Claes Schiblers PR ex

PROGRAM BITEN 82-1

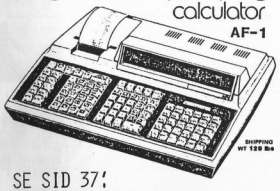
Innehåll

Årsmötesprotokoll 3
Pgm-bitar 5
Utmaningen 6
Mullers metod 6; Dse-funktion 9; Superprimtal 9; Skrivkodstabeller 11; Primtalstest 12; 1/a+1/b=1/c 13; Mr Magic 14
Snabbt kalenderprogram 17
Nyheter på TI-58 C 18
Hobbit 21
Djupdykning i TI-59 (3) 24
Tävling - sänka skepp (vinnande pgm) 26
Svar om HP-41 C 27
Avancerade programmeringsmetoder - 2 28
Å, Ä och Ö på PC-100! 31
Kortaste avstånd mellan korsande linjer 32
Rättelse: Tippa med TI-59 34
Tredjegradsekvation 35
Exekveringstider 36
Senaste nytt från USA 37
Programbiblioteket 38

SENASTE NYTT FRÅN USA!

ÄR DETTA RÄKNAREN
SOM SLÅR UT TI-59?

OK# INSTRUMENTS
programmable prompting calculator
AF-1



SE SID 37!

Dyre medlem!

När Programbiten nu börjar sin fjärde årgång är det faktiskt med ganska stor optimism. Medlemstillströmningen är tillfredsställande, och som Du ser i kuvertpåsen har vi tack vare bl a Björn Gustavssons och Jan Gustafssons insatser fått dels ett "Komplement till instruktionsboken" som framför allt kommit till med tanke på nya medlemmar men som vi hoppas kommer att uppskattas även av de gamla, dels ett register för våra tre första årgångar. Vidare har vi också en katalog med beskrivningar av våra svenska program som alltså inte tar upp plats inne i numret.

"Utmaningen" är den här gången rekordlång med elva sidor och många intressanta bidrag. Björn Gustavsson har flera andra bra bidrag också, t ex om mer avancerad sökning i register. Spelet Hobbit av vår jugoslaviske medlem Dejan Ristanović är både ett bra spel och ett välgjort program som tål att granskas.

En svensk programkatalog går alltså till alla medlemmar. En katalog över tillgängliga belgiska och engelska program har just tryckts, men den är så tung att vi endast sänder ut den på beställning till självkostnadspris.

En sak till som just nu trycks är en informationsbroschyr om föreningen med avtryck av sidor i gamla nummer av Programbiten. Om Du inte själv kan visa vårt medlemsblad för intresserade vänner så kan Du be oss skicka den här broschyren.

Sedan sist har vi också haft årsmöte med åtföljande samvaro. Även om deltagarantalet inte är så imponerande varken absolut eller procentuellt så var vi i år fler än vi brukar, och vi hade en intressant diskussion om föreningens fortsatta inriktning som gjordes mer flexibel genom att en stadgeparagraf ändrades.

Överhuvudtaget var det en trevlig träff, och särskilt var det trevligt att träffa medlemmar som kommit resande ganska långtifrån. TI:s nya version av hemdatorn TI-99/4 A tog en del uppmärksamhet efter årsmötet, men det hanns också med att prata en del om programmering.

För att även i fortsättningen försöka hålla medlemsbladet på hög nivå har vi fortfarande behov av bidrag, men även av granskare till de bidrag som kommer in. Tyvärr är tekniska/vetenskapliga program ganska sällsynta, men när vi får dem vill vi ju också att någon kunnig person skall se igenom dem. Hör av Dig om Du har tid och gärna kommer i kontakt med andra programmerare.

Våren är kanske på väg. Innan de riktigt varma kvällarna kommer träffas vi en gång till ute i Farsta före Valborg.

Lars Hedlund, ordf

Medlemsträff

Du som kan ta Dig till Stockholms södra förorter en lördageftermiddag är välkommen till nästa medlemsträff i Folke Johanson Ingenjörsbyrås lokaler vid Farsta Centrums tunnelbanestation lördagen den 24 april 1982 kl. 14 - 18.

Kom och bekanta Dig med oss och prata programmering. Ring på vid det runda trapphuset på stationens sydöstra sida.

REDAKTIONEN

har kanske litet obestämda gränser, men här är dock två av de mest idoga medlemmarna. Överst Björn Gustavsson, välkänd från "Utmaningen" och många andra artiklar, och nedtill Ingvar Magnusson, "a new giant in TI-57 programming" enligt TI PPC Notes och dessutom trogen manusmonterare. En annan trogen medarbetare i praktiska frågor är Claes Schibler som finns i tredubbel upplaga i 79-3&4.

TILLFÄLLE !
20 ex av Sourcebook kvar för 80 skr/st

Texas Instruments som tar in Sourcebook för oss erbjöd oss för en tid sedan att köpa den upplaga som fanns i lager till nedsatt pris. Vi låter prisnedsättningen gå vidare till medlemmarna och kan nu erbjuda boken för 80 kr.

Dessutom har vi ett paketpris för de tre första årgångarna av Programbiten - 200 kr! Dessa priser gäller nu (beställ genom att sätta in motsvarande belopp på postgiro 430 01 59 - 3):

Sourcebook for Programmable Calculators (liten restupplaga)	80,-
Programbiten, årgång 1981	100,-
-"- -"- 1980	80,-
-"- -"- 1978/79	60,-
-"- , tre årgångar 1978/79 - 1981	200,-
Patenthandlingar TI-59	25,-
Astronomical Formulae for Calculators (viss leveranstid)	55,-
40 st tomma magnetkort och plånbok	108,-
Tom magnetkortsplånbok	10,-
Föreningens programmeringsblanketter, olika typer, block om 50 blanketter (se PB 79-3&4)	11,-

SKRIVREGLER

Vi ser fram emot bra bidrag till Programbiten, men gör oss ännu gladare genom att följa de här skrivreglerna:

Spaltbredd max 11,5 cm och radavstånd "1 1/2 kugg".

Kontrasten är mycket viktig! Använd nytt färgband och sänd alltid med magnetkort som ersättes av oss. Listning vill vi ändå gärna ha, men montera den inte!

PROGRAMBITEN och FÖRENINGEN PROGRAMBITEN
c/o Hedlund
Årstavägen 27 6 tr
121 68 JOHANNESHOV

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, magnetkortsinspelning etc.

Tryck: K-TRYCK AB STOCKHOLM

ÅRSMÖTES PROTOKOLL

ÅRSMÖTE I FÖRENINGEN PROGRAMBITEN

TID: 82.01.30

PLATS: Folke Johanson Ingenjörsbyrå's lokaler, FARSTA.

Närvarande: 21 medlemmar

1. **Val av ordförande och sekreterare för mötet.**

   Till ordförande valdes Lars Hedlund och till sekreterare Lars Jansson

2. **Val av justeringsmän.**
   Arne Andersson och Björn Gustavsson valdes.

3. **Tid och sätt för kallelsen godkändes.**

4. **Dagordningen godkändes.**

5. **Verksamhetsberättelsen lästes upp.**
   Verksamhetsberättelsen godkändes.

6. **Balansräkning och revisionsberättelse.**

   Ordinarie revisor hade i sista stund meddelat förhinder, varför revisorssuppleant Lars Winter tillsammans med Arne Andersson (av mötet vald att biträda) fick i uppdrag att sammanställa revisionsberättelsen. Så fort detta arbete har fullföljts skall revisionsberättelsen sändas ut till mötesdeltagarna för godkännande/kommentar. Resultatet skall därefter presenteras i Programbiten.

   Balansräkningen redovisades av kassören.
   Balansräkningen godkändes.

7. **Styrelsens ansvarsfrihet efterfrågades.**
   Mötet beviljade styrelsen ansvarsfrihet under förutsättning att revisionsberättelsen kunde godkännas av mötesdeltagarna (Se pkt. 6 ovan).

8. **Föreningens årsavgift.**
   Tidigare årsavgiften 120 kr föreslogs. Mötet beslöt att ta upp frågan under pkt.10.

9. **Frågan om ändring av stadgarnas paragraf 8 togs upp.**

   En livlig debatt följde varvid följande frågor togs upp:

   Bör vi intressera oss även för datorer? I så fall, vilka?
   Hur ställer vi oss i förhållande till andra räknare?
   Bör vi begränsa oss till format, språk, eller tillverkare?
   Eller bör vi begränsa oss till apparater som inte kräver nätanslutning för att kunna fungera?

   Efter att dessa frågor ventilerats en stund beslöt mötet att fortsätta med denna fråga efter kaffepausen.

Forts. **Årsmötesprotokoll**

Under denna punkt togs även två andra frågor upp, nämligen:

Marknadsföring av föreningen och medlemstillströmning.

Beträffande marknadsföringen föreslog Sven Carlson att en marknadsföringsgrupp för föreningens spridning skulle bildas. Sven Carlson erbjöd sig även att samordna en sådan grupp.

Mötet beslöt att frågan skulle tas upp på nytt efter kaffepausen.

Claes Schibler rapporterade om medlemstillströmningen:
Föreningen har nu fått in 60 % fler årsavgifter än förra året vid samma tidpunkt.

10. Årsplan för 82.

Föreningens årsplan diskuterades varvid följande framkom:

Tidningstiderna är bra och kvaliteten på tidningen och dess medarbetare bedömdes som mycket god. Speciellt Björn Gustavsson berömdes för sin insatser.

Mötet beslöt att samma inriktning som föregående år skulle gälla, dvs 4 tidningar med minst 40 sidor vardera.

Frågan om årsavgift togs upp på nytt.
Mötet beslöt att 120 kr skall fortsätta att gälla. Dock har styrelsen fullmakt att, om så visar sig vara nödvändigt, ändra medlemsavgiften för nästkommande år redan i god tid före slutet av innevarande år. Om så sker skall detta presenteras i tidningen.

11. Val av ordförande.
Valberedningens förslag skickades runt, varefter val av ordförande vidtogs.

Till ordförande valdes Lars Hedlund.

12. Val av övriga styrelseledamöter.
Styrelsens storlek, sammmansättning och antal suppleanter diskuterades. Följande styrelse valdes:

Kassör:Sven Carlson
Sekreterare: Björn Svensson
Materialförvaltare: Claes Schibler
Klubbmästare: Anders Brolin
Styrelsemedlem: Ingvar Magnusson
Styrelsemedlem: Björn Gustavsson

Programförmedlare: Bo Nordlin
(Ej ordinarie styrelsemedlem, dock suppleant)

13. Val av revisor.
Två revisorer och en suppleant valdes.

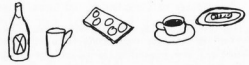
Revisorer: Lars Winter ,
Arne Andersson

Revisorssuppleant: Arne Melander

14. Val av valnämnd.

Till valnämnd valdes: Lars Jansson
Gunnar Svanborg

Mötet tog paus för kaffe och smörgåsar kl. 15:15 - 16:45



15. Övriga frågor.

Stadgeändring av paragraf 8 diskuterades ingående. Efter votering mellan två kvarstående likartade förslag beslöt mötet att paragraf 8 skulle få följande lydelse:

Föreningen Programbiten startade som en förening för användare av programmerbara räknare vars programspråk är sådant som hos Texas Instruments TI57/58/59. Föreningen skall bygga vidare på dessa traditioner och även vid behov vidga sina intresseområden i takt med den tekniska utvecklingen och medlemmarnas önskemål.

Övriga beslut som togs var:

För att styrelsen skall vara beslutsmässig måste minst fyra medlemmar vara närvarande.

En marknadsföringsgrupp skall bildas med Sven Carlson som sammankallande. Gruppen skall sammansättas av frivilliga krafter.

16. Mötet avslutades kl. 17:30

Vid protokollet

Lars Jansson

Lars Jansson

Justeras

Lars Hedlund

Lars Hedlund

Arne Andersson Björn Gustavsson

Arne Andersson Björn Gustavsson

Följande medlemmar deltog i årsmötet. De som har märkts med * har meddelat att de godkänner revisionsberättelsen (76 %; inga har meddelat att de inte godkänner den). Bo Nordlin *, Björn Gustavsson *, Arne Andersson *, Klas Schöldström, Anders Persson *, Björn Nordell *, Arne Melander *, Olof Magnusson *, Erik Stoy, Carl-Magnus Ekermann *, Anders Brolin *, Lars Winter *, Anders Nawroth, Sven Carlson *, Ingvar Magnusson *, Caj Erlö *, Hans Hellström, Lars Jansson *, Gunnar Svanborg, Lars Hedlund *, Claes Schibler *. ■

Egen modul!

Så här skriver Maurice Swinnen i senaste numret av TI PPC Notes:
"Jag har (med avsikt) dragit ut på tiden de senaste två månaderna och hoppats att TI till sist skulle tala om hur nästa räknare skulle se ut. Om den visar sig vara en "förfinad TI-59" med t ex moduler som man själv kan spela in är det ingen mening att vi nu utvecklar en modul. --- Men om den nya visar sig vara något helt skilt från TI-59 (kanske en handdator à la Radio Shack eller Sharp) då fortsätter vi med vår egen modul. Då blir inte TI-59 omodern så snart. Under inga förhållanden vill jag göra av med (våra gemensamma) pengar på något som kan bli föråldrat innan vi är färdiga med det."
Med dessa tungt vägande ord avvaktar vi den vidare utvecklingen, SE ÄVEN "PS" PÅ S. 26!

ATT "NEUTRALISERA" EN INSTRUKTION

Instruktionen "Lbl" kan användas för att "neutralisera" en följande instruktion. Här ett exempel för att beräkna en siffras skrivkod: (Markús S. Markússon)

```
000  76 LBL   005  77  GE   009  76 LBL
001  59 INT   006  00  00   010  01  1
002  85  +    007  10  10   011  95  =
003  32 X:T   008  03  3    012  92 RTN
004  08  8
```

TEST OM ETT TAL ÄR "NOLL"

Man måste ibland testa om ett tal är noll. Om en beräkning ger 4.1-09 - skall man anse att det är noll? Om man väljer t ex kvantiteten 1-08 att jämföra med kan man använda detta program:

```
000  76 LBL   004  52 EE   007  22 INV
001  67  EQ   005  22 INV   008  58 FIX
002  58 FIX   006  52 EE    009  92 RTN
003  08  8
```

Men om man senare i samma program vill jämföra med 1-04? Ja, en ny liknande rutin, men 1-10 skulle inte gå alls. Denna pgm-bit löser problemet:

```
000  76 LBL   006  53  (    012  69 OP
001  16 A'    007  50 I×I    013  10  10
002  22 INV   008  65  ×    014  =
003  28 LOG   009  32 X:T   015  92 RTN
004  32 X:T   010  54  )
005  65  ×    011  59 INT
```

Jämförelsetalet "noll" lägges i t-registret och antalet signifikanta siffror efter decimalkommat ges före anropet A'.

(Markús S. Markússon)

VEKTORSUMMERING

Per Eriksson har skickat in pgm 1 som summerar vektorer. Nollställning som skall utföras före varje beräkning görs med E. Mata sedan in alla vektorer på följande sätt: Mata in vektorns längd och tryck A. Mata in vinkeln och tryck A igen. Efter varje vinkelinmatning visas summan av de inmatade vektorerna med längden i X-registret och vinkeln i T-registret.

Pers program använder en flagga för att all inmatning skall kunna göras på en tangent. Björn Gustavsson föreslog att programmet kan göras enklare och kortare som i program 2, om inmatningsrutinen ändras. Björns program handhas så här:

1. Tryck E. Ignorera displayen.
2. Mata in vektorns längd, tryck x:t, mata in vinkeln, tryck R/S eller A. Upprepa för alla vektorer.
3. Tryck E. Summan av de inmatade vektorerna visas med vinkeln i X-registret och längden i T-registret. Samtidigt nollställs summationsregistren för en ny beräkning.

Om man i steg 3 ångrar sig efter att ha tryckt E, kan man trycka R/S eller A och fortsätta som om inget hänt. Detta förutsätter dock att X- och T-register inte har ändrats.

```
000  76 LBL   022  32 X:T   000  76 LBL   013  11  A
001  11  A    023  43 RCL   001  15  E    014  37 P/R
002  87 IFF   024  00  00   002  25 CLR   015  44 SUM
003  01  01   025  22 INV   003  48 EXC   016  01  01
004  22 INV   026  37 P/R   004  00  00   017  32 X:T
005  29 CP    027  22 INV   005  32 X:T   018  44 SUM
006  85  +    028  86 STF   006  00  0    019  00  00
007  32 X:T   029  01  01   007  48 EXC   020  25 CLR
008  95  =    030  32 X:T   008  01  01   021  29 CP
009  86 STF   031  91 R/S   009  22 INV   022  92 RTN
010  01  01   032  76 LBL   010  37 P/R   023  61 GTO
011  91 R/S   033  15  E    011  92 RTN   024  11  A
012  76 LBL   034  29 CP    012  76 LBL
013  22 INV   035  25 CLR
014  37 P/R   036  42 STO
015  44 SUM   037  01  01
016  00  00   038  42 STO
017  32 X:T   039  02  02
018  44 SUM   040  22 INV
019  01  01   041  56 STF
020  43 RCL   042  01  01
021  01  01   043  91 R/S
```

Program 1 till vänster,

program 2 till höger.

JÄMFÖRELSE AV TVÅ NÄSTAN LIKA STORA TAL

När x och y är av samma storleksordning blir testet "x - y = 0?" något tvivelaktigt, men om man i stället testar "1 - y/x = 0?" förloras inga signifikanta siffror och testet kan användas med största förtroende. Ett annat exempel är "$x^2 + y^2$ = 0?". Här kan man använda "1 + y^3/x^2 = 0?". (Markús S. Markússon)

STOPP I EXEKVERING GENOM "SST"

Av något underligt skäl stoppas ett program av instruktionen SST (41) följd av D.MS eller P/R. Om någon vet något mer om detta eller andra besynnerligheter med SST, skriv en rad!

(Markús S. Markússon) ■

UTMANINGEN!

Redaktörens adress:

Björn Gustavsson
Nordlandervägen 12 A

777 00 SMEDJEBACKEN

Hej!

Välkomna till en ny Utmaning. Jag har mottagit ovanligt många brev med lösningar till det här numret. Det här är därför den hittills längsta Utmaningen, hela elva sidor.

Det kanske märks att jag har köpt en ny skrivmaskin. Maskinen heter Olivetti Praxis 35 och har utbytbara skrivhjul och tre olika bredder på tecknen. Jag hoppas därför att den här Utmaningen åtminstone typografiskt är bättre än de föregående.

Nu går det att köpa magnetkort för Utmaningen 82-1 och framåt från programförmedlare Bo Nordlin för normalt programpris 30 kr (+10 kr för varje beställning). Jag hoppas att jag kan sammanställa magnetkort även för de tidigare utmaningar som Sven Östberg redigerade.

MULLERS METOD (9)

Problem 9 gick ut på att göra ett program som löser ekvationen $f(x) = 0$. Markús Sveinn Markússon har gjort ett utmärkt program som kan ge alla rötter, både reella och komplexa, av ett polynom av upp till 35:e graden. Det är också möjligt att få en eller flera rötter av en egen funktion $f(x)$ som matas in i programminnet.

Programmet använder Mullers metod, som beskrivs så här av Markús:

> En av många metoder som används för att lösa icke-linjära ekvationer är Mullers metod. Den får många lovord i boken "Elementary Numerical Analysis: An Algorithimic Approach" av Conte och de Boor. Den här metoden kan användas för att finna vilket föreskrivet antal nollställen, reella eller komplexa, som helst av en godtycklig funktion. Den konvergerar nästan kvadratiskt i närheten av en rot, kräver inte någon beräkning av funktionens derivata och ger både reella och komplexa rötter.

I Mullers metod utgår man från tre approximationer x_i, x_{i-1}, x_{i-2} av roten och beräknar en ny approximation x_{i+1} som är ett nollställe till parabeln som går genom punkterna $(x_i, f(x_i))$, $(x_{i-1}, f(x_{i-1}))$ och $(x_{i-2}, f(x_{i-2}))$. Förhoppningsvis är x_{i+1} en bättre approximation till roten än de föregående.

Den som är intresserad av hur beräkningarna görs i detalj kan skicka 3:30 i frimärken till mig, så skickar jag kopior av 4 sidor ur den ovan nämnda Conte & de Boors bok. Där finns en beskrivning av Mullers metod, algoritmen för beräkning och ett FORTRAN-program.

```
000  76 LBL     030  95  =     060  01  01     090  00  00     120  05  05     150  00  0      180  01  01     210  22 INV
001  11  A      031  42 STO     061  34  34     091  96  96     121  85  +     151  01  1      181  98 ADV     211  87 IFF
002  22 INV     032  00  00     062  04  4      092  08  8      122  01  1     152  27  7      182  69 OP      212  00  00
003  76 LBL     033  69 OP      063  04  4      093  08  8      123  95  =     153  03  3      183  05  05     213  02  02
004  16 A'      034  00  00     064  06  6      094  44 SUM     124  69 OP     154  03  3      184  21  18     214  18  18
005  86 STF     035  02  2      065  00  0      095  02  02     125  99 PRT     155  03  3      185  01  1      215  03  3
006  00  00     036  00  0      066  00  0      096  02  XIT     126  69 OP     156  06  6      186  92 RTN     216  69 OP
007  22 INV     037  03  3      067  00  0      097  93  ,      127  30  30     157  69 OP     187  99 PRT     217  17  17
008  52 ENG     038  00  0      068  01  1      098  09  9      128  72 ST*     158  01  01     188  42 STO     218  02  2
009  22 INV     039  04  4      069  02  2      099  22 INV     129  00  00     159  98 ADV     189  08  08     219  00  0
010  58 FIX     040  01  1      070  42 STO     100  67  EQ     130  97 DSZ     160  69 OP     190  25 CLR     220  69 OP
011  42 STO     041  69 OP      071  01  01     101  01  01     131  05  05     161  00  0      191  03  3      221  07  07
012  05  05     042  01  01     072  00  0      102  05  05     132  00  00     162  04  4      192  05  5      222  69 OP
013  69 OP     043  02  2      073  42 STO     103  03  3      133  77  77     163  04  4      193  69 OP     223  19  19
014  25  25     044  02  2      074  00  00     104  76 LBL     134  25 CLR     164  92 RTN     194  04  04     224  25 CLR
015  42 STO     045  02  2      075  69 OP      105  01  1      135  02  2      165  99 PRT     195  03  3      225  68 NOP
016  06  06     046  07  7      076  00  00     106  44 SUM     136  04  4      166  -2 STO     196  02  2      226  93  ,
017  85  +      047  01  1      077  43 RCL     107  02  02     137  02  2      167  07  07     197  03  3      227  01  1
018  32 XIT     048  07  7      078  03  03     108  25 CLR     138  00  0      168  25 CLR     198  02  2      228  32 XIT
019  03  3      049  03  3      079  69 OP      109  43 RCL     139  03  3      169  02  2      199  03  3      229  01  1
020  05  5      050  05  5      080  23  23     110  02  02     140  02  2      170  01  1      200  07  7      230  94 +/-
021  77  GE     051  02  2      081  95  =      111  01  01     141  03  3      171  02  2      201  03  3      231  22 INV
022  00  00     052  00  0      082  01  1      112  01  01     142  01  1      172  00  0      202  06  6      232  96 WRT
023  28  28     053  69 OP      083  00  0      113  98 ADV     143  06  6      173  01  1      203  06  6      233  02  2
024  69 OP      054  02  02     084  95  =      114  67  EQ     144  02  2      174  07  7      204  42 STO     234  94 +/-
025  99  99     055  98 ADV     085  22 INV     115  05  05     145  69 OP     175  03  3      205  69 OP     235  22 INV
026  32 XIT     056  69 OP      086  59 INT     116  43 RCL     146  02  02     176  03  3      206  02  02     236  96 WRT
027  91 R/S     057  05  05     087  29 CP      117  06  06     147  04  4      177  03  3      207  98 ADV     237  25 CLR
028  02  2      058  87 IFF     088  22 INV     118  75  -      148  04  4      178  06  6      208  69 OP     238  61 GTO
029  05  5      059  00  00     089  67  EQ     119  43 RCL     149  02  2      179  69 OP     209  05  05     239  12  B
```

Pgm 1a. (Markús S. Markússon.) Mullers metod (inmatning).

Pgm 1a & 1b är Markús' mycket användarvänliga program för Mullers metod. Pgm 1a är en inmatningsrutin. Det spelas in på ett eget magnetkort i normaluppdelning. Pgm 1b utför själva beräkningarna. Det matas in och spelas in i normaluppdelning på ett magnetkort.

Programmet kan köras både med och utan skrivare. Om skrivare är ansluten kan du läsa i klartext vad som skall matas in härnäst. Rötterna skrivs då också ut i tur och ordning utan att programmet stoppas. Om skrivaren däremot inte är ansluten stannar programmet och visar en rot i taget.

I bruksanvisningen nedan skall stegen 2a, 3a, 10a och 11a utföras om funktionen är ett polynom. I annat fall utförs stegen 2b och 7b.

I steg 4 bestämmer ϵ_x hur stort felet i roten maximalt får vara. I steg 5 bestämmer ϵ_f hur mycket rotens funktionsvärde får skilja sig från 0. När ett värde har hittats som uppfyller ett av dessa krav avbryts sökandet efter bättre värden till den roten. Om funktionen är ett polynom försöker programmet beräkna ännu en rot, om inte alla rötter är funna.

```
000  76 LBL     060  97 DSZ     120  19 D'     180  43 RCL     240  43 RCL     300  04  04     360  32 XIT     420  42 STO
001  17 B'      061  05  05     121  92 RTN     181  23  23     241  20  20     301  10 E'      361  43 RCL     421  03  03
002  42 STO     062  00  00     122  76 LBL     182  44 SUM     242  49 PRD     302  82 HIR     362  20  20     422  43 RCL
003  04  04     063  45  45     123  12 B      183  11  11     243  02  02     303  15  15     363  17 B'      423  06  06
004  32 XIT     064  43 RCL     124  42 STO     184  19 D'      244  49 PRD     304  42 STO     364  43 RCL     424  49 PRD
005  42 STO     065  18  18     125  10  10     185  10 E'      245  21  21     305  01  01     365  17  17     425  05  05
006  03  03     066  32 XIT     126  42 STO     186  42 STO     246  52 EE      306  82 HIR     366  32 XIT     426  67  EQ
007  76 LBL     067  43 RCL     127  12  12     187  17  17     247  43 RCL     307  16  16     367  43 RCL     427  04  04
008  18 C'      068  24  24     128  32 XIT     188  32 XIT     248  17  17     308  42 STO     368  18  18     428  36  36
009  36 PGM     069  95  =      129  42 STO     189  42 STO     249  75  -      309  02  02     369  17 B'      429  69 OP
010  04  04     070  92 RTN     130  09  09     190  18  18     250  48 EXC     310  36 PGM     370  42 STO     430  36  36
011  13  C      071  76 LBL     131  42 STO     191  22 INV     251  15  15     311  05  05     371  19  19     431  11  A
012  92 RTN     072  16 A'      132  11  11     192  97 DSZ     252  95  =      312  13  C      372  32 XIT     432  22 INV
013  76 LBL     073  65  ×      133  93  ,      193  43 RCL     253  32 XIT     313  36 PGM     373  42 STO     433  37 P/R
014  19 D'      074  53  (      134  05  5      194  08  08     254  44 SUM     314  04  04     374  20  20     434  32 XIT
015  43 RCL     075  50 IX1     135  76 LBL     195  77  GE     255  18  18     315  17 B'      375  13  RCL    435  33 X#
016  12  12     076  55  ÷      136  13  C      196  04  04     256  75  -      316  36 PGM     376  21  21     436  42 STO
017  32 XIT     077  43 RCL     137  42 STO     197  15  15     257  48 EXC     317  05  05     377  32 XIT     437  04  04
018  43 RCL     078  07  07     138  23  23     198  43 RCL     258  16  16     318  14  D      378  43 RCL     438  02  2
019  11  11     079  54  )      139  92 RTN     199  16  16     259  95  =      319  65  ×      379  22  22     439  04  4
020  92 RTN     080  59 INT     140  76 LBL     200  75  -      260  17 B'      320  87 IFF     380  17 B'      440  42 STO
021  76 LBL     081  69 OP      141  14  D      201  48 EXC     261  02 HIR     321  01  01     381  42 STO     441  01  01
022  10 E'      082  10  10     142  22 INV     202  14  14     262  55  55     322  03  03     382  21  21     442  73 RC*
023  87 IFF     083  24 CE      143  76 LBL     203  95  =      263  32 XIT     323  29  29     383  48 EXC     443  01  01
024  00  00     084  95  =      144  15  E      204  42 STO     264  82 HIR     324  00  0      384  11  11     444  69 OP
025  04  04     085  92 PTN     145  86 STF     205  02  02     265  56  56     325  42 STO     385  42 STO     445  21  21
026  80  80     086  76 LBL     146  01  01     206  43 RCL     266  02  2      326  02  02     386  03  03     446  65  ×
027  42 STO     087  11  A      147  01  1      207  15  15     267  82 HIR     327  82 HIR     387  44 SUM     447  32 XIT
028  03  03     088  98 ADV     148  42 STO     208  75  -      268  65  65     328  06  06     388  11  11     448  43 RCL
029  32 XIT     089  04  4      149  19  19     209  48 EXC     269  82 HIR     329  82 HIR     389  32 XIT     449  04  04
030  42 STO     090  07  7      150  00  0      210  13  13     270  66  66     330  15  15     390  42 STO     450  94 +/-
031  04  04     091  69 OP      151  42 STO     211  95  =      271  82 HIR     331  85  +      391  22  22     451  95  =
032  43 RCL     092  04  04     152  20  20     212  42 STO     272  13  13     332  43 RCL     392  48 EXC     452  32 XIT
033  06  06     093  01  1      153  42 STO     213  01  01     273  22 INV     333  02  02     393  12  12     453  65  ×
034  44 SUM     094  16 A'      154  22  22     214  43 RCL     274  44 SUM     334  65  ×      394  42 STO     454  43 RCL
035  00  00     095  42 STO     155  43 RCL     215  19  19     275  03  03     335  82 HIR     395  10  10     455  03  03
036  48 EXC     096  11  11     156  23  23     216  32 XIT     276  82 HIR     336  16  16     396  44 SUM     456  95  =
037  05  05     097  44 SUM     157  22 INV     217  43 RCL     277  14  14     337  95  =      397  12  12     457  74 SM+
038  42 STO     098  02  02     158  44 SUM     218  20  20     278  22 INV     338  04  04     398  43 RCL     458  01  01
039  02  02     099  69 OP      159  11  11     219  17 B'      279  44 SUM     339  10  10     399  07  07     459  32 XIT
040  42 STO     100  06  06     160  42 STO     220  82 HIR     280  04  04     340  49 PRD     400  75  -      460  69 OP
041  18  18     101  02  2      161  21  21     221  03  03     281  36 PGM     341  01  01     401  36 PGM     461  21  21
042  01  1      102  05  5      162  19 D'      222  32 XIT     282  04  04     342  49 PRD     402  05  05     462  74 SM+
043  42 STO     103  69 OP      163  19 D'      223  82 HIR     283  10 E'      343  02  02     403  12  B      463  01  01
044  01  01     104  04  04     164  42 STO     224  04  04     284  43 RCL     344  82 HIR     404  55  ÷      464  69 OP
045  73 RC*     105  15  15     165  14  D      225  13  13     285  19  19     345  19  19     405  19 D'      465  21  21
046  00  00     106  32 XIT     166  32 XIT     226  82 HIR     286  32 XIT     346  44 SUM     406  22 INV     466  97 DSZ
047  85  +      107  16 A'      167  42 STO     227  05  05     287  43 RCL     347  01  01     407  37 P/R     467  05  05
048  18 C'      108  94 +/-     168  04  14     228  32 XIT     288  20  20     348  82 HIR     408  32 XIT     468  02  02
049  65  ×      109  42 STO     169  43 RCL     229  82 HIR     289  17 B'      349  16  16     409  95  =      469  42  42
050  53  (      110  12  12     170  23  23     230  06  06     290  69 OP      350  44 SUM     410  32 XIT     470  97 DSZ
051  69 OP      111  94 +/-     171  44 SUM     231  02  2      291  23  23     351  02  02     411  25 CLR     471  06  06
052  30  30     112  69 OP      172  11  11     232  42 STO     292  10  10     352  36 PGM     412  77  GE     472  01  01
053  73 RC*     113  06  06     173  01  1      233  01  01     293  43 RCL     353  05  05     413  47  47     473  47  47
054  00  00     114  32 XIT     174  10 E'      234  42 STO     294  17  17     354  15  E      414  84  84     474  98 ADV
055  65  ×      115  87 IFF     175  42 STO     235  02  02     295  32 XIT     355  43 RCL     415  11  11     475  98 ADV
056  32 XIT     116  07  07     176  15  15     236  43 RCL     296  43 RCL     356  19  19     416  87 IFF     476  98 ADV
057  54  )      117  01  01     177  32 XIT     237  19  19     297  18  18     357  85  +      417  00  00     477  91 R/S
058  44 SUM     118  21  21     178  42 STO     238  49 PRD     298  17 B'      358  01  1      418  04  04     478  76 LBL
059  18  18     119  91 R/S     179  16  16     239  01  01     299  36 PGM     359  95  =      419  77  77     479  71 SBR
```

Pgm 1b. (Markús S. Markússon.) Mullers metod (beräkning).

RÅD VID INSKICKNING AV PROGRAM

Om du har TI-59, skicka med magnetkort. Normalt skickar jag tillbaka blanka magnetkort inom några dagar. Skicka kopior, inte originalkort.

En enkel gardering mot fellåsningar består i att spela in programmet på båda sidor av kortet, om det endast tar upp ett block.

Skicka också med listning om du har skrivare.

Om funktionen inte är ett polynom avbryts körningen när en rot har hittats. Om det finns fler rötter kan användaren hitta dem genom att mata in en ny approximation till x_1 och h_1 i steg 8 och 9.

Programmet använder som begynnelseapproximationer $x_0 = x_1 - h_1$, x_1 och $x_2 = x_1 + h_1$, där x_1 och h_1 kan väljas av användaren. Om användaren inte matar in några värden, använder programmet $x_0 = -0.4$, $x_1 = 0.1$, $x_2 = 0.6$ och $h_1 = 0.5$.

Bruksanvisning:

1. Läs in inmatningsprogrammet.
2a. Om funktionen är ett polynom av n:te graden, mata in n och tryck A.
2b. Om funktionen inte är ett polynom, tryck A'.
3a. Mata in alla koefficienter, med början vid koefficienten för x^0. Tryck R/S efter varje koefficient.
4. Mata in ϵ_x och tryck R/S.
5. Mata in ϵ_f och tryck R/S.
6. Sätt in block 1 & 2 av beräkningsprogrammet i kortläsaren.
7b. Mata in funktionen: Tryck GTO SBR LRN f(x) RTN LRN. Vid anropet av funktionen finns Re x i displayen och Im x i T-registret. När funktionen avslutas skall Re f(x) finnas i displayen och Im f(x) i T-registret. Om funktionen är reell, ignoreras T-registret.
8. Om så önskas, mata in x_1. Tryck Re x_1 x:t Im x_1 B.
9. Om så önskas, mata in h_1 och tryck C.
10. Tryck D för reella rötter. Första roten visas.
10a. Tryck R/S n-1 gånger om skrivaren inte är ansluten för att visa resterande rötter.
11. För komplexa rötter tryck E. Re x visas i displayen och Im x finns i T-registret.
11a. Tryck R/S n-1 gånger för att visa resterande rötter om skrivaren inte är ansluten.

```
-MULLER-        X-EPSILON:
                      1. -08
X*00
     -6.8       F-EPSILON:
                      1. -11
X*01
     10.8            ROOTS:

X*02                 1.        +
    -10.8           -1.        J
X*03                 1.        +
      7.4            1.        J
X*04                 1.7       +
     -3.7            0.        J
X*05                 0.        +
      1.       -1.414213562    J
                     0.        +
               1.414213562     J
```

Figur 1. Exempel på användning av pgm 1.

```
-MULLER-          480 42 STO   497 19 D'
                  481 11 11    498 42 STO
                  482 94 +/-   499 01 01
X-EPSILON:        483 42 STO   500 32 X:T
      1. -08      484 01 01    501 42 STO
                  485 32 X:T   502 02 02
                  486 42 STO   503 36 PGM
F-EPSILON:        487 12 12    504 05 05
                  488 94 +/-   505 13 C
      1. -11      489 42 STO   506 36 PGM
                  490 02 02    507 04 04
                  491 36 PGM   508 12 B
   ROOTS:         492 05 05    509 85 +
                  493 17 B'    510 02 2
                  494 36 PGM   511 95 =
                  495 04 04    512 92 RTN
 .2612287672   +  496 10 E'
-1.465902909   J
```

Figur 2. Exempel på användning av pgm 1.

Några anmärkningar:

1. När f(x) matas in är det lämpligt att spara Re x och Im x i R_{11} respektive R_{12}. Om detta görs kan funktionen anropas från tangentbordet och subrutin D' kan ändå anropas av funktionen (se anmärkning 2). Funktionen anropas med E' och Re x i displayen och Im x i T-registret. Detta kan ofta vara till nytta, t ex om man vill veta om en rots konjugat också är en rot eller om du behöver några ledtrådar för att välja ut ett lämpligt värde för x_1.

2. Subrutin D' återkallar Re x och Im x från R_{11} respektive R_{12}. Detta kan korta ner en funktion, men snabbare är att skriva RCL 12 x:t RCL 11 direkt.

3. Om f(x) är en reell funktion skall T-registret vara tomt vid slutet av subrutinen. Det är tomt när funktionen anropas.

4. Om |Re x| och/eller |Im x| är mindre än ϵ_x, antas Re x och/eller Im x vara noll och trycks och visas så.

5. Om programmet söker efter en rot av en användar-definerad funktion kan programmet stoppas när som helst med R/S och en ny körning kan startas igen vid steg 8.

6. Eftersom Mullers metod är nästan kvadratiskt konvergent i närheten av en rot, så ökas inte exekveringstiden särskilt mycket om ett ϵ_x-värde = $1*10^{-8}$ används jämfört med ϵ_x = 0.001. Detta gäller också för ϵ_f.

Exempel: I figur 1 visas hur ekvationen $x^5 - 3.7x^4 + 7.4x^3 - 10.8x^2 + 10.8x - 6.8 = 0$ har lösts med Markús' program. Rötterna är alltså 1.7, 1-i, 1+i, $\sqrt{-2}$, $-\sqrt{-2}$, om man betecknar den imaginära enheten med i, i stället för j som programmet skriver ut.

I figur 2 har ekvationen $e^{-x}+x^2+2 = 0$ lösts. Roten är 0.2612287672+1.465902909i. Är också konjugatet en rot? För att besvara den frågan behöver vi bara trycka x:t +/- x:t E', som beräknar funktionsvärdet för konjugatet. Resultatet blir $10^{-12}+3\cdot10^{-13}$i, dvs nästan 0. Även konjugatet är en rot.

Observera: Eftersom funktionen skall kunna beräkna komplexa uttryck användes ML-04 och ML-05 som subrutiner.

DSE-FUNKTION (11)

I förra Utmaningen skrev jag om Markús Markússons program för DSE-funktion. Jag skrev att rutinen inte kunde användas i en modul, och gjorde därför om den så att den med hjälp av en flagga kunde användas i en modul. Jag efterlyste också bättre sätt att samordna huvudprogram och modulprogram.

Markús själv har löst problemet. Han skriver:

> När jag läste meningen "Den här rutinen kan inte användas som subrutin i t ex en modul" i Utmaningen PB 81-4 var jag övertygad om att det faktiskt stämde. Det var i december. Men i går upptäckte jag att det fungerar riktigt om det villkorliga hoppet GE XXX flyttas in i huvudprogrammet. Det var faktiskt vad jag hade i tankarna från början, men jag underlät att skilja huvudprogrammet från subrutinen. Programmet som jag skickade var bara ett som skulle visa hur metoden fungerar.

Han förbättrade också mitt exempel i förra numret. Om vi antar att pgm 2 finns i en modul med programnumret 2, kan det anropas som i pgm 3.

```
000  76 LBL     007  22 INV     014  32 X:T     021  22 INV
001  11  A      008  59 INT     015  00  0      022  28 LOG
002  53  (      009  65  ×      016  54  )      023  54  )
003  43 RCL     010  05  5      017  22 INV     024  42 STO
004  00  00     011  22 INV     018  59 INT     025  00  00
005  75  -      012  28 LOG     019  65  ×      026  59 INT
006  53  (      013  85  +      020  03  3      027  92 RTN
```

Pgm 2. (Markús S. Markússon.) DSE-funktion för användning i modul.

```
000  76 LBL     007  00  0      014  42 STO     021  11  A
001  11  A      008  00  0      015  00  00     022  77  GE
002  25 CLR     009  00  0      016  59 INT     023  00  00
003  01  1      010  03  3      017  42 STO     024  17  17
004  05  5      011  00  0      018  00  00     025  92 RTN
005  93  .      012  00  0      019  36 PGM
006  00  0      013  01  1      020  02  02
```

Pgm 3. (Markús S. Markússon.) Exempel på anrop av pgm 2.

SUPERPRIMTAL (16)

Problem 16 gick ut på att bestämma alla superprimtal. Ett tal är ett superprimtal, om talet är ett primtal och om alla tal, som erhålls genom att stryka en siffra i taget från höger, också är primtal. Det bör kanske också påpekas att 1 inte är ett primtal.

Det finns olika, mer eller mindre snabba, metoder för att lösa problemet. Patrik Johansson, Ursviken, och Erik Stoy, Svalöv, har båda gjort snabba program.

Den långsammaste metoden består i att testa de udda talen 3, 5, 7, 9,..., för att se vilka som är superprimtal. Ett sådant program blir utomordentligt långsamt. Jag har själv provat den metoden på en av ASEAs datorer under en praktikvecka på ASEA i Ludvika. Det tog några timmar att skriva ut superprimtalen under en miljon, och den datorn var snabb.

Snabbare går det om man utgår från de ensiffriga primtalen 2, 3, 5 och 7, som också är superprimtal. Från dessa byggs tvåsiffriga tal upp genom att tillfoga slutsiffrorna 1, 3, 7 och 9 (ett tal som slutar på 5 är delbart med 5). Man får då talen 21, 23, 27, 29, 31, 33, 37, 39, 51, 53, 57, 59, 71, 73, 77, 79. Man testar vilka av dessa som är primtal. De som är primtal är också superprimtal.

De tvåsiffriga superprimtalen är följande: 23, 29, 31, 37, 53, 59, 71, 73 och 79. Från dessa kan tresiffriga tal byggas upp genom att tillfoga slutsiffrorna 1, 3, 7 och 9. Genom att primtalstesta får man de tresiffriga superprimtalen. Man fortsätter på detta sätt tills man har n-siffriga tal, och inga primtal hittas bland dessa. Då är alla superprimtal funna.

```
000  25 CLR     025  99 PRT     050  73 RC*     075  09  9      100  88  88     125  05  05     150  18  18     175  00  00
001  91 R/S     026  42 STO     051  89 89      076  09  9      101  08  8      126  04  4      151  67  EQ     176  00  00
002  00  0      027  01  01     052  65  ×      077  95  =      102  06  6      127  08  8      152  01  01     177  75  -
003  76 LBL     028  03  3      053  01  1      078  53  (      103  42 STO     128  42 STO     153  46  46     178  01  1
004  11  A      029  99 PRT     054  00  0      079  02  2      104  00  00     129  88  88     154  32 X:T     179  07  7
005  36 PGM     030  42 STO     055  85  +      080  01  1      105  73 RC*     130  08  8      155  09  9      180  95  =
006  02  02     031  02  02     056  02  2      081  00  0      106  00  00     131  03  3      156  77  GE     181  42 STO
007  71 SBR     032  05  5      057  08  8      082  95  =      107  82 HIR     132  97 DSZ     157  00  00     182  89 89
008  02  02     033  99 PRT     058  08  8      083  59 INT     108  37  37     133  35  35     158  17  17     183  00  0
009  39  39     034  42 STO     059  22 INV     084  42 STO     109  82 HIR     134  01  01     159  01  1      184  63 EX*
010  09  9      035  03  03     060  82 HIR     085  35  35     110  16  16     135  03  03     160  44 SUM     185  87  87
011  00  0      036  07  7      061  18  18     086  29 CP      111  55  ÷      136  82 HIR     161  89  89     186  72 ST*
012  22 INV     037  99 PRT     062  95  =      087  01  1      112  82 HIR     137  16  16     162  73 RC*     187  89  89
013  58 FIX     038  42 STO     063  82 HIR     088  82 HIR     113  17  17     138  72 ST*     163  89  89     188  97 DSZ
014  22 INV     039  04  04     064  06  6      089  07  07     114  95  =      139  87  87     164  29 CP      189  87  87
015  57 ENG     040  01  1      065  22 INV     090  04  4      115  22 INV     140  99 PRT     165  22 INV     190  51 BST
016  25 CLR     041  42 STO     066  77  GE     091  87 IFF     116  59 INT     141  01  1      166  67  EQ     191  97 DSZ
017  02  2      042  89 89      067  00  00     092  00  00     117  67  EQ     142  44 SUM     167  00  00     192  89  89
018  91 R/S     043  01  1      068  70  70     093  00  00     118  01  01     143  87  87     168  47  47     193  01  01
019  03  3      044  08  8      069  22 INV     094  98  98     119  44  44     144  05  5      169  01  1      194  83  83
020  91 R/S     045  42 STO     070  86 STF     095  85  +      120  69 OP      145  32 X:T     170  08  8      195  61 GTO
021  09  9      046  87  87     071  00  00     096  04  4      121  30  30     146  02  2      171  32 X:T     196  00  00
022  69 OP      047  01  1      072  34 FX      097  07  7      122  97 DSZ     147  82 HIR     172  43 RCL     197  40  40
023  17  17     048  82 HIR     073  85  +      098  95  =      123  88  88     148  38  38     173  87  87
024  02  2      049  08  08     074  01  1      099  42 STO     124  01  01     149  82 HIR     174  67  EQ
```

Pgm 4. (Patrik Johansson.) Superprimtal. Inspelning: Mata in pgm och ladda R_{36}-R_{86} (registerlistning, se vänster-spalten på följande sida). Spela in i normaluppdelning.

Patrik Johansson har enligt denna metod gjort pgm 4, som på ungefär 3 h 55 min ger resultatutskriften i figur 3. Som synes finns det inga niosiffriga superprimtal.

Programmet körs genom att man läser in block 1, trycker A, läser in block 1 igen samt läser in resterande två block. Programmet startar automatiskt. På samma sätt körs Erik Stoys program, beskrivet nedan.

I Patriks program lagras alla superprimtal med ett visst antal siffror i R_1-R_{17}. Till vart och ett av dessa tal fogas slutsiffrorna 1, 3, 7 eller 9. De superprimtal som då upptäcks lagras i R_{18}-R_{34}. När alla m-siffriga superprimtal har hittats flyttas dessa från R_{18}-R_{34} till R_1-R_{17}, och sökningen efter (m+1)-siffriga superprimtal startas.

10.	36	2.	54	6.	72
2.	37	6.	55	6.	73
10.	38	4.	56	4.	74
2.	39	6.	57	2.	75
4.	40	8.	58	4.	76
2.	41	4.	59	6.	77
4.	42	2.	60	2.	78
6.	43	4.	61	6.	79
2.	44	2.	62	4.	80
6.	45	4.	63	2.	81
4.	46	8.	64	4.	82
2.	47	6.	65	2.	83
4.	48	4.	66	4.	84
6.	49	6.	67	4.	85
6.	50	2.	68	2.	86
2.	51	4.	69		
6.	52	6.	70		
4.	53	2.	71		

Pgm 4, registerlistning.

2.	593.		7393.		593933.
3.	599.		23333.		599993.
5.	719.		23339.		719333.
7.	733.		23399.		719391.
23.	739.		23993.		739393.
29.	797.		29399.		739397.
31.	2333.		31193.		739399.
37.	2339.		31379.		2339933.
53.	2393.		37337.		2393333.
59.	2399.		37339.		2939999.
71.	2939.		37397.		3733799.
73.	3119.		59393.		5939933.
79.	3137.		59399.		7393913.
233.	3733.		71933.		7393931.
239.	3739.		73331.		7393933.
293.	3793.		73939.		23399339.
311.	3797.		23993.		29399999.
313.	5939.		23993.		37337999.
317.	7193.		29399.		59393339.
373.	7331.		37379.		73939133.
379.	7333.		37393.		

Figur 3. Utskrift av pgm 4.

Erik Stoys pgm 5 ger på den något kortare tiden 3 h 38 min utskriften i figur 4. Jag har ändrat programmet något.

Eriks program håller bara reda på <u>ett</u> tal som modifieras efter hand. Om talet har konstaterats vara ett superprimtal ändras det genom tillfogning av slutsiffran 1. Om talet har konstaterats vara sammansatt ökas slutsiffran med 2. Om inte det är möjligt kapas sista siffran bort, och den nya sista siffran ökas så att den blir udda. Denna metod ger inte superprimtalen i ordning.

10.	01	2.	19	6.	37
2.	02	6.	20	6.	38
10.	03	4.	21	4.	39
2.	04	6.	22	2.	40
4.	05	8.	23	4.	41
2.	06	4.	24	6.	42
4.	07	2.	25	2.	43
6.	08	4.	26	6.	44
2.	09	2.	27	4.	45
6.	10	4.	28	2.	46
4.	11	8.	29	4.	47
2.	12	6.	30	2.	48
4.	13	4.	31	4.	49
6.	14	6.	32	2.	50
6.	15	2.	33	2.	51
2.	16	4.	34	1.	52
6.	17	6.	35		
4.	18	2.	36		

Pgm 5, registerlistning.

2.	2939999.		3739.		71933.
23.	2939999.		37397.		719391.
233.	29399999.		379.		73.
2333.	3.		3793.		733.
23333.	31.		3797.		7331.
23339.	311.		5.		7333.
2339.	3119.		53.		73391.
23399.	31193.		59.		739.
23993.	313.		593.		7393.
239933.	3137.		5939.		73939.
23999333.	31379.		59393.		739391.
239.	317.		593933.		7393913.
2393.	37.		5939333.		73939133.
2399.	373.		59393339.		739393.
23993.	3733.		59399.		7393931.
239993.	37337.		593993.		7393933.
2399333.	373379.		599.		739397.
29.	3733799.		7.		739399.
293.	37337999.		71.		79.
2939.	37339.		719.		797.
29399.	373393.		7193.		

Figur 4. Utskrift av pgm 5.

Programmet startas på samma sätt som Patriks program. Däremot stannar det inte automatiskt, utan när alla superprimtal är utskrivna skrivs 1 ut, följd av en massa primtal som börjar med 1. (Dessa är inte superprimtal eftersom 1 inte är ett superprimtal.)

Vid tidtagningen stoppade jag klockan när ettan skrevs ut, liksom jag stoppade klockan när Patriks program visade en nolla i displayen. För båda programmen tog det nämligen en viss tid innan de "fattade" att alla superprimtalen var utskrivna.

```
   01234567          0123456789
00 0123456        00  012345678
10 789ABCDE        10 789ABCDE78
20 -FGHIJKL       20 -FGHIJKLMN
30 MNOPQRST       30 MNOPQRSTMN
40 .UVWXYZ+       40 .UVWXYZ+X*
50 X*Fπe()×       50 X*Fπe()×X*
60 †%!/=*X$       60 †%!/=*X$%?
70 %?÷θ∏∫Σ        70 %?÷θ∏∫Σ?
                  80  012345678
                  90 789ABCDE78
```

Figur 5. Skrivkodstabeller.

000	91 R/S	015	43 RCL	030	42 STO	045	05 05
001	57 ENG	016	92 92	031	00 00	046	97 DSZ
002	69 OP	017	82 HIR	032	43 RCL	047	00 00
003	00 00	018	07 07	033	95 95	048	00 00
004	98 ADV	019	43 RCL	034	82 HIR	049	32 32
005	43 RCL	020	93 93	035	35 35	050	97 DSZ
006	90 90	021	82 HIR	036	43 RCL	051	01 01
007	82 HIR	022	06 06	037	94 94	052	00 00
008	06 06	023	43 RCL	038	82 HIR	053	02 02
009	43 RCL	024	96 96	039	36 36	054	22 INV
010	91 91	025	82 HIR	040	43 RCL	055	57 ENG
011	82 HIR	026	05 05	041	97 97	056	25 CLR
012	07 07	027	69 OP	042	82 HIR	057	81 RST
013	69 OP	028	05 05	043	37 37		
014	05 05	029	7 7	044	69 OP		

```
.1110102030405      90
 0.111060708        91
 0.111050607        92
.111000102030304    93
.000101010101       94
 0.000000001        95
.11100000101        96
 0.00010101         97
44444.00000002      98
```

Pgm 6. (Per Eriksson.) "Liten" skrivkodstabell. Inspelning: Mata in pgm och ladda R_{90}-R_{98}. Spela in block 1 i normaluppdelning.

▷000	91 R/S	020	00 00	040	02 02	060	43 RCL
▷001	03 3	021	00 00	041	35 35	061	99 99
002	65 ×	022	06 06	042	42 STO	062	82 HIR
003	22 INV	023	07 IFF	043	00 00	063	07 07
▷004	86 STF	024	00 00	044	01 1	064	69 OP
005	00 00	025	00 00	045	00 0	065	05 05
▷006	43 RCL	026	01 01	046	98 ADV	066	43 RCL
007	95 95	027	06 6	047	69 OP	067	93 93
008	95 =	028	69 OP	048	17 17	068	82 HIR
009	82 HIR	029	17 17	049	22 INV	069	07 07
010	36 36	030	25 CLR	050	58 FIX	070	43 RCL
011	69 OP	031	98 ADV	051	09 9	071	94 94
012	05 05	032	98 ADV	052	42 STO	072	82 HIR
013	43 RCL	033	98 ADV	053	00 00	073	06 06
014	96 96	034	81 RST	054	69 OP	074	43 RCL
015	82 HIR	035	76 LBL	055	00 00	075	97 97
016	37 37	036	11 A	056	43 RCL	076	82 HIR
017	82 HIR	037	36 PGM	057	98 98	077	08 08
018	38 38	038	02 02	058	82 HIR	078	61 GTO
019	97 DSZ	039	71 SBR	059	08 08	079	00 00
						080	04 04

```
1000001020304.      93
1000000000100.      94
        10000.      95
 1010101010.        96
1000506070809.      97
1000607080912.      98
1000102030405.      99
```

Pgm 7. (Niklas Ramström.) "Stor" skrivkodstabell. Inspelning: Mata in pgm och ladda R_{93}-R_{99}. Spela in block 1 i normaluppdelning.

SKRIVKODSTABELLER (17)

Till problem 17, som gällde att skriva ut tabellerna i figur 5, har lösningar inkommit från Per Eriksson, Niklas Ramström, Dan Altemyr, Lars Winter, Olof Magnusson och Lars Hedlund. Alla lösningar kan inte återges här, så jag har valt ut tre.

Den snabbaste är pgm 6 av Per Eriksson. På 44 sekunder skriver det ut 10 "små" tabeller. Det körs så här: Läs in programmet, mata in antal tabeller och tryck A. Tryck 7 EE när ett blinkande tal visas. Utskriften startar.

Programmet använder St flg Ind 7 INV-sekvensen, som beskrevs i PB 81-3 s 29-32, för att komma in i fast mode. Personligen tycker jag att den sekvensen är mer lämplig för TI-58, som jag antyder i "Nyheter på TI-58C" någon annanstans i det här numret.

I Pers program laddas skrivkoderna för nästa rad i utskriften genom direkt summering till skrivregistren med hjälp av HIR-instruktionen. Detta gör programmet mycket snabbt. Alla skrivkoder hämtas från R_{90}-R_{98}, som ju ligger i samma block som programmet.

Niklas Ramström, Uppsala, har gjort pgm 7 som skriver ut den "stora" tabellen. Det körs i fast mode, och startas på sedvanligt sätt: Läs in block 1, tryck A, läs in block 1 igen.

Slutligen, Dan Altemyr, Göteborg, har gjort pgm 8 som skriver ut <u>båda</u> tabellerna. Tryck A för "liten" tabell och B för "stor" tabell.

000	76 LBL	040	05 5	080	01 01	120	00 0
001	12 B	041	95 =	081	00 00	121	00 0
002	25 CLR	042	69 OP	082	75 -	122	00 0
003	02 2	043	02 02	083	01 1	123	00 0
004	42 STO	044	05 5	084	00 0	124	95 =
005	02 02	045	00 0	085	01 1	125	69 OP
006	61 GTO	046	06 6	086	00 0	126	01 01
007	00 00	047	00 0	087	22 INV	127	43 RCL
008	15 15	048	77 GE	088	44 SUM	128	05 05
009	76 LBL	049	00 0	089	06 06	129	69 OP
010	11 A	050	08 8	090	08 8	130	02 02
011	25 CLR	051	00 0	091	00 0	131	43 RCL
012	42 STO	052	09 9	092	09 9	132	04 04
013	02 02	053	42 STO	093	22 INV	133	44 SUM
014	22 INV	054	07 07	094	44 SUM	134	05 05
015	86 STF	055	01 1	095	01 1	135	43 RCL
016	01 01	056	00 0	096	09 9	136	07 07
017	25 CLR	057	01 1	097	01 1	137	69 OP
018	42 STO	058	00 0	098	02 2	138	03 03
019	03 03	059	01 1	099	95 =	139	43 RCL
020	98 ADV	060	00 0	100	69 OP	140	06 06
021	69 OP	061	01 1	101	03 03	141	44 SUM
022	00 00	062	00 0	102	69 OP	142	07 07
023	08 8	063	01 1	103	05 05	143	69 OP
024	44 SUM	064	00 0	104	69 OP	144	05 05
025	02 02	065	42 STO	105	42 STO	145	97 DSZ
026	01 1	066	04 04	106	43 RCL	146	02 02
027	00 0	067	42 STO	107	03 03	147	01 01
028	02 2	068	06 06	108	85 +	148	04 04
029	00 0	069	06 6	109	28 LOG	149	98 ADV
030	03 3	070	00 0	110	59 INT	150	25 CLR
031	00 0	071	07 7	111	65 ×	151	92 RTN
032	04 4	072	00 0	112	02 2		
033	42 STO	073	08 8	113	85 +		
034	05 05	074	00 0	114	93 .		
035	65 ×	075	09 9	115	00 0		
036	01 1	076	01 1	116	01 1		
037	00 0	077	02 2	117	95 =		
038	00 0	078	87 IFF	118	65 ×		
039	85 +	079	01 01	119	01 1		

Pgm 8. (Dan Altemyr.) Båda skrivkodstabellerna.

Dan skriver att

> Programmet tar 25 resp 30 s i anspråk för utskrift, vilket naturligtvis går att korta ned om man gör sig omaket att skriva om programmet till "fast mode", men det tycker jag sker på bekostnad av bekvämligheten vid användning. Man bör nog inte bara räkna in exekveringstiden i den totala användningstiden för ett så här kort program. Det tar ju trots allt några sekunder att klabba med magnetkort fram och tillbaka, att leta upp nätsladden till skrivaren, ansluta kalkylatorn osv. Det är tyvärr ett allmänt intryck, tycker jag i alla fall, att många program blir onödigt otympliga och krångliga att använda, bara för att snabba upp ett program som ändå tar lång tid på sig. Nog om detta.
>
> Utskriften från programmet överenstämmer helt med problemtexten, men jag anser att TI:s egen version, se t.ex. instruktionsboken, är mer estetiskt tilltalande. [*Här har en programbeskrivning utesluts.*] En sådan lista har jag haft fasttejpad på skrivaren ända sen den köptes. Det är rätt praktiskt, man kan hålla mer väsentliga saker i tankarna än alla dumma skrivarkoder som inte är numrerade på något speciellt intelligent sätt.

Olof Magnusson, Uppsala, har också skickat en annan sorts skrivkodstabell. Detta gav mig idén till problem 21 i slutet av Utmaningen.

PRIMTALSTEST (3 & 18)

I Utmaningen i PB 81-4 beskrev jag en metod för att bevisa att ett tal är primtal. Metoden hade nackdelen att n-1 måste faktoriseras. Om man nöjer sig med att vara nästan säker på att ett tal är ett primtal, kan följande enkla algoritm (ur The Art of Computer Programming, Vol 2, av D.E. Knuth) användas:

Algoritm P (Sannolikhets bestämt primtalstest). Algoritmen försöker ta reda på om ett udda heltal n är ett primtal. Låt $n = 1+2^k q$, där q är udda.

P1. [Alstra x.] Låt x vara ett slumpmässigt valt heltal i intervallet $1<x<n$.
P2. [Exponentiera.] Sätt $j \leftarrow 0$ och $y \leftarrow x^q$ mod n.
P3. [Färdigt?] Om $j = 0$ och $y = 1$, eller om $y = n-1$, avbryt algoritmen och säg att "n är troligen ett primtal". Om $j>0$ och $y = 1$, gå till steg P5.
P4. [Öka j.] Öka j med 1. Om $j<k$, sätt $y \leftarrow y^2$ mod n och gå tillbaka till steg P3.
P5. [Ej primtal.] Avbryt algoritmen och säg att "n är definitivt inte ett primtal".

Det går att bevisa att algoritmen kommer att ta fel högst 1/4 av alla gånger, för alla n. Antag att vi upprepar algoritmen ett antal gånger, och ser till att x väljs slumpmässigt varje gång. Om algoritmen svarar en enda gång att n inte är ett primtal, kan vi vara absolut säkra på att n är sammansatt.

Om å andra sidan algoritmen 25 gånger i rad meddelar att n troligen är ett primtal, kan vi vara nästan säkra på att n är ett primtal. Sannolikheten för att algoritmen har tagit fel 25 gånger i rad är $(1/4)^{25} = 8.9 \cdot 10^{-16}$ eller ungefär en chans på 10^{15}. Knuth påpekar i sin bok: "Det är mycket mer troligt att vår dator har tappat en bit i sina beräkningar, på grund av hårdvarufel eller kosmisk strålning, än att algoritm P har gissat fel gång på gång!"

För den intresserade skall den grundläggande idén bakom algoritm P kort nämnas. Om $n = 1+2^k q$ är ett primtal och x^q mod $n \neq 1$, kommer sekvensen av värden

x^q mod n, x^{2q} mod n, x^{4q} mod n,..., $x^{2^k q}$ mod n

att sluta med 1 (detta följer av Fermats sats eftersom den sista exponenten är n-1), och värdet alldeles före 1 kommer att vara n-1. Detta följer av att de enda lösningarna till $y^2 \equiv 1$ (mod p) är $y \equiv 1$ och $y \equiv p-1$.

Jag har gjort ett program för algoritm P, där pgm 8 från Utmaningen 81-4 ingår. Tyvärr klarar inte pgm 8 särskilt stora tal. Gösta Blume skickade då ett program som han för två år sedan fick som lösning till en "utmaning" i tidskriften Elementa. Programmet är gjort av Patrik Delin och Mikael Rittri, båda från Göteborg, och det beräknar a^x mod n för ända upp till 12-siffriga n!

Problemet med en sådan beräkning består i att produkten modulo n av två tal, som kan vara så stora som n-1, måste kunna beräknas exakt. Om n då är 10-siffrig kan produkten bli 20-siffrig. Patrik och Mikael har löst problemet genom att utföra multiplikationen i flera steg, där mellanresultaten aldrig överskrider räknarens 13-siffriga kapacitet.

Jag har satt in deras subrutin för multiplikationen i pgm 8 från Utmaningen 81-4. Resultatet blev pgm 9. Subrutinen omfattar steg 000-057. I steg 067-077 beräknas en konstant - som används i subrutinen - så att inga mellanresultat får fler än 12 siffror. Jag litar nämligen inte på att räknaren räknar helt riktigt med 13 siffror. I den belgiska TISoft tidskriften fanns en artikel, som talade om att den trettonde siffran i vissa fall kan bli felaktig vid multiplikation.

Denna ändring får till följd att programmet "endast" klarar maximalt 11-siffriga n. (Steg 071 kan ändras till "2" på egen risk, för att kunna använda 12-siffriga n.)

Bruksanvisning för att beräkna a^x mod n med pgm 9: Mata in a, tryck A. Mata in n, tryck B. Mata in x, tryck C. Minsta positiva rest visas.

Se problem 23 i slutet av den här avdelningen.

```
000  76 LBL     033  85  +      066  00  00      099  05  05
001  16 A'      034  43 RCL     067  28 LOG      100  42 STO
002  65  ×      035  03  03     068  59 INT      101  02  02
003  43 RCL     036  75  -      069  75  -       102  43 RCL
004  04  04     037  53  (      070  01  1       103  07  07
005  75  -      038  29 CP      071  01  1       104  16 A'
006  59 INT     039  55  ÷      072  95  =       105  42 STO
007  42 STO     040  43 RCL     073  22 INV      106  07  07
008  01  01     041  00  00     074  28 LOG      107  43 RCL
009  95  =      042  54  )      075  52 EE       108  05  05
010  65  ×      043  59 INT     076  42 STO      109  42 STO
011  53  (      044  65  ×      077  04  04      110  02  02
012  53  (      045  43 RCL     078  01  1       111  16 A'
013  43 RCL     046  00  00     079  42 STO      112  42 STO
014  02  02     047  95  =      080  07  07      113  05  05
015  75  -      048  42 STO     081  25 CLR      114  02  2
016  53  (      049  03  03     082  42 STO      115  32 X:T
017  24 CE      050  43 RCL     083  03  03      116  43 RCL
018  55  ÷      051  01  01     084  92 RTN      117  06  06
019  43 RCL     052  22 INV     085  76 LBL      118  77  GE
020  00  00     053  67  EQ     086  13  C       119  00  00
021  54  )      054  16 A'      087  55  ÷       120  87  87
022  59 INT     055  48 EXC     088  02  2       121  43 RCL
023  65  ×      056  03  03     089  75  -       122  05  05
024  43 RCL     057  92 RTN     090  59 INT      123  42 STO
025  00  00     058  76 LBL     091  42 STO      124  02  02
026  54  )      059  11  A      092  06  06      125  43 RCL
027  55  ÷      060  42 STO     093  95  =       126  07  07
028  43 RCL     061  05  05     094  29 CP       127  61 GTO
029  04  04     062  92 RTN     095  67  EQ      128  16 A'
030  54  )      063  76 LBL     096  01  01
031  42 STO     064  12  B      097  07  07
032  02  02     065  42 STO     098  43 RCL
```

Pgm 9. (Red.) Beräkning av a^x mod n för stora n.

1/A + 1/B = 1/C (14)

Problem 14 gällde att göra ett program som ger alla lösningar till ekvationen $1/a + 1/b = 1/c$, där a, b, c är positiva heltal. Två lösningar har inkommit. Den första fick jag från Markús Markússon (som jag av utrymmesskäl inte tog med i Utmaningen 81-4).

Markús har härlett den allmänna lösningen:

$$k \cdot (I,\ I(I-1),\ I-1),\quad I = 2,\ 3,\ 4,\ldots$$

där k är ett heltal större än noll. Om man t ex $I = 2$ och $k = 1$ erhålls lösningen (2,2,1) eller $a = 2$, $b = 2$, $c = 1$. Om $I = 1$ och $k = 2$ erhålls lösningen (4,4,2).

Man kan naturligtvis kasta om värdena på a och b.

Markús' program ger en mycket elegant utskrift med a, b och c på samma rad, men det ger inte alla lösningar.

Gösta Blume har gjort ett program som teoretiskt sett kan ge alla lösningar. Han har också skrivit en utmärkt redogörelse över den bakomliggande teorin. Här återges hela redogörelsen:

> I Utmaningen, PB 81-3, efterlyses "ett pgm som ger alla lösningar till 1/a + 1/b = 1/c, där a, b och c är positiva heltal". För varje givet värde på c har likheten ett begränsat antal lösningar i form av talpar (a,b). Men eftersom det i uppgiften inte förutsätts, att någon av de tre parametrarna, inte ens c, har ett fast värde, så blir antalet lösningar uppenbarligen oändligt stort.
>
> För att åtminstone börja beta av lösningsmängden behandlar man lämpligen ett c-värde i taget. Man kan då för ett givet c och successiva värden på a söka de $b = ac/(a-c)$ som är heltal.
>
> Den första frågan blir då, inom vilket område a kan variera. Det är uppenbart, att det minsta a-värde som kan tänkas är $a = c+1$, varvid b generellt blir $b = c(c+1)$. Av symetriskäl blir då det största a-värdet $= c(c+1)$ och motsvarande $b = c+1$.
>
> Det område inom vilket a kan variera är sålunda $(c+1)$ till $c(c+1)$. Den nämnda symmetrin gör emellertid att vi kan nöja oss med att avsöka endast halva detta område. För $c = 6$ gäller t ex
>
> | | | | | | | | | | |
> |---|---|---|---|---|---|---|---|---|---|
> | a = | 7 | 8 | 9 | 10 | 12 | 15 | 18 | 24 | 42 |
> | b = | 42 | 24 | 18 | 15 | 12 | 10 | 9 | 8 | 7 |
>
> Har vi sålunda fått a- och b-värdena för området från $a = c+1$ till $a (= b) = 2c$, så har vi tillgång till alla sökta (a,b)-par.
>
> Eftersom det torde vara svårt, kanske omöjligt men säkerligen tidsödande att direkt ur c beräkna de värden på a som gör b till ett helt tal eller, vilket är samma sak, som gör att bråket $ac/(a-c)$ kan förkortas med (a-c), ser det ut som om vi skulle bli tvungna att testa vart och ett av alla de a-värden som faller inom det fastställda sökområdet. Men fullt så illa är det inte. Det finns nämligen åtminstone ett fall, där vi direkt och generellt kan se, att den nämnda förkortningen inte kan göras: antag att både a och c är udda. Då blir täljaren, ac, udda medan nämnaren, (a-c), blir jämn, dvs innehåller en eller flera faktorer 2 som inte kan förkortas bort. Vi kan därför generellt säga, att om c är udda behöver vi endast pröva jämna värden på a. Någon motsvarande regel finns inte för det fallet att c är jämn.

För styrning av pgm kan man antingen testa varje a mot 2c och gå ur loopen när detta värde nåtts, eller kan man för varje nytt c beräkna antalet testvarv och låta Dsz sköta övervakningen. Troligen är Dsz-alternativet något snabbare.

Antag då till att börja med att c är udda. Både (c+1), som vi skall starta med, och 2c, där vi skall stanna, är då jämn. Eftersom c är udda, skall endast jämna a testas. Antalet testvarv blir då (2c-(c+1))/2+1 = (c+1)/2, och steglängden är 2. Om c däremot är jämnt, blir antalet steg (2c-(c+1))+1 = c, och steglängden är 1.

I pgm [*pgm 10*] finns a, b och c i resp R_1, R_2 och R_3. R_4 innehåller steglängden och R_0 är Dsz-reg som räknar ner testvarven 000 - 039, varav själva heltalstesten ligger i 000 - 019. 030 - 036 ger utskrift av a och b. 040 - 044 initierar start för nästa c-värde, vilken start sedan sker automatiskt.

Pgm startas genom inslagning av ett första c-värde på A. I 045 - 064 testas huruvida c är udda eller jämn. På vägen till den egentliga testen lagras c i Dsz-reg (R_0), steglängden 1 i R_4 och första a-värdet (c+1) i R_1, alltså alla de data som gäller för fallet "jämn c". Visar sedan testen att c verkligen är jämn, går körningen direkt upp till pgm första del, test för heltals-b. - Skulle däremot testen i 056....visa, att c är udda korrigeras (065 - 073) dels varvsiffran (i R_0) till (c+1)/2, dels steglängden i R_4 till 2.

40 à 45 a-värden behandlas per minut.

Det kan vara värt att påpeka att TISoft, som problemet hämtades ifrån, inte hade kommit på att endast jämna a-värden behöver provas om c är udda.

000	43	RCL	021	97	DSZ	042	43	RCL	063	00	00
001	01	01	022	00	00	043	03	03	064	00	00
002	45	x	023	00	00	044	98	ADV	065	43	RCL
003	43	RCL	024	40	40	045	99	PRT	066	01	01
004	03	03	025	43	RCL	046	98	ADV	067	55	÷
005	55	÷	026	04	04	047	42	STO	068	02	2
006	53	(	027	44	SUM	048	00	00	069	42	STO
007	43	RCL	028	01	01	049	85	+	070	04	04
008	01	01	029	81	RST	050	01	1	071	95	=
009	75	-	030	43	RCL	051	42	STO	072	42	STO
010	43	RCL	031	01	01	052	04	04	073	00	00
011	03	03	032	99	PRT	053	54	)	074	81	RST
012	95	=	033	43	RCL	054	42	STO	075	76	LBL
013	42	STO	034	02	02	055	01	01	076	11	A
014	02	02	035	99	PRT	056	55	÷	077	29	CP
015	22	INV	036	98	ADV	057	02	2	078	42	STO
016	59	INT	037	61	GTO	058	95	=	079	03	03
017	67	EQ	038	00	00	059	22	INV	080	61	GTO
018	00	00	039	20	20	060	59	INT	081	00	00
019	30	30	040	69	OP	061	22	INV	082	44	44
020	22	INV	041	23	23	062	67	EQ			

Pgm 10. (Göata Blume.) Löeningar till 1/a + 1/b = 1/c.

MR MAGIC (19)

Problemet om Mr Magic stod i PB 81-4 s 36 i en artikel om ett Mr Magic-program. Det är formellt upptaget som ett problem i slutet av den här Utmaningen.

Olof Magnusson, Uppsala, har skickat två versioner. Båda använder knepet med fördröjt modulanrop (se Komplement till instruktionsboken s 7), vilket gör att R/S inte behöver följas av någon ytterligare tangent.

Eftersom Lars Hedlund föreslog problemet har jag låtit honom titta på Olofs program. Lars gjorde då ett eget program. Både Olofs ursprungliga program och Lars' program är bra gjorda, men eftersom jag tycker att Lars' program är något bättre för "praktiskt trolleri" har jag valt att ta med Lars' program.

Lars skriver så här om sitt program:

Jag har tittat på Olof Magnussons program och använt samma modulsubrutin som han, dvs ML-13 SBR 044 (D) som ställer flaggorna 0 och 1. Olof förklarade sitt program efter årsmötet och föreslog att man t ex kunde mata in godtycklig skrivkod på det här sättet, dvs att t ex få skrivaren att skriva namnet på en person som gått ut ur rummet.

Mitt program [*pgm II*] arbetar för övrigt med de idéer jag fick när vi publicerade Mr. Magic i förra numret. Det startas med A. 9999 (utan punkt) blinkar en gång och därefter 9999. (med punkt) fyra gånger. R/S under en av blinkarna med punkt ger först utskrift av färg (klöver, ruter, hjärter, spader), därefter hög eller låg (2-5 "hacka", 6-9 "hacka", 10-13(K) "honnör" och som sista alternativ "ess") och till sist exakt valör (2,3,4,5 resp. 6,7,8,9 eller 10,Kn, D,K). Korrigering av feltryckning görs med R/S under blink utan punkt ("Fel, jag gissar igen!).

Steg 000-026 innehåller "Ta ett kort!" och dszladdning för tre svarsomgångar. 027-037 förbereder modulanropet (ett R/S från tangentbordet när som helst härefter medför att pgm 13 SBR 044 anropas och flagga 0 ställs varefter återhopp görs till huvudprogrammet där uthoppet skedde). 038-047 är blinkning utan punkt - om R/S tryckts ger flaggtestet hopp till 066 där dsz-registret ökas och "Fel, jag gissar igen!" skrives. 048-062 är fyra blinkningar med punkt - flaggtestet ger här ev. hopp till 092 där RCL 01 anger vilken svarsomgång det är. Första omgången ("3") är bestämning av färg i 103-118. Andra omgången ("2") är 137-153-105-118 som anger "hacka", "honnör" eller "ess". Om det var ett ess behövs ingen tredje omgång (137-146). För övriga valörer lagras RCL 00 i R03. Tredje omgången ("1") är 126-136-105-118 och där sammansätts uppgifterna från andra och tredje omgången till rätt valör.

000	76	LBL	040	09	9	080	43	RCL	120	98	ADV
001	11	A	041	09	9	081	33	33	121	98	ADV
002	98	ADV	042	66	PAU	082	69	OP	122	98	ADV
003	43	RCL	043	66	PAU	083	04	04	123	91	R/S
004	54	54	044	87	IFF	084	69	OP	124	61	GTO
005	69	OP	045	00	00	085	05	05	125	11	A
006	01	01	046	00	00	086	69	OP	126	02	2
007	43	RCL	047	66	66	087	00	00	127	05	5
008	55	55	048	32	XIT	088	98	ADV	128	85	+
009	69	OP	049	04	4	089	61	GTO	129	04	4
010	02	02	050	42	STO	090	00	00	130	65	×
011	43	RCL	051	00	00	091	27	27	131	43	RCL
012	56	56	052	32	XIT	092	43	RCL	132	03	03
013	69	OP	053	66	PAU	093	01	01	133	95	=
014	03	03	054	66	PAU	094	32	XIT	134	61	GTO
015	43	RCL	055	87	IFF	095	01	1	135	01	01
016	57	57	056	00	00	096	67	EQ	136	05	05
017	69	OP	057	00	00	097	01	01	137	01	1
018	04	04	058	92	92	098	26	26	138	32	XIT
019	69	OP	059	97	DSZ	099	02	2	139	43	RCL
020	05	05	060	00	00	100	67	EQ	140	00	00
021	69	OP	061	00	00	101	01	01	141	22	INV
022	00	00	062	53	53	102	37	37	142	67	EQ
023	98	ADV	063	61	GTO	103	04	4	143	01	01
024	03	3	064	00	00	104	09	9	144	47	47
025	42	STO	065	38	38	105	44	SUM	145	69	OP
026	01	01	066	69	OP	106	00	00	146	31	31
027	22	INV	067	21	21	107	73	RC*	147	42	STO
028	86	STF	068	43	RCL	108	00	00	148	03	03
029	00	00	069	30	30	109	69	OP	149	04	4
030	36	PGM	070	69	OP	110	02	02	150	05	5
031	13	13	071	01	01	111	69	OP	151	61	GTO
032	71	SBR	072	43	RCL	112	05	05	152	01	01
033	00	00	073	31	31	113	69	OP	153	05	05
034	42	43	074	69	OP	114	00	00			
035	36	PGM	075	02	02	115	97	DSZ			
036	13	13	076	43	RCL	116	01	01			
037	51	EST	077	32	32	117	00	00			
038	09	9	078	69	OP	118	27	27			
039	09	9	079	03	03	119	25	CLR			

Pgm 11. (Lars Hedlund.) Mr Magic.

PROBLEM

Eftersom det är nytt år kommer här en lista över alla gamla problem, sammanställd av Jan Gustafsson (skulle egentligen ingå i registret men togs bort av utrymmesskäl). Däremot skulle det ta upp för mycket plats att ta med problemformuleringarna. Dessa finns emellertid samlade i Utmaningen 81-2.

Därefter följer nya problem.

1. EXPONENTEN
80:2, s. 5. 80:3, s. 4. xxx Matematik

2. DUBBEL PRECISION
80:2, s. 5. 81:1, s. 6. 81:3, s. 5.

3. PRIMTAL
80:2, s. 5. 80:3, s. 5-7. 80:4, s. 4-5. 81:1, s.6-7. 81:2, s. 4. 81:3, s. 6. 81:4, s. 7-8. xxx Matematik

4. KONSTANTEN e
80:2, s. 5. 80:3, s. 4-5. 80:4, s. 5-7. 81:1, s.8. xxx Matematik

5. SKRIVARKOD
80:3, s. 6. 81:3, s. 4-5. x) xxx Textbehandling

6. HIR-STACKEN
80:3, s. 6. 81:1, s. 8. xxx HIR

7. PERMUTATIONER OCH KOMBINATIONER
80:3, s. 6-7. 81:1, s. 8-9.

8. EKVATIONSSYSTEM
80:4, s. 7. 81:1, s. 9-10. 81:2, s. 4-6. xxx Matematik

9. EKVATIONEN f(x) = 0
81:2, s. 7. xxx Matematik

10. LABYRINTEN
81:2, s. 7-8. 81:3, s. 4. 81:4, s. 6-7 och 8. xxx Spel

11. DSE-FUNKTION
81:3, s. 6. 81:4, s. 4. xxx Programmering

12. LUFFARSCHACK PÅ STOR SPELPLAN
81:3, s. 6. xxx Spel

13. SORTERING
81:3, s. 6. 81:4, s. 4-5. xxx Sortering

14. 1/a + 1/b = 1/c
81:3, s. 7. xxx Matematik

15. $1^1 + 2^2 + 3^3 + \ldots + 20^{20}$
81:3, s. 7. 81:4, s.5. xxx Programmering

16. SUPERPRIMTAL
81:4, s. 9. xxx Matematik

17. SKRIVKODSTABELL
81:4, s. 9 och 8. xxx Textbehandling

18. a^x mod n I DUBBEL PRECISION
81:4, s. 9.xxx Matematik

19. Mr Magic.

Gör ett program, som liksom programmet "Mr Magic" i PB 81-4 s 36-37 "gissar" spelkort, men med val av fyra möjligheter vid varje R/S. Programmet klarar då rätt färg på en gång, och valören i två gissningar. (Lars Hedlund.)

20. AOS på HP-41C.

Gör ett program för HP-41C, så att man kan räkna med AOS som på TI-59. (Björn Nordell.)

21. Praktisk skrivkodstabell.

Konstruera en praktisk skrivkodstabell, där det är enkelt att hitta skrivkoden för ett speciellt tecken. Tabellerna får se ut hur som helst. Det är inte heller nödvändigt att ha ett program som skriver ut tabellen; tabellen kan lika gärna skrivas ut på skrivmaskin. Det viktiga är hur praktisk användbar tabellen är. (Red.)

22. Patienseläggare.

Gör ett program som skriver utläggningen av patiensen "7x7", såväl från start som efter omflyttning och efter hopsamling och upprepad utläggning.

Med kort lägges patiensen så här. Alla kort läggs med bildsidan upp i 7 rader med 7 kolumner. Resterande tre kort läggs vid sidan och kallas "handkort". Endast det nedersta (främsta) kortet i varje kolumn samt handkorten är tillgängliga för flyttning. Ess får läggas upp som start för särskilda högar och i övrigt får ett kort endast läggas på ett annat av samma färg av närmast högre eller lägre valör. Hela sviter får ej flyttas. "Handen" får ha högst tre kort.

Då inget kort mer kan flyttas samlas handkorten med bildsidan upp, därefter första kolumnens kort sammanskjutna ovanpå dessa (det "flyttbara" kortet överst) och sedan övriga kolumner. "Esshögarna" ligger kvar. Därefter läggs de hopsamlade korten ut på vanligt sätt utan att blandas, dvs från en hög med baksidan upp och en rad i taget (med bildsidan upp som förut). De tre sista blir "handkort". Om man vill sätta en gräns för att patiensen "går ut" kan man göra det efter tre utläggningar - annars går det att hålla på tills den är slut.

Skrivaren kan ju skriva exakt sju "kolumner" med två tecken + mellanrum mellan kolumnerna. Det är tillåtet att vända på utskriften så att man arbetar med den översta raden i st f den understa. (Lars Hedlund.)

23. Primtalstest.

Skriv med hjälp av pgm 9 [Utmaningen 82-1] ett program för primtalstest enligt algoritm P [Utmaningen 82-1]. (Red.)

24. Trafikräkning.

Gör ett programsystem som kan användas för att beräkna hur många bilar som passerar ett område. Det skall användas så här: En person står vid infarten av området och matar in bilnumren (endast siffrorna i bilnumret, ej bokstäverna) på alla bilar som åker in i området i en TI-59. En annan person står vid utfarten och matar in bilnumren på utfarande bilar. Med hjälp av ett program skall man sedan kunna få reda på hur många av de infarande bilarna som körde ut igen.

Mer än 100 bilnummer bör klaras av. Genom packning av minnen kan man anteckna avsevärt flera. Denna metod som kan visa numren i tidsordning är intressantare än att t ex anteckna förekomst eller frekvens i t ex 10 sifferplatser i ett antal minnen. (Hans Hellström.)

25. Registerblockhantering.

Gör subrutiner som klarar att

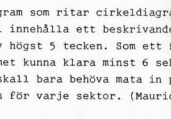
a) kopiera innehållet i ett registerblock till ett annat registerblock,

b) skifta innehåll mellan två registerblock.

Med registerblock menas alla register från R_i till R_j (i<j). Subrutinerna bör vara så praktiskt användbara som möjligt. (Hans Hellström.)

26. Cirkeldiagram.

Gör ett program som ritar cirkeldiagram. Varje sektor skall innehålla ett beskrivande namn, bestående av högst 5 tecken. Som ett förslag bör programmet kunna klara minst 6 sektorer. Användaren skall bara behöva mata in procental och namn för varje sektor. (Maurice E.T. Swinnen.)



[Kommentar: Maurice föreslog problemet i senaste numret av TI PPC Notes (v7n3p3), eftersom flera medlemmar hade frågat om det går att göra cirkeldiagram med TI-59/PC100. Han avslutade: "Our friends from Programbiten will call this one an 'Utmaning.' And it is one that counts!"]

På återhörande

Björn Gustavsson ■

Snabbt kalenderprogram

av Björn Gustavsson

I PB 80-4 nämndes kalendertävlingen som pågick i USA mellan TI-59 och HP-41C. Segern togs hem av RPN-programmerarna. Men Palmer O. Hanson lyckades i alla fall få ner tiden för TI-59 till under en och halv minut. Här presenteras detta program. Jag har ändrat det något för att få "svensk" utskrift; i USA är söndag första veckodag, medan måndag är första veckodag i Sverige.

Inmatning: Slå av och på räknaren. Mata in steg 000-477 enligt listning. Spela in block 1 och 2 i normaluppdelning. Ladda R_8-R_{59} enligt registerlistning. Spela in block 3 och 4.

Bruksanvisning:

1. Se till att standardmodulen är isatt.
2. Läs in block 1. Tryck A.
3. Läs in block 1 igen. 2 visas.
4. Läs in block 2. 3 visas.
5. Läs in block 3. 4 visas.
6. Läs in block 4. Skriver skriver "YEAR?".
7. Mata in år och tryck R/S.
8. Önskas även efterföljande år, tryck R/S igen.

```
YEAR?

      1982.                JAN
M  T  D  T  F  L  S
                 1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

      1982.                FEB
M  T  D  T  F  L  S
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28

      1982.                MAR
M  T  D  T  F  L  S
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

31.00000251331	08	2.01000700001	34
28.00000211714	09	2.010010000011	35
31.00000301335	10	2.010011000012	36
30.00000133335	11	2.010012000201	37
31.00000301325	12	2.010201000202	38
30.00000254131	13	2.010202000203	39
31.00000254127	14	2.010203000204	40
30.00000134122	15	2.010204000205	41
30.00000361733	16	2.010205000206	42
31.00000322637	17	2.010206000207	43
30.00000313242	18	2.01020700021	44
31.00000161715	19	2.010210000211	45
28.00000211714	20	2.010211000212	46
0.	21	2.010212000301	47
98.99	22	2.010301000302	48
0.	23	2.010302000303	49
1.000030000037	24	2.010303000304	50
1.00000032	25	2.010304000305	51
1.003700001	26	2.010305000306	52
1.000027000036	27	2.010306000307	53
2.010000000002	28	2.01030700031	54
2.010002000003	29	2.010310000311	55
2.010003000004	30	2.010311000312	56
2.010004000005	31	2.010312000401	57
2.010005000006	32	2.010401000402	58
2.010006000007	33	2.010402	59

000	00	0	060	42	STO	120	59	INT	180	29	CP	240	82	HIR	300	61	GTO	360	07	07	420	24	24
001	*00	0	061	05	05	121	94	+/-	181	22	INV	241	07	07	301	03	03	361	85		421	00	0
002	00	0	062	59	INT	122	85	+	182	67	EQ	242	73	RC*	302	32	32	362	43	RCL	422	67	EQ
003	76	LBL	063	42	STO	123	53	(	183	03	03	243	00	00	303	73	RC*	363	02	02	423	02	02
004	11	A	064	03	03	124	43	RCL	184	52	52	244	82	HIR	304	00	00	364	75	-	424	18	18
005	36	PGM	065	42	STO	125	06	06	185	49	DP	245	08	08	305	22	INV	365	53	(	425	49	DP
006	02	02	066	06	06	126	55	÷	186	29	29	246	69	OP	306	44	SUM	366	24	CE	426	30	30
007	71	SBR	067	22	INV	127	04	4	187	43	RCL	247	05	05	307	00	00	367	55	÷	427	06	6
008	02	02	068	44	SUM	128	54	)	188	04	04	248	97	DSZ	308	52	EE	368	07	7	428	67	EQ
009	39	39	069	05	05	129	59	INT	189	55	÷	249	04	04	309	82	HIR	369	54	)	429	02	02
010	09	9	070	01	1	130	75	-	190	02	2	250	02	02	310	08	08	370	59	INT	430	42	42
011	00	0	071	00	0	131	53	(	191	05	5	251	16	16	311	73	RC*	371	42	STO	431	04	4
012	22	INV	072	49	PRD	132	93	.	192	95	=	252	43	RCL	312	00	00	372	04	04	432	67	EQ
013	58	FIX	073	05	05	133	07	7	193	42	STO	253	02	02	313	22	INV	373	65	×	433	02	02
014	22	INV	074	68	NOP	134	05	5	194	04	04	254	29	CP	314	59	INT	374	07	7	434	34	34
015	57	ENG	075	68	NOP	135	85	+	195	22	INV	255	67	EQ	315	82	HIR	375	95	=	435	61	GTO
016	02	2	076	68	NOP	136	53	(	196	59	INT	256	03	03	316	07	07	376	32	X:T	436	02	02
017	91	R/S	077	53	(	137	43	RCL	197	22	INV	257	37	37	317	69	OP	377	73	RC*	437	18	18
018	03	3	078	03	3	138	06	06	198	67	EQ	258	44	SUM	318	30	30	378	01	01	438	69	OP
019	91	R/S	079	06	6	139	55	÷	199	03	03	259	00	00	319	73	RC*	379	82	HIR	439	23	23
020	04	4	080	05	5	140	01	1	200	52	52	260	69	OP	320	00	00	380	08	08	440	08	8
021	91	R/S	081	65	×	141	00	0	201	43	RCL	261	00	00	321	22	INV	381	43	RCL	441	42	STO
022	69	OP	082	43	RCL	142	00	0	202	04	04	262	32	X:T	322	44	SUM	382	03	03	442	01	01
023	00	00	083	06	06	143	54	)	203	55	÷	263	02	2	323	00	00	383	69	OP	443	97	DSZ
024	98	ADV	084	75	-	144	59	INT	204	04	4	264	67	EQ	324	85	+	384	06	06	444	05	05
025	68	NOP	085	01	1	145	65	×	205	95	=	265	03	03	325	43	RCL	385	43	RCL	445	01	01
026	68	NOP	086	68	NOP	146	93	.	206	22	INV	266	30	30	326	22	22	386	24	24	446	67	EQ
027	68	NOP	087	85	+	147	07	7	207	59	INT	267	05	5	327	95	=	387	82	HIR	447	25	CLR
028	68	NOP	088	03	3	148	05	5	208	67	EQ	268	67	EQ	328	82	HIR	388	05	05	448	91	R/S
029	68	NOP	089	01	1	149	54	)	209	03	03	269	03	03	329	06	06	389	43	RCL	449	42	STO
030	68	NOP	090	65	×	150	59	INT	210	52	52	270	11	11	330	73	RC*	390	25	25	450	05	05
031	68	NOP	091	43	RCL	151	54	)	211	69	OP	271	69	OP	331	00	00	391	82	HIR	451	61	GTO
032	04	4	092	01	01	152	75	-	212	39	39	272	20	20	332	82	HIR	392	06	06	452	01	01
033	05	5	093	75	-	153	53	(	213	61	GTO	273	06	6	333	05	05	393	43	RCL	453	67	67
034	01	1	094	03	3	154	24	CE	214	03	03	274	67	EQ	334	69	OP	394	26	26	454	76	LBL
035	07	7	095	01	1	155	55	÷	215	52	52	275	03	03	335	00	00	395	82	HIR	455	13	C
036	01	1	096	05	5	156	07	7	216	44	SUM	276	03	03	336	22	CLR	396	07	07	456	06	STF
037	03	3	097	03	3	157	54	)	217	00	00	277	01	1	337	98	ADV	397	43	RCL	457	01	01
038	03	3	098	32	X:T	158	59	INT	218	73	RC*	278	67	EQ	338	87	IFF	398	27	27	458	61	GTO
039	05	5	099	43	RCL	159	65	×	219	00	00	279	02	02	339	01	01	399	82	HIR	459	04	04
040	07	7	100	01	01	160	07	7	220	82	HIR	280	97	97	340	04	04	400	00	08	460	66	66
041	01	1	101	07	7	161	95	=	221	36	36	281	03	3	341	73	73	401	69	OP	461	76	LBL
042	69	OP	102	01	01	162	42	STO	222	44	SUM	282	67	EQ	342	43	RCL	402	05	05	462	14	D
043	03	03	103	09	09	163	02	02	223	00	00	283	12	02	343	01	01	403	69	OP	463	22	INV
044	69	OP	104	69	OP	164	07	7	224	73	RC*	284	91	91	344	32	X:T	404	00	00	464	86	STF
045	05	05	105	36	36	165	44	SUM	225	00	00	285	73	RC*	345	01	1	405	32	X:T	465	01	01
046	98	ADV	106	61	GTO	166	01	01	226	82	HIR	286	00	00	346	09	9	406	48	EXC	466	42	STO
047	25	CLR	107	01	01	167	43	RCL	227	06	06	287	52	EE	347	67	EQ	407	02	02	467	01	01
048	91	R/S	108	23	23	168	20	20	228	43	RCL	288	61	GTO	348	04	04	408	32	X:T	468	25	CLR
049	76	LBL	109	53	(	169	22	22	229	22	22	289	03	03	349	38	38	409	05	5	469	91	R/S
050	12	B	110	93	.	170	09	09	230	82	HIR	290	13	13	350	69	OP	410	67	EQ	470	61	GTO
051	98	ADV	111	04	4	171	43	RCL	231	36	36	291	73	RC*	351	21	21	411	02	02	471	00	00
052	22	INV	112	65	×	172	03	03	232	69	OP	292	00	00	352	02	2	412	42	42	472	60	60
053	86	STF	113	43	RCL	173	55	÷	233	20	20	293	52	EE	353	09	9	413	03	3	473	22	INV
054	01	01	114	01	01	174	04	4	234	73	RC*	294	61	GTO	354	42	STO	414	67	EQ	474	86	STF
055	32	X:T	115	85	+	175	95	=	235	00	00	295	03	03	355	00	00	415	02	02	475	01	01
056	01	1	116	02	2	176	42	STO	236	44	SUM	296	21	21	356	73	RC*	416	34	34	476	25	CLR
057	42	STO	117	93	.	177	04	04	237	00	00	297	73	RC*	357	01	01	417	02	2	477	91	R/S
058	01	01	118	03	3	178	22	INV	238	59	INT	298	00	00	358	59	INT	418	67	EQ			
059	32	X:T	119	54	)	179	59	INT	239	59	INT	299	52	EE	359	42	STO	419	02	02			

Nyheter på TI 58 C

av Björn Gustavsson

Mina tidigare artiklar har varit helt inriktade på hur fast mode kan användas på TI-59. Anledningen är helt enkelt att det har varit svårt eller omöjligt att använda fast mode på TI-58/58C. Nu har emellertid två nya metoder för körning av program i fast mode dykt upp. En av metoderna, St flg Ind 7 INV från tangentbordet, beskrevs i PB 81-3 s 29-32. Den andra metoden kräver att en hexkod sätts in i programmet, vilket endast kan göras med en serie knapptryckningar från tangentbordet.

På TI-59 är alltså den gamla modulmetoden överlägsen. Men på TI-58/58C är läget ett annat. På TI-58 förstörs ju programmet vid övergång till fast mode med modulmetoden. Därför är metoden som beskrevs i PB 81-3 utmärkt för TI-58.

På TI-58C fungerar inte modulmetoden. Den programmerbara metoden med hexkoden är däremot mycket lämplig. Hexkoden behöver bara skapas en gång - det kontinuerliga minnet bevarar ju programmet. (Tyvärr går det inte att göra samma sak med TI-59. Hexkoder kan inte spelas in på magnetkort, utan måste skapas på nytt när räknaren har varit avslagen.)

Här kommer alltså en beskrivning av hur fast mode används på TI-58C. Samma metod kan i princip användas på TI-59, men det rekommenderas inte. Därefter kommer en beskrivning av en upptäckt av fyra extra register på TI-58C.

Artikeln bygger på tre artiklar i TI PPC Notes av Patrick Acosta (v6nlp15-16, v6n8p3-4, v6n9/10p14-15).

000	91	R/S	006	85	+	012	22	INV	018	61	GTO
001	76	LBL	007	52	EE	013	57	ENG	019	00	00
002	11	A	008	01	1	014	86	STF	020	21	21
003	60	DEG	009	02	2	015	40	IND			
004	25	CLR	010	94	+/-	016	12	12			
005	04	4	011	95	=	017	00	0			

Figur 1

021	25	CLR	026	09	09	031	25	25	036	61	GTO
022	91	R/S	027	65	×	032	01	1	037	00	00
▷023	42	STO	028	97	DSZ	033	95	=	038	23	23
024	09	09	029	09	09	034	99	PRT			
▷025	43	RCL	030	00	00	035	91	R/S			

Figur 2

FAST MODE PÅ TI-58C

Den hexadecimala, programmerbara metoden är ganska lik metoden i PB 81-3. I stället för St flg Ind 7 INV från tangentbordet programmeras St flg Ind h12, där h12 är en hexadecimal kod eller hexkod. Vad är en hexkod?

Varje programsteg kan faktiskt innehålla tal mellan 0 och 255. Normalt går det inte att mata in fler än 100 av dessa, nämligen de vanliga tangentkoderna 00-99. Med vissa knep går det emellertid att mata in andra koder, som - något felaktigt - kallas hexkoder.

Eventuellt kommer en artikel i detta eller nästa nummer om hur hexkoder i allmänhet kan matas in. Här skall endast beskrivas hur den speciella hexkoden för övergång till fast mode matas in.

Liksom vid tangentbordsmetoden måste displayen innehålla ett 13-siffrigt tal. Om detta tal är lagrat i register nn kan följande sekvens användas: Lbl A Fix 0 Deg RCL nn St flg Ind h12 Nop GTO nnn. Rutinen i fast mode skall börja vid steg nnn. För att det skall fungera måste också den tolfte siffran i talet vara mindre än 4 och den trettonde vara 2, 4 eller 6.

Jag har ändrat den rutinen något och gjort en initieringsrutin som är 21 steg lång. Denna rutin återfinns i figur 1. Steg 016 är där hexkoden h12, som måste skapas på ett särskilt sätt. Det 13-siffriga talet skapas nu i steg 005-013, i stället för att hämtas från ett register.

Rutinen behöver bara matas in en gång. Det kontinuerliga minnet bevarar ju rutinen även om räknaren slås av. Alla program kan börja på steg 021.

Så här görs inmatningen:

1. Se till räknarens uppdelning är 3 Op 17.
2. Tryck GTO 16 LRN GTO BST LRN Pgm 12 SBR 444 R/S P/R LRN Ins LRN RST. Detta sätter in hexkoden.

3. Nu kan resten av programmet matas in på vanligt sätt. Observera: Hexkoden kommer att förstöras om den flyttas från steg 016. Använd alltså inte Del eller Ins före steg 017. Se också till att inte skriva över hexkoden med något annat.

De flesta vanliga begränsningarna för fast mode gäller (se t ex Komplement till instruktionsboken). Men det finns en viktig skillnad. Ett program som körs i normal mode kan anropa en subrutin i fast mode! Rutinen i fast mode avslutas med RTN som en vanlig subrutin. Däremot kan ett program i fast mode fortfarande inte använda subrutiner.

Exempel: I figur 2 finns ett enkelt program för fakultetsberäkning. Det matas in på vanligt sätt med början på steg 021. Det körs genom att först trycka A. Noll visas. Nu kan ett heltal (större än noll) matas in. Tryck R/S. Fakultetet beräknas och visas (om skrivare är ansluten trycks även fakultetet).

I figur 3 finns ett mer avancerat fakultetsprogram som också testar indata och ger rätt svar för 0. För att snabba upp programmet används en subrutin i fast mode. Programmet körs genom att mata in ett heltal och trycka B. Om talet inte är ett heltal och/eller mindre än 0 beräknas fakultet för absolutbeloppet och heltalsdelen och visas blinkande.

Steg 021-028 är subrutinen i fast mode. Huvudprogrammet börjar på steg 029. Först lagras heltalsdelen av absolutbeloppet av talet i R_9. Samtidigt beräknas skillnaden mellan det som matades in och R_9. Om den skillnaden är skild från noll är det inmatade talet otillåtet. Skillnaden lagras i R_0 med negativt tecken. Sedan anropas subrutinen i fast mode, som beräknar fakultet av talet i R_9.

Eftersom resultatet är lagrat som en vilande operation störs det inte av RCL 00 $\sqrt{x}$, som orsakar felindikering på grund av kvadratrotsurdragning ur negativt tal om det inmatade talet var felaktigt. Sedan fullföljer 1 = den vilande operationen. Om resultatet är 0 visas 1, eftersom 0-fakultet är 1.

021	43	RCL	030	12	B	039	42	STO	048	67	EQ
022	09	09	031	75	-	040	00	00	049	00	00
023	65	×	032	59	INT	041	11	A	050	52	52
024	97	DSZ	033	50	I×I	042	43	RCL	051	92	RTN
025	09	09	034	42	STO	043	00	00	052	01	1
026	00	00	035	09	09	044	34	√x	053	92	RTN
027	21	21	036	95	=	045	01	1			
028	92	RTN	037	50	I×I	046	95	=			
029	76	LBL	038	94	+/-	047	29	CP			

Figur 3

Nu finns det bara en nackdel med att använda subrutiner i fast mode: Om skrivaren är ansluten ställs Trace-läget in vid RTN från subrutinen. Det rekommenderas därför att köra programmet i figur 3 utan skrivare.

Om man vill kunna anropa flera subrutiner i fast mode kan programmet i figur 4 användas. Steg 016 skall fortfarande vara en hexkod. Anrop sker med nnn A', där nnn är den adress som subrutinen börjar vid. I programmet används som synes St flg h12 i stället för St flg Ind h12. Detta går bra på TI-58C men inte på TI-59.

EXTRA REGISTER PÅ TI-58C

TI-58C har 32 extra, dolda programsteg, nämligen stegen 480-511. Motsvarande register kommer här att kallas R_{56}-R_{59}, dvs som på en TI-59. Vi kommer att se att det finns goda skäl för att numrera registren så.

TI-58C har alltså ett 512-stegs programminne, som är det enda som är kontinuerligt i räknaren. R_{59} används för att bevara inställd uppdelning och fast decimalplacering när räknaren slås av.

De första 13 siffrorna i R_{59} är ln 10 utan exponent (motsvarar steg 487-481; se Samband programsteg - dataregister i Komplement till instruktionsboken). Om inte detta tal finns där när räknaren slås på raderas hela programminnet för att skydda användaren från oanad minnesförlust. I stället drabbas användaren av känd minnesförlust.

De följande två siffrorna (som normalt lagrar exponent) bestämmer den uppdelning som skall ställas in när räknaren slås på. Dessa siffror kan ändras med metod som beskrivs nedan, med resultat att en del pseudo-uppdelningar kan fås.

Den sista siffran (som normalt lagrar tecknen) bestämmer den fasta decimalinställningen. Fix 9 lagras som en nolla. Fix n för $0 \leq n \leq 8$ lagras som n+2. Men 8+2 = 10, och man kan fråga sig hur det kan lagras i en enda siffra. Det lagras som en hexkod, och visas som en nolla, samtidigt som tiotalssiffran i samma programsteg verkar vara höjd med ett.

Om man sätter siffran lika med 1 och slår av och på räknaren, fås en sorts Fix -1 display som uppför sig underligt.

För att kunna ändra något i R_{59} matar man först in följande program: Lbl A Fix 0 R/S. Tryck sedan från tangentbordet 609 + EE 9 +/- = INV EE A St flg Ind 7 INV LRN. Du skall nu vara vid steg 480. Tryck ABS RCL. Slå av och på räknaren, tryck Op 16 och se 0.09. Det betyder att det finns 1200 programsteg och inga register.

Naturligtvis finns det inget minne bortom steg 511, så det är ingen idé att mata in något där. Men det är möjligt att gå till steg 480 med GTO 480, och mata in rätt kod för någon av de primära uppdelningarna i figur 5. Slå sedan av och på räknaren för att ställa in uppdelningen.

Om man väljer någon av de fyra första primära uppdelningarna, kommer sedan räknaren att uppföra sig som en TI-59. Alla TI-59s uppdelningar kan ställas in med Op 17. Dessa pseudo-59 uppdelningar finns uppräknade i figur 5.

Alla register kommer då att numreras som på TI-59. R_0 består t ex av stegen 952-959 och är oanvändbart eftersom inget minne finns vid dessa steg. Registren 56-99 är däremot användbara.

Exempel: Vi ställer in den första primära uppdelningen genom att skriva List STO i steg 480-481 och slå av och slå på räknaren. Om vi nu trycker 6 Op 17 ställs uppdelningen 479.59 in. Det finns 480 programsteg och 60 register, varav bara R_{56}-R_{59} kan användas. Det går nu att använda ett program som är 480 steg långt och ändå ha fyra register (samt HIR-register).

Det finns några risker med att använda R_{59}. Dels ändras innehållet i R_{59} om Fix n eller Op 17 används. Dels måste man återställa ln 10 innan man slår av räknaren - i alla fall om man vill ha programmet kvar.

Det går också att använda steg 480-511 för program. Lämplig uppdelning är då 5 Op 17. Då har man 512 steg men inga register (förutom HIR-register). Även då måste man återställa ln 10 innan räknaren slås av.

000	91	R/S	005	00	00	010	02	2	015	86	STF
001	76	LBL	006	04	4	011	94	+/-	016	12	12
002	16	A^x	007	85	+	012	95	=	017	00	0
003	60	DEG	008	52	EE	013	22	INV	018	83	GO*
004	42	STO	009	01	1	014	57	ENG	019	00	00

Figur 4

En pseudo-59 uppdelning kommer att ändras till motsvarande primära uppdelning (se figur 5) om räknaren slås av. Därför bör någon av de fyra första primära uppdelningarna ställas in innan räknaren slås av. Detta kan göras genom att gå till steg 480 och skriva in rätt koder, eller genom att exekvera i program eller från tangentbordet: 10 lnx x 1 EE 19 = STO 59 CLR.

Kod i steg... 480	481	Primär uppdelning	Antal register	Pgm-steg	Pseudo-59 uppdelning
90	41	239.89	90	240	--
80	41	159.99	100	160	--
70	41	79.09	110	80	--
60	41	0.19	120	0	--
50	41	0.09	0	1200	--
40	41	0.19	0	1120	--
30	41	0.29	0	1040	--
20	41	0.39	0	960	959.00
10	41	0.49	0	880	879.09
00	41	0.59	0	800	799.19
90	40	0.69	0	720	719.29
80	40	0.79	0	640	639.39
70	40	0.89	0	560	559.49
60	40	479.00	0	480	479.59
50	40	399.09	10	400	399.69
40	40	319.19	20	320	319.79
30	40	239.29	30	240	239.89
20	40	159.39	40	160	159.99
10	40	79.49	50	80	--
00	40	0.59	60	0	--

Not: Koden 41 kan bytas ut mot koderna 42-49. Pseudo-59 uppdelningar har det antal programsteg och register som en TI-59 har.

Figur 5

Yrkesmässig programförsäljning

BYGGNADSTEKNIK

Kontinuerliga balkar, ramar, takstolar och fackverk, dimensionering av betong, stål och trä, stödmurar, pålgrupper, grundplattor, plattor, BBK-program. Moduler Structural Engineering och Baustatik II.

Carl-Adolf Granholm
Odinsgatan 6
411 03 GÖTEBORG
tel 031/80 45 75

Lars Hedlund
Årstavägen 27
121 68 JOHANNESHOV
tel 08/91 38 97

HOBBIT

av Dejan Ristanović
Gvozdićeva 34
YU-11000 BELGRAD

Svensk text av Lars Hedlund

Dejan Ristanović i Belgrad, Jugoslavien, har sänt in det här spelprogrammet till oss. Han har haft bidrag införda i både TI PPC Notes och TISofts medlemsblad. Detta anser han själv vara ett av de bästa TI-59-program han gjort. Han spelade först "Hobbit" på sin TRS-80 hemdator och lyckades därefter göra om det för TI-59. "Hobbit" är det engelska ordet för "hob", dvs folket i Tolkiens böcker. Bakgrunden till det här spelet tycks vara en fantasifull blandning av Tolkien och annat så vi har översatt de engelska orden ganska fritt.

Vad spelet går ut på. Spelaren ger sig in i en labyrint, där en trollkarl förvarar "Orb of Zot". Trollkarlen är inte där, och spelaren skall försöka stjäla "Orb of Zot" fastän han inte vet var den finns. Det finns demoner i labyrinten som stjäl ädelstenar från spelaren och i värsta fall dödar honom, men han kan också döda demonerna. Om spelaren är för långsam med att hitta "Orb of Zot" dyker trollkarlen upp och dödar honom. Om man klarar sig från detta räknas det till sist ihop en poängsumma (score, SCR) som bl a tar hänsyn till hur många demoner man avlivat. Om man är försiktig rymmer man givetvis så fort man fått tag på "Orb of Zot".

Spelet kan spelas i olika svårighetsgrader (1 - 9).

Labyrinten är en spelplan med 100 rutor numrerade från 00 till 99, där 00 är "NV" hörnet (upptill t v), 09 "NO" hörnet (upptill t h), 90 "SV" hörnet (nedtill t v) och 99 "SO" hörnet (nedtill t h). Planen är en s k torus, dvs nord- och sydsidan är en gemensam linje och man kan alltså gå söderut från ruta 94 till ruta 04, och likaså gör en gemensam väst- och östsida att man kan gå österut från t ex ruta 49 varvid man dock inte hamnar på 40 utan på 50.

Man kan röra sig och vidta åtgärder enligt följande:

A Flytta ett steg åt väster
B -"- öster
C -"- norr
D -"- söder
E Använda lysbloss
A' Döda demon (bara om man har en amulett)
B' Teleportera (bara om man har en runstav)
D' Lämna labyrinten.

Följande varelser och föremål finns utspridda i labyrinten:

1. "Orb of Zot" som ser ut att vara en "varp" (se 5).
2. Demoner (antalet beror på svårighetsgraden). Skrives DEMON.
3. Amuletter (7 st). Skrives AMULE.
4. Extra lysbloss (man får några från starten). Skrives FLARE.
5. "Varpar" (antalet beror på svårighetsgraden). Skrives WARP.
6. Runstaven som ser ut att vara en demon. Skrives STAFF.
7. Ädelstenar på alla rutor som inte är upptagna av annat. Skrives GEMS.

Och här något hur de olika elementen i spelet fungerar:

Lysbloss. Tryck ett rutnummer och E. Om det "bara" finns ädelstenar där visas ingenting. Annars skrivs något: TRACK (spår) om Du har varit där förut; AMULE för amulett; DEMON för demon eller runstav; WARP för "varp" eller Orb of Zot. Förrådet med lysbloss är ej obegränsat, men om man hamnar på en ruta där det skrivs FLARE har man fått nio nya.

Ädelstenar behövs om man ännu inte fått tag i någon amulett för att betala demoner så att man inte skall bli dödad (programmet sköter om att 0 - 10 st tas från ens förråd och det skrivs ut BRIBE - muta). När man för första gången kommer till en "tom" ruta fylls förrådet av ädelstenar på och det totala antalet skrivs ut som GEMS - så länge som man inte har någon amulett för då behöver man ju inte den upplysningen längre. Antalet ädelstenar beräknas dock alltid.

Amuletter. Om man hamnar på en ruta med amulett skrivs AMULE och amulettens nummer. Den inte bara skyddar mot demoner, man kan döda demoner med den också när man står på demonens ruta genom att trycka amulettens nummer och A'. (Om man inte har den amuletten eller står på fel ruta skrivs SORRY.) Amulettens energi räcker bara till för att ta kål på ett begränsat antal demoner, men den skyddar alltid innehavaren från att bli dödad av en demon.

"Varpar". Om man hamnar på en ruta med en "varp" kastas man i väg till en annan ruta (som naturligtvis skulle kunna innehålla ytterligare en "varp" så att man blir kastad till en tredje ruta! Det kan t o m inträffa att två "varpar" har varandras nummer så att man bara kastas fram och tillbaka. I så fall måste man avbryta spelet med R/S och börja om. Programutrymmet räckte inte till för att gardera mot detta).

Orb of Zot ser ut som en "varp". Om man emellertid flyttar till hans ruta kastas man inte till en bestämd ruta utan flyttas ytterligare ett steg i samma riktning. Amta att Orb är på 31 och Du står på 32. Du trycker A (gå ett steg åt väster). Du hamnar nu inte på 31 utan på 30. Om Du nu trycker B (gå ett steg åt öster) hamnar Du på 32! Detta är det enda sättet att ta reda på var Orb of Zot är. För att komma till Orb of Zot måste man använda ett speciellt hjälpmedel, nämligen teleportering.

Teleportering. För att teleportera måste man ha en runstav. Slå numret dit Du vill bli teleporterad och tryck B'. Men nu förlorar Du alla Dina amuletter och ädelstenar! Det gäller alltså att veta exakt vad man gör när man teleporterar så att man inte hamnar på en vanlig "varp" (därför att man tror att det är Orb of Zot) och blir kastad till en ruta med en demon för då har man varken amulett eller ädelstenar och blir alltså dödad.

Runstav. Det går inte att använda lysbloss för att hitta runstaven för då skrivs ju DEMON. Det går bara att hitta den genom att gå till rätt ruta för då skrivs STAFF och då har man den så att man kan teleportera.

Inspelning och användning av magnetkort

Programmet består som synes av två stycken "sida 1" och en "sida 2" och en "sida 3". Vr hanteringssynpunkt är det nog bäst att spela in de båda sidorna "1" på samma kort och sidorna "2" och "3" på ett annat kort. Det blir ju då självfallet mycket viktigt att märka sidorna på kortet "1" med "start" och "fortsättning". (Allt i normaluppdelning!)

Kör programmet så här: Läs "start"-sida 1 och sidorna 2 och 3. Mata in ett slumptalsfrö och tryck E. Mata in svårighetsgrad (1-9) och tryck A. Nu arbetar räknaren i flera minuter. Sätt under tiden in "fortsättnings"-sida 1 i kortläsaren. När "HOBBIT" skrivits läser maskinen denna sida. Tryck nu R/S så har Du Din startposition. Nu är det bara att välja strategi och använda tangenterna A - E och A', B' eller D'. Efter avslutat spel kan Du ta ut kortet ur kortläsaren och läsa "start"-sidan igen - sidorna 2 och 3 finns ju redan inlästa. Mata in slumptalsfrö etc precis som ovan.

För att vinna måste man lägga beslag på Orb of Zot och fly innan trollkarlen återvänder. Men man kan också försöka döda ytterligare några demoner. Orb of Zot kan emellertid inte skydda mot trollkarlen så när denne återvänder har man alltid förlorat.

Strategin blir alltså troligen denna: Använd först lysbloss för att se att Du inte kommer till en demon. När Du har minst tio ädelstenar eller en eller flera amuletter kan Du gå utan lysbloss. Avliva alltid demoner om Du har en fungerande amulett, det ger poäng. Använd ett rutat papper där Du kan rita in hur Du går och försök gå så systematiskt som möjligt (alltsomoftast blir Du ju "kastad" till ett nytt ställe). Lokalisera Orb of Zot samt runstaven som ju behövs för att hämta Orb.

Börja gärna med låg svårighetsgrad innan Du har utvecklat strategin. Om Du tycker att Du uppnått hög poäng i något parti så kan Du ju meddela Programbiten den tillsammans med tillhörande slumptalsfrö och svårighetsgrad så får vi kanske se vem som är bäst på att spela Hobbit.

NUMERIC	REG	ALPHA
22173036.	41	GEMS
43133533.	42	WARP
3735131526.	43	TRACK
2127133517.	44	FLARE
1330412717.	45	AMULE
1617303231.	46	DEMON
1617303231.	47	DEMON
33323640.	48	POS.
1435241417.	49	BRIBE
3637132121.	50	STAFF
49332410.	51	YDU
2732363773.	52	LDST*
4324461335.	53	WIZAR
1600351737.	54	D RET
4135311716.	55	URNED
4324317373.	56	WIN**
2624272773.	57	KILL*
3632353545.	58	SORRY
32351473.99937	59	

På följande sida finns listning till programmets tre delar. Överst "start"-program 1 t v och "fortsättnings"-program 1 t h, nedtill kortsida 2 och här till vänster står alltså den registerlistning som motsvarar kortsida 3. R_{59} (13 siffror) matas förstås in i två delar (t ex heltalsdel och decimaldel) med STO 59 resp. SUM 59.

START

000	01	1
001	91	R/S
002	61	GTO
003	01	01
004	04	04
005	76	LBL
006	10	E'
007	01	1
008	00	0
009	00	0
010	76	LBL
011	16	A'
012	65	×
013	36	PGM
014	15	15
015	71	SBR
016	88	DMS
017	95	=
018	59	INT
019	92	RTN
020	76	LBL
021	19	D'
022	42	STO
023	02	02
024	10	E'
025	85	+
026	29	CP
027	22	INV
028	67	EQ
029	00	00
030	24	24
031	43	RCL
032	02	02
033	85	+
034	43	RCL
035	12	12
036	95	=
037	55	÷
038	01	1
039	00	0
040	00	0
041	00	0
042	85	+
043	43	RCL
044	13	13
045	95	=
046	65	×
047	82	HIR
048	13	13
049	95	=
050	72	ST*
051	14	14
052	92	RTN
053	76	LBL
054	17	B'
055	42	STO
056	15	15
057	75	-
058	53	(
059	24	CE

KORTSIDA 2

060	55	÷
061	04	4
062	54	)
063	59	INT
064	42	STO
065	14	14
066	65	×
067	04	4
068	00	0
069	94	+/-
070	65	×
071	03	3
072	85	+
073	01	1
074	02	2
075	95	=
076	22	INV
077	28	LOG
078	52	EE
079	82	HIR
080	01	01
081	25	CLR
082	01	1
083	06	6
084	44	SUM
085	14	14
086	73	RC*
087	14	14
088	55	÷
089	82	HIR
090	13	13
091	95	=
092	42	STO
093	13	13
094	22	INV
095	59	INT
096	22	INV
097	59	INT
098	13	13
099	65	×
100	01	1
101	00	0
102	00	0
103	00	0
104	95	=
105	42	STO
106	12	12
107	59	INT
108	22	INV
109	44	SUM
110	12	12
111	92	RTN
112	76	LBL
113	15	E
114	32	XIT
115	04	4
116	00	0
117	36	PGM
118	01	01
119	71	SBR

START

120	00	00
121	12	12
122	32	XIT
123	42	STO
124	09	09
125	92	RTN
126	76	LBL
127	11	A
128	42	STO
129	03	03
130	01	1
131	19	D'
132	43	RCL
133	15	15
134	82	HIR
135	04	04
136	05	5
137	00	0
138	00	0
139	19	D'
140	43	RCL
141	03	03
142	85	+
143	03	3
144	95	=
145	42	STO
146	00	00
147	04	4
148	00	0
149	00	0
150	19	D'
151	97	DSZ
152	00	00
153	01	01
154	47	47
155	00	0
156	00	0
157	00	0
158	19	D'
159	07	7
160	42	STO
161	00	00
162	03	3
163	00	0
164	00	0
165	85	+
166	43	RCL
167	00	00
168	95	=
169	19	D'
170	97	DSZ
171	00	00
172	01	01
173	62	62
174	01	1
175	05	5
176	85	+
177	43	RCL
178	03	03
179	95	=

FORTSÄTTNING

180	42	STO
181	00	00
182	10	E'
183	29	CP
184	67	EQ
185	01	01
186	82	82
187	19	D'
188	97	DSZ
189	00	00
190	01	01
191	82	82
192	01	1
193	00	0
194	00	0
195	19	D'
196	43	RCL
197	15	15
198	02	2
199	04	04
200	01	1
201	01	1
202	42	STO
203	16	A'
204	06	6
205	42	STO
206	85	+
207	08	8
208	05	5
209	95	=
210	42	STO
211	08	08
212	98	ADV
213	02	2
214	03	3
215	03	3
216	02	2
217	01	1
218	04	4
219	69	DP
220	00	00
221	69	DP
222	00	00
223	01	1
224	04	04
225	04	4
226	04	4
227	02	2
228	07	7
229	00	0
230	00	0
231	00	0
232	00	0
233	69	DP
234	00	00
235	69	DP
236	05	05
237	69	DP
238	00	00
239	69	DP

KORTSIDA 2

240	82	HIR
241	07	07
242	43	RCL
243	45	45
244	18	C'
245	71	SBR
246	59	INT
247	99	PRT
248	22	INV
249	28	LOG
250	52	EE
251	44	SUM
252	05	05
253	25	CLR
254	61	GTO
255	75	-
256	22	INV
257	86	STF
258	06	06
259	43	RCL
260	47	47
261	18	C'
262	43	RCL
263	05	05
264	29	CP
265	22	INV
266	67	EQ
267	86	STF
268	14	14
269	43	RCL
270	49	49
271	18	C'
272	01	1
273	00	0
274	15	E'
275	99	PRT
276	22	INV
277	44	SUM
278	10	10
279	43	RCL
280	10	10
281	29	CP
282	77	GE
283	01	01
284	96	96
285	01	1
286	94	+/-
287	49	PRD
288	06	06
289	43	RCL
290	52	52
291	69	OP
292	02	02
293	43	RCL
294	51	51
295	69	OP
296	01	01
297	69	OP
298	05	05
299	03	3

300	06	6
301	01	1
302	05	5
303	03	3
304	05	5
305	69	DP
306	04	04
307	43	RCL
308	06	06
309	69	DP
310	06	06
311	25	CLR
312	91	R/S
313	76	LBL
314	19	D'
315	22	INV
316	87	IFF
317	00	00
318	02	02
319	85	85
320	43	RCL
321	56	56
322	61	GTO
323	02	02
324	91	91
325	76	LBL
326	16	A'
327	02	IFF
328	06	06
329	32	XIT
330	22	INV
331	28	LOG
332	52	EE
333	82	HIR
334	02	02
335	25	CLR
336	43	RCL
337	05	05
338	55	÷
339	82	HIR
340	12	12
341	95	=
342	59	INT
343	55	÷
344	01	1
345	00	0
346	95	=
347	22	INV
348	59	INT
349	32	XIT
350	25	CLR
351	67	EQ
352	32	XIT
353	93	,
354	04	4
355	67	EQ
356	32	XIT
357	82	HIR
358	12	12
359	44	SUM

KORTSIDA 2

360	05	05
361	43	RCL
362	57	57
363	69	DP
364	00	00
365	18	C'
366	69	DP
367	26	26
368	61	GTO
369	75	-
370	43	RCL
371	04	04
372	32	XIT
373	82	HIR
374	14	14
375	67	EQ
376	01	01
377	00	00
378	71	SBR
379	59	INT
380	42	STO
381	04	04
382	61	GTO
383	01	01
384	04	04
385	76	LBL
386	32	XIT
387	43	RCL
388	58	58
389	69	OP
390	00	00
391	18	C'
392	61	GTO
393	01	01
394	96	96
395	76	LBL
396	17	B'
397	22	INV
398	87	IFF
399	01	01
400	32	XIT
401	42	STO
402	04	04
403	32	XIT
404	25	CLR
405	42	STO
406	10	10
407	42	STO
408	05	05
409	82	HIR
410	14	14
411	22	INV
412	67	EQ
413	01	01
414	04	04
415	01	1
416	94	+/-
417	82	HIR
418	04	04
419	43	RCL

FORTSÄTTNING

420	59	59
421	86	STF
422	00	00
423	61	GTO
424	03	03
425	89	89
426	76	LBL
427	59	INT
428	43	RCL
429	01	01
430	55	÷
431	01	1
432	00	0
433	00	0
434	95	=
435	22	INV
436	59	INT
437	65	×
438	01	1
439	00	0
440	00	0
441	95	=
442	92	RTN
443	86	STF
444	01	01
445	43	RCL
446	50	50
447	18	C'
448	61	GTO
449	75	-
450	76	LBL
451	15	E
452	22	INV
453	97	DSZ
454	11	11
455	32	XIT
456	71	SBR
457	43	RCL
458	29	CP
459	67	EQ
460	01	01
461	96	96
462	55	÷
463	01	1
464	00	0
465	00	0
466	85	+
467	04	4
468	02	2
469	95	=
470	42	STO
471	00	00
472	73	RC*
473	00	00
474	18	C'
475	61	GTO
476	01	01
477	96	96
478	00	0
479	00	0

FORTSÄTTNING

000	76	LBL
001	10	E'
002	65	×
003	36	PGM
004	15	15
005	71	SBR
006	88	DMS
007	95	=
008	59	INT
009	92	RTN
010	76	LBL
011	18	C'
012	69	OP
013	02	02
014	69	OP
015	05	05
016	69	OP
017	00	00
018	92	RTN
019	76	LBL
020	43	RCL
021	42	STO
022	15	15
023	75	-
024	53	(
025	24	CE
026	55	÷
027	04	4
028	54	)
029	59	INT
030	42	STO
031	14	14
032	65	×
033	04	4
034	95	=
035	94	+/-
036	65	×
037	03	3
038	85	+
039	01	1
040	02	2
041	95	=
042	22	INV
043	28	LOG
044	52	EE
045	82	HIR
046	03	03
047	25	CLR
048	01	1
049	06	6
050	44	SUM
051	14	14
052	73	RC*
053	14	14
054	55	÷
055	82	HIR
056	13	13
057	29	CP
058	42	STO
059	13	13

FORTSÄTTNING

060	22	INV
061	59	INT
062	22	INV
063	44	SUM
064	13	13
065	61	GTO
066	01	1
067	00	0
068	00	0
069	00	0
070	76	LBL
071	42	STO
072	12	12
073	59	INT
074	22	INV
075	44	SUM
076	12	12
077	92	RTN
078	76	LBL
079	11	A
080	25	CLR
081	94	+/-
082	76	LBL
083	12	12
084	01	1
085	42	STO
086	02	02
087	61	GTO
088	01	01
089	00	00
090	76	LBL
091	13	C
092	25	CLR
093	94	+/-
094	76	LBL
095	14	D
096	01	1
097	00	0
098	42	STO
099	02	02
100	42	STO
101	02	02
102	44	SUM
103	04	04
104	86	STF
105	06	06
106	43	RCL
107	48	48
108	69	OP
109	04	04
110	43	RCL
111	04	04
112	29	CP
113	77	GE
114	01	01
115	21	21
116	65	×
117	01	1
118	00	0
119	00	0

START

120	95	=
121	42	STO
122	01	01
123	71	SBR
124	59	INT
125	42	STO
126	04	04
127	69	DP
128	04	04
129	69	DP
130	65	×
131	71	SBR
132	43	RCL
133	42	STO
134	01	01
135	29	CP
136	67	EQ
137	02	02
138	00	0
139	55	÷
140	01	1
141	00	0
142	00	0
143	95	=
144	59	INT
145	32	XIT
146	25	CLR
147	67	EQ
148	03	03
149	70	70
150	01	1
151	67	EQ
152	01	01
153	96	96
154	04	4
155	67	EQ
156	02	02
157	56	56
158	03	3
159	67	EQ
160	02	02
161	36	36
162	05	5
163	67	EQ
164	04	04
165	43	43
166	43	RCL
167	44	44
168	18	C'
169	09	9
170	44	SUM
171	11	11
172	76	LBL
173	75	-
174	01	1
175	21	21
176	65	×
177	85	+
178	43	RCL
179	12	12

KORTSIDA 2

180	95	=
181	55	÷
182	01	1
183	00	0
184	00	0
185	00	0
186	85	+
187	43	RCL
188	13	13
189	95	=
190	65	×
191	82	HIR
192	13	13
193	95	=
194	72	ST*
195	14	14
196	97	DSZ
197	08	08
198	02	02
199	14	14
200	43	RCL
201	53	53
202	69	OP
203	01	01
204	43	RCL
205	54	54
206	69	OP
207	03	03
208	43	RCL
209	54	54
210	18	C'
211	61	GTO
212	02	02
213	85	85
214	25	CLR
215	91	R/S
216	05	5
217	10	E'
218	44	SUM
219	10	10
220	43	RCL
221	05	05
222	29	CP
223	22	INV
224	67	EQ
225	75	-
226	43	RCL
227	59	59
228	67	EQ
229	04	04
230	43	RCL
231	10	10
232	69	OP
233	06	06
234	61	GTO
235	75	-
236	43	RCL
237	59	59
238	22	INV
239	59	INT

Display — "teckenruta"?

Tidskriften "Språkvård" som utges av Svenska Språknämnden skrev följande i sitt nr 2-1981:

"På vissa ur, på miniräknare m.m. finns det ett litet "fönster" där siffror, bokstäver och andra tecken visas. Ett vanligt ord för detta, också i svensk text är det engelska ordet *display*. Samma ord används i andra länder, t.ex. Danmark, Norge och Finland. Inom Nordiska språksekretariatet har man försökt få fram ett helt gemensamt nordiskt ersättningsord för display. Resultatet har blivit *teckenruta* (norska *tegnrute*, danska *tegnrude*). Svenska språknämndens styrelse har beslutat att för svenskans del rekommendera *teckenruta* i st. f. *display*. Det är nämndens förhoppning att detta nybildade ord snabbt skall slå igenom, så att det onödiga och mera svårhanterliga engelska lånordet *display* kan trängas tillbaka."

Har Du några åsikter i frågan så skriv till Programbiten! Själva har vi ännu inte tagit ståndpunkt för det ena eller det andra. ■

DJUP-DYKNING I TI-59 (3)

av BJÖRN GUSTAVSSON

I PB 80-3 s14-15 och PB 80-4 s23 gjorde Lars Hedlund djupdykningar i TI-59 och listade den fasta programvaran. I den här djupdykningen skall vi titta på hur 59:an (och 58:an) beräknar lnx, log, INV lnx och INV log. I nästa djupdykning tar vi itu med de trigonometriska funktionerna. - Källor: Display och TISofts medlemstidskrift.

Som konstaterades i tidigare djupdykningar finns vissa konstanter permanent lagrade i TI-59. De som är intressanta för oss just nu är följande:

ln 10
ln 2
ln 1.1
ln 1.01
ln 1.001
ln 1.0001
ln 1.00001
ln 1.000001

vilka används för beräkning av ln x och e^x. $^{10}\log x$ och 10^x beräknas från ln x och e^x enligt:

$$^{10}\log x = \ln x / \ln 10$$
$$10^x = e^{x \cdot \ln 10}$$

BERÄKNING AV ln x (x>0)

Alla tal lagras i tiopotensnotation oavsett hur de visas i displayen:

$$x = m \cdot 10^p \quad (x>0,\ 1 \leq m < 10)$$

p är exponenten, m är mantissan. Därav följer att

$$\ln x = \ln m + p \cdot \ln 10$$

Konstanten ln 10 finns ju lagrad, så det är enkelt att beräkna p ln 10. Återstår att beräkna ln m.

Först multipliceras m med 2 så många gånger att det nya resultatet kommer så nära 10 som möjligt utan att överskrida 10. Detta upprepas med faktorn 1.1 på samma sätt; och sedan med alla faktorerna 1.01, 1.001... 1.000001. När det är klart har vi ett tal r, som ligger nära 10 men $r \leq 10$. Följande gäller

$$m \cdot \text{faktorer} = r \ (r \leq 10).$$

"Faktorer" är produkten av alla faktorer vi multiplicerade med. Likheten kan skrivas om som

$$m = r/\text{faktorer}$$

eller

$$\ln m = \ln r - \ln (\text{faktorer}).$$

Eftersom logaritmerna för alla faktorer finns lagrade återstår att beräkna ln r. Som vi tidigare sagt ligger r nära 10. Då bör $\ln r \approx \ln 10$. Nu visar det sig emellertid att den approximationen inte är tillräckligt bra - när jag gjorde ett försök att beräkna ln 18.12 fick jag ett fel redan i sjätte decimalen.

Ett bättre värde på ln r får man med

$$\ln r \approx \ln 10 + r/10 - 1,$$

och det är också så TI-59 beräknar ln r.

(Den approximationen kan härledas så här: Vi söker en linear funktion $f(x) = a + bx$, sådan att $f(x) \approx \ln x$, för $x \approx 10$. Koefficienterna a och b måste bestämmas. Om vi kräver att $\ln 10 = f(10)$, och att derivatornas funktionsvärden skall vara lika för x = 10 fås ekvationssystemet:

$$\begin{cases} \ln 10 = a + 10b \\ 1/10 = b \end{cases}$$

(Derivatan av ln x är som bekant 1/x.)

Lösningen är $a = \ln 10 - 1$, $b = 1/10$; $f(x) = \ln 10 - 1 + x/10$.)

Då blir

$$\ln m \approx \ln 10 + r/10 - 1 - \ln (\text{faktorer})$$

och

$$\ln x \approx (p + 1) \cdot \ln 10 - \ln (\text{faktorer}) + r/10 - 1.$$

Exempel: Vi beräknar ln 18.12. Då är $x = 1.812 \cdot 10^1$, dvs m = 1,812, p = 1.

		2·ln 10
1.812	·2	-ln 2
3.624	·2	-ln 2
7.248	·1.1	-ln 1.1
7.9728	·1.1	-ln 1.1
8.77008	·1.1	-ln 1.1
9.647088	·1.01	-ln 1.01
9.74355888	·1.01	-ln 1.01
9.840994469	·1.01	-ln 1.01
9.939404413	·1.001	-ln 1.001
9.949343818	·1.001	-ln 1.001
9.959293162	·1.001	-ln 1.001
9.969252455	·1.001	-ln 1.001
9.979221707	·1.001	-ln 1.001
9.989200929	·1.001	-ln 1.001
(Faktorn 1.0001 kan ej användas.)		
9.99919013	·1.00001	-ln 1.00001
9.999290122	·1.00001	-ln 1.00001
9.999390115	·1.00001	-ln 1.00001
9.999490109	·1.00001	-ln 1.00001
9.999590104	·1.00001	-ln 1.00001
9.999690099	·1.00001	-ln 1.00001
9.999790096	·1.00001	-ln 1.00001
9.999890094	·1.00001	-ln 1.00001
9.999990093 (= r)		+r/10-1
		2.897016301

Den sista faktorn, 1.000001, kunde inte användas.

BERÄKNING AV e^x

Resultatet av e^x skall lagras i tiopotensnotation och därför bestäms först vilken 10-exponent resultatet skall ha. Samtidigt reduceras x så att $0 \leq x < \ln 10$, och alltså $1 \leq e^x < 10$. Detta görs genom att minska eller öka x med ln 10 tills x ligger inom rätt gränser.

Exempel:

$$e^8 = e^8/10 \cdot 10^1 = e^{8-\ln 10} \cdot 10^1 = e^{8-3\cdot\ln 10} \cdot 10^3 \approx e^{1.09224472} \cdot 10^3$$

$e^{1.09224472}$ är mindre än 10.

Därefter gäller det att bestämma e^x för det reducerade x. Det görs genom att uttrycka x som en summa av logaritmer:

$$x = a_1 \cdot \ln 2 + a_2 \cdot \ln 1.1 + a_3 \cdot \ln 1.01 + \ldots + a_7 \cdot \ln 1.000001 + r$$

Då blir

$$e^x = 2^{a_1} \cdot 1.1^{a_2} \cdot 1.01^{a_3} \cdot \ldots \cdot 1.000001^{a_7} \cdot e^r$$

Beräkningen utförs så att man subtraherar ln 2 från x så många gånger att det nya resultatet kommer så nära noll som möjligt utan att bli mindre än noll. För varje subtraktion multiplicerar man det som skall bli e^x med 2. Sedan gör man likadant med ln 1.1; och sedan med ln 1.01, ln 1.001... ln 1.000001.

När det är klart har vi ett tal r kvar, som ligger mycket nära 0, men $r \geq 0$. Vi måste bestämma e^r. Som bekant är

$$e^x = 1 + x + x^2/2! + x^3/3! + \ldots + x^n/n! + \ldots$$

Nu när r är mycket nära 0 kan e^r approximeras med

$$e^r \approx 1 + r.$$

Det sista som återstår att göra är att multiplicera den tidigare erhållna produkten med (1 + r).

Exempel: Beräkna e^8. Först görs detta om till $e^{8-3\cdot\ln 10} \cdot 10^3 = e^{1.092244721} \cdot 10^3$.

1.092244721-ln 2	2
0.3990975405-4·ln 1.1	1.1^4
0.0178568212-ln 1.01	1.01
0.0079064904-7·ln 1.001	1.001^7
0.0090099881-9·ln 1.0001	1.0001^9
0.0000100331-ln 1.00001	1.00001
0.0000000331 (= r)	·(1+r)
	2.980957987

(Den sista konstanten kunde inte användas.) Detta resultat skall multipliceras med 10^3, dvs $e^8 = 2980.957987$.

Multiplikationer och divisioner tar förhållandevis lång tid för en räknare att utföra. Det är därför viktigt för konstruktörerna som gör mikroprogrammen för t ex lnx och INV lnx att dessa innehåller så få multiplikationer och divisioner som möjligt. Hur många multiplikationer måste utföras i metoderna beskrivna ovan?

Svaret är <u>ingen</u>. 2·a t ex kan ju beräknas som a + a. a·1.001 är samma sak som a + 0.001·a. Att beräkna 0.001a är mycket enkelt - det är bara att flytta decimaltecknet 3 steg åt vänster. En sådan <u>skiftning</u> av siffrorna kan räknaren göra mycket snabbt.

Slut på djupdykningen för den här gången. ■

VINNANDE PROGRAMMET I

TÄVLING: SÄNKA SKEPP

I nr 81-3 ordnade Anders Persson en tävling mellan olika "sänka skepp"-program. Segrare i tävlingen blev Björn Edman i Malmö. Sedan numret kommit ut har med rätta undrats hur det segrande programmet såg ut. Här är det och anvisningarna är dessa:

Initiering: ange slumptalsfrö (reellt tal), tryck RST R/S. Placering av fartyg kan ske antingen manuellt eller automatiskt. Manuellt: Tryck E. Positionerna matas sedan in med R/S för den största först och sedan "treorna" och sist "tvåorna". Automatiskt: tryck D så läggs båtarna ut slumpmässigt. För att skjuta: ange positionen och tryck A. För att erhålla svarsskott: tryck B (displayen skall vara 0). Svarsskotten söker slumpmässigt i diagonala rader.

000	47 CMS	060	08 08	120	01 1	180	12 B	240	03 03	300	09 09
001	42 STO	061	59 FIX	121	95 =	181	09 CP	241	44 SUM	301	01 1
002	09 09	062	01 01	122	72 ST*	182	87 IFF	242	14 14	302	94 +/-
003	00 0	063	25 CLR	123	00 00	183	00 00	243	01 1	303	49 PRD
004	08 8	064	92 RTN	124	69 OP	184	02 02	244	00 0	304	15 3
005	00 0	065	76 LBL	125	20 20	185	15 15	245	32 X:T	305	01 1
006	00 0	066	18 C'	126	97 DSZ	186	67 EQ	246	77 GE	306	00 0
007	02 2	067	43 RCL	127	03 03	187	02 02	247	02 02	307	00 0
008	94 +/-	068	09 09	128	01 01	188	57 57	248	19 19	308	65 ×
009	42 STO	069	28 LOG	129	04 04	189	42 STO	249	29 CP	309	43 RCL
010	05 05	070	33 X²	130	93 .	190	05 05	250	77 GE	310	10 10
011	02 2	071	22 INV	131	01 1	191	86 STF	251	02 02	311	22 INV
012	42 STO	072	59 INT	132	44 SUM	192	00 00	252	56 56	312	59 INT
013	03 03	073	42 STO	133	14 14	193	01 1	253	61 GTO	313	95 =
014	02 2	074	09 09	134	97 DSZ	194	42 STO	254	02 02	314	22 INV
015	00 0	075	92 RTN	135	02 02	195	14 14	255	19 19	315	5'' INT
016	42 STO	076	76 LBL	136	00 00	196	42 STO	256	91 R/S	316	65 ×
017	00 00	077	15 E	137	97 97	197	10 10	257	22 INV	317	01 1
018	93 .	078	03 3	138	97 DSZ	198	94 +/-	258	86 STF	318	00 0
019	09 9	079	00 0	139	01 01	199	42 STO	259	00 00	319	00 0
020	08 8	080	42 STO	140	00 00	200	11 11	260	97 DSZ	320	95 =
021	02 2	081	00 00	141	85 85	201	93 .	261	04 04	321	59 INT
022	42 STO	082	03 3	142	25 CLR	202	01 1	262	03 03	322	55 ÷
023	02 02	083	42 STO	143	91 R/S	203	42 STO	263	29 29	323	01 1
024	05 5	084	01 01	144	76 LBL	204	12 12	264	18 C'	324	00 0
025	42 STO	085	93 .	145	11 A	205	94 +/-	265	65 ×	325	95 =
026	01 01	086	01 1	146	55 ÷	206	42 STO	266	01 1	326	42 STO
027	43 RCL	087	42 STO	147	01 01	207	13 13	267	00 0	327	07 07
028	05 05	088	04 04	148	00 0	208	01 1	268	85 +	328	91 R/S
029	85 +	089	04 4	149	95 =	209	00 0	269	02 2	329	43 RCL
030	43 RCL	090	75 -	150	32 X:T	210	42 STO	270	00 0	330	15 15
031	02 02	091	43 RCL	151	03 3	211	03 03	271	95 =	331	85 +
032	95 =	092	01 01	152	00 0	212	61 GTO	272	59 INT	332	43 RCL
033	72 ST*	093	59 INT	153	42 STO	213	02 02	273	42 STO	333	07 07
034	00 00	094	95 =	154	00 00	214	27 27	274	00 00	334	95 =
035	69 OP	095	42 STO	155	01 1	215	22 INV	275	25 CLR	335	59 INT
036	20 20	096	02 02	156	06 6	216	67 EQ	276	63 EX*	336	07 07
037	97 DSZ	097	43 RCL	157	42 STO	217	02 02	277	00 00	337	91 R/S
038	01 01	098	01 01	158	01 01	218	27 27	278	67 EQ	338	76 LBL
039	00 00	099	85 +	159	73 RC*	219	69 OP	279	02 02	339	98 ADV
040	29 29	100	01 1	160	00 00	220	23 23	280	64 64	340	25 CLR
041	93 .	101	95 =	161	22 INV	221	73 RC*	281	42 STO	341	42 STO
042	00 0	102	42 STO	162	59 INT	222	03 03	282	64 64	342	05 05
043	01 1	103	03 03	163	67 EQ	223	42 STO	283	42 STO	343	93 .
044	07 7	104	43 RCL	164	01 01	224	14 14	284	04 04	344	08 8
045	09 9	105	01 01	165	74 SBR	225	69 OP	285	22 INV	345	09 9
046	42 STO	106	85 +	166	69 OP	226	25 25	286	97 DSZ	346	01 1
047	05 05	107	01 1	167	20 20	227	22 INV	287	08 08	347	01 1
048	93 .	108	85 +	168	97 DSZ	228	97 DSZ	288	98 ADV	348	94 +/-
049	08 8	109	43 RCL	169	01 01	229	05 05	289	02 2	349	42 STO
050	00 0	110	04 04	170	01 1	230	02 02	290	93 .	350	05 05
051	00 0	111	95 =	171	59 59	231	57 57	291	02 2	351	71 SBR
052	02 2	112	92 RTN	172	25 CLR	232	43 RCL	292	42 STO	352	00 00
053	97 DSZ	113	55 ÷	173	91 R/S	233	07 07	293	15 15	353	14 14
054	03 03	114	01 1	174	25 CLR	234	85 +	294	93 .	354	93 .
055	00 00	115	00 0	175	63 EX*	235	43 RCL	295	05 5	355	01 1
056	22 22	116	85 +	176	00 00	236	14 14	296	32 X:T	356	00 0
057	01 1	117	43 RCL	177	59 INT	237	95 =	297	18 C'	357	08 8
058	00 0	118	01 01	178	91 R/S	238	32 X:T	298	77 GE	358	08 8
059	42 STO	119	85 +	179	76 LBL	239	73 RC*	299	03 03	359	42 STO

360	05 05	420	69 OP								
361	02 2	421	10 10								
362	42 STO	422	42 STO								
363	03 03	423	53 53								
364	71 SBR	424	43 RCL								
365	00 00	425	50 50								
366	48 48	426	30 TAN								
367	25 CLR	427	69 OP								
368	42 STO	428	10 10								
369	24 24	429	29 CP								
370	42 STO	430	77 GE								
371	29 29	431	04 04								
372	61 GTO	432	34 34								
373	02 02	433	93 .								
374	89 89	434	01 1								
375	76 LBL	435	65 ×								
376	14 D	436	43 RCL								
377	15 E	437	53 53								
378	02 2	438	95 =								
379	42 STO	439	42 STO								
380	59 59	440	55 55								
381	18 C'	441	43 RCL								
382	65 ×	442	55 55								
383	03 3	443	44 SUM								
384	06 6	444	52 52								
385	00 00	445	43 RCL								
386	95 =	446	52 52								
387	42 STO	447	71 SBR								
388	50 50	448	01 01								
389	43 RCL	449	13 13								
390	59 59	450	66 PAU								
391	32 X:T	451	43 RCL								
392	43 RCL	452	01 01								
393	50 50	453	32 X:T								
394	37 P/R	454	43 RCL								
395	32 X:T	455	03 03								
396	85 +	456	77 GE								
397	05 5	457	04 04								
398	95 =	458	62 62								
399	59 INT	459	61 GTO								
400	85 +	460	04 04								
401	53 (	461	41 41								
402	53 (	462	18 C'								
403	32 X:T	463	65 ×								
404	85 +	464	05 5								
405	03 3	465	85 +								
406	54)	466	05 5								
407	59 INT	467	05 5								
408	55 ÷	468	95 =								
409	01 1	469	44 SUM								
410	00 0	470	50 50								
411	95 =	471	93 .								
412	42 STO	472	04 4								
413	52 52	473	44 SUM								
414	71 SBR	474	59 59								
415	01 01	475	61 GTO								
416	13 13	476	03 03								
417	43 RCL	477	89 89								
418	50 50										
419	38 SIN										

PS

I allra sista minuten innan vi lämnar manus till tryckning kom ett telefonsamtal från Hans Nielsen i den danska Program-klubben. Han har kontakt med en konstruktör av en programmerbar modul till TI-59. Den skulle omfatta 10.000 steg och kunna programmeras individuellt. Den skulle kosta max 1.800 skr men redan med en upplaga av 50 st skulle priset kunna gå ned till ca 1.300 kr. Dyrt, men kanske ändå intressant? Tänk på saken och skriv om Du känner för den! Vi tar förstås in den vidare information vi får från Danmark. Men konstruktören vill också gärna veta vårt intresse. PPS: Maurice Swinnen, tillfrågad per telefon, trodde intresset skulle kunna vara stort i USA. OBS! INGET APRILSKÄMT!!!

Björn säger att autoplaceringen inte är särskilt avancerad därför att han ville ha med allt på ett kort, och med någon erfarenhet kan man kanske därför ana delar av placeringen. Om räknaren har fått in en träff svarar Du med "2" - "4" innan Du trycker B. Det diagonala sökandet upphör då för tillfället, och den söker runt träffen tills båten är sänkt. Svara 0 när den missar och 2 - 4 vid träff hela tiden.

Svar om HP-41 C

Efter att ha tagit del av Anders Perssons svar ville jag bara göra några kommentarer om skyddssiffrorna.

Anders P. hade räknat ut inverterade värdet av 3, som förstås blev 0.3333333333, där den sista trean inte syns på displayen. Sedan multiplicerade han med 3, och resultatet blir då 0.9999999999 . Avrundat till 9 decimaler blir detta 1.000000000 förstås, och det var detta som Anders P. såg. För att avgöra om talet verkligen är exakt ett, kan man göra på två sätt. Antingen subtraherar man 1, och ska då förstås få noll som resultat, men i detta fall blir det -1.0E-10. Eller också multiplicerar man med 10, och i detta fall får man 9.999999999 som resultat, och då får alla 10 siffrorna plats i displayen, och någon avrundning sker därför inte. Anledningen till att HP-41 beter sig så här beror på att den saknar "fritt format" på sin display, dvs ett format där alla decimaler som inte är nollor på slutet skrivs ut. Man måste alltid tala om för HP-41 hur många decimaler man vill ha på displayen (0-9 st). Och om talet är mindre än ett skrivs nollan före decimalkommat ut (ja man kan faktiskt säga åt HP-41 att skriva ett decimalkomma i stället för decimalpunkt!). HP-41 kan m.a.o. aldrig skriva ut .9999999999 utan måste skriva det som 1.000000000 avrundat.

Internt lagras varje tal i 7 bytes, med två siffror i varje byte. Talets tecken upptar 1/2 byte, sedan upptar mantissans 10 siffror 5 bytes, och slutligen upptar exponenten 1 1/2 byte. Negativa exponenter lagras i komplementform, dvs talet 0.1 lagras så här internt: 0100000000999, där 999 är exponentdelen. Ett minus lagras som en 9:a, -234 lagrat internt då som 9234000000002. Om det ligger en etta i första halva byten, betyder det att registret i fråga innehåller alfanumeriska tecken. Rent tekniskt finns det m.a.o. förutsättningar att låta exponenten variera mellan +499 och -499, men HP har i sin mikroprogrammering valt att begränsa exponenten till mellan +99 och -99. Men några fantaster i USA har faktiskt lyckats lura mikroprogrammeringen, och fått en HP-41:a att räkna inom hela exponentområdet +499 - -499. På TI-59 och andra TI-räknare lagras talen internt i 8 bytes. 1/2 byte rymmer både talets och exponentens tecken, en byte rymmer exponentens två siffror, och sedan finns det 6 1/2 bytes kvar för mantissan, vilket räcker till att lagra 13 siffror. Men TI:s AOS-system kräver att operationsstacken inte bara kommer ihåg en siffra utan också vilken operation (plus, minus etc) som ska utföras på denna siffra - och då går det åt 1/2 byte till. Operationsstacken klarar därför bara av att komma ihåg 12 siffror i mantissan. *[Se red. anm.]* HP:s operationsstack behöver inte denna information, eftersom man i RPN själv måste hålla reda på vilken operation som ska utföras just då.

I HP:s mikroprogrammering finns det egentligen bara en genuin trigonometrisk funktion - TAN. Både SIN och COS använder TAN som en subrutin, och när TAN väl är uträknad, går det ganska snabbt att beräkna både SIN och COS. Och när man beräknar P-R behöver man ju både SIN och COS på en gång. Man beräknar då TAN, och omvandlar TAN först till SIN, sedan till COS. P-R är på HP-41 mycket riktigt skriven i mikrokod. Gissningsvis är mikrokoden för SIN och COS på TI-59 i princip ganska lik motsvarande kod i HP-41. Men P/R på TI-59 är inte skriven i mikrokod, utan använder först SIN och sedan COS , och båda dessa använder i sin tur troligen TAN. Resultatet blir att när man på TI-59 gör P/R-omvandlingen, får den stackars TI-59-an beräkna TAN för samma vinkel två gånger!! Och det är TAN-beräkningen som tar tid - allt det övriga går relativt snabbt att räkna ut.

Det är inte alltid som regeln "ju mer ström en räknedosa drar, ju snabbare är den" gäller. HP-41 drar t ex bara bräkdelen så mycket ström som HP-67, men är ändå ca dubbelt så snabb på små enkla benchmarks. Och HP-41 är ett par år yngre än TI-59, och utvecklingen på elektronikområdet hinner ju en del på några år. Men vi kunde ju göra ett enkelt prov. Följande lilla slinga

```
01 100
02 LBL 00
03 DSE X
04 GTO 00
05 END
```

som snurrar runt 100 varv, tar det min HP-41:a 9,5 sekunder att utföra (hastigheten kan variera ca 10 % om man jämför olika HP-41:or). Hur lång tid tar det att köra motsvarande slinga på en TI-59:a? På HP-41 tar slingan inte ett dugg längre tid att köra, även om den befinner sig i ett mycket stort program. Och på HP-41 kan man bara hoppa till labels.

En fickdator med Pascal som språk lär det nog dröja innan vi får se. Alla dagens mikrodatorer med mer kraftfulla språk än BASIC har all eller stor del av systemet på flexskivor, och sedan läser man in det man behöver för tillfället i minnet, t ex en texteditor eller en kompilator. Så för att en fickdator ska kunna ha Pascal som språk måste den antingen ha mycket stort RAM-minne, 200 - 500 K, eller också en flexskiveenhet, likaså den i fickformat och batteridriven. Men det finns ju andra "små" språk än BASIC - Quasar/Panasonic lär hålla på att ta fram FORTH till sin fickdator. Och LISP lär kunna få plats i ganska litet minnesutrymme ("litet" betyder här högst ca 40 K).

Paul Schlyter
Nybrogatan 75 E
114 40 Stockholm

Red. anm. Det där med att operationsstacken bara skulle kunna komma ihåg 12 siffror stämmer inte. Gör operationen 1 INV lnx + 1 = CLR. (1 INV lnx dvs e har 13 signifikanta siffror.) HIR 11 - 1 INV lnx = ger noll och inget annat. HIR-register 1 har alltså kommit ihåg alla 13 siffrorna.

Avancerade programmeringsmetoder -2

av BJÖRN GUSTAVSSON

Den första artikeln i PB 81-4 s 12-14 handlade om sekventiell och binär sökning. Nu går vi vidare med mer avancerade metoder. Den första artikeln ligger till grund och bör därför ha lästs.

Hashing

De två tidigare metoderna har ett nära samband med hur människor söker efter något i en tabell. Det finns en mycket bättre metod, som är olämplig för människor, men som utnyttjar en dators eller räknares förmåga att göra snabba beräkningar.

Man använder argumentet K i en aritmetisk beräkning. Resultatet som fås blir adressen till K. Man beräknar alltså en funktion f(K).

Ett exempel: I tabell 1 finns tio svenska ord. Programmet i figur 1 är gjort så att det gör om vart och ett av de tio orden till ett unikt tal mellan 1 och 11. Orden har lagrats i motsvarande position i tabellen. BIL t ex har matats in som 14 24 27 00 00 och programmet i figur 1 har körts. Resultatet blev 5, och därför lagrades ordet BIL i position 5. Alla ord har matats in som tiosiffriga skrivkoder, med efterföljande nollor om nödvändigt.

Helt naturligt blir den här metoden snabbare än de tidigare. Endast en enda beräkning behöver ju göras för att få adressen till en nyckel, oavsett både tabellens storlek och nyckelns placering i tabellen. Söktiden vid användning av de andra metoderna är beroende på tabellens storlek.

Tyvärr finns det få funktioner med egenskapen att alla nycklar ger olika adresser. Det kan alltså ta tid att hitta en lämplig funktion. Funktionen i figur 1 hittades efter några timmar av ett program som letade efter funktioner. Om man inte heller packar tabellen lika full som tabell 1 är det lättare att hitta en funktion.

Men den här metoden har ett allvarligt fel. Antag att vi vill sätta in en ny nyckel i tabell 1, vars enda tomma position är position 10. Risken är stor att den nya nyckeln ger ett funktionsvärde skilt från 10. Om detta händer är tabellen förstörd och en ny funktion måste hittas.

Även om man inte i praktiken fyller en tabell till mer än 80-90 procent, är ändå risken stor att tabellen blir förstörd om några nycklar läggs till. Det är tydligen nödvändigt att känna hela tabellens innehåll innan man letar rätt på en funktion.

Om vi i stället tillåter att olika nycklar ger samma funktionsvärde och finner en metod för att ändå få det att fungera får vi en mer flexibel metod.

Metoden kallas "hashing". (Verbet "hash" betyder "finhacka", och metoden kan sägas bestå i att man "finhackar nycklarna och använder delar av nycklarna för adressen".) Vi beräknar en hash-funktion h(K) och använder det värdet som startregister vid sökningen.

Troligen kommer det att finnas nycklar för vilka det gäller att $K_i \neq K_j$, men $h(K_i)=h(K_j)$. Detta kallas kollision, och det finns olika metoder att hantera sådana. Vi skall titta på en sådan metod.

Men först skall vi titta på hur en hash-funktion kan väljas. Det skall gälla att $1 \leq h(K) \leq M$, där M är antal register som vi vill använda till tabellen. Enklast är divisionsmetoden:

h(K)=(K mod M)+1

där K mod M betyder "resten när K delas med M". Detta kan lätt programmeras som

- (CE DIV M) Int x M =

1	SPIS	65	×	04	4
2	BUSKE	93	.	95	=
3	BUSS	00	0	22	INV
4	FYSIK	08	8	59	INT
5	BIL	05	5	65	×
6	INNE	05	5	01	1
7	NATUR	06	6	01	1
8	KAMIN	09	9	85	+
9	HUS	09	9	01	1
10		00	0	95	=
11	UTE	08	8	59	INT

Figur 1.

Tabell 1.

Det är viktigt att välja ett värde på M så att h(K) beror på alla siffror i K. M=100 är klart olämpligt, eftersom h(K) då blir de två sista siffrorna i K ökade med 1. Allmänt gäller på TI-58/59 att jämna M-värden och M-värden delbara med 5 är olämpliga. Helst bör M vara ett primtal med vissa villkor uppfyllda, men på TI-58/59 har man inte alltid så stor valfrihet, och ett udda M-värde ej delbart med 5 är acceptabelt.

När man har en lämpligt hash-funktion gäller det att konstruera en algoritm som kan hantera kollisioner. Enklast är att söka efter K i registren med numren: h(K), h(K)-1, h(K)-2... 1, M, M-1... h(K)+1, tills antingen K hittas eller ett tomt register hittas. Följande algoritm gör detta.

Algoritm H (Sökning i hash-tabell). Den här algoritmen söker efter en given nyckel K i en tabell med M element, betecknade R_1, R_2, R_3... R_M. Ett tomt element betecknas med 0.

H1. [Initiera.] Sätt i=h(K). (Nu är $1 \leq i \leq M$.)
H2. [Misslyckad?] Om R_i=0, skriv 0 och avbryt. (Sökningen misslyckad.)
H3. [Lyckad?] Om R_i=K, skriv i och avbryt. (Sökningen lyckad.)
H4. [Nästa nyckel.] Sätt i=i-1; om i=0, sätt i=M. Gå tillbaka till H2.

För att sätta in en nyckel i tabellen utförs först algoritm H och sedan sätts R_i=K.

Utgående från algoritm H har jag gjort pgm 3. E' ställer in rätt uppdelning och raderar alla register. A söker efter en nyckel. B sätter in en nyckel i tabellen.

För att klargöra litet tydligare hur "hashing" fungerar skall vi titta på ett exempel. Tio svenska ord finns i tabell 2. Dessa skall sättas in i en liten hash-tabell med plats för 11 ord. Som hash-funktion används då h(K)=(K mod 11)+1, dvs resten när K delas med 11 ökad med 1. Detta beräknas med sekvensen - (CE DIV 11) Int x 11 =.

ENHET (5)	1	SISTA (4)	1	HOPPA (3)
METER (5)	2	RINGA (5)	2	PACKA (2)
PILOT (11)	3	HOPPA (3)	3	METER (5)
HOPPA (3)	4	METER (5)	4	SISTA (4)
RINGA (5)	5	ENHET (5)	5	RINGA (5)
MINUT (10)	6	OFFER (6)	6	OFFER (6)
SISTA (4)	7	STACK (7)	7	STACK (7)
PACKA (2)	8		8	
OFFER (6)	9	PACKA (2)	9	ENHET (5)
STACK (7)	10	MINUT (10)	10	MINUT (10)
	11	PILOT (11)	11	PILOT (11)

Tabell 2. Tabell 3. Tabell 4.

Pgm 3 kan användas om 99 i stegen 024-023, 029-030 samt 052-053 ändras till 11. Resultatet av insättningen visas i tabell 3. Siffrorna inom parentes anger hash-adress.

Orden ENHET, METER och RINGA har alla hash-adressen 5. Eftersom ENHET sattes in först var adress 5 ledig. Men när sedan METER sattes in var position 5 upptagen, och ordet försköts till position 4. När RINGA sattes in var även position 3 upptagen (av HOPPA), så det ordet hamnade i position 2.

De flesta orden har faktiskt lagrats nära sina hash-adresser. Ordet PACKA har förskjutits längst: från 2 till 9, men PACKA sattes också in ganska sent.

Vad kan vi vänta oss för söktider? Genomsnittligt är "hashing" snabbare än t ex binär sökning. Tiden för sökning i en hash-tabell är beroende på hur full tabellen är, inte hur stor den är. Man bör aldrig låta en hash-tabell bli lika full som tabell 3, utan bara låta den bli fylld till 80-90 procent.

```
000  76 LBL     018  82 HIR    ▷036  73 RC*     054  61 GTO
001  10 E'      019  04  04     037  00  00     055  00  00
002  25 CLR     020  75  -      038  29 CP      056  34  34
003  01  1      021  53  (      039  67  EQ    ▷057  43 RCL
004  00  0      022  24 CE      040  00  00     058  00  00
005  69 OP      023  55  ÷      041  59  59    ▷059  99 PRT
006  17  17     024  09  9      042  32 X:T     060  92 RTN
007  47 CMS     025  09  9      043  82 HIR     061  76 LBL
008  92 RTN     026  54  )      044  14  14     062  12  B
009  76 LBL     027  59 INT     045  67  EQ     063  11  A
010  11  A      028  65  ×      046  00  00     064  82 HIR
011  98 ADV     029  09  9      047  57  57     065  14  14
012  69 OP      030  09  9      048  97 DSZ     066  72 ST*
013  00  00     031  85  +      049  00  00     067  00  00
014  69 OP      032  01  1      050  00  00     068  00  0
015  02  02     033  95  =      051  36  36     069  92 RTN
016  69 OP     ▷034  42 STO     052  09  9
017  05  05      035  00  00     053  09  9
```

Pgm 3. Sökning i hash-tabell.

```
000  76 LBL     023  09  9      046  00  00     069  00  00
001  10 E'      024  09  9      047  57  57     070  57  57
002  25 CLR     025  54  )      048  97 DSZ     071  77  GE
003  01  1      026  59 INT     049  00  00     072  00  00
004  00  0      027  65  ×      050  00  00     073  77  77
005  69 OP      028  09  9      051  39  39     074  32 X:T
006  17  17     029  09  9      052  09  9      075  72 ST*
007  47 CMS     030  85  +      053  09  9      076  00  00
008  92 RTN     031  01  1      054  61 GTO    ▷077  00  0
009  76 LBL     032  95  =      055  00  00     078  67  EQ
010  16 A'      033  92 RTN     056  32  X:T    079  00  00
011  98 ADV     034  76 LBL    ▷057  43 RCL     080  59  59
012  69 OP      035  11  A     ▷058  00  00     081  97 DSZ
013  00  00     036  16 A'     ▷059  99 PRT     082  00  00
014  69 OP     ▷037  42 STO      060  92 RTN    083  00  00
015  02  02     038  00  00      061  76 LBL    084  66  66
016  69 OP     ▷039  73 RC*      062  12  B     085  09  9
017  05  05      040  00  00     063  16 A'     086  9   9
018  75  -      041  22 INV    ▷064  42 STO     087  61 GTO
019  53  (      042  77  GE     ▷065  00  00     088  00  00
020  24 CE      043  00  00     ▷066  73 RC*     089  64  64
021  55  ÷      044  58  58      067  00  00
022  32 X:T     045  67  EQ      068  67  EQ
```

Pgm 4. Sökning i ordnad hash-tabell.

Allmänt gäller också att misslyckade sökningar tar längre tid än lyckade. En misslyckad sökning kan bara sluta vid en tom position. En misslyckad sökning i tabell 3 måste sluta vid position 8, oavsett värdet på h(K).

Det går emellertid att snabba upp misslyckade sökningar. Antag att nycklarna sätts in i tabellen i omvänd alfabetisk ordning. För alla nycklar K_i gäller då, att alla nycklar mellan adressen $h(K_i)$ och den verkliga positionen för K_i måste vara alfabetiskt större än K_i. En sökning efter K_i kan då avbrytas om en nyckel som är alfabetiskt mindre än K_i påträffas.

Tabell 4 har skapats genom att sätta in nyclarna från tabell 2 i en ny tabell i omvänd alfabetisk ordning; alltså i ordningen STACK, SISTA, RINGA, PILOT... HOPPA, ENHET.

Läsaren bör själv prova att skapa tabellerna 3 och 4 utgående från tabell 2 för att bättre förstå hur det fungerar.

För att söka i tabell 4 kan följande algoritm användas.

Algoritm O (Sökning i ordnad hash-tabell). Den här algoritm söker efter en given nyckel K i en ordnad hash-tabell.

O1. [Initiera.] Sätt i←h(K).
O2. [Misslyckad?] Om R_i<K, skriv O och avbryt. (Sökningen misslyckad.)
O3. [Lyckad?] Om R_i=K_i, skriv i och avbryt. (Sökningen lyckad.)
O4. [Nästa nyckel.] Sätt i←i-1. Om i=0, sätt i←M. Gå tillbaka till steg O2.

Denna algoritmen fungerar bra, vilket läsaren kan övertyga sig om genom att genomföra algoritmen i huvudet, om nycklarna har satts in som i tabell 4. Men antag att man vill sätta in fler ord i en ordnad hash-tabell. Är det möjligt? Ja, följande algoritm klarar det.

Algoritm I (Insättning i en ordnad hash-tabell). Den här algoritmen sätter in en ny nyckel K i en ordnad hash-tabell.

I1. [Initiera.] Sätt i←h(x).
I2. [Jämför.] Om R_i<K, skifta värdena på R_i och K.
I3. [Färdigt?] Om K=0, avbryt.
I4. [Nästa.] Sätt i←i-1. Om i=0, sätt i←M. Gå tillbaka till steg I2.

Pgm 4 har gjorts efter dessa algoritmer. Lablarna har sedvanliga betydelser. Observera att sökrutinen faktiskt är kortare än motsvarande i pgm 3.

En tillämpning: Morse-program

Morse-alfabetet består av punkter och streck, eller korta och långa signaler. Se tabell 5. Vi skall här se hur ett program som omvandlar mellan morse och vanliga tecken, och tvärtom, kan göras.

Vi väljer att representera tecknen med den vanliga skrivkodstabellens koder. Då betyder t ex 13 A, 14 B etc. Vi vill också kunna omvandla bokstäverna Å, Ä och Ö, och därför låter vi dessa tecken få koderna 47-49. (47=Å, 48=Ä, 49=Ö.) Dessa skrivs inte ut riktigt på skrivaren, men det kan inte hjälpas.

Morse-koderna beteckna så här: 1 betyder punkt och 3 betyder streck. Exempel: -... (betyder B) betecknas med 3111.

Då övergår vi till programuppbyggnaden. Första uppgiften för programmet är att omvandla ett tecken till morse-kod. Om vi tillåter skrivkoderna 1-49 kan morse-koderna lagras i motsvarande register och återkallas indirekt. Siffrorna och bokstäverna, samt "-" och "." som råkar ha koder mellan 1-49 kan då omvandlas.

I pgm 5 har i stället R_{51}-R_{99} använts, och då läggs först 50 till skrivkoden innan indirekt återkallning utförs.

Nästa uppgift är att omvandla från morse-kod till skrivkod. I det här fallet måste en sökning göras, eftersom vissa morse-koder är ensiffriga och andra sexsiffriga. Om man söker sekventiellt kan samma tabell som används vid omvandling från tecken till morse-kod användas. Jag har sett program som gör så, resulterande i exekveringstider på uppemot 30 sekunder.

I pgm 5 har i stället en ordnad hash-tabell använts. Den ligger i R_1-R_{49}, och det är anledningen till att den andra tabellen ligger i R_{51}-R_{99}. Sökningen tar ofta under 5 sekunder. Morse-koden består av heltalsdelen av talet i registret och skrivkoden består av decimaldelen. Vid själva sökningen ignoreras decimaldelen.

Programmet innehåller också rutiner för utmatning på både skrivare och till display. Om skrivare är ansluten skrivs både morse-kod och tecken ut. Utan skrivare visas morse-kod eller skrivkod i displayen, beroende på vilken omvandling som har gjorts.

Programmet körs så här:

1. Initiera med E'.
2a. Mata in en skrivkod och tryck A. Morse-kod visas och trycks ut.
2b. Mata in en morse-kod och tryck B. Skrivkoden visas och trycks ut. Om morsekoden inte finns visas i stället 9.9999999 99.

Detta var allt för den här gången. Jag har ännu inte bestämt slutgiltigt vad nästa artikel kommer att handla om, men jag har funderingar på sorteringsmetoder. Förslag och synpunkter från läsarna är välkomna.

▷000	69	DP	022	05	5	044	04	4	066	09	9
001	04	04	023	00	0	045	09	9	067	61	GTO
002	32	XIT	024	44	SUM	046	85	+	068	00	00
003	69	DP	025	00	00	047	01	1	069	49	49
004	06	06	026	73	RC+	048	95	=	▷070	73	RC+
005	43	RCL	027	00	00	▷049	42	STO	071	00	00
006	00	00	028	42	STO	050	00	00	072	22	INV
007	42	STO	029	00	00	▷051	73	RC+	073	59	INT
008	76	LBL	030	32	XIT	052	00	00	074	65	×
009	10	E'	031	81	RST	053	59	INT	075	01	1
010	25	CLR	032	76	LBL	054	22	INV	076	00	0
011	01	1	033	12	B	055	77	GE	077	00	0
012	00	0	034	75	-	056	00	00	078	95	=
013	69	OP	035	53	(	057	82	82	079	42	STO
014	17	17	036	24	CE	058	67	EQ	080	00	00
015	98	ADV	037	55	÷	059	00	00	081	81	RST
016	92	RTN	038	32	XIT	060	70	70	▷082	00	0
017	76	LBL	039	04	4	061	97	DSZ	083	35	1/X
018	11	A	040	09	9	062	00	00	084	92	RTN
019	42	STO	041	54	)	063	00	00			
020	00	00	042	59	INT	064	51	51			
021	32	XIT	043	65	×	065	04	4			

0.	01	1113.42	34	1.	67
1.17	02	13313.47	35	33111.	68
3.37	03	1311.27	36	33311.	69
11.24	04	33111.1	37	311113.	70
1131.21	05	11111.06	38	1131.	71
13333.02	06	1313.48	39	331.	72
13.13	07	11113.05	40	1111.	73
111.36	08	33311.11	41	11.	74
1331.33	09	0.	42	1333.	75
1333.25	10	131313.4	43	313.	76
11133.04	11	0.	44	1311.	77
33331.12	12	3131.15	45	33.	78
311113.2	13	31111.07	46	31.	79
33333.01	14	3133.45	47	33.	80
11333.03	15	0.	48	31.	81
113.41	16	3331.49	49	333.	82
0.	17	0.	50	1331.	83
311.16	18	33333.	51	3313.	84
0.	19	13333.	52	131.	85
313.26	20	11333.	53	111.	86
0.	21	11133.	54	3.	87
0.	22	11113.	55	33.	88
31.31	23	11111.	56	31.	89
33.3	24	31111.	57	131313.	90
3111.14	25	33111.	58	113.	91
131.35	26	33311.	59	1113.	92
3113.44	27	33111.	60	133.	93
133.43	28	33311.	61	3113.	94
3311.46	29	33331.	62	3133.	95
331.22	30	13.	63	3311.	96
3313.34	31	3111.	64	13313.	97
333.32	32	3131.	65	1313.	98
1111.23	33	311.	66	3331.	99

Pgm 5. Morse-program.

Å,Ä och Ö på PC 100 !

När jag nyligen experimenterade med hexkoder hittade jag plötsligt en sekvens av vanliga instruktioner och en hexkod, som gav mycket underliga utskrifter på skrivaren. Jag började göra mer systematiska undersökningar och lyckades till slut skriva Ä och Ö! Vid fortsatta experiment lyckades jag också få fram något som liknar Å. Det är också möjligt att det går att få fram en mer högupplösande grafisk mode, men det har jag ännu inte hunnit undersöka.

Däremot har jag gjort ett program som skriver ut mitt namn riktigt:

BJÖRN GUSTAVSSON

Det är fortfarande några detaljer i räknarens beteende som jag inte förstår, men jag hoppas att ha löst det till nästa nummer.

Under väntetiden kan du prova programmet nedan.

Bruksanvisning:

1. Mata in programmet.
2. Tryck GTO 016 9 Op 17 CLR Pgm 12 SBR 444 R/S D.MS LRN Ins LRN RST A. Detta sätter in hexkoden och startar programmet.

000	92	RTN	033	07	7	066	01	1	099	04	4
001	76	LBL	034	01	1	067	02	2	100	01	1
002	11	A	035	06	6	068	08	8	101	03	3
003	61	GTO	036	42	STO	069	95	=	102	06	6
004	00	00	037	02	02	070	69	OP	103	03	3
005	27	27	038	33	X²	071	02	02	104	07	7
006	68	NOP	039	65	×	072	43	RCL	105	69	OP
007	68	NOP	040	05	5	073	02	02	106	02	02
008	76	LBL	041	01	1	074	65	×	107	01	1
009	17	B'	042	33	X²	075	04	4	108	03	3
010	69	OP	043	75	-	076	75	-	109	04	4
011	05	05	044	06	6	077	09	9	110	02	2
012	26	2ND	045	05	5	078	01	1	111	03	3
013	56	DEL	046	08	8	079	95	=	112	06	6
014	88	DMS	047	02	2	080	69	OP	113	03	3
015	21	2ND	048	09	9	081	03	03	114	06	6
016	53	(	049	42	STO	082	17	B'	115	03	3
017	00	0	050	01	01	083	01	1	116	02	2
018	00	0	051	95	=	084	04	4	117	69	OP
019	00	0	052	69	OP	085	02	2	118	03	03
020	00	0	053	01	01	086	05	5	119	03	3
021	00	0	054	09	9	087	03	3	120	01	1
022	00	0	055	07	7	088	02	2	121	69	OP
023	00	0	056	05	5	089	03	3	122	04	04
024	00	0	057	07	7	090	05	5	123	69	OP
025	00	0	058	85	+	091	03	3	124	05	05
026	25	CLR	059	43	RCL	092	01	1	125	58	FIX
027	47	CMS	060	01	01	093	69	OP	126	09	09
028	58	FIX	061	95	=	094	01	01	127	25	CLR
029	08	08	062	33	X²	095	08	8	128	92	RTN
030	69	OP	063	85	+	096	00	0			
031	00	00	064	09	9	097	02	2			
032	98	ADV	065	00	0	098	02	2			

Kortaste avstånd mellan korsande räta linjer

av GÖSTA BLUME

Två räta linjer som ligger i ett plan måste antingen vara parallella eller skära varandra. I det tredimensionella rummet tillkommer ett tredje alternativ: linjerna kan korsa varandra på samma sätt som två godtyckliga linjer, en på vardera av två motsatta väggar i ett rum,gör om de inte råkar vara parallella, eller som två godtyckliga linjer, en på vardera av två sammanstötande väggar, gör om de inte skär varandra i hörnet mellan väggarna.

Det händer väl inte så ofta, att man behöver känna det kortaste avståndet mellan två korsande linjer. Men om det händer är det inte så lätt att hitta några färdiga formler eller beräkningsmetoder. Och problemet hör inte till de allra enklaste att knäcka på egen hand och även om man lyckas med det så blir räkningarna långa och besvärliga. Här är ett pgm som klarar saken.

-oOo-

I rymdanalysen beskrivs en rät linje vanligen av två ekv så här

$$\frac{x-x_1}{l} = \frac{y-y_1}{m} = \frac{z-z_1}{n}$$

där (x_1,y_1,z_1) är koordinaterna för en given punkt på linjen. l, m och n är riktningskoefficienter, som är proportionella mot cosinerna för vinklarna mellan linjen och de tre koordinataxlarna. Känner man två punkter på linjen, kan man beräkna dessa koefficienter som $l=x_2-x_1$, $m=y_2-y_1$, $n=z_2-z_1$.

I dessa ekv är (x,y,z) en löpande punkt v.s.h. på linjen. En bestämd punkt kan definieras genom att man sätter vart och ett av de lika stora bråken = en fast kvantitet, alltså t ex

$$\frac{x'-x_1}{l} = \frac{y'-y_1}{m} = \frac{z'-z_1}{n} = k$$

där $x' = kl + x_1$ är den definierade punktens x-koordinat, och motsvarande för y' och z'. Genom att ändra k kan vi alltså flytta punkten som vi vill på linjen.

Antag nu, att vi har två räta linjer, L_1 och L_2, vardera bestämd av två punkter, P_{11} och P_{12} på L_1 och P_{21} och P_{22} på L_2. Vi söker de punkter Q_1 och Q_2 på linjerna, mellan vilka avståndet är minst.

Den fullständiga härledningen av alla formlerna är rätt utrymmeskrävande, varför jag nöjer mej med att skissera problemets behandling.

Med lätt insedda beteckningar har vi de båda linjernas ekvationer

$$\frac{\xi_1-x_{11}}{l_1} = \frac{\eta_1-y_{11}}{m_1} = \frac{\zeta_1-z_{11}}{n_1} = k_1$$

resp.

$$\frac{\xi_2-x_{21}}{l_2} = \frac{\eta_2-y_{21}}{m_2} = \frac{\zeta_2-z_{21}}{n_2} = k_2$$

där, för vissa, ännu okända k-värden, ξ_1, η_1, ζ_1 är koordinaterna för punkterna Q_1 och Q_2.

Avståndet mellan Q_1 och Q_2,

$$d = \sqrt{(\xi_1-\xi_2)^2+(\eta_1-\eta_2)^2+(\zeta_1-\zeta_2)^2}$$

är nu en funktion av enbart k_1 och k_2 och vi har alltså att söka de k-värden som gör d så litet som möjligt. Vi beräknar därför $\partial d/\partial k_1$ och $\partial d/\partial k_2$ och sätter båda dessa partiella derivator = 0. Vi får då ett system av två linjära ekv, varur k_1 och k_2 kan beräknas. Hela räkningen är, trots sin principiella enkelhet, så pass trasslig, att man för att inte tappa bort sej måste införa hjälpstorheter i flera nivåer. Uttryckt i sådana storheter kan ekv-systemets lösning skrivas

$$k_1 = \frac{a_{22}h_1 - a_{12}h_2}{a_{11}a_{22}-a_{12}^2}\ ;\ k_2 = \frac{a_{12}h_1 - a_{11}h_2}{a_{11}a_{22}-a_{12}^2}$$

där

$$a_{ij} = l_i l_j + m_i m_j + n_i n_j$$

$$h_i = -[l_i(x_{11}-x_{21}) + m_i(y_{11}-y_{21}) + n_i(z_{11}-z_{21})]$$

$$l_i = x_{i2}-x_{i1} \quad m_i = y_{i2}-y_{i1} \quad n_i = z_{i2}-z_{i1}$$

$$\xi_i = k_i l_i + x_{i1} \quad \eta_i = k_i m_i + y_{i1} \quad \zeta_i = k_i n_i + z_{i1}$$

(i,j = 1,2)

En enkel koll på resultatet av avståndsbestämningen kan vi få genom att beräkna de båda kvantiteterna

$$c_i = l_i\lambda + m_i\mu + n_i\nu\ ,\quad (i = 1, 2)$$

där λ, μ, och ν är riktningskoefficienterna för sträckan Q_1Q_2 ($\lambda = \xi_2 - \xi_1$ etc.). c_i är nämligen proportionell mot cosinus för vinkeln mellan Q_1Q_2 och L_i. Båda vinklarna skall emellertid vara räta, alltså deras cosiner = 0 och följaktligen också de båda c_i=0.

-oOo-

Pgm för TI-59 med PC-100 kan se ut som i fig 1.

Handhavandet blir detta:

Slå in x_{11} på A och y_{11}, z_{11}, x_{21}, y_{21} och z_{21} på R/S.
(Obs P_{11} på L_1, P_{21} på L_2! likgiltigt vilken punkt på resp linje som utses till P_{11})
Slå på motsvarande sätt in P_{12} och P_{22} på B
Tryck C: d, Q_1 (3 koordinater), Q_2 (dito) skrivs ut
Tryck (om så önskas) D, varvid 90° (inom avrundningsnoggrannhetens gränser) skall visas två gånger för koll av vinkeln mellan Q_1Q_2 och linjerna.
Registerreview (fig 2) kan fås med E.

Som exempel väljes (fig 3) en huvuddiagonal AB och en sidodiagonal CD i en rätvinklig parallellepiped. Vi väljer (godtyckligt) A till P_{11} och C till P_{21}.

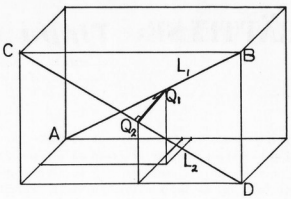


Fig 3

De fyra punkternas koordinater må vara P_{11}(0/0/0), P_{12}(4/10/6), P_{21}(4/0/6) och P_{22}(4/10/0).
P_{11}och P_{21} slås in på A, P_{12} och P_{22} på B. Tryckning på C ger d = 1,864... samt koordinaterna för Q_1 och Q_2, + visas som listning av R1 - R7 -
D visar 90° två gånger.
Registren R8 - R20, som innehåller värden på hjälpvariablerna, listas efter tryckning på E.
Sträckan Q_1Q_2 är inritad i fig 3.

N.B. Betr. listningen i pos 282 - 334 se PB81-2 sidan 3.

blm

Fig 2

1.864109298	d	01
2.262548263	ξ_1	02
5.656370656	η_1	03
3.393822394	ζ_1	04
4.	ξ_2	05
5.308880309	η_2	06
2.814671815	ζ_2	07
4.	l_1	08
10.	m_1	09
6.	n_1	10
0.	l_2	11
10.	m_2	12
-6.	n_2	13
152.	a_{11}	14
136.	a_{22}	15
64.	a_{12}	16
52.	h_1	17
-36.	h_2	18
.5656370656	k_1	19
.5308880309	k_2	20

000	76 LBL	050	33 X²	100	06 06	150	16 16	200	65 ×	250	43 RCL	300	02 02	350	10 10
001	11 A	051	85 +	101	75 -	151	65 ×	201	43 RCL	251	06 06	301	82 82	351	65 ×
002	32 X:T	052	43 RCL	102	43 RCL	152	43 RCL	202	09 09	252	75 -	302	75 -	352	43 RCL
003	02 2	053	12 12	103	03 03	153	18 18	203	95 =	253	43 RCL	303	59 INT	353	28 28
004	42 STO	054	33 X²	104	54)	154	54)	204	44 SUM	254	03 03	304	42 STO	354	94 ±
005	00 00	055	85 +	105	42 STO	155	55 ÷	205	03 03	255	54)	305	59 59	355	22 INV
006	42 STO	056	43 RCL	106	22 22	156	53 (	206	43 RCL	256	42 STO	306	54)	356	39 COS
007	01 01	057	85 +	107	85 +	157	43 RCL	207	19 19	257	27 27	307	53 (	357	66 PAU
008	32 X:T	058	33 X²	108	43 RCL	158	14 14	208	65 ×	258	33 X²	308	24 CE	358	43 RCL
009	72 ST×	059	95 =	109	10 10	159	65 ×	209	43 RCL	259	85 +	309	65 ×	359	11 11
010	00 00	060	42 STO	110	65 ×	160	43 RCL	210	10 10	260	53 (	310	71 SBR	360	65 ×
011	69 OP	061	15 15	111	53 (	161	15 15	211	95 =	261	43 RCL	311	02 02	361	43 RCL
012	20 20	062	43 RCL	112	43 RCL	162	75 -	212	44 SUM	262	07 07	312	82 82	362	26 26
013	92 RTN	063	08 08	113	07 07	163	43 RCL	213	04 04	263	75 -	313	85 +	363	85 +
014	61 GTO	064	65 ×	114	75 -	164	16 16	214	43 RCL	264	43 RCL	314	43 RCL	364	43 RCL
015	00 00	065	43 RCL	115	43 RCL	165	33 X²	215	20 20	265	04 04	315	59 59	365	12 12
016	09 09	066	11 11	116	04 04	166	54)	216	65 ×	266	54)	316	54)	366	65 ×
017	76 LBL	067	85 +	117	54)	167	42 STO	217	43 RCL	267	42 STO	317	69 OP	367	43 RCL
018	12 B	068	43 RCL	118	42 STO	168	24 24	218	11 11	268	28 28	318	04 04	368	27 27
019	75 -	069	09 09	119	23 23	169	95 =	219	95 =	269	33 X²	319	73 RC×	369	85 +
020	73 RC×	070	65 ×	120	95 =	170	42 STO	220	44 SUM	270	95 =	320	58 58	370	43 RCL
021	01 01	071	43 RCL	121	42 STO	171	19 19	221	05 05	271	34 √x	321	69 OP	371	13 13
022	95 =	072	12 12	122	17 17	172	43 RCL	222	43 RCL	272	42 STO	322	06 06	372	65 ×
023	72 ST×	073	85 +	123	43 RCL	173	16 16	223	20 20	273	01 01	323	01 1	373	43 RCL
024	00 00	074	43 RCL	124	11 11	174	65 ×	224	65 ×	274	08 8	324	44 SUM	374	28 28
025	69 OP	075	10 10	125	65 ×	175	43 RCL	225	43 RCL	275	32 X:T	325	58 58	375	95 =
026	20 20	076	65 ×	126	43 RCL	176	17 17	226	12 12	276	01 1	326	43 RCL	376	22 INV
027	69 OP	077	43 RCL	127	21 21	177	75 -	227	95 =	277	42 STO	327	58 58	377	39 COS
028	21 21	078	13 13	128	85 +	178	43 RCL	228	44 SUM	278	59 59	328	22 INV	378	92 RTN
029	92 RTN	079	95 =	129	43 RCL	179	14 14	229	06 06	279	61 GTO	329	77 GE	379	76 LBL
030	61 GTO	080	42 STO	130	12 12	180	65 ×	230	43 RCL	280	02 02	330	02 02	380	15 E
031	12 B	081	16 16	131	65 ×	181	43 RCL	231	20 20	281	96 96	331	96 96	381	02 2
032	76 LBL	082	43 RCL	132	43 RCL	182	18 18	232	65 ×	282	01 1	332	98 ADV	382	01 1
033	13 C	083	08 08	133	22 22	183	54)	233	43 RCL	283	00 0	333	00 0	383	32 X:T
034	43 RCL	084	65 ×	134	85 +	184	55 ÷	234	13 13	284	49 PRD	334	92 RTN	384	61 GTO
035	08 08	085	53 (	135	43 RCL	185	43 RCL	235	95 =	285	59 59	335	76 LBL	385	02 02
036	33 X²	086	43 RCL	136	13 13	186	24 24	236	44 SUM	286	49 PRD	336	14 D	386	96 96
037	85 +	087	05 05	137	65 ×	187	95 =	237	07 07	287	59 59	337	43 RCL		
038	43 RCL	088	75 -	138	43 RCL	188	42 STO	238	53 (	288	85 +	338	08 08		
039	09 09	089	43 RCL	139	23 23	189	20 20	239	43 RCL	289	01 1	339	65 ×		
040	33 X²	090	02 02	140	95 =	190	43 RCL	240	05 05	290	05 +	340	43 RCL		
041	85 +	091	54)	141	42 STO	191	19 19	241	75 -	291	28 LOG	341	26 26		
042	43 RCL	092	42 STO	142	18 18	192	65 ×	242	43 RCL	292	59 INT	342	85 +		
043	10 10	093	21 21	143	43 RCL	193	43 RCL	243	02 02	293	65 ×	343	43 RCL		
044	33 X²	094	85 +	144	15 15	194	08 08	244	54)	294	02 2	344	09 09		
045	95 =	095	43 RCL	145	65 ×	195	95 =	245	42 STO	295	92 RTN	345	65 ×		
046	42 STO	096	09 09	146	43 RCL	196	44 SUM	246	26 26	296	53 (	346	43 RCL		
047	14 14	097	65 ×	147	17 17	197	02 02	247	33 X²	297	24 CE	347	27 27		
048	43 RCL	098	53 (	148	75 -	198	43 RCL	248	85 +	298	85 +	348	85 +		
049	11 11	099	43 RCL	149	43 RCL	199	19 19	249	53 (	299	71 SBR	349	43 RCL		

RÄTTELSE: *Tippa med TI 59*

av BERTIL WALLIN · Karlskoga

Tyvärr råkade vi göra en ordentlig tabbe när vi publicerade den här artikeln på s. 26 - 27 i nr 81-4. Hela listningen till huvudprogrammet föll ut, så det kan inte ha varit särskilt lätt att få några större tipsvinster med hjälp av artikeln.

Här kommer nu listningen tillsammans med en utskrift av det exempel som fanns i vänsterspalten på s. 27. Anvisningarna för körning på s. 26, högra spalten, gäller.

Det program som fanns på s. 27, högerspalten, gäller sannolikhetsfördelningen i huvudprogrammet. B- och H-värden från detta fordras för körning. Programmet kan användas för att analysera varje särskild match. K-värdet framgår av strategiplanen s. 26. I huvudprogrammet är den satt till 0,1. Indata ges på följande sätt: H-värde tangent E, B-värde R/S, K-värde R/S. Uppdelning 10 Op 17.

```
TIPS*V. NR          .1889822365   %HEM
.1111111111         .6713071523   %BRT
     N=?
        10.                       .4635231383   %HEM
                                  .4857022604   %BRT
.5292738139   %HEM
.3956518422   %BRT
                                  2GRD1
                                  2
.3828698548   %HEM                1GRD2
.4832788619   %BRT                2
                                  1
                                  2GRD1
.3453266907   %HEM                1
.4759242176   %BRT                2
                                  2
.4017766953   %HEM                XGRD1
.5061862178   %BRT                2
                                  1GRD2
.6839101615   %HEM                1GRDX
.2789213562   %BRT
                                  S=?
                                    0.789
.5763853992   %HEM
.3454792759   %BRT
                                  1
                                  2
.6837745848   %HEM                2          NUMERIC  REG ALPHA
.2909060279   %BRT                2   3664710000.  46 S=?
                                  2   3164710000.  47 N=?
0.677964473   %HEM                1   4240003135.  48 V. NR
        0.1   %BRT                2GRD1  3724333651.  49 TIPS*
                                  2GRD1    61231730.  50 %HEM
                                  1        61143537.  51 %BRT
      0.05    %HEM                1       222351603.  52 1GRD2
.7582482905   %BRT                1GRDX   322351602.  53 2GRD1
                                  2       200000000.  54 1
.5691423693   %HEM                2       222351650.  55 1GRDX
0.255020575   %BRT                2       300000000.  56 2
                                          322351650.  57 2GRDX
.4486067977   %HEM                        5022351602. 58 XGRD1
.4488612788   %BRT                        5022351603. 59 XGRD2
```

```
000 91 R/S    060 00 00     120 43 RCL    180 43 RCL    240 08 08     300 73 RC*    360 15 E      420 32 X:T
001 02 2      061 40 IND    121 15 15     181 10 10     241 67 EQ     301 08 08     361 43 RCL    421 95 =
002 00 0      062 17 17     122 42 STO    182 72 ST*    242 00 00     302 32 X:T    362 49 49     422 50 I×1
003 42 STO    063 85 +      123 10 10     183 08 08     243 00 00     303 43 RCL    363 08 08     423 77 GE
004 07 07     064 43 RCL    124 43 RCL    184 69 OP     244 43 RCL    304 15 15     364 01 01     424 04 04
005 73 RC*    065 15 15     125 16 16     185 07 06     245 14 14     305 77 GE     365 43 RCL    425 31 31
006 07 07     066 95 =      126 42 STO    186 69 OP     246 32 X:T    306 03 03     366 48 48     426 43 RCL
007 72 ST*    067 48 EXC    127 11 11     187 28 28     247 73 RC*    307 13 13     367 69 OP     427 55 55
008 00 00     068 15 15     128 69 OP     188 48 EXC    248 08 08     308 68 NOP    368 02 02     428 61 GTO
009 59 INT    069 55 ÷      129 30 30     189 51 SI     249 69 OP     309 68 NOP    369 69 OP     429 03 03
010 63 EX*    070 43 RCL    130 01 1      190 69 OP     250 28 28     310 61 GTO    370 08 05     430 20 20
011 00 00     071 15 15     131 03 3      191 04 04     251 85 +      311 04 04     371 69 OP     431 43 RCL
012 22 INV    072 95 =      132 08 8      192 43 RCL    252 63 EX*    312 36 36     372 00 00     432 53 ×
013 59 INT    073 48 EXC    133 42 STO    193 11 11     253 08 08     313 43 RCL    373 91 R/S    433 61 GTO
014 52 EE     074 16 16     134 17 17     194 72 ST*    254 42 STO    314 56 56     374 99 PRT    434 03 03
015 04 4      075 55 ÷      135 61 GTO    195 08 08     255 15 15     315 15 15     375 43 RCL    435 20 20
016 22 INV    076 43 RCL    136 00 00     196 69 OP     256 95 =      316 03 03     376 47 47     436 75 -
017 52 EE     077 15 15     137 39 39     197 06 06     257 42 STO    317 20 20     377 69 OP     437 93 .
018 69 OP     078 95 =      138 43 RCL    198 98 ADV    258 16 16     318 43 RCL    378 02 02     438 01 1
019 20 20     079 49 PRD    139 09 09     199 69 OP     259 73 RC*    319 58 58     379 69 OP     439 00 0
020 72 ST*    080 13 13     140 22 INV    200 28 28     260 08 08     320 69 OP     380 05 05     440 32 X:T
021 00 00     081 43 RCL    141 49 PRD    201 02 2      261 77 GE     321 02 02     381 91 R/S    441 95 =
022 69 OP     082 16 16     142 15 15     202 42 STO    262 02 02     322 69 OP     382 99 PRT    442 50 I×1
023 20 20     083 49 PRD    143 22 INV    203 05 05     263 82 82     323 05 05     383 42 STO    443 77 GE
024 69 OP     084 12 12     144 49 PRD    204 04 4      264 43 RCL    324 43 RCL    384 09 09     444 04 04
025 27 27     085 69 OP     145 16 16     205 06 6      265 16 16     325 15 15     385 98 ADV    445 51 SI
026 22 INV    086 30 30     146 43 RCL    206 32 X:T    266 77 GE     326 72 ST*    386 02 2      446 43 RCL
027 97 DSZ    087 97 DSZ    147 16 16     207 43 RCL    267 03 03     327 08 08     387 00 0      447 57 57
028 05 05     088 05 05     148 44 SUM    208 08 08     268 00 00     328 69 OP     388 42 STO    448 61 GTO
029 00 00     089 00 00     149 11 11     209 67 EQ     269 43 RCL    329 28 28     389 08 08     449 03 03
030 34 34     090 39 39     150 43 RCL    210 02 02     270 15 15     330 43 RCL    390 42 STO    450 20 20
031 61 GTO    091 43 RCL    151 15 15     211 18 18     271 32 X:T    331 14 14     391 07 07     451 43 RCL
032 00 00     092 12 12     152 44 SUM    212 22 INV    272 73 RC*    332 61 GTO    392 02 2      452 53 ×
033 05 05     093 34 FX     153 10 10     213 86 STF    273 08 08     333 02 02     393 42 STO    453 61 GTO
034 69 OP     094 42 STO    154 02 2      214 00 00     274 77 GE     334 26 26     394 05 05     454 03 03
035 30 30     095 12 12     155 22 INV    215 61 GTO    275 03 03     335 76 LBL    395 01 1      455 20 20
036 02 2      096 43 RCL    156 49 PRD    216 04 04     276 18 18     336 14 B      396 04 4      456 76 LBL
037 42 STO    097 13 13     157 10 10     217 08 08     277 43 RCL    337 72 ST*    397 32 X:T    457 13 C
038 05 05     098 34 FX     158 22 INV    218 02 2      278 59 59     338 07 07     398 01 1      458 43 RCL
039 73 RC*    099 42 STO    159 49 PRD    219 00 0      279 61 GTO    339 69 OP     399 49 PRD    459 44 44
040 00 00     100 13 13     160 11 11     220 42 STO    280 03 03     340 27 27     400 06 06     460 69 OP
041 52 EE     101 86 STF    161 43 RCL    221 08 08     281 20 20     341 22 INV    401 42 STO    461 02 02
042 94 +/-    102 00 00     162 12 12     222 69 OP     282 43 RCL    342 07 DSZ    402 12 12     462 69 OP
043 02 2      103 01 1      163 44 SUM    223 00 00     283 15 15     343 05 05     403 42 STO    463 05 05
044 22 INV    104 01 1      164 10 10     224 42 STO    284 32 X:T    344 03 03     404 13 13     464 91 R/S
045 52 EE     105 01 1      165 43 RCL    225 25 25     285 73 RC*    345 48 48     405 42 STO    465 99 PRT
046 42 STO    106 43 STO    166 13 13     226 65 ×      286 08 08     346 25 CLR    406 00 00     466 32 X:T
047 15 15     107 17 17     167 44 SUM    227 01 1      287 77 GE     347 91 R/S    407 81 RST    467 02 2
048 59 INT    108 61 GTO    168 11 11     228 00 0      288 02 02     348 02 2      408 01 1      468 00 0
049 48 EXC    109 00 00     169 02 2      229 95 =      289 95 95     349 42 STO    409 42 STO    469 42 STO
050 15 15     110 39 39     170 22 INV    230 22 INV    290 68 NOP    350 05 05     410 12 12     470 08 08
051 22 INV    111 01 1      171 49 PRD    231 23 LNX    291 68 NOP    351 69 OP     411 42 STO    471 32 X:T
052 59 INT    112 00 0      172 10 10     232 22 INV    292 61 GTO    352 26 26     412 13 13     472 98 ADV
053 52 EE     113 00 0      173 22 INV    233 59 INT    293 04 04     353 43 RCL    413 61 GTO    473 61 GTO
054 02 2      114 22 INV    174 49 PRD    234 42 STO    294 16 16     354 06 06     414 00 00     474 02 02
055 22 INV    115 49 PRD    175 11 11     235 14 14     295 43 RCL    355 67 EQ     415 05 05     475 26 26
056 52 EE     116 15 15     176 43 RCL    236 04 4      296 54 54     356 00 00     416 00 0      476 68 NOP
057 42 STO    117 22 INV    177 50 50     237 06 6      297 61 GTO    357 01 01     417 93 .      477 68 NOP
058 16 16     118 49 PRD    178 69 OP     238 32 X:T    298 03 03     358 91 R/S    418 01 1      478 68 NOP
059 87 IFF    119 16 16     179 04 04     239 43 RCL    299 00 20     359 76 LBL    419 00 0      479 68 NOP
```


Tredje-grads-ekvation

av BJÖRN GUSTAVSSON

I PB 79-2 s 8-9 fanns ett 317 steg långt program för lösning av tredjegradsekvationer. Programmet som presenteras här är endast 172 steg långt och dessutom användarvänligare. Originalprogrammet gjordes av Göran Åhlström och Jan Söderkvist från Ludvika. Jag har med viss hjälp från Lars Hedlund kortat ner det ännu mer.

Bruksanvisning:

Programmet löser en tredjegradsekvation på formen $x^3 + ax^2 + bx + c = 0$.

1. Mata in a och tryck A.
2. Mata in b och tryck B
3. Mata in c och tryck C. Den första reella roten visas.
4. Tryck D.

5a. Om displayen inte blinkar visas den andra reella roten. Tryck E för att visa den tredje reella roten.

5b. Om displayen blinkar visas realdelen av de komplexa rötterna. Tryck E för att visa imaginärdelen.

```
000 76 LBL    043 75 -      086 34 FX     129 85 +
001 34 FX     044 43 RCL    087 45 Y×     130 43 RCL
002 75 -      045 00 00     088 03 3      131 02 02
003 43 RCL    046 33 X²     089 95 =      132 61 GTO
004 00 00     047 95 =      090 22 INV    133 34 FX
005 95 =      048 94 +/-    091 39 COS    134 76 LBL
006 92 RTN    049 42 STO    092 55 ÷      135 14 B
007 76 LBL    050 04 04     093 03 3      136 87 IFF
008 16 A*     051 32 X:T    094 95 =      137 00 00
009 22 INV    052 55 ÷      095 42 STO    138 00 00
010 45 Y×     053 02 2      096 05 05     139 99 99
011 32 X:T    054 85 +      097 86 STF    140 43 RCL
012 03 3      055 43 RCL    098 00 00     141 01 01
013 65 ×      056 00 00     099 01 1      142 85 +
014 32 X:T    057 65 ×      100 02 2      143 43 RCL
015 69 OP     058 53 (      101 00 0      144 02 02
016 10 10     059 33 X²     102 44 SUM    145 85 +
017 95 =      060 75 -      103 05 05     146 95 =
018 92 RTN    061 43 RCL    104 02 2      147 55 ÷
019 76 LBL    062 01 01     105 65 ×      148 02 2
020 11 A      063 55 ÷      106 43 RCL    149 94 +/-
021 42 STO    064 02 2      107 04 04     150 61 GTO
022 00 00     065 95 =      108 34 FX     151 34 FX
023 92 RTN    066 94 +/-    109 65 ×      152 76 LBL
024 76 LBL    067 42 STO    110 43 RCL    153 15 E
025 12 B      068 02 02     111 05 05     154 25 CLR
026 42 STO    069 42 STO    112 39 COS    155 87 IFF
027 01 01     070 06 06     113 61 GTO    156 00 00
028 92 RTN    071 33 X²     114 34 FX     157 00 00
029 76 LBL    072 75 -      115 44 SUM    158 99 99
030 13 C      073 32 X:T    116 02 02     159 03 3
031 22 INV    074 65 ×      117 94 +/-    160 34 FX
032 86 STF    075 33 X²     118 85 +      161 55 ÷
033 00 00     076 95 =      119 43 RCL    162 02 2
034 60 DEG    077 29 CP     120 06 06     163 65 ×
035 32 X:T    078 77 GE     121 95 =      164 53 (
036 43 RCL    079 01 01     122 16 A*     165 43 RCL
037 01 01     080 14 14     123 48 EXC    166 01 01
038 55 ÷      081 43 RCL    124 02 02     167 75 -
039 03 3      082 06 06     125 16 A*     168 43 RCL
040 22 INV    083 55 ÷      126 24 CE     169 02 02
041 49 PRD    084 43 RCL    127 42 STO    170 95 =
042 00 00     085 04 04     128 01 01     171 92 RTN
```

Några knep ur programmet: Kubikrot kan beräknas med sekvensen INV y^x x:t 3 x x:t Op 10 = CE, vilket fungerar för båda positiva och negativa tal. - Felindikering kan skapas med ett programsteg på följande sätt: RCL 01 + RCL 02 + = (steg 148-154). "+" före "=" avslutar beräkningen och ger felindikering.

En lämpligt vald label kan samtidigt fungera som en instruktion och spara steg. Vid steg 121 finns sekvensen GTO √X SUM 02 ... Detta fungerar dels som GTO √X, dels som √X SUM 02 vid inhopp vid steg 122. ■

KOMMENTARER TILL:

Komplement till instruktionsboken för TI 58, TI 58C, TI 59

Programmen högst upp i andra spalten på sidan 5 i Komplement till instruktionsboken kan kortas med 2 programsteg, som programmen nedan visar. Vi valde att använda de längre varianterna i Komplementet, eftersom vi tror att de är något lättförståeligare.

Vi har ansträngt oss för att undvika fel i Komplementet, och vi hoppas att det nu är felfritt. Men vi är medvetna om att fel fortfarande kan finnas kvar, och därför utfäster vi en belöning på 5 magnetkort till alla som hittar ett fel. Om fel hittas kommer vi eventuellt att skicka ut ett rättningsblad.

Lars Hedlund
Björn Gustavsson

```
TI-58/58C                TI-59

000  76 LBL              000  76 LBL
001  11  A               001  11  A
002  85  +               002  85  +
003  03  3               003  06  6
004  69 OP               004  69 OP
005  17 17               005  17 17
006  93  .               006  93  .
007  06  6               007  06  6
008  01  1               008  01  1
009  95  =               009  95  =
010  42 STO              010  42 STO
011  29 29               011  59 59
012  02  2               012  05  5
013  69 OP               013  69 OP
014  17 17               014  17 17
015  61 GTO              015  61 GTO
016  02 02               016  04  4
017  46 46               017  86 86
```

■

Exekveringstider

Ur BTIUC:s (British Texas Instruments Users Club) medlemsblad (2-2-5) har vi hittat den här artikeln rörande 59:ans exekveringstider. Nedanstående tider, som är uppmätta av Stephen Tyler, ska inte uppfattas som helt exakta. Tiderna beror på kalkylatorns klockhastighet, som kan variera något mellan olika maskiner. Dessutom varierar hastigheten även en aning med temperaturen.

Tiderna, som samtliga är angivna i millisekunder, är också giltiga för 58:an, med ett viktigt undantag: hopp till label. Den maximala tiden är självfallet mindre för ett sådant hopp, på grund av att 58:an bara har 480 steg att genomsöka, mot 59:ans 960 steg.

Register funktioner:

funktion	tid direkt	tid indirekt
STO nn	79 ms	105 ms
RCL nn	79	105
SUM nn	88	113
Prd nn	88	113
INV SUM nn	100	125
INV Prd nn	112	135
Exc nn	84	109

CMs 44-50 ms (för uppdelning 959-159.99)
CP 35 ms

HIR 0X	79 ms	HIR 5X	86 ms
1X	78	6X	89
2X	78	7X	89
3X	86	8X	89
4X	87	9X	89

Där X anger vilket stackregister (1-8) operationen avser.

Hopp till label och direkt-adress:

SBR (A-E, A'-E')	Lbl A INVSBR	95-1660 ms
SBR label	Lbl + INVSBR	129-1700
SBRInd nn		143-175
SBR mmm		143-167
GTO label		78-1650
GTOInd nn		89-112
GTO mmm		89-116
If flg n mmm		135-158

Trigonometriska funktioner:

funktion	vinkelmått	tid
sinus	grader	407-418 ms
cosinus	grader	407-418
tangens	grader	310-315
sinus	radianer	324-402 ms
cosinus	radianer	324-402
tangens	radianer	223-289
sinus	gon	342-372 ms
cosinus	gon	344-375
tangens	gon	250-280

Tiderna gäller för vinklar mellan 1-100 grader
0,1-10 rad.
resp. 1-99 gon

funktion	vinkelmått	tid
arc sin (0,1)	grader	202 ms
(1,0)	grader	126
arc cos (0,1)	grader	197
(1,0)	grader	87
arc tan (0,1)	grader	95
(1,0)	grader	93
(10,0)	grader	73
arc sin (0,1)	radianer	216 ms
(1,0)	radianer	163
arc cos (0,1)	radianer	235
(1,0)	radianer	89
arc tan (0,1)	radianer	124
(1,0)	radianer	123
(10,0)	radianer	110
arc sin (0,1)	gon	224 ms
(1,0)	gon	149
arc cos (0,1)	gon	231
(1,0)	gon	88
arc tan (0,1)	gon	98
(1,0)	gon	123
(10,0)	gon	111

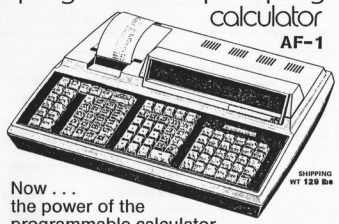
Inställning av vinkelmått (deg, rad och grad) tar samtliga 16 ms.

Senaste nytt från USA

Medan vi håller på med att göra i ordning detta nummer av Programbiten fick vi just ett nummer av TI PPC Notes. Där presenterar man en ny räknare som sägs skola introduceras den 1 april 1982. Det är inte Texas som gjort den utan ett helt nytt företag som heter OK Instruments. Maurice Swinnen säger att han fått uppgifterna genom det tryckeri som trycker firmans broschyrer.

Vi har inte haft tid att översätta broschyruppgifterna utan visar här exakt den sida som TI PPC Notes hade infört. Maurice tror att telefonnumret har något att göra med de fingerskivor som man använder i USA (ABC 2, DEF 3, GHI 4, JKL 5, MNO 6, PQR 7, ST 8, UVW 9, XYZ 0). Har Du ett förslag till samband? Liksom en känd publikation en gång gjorde erbjuder Programbiten till först öppnade rätta lösning en gratis fotvandring till Sigtuna.

OK# INSTRUMENTS
programmable prompting calculator
AF-1



Now . . . the power of the programmable calculator can be made available to everyone.

The **AF-1** card programmable prompting printing calculator is designed to bridge the gap between simple desktop calculators and computers. A powerful asset to business and technical operations alike.

Its business capability ranges from solving intricate financial analyses and long-range forecasting, to simpler operations like payroll and amortization.

For technical applications there are **7** scientific functions on the keyboard, while **18** program memory locations and **5** data registers are available for complecated programming.

Short problems can be key-programmed. But larger custom-designed programs are easy to write and record permanently on **5 by 20** inch magnetic cards.

The **AF-1** is also supplied with a Basic Applications Library containing 10 prerecorded programs.

Although the **AF-1** has enormous calculating power, it can be operated as easily as a simple general purpose calculator. Its left-to-right algebraic operating system (AOS) allows problems to be entered just as they're written. And on the **AF-1** the answers can be displayed or printed.

Whether you are a businessman, engineer, or scientist, whenever you require special mathematical techniques, the **AF-1** aids you, and your assistants in the solution to problems.

OK:Okefenoke Swamp

The AF-1 combines the convenience, simplicity and value of a desktop calculator with the powerful features found only on its more expensive relative — the co^mputer.

Prompting. The alphanumeric prompting feature used in conjunction with programming, displays letters, numbers and special symbols that let you make words and phrases that will later "ask" for entries or decisions to solve the problem.

The **AF-1's** large (1¼ by 3⅝ inch) 20-character light emitting diode display (5 by 7 dot matrix) "asks" you for your input, in terms you understand, at each stage of the problem—then waits for your keyed in response before it continues.

This rapid dialogue lets you solve a problem using different inputs, letting you explore multiple options. And, should your dialogue be interrupted, just leave the **AF-1** on and its display will tell you where you are when you return.

Programming. Easy to learn. No codes or special rules to master. More time can be spent formulating problems. And it's not necessary to learn all the functions to write simple programs. The **AF-1** can also handle very large problems with its: **5** data registers, **18** program memory locations, **99** flags, **2** branching instructions, **2** subroutine levels, Alphanumeric prompting, Choice of labels or absolute addressing, Direct or indirect addressing.

Prerecorded programs are also supplied. So easy to use, a person merely needs a general concept of what's to be solved to have a solution in seconds. People with a minimum math background can use prerecorded programs (or programs developed by others) with a minimum amount of instruction.

Any program can be recorded on blank magnetic cards for continual use.

Printing. Quiet thermal printer prints any number that appears on the display. Up to 20 characters (5 by 7 dot matrix) on 2⅜-inch wide thermal paper. A scaled replica of the display.

Fast and reliable, the **AF-1's** printer delivers a "hard" permanent copy of all your calculations and results. Identifies pertinent data and answers.

Optional Memory Expansion Modules are available to increase either program locations or data memories. The following configurations are available:

Option	Memory	Prog. Steps
1	10	19
1A	19	5
2	31	38
2A	65	19
3	43	57
3A	91	19

Optional Libraries
Six libraries containing well over 10 different programs are available: Finance with 2 programs, Electrical Engineering with 6 programs, Math I has 2 programs, Math II, 6 programs. Statistics, 9 programs and Surveying, 7 programs.

The **AF-1** comes equipped with customized software: A 2-page Operating Manual details all keystrokes and operations. A 6-page Programming Manual provides comprehensive, detailed information and numerous examples on how to program. A Basic Library Manual shows you how to use the 8 prerecorded programs and the 2 diagnostic tests that are contained on the 5 prerecorded magnetic cards. Three blank cards and a head cleaner are also included.

There's a 50-sheet tablet of Coding Forms and User-Instructions to help you write your own programs. And finally, a dust cover and a 3-wire 120-volt power cord which plugs into a standard 115-volt outlet complete the package. (Note: The **AF-1** can also be operated by 220-volts by placing its voltage switch in the 220 position and changing the power cord.)

CALL OUR TOLL FREE NUMBER 27745-3665
FOR PRICE QUOTATION BASED ON DESIRED OPTIONS

Övriga funktioner:

D.MS	726 ms	INV D.MS	750 ms
EE	36	Eng	55
INV EE	50	INV	12
(	31	)	36
CLR	17	CE	14
St flg n	63	INV St flg n	74
Int	38	INV Int	50
Fix n	80-122	|x|	34
CP	35	x⇄t	34
Pause	456	Nop	14
+/-	14	π	36
x^2 (x=1-100)	46-68	$\sqrt{x}$	50-54

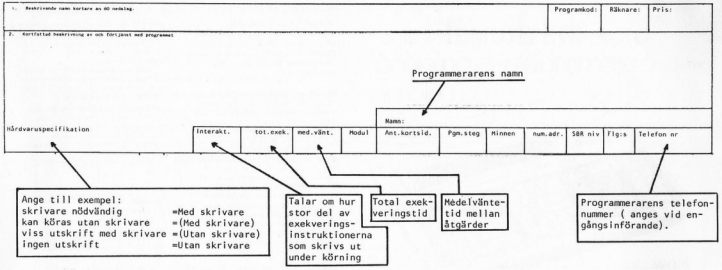
funktion	grader	radianer	gon
P→R	635 ms	631 ms	633 ms
INV P→R	790	737	777

De största områdena som inte behandlas i denna artikel är OP-funktionerna (OP 00-39), villkorliga hopp, Dsz funktionen samt de invecklade möjligheterna när de två sistnämnda funtionerna används indirekt.

BN

PROGRAMBIBLIOTEKET

Nedanstående mall kan rekvireras från föreningen.
Använd den då Du skickar in program!



Beställning av program

Beställning av program sker genom inbetalning av aktuellt belopp till föreningens postgiro 430 01 59 - 3. Utom programmens pris betalas för varje hel beställning en beställningsavgift på 10 kr.
Två alternativ finns för svenska program till 58/59:

med magnetkort 30 kr
utan magnetkort 15 kr. } + best.avg. 10 kr

För övriga program (utom de tyska i nr 81-2) finns ej magnetkort och enhetspriset är 15 kr.
De tyska kostar 25 kr utan magnetkort.
I samtliga fall tillkommer beställningsavgift på 10 kr per beställning.
Endast svenska och engelska program lagerförs. Övriga beställes av klubben till den utländska klubben en gång per månad, och det kan alltså bli viss leveranstid.

Förteckning över svenska program sänds ut till alla medlemmar. Förteckning över ca 800 engelska och belgiska program kan beställas genom insättning av 20 kr på postgiro 430 01 59 - 3.

Insändande av program

1. Programmet skänks till Föreningen Programbiten.
   För varje program Du skänker har Du rätt att välja 2 valfria program ur Programbiblioteket på magnetkort eller 3 st utan magnetkort.
   (Valda program får ej stå under rubriken Engångsinförande.)
   Mall för insändande av program kan gratis erhållas från föreningen.

2. Engångsinförande.
   Om Du önskar marknadsföra Dina privata program kan Du göra det under denna rubrik. Programbeskrivning (enligt mall) införs en gång i Programbiblioteket, utan kostnad för Dig. Du svarar för att leverans sker till eventuella beställare.

Program till Programbiblioteket insändes till
Programförmedlare Bo Nordlin
Stora Björnens gata 70
136 64 HANDEN

1. Beskrivande namn kortare än 80 tecken.

Programkod:	Räknare:	Pris:
B21a	59	15/30:-

Function tabulator and plotter

2. Kortfattad beskrivning av och förtjänst med programmet.

Given a variable and a function this program tabulates (3 lines/min.) and plots the values. Values are printed within the interval +/- 0,0000 9999,5. The program plots a maximum of 47 in the form (f(x), y(G(x))) according to user wishes. The program also determines max. & min. values to get optimum plot resolution.
Obs programdokumentationen är på engelska.

Namn: Markús Sveinn Markússon

Hårdvaruspecifikation	Interakt.	tot.exek.	med.vänt.	Modul	Ant.kortsid.	Pgm.steg	Minnen	num.adr.	SBR niv	Flg:s	Telefon nr
Med skrivare	0%		20 s.	—	2	468+	13+	ja	2	1	

1. Beskrivande namn kortare än 80 tecken.

Programkod:	Räknare:	Pris:
B21b	58	15/30:-

Tredimensionell rymdjakt

2. Kortfattad beskrivning av och förtjänst med programmet.

Med hjälp av ledtrådar som programmet ger ska Du lokalisera och skjuta ner det fientliga rymdskeppet, som kretsar runt i den tredimensionella rymden. Till din hjälp har du bl.a. en ljusbomb.

Namn: Lars Nilsson

Hårdvaruspecifikation	Interakt.	tot.exek.	med.vänt.	Modul	Ant.kortsid.	Pgm.steg	Minnen	num.adr.	SBR niv	Flg:s	Telefon nr
Utan skrivare			15 s.	1	2	400	10	ja	1	—	

1. Beskrivande namn kortare än 80 tecken.

Programkod:	Räknare:	Pris:
B21c	59	15/30:-

Rymdprogram

2. Kortfattad beskrivning av och förtjänst med programmet.

Programmet beräknar en satellits höjd över marken, omloppstid, fart och omloppsbanans längd, när en av dessa är känd. Programmet kan också beräkna jordens medelpunktsvinkel mellan två orter. Med hjälp av den kan man sedan utföra olika beräkningar, t ex räkna ut hur högt över marken ett föremål måste befinna sig vid den ena orten för att vara synligt på den andra, avståndet mellan orterna kan också beräknas. Programmet antar att jorden är ett klot.

Namn: Patrik Johansson

Hårdvaruspecifikation	Interakt.	tot.exek.	med.vänt.	Modul	Ant.kortsid.	Pgm.steg	Minnen	num.adr.	SBR niv	Flg:s	Telefon nr
(Med skrivare)	0%	ca. 25s.	5 s.	—	3	699	10	ja	2	2	

1. Beskrivande namn kortare än 80 tecken.

Programkod:	Räknare:	Pris:
B21d	(58)59	15/30:-

Super Master Mind

2. Kortfattad beskrivning av och förtjänst med programmet.

Detta är en bearbetning av B12b, beskrivet i PB 81-2. Den här versionen kan köras utan skrivare. Dessutom är det "supersnabbt", troligen det snabbaste Master Mind program som finns till TI 59. Programmet följer helt Master Minds regler. Antalet "hål" och "färger" är valbara upp till 10.

Namn: Björn Gustavsson

Hårdvaruspecifikation	Interakt.	tot.exek.	med.vänt.	Modul	Ant.kortsid.	Pgm.steg	Minnen	num.adr.	~~SBR niv~~ fast-mode	Flg:s	Telefon nr
Utan skrivare				1	1	230	30	ja		—	

1. Beskrivande namn kortare än 80 tecken.

Programkod:	Räknare:	Pris:
B21e	59	15/30:-

Cubic and quartic equation solver

2. Kortfattad beskrivning av och förtjänst med programmet.

This program solves directly cubic and quartic equations having real coefficients, solutions are usually correct within 10-11 significant digits. A special program is included, which determines the accuracy of solutions.
Programdokumentationen är på engelska.

Namn: Markús Sveinn Markússon

Hårdvaruspecifikation	Interakt.	tot.exek.	med.vänt.	Modul	Ant.kortsid.	Pgm.steg	Minnen	num.adr.	SBR niv	Flg:s	Telefon nr
(Med skrivare)	0%	23-73 s	—	-	4	720+240	30	yes	2	1	

PROGRAM FÖRMEDLARENS SPALT

Tyvärr har det inkommit ganska få program till programförmedlingen, därför uppmanar jag Dig att skicka in programmen som Du har i Dina gömmor. Kom ihåg att varje godkänt program berättigar till två program inspelade på magnetkort eller tre program utan magnetkort. Du kan välja bland 61 svenska program eller bland de hundratals engelska och belgiska program som finns presenterade i en tjock lunta som kan köpas till självkostnadspris (20:-).

Om Du skickar in ett program skriv då program-dokumentationen på skrivmaskin eller med svart bläck- eller tuschpenna (det går naturligtvis också bra att skicka in kopior på Din egen dokumentation)

Adressen till programförmedlingen är:
Bo Nordlin
Stora Björnens g. 70
136 64 Handen

■

Rekommenderade fackhandlare

Under den här rubriken låter vi de återförsäljare som säljer Texas Instruments och som är medlemmar i föreningen annonsera:

Karl D. Björkmans AB
Stensätravägen 5-7
127 03 SKÄRHOLMEN
Tel: 08/97 95 50
Filial: Humlegårdsgatan 22
Tel: 08/62 65 81

Blids Bokhandel
Holmgatan 11
791 23 FALUN
Tel: 023/280 50

Blijenburgh Electronics
Kungsholmsgatan 20
104 20 STOCKHOLM
Tel: 08/54 18 75

Bokman i Spiralen
Drottninggatan 59
600 02 NORRKÖPING
Tel: 011/12 22 00

Dawidssons Maskinaffär AB
Aspholmsvägen 1
702 27 ÖREBEO
Filial: Stortorget 8
Tel: 019/13 64 50

Elikon
Regeringsgatan 30
111 53 STOCKHOLM
Tel: 08/21 93 00

Gumperts Universitetsbokhandel
Norra Hamngatan 26
401 25 GÖTEBORG
Tel: 031/23 54 80
Filial:
Västra Storgatan 16
552 55 JÖNKÖPING
Tel: 036/11 92 40

Kontorskonsult AB
Ågatan 23
581 02 LINKÖPING
Tel: 013/13 01 75

Kontorssystem AB
Skeppsbrohuset
411 18 GÖTEBORG
Tel: 031/17 44 55

Maskinaffären Fyris
Kungsgatan 32
753 21 UPPSALA
Tel: 018/14 90 15

M-Center
Stora Gatan 4
720 13 VÄSTERÅS
Tel: 021/12 38 00

Minital
Norra Allégatan 8
402 32 GÖTEBORG
Tel: 031/11 01 54

Wettergrens Bokhandel
V:a Hamngatan 22
Avenyn 21
Vasagatan 22
Frölunda Torg
Wieselgrensplatsen
GÖTEBORG
Tel: 031/17 00 90
Kungsmässan
KUNGSBACKA
Tel: 0300/160 90

AB Montins Bokhandel
Hovrättsesplanaden 15
65 100 VASA 10
FINLAND
Tel: 961/24 25 00

K-TRYCK AB, STOCKHOLM