, minneskort, skivminne, modulen Editor/assembler.

TI COMAL80

COMAL80 är ett språk utvecklat i Danmark. Det bygger delvis på BASIC och är lika lätt att lära sig. Samtidigt har det konstruerats för att ge mer lättlästa och mer lättändrade program. Sö har beslutat att COMAL80 skall användas i skolornas datorundervisning.

TI:s version har dessutom samma möjligheter som Extended BASIC att hantera grafik.

Nödvändiga tillbehör: Modulen COMAL80. För att spara program: Kassettbandspelare eller skivminne.

TI PILOT

PILOT är ett språk som är särskilt lämpligt för datorstödd undervisning. Lärare kan lätt lära sig språket och göra egna undervisningsprogram.

Nödvändiga tillbehör: Flexskiva med TI PILOT, minneskort, P-kod kort, utbyggnadsenhet, skivminne.

Instruktioner i TI BASIC

Kommandon

LIST: visar programmet i minnet.

NEW: tar bort programmet i minnet. Använd det innan du matar in ett nytt program.

RUN: "kör" eller utför programmet i minnet.

SAVE: används för att spara programmet i minnet på kassett (SAVE CS1) eller på flexskiva (SAVE DSK1.FILNAMN).

OLD: används för att hämta ett program från kassett (OLD CS1) eller flexskiva (OLD DSK1.FILNAMN) till datorns minne.

EDIT: EDIT 100 visar rad 100 så att den kan ändras.

BYE: lämnar TI BASIC.

Instruktioner

Ett exempel visas tillsammans med alla instruktioner.

LET: LET PRIS=150 ger värdet av uttrycket till höger om likhetstecknet till variabeln PRIS. (LET får uteslutas i TI BASIC. Man kan alltså skriva PRIS=150.)

INPUT: INPUT PRIS stannar programmet och väntar på inmatning av ett tal. Värdet tilldelas variabeln PRIS.

PRINT: PRINT "HEJSAN" skriver ut HEJSAN på skärmen. PRINT PRIS skriver ut värdet av variabeln PRIS på skärmen.

IF-THEN: IF A=5 THEN 40 testar om A är lika med 5. Om så är fallet försätter programmet på rad 40. Annars försätter programmet till nästa rad.

GOTO: GOTO 500 fortsätter programkörningen på rad 500.

FOR-NEXT: FOR I=1 TO 10 en slinga som upprepas tio gånger. I har i tur och ordning värdena 1, 2, 3, 4, 5, 6, 7, 8, 9 och 10.

DATA: DATA 31,14,90 lagrar data i programmet. Dessa kan läsas med READ-instruktionen.

READ: READ X,Y,Z ger de tre variablerna värden från en DATA-sats någon annanstans i programmet.

REM: REM MOMSPROGRAM ignoreras av datorn, avsedd som en kommentar till mänskliga läsare.

CALL CHAR: CALL CHAR(33,"0000FF") omdefinierar utropstecknet (ASCII-kod 33) till ett nytt mönster.

CALL COLOR: CALL COLOR(1,16,13) sätter färg på teckengrupp 1, vit (16) som förgrundsfärg och mörkgrön (13) som bakgrundsfärg.

CALL SCREEN: CALL SCREEN(9) ändrar färgen på skärmen till mellanröd.

CALL SOUND: CALL SOUND(1000,440,1) spelar en ton med frekvensen 440 Hz med volymen 1 (ganska högt) under en sekund (1000 millisekunder).

CALL HCHAR: CALL HCHAR(1,4,42,10) ritar en vågrät rad med 10 asterisker (kod 42) med början vid rad 1, kolumn 4.

CALL VCHAR: CALL VCHAR(2,5,42,10) ritar en lodrät linje med 10 asterisker (kod 42) med början vid rad 2, kolumn 5.

FÄRGKODER

Färgkod Färg

- | | |
|----|----------------|
| 1 | genomskinlig |
| 2 | svart |
| 3 | mellangrön |
| 4 | ljusgrön |
| 5 | mörkblå |
| 6 | ljusblå |
| 7 | mörkröd |
| 8 | cyan |
| 9 | mellanröd |
| 10 | ljusröd |
| 11 | mörkgul |
| 12 | ljusgul |
| 13 | mörkgrön |
| 14 | magenta (lila) |
| 15 | grå |
| 16 | vit |

BÖCKER OM BASIC

Texas Instruments, Bruksanvisning för TI-99/4A.

Texas Instruments, BASIC på Texas Instruments hemdator, Studentlitteratur 1983.

Program i TI BASIC

Programmen på följande sidor kan hjälpa dig att bli bekant med TI BASIC. Bredvid varje rad finns en förklaring till vad den gör.

Om du gör ett fel medan du matar in en rad, använd vänsterpil (FCTN S) för att gå tillbaka till felet. Skriv sedan in det riktiga. Hela raden kan också tas bort med FCTN 3.

Om du hittar ett fel när du redan har tryckt ENTER, mata in raden igen med radnummer och tryck ENTER. Den nya raden ersätter den gamla i minnet.

Du kan även använda kommandot EDIT som beskrivs i Bruksanvisningen.

Skriv RUN och tryck ENTER när du har matat in programmet för att köra det.

Enkel färgdemonstration

Det första programmet visar alla färger som datorn kan hantera.

100 CALL CLEAR	Töm skärmen.
110 CALL SCREEN(8)	Sätt skärmfärgen till cyan (ljus blå).
120 FOR I=2 TO 16	Börja en slinga som genomlöper färgerna 2 till 16.
130 CALL COLOR(I,I,I)	Sätt förgrunds- och bakgrunds-färg för uppsättning I till färg I (2 till 16).
140 FOR J=19 TO 21	Börja en slinga som genomlöper raderna 19 till 21.
150 CALL HCHAR(J,I*2-1,24+8*I,2)	Sätt ut två tecken på skärmen.
160 NEXT J	Avsluta slinga.
170 NEXT I	Avsluta slinga.
180 GOTO 180	Avsluta inte programmet förrän FCTN 4 trycks.

Momsberäkningar

De följande programmen visar hur ett enkelt program för momsberäkning utvecklas. Den första varianten läser in priset i variabeln P, beräknar momsen M som $P \cdot 0.2346$ (pris gånger 23,46 procent) samt skriver ut M.

100 INPUT P	Läs in ett tal och ge värdet till variabeln P.
110 M=P*0.2346	Beräkna moms som priset gånger 23,46 % och ge variabeln M detta värde.
120 PRINT M	Skriv ut momsen

Låt oss nu bygga ut detta program. Vi lägger till beräkning av pris med moms. Dessutom byter vi variabelnamnen P och M mot namn som säger något vad det handlar om - PRIS, MOMS och PRISMEDMOMS.

100 INPUT PRIS	Läs in pris till variabeln PRIS.
110 MOMS=PRIS*0.2346	Beräkna moms.
120 PRISMEDMOMS=PRIS+MOMS	Beräkna pris med moms som pris+moms.
130 PRINT MOMS;PRISMEDMOMS	Skriv ut moms och pris med moms.

Detta fungerar ju - men särskilt användarvänligt är det inte. Hur kan man veta vad som skall matas in när man ser ett ensamt frågetecken på skärmen? Och vad betyder siffrorna som skrivs ut?

Vi ändrar programmet på följande sätt:

90 CALL CLEAR	Töm skärmen.
100 INPUT "VARANS PRIS? ":PRIS	Fråga efter varans pris.
110 MOMS=PRIS*0.2346	Beräkna moms.
120 PRISMEDMOMS=PRIS+MOMS	Beräkna pris med moms.
130 PRINT "PRIS: ";PRIS	Skriv ut pris.
140 PRINT "MOMS: ";MOMS	Skriv ut momsen.
150 PRINT "PRIS MED MOMS: ";PRISMEDMOMS	Skriv ut pris med moms.
160 PRINT ::	Två tomrader.
170 GOTO 100	Upprepa.

Detta program stoppas inte förrän du trycker FCTN 4.

Å, Ä och Ö

Det är lätt att själv definiera Å, Ä och Ö.

Prova följande program:

100 CALL CHAR(91,"00280038447C4444")	Definiera
110 CALL CHAR(92,"0028003844444438")	svenska
120 CALL CHAR(93,"00100038447C4444")	tecken.
130 PRINT "EN SJ\MAN [LSKAR HAVETS V]G"	Exempel.
140 GOTO 140	Fortsätt tills FCTN 4 trycks.

Du kan använda Å, Ä och Ö i dina egna program genom att lägga raderna 100-120 först och sedan i alla PRINT-satser skriva Å som höger hakparentes, Ä som vänster hakparentes och Ö som bakvänt bråkstreck. När du kör programmet visas de svenska tecknen.

Medelvärde

Följande program beräknar medelvärde för ett antal tal. När du kör det frågar det efter ett tal. När ett tal har matats in frågar det efter ett nytt. När du har matat in alla tal, svara 999. Då beräknar datorn medelvärdet och skriver ut. Du kan ändra 999 i rad 170 till ett annat tal du vill kunna mata in 999 som ett tal.

OBS! I listningen ingår bokstaven Ä. Denna måste du mata in som vänster hakparentes (FCTN R).

100 CALL CHAR(91,"00280038447C4444")	Definiera
110 CALL CHAR(92,"0028003844444438")	svenska
120 CALL CHAR(93,"00100038447C4444")	tecken.
130 CALL CLEAR	Töm skärmen.
140 ANTAL=0	Nollställ räknaren för antal inmatade tal.
150 SUMMA=0	Nollställ talens summa.
160 INPUT "TAL: ":TAL	Fråga efter ett tal.
170 IF TAL=999 THEN 210	Testa om slut på tal.
180 SUMMA=SUMMA+TAL	Summera talet.
190 ANTAL=ANTAL+1	Räkna upp antalet.
200 GOTO 160	Upprepa.
210 PRINT "MEDELVÄRDE: ";SUMMA/ANTAL	Beräkna och skriv ut medelvärde.
220 INPUT "NY BERÄKNING? ":SVAR\$	Fråga om ny beräkning.
230 IF SVAR\$="J" THEN 140	Hoppa tillbaka om svar "J".

Sinuskurva

Detta enkla program frågar efter ditt namn (högst 10 tecken) och ritar sedan en sinuskurva med namnet. Prova!

100 CALL CHAR(91,"00280038447C4444")	Definiera
110 CALL CHAR(92,"0028003844444438")	svenska
120 CALL CHAR(93,"00100038447C4444")	tecken.
130 CALL CLEAR	Töm skärmen.
140 X=0	Nollställ vinkel.
150 INPUT "VAD HETER DU? ":NAMN\$	Fråga efter namn.
160 PRINT TAB(10+9*SIN(X));NAMN\$	Skriv ut namn vid position som bestäms av sinus- funktionens värde.
170 X=X+0.3	Öka X-värdet.
180 GOTO 160	Upprepa.

Syntetiskt tal

Om du har modulen Terminal Emulator II och "prator" (TI:s enhet för syntetiskt tal) är det här något för dig. Mata in vad som helst och datorn säger det! Den pratar engelska bäst, men klarar även svenska med en viss accent.

100 OPEN #1:"SPEECH",OUTPUT	Öppna pratorns fil.
110 INPUT A\$	Läs in något att säga.
120 PRINT #1:A\$	Skriv till pratorn.
130 GOTO 110	Upprepa.

Ljud

Det här programmet spelar en hel oktav med toner (med halva steg). Du anger bara startfrekvensen, t ex 262 för ettstrukna C.

100 INPUT "STARTFREKVENNS? ":FREKVENNS	Läs in startfrekvens.
110 FOR I=1 TO 12	Börja en slinga.
120 CALL SOUND(300,FREKVENNS,0)	Skapa ljud med längden 300 millisekunder, frekvensen FREKVENNS och volymen 0.
130 FREKVENNS=FREKVENNS*2^(1/12)	Beräkna nästa frekvens.
140 NEXT I	Avsluta slinga.
150 FOR I=1 TO 12	Börja slinga.
160 CALL SOUND(300,FREKVENNS,0)	Skapa ljud.
170 FREKVENNS=FREKVENNS*2^-(1/12)	Beräkna nästa frekvens.
180 NEXT I	Avsluta slinga.

Tärningar

Det här exemplet visar en enkel simulering av tärningar. Du anger antal kast och sedan "kastar" datorn två tärningar så många gånger. Funktionen RND som används ger ett tal mellan 0 och 1. När detta multipliceras med 6 fås ett tal mellan 0 och 6. När man tar heltalsdelen med INT fås ett heltal 0 till 5. Slutligen lägger man bara till 1.

100 CALL CLEAR	Töm skärmen.
110 RANDOMIZE	Ge slumpalsgeneratorn ett slumpmässigt startvärde.
120 INPUT "ANTAL KAST? ":ANTAL	Fråga efter antal kast.
130 FOR KAST=1 TO ANTAL	Börja en slinga.
140 TARNING1=INT(6*RND)+1	Slumpa fram kast 1.
150 TARNING2=INT(6*RND)+1	Slumpa fram kast 2.
160 PRINT TARNING1;TARNING2,TARNING1+TARNING2	Skriv ut kasten och summan.
170 NEXT KAST	Avsluta slinga.

Mer om TI LOGO

I det här avsnittet kommer vi att gå igenom grundläggande LOGO-kommandon. Några exempel som visar LOGOs fantastiska grafiska möjligheter visas också på slutet.

LOGO är redan från början vänligare än andra språk. När man går in i LOGO visas: "WELCOME TO TI LOGO!" Detta visas också av dess felmeddelanden. Om du t ex skriver HEJ svarar LOGO:

```
TELL ME HOW TO HEJ
```

Du kan då lära LOGO att säga HEJ. Skriv bara TO HEJ och du kommer in i ett särskilt redigeringsläge. Se till att det står:

```
TO HEJ  
PRINT "HEJ  
END
```

och tryck BACK (FCTN 9). Nu vet LOGO hur man säger HEJ. Skriv HEJ och LOGO svarar HEJ.

LOGOs vänliga sköldpadda

Låt oss nu "prata" med LOGOs vänliga sköldpadda (turtle). Vi skriver alltså:

```
TELL TURTLE
```

Ungefär i mitten av skärmen dyker då sköldpaddan upp i form av en triangel. Skriv

```
FORWARD 30
```

så går sköldpaddan 30 steg framåt, vilket är rakt uppåt. Den lämnar ett streck efter sig. Vi kan vrida sköldpaddan genom att skriva t ex

```
RIGHT 90
```

vilket vrider den 90 grader åt höger. Du ser att sköldpaddan vrider sig och nu är riktad "österut" (rakt åt höger). Skriv

```
FORWARD 30
```

så går sköldpaddan framåt, dvs åt höger och vi har två streck i rät vinkel. Om vi skriver

```
RIGHT 90  
FORWARD 30  
RIGHT 90  
FORWARD 30
```

så har vi en fullständig kvadrat.

Om vi ofta vill rita kvadrater kan vi skriva

TO KVADRAT

och tala om för LOGO hur man ritar en kvadrat, t ex:

```
FORWARD 50  
RIGHT 90  
FORWARD 50  
RIGHT 90  
FORWARD 50  
RIGHT 90  
FORWARD 50
```

Denna kvadrat är större än vår föregående: varje sida är 50 enheter i stället för 30.

Nu kanske skärmen börjar se rörig ut. I så fall kan du tömma den om du skriver

CLEARSCREEN

På sidan 26 finns en annan definition av KVADRAT där du anger storleken. Den används så här:

KVADRAT 25

ritar en kvadrat med sidan 25 och

KVADRAT 70

ritar en kvadrat med sidan 70.

I definitionen har REPEAT 4 använts. Detta säger åt LOGO att repetera kommandona

FORWARD :STORLEK RIGHT 90

inom hakparenteser 4 gånger. :STORLEK har det värde som anges efter KVADRAT, t ex 25 eller 70.

Innan vi lämnar vår vän sköldpaddan så kan vi titta på några fler kommandon som den förstår.

HIDETURTLE säger åt sköldpaddan att bli osynlig. Linjer kan fortfarande ritas (och det går något snabbare).

SHOWTURTLE säger åt sköldpaddan att bli synlig igen efter HIDETURTLE.

PENUP lyfter upp pennan som sköldpaddan ritar med. Sköldpaddan kan då flyttas vart som helst utan att rita någon ny linje.

PENDOWN för ner pennan igen så att linjer ritas när sköldpaddan rör sig.

HOME flyttar sköldpaddan till dess "hem" mitt på skärmen utan att rita någon linje. Den riktas också in "norrut" (rakt upp).

SETHEADING följt av antal grader ställer in riktningen som

sköldpaddan skall röra sig i. Graderna anges som på en kompass, dvs SETHEADING 0 riktar sköldpaddan norrut (uppåt), SETHEADING 90 österut (höger), SETHEADING 180 söderut (nedåt) och SETHEADING 270 västerut (vänster).

NOTURTLE tar bort sköldpaddan från skärmen och raderar det som har ritats.

Skriv nu NOTURTLE för att ta bort sköldpaddan.

LOGOs sagoväsen

LOGO har också andra grafiska möjligheter, t ex "sagoväsen" eller "sprites" som det heter på engelska. De kallas sagoväsen eftersom de kan bära olika skepnader, röra sig självständigt på skärmen och passera över andra tecken.

Från början är alla sagoväsen osynliga. Låt oss plocka fram ett sagoväsen. Skriv

TELL SPRITE 0

för att tala med sagoväsen nr 0.

Vi måste nu tala om dels hur sagoväsendet skall se ut ("vilken skepnad det skall bära"). Skriv

CARRY :BALL

och sagoväsendet kommer att se ut som en boll. Vi ser ännu inte bollen. Sedan måste vi ge den en färg:

SETCOLOR :RED

När vi slutligen ger den en position dyker den upp. Det lättaste är att skriva

HOME

vilket flyttar bollen till mitten av skärmen (vi kan använda samma kommando för sköldpaddan).

Vi kan sätta fart på bollen med

SETSPEED 20

vilket ger bollen hastigheten 20 riktad rakt uppåt. När bollen når övre kanten av skärmen dyker den upp vid nedre kanten och fortsätter uppåt.

Maxhastigheten är 127. Prova

SETSPEED 127

Nu går det undan! Om vi vill stoppa bollen sätter vi hastigheten till 0:

SETSPEED 0

Vi kan också få bollen att röra sig i andra riktningar:

SETHEADING 45

SETSPEED 10

SETHEADING har samma betydelse som för sköldpaddan. 45 betyder att sagoväsendet skall röra sig mot nordost (snett uppåt till höger).

Om vi tillfälligt vill stoppa alla sagoväsen skriver vi

FREEZE

De startar igen med

THAW

Hur gör man om man inte vill att sagoväsendet skall se ut som en boll? Man kan använda kommandot MAKESHAPE följt av ett tal mellan 1 och 25. Prova t ex

MAKESHAPE 6

Du får upp ett rutnät med 16 gånger 16 rutor. Där kan du rita in hur ett sagoväsen skall se ut. Använd piltangenterna E, S, D och X. För att sudda används enbart tangenterna. För att rita används tangenterna tillsammans med FCTN. Tryck BACK (FCTN 9) när du är färdig.

För att få sagoväsendet att bära den nya skepnaden skriver du

CARRY 6

När man slutligen vill lämna TI LOGO skriver man

BYE

Och LOGO svarar "AND A PLEASANT DAY TO YOU!"

Skepnader som redan finns

- 1 PLANE (flygplan)
- 2 TRUCK (lastbil)
- 3 ROCKET (raket)
- 4 BALL (boll)
- 5 BOX (kvadrat)

Du kan använda antingen numret eller namnet:

CARRY :PLANE

CARRY 1

Färger i TI LOGO

Alla färger har ett namn. Du kan ange antingen namnet eller numret. Exempelvis:

SETCOLOR :YELLOW

SETCOLOR 10

- 0 CLEAR (genomskinlig)
- 1 BLACK (svart)
- 2 GREEN (mellangrön)
- 3 LIME (ljusgrön)
- 4 BLUE (mörkblå)
- 5 SKY (ljusblå)
- 6 RED (mörkröd)
- 7 CYAN (cyan)
- 8 RUST (mellanröd)
- 9 ORANGE (ljusröd)
- 10 YELLOW (mörkgul)
- 11 LEMON (ljusgul)
- 12 OLIVE (mörkgrön)
- 13 PURPLE (magenta)
- 14 GRAY (grå)
- 15 WHITE (vit)

Några exempel

Definitionerna på nästa sida matas in på följande sätt:

Skriv TO och namnet (t ex TO KVADRAT). Tryck ENTER. Nu övergår LOGO till ett redigeringsläge.

Mata in definitionen. Om du gör fel kan du använda piltangenterna (FCTN S, D, E och X) för att flytta markören och DEL (FCTN 1) för att radera ett tecken.

Tryck BACK (FCTN 9) när du är klar.

Nu behöver du bara skrivit namnet på något av orden som du har definierat. Vissa ord måste följas av ett tal. Några exempel:

KVADRAT 50 (ritar en kvadrat med sidan 50)

PENTAGON 35 (ritar en femhörning med sidan 35)

SQ 123 (kan ej beskrivas! Måste ses! Prova med olika tal efter SQ)

START (sätter fart på en studsande boll)

STUDS (förutsätter att ett sagoväsen finns i rörelse och får det i så fall att studsa)

```
TO PENTAGON :STORLEK
TELL TURTLE
REPEAT 5 [FORWARD :STORLEK RIGHT
  72 ]
END
```

```
TO SQ :VINKEL
TELL TURTLE
CLEARSCREEN
SXY 0 29
SQUIRAL :VINKEL 5
END
```

```
TO SQUIRAL :VINKEL :LAENGD
RIGHT :VINKEL
FORWARD :LAENGD
IF :LAENGD > 100 STOP
SQUIRAL :VINKEL :LAENGD + 2
END
```

```
TO KVADRAT :STORLEK
REPEAT 4 [FORWARD :STORLEK RIGHT
  90 ]
END
```

```
TO STUDS
IF BOTH XCOR > 105 XVEL > 0 SXV
- XVEL
IF BOTH XCOR < - 105 XVEL < 0 SX
V - XVEL
IF BOTH YCOR > 72 YVEL > 0 SYV -
YVEL
IF BOTH YCOR < - 72 YVEL < 0 SYV
- YVEL
STUDS
END
```

```
TO START
TELL SPRITE 0
CARRY :BALL
SETCOLOR :RED
HOME
SETHEADING 67
SETSPED 50
STUDS
END
```

Användarföreningar

Sverige

Föreningen Programbiten
c/o Lars Hedlund
Årstavägen 27
121 68 JOHANNESHÖV

Belgien

TI Software Exchange
P. B. 63
B-2080 KAPELLEN
Kontaktperson: Thomas Coppens
Tel: 00932-(ton)-3-665 05 97
endast torsdag eftermiddagar

Holland

Software Department
J. Barnier
Rozenstraat 36
NL-8471 KM WOLVEGA

Danmark

Programklubben
c/o Hans Peder Nielsen
Vestervold 16
DK-6800 Varde

USA

International 99/4 Users' Group
7908 N.W. 23 RD Street, Suite 5
Bethany, OK 73008

International Home Computer Users' Association
P.O. Box 2403
DEL MAR, CA 92014

Storbritannien

157 Bishopsford Road, Morden, Surrey
eller direkt till ordföranden:
Paul Dicks
c/o Pershke Price Service Organisation Ltd
Dover House, 141 Morden Road
Mitcham Surrey CR4 4XB

Västtyskland

Egon Müller
Postfach 3741
2900 Oldenburg
Tel 00949-(ton)-441-75106

Luecke Elektronik
Achmühlstrasse 3
8076 Ebenhausen-Werk
Tel 00949-(ton)-8453-400

Fachverlag
Reinhold Hasse
Raabestrasse 5
5413 Bendorf
Tel 00949-(ton)-2622-2984

Rainer Bartel
Tapenbeekstrasse 61
2000 Hamburg 20

PH Color-Technik
Paul Hinz
Postfach 1265
7238 Obendorf
Tel 00949-(ton)-7423-2532

Tidskrifter

99'er Home Computer Magazine

En fristående användartidskrift har startats i USA för användare av TI-99/4A. Den utges en gång i månaden och innehåller detaljerade artiklar inom flera områden: introduktion till BASIC-programmering, tekniska frågor, programbeskrivningar, genomgång av nya tillbehör, recensioner av programvara, datorstödd undervisning, introduktion till språken som finns till TI-99/4A samt färdiga program som du själv kan mata in.

Adressen är:
99'er Home Computer Magazine
P.O. Box 5537
Eugene, OR 97405

Tidskriften kan köpas i Sverige hos återförsäljare för hemdatorn.

Nittinian

En svensk fristående användartidskrift som ges ut av Föreningen Programbiten. Än så länge har två nummer kommit ut. I dessa fanns bl a artiklar om: introduktion till BASIC-programmering och grafik, genomgång av programvara och tillbehör, presentation av programspråk, teknisk information, frågeavdelning samt programlistningar.

Adress till Föreningen Programbiten:
Föreningen Programbiten
c/o Lars Hedlund
Årstavägen 27
121 68 JOHANNESHOV

Adress till redaktionen:
Nittinian
c/o Schibler
Wahlbergsgatan 6 n b
121 46 JOHANNESHOV

En översikt över datorns utveckling

3000 f.Kr.

Ett av de äldsta beräkningshjälpmedlen är abakusen (en kulram) som uppfanns i Asien. Den kinesiska abakusen - kallad "suan-pan" - utvecklades omkring 3000 f.Kr. Den förbättrades under de följande århundrandena och den används än i dag av många människor.

1600-talet

Omkring 1640 uppfann och byggde den franske matematikern Blaise Pascal den första adderingsmaskinen med minnessiffra. (Pascal har förresten givit namn åt ett programspråk.)

I slutet av 1600-talet uppfann den tyske filosofen Gottfried Wilhelm Leibniz en maskin som multiplicerade genom snabba, upprepade additioner.

1800-talet

Den franske uppfinnaren Joseph M. Jacquard konstruerade år 1801 en vävstol som vävde tyg efter mönster. Vävstolen valde ut färgade trådar för olika delar av vävnaden enligt ett hålkort. Jacquards hålkort gjorde det möjligt att ge en maskin speciella instruktioner för en uppgift och att byta instruktioner helt enkelt genom att byta kort.

Den engelske matematikern Charles Babbage konstruerade en beräkningsapparat innehållande en modern dators alla beståndsdelar. Eftersom den enda energikällan på 1820-talet var ångmaskiner, kunde han aldrig bygga konstruktionen.

Amerikanen William S. Burroughs konstruerade 1888 den första additionsmaskinen som kunde lagra data.

Hermann Hollerith och John Shaw Billings uppfann en hålkortsmaskin avsedd för folkräkningar. Den användes i den amerikanska folkräkningen 1890.

1930-talet

Vannevar Bush, en amerikansk ingenjör, konstruerade och byggde den första analoga datorn.

Mellan 1937 och 1944 byggde Howard H. Aiken från Harvard University den första digitala datorn i samarbete med IBM. Mark I, eller den Automatiskt Sekvensstyrda Kalkylatorn, kunde utföra tre additioner i sekunden. Den var 2,4 m hög och 15,5 m lång.

1940-talet

Ingenjörerna John Mauchly och John Presper Eckert vid Pennsylvaniauniversitet byggde på 1940-talet ENIAC (Electronic Numerical Integrator and Computer). ENIAC hade 18000 radiorör, men var snabbare än tidigare maskiner.

1948 uppfann Bell Labs transistorn som till slut ersatte de utrymmeskrävande radorören.

1950-talet

UNIVAC (Universal Automatic Computer) var den första datorn som konstruerades för försäljning. 1951 installerades den på folkräkningsverket i USA.

1954 konstruerade Texas Instruments den första kommersiellt framställda transistorradion. Som resultat av projekt blev TI det första företaget som massstillverkade billiga germaniumtransistorer.

1958 konstruerade Jack S. Kilby på TI den första integrerade kretsen (IC): en enda kiselskiva med kompletta elektroniska kretsar.

1967

1967 framställde TI den första miniräknaren.

1970

Mikroprocessorn i ett enda chip som innehåller hela centralenheten uppfanns 1970 av Gary Boone på TI. Idag används mikroprocessorer som "hjärnor" i en mängd elektroniska produkter, som t ex i TI-99/4A.

1976

1976 utvecklade TI tekniken att "omprogrammera" räknare, utbildningsmaskiner, hemdatorer genom att sätta i en ny programmodul.

1978

1978 introducerade TI Speak & Spell, en apparat som utnyttjade TI:s uppfinning av syntetiskt tal på ett enda chip.

1981

1981 introducerades TI LOGO. LOGO var det första programspråket som låter barn skapa en lärorik omgivning. I TI LOGO lär barnen TI-99/4A att dra streck och skapa rörlig grafik i många färger.

Datorordlista

ASCII - Amerikansk standardkod för informationsutbyte. Den kod som används i många datorer för att representera bokstäver, siffror och specialtecken.

Assembler - Ett språk som ligger mycket nära datorns "eget" språk.

BASIC - "Beginners All-purpose Symbolic Instruction Code". Ett vanligt programmeringspråk som är lätt att lära sig. Det används i de flesta persondatorer.

Baud - Enhet för överföringshastighet mellan t ex dator och skrivare. Motsvarar vanligen antal bitar per sekund.

Binär - ett talsystem som bara har två siffror, 0 och 1. Används i datorn eftersom en ledning kan antingen leda ström (1) eller inte leda ström (0).

Bit - binär siffra. Kan anta värdena 0 och 1.

Brytpunkt - en rad i ett program angiven med BREAK-kommandot där exekveringen stoppas. Efter en brytpunkt kan man skriva ut variabler m m i kommandoläget. Programkörningen kan sedan fortsättas med CONTINUE, om programmet ej har ändrats.

Buffert - En del av minnet avsett för tillfällig lagring av inlästa poster eller poster som skall skrivas ut.

Bug - Datorslang för fel i program eller hårdvara.

Byte - Åtta bitar (binära siffror) som behandlas som en enhet, ofta representerande ett tecken. Datorns minneskapacitet uttrycks ofta i antalet bytes.

Centralenhet - datorns "hjärna": den del av datorn som utför instruktioner i ett program, gör beräkningar m m.

Data - information.

Datorstödd undervisning - användning av datorn för undervisning i olika ämnen.

Default - se förvalt standardvärde.

Diskett - se flexskiva.

Editor - ett speciellt program som används för att mata in och redigera program. Editorer som är speciellt lämpliga för textbehandling brukar kallas ordbehandlare. Det här häftet är skrivet med hjälp av ordbehandlaren TI Writer.

Fil - en samling data av samma typ på t ex diskett eller kassett.

Flexskiva - en böjbar plastskiva med ett magnetiskt skikt på. Används för snabbare datalagring än kassetband.

Hertz (Hz) - enhet för frekvens. 1 Hz = 1 svängning per sekund.

Fält - en samling av numeriska eller strängvariabler ordnade i en lista eller matris för att datorn ska kunna hantera dem enkelt.

Hänvisning till ett element i fältet görs med ett index som anger elementets placering i fältet.

Förvalt standardvärde - standardegenskaper eller standardvärden som datorn förutsätter om vissa specifikationer utelämnas i en instruktion eller ett program.

Hexadecimal - ett talsystem med basen 16 som alltså har 16 siffror: 0-9 och A-F. Det används ofta som ett förkortat skrivsätt för binära tal. Fyra binära siffror motsvarar en hexadecimal siffra så 1010 motsvarar A, 1011 motsvarar B och 10101011 motsvarar AB. Hexadecimala tal används för definition av grafiska tecken i CALL CHAR-instruktionen.

Hårdvara - kretsarna i datorn med tillbehör, i motsats till program.

Index - ett numeriskt uttryck som anger ett visst element i ett fält. I TI BASIC skrivs indexet inom parenteser omedelbart efter fältnamnet.

Inmatning - att överföra data till minnet.

Interpretator - programmet i datorn som översätter de BASIC-satser du matar in till maskinspråk.

I/O - inmatning/utmatning.

Kommando - en instruktion som datorn utför omdelbart. Kommandon är inte en del av ett program och matas därför in utan radnummer.

Kommandoläge - när inte något program körs är datorn i kommandoläge och utför då varje uppgift när den matas in.

Loop - se slinga.

Markör - en symbol som anger var nästa tecken kommer att visa sig på skärmen när du trycker ned en tangent.

Masslagringsenhet - ett tillbehör, t ex kassettbandspelare eller skivminne som lagrar program eller data för framtida användning.

Maskinspråk - det språk som datorn direkt förstår. Det består av grupper av binära tal.

Minne - se RAM, ROM och masslagringsenhet.

Mjukvara - program. Här ingår program som är inbyggda i datorn, modulprogram och program inmatade av användaren.

Post - en samling sammanhörande data, som t ex löneutbetalningar till en person eller en students skrivningspoäng. En grupp av liknande poster, t ex alla poster i ett företags avlöningslista, kallas fil.

Program - en samling instruktioner vilka talar om för datorn hur den skall utföra en uppgift.

RAM - "random access memory", skriv/läs minne. Det minne som program och data lagras i under körning. Data lagrade i RAM raderas när datorn slås av eller när man lämnar BASIC.

ROM - "read only memory", läsminne. Vissa av datorns instruktioner lagras permanent i ROM och kan utnyttjas men inte ändras. ROM raderas inte när datorn stängs av.

Slinga - en grupp på varandra följande programrader vilka utförs flera gånger, ofta ett bestämt antal gånger.

Sträng - en följd av tecken behandlade som en enhet.

Tecken - bokstav, siffra, skiljetecken eller speciella grafiska symboler.

Tolk - se interpretator.

Variabel - ett namn som ges till ett värde vilket kan variera under programkörningen. Du kan föreställa dig en variabel som en plats i minnet där värden kan ersättas med nya värden under programkörning.



**TEXAS
INSTRUMENTS**