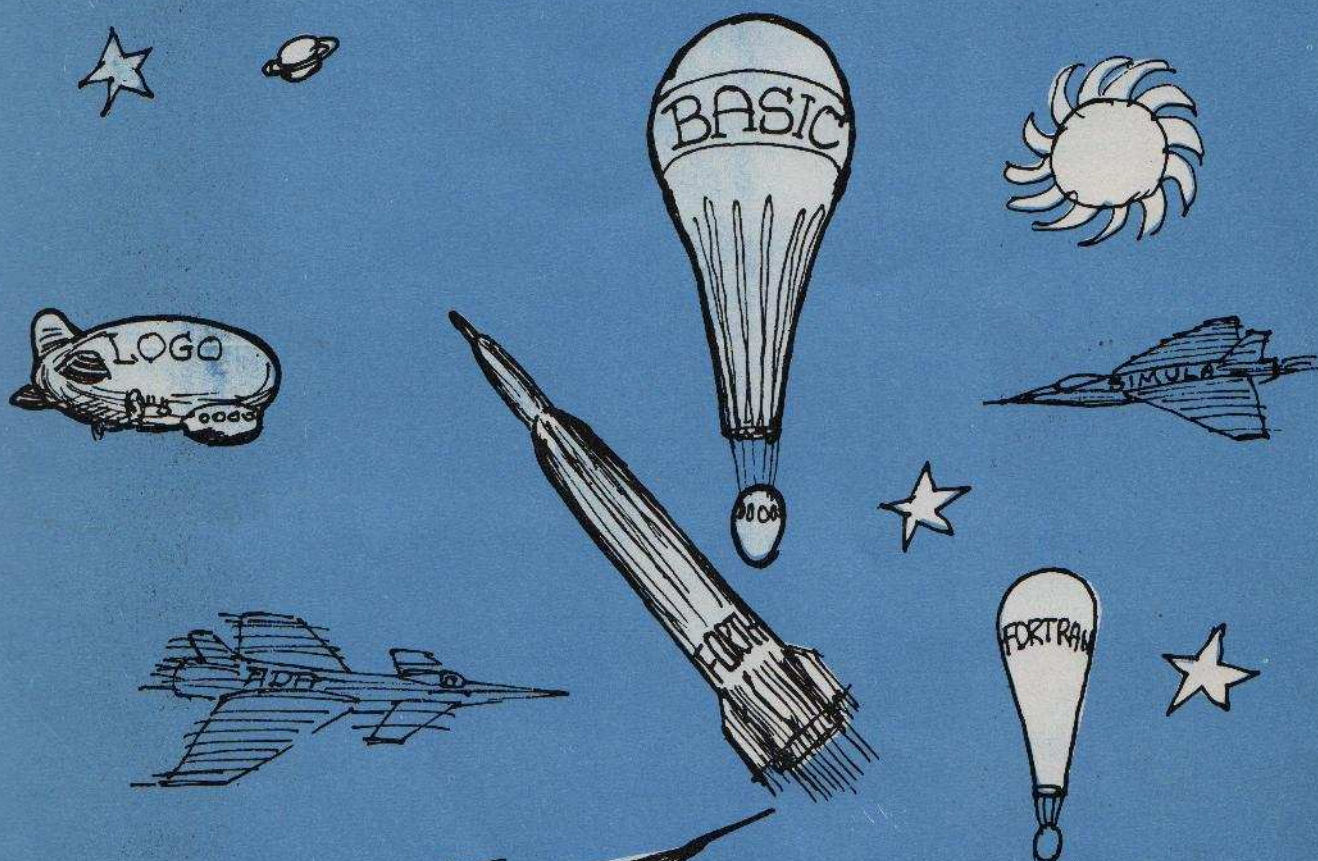


nittinian



NITTINIAN NR 2

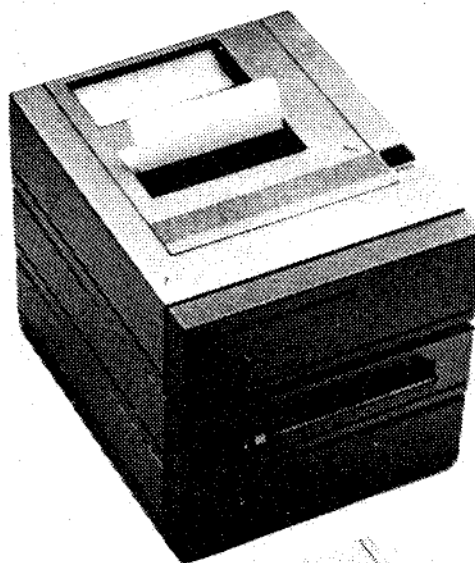
- 4 TEKNISKA AVDELN.
- 6 MÄSSAN I SOLLENTUNA
- 7 PROGRAMFÖRMEDELAREN
- 8 SPELHALLEN - MUNCHMAN
- 10 DOMINION, ONE-CHECK
OCH DATAKRYSSSET
- 12 99ans MÅNGA SPRÅK
- 14 EXTENDED BASIC
- 16 FORTH
- 18 GRAFISKA SKOLAN.
- 20 ANNONSERING

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 kompaktdatorer 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 grundutförandet.



**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.

Med det här numret av medlemsbladet som Du nu får kan man säga att 99-användarna "kommit tillbaka" till Föreningen Programbiten. "Kommit tillbaka" därför att Programbiten varit användarföreningen för TI:s programmerbara räknare alltsedan 1978 medan "Föreningen Nittinian" först bildades som en fristående förening även om det var under Programbitens "beskydd". Men nu är vi alltså alla medlemmar av Programbiten, både användare av hemdatorer och programmerbara räknare.

I programbitens verksamhet har det ingått att sända ut ett medlemsblad fyra gånger om året där vi skrivit om programmeringsfinesser och publicerat program. Dessutom har det byggts upp ett programbibliotek (för programmerbara räknare) dels genom att den som skänkte ett användbart program fick två andra i utbyte, dels (och inte minst) genom att klubbar i andra länder ställt sina programbibliotek till förfogande mot mycket låga avgifter. Vidare har vi förmedlat böcker av gemensamt intresse.

Förhoppningsvis kommer vi att gå vidare i liknande banor för 99-ägarna. När det gäller programbibliotek har Thomas Coppens på TIsoft i Belgien varit mycket tillmötesgående och låtit oss få förmedla alla deras program för TI-99/4A. Vad Thomas hoppas få i utbyte (utom en ringa förmedlingsavgift) är tillgång till trevliga program som görs av Programbitens medlemmar. Den utmaningen tycker jag vi gärna bör anta.

När det gäller föreningsservice för 99-ägarna får vi väl säga att vi ännu är i uppbyggnadsskedet. Det betyder — eftersom vi är en förening och inte en kommersiell servicecentral — att vi är beroende av intresserade krafter som kan tänka sig att åta sig någon viss uppgift inom föreningen. Det kan var programförmedling eller andra servicefunktioner där man kommer i kontakt med medlemmarna eller också kan det vara granskning av bidrag till medlemsbladet vilken ju är en förutsättning för att detta skall hålla en god nivå. Hör av Dig till oss, det är ett trevligt sätt att bli bekant med andra programmeringsintresserade.

Adressen dit Du som 99-användare kan skriva är t ex
Björn Gustavsson
c/o Schibler

Wahlbergsgatan 6
121 46 JOHANNESHOV

Med bästa hälsningar

Lars Hedlund ordf

NR 2

Redaktionen består av Björn Gustavsson, som är vår tekniska redaktör, Peter Olin, som vår spelredaktör, Yvonne Thorfve, en hängiven nittinia-grafiker, vår grafiska redaktör och Roger Everett som är illustratör och produktionsansvarig. Ansvarig utgivare är föreningen Programbitens ordförande, Lasse Hedlund. Samtliga i tidningsredaktionen är privatägare av Texas hemdatorer och är oberoende av Texas Instruments — med tillfälligt undantag för Björn som är sommarpraktikant där. Till nästa nummer kommer vi att utökas med Bo Nordlin från Programbiten. Han skriver TI-spalten i Datorhobby och är flitigt i elden i Programbiten. Han har lovat att vara redaktör för cc-40 materialet.

Detta nummer fortsätter med ett taktiskt spel för flera från Peter Olin, fortsättningen på "One-check", Yvonne och Rogers iakttagelser på Mikrodatormässan, en presentation av extended basic, ett upprop från föreningens nya programförmedlare, Hans Attersjö, ett sommarkryss på datorspråk, utmaningen till Mästargalleriet och en del annat. Allt för att underhålla, informera och framförallt inspirera dej.

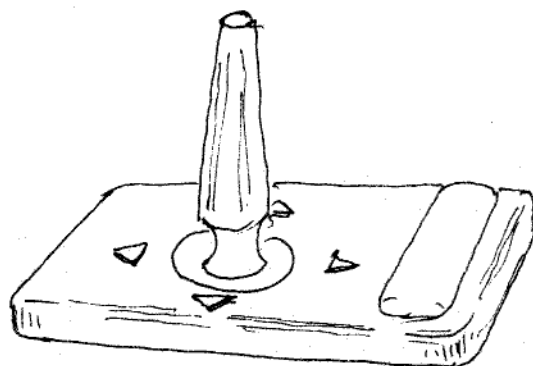
Hälsningar

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.

PRENUMERERA

Betala in 100 kronor till Nittinians postgiro 19 83 00-6 och markera «1983 abonnemang»



TEKNISKA AVDELNINGEN

Den här avdelningen kommer att handla om lite mer tekniska tillämpningar för 99:an. Vi kommer bl a att skriva om 99:ans hårdvarumässiga uppbyggnad och beskriva enkla tillbehör som man kan bygga själv för att t ex styra olika saker med 99:an. Vi har en del idéer, men förslag på användbara tillbehör och/eller ritningar är välkomna.

Vi kommer också att svara på frågor om hårdvara (datorn och tillbehör) eller mjukvara (program). Skicka in frågor och förslag — det är Du som bestämmer vad vi skall skriva om.

Vi som är redaktörer för avdelningen heter Björn Gustavsson och Lars Magnusson.

Vi inleder avdelningen med ett vanligt problem — joysticks som inte rör sig uppåt. De flesta ägare av hemdatorn har säkert själva upplevt detta. Jag var själv lika förvånad tills jag följde det goda rådet: "När inget annat hjälper, läs bruksanvisningen."

I bruksanvisningen står det nämligen att ALPHA LOCK ej får vara nedtryckt när man använder joysticks.

NYHETER

Ti har introducerat en ny serie inlärningsspel i Sverige. Dessa spel bygger på något eller några av de fyra räknesätten. I Meteor Multiplication t ex blir man anfallen av meteoror, där varje meteor har en multiplikationsuppgift. Genom att rikta kanonen mot en meteor och ge rätt svar kan man skjuta sönder den.

Spelen heter: Alligator Mix, Alien Addition, Demolition Division, Dragon Mix, Minus Mission, Meteor Multiplication, Math Games 2 och Math Games 6.

EDITOR/ASSEMBLER MANUAL ERRORS

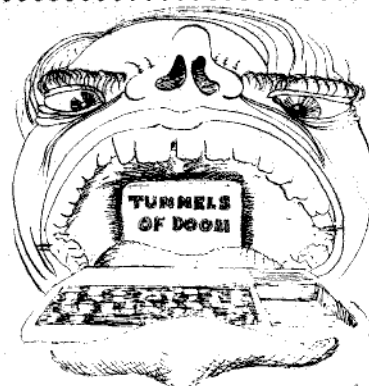
Until now the following errors have been found in this manual:

PAGE	SECTION	DESCRIPTION
42	3.1.3.1	In the first paragraph, last sentence: change "least" into "most".
83	6.2	Line 5 should read: "plus >18 (the value in memory byte 2123)".
92	6.10	In the second line of the example: change "value of addr." into "value in addr."
103	6.14.2	In the example: change "MOV *11,1" into "MOV *11"
127	7.20.1	In next to last line: change ">2220" into ">C220"
142	8.2	In the example-table at the bottom of the page: the columns under Source and Destination must be swapped. (Compare with table on page 140).

- 168 10.5 In the example change ">2A41" into
"02A41", and change "Register 3" into
"Register 2".
- 214 This page is an exact copy of page 212;
the real page is missing, containing for
example the chapter concerning PSEG.
- 230 14.4.3.1 Last paragraph: change "source listing"
into object file".
- 231 14.4.3.1 Replace the second paragraph with the
following text: To run the program,
select the LOAD AND RUN option on the TI
Editor/Assembler. The file name is
DSK1.TOMB, the program name is START.
Alternatively, you may run the same from
TI EXTENDED BASIC. However, the object
file TOMB is in truncated format, so the
file TOMBS must be reassembled.
Hereafter the following program may be
run:
- ```

100 CALL INIT
110 CALL LOAD("<object name>")
120 CALL LINK

```
- 262 16.2.4 Add the following note: NOTE: Some  
devices modify the GROM read address.  
RS232 is a known offender. If  
your program access this device,  
you should save the current GROM address  
before the I/O operation (See Section  
16.5.2), and restore it afterwards (See  
Section 16.5.1). Otherwise the program  
will not be able to return to the  
EDITOR/ASSEMBLER or to BASIC, or perform  
a "BLWP @GPLLNK" properly.
- 289 15.2.1 Change line 130 in the BASIC program into  
CALL LOAD("DSK1.BSCUP","DSK2.STRING").  
This assumes, that the source file on the  
next page has been entered using the  
EDITOR, and saved as "DSK2.STRING", and  
that the ASSEMBLER has been run, using  
"DSK2.STRING" as source file and  
producing "DSK2.STRINGO" as object file.
- 328 21.1 The default for Register 7 is >07 in TI  
BASIC and EXTENDED BASIC.
- 335 21.5 In the second paragraph: change ">00 or  
>04" into ">03 or >07". In the next to  
last paragraph include: change ">00 or  
>04" into ">7F or >FF".
- 344 21.7.1 Change line 2 into "AI R0,->20".
- 415 24.4.8 The second instance of "GRMRD" must be  
changed into "GRMWA EQU >9802".
- 416 24.4.8 Change the second line into: "NUMREF  
EQU >200C".



Ett nytt adventurespel har äntligen kommit till Sverige — Tunnels of doom. I en labyrint med upp till tio våningar skall du söka efter en kung. Massor av monster gör det hela lite roligare.

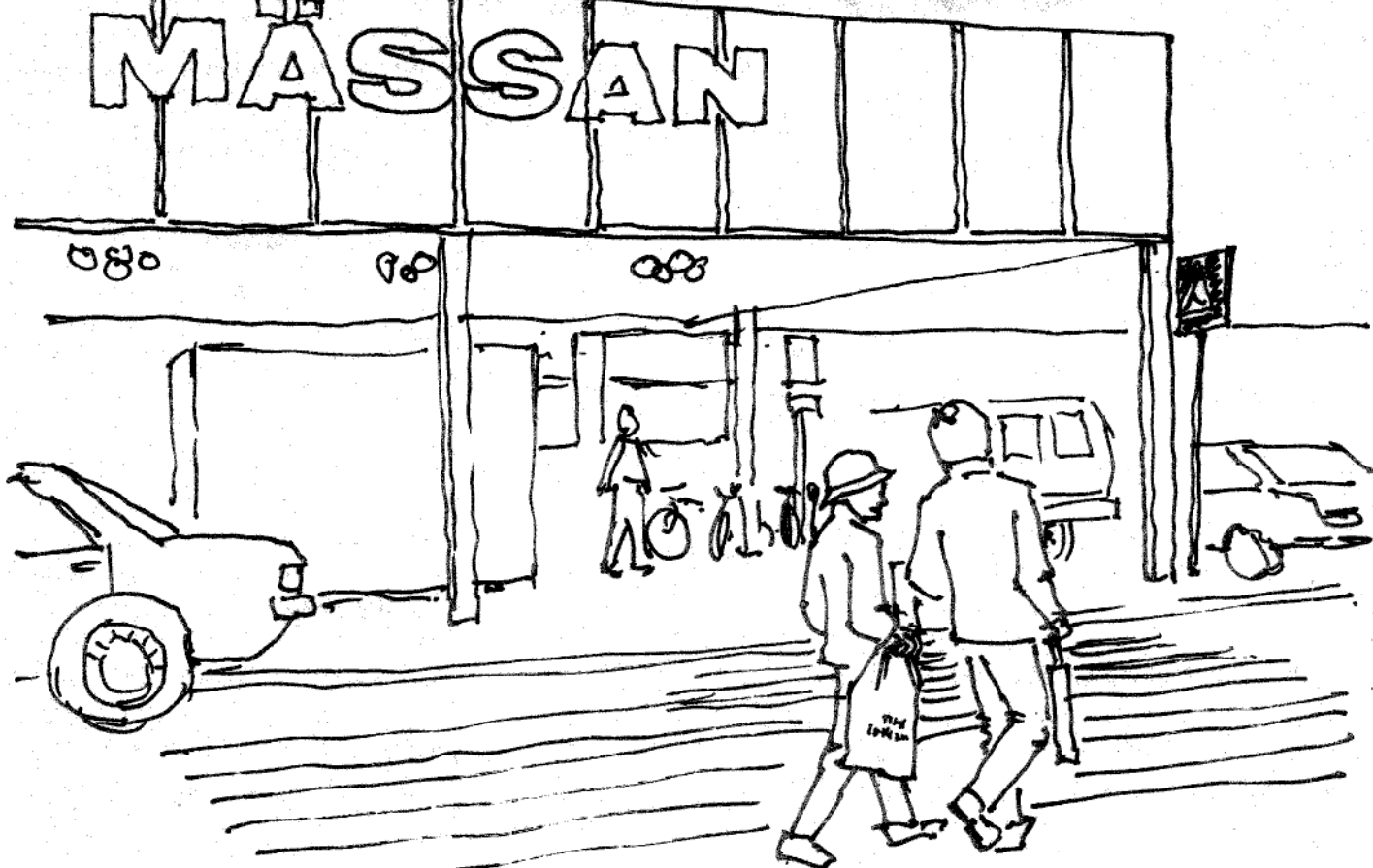
I motsats till Adventure som endast innehåller text finns här utmärkt grafik. När man går i korridorerna mellan rummen visas omgivningarna tredimensionellt (i perspektiv). När man kommer in i ett rum ser man rummet med innehåll — monster och skatter — tvådimensionellt ovanifrån. Det gäller att bestämma sig om man skall fly eller slåss.

Man behöver heller inte vara ensam. Man kan låta upp till fyra figurer följas åt.

Vår spelredaktör, Peter Olin, köpte Tunnels of doom från USA förra året och hans recension kommer i nästa nummer.



För den som saknat svensk litteratur om hemdatorn kan vi meddela att Studentlitteratur kommer med tre titlar under sommaren/hösten: BASIC på Texas Instruments hemdator, Lär dig använda TI-99/4A och Avancerad BASIC på Texas hemdator. Även TI Sverige håller på att ta fram svensk litteratur.



## NITTINIAN PÅ MÄSSAN

Årets Mikrodatormässan i Sol-lentuna Mässhall slog alla rekord för antal besökare. En del av nittinians redaktion bor ovanför mässhallen och kan rapportera att reportagerna i tidningar och lokalradion överdrev inte. Antalet besökare måste ha varit otroligt. Vi stötte på grannar som hade haft världens jobb att övertyga polisen om att de verkligen bodde här och inte alls ville parkera sina bilar på nån äng en kilometer bort på andra sidan byn när de betalade hyra för garageplats här! Dessutom vimlade det av vilsekommet folk som ville hitta från pendelelandet till mässan eller tvärtom.

Men till själva utställningen: Texas Instruments hade blivit tilldelat en plats längst in bredvid Sinclair och Atari. TI hade ställt upp nio st nittinior i en rad och försett dom med alla möjliga spel. En ständig klunga på cirka femtitalet spelsugna entusiaster stod med ryggen massivt vände mot Atari och prövade sin lycka på Parsec, Munch man, TI Invaders och ett av vår förträff-

lige redaktör BG spel vid namn ROBOT. Nittinian hade utformat fyra st Mästerskapsbrev på pergament för utdelning till dem som fick högsta poäng i respektivet spel. Tyvärr var dels trängseln för häftig för rekordspel och dels tycks någon har tyckt så mycket om våra brev att de har tagit dem med sej. Mer om mästerskapsbrev på annan plats.

På andra sidan TI-ståndet fanns att antal väluniformerade TI personer med uppgift att visa cc-40, den nya kompakta datorn från TI, och PC — inte personlig utan PROFESSORIAL COMPUTER från TI i antagligen trettitegelklassen.

TI-butiksgondolen i blått med en nittinia apterad i den och en fristående komplett uppställning med expansionsbox, drives etc utgjorde resten av TI-montern.

## NITTINIANS T-SHIRTS

Personalen på spelsidan — dvs sidan där TI-99orna stod var klädda i våra (tidningen nittians) trevliga T-Shirts med en cowboy som "hackar på häcken" ut i öknen i Texas. Finns

det andra som vill köpa en sådan till "självkostnadspris = 25 kr + ev moms" så går det bra.

Av övriga utställare hade vi väntat oss en hel del, och kanske just därför blev vi lite besvikna. Men den stora kundkategorin tycks fortfarande vara företag och den enorma anstormning av nyfikna privatpersoner som letade hemdatorer och kringutrustning måste ha funnit alla dessa utbudna konsulttjänster och stora datorsystem för administrativa och konstruktiva uppgifter något förbryllande. Det kommer dock en mässa till i "Hemelektronik" i september och vi får se om man kan hitta lämplig kringdon typ printers, modem och grafikmjuk- och hårdvara då.

Ett intryck är dock bestående — TEXAS GJORDE DET PÅ SOLLENTUNAMÄSSAN och ingen annan kom i närheten. Vi bugar oss för Nils, Gunnar, Peder, Björn, Diella, Martin och de övriga som gjorde det. Vi hörde dessutom att EXPERTEN skall sälja TI-99an nu.





## PROGRAMFÖRMEDLING

NU ÄR DET DAGS FÖR OSS TI99/4-ÄGARE ATT FÅ EN EGEN PROGRAMM-BANK! VI HAR NÄMLIGEN FRÅN DEN BELGISKA FÖRENINGEN TI-SOFT ERHÅLLIT CA 140 PROGRAMM OCH VI HAR NOG NÄSTAN HUNDRA ST TILLGÅNGLIGA HÄR OCKSÅ.

DESSA PROGRAMM ÄR GIVETVIS AV MYCKET BLANDAD KVALITET MEN DET FINNS GODBITAR SOM ORDBEHANDLING, KERMIT, HANGMAN(PÅ SVENSKA), BEETH NOS, MAILING LIST MED MERA.

I STARTSKEDET KOMMER VI EJ ATT SÄLJA NÅGRA PROGRAMM UTAN VI KOMMER ENBART ATT BYTA PROGRAMM. 1 ST INSKICKAT BASIC PROGRAMM GER 3 ST PROGRAMM I RETUR OCH 1 ST ASSEMBLER PROGRAMM GER 6 ST I RETUR. GIVETVIS MÅSTE DE INSKICKADE HÅLLA EN VISS STANDARD OCH DE FÅR EJ HELLER VARA KOPIOR AV PROGRAMM MED COPYRIGHT.

VI RÄKNAR MED ATT I NÄSTA NUMMER AV NITTIONIAN PUBLICERA EN FÖRTECKNING ÖVER TILLGÅNGLIGA PROGRAM TILLSAMMANS MED ETT FORMULÄR ATT ANVÄNDA VID INSÄNDANDE AV PROGRAM. NI ÄR GIVETVIS MYCKET VÄLKOMNA ATT REDAN NU SKICKA IN PROGRAMM.

SOM NÄMT MÅSTE PROGRAMMEN HÅLLA EN VISS STANDARD OCH VI FÖRBEHÅLLER OSS RÄTTEN ATT RETURNERA PROGRAMM EV. MED FÖRSLAG TILL ÄNDRINGAR.

TILL SIST NÅGRA PRAKTISKA TIPS FÖR SJÄLVA PROGRAMM SKRIVANDET. FÖRSÖK ATT SKRIVA PROGRAMMEN SÅ ATT DE EJ KAN "SPÄRA UR" OAVSETT VILKA TANGENTER MAN TRYCKER PÅ OCH FÖRSÖK ATT ANVÄNDA LITE ORGYNELLARE PROGRAMNAMN ÄN SPACE INVADER OCH STARTRECK!

SKICKA ERA PROGRAMM TILL:

NITTIONIAN  
C/O SCHIBLER  
WAHLBERGSG. 6 NB  
121 46 JOHANNESHÖV

MED HÄLSNINGAR  
H. ATTERSJÖE

## PRESENTATION AV MUNCH MAN av Peter Olin

Fyra listiga Hoonos är på jakt efter din Munch Man. Kommer han att hinna till ett energipiller för att vända attacken eller kommer han att bli uppäten av en Hoono!

Så inleds manualen till en av TI:s nya spelmoduler — Munch Man, som garanterat kommer att ge dig många roliga timmar framför din dator. När du har pluggat in modulen och valt Munch Man visas tittelskärmen, som består av ordet "Munch Man" skrivet med bokstäver uppbyggda av de figurer som finns med i spelet. För att starta spelet trycker man på vilken tangent som helst och då uppenbarar sig spelplanen.

I mitten av spelplanen finns Munch Mans bo och Vid hörnen finns fyra celler som de farsansfulla Hoonos bor i. En Hoono är ett slags monster som definitivt är ovänligt inställda till din Munch Man. De fyra Hoonos har olika färger: en röd, en lila, en blå och en gul. De har också olika intelligensnivå — den röda är smartast och den gula är dummast.

Själva spelplanen består av en labyrinth där kampen pågår. Munch Man lägger ut en kedja efter sig där han grå fram i labyrinthen. När hela labyrinthen är sammankopplad av en enda sammanhängande kedja kommer en ny labyrinth, likadan som den förra. Men nu är de fyra Hoonos snabbare och har antagit en annan skepnad.

Man kan få poäng på följande sätt:

1. För Varje länk i kedjan du lägger ut får du 10 poäng.
2. När Du har ätit ett av de fyra energipillren (70 poäng) som finns längst ut i hörnen av



labyrinth blir Munch Man snabbare och Hoonos blir svarta. Nu är rollerna ombytta och Munch Man kan äta Hoonos som nu flyr för livet. För den första Hoonon som man äter får man 100 poäng, för den andra 200 poäng, den tredje 400 poäng och för den fjärde 800 poäng.

Om man fångar alla Hoonos på alla energipiller kan man få maximalt 8700 poäng per labyrinth. När energin från pillret börjar ta slut blinkar Hoonos i sina normala färger och när labyrinthen blinkar rött är ordningen återställd och du måste åter akta dig för Hoonos.

Om Munch Man blir fångad av en Hoono förlorar han ett av de tre liv han hade från början. När han mister sitt sista liv är spelet slut och eventuell High Score registreras. man kan emellertid utöka antalet liv genom att få många poäng. Var tiotusenden poäng får man nämligen ett extra liv.

Munch Man kräver ingen kringutrustning men jag rekommenderar joystick och tror att den är nödvändig för att uppnå bra resultat.

## MUNCH MAN — PAC MAN

För dem som inte vet vad PAC MAN är ger jag en kort beskrivning.

PAC MAN är en spelautomat från ATARI som Munch Man och många andra liknande spel bygger på. I mitten av en labyrinth finns ett bo som innehåller 4 spöken motsvarande Munch Mans Hoonos. I hörnen av labyrinthen finns energipil-



ler. Gångarna i PAC MAN är fyllda med prickar som man skall äta till skillnad från Munch Man där man lägger ut en kedja. Men grundidén är densamma: åk igenom hela labyrinthen men se upp för monstren.

Under spökenas bo uppenbarar sig två gånger per labyrinth en bonusfrukt som ändras efter varje labyrinth (körsbär, jordgubbe, persika, äpple etc). Är man tillräckligt skicklig kommer andra föremål än frukter också.

I PAC MAN är bonusfrukten en sporre att rensa labyrinthen eftersom man vill se vilken nästa bonusfrukt är. (Man ser verkligen vad det är för frukt; grafiken är nämligen mycket bra.) Det finns en liknande sporre i Munch Man eftersom man hela tiden vill se vilken form Hoonos har nästa gång. TI får förstås, men jag gillar bonusfrukten bättre.

PAC MAN har ytterligare en sporre: med oregelbundna mellanrum mellan labyrintherna ges en liten uppvisning med fin grafik.

Speltiden på PAC MAN är dock kortare, man har tre liv från början och får ett nytt vid 10.000 poäng men sedan är det stopp. I Munch Man får man som bekant ett nytt liv vid var tiotusende poäng.

PAC MAN kostar förresten 2 kr per spel.

Mitt slutgiltiga omdöme är att Munch Man är ett roligt spel och leder klart vid en jämförelse med VIC-20:s VIC MEN och PAC MAN till ATARI VCS TV-spel.



Men om jag hade obegränsat med enkronor skulle jag nog spela på PAC MAN automaten.

## TESTMOD

Till Munch Man finns också en testmod som inte är dokumenterad i manualen. För att komma in i denna krävs att du trycker inom tre sekunder efter att titelskärmen har kommit fram.

Väl inne i testmod skriver datorn:

## RND(0-2)

Svara med 0,1 eller 2.0 ger relativt snälla och långsamma Hoonos medan 2 ger mycket smarta och snabba Hoonos.

Nästa fråga gäller vilken labyrint du vill ha.

## SCN(0-19)

Svara med 0 och du får labyrint nr 1, med 19 och du får labyrint nr 20. Det är möjligt att välja vilken som helst mellan 11 och 20.

Den sista frågan gäller hur många liv Munch Man skall ha från början.

## MM(1-9)

Där kan du välja vilket antal som helst mellan ett och nio. Därefter kan man börja spela på vanligt sätt.

(Anledningen till att detta testmod finns är att TI använde det vid inprovningen av Munch Man och att de av någon anledning inte tog bort det från den färdiga modulen. Reds. anm.)

Jag tycker personligen att man inte skall använda testmoden innan man lyckats klara de 20 första labyrinterna eftersom det tar bort en del av spänningen. Man är nämligen ganska nyfiken på hur Hoonos ser ut i nästa labyrint vilket gör att man kämpar lite extra.

## MINI-ADVENTURE

Nittinian presenterar: MINI-ADVENTURE. Ger dig följande unika fördelar:

- Kan köras i både TI BASIC och Extended BASIC.
- Inga långa listningar att knäppa in.
- Mycket snabbt program — inga långa väntetider.
- Roar familjen i timmar.<sup>1</sup>
- MINI-ADVENTURE förstår alla kommandon som du matar in.
- På begäran tar vi fram versioner för andra språk som t ex 9900-assembler, LOGO, Pascal, FORTH m fl. Vi kan även ordna versioner för andra datorer.
- Ett otroligt lågt pris.<sup>2</sup>

Om du är medlem i Föreningen 99:an gör du på följande sätt för att få MINI-ADVENTURE. Skriv namn, adress, vilken dator du äger, önskat programspråk på en lapp. Stoppa denna i ett kuvert. Slå sedan upp sidan XX i denna tidning så hittar du listningen till MINI-ADVENTURE.

<sup>1</sup> Om familjen har en IK under 30.

<sup>2</sup> Gratis för medlemmar av Föreningen 99:an.

## HUR MYCKET MINNE HAR JAG KVAR?

Har man Extended BASIC och vill ha reda på hur mycket minne man har kvar skriver man bara SIZE. Detta kommando finns tyvärr inte i TI BASIC. I stället kan man mata in följande rader:

1 A = A + 8

2 GOSUB 1

och skriva RUN. Efter en stund visas

\*MEMORY FULL IN 1

Skriv nu PRINT A så visas hur många bytes som är kvar av minnet.

Du bör inte ha radnummren 1 och 2 i ditt program när du använder det här tricket eftersom de då skrivs över. När du vet hur mycket du har kvar tar du bort rad 1 och 2 och kan åter köra ditt program.

## STUDSAND SPRITE

Följande enkla program skapar en sprite som studsar omkring på skärmen:

```
100 CALL CLEAR :: CALL COLOR(2,6,6):: PRINT RPT$("+",252);RPT$
 ("+",252);RPT$("+",56):: R=40
 :: C=30 :: CALL SPRITE(1,42,2,25,17,R,C)
110 FOR K=1 TO 900 :: CALL POSITI
 ON(1,Y,X):: R=R+80*((Y+R)/200)
 -(Y+R/1):: C=C+60*((X+C)/250)
 -(X+C/1):: CALL MOTION(1,R,C):: NEXT K
```

Craig Miller  
San Dimas, CA

## SIDAN XX

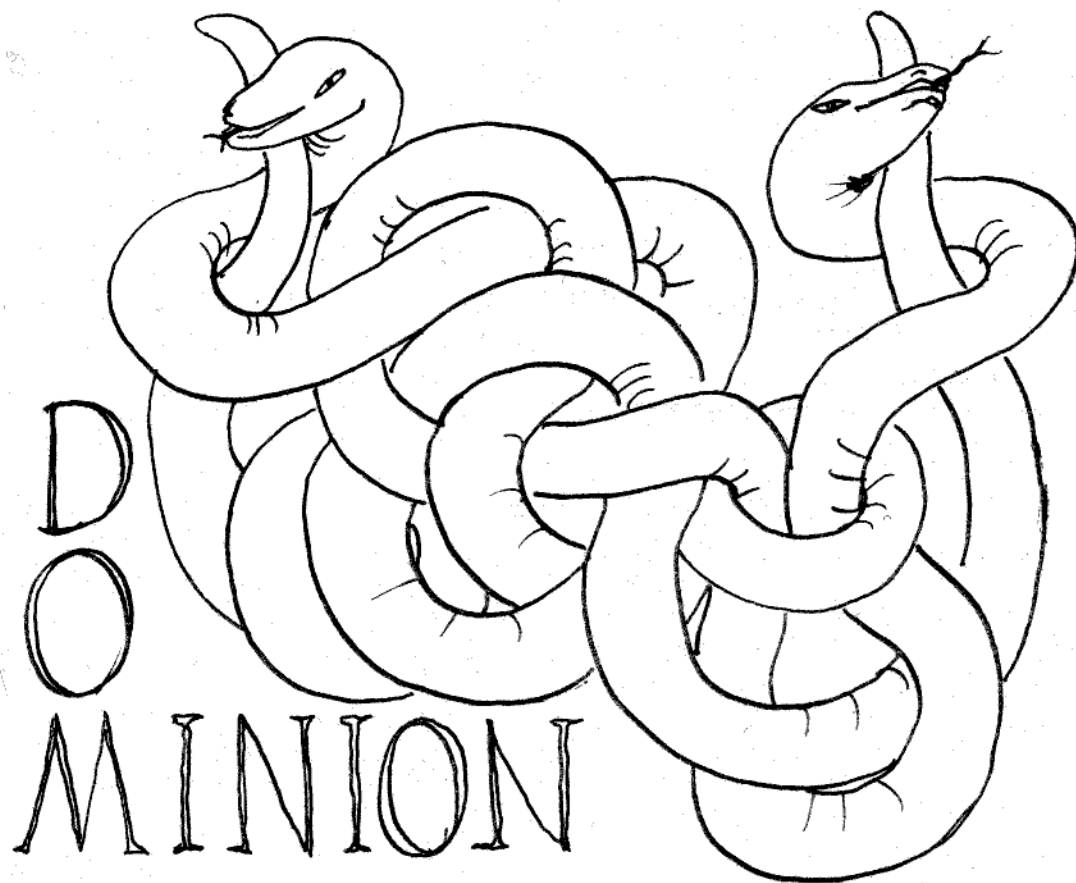
100 CALL CLEAR

110 PRINT: "DU BEFINNER DIG I EN GROTTA."

120 PRINT "I VILKEN RIKTNING VILL DU": "FLYTTA":

130 INPUT AS

140 GOTO 110



Dominion är ett spel för två personer med var sin joystick. De får var sin orm som slingrar fram och försöker mota in motståndaren i ett hörn. För att komma ut kan man skjuta sej en väg genom motståndaren. Inget för dem med ormmardrömmar!

```

100 REM *****
110 REM ** DOMINION **
120 REM *****
130 REM
140 REM AV PETER OLIN
150 REM
160 CALL CLEAR
170 CALL SCREEN(7)
180 REM
190 REM ** VIT SPELARE **
200 CALL COLOR(9,16,1)
210 CALL CHAR(96,"FFFFFFFFFFFF
FF")
220 REM
230 REM ** SVART SPELARE **
240 CALL COLOR(10,2,1)
250 CALL CHAR(104,"FFFFFFFFFFFF
FFF")
260 REM
270 REM ** BLI VIGGAR **
280 CALL COLOR(11,5,1)
290 CALL CHAR(112,"FFFFFFFFFFFF
FFF")
300 REM
310 REM
320 REM ** RITA RAM **
330 CALL HCHAR(1,2,112,31)
340 CALL VCHAR(1,2,112,24)
350 CALL VCHAR(1,32,112,24)
360 REM ** SS=SVARTS SKOTT
SV=VITS
VARIERAR MELLAN
0 DCH 5
370 SS=INT(RND*6)
380 SV=INT(RND*6)
390 REM ** STARTPOSITIONER
RPS,KPS=SVARTS
RAD&KOLUMN
RPV,KPV=VITS

```

```

KOLUMN
400 RPS=10
410 KPS=27
420 RPV=11
430 KPV=7
440 REM ** STARTRIKTNINGAR
XVI,YVI=VITS
HASTIGHETER I
X- RESP.Y-LED
450 REM ** XSI,YSI=SVARTS
HASTIGHETER I
X- RESP. Y-LED
460 YVI=0
470 XVI=1
480 XSI=-1
490 YSI=0
500 REM ** B\NJA SPELET **
510 REM
R \ D
520 CALL JOYST(1,XV,YV)
530 IF (XV=0)*(YV=0) THEN 570
540 REM IF (XV<0)*(YV<0) THEN
470
550 XVI=XV/4
560 YVI=YV/-4
570 RPV=RPV+YVI
580 KPV=KPV+XVI
590 CALL KEY(1,K,S)
600 IF (SV=0)*(K)-1 THEN 970
610 CALL GCHAR(RPV,KPV,DOT1)
620 CALL VCHAR(RPV,KPV,96)
630 IF DOT1<32 THEN 780
640 REM
S V A R T
650 CALL JOYST(2,XS,YS)
660 IF (XS=0)*(YS=0) THEN 700
670 REM IF (XS<0)*(YS<0) THEN
580
680 XSI=XS/4
690 YSI=YS/-4

```

```

700 RPS=RPS+YSI
710 KPS=KPS+XSI
720 CALL KEY(2,K1,S1)
730 IF (SS=0)*(K1)-1 THEN 1010
740 CALL GCHAR(RPS,KPS,DOT2)
750 CALL VCHAR(RPS,KPS,104)
760 IF DOT2<32 THEN 850
770 GOTO 520
780 REM ** C R A S H **
790 PTSS=PTSS+1
800 FOR I=1 TO 30
810 CALL VCHAR(RPV,KPV,112)
820 CALL VCHAR(RPV,KPV,32)
830 NEXT I
840 GOTO 900
850 PTSV=PTSV+1
860 FOR I=1 TO 30
870 CALL VCHAR(RPS,KPS,104)
880 CALL VCHAR(RPS,KPS,112)
890 NEXT I
900 REM
910 PRINT "VIT "IPTS,"SVART "IP
TSS
920 IF (PTSV=5)+(PTSS=5) THEN 104
0
930 CALL KEY(1,A,B)
940 CALL KEY(2,C,D)
950 IF (A)-1*(C)-1 THEN 160 ELS
E 930
960 REM ** VIT SKJUTER **
970 CALL HCHAR(RPV,KPV,32)
980 SV=SV-1
990 GOTO 610
1000 REM ** SVART SKJUTER **
1010 CALL HCHAR(RPS,KPS,32)
1020 SS=SS-1
1030 GOTO 740
1040 END

```

# ONE CHECK

I vårt förste nummer visade vi hur man översatte från microsoft till ti-basic med David Ahls spel One-Check som exempel. Här följer Rogers förslag med lite grafik och omarbetning. För dej som vill spela - varsågod! Och för dej som vill ha lite grafiskt pyssel - här är ett läckert brädspel som är mycket tacksam för grafisk behandling ur den högre skolan.

```
10 CALL CLEAR
20 PRINT TAB(12);"ONE CHECK"
30 PRINT TAB(8);"CREATIVE COMPUT
ING"
40 PRINT ::
50 DIM A(64)
60 DIM B$(8,8)
70 PRINT ::
80 PRINT "48 CHECKERS ARE PLACED
ON THE OUTSIDE TWO SPACES OF
A 64 SPACE CHESSBOARD."
90 PRINT "REMOVE AS MANY CHECKER
S AS POSSIBLE BY DIAGONAL JUMPS
."
100 FOR I=1 TO 8
110 FOR J=1 TO 8
120 B$(I,J)="ee"
130 NEXT J
140 NEXT I
150 PRINT "WHEN YOU HAVE NO POSS
IBLE JUMPS LEFT INPUT 0,0 "
160 PRINT ::
170 PRINT
180 GOTO 350
185 REM TRYCKER SPELPLANET
190 C=1
195 GOSUB 710
200 PRINT TAB(6);"1 2 3 4 5 6 7
8"
210 FOR I=1 TO 8
220 PRINT TAB(2);C;
```

```
230 C=C+1
240 FOR J=1 TO 8
250 PRINT B$(I,J);
260 NEXT J
270 PRINT
280 PRINT TAB(5);
290 FOR J=1 TO 8
300 PRINT B$(I,J);
310 NEXT J
320 PRINT :
330 NEXT I
340 RETURN
350 FOR I=7 TO 8
360 FOR J=1 TO 8
370 B$(I,J)="mn"
380 NEXT J
390 NEXT I
400 FOR J=1 TO 2
410 FOR I=1 TO 8
420 B$(I,J)="mn"
430 NEXT I
440 NEXT J
450 FOR J=7 TO 8
460 FOR I=1 TO 8
470 B$(I,J)="mn"
480 NEXT I
490 NEXT J
500 FOR I=1 TO 2
510 FOR J=1 TO 8
520 B$(I,J)="mn"
530 NEXT J
```

```
540 NEXT I
550 GOSUB 190
560 INPUT "JUMP FROM " :K,L
565 IF ((K=0)*(L=0)) THEN 800
570 IF B$(K,L)="mn" THEN 600
580 PRINT "CAN'T DO THAT"
590 GOTO 560
600 INPUT "TO " :O,P
610 IF B$(O,P)="ee" THEN 640
620 PRINT "CAN'T DO THAT "
630 GOTO 600
640 IF ((O=K+2)+(O=K-2)+(P=L+2)+(
P=L-2)) THEN 660
650 GOTO 620
660 B$(K,L)="ee"
670 B$(O,P)="mn"
680 B$((K+O)/2,(L+P)/2)="ee"
690 GOSUB 190
700 R=R+1
705 GOTO 560
710 CALL CHAR(101,"")
720 CALL CHAR(109,"FFFFFFFFFFFF
FFF")
730 CALL CHAR(110,"FFFFFFFFFFFF
FFF")
740 CALL COLOR(10,7,6)
750 CALL COLOR(9,6,6)
760 RETURN
790 GOTO 560
800 PRINT "YOU HAVE "148-R;"COUN
TERS LEFT"
1000 STOP
```

## DATAKORSORDET

```
100 REM *****
101 REM * KRYSS *
102 REM *AV YVONNE & ROGER*
103 REM *****
111 REM : MATA ENDAST IN RADER
460 - 600 SOM PRINTAR KRYSSET.
LYCKA TILL.
130 DIM H$(9,12)
140 DIM V$(9,12)
150 H$(1,6)="SPELARENS DRAG"
160 H$(2,3)="SOPA RENT"
170 H$(2,8)="FLOTT BIL"
180 H$(3,2)="BAS SEXTON"
190 H$(3,8)=CHR$(73+85)
200 H$(3,6)="VATTENDRAG"
210 H$(4,2)="TI NYTT, HALLELUJAH
"
220 H$(5,1)="EXTRA SOM SAKTA"
230 H$(5,6)="STIG IN I DATORN"
240 H$(6,4)="GENITIVPREPOSITIONE
N"
250 H$(7,1)="INLEDNING TILL SLIN
GA"
260 H$(7,5)="SJUNDE"
270 H$(7,10)="VINKELFUNKTION"
280 H$(8,2)="KNALLEBY"
290 H$(8,8)="LOGISK +"
```

```
300 H$(9,3)="RECENSERAS I NR 2"
310 V$(1,8)="HARD COPY"
320 V$(1,9)="KATASTROFAL"
330 V$(2,3)="HOPP I SLINGA"
340 V$(2,4)="EXTENDED T EX"
350 V$(3,2)="KOMPLETT"
360 V$(3,6)="TITTA"
370 V$(4,7)="MOT I"
380 V$(4,10)="VAGNMATSAL"
390 V$(7,2)="OBEHINDRAD"
400 V$(7,3)="READ ONLY MEMORY"
410 V$(7,6)="NRS 1 IDENTITET"
420 V$(6,5)="MYSPLATSEN"
430 V$(8,4)=CHR$(82+85)
440 V$(8,8)="IF IN SWEDEN"
450 V$(8,9)=CHR$(82+65)
460 FOR I=1 TO 9
465 DIM A$(9,12)
470 FOR J=1 TO 12
480 READ A$(I,J)
484 NEXT J
486 NEXT I
490 DATA a,a,a,a,a,a,b,b,b,b,a,a
a,a,b,b,b,b,a,a,b,b,a,a,a,a,b,b,b
a,b,a,b,b,b,a,a,a
500 DATA a,b,b,b,b,b,b,b,b,b,b,b
b,b,b,b,a,a,b,b,b,b,b,b,a,a,a,b
b,a,a,a,b,b,a,a,b
510 DATA b,b,b,a,b,b,b,b,a,a,b,b,b
```

```
a,a,b,b,b,b,b,b,a,b,b,b,a,a,b,b,b
a,b,b,b,b,b,b,a,a
520 CALL CHAR(97,"FFFFFFFFFFFFFF
FF")
530 CALL CHAR(98,"FF818181818181
FF")
540 CALL CHAR(99,"00FE02020702")
550 A$(5,4)=CHR$(99)
560 A$(4,12)=CHR$(99)
570 PRINT TAB(6);"123456789012"
580 C=1
590 FOR I=1 TO 9
600 PRINT TAB(3);C;
610 FOR J=1 TO 12
620 PRINT A$(I,J);
630 NEXT J
640 C=C+1
650 NEXT I
660 INPUT "VILL DU SE SVARET INF
UT J":B$
670 IF B$="J" THEN 460
680 STOP
690 DATA a,a,a,a,a,I,N,P,U,T,a,a
a,a,N,E,U,a,a,R,R,a,a,a,a,H,E,X
a,I,a,I,U,a,a,a
700 DATA a,E,X,I,T,A,S,I,N,S,T,R,U
U,L,T,R,a,E,N,T,E,R,a,M,a,a,a,A
,V,a,a,a,L,a,a,E
710 DATA F,O,R,a,R,A,D,a,a,S,I,N
a,B,O,R,I,S,a,O,R,a,a,T,a,a,M,U
N,C,H,M,A,N,a,a
```



## 99:ANS MÅNGA SPRÅK

av Björn Gustavsson

En dator är en fantastisk apparat. Den kan göra beräkningar, vara till hjälp i utbildning och ge avkoppling med fantastiska spel. Men allt detta förutsätter att man på något sätt får kontakt med datorn, kan kommunicera med den.

Det finns tre möjligheter:

1. Användaren måste lära sig datorns eget språk, bestående av "ettor och nollor".
2. Datorn måste lära sig användarens språk, dvs svenska, engelska eller något annat språk.
3. Användaren och datorn lär sig båda ett mellanliggande språk.

Man säger att ju närmare ett språk liknar datorns, ju lägre är dess nivå. Och tvärtom, ju mer det liknar människans, ju högre är dess nivå.

### MASKINSPRÅK

Låt oss först titta på den lägsta nivån av språk — kallat maskinspråk. Eftersom elektricitet kan antingen vara av eller på kan maskinspråk konstrueras med bara två "ord". Människor brukar representera maskinspråk med siffrorna 0 och 1, där 1 betyder "ström på" och 0 "ström av".

I figur 1 hittar du ett exempel på maskinspråk. Det är inte särskilt lätt att förstå för en människa, eller hur? Fördelen är att man har perfekt kontroll över maskinen eftersom ingenting går förlorat i någon översättning.

### ASSEMBLERSPRÅK

Eftersom människor har så svårt att använda maskinspråk har språk på högre nivå utvecklats. Nästa nivå är assemblyspråk. Man slår ihop siffrorna i maskinspråket i grupper och ger dessa alfabetiska namn som är lite lättare att komma ihåg. För att datorn skall kunna förstå assemblyspråk måste det först översättas till maskinspråk. Denna översättning kan datorn klara själv om man har givit den ett program som kallas assembler. Detta program, skrivet i maskinspråket, översätter program i assembly till maskinspråk.

Ett exempel på assembly finns i figur 2.

Assembler:

```

SUBMS DEF START
START EQU %820
 LMP1 SUBMS
 LI R2, REGLD
 BL 2, ADDR5
 LI R0, %1800
 LI R1, %2000
 BLMP %V65M
 LI R0, %1800
 CLR R2
 CLR R1
 LOOP1
 LOOP2
 SWPB R1
 GET OUR WORKSPACE
 LOAD VDP REGISTERS
 BEGINNING OF SPRITE ATTR. LIST
 END OF SAB
 BEGINNING ADDR OF SCREEN IMAGE TABLE
```

Maskinspråk:

```

00000010
11100000
10000011
00100000
00100010
00100010
01111111
01110100
00000110
10100000
01111100
11010110
```



### HÖGNIVÅSPRÅK

Även assemblyspråk är svårt för en människa att arbeta med. Eftersom varje instruktion i ett assemblyprogram svarar mot en instruktion i maskinspråk behövs väldigt många instruktioner och programmeraren måste hålla reda på massor med detaljer som är oväsentligt för hans problem. Man försöker därför att i stället använda högnivåspråk som ligger närmare mänskliga språk.

I högnivåspråk motsvarar vanligtvis ett ord mängder med instruktioner i maskinspråk. Eftersom varje ord är mycket kraftfullare blir programmen kortare och programmeraren kan ägna sig åt det väsentliga i problemet.

För att en dator skall förstå ett högnivåspråk måste ett särskilt program finnas som översätter språket till maskinspråk.

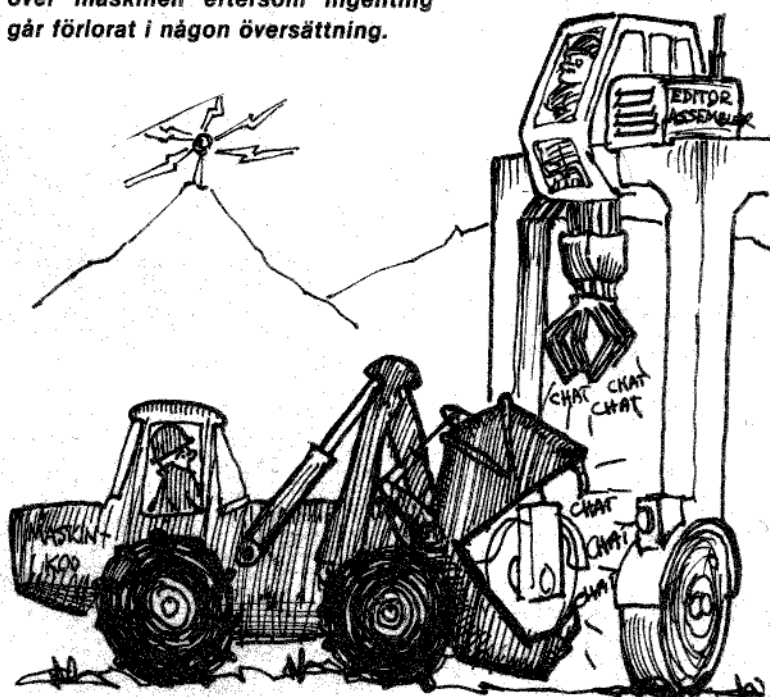
### KOMPILERING OCH INTERPRETERING

Översättning från högnivåspråk till maskinspråk kan göras på två sätt.

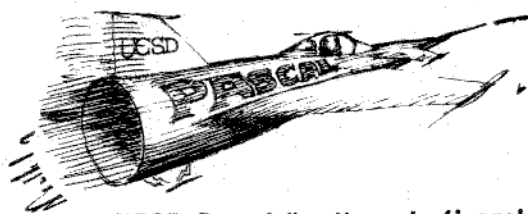
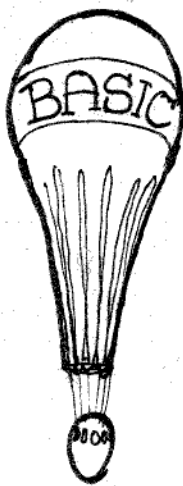
Om man översätter hela programmet på en gång till maskinspråk är det ett "kompilerat" språk. Programmet som gör översättningen kallas "kompilator". Resultatet av kompileringen kan sparas och köras hur många gånger som helst.

Om datorn däremot utför varje instruktion så fort den är översatt är språket ett "interpreterande" eller "tolkat" språk. Därför måste programmet översättas varje gång det körs. Därför är kompilerade språk mycket snabbare än tolkade språk.

Men även tolkade språk har sina fördelar. De är vanligen lättare att använda. Så fort alla instruktioner är inmatade kan vi köra programmet utan att först behöva kompilera det (vilket kan ta flera minuter). Dessutom kan datorn tala om för oss på en gång när vi matar in en rad att den är felaktig.



99 an



TI BASIC är det tolkade högnivåspråk som alltid finns i TI-99/4A. Det är lätt att lära sig och använda.

#### TI EXTENDED BASIC

Man får Extended BASIC till 99:an genom att stoppa in en modul. Språket liknar TI BASIC och kan göra allt som det kan plus mer. Se presentationen i detta nummer av Nittinian.



TI LOGO är ett annat tolkat språk. Det är mycket lämpligt för utbildning eftersom det är mycket lätt att lära sig. Grafiken hanteras genom att man styr en "sköldpadda" som är doppad i bläck och lämnar ett språk efter sig på skärmen. Figur 3 visar ett exempel.

LOGO hör till de språk som kan utökas med nya instruktioner. Om man behöver ett ord som ritar ett hus talar man helt om hur det skall gå till med hjälp av de ord som redan finns. Detta nya ord kan sedan användas närhelst man behöver ett hus.

#### PROCEDURES

```
TO PENTAGON :STORLEK
TELL TURTLE
REPEAT 5 [FD :STORLEK RT 72]
END
```

```
TO SQUIRAL :ANGLE
TELL TURTLE CS
PENUP FORWARD 29 PENDOWN
SQ :ANGLE 5
END
```

UCSD Pascal är ett av de få språk som är kompilerande på hemdatorn. Versionen till 99:an innehåller funktioner som ger programmeraren nästan lika bra kontroll över datorn som assemblerspråk. Pascal är mycket lämpligt för stora, komplicerade ekonomiska eller tekniska program. Eftersom det är ett kompilerat språk är det betydligt snabbare än BASIC.

ASIC är ett tolkat språk och dess interpretator är skriven i TI BASIC! Alla ägare av grundenheten kan alltså använda detta språk. (Listning och beskrivning av språket kommer i Nittinian.) Grafiken är lättare att använda i

#### SAMMANSTÄLLNING UTRUST. NINGSBEHOV OCH PRISER

Sammanställningen nedan ger utrustningen som behövs för de olika språken och cirkapriser. Priserna kan naturligtvis ändra sig snabbt. När minneskort som kan användas utan utbyggnadslåda blir tillgängliga i Sverige sjunker priset för bl a TI LOGO med några tusenlappar.

Pris för kassettbandspelare och TV eller monitor är ej medräknade i priserna.

#### TI BASIC

Kassettkabel. Pris: 3144.

#### TI EXTENDED BASIC

Kassettkabel, modulen "Extended BASIC". Pris: 4139.

#### TI LOGO

Kassettkabel, minneskort 32K, utbyggnadslåda, modulen "TI LOGO". Pris: 9869.

#### UCSD PASCAL, FÖR ATT KÖRA PROGRAM

Kassettkabel, minneskort 32K, utbyggnadslåda, P-kod-kort. Pris: 10824.

#### UCSD PASCAL, FÖR ATT SKRIVA EGNA PROGRAM

Minneskort 32K, utbyggnadslåda,

ASIC än i BASIC. Nackdelen är främst att det är långsamt, eftersom det är tolkat. (Språket tolkas ju av BASIC som i sin tur tolkas.)

Eftersom ASICs interpretator är skriven i BASIC är det lätt att göra egna ändringar, och den finns även i en svensk version!

#### TI FORTH

Ett fantastiskt språk som inte liknar något annat. Dess främsta fördelar är att det kräver lite minne, är snabbt och tillåter lättlästa program. Med FORTH har man samma kontroll över datorn som med assembler, men eftersom FORTH är ett högnivåspråk är det förstås lättare att använda. FORTH är både kompilerande och interpreterande. Se mer om FORTH i en artikel i detta nummer.

skrivminne, P-kod-kort, kompilator, editor, filhanteringssystem på diskett. Pris: 18925.

#### ASSEMBLER, SKRIVA KORTARE PROGRAM

Kassettkabel, Mini Memory. (Med denna utrustning kan du skriva kortare assemblerprogram samt köra assemblerprogram kortare än 4K.) Pris: 4339.

#### ASSEMBLER, KÖRA LÄNGRE PROGRAM

Kassettkabel, minneskort 32K, utbyggnadslåda, en av följande: Mini Memory, Extended BASIC eller Editor/Assembler moduler. Pris: 8869.

#### ASSEMBLER, SKRIVA LÄNGRE PROGRAM

Minneskort 32K, utbyggnadslåda, skrivminne, Editor/Assembler modulen. Pris: 15325.

#### FORTH

Minneskort 32K, utbyggnadslåda, skrivminne, Editor/Assembler modulen, diskett med FORTH. Kan ännu ej köpas i Sverige. Pris: 15325 + pris för disketten med FORTH.

#### ASIC

Kassettkabel. Pris: 3144.

# S många språk



# EXTENDED

## PRESENTATION AV TI EXTENDED BASIC

av Björn Gustavsson

Det programspråk som de flesta först kompletterar sin 99:a med är antagligen **TI EXTENDED BASIC**. Vi skall här ta en titt på detta språk och beskriva de utökningar som har gjorts jämfört med **TI BASIC**.

Det finns ca 40 nya eller utökade kommandon, instruktioner, funktioner och subprogram (som anropas med CALL). Det finns också en del "små" förbättringar som gör språket lättare att arbeta med.

### "SMÅ" FÖRBÄTTRINGAR

Det går t ex att skriva flera instruktioner på samma rad, vilket spar minne och ger snabbare programkörning. Instruktionerna separeras med ::. Exempel:

```
10 FOR I=1 TO 10 :: A(I)=0 :: NEXT I
```

En mycket viktig förbättring är också att instruktioner kan följa THEN i en IF-sats:

```
10 IF A=B THEN PRINT "LIKA" ELSE PRINT "OLIKA"
```

### IN- OCH UTMATNING

INPUT och PRINT finns kvar förstås. Men nu finns också LINPUT, ACCEPT, DISPLAY, PRINT USING och DISPLAY USING.

LINPUT är en version av INPUT som läser en hel rad åt gången till en strängvariabel. Alla tecken, inklusive komma, kan då ingå i strängen.

ACCEPT läser liksom INPUT tecken från tangentbordet, men man har här möjlighet att bestämma:

- Var på skärmen tecknen skall matas in.
- Vilka sorts tecken som får matas in, t ex bara stora bokstäver, siffror eller bokstäverna J, Y och N.
- Om ljudsignal skall ges före inmatning.
- Om skärmen skall raderas före inmatning.
- Hur många tecken som får matas in och om raden skall blankas före inmatning.

DISPLAY skriver liksom PRINT ut en text men man kan här bestämma var texten skall skrivas, om ljudsignal skall ges, om skärmen skall blankas helt eller delvis före utskrift.

DISPLAY USING och PRINT USING är lämpliga t ex när man skriver tabeller. De gör det möjligt att bestämma bl a hur många siffror tal skall skrivas ut med och få alla decimalpunkter under varandra.

### EGNA SUBPROGRAM

I EXTENDED BASIC (och även i TI BASIC) finns många inbyggda subprogram som anropas med CALL. Exempel är CLEAR, COLOR, HCHAR etc. Men det går även att definiera egna subprogram. Dessa inleds med SUB <namn> och eventuellt en parameterlista inom parenteser.

Ett exempel på ett subprogram är följande som beräknar fakultet:

```
2150 SUB FAKULTET(N,A)
2160 A=1
2170 FOR I=2 TO N :: A=I*A :: NEXT I
2180 SUBEND
```

(Fakulteten för ett tal n beräknas som  $1*2*3*...*n$ . Exempelvis är 5-fakultet  $1*2*3*4*5=120$ .)

Subprogrammet anropas med CALL FAKULTET(<uttryck>,<variabel>). Programmet beräknar då fakultet för <uttryck> och lämnar resultatet i <variabel>.

Exempel:

```
100 CALL FAKULTET(5,R)
110 PRINT "5-FAKULTET =" ; R
RUN
5-FAKULTET = 120
```

Alla variabler som används i subprogrammet är lokala. Detta innebär att samma variabelnamn kan användas i huvudprogrammet eller i andra subprogram utan att påverka varandra. Variabelnamnen A och N kan alltså användas utan problem i huvudprogrammet som anropar FAKULTET.



# ENDED BASIC

Detta innebär att man kan lagra ofta använda subprogram på diskett och foga ihop det med andra program utan att behöva tänka på variabelnamnen, vilket man ju måste när man använder vanliga subrutiner (GOSUB - RETURN). Andra fördelar är ju att CALL FAKULTET säger mycket mer än t ex GOSUB 2150 och alltså gör programmet lättare att läsa.

## PROGRAMLAGRING

RUN kan användas som en instruktion på formen RUN "<filnamn>". Följande rad i ett program

```
2030 RUN "DSK1.HANGMAN"
```

hämtar programmet HANGMAN i skivenhet 1 och kör det. (RUN "CSI" fungerar också.) Detta gör det möjligt att bryta upp stora program i små delprogram som automatiskt läser in nästa del.

En liten detalj som retat en del personer är att filnamnet efter RUN måste vara en strängkonstant, ej en strängvariabel! Detta sätter effektivt en käpp i hjulet på försök att skriva ett allmänt program som visar vilka BASIC-program som finns på en disk och låter en välja vilket man vill köra.

En annan intressant nyhet om man har skivenhet är automatstart av program. När EXTENDED BASIC startas upp söker den alltid efter filen LOAD i skivenhet 1. Om den hittas hämtas programmet i den och körs.

Om man har skivenhet går det också att skarva ihop BASIC-programmet i datorn med ett program från skivan. På det sättet kan man lägga till vanliga subprogram till ett program. Tyvärr fungerar detta inte med kassetbandspelare!

Slutligen vad beträffar programlagring kan man listsky i ett program. Detta fungerar med både kasset och diskett. Programmet går att köra, men kan inte listas, ändras eller sparas på nytt.

## SPRITES

Sprites (svenska "andeväsen", "sagoväsen") är grafiska figurer som rör sig med jämna rörelser över skärmen. Så fort en sprite har satts i rörelse av ett program fortsätter den att röra sig tills programmet ändrar rörelsen. En sprite kan passera över andra tecken på skärmen. Ett fantastiskt hjälpmedel i spelprogram!

För att manipulera sprites finns följande subprogram:

|       |                                                                                |
|-------|--------------------------------------------------------------------------------|
| CHAR  | Bestämmer hur ett tecken som skall användas för en sprite skall se ut.         |
| COINC | Kontrollerar om två sprites eller en sprite och punkt på skärmen sammanfaller. |
| COLOR | Bestämmer färgen för en sprite.                                                |

DELSprite Tar bort sprites.

DISTANCE Ger avståndet mellan två sprites eller en sprite och en punkt.

LOCATE Flyttar en sprite till en viss punkt på skärmen.

MAGNIFY Ändrar storleken på alla sprites.

MOTION Bestämmer en sprites hastighet.

PATTERN Bestämmer vilket tecken som en sprite skall bestå av.

POSITION Tar reda på var på skärmen en sprite är.

SPRITE Skapar en sprite, bestämmer hur den ser ut, dess färg, position och rörelse.

## FELHANTERING

EXTENDED BASIC innehåller instruktioner som gör det möjligt för ett program att självt hantera fel som normalt skulle stoppa det med ett felmeddelande. Dessa är: CALL ERR, ON ERROR, ON WARNING och RETURN.

Det går dessutom att hantera brytpunkter inklusive en åstadkommen genom FCTN 4 från tangentbord. Ett spelprogram kan man alltså göra omöjligt att bryta från tangentbordet, vilket kan vara roligt ibland.

## ASSEMBLERPROGRAM

Assemblerprogram kan laddas och köras från EXTENDED BASIC om minnesexpansion finns. Subprogrammen INIT, LOAD, LINK och PEEK används för detta ändamål. För att däremot skriva assemblerprogram behövs Mini Memory eller Editor/Assembler.

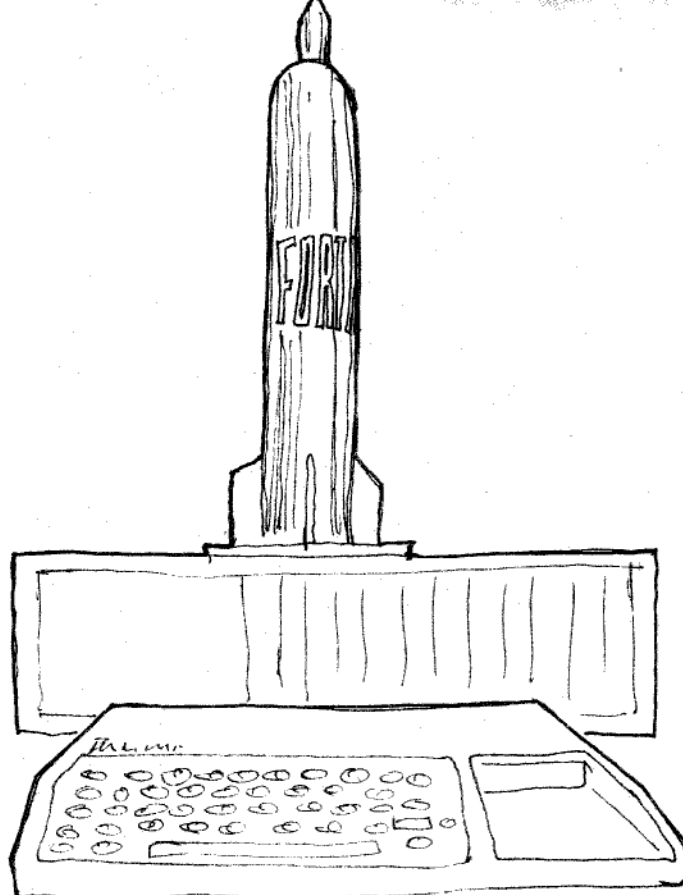
## HASTIGHET

Om man inte använder sprites är nyare versioner av EXTENDED BASIC (bör gälla samtliga som sålts i Sverige) ca 40 procent snabbare än TI BASIC. Den äldre versionen är något långsammare än TI BASIC.

## SLUTSUMMERING

Således innehåller EXTENDED BASIC en hel del användbara nyheter som gör språket mycket kraftfullt, även jämfört med många gånger dyrare datorer. Det är främst två saker som jag önskar mig:

1. Att TI hade gjort EXTENDED BASIC betydligt snabbare.
2. Att de hade byggt in EXTENDED BASIC i stället för TI BASIC i 99:an från början.



## FORTH — FJÄRDE GENERATIONENS HÖGNIVÅSPRÅK

av Björn Gustavsson

Tycker du att BASIC är för långsamt? Tycker du att Pascal ger för dålig kontroll över datorn? Tycker du att assembler är för besvärligt att använda?

Då är kanske FORTH det rätta språket för dig. Under våren har TI FORTH introducerats i USA och vi får hoppas att det snart kommer till Sverige.

Låt oss titta på några av FORTHS fördelar:

### Snabbhet

FORTH är det snabbaste högnivåspråket för 99:an. (Assembler är snabbare, men det är inget högnivåspråk.) Man räknar med att FORTH är ca 20-75 procent långsammare än ren maskinkod. Som ett exempel kan nämnas att jag provade att ta tid på en tom loop i FORTH:

```
: TOM-LOOP 1000 0 DO LOOP ;
```

Jag hann aldrig starta klockan! Loopen ser ut så här i TI BASIC:

```
100 FOR I = 1 TO 1000
110 NEXT I
```

### Utbyggbarhet

FORTH innehåller inte ett fast antal instruktioner som t ex BASIC. Språket bygger just på att det kan utökas i

oändlighet. Man definierar nya "ord" (instruktioner) genom att skriva kolon, det nya ordets namn och sedan definitionen med hjälp av ord som FORTH redan kan. Definitionen avslutas med semikolon.

När man programmerar bygger man upp fler och fler nya ord. Till slut har man ett ord, eller en grupp av ord, som löser hela problemet. Man behöver då endast skriva ordets namn så utförs det.

I FORTH är program och ord samma sak. Eftersom man kan ha många ord definierade samtidigt kan man alltså ha flera program i minnet samtidigt. Varje program är åtkomligt genom att endast skriva dessa man.

Jämför det med BASIC där du bara har ett "ord" — RUN.

### Strukturerat

Man har i FORTH goda möjligheter att skriva strukturerade program, dvs program som är lättlästa och lätta att ändra — även för andra än programmeraren själv. BASIC:s GOTO saknas helt i FORTH och jag kan inte tänka mig något tillfälle där det skulle behövas.

I stället styr man programflödet med satser som IF — ELSE — ENDIF, BEGIN — UNTIL, BEGIN — WHILE — REPEAT samt CASE. (Dessa har likheter med Pascals styrsatser, men är enligt min mening kraftfullare.) DO —

LOOP är en motsvarighet till BASIC:s FOR — NEXT.

### Kontroll

Man har så gott som samma kontroll över datorn som assembler. Men eftersom FORTH är ett högnivåspråk är det mycket lättare att använda. Ofta kan man prova en idé direkt från tangentbordet vilket är om möjligt i assembler (man måste skriva ett program och assemblera det) och BASIC.

Om man trots allt inte kan göra någonting direkt i FORTH kan man definiera ett ord i assembler. Detta ord används som vilket ord som helst. Man kan t o m skriva en kraftfull assembler i FORTH relativt lätt. Enda gången jag har behövt definiera ett ord i assembler var när jag skulle kontrollera in/ut-portarna direkt. (Det har man sällan anledning att göra.)

### Lätt att använda

Det har väl redan framgått att FORTH är lätt att använda när man väl kan det. Däremot kan det ta längre tid att lära sig det än andra språk, eftersom FORTH skiljer sig rejält från övriga datorspråk. Men det är väl värt att lära sig. Man kan göra kraftiga tidsbesparingar när man väl kan det.

Det kan gå tio gånger snabbare att göra ett program i FORTH jämfört med assembler. Ofta är just assembler det enda realistiska alternativet.

Jämfört med andra högnivåspråk brukar det ta hälften så lång tid att göra motsvarande program i FORTH.

### Kompakt

FORTH-kod är mycket kompakt, t o m kompaktare än assembler. Detta beror på att programmet som man matar in kompileras till en form som dels är väldigt kompakt (anrop av ett ord tar bara upp 2 bytes), dels kan interpreteras snabbt. FORTH är alltså både kompilerande och interpreterande.

Hur mycket minne behövs för att kunna köra FORTH? 16K RAM räcker gott och väl. I det ryms hela FORTH-systemet inklusive kompilator, interpretator, editor, assembler och plats för egna ord! Jämför det med TI BASIC där 16K RAM används bara för egna program. Dessutom finns en stor BASIC-tolk i ROM (24K).

### Portabelt

FORTH-program är också portabla, vilket innebär att de lätt kan flyttas mellan olika datorer. Detta beror på att stora delar av FORTH är skrivet i FORTH. Endast de grundläggande orden är skrivna i assembler. Det enda som är olika i skilda FORTH-system är assemblerdelen.

Nu har jag räknat upp FORTHs fördelar. Vad har då FORTH för nackdelar? Inga, säger kanske en FORTH-anhängare. Men här är några egenskaper som betraktas som nackdelar av en somliga.

### Flyttal och stränghantering saknas

Från början finns inga rutiner för flyttals- och stränghantering. Alla beräkningar utförs på heltal eller dubbla heltal. Dubbla heltal har en kapacitet på ca 9 siffror. FORTH-anhängarens "försvar" för detta är att flyttal inte behövs: Flyttalsaritmetik är långsam och man klarar sig bra med heltal och dubbla heltal.

Men om man ändå vill ha flyttalsaritmetik kan man själv lägga till det. På 99:an går det att anropa rutinerna i BASIC-tolken. Stränghantering är också lätt att lägga till.

### Svåräst?

Den allvarigaste kritiken är nog påståendet att FORTH är ett svåräst språk. Ofta beror påståendet på oekantskap med ett språk som skiljer sig ordentligt från andra språk. Men faktum är att man KAN skriva väldigt svårästa program i FORTH. Å andra sidan går det att skriva väldigt lättläs-

ta program i FORTH. Programmeraren är här i högsta grad ansvarig.

Det viktigaste att tänka på är följande:

Skriv korta enkla definitioner, högst några rader långa.

Välj beskrivande namn.

Kommentera ordentligt.

Det är brott mot bl a dessa enkla regler som leder till svårästa program.

### Ansvar

Programmeraren har ett stort ansvar. I andra språk som BASIC och Pascal kan man göra dumma misstag och tolken eller kompilatorn svarar med ett felmeddelande. Man kan också avbryta ett program som inte fungerar. I FORTH finns däremot få säkerhetskontroller. Resultatet av ett misstag brukar bli en systemkrasch.

Men man kan själv bygga in säkerhetskontroller.

Filosofin bakom detta är att FORTH-programmeraren inte vill att systemet gör en massa onödiga säkerhetskontroller när ett snabbt program behövs. Behövs säkerhetskontroller bygger man själv in dem och tar bort dem när de inte längre behövs.

Vill man inte ha detta ansvar väljer man lämpligen Pascal eller BASIC.

### RPN

FORTH är ett språk som delar världens programmerare i två halvor: de som älskar FORTH och de som hatar FORTH. Ingen torde vara helt oberörd. En av orsakerna är att FORTH använder omvänd polsk notation (RPN). Det innebär t ex att man inte skriver:

$$2 + 3 * 5 + 4$$

som i BASIC utan:

$$2 \ 3 \ 5 \ * \ + \ 4 \ +$$

Just skillnaden mellan dessa två sätt att uttrycka samma formel har varit föremål för många diskussioner. RPN är skrivsättet som är enklare för datorn att hantera. I BASIC-tolkar brukar algebraiska uttryck först göras om till RPN innan de beräknas! Om man har lust kan man naturligtvis utöka FORTH så att algebraiska uttryck godtas (det är ganska lätt), men vem vill det?

När man väl har lärt sig RPN brukar man tycka att det är riktigt smart. Faktum är att det är svårt att tänka sig FORTH utan RPN.

### TI FORTH

TI kallar sin version TI FORTH. TI har utgått från en dialekt som kallas fig-FORTH som skapades 1978 av FORTH Interest Group i Kalifornien i avsikt att sprida FORTH. Källkod är tillgänglig för de flesta populära mikroprocessorer för en billig penning.

FIG ger ut tidskriften FORTH Dimensions sex gånger om året.

TI har gjort en del förbättringar bl a i avsikt att snabba upp språket så mycket som möjligt. TI har också lagt in vissa förbättringar som diskuterats i FORTH Dimensions.

Ordet SYSTEM ger åtkomst av ett tiotal olika funktioner unika för 99:an:

Skrivning och läsning i VDP RAM (nödvändigt för att göra grafik).

Skrivning i VDP-register.

Anrop av rutiner i ROM (GPLLNK, XMLLNK och DSRLNK).

Initiering av disketter.

Det finns inga direkta ord för grafik men det är enkelt att definiera egna med hjälp av assemblermanualen. Troligen medföljer också exempel på definitioner i manualen för FORTH eller på diskett. (Jag har inte sett manualen.)

TI har också gjort det möjligt att kontrollera avbrott i FORTH. Sextio gånger i sekunden alstras ett avbrott då processorn avbryter vad den håller på med och hoppar till en särskild avbrottsrutin. När denna rutin slutfört sin uppgift fortsätter datorn med sin förra uppgift. Denna avbrottsrutin kan skrivas i FORTH! Detta öppnar fantastiska möjligheter i t ex spelprogram. Skenbart kan man få datorn att utföra två uppgifter samtidigt.

Jag gjorde en enkel avbrottsrutin som fick en sprite att studsa omkring på skärmen. Samtidigt kunde jag använda tangentbordet som vanligt. En idé som jag ännu inte hunnit prova är att låta utskrift på skrivaren vara avbrottsstyrt så att jag slipper sitta och vänta.

Slutligen kan man lagra kompilerad FORTH-kod på diskett. Normalt kan man bara lagra källkoden på diskett. Möjligheten att lagra kompilerad kod spar tid. Jag har själv gjort massor med utökningar som jag brukar läsa in varje gång jag startar upp systemet. Innan jag började använda den kompilerade versionen tog det några minuter att läsa in allt.

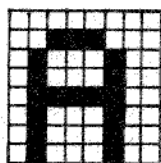


## ATT ANVÄNDA GRAFIKEN, DEL 1

av Björn Gustavsson

**I den här artikelserien kommer vi att titta på hur 99:ans kraftfulla färggrafik används. I den här delen kommer grunderna att behandlas. I kommande artiklar kommer vi bl a att titta på sprites i Extended BASIC.**

99:ans grafik bygger på att man gör egna tecken. Från början finns det hel del tecken, t ex siffror och bokstäver. Varje tecken är uppbyggt av 64 små punkter, ordnade i ett 8x8 rutnät. Tecknet "A" ser t ex ut så här:



Nu kan man själv göra tecken genom att bestämma vilka av punkterna som skall vara tända och vilka som skall vara släckta. Man kan t ex definiera Å, Ä och Ö (se Tekniktips i nr 1).

Låt oss steg för steg konstruera några tecken. Vi bestämmer oss för att göra tecken för spader, klöver, hjärter och ruter. Om vi börjar med spader gör vi så här:

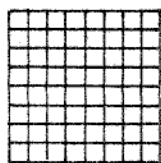
**STEG 1.** Vi tar ett vanligt rutat papper och ritat upp en ram så att vi får ett rutnät med 8x8 rutor.

**STEG 2.** Vi ritat upp spader genom att sätta kryss för de rutor som skall vara tända. Nu kanske någon tycker att det inte är särskilt likt spader. Men tänk på att figuren kommer att vara mycket mindre på skärmen och då blir det faktiskt ganska likt.

**STEG 3.** Nu skall vi göra om figuren till koder som datorn förstår. Först förstärker vi den vertikala mittlinjen. Nu är figuren uppdelad i grupper om fyra rutor. Innehållet i dessa fyra rutor kan representeras med en kod, bestående av en bokstav eller en siffra, enligt tabell 1. Fyra tända punkter betecknas t ex med "F" och tre tända punkter till höger med "7".

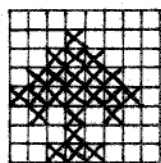
Nu kodar vi figuren! Vi går uppifrån och ner, och börjar med vänstra halvan. Först har vi fyra släckta punkter. Enligt tabellen är koden "O". Vi skriver alltså en nolla till vänster om motsvarande rutor. Därefter har vi en fylld ruta längst till höger. Koden är "1". Sedan två rutor till höger, vilket blir "3". På detta sätt fortsätter vi tills allt är ifyllt.

**STEG 4.** Vi grupperar koderna två och två på en enda lång rad. För spader får vi alltså koden 0010387CFE541038.



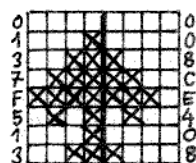
### STEG 1

RITA UPP ETT  
RUTNÄT PÅ  
8x8 RUTOR.



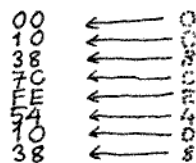
### STEG 2

RITA IN DIN  
FIGUR I RUT-  
NÄTET.



### STEG 3

FÖRSTÄRK DEN  
VERTIKALA MITT-  
LINJEN. SKRIV  
UT KODERNA PÅ  
RESPEKTIV SIDA AV  
FIGUREN.

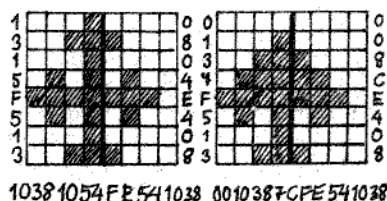


### STEG 4

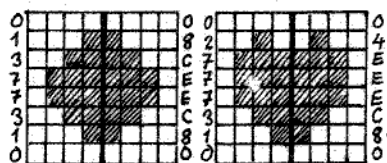
GRUPPERA  
KODERNA TVÅ  
OCH TVÅ OCH  
PLACERA DESSE  
EFTER VARANDRA.

00 10 38 7C FE 54 10 38

Vi gör precis likadant med de andra tecknen.



1038 1054 FE 54 1038 0010387CFE541038



00183C7E7E7E3C1800 00247E7E7E3C1800

Nu måste vi bestämma vilka ordningsnummer våra tecken skall ha. Tabell 2 visar vilka ordningsnummer de vanliga tecknen har. "A" har t ex numret 65. I stället för "ordningsnummer" talar man vanligen om "ASCII-kod".

Vi kommer knappast att behöva små versaler, så vi "knycker" ASCII-koderna 96, 97, 104 och 105 och definierar om dessa. I BASIC-programmet härintill definieras tecknen i raderna 110-140 (vad övriga rader gör kommer vi till). CALL CHAR (97, "0010387CFE541038") talar om hur tecken 97 skall se ut, dvs i det här

fallet som spader. I övrigt låter vi tecknen 96 vara klöver, 104 vara ruter och 105 hjärter.

Nu måste vi hitta på någonting som skriver ut tecknen på skärmen. Det görs på raderna 180-230. 180 och 230 innehåller en FOR-NEXT-loop som utförs 50 gånger. För att ha lite roligt bestämmer vi oss för att slumpa fram vilka tecken som skall skrivas och var de skall skrivas.

Vi sätter alltså

$KORT = 96 + INT(2 * RND)$

i rad 190. RND ger ju ett slumptal mellan 0 och 1 (aldrig lika med 1).  $2 * RND$  ger alltså ett slumptal mellan 0 och 2. Tar man då INT (heltalsdelen) får man 0 eller 1. KORT blir alltså 96 eller 97. Sedan testar vi

IF RND RND THEN 220

I hälften av fallen sker hopp till 220. I övriga fall ökas KORT med 8 och vi får 104 eller 105. På detta sätt kan alltså vilket som helst av spader, klöver, ruter eller hjärter väljas ut.

CALL HCHAR(rad, kolumn, tecken)

skriver ut ett tecken på en viss rad och kolumn. Raderna är numrerade från 1-24 där 1 är översta raden och kolumnerna från 1-32 där 1 är kolumnen längst till vänster. CALL HCHAR(12,16,97) skriver alltså ut spader (tecken 97) på rad 12, kolumn 16, alltså ungefär mitt på skärmen.

På rad 220 står det nu

CALL  
HCHAR(RND\*20+2,RND\*28+2,  
KORT)

vilket skriver ut ett slumpmässigt tecken på en slumpmässig plats på skärmen.

För att få olika resultat varje gång vi kör programmet har vi satt in RANDOMIZE på rad 100. Det ger slumpmässiga startvärde. För att kunna njuta av den färdiga bilden hur länge vi vill har vi satt in

240 GOTO 240

sist. för att stoppa programmet måste vi trycka FCTN 4.

Nu återstår bara raderna 150-170. CALL CLEAR raderar skärmen. CALL SCREEN(16) ändrar färgen på skärmen. Se tabell 3. Färgkod 16 betyder alltså vit. Till sist sätter CALL COLOR(10,7,1) färg på hjärter och ruter. Man sätter färg på en hel grupp med åtta tecken på en gång. Alla grupper är numrerade enligt tabell 4. Tecken 32-39 ingår i grupp 1, tecken 40-47 ingår i grupp 2 osv.

Hjärter och ruter, tecken 104 och 105, ingår i grupp 10. Därför anger vi 10

först i CALL COLOR. Därefter anger vi mörkt röd som färg på de punkter som är tända. Detta är färgkod 7. Slutligen måste vi ange färgen på de släckta punkterna. Den sätter vi till genomskinlig (kod 1) så får den samma färg som bakgrunden.

Varför ändrar vi inte färgen på spader och klöver? Därför att den normala färgen på alla tecken är svart.

I nästa artikel kommer vi att titta på fler grafiska möjligheter.

```
100 RANDOMIZE
110 CALL CHAR(96,"10381054FE5410
38")
120 CALL CHAR(97,"10387CFE541038
")
130 CALL CHAR(104,"00183C7E7E3C1
800")
140 CALL CHAR(105,"00247E7E7E3C1
800")
150 CALL CLEAR
160 CALL SCREEN(16)
170 CALL COLOR(10,7,1)
180 FOR I=1 TO 50
190 KORT=96+INT(2*RND)
200 IF RND>RND THEN 220
210 KORT=KORT+8
220 CALL HCHAR(RND*20+2,RND*28+2
,KORT)
230 NEXT I
240 GOTO 240
```



TAB 1

ASCII-KODSTABELL

|       |      |       |
|-------|------|-------|
| 32    | 64 @ | 96    |
| 33 !  | 65 A | 97 a  |
| 34 "  | 66 B | 98 b  |
| 35 #  | 67 C | 99 c  |
| 36 \$ | 68 D | 100 d |
| 37 %  | 69 E | 101 e |
| 38 &  | 70 F | 102 f |
| 39 ^  | 71 G | 103 g |
| 40 <  | 72 H | 104 h |
| 41 >  | 73 I | 105 i |
| 42 +  | 74 J | 106 j |
| 43 =  | 75 K | 107 k |
| 44 ,  | 76 L | 108 l |
| 45 -  | 77 M | 109 m |
| 46 .  | 78 N | 110 n |
| 47 /  | 79 O | 111 o |
| 48 0  | 80 P | 112 p |
| 49 1  | 81 Q | 113 q |
| 50 2  | 82 R | 114 r |
| 51 3  | 83 S | 115 s |
| 52 4  | 84 T | 116 t |
| 53 5  | 85 U | 117 u |
| 54 6  | 86 V | 118 v |
| 55 7  | 87 W | 119 w |
| 56 8  | 88 X | 120 x |
| 57 9  | 89 Y | 121 y |
| 58 :  | 90 Z | 122 z |
| 59 ;  | 91 [ | 123 { |
| 60 <  | 92 \ | 124   |
| 61 =  | 93 ] | 125 } |
| 62 >  | 94 ^ | 126 ~ |
| 63 ?  | 95 _ | 127   |

TAB 2

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            |

TAB 3

TECKENGRUPP

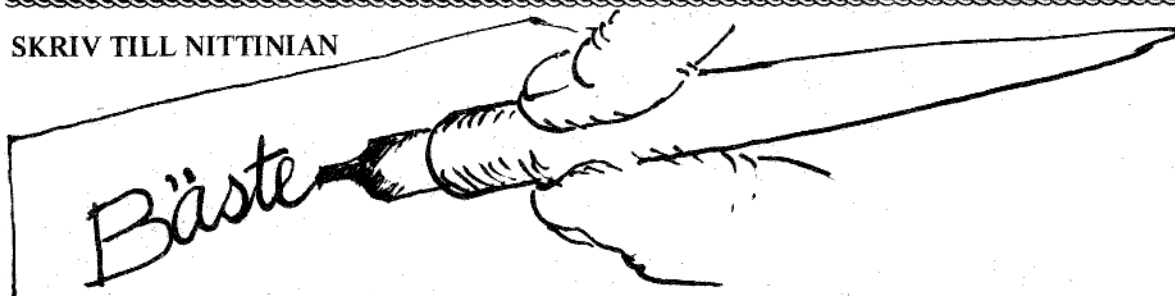
ASCII-KODER

|    |           |
|----|-----------|
| 1  | 32 - 39   |
| 2  | 40 - 47   |
| 3  | 48 - 55   |
| 4  | 56 - 63   |
| 5  | 64 - 71   |
| 6  | 72 - 79   |
| 7  | 80 - 87   |
| 8  | 88 - 95   |
| 9  | 96 - 103  |
| 10 | 104 - 111 |
| 11 | 112 - 119 |
| 12 | 120 - 127 |
| 13 | 128 - 135 |
| 14 | 136 - 143 |
| 15 | 144 - 151 |
| 16 | 152 - 159 |

OBS! GRUPP 15 OCH 16 FINNS  
EJ I EXTENDED BASIG.

TAB 4

## SKRIV TILL NITTINIAN



Vår brevlåda svämmer över med brev från alla som vill veta mera om nittinian, och vi hoppas att tidningen talar för oss och förklarar vad vi gör och vad vi och ni vill göra. Mening-

en är att var och en skall skriva tidningen och att det ska bli en tummelplats för 99- och 40ägare.

För att man skall känna sej inspirerad att skicka in bidrag

har vi tänkt oss att bidrag honoreras med 3kr per spaltcentimeter i publicerat skick. Ett spel med illustration och förklaringar som upptar en sida ger cirka 250 kr.

## ANNONSER

En användartidning är den naturliggaste kanalen för annonsering av mjuk- och hårdvara för TI-ägare. Därför välkomna annonsörer av alla sorter! Sidorna 2, 3 och 4 på omslaget

kostar 4 kr per spalt mm. Samma pris gäller annonser i texten, medan vår specialavdelningen "PULOCHPRYLMARKNADEN" kostar 2 kr per spalt mm.



## Kommande nummer

Flera spel i ti basic, extended  
och assembler.

En specialavdelning för  
dej som har en cc 40.

Fortsättningen på Björns  
grafiska skolan.

Hobbyavdelning för dej som  
vill bygga egen hårdvara

Lär dig över programmen i vår  
nystartade programförmedling.

Lär dig och reportage över  
höstens många datamässor.

Fortsättningen skall vara  
spännande efter denna  
Pangsummer!

## ANNONSBOOKNING

08-

357768

### NYTT-NYTT-NYTT

**EXT BASIC CATERPILLAR** 120:-  
Du är en mask med en  
evigt växande svans.  
Aktå dig att snubbla på  
svansen!

**Kräver Minimum ROBOT** 120:-  
Du jagas av robotar som  
blir fler och fler i ett  
minut. Snabbt spel!

Ring och Beställ!  
T-DATA 08-3555 04

### SPÄNNANDE PÅ SVENSKA

#### GLUFS

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!



Ring och Beställ!

#### T-KENO

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

#### KALAH

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

#### DU ÄR HANGD!

Klassisk med massor av ord.

1 SPEL 75:-

T-DATA 08-3555 04

2 SPEL 140:-