

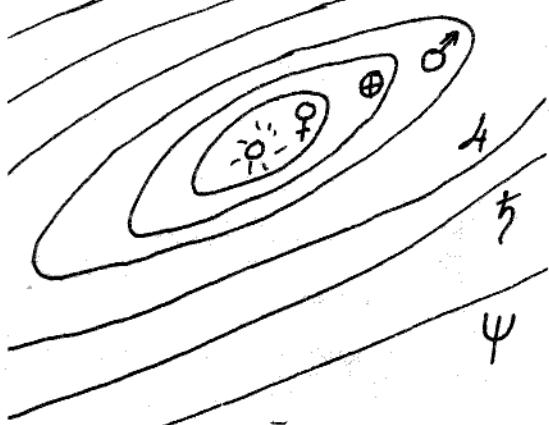
# nittinian EDITION

TI 59  
PÅ  
TI-99 4A



samt

PLANETARIUM



## Innehåll

Redaktören	2	
Medlemsmöte i Lund	2	
Ordförande	3	
Utmaningen	4-7	
Sortering	8	
TI-59	n-te gradspolynom	9
	Planetarium	10-11
	Parabelanpassning	12-16
Favorit i repris	17	
Ta det piano	17	
Tips och Tricks	18	
Polsk notation	19	
Assembler skolan del 2	20-21	
Datakommunikation med modem	22	
Datavision	22	
Annons	23	
Programbanken	24	

ISSN 0281-1146

# Redaktören . .

Du håller nu i nummer två av Programbiten för år 1984. Efter monteringsvällar på ideell grund har redaktionsmedarbetarna äntligen fått se ett resultat av sitt frivilliga arbete, ett nytt nummer!

Texas räknaren TI-59 ger sig inte med att bli utklassad av hemdatorerna utan kommer i detta nummer med ett mycket intressant program för beräkningar av planeters vandring på vår himmel, timme för timme, sekund för sekund. 99-an har fortfarande en plats bland de större beräkningsmaskinerna. Jag skulle med glädje se detta program utvecklas och förbättras för anpassning till 99-an. Kanske något för utmaningen?

Annars innehåller detta nummer knep för TI99/4A till exempel hur du "poppar" upp subrutinstacken i Extended-Basic.

En ny sida har dykt upp vilken tar upp tips och knep vilka inte står i manualer utan som "hackers" har listat ut genom "trial and error".

Tyvärr finns det väl inte speciellt många program för TI-BASIC, men käre medlem DU måste skicka in Basic program ty det är just så att en del av dem som skriver program för publicering i tidningen är nog lite för avancerade till att sitta och skriva enkla Basicprogram vilket du nog ser av innehållet i tidningen.

Utanförningen tar upp sorteringsrutiner för olika program-språk, det finns även ett tillägg om mera sorteringsar som kom oss tillhandha efter det att utaningsredaktören skrivit färdigt sin del.

Ett problem som tagits upp i utanförningen är, hjälpprogram för 59-an skrivet i Extended-Basic för 99an.

Stränghantering tas även upp med två korta subrutiner i Extended-Basic.

Polsk notation i Basic? Javisst! se programmet "Den Italienske Programmerarens Helafton" en festligt program för dig som inte har Forth, går säkert att bygga ut efter behag och kunskap.

Att använda sin dator till att spela piano med går relativt bra, ett enkelt program som kan skrivas i TI-Basic eller Extended-Basic finns också med i detta nummer.

Som ni märker hänger tidningen Programbiten med och försöker fylla sig med bra, rolig, svår, intressant, lätt och spännande läsning.

Så dyre medlem starta nu din färd i matematikens och beräkningsmaskinernas värld!

## I REDAKTIONEN:

Redaktör	Göran Nygren
Biträende redaktör	Michael öhman
Utaningsredaktör	Anders Persson
Programförmeldare	Björn Mårtensson

Föreningens och redaktionens adress:

Föreningen Programbiten  
c/o Schibler  
Wahlbergsgatan 6 1 tr ned  
121 46 Johanneshov  
Postgiro 19 83 00 - 6

+++++ DATAINSPEKTIONENS LICENSNUMMER +++++  
82100488  
+++++ +++++++

Medlemsavgift för 1984 är.....120 SEK  
årgång 1983 (nr 1, 2, 3 + 4/5) kostar 80 SEK

## Medlemsträff

Det kom ett brev till mig dagarna innan vi skulle lämna in Programbiten nr 2.84 till tryckning. Vi hann i alla fall med att montera det "i sista stund".

Hej Göran!

Efter det att jag hörde att medlemsträffen i Stockholm, lördagen 15/9 blev en succé med mycket folk (ca 65 st. Claes räknade. red. anm.). Anser jag att det är orättvist att bara Stockholm skall ha medlemsträffar. Istället för att gå omkring och klaga över detta. Har jag tagit saken i egna händer och åstadkommit detta:

EN MEDLEMSTRÄFF I LUND !!!

Lördagen 17/11 (november)  
Lokal: Lunds Tekniska Högskola,  
E-huset, Ole Römers väg 3, Lund.  
Tid: kl. 14-18

### TEMA:

Utrustning till 99:an.  
Vilket innebär några stycken TI99/4A.  
Speech Synthesizer.  
Expansionsbox.  
32kB RAM expansionsminne.  
RS232-kort. Skrivare (printer).  
Diskssystem, flexskive-enheter och flexskive-kontroller,  
dubbelsidiga halvhöjdsdrivrar vilka får  
plats i expansionsboxen.

Mini-Memory modul.  
Delvis hembyggd RAM expansionsminne.  
Extended-Basic modul.  
Editor/Assembler modul.  
Ti-Writer.  
Multiplan.  
USCD-Pascal system komplett med P-kods kort alla  
systemprogram (Editor, Compiler, Assembler Linker, dokumentation m.m.).

99:er Magazine för ett helt år.  
Parsec. Munch-Man. Adventure.  
A-register plus mycket annat.  
Dessutom tillgång till kvalificerat folk som du kan  
fråga och få svar om det mesta.  
Kanske någon av er i Stockholm vill komma ner också?  
Alltså medlemmar i södra Sverige nu har ni chansen att  
bräda oss i huvudstaden. Vi har inte allt här i  
Stockholm! ? (red.anm)

Medlemsträffen i Lund blir av lördagen den 17 november.  
Kl. 14-18.  
Lunds Tekniska Högskola,  
E-huset. Ole Römers väg 3. Lund.

Väl mött alla 99:e användare i Södra Sverige.

Hälsningar  
Anders Persson



# ORDFÖRANDE har ORDET

Fortsatt tack för brev!

Vad gör Du egentligen med Din 99-a? Varför skaffade Du en hemdator överhuvudtaget? Och varför en 99-a?

Jag uppskattar alla brev vi får med frågor om det ena och det andra och vi svarar så gott vi kan. Jag skulle emellertid bli väldigt glad också för brev som berättar om Dig och Din 99-a. Skriv sådana brev! Skriv gärna sådana brev som vi får publicera! Jag tror att det finns ett intresse bland andra medlemmar för sådant. Har Du t ex nyttja av maskinen? Har Du nöje av den? Är den en teknisk och intellektuell utmaning? Eller vad är den för Dig?

En läkare har skrivit och berättat hur han använder maskinen i sitt jobb! Det kommer nog en artikel om det.

I förra numret påstod jag att ryktet om vår död var betydligt överdrivet. Mycket påpassligt kommer tidningen MIKRODATORN nu med en artikel om 'En dators död' - TI 99/4A nämligen! Det är mycket spännande läsning som framför allt skildrar mätspelet i ett storföretag - Texas Instruments förstås. Det handlar inte särskilt mycket om produkten - de maskiner som vi har hemma nu. Det är väl framför allt dem och deras användningsmöjligheter som intresserar oss tror jag.

Och användningsmöjligheterna ökar.

Allt fler tillbehör verkar bli tillgängliga genom olika företag. Vi har fått en hel del information. Det blir så småningom en artikel om detta. Troligen kommer det också en del reklam. Jag tycker att framförallt de nya kontrollkorten för skivenheter och nya skivenheter med hög lagringskapacitet är intressanta. Här ligger Texas egen grejor i lä! Det dyker också upp fler spel. Bl a ser vi en del ATARI-spel som anpassats till 99-an och distribueras på modul.

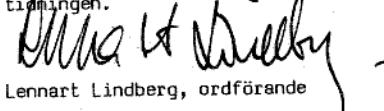
Vi får nog emellertid räkna med att de här grejorna bara kommer att kunna skaffas per postorder. Antalet butiker som fortfarande säljer 99-tillbehör över disk kan förväntas gå mot noll. Jag undrar faktiskt om vi inte snart kan känna oss som 'VETERAN HACKERS'. Jag kan se rubrikerna: "99-an - den första RIKTIGA hemdatorn! Tidiga exemplar i gott skick stiger i pris!"

Internationella kontakter: vi har genom Peter Odelryd hittat en engelsk 99-förening som ger ut en mycket matnyttig tidning. Få se om vi kan vidarebefordra en del därifrån i vår tidning.

Så FORTH då. Jag gillar FORTH även om jag inte är helt säker på att kärleken är besvärad. I sommar har jag haft tillfälle att - helt amatörmässigt - arbeta med VIC-64 FORTH. 99-ans FORTH är i flera avseenden alldeles utmärkt bra. Editorn t ex. Särskilt eleganta är en del saker som Björn Gustavsson gjort i föreningens FORTH.

--> Nu känner jag att jag gjort det igen! Jag menar att jag skrivit några rader om det som kan intressera dem som har råd, lust och ork att gå vidare i sin datorhobby mot fler prylar och mer avancerad användning. Vi har många som arbetar med Basic eller X-basic och kassettsbandspelare och tycker att det svarar mot rimliga ambitioner. Vi skall försöka göra mer för Er. Problemet är ju dock ett de som skriver och är aktiva - både brevskrivare och tidningsskribenter - är de som har de mest avancerade behoven och ideerna.

Alltså! Låt mig få höra mera ifrån DIG somm jobbar med Basic och kassett! Jag tar också gärna emot anmälningar från någon som vill särskilt ta hand om dessa frågor i tidningen.

  
Lennart Lindberg, ordförande

## RÄTTELSE

Vår vän i USA, Maurice E. T. Swinnen, läser med stort nöje vår tidning, säger han. Jag undrar hur vi skall tolka det, vetandet att han inte kan svenska så vidare bra? Noggrann är han i alla fall, ty i nr 1/84 har han upptäckt ett fel i en programlistning! Det handlar om sid 26, 'Snabbare grafik i TI Basic?', exempel 4. På programraderna 190 och 200 är Call Char-satserna ofullständiga. Maurice skickar med en ny programlista, där endast raderna 190 och 200 ändrats. Den följer här:

```
100 CALL CLEAR
110 A$="0000000000000001038107C545428
44FFFFFFFFFFFF"
120 FOR I=0 TO 23
130 CALL CHAR(I+128,"0")
140 CALL HCHAR(I+1,14,I+128,5)
150 CALL SOUND(-100,930-I*30,1,940-I*3
0,I,950-I*30,I)
160 NEXT I
170 FOR I1=150 TO 128 STEP -1
180 FOR I2=I TO 17 STEP 2
190 CALL CHAR(I1,(SEG$(A$,12,16)))
200 CALL CHAR(I1,(SEG$(A$,12+16,16)))
210 CALL SOUND(-1000,-4,0)
220 NEXT I2
230 NEXT I1
240 GOTO 120
```

Maurice är mjukvarukonsult med specialisering på Texas Instrument. Han får tillgång till intressanta såväl prylar som program. Nyligen berättade han att det finns en modul som heter Disk Fixer, som primärt tydligen är tillverkad för att kunna reparera disketter med dåliga sektorer eller som på annat sätt kommit i olag. Den kan också användas för att t ex analysera och kanske så småningom kopiera även kopieringsskyddade program. Disk Fixer kommer från Navarone. Pris i USA c:a 40 dollar.

### ANVÄNDARTIPS MED MINI MEMORY

Denna bok skrevs av Björn Gustavsson på uppdrag av Texas Instruments. Precis när boken var klar lade TI ner hemdatorn. Föreningen fick då rätten till boken, och har tryckt upp den. Boken innehåller 60 sidor och kostar för medlemmar 60 SEK som sätts in på föreningens postgiro 19 83 00-6.

### VIKTIGT MEDDELANDE

Föreningen kan från och med nr 84-1 inte lämna någon ersättning för artiklar och program enligt notis i Nittinian nr 2.

Annonser, insatta av enskild medlem (ej företag), som gäller försäljning av moduler eller andra tillbehör i enstaka exemplar är gratis.

Övriga annonser kostar 2000 SEK per helsida, förutom baksidan som kostar 3000 SEK.

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 Programbiten. Förbudet gäller varje form av mångfaldigande genom tryckning, duplicering, stencilering, bandinspelning, diskettinspelning etc.

# UTMANINGEN!

Redaktörens adress:  
Anders Persson  
Kämnärvägen 4:1078  
222 45 LUND

Här kommer Utmaningen så att ni får något att sysselsätta er med de dagar då det är regnigt.

Några personer har påpekat att vissa av de problem som förekommer i Utmaningen är så komplicerade eller omfattande att lösa att ingen rimligtvis kommer att gå ifland med uppgiften, inom ramen för Utmaningen alltså. Man kan ju inte förvänta sig att någon skall arbeta i månader med ett program bara för att det fanns i Utmaningen. Själv anser jag dock inte att omfattande problem (läs verkliga utmaningar) skall tas bort bara för att de är arbetsamma. Det är inte alls nödvändigt att presentera en fix och färdig lösning till varje problem. Någon har kanske idéer om hur vissa delar av ett program bör utformas, även om vederbörande inte har intresse av att ge sig på och skriva hela programmet. Skriv då ner dessa idéer på ett papper och skicka in dem till mig så att vi kan få till stånd en diskussion om hur problemet bör tacklas här i Utmaningen. Hjälp till självhjälp, med andra ord.

## 47. SORTERING

Att det skulle gå att sortera på många olika sätt var väl väntat. Hitintills har vi sett tre olika program i tidningen, samtliga skrivna i olika programspråk (om man räknar TI BASIC och Extended BASIC som olika). Här kommer två program till, fortfarande i nya programspråk!

Det första programmet, vilket kommer från Björn Gustavsson, är skrivet i i FORTH (förstås.) Björn har dock inte skrivit det själv, utan hämtat rutinen från tidningen Forth Dimensions. Författare är Marc Perkel från Missouri. Det är här frågan om en rekursiv implementation av Quicksort. Detta innebär att den inte är lika effektiv som t.ex. Pascalprogrammet i förra numret, men då Forth är ett snabbt språk går sorteringen ändå snabbare! Programmet nedan sorterar 1000 heltal (16-bitars) på ungefär 10 sekunder. Pascal behövde 20-22 sekunder, ibland 30 beroende på hur talen var ordnade från början.

Observera nu att detta inte är en rättvis jämförelse, eftersom Forth-rutinen inte är lika effektiv som motsvarigheten i Pascal. Trots detta är Forth alltså snabbare!

I nuvarande skick sorterar ordet SORT listor med åttabitars tal. Genom att byta ut alla C och C! mot och ! sorteras 16-bitars tal i stället. Modifieringen av adresser på rad 9 (där D+ används på ett smart sätt) måste naturligtvis också ändras. 16-bitars tal tar ju upp två adresser.

En annan intressant effekt kan man få genom att använda V respektive V!. Då kan man sortera bildskärmen, och på så sätt verkligen se hur Quicksort arbetar. Pedagogiskt värre! Ordet MYSELF i slutet av ordet är Forths variant av rekursion, där ordet helt enkelt anropar sig själv.



```

0 C Recursive Quick Sort - Sorts an
array of bytes.
1
2 O VARIABLE MIDDLE
3 : EXCHANGE DUP CA SWAP CA ROT C!
   SWAP C! !
4 : SORT ( start end -- )
5 2DUP 2DUP OVER = 2/ + CA
   MIDDLE !      ( pick middle one )
6   BEGIN
   ( scan right then left )
7     SWAP BEGIN DUP CA MIDDLE @
   < WHILE 1+ REPEAT
8     SWAP BEGIN DUP CA MIDDLE @
   > WHILE 1- REPEAT
9     2DUP > NOT IF 2DUP EXCHANGE
   1 -i D+ ENDIF
10    2DUP >      ( until
   partitions cross)
11    UNTIL SWAP ROT ( sort both
   pieces)
12  2OVER 2OVER = ROT ROT = < IF
   2SWAP ENDIF
13  2DUP < IF MYSELF ELSE 2DROP
   ENDIF ( smallest first )
14  2DUP < IF MYSELF ELSE 2DROP
   ENDIF ! ( then large piece )
15

```

Forthrutinen ovan kan utan problem köras i föreningens Forth.

När det är verkligt bråttom - och vem vill vänta längre än nödvändigt - duger naturligtvis bara program i assembler. Även här är det dock viktigt med vettiga algoritmer, även om programmeringssättet har större betydelse än när det gäller högnivåspråk.

Ingen har behagat skicka in något sorteringsprogram i assembler, varför jag visar mitt eget i stället. Det är avsett att anropas från Pascal, med det bör inte vara några problem att utnyttja metoden i andra sammanhang.

UCSD-Pascal arbetar ju med en virtuell stackmaskin, den så kallade p-maskinen. I implementationen på TI 99/4A används R10 i Pascals workspace som stackpekare. Vid anrop av en FUNCTION eller PROCEDURE stackas parametrarna i den ordning de är angivna. Sista parametern kommer alltså överst. Pascal tillåter att ett assemblerprogram använder R0-R7, men det är inte tillräckligt i det här fallet. Därför börjar jag med en BLWP för att undvika att oroa Pascals workspace. Återhopp till Pascal ska ske med B \*R11. Dessutom måste man se till att ändra stackpekaren så att parametrarna inte längre finns på stacken. I det här fallet ska två saker anges vid anropet. Överst på stacken finns antalet tal som ska sorteras, och därunder adressen där talen börjar.

För övrigt är assemblerprogrammet skrivet tämligen rakt fram, som en kopia av Pascalprogrammet i förra Utmaningen. Vissa minnesceller har jag dessutom givit sama namn som motsvarande variabler i Pascal (och BASIC). Vissa rader är också kommenterade med de Pascalrader sm de motsvarar. Tecknet ::= betyder att två variabler byter värden med varandra.

Vid labeln INNERLOOP finns ett trick som kallas för "loop unwrapping". Eftersom det är oekonomiskt (tidsmässigt) att avsluta en loop med ett ovillkorligt hopp, har jag "vridit runt" en loop. Däriigenom får jag det villkorliga hopp som, eventuellt, bryter slingan i slutet på densamma, och behöver alltså inte öda tid på ett onödigt ovillkorligt hopp vid varje varv. Det blir naturligtvis nödvändigt att hoppa in i slingan, eftersom startpunkten inte kommer "överst", men det gör man ju bara en gång vid varje tillfälle när slingörningen påbörjas.

```

; Procedur som sorterar integers i växande ordning
; Anropas från Pascal med QSORT(A,N);
; Deklareras som PROCEDURE QSORT(VAR A:VECTOR;
; N:INTEGER); EXTERNAL;
; Vector skall vara en ARRAY (...) OF INTEGER;

;-----;
; Workspace för subrutinen vid START
;
; R0 Array basadress
; R1 Array slutadress
; R2 L
; R3 R
; R4 I
; R5 J
; R6 KEY
; R7 Temp
; R8 Temp
; R9 Subfil stack-pekar
; R10 Huvudstackpekar (kopia från Pascal WS)
; R11
; R12
; R13 Anropande programs WP
; R14 Anropande programs PC
; R15 Anropande programs ST
;-----

; Vektorn i Pascalprogrammet kan indexeras [1..n]
; Detta program förutsätter endast att n indikerar
; antalet element, ej sista index
;-----;

        .RELPROC QSORT,2
LIMIT    .EQU 16           ;Quick/Insert-gräns
        .BLWP ESORTING ;Anropa Quicksort
        .AI R10,4       ;Simulera två stack-pop
        .B  *R11         ;Tillbaka till Pascal
;-----;
SORTING .WORD SORTWS ;Quicksort hoppvektor
        .WORD START
;-----;

; Hämta stackpekan från anropande programs WS
; Hämta parametrarna i anropet
; Räkna ut var listan A slutar
;-----;

START   MOV E14H(R13),R10
        LI R9,ENDOFSTK
        MOV *R10+,R1
        MOV *R10+,R0
        DEC R1
        SLA R1,1
        A R0,R1

; Initiera L och R, räkna ut om Quicksort behövs
        MOV R0,R2      ;L:=1
        MOV R1,R3      ;R:=N
        MOV R1,R7
        S R0,R7

        CI R7,LIMIT
        JLE INSERT

; Huvudloopen i Quicksort
;-----;

MAINLOOP MOV R2,R7
        SRL R7,1
        MOV R3,R8
        SRL R8,1
        A R8,R7
        ANDI R7,0FFFH ;R7:=INT((L+R)/2)
        MOV *R7,R8
        MOV E2(R2),*R7
        MOV R8,E2(R2) ;A[(L+R)/2]:=A[L+1]

        C E2(R2),*R3
        JLT NOSWAP1
        MOV E2(R2),R8
        MOV *R3,E2(R2)
        MOV R8,*R3 ;A[L+1]:=A[R]

```

```

NOSWAP1 C *R2,*R3
        JLT NOSWAP2
        MOV *R2,R8
        MOV *R3,*R2
        MOV R8,*R3 ;A[L]:=A[R]

NOSWAP2 C E2(R2),*R2
        JLT NOSWAP3
        MOV E2(R2),R8
        MOV *R2,E2(R2)
        MOV R8,*R2 ;A[L+1]:=A[L]

NOSWAP3 MOV R2,R4
        INCT R4      ;I:=L+1
        MOV R3,R5      ;J:=R
        MOV *R2,R6      ;KEY:=A[L]
        JMP INCLOOP

INNERLOOP MOV *R4,R8      ;Loop unwrapping
        MOV *R5,*R4
        MOV R8,*R5 ;A[I]:=A[J]

INCLOOP  INCT R4      ;I:=I+1
        C *R4,R6
        JLT INCLOOP ;A[I]KEY ?
        C R4,R5
        JLE INNERLOOP ;Om I<=J så fortsätt loopen

OUT      MOV *R2,R8
        MOV *R5,*R2
        MOV R8,*R5 ;A[L]:=A[J]

DEL1     MOV R5,R7      ;Quicksort för sub-filer?
        S R2,R7      ;R7:=J-L
        MOV R3,R8
        S R4,R8
        INCT R8      ;R8:=R-I+1
        CI R7,LIMIT
        JH DEL2
        CI R8,LIMIT
        JH DEL2

        CI R9,ENDOFSTK ;LVL=0?
        JEQ INSERT      ;Slut på Quicksort?

        MOV *R9+,R2      ;POP L
        MOV *R9+,R3      ;POP R
        JMP MAINLOOP

DEL2     C R7,R8      ;Bestäm vilken subfil som är
        JL ELSE2      ;minst

        MOV R2,ELSFL
        MOV R5,ELSR
        DECT ELSFR
        MOV R4,ESSFL
        MOV R3,ESSFR
        JMP DEL3

ELSE2   MOV R4,ELSFL
        MOV R3,ELSR
        MOV R2,ESSFL
        MOV R5,ESSFR
        DECT ESSFR

DEL3     CI R7,LIMIT      ;Liten subfil stor nog för
        JLE THEN3      ;Quicksort?
        CI R8,LIMIT
        JH ELSE3

THEN3   MOV ELSFL,R2      ;Quicksort stor endast
        MOV ELSFR,R3
        JMP MAINLOOP

ELSE3   DECT R9      ;Stacka stora subfilen
        MOV ELSFR,*R9      ;Push R
        DECT R9
        MOV ELSFL,*R9      ;Push L
        MOV ESSFL,R2      ;Sortera liten subfil
        MOV ESSFR,R3
        JMP MAINLOOP

```

;-----  
;Insertsortering som avslutning

```

INSERT    MOV R1,R4
          DECT R4      ;I:=N-1
          C   R4,R0
          JL  LEAVE    ;Ingen sorterings alls?

FORI     C   E2(R4),*R4
          JGT NEXTI   ;Om nästa värde större: OK

          MOV *R4,R6   ;KEY:=A[I]
          MOV R4,R5
          INCT R5      ;J:=I+1

WHILE    MOV *R5,E-2(R5) ;A[J-1]:=A[J]
          INCT R5      ;J:=J+1
          C   R5,R1
          JH  ENDWHILE ;J>N?
          C   *R5,R6   ;A[J]=KEY?
          JLT WHILE

ENDWHILE MOV R6,E-2(R5) ;A[J-1]:=KEY
NEXTI    DECT R4
          C   R4,R0
          JHE FORI    ;Förbi array startadr.?

LEAVE    RTWP      ;Till huvudnivå (asm)

;-----
;**** Data area ****

```

```
;-----  
;**** Data area ****  
;-----
```

```
SORTWS .BLOCK 20H,0 ;Workspace, sorteringspgm  
SUBFSTK .BLOCK 40H,0 ;Intern subfilstack  
ENDOFSTK .EQU SUBFSTK+40H
```

```
LSFL    .WORD 0 ;Large SubFile Left  pekare
LSFR    .WORD 0 ;Large SubFile Right pekare
SSFL    .WORD 0 ;Small Subfile Left   pekare
SSFR    .WORD 0 ;Small Subfile Right  pekare

        .END
```

Härmed kan man väl säga att vi har sorterat färdigt när det gäller tal. En rutin (skriven i assembler) som sorterar strängar i BASIC genom att läsa av de pekare BASIC använder och ändra endast dessa vore förstås trevlig...

45. HJÄLPPROGRAM FÖR TI 59

Lennart Thelander har tagit sig an problem 45a. Här nedan kommer ett program som listar TI 59-program på skärmen eller på skrivare, om tangentkoderna matas in. Det finns också vissa möjligheter till redigering av det inmatade programmet. Lennarts program är skrivet för Extended BASIC och användes så här:

Efter start visas en minneslista över vilka kommando som finns.

HELP förklarar dessa kommando lite bättre.

LEARN innehåller inmatning av koderna. Med upp- och nerpilarna kan man flytta sig i ett program utan att ändra något. Avslutning av Learn-läge får genom att mata in något otillåtet, text alltså. Om negativa koder anges kommer man åt vissa speciaffunktioner, såsom Insert, Delete och Backstep flera redar i taget.

TYPE skriver ut 59-programmet på skärmen. Man kan börja på vilken rad som helst.

PRINT är samma sak, fast på skrivare. Filnamnet finns på rad 510. Där får du ändra om du har din skrivare annorlunda ansluten.

EXIT avslutar programkörningen.

GOTO möjliggör ett hopp till vilken rad som helst i 59-programmet.

CLEAR raderar hela 59-programmet.

De mest dramatiska av funktionerna ovan kräver att man kvitterar en fråga innan de verkligen utförs.

Programmet ser ut så här:

```

100 ON ERROR 140
110 ON WARNING NEXT
120 GOTO 130 :: DIM P(959),OUT$(99):: ST,SCREEN,
A$,K,S,PC,I,MX :: CALL KEY :: CALL CLEAR :: !BP-
130 CALL CLEAR :: PRINT "INITIALIZING": :: :: ::
FOR I=0 TO 99 :: READ OUT$(I):: NEXT I
140 PRINT "PUSH A KEY::"Help Learn Type P
rint Exit Goto Clear"
150 PRINT :: :
160 ON ERROR 140
170 ACCEPT AT(23,1)SIZE(1):A$ :: PRINT
180 IF A$=="H" THEN 260
190 IF A$=="L" THEN 310
200 IF A$=="T" THEN 450
210 IF A$=="P" THEN 500
220 IF A$=="E" THEN 290
230 IF A$=="G" THEN 300
240 IF A$=="C" THEN 920
250 PRINT "ILLEGAL COMMAND" :: GOTO 140
260 PRINT "Type MEANS TYPE ON SCREEN Print MEA
NS PRINT ON PRINTERLearn MEANS INPUT OF PROG
RAM USE BLANK INPUT TO EXIT"
270 PRINT "Exit MEANS EXIT THIS PROGRAMGoto MEAN
S GOTO PROGRAMSTEP Clear MEANS CLEAR PROGRAM
"
280 PRINT :: :: GOTO 140
290 PRINT "DO YOU WANT TO EXIT (Y/N)" :: ACCEPT
AT(23,28)SIZE(1)BEEP:A$ :: IF A$=="Y" THEN S
TOP ELSE 140
300 INPUT "WHICH LINE? ::PC :: GOTO 140
310 PRINT "TYPE SPACES TO EXIT.";"TYPE NEGATIVE
FOR SPECIAL INSTRUCTIONS"
320 PRINT USING "### ##":PC,P(PC)
330 ACCEPT AT(23,5)SIZE(-2):A$ :: CALL KEY(O,K,S
):: I=VAL(A$)
340 IF I>0 THEN 420
350 INPUT "TYPE: Ins Del Exit Bst ::A$"
360 IF A$<>"I" THEN 380
370 FOR I=MX TO PC STEP -1 :: P(I+1)=P(I):: NEXT
I :: MX=MX+1 :: P(PC)=0 :: GOTO 320
380 IF A$<>"D" THEN 400
390 FOR I=PC TO MX :: P(I)=P(I+1):: NEXT I :: PC
=MX=0 :: MX=MX-1 :: GOTO 320
400 IF A$=="B" THEN INPUT "HOW MANY LINES BACK? "
I :: PC=PC-I
410 GOTO 320
420 P(PC)=I :: MX=MAX(MX,PC)
430 IF K=11 THEN PC=PC-2
440 PC=PC+1 :: GOTO 320
450 SCREEN=-1 :: GOTO 520
460 PRINT USING "### ##":ST,P(ST):: RETURN
470 PRINT #1,USING "### ##":ST,P(ST):: RETURN
480 DISPLAY AT(23,8):A$ :: RETURN
490 PRINT #1:A$ :: RETURN
500 SCREEN=0
510 OPEN #1:"RS232/2.BA=9600.DA=8"
520 INPUT "START AT 000 (Y/N)? ::A$ :: IF A$=="Y"
THEN ST=0 ELSE ST=PC
530 PRINT "TYPE 'S' TO STOP";"HOLD ANY KEY TO P
AUSE TYPING"::
540 IF ST>MX THEN IF SCREEN=0 THEN PRINT #1: :: ::
CLOSE #1 :: GOTO 140 ELSE PRINT #1: :: ::
GOTO 140
550 CALL KEY(O,K,S):: IF K=83 THEN ST=1000 :: GO
TO 540
560 CALL KEY(O,K,S):: IF S>0 AND SCREEN THEN 55
0
570 IF SCREEN THEN GOSUB 460 ELSE GOSUB 470
580 I=P(ST):: IF I=82 OR I=36 OR(I>39 AND I<45)O
R I=48 OR I=49 OR I=69 OR(I>61 AND I<65)OR(I
>71 AND I<75)OR I=83 OR I=84 THEN 640
590 IF I=61 OR I=67 OR I=71 OR I=77 THEN 690

```

```

600 IF I=97 OR I=87 THEN 850
610 A$=OUT$(I)
620 IF SCREEN THEN GOSUB 480 ELSE GOSUB 490
630 ST=ST+1 :: GOTO 540
640 A$=OUT$(P(ST)):: ST=ST+1 :: IF SCREEN THEN G
OSUB 480 ELSE GOSUB 490
650 IF SCREEN THEN GOSUB 460 ELSE GOSUB 470
660 IF P(ST)<10 THEN A$=" 0"&STR$(P(ST))ELSE A$=
" "&STR$(P(ST))
670 IF SCREEN THEN GOSUB 480 ELSE GOSUB 490
680 ST=ST+1 :: GOTO 540
690 A$=OUT$(P(ST)):: ST=ST+1 :: IF SCREEN THEN G
OSUB 480 ELSE GOSUB 490
700 IF SCREEN THEN GOSUB 460 ELSE GOSUB 470
710 I=P(ST):: ST=ST+1 :: IF I=40 THEN 740
720 IF I<10 THEN 790 ELSE A$=OUT$(I)
730 IF SCREEN THEN GOSUB 480 :: GOTO 540 ELSE GO
SUB 490 :: GOTO 540
740 A$="IND" :: IF SCREEN THEN GOSUB 480 ELSE GO
SUB 490
750 IF SCREEN THEN GOSUB 460 ELSE GOSUB 470
760 IF P(ST)<10 THEN A$=" 0"&STR$(P(ST))ELSE A$=
" "&STR$(P(ST))
770 ST=ST+1 :: IF SCREEN THEN GOSUB 480 ELSE GOS
UB 490
780 GOTO 540
790 IF I<10 THEN A$=" 0"&STR$(I)ELSE A$=" "&STR$(
I)
800 IF SCREEN THEN GOSUB 480 ELSE GOSUB 490
810 IF SCREEN THEN GOSUB 460 ELSE GOSUB 470
820 IF P(ST)<10 THEN A$=" 0"&STR$(P(ST))ELSE A$=
" "&STR$(P(ST))
830 ST=ST+1 :: IF SCREEN THEN GOSUB 480 ELSE GOS
UB 490
840 GOTO 540
850 A$=OUT$(P(ST)):: ST=ST+1 :: IF SCREEN THEN G
OSUB 480 ELSE GOSUB 490
860 IF SCREEN THEN GOSUB 460 ELSE GOSUB 470
870 IF P(ST)>40 THEN 890 ELSE A$="IND" :: ST=ST
+1 :: IF SCREEN THEN GOSUB 480 ELSE GOSUB 49
0
880 IF SCREEN THEN GOSUB 460 ELSE GOSUB 470
890 IF P(ST)<10 THEN A$=" 0"&STR$(P(ST))ELSE A$=
" "&STR$(P(ST))
900 ST=ST+1 :: IF SCREEN THEN GOSUB 480 ELSE GOS
UB 490
910 GOTO 700
920 INPUT "DO YOU WANT CLEAR (Y/N)? ":"A$ :: IF A
$<>"Y" THEN 140
930 FOR I=0 TO MX :: P(I)=0 :: NEXT I :: MX=0 :: :
GOTO 140
940 ****
950 !@P+
960 DATA 0,1,2,3,4,5,6,7,8,9,E',A,B,C,D,E,A',B',
C',D'
970 DATA CLR,2ND,INV,LNX,CE,CLR,2ND,INV,LOG,CP,T
AN,LRN,X>T,XA2,SQRT,1/X,PGM,P->R,SIN,COS
980 DATA IND,SST,STO,RCL,SUM,YAX,INS,CMS,EXC,PRD
,IXI,BST,EE,(,),/,DEL,ENG,FIX,INT
990 DATA DEG,GTO,PGM IND+EXC IND,PRD IND,X,PAUSE
,X=T,NOP,OP,RAD,SBR,STO IND,RCL IND,SUM IND,
-,LBL,X>T,SIGMA+,ABS
1000 DATA GRAD,RST,HIR,GTO IND,OP IND,+,ST FLG,IF
FLG,D.MS,PI,LIST,R/S,RTN (INV SBR),,,/-,-,=
,WRITE,DSZ,ADV,PRT

```

Lennart har valt ett utseende på listningen som nära ansluter till den som PC-100C producerar. Dock har vissa koder fått ett annorlunda utseende, vilket framgår av datasatserna i slutet av programmet (för den som inte provar själv).

Härmed lämnar vi lösningarna för den här gången och övergår till olösta problem.

#### TERMINALPROGRAM (49)

Från Björn Gustavsson kommer önskemål om ett program som möjliggör kommunikation med andra datorer. Björn önskar sig möjlighet att använda olika hastigheter (110-9600 baud) och gärna även olika hastigheter för sändning och mottagning (t.ex. inte helt ovanliga 1200/75).

Genom att utnyttja de möjligheter som RS232-kortets fasta programvara erbjuder kan man säkerligen få ett effektivt och bra program. Den som är intresserad kan i den tekniska beskrivningen se varför programmet ser ut som det gör. "The 9900 Family Data Book" talar om hur 9902 (Asynchronous Communications Controller, använd i RS232-kortet) fungerar. Den boken är för övrigt intressant för alla som är intresserade av 99:ans konstruktion.

#### HJÄLPPROGRAM FÖR TI 59 (45)

Till simuleringen av en TI 59 (problem 45c och 45d) kan man lägga simulering av modul på flexskiva. Om minnet räcker till går det naturligtvis bra att använda RAM till modulprogram, men då kan man ju (nästan) lika gärna utnyttja minnet till "vanlige" programsteg. Det är väl också lämpligt att utnyttja kassett eller diskett som magnetkort.

Jag har hört talas om vissa försök till ett simuleringsprogram. Problemen tycks vara störst när det gäller beräkningen av uttryck. Kanske någon har några tips att komma med i den frågan?

#### MERGE FRÅN KASSETT (50)

Flera personer, bl.a. Bror Nilsson och Tomas Åström önskar sig att MERGE ska fungera även från kassett. Extended BASIC tillåter som bekant MERGE, men endast från diskett. Min personliga uppfattning är att det går att åstadkomma från kassett, men utan expansionsminne är det knappast möjligt. Kanske någon har en annan uppfattning?

Bror Nilsson önskar sig också programvaran från Programming Aids II och III överförd till kassett för användning tillsammans med Extended BASIC eller Minimemory. Personligen har jag inte utnyttjat dessa program, men att modifiera dem så att de passer Extended BASIC torde inte vara någon konst. Däremot lär det inte vara inom lagens råmärken att norpa II:s program på det här sättet (även om "de lärde" förvisso tvistar om vad som egentligen är tillåtet). Därför tar jag inte upp detta som ett särskilt problem. Det finns naturligtvis inget som hindrar att någon skriver ett eget program som gör samma sak och presenterar det under Utmaningen.

När det gäller ett annat problem från Bror kan jag dock komma till omedelbar hjälp. Bror önskar sig nämligen en möjlighet att byta modul utan att återgång till titelbildet sker. Det är nämligen lätt ordnat. Varje modul är nämligen försedd med en kontakt som inte gör något annat än just RESET på processorn. (9900 gör då ett hopp till en adress som anges i en hoppvektor på 0000H i minnet, vilket i det här fallet är i monitor-ROM.) Genom att koppla bort denna RESET sker ingen återgång till titelbilden. Om man tar isär en modul och vänder lödsidan uppåt med kontakten mot sig har man RESET-kontakten längst till vänster. Den är förbunden med ett 100-ohms motstånd. Sätt en tejpbit över kontakten, och du kan sätta in moduler utan att titelbilden uppenbarar sig. Genom att ta bort motståndet får en mer permanent modifiering. Om man vill kan man ta bort motstånden i ALLA moduler som man har (de är ju lätt att sätta tillbaka om man så vill) och sätta en RESET-knapp på datorn i stället.

Observera dock att det inte är säkert att det går att byta modul utan att datorn hänger sig. Om man till exempel använder Extended BASIC och plötsligt rycker ut den modulen blir det tvärstopp, eftersom datorn då och då sysslar med att läsa GPL-program från modulen. Däremot går det bra att starta Forth med Extended BASIC, ta ut modulen och sätta in en annan "preparerad" modul om man t.ex. vill disassemblera ROM i den. Dessutom kan det inträffa att datorn får nippran av det faktum att man sätter in fler kretser, även om det inte sker någon RESET. I sådana fall får man vackert trycka på QUIT eller, i svårtalade fall, stänga av maskinen.

Härmed får det vara nog för den här gången. Ha det så bra!

# Mera Sorteringar

## Mera sorterings... ( urk )

Av Micael Dahlquist.

Redan i tidigare nummer av NITTINIAN ( och PROGRAMBITEN ) så här olika sorteringsmetoder diskuterats. Det finns många olika sätt att sortera tal på och den jag mest gillar är den så kallade BUBBEL-metoden.

Tekniken bakom BUBBEL-metoden är följande: talen som skall sorteras ställs på varandra i en stor stapel, en pekare ställs på de två översta talen därefter jämförs dessa och om det undre talet är lägre än det övre så byter dess plats. Pekaren ökas sedan på och pekar då på tal nummer 2 och tal nummer 3, dessa jämförs etc. När hela stapeln körts igenom testas en flagga, som blivit ställd endast om två tal bytt plats och om den är ställd kallas stapeln igen ( Detta betyder att när sista plats-bytet är gjort så körs ändå en sista koll ). Effekten blir att dom låga talen i stapeln liksom "bubblar" upp.

BUBBEL-metoden är definitivt inte den snabbaste metoden när det gäller sorterings och om man dessutom skriver rutinen i ( TI ) BASIC, så blir resultatet inte alltför snabbt. Därför tyckte jag att det skulle vara kul om man skrev det i FORTH och för att vara säker på att rutinen skrevs, så skrev jag den själv. Resultatet var bättre än väntat; enligt 99'er HOME COMPUTER magazine, July 1983, så sorterar nedanstående BASIC-program 100 tal på cirka 186 sekunder. Mitt FORTH-program sorterar samma talmängd på mindre än 20 sekunder !

Men först en titt på dess motsvarighet i BASIC ( Programmet är hämtat från 99'er HOME COMPUTER magazine, July 1983 ).

```
100 REM **BUBBLE SORT**
110 DIM A(50)
120 N=50
130 CALL CLEAR
140 REM **CHOOSE AND PRINT RANDOM ITEMS**
150 FOR I=1 TO N
160 RANDOMIZE
170 A(I)=INT(RND*100)+1
180 PRINT A(I);
190 NEXT I
200 PRINT ::::
210 REM **SORT RUTINE**
220 FLAG=0
230 FOR I=1 TO N-1
240 IF A(I)<=A(I+1)THEN 290
250 CHANGE=A(I)
260 A(I)=A(I+1)
270 A(I+1)=CHANGE
280 FLAG=1
290 NEXT I
300 IF FLAG=1 THEN 220
600 REM **PRINT SORTED ITEMS**
610 FOR I=1 TO N
620 PRINT A(I);
630 NEXT I
640 STOP
```

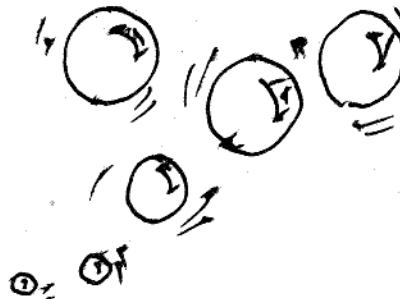
Då man först tittar lite flyktigt på programmet så ser man direkt tre olika skeden: gör massa slumptal ( rad 140-200 ), sortera ( rad 210-300 ) och skriv ut ( rad 600-640 ). Dessutom kan man i början av programmet ställa in önskat antal sorterade tal. Mitt FORTH-program följer därför också dessa tre skeden.

```
SCR #XX
0 ( BUBBEL-metoden ) DECIMAL
1 2 O VARIABLE A 50 ALLOT
3 4 HERE VARIABLE RND
5 6 : RANDOM RND E 31421 * 6927 + DUP RND ! ;
7 : CHOOSE RANDOM U* SWAP DROP ;
8
9 : RAND PAGE 20 O DO 100 CHOOSE DUP . CR A I 2* +
LOOP ;
10 : HEAD 1 19 O DO A I 2 * + DUP DUP E SWAP 2+ E 2DUP
> IF ROT
11 : ROT OVER 2+ ! ! O AND ELSE DROP DROP DROP THEN
LOOP ;
12 : PRINT 20 O DO 10 I GOTOXY A I 2 * + E . LOOP ;
13
14 : SORT1 RAND BEGIN HEAD UNTIL PRINT ;
15 : SORT2 RAND BEGIN HEAD PRINT KEY DROP UNTIL ;
```

Vad som direkt bör tilläggas är att rad nummer 4 till och med rad nummer 7 är hämtade från boken STARTING FORTH av Leo Brodie och kan väl fortfarande beställas av ABC-klubben.

Ordet SORT1 startar hela programmet; tjugo slumpmässigt valda tal skrivs ut på skärmen i en stapel, talen sorteras och skrivs sedan ut i en rad bredvid. Om man nu vill se hur det "bubblar" så använd ordet SORT2 som till skillnad från SORT1 stannar och skriver ut alla talen efter varje "stapel-koll", samt inväntrar att en knapp trycks ned.

Lycka till med tal-sorteringar i fortsättningen ! ■



## STRÄNGHANTERING

Här är två st. korta sub-program i Extended Basic för insättning resp. borttagning av en delsträng i/från en huvudsträng. På grund av hastighetsskäl har jag inte lagt till några kontroller av indata till subprogrammen.

```
1000 SUB INS(STRING$,INSERT$,POSI)
1010 P2=POSI-1
1020 LENGTH=LEN(STRING$)
1030 FIRST$=SEG$(STRING$,1,P2)
1040 LAST$=SEG$(STRING$,POSI,LENGTH-(LENGTH-P2))
1050 STRING$=FIRST$&INSERT$&LAST$
1060 SUBEND
```

```
2000 SUB DEL(STRING$,POSI,LENG)
2010 P2=POSI-1
2020 LENGTH=LEN(STRING$)
2030 FIRST$=SEG$(STRING$,1,P2)
2040 LAST$=SEG$(STRING$,POSI+LENG,LENGTH-(P2+LENG))
2050 STRING$=FIRST$&LAST$
2060 SUBEND
```

Bengt Fahlgren

71/59

# n-te-gradspolynom

Raul Hofman

PARGAS 2/1-84 Hej Programbiten

I ett nummer av PB efterfrågades det efter ett lämpligare PGM än ML - 07 för att beräkna värdet av ett n:te grads polynom utan begränsning av gradantalet. Jag bifogar två versioner. En rymmer i dosa som dosa, en för dem som har printer och kör med TI-59.

Matematiken är samma som beskrivs i ML-07 metodbeskrivning fastän i omvänt ordning. Ingen reservation för gradantalet. Två behövliga minnen. En delkalkyl ledande spikrakt mot målet sker vid varje inknappling av koeficienterna, från den högsta till den lägsta. Härvid är man inte medveten om att hela uträkningen är utförd då den 0:te termen slås in (54:grads eller 1024:grads polynom).

$$y(x) = a(0)x^0 + a(1)x^1 + \dots + a(n)$$

Ett sifferexempel:

Vad är  $f(-3)$  då  $f(x) = 2x^3 - 3x^2 + 5x - 8$  ?

2	-3	5	-8
-3	-6	27	-96

-----  
-9      32    -104    svar: -104

Blev det klart?  
 $-3 * 2 = -6$   
 $-6 + (-3) = -9$   
 $-3 * (-9) = 27$   
 $27 + 5 = 32$   
 $-3 * 32 = -96$   
 $-96 + (-8) = -104$

Programbeskrivning version 1:

- Knappa in programmets 26 steg.
  - Initiera med RST.
  - $x(0)$  följt av R/S.
  - $a(0)$  följt av R/S.
  - $a(1)$  följt av R/S.
- O.S.V.
- N+2. E ger  $f(x_0)$  omedelbart.

n-te gr. v.l.  
 000 91 R/S  
 001 42 STD  
 002 00 00  
 003 65 X  
 004 91 R/S  
 005 85 +  
 006 91 R/S  
 007 95 =  
 008 42 STD  
 009 01 01  
 010 65 X  
 011 43 RCL  
 012 00 00  
 013 85 +  
 014 91 R/S  
 015 95 =  
 016 61 GTD  
 017 00 00  
 018 07 07  
 019 76 LBL  
 020 15 E  
 021 25 CLR  
 022 43 RCL  
 023 01 01  
 024 47 CMS  
 025 81 PST

Programbeskrivning version 2:

- Initiering med E' utskriver text som begär x-värdet.
  - Utskriver x-värdet och kvitterar samt begär en koefficient, allt via R/S.
  - Kvitterar föregående koefficient och begär följande, R/S
- ..... O.S.V.

E ger  $f(x)$  kvitterat. Nollställer och frågar efter ett nytt  $x(0)$ .

PS. Priset på snabbhet i detta PGM. är självfallet den att man inte kan beräkna samma polynoms värde för andra x utan att köra koeficienterna på nytt.

1 PROGRAMMERARE: RAUL HOFMAN		R/S	
n-te-gradspolynom			
		INIT	

## n-te-gradspolynom version 2

000	76	LBL	060	02	02	120	05	5
001	10	E'	061	01	1	121	06	6
002	98	RDV	062	07	7	122	69	DP
003	04	4	063	00	0	123	04	04
004	04	4	064	00	0	124	43	RCL
005	00	0	065	00	0	125	00	00
006	00	0	066	00	0	126	65	X
007	03	3	067	69	DP	127	91	R/S
008	02	2	068	03	03	128	69	DP
009	01	1	069	69	DP	129	06	06
010	05	5	070	05	05	130	95	=
011	02	2	071	69	DP	131	42	STD
012	03	3	072	00	00	132	01	01
013	69	DP	073	98	ADV	133	85	+
014	01	01	074	04	4	134	91	R/S
015	01	1	075	04	4	135	69	DP
016	03	3	076	06	6	136	06	06
017	05	5	077	04	4	137	95	=
018	05	5	078	69	DP	138	65	X
019	03	3	079	01	01	139	43	RCL
020	01	1	080	69	DP	140	00	00
021	05	5	081	05	05	141	95	=
022	06	6	082	25	CLR	142	61	GTD
023	69	DP	083	91	R/S	143	01	01
024	02	02	084	42	STD	144	31	31
025	04	4	085	00	00	145	76	LBL
026	02	2	086	04	4	146	15	E
027	02	2	087	04	4	147	98	RDV
028	04	4	088	00	0	148	02	2
029	01	1	089	00	0	149	01	1
030	03	3	090	69	DP	150	05	5
031	00	0	091	04	04	151	05	5
032	00	0	092	43	RCL	152	04	4
033	69	DP	093	00	00	153	04	4
034	03	03	094	69	DP	154	05	5
035	03	3	095	06	06	155	06	6
036	05	5	096	69	DP	156	69	DP
037	06	6	097	00	00	157	04	4
038	03	3	098	01	1	158	53	C
039	03	3	099	03	3	159	43	RCL
040	06	6	100	05	5	160	01	01
041	00	0	101	05	5	161	55	÷
042	00	0	102	03	3	162	43	RCL
043	69	DP	103	01	1	163	00	00
044	04	04	104	05	5	164	54	)
045	69	DP	105	06	6	165	69	DP
046	05	05	106	06	6	166	06	06
047	69	DP	107	04	4	167	69	DP
048	00	00	108	69	DP	168	00	00
049	03	3	109	01	01	169	98	RDV
050	06	6	110	69	DP	170	98	RDV
051	04	4	111	05	05	171	98	RDV
052	02	2	112	69	DP	172	47	CMS
053	01	1	113	00	00	173	25	CLR
054	03	3	114	01	1	174	61	GTD
055	03	3	115	03	3	175	00	00
056	05	5	116	05	5	176	74	74
057	06	6	117	05	5	177	91	R/S
058	02	2	118	03	3			
059	69	DP	119	01	1			

► 16

# TI 59

## PLANETARIUM

1	PROGRAMMERARE: RAUL HOFMAN	2
PLANETARIUM	9.12.83, A.C.	
φ (0)	φ (23)	T → INIT
GHA (0)	GHA (23)	AT START

PARGAS 27/12-83 Hej Programbiten

I PB 81-1 sid 3, skrev Paul Schlyter och eftertraktade ett snabbare planetprogram. Jag har denna gång begränsat programmet till Venus, Mars, Jupiter, Saturnus, stjärna och månen. Allt baserar sig på en lämplig approximation av deklinationen och timvinkel som endast för månens del är en aning optimistisk. Ni behöver Nautisk Almanack (Svensk Sjöfartskalender) och knappar in timvinkel och deklination (siderisk timvinkel + vårdagjämningens timvinkel och deklination för en stjärna) för 0h och 23h. Programmet lägger räta linjer genom ändpunkterna och använder dessa som genererare av behövliga timvinkelar och deklinationer. Under körningen skrives tiden i GMT ut (HH.MMss), höjden (DD.MMss) + text, bäringen (kompasskurser till objektet) (DDD.d).

Beräknandet av varje talvärde tar ca 11 s, d.v.s ett värdepar 22 sek. Jag har inte försökt sporta med att minimera tiden då man uppenbarligen skulle förlora i överskådighet.

Jag har även en Sharp EL-5103 som med sina 48 steg precis rymmer höjd - och asimutformlerna. Med den kommer man ned till 11 sek för ett värdepar för ett objekt. (Inmatning fi, delta, t --> h, A).

Jag övergår nu till programbeskrivningen. Sedan får programmet tala för sig själv. Observera att timvinkeln för 23h måste ökas med 360 grader emedan det finns en nollpunkt mellan 0h och 23h men det inverkar ju inte på de trigonometriska funktionerna (mod 360 grader).

### PLANETARIUM

- INIT med begynnelsetiden T i GMT via E'
- GHA(0) via A
- delta(0) via A'
- GHA(23) + 360 grader via B
- delta(23) via B'

Stegen 2 - 5 upprepas sammanlagt sex gånger för Venus, Mars, Jupiter, Saturnus, Stjärna, Månen. Vänta någon sekund efter varje B' så att parametrarna hinner uträknas och placeras i rätta minnen.

- Latituden fi via C (GG.MMss)
- Longituden lambda via D (obs +E, -W i motsats till amerikanskt förfarande)
- Delta T via E (HH.MMss, t.ex .15 om du vill ha värden varje kvart).

Efter 8 startar PGM och T skrives ut jämta höjder namn på objekten och bäringen. Sedan följer T + DeltaT o.s.v.

Raul Hofman

Kör exempel: Se tabell ur Nautical Almanac sid 110+111, vid programmet PARABELANPASSNING.

KNAPPA IN	TRYCK
1. INITiering med	0 E'
2. GHA(0) VENUS	185.2512 A
3. delta(0) VENUS	N 20.3148 A'
4. GHA(23) VENUS	170.0806 + 360 = B
5. delta(23)VENUS	N 20.4648 B'
2. GHA(0) MARS	25.5112 A
3. delta(0) MARS	S 17.1012 A'
4. GHA(23) MARS	12.0354 + 360 = B
5. delta(23)MARS	S 17.0754 B'
2. GHA(0) JUPI	325.0642 A
3. delta(0) JUPI	S 22.4824 A'
4. GHA(23) JUPI	311.0854 + 360 = B
5. delta(23)JUPI	S 22.4900 B'
2. GHA(0) SATU	28.0642 A
3. delta(0) SATU	S 12.4448 A'
4. GHA(23) SATU	14.0654 + 360 = B
5. delta(23)SATU	S 12.4348 B'
STAR = ARCTURUS	
2. GHA(0)STAR=SHA(STAR)	146.1542(DMS)+
	CHA(ARIES) 247.4112(DMS)= INVDMDS A
3. delta(0) STAR	N 19.1554 A'
4. GHA(23)STAR=SHA(STAR)	146.1542(DMS)+
	CHA(ARIES) 233.3754(DMS)+360=INVDMDS B
5. delta(23)STAR	N 19.1554 B'
2. GHA(0) MONE	189.2336 A
3. delta(0) MONE	N 20.3148 A'
4. GHA(23) MONE	162.1912 + 360 = B
5. delta(23)MONE	22.5436 B'

### OBS-ORT STOCKHOLM

- LAT fi N 59.31 C
- LONG lambda W 18.21 D
- deltaT 1 E

Därmed startar beräkningarna och inom kort hörs ett PRRRRRRT.

OBS! Decimalgradsdelen i tabellen multipliceras med 6 varvid sekunder erhålls.  
Exempel: delta(0)MONE N 20° 31.8 --> N 20.3148

1	PROGRAMMERARE: RAUL HOFMAN	2
PARABELANPASSNING INMATNING DATA		
φ (0)	φ (11)	φ (23)
GHA (0)	GHA (11)	GHA (23)

### PROGRAM 2

1	PROGRAMMERARE: RAUL HOFMAN	2
PARABELANP. TESTPROGRAM 1-8 objekt		
TEST		TEXT

### PROGRAM 3

1	PROGRAMMERARE: RAUL HOFMAN	2
PARABELANPASSNING 1-8 objekt BERÄKNING		
LAT, φ	LONG, λ	GMT

# TI 59

DBS-ORT PLANETARIUM Utskrift för maj 30 1984 varje timme. Plats Stockholm.

59.58333333  
18.5 LAT LONG

maj 31

0.		4.		8.		12.		16.		20.		24.
-7.3133	VNUS	14.3717	VNUS	43.1932	VNUS	47.324	VNUS	20.5345	VNUS	-4.2118	VNUS	-7.194
22.5		74.2		130.6		214.		275.6		326.2		22.2
5.1419	MARS	-22.0537	MARS	-46.0901	MARS	-36.3658	MARS	-7.1905	MARS	12.2146	MARS	4.5031
222.1		273.6		338.8		57.3		111.6		165.7		223.3
6.3055	JUPI	0.0854	JUPI	-26.3305	JUPI	-51.3346	JUPI	-41.5536	JUPI	-12.3331	JUPI	6.3837
164.8		219.6		269.7		336.1		60.		113.9		165.8
8.3356	SATU	-19.2628	SATU	-42.0955	SATU	-31.4427	SATU	-2.2826	SATU	16.5848	SATU	8.1155
225.8		278.		342.7		57.2		111.2		167.2		226.8
35.0925	STAR	5.4659	STAR	-10.5548	STAR	-0.1199	STAR	28.101	STAR	49.2327	STAR	34.4153
246.3		298.8		353.1		48.9		100.		170.2		247.3
-7.3235	MONE	15.155	MONE	43.2602	MONE	48.4813	MONE	24.0309	MONE	-0.4949	MONE	-6.2446
26.4		75.9		130.3		211.4		272.7		321.3		14.1
1.		5.		9.		13.		17.		21.		25.
-3.4813	VNUS	22.055	VNUS	48.1236	VNUS	42.2139	VNUS	13.2935	VNUS	-7.4635	VNUS	-3.384
36.1		86.7		149.5		232.4		288.		339.9		35.8
-0.2844	MARS	-29.3419	MARS	-47.3416	MARS	-29.4338	MARS	-0.3555	MARS	13.1651	MARS	-0.5822
235.3		287.2		359.8		72.4		124.4		180.5		236.4
7.3608	JUPI	-5.1928	JUPI	-34.0616	JUPI	-53.1328	JUPI	-34.5345	JUPI	-5.5848	JUPI	7.3556
178.8		232.4		283.2		358.8		75.3		126.3		179.8
2.3157	SATU	-26.4708	SATU	-43.0758	SATU	-24.5222	SATU	4.1514	SATU	17.3923	SATU	2.0525
239.2		291.8		2.7		72.		124.3		182.6		240.2
27.5221	STAR	-0.2533	STAR	-10.5331	STAR	6.0291	STAR	35.2555	STAR	49.1421	STAR	27.2248
260.6		311.6		7.5		61.6		114.3		192.1		261.5
-3.2547	MONE	22.3734	MONE	48.2003	MONE	44.1007	MONE	16.5636	MONE	-4.3627	MONE	-3.4117
39.4		88.		148.5		229.8		284.9		334.1		27.2
2.		6.		10.		14.		18.		22.		26.
1.2025	VNUS	29.3958	VNUS	50.4811	VNUS	35.4648	VNUS	6.3548	VNUS	-9.2842	VNUS	1.2753
49.2		99.8		170.8		248.4		300.4		354.		48.8
-7.1034	MARS	-36.284	MARS	-46.1234	MARS	-22.1513	MARS	5.0857	MARS	12.1453	MARS	-7.4414
248.1		302.3		20.8		86.1		137.6		195.2		249.2
6.5031	JUPI	-11.4852	JUPI	-41.1253	JUPI	-51.5239	JUPI	-27.2203	JUPI	-0.2304	JUPI	6.4224
192.7		244.8		298.3		21.6		88.9		139.		193.7
-4.2343	SATU	-33.2433	SATU	-41.2828	SATU	-17.2504	SATU	9.591	SATU	16.182	SATU	-4.531
252.2		307.		22.4		85.5		137.9		197.9		253.1
20.1659	STAR	-5.2914	STAR	-8.5801	STAR	13.0523	STAR	41.5055	STAR	46.1943	STAR	19.4706
273.7		324.9		21.8		74.2		130.6		212.8		274.6
1.5616	MONE	30.0219	MONE	51.0757	MONE	38.0603	MONE	10.1343	MONE	-6.5308	MONE	0.2925
51.9		100.7		169.		245.8		296.8		347.3		39.8
3.		7.		11.		15.		19.		23.		27.
7.3529	VNUS	36.5502	VNUS	50.3345	VNUS	28.2727	VNUS	0.3218	VNUS	-9.1929	VNUS	
61.8		114.2		193.		262.6		313.1		368.		
-14.2999	MARS	-42.1405	MARS	-42.203	MARS	-14.3919	MARS	9.3433	MARS	9.2122	MARS	
260.7		319.4		40.2		99.		151.4		209.5		
4.1741	JUPI	-19.0001	JUPI	-47.1729	JUPI	-47.5059	JUPI	-19.4755	JUPI	3.5458	JUPI	
206.4		257.1		315.8		42.2		101.6		152.2		
-11.5039	SATU	-38.4447	SATU	-37.2857	SATU	-9.4918	SATU	14.2035	SATU	13.0329	SATU	
265.		323.9		40.8		98.5		152.2		312.7		
12.4801	STAR	-9.0439	STAR	-5.1847	STAR	20.3502	STAR	46.45	STAR	41.1357	STAR	
286.3		338.8		35.6		86.8		149.3		231.2		
8.1605	MONE	37.0753	MONE	51.1744	MONE	31.1356	MONE	4.1226	MONE	-7.3013	MONE	
64.		114.6		190.6		259.9		308.9		347.3		

## PLANETARIUM

000	76	LBL	060	69	DP	120	00	00	180	39	CDS	240	06	6	300	71	SBR	360	00	00	420	02	02
001	10	E'	061	35	35	121	01	1	181	65	X	241	69	DP	301	03	03	361	85	+	421	03	3
002	42	STD	062	97	DSZ	122	00	0	182	43	RCL	242	04	04	302	72	72	362	43	RCL	422	06	6
003	00	00	063	06	06	123	42	STD	183	03	03	243	61	GTO	303	69	DP	363	06	06	423	02	2
004	01	1	064	00	00	124	08	08	184	39	CDS	244	03	03	304	06	06	364	95	=	424	00	0
005	00	0	065	38	38	125	71	SBR	185	95	=	245	00	00	305	43	RCL	365	22	INV	425	03	3
006	42	STD	066	91	R/S	126	00	00	186	22	INV	246	02	02	306	02	02	366	88	DMS	426	02	02
007	07	07	067	53	(	127	67	67	187	38	SIN	247	05	5	307	38	SIN	367	42	STD	427	03	3
008	91	R/S	068	22	INV	128	71	SBR	188	42	STD	248	04	4	308	75	-	368	00	00	428	07	7
009	76	LBL	069	86	STF	129	00	00	189	04	04	249	01	1	309	43	RCL	369	61	GTO	429	03	3
010	11	A	070	40	IND	130	79	79	190	22	INV	250	03	3	310	01	01	370	01	01	430	07	7
011	88	DMS	071	07	07	131	85	+	191	88	DMS	251	03	3	311	38	SIN	371	11	11	431	69	DP
012	42	STD	072	69	DP	132	43	RCL	192	42	STD	252	02	2	312	65	X	372	53	(	432	03	3
013	03	03	073	27	27	133	05	05	193	09	09	253	04	4	313	43	RCL	373	53	(	433	69	DP
014	91	R/S	074	86	STF	134	95	=	194	69	DP	254	69	DP	314	04	04	374	43	RCL	434	05	05
015	76	LBL	075	40	IND	135	42	STD	195	00	00	255	04	04	315	38	SIN	375	09	09	435	02	2
016	16	A'	076	07	07	136	03	03	196	87	IFF	256	61	GTO	316	95	=	376	65	X	436	07	7
017	88	DMS	077	54	)	137	71	SBR	197	01	01	257	03	03	317	55	+	377	01	1	437	01	1
018	42	STD	078	92	RTN	138	00	00	198	02	02	258	00	00	318	43	RCL	378	00	0	438	03	3
019	01	01	079	53	(	139	79	79	199	20	20	259	03	3	319	04	04	379	45	YX	439	03	3
020	91	R/S	080	73	RC*	140	42	STD	200	87	IFF	260	06	6	320	39	CDS	380	04	4	440	07	7
021	76	LBL	081	08	08	141	02	02	201	02	02	261	01	1	321	55	+	381	85	+	441	00	0
022	12	B	082	69	DP	142	03	3	202	02	02	262	03	3	323	01	01	382	93	.	442	00	0
023	88	DMS	083	28	28	143	06	6	203	33	33	263	03	3	323	01	01	383	05	5	443	69	DP
024	42	STD	084	85	+	144	00	0	204	87	IFF	264	07	7	324	39	CDS	384	54	)	444	04	04
025	04	04	085	73	RC*	145	32	XIT	205	03	03	265	04	4	325	95	=	385	59	INT	445	43	RCL
026	91	R/S	086	08	08	146	43	RCL	206	02	02	266	01	1	326	22	INV	386	55	+	446	01	01
027	76	LBL	087	65	X	147	03	03	207	46	46	267	69	DP	327	39	CDS	387	01	1	447	69	DP
028	17	B'	088	43	RCL	148	77	GE	208	87	IFF	268	04	04	328	42	STD	388	00	0	448	06	06</td

**TI 59**

# PARABELANPASSNING

PARGAS 12/1-84 Hej Lars!

Vi har inte tidigare haft kontakt emedan jag är ny medlem med koden Q205. Däremot har jag skickat tre program till Claes Schibler. Själv var jag ganska nöjd med dem, då det inte tagit så förskräckligt lång tid att göra dem. Nu kommer jag däremot med tre nya av grövsta kaliber. De är kollade och kollade och åter kollade.

Taket på 59-an kom emot då jag ville ha nyttas av PC-100. Det matematiska har jag löst själv och deklarerar endast kort att både deklination som timvinkel inom ett dygn följer en parabel

$at^2 + bt + c$ . (3 olika fall)

Denna approximation är märkt nog helt tillfyllest för månen, och felgränsen är i allmänhet  $\pm 0,1'$ , d.v.s. Nautical Almanacs noggrannhet. Jag bifogar två fotostatkopior ur Nautical Almanac 1984 så att du själv kan prova.

## 1) Programbeskrivning:

1. Läs in bågge sidorna av PGM Parabelanpassning.
2. Följ instruktionerna du erhåller med E': (se remsan). Vid C' sker beräkning och omplacering av a(1), b(1), c(1) och a(2), b(2), samt c(2).
3. Ovanstående upprepas med följande objekt.  
o.s.v.
4. Som ovan.

Du får fritt välja mellan 1 - 8 objekt. Programmet determinerar i normaluppsättning max. 16 parabler.

2) Eftersom du är skeptisk angående ovanstående överläser du program 1, bevarande minnena fylda, med kontrollprogrammet (sida 1 och 2) följande anvisningar du erhåller då du trycker E' (snopet va?). Om du tryckt in GHA och DELTA för 8 objekt rekommenderar jag att du går på kaffe och låter skrivaren göra sitt, nämligen prestera 16 rader med 24 värden per lodrät rad. Den stannar nog av sig själv för att PGM 1 och PGM 2 knyts ihop med R09.

3) Nu tror du det är värt att ta reda på vilken höjd DD.MM (observera Nautical Almanacs format) och bärings (riktning), N=0, E=90, S=180, W=270.

Du överläser med PGM 3. E' säger vad du skall trycka in (se remsan). Huvudprogrammet är en oändlig loop som håller reda på tiden utskrivande GMT i OP 04 och ökar denne tid T med valt tillskott. Likaså behöver man inte hålla reda på hur många objekten ursprungligen var ty R09 är med på noterna. För att skilja mellan objekten har jag kallat dem 1, 2, 3, 4, 5, 6, 7, och 8 med tillhörande beteckningar H1, H2, ..., H8.

Raul Hofman

Jag är övertygad att detta program kommer att glädja dem som är intresserade av praktisk (men noggrann) astronomi. Kanske programmet rent av kan kallas vetenskapligt - tekniskt. Hur det än är blir jag mer och mer imponerad av 59:ans möjligheter och AOS förträfflighet.

Ett litet komplement till inmatningsanvisningen: Även dagarna kan varieras.

Nedan en fingervisning:

objekt	dygn	objekt	dygn	objekt	dygn
8	1				
6	1	1	2		
5	1	1	3		
4	2				
4	1	2	2		
4	1	1	4		
3	2	2	1		
2	3	2	1		
2	4				
1	5	3	1		
1	3	2	2	1	1
1	6	2	1		
1	4	2	2		
1	5	1	2	1	1
1	6	1	2		
1	5	1	3		
1	8				

PARABELANPASSNING  
E INIT  
A GHA 0  
A' ^ 0  
B GHA11  
B' ^ 11  
C GHA23  
C' ^ 23  
KOMB. MED 1-8 DBJ.  
R9, R11-R58

TESTPROGRAM TILL  
PARABELANPASSNING  
MED R9, R11-R58 (MAX)  
FYLLDA, UTSKRIVER GHA  
OCH ^.. TRYCK ENDAST A

1-8 DBJ. E: INIT  
A: LAT  
B: LONG  
C: GMT  
D: ^T

Program 3 E'

G.M.T.	ARIES			VENUS			-3.4			MARS			-1.5			JUPITER			-2.1			SATURN			+0.5			STARS				
	G.H.A.	Dec.	G.H.A.	Dec.	G.H.A.	Dec.	G.H.A.	Dec.	G.H.A.	G.H.A.	Dec.	G.H.A.	G.H.A.	Dec.	G.H.A.	Dec.	G.H.A.	Dec.	G.H.A.	Dec.												
20 00 237 412	185 252	+520 310	3	25 512 517	102	325 267 522	-48.4	25 667 512	448	43.5	345 094	43.5	315 365	549	222	Acamer	315 365	549	222	447	47.0	235 523	104	19 216 106	119	195	94 565	5	94 565	5	94 565	5
01 232 457	200 245	325	101	340 54	40	355 121	100	355 121	48.5	56 120	44.7	48.5	120	44.7	01	Achenor	315 365	549	222	447	47.0	235 523	104	19 216 106	119	195	94 565	5	94 565	5	94 565	5
02 277 462	215 337	311	01	340 54	40	355 121	100	355 121	48.5	56 120	44.7	48.5	120	44.7	01	Adabora	255 302	522	017	446	47.0	235 523	104	19 216 106	119	195	94 565	5	94 565	5	94 565	5
03 302 486	220 230	33	308	71	007	098	25 176	48.5	38 172	44.6	48.5	172	44.6	01	Adabora	291 153	N1	251	446	47.0	235 523	104	19 216 106	119	195	94 565	5	94 565	5	94 565	5	
04 307 522	245 223	345	86 19	075	069	25 176	48.5	38 172	44.6	48.5	172	44.6	01	Adabora	290 215	N1	251	446	47.0	235 523	104	19 216 106	119	195	94 565	5	94 565	5	94 565	5		
05 312 552	230 215	351	101	019	074	07	40 19	48.5	40 19	48.5	40 19	48.5	01	Alith	166 394	N56	63	106	119	195	100	120	357	90 556	5	90 556	5	90 556	5			
06 337 560	275 208	N62	310	116	020	102	517	096	55 229	522	48.6	118 224	44.5	103 297	521	48.6	103 297	521	48.6	103 297	521	48.6	103 297	521	48.6	103 297	521	48.6	103 297	521		
W 07 332 565	280 200	364	131	134	015	075	07	50 256	48.6	131 25	44.5	131 25	44.5	131 25	44.5	131 25	44.5	131 25	44.5	131 25	44.5	131 25	44.5	131 25	44.5	131 25	44.5	131 25	44.5			
A 08 301 364	205 193	320	185	..	131	146	084	85 283	48.6	166 148	277	166 148	277	166 148	277	166 148	277	166 148	277	166 148	277	166 148	277	166 148	277	166 148	277	166 148	277			
D 09 305 348	210 182	321	80	122	459	281	450	085	45 263	48.6	165 160	311	165 160	311	165 160	311	165 160	311	165 160	311	165 160	311	165 160	311	165 160	311	165 160	311				
N 10 308 345	213 178	323	87	176	177	099	093	115 338	48.7	176 329	32	176 329	32	176 329	32	176 329	32	176 329	32	176 329	32	176 329	32	176 329	32	176 329	32					
E 11 308 345	210 170	320	86	191	192	084	084	120 365	48.7	192 165	36	192 165	36	192 165	36	192 165	36	192 165	36	192 165	36	192 165	36	192 165	36	192 165	36					
F 12 303 357	206 292	S10	050	145 392	522	487	070	206 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3					
G 13 303 357	208 292	S10	050	145 392	522	487	070	208 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3					
H 14 303 357	213 292	S10	050	145 392	522	487	070	213 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
I 15 303 357	218 292	S10	050	145 392	522	487	070	218 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
J 16 303 357	223 292	S10	050	145 392	522	487	070	223 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
K 17 303 357	228 292	S10	050	145 392	522	487	070	228 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
L 18 303 357	233 292	S10	050	145 392	522	487	070	233 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
M 19 303 357	238 292	S10	050	145 392	522	487	070	238 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
N 20 303 357	243 292	S10	050	145 392	522	487	070	243 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
O 21 303 357	248 292	S10	050	145 392	522	487	070	248 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
P 22 303 357	253 292	S10	050	145 392	522	487	070	253 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
Q 23 303 357	258 292	S10	050	145 392	522	487	070	258 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
R 24 303 357	263 292	S10	050	145 392	522	487	070	263 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
S 25 303 357	268 292	S10	050	145 392	522	487	070	268 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
T 26 303 357	273 292	S10	050	145 392	522	487	070	273 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
U 27 303 357	278 292	S10	050	145 392	522	487	070	278 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
V 28 303 357	283 292	S10	050	145 392	522	487	070	283 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
W 29 303 357	288 292	S10	050	145 392	522	487	070	288 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
X 30 303 357	293 292	S10	050	145 392	522	487	070	293 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
Y 31 303 357	298 292	S10	050	145 392	522	487	070	298 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
Z 32 303 357	303 292	S10	050	145 392	522	487	070	303 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
A 33 303 357	308 292	S10	050	145 392	522	487	070	308 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
B 34 303 357	313 292	S10	050	145 392	522	487	070	313 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
C 35 303 357	318 292	S10	050	145 392	522	487	070	318 292	510	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3	122 443	44.3							
D 36 303 357	323 292	S10																														

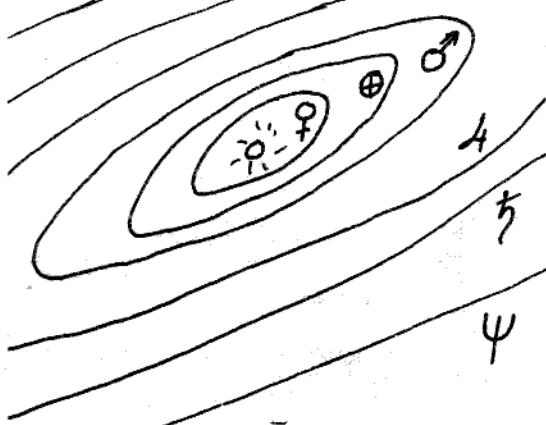
# nittinian EDITION

TI 59  
PÅ  
TI-99 4A



samt

PLANETARIUM



## Innehåll

Redaktören	2	
Medlemsmöte i Lund	2	
Ordförande	3	
Utmänning	4-7	
Sortering	8	
TI-59	n-te gradspolynom	9
	Planetarium	10-11
	Parabelanpassning	12-16
Favorit i repris	17	
Ta det piano	17	
Tips och Tricks	18	
Polsk notation	19	
Assembler skolan del 2	20-21	
Datakommunikation med modem	22	
Datavision	22	
Annons	23	
Programbanken	24	

ISSN 0281-1146

# TI 59

Utskrift av testprogrammet. Jämför med kopian ur Nautical Almanac.  
STAR får ni plussa ihop själva = (ARIES + ARCTURUS)

## STAR=ARKTURUS

		MÅNE	SOL	ARIES	VENUS	MARS	JUPIT	SATURN	STAR
16.	00								
61.	01								
8.	02								
.1067068328	03								
0.	04								
10.	05								
59.	06								
0.	07								
13.	08								
0.	09								
10.									
189. 3933333	11	189. 236	180. 384	247. 412	185. 252	25. 512	325. 067	28. 067	60. 569
14. 49322903	12	203. 531	195. 383	262. 437	200. 245	40. 544	340. 094	43. 092	72. 281
.0007894159	13	218. 226	210. 382	277. 462	215. 237	55. 576	355. 121	58. 117	84. 12
19. 65833333	14	232. 52	225. 382	292. 486	230. 23	71. 007	10. 148	73. 143	96. 088
.1616798419	15	247. 212	240. 381	307. 511	245. 222	86. 039	25. 175	88. 169	108. 184
.00008827404	16	261. 504	255. 38	322. 536	260. 215	101. 071	40. 203	103. 195	120. 408
180. 64	17	276. 195	270. 379	337. 561	275. 207	116. 103	55. 23	118. 221	133. 16
14. 99871487	18	290. 484	285. 378	352. 585	290. 2	131. 134	70. 257	133. 247	146. 04
.0000071366	19	305. 173	300. 378	8. 01	305. 192	146. 166	85. 284	148. 274	159. 048
21. 765	20	319. 461	315. 377	23. 035	320. 195	161. 198	100. 311	163. 301	172. 185
.0063268555	21	334. 148	330. 376	38. 059	335. 177	176. 229	115. 338	178. 328	185. 449
.0000104304	22	348. 434	345. 375	53. 084	350. 17	191. 261	130. 365	193. 355	199. 241
247. 6866667	23	3. 119	0. 374	68. 109	5. 163	206. 293	145. 392	208. 382	213. 161
15. 04132686	24	17. 403	15. 373	83. 133	20. 155	221. 324	160. 419	223. 41	227. 21
.00000104304	25	32. 086	30. 372	98. 158	35. 148	236. 356	175. 446	238. 438	241. 386
0.	26	46. 368	45. 371	113. 183	50. 14	251. 387	190. 473	253. 466	256. 091
0.	27	61. 05	60. 371	128. 207	65. 133	266. 419	205. 5	268. 495	270. 523
0.	28	75. 33	75. 37	143. 232	80. 125	281. 45	220. 527	283. 523	285. 484
185. 42	29	90. 009	90. 369	158. 256	95. 118	296. 482	235. 554	298. 552	300. 572
14. 98754556	30	104. 288	105. 368	173. 281	110. 111	311. 513	250. 581	313. 581	316. 189
.0000027448	31	118. 565	120. 367	188. 305	125. 103	326. 545	266. 008	329. 01	331. 533
20. 53	32	133. 242	135. 366	203. 33	140. 096	341. 576	281. 035	344. 04	347. 406
.0109453228	33	147. 517	150. 365	218. 354	155. 088	357. 008	296. 062	359. 069	3. 407
.0000032938	34	162. 192	165. 364	233. 379	170. 081	12. 039	311. 089	14. 099	19. 536
25. 85333333	35								
15. 05305995	36	19. 395	21. 459	0.	20. 318	-17. 101	-22. 483	-12. 447	19. 159
.0. 0000164649	37	19. 491	21. 463	0.	20. 325	-17. 1	-22. 483	-12. 447	19. 167
-17. 17	38	19. 587	21. 467	0.	20. 331	-17. 1	-22. 484	-12. 446	19. 175
.0016666667	39	20. 081	21. 47	0.	20. 338	-17. 098	-22. 484	-12. 446	19. 183
-2.	40	20. 175	21. 474	0.	20. 344	-17. 097	-22. 484	-12. 445	19. 192
325. 1116667	41	20. 267	21. 478	0.	20. 351	-17. 096	-22. 484	-12. 445	19. 202
15. 04522398	42	20. 358	21. 482	0.	20. 357	-17. 095	-22. 485	-12. 444	19. 212
.0. 0000065876	43	20. 448	21. 485	0.	20. 364	-17. 094	-22. 485	-12. 444	19. 222
-22. 80666667	44	20. 537	21. 489	0.	20. 37	-17. 093	-22. 485	-12. 443	19. 233
.0. 0004726614	45	21. 025	21. 493	0.	20. 377	-17. 092	-22. 485	-12. 443	19. 244
.0. 0000016469	46	21. 112	21. 496	0.	20. 383	-17. 091	-22. 486	-12. 442	19. 256
28. 11166667	47	21. 198	21. 5	0.	20. 39	-17. 09	-22. 486	-12. 442	19. 268
15. 04165569	48	21. 283	21. 504	0.	20. 397	-17. 09	-22. 486	-12. 442	19. 281
.0. 0001806015	49	21. 367	21. 507	0.	20. 403	-17. 088	-22. 487	-12. 441	19. 294
-12. 74666667	50	21. 449	21. 511	0.	20. 41	-17. 087	-22. 487	-12. 441	19. 308
.0. 000787769	51	21. 531	21. 515	0.	20. 416	-17. 086	-22. 487	-12. 44	19. 322
.0. 0000027448	52	22. 012	21. 518	0.	20. 423	-17. 085	-22. 487	-12. 44	19. 337
60. 94805556	53	22. 091	21. 522	0.	20. 429	-17. 084	-22. 488	-12. 44	19. 352
11. 41291676	54	22. 17	21. 525	0.	20. 436	-17. 083	-22. 488	-12. 44	19. 367
.0. 1067068328	55	22. 247	21. 529	0.	20. 442	-17. 082	-22. 488	-12. 439	19. 383
19. 265	56	22. 323	21. 532	0.	20. 449	-17. 081	-22. 488	-12. 438	19. 4
.0. 0122858214	57	22. 399	21. 536	0.	20. 455	-17. 08	-22. 489	-12. 438	19. 417
.0. 0003890759	58	22. 473	21. 539	0.	20. 462	-17. 079	-22. 489	-12. 437	19. 434
0.	59	22. 546	21. 543	0.	20. 468	-17. 078	-22. 49	-12. 437	19. 452

## Program 1 i PARABELANPASSNING

INMATNING upp till 8 objekt.									
000	76	LBL	060	02	2	120	03	3	180
001	15	E	061	07	7	121	04	4	181
002	01	I	062	06	6	122	65	x	182
003	01	I	063	95	=	123	43	RCL	183
004	42	STD	064	42	STD	124	08	08	184
005	07	07	065	08	08	125	42	STD	185
006	92	RTH	066	43	RCL	126	03	03	186
007	76	LBL	067	02	02	127	95	=	187
008	75	-	068	42	STD	128	42	STD	188
009	88	DMS	069	43	RCL	129	02	02	189
010	42	STD	070	01	01	130	43	RCL	190
011	01	I	071	95	=	131	04	04	191
012	29	C	072	77	GE	132	55	+	192
013	91	R/S	073	00	00	133	02	2	193
014	76	LBL	074	89	89	134	05	5	194
015	16	A'	075	03	3	135	03	3	195
016	88	DMS	076	06	6	136	75	-	196
017	42	STD	077	00	0	137	43	RCL	197
018	04	04	078	55	+	138	05	05	198
019	91	R/S	079	02	2	139	55	+	199
020	76	LBL	080	05	5	140	01	1	200
021	12	B	081	03	3	201	00	00	261
022	88	DMS	082	94	+-	202	03	3	262
023	42	STD	083	95	=	203	03	3	263
024	02	02	084	44	SUM	144	03	06	204
025	91	R/S	085	08	08	145	06	06	264
026	76	LBL	086	61	GTD	146	55	+	206
027	18	B'	087	01	01	147	02	2	207
028	88	DMS	088	09	09	148	07	7	208
029	42	STD	089	43	RCL	149	06	6	209
030	05	05	090	03	03	150	95	=	210
031	91	R/S	091	75	-	151	42	STD	211
032	76	LBL	092	43	RCL	152	00	00	212
033	13	C	093	02	02	153	32	XIT	213
034	88	DMS	094	95	=	154	53	c	214
035	42	STD	095	77	GE	155	43	RCL	215
036	03	03	096	01	01	156	06	06	216
037	91	R/S	097	09	09	157	75	-	217
038	76	LBL	098	03	3	158	43	RCL	218
039	18	C'	099	00	0	159	05	05	219
040	88	DMS	100	55	+	160	54	>	220
041	42	STD	101	02	2	161	55	+	221
042	06	06	102	03	3	162	01	1	222
043	43	RCL	103	95	=	163	02	2	223
044	01	I	104	44	SUM	164	75	-	224
045	55	+	105	08	08	165	03	3	225
046	02	2	106	03	3	166	04	4	226
047	05	5	107	00	0	167	65	x	227
048	03	3	108	85	+	168	32	XIT	228
049	75	-	109	53	c	169	95	=	229
050	43	RCL	110	43	RCL	170	42	STD	230
051	02	02	111	03	03	171	05	05	231
052	55	+	112	75	-	172	43	RCL	232
053	01	I	113	43	RCL	173	00	00	233
054	0								

# TI 59

## Program 2 TESTPROGRAMMET 1 PARABELANPASSNING

```

000 53 < 060 06 06 120 01 01 180 04 04 240 01 01 300 03 3 S 360 00 0 SPC
001 53 < 061 02 2 121 97 DSZ 181 69 DP 241 03 3 R 301 06 6 E 361 00 0 E
002 53 < 062 04 4 122 05 05 182 05 05 242 05 5 302 69 DP 362 01 1 N 362 01 1 N
003 43 RCL 063 42 STD 123 00 00 183 69 DP 243 01 1 9 303 02 02 305 06 6 K 363 07 7 D
004 04 04 064 00 00 124 61 61 184 00 00 244 02 2 304 02 2 3 R 364 03 3 N
005 65 x 065 00 0 125 98 ADV 185 03 3 P 245 05 5 305 06 6 3 R 365 01 1 N
006 01 1 066 42 STD 126 98 ADV 186 03 3 246 07 7 306 03 3 3 R 366 01 1 D
007 00 0 067 02 02 127 98 ADV 187 01 1 A 247 03 3 307 05 5 I 367 06 6 D
008 00 0 068 03 3 128 91 R/S 188 03 3 248 05 5 R 308 02 2 4 V 368 69 DP
009 54 ) 069 06 6 129 43 RCL 189 69 DP 249 00 0 1 309 04 4 4 V 369 03 03
010 42 STD 070 00 0 130 04 04 190 01 01 250 02 2 310 04 4 4 V 370 01 1 A
011 04 04 071 32 XIT 131 75 3 R 251 69 DP 311 02 2 371 03 3 S
012 59 INT 072 73 RC* 132 03 3 192 05 5 252 02 02 312 01 1 E 372 03 3 S
013 85 + 073 01 01 133 06 6 193 01 1 A 253 02 2 1 313 07 7 373 06 6 T
014 53 < 074 69 DP 134 00 0 194 03 3 254 02 0 314 69 DP 374 03 3 T
015 53 < 075 21 21 135 95 = 195 01 1 B 255 00 0 315 03 03 3 R 375 07 7
016 53 < 076 85 + 136 42 STD 196 04 0 256 03 3 316 03 3 R 376 00 0 SPC
017 43 RCL 077 73 RC* 137 04 04 197 01 1 E 257 05 5 317 05 5 R 377 00 0 SPC
018 04 04 078 01 01 138 61 GTO 198 07 7 258 00 0 5 318 00 0 0 SPC
019 22 INV 079 65 x 139 00 00 199 02 2 L 259 06 6 319 00 0 0 SPC
020 59 INT 080 43 RCL 140 99 99 200 07 2 260 01 1 8 320 02 2 2 G
021 65 x 081 02 02 141 76LBL 201 69 DP 261 01 1 8 321 02 2 2 H
022 05 5 082 69 DP 142 10 E* 202 02 02 262 69 DP 322 02 2 2 H
023 55 + 083 21 21 143 03 3 203 01 1 263 03 03 323 03 3 3 H
024 03 3 084 85 + 144 07 7 T 204 03 3 A 264 05 5 ( 324 01 1 A
025 85 + 085 73 RC* 145 69 DP 205 03 3 265 05 5 325 03 3 A
026 93 . 086 01 01 146 01 01 206 01 1 N 266 03 3 M 326 00 0 CLR
027 00 0 087 65 x 147 01 1 E 207 03 3 P 267 00 0 327 00 0
028 05 5 088 43 RCL 148 07 7 208 03 3 P 268 01 1 A 328 69 DP
029 54 > 089 02 02 149 03 3 209 01 1 A 269 03 3 329 04 04
030 65 x 090 33 X* 150 06 6 210 01 1 270 04 4 X 330 69 DP
031 01 1 091 95 = 151 03 3 211 03 3 271 04 4 X 331 05 05
032 00 0 092 22 INV 152 07 7 212 06 6 S 272 05 5 332 69 DP
033 54 > 093 88 DMS 153 03 3 213 69 DP 273 06 6 ) 333 00 00
034 59 INT 094 42 STD 154 03 3 214 03 3 274 69 DP 334 03 3 O
035 55 + 095 04 04 155 03 3 215 03 3 275 04 04 335 02 2
036 01 1 096 77 GE 156 05 5 S 216 06 6 276 69 DP 336 01 1 C 047 11 H
037 00 0 097 01 01 157 69 DP 217 03 3 277 05 05 337 05 5 142 10 E*
038 54 > 098 29 29 158 02 02 218 01 1 278 69 DP 338 02 2 H
039 54 > 099 71 SBR 159 03 3 219 02 2 279 00 00 339 03 3 H
040 55 + 100 00 00 160 02 2 220 04 4 280 02 2 340 00 0 SPC
041 01 1 101 00 00 161 02 2 221 03 3 N 281 01 1 F 341 00 0 SPC
042 00 0 102 99 FRT 162 02 2 222 01 1 282 04 4 Y 342 07 7
043 00 0 103 69 DP 163 03 3 223 02 2 G 283 05 5 343 06 5
044 54 > 104 22 22 164 05 5 224 02 2 284 02 2 344 69 DP
045 92 RTN 105 43 RCL 165 01 1 A 225 69 DP 285 07 7 345 01 01
046 76LBL 106 06 06 166 03 3 226 04 04 286 02 2 346 04 4
047 11 R 107 42 STD 167 03 3 227 69 DP 287 07 7 347 00 0
048 43 RCL 108 01 01 168 05 M 228 05 05 288 01 1 D 348 03 3 T
049 09 09 109 97 DSZ 169 69 DP 229 69 DP 289 06 6 349 07 7 R
050 65 x 110 00 00 170 03 03 230 00 00 290 69 DP 350 03 3
051 02 2 111 00 00 171 03 3 231 03 3 291 01 01 351 05 5
052 95 = 112 72 72 172 07 7 T 232 00 0 M 292 01 1 A 352 04 4
053 42 STD 113 38 ADV 173 02 2 I 233 01 1 E 293 03 3 353 05 5
054 05 05 114 03 3 174 04 4 234 07 7 294 05 5 , 354 01 1 C
055 01 1 115 44 SUM 175 02 2 L 235 01 1 D 295 07 7 355 05 5
056 01 1 116 06 06 176 07 7 236 06 6 296 04 4 U 356 69 DP
057 42 STD 117 43 RCL 177 02 2 L 237 00 0 SPC 297 01 1 T 357 02 02
058 01 01 118 06 06 178 07 7 238 00 0 298 03 3 T 358 02 2
059 42 STD 119 42 STD 179 69 DP 239 07 7 299 07 7 359 06 6 K

```

ENTALSSIFFROR  
0 1 2 3 4 5 6 7

TIOTALS-SIFFROR	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	-	8	9	A	B	C	D	E
2	M	N	D	P	Q	R	S	T
3	U	V	W	X	Y	Z	+	
4	x	*	r	a	e	*	x	
5	↑	%	↓	=	*			
6	z	?	?	?	?			
7	z	?	?	?	?			

## Program 3 i PARABELANPASSNING INMATNING OBSORT= LAT, LONG, TID i GMT, delta-TID

```

000 76LBL 060 54 > 120 00 00 180 42 STD 240 39 COS 300 61 GTO 360 19 D* 420 03 3
001 15 E 061 65 x 121 15 E 181 05 05 241 55 + 301 02 02 361 69 DP 421 07 7
002 01 1 062 01 1 122 02 2 182 43 RCL 242 43 RCL 302 14 14 362 02 02 422 19 D* 423 01 1
003 01' 1 063 00 0 123 09 9 183 01 01 243 06 06 303 02 2 363 69 DP 423 01 1
004 42 STD 064 54 > 124 05 5 184 38 SIN 244 39 COS 304 03 3 364 05 05 424 04 4
005 07 07 065 59 INT 125 42 STD 185 65 x 245 95 = 305 00 0 365 69 DP 425 06 6
006 92 RTN 066 55 + 126 59 59 186 43 RCL 246 22 INV 306 03 3 366 00 00 426 02 2
007 53 < 067 01 1 127 98 ADV 187 04 04 247 39 COS 307 32 XIT 367 92 RTN 427 69 DP
008 73 RC* 068 00 0 128 02 2 188 38 SIN 248 42 STD 308 61 GTO 368 76LBL 428 01 01
009 07 07 069 54 > 129 02 2 189 85 + 249 06 06 309 02 02 369 10 E* 429 02 2
010 69 DP 070 54 > 130 03 3 190 43 RCL 250 69 DP 310 14 14 370 98 ADV 430 07 7
011 27 27 071 55 + 131 00 0 191 01 01 251 00 00 311 02 2 371 69 DP 431 03 3
012 85 + 072 01 1 132 03 3 192 39 COS 252 01 1 312 03 3 372 00 00 432 02 2
013 73 RC* 073 00 0 133 07 7 193 65 x 253 04 4 313 00 0 373 02 2 433 03 3
014 07 07 074 00 0 134 32 XIT 194 43 RCL 254 69 DP 314 04 4 374 02 2 434 01 1
015 65 x 075 54 > 135 43 RCL 195 04 04 255 04 04 315 32 XIT 375 00 0 435 02 2
016 43 RCL 076 92 RTN 136 03 03 196 39 COS 256 01 1 316 61 GTO 376 01 1 436 02 2
017 03 03 077 53 < 137 22 INV 197 65 x 257 08 8 317 02 02 377 01 1 437 19 D*
018 69 DP 078 53 < 138 88 DMS 198 43 RCL 258 00 0 318 14 14 378 00 0 438 01 1
019 27 27 079 43 RCL 139 32 XIT 199 05 05 259 32 XIT 319 02 2 379 00 0 439 05 5
020 85 + 080 06 06 140 69 DP 200 39 COS 260 43 RCL 320 03 3 380 03 3 440 06 6
021 73 RC* 081 65 x 141 04 04 201 95 = 261 05 05 321 00 0 381 02 2 441 02 2
022 07 07 082 01 1 142 32 XIT 202 22 INV 262 77 GE 322 05 5 382 69 DP 442 69 DP
023 65 x 083 00 0 143 95 = 203 38 SIN 263 02 02 323 32 XIT 383 02 02 443 01 01
024 43 RCL 084 85 + 144 69 DP 204 42 STD 264 74 74 324 61 GTO 384 01 1 444 02 2
025 03 03 085 93 . 145 06 06 205 06 06 265 03 3 325 02 02 385 04 4 445 02 2
026 33 X* 086 05 5 146 98 ADV 206 22 INV 266 06 6 326 14 14 386 02 2 446 03 3
027 69 DP 087 54 > 147 71 SBR 207 88 DMS 267 00 0 327 02 2 387 05 5 447 00 0
028 27 27 088 59 INT 148 00 00 208 42 STD 268 75 - 328 03 3 388 04 4 448 03 3
029 54 > 089 55 + 149 07 07 209 08 08 269 43 RCL 329 00 0 389 00 0 449 07 7
030 92 RTN 090 01 1 150 85 + 210 69 DP 270 06 06 330 06 6 390 01 1 450 19 D*
031 53 < 091 00 0 151 43 RCL 211 00 00 271 95 = 331 32 XIT 391 07 7 451 01 1
032 53 < 092 54 > 152 02 02 212 83 GB* 272 42 STD 332 61 GTO 392 06 6 452 06 6
033 53 < 093 92 RTN 153 95 = 213 59 59 273 06 06 333 02 02 393 02 2 453 06 6
034 43 RCL 094 76LBL 154 42 STD 214 32 XIT 274 71 SBR 334 14 14 394 69 DP 454 02 2
035 08 08 095 11 R 155 05 05 215 69 DP 275 00 00 335 02 2 395 03 03 455 69 DP
036 65 x 096 88 DMS 156 71 SBR 216 04 04 276 77 77 336 03 3 396 02 2 456 01 01
037 01 1 097 42 STD 157 00 00 217 71 SBR 277 69 DP 337 00 0 397 04 4 457 07 7
038 00 0 098 01 1 158 07 07 218 00 00 278 06 06 338 07 7 398 03 3 458 05 5
039 00 0 099 91 R/S 159 42 STD 219 31 31 279 98 ADV 339 32 XIT 399 01 1 459 03 3
040 54 > 100 76LBL 160 04 04 220 69 DP 280 97 DSZ 340 61 GTO 400 02 2 460 07 7
041 42 STD 101 12 B 161 03 3 221 06 06 281 09 09 341 02 02 401 04 4 461 19 D*
042 08 08 102 88 DMS 162 06 6 222 08 8 282 01 01 342 14 14 402 03 3 462 98 RDV
043 59 INT 103 42 STD 163 00 0 223 44 SUM 283 47 47 343 02 2 403 07 7 463 98 RDV
044 85 + 104 02 02 164 32 XIT 284 43 RCL 344 03 3 404 69 DP 464 98 RDV
045 53 < 105 91 R/S 165 43 RCL 225 43 RCL 345 01 1 405 04 04 465 15 E
046 53 < 106 76LBL 166 05 05 226 04 04 286 42 STD 346 00 0 406 69 DP 466 25 CLR
047 53 < 107 13 C 167 77 GE 227 38 SIN 287 09 09 347 32 XIT 407 05 05 467 91 R/S
048 43 RCL 108 88 DMS 168 01 01 228 75 - 288 43 RCL 348 61 GTO 408 69 DP
049 08 08 109 42 STD 169 73 73 229 43 RCL 289 10 10 349 02 02 409 00 00 LABLAR
050 22 INV 110 03 03 170 61 GTO 230 01 01 290 44 SUM 350 14 14 410 01 1
051 59 INT 111 91 R/S 171 01 01 231 38 SIN 291 03 03 351 02 2 411 03 3
052 65 x 112 76LBL 172 82 82 232 65 x 292 61 GTO 352 03 3 412 06 6
053 05 5 113 14 D 173 43 RCL 233 43 RCL 293 01 01 353 01 1 413 02 2
054 55 + 114 88 DMS 174 05 05 234 06 06 294 21 21 354 01 1 414 69 DP
055 03 3 115 42 STD 175 75 - 235 38 SIN 295 02 2 355 32 XIT 415 01 01 415 01 01
056 85 + 116 10 10 176 03 3 236 95 = 296 03 3 356 61 GTO 416 02 2 113 14 D
057 93 . 117 43 RCL 177 06 6 237 55 + 297 00 0 357 02 02 417 07 7 360 19 D*
058 00 0 118 09 09 178 00 0 238 43 RCL 298 02 2 358 14 14 418 01 1 369 10 E*
059 05 5 119 42 STD 179 95 = 239 01 01 299 32 XIT 359 76LBL 419 03 3

```

7 ➤

De Bilt, 17-9-84

Redaktörens Adress:  
Anders Persson  
Klippervägen 41078  
222 45 LUND

Bäste Claes,

Det tog en tid, men här är ett bidrag från mig igen.

Låt oss starta med några tillägg till problem 45 av Utmaningen:

- 45a: Det kommer en tid när TI slutar tillverka termopapper! Och förutom det, är termopapper mycket dyrt!
- 45b: Ändring från labels till absolutadresser är alltid möjligt, men inte tvärtom, som Anders föreslår. (Hopp i mitten av kommandon!)
- 45c/d: Ändring av ett program under kontroll av sig självt är mycket viktigt, så OP 16 och OP 17 måste fungera!

Ett nytt problem till Utmaningen:

- 51: Beräkning av  $\ln x$  med många siffror.

Problem 4 och 29 var av samma slag, där frågades efter utökad precision av matematiska konstanterna: e och pi. Den bästa lösningen till 'e' kom från Björn Gustavsson - 1300 siffror och var införd i PB 80-4.

## n-te-gradspolynom

Exempel ur TI 59 Master Library-bok

$$2 - 3X + X^2, \quad X = +2 \rightarrow f(+2) = 0 \\ X = -1 \quad f(-1) = 6$$

Exempel ur SR-52 Mathematics Applications....

$$1 + X + 2X^2 + 3X^3 + 4X^4, \quad X = +2 \rightarrow f(+2) = 99 \\ X = -2 \quad f(-2) = 47$$

$$1 + 7X + 2X^2 + 3X^3 + 4X^4, \quad X = +2 \rightarrow f(+2) = 111$$

9 ➤

### exempel version 2

X DCH RND VIR R/S	VIR R/S	X DCH RND VIR R/S	VIR R/S
X=		X=	
R(N)=		R(N)=	
-3.	X	-2.	X
2.	R(N)	4.	R(N)
-3.	R(N)	3.	R(N)
5.	R(N)	2.	R(N)
-8.	R(N)	1.	R(N)
-104.	F(X)	1.	R(N)
		99.	F(X)
X=		X=	
X DCH R(N) VIR R/S	VIR R/S	X DCH R(N) VIR R/S	VIR R/S
X=		X=	
R(N)=		R(N)=	
2.	X	-2.	X
1.	R(N)	4.	R(N)
-3.	R(N)	3.	R(N)
2.	R(N)	2.	R(N)
0.	F(X)	1.	R(N)
X=		X=	
X DCH R(N) VIR R/S	VIR R/S	X DCH R(N) VIR R/S	VIR R/S
X=		X=	
R(N)=		R(N)=	
-1.	X	4.	R(N)
1.	R(N)	3.	R(N)
-3.	R(N)	2.	R(N)
2.	R(N)	7.	R(N)
6.	F(X)	1.	R(N)
X=		X=	
X DCH R(N) VIR R/S	VIR R/S	X DCH R(N) VIR R/S	VIR R/S
X=		X=	
R(N)=		R(N)=	
111.	F(X)	111.	F(X)

Lösningen av pi var skriven av Jovan Puzovic' - 1188 siffror i PB 83-2 och Renand de la Faille - 1287 siffror i PB 83-4.

Båda programmen använde inverterade serier för att komma till dessa värden.

Nyligen har jag funnit dessa serier för att beräkna  $\ln(x+1) \approx 0$

$$\ln(x+1) = \ln x + 2 \left[ \frac{1}{2x+1} + \frac{1}{3} \left( \frac{1}{2x+1} \right)^2 + \frac{1}{5} \left( \frac{1}{2x+1} \right)^4 + \dots \right]$$

Tag  $x=1$  och ni finner  $\ln 2$  etc.

Om ni jämför med detta:

$$\arctan x = \frac{1}{x} - \frac{1}{3} \left( \frac{1}{x} \right)^3 + \frac{1}{5} \left( \frac{1}{x} \right)^5 - \dots$$

Ni kan se att de är mycket lika. Som resultat av det försökte jag anpassa Jovan Puzovic's program i PB 83-2, men när det kört färdigt fann jag att några minnen innehöll 13 siffror, istället för väntade 12.

Kan någon lösa detta mysterium? och finns någon som vet en bok som innehåller värdet av  $\ln x$  med mer än 13 siffror?

Nästa ämne:

Finn det någon medlem som kan översätta HP-41 programmet "HP-41 calculates "e" to 2900 D" ur HP PPC Calculator Journal V9 N1 P12/13 till Basic och/eller flödesdiagram. Varje gång jag försöker slutar det med kaos.

Behöver någon en kopja av artikeln skriv till mig!

Robert A.H. Prins  
A. Nobellalaan 112  
NL-3731 DX De Bilt  
Holland.

## PARABELANPASSNING

1-8 OBJ.E: INIT      UTSKRIFT program 3 i  
A: LAT                    PARABELANPASSNING  
B: LONG  
C: GMT  
D: ^T

Ur Den Svenska Almanackan för maj 30 1984:  
solförmörkelse kl 18.56 SNT = 17:56 GMT  
MAXIMUM.

17.	GMT	17.4	GMT	17.3	GMT
16.552	H1	12.21	H1	13.283	H1
285.1	B	299.	B	291.	B
16.389	H2	11.503	H2	13.01	H2
285.	B	293.1	B	291.1	B
-28.588	H3	-30.187	H3		
339.2	B	350.6	B		
13.253	H4	8.429	H4	17.4	GMT
288.1	B	296.3	B		
-0.279	H5	3.814	H5	12.21	H1
124.4	B	133.1	B	293.	B
-34.572	H6	-29.553	H6	11.503	H2
75.4	B	84.6	B	293.1	B
4.236	H7	8.227	H7		
124.3	B	133.3	B	17.496	GMT
33.596	H8	38.389	H8		
110.2	B	120.4	B		
				11.148	H1
				295.	B
				10.406	H2
				295.2	B
17.196	GMT	18.	GMT		
				18.	GMT
14.364	H1	10.096	H1		
289.	B	296.9	B		
14.128	H2	9.32	H2		
289.	B	297.2	B		
-29.461	H3	-30.36	H3	10.096	H1
344.9	B	356.4	B	296.9	B
11.02	H4	6.287	H4	9.32	H2
292.2	B	300.5	B	297.2	B
1.353	H5	5.193	H5		
128.7	B	137.5	B	18.1	GMT
-32.273	H6	-27.222	H6		
80.1	B	89.	B		
6.269	H7	10.1	H7		
128.7	B	137.9	B		
36.218	H8	40.491	H8		
115.2	B	126.	B		
				18.1	GMT
				H1= MÅNE	
				H2= SOL	
				H3= ARIES	
				H4= VNUS	
				H5= MARS	
				H6= JUPI	
				H7= SATU	
				H8= STAR	

# Favorit i repris

## NUMMERSÄNDARE mod.manuell

Brukar du fastna med fingret i nummerskivan och tycker att knapptelefonen är för dyr så är detta lösningen på ditt problem. Denna metod är enkel och kräver varken dator, lödkolv eller störande högtalare. Om du någon gång tittat inuti en telefon, har du kanske lagt märke till den anordning som frambringar nummersignalen. När nummerskivan återgår till sitt viloläge alstrar den ett signaltåg som ett telefonnummer, och kopplar samtalet. Nedanstående tabell visar antalet pulser hos signaltåget vid olika siffror:

Sifrapå tel.

Signaltåg

0	X
1	XX
2	XXX
3	XXXX
4	XXXXX
5	XXXXXX
6	XXXXXX
7	XXXXXXX
8	XXXXXXXX
9	XXXXXXXX

Tillvägagångsätt:

Du kan simulera ett siffernedslag genom att i snabb takt ( ung. 2 nedslag/sek ) göra ( SIFFRAN=1 ) nedslag på din telefon-klyka. Genom att upprepa denna procedur kan du alltså ringa ett telefonnummer utan att röra fingerskivan.

Övn.ex:

Telefonnummer: 9051...

Nedslagssekvens:

9= XXXXXXXXX  
0= X  
5= XXXXX  
1= XX

Eftersom ditt finger svälligen kan anses vara en apparat med elektrisk, mekanisk eller akustisk förbindelse "torde" denna metod vara tillåten. Möjligens kunde detta ev. "fingerbrott" rubriceras som skedegörelse. Troligen är det dock fråga om normal försitning vilket inte medför straffansvar.

Eder trogne lagvrängare

Bengt Fahlgren

## TA det Piano

```

10 REM MUSIKPROGRAM DÄR DU ANVÄNDER DATORN
   SOM ETT PIANO
20 REM ALLA NORMAL TONER KAN SPELAS MED TA
   NGENTERNA
100 CALL KEY(0,K,S):: IF S=0 THEN 100
105 REM KOLLAR ATT INTE ÖNSKADE TANGENTER
   TRYCKS NER, TAR ÄVEN HAND OM UDDA ASCII
   -TAL
110 IF K=45 THEN 100
120 IF K<44 THEN 100 ELSE IF K>46 AND K<58
   THEN 100 ELSE IF K>60 AND K<65 THEN 100
130 IF K>90 AND K<96 THEN 100 ELSE IF K>96
   THEN 200 ! CHECKAR OM STORA ELLER SMA B
   OKSTAVER TRYCKS NER
140 IF K=44 THEN CALL SOUND(-100,1568,0):::
   GOTO 100
150 IF K=46 THEN CALL SOUND(-100,1760,0):::
   GOTO 100
160 IF K=59 THEN CALL SOUND(-100,698,0)::: G
   OTO 100
170 IF K=58 THEN CALL SOUND(-100,1661,0):::
   GOTO 100
180 IF K=60 THEN CALL SOUND(-120,-8,0)::: GO
   TO 100
185 REM STORA BOKSTAVER ASCII=65->90
190 ON K-64 GOTO 210,220,230,240,250,260,27
   0,280,290,300,310,320,330,340,350,360,3
   70,380,390,400,410,420,430+440,450,460
195 REM SMA BOKSTAVER ASCII =97->122
200 ON K-96 GOTO 470,480,490,500,510,520,53
   0,540,550,560+570,580,590,600,610,620,6
   30,640,650,660,670,680,690,700,710,720
205 REM STORA BOKSTAVER=HALVTONER+NOISE
210 CALL SOUND(-120,466,0)::: GOTO 100
220 CALL SOUND(-120,-6,0)::: GOTO 100
230 CALL SOUND(-120,-1,0)::: GOTO 100
240 CALL SOUND(-120,622,0)::: GOTO 100
250 CALL SOUND(-120+156,0)::: GOTO 100
260 CALL SOUND(-120,740,0)::: GOTO 100
270 CALL SOUND(-120+831,0)::: GOTO 100
280 CALL SOUND(-120,932,0)::: GOTO 100
290 CALL SOUND(-120+311,0)::: GOTO 100
300 CALL SOUND(-120,1109,0)::: GOTO 100
310 CALL SOUND(-120,1245,0)::: GOTO 100
320 CALL SOUND(-120+1480,0)::: GOTO 100
330 CALL SOUND(-120,-4,0)::: GOTO 100
340 CALL SOUND(-120,-5,0)::: GOTO 100
350 CALL SOUND(-120,370,0)::: GOTO 100
360 CALL SOUND(-120,415,0)::: GOTO 100
370 CALL SOUND(-120,117,0)::: GOTO 100
380 CALL SOUND(-120,185,0)::: GOTO 100
390 CALL SOUND(-120,554,0)::: GOTO 100
400 CALL SOUND(-120,208,0)::: GOTO 100
410 CALL SOUND(-120,277,0)::: GOTO 100
420 CALL SOUND(-120,-7,0)::: GOTO 100
430 CALL SOUND(-120,139,0)::: GOTO 100
440 CALL SOUND(-120,-2,0)::: GOTO 100
450 CALL SOUND(-120,233,0)::: GOTO 100
460 CALL SOUND(-120,-3,0)::: GOTO 100
465 REM SMA BOKSTAVER HELTONER
470 CALL SOUND(-100,294,0)::: GOTO 100
480 CALL SOUND(-100,1175,0)::: GOTO 100
490 CALL SOUND(-100,988,0)::: GOTO 100
500 CALL SOUND(-100,349,0)::: GOTO 100
510 CALL SOUND(-100,131,0)::: GOTO 100
520 CALL SOUND(-100,392,0)::: GOTO 100
530 CALL SOUND(-100,440,0)::: GOTO 100
540 CALL SOUND(-100,494,0)::: GOTO 100
550 CALL SOUND(-100,220,0)::: GOTO 100
560 CALL SOUND(-100,523,0)::: GOTO 100
570 CALL SOUND(-100,587,0)::: GOTO 100
580 CALL SOUND(-100,659,0)::: GOTO 100
590 CALL SOUND(-100,1397,0)::: GOTO 100
600 CALL SOUND(-100,1319,0)::: GOTO 100
610 CALL SOUND(-100,247,0)::: GOTO 100
620 CALL SOUND(-100,262,0)::: GOTO 100
630 CALL SOUND(-100,110,0)::: GOTO 100
640 CALL SOUND(-100,147,0)::: GOTO 100
650 CALL SOUND(-100,330,0)::: GOTO 100
660 CALL SOUND(-100,165,0)::: GOTO 100
670 CALL SOUND(-100,196,0)::: GOTO 100
680 CALL SOUND(-100,1047,0)::: GOTO 100
690 CALL SOUND(-100,123,0)::: GOTO 100
700 CALL SOUND(-100,880,0)::: GOTO 100
710 CALL SOUND(-100,175,0)::: GOTO 100
720 CALL SOUND(-100,784,0)::: GOTO 100

```

# Tips och tricks

Här börjar en artikel med tips och tricks, vilka du kan utföra med din dator. Jag hoppas att denna artikel kommer att fortsätta så länge tidningen Programbiten finns, och den kommer att finnas så länge det finns en TI99/4A.

Vi börjar med hur du kopieringskyddar dina flexskivor med hjälp av Disk-Manager Modulen. Man undrar varför inte detta enkla knep står i Manualen, Disk Memory System? Man önskar att liknande saker borde stå i respektive manual.

Alltså så här gör du:

1. Sätt in din Disk Manager Modul
2. Sätt på alla nödvändiga enheter
3. Gå till titelskärmen och tryck siffran 2 för Disk Manager
3. Välj vilken meny du vill
4. Tryck ned FCTN och samtidigt bokstaven X
5. Tryck ner X tio gånger. Tionde gången hör du en ljudsignal och tecknen > < kommer att visa sig ovanför menylistan högst upp på skärmen.

Så här ser det ut:

> <

- DISK COMMANDS
1. CATALOG DISK
  2. BACKUP DISK
  3. MODIFY DISK NAME
  4. INITIALIZE NEW DISK

YOUR CHOICE? 1

När du nu formatterar din nya flexskiva, kommer kopieringskyddet att skrivas på sektor 0. Skyddet skrivas in när programmet startar -(val 4 på menylistan= INITIALIZE NEW DISK )- initieringsrutinen. Försöker du sedan att kopiera program med hjälp av Disk-Manager Modulen, givetvis med program på den skyddade flexskivan, kommer du att se och höra något otrevligt. En WARNING! gör alltid en kopia av dina program på en oskyddat -(unprotected)- flexskiva, eftersom skyddet ej går att ta bort. För att ta bort skyddet, måste du omformattera flexskivan. Om du inte vill skydda din flexskiva från obehörig kopiering tryck bara en gång till på FCTN X och tecknen försvinner från sin plats på skärmen eller tryck FCTN += (QUIT) så kommer du istället tillbaks till titelskärmen. Enkelt, eller hur? I kombination med kopierings och listskyddet som finns i Extended Basic ( SAVE DSK1.XXXX,PROTECTED ) bör dina program vara ordentligt skyddade från obehörig kopiering.

Här kommer några adresser vilka används av Extended Basic och expansionminnet. de första gör samma sak som att skriva BYE eller att trycka FCTN += (QUIT). Det är 3 olika adresser beroende på vilken datorversion du har, prova en i taget.

1. CALL LOAD(-31885,255)
2. CALL INIT :: CALL PEEK(2,A,B)::  
CALL LOAD(-31804,A,B)
3. CALL LOAD(-31804,0,36)

Prova endast ett av alternativen. Var inte rädd att leka med maskinkoder inte mycket kan hänta, för det mesta läser sig datorn eller du får underliga resultat. Datorn blir "knäpp".

Här kommer en annan adress. Den "läser" FCTN += (QUIT), d.v.s. den sluter tillfälligt att fungera.

CALL INIT :: CALL LOAD(-31806,16)

Vill du kunna använda den igen skriv bara

CALL LOAD(-31806,0)

Om du har en talsyntes:

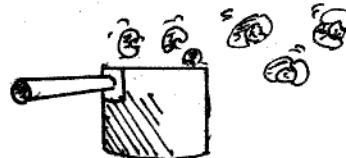
Speech Synthesizer och Terminal Emulator II. Så kan du få roligt om du gör så här.

1. Ladda in ett program i Basic.
2. Mata in LIST "SPEECH".
3. Lyssna noga.

Du har en vän om du känner dig ensam någon gång. Jag vet att man kan skriva och läsa adresser direkt till talsyntesen. Jag har själv provat det, men glömt adresserna. Vi kommer att ta upp det i nästa nummer. Har ni kära läsare och medlemmar fler tips och tricks. Skriv till oss under titel "TIPS OCH TRICKS". Det var allt för den här gången.

Vi möts i nästa nummer!

Göran Nygren



Hur man poppar subrutin stacken.

Av Tony Rogvall.

Bakgrund till den här programlistningen har en liten historia. När jag började arbeta med basic på 99:an upptäckte jag att det fanns en del kommandon. Bl.a kommandon som kunde peeka och poka i minnet, men också kommandon för stack hanteringen, när man använder GOSUB - RETURN. För jag tyckte att det skulle vara bra att vid t.ex ett error i en subrutin av 2:a, 3:e ordningen direkt kunna komma tillbaka till en huvudmeny, styrförprogram eller dylikt.

Efter jag hade skaffat mig minnesexpansion började jag leta efter den pekare som hade hand om stacken, men kunde inte hitta den. Efter mycket slit och mödosamt provande kom jag fram till vilken adress det var och hur den fungerade. Adressen kallas VS PTR ( VDP stack pointer ) och används åtta bytes per stackat element. VS PTR hittas på adress >836E eller -31890 och måste minskas med först åtta sedan antalet subrutiner som man vill plocka ur stacken multiplicerat med åtta. Jag hoppas att adressen är den samma på alla maskiner. Skicka gärna in program som använder den adress till tidningen.

```
100 PRINT "START"
110 GOSUB 150
120 PRINT "STOP"
130 END
140 REM *** SUBRUTIN 1 ***
150 GOSUB 190
160 PRINT "1:A"
170 RETURN
180 REM *** SUBRUTIN 2 ***
190 GOSUB 230
200 PRINT "2:A"
210 RETURN
220 REM *** SUBRUTIN 3 ***
230 GOSUB 270
240 PRINT "3:E"
250 RETURN
260 REM *** POP RUTIN ***
270 CALL PEEK(-31890,A,B)
280 CALL LOAD(-31890,A,B-8-16)
290 RETURN
```

ETT STORT PS.

Till nästa nummer hade jag tänkt att försöka publicera en minneskarta över systemvariabler för extended basic. Men för att få en så fulständig minneskarta som möjligt så skulle jag vilja ha lite hjälp. Skicka ett brev och berätta vilka adresser ni hittat och vad de gör. Skriv "adressjakten" utan på brevet så vet Claes "skägget" Schibler vem som skall ha brevet. Trevlig jakt.

# Polsk Notation

DEN ITALIENSKE PROGRAMMERARENS HELAFTON.  
eller Polsknotation i BASIC.

Av Tony Rogvall.

Det här programmet är till för er som vill lära er att handskas med polsknotation. När jag skrev det så försökte jag så långt som möjligt att härlma FORTHens syntax. Programmet jobbar endast med tal, det här är inte FORTH skriven i BASIC. Jag har dock lagt in kommandon för stackhantering.

Programmet nedan illustrerar hur man inte skall skriva ett program, i regel. Men i ett försök att få upp hastigheten i programmet så flyttade jag om subrutinerna lite. Rem satser o.dyl är borttagna (aldrig ditsatta). Pre-scan kopplas bort för att få snabbare uppstart.

Program beskrivning:

Stack-hanterings-kommandon.

DROP renser toppen på stacken.

OVER kopierar det nästöversta elementet på stacken.

DUP dubblerar toppen på stacken.

ROT roterar de tre översta elementen.

SWAP byter plats på de två översta elementen.

Utskrifts-kommandon

.S skriver ut innehållet i stacken utan att förstöra den.

.V skriver ut variabler som finns definierade.

. (punkt) skriver det översta elementet och poppar stacken.

Radering.

DEL<v> raderar en variabel.

EMPTY ALL, S eller V. ALL tömmer både variabler och stack. S tömmer stacken. V tömmer variabler.

= variabel tilldelning.

Räknesätt.

Multiplikation, division, addition, subtraktion, potenser. (\*, /, +, -, 0).

Beräkningar i polsknotation är inte svårt, men lite omständigt om man är ovan. T. ex om man vill addera 6 och 4 skriver man 6 <sp> 4 <sp> + (<sp> står för mellanslag). Svaret hittar man på stacken, genom att skriva . <enter> så skrivas svaret ut. Det går självfallet att skriva i en följd: 6 <sp> 4 <sp> + <sp> . <enter>.

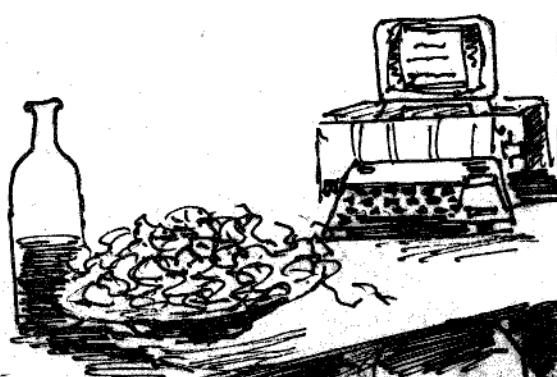
Om jag skulle vilja göra lite svårare beräkningar t.ex (6+4)\*9 kan jag göra på två sätt  
1) 6 <sp> 4 <sp> + <sp> 9 <sp> \* <sp> . <enter>  
2) 9 <sp> 6 <sp> 4 <sp> + <sp> \* <sp> . <enter>

Man kan skapa variabler genom att skriva <värdet> <sp> = <sp> <variabelnamn> <enter>. Du kan inspektera variabel stacken genom att skriva <V>. Man använder variabeln genom att refera till den, exempel:

1) A <sp> = <sp> 10

2) A <sp> 10 <sp> + <sp> . <enter>.

Lek lite med det här programmet sedan kan ni köpa föreningens FORTH. FORTH går FORT.



```

10 DIM S(50),C$(21),VA$(30),VA(30)
20 GOTO 30 :: CALL SCREEN :: CALL COLOR :: CALL SOUND
   :: CALL CLEAR :: I$,D$,L,S
   P,VP,T,E,V,I,J,R$
30 DATA +,-,*/,0,SWAP,DROP,OVER,ROT,DUP,=,.,S.,V,DEL
   ,EMPTY
40 GOTO 460 :: !EP-
50 ON ERROR 60 :: D$=SEG$(I$,1,POS(I$," ",1)-1):: L=LE
   N(D$):: ON ERROR STOP :: RETURN
60 PRINT D$&"? ";:: ON ERROR STOP :: RETURN 70
70 RETURN
80 I$=SEG$(I$,POS(I$," ",L)+1,LEN(I$)):: RETURN
90 S(SP)=V :: SP=SP+1 :: RETURN
100 SP=SP-1 :: V=S(SP):: RETURN
110 GOSUB 100 :: E=V :: GOSUB 100 :: RETURN
120 IF SP<2 THEN RETURN ELSE T=S(SP-1):: S(SP-1)=S(SP-2
   ):: S(SP-2)=T :: RETURN
130 IF SP<1 THEN RETURN ELSE GOSUB 100 :: RETURN
140 IF SP=0 THEN 170 ELSE GOSUB 100 :: PRINT STR$(V)&" "
   :: RETURN
150 FOR I=0 TO SP-1 :: PRINT STR$(S(I))&" ";:: NEXT I :
   : IF I=0 THEN 170 ELSE RETURN
160 FOR I=1 TO VP :: PRINT VA$(I)&" ";:: NEXT I :: IF V
   P=0 THEN 170 ELSE RETURN
170 PRINT "empty" :: RETURN
180 IF SP<2 THEN RETURN ELSE V=S(SP-2):: GOSUB 90 :: RE
   TURN
190 IF SP<3 THEN RETURN ELSE T=S(SP-3):: S(SP-3)=S(SP-2
   ):: S(SP-2)=S(SP-1):: S(SP-1)=T :: RETURN
200 IF SP<1 THEN RETURN ELSE V=S(SP-1):: GOSUB 90 :: RE
   TURN
210 V=V+E :: RETURN
220 V=V-E :: RETURN
230 V=V-E :: RETURN
240 V=V/E :: RETURN
250 V=V\0E :: RETURN
260 GOSUB 50 :: FOR I=1 TO 19 :: IF SP=0 OR C$(I)=D$ TH
   EN PRINT "invalid variable ";::RETURN
270 NEXT I :: T=VP :: FOR J=0 TO VP :: IF D$=VA$(J)THEN
   VP=J :: GOTO 290
280 NEXT J :: T=T+1
290 VP=VP+1 :: VA$(VP)=D$ :: GOSUB 100 :: VA(VP)=V :: G
   OSUB 80 :: VP=T :: RETURN
300 GOSUB 50 :: FOR I=0 TO VP :: IF VA$(I)=D$ THEN 320
310 NEXT I :: RETURN
320 GOSUB 80 :: FOR J=I TO VP :: VA$(J)=VA$(J+1):: VA(J
   )=VA(J+1):: NEXT J :: VP=VP-1:: RETURN
330 GOSUB 50 :: IF D$="ALL" THEN SP,VP=0 ELSE IF D$="S"
   THEN SP=0 ELSE IF D$="V" THEN VP=0 ELSE PRINT D$;"? ";
340 GOSUB 80 :: RETURN
350 GOSUB 50 :: IF D$="-" OR D$="+" THEN 390
360 IF D$="" THEN L=1 :: GOSUB 80 :: GOTO 350
370 ON ERROR 440
380 V=VAL(D$):: GOSUB 90 :: L=1 :: T=0 :: GOTO 400
390 R$=D$ :: T=1
400 ON ERROR STOP :: GOSUB 80
410 IF T THEN RETURN
420 IF I$<>"" THEN 350
430 R$="" :: RETURN
440 RETURN 390
450 PRINT "ok" :: CALL SOUND(75,220,5,221,5,222,5):: IN
   PUT ""::I$ :: IF I$="" THEN 450 ELSE I$=I$&" " :: RETURN
460 CALL SCREEN(2):: CALL CLEAR :: FOR I=1 TO 16 :: REA
   D C$(I):: NEXT I :: FOR I=0 TO 12 :: CALL COLOR(I,16,2)
   :: NEXT I :: SP,VP=0
470 GOSUB 450
480 GOSUB 350 :: IF R$="" THEN 470
490 FOR I=0 TO VP :: IF VA$(I)=R$ THEN V=VA(I):: GOSUB
   90 :: GOTO 560
500 NEXT I
510 FOR I=1 TO 16 :: IF R$=C$(I)THEN GOTO 530
520 NEXT I :: PRINT R$&"? ";:: GOTO 470
530 IF I>5 THEN I=I-5 :: GOTO 550
540 IF SP>1 THEN GOSUB 110 :: ON I GOSUB 220,230,210,24
   0,250 :: GOSUB 90 :: GOTO
   560 ELSE 560
550 ON I GOSUB 120,130,180,190,200,260,140,150,160,300,
   330
560 IF I$<>"" THEN 480 ELSE 470

```

# ASSEMBLER-SKOLAN DEL II

av Tony Rogvall & Michael Öhman

För att göra olika saker i assembler måste man kunna spara data på olika sätt. Dels TEXT-, DATA-, och Buffert lagringar.

TEXT,

Exempel.

Först definierar vi en sträng:

D1 TEXT 'KALLA MIG KALLE.'

"D1" är vad man i assembler sammanhang kallas label eller etikett och skulle kunna jämnställas med en variabel. I assembler är alltid fältet längst till höger reserverat för en etikett. En etikett refererar alltid till en adress där data, program eller text finns.

"TEXT" hamnar i OP-code fältet (instruktions fält) men är inte en instruktion till processorn utan en instruktion till assemblatorn. Den talar om att det efterföljande fältet är en text sträng. Text strängen markeras med apostrof och inte som i basic med sitiations tecken.

För detta program krävs en referens till en rutin som skriver i VDP-ram. Rutinen heter VSBW (eng. vdp single byte write) och skriver en byte till till VDP. Rutinen är beroende av vad som finns i R0 och R1. I R0 laddar man skärmadressen och i MSBY R1 laddas tecknet. I assembler kan man direkt referera till den genom att skriva:

REF VSBW. Detta medför att namnet VSBW tilldelas en adress när programmet laddas och körs.

I line-by-line-assembler måste en etiket användas. En etiket tilldelas ett värde genom att sätt en equate:

VS EQU >6024.

För att inte krångla till det allt för mycket låter jag programmet dyka (göra en evighets loop).

Exempel 1 skriver ut strängen på skärmen med hjälp av indirekt adressering med automatiskt ökande.

Definitions block för:

EDITOR/ASSEMBLER.

```
I     DEF ST          I
I     REF VSBW        I
```

MINI-MEMORY.

```
I VS   EQU >6024      I
```

```
D1   TEXT 'KALLA MIG KALLE.'
ST   LI  R0,488    * Ladda R0 med skärm adress.
      LI  R3,16     * R3=tecknen räknare.
      LI  R2,D1     * R2 innehåller adress till
sträng.
L1   MOVB *R2+,R1  * Flyttar från sträng till
register 1.
      BLWP EVS     * (VSBW) är för E/A
      INC R0       * Öka skärm adress.
      DEC R3       * Minska tecknen räknare.
      JNE L1       * Är R3=0 ? Om inte hoppa till L1
L2   JMP L2       * EVIGHETS LOOP
```

Exempel 2 använder samma definitioner som i exempel 1, men skall exemplifiera användningen av indexerad adressering.

```
ST   LI  R0,488    * Skärm adress laddas i R0.
      CLR R3      * R3=index.
L1   MOVB ED1(R3),R1 * Flyttar över från strängen
till R1.
      BLWP EVSBW  * Skriver tecknen på skärmen.
      INC R0       * Öka skärm adress.
      INC R3       * Öka index
      CI  R3,16     * Är hela strängen på skärmen ?
      JNE L1       * Nej, fortsätt att skriv.
L2   JMP L2       * Ja, DYK!
```

För att försäkra mig om att ni får in programmet i datorn, skall jag steg för steg förklara hur det går till praktiskt.

För EDITOR/ASSEMBLER.

1. Ladda in EDITOR.
2. Skriv in definitionsblocket och programmet.
3. Skriv "END" som sista rad, i instruktions-fältet.
4. Gå tillbaka till menyn och spara programmet.
5. ladda in assemblatorn och döp objekt filen.
6. Som option matar du in ett "R", som står för att du vill använda prefikset "R" för att beteckna register.
7. Välj "LOAD AND RUN". Starta programmet med ST.

För MINI MEMORY.

1. Ladda in line-by-line assembler (om den inte redan finns).
2. Välj MINI MEMORY som val från huvud menyn.
3. Skriv in definitions blocket för mini memory.
4. Mata in programmet.
5. När programmet är klart så måste du justera två pekare: LFAM (eng. last free address in memory) och FFAM (eng. first free address in memory). Genom att skriva AORG >701C (AORG skall stå i instruktions fältet) så kommer assemblatorn att hoppa till LFAM. Där skriver du in adressen efter den sista adressen som du använder för programmet. Skriv in "DATA >adress". När du tryckt enter så ligger du på adress >701E (LFAM) där skriver du "DATA >7FE0" vilket innebär den sista användbara adressen i minnet.
6. Skriv "AORG >7FE0" och "TEXT 'ST'" (ST med fyra mellanslag) vilket är program namnet, sedan enter. Därefter "DATA >7D10" som är en pekare mot programmets start adress.
7. Skriv "END" och enter.
8. Välj "RUN" på menyn och skriv "ST".

Använd denna instruktionslista för kommande bruk.

DATA,

I det här fallet sparas värdet av talet istället för ascii kodén som ovan.

D1 DATA >1234,>5678

>12 kommer att lagras på adressen D1 pekar på. >13 på följande byte osv.  
Även i det här fallet är DATA instruktionen bara gjord för assemblatorn.

till sist så har vi:

BSS, Block starting with symbol.

D1 BSS >20

denna instruktion sparar 32 bytes med början vid D1. När assemblatorn assemblar motsvarar en address D1. Om assemblatorns address ligger på >A000 när instruktionen utförs får D1 värdet >A000, nästa instruktion kommer att hamna 32 bytes framåt dvs. >A020.

## ARBETSREGISTER OCH PEKARE

Den absolut första instruktionen som jag tycker att man måste sätta sig in i hur den fungerar, för att förstå assemblerprogrammering på 99:an, är LWPI.

LWPI (eng. load workspace pointer immediate) laddar arbetsregisterpekaren (WP) med en godtycklig adress (inom adress området >0000->FFFF). WP pekar, som nämnts i förra nummret, ut en area i minnet där mjukvaruregisterna hamnar (se bild 1).

Det finns 16 mjukvaruregister som kan användas åt gången, de numreras från 0 till 15. I ett assemblator program kan man refera till dem med ett "R" och sedan register nummret. De flesta instruktioner använder sig av arbetsregisterna.

### ADRESSERINGS SÄTT

Det finns fem sätt att nå minnet eller registerna. Deras syntax visas nedan. "n" är ett tal mellan 0 och 15, vilket bestämmer register numret.

1. Direkt register  
Rn
2. Indirekt register  
\*Rn
3. Indirekt register med automatisk ökning.  
\*Rn+
4. Direkt minne  
Eminne
5. Indexerat minne  
Eminne(Rn) OBS n<0

De flesta instruktionerna fodrar två register eller minnesadresser, t.ex vid addering, subtrahering, flyttning eller multiplicering. Sådana instruktioner säger man ha en källa och en destination. Källan är det register eller den minnesadress som står närmast och viceversa.

Exempel: syntax  
( instruktion källa,destination )  
MOV RO , E>A000

Andra instruktioner har bara källa.

Exempel:  
CLR RO

Den här instruktionen heter clear och nollställer alla bitar i ett 16-bitars ord. Den kan även använda de andra adresserings sätteten.

LWPI >A000  
LWPI är en instruktion som kräver ett direkt värde, man kan alltså inte använda någon utav de fem adresserings sätten på den här instruktionen. Instruktionen laddar WP med värdet till höger.

Att flytta om kring i minnet.

MOV och MOVB är två instruktioner som används för att kopiera minne/register innehåll. MOVB (eng. MOV BYTE) flyttar en byte från källan till destinationen medan MOV flyttar ett helt 16 bitars ord.

Här är några exempel på hur man använder de fem adresserings sätteten.

MOV RO,R1  
kopierar innehållet i RO till R1.

MOV \*RO,R1  
kopierar innehållet på adress som finns i RO till R1.

MOV \*RO+,R1  
kopierar innehållet på adress som finns i RO till R1, ökar sedan adressen i RO med två.

MOVB \*RO+,R1  
byte instruktionen gör likadant som ovan men ökar bara RO med ett.

MOV E>A000,R1  
kopierar innehållet på adress A000 till R1.

MOV E>A000(R1),RO  
kopierar innehållet på den adress som fås genom att A000 adderas med innehållet i R1 till RO.

MOVB \*RO+,\*R1+  
kopierar en byte från adress som finns i RO till adress som finns i R1. Båda registerna ökas med ett efter kopiering. Om denna instruktion brukas i en loop kan den vara mycket användbar för att kopiera stora mängder text mellan olika minnes areor.

BILD 1

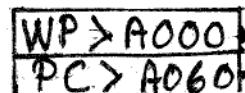
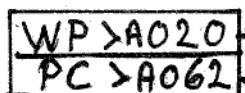


BILD 2



REGISTER AREA

MSB, RO	LSB, RO
MSB, R1	LSB, R1
:	:
MSB, R15	LSB, R15
MSB, RO	LSB, RO
MSB, R1	LSB, R1
:	:
MSB, RM	LSB, R15
>AO	>60
LWPI	→
>AO	20
:	:
020.	020.

PROGRAM AREA.

S E I K O S H A

GP-100 A/II PRINTER

STORA och små bokstäver även dubbelbred text Å Ä Ö och grafik.  
Med interface för RS-232  
OBETYDLIGT använd 2.400:-

TI JOYSTICKS HELT NYA 125:-

08/ 765 47 78

# DATAkommunikation över TELENätet med MODEM

av Joakim Söderberg och Michael Öhman

## TECKENSET:

Datorer kan kommunicera med varandra över telefon nätet. Förutsättningen är att de förstår varandra. Man har därför infört teckensed som är standardiserade. Dessvärre finns det ett flertal sk standardiserade teckensed, men det vanligaste är ASCII som TEXASen använder. ASCII står för American Standard Code for Information Interchange.

## MODEMET:

Modem står för MOdulator DEModulator. För att kunna skicka och ta emot data så behövs ett modem. Signalerna ut från datorn måste göras om så de passar telenätet. Telenätet har sin aktiva gräns mellan 50 och 2400 Hz. Man utgår från en bärväg (carrier) och sedan sänder man en högre eller lägre frekvens för att indikera en etta eller nolla. En modulator sänder, dvs. gör om datorns signaler till frekvenserna vi diskuterat ovan. Gissa vad en demodulator gör ??  
Ja, omvandlar frekvenser givna ovan till data signaler.

## ANSWER-ORIGINATE:

För att börja enkelt (det slutar enkelt också). Om vi har två exakt likadana modem, då sänder vi och tar emot på samma frekvens, det är inte bra ! (Observera INTE)  
Därför låter man den ena terminalen/datorn vara ett ORIGINATE och den andra ett ANSWER. Dessa två går hand i hand. Ett ORIGINATE modem sänder på den frekvens ANSWER modemet tar emot på och vice versa. Europa har en standard och Amerika en annan, ingen kan samsas. Europeernas heter CCITT och då kallas ORIGINATE för Kanal 1 eller A samt ANSWER för kanal 2 eller B. Ett A modem brukar inleda ett 'samtal' mellan datorer/terminaler.

## NITTINIANS DATANÄT

Vi (och många med oss) tycker att det vore kul med ett datanät för medlemmar i föreningen. Vi hoppas att det tas upp på allvar i föreningen.

Alla som vill vara med måste förstås ha ett modem och ett RS232 (RS232 används för att koppla in dig på en standard kommunikations uppsättning). Inom föreningen kan vi ta fram ett modem fast det beror ju på underlaget. Hur många är intresserade av en DATABASE ?

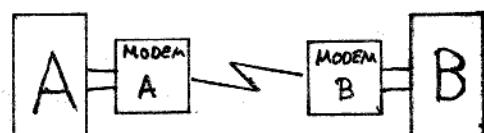
Priset kommer isäfall till medlem att ligga mellan 800-1000 Skr. Sedan tillkommer ett gränsnitt sk RS232 för c:a 1000 Skr.

## DU SOM ÄR INTRESSERAD SKRIV TILL OSS:

Föreningen Programbiten  
c/o Schibler  
Wahlbergsgatan 6 nb  
121 46 Johanneshov

Märk kuvertet MODEM.

Nu är det slut men vi återkommer.



## DATAVISION

av Michael Öhman

Datevision är den största databasen i sverige. TEXASen har alla möjligheter att koppla upp sig mot den. Det behövs bara ett terminalprogram. I BASIC är det dessvärre omöjligt, så programmet måste vara skrivet i maskinkod. I föreningen kommer en modul att säljas för c:a 550-600 kr. Programmet måste först och främst kunna kommunicera med 1200 BPS (=bits per second = baud) i mottagar riktningen och 75 BPS i sändar riktningen. Skärmen skall också kunna generera 40x24 tecken med en individuell färg på varitida bokstav. TEXASen har något dålig färgupplösning så en färgkollision i viss mån är oundviklig. Men det är sällan många färgändringar efter varandra. För er som vill kunna koppla upp er mot DATAVISIONEN hör av er till föreningen. Programmet kommer att kunna spara skärmar till Kassett/Diskett skriva ut på printer osv.

Addressen till föreningen står till vänster. ■

Med full duplex menas att man sänder och tar emot samtidigt. Det är omöjligt i BASIC, så du som är intresserad av att kommunicera med en annan texasägare hör av dig. En del assemblerprogram finns att tillgå som använder full duplex. Det behövs ett minimemory och ett RS232 samt ett modem. Spel av olika slag kan spelas med hjälp av modem, vad skulle ni säga om ett wargames där bågge två spelar samtidigt Ryssland mot USA. Hjälp ! TELEverkets önskedräm ? ■

## FÖRENINGENS TILLBEHÖRSFÖRSÄLJNING

Följande finns att köpa för medlemmar genom att motsvarande belopp insättes på postgiro 19 83 00-6

Användartips med Mini Memory	60:-
FORTH (ange kassett/diskett-version)	
(se annons till höger = nästa sida)	250:-
Nittinian T-tröja	40:-
99'er Magazine nr. 11,12/82 nr. 1-6,8,9,11/83	20:-
Nittinian, årgång 1983	80:-
Astronomical Formulae for Calculators (viss leveranstid)	55:-
Programbiten, årgång 1983	80:-
Programbiten, årgång 1982	80:-
Programbiten, årgång 1981	60:-
Programbiten, årgång 1980	60:-
Programbiten, årgång 1978/79	60:-
Programbiten, fem årgånger 1978-1983	280:-
Katalog med belgiska och engelska program för räknare TI 57, TI 58, TI 59	20:-
Patenthandlingar TI-59	25:-
Föreningens programmeringsblanketter (TI-59), olika typer, block om 50 blanketter (se PB 83-1 sidan 30)	11:-
40 st tomma magnetkort och plånbok	150:-
Tom magnetkortsplånbok	10:-

# FÖRENINGENs FORTH

## TI FORTH

Denna beskrivning över programspråket  
TI FORTH  
erhållen från Maurice Swinnen, USA  
kan köpas genom:

FÖRENINGEN PROGRAMBITEN,	POSTGIRO: 19 83 00 - 6
för MEDLEMMAR	för ICKE MEDLEMMAR
190 SEK	270 SEK

Föreningen har även programmet på diskett, vilken kostar:  
försäljning för MEDLEMMAR försäljning för ICKE MEDLEMMAR  
60 SEK 100 SEK

För beskrivning och diskett tillsammans:  
försäljning för MEDLEMMAR försäljning för ICKE MEDLEMMAR  
250 SEK 370 SEK

Föreningen har även en egen FORTH framtagen av Björn Gustavsson,  
vilken inklusive kortfattad beskrivning kostar:  
försäljning för MEDLEMMAR försäljning för ICKE MEDLEMMAR  
250 SEK 370 SEK

Vid SAMTIDIG beställning av TI FORTH och Föreningens FORTH  
försäljning för MEDLEMMAR försäljning för ICKE MEDLEMMAR  
400 SEK 600 SEK

Medlem i FÖRENINGEN PROGRAMBITEN blir man enklast genom att betala  
in 120 SEK på POSTGIRO: 19 83 00 - 6.

Reservation mot prisändringar på grund av materialfördyrningar  
utanför vår kontroll.

Stockholm 1984-06-20

### DATAVISION MODUL

Modulen gör att du kan kommunicera med DATAVISION eller liknande baser.

Utskrift för printer finns tillgänglig, sparning av skärmar på kassett eller diskett.

Nödvändig utrustning:

RS232 interface

Televerkets modem eller liknade

Modulen kommer med instruktioner om hur du kopplar upp dig med Televerkets modem.

Pris: 565:-

### 16 KRAM CMOS MODUL

utan hölje. Modulen kan simulera ett ROM, inläddning sker från T.ex FORTH vilken medför att minimum utrustning är X-basic, MINIMEMORY, eller ED/ASM och 32 K minnesutökning. Ram'et är batteribackupat så programmet ligger kvar minst 2 veckor. Denna modul används i princip för att testa ut hur egna program i modulform fungerar innan man bränner prommar vilket blir klart jobbigare.

Pris: 675:-

### DISK DRIVER

Fabrikens nya driver från Shugart. Kan direkt kopplas in i expansionslåda.

Enkelsidig: 1200:-  
Dubbelsidig: 1600:-

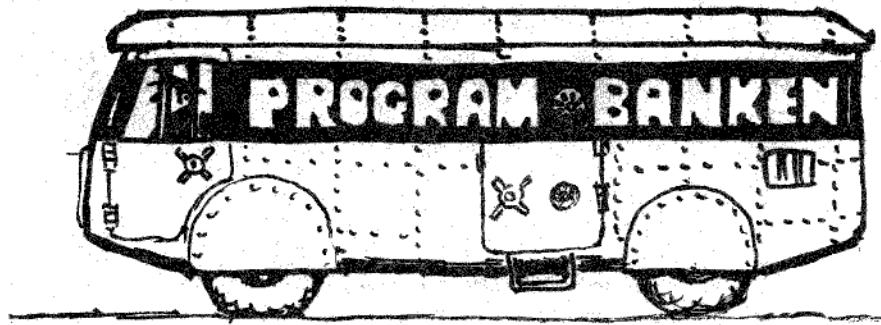
Specialerbjudande: Halvhöjdsdrivar. Två stycken får plats i expansionslådan. Grenkontakt och monteringsanvisning medföljer.

Pris per PAR: 3500:-

Även begagnade enkelsidiga driver finns.

Eventuell frakt tillkommer.

Björn Gustavsson  
Lagfartsvägen 46  
145 58 Norsborg



Här visas en lista på de tillgängliga programmen i Programbanken. Programmen kan numera köpas. Priset är uppdelat i två delar, dels en startkostnad som täcker kostnaden för porto och mediet, dels en kopieringsavgift för varje program.

Startkostnad, kassett	35:-
Startkostnad, diskett	55:-
Kopieringsavgift, kassett	15:-
Kopieringsavgift, diskett	10:-

Som förut får du tre program i utbyte när du skickar in ett BASIC-program.

#### Programkategori:

- 14 = Ekonomi
- 20 = Regression, Kurvanpassning
- 21 = Statistik, Varians
- 29 = Statistik, Sannolikhet
- 30 = Linjär algebra
- 39 = Allmän matematik
- 65 = Elektronik

- 78 = Astronomi
- 90 = Praktiska programhjälpmödel
- 91 = Spel
- 92 = Utbildning
- 96 = Musik
- 97 = Demo och Musik
- 99 = Övrigt

01201001E --- LINEAR REGRESSION  
 01201002E --- TREND LINES ANALY  
 01391001E --- 8 MATH PROGRAMS  
 01391002E --- SIMULT. EQUATIONS  
 01651001E --- ELECTRONICS  
 01781001E --- GEOSTAT. SATELITE  
 01901001E --- P WORD-PROCESSOR 1  
 01901002E --- MAILING LIST  
 01911028H --- CAR DRIVER  
 01911029H --- ANIMALS  
 01911030H --- SPACE INVADER  
 01921006H --- X GENETICS  
 01971001H --- C 99/4A DEMO  
 01991001E --- X ANCESTRAL FILER  
 01991002H --- CALENDAR W. PRINT  
 01992001E --- MOON CYCLE  
 02911001E --- MASTERMIND  
 02911002E --- X SCHMOO  
 02911003E --- STARTRECK  
 02911004E --- AIR - SEA BATTLE  
 02911005E --- QUEENS  
 02911006E --- OTHELLO 2-SPELARE  
 02911007E --- CHASE  
 02911008E --- PUZZLE 15  
 02911009E --- NUMBERS AWAY  
 02911010E --- AWARI  
 02911011E --- KENO  
 02911012E --- +D STARGUARD  
 03911013E --- MINER  
 03911014E --- C YAHTZEE  
 03911015E --- BACKGAMMON  
 03911016E --- 3D TIC-TAC-TOE  
 03911017E --- X BLACKJACK  
 03911018E --- X NOT ONE  
 03911019E --- X POKER  
 03911021E --- DONKEY'S TAIL  
 03911023E --- X SPACE GEMS  
 03911024E --- SCORE FOUR

De två första siffrorna i programnumret anger vilken diskett programmet ligger på.

De nästa två siffrorna anger programkategori se nedan Femte siffran anger ursprungsland:

1 = USA, 2 = Holland, 7 = Sverige.

De tre sista siffrorna är ett löpnummer och bokstaven som finns sist anger vilket språk som används i programmet: E = Eng., H = Holl., F = Fra., S = Svenska. Bokstäverna efter numret talar om vilken utrustning som krävs: X = Ext. Basic, D = Diskett, P = Printer E = Minnesexp., J = Joystick, C = Call Files (1) måste användas om man har disk. Ett + anger att programmet är delat i flera delar.

07141001H	---X ANNUITIES
07901005H	--- MORSE
07902002	---F BANNER (BIG TEXT)
07911035H	---X WARI
07911036H	--- LABYRINTH
07911037H	---X ELIZA
07911038H	--- SECRET THOUGHTS
07911039H	---X CATCH THE CHAR.
07911040H	---X BLACKJACK
07911041H	---X SPACE BATTLE
07911042H	---X MAGIC WORD
07911043H	--- JACKPOT
07911044H	--- LABYRINTH 2
07921007E	--- PAST TENSE
07962013	--- LOOKING THROUGH U
07971003H	--- BIG CHARACTERS
08901006E	--- XE UNIV. TITLE PAGE
08902003	--- LARGE CHARACTERS
08911039E	---X SPRITE CHASE
08911046E	---X CHECKERS
08911049E	---X SWORDS & SORCERY
08911051E	---X EGGWARS
08961001E	---X KILLING ME SOFTLY
08962014	---X CHOPIN OP 10 #3
09291001H	--- PROBABILITY
09391003H	--- DIFFERENTIAL EQ.
09911052E	--- DEEP SPACE
09911053F	--- ATOMES
09911054H	--- PLANETARY LANDER
09911056H	---X LUNAR LANDER
09911057F	--- TOWERS OF HANOI
09911058E	--- BREAKOUT
09911059H	---J DRAWING ON SCREEN
09911060H	---JX TREASURE HUNT
09912008F	---X ASTEROID
09912009	--- BATTLE STAR
09991003F	--- BIORYTHM
10392002	--- PRIME NUMBERS
10901007H	--- LOTTO (HOLL.)
10901008E	--- VERY LARGE CHAR.
10901009E	--- KEY DEFINER 99/4
10901010E	---C TEXTPRO (99/4)
10911061H	---X KERMIT
10911062H	---JX FIND THE GUN
10911063E	--- CHASE A TARGET
10911064H	--- RUSSIAN ROULETTE
10911065E	--- CAMEL
10921008E	--- COLOR FRACTIONS
11901011S	--- STAPELDIAGRAM
11911055S	---X HANGNING
11917001S	---JX INKRAEKTARNA
11917002S	---JX PAERON