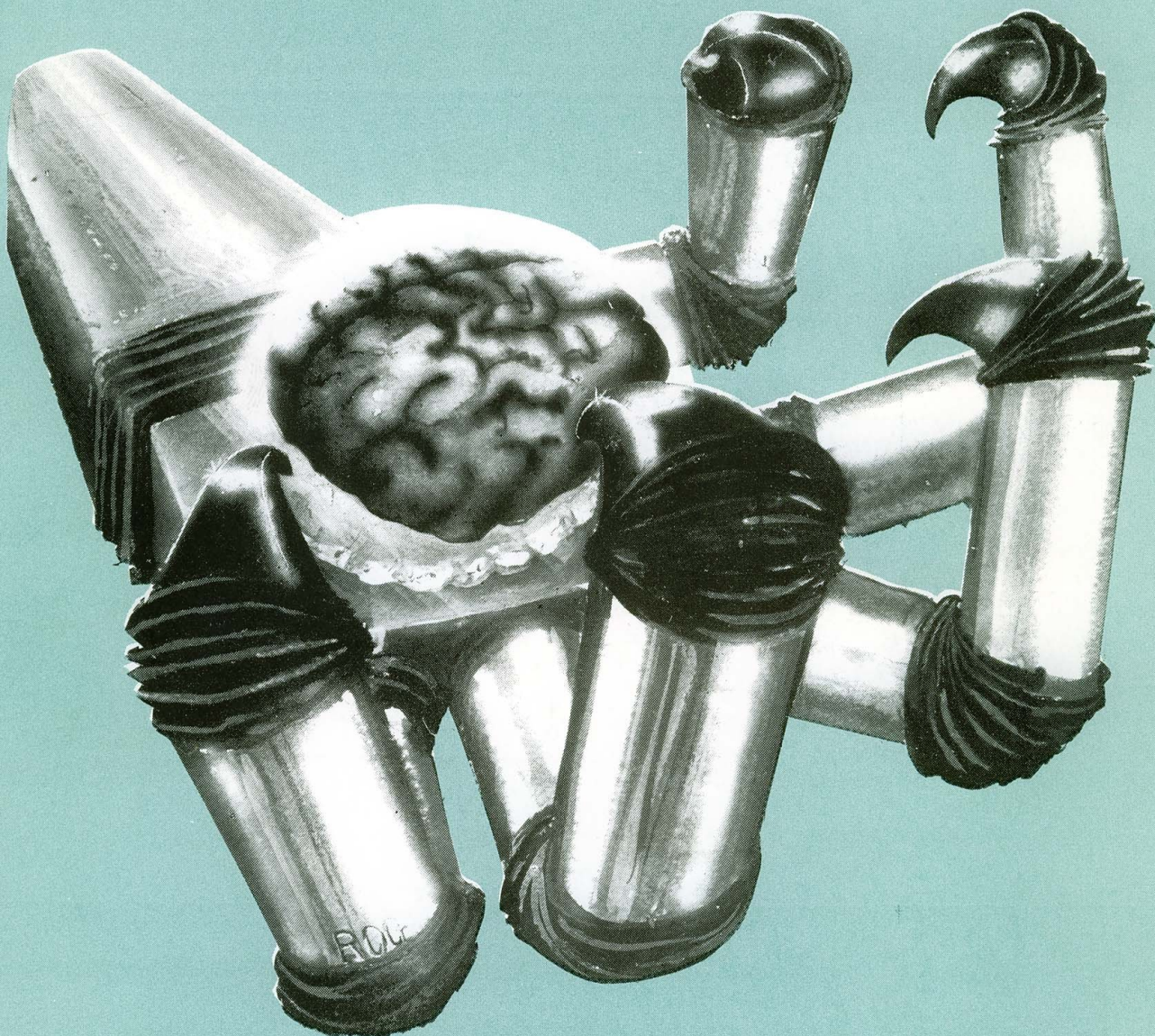


#nittinian



Innehåll nr 1

Datorernas krig	4	Köpt vaddå?!	11
Sprites i TI basic	6	Minimemory	12
Nya produkter	7	Tekniktips	13
Engelsk-svensk		Murphy's law	
ordlista	8	och TI-99	14
Earth attack	10	Översättning	15

Nu är den här, kompaktdatorn från Texas Instruments!

CC-40

För nybörjare, specialister,
studerande, forskare, affärs-
män . . .

CC-40 är den första modellen i
en ny familj av kompaktdato-
rer från Texas Instruments.

Programmeras i en utökad
version av BASIC. Färdiga
program finns i "Solid State
Software"-moduler och i små
kassettband, "Wafertape".

Hela systemet är batteridrivet
och ryms ledigt i en portfölj.

Upp till 200 timmars drift på
4 st alkaliska batterier. 6 K
RAM-minne och 34 K inbyggt
ROM-minne i grundutfö-
randet.



**TEXAS
INSTRUMENTS**

Skrivare/Plotter HX-1000

X-Y grafik i 4 färger. För tex
cirkeldiagram, histogram och
regressionskurvor. Skriver ut
programlistor och resultat.
Upp till 36 tecken per rad.
Skriver på 57 mm vanligt
papper.

Wafertape HX-2000

Digital mikrokassettenhet som
ger 48 K lagringsutrymme i
varje kassett. Snabb tillgång
till program och data (8000
baud). Styrs helt från datorn.

RS-232 parallell- interface HX-3000

För anslutning av CC-40 till

matrisskrivare, modem eller
annan dator.

Programvara

18 programpaket finns redan
klara. 4 "Solid State Software"-
moduler (upp till 128 K) och
14 Wafertapekassetter.
Ledtexter på engelska, tyska
eller franska.

Föreningen 99:an
c/o A Brolin
S:t Göransgatan 153/801
S-112 51 Stockholm
Sverige

BÄSTE 99-FANTAST!

Du har nu i din hand det första numret av den efterlängtnade användartidningen till Texas Instruments Hemdator TI-99/4A.

99'an är medlemsbladet för föreningen 99'an, som är en dotterförening till föreningen Programbiten. Från Programbiten kommer två av medarbetarna i redaktionen: Björn Gustavsson och Anders Brolin. Björn har tidigare "vänt ut och in" på de programmerbara Texas-räknarna TI-58 och TI-59. Numera sysselsätter han sig enbart med datorer. Anders har bl. a. översatt delar av manualen till TI-99/4A och svarat på telefonförfrågningar om räknare och hemdatorer. Den tredje medarbetaren är Roger Everett, som står för det mesta av det konstnärliga arbetet. Roger har jobbat mycket med tidningsproduktion och grafisk produktion tidigare.

Det är redaktionens förhoppning att du kommer att ha stor glädje av den här tidningen. Det första numret består nästan uteslutande av material som vi i redaktionen har tagit fram. Några av bidragen har dock skickats in av andra. Vi är mycket tack-samma för detta och kommer så snart som möjligt att honorera de införda bidragen. I framtiden hoppas vi att även andra medlemmar i föreningen 99'an kommer att bidra med artiklar och tips om hur man kan använda sin TI-99/4A på bästa sätt. Vi vill med andra ord att 99'an ska bli en tidning för medlemmarna av medlemmarna.

TREVLIG LÄSNING!!

NITTINIAN

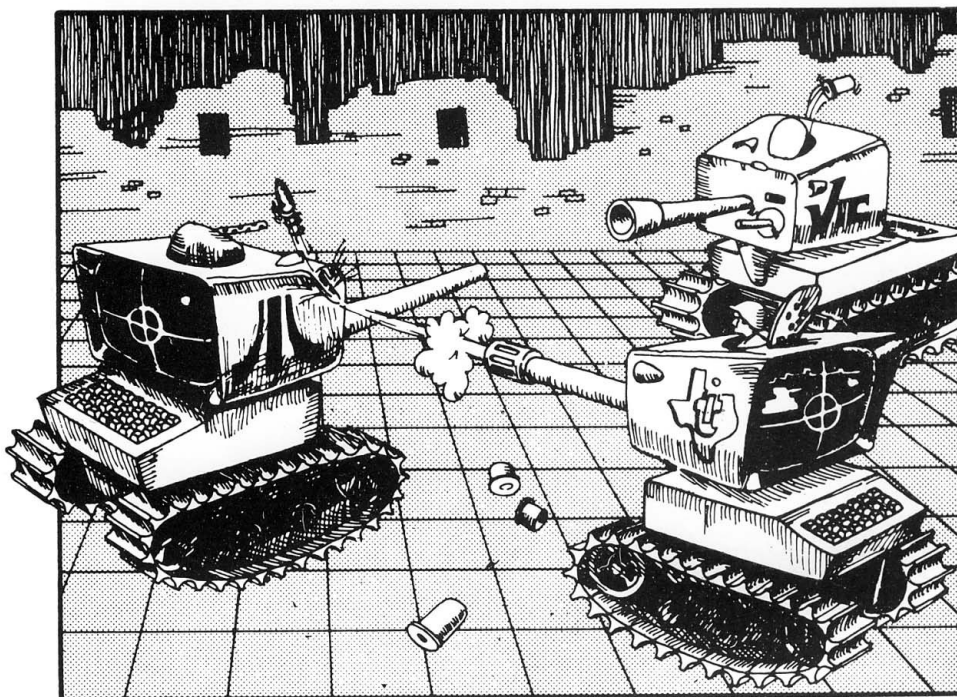
Medlemsblad för "Föreningen 99'an". Lösnummerpris 20:—, medlemsavgift 1983 100:—. Ansvarig utgivare: Anders Brolin. För insänt ej beställt material ansvaras ej.

För KOMMERSIELLT bruk gäller följande:

Mångfaldigandet av innehållet i denna skrift, helt eller delvis, är enligt lag om upphovsrätt av den 30 december 1960 förbjudet utan medgivande av FÖRENINGEN 99'an. Förbudet gäller varje form av mångfaldigande genom tryckning, duplicering, stencilering, bandinspelning, diskettinspelning etc.

Redaktionens adress:
Nittinian
Krysshämmarvägen 10
171 57 SOLNA

Tel: 08-27 96 30



"Datorernas krig"-spelet för 80-talet !

Vad kan vara mer spännande än Munch Man, mer dödligt än Space Invaders, och har fler överraskningar än Adventure eller Tunnels of Doom? Jo, "Datorernas Krig" – det mest avancerade spelet av dem alla.

Detta nya spel, med höga insatser, finns ännu inte tillgängligt på kassett, diskett eller programmodul. Men av den prototyp som vi haft tillfälle att undersöka, framgår med all önskvärd tydlighet att detta spel har framtiden för sig.

Datorernas Krig har alla finesser som ska finnas med i ett "riktigt" datorspel. Det har superb grafik, kan spelas direkt från tangentbordet eller med joysticks, och har bara en svårighetsgrad. Tyvärr är bruksanvisningen mycket dåligt skriven, något som gäller väldigt många datorspel. Däremot är reglerna helt fantastiska. Du kan luras, ljuga, bryta löften, inte leverera produkter eller lägga krokben för din bästa vän, allt utan att förlora en enda poäng. Bonuspoäng utdelas om du förannonserar dina produkter ett halvår i förväg,

om du har en usel servicegrad mot köparen, och om du hela tiden byter ståndpunkt.

I version 1.0 av Datorernas Krig deltar tre spelare. Alla tre är stora amerikanska bolag och ett av målen med spelet är att låta det förbli så eller att eliminera de andra två deltagarna. Extra poäng ges till den spelare som kan köpa in de flesta utländska komponenterna utan att få tillverkaren att själv komma med i spelet.

Spelet börjar med att varje deltagare väljer ut spelpjäser från en "Job Pool". Varje spelare kan ha hur många elektronikingenjörer som helst – de är vana att arbeta gratis. Därefter väljer man ut en reklambyrå som är van att övertyga 9 filmstjärnor av 10. Till slut väljer man sin försäljningschef, helst ung men med begynnande flint. Nästa anhalt är banken. Varje spelare kan ta flera miljoner från banken, och beroende på framgången kan han alltid komma tillbaka för att hämta mer.

KONTAKTER ÖVER KÖLEN

Från Norge har Herbert Jansen hört av sig. Herbert är "hobbyprogrammerare" enligt vad han själv skriver. Hans utrustning består av TI-99/4A, kassettbandspelare och Extended BASIC. Ursprungligen kommer han från Holland och är därför medlem i den Holländska TI99-klubben. Från den holländska användartidningen "Tijdsningen" har han saxat följande lilla programsnutt som visar ett exempel på "kapslade" slingor (två FOR-TO-NEXT-slingor i varandra).

```

50 REM TI-BASIC PROGRAM          190 FOR J=1 TO 12
    UR TIJDNINGEN,                200 A=A-2
    HOLLAND                       210 FOR Z=J TO 25-J
60 REM AV JAN BARNIER            220 CALL HCHAR(Z,4+J,40+(J-1)*8,A)
    NOVEMBER 1982                230 NEXT Z
100 CALL CLEAR                   240 NEXT J
110 CALL SCREEN(5)              250 C=INT(15*RND)+2
120 PRINT "          VAENTA EN 260 N=30*C+110
STUND"::::::::::                270 CALL SOUND(500,N,1,2*N,3)
130 RANDOMIZE                   280 CALL COLOR(2,1,C)
140 A=26                        290 FOR J=2 TO 16
150 FOR J=5 TO 17               300 CALL COLOR(J,1,C)
160 CALL CHAR(J*8,"")           310 NEXT J
170 NEXT J                      320 GOTO 250
180 CALL CLEAR

```


Nu raderas bildskärmen och den slutgiltiga målsättningen med spelet avslöjas . . .

Sno åt dig så mycket som möjligt ur Svenssons plånbok, utan att ta bort leendet på hans läppar!

Nu börjar fas 1 av spelet. Varje deltagare får reda på att han ska ta fram och bygga en hemdator. Enligt reglerna får ingen av dessa datorer vara kompatibla med någon annan dator. Dessutom måste de alla ha helt olika tangentbord. Utöver detta måste varje tangentbord ha minst två tangentfunktioner som driver användaren helt vansinnig. Konstruktionsändringar kan göras när som helst under spelets gång och bonuspoäng ges till den spelare som får sin hårdvara och mjukvara att framstå som uråldrig inom de två första produktionsåren. Naturligtvis kommer inga produkter tas emot som inbyte när de väl har blivit gammalmodiga och köparen vill byta upp sig.

Spelet har nu fortskridit så långt att alla deltagarna har färdigställt sina första produkter och avslutat den första testförsäljningen. Enligt vad vi kan se har alla redan hunnit göra ett par misstag. Spelare nummer 1 har totat ihop sitt system så att grundenhet och monitor kostar över 10.000 kr. Alldeles för mycket för Svenssons plånbok. Spelare nummer 2 har fått fram ett knasigt tangentbord som inte har rörliga tangenter. Spelare nummer 3 ligger också för högt i pris med sin första version, men planerar ett släppa en billigare modell, och håller hårt i sin annonsbudget till denna nyare modell finns färdig.

Allt eftersom spelet fortsätter märker deltagarna suget från marknaden. Folk i utbildningssystemet faller pladask för de nya produkterna, och en helt ny industri växer upp. Spelare nummer 1 släpper nu ett helt nytt tangentbord och sänker dessutom priset på sin maskin. Försäljningen bara ökar och hans enhet för konstgjort tal är den bästa på marknaden. Spelare nummer 2 har de bästa rymdspelen på marknaden och får massvis med draghjälp från inhyrda programskrivare. Spelare nummer 3 kommer nu starkt i försäljningen med sin nya modell och ser ut att kunna hota ledaren.

Spelare nummer 3 gör nu ett drag som placerar honom i ledningen. Det ser ut som om han har hyrt in en av piloterna från rymd färjan för att sköta marknadsföringen. Han har redan fått flera bonuspoäng för felaktig och missvisande reklam. Hans försäljningsavdelning har arbetat övertid med att charma in sig hos landets största återförsäljningskedja, en taktik som verkligen har lyckats. Spelare nummer 2 svarar med att lägga beslag på rättigheterna till århundradets största spelsuccé . . . Pac man. För att inte förlora allt, stoppar spelare nummer 1 in sin Video Graphics-modul och tar fram en ny Espansionsbox som ska ersätta det berg av kringutrustningsenheter som har växt ut till en meters längd till höger om tangentbordsenheten. (Extra poäng för att han annonserade om den i förväg.)

Flera nya bilder rullar förbi på skärmen där spelarna visar upp nya hårdvaruprodukter och ny programvara. Alla tre spelarna har nu så små marginaler på produkterna att återförsäljarna måste kasta produkterna på köparna. För att få så stor marknadsandel som möjligt har spelarna nu börjat massidistribuera sina produkter via butikskedjor och postorderföretag som inte vet ett smack om datorer. Varje spelare får bonuspoäng för total avsaknad av service till köparna.

Det är nu spelare nummer 1 står på tur. Han trycker ned några CTRL-tangenter och skärmen blinkar till med sina 16 färger. Vad är det som händer, frågar jag mig. Har systemet hängt sig? Använder han en gammal version av Extended BASIC? Så plötsligt . . . skärmen fylls med stora bokstäver . . . KRIG! Från enheten för konstgjort tal hörs en röst säga: "Om några veckor kommer vi att sänka priset på vårt datorsystems grundenhet!". (Självfallet får han bonuspoäng för att tala om det i förväg.) Nu pågår kriget för fullt. Det verkar som om spelare nummer 1 har övertygat de flesta av sina återförsäljare att om de förlorar på att sälja grundenheten, så kommer de att tjäna en förmögenhet på att sälja programmoduler. Allt detta ingår i ett nytt återförsäljaravtal. Eftersom spelare nummer 1 dessutom har ensamrätten på tillverkningen av dessa moduler, har han också kännedom om hemliga detaljer i modulerna, som han aldrig kommer att dela med sig av till återförsäljare eller konsumenter.

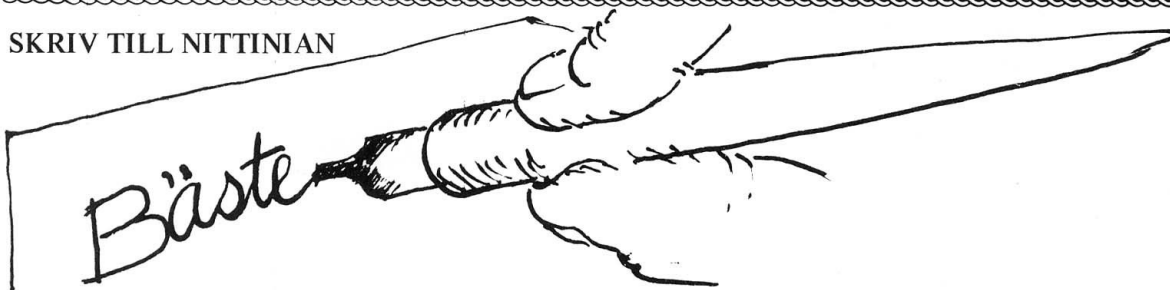
Spelare nummer 2 och 3 är totalt förvirrade! Vad sjutton ska vi ta oss till? Spelare nummer 3 gör det första draget in i KRIGSZONEN. Övertygad om att han har en större marginal på sin produkt än spelare nummer 1, och att herr och fru Svensson inte begriper skillnaden mellan 5K och 16K, sänker också han priset. Spelare nummer 2 bestämmer sig därefter för att han har större marginal på mjukvaran än på hårdvaran, så han erbjuder gratis programvara för spel och hushållsapplikationer med varje köpt enhet. Spelare nummer 2 har under tiden fått stora problem med tillverkningen av andra produkter och skär nu ned på personalen.

Med spelstrategin upplagd och alla pjäserna på rätt ställe är det nu dags för att få full kontroll över det interna RAM-minnet. De har gjort allt som stått i deras makt till nu, och med Kapten Kirk och Mr Spock vid spakarna, försäljare från Kiviks marknad och Ronald McDonald full med nya ideer och fickorna fulla med pengar, är det hög tid att gräva ned sig i skyttegravarna inför vinterkriget.

Vem som ska vinna slaget i konsumenternas fickor står fortfarande skrivet i stjärnorna. Det ser i alla fall ut att bli en givande kamp (mest för deltagarna).

Rekommenderat försäljningspris för "Datorernas Krig" ligger runt en halv miljard kronor.

SKRIV TILL NITTINIAN



Det här är en tidning för hemdatoranvändare gjord av hemdatoranvändare. Det betyder att vi vill att du som läser också ska skriva artiklar till tidningen. Skriv om allt hellan himmel och jord som rör hemdatorer, men skriv helst om sådant som du tror kan vara av intresse för andra medlemmar. Dvs skriv om sådant du själv vill läsa.

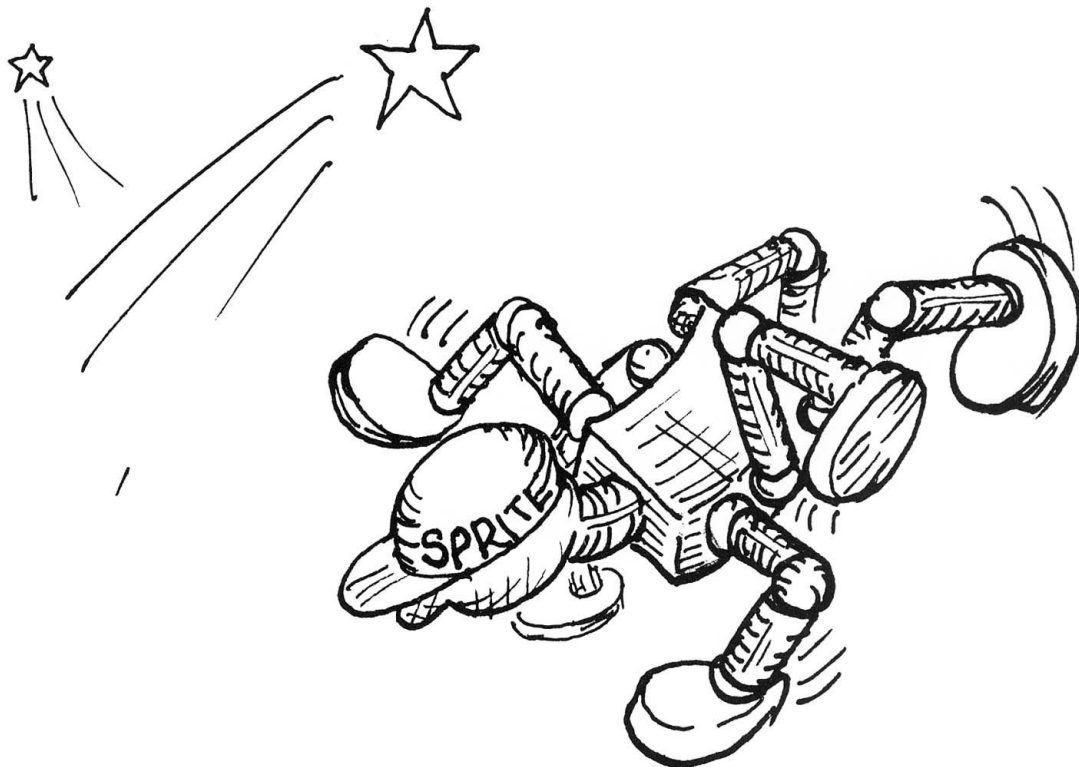
Försök att strukturera ditt material så att vi inte behöver skriva om ditt bidrag. Om du skickar in ett program eller en subrutin, måste du bifoga en kassett eller en diskett (som vi självfallet returnerar till dig). Använd en tom kassett och spela in ditt program i början på den.

För att vi ska kunna publicera ditt bidrag måste vi dessutom få en tydlig programbeskrivning och en ordentlig användarinstruktion. Det räcker t. ex. inte med att skriva "Programmet startas med RUN".

Skriv tydligt och helst på maskin. Tänk på att använda dubbelt radavstånd och lämna ordentliga marginaler.

Välkommen med ditt bidrag!

Att skapa sprites i TI basic



Peter Olin har undersökt hur man kan skapa sprites i TI BASIC med hjälp av Mini Memory-modulen och vad han har kommit fram till kan du läsa i artikeln nedan. Men det krävs mer undersökning av den här metoden. Varför kan tex bara 4 sprites skapas samtidigt? Om du kommer på något vill vi gärna att du skriver några rader om det till oss.

—Redaktören.

I International 99/4 User's Group's Newsletter (nov 1982) fanns följande BASIC-program:

```
100 CALL CLEAR
110 CALL POKEV(768,98,129,161,1,
208)
120 CALL POKEV(1920,50,50)
130 CALL LOAD(-31878,1)
140 GOTO 140
```

För att det skall fungera måste du ha Mini Memory-modulen inpluggad. Programmet sätter igång en sprite som åker snett över skärmen.

Så här fungerar det:

Rad 110: 768 är första adressen i spritebeskrivningstabellen. Nästa spritebeskrivning börjar på adress 772, och följande på 776 etc.

98 är dot-raden som spriten skall börja på och 128 är dot-kolumnen.

161 är ASCII-koden för spritens tecken + 96, dvs tecknet är A (ASCII-kod 65).

1 är spritens färgkod-1.

208 markerar slutet på tabellen.

Om du vill definiera ytterligare sprites måste du skriva 208 allra sist i sista spritebeskrivningen.

Exempel 1:

```
100 CALL POKEV(768,98,128,161,1)
110 CALL POKEV(772,35,200,162,10
208)
```

Detta definierar två sprites: Ett svart A som startar på (98,128) och ett gult B som startar på (35,200).

Rad 120: 1920 är första adressen i spritehastighetstabellen. Värdena 50,50 är hastigheterna i Y-led respektive X-led. Nästa sprites hastighet beskrivs på adress 1924, och den efterföljande på adress 1928 etc. Följande exempel sätter fart på de två sprites som beskrivs i exempel 1:

de två sprites som beskrivs i exempel 1:

```
120 CALL POKEV(1920,50,50)
130 CALL POKEV(1924,-30,120)
```

Rad 130: -31878 är en adress i CPU-RAM och 1 är antalet sprites som finns beskrivna. Det här exemplet startar de två sprites som beskrivits i de tidigare exemplen och håller dem igång under hela tiden som programmet körs. Exempel 3:

```
140 CALL LOAD(-31878,2)
150 GOTO 150
```

Programmet stoppas med FCTN CLEAR. För att se dessa sprites bättre kan man lägga till

```
80 CALL CLEAR
90 CALL SCREEN(16)
```

För att få bort underligheterna som bildas på skärmen skriv bara:

```
CALL LOAD(-31878,0)
```

Tyvärr har jag inte lyckats få fram fler än 4 sprites på det här sättet och jag är tacksam om jag får uppslag av er.

NYA PRODUKTER

CC-40

av Björn Gustavsson

Texas Instruments har nyligen presenterat en s k "Compact Computer" – CC-40. Den är batteridrivnen och kan tas med, tillsammans med all kringutrustning, i en portfölj. Men den innehåller en BASIC jämförbar med 99:ans Extended BASIC.

Utseendemässigt liknar den de räknedosor programmerbara i BASIC som kommit tidigare. CC-40 är dock något större och tangentbordet är därför något lätthanterligare. (Formatet är 23 x 15 x 2,5 cm.) Displayen som är av LDC-typ kan visa 31 tecken. Dock kan rader upp till 80 tecken inmatas och visas genom att rulla texten åt vänster eller höger.

Tangenterna sitter som på en skrivmaskin. Det finns också ett separat numeriskt tangentbord till höger om det vanliga.

CC-40 är så strömsnål att 4 små ficklampsbatterier kan driva den i ca 200 timmar. (Det går också att köpa en batterieliminatör.) Dessutom bevaras programmet i minnet medan den är avstängd. Strömsnålheten har dock sina nackdelar: Den är inte lika snabb som en "vanlig" dator.

Från början innehåller CC-40 34 K ROM för BASIC-tolken och 6 K RAM för egna program. Minnet kan byggas ut till maximalt 34 K RAM. Man kan ansluta en modul i taget och på så sätt få ända upp till 128 K ROM.

CC-40 kan programmeras i BASIC och assembler. BASICen är mycket avancerad jämförd med övriga BASIC-datorer i fickformat (från Sharp och CASIO bl a). Den har det mesta som Extended BASIC har, förutom grafiken förstås. Dessutom kan man svara med ett uttryck i en INPUT-sats, vilket är lika användbart som ovanligt.

Om man tycker att BASIC är för långsamt kan man använda TMS 7000 assembler. Assembler finns i en modul. Assemblerade assemblerprogram kan laddas med POKE direkt från BASIC.

För närvarande finns tre kringutrustningsenheter:

Färgskrivare/plotter: Plottar eller skriver i röd, blå, grön eller

svart färg. Man kan välja mellan 10 olika textstorlekar. Den skriver med upp till 11 tecken i sekunden.

RS-232-interface: Används när man vill koppla CC-40 till t ex en vanlig 80-kolumners datorskrivare eller till en dator (t ex 99:an). Med hjälp av ett modem kan man också skicka program mellan två datorer över telefonnätet.

"Wafertape". En specialbandspelare som använder kassettband i en oändlig slinga. Den styrs helt och hållet från datorn vilket eliminerar allt manuellt letande efter program på bandet. Ange endast programets namn och CC-40 letar rätt på det. Överföringshastigheten är också högre än med en kassettbandspelare. Det tar ca 10 sekunder att ladda 4 K. En wafertape-kassett rymmer maximalt 48 K.

Streckkodsläsare och TV-interface håller på att tas fram.



TI-99/2

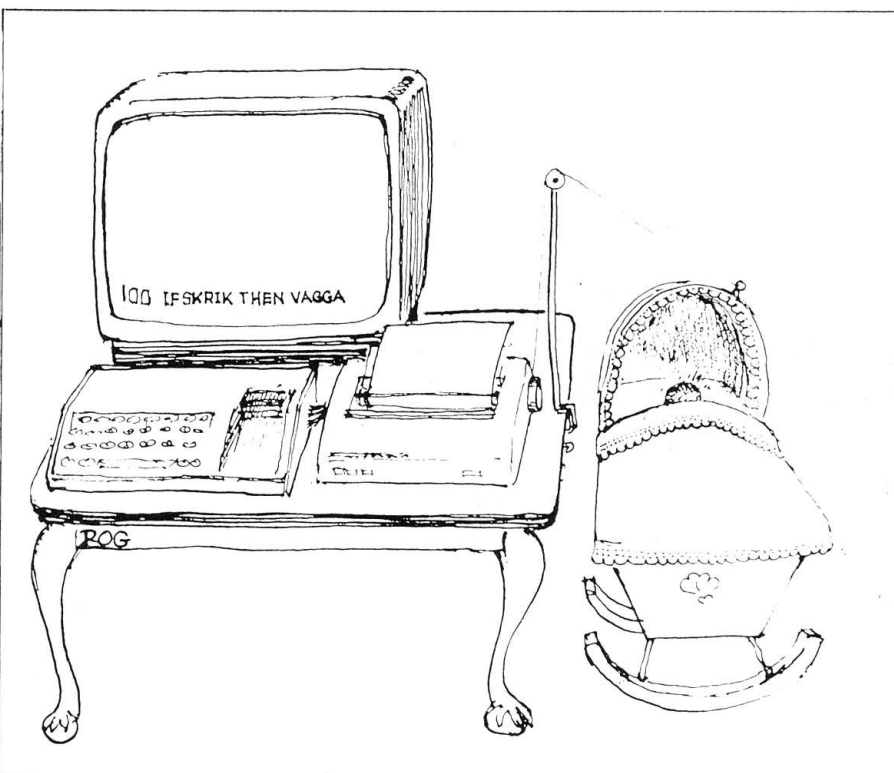
av Björn Gustavsson

I USA finns lillebror till TI-99/4A, TI-99/2, som beräknas komma till Sverige hösten 1983. Priset i USA är 99 dollar, vilket är lika mycket som ZX-81 kostar. (ZX-81 kostar i Sverige 995:– monterad och 895:– i byggsats.) 99/2 har liksom 99/4A ett riktigt skrivmaskinstangentbord och ansluts till en vanlig TV.

Den innehåller i grundutförande 4,2 K RAM, vilket kan byggas ut till totalt 36,2 K. 99/2 har samma BASIC som 99/4A, vilket innebär att dess program kan köras på 99/4A. Däremot är det lite svårare att göra tvärtom. 99/2 saknar bl a 99/4A:s färggrafik.

Det går att ansluta programmoduler även till 99/2. En av modulerna innehåller en "BASIC-skola" som lär nybörjaren BASIC.

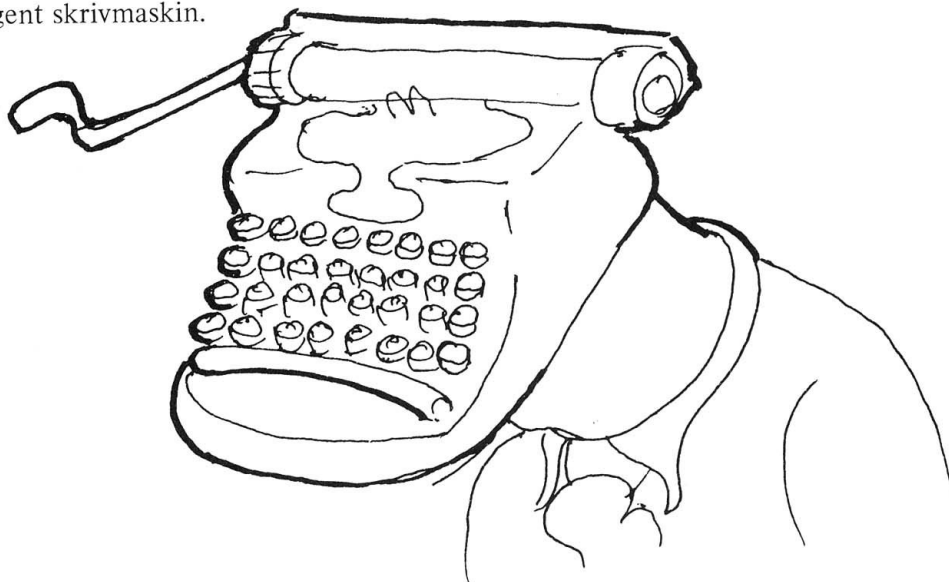
99/2 kan använda samma tillbehör som CC-40 (se separat artikel). För närvarande finns färgskrivare/plotter, wafertape och RS-232 interface.



ENGELSK-SVENSK DATAORDLISTA

av Björn Gustavsson

ALGORITHM	Algoritm. En algoritm beskriver steg för steg en lösning till ett givet problem. Den måste slutföras i ett ändligt antal steg. Algoritmen kan uttryckas i vilket språk som helst. Ett exempel på en enkel algoritm är följande: 1. Sätt i nyckeln i nyckelhålet. 2. Vrid nyckeln ett kvarts varv åt höger. 3. Fatta dörrhandtaget. 4. Drag i dörrhandtaget tills dörren är öppen.
ALPHANUMERIC	Alfanumerisk. Bokstäver, siffror och specialtecken.
ANALOG	Analog. Enkelt uttryckt, något som avläses med en visare. Motsatsen är digital. (En klocka med visare visar tiden analogt, medan ett digitalur visar tiden digitalt.)
ASCII	American Standard Code for Information Interchange. Den kod som används i 99:an för att representera alfanumeriska tecken. Bokstaven A representeras t ex av talet 65.
ASSEMBLY LANGUAGE	Assembler, svensk standard assembleringsspråk. Symboliskt maskinspråk. Ett språk som ligger mycket nära datorns maskinspråk. Se MACHINE LANGUAGE.
ASSEMBLER	Assembler. Ett program som översätter program i assembler till maskinspråk. Se ASSEMBLY LANGUAGE, MACHINE LANGUAGE.
BASIC	Det vanligaste högnivåspråket på mikrodatorer. Dess stora fördel är att det är lätt att lära sig.
BAUD	Enhet i vilken man mäter överföringshastighet mellan t ex en dator och en skrivare. Motsvarar vanligen (men inte alltid) bit per sekund.
BCD	Binary Coded Decimal = binärkodad decimal notation.
BENCHMARK	Ett testprogram för att testa en dators kapacitet, t ex dess snabbhet.
BINARY	Binär.
BIT	Bit. Binär siffra. Den minsta logiska enheten. Kan endast anta värdena 0 eller 1.
BUG	Fel.
BUS	En förbindelse mellan olika delar i ett datorsystem.
BYTE	En byte = 8 bits. En dators minnesstorlek mäts i Kbyte = 1024 bytes. Se WORD.
COMPILER	Kompilator. Ett komplicerat program som översätter ett högnivåspråk till maskinspråk. Det finns till 99:an en Pascal-kompilator. Se HIGH LEVEL LANGUAGE.
COMPATIBLE	Kompatibel. Två datorer sägs vara kompatibla om man kan köra samma program på dem utan större ändringar.
CRT	TV eller monitor.
CURSOR	Markör. Pekar ut var på skärmen nästa tecken kommer att skrivas. Är i BASIC en fylld ruta.
D/A CONVERTER	En enhet som gör om digitala signaler till analoga. Motsatsen är A/D CONVERTER.
DEBUGGING	Debugging, svensk standard "inkörning av program". När man upptäcker, spårar upp och rättar till fel i ett program. Se BUG.
DIGITAL	Digital. Se ANALOG.
DOS	Disk Operating System. Det som styr diskettenheterna till 99:an.
EDITOR	Ett program för att skriva och redigera text. Används bl a tillsammans med en assembler för att mata in och ändra assemblerprogram. En texteditor används som en intelligent skrivmaskin.



EPROM	Ett ROM som användaren kan programmera och sedan radera med ultraviolett ljus.
EXTERNAL MEMORY	Minne utanför datorn.
FILE	Fil. En mängd data lagrad på kassettband eller diskett.
FIRMWARE	Program lagrat i ROM, PROM eller EPROM.
HARD COPY	Utskrift på papper.
HARDWARE	Hårdvara, maskinvara. Själva datorn utan program.
HEXADECIMAL	Hexadecimal. Tal med basen 16. I detta talsystem används våra vanliga siffror plus bokstäverna A–F: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
HIGH LEVEL LANGUAGE	Högnivåspråk. Ett språk som mer än assemblerspråk liknar mänskliga språk och därför är lättare att använda. Ett högnivåspråk kräver en INTERPRETER eller en COMPILER. Exempel: BASIC, Pascal.
I/O	Inmatning till yttre enheter.
INTERFACE	Interface, gränssnitt. T ex en ihopkoppling mellan skrivare och dator.
INTERNAL STORE	Primärminne.
INTERPRETER	Tolk, interpretator. En interpretator utför en instruktion i taget i ett program (t ex BASIC-program).
MASS STORAGE	Massminne. Kan lagra väldigt stora datamängder. Exempel: Disketter, kassettband.



MACHINE LANGUAGE	Maskinspråk. Ett språk som en dator direkt kan "förstå" och utföra. Består av "ettor och nollor". Maskinspråk är olika på olika datorer. 99:an använder maskinspråket till mikroprocessorn TMS9900.
MODEM	Gör det möjligt att per telefon överföra program eller data mellan två datorer.
OBJECT PROGRAM	Objektprogram. Det program som en assembler eller kompilator producerar från källprogrammet. Se SOURCE PROGRAM.
OPERATING SYSTEM	Operativsystem. Det program som styr datorn.
PASCAL	Ett högnivåspråk som kan köpas till 99:an.
PERIPHERALS	Yttre enheter, t ex skrivare, diskettenheter, kassettbandspelare.
PROM	Programmerbar ROM.
RAM	Minne som man kan läsa och skriva i.
ROM	Minne som man endast kan läsa. Förprogrammeras på fabriken.
RS232	Interface för att koppla t ex skrivare till 99:an.
SOFTWARE	Mjukvara, programvara.
SOURCE PROGRAM	Källprogram. Det program som användaren matar in. En kompilator eller assembler översätter sedan detta till ett objektprogram.
WORD	Ord. 1 ord = 2 bytes på 99:an.



Från Håkan Göransson i Halmstad har vi fått in ett mycket underhållande spelprogram: Earth Attack – spelprogram för TI-99/4A. Programmet är hämtat från den amerikanska tidningen PCT och är från början konstruerat av Peter Clisby, USA, men Håkan har lagt till att man kan välja svårighetsgrad och har dessutom ändrat litet på tecknen.

Spelet går till så här:

När du startar spelet frågar datorn efter svårighetsgraden, som du anger mellan 1 och 2.

När spelet sedan börjar, kommer du att se ett flertal "korridorer" uppenbara sig till vänster och en vertikal linje till höger. Poängen visas hela tiden i det övre högra hörnet.

Ditt "skepp" sätts ut i mitten av skärmen och spelet kan börja. Marsianerna (The Aliens) kommer i en

stadig ström från vänstra sidan av "korridorerna". Du måste då flytta ditt skepp uppåt eller nedåt med A respektive Z-tangenterna, så att du kommer i höjd med den korridor där marsianen befinner sig. När du gjort så, skjuter du honom (eller henne) med "."-tangenten. Varje marsian ger dig 100 poäng.

Det låter lätt, eller hur? Men det finns en hake . . . Marsianerna skjuter nämligen tillbaka!

Maxpoängen är 10000 poäng (Håkan har aldrig nått den, säger han själv). Om du når 10000 poäng, kommer ett meddelande upp på skärmen, och datorn frågar om du vill spela igen. Tryck Y för ja och N för nej.

Om du själv blir skjuten, nollställs poängen och spelet börjar om på nytt efter en kort paus.

Om du väljer svårighetsgrad 2, kommer marsianerna att skjuta oftare.

Programförklaring

till EARTH ATTACK-spel:

Rader	
10	Randomisera.
20-30	Radera skärmen, sätt svart färg.
40-90	Definiera tecken.
100-250	Skriv ut skärmen.
260-290	Definiera variabler.
300-330	Skriv ut skepp och alien
340-380	Kollar tangentbordet.
390-430	Flyttar alien framåt.
440	Kolla om alien har passerat dig.
450	Alien laserrutin.
470-510	Flyttar skepp uppåt.
520-560	Flyttar skepp nedåt.
570-690	Din laser skjuter/test om alien blivit träffad.
700-750	Din laser om alien inte blivit träffad.
760-840	Poängräkning, visning på skärmen.
850	Sätter färgen till mörkgul.
860-1000	END-rutin.
1010-1080	Aliens laser/om du blivit träffad.
1090-1140	Aliens laser/om du inte blivit träffad.
1150-1230	"Aliens besegrade"-rutin.

LYCKA TILL!

```

1 REM *EARTH ATTACK*
2 REM *PCT-82*
3 CALL CLEAR
4 INPUT "GRADE(1-2)? ":GRADE
5 IF GRADE=1 THEN 10
6 LET SCORE=5000
7 RANDOMIZE
8 CALL CLEAR
9 CALL SCREEN(2)
10 CALL CHAR(33,"8181423C6AFF7E2
4")
11 CALL CHAR(40,"00021E36FE361E0
2")
12 CALL CHAR(34,"834421800221244
9")
13 CALL CHAR(64,"000000FFFF")
14 CALL CHAR(72,"FF8080808080808
0")
15 CALL CHAR(73,"808080808080808
0")
16 REM PRINT SCORE
17 CALL HCHAR(1,25,48,4)
18 CALL COLOR(3,16,2)
19 CALL COLOR(4,16,2)
20 REM PRINT SCENERY
21 FOR F=2 TO 20 STEP 3
22 CALL HCHAR(F,1,72,20)
23 CALL HCHAR(F+2,1,73,20)
24 CALL SOUND(100,200,1)
25 NEXT F
26 CALL COLOR(6,4,1)
27 CALL SOUND(100,-3,1)
28 REM PRINT BACKLINE
29 CALL CHAR(74,"808080808080808
0")

```

```

240 CALL VCHAR(2,30,74,21)
250 CALL SOUND(100,500,1)
260 REM START GAME
270 X=INT(6*RND)*3+3
280 Y=1
290 H=12
300 CALL HCHAR(H,22,40)
310 CALL HCHAR(X,Y,33)
320 CALL COLOR(1,9,1)
330 CALL COLOR(2,13,1)
340 CALL KEY(0,K,S)
350 IF S=0 THEN 390
360 IF K=65 THEN 470
370 IF K=90 THEN 520
380 IF K=46 THEN 570
390 CALL GCHAR(X,Y+1,C)
400 IF C=40 THEN 850
410 CALL HCHAR(X,Y+1,33)
420 CALL HCHAR(X,Y,32)
430 Y=Y+1
440 IF Y=30 THEN 850
441 IF SCORE=5000 THEN 442 ELSE
450
442 IF RND<.3 THEN 1010
450 IF RND<.9 THEN 1010
460 GOTO 340
470 IF H=3 THEN 390
480 CALL HCHAR(H-3,22,40)
490 CALL HCHAR(H,22,32)
500 H=H-3
510 GOTO 390
520 IF H=21 THEN 390
530 CALL HCHAR(H+3,22,40)
540 CALL HCHAR(H,22,32)
550 H=H+3
560 GOTO 390
570 IF H=30 THEN 700

```

```

580 CALL HCHAR(H,Y+1,64,21-Y)
590 CALL SOUND(10,900,1)
600 CALL COLOR(5,16,1)
610 CALL COLOR(5,1,1)
620 CALL HCHAR(X,Y,34)
630 CALL HCHAR(X,Y,32)
640 CALL SOUND(-500,-5,1)
650 GOSUB 760
660 CALL HCHAR(X,1,32,21)
670 X=INT(6*RND)*3+3
680 Y=1
690 GOTO 300
700 CALL HCHAR(H,1,64,21)
710 CALL SOUND(10,900,1)
720 CALL COLOR(5,16,1)
730 CALL COLOR(5,1,1)
740 CALL HCHAR(H,1,32,21)
750 GOTO 390
760 CALL GCHAR(1,26,C)
770 IF C=57 THEN 800
780 CALL HCHAR(1,26,C+1)
790 RETURN
800 CALL HCHAR(1,26,48)
810 CALL GCHAR(1,25,C)
811 IF C=53 THEN 812 ELSE 820
812 SCORE=5000
820 IF C=57 THEN 1150
830 CALL HCHAR(1,25,C+1)
840 RETURN
850 CALL SCREEN(11)
860 CALL SOUND(10,-2,1)
870 CALL SOUND(500,-4,1)
880 CALL HCHAR(H,22,34)
890 CALL VCHAR(5,5,33,5)
900 CALL HCHAR(5,5,33,4)
910 CALL HCHAR(7,5,33,3)
920 CALL HCHAR(9,5,33,4)
930 CALL VCHAR(5,10,33,5)

```

```

940 CALL VCHAR(5,13,33,5)
950 CALL HCHAR(5,10,33,3)
960 CALL VCHAR(5,15,33,5)
970 CALL VCHAR(6,18,33,3)
980 CALL HCHAR(5,15,33,3)
990 CALL HCHAR(9,15,33,3)
1000 FOR D=1 TO 1000
1001 NEXT D
1002 GOTO 10
1010 IF H=3 THEN 1090
1020 CALL HCHAR(X,Y+1,64,21-Y)
1030 CALL SOUND(10,-3,1)
1040 CALL COLOR(5,9,1)
1050 CALL COLOR(5,1,1)
1060 CALL HCHAR(X,Y+1,32,21-Y)
1070 CALL SOUND(1000,-7,1)
1080 GOTO 850
1090 CALL HCHAR(X,Y+1,64,30-Y)
1100 CALL SOUND(10,-3,1)
1110 CALL COLOR(5,9,1)
1120 CALL COLOR(5,1,1)
1130 CALL HCHAR(X,Y+1,32,30-Y)
1140 GOTO 340
1150 CALL CLEAR
1160 CALL SCREEN(7)
1170 PRINT "BÄR! DU HAR RÄDDAT J
ORIENT!":
1180 PRINT "TRYCK J FÖR NYTT SPE
L BÄR!":
1185 PRINT "N FÖR ATT SLUTA.":
1190 CALL KEY(0,K,S)
1200 IF S=0 THEN 1190
1210 IF K=74 THEN 1
1215 IF K=78 THEN 1220 ELSE 1190
1220 CALL CLEAR
1230 END

```

Köpt vaddå?

av Everett

Hej Brorsan och Svägerskan! Trevligt att ha er på besök. Men det blir inte varje dag man träffas eller hur. Häng av er och komma in i vardagsrummet där jag kan visa upp mitt senaste inköp – en hemdata.

–Jasså, har du köpt dej en hemdata. Ja, jag har funderat över det där med data förut men jag har bestämt mej att de inte kan göra något bättre än människan kan så jag avstår. Du behöver inte visa upp din – jag har fått dem demonstrerade många gånger förr. Hur har du det med båten?

–Tack bra. Men nu är det vinter. Vad menar du att datan gör ingenting bättre än människan gör?

–Ja ingenting väsentligt, eller hur. Datana bara kopierar, imiterar människan. Och imitationen blir alltid sämre än originalet. Eller hur? Och säj mej: Varför köpte du datan? Du måste ha haft någon idé om vad du skulle ha den till! Men jag har aldrig träffat någon som kan ge mej en rationell förklaring till varför dom köpte en data. Javisst, dom har den för diverse videospel, eller för att spela schack emot – men vad är det för förklaring egentligen?

–Du! Brorsan! Vad kan du ge för vettig förklaring för att du köpte din nya bil. Eller din ljusgrå kostym – den där snedrandiga slipsen – eller den där

apelsinmarmeladen du äter till fruskost. Skall jag försöka förklara varför jag köpte datan? Tror du att du kan lyssna till något som inte kanske är hundra procentigt rationellt – eller skall jag slå in det med paragraftecken före varje rad för att göra det mer apitligt?

–OK då. Gör som du vill. Jag lyssnar.

–När du köpte din första bil då visste du bara att du måste ha en bil – först efteråt upptäckte du en massa grejor som du kunde göra som du inte ens hade funderat på innan du köpte den. Inte visste jag heller varför jag skulle köpa en data – jag kunde ingenting om dom annat än att de skulle börja påverka mitt liv påtagligt. Så, liksom många andra, köpte jag först och tog reda på varför efteråt. Kanske en data inte är mera än en människa – men en människa med en data är mycket mer än en som är utan. Nu och för all framtid. Så vill du behålla din nya bil, din gråa kostym med den randiga slipsen och din apelsinmarmelad, råder jag dig att investera i en data.

–Usch då. Vad säger du älskling? Skall vi skaffa oss en data?

–Never.

Suck!

SPANNANDE

PÅ SVENSKA

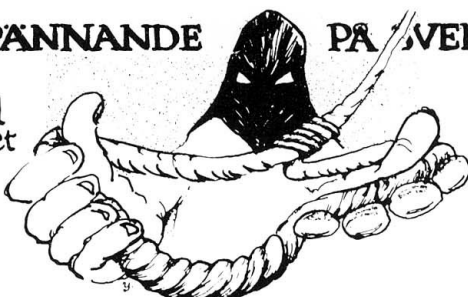
GLUF'S

I kakan finns en förgiftad bit. Inbakad eller synlig? Det bestämmer du! Natti-natti för den som glufsar fel!

ORDEOLOGI

Obs! 50:-

Ett måste för korsords-entusiasten!



T-KENO

Gamblingsuccé i USA!
Olika insatser. Vinster 5-5000:-

KALAH

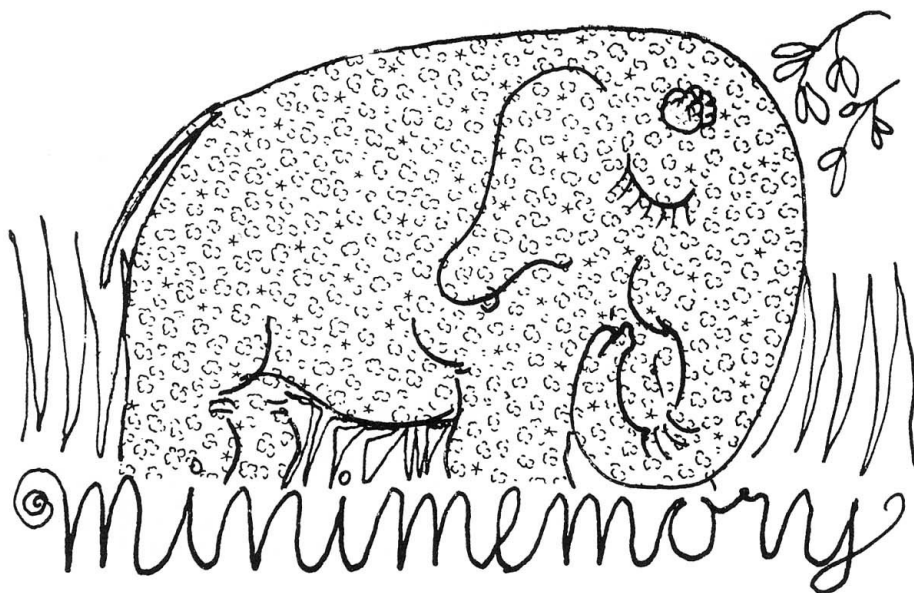
2000-årig strategi. Spela mot din kompis eller självaste skirken!

DU ÄR HÄNGD!

Klassisk med massor av ord.

1 SPEL 75:- T-DATA 08-35 55 04

2 SPEL 140:-



av Björn Gustavsson

En av de mer användbara modulerna från TI är Mini Memory modulen. Den kan mer än namnet antyder. Den innehåller sammanlagt 14 K minne 4 K RAM, 4 K ROM och 6 K GROM (Graphics Read Only Memory).

RAM är ju den sorts minne som BASIC-programmen lagras i. När man slår av datorn försvinner programmet. Här har vi MM-modulens första överraskning: Man kan lagra ett program i den, plocka ur modulen och hantera den precis som en kassett eller diskett. När man sedan sätter tillbaka den i datorn finns programmet kvar i modulen och kan enkelt laddas in. Detta är möjligt tack vare batterier som är inbyggda i modulen. Batterierna räcker åtminstone några år, och när de är slut kan man byta ut modulen mot en ny.

Förutom batteriförsett RAM innehåller modulen GROM och ROM som utökar TI BASIC med sju nya instruktioner, inklusive PEEK och POKE. Samma GROM och ROM gör det möjligt att använda systemrutiner från assemblerprogram. Dessutom finns det något man kan ha nytta av vid assemblerprogrammering: en debugger kallad EASY bug. Nu kan man fråga sig vad man har för nytta av allt detta utan en assemblerator. Svar: En kassett med en assemblerator medföljer modulen. På samma kassett finns också ett demonstrationsprogram som visar TI-99/4A:s grafiska möjligheter.

Låt oss undersöka MM-modulens möjligheter lite närmare:

EXTRA FILER

MM-modulen kommer säkert att användas i stället för bandspelare för att lagra BASIC-program. Med modulen på plats har man en extra fil som heter "MINIMEM". För att spara ett BASIC-program skriver man bara SAVE MINIMEM. För att hämta tillbaka det skriver man OLD MINIMEM. Alltså precis som vilken annan fil som helst. Skillnaden är bara att det går mycket snabbare t. o. m. snabbare än att lagra på diskett.

Det går också att öppna filen från ett BASIC-program och använda den för datalagring.

Om man har minnesexpansion kan man även använda den som en fil. SAVE EXPMEM2 sparar ett program i 24K-delen av expansionen.

ROM OCH GROM

I ROM och GROM finns extra BASIC-instruktioner, länkar till systemrutiner samt debuggern EASY BUG.

Följande extra instruktioner finns:

CALL PEEK läser bytes från CPU-RAMet till variabler. CALL PEEK(8192,A,B,C) t ex läser tre bytes från adresserna 8192, 8193 och 8194 och tilldelar dessa till variablerna A, B och C.

CALL PEEKV fungerar som CALL PEEK, förutom att läsningen sker från VDP-RAM. (BASIC-program lagras i VDP-RAM, medan assemblerprogram måste lagras i CPU-RAM.)

CALL POKEV lagrar data i VDP-RAM.

CALL CHARPAT är motsatsen till CALL CHAR, dvs CALL CHARPAT(91, D\$) ger en sträng som visar hur ascii-tecken 91 är definierat. (Se CALL CHAR i BASIC-manualen.)

Det återstående tre instruktionerna gör det möjligt att ladda och köra assemblerprogram. CALL INIT initierar CPU-minnet för laddning av assemblerprogram. CALL LOAD laddar ett assemblerprogram.

CALL LOAD ("DSK1.DEMO") laddar ett assemblerprogram från filen DEMO i drive 1. Alternativt kan man skriva CALL LOAD(9197, 85,40) vilket direkt placerar talen 85 och 40 i 8197 och 8198.

CALL LINK, slutligen, kör ett assemblerprogram. När det är färdigt sker återhopp till BASIC.

ÅTERKOMST AV SYSTEMRUTINER

Modulen innehåller olika hjälprutiner som kan anropas från assemblerprogram. För de intresserade följer här en kortfattad lista över tillgängliga rutiner.

VDP single byte write — skriv en byte i VDP RAM.

VDP multiple byte write — skriv flera bytes i VDP RAM.

VDP single byte read och VDP multiple byte read — läs en resp flera bytes från VDP RAM.

VDP write to register — skriv en byte till något av VDP-registren.

Keyboard scan — avsök tangentbord och ge en tangentkod och statuskod (motsvarar CALL KEY i BASIC).

Load standard character set — ladda de vanliga definitionerna av tecknen.

Load small character set — laddar små bokstäver.

Execute power-up routine — den rutin som utförs när datorn slås på. Accept tone och Bad response tone — ger olika ljudsignaler.

Bit reversal routine — ger en spegelbild av en byte (det är ofta praktiskt för att spegelvända en tecken-definition).

Cassette device service routine — gör det möjligt att använda kassettbandspelaren från ett assemblerprogram.

Load lower case character set — ladda de små versalerna.

Dessutom finns det en mängd rutiner som räknar med flyttal.

Slutligen finns det rutiner som kopplar ihop assemblerrutiner med BASICen. Man kan t ex överföra värden till/från BASIC-variabler. Det finns också möjlighet att ge BASIC-programmet en felkod om något går galet.

EASY BUG

I modulens ROM finns också EASY BUG. Med EASY BUG kan man:

1. Lista och ändra CPU-minnets innehåll.
2. Lista innehållet i GROM.
3. Lista och ändra VDP-RAM.
4. Exekvera assemblerprogram.
5. Hantera in/ut-portarna.
6. Spara assemblerprogram på kassett.
7. Läs in ett program från kassett.

ASSEMBLATORN

Med MM-modulen följer en "line-by-line" assembler på kassett. Den kan läsas in och lagras i modulens RAM och man har fortfarande 2 K kvar för egna program. Assemblatorn assemblerar varje rad direkt när den matas in och lagrar objekt-koden direkt i minnet. En trevlig egenskap är att den ger ett felmeddelande omedelbart när man slår in ett felaktigt tecken.

En del källkod bevaras i en nio sidors textbuffert och man kan kontrollera att man har matat in programmet riktigt genom att visa texten med upp- och nerpilarna. Det går emellertid inte att ändra eller spara källkoden.

"LINES"

LINES är ett demonstrationsprogram med på samma kassett som assemblern. Det ritar linjer på skärmen med en mängd olika färger. Riktigt dekorativt.

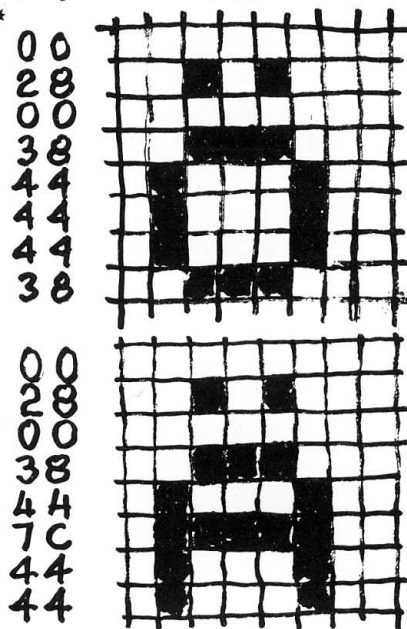
SLUTSUMMERING

Utan tvekan en riktigt prisvärd modul, både för BASIC- och assemblerprogrammerare. Den utgör enda möjligheten att programmera i assembler utan minnesexpansion och diskettenheter.

Observera dock att MMM-manualen inte lär ut assemblerprogrammering. Man måste ha manualen "Editor/Assembler" och förkunskaper om assemblerprogrammering.

TEKNIKTIPS

Å, Ä OCH Ö på 99:AN
av Björn Gustavsson



Tyvärr finns ju å, ä och ö inte i 99:ans teckenuppsättning. Som tur är det lätt att definiera om tre tecken.

Det är lämpligt att definiera om [som Ä, \ som Ö och] som Å. Varför just dessa tecken? frågar sig vän av ordning. Anledningen är att tecknen har ascii-koderna 91, 92 och 93, som är de vanliga ascii-koderna för ä, ö och å på andra datorer och skrivare. Genom att också vi använder dessa koder kan vi använda en vanlig skrivare (som kopplas över RS232-interface) eller överföra program till en annan dator.

Observera att ordningen tyvärr är: Å=91, Ö=92, Ä=93. Det beror på att tyskarna fick bestämma ordningen i den europeiska varianten av ascii. De använder varianten: Ä=91, Ö=92 och β=93. Det måste vi svenskar anpassa oss efter.

I TI BASIC definierar man ä, ö, å och följande tre rader:

```
CALL CHAR(91,"00280038447C4444")
CALL CHAR(92,"0028003844444438")
CALL CHAR(93,"00100038447C4444")
```

I Extended BASIC kan man också använda dessa tre rader, men man kan också skriva:

```
CALL CHAR (91,"00280038447C4444
002800384444443800100038447C44
44")
```

Exempel hittar du i programmet "Skrivmaskinstränare".

MARKÖRSTYRNING I TI BASIC
av Björn Gustavsson

Vi skall i den här och kommande artiklar titta på hur olika instruktioner i TI Extended BASIC faktiskt kan utföras i TI BASIC med hjälp av en subrutin.

Här kommer en subrutin som simulerar DISPLAY AT att presenteras, dvs möjligheten att skriva ut text var som helst på skärmen. Här är listningen:

```
1000 REM MARKÖRSTYRNING
1010 FOR Z=1 TO LEN(Z$)
1020 CALL VCHAR(RAD,KOL,ASC
      (SEG$(Z$,Z,1)))
1030 KOL=1-KOL*(KOL<>32)
1040 RAD=RAD-(KOL=1)
1050 NEXT Z
1060 RETURN
```

Radnumren i subrutinen kan ändras fritt eftersom inga GOTO eller GOSUB används i rutinen. Den anropas så här:

```
90 RAD=3
100 KOL=10
110 Z$="TEXT SOM SKRIVS UT"
120 GOSUB 1000
```

I det här fallet skrivs texten TEXT SOM SKRIVS UT fr. o. m. rad 3, kolumn 10 på skärmen. Det går också bra att skriva ut tal:

```
170 RAD=2
180 KOL=1
190 Z$="RESULTAT: "&STR$(TAL)
200 GOSUB 1000
```

Om variabeln TAL har värdet 23 skrivs RESULTAT: 23 ut på rad 2, kolumn 1 på skärmen.

föreningen 99:an
söker en
PROGRAMFÖRMEDLARE
med diskett



MURPHYS LAW och TI 99

av Björn Gustavsson

Murphys tredje lag lyder: "Allting som kan gå gale, kommer att gå gale." Till detta har Pincus tillagt: "Och vid värsta möjliga tidpunkt." (Någon har sagt: "Murphy var en optimist.")

Betydelsen av Murphys lagar är väl inpräglad i alla professionella programmerares hjärnor. För att undvika kostsamma misstag har "proffsen" utvecklat vissa regler för att minimera antalet problem som dyker upp vid programutveckling. I den här artikeln skall jag beskriva några av dessa regler.

REGEL 1: NAMNGE VARIABLERNA

De flesta nybörjare använder korta namn som X1, X2, X3 etc. Problemet med den sortens "namn" är att komma ihåg vad variabeln representerar. Som tur är kan man använda långa variabelnamn i både vanlig och Extended BASIC. Men tyvärr blir programmet långsammare av långa namn. En lämplig kompromiss är att använda namn med fyra till fem bokstäver som någorlunda talar om vad den används till. Använd inga namn som du riskerar att stava fel.

När det gäller loop-variabler, alltså sådana som används i FOR — NEXT loopar är det faktiskt lämpligt att använda ett kort namn. En

lämplig standard för detta är att använda namnen I till N. (Detta har anor från FORTRAN, det allra första högnivåspråket.)

Du bör också ha en lista över alla variabler. Och jag menar alla. Frestelsen kan vara stor att lägga till en variabel till programmet utan att skriva in den på listan. En ofullständig lista är sämre än ingen alls.

REGEL 2: SPAR PROGRAMMET

Gör det till en vana att alltid spara programmet när du har matat in maximalt 75 nya rader (50 rader om du har skivenhet). Viktiga program skall alltid sparas flera gånger på olika kassetter eller disketter.

REGEL 3: TA'T LUGNT

Hur många skyndar sig inte att prova programmet när de sista raderna matats in? Om du brukar göra så, bli inte förvånad om Murphy tittar fram och datorn "hänger sig" så att man bara kan slå av den. För att undvika detta skall man alltid spela in hela programmet innan man provkör.

Ta det också lugnt när du provkör det. En lag säger nämligen att ett program aldrig fungerar första gången. Skriv ner på ett papper vad du tänker testa första gången du kör programmet. Kör programmet, och notera fel som du upptäcker (och som inte stoppar programmet).

Detta av följande orsaker:

När man ändrar en rad nollställs alla data som eventuellt har matats in. Detta måste matas in igen.

Du kanske inte upptäcker ett stort fel för att du rättar till ett litet fel.

Du kan glömma bort att anteckna ändringen.

Det tar betydligt längre tid att rätta ett fel i taget.

När du har testat så mycket som möjligt, stanna programmet och rätta till fel. Anteckna vilka fel du rättar till. Du kan behöva ändra tillbaka.

REGEL 4: TESTA, TESTA, TESTA

Det är väldigt svårt att få fram ett felfritt program och praktiskt taget omöjligt om programmet är stort och komplicerat. Därför bör man testa, testa och åter testa.

(Bearbetning av en artikel från 99'er Magazine v1n3p32.)

Översättning

```
2 PRINT TAB(30);"ONE CHECK"
4 PRINT TAB(15);"CREATIVE COMPUTING MORRISTOWN, NEW JERSEY"
6 PRINT:PRINT:PRINT
8 DIM A(64)
10 PRINT "SOLITAIRE CHECKER PUZZLE BY DAVID AHL"
15 PRINT
20 PRINT "48 CHECKERS ARE PLACED ON THE 2 OUTSIDE SPACES OF A"
25 PRINT "STANDARD 64-SQUARE CHECKERBOARD. THE OBJECT IS TO"
30 PRINT "REMOVE AS MANY CHECKERS AS POSSIBLE BY DIAGONAL JUMPS"
35 PRINT "AS IN STANDARD CHECKERS). USE THE NUMBERED BOARD TO"
40 PRINT "INDICATE THE SQUARE YOU WISH TO JUMP FROM AND TO. ON"
45 PRINT "THE BOARD PRINTED OUT ON EACH TURN '1' INDICATES A"
50 PRINT "CHECKER AND '0' AN EMPTY SQUARE. WHEN YOU HAVE NO"
55 PRINT "POSSIBLE JUMPS REMAINING, INPUT A '0' IN RESPONSE TO"
60 PRINT "QUESTION 'JUMP FROM ?'"
62 PRINT
65 PRINT "HERE IS THE NUMERICAL BOARD:"
66 PRINT
70 FOR J=1 TO 57 STEP 8
74 PRINT J;TAB(4);J+1;TAB(8);J+2;TAB(12);J+3;TAB(16);J+4;TAB(20);J+5;
75 PRINT TAB(24);J+6;TAB(28);J+7
76 NEXT J
77 PRINT
78 PRINT "AND HERE IS THE OPENING POSITION OF THE CHECKERS."
79 PRINT
```

IF NYBÖRJARE THEN READ
av Everett

Först kom jag hem med nittinian, kopplade upp den, kasta "Röster i Radio och TV, sökt frenetiskt efter min gamla Philips monospelare från sextitalet, koppla på den efter världens anordning för att anpassa dinkontakterna till den japanska snoppen och knacka in "inchworm", "graphics match" etc.

Nästa dag var jag på bokhandeln och köpte en stor läcker bok från Creative Computer med massor av games för min nittinia. LADDELADDELADDELADDE... gick tangenterna och jag knackade in "ONE CHECK" på nån timme sådär. Och nu bara en test run innan jag kallar hit gänget!

Trodde jag.

I min okunnighet.

Och utan att ha läst Murphys lag i denna tidning. Och utan att ha plöjt mej genom "manualen".

FOR-NEXT ERROR

Jag kunde sätta REM på alla FOR-NEXT ställen, men då blev det

INCORRECT STATEMENT

osv.

Nu fattade jag att olika datorer talar olika "dialekter" och att jag måste "översätta" programmet för min nittinia.

Det första anmärkningsvärda är TAB(42) — nittinian har ju bara trettitalet positioner på bredden. ONE CHECK är nio bokstäver lång, mittpunkten på skärmen ligger på femton, börja då på ena sidan, 15-4=11. Ändra till TAB(11);ONE CHECK.

Samma sak men CREATIVE COMPUTING osv. Men det räknar du ut själv nu när du har algoritmen.

Sedan ser jag att varje rad har ett nummer — behövs inte på nittinian — men raderna innehåller flera "commands" eller "statements". Det går inte på nittinian. Dessa olika commands och statements är åtskilda av kolontecken:. Detta innebär att varje kolon kan vara en signal till mej att avsluta raden där och börja på en ny — med nytt radnummer.

Det finns naturligtvis undantag — vill man åstadkomma några raders mellanrum i "microsoft" basic kan man XXX PRINT:PRINT:PRINT.

I TI-Basic kan man lämpligen skriva XXX PRINT:: för att åstadkomma samma sak.

En stor fördel med att ha programmet i maskinen — trots att maskinen inte kan tolka det, är att kunna använda EDIT mode. Jag arbetar med ett

```
80 FOR J=1 TO 64
82 A(J)=1
84 NEXT J
86 FOR J=19 TO 43 STEP 8
88 FOR I=J TO J+3
90 A(I)=0
92 NEXT I
94 NEXT J
96 N=0
98 GOTO 340
100 INPUT "JUMP FROM";F
105 IF F=0 THEN 500
110 INPUT "TO";T
112 PRINT
120 REN *** CHECK LEGALITY OF MOVE
120 F1=INT((F-1)/8)
130 F2=F-8*F1
140 T1=INT((T-1)/8)
150 T2=T-8*T1
160 IF F1>7 THEN 230
170 IF T1>7 THEN 230
180 IF F2=8 THEN 230
190 IF T2=8 THEN 230
200 IF ABS(F1-T1)<2 THEN 230
210 IF ABS(F2-T2)<2 THEN 230
212 IF A(T1+T2)/2=0 THEN 230
215 IF A(F)=0 THEN 230
220 IF A(T)=1 THEN 230
225 GOTO 250
230 PRINT "ILLEGAL MOVE. TRY AGAIN..."
240 GOTO 100
245 REN *** UPDATE BOARD
250 A(T)=1
260 A(F)=0
270 A((T+F)/2)=0
290 N=N+1
310 REN *** PRINT BOARD
340 FOR J=1 TO 57 STEP 8
350 FOR I=J TO J+7
360 PRINT A(I);
370 NEXT I
380 PRINT
390 NEXT J
400 PRINT
410 GOTO 100
490 REN *** END GAME SUMMARY
500 S=0
510 FOR I=1 TO 64
520 S=S+A(I)
530 NEXT I
540 PRINT "YOU MADE";N;"JUMPS AND HAD";S;"PIECES"
550 PRINT "REMAINING ON THE BOARD."
560 PRINT
562 INPUT "TRY AGAIN";A$
570 IF A$="YES" THEN 70
575 IF A$="NO" THEN 600
580 PRINT "PLEASE ANSWER 'YES' OR 'NO'."
590 GOTO 562
600 PRINT
610 PRINT "O.K. HOPE YOU HAD FUN!!"
999 END
```

tiotal rader på skärmen, säg rader 100, 110, 120, 130 osv. Sedan tar jag på nytt rad 100, DELETE fr o m första kolontecknet, sätter upp nytt rad 103 och skriver ut det andra command eller statement, som jag har kvar på den första utskriften, högst upp på skärmen. Det är bra att ha minst 10 som avstånd mellan radnummer. Detta innebär att man skall RESEQUENCE 10,10 innan man börjar att översätta.

Nu kommer programmet i TI-basics programformat. Det återstår dock några viktiga punkter som måste förändras innan programmet fungerar i nittinian.

För det första finns det commands, och i synnerhet definitions, som är olika från ena dialekten till den andra. Det finns två sätt att ta reda på vad som är vilket. Man skaffar sej handboken för dialekten det gäller — om man har en vän med en annan dator och vill göra avskrifter av hans program till exempel.

Eller man köper "Liens Basic Handbok" finns på svenska. Där finns de flesta basic fraser ur de otaliga dialekterna med synonymer och testprogram. I Hangmanexemplet finns MIDfunktionen som inte finns i TI-basic. Man använder SEG istället. För det andra finns det en källa till fel i formatet med flera commands per rad då det gäller IF... THEN satsen. Oftast följs THEN av en direkt command. I TI-basic måste den följas av en radhänvisning. Sedan följs den ofta av ytterligare en command på samma rad. Denna command är också villkorsbunden av IF... THENsatsen.

EXEMPEL:

330 IF N=1 THEN GOSUB 3000:PRINT

kan översättas med:

eller:

330 IF N=1 THEN	330 IF N=1 THEN
333 GOSUB 3000	333 GOTO 340
336 PRINT	335 GOSUB 3000
	337 PRINT

Om du nu följer efter i mina fotspår, då har du ett engelskt ONE CHECKspel, utan ljud eller färggrafik och byggt på en tämligen gammal form av basic. Hur man översätter den till svenska, får radjusterade texter och dekorerar med färg, teckningar och ljud, återkommer jag till i nästa nummer.



Texas Instruments hemdator.

Texas Instruments hemdator är en riktig dator. Som hela familjen kan arbeta med, räkna på, lära av eller spela med.

Det finns mer än 1000 färdiga program från hela världen – från rymdspel och schack till budgetering av familjens ekonomi. Lägg bara in programmen och du är klar att börja.

När du själv vill programmera ger den utförliga handboken dig instruktioner, steg för steg. Spännande och inte alls svårt.

Texas Instruments hemdator har hela 16 Kbyte användarminne (RAM) redan från början men stannar inte vid detta – utvecklingsmöjligheterna är mycket stora. Du kan tex ansluta en talenhet som ger din dator röst.

Kom in och prova Texas Instruments hemdator – den är mer än värd sitt pris.



TEXAS INSTRUMENTS

I NÄSTA NUMMER INVIGS SPELHALLEN MED
MÄSTERGALLERIET, SPEL OCH RECENSIONER.
EXTENDED BASIC GENOMGÅS, TEKNIKTIPS
OCH MITTINITTINIAN GÅR PÅ MÄSSAN.