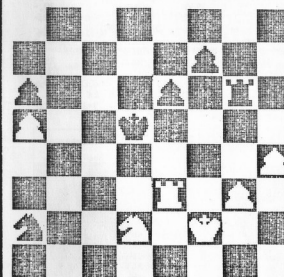


PROGRAM BITEN 82-83

SCHACK PÅ TI-59
SE S. 32

SNABBTRITARE
SE S. 18

Å, Ä DCH Ö PÅ PC-100
SE S. 28



Innehåll

Utmaningen	4
Cirkeldiagram 4; $1/a+1/b=1/c$ 7; Dubbel precision 7; $f(x)=0$ 9; Problem 10	
Programbitar	11
Hexkoder	13
Test av TI-57 LCD	16
Snabbtritare	18
En jämförelse mellan TI-59 och TI-88	20
Printalsprogram på TI-88	22
Varför inte BASIC?	24
Gardering mot strömvabrott	26
Utländska klubbar	27
Å, Ä och Ö på PC-100 - på riktigt!	28
Diofantiska ekvationen $ax+by=c$ - 2	29
Kontrollsumma	30
Schack på TI-59	32
Mall till "skrivmaskin"	36
Rättelse Kalkylatorstatus	36
Skrivkodsomvandlare	37
Programbiblioteket	38



Dyre medlem!

Du har nu i Din hand ett ganska fullmatat nummer av vårt medlemsblad. Tillgång på bra både eget och utländskt material har gjort att flera bidrag får vänta till annat tillfälle.

Michael Sperbers schackprogram med analys av programmet, en riktig utskrift med Å, Å och Ö (som inte är aprilskämt!), en systematisk beskrivning av "hexkoder" och mer material om TI-88 är höjdpunkterna tillsammans med teckenskriftdiagrammet i "Utnaningen".

En annan läsvärd artikel är Björn Gustavssons "Kontrollsumma". Hans program blir troligen antaget som TI PPC:s officiella med vilket kontrollsumman kommer att beräknas för varje program som publiceras i TI PPC Notes. I Programbiten kommer vi att använda det för om nästa nummer såvida ingen har något att invända.

En stor ändring i Programbiblioteket är att priset på programmen sänks radikalt. Beställningsavgiften 10 kr finns kvar, men ett svenskt program med magnetkort kostar nu bara 15 kr medan ett utan kostar 10 kr. Utländska program har också blivit billigare. Dessutom har "mailen" för beskrivning av programmet ändrats så att den nu lätt kan fyllas i på maskin med normal vals. Släng Dina gamla mailar om du har några kvar och beställ nya från oss om du tänker sända in program. En ny mail bifogas.

Björn Gustavssons förekommer som synes även i detta nr som författare eller översättare till ett stort antal artiklar. Trots att han f n gör militärtjänst har han engagerat sig hårt för Programbiten under veckosluten. Personligen skulle jag önska att han kunde ta över redaktörskapet helt för om nästa sommar, men det får framtiden – och årsmötet – utvisa.

Lars Hedlund, ordf

Medlemsträff

Tyvärr byter vi lokal den här gången – vi flyttat tillbaka till "Dalens träffcentrum" vid T-banestation Sandsborg i Stockholms södra förorter (T linje 18; med bil kör man in i Söfjellundsvägen österut och korsar T-banan). Tiden är lördagen den 20 november kl. 14 - 18. En del av tiden anslås till förberedda inlägg och resten till att knyta och vidareutveckla personliga kontakter.

SKRIVREGLER

Tyvärr måste vi upprepa våra skrivregler – för visst är det roligt att få bra bidrag, men om de inte kan tas in utan att skrivas om så ger de ju oss alldeles onödigt jobb som tar tid som vi kunde använda till bättre saker. Alltså:

1. bra kontrast (nytt färgband)
2. rätt spaltbredd (max 11,5 cm)
3. 1 1/2 kulsbåds radavstånd om ej annat överenskommit.

Rätta ej i texten utan markera fel i marginalen och skriv om hela raden (eller mer) på särskilt ställe. Vi klipper och monterar.

Sänd alltid med magnetkort! Särskilt sedan vi börjat med kontrollsumma i nästa nummer behöver vi dem. Samma antal tomma magnetkort åter-sändes av oss.

FÖRENINGENS TILLBEHÖRSFÖRSÄLJNING

Följande finns att köpa genom insättning av motsvarande belopp på postgiro 430 01 59 - 3:

Programbiten, årgång 1981	100,-
"-", -"- 1980	80,-
"-", -"- 1978/79	60,-
"-", -"- tre årgångar 1978/79 - 1981	200,-
Patenthandlingar TI-59	25,-
Astronomical Formulae for Calculators (något ex. i lager, annars viss leveranstid)	55,-
40 st tomma magnetkort och plånbok	108,-
Ton magnetkortplånbok	10,-
Föreningens programmeringsblanketter, olika typer, block om 50 blanketter (se PB 79-384) 11,-	
Sourcebook är slutsåld.	

MEDARBETARE



Till vänster Patrik Johansson, Utnaningen (bl a Utnaningen och "Rymdprogram") och till höger Per Eriksson, Sandviken (också Utnaningen).



INSÄNT



Föreningen Programbiten
c/o Hedlund
Arstavägen 27 6 tr
121 68 JOHANNESHOV

Sentidigt som jag helt instämmer i Per Erik Holsbergers temperamentsfulla insändare ang "teckenruta" som nordisk term för vad vi tekniker och "dataanubbar" kallar "display", skulle jag vilja tillfoga ytterligare några sakargument mot förslaget.

1. "Display", -en, -er, -erna" är kortare än "teckenruta", -n, -or, -orna".
2. "Display" har hävd inom tekniken, visserligen inte från 1960-talet men så långt tillbaka i tiden som det finns något som motsvarade vad vi kallar så.
3. Om vi ovillkorligen måste ha ett äldre språkbruk att åberopa, så kommer ordet "ruta" enligt Wesslens "Värd ord" av lågty. "röta" med betydelsen "fyrkantigt vapensfält". Hur går det om en uttillverkare tar sig det ordet före att göra "rutan" rund?
4. Var skall gränsen dras? Jag har en räkare där displayen är 11x11 mm och där sammanhängande text kan visas. Är det en "teckenruta"? Och de betydligt större displayer, som nyttjas tillsammans med lite större datorer och som vilken dag som helst kan komma att finna sin väg till miniräknarna?
5. Hur går det med vårt grannland Finland? De förstår "display", men knappast "teckenruta".

-oO-

Jag hoppas och tror, att "teckenrutan" kommer att gå samma snävliga öde tillmötes som det tidigare vetillse förelaget om "räknarens" som beteckning på våra, låt vara så, räknarens som dock kan göra tio gånger mer än 1950-talets stora, långa och bullriga maskiner.

Gösta Blum

PROGRAMBITEN OCH FÖRENINGEN PROGRAMBITEN
c/o Hedlund
Arstavägen 27 6 tr
121 68 JOHANNESHOV

För KOMMERIELLT bruk gäller följande:

Hängfelsgodset av innehåller 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 hängfelsgodset genom tryckning, duplicering, stencilering, bandinspelning, magnetkortsinspelning etc.

Tryck: K-TRYCK AB STOCKHOLM

NY BASIC-RÄKNARE FRÅN TI?

Bestämda rykten pekar på att TI tänker ta upp konkurrensen med basic-räknare av typ Sharp 1500 som väl f n har det största basic-språket och största minnet bland basic-räknarna. Sharp 1500 är ju också den som har den roliga skrivaren där pennor ritar i fyra färger och rem-sann kan gå fram- och baklänges, vilket ju medför utomordentliga möjligheter vid diagramritning.

PROGRAMMERBAR MODUL FÖR TI-59

I ett P.S. i PB 82-1 s 26 berättade vi om en dansk som experimenterat med att bygga en programmerbar modul för TI-59. Jag hade tillfälle att besöka honom under våren och se att det faktiskt fungerade. Det var emellertid, åtminstone på detta experimentstadium, en mycket utrymmskrivande apparat och även i praktiskt avseende ofullkommen. Man fick programmera varje siffra i koden binärt, och om det blev något fel under de 2000 stegen i en moduldel var det bara att börja från början igen.

Eftersom bygget av modulen var en fritidsysselsättning har konstruktören inte haft tid att fram till nu genomföra alla nödvändiga förbättringar. Han har emellertid skaffat en lämplig mikroprocessor för att underlätta programmeringen. Den färdiga modulen blir en mindre fristående låda.

Priset som nämndes i PB 82-1 ser ut att vara någorlunda rättpekan. Hör av Dig till oss om Du är intresserad.

MODULVÄJARE

Som omnämnts i tidigare nummer av PB finns en modulväljare att beställa från Systems 7, 1720 West Belt, Houston TX 77043, USA (ny adress!).

Modulväljaren ansluts via det vanliga modulfacket på räknaren och kan innehålla upp till 4 moduler (eller 7 osv om flera kombineras!). Den finns i två versioner, en manuell och en som kan styras från program. Den manuella version är utformad som en ersättning för locket till laddningsfacket på PC-100 men kan även användas fristående. Enligt prisuppgift från företaget ovan (mars 1982) kostar den manuella väljaren \$98.00 och den automatiska \$169.00. Om någon vill ringa till Mr. Harvey Sperber för information för 9,25 kr/min är numret 0091-713-468-4394.

UTMANINGEN!

Redaktörens adress:
Björn Gustavsson
Nordlandvägen 12 A
777 00 SÖDERBACKEN

Hej!

Välkomna till Utmaningen! Sommaren blev ju bättre än väntat, men det har ändå inkommit en del lösningar. Faktiskt har en del material blivit över och kommer i nästa Utmaningen.

Höjdpunkter den här gången är de två programmen för cirkeldiagram av Lars Nilsson resp Robert Prins.

NOTISER

Material som inte tagits med: Lösningar på problem 27, 30, 32 och 33. Kommer i nästa Utmaningen.

Beträffande problem 27 - påsكدagens datum - har Bertil Wallin påpekat att metoden inte är precis ny som det påstås. Den konstruerades i början av 1800-talet och kallas "Gauss's algorithms for determining Easter dates".

Paul Schlyter som har granskat mitt HP-41C-program för problem 20 - AOS-simulator - har följande kommentarer: Man kan kalla tabellerna för Lbl "+", Lbl "-" etc i stället för Lbl "ADD", Lbl "SUB" etc. Detta spar minnesutrymme och är tydligare. ISG och DSE kan användas i stället för sekvenser som 1 ST+ 30, 1 ST- 29 m fl.

Några synpunkter på programskrivning:

Ett program blir, tycker jag, mycket användbarare om man slipper ställa in uppdelning- eller manuellt innan man läser in korten. Programmet själv bör utföra alla sådana rutingör-mål.

Vissa skrivare, däribland min egen, trycker ihop texten så att den blir oläslig på den Första raden efter avrivning av remsan. Mitt

förslag är att i program alltid inleda utskriften med Adv. Detta har den ytterligare fördelen att de nya utskrifterna separeras från ev tidigare.

RÄTTELSE

Tyvärr blev problemet "Flera program på ett kort" felnumrerat i Förra Utmaningen och fick samma nummer som "Partialbrädduppdelning". Rätt nummer är 33.

I "Avancerade programmeringsmetoder" har jag översatt "sequential" litet väl rakt på sak utan att kontrollera den svenska stavningen. Korrekt svensk stavning är "sekvensiell".

CIRKELDIAGRAM (26)

Lars Nilsson, Landskrona, har skickat en lösning till problem 26: cirkeldiagram. Problemet var formulerat på följande sätt: "Gör ett program som ritat cirkeldiagram. Varje sektor skall innehålla ett beskrivande namn, bestående av högst 5 tecken." Både Lars Nilsson och Lars Hedlund har påpekat att det är så gott som omöjligt att skriva ett namn i varje sektor. Jag håller med, och därför utgår kravet att ett namn skall ingå i varje sektor.

Lars Nilssons program (pgm 1) skriver en lista över namn och procenttal. Sektorerna ritas i samma ordningsmönster. Maximalt antal sektorer är 10.

Bruksanvisning:

1. Läs in block 1, 2 & 3.
2. Initiera Grafisk Mode genom att trycka: GTO 008 10 P 17 CLR Pgm 19 SBR 045 P/R LRN Ins LRN CLR RST 6 Op 17 CLR.
3. Starta programmet med SBR 278.
4. I displayen visas hur många sektorer som kan tas emot. Mata in skrivningen för namnet och tryck R/S. Mata in procenttalet (endast heltal) och tryck R/S.
5. Upprepa steg 4 tills alla sektorer är inmatade. Tryck sedan R/S två gånger för att starta plottningen av cirkeldiagrammet.
6. För att Rita ett nytt cirkeldiagram, tryck RST och gå tillbaka till steg 3.

Metod: Cirkeldiagrammet plottas enligt PLOT 60-programmen. Men när någon begränsningslinje skall ritas i positionen 25+X eller 75+X (parallellt med y-axeln som går vinkelrätt mot remsan) skrivs en horisontell linje ut.

Förslag till vidare utveckling av programmet:

1. Det blir snabbare om det körs i Fast mode. I artikeln "Snabbare" Förklaras hur Grafisk mode och Fast mode kan användas i samma program.
2. Det kan göras mer användarvänligt genom att reducera antalet manuella tangentnedslag. T ex kan programmet utföra 6 Op 17 CLR (sist i initieringen). R/S bör räcka för att göra ett nytt diagram (ej RST SBR 278).
3. Programmet bör testa att summan av procent-talen verkligen blir 100!
4. Adv bör användas före första utskriften. På min skrivare blir rubriken oläslig om remsan rivits av ett av mats fram.
5. Det går att vinna steg genom att avlägsna Nopar och tomta steg.

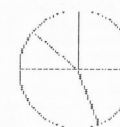
-000-

Lars Nilsson har i ett senare brev föreslagit följande ändring av programmet: GTO 231 LRN 82 LRN och spela in på kort. Detta ger en något snyggare utskrift, tycker Lars.

CIRKELDIAGRAM

NAVN PRC.

GSR 24%
FTR 20%
CIR 30%
NSR 12%
NSR 14%



Figur 1. Exempel på ett cirkeldiagram ritat av pgm 1.

Lagom till veckoslutet här jag skulle göra den sista finslipningen av Utmaningen ankom ett cirkeldiagramprogram av Robert Prins i fast mode (pgm 6). Robert skriver själv att programmet endast är en "enkel" version. Jag hoppas kunna ta med den utökade versionen i nästa Utmaningen.

000 87 HDP	060 95 =	120 13 13	180 42 STD	240 07 07	300 00 0	360 95 =	420 61 GTO
001 87 GDO	061 34 FX	121 61 GTO	181 09 09	241 69 DP	301 42 STD	361 44 SUM	421 04 04
002 87 HDP	062 00 00	122 00 00	182 00 00	242 00 00	302 00 00	362 00 00	422 00 00
003 87 HDP	063 00 00	123 00 00	183 29 29	243 86 STF	303 01 07	363 86 STF	423 86 STF
004 87 HDP	064 00 00	124 00 00	184 00 00	244 01 01	304 00 00	364 00 00	424 00 00
005 25 CLR	065 07 7	125 00 00	185 45 =	245 43 PCL	305 05 05	365 43 PCL	425 43 PCL
006 43 DP	066 00 00	126 00 00	186 00 00	246 00 00	306 00 00	366 00 00	426 00 00
007 05 05	067 42 STD	127 00 00	187 54 =	247 87 IFF	307 00 00	367 87 IFF	427 87 IFF
008 22 LNX	068 00 00	128 00 00	188 00 00	248 00 00	308 00 00	368 00 00	428 00 00
009 00 00	069 43 RCL	129 45 =	189 85 =	249 02 02	309 31 31	369 02 02	429 31 31
010 00 00	070 00 00	130 43 RCL	190 44 =	250 89 =	310 02 02	370 03 03	430 14 14
011 00 00	071 00 00	131 05 05	191 54 =	251 87 IFF	311 02 02	371 03 03	431 25 25
012 00 00	072 46 46	132 32 XIT	192 44 =	252 82 82	312 32 32	372 03 03	432 36 36
013 00 00	073 46 46	133 32 XIT	193 44 =	253 02 02	313 32 32	373 03 03	433 47 47
014 00 00	074 00 00	134 43 RCL	194 44 =	254 82 82	314 32 32	374 03 03	434 58 58
015 00 00	075 44 SUM	135 00 00	195 82 HIR	255 03 03	315 03 03	375 03 03	435 69 69
016 00 00	076 00 00	136 00 00	196 82 HIR	256 03 03	316 03 03	376 03 03	436 80 80
017 00 00	077 43 RCL	137 77 GE	197 73 RCL	257 42 STD	317 33 33	377 03 03	437 91 91
018 98 RDV	078 00 00	138 00 00	198 00 00	258 00 00	318 00 00	378 00 00	438 02 02
019 98 RDV	079 94 94	139 23 23	199 95 =	259 61 GTO	319 04 04	379 61 GTO	439 09 09
020 98 RDV	080 00 00	140 00 00	200 00 00	260 00 00	320 00 00	380 00 00	440 00 00
021 98 RDV	081 01 01	141 77 GE	201 13 13	261 00 00	321 00 00	381 00 00	441 11 11
022 98 RDV	082 43 RCL	142 32 XIT	202 13 13	262 00 00	322 00 00	382 00 00	442 22 22
023 98 RDV	083 43 RCL	143 23 23	203 28 LND	263 03 03	323 00 00	383 03 03	443 33 33
024 98 RDV	084 07 07	144 32 XIT	204 22 INV	264 03 03	324 00 00	384 03 03	444 44 44
025 68 HDP	085 42 STD	145 95 =	205 28 LND	265 04 04	325 00 00	385 04 04	445 55 55
026 98 HDP	086 01 01	146 01 01	206 28 LND	266 04 04	326 00 00	386 04 04	446 06 06
027 98 HDP	087 01 01	147 95 =	207 28 LND	267 04 04	327 00 00	387 04 04	447 17 17
028 09 09	088 01 01	148 01 01	208 28 LND	268 04 04	328 00 00	388 04 04	448 28 28
029 09 09	089 01 01	149 01 01	209 01 01	269 04 04	329 00 00	389 04 04	449 39 39
030 25 CLR	090 01 01	150 01 01	210 01 01	270 04 04	330 00 00	390 04 04	450 50 50
031 25 CLR	091 01 01	151 95 =	211 95 =	271 69 DP	331 00 00	391 04 04	451 61 61
032 25 CLR	092 01 01	152 95 =	212 95 =	272 69 DP	332 00 00	392 04 04	452 72 72
033 25 CLR	093 01 01	153 95 =	213 95 =	273 69 DP	333 00 00	393 04 04	453 83 83
034 25 CLR	094 01 01	154 95 =	214 95 =	274 69 DP	334 00 00	394 04 04	454 94 94
035 01 01	095 01 01	155 95 =	215 22 INV	275 69 DP	335 00 00	395 04 04	455 05 05
036 01 01	096 01 01	156 95 =	216 22 INV	276 69 DP	336 00 00	396 04 04	456 16 16
037 01 01	097 01 01	157 95 =	217 22 INV	277 00 00	337 00 00	397 04 04	457 27 27
038 01 01	098 01 01	158 95 =	218 22 INV	278 00 00	338 00 00	398 04 04	458 38 38
039 01 01	099 01 01	159 95 =	219 22 INV	279 00 00	339 00 00	399 04 04	459 49 49
040 25 INV	100 01 01	160 26	220 22 INV	280 00 00	340 00 00	400 04 04	460 60 60
041 44 SUM	101 01 01	161 62	221 74 RCL	281 40 40	341 40 40	401 04 04	461 71 71
042 44 SUM	102 01 01	162 62	222 74 RCL	282 00 00	342 00 00	402 04 04	462 82 82
043 44 SUM	103 01 01	163 62	223 74 RCL	283 00 00	343 00 00	403 04 04	463 93 93
044 01 01	104 01 01	164 62	224 74 RCL	284 40 40	344 40 40	404 04 04	464 04 04
045 01 01	105 01 01	165 62	225 74 RCL	285 40 40	345 40 40	405 04 04	465 15 15
046 43 RCL	106 01 01	166 62	226 74 RCL	286 40 40	346 40 40	406 04 04	466 26 26
047 75 05	107 01 01	167 62	227 74 RCL	287 40 40	347 40 40	407 04 04	467 37 37
048 25 CF	108 01 01	168 62	228 74 RCL	288 40 40	348 40 40	408 04 04	468 48 48
049 04 04	109 01 01	169 62	229 74 RCL	289 40 40	349 40 40	409 04 04	469 59 59
050 02 02	110 01 01	170 62	230 74 RCL	290 40 40	350 40 40	410 04 04	470 70 70
051 34 34	111 01 01	171 62	231 74 RCL	291 40 40	351 40 40	411 04 04	471 81 81
052 32 XIT	112 01 01	172 62	232 74 RCL	292 40 40	352 40 40	412 04 04	472 92 92
053 32 XIT	113 01 01	173 62	233 74 RCL	293 40 40	353 40 40	413 04 04	473 03 03
054 94 94	114 01 01	174 14 14	234 74 RCL	294 40 40	354 40 40	414 04 04	474 14 14
055 94 94	115 01 01	175 14 14	235 74 RCL	295 40 40	355 40 40	415 04 04	475 25 25
056 00 00	116 09 09	176 95 =	236 74 RCL	296 00 00	356 00 00	416 04 04	476 36 36
057 13 13	117 01 01	177 95 =	237 74 RCL	297 00 00	357 00 00	417 04 04	477 47 47
058 32 XIT	118 09 09	178 95 =	238 74 RCL	298 00 00	358 00 00	418 04 04	478 58 58
059 32 XIT	119 09 09	179 95 =	239 74 RCL	299 01 01	359 06 06	419 04 04	479 69 69

Pgm 1. (Lars Nilsson). Cirkeldiagram. Inspelning: 152492617, 271621322, 351330000. Spela in steg 000-479 och Ladda R30-R56. Spara in block 1, 2 & 3 i normaluppdelning.

6767676767, 152492617, 271621322, 351330000.

30 31 32 33 3113303100, 32535215540, 6100000000, 6100000000.

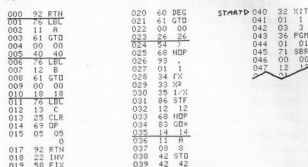
1. Ladda programmet, block 1 & 2.
2. Initiera: 10 op 17 CLR GTO 016 Pgm 19 SBR 045 P/R L/RN Ins (skapar hexkod för grafisk mode) SST (till steg 032) Ins (skapar hexkod för fast mode) L/RN RST CLR 6 op 17. (Roberts kommentar: "Jag är säker på att Björn Gustavson hatar så här mycket knappknäppande!")
3. Mata in antal resor och tryck A. (Förlägg: Använd fler än två resor endast när du har väldigt gott om tid!)
4. Mata in procentdelen för alla sektorer och tryck R/S. Se till att displayen är mjuk när du trycker R/S eftersom decimalpunkt-knepen används. (Förlägg: Avrunda alla procenttal till 0,5 % och se till att de är mindre än 49,5 %.) Maximalt antal procenttal är 40. (Försök inte mata in 40 at 2,5 % sektorer!)
5. Tryck R/S när du har matat in alla procenttal.

Exempel: (Efter initiering) 2 A 20 R/S 12,5 R/S 5 R/S 12,5 R/S 2,5 R/S 7,5 R/S 2,5 R/S 10 R/S 25 R/S 2,5 R/S R/S 10. Resultatet visas i fig 3.

Pgm 6a. (Robert Prins.) Cirkeldiagram.

0005 92 R/L	040 03 3	120 00 00	080 02 02
0001 76 LBL	061 00 00	121 61 GTO	181 65 DP
0002 11 A	062 55 1/X	122 00 00	182 05 05
0003 61 GTO	063 42 STD	123 00 00	183 05 05
0004 00 00	064 04 04	124 00 00	184 02 02
0005 40 00	065 05 05	125 00 00	185 00 00
0006 75 LBL	066 05 05	126 00 00	186 01 01
0007 12 B	067 05 05	127 00 00	187 05 05
0008 00 00	068 05 05	128 14 14	188 97 DSZ
0009 18 LBL	069 05 05	129 04 04	189 05 05
0010 42 STD	070 05 05	130 04 04	190 01 01
0011 LBL	071 05 05	131 44 SUR	191 98 STP
0012 32 XIT	072 05 05	132 44 SUR	192 98 STP
0013 01 01	073 05 05	133 44 RCL	193 09 09
0014 69 DP	074 05 05	134 44 RCL	194 09 09
0015 74 SHV	075 06 06	135 42 STD	195 32 XIT
0016 74 SHV	076 06 06	136 42 STD	196 32 XIT
0017 12 B	077 26 26	137 44 RCL	197 02 02
0018 12 B	078 26 26	138 44 RCL	198 02 02
0019 30 FRH	079 44 FRD	139 44 RCL	199 44 FRD
0020 61 GTO	080 44 FRD	140 42 STD	200 42 STD
0021 61 GTO	081 44 FRD	141 12 12	201 00 00
0022 61 GTO	082 44 FRD	142 12 12	202 42 STD
0023 61 GTO	083 44 FRD	143 12 12	203 42 STD
0024 68 HDP	084 01 01	144 44 RCL	204 44 RCL
0025 68 HDP	085 01 01	145 44 RCL	205 44 RCL
0026 01 1	086 44 HDP	206 42 STD	207 44 RCL
0027 01 1	087 44 HDP	208 42 STD	208 44 RCL
0028 33 X/X	088 44 HDP	209 42 STD	209 44 RCL
0029 33 X/X	089 44 HDP	210 42 STD	210 44 RCL
0030 86 STP	090 93 RTH	211 00 01	211 20 RCL
0031 35 1/X	091 93 RTH	212 93 RTH	212 93 RTH
0032 35 1/X	092 93 RTH	213 93 RTH	213 93 RTH
0033 26 26	093 26 26	214 00 02	214 00 02
0034 26 26	094 26 26	215 00 02	215 00 02
0035 00 0	095 00 0	216 02 02	216 02 02
0036 00 0	096 00 0	217 00 00	217 00 00
0037 00 0	097 00 0	218 00 00	218 00 00
0038 00 0	098 00 0	219 00 00	219 00 00
0039 00 0	099 00 0	220 97 DSZ	220 97 DSZ
0040 02 3	100 97 DSZ	221 00 00	221 00 00
0041 01 01	101 00 00	222 02 02	222 02 02
0042 01 01	102 02 02	223 02 02	223 02 02
0043 01 01	103 02 02	224 02 02	224 02 02
0044 00 00	104 73 RCL	225 02 02	225 02 02
0045 00 00	105 73 RCL	226 74 SHV	226 74 SHV
0046 01 1	106 59 DP	227 59 DP	227 59 DP
0047 01 1	107 59 DP	228 59 DP	228 59 DP
0048 00 00	108 44 RCL	229 44 RCL	229 44 RCL
0049 00 00	109 44 RCL	230 44 RCL	230 44 RCL
0050 44 44	110 44 RCL	231 44 RCL	231 44 RCL
0051 01 01	111 44 RCL	232 44 RCL	232 44 RCL
0052 01 01	112 44 RCL	233 44 RCL	233 44 RCL
0053 44 44	113 44 RCL	234 44 RCL	234 44 RCL
0054 93 93	114 22 INV	235 04 04	235 04 04
0055 93 93	115 22 INV	236 04 04	236 04 04
0056 42 STD	116 54 54	237 04 04	237 04 04
0057 42 STD	117 54 54	238 04 04	238 04 04
0058 69 DP	118 20 20	239 04 04	239 04 04
0059 33 33	119 72 STP	240 04 04	240 04 04

Figur 3. Exempel på ett cirkeldiagram ritat av pgm 6.



Pgm 6b. (Robert Prins.) Visar hur första delen av pgm 6a ser ut efter initiering.

0005 92 R/L	040 03 3	120 00 00	080 02 02
0001 76 LBL	061 00 00	121 61 GTO	181 65 DP
0002 11 A	062 55 1/X	122 00 00	182 05 05
0003 61 GTO	063 42 STD	123 00 00	183 05 05
0004 00 00	064 04 04	124 00 00	184 02 02
0005 40 00	065 05 05	125 00 00	185 00 00
0006 75 LBL	066 05 05	126 00 00	186 01 01
0007 12 B	067 05 05	127 00 00	187 05 05
0008 00 00	068 05 05	128 14 14	188 97 DSZ
0009 18 LBL	069 05 05	129 04 04	189 05 05
0010 42 STD	070 05 05	130 04 04	190 01 01
0011 LBL	071 05 05	131 44 SUR	191 98 STP
0012 32 XIT	072 05 05	132 44 SUR	192 98 STP
0013 01 01	073 05 05	133 44 RCL	193 09 09
0014 69 DP	074 05 05	134 44 RCL	194 09 09
0015 74 SHV	075 06 06	135 42 STD	195 32 XIT
0016 74 SHV	076 06 06	136 42 STD	196 32 XIT
0017 12 B	077 26 26	137 44 RCL	197 02 02
0018 12 B	078 26 26	138 44 RCL	198 02 02
0019 30 FRH	079 44 FRD	139 44 RCL	199 44 FRD
0020 61 GTO	080 44 FRD	140 42 STD	200 42 STD
0021 61 GTO	081 44 FRD	141 12 12	201 00 00
0022 61 GTO	082 44 FRD	142 12 12	202 42 STD
0023 61 GTO	083 44 FRD	143 12 12	203 42 STD
0024 68 HDP	084 01 01	144 44 RCL	204 44 RCL
0025 68 HDP	085 01 01	145 44 RCL	205 44 RCL
0026 01 1	086 44 HDP	206 42 STD	206 44 RCL
0027 01 1	087 44 HDP	207 44 RCL	207 44 RCL
0028 33 X/X	088 44 HDP	208 42 STD	208 44 RCL
0029 33 X/X	089 44 HDP	209 42 STD	209 44 RCL
0030 86 STP	090 93 RTH	210 42 STD	210 44 RCL
0031 35 1/X	091 93 RTH	211 00 01	211 20 RCL
0032 35 1/X	092 93 RTH	212 93 RTH	212 93 RTH
0033 26 26	093 26 26	213 93 RTH	213 93 RTH
0034 26 26	094 26 26	214 00 02	214 00 02
0035 00 0	095 00 0	215 00 02	215 00 02
0036 00 0	096 00 0	216 02 02	216 02 02
0037 00 0	097 00 0	217 00 00	217 00 00
0038 00 0	098 00 0	218 00 00	218 00 00
0039 00 0	099 00 0	219 00 00	219 00 00
0040 02 3	100 97 DSZ	220 97 DSZ	220 97 DSZ
0041 01 01	101 00 00	221 00 00	221 00 00
0042 01 01	102 02 02	222 02 02	222 02 02
0043 01 01	103 02 02	223 02 02	223 02 02
0044 00 00	104 73 RCL	224 02 02	224 02 02
0045 00 00	105 73 RCL	225 02 02	225 02 02
0046 01 1	106 59 DP	226 74 SHV	226 74 SHV
0047 01 1	107 59 DP	227 59 DP	227 59 DP
0048 00 00	108 44 RCL	228 59 DP	228 59 DP
0049 00 00	109 44 RCL	229 44 RCL	229 44 RCL
0050 44 44	110 44 RCL	230 44 RCL	230 44 RCL
0051 01 01	111 44 RCL	231 44 RCL	231 44 RCL
0052 01 01	112 44 RCL	232 44 RCL	232 44 RCL
0053 44 44	113 44 RCL	233 44 RCL	233 44 RCL
0054 93 93	114 22 INV	234 44 RCL	234 44 RCL
0055 93 93	115 22 INV	235 04 04	235 04 04
0056 42 STD	116 54 54	236 04 04	236 04 04
0057 42 STD	117 54 54	237 04 04	237 04 04
0058 69 DP	118 20 20	238 04 04	238 04 04
0059 33 33	119 72 STP	239 04 04	239 04 04

$$1/A + 1/B = 1/C \quad (14)$$

Gösta Blume skrev om problemet "1/a + 1/b = 1/c" i förra Utmaningen. När utredar har problemet vidare. Följande är en direkt fortsättning på hans notis från förra Utmaningen.

I anslutning till det föregående kanske det kan vara värt att påpeka en sak, som gäller vårt problem och som i första ögonblicket kanske är något överraskande. C kan ju alltid, liksom varje annat heltal, skrivas som en produkt av primtalspotenser så

$$C = p_1^{a_1} p_2^{a_2} p_3^{a_3} \dots p_n^{a_n}$$

där p -värdena är de i C ingående primfaktorerna och a -värdena de potenser med vilka respektive primfaktor ingår. Antalet heltalslösningar till vår ekv är nu enbart

beroende av a -värdena och således fullkomligt oberoende av primtalens storlek. Så har t ex $2^2 \cdot 3^5 = 60$ lika många lösningar, nämligen 23, som $11 \cdot 17^2 \cdot 19 = 60401$. Man nedan fallat C=60 kan lösas på en eller annan minut med ett genomslagspgm, skulle det större C-värdet kräva drygt 500 ggr så lång tid, dvs åtskilliga timmar, får det lösas med samma metod. Det är därför i första hand för C-värdena bestående av, eller innehållande, några relativt stora primfaktorer som en mera direkt lösningsmetod skulle komma till sin rätt. I brist på en sådan kan man många gånger komma snabbare fram genom att för hand ta fram de ovan visade bräken och sedan med ett yttre enkelt beräkningspgm ta fram a och b ur en ledningsvis givna formeln.

Till ledning för dem som vill övertygas vill prova på att göra ett direkt pgm för lösning av vår ekv så påminnas om det uppenbara faktum, att ett C-värde skrivet ent (1) ovan innehåller

$$(a_1+1)(a_2+1)(a_3+1)\dots(a_n+1)$$

faktorer, inkl 1 och C självt.

1	a-förd.	2	a-förd.
2	ant. lös.	3	ant. lös.
1	1	2	2
2	2	3	3
3	3	4	4
4	4	5	5
5	5	6	6
6	6	7	7
7	7	8	8
8	8	9	9
9	9	10	10
10	10	11	11
11	11	12	12
12	12	13	13
13	13	14	14
14	14	15	15
15	15	16	16
16	16	17	17
17	17	18	18
18	18	19	19
19	19	20	20
20	20	21	21
21	21	22	22
22	22	23	23
23	23	24	24
24	24	25	25
25	25	26	26
26	26	27	27
27	27	28	28
28	28	29	29
29	29	30	30
30	30	31	31
31	31	32	32
32	32	33	33
33	33	34	34
34	34	35	35
35	35	36	36
36	36	37	37
37	37	38	38
38	38	39	39
39	39	40	40
40	40	41	41
41	41	42	42
42	42	43	43
43	43	44	44
44	44	45	45
45	45	46	46
46	46	47	47
47	47	48	48
48	48	49	49
49	49	50	50
50	50	51	51
51	51	52	52
52	52	53	53
53	53	54	54
54	54	55	55
55	55	56	56
56	56	57	57
57	57	58	58
58	58	59	59
59	59	60	60

Fed reservation för ev fel ges också här nedan en sammanställning av hur antalet ekv-lösningar varierar med potenserna a_i för några enkla fall. a_i -värdena har därvid för överskådhetens skull i förekommande fall ordnats i fallande storleksordning. Således faller t ex $C=2^2 \cdot 3^5 \cdot C=2^2 \cdot 5^2 \cdot 7$ och $C=11 \cdot 17^2 \cdot 19$ alla under a_i -fördelning 2,1,1 och har således 23 lösningar.

För ett tal, som är en produkt av potenser a_i av n olika, godtyckliga primtal, är totala antalet heltalslösningar N_n (inklusive lösningar som är symmetriskt med avseende på a och b) $\frac{1}{2}(n+1) \cdot \sum_{i=1}^n (a_i+1) \cdot \prod_{j=1}^n (a_j+1)$

där " n " står för "alla n -lediga produkter av a_i ".

Jag har inte kunnat bevisa denna sats generellt, men den stämmer för alla i tabellen givna, delvis empiriskt funna värden.

Fed utställning av symmetriska lösningar blir antalet lösningar $N_n = (N_n' + 1)/2$.

För $n=3$ fås t ex

$$N_3' = 8a_1a_2a_3 + 2(a_1a_2 + a_1a_3 + a_2a_3) + 2(a_1 + a_2 + a_3) + 1$$

$$N_3 = (N_3' + 1)/2$$

Fed $a_1=3$ och $a_2=a_3=2$ blir $N_3' = 175$ och $N_3 = 88$.

DUBBEL PRECISION (2)

Som kommentar till förra Utmaningen skriver Markus Svein Markusson följande:

Jag tilltog nyss Programmet 82-2 och njt som allt. Angående mitt "D.P.-program som framvisades i Utmaningen frågar du: "Nu undrar jag, hur denna metod alltid rätt resultat?" Din kommentar är välgrundad men jag tvivlar inte på att metoden fungerar. Tag till exempel $x = 1112222$ och $y = 111444$ (Jag använder sexstaviga tal för enkelhetens skull - principen är densamma). När x divideras med y får man uppenbarligen en nia som första siffra och inte en atta som min metod kanske skulle antyda. Divisionen utförs på följande sätt:

$$k=0, x=1000 \cdot x=1112222000 (=111000 \cdot 10^3 + 222000)$$

$$i = \text{int}(x/y) = 111000/111 = 1000$$

$$x = (x - y \cdot i) \cdot 10^3 + (x_0 - y_0) \cdot 10^0 = 222000 - 444000 = -222000$$

Resten är nu:

$$i_p = x/y$$

Jag väntade nästan att du skulle reagera när du såg multiplikationen i efterdom den ofta ger ett 13-siffrigt tal som resultat. Jag har emellertid testat detta och funnit att resultatet är exakt (än så länge). Om detta inte vore fallet skulle man kunna kalla programmet "Enkel precision på ett tillräckligt sätt!"

k representerar förresten de första 9 eller 10 och inte de 8 eller 9 första siffrorna av kvoten som det påstås i algoritmen för "D.P."-programmet.

Gösta Blume har gjort pgm 2 som beräknar kvadratroten ur ett högst 12-siffrigt med ca 20 siffrors precision. Här är hans egen beskrivning av metoden och programmet:

Idet här sammanhanget avses att med ca 20 siffrors precision ta fram kvadratroten ur ett högst 10- till 12-siffrigt heltal (se dock längre fram).

Metoden bygger på att man förvandlar kvadratroten till ett kedjebråk (min gamla utkastade hobby!) med en noggrannhet på ca 10^{-20} och sedan med en eller annan känd metod utför divisionen av täljaren med nämnaren.

Principen för kvadratroten förvandling till kedjebråk är denna, här tillämpad på $\sqrt{25}$:

$$\begin{aligned}\sqrt{25} &= 5 + \frac{\sqrt{25}-5}{1} = 5 + \frac{3}{\sqrt{25}+5} = 5 + \frac{1}{\frac{1}{3}} \\ \frac{1}{\frac{1}{3}} &= \frac{\sqrt{25}+5}{3} = 3 + \frac{\sqrt{25}-4}{3} = 3 + \frac{1}{\frac{3}{\sqrt{25}-4}} = 3 + \frac{1}{\frac{3}{4} + \frac{3}{\sqrt{25}-4}} \\ \frac{1}{\frac{3}{4} + \frac{3}{\sqrt{25}-4}} &= 2 + \frac{\sqrt{25}-4}{4} = 2 + \frac{1}{\frac{4}{\sqrt{25}-4}} = 2 + \frac{1}{\frac{4}{4} + \frac{4}{\sqrt{25}-4}} \\ \frac{1}{\frac{4}{4} + \frac{4}{\sqrt{25}-4}} &= 1 + \frac{\sqrt{25}-4}{4} = 1 + \frac{1}{\frac{4}{\sqrt{25}-4}} = 1 + \frac{1}{\frac{4}{4} + \frac{4}{\sqrt{25}-4}} \\ \frac{1}{\frac{4}{4} + \frac{4}{\sqrt{25}-4}} &= 1 + \frac{\sqrt{25}-4}{4} = 1 + \frac{1}{\frac{4}{\sqrt{25}-4}} = 1 + \frac{1}{\frac{4}{4} + \frac{4}{\sqrt{25}-4}}\end{aligned}$$

Vi ser nu tillbakå vid x_1 och har därför $\sqrt{25} = 5 + \frac{1}{\frac{1}{3} + \frac{1}{2 + \frac{1}{1 + \frac{1}{2 + \frac{1}{3}}}}}$ eller, kortare $= 5 + \frac{1}{2 + \frac{1}{3 + \frac{1}{2 + \frac{1}{3}}}}$.

alltså en fyrstörig period 3,2,3,10 av partial nämnare.

Generellt kan bl a följande egenskaper bevisas hos följden av partialnämnare i kedjebråks utveckling ur kvadratrötter av hela tal:

1. följden är alltid periodisk, även om perioden kan vara så lång, att man inte får tillfälle att se den i dess helhet,

2. periodens sista tal är alltid dubbelt så stort som kvadratroten heltalsdel (när $10 = 2x_5$),

3. resten av perioden, alltså bortsett från dennas sista tal, är alltid symmetrisk (när 3,2,3).

För beräkning av kedjebråket (med bortseende från heltalsdelen, som ju alltid kan läggas till när resten är beräknad) successiva närmvärden finns det en enkel algoritm som ser ut så här:

	3	2	3	10	3	2	3	10	(3)
1	0	1	2	7	22	518	1777	18288	
0	1	3	7	24	247	765	1777	6096	52737 (194307)
+	+	+	+	+	+	+	+	+	(+)

Den första raden är, som synes, de beräknade partialnämnarna. Andra och tredje raden ger de successiva närmvärdena täljare resp nämnare, varvid de två första talen i varje rad är symboliska startvärden.

Varje täljare/nämnare är produkten av resp partialnämnare och närmast föregående närmvärdestäljare/-nämnare, ökad med den före sistnämnda täljare/-nämnare stående täljare/nämnare: $t_n m_n, t_{n-1} + t_{n-2}$ och motsvarande för nämnarna.

Följden av närmvärden har följande egenskaper: inget bråk kan förkortas, varje närmvärde är det bästa som kan åstadkommas med täljare och/eller nämnare som inte är större än det aktuella närmvärdest, varnatt närmvärde är för stort, varnatt för litet,

skillnaden mellan två på varandra följande närmvärden är (numeriskt) lika med det inverterade värdet av deras nämnare produkt.

De båda sistnämnda egenskaperna är särskilt viktiga. Det sista av de ovan framkända närmvärdena var 18288/52737, och det är för litet (se plus- och minusmarkeringarna under bråken). Vi vet då att nästa närmvärde kommer att bli för stort och att det ej kommer att skilja sig från det sista mer än $1/(62737 \cdot 18288)$ nämnare, och alltså med säkerhet mindre än $1/62737^2$. Och det exakta värdet på vår kvadratröt ligger emellan dess båda värden, varför vi med säkerhet kan säga, att felet i vårt sista närmvärde är betydligt mindre än $1/62737^2$ dvs $3 \cdot 10^{-10}$.

Det är nu ingenting som hindrar att vi fortsätter beräkningen av närmvärdestörbråk så långt som vår räkneskap kapacitet räcker, dvs till 12 till 13 siffror, vilket teoretiskt skulle motsvara en precision av 25 siffror, kvadratroten heltalsdel oroknad.

Kruket är nu, att det kan bli svårt att utföra divisionen täljare/nämnare med så stor noggrannhet. Härifrån finns det emellertid flera metoder, ett par av den visade här i Utmanningen. I det primitiva pgm som jag använder en enkla metod som tar fram kvoten med en eller två siffror i sänder, vilket förutsätter att vi inte går längre med närmvärden än till en 10-, möjligen 11-siffrigt nämnare.

Det finns nu en möjlighet att i gynnsamma fall vinna ytterligare ett par siffror. Redan i början av vår beräkning kunde vi göra oss kvitt roten heltalsdel. Det tricket kan vi utsträcka till att i stället för $\sqrt{25}$ beräkna $\sqrt{2500000000}$ (Jämt antal nollor, natur-

ligtvis!) och få en heltalsdel som är 52915, som ger oss fyra decimaler till men som vi inte behöver bekymra oss om. Detta är också i många fall vägen till att beräkna kvadratrötter ur brutna tal.

—o—o—

Pgm 2 är ett något primitivt pgm. Det används så här.

1. S13 in radikanden, N , på A. Då skrivs först ut N , $\sqrt{N}$ och entVI . Efter skrivs successiva partialnämnare och motsvarande närmvärdena täljare och nämnare, så dock endast för att rotens decimaldel. På A1 pos 067 ger tid för att stoppa körningen enl 2 nedan. Vid behov kan Pau bytas ut mot R/S.

2. Efter stopp enl 1 kan senaste närmvärdestäljare och nämnare samt motsvarande värde på fracVI tas fram med sina skyddsiffror. Dessutom visas inverterade värdet av nämnarens kvadrat för att underlätta bedömningen av förväntad precision. Detta åstadkom genom tryckning av B.

3. Efter 1 eller 2 kan en (primitiv) division av senaste närmvärdestäljare genom inmatning av antalet samtliga decimaler och tryckning av C. Högst (12 - antalet siffror i nämnaren) får begäras. Härav följer, att man inte bör låta nämnarens värde (nämnarvärd) överstiga 10^{10} . Den eller de siffror decimaler/överslag, föregående av U för att säkerställa, att inga nollor tappas bort om två eller fler decimaler begärs.

4. Det kan vara klokt att i god tid stoppa körningen enl 1 och ta fram värdena med 2. Man kan sedan fortsätta med följande närmvärde genom tryckning av R/S. Vill man gå direkt från 1 till fortsatt körning, dvs utan att ha tryckt B, trycks D.

"28 A"	28.	3.	10.
5.291502622	56441.	1179832025.	
	154307.	4047414617.	
3.	1.	2.	
1.	131570.	3654137473.	
3.	451351.	1.2535522	10
2.	154351.	3654137473.	R/S
7.	1543560.	1.2535522	0.00
3.	10.	2.	
7.	4645080.	1.2535522	-2.05 02
24.	15934951.	2915026221.	
10.	3.	2.	
72.	451351.	6.367801-21	
247.	1543560.		
3.	10.		
229.	4645080.		
765.	15934951.		
	0.		
	0.29		
	0.18		
	0.02		
	0.62		
	0.21		
	0.19		
	0.18		
	0.21		
	0.31		
	0.00		
10.	3.		
18288.	1146413777.		
62737.	393277344.		

Figur 2. Exempel på kvadratroten ur 28 med dubbel precision med pgm 2.

5. Vill man räkna fram (genom division enl 3) något tidigare närmvärde eller man i detta tillfälle (inkl skyddsiffrorna), trycker C, därefter nämnaren (dito) på R/S och antalet samtliga decimaler (se 3) på R/S.

000	43	RCL	056	07	07	112	47	CHS	168	43	RCL
001	09	05	027	48	EOG	113	INV		169	09	07
002	43	RCL	058	08	08	114	58	FIN	170	54	+
003	43	RCL	059	08	08	114	58	FIN	170	54	+
004	02	02	060	07	07	116	01	01	172	43	RCL
005	04	2	061	43	RCL	117	88	INV	173	88	+
006	25	2	062	07	07	118	99	PRT	174	95	+
007	43	RCL	063	02	02	119	02	02	175	02	+
008	03	03	064	72	CLB	120	99	PRT	176	43	RCL
009	95	065	17	21	25	INV	177	10	177	10	+
010	59	INT	066	98	RCL	122	42	STD	178	25	+
011	42	STD	067	98	RCL	122	07	07	179	42	STD
012	04	04	068	81	PRT	124	01	01	180	11	+
013	43	RCL	069	72	CLB	126	43	STD	181	43	RCL
014	65	070	12	12	CLB	126	03	03	182	65	+
015	99	PRT	071	43	RCL	128	65	STD	183	99	+
016	03	03	072	05	05	128	06	06	184	00	+
017	75	05	073	99	PRT	130	07	07	185	01	+
018	43	RCL	074	75	05	130	07	07	186	94	+
019	02	02	075	92	EE	131	86	STF	187	94	+
020	02	02	076	95	EE	132	00	00	188	43	RCL
021	02	02	077	99	PRT	132	61	STD	189	10	+
022	02	02	078	25	CLB	136	00	00	190	10	+
023	33	34	079	88	INV	138	00	00	191	58	FIN
024	94	+	080	43	RCL	138	76	CLB	192	40	INV
025	85	+	081	07	07	137	15	5	193	00	+
026	43	RCL	082	99	PRT	139	00	00	194	00	+
027	01	01	083	75	05	139	00	00	195	07	+
028	55	+	084	32	EE	140	01	FIN	196	00	+
029	43	RCL	085	99	PRT	141	00	FIN	197	99	PRT
030	43	RCL	086	99	PRT	141	00	FIN	198	61	STD
031	02	02	087	25	CLB	143	25	CLB	199	02	+
032	42	STD	088	99	PRT	145	42	STD	200	95	+
033	03	03	089	99	PRT	145	42	STD	201	99	+
034	03	03	090	05	05	145	42	STD	202	00	+
035	07	07	091	43	RCL	147	00	00	203	00	+
036	17	17	092	07	07	148	43	STD	204	76	CLB
037	43	RCL	093	07	07	148	43	STD	205	10	+
038	04	04	094	99	PRT	151	25	CLB	206	25	CLB
039	43	RCL	095	99	PRT	151	25	CLB	207	25	CLB
040	05	05	096	99	PRT	152	86	STF	208	10	+
041	05	05	097	92	EE	153	86	STF	209	10	+
042	43	RCL	098	99	PRT	155	99	PRT	210	10	+
043	05	05	099	99	PRT	155	99	PRT	211	INV	+
044	05	05	100	43	RCL	157	11	11	212	25	CLB
045	06	06	101	98	INV	157	11	11	213	25	CLB
046	06	06	102	98	INV	157	11	11	214	05	+
047	06	06	103	98	INV	157	11	11	215	05	+
048	07	07	104	75	CLB	158	04	04	216	05	+
049	05	05	105	07	07	159	01	01	217	01	+
050	05	05	106	35	15	161	43	CLB	218	07	+
051	43	RCL	107	98	INV	163	98	9	219	07	+
052	04	04	108	98	INV	165	98	9	220	13	+
053	48	EOG	109	81	PRT	165	42	STD			
054	07	07	110	75	CLB	166	04	04			
055	49	PRT	111	18	R	167	65	04			

Pgm 3. (Gösta Blume.) Kvadratröt med dubbel precision.

ETT ANTAL SATT ATT LÖSA EKV $F(X) = 0$ (9)

Göran Ahlström har skickat flera program som löser ekvationen $f(x) = 0$ och en artikel med rubriken "Tema med variationer eller Ett antal sätt att lösa ekvationen $f(x) = 0$ ". Artikeln beskriver ett antal varianter av Regula Falsi metoden. Tre av programmen och en förkortad version av artikeln återges nedan. (Som vanligt är jag tvungen att tänka på utrymmet.)

Jag har avskiktat gjort programmen (utom Stefansens metod) väldigt lika varandra eftersom de är väldigt beaktade och t o m förbättrade av varandra i vissa fall. Om man vill så vore det inget större problem att göra ett stort pgm, och sedan vilja metoden. Ett sådant pgm skulle dock bli långsammare. Som en följd av detta använder program $R_1 - R_7$ (utom Stefansens-pgmet som använder $R_1 - R_7$). Bruksanvisningarna är lika för samtliga pgm. (Med en inräkning.)

Vilka metoder är det då, och hur bra är de?

Till att börja med är det Regula Falsi, som är en bra metod så tillvida att den alltid konvergerar, men den har nackdelen att vara långsam, då den enbart har linjär konvergens.

En metod med kvadratisk konvergens är Steffensens metod. Denna konvergerar ofta mycket snabbt, vilket har gjort den till en ofta använd metod för att lösa system av icke-linjära ekv.

Slutligen kvarstår så de båda överlägsna metoderna: Illinois och Pegasus. De är båda förbättringar av Regula Falsi, och Pegasus i sin tur en förbättring av Illinois. [Om för tas inte Illinois-programmet med här. Red.]

Skillnaden mellan dessa och Regula Falsi är att medan RF alltid har en fast punkt när den konvergerar så avbryts detta i de båda andra och ändpunkterna modifieras för att hindra en punkt att ligga fast, varvid konvergensen går mycket fortare.

Bruksanvisning:

1. Läs in programmet.
2. Definiera funktionen genom att trycka GTO E LRN och knappa sedan in den på vanligt sätt med ett RTN sista. "x" får användas.
3. Mata in maxfel ϵ och tryck D.
4. Mata in den första approximationen x_0 och tryck A.
5. Mata in den andra approximationen x_1 och tryck B. (Ska ej utföras för Steffensens metod då denna endast används en begynnelseapproximation. Lbl B får för övrigt odefinierad).
6. Tryck C. Efter ett tag visas roten i displayen.

Kommentarer: x_0 och x_1 är två olika approximationer till den sökta roten x. De måste ligga på varsin sida av x och med x_1 som den förhoppningsvis bättre approximationen av de två.

Tips: Genom att mata in ett värde och trycka E kan man prova sig fram till en approximation som ger ett funktionsvärde tillräckligt nära 0. (E beräknar ju funktionsvärdet för talet i displayen.)

```
0001 76 LBL 015 02 02 036 01 01 054 61 670
0002 43 RCL 020 35 1/X 038 35 04 044
0003 01 016 021 65 02 039 43 RCL 027 29 000
0004 71 SBR 022 43 RCL 040 03 055 03 070
0005 00 010 023 02 02 041 50 1/1 056 76 LBL
0006 72 72 024 33 X 042 95 1/1 057 76 LBL
0007 02 810 025 95 1/1 043 50 1/1 058 76 LBL
0008 02 02 026 94 +- 044 34 X/T 062 42 S/T
0009 85 027 85 01 045 43 RCL 063 01 01
0010 43 RCL 028 43 RCL 046 04 064 76 LBL
0011 01 01 047 01 01 048 09 06
0012 90 030 35 +- 049 00 06
0013 71 SBR 051 43 RCL 052 57 06
0014 00 052 02 03 050 43 RCL 066 04 04
0015 72 72 053 01 03 051 01 03
0016 75 75 054 25 RCL 052 42 S/T
0017 43 RCL 053 43 RCL 053 01 07 15 E
```

Pgm 4. (Göran Åhrström.) Steffensens metod.

Referenser:

Powell, M. and Jarrat, P.: A modified Regula Falsi method for computing the root of an equation. BIT 11 (1971) s 168-174

Powell, M. and Jarrat, P.: The "Pegasus" method for computing the root of an equation. BIT 12 (1972) s 503-508

Dahlquist, G. and Björk, Å.: Numerical Methods. Prentice Hall, Englewood Cliffs, New Jersey 1974 s 222-232

Wittmeyer-Koch, L.: Kompendium i elementär numerisk analys. MAI (Matematiska Institutionen), LITH, Linköping, Linköping 1981.

PROBLEM

34. Förenklad snabbritare.

Förenkla Dejan Ristanovićs program "Snabbritare" under förutsättningen att operatören själv översätter de binära talen (upp till 11111₂) till decimala och packa registerinnehåll med två rader (40 teckenutrymmen) per register. (Lars Hedlund.)

På återbörande

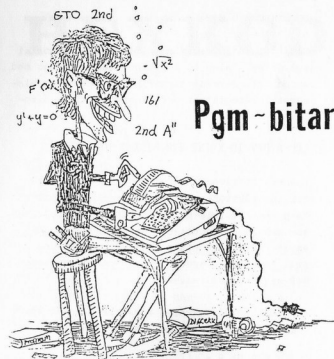
Björn Gustavsson

```
0001 76 LBL 028 75 +- 054 43 RCL 084 01 01
0002 13 C 039 43 RCL 087 07 085 43 RCL
0003 43 RCL 030 02 03 055 04 06
0004 01 01 031 95 +- 059 00 00 087 42 S/T
0005 01 01 032 84 +- 060 24 02 088 05 02
0006 01 01 033 85 +- 061 43 RCL 089 61 GTO
0007 12 12 034 43 RCL 062 04 00 090 05 04
0008 42 S/T 035 03 03 063 65 04 091 16 16
0009 02 02 036 05 05 064 43 RCL 092 43 RCL
0010 43 RCL 037 42 S/T 065 06 06 093 05 05
0011 33 C 038 05 05 066 25 C 094 32 S/T
0012 71 SBR 039 71 SBR 067 29 C 095 76 LBL
0013 01 01 040 01 01 068 1HV 096 11 11
0014 12 12 041 12 12 069 77 6E 097 42 S/T
0015 42 S/T 042 42 S/T 070 00 00 098 01 01
0016 04 04 043 06 06 071 81 299 35 FTH
0017 53 C 044 01 1 072 05 05 101 12 12
0018 43 RCL 045 75 +- 073 05 05 101 12 12
0019 03 03 047 03 03 075 03 03 103 03 03
0020 75 75 048 1/1 1 076 01 01 104 11 11
0021 43 RCL 049 35 +- 077 04 06 105 76 LBL
0022 01 01 050 05 05 078 61 GTO 106 05 05
0023 54 +- 051 05 05 079 00 00 107 42 S/T
0024 54 +- 052 1/1 1 080 1/1 1 108 1/1 1
0025 53 C 053 95 +- 081 43 RCL 109 35 FTH
0026 43 RCL 054 50 1/1 082 43 RCL 110 15 E
0027 04 04 055 32 X/T 083 42 S/T
```

Pgm 3. (Göran Åhrström.) Regula Falsi.

```
0001 76 LBL 023 85 +- 064 95 +- 099 02 02
0002 13 C 034 43 RCL 067 29 C 100 43 RCL
0003 43 RCL 035 03 03 068 1HV 101 05 05
0004 01 01 036 95 +- 069 77 6E 102 42 S/T
0005 71 SBR 037 42 S/T 070 00 00 103 03 03
0006 01 01 038 05 05 071 35 02 104 43 RCL
0007 29 29 039 71 SBR 072 43 RCL 105 05 05
0008 42 S/T 040 01 01 073 04 04 106 61 GTO
0009 02 02 041 29 29 074 1/1 1 107 00 00
0010 43 RCL 042 42 S/T 075 51 C 108 11 11
0011 02 02 043 06 06 076 24 C 109 42 S/T
0012 71 SBR 044 01 1 077 85 +- 110 05 05
0013 01 01 045 75 75 078 43 RCL 111 76 LBL
0014 42 S/T 046 03 03 079 06 06 112 76 LBL
0015 04 04 047 03 03 080 95 112 01 01
0016 04 04 048 50 1/1 081 49 FRB 114 42 S/T
0017 43 RCL 049 43 RCL 082 05 05 115 01 01
0018 01 01 050 43 RCL 083 05 05 116 01 01
0019 01 01 051 05 05 084 05 05 117 42 S/T
0020 03 03 052 50 1/1 085 42 S/T 118 12 12
0021 43 RCL 053 95 +- 086 95 +- 119 42 S/T
0022 02 02 054 50 1/1 087 43 RCL 120 03 03
0023 02 02 055 01 01 088 06 06 121 57 6E
0024 54 +- 056 43 RCL 089 61 GTO 122 76 LBL
0025 42 S/T 057 07 07 090 05 05 123 76 LBL
0026 53 C 058 77 6E 091 14 14 124 42 S/T
0027 04 04 059 01 01 092 04 04 125 97 6E
0028 04 04 060 09 09 093 03 03 126 97 6E
0029 05 05 061 57 6E 094 42 S/T 127 42 S/T
0030 43 RCL 062 34 04 095 01 01 128 15 E
0031 02 02 063 65 05 096 02 02 129 42 S/T
0032 01 01 064 43 RCL 097 04 04 130 43 RCL
0033 01 01 065 04 06 098 04 06 131 43 RCL
```

Pgm 5. (Göran Åhrström.) Pegasus-metoden.



HISTOGRAMSUBROUTINER

Bill Beebe Jr har skrivit pgm 1 som är hämtat ur TI PPC Notes v6n9/10p10. På mindre än tre sekunder trycker det ut en stolpe vars längd bestäms av ett tal i displayen mellan 0 och 20. Enda nackdelen med programmet är att stolparna går från höger till vänster. Ofta vill man hellre ha stolparna från vänster till höger. (Det går visserligen att vända upp och ner på listningen, men då blir också eventuellt text upp och nervänd.)

Det är emellertid lätt att ändra programmet. Om man byter ordningen i vilka skrivregistren laddas till den omvända (Op 4 - ... - Op 1 i stället för Op 1 - ... - Op 4) samt ställer in Fix 8 går stolparna från vänster till höger. Detta har gjorts i pgm 2.

Bruksanvisning för båda programmen: Placera ett tal mellan 0 och 20 - som representerar stolpens längd - i displayen och tryck eller anropa A.

Björn Gustavsson

LISNING AV TANGENTKRODER

TI PPC Notes 1 USA har presenterat följande lilla pgm, som - förutom att det gör vad rubriken angir - innehåller några intressanta programbitar.

Genom att lagra ett lämpligt tal i ett register (här R00) och sedan andra uppställningen, skapar programmet självt ytterligare pgm-steg, som det därefter använder som SBR.

för att slutligen - efter avslutad huvudrutin - återgå till gruppindelningen.

Program

```
0000 LBL A 1 0 0 STO 01 9 2 0 0 . 7 6 0 8
016 6 9 STO 00 LBL A' 0 0 P 17 SBR 09 55
028 6 0 P 17 0 P 20 DSE 1 A' E/S
```

Startas genom att trycka A

FSH

ATT SLÄCKA TI-57:S DISPLAY

I TI PPC Notes v7n6p9 (som har hämtat det ur den franska tidskriften *L'Ordinateur de Poche*) beskrivs hur man kan släcka TI-57:s display. Placera först sekvensen 2nd Exc SST Lbl 1 vid steg 48. När detta har gjorts kan räknarens display när som helst släckas med sekvensen SBR 2nd 48 INV STO 3 +/- +/- och åter tändas med sekvensen INV Fix CLR.

TI PPC Notes påstår att man på det här sättet har fått en TI-57C. Praktiska prov visar dock att inte endast displayen behöver energi, för ackumulatorerna blev urladdade under en natt.

Björn Gustavsson

```
0000 03 3 020 03 3 040 03 3 060 79 INT
0001 02 2 021 03 0P 041 02 2 061 55 +-
0002 03 3 022 49 0P 042 03 3 062 05 +-
0003 02 2 023 03 0P 043 02 2 063 95 +-
0004 03 3 024 03 0P 044 03 3 064 45 +-
0005 02 2 025 05 2 045 02 2 065 01 1
0006 03 3 026 03 3 046 47 0P 066 00 0
0007 02 2 027 05 2 047 04 0P 067 95 +-
0008 03 3 028 03 3 048 65 0P 068 50 1/1
0009 02 2 029 03 3 049 05 05 069 42 S/T
0010 49 0P 030 03 3 050 50 00 070 60 00
0011 01 01 031 02 2 051 76 LBL 071 83 GTO
0012 03 3 032 11 1 052 11 1 072 60 00
0013 02 2 033 04 2 053 49 0P
0014 03 3 034 03 3 054 05 05
0015 02 2 035 03 03 055 55 +-
0016 03 3 036 03 3 056 05 05
0017 02 2 037 02 2 057 75 +-
0018 03 3 038 03 3 058 03 3
0019 02 2 039 02 2 059 85 +-
0020 03 3 040 03 3 060 04 4
0021 02 2 041 02 2 061 05 +-
0022 03 3 042 03 0P 062 05 05
0023 02 2 043 03 0P 063 05 05
0024 03 3 044 03 3 064 05 05
0025 02 2 045 03 3 065 05 05
0026 03 3 046 03 3 066 05 05
0027 02 2 047 03 3 067 05 05
0028 03 3 048 03 3 068 05 05
0029 02 2 049 03 3 069 05 05
0030 03 3 050 03 3 070 50 1/1
0031 02 2 051 03 3 071 50 1/1
0032 03 3 052 03 3 072 50 1/1
0033 02 2 053 03 3 073 50 1/1
0034 03 3 054 03 3 074 50 1/1
0035 02 2 055 03 3 075 50 1/1
0036 03 3 056 03 3 076 50 1/1
0037 02 2 057 03 3 077 50 1/1
0038 03 3 058 03 3 078 50 1/1
0039 02 2 059 03 3 079 50 1/1
0040 03 3 060 03 3 080 50 1/1
0041 02 2 061 03 3 081 50 1/1
0042 03 3 062 03 3 082 50 1/1
0043 02 2 063 03 3 083 50 1/1
0044 03 3 064 03 3 084 50 1/1
0045 02 2 065 03 3 085 50 1/1
0046 03 3 066 03 3 086 50 1/1
0047 02 2 067 03 3 087 50 1/1
0048 03 3 068 03 3 088 50 1/1
0049 02 2 069 03 3 089 50 1/1
0050 03 3 070 03 3 090 50 1/1
0051 02 2 071 03 3 091 50 1/1
0052 03 3 072 03 3 092 50 1/1
0053 02 2 073 03 3 093 50 1/1
0054 03 3 074 03 3 094 50 1/1
0055 02 2 075 03 3 095 50 1/1
0056 03 3 076 03 3 096 50 1/1
0057 02 2 077 03 3 097 50 1/1
0058 03 3 078 03 3 098 50 1/1
0059 02 2 079 03 3 099 50 1/1
0060 03 3 080 03 3 100 50 1/1
```

00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000

Pgm 1. (Björn Gustavsson.)

```
0000 03 3 020 03 3 040 03 3 060 04 4
0001 02 2 021 03 0P 041 02 2 061 05 +-
0002 03 3 022 49 0P 042 03 3 062 05 +-
0003 02 2 023 03 0P 043 02 2 063 05 +-
0004 03 3 024 03 0P 044 03 3 064 05 +-
0005 02 2 025 05 2 045 02 2 065 05 +-
0006 03 3 026 03 3 046 47 0P 066 00 0
0007 02 2 027 05 2 047 04 0P 067 95 +-
0008 03 3 028 03 3 048 65 0P 068 50 00
0009 02 2 029 03 3 049 05 05 069 42 S/T
0010 49 0P 030 03 3 050 50 1/1 070 50 1/1
0011 01 01 031 02 2 051 76 LBL 071 83 GTO
0012 03 3 032 11 1 052 11 1 072 60 00
0013 02 2 033 04 2 053 49 0P
0014 03 3 034 03 3 054 05 05
0015 02 2 035 03 03 055 55 +-
0016 03 3 036 03 3 056 05 05
0017 02 2 037 02 2 057 75 +-
0018 03 3 038 03 3 058 03 3
0019 02 2 039 02 2 059 75 +-
0020 03 3 040 03 3 060 04 4
0021 02 2 041 02 2 061 05 +-
0022 03 3 042 03 0P 062 05 05
0023 02 2 043 03 0P 063 05 05
0024 03 3 044 03 3 064 05 05
0025 02 2 045 03 3 065 05 05
0026 03 3 046 03 3 066 05 05
0027 02 2 047 03 3 067 05 05
0028 03 3 048 03 3 068 05 05
0029 02 2 049 03 3 069 05 05
0030 03 3 050 03 3 070 50 1/1
0031 02 2 051 03 3 071 50 1/1
0032 03 3 052 03 3 072 50 1/1
0033 02 2 053 03 3 073 50 1/1
0034 03 3 054 03 3 074 50 1/1
0035 02 2 055 03 3 075 50 1/1
0036 03 3 056 03 3 076 50 1/1
0037 02 2 057 03 3 077 50 1/1
0038 03 3 058 03 3 078 50 1/1
0039 02 2 059 03 3 079 50 1/1
0040 03 3 060 03 3 080 50 1/1
0041 02 2 061 03 3 081 50 1/1
0042 03 3 062 03 3 082 50 1/1
0043 02 2 063 03 3 083 50 1/1
0044 03 3 064 03 3 084 50 1/1
0045 02 2 065 03 3 085 50 1/1
0046 03 3 066 03 3 086 50 1/1
0047 02 2 067 03 3 087 50 1/1
0048 03 3 068 03 3 088 50 1/1
0049 02 2 069 03 3 089 50 1/1
0050 03 3 070 03 3 090 50 1/1
0051 02 2 071 03 3 091 50 1/1
0052 03 3 072 03 3 092 50 1/1
0053 02 2 073 03 3 093 50 1/1
0054 03 3 074 03 3 094 50 1/1
0055 02 2 075 03 3 095 50 1/1
0056 03 3 076 03 3 096 50 1/1
0057 02 2 077 03 3 097 50 1/1
0058 03 3 078 03 3 098 50 1/1
0059 02 2 079 03 3 099 50 1/1
0060 03 3 080 03 3 100 50 1/1
```

Pgm 2. (Björn Gustavsson.)

00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000

Pgm 3. (Björn Gustavsson.)

APPROXIMERING AV DERIVATA

FÖR TI-57

Här visas ett kort program för approximering av derivator på TI-57.

Formeln som används är:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

F(x) kan vara upp till 27 programsteg långt. - kan användas, men inte registren 0-2 och avsluta f(x) med INV SBR.

Exempel:

Beräkna f'(1) och f'(3) för f(x)=3x³-2x+1.
Sätt h till .001.
Tryck GTO 0 LBN och slå in funktionen (se steg 22-34 i listningen).
Sätt fix till 2:FIX 2
Slå in h: 0,001 STO 1
Slå in första x-värdet: 0,1 STO 0
Beräkna f'(1):RST R/S ger -1,91
Slå in andra x-värdet:3 STO 0
Beräkna f'(3):RST R/S ger 79,00

*(h skall vara omkring .001-.0001 och fungerar som noggrannhet.)

Thomas Fänge

STJÄRNSKYTTE			
STEG	INSTRUKTION	DATA	REMARK
0	h	0	f(x)
1	h	2x	1
2	Används	2x	2
3		2x	3
4		2x	4
5	JAGD	2x	5
6	JAGD	2x	6
7	10	2x	7
8		2x	8
9		2x	9

ETT KNEP MED HIR

Följande rutin delar ett tvåsiffrigt heltal i två delar. Talen 10 i rutinen kan ersättas med andra tal. De mest signifikanta siffrorna placeras i T-registret och de resterande i displayen.

LBL A DIV 10 X INT HIR 51 X: T 10 = RTN

Knepet att påverka en vanlig operation med HIR-instruktionen kan användas i andra sammanhang (se artikeln om skrivkodsomvandlare), men observera följande: Om den vilande operationen är "+", "DIV" eller "INV X" ändras den till "+", "x" respektive "y/x" om motsvarande HIR-register ändras.

Exempel: Följande sekvens ger resultatet 10. Det är inte 2-8 som beräknas, utan 2+8.

5 - 2 HIR 01 8 =

Markus Sveinn Markússon/Björn Gustavsson

Stjärnskytte

Flera läsare har uttryckt sin uppskattning av spelet "Stjärnskytte" från PB 81-3 s 20. Markus Sveinn Markússon är för försök med "penna och massor av papper" övertygad om att elvaskottserien från förra numret är den kortast möjliga. Han påpekar också att sex skott krävs för att sluta i startpositionen (5, 2, 1, 2, 5 och 1).

Från 52-Notes har jag fått idén till ytterligare några problem gällande stjärnskytte.

- Skjut bort alla stjärnor.
- Utgå från ett svart hål i mitten omgivet av stjärnor. Skjut tills du får en stjärna i mitten omgivet av svarta hål. Detta är alltså raka motsatsen till det vanliga stjärnskytte. (Programmet kan inte användas.)

Lösningar kan skickas till mig.

Björn Gustavsson

MANUSSTOPP PB 82-4

är den 15 november. Sedan måste monteringsarbetet börja för att tryckeriet skall hinna med innan postens julrusch börjar.

HEXKODER

av Björn Gustavsson

VAD ÄR HEXKODER?

Varje programsteg tar upp en byte (grupp av 8 binära siffror) och är lagrad i BCD-format (BCD=Binary Coded Decimal). Det betyder att varje enskild siffra (0-9) är lagrad i fyra bitar (bit=binary digit=binär siffra). Exempel: Koden 59 (Int) lagras i ett programsteg som 0101 1001 (binärt) där 0101 motsvarar 5 och 1001 motsvarar 9. (Se tabell 1 om du är osäker på det binära eller hexadecimala talsystemet.)

Vi ser i tabell 1 att man kan kombinera 4 bitar på 16 olika sätt, men endast 10 av dessa används i BCD-format. Men med vissa knep kan man skapa koder vars entalssiffror innehåller någon av de "förbjudna" koderna 1010-1111 (binärt) eller A-F (hexadecimalt). Eftersom det i datorsammanhang är betydligt vanligare att använda det hexadecimala talsystemet än det binära kallas sådana koder något oegentligt "hexkoder". Räkaren är inte alls beredd på sådana koder och underliga saker inträffar när den stöter på dem. Det mest kända exemplet är väl "Graphics mode" som orsakas av att hexkoden 1F avbryter skrivarens utskrift.

TABELL 1

Binärt	Decimalt	Hexadecimalt
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010		A
1011		B
1100		C
1101		D
1110		E
1111		F

HUR FRAMSTÄLLS HEXKODER?

Det finns flera metoder för att framställa hexkoder. Men den grundläggande principen är densamma och också begränsningarna. Man kan endast skapa en hexkod på ett steg delbart med 8 (000, 008, 016,...). Hexkoder kan heller inte flyttas med Ins eller Del - alla sådana försök förvandlar den till en vanlig BCD-kod.

Principen för att skapa en hexkod är följande: Man lagrar en vanlig BCD-kod ab på ett steg n delbart med 8, försätter räknaren i ett särskilt tillstånd med en eller annan "underlig" sekvens och trycker LBN. Nu finns det skenbart en kod ad på steg n. Varje gång man trycker Ins utförs subtraktionen ab-ad och resultatet ersätter den ursprungliga koden ab. Därefter trycker man LBN igen samt trycker RST CLR för att räknaren skall fungera normalt igen.

Den ovan nämnda subtraktionen utförs som en vanlig decimal subtraktion med följande undantag: Vid lån från tiotalssiffran till entalssiffran lånas 16 enheter (ej 10). Det förutsätts alltid finnas en hundratalsiffra som (normalt) lån kan göras ifrån. Exempel: Subtraktionen 74-55 utförs så här:

$$\begin{array}{r} 74 \\ -55 \\ \hline 1F \end{array}$$

(16+4-5 blir 15 som enligt tabell 1 hexadecimalt motsvaras av F.)

Det finns i princip två metoder för att försätta räknaren i det tillstånd som gör hexkodframställning möjlig: en med modul och en utan modul.

CROM-METODEN

CROM-metoden är den äldsta och behandlas därför först. Den kräver att en modul är installerad - för sekvenserna som gör hexkodframställning möjlig: en med modul och en utan modul. Så här gör man: Man ändrar uppdeleningen till 9 eller 10 op 17 och ser till att stegräknaren pekar på steget där hexkoden skall skapas. Det steget skall innehålla koden

ab. Sedan trycker man antingen "CLR Pgm 19 SBR 045 P/R" eller "Pgm 12 SBR 444 R/S D.MS". Därefter trycker man LRN, trycker Ins en eller flera gånger och trycker LRN RST CLR.

Med den här metoden hämtas koden *ad* från motsvarande steg i den fasta programvaran (se komplement till instruktionsboken s 7). I tabell 2 återges värdet på *ad* för alla steg n delbara med 8 under 240.

Exempel: Vi skall framställa koden 1F på steg 024. Tabell 2 visar att koden på steg 024 i den fasta programvaran är 55 (DIV). Vi måste då hitta ett tal *ab* som gör att *ab* - 55 = 1F. Genom att prova sig fram (eller genom någon intelligentare metod) finner man att 74 - 55 = 1F. (Se räknexemplet ovan.)

Hexkoden sätts alltså in på följande sätt: GTO 024 LRN SUMIND BST LRN 10 op 17 CLR Pgm 19 SBR 045 P/R LRN Ins LRN CLR RST.

Om man sedan trycker GTO 024 LRN ser man "024 25". Hexkoden 1F visas alltså som koden 25 (CLR). Men det är absolut inget CLR - den här koden avbryter en pågående utskrift på skrivaren.

Det är nu lämpligt att förklara reglerna för hur en hexkod visas i displayen. Allmänt gäller: Entalssiffran som visas är hexkodens entalssiffran minskad med 10 och tiotalssiffran är hexkodens tiotalssiffran ökad med 1. Exempel:

Hexkod	Visas	Acostas beteckning på hexkoden
0A	10	h10
0B	11	h11
0C	12	h12
0D	13	h13
0E	14	h14
0F	15	h15
2A	30	h30
2F	35	h35
9A	00	h00
9E	04	h04
9F	05	h05

Det är bra att kunna detta samband utan till. Patrick Acosta, en av pionjärerna i TI PPC beträffande hexkoder, använder genomgående beteckningen *h*, där *h* är utseendet i displayen enligt tabellen ovan. Det är också nödvändigt att kunna när *pgm* 1 används.

Pgm 1 beräknar vilken kod *ab* som skall användas för att erhålla en viss hexkod, angiven i *h*-format.

Bruksanvisning:

1. Mata in *ad* för steget (se tabell 2) och tryck A.
2. Mata in önskad hexkod i *h*-format och tryck B. *ab* visas i displayen, och i t-registret finns antalet Ins som måste utföras.

Exempel: Skapa hexkoden 9E (h04) på steg 016. *ad* för steg 016 är enligt tabell 2 55. Med *pgm* 1 i räknen tryck 55 A 4 B. 53 visas: genom att trycka x:t visas 1. Det innebär att h04 skapas genom att placera kod 53 ("") på steg 016 och sedan trycka GTO 016 9 op 17 Pgm 12 SBR 444 R/S D.MS LRN Ins LRN RST CLR. (Jämför PB 82-1 s 311 h04 är känd som "super clear".)

000 16 LBL	036 75	040 75	Till vänster:	000 76 ETH
001 16 R	037 06	041 06	"Pgm 1"	001 76 LBL
002 45	038 05	042 05	"Pgm 1"	002 11 R
003 93	039 43	043 RCL	Behöverlig	003 41 GTO
004 01	040 01	044 01	utgångskod.	004 01 GTO
005 45	041 95	045		005 33
006 59	042 0P	046 0P		006 05
007 32	043 24	047 24		007 05 05
008 51	044 23	048 23		008 00 0
009 32	045 77	049 77		009 12 B
010 01	046 00	050 00		010 00 0
011 00 0	047 38	051 38		011 00 0
012 95	048 40	052 40		012 00 0
013 92 ETH	049 04	053 04		013 76 LBL
014 76 LBL	050 45	054 45		014 15 E
015 11 R	051 32	055 32		015 25 CLR
016 16 R	052 43	056 43		016 01 L
017 42 STD	053 00	057 00		017 85 +
018 01 01	054 00	058 00		018 01 L
019 32	055 43	059 43		019 09 9
020 42 STD	056 02	060 02		020 01 E
021 00 00	057 95	061 95		021 01 L
022 00 0	058 95	062 95		022 01 E
023 92 ETH	059 01	063 01		023 94 +
024 76 LBL	060 95	064 95		024 22 INV
025 12 B	061 95	065 95		025 52 EE
026 16 R	062 22	066 22		026 95
027 42 STD	063 59	067 59		027 60 DEG
028 03 03	064 45	068 45		028 91
029 32	065 01	069 01		029 85
030 42 STD	066 00	070 00		030 85
031 02 02	067 00	071 00		031 81 81
032 00 0	068 95	072 95		032 69 DP
033 42 STD	069 43	073 43		033 05 05
034 04 04	070 04	074 04		034 05 05
035 32	071 95	075 95		035 42 STD
	072 92	076 92		036 05 05
				037 95 +
				038 01

TABELL 2

Steg (n)	Kod (ad)	Steg (n)	Kod (ad)
000	82	120	53
008	04	128	54
016	55	136	43
024	55	144	82
032	65	152	06
040	54	160	43
048	53	168	75
056	54	176	65
064	32	184	55
072	03	192	44
080	03	200	44
088	43	208	05
096	53	216	82
104	54	224	44
112	43	232	44

DEN MODULFRIA METODEN

Nu är det dags att förklara den modulfria metoden som har flera fördelar jämfört med CROM-metoden: Hexkoder kan skapas på steg upp till 872. Uppdelning är valfri. Det är lättare att beräkna vilken utgångskod man skall använda. Färre manuella tangentnedslag är nödvändiga.

Metoden utnyttjar *exekveringsbufferten* - ett särskilt register dit 8 programsteg i taget kopieras vid programkörning. Med sekvensen STF Ind 7 EE kan man exekvera ett GTO utan att ändra innehållet i *exekveringsbufferten*. (Se PB 81-3 s 31-32 för ytterligare detaljer.) Den skenbara koden *ad* finns då i *exekveringsbufferten* och en hexkod skapas då med hjälp av Ins på vanligt sätt.

Så här går det till: Bestäm först oktettnumret för *n*. (n delat med 8.) Läg därefter följande initieringsrutin någonstans i programmet: 1 + oktettnumret 9 EE 11 +/- = Deg R/S STF Ind 0. (Viktigt: *ad* får inte ligga på första steget i en oktett.) Om nu koden 00 placeras på steg *n* så är *ab* = 0 och 0 - *ad* = önskad hexkod. 0 är alltså lika med 0 - önskad hexkod (subtraktion på "hexadecimalt" vis mellan entals-siffrorna).

Exekvera initieringsrutinen och tryck: SST SST 7 EE LRN Ins LRN CLR. Hexkoden finns nu på steg *n*.

Exempel: Se artikel om A, X och Ö på PC-100 (del av programmet återges här - se *pgm* 2). Man vill på steg 008 få koden 1F, som medför Graphics Mode genom att avbryta *exekveringen* av Op 05 på steget 006-007. Oktettnumret blir 1 (=8/8). *ad* fås ur 00 - 1F, vilket som entals-siffror ger 16 - F = 1 och som tiotalssiffror 9 (vi har lånat!) - 1 = 8, dvs 81. På steget 105 - 031 finns erforderlig sekvens.

Vad man måste göra manuellt är att trycka E SST SST 7 EE LRN 2nd Ins LRN. Detta är 8 1/2 manuella tangentnedtryckningar (två nedtryckningar av samma tangent räknas som 1 1/2 nedtryckningar!).

På steg 009 finns som synes koden 12 (B). Den står där för att efter initiering ge 92 (RTN). Det går då att anropa Op 05 och hexkoden med SBR 006 och få ett normalt subrutinåterhopp till huvudprogrammet. Genom försök visade det sig att 80 adderades till steget efter hexkoden 92 = 80 + 12. Robert Prins har aviserat ett sätt att kontrollera de sex stegen efter en hexkod och vi hoppas återkomma om detta.

Ett annat exempel: Man vill på steg 016 börja ett program i fast mode. Erforderlig hexkod är 0C. Oktettnumret är 16/8 = 2. Utgångskoden i formeln blir 00 - 0C = 94. Initieringsrutinen på lämpligt ställe i programmet blir 1 + 2 9 EE 11 +/- = Deg R/S STF Ind +/- (=94). Som vanligt trycks sedan (efter att ha exekverat initieringsrutinen) SST SST 7 EE LRN Ins LRN. (Se Nyheter på TI-58C, PB 82-1 s 18-20.)

HUR ANVANDS HEXKODER?

Följande förteckning över vad några hexkoder gör för något är tagen ur TI PPC Notes.

9E (h04) fungerar som "super clear" och raderar displayen, t-registret, EE och Eng, subrutinadressregistret, felindikation, HIR-registren, nollställer alla flaggor samt raderar (utom på TI-58 C) även *programminnet*! Likadant fungerar 2C (h32), 2D (h33), 3F (h45), 4D (h53), 5E (h64), 6B (h71), 6C (h72) och 7D (h83).

2B (h31), 3B (h41) och 4B (h51) fungerar som en sinusinstruktion. Med INV framför blir det något konstigt.

4E (h54) kopierar t-registret till displayen om det kommer omedelbart efter x:t och om det finns vilande operationer. Om något kommer mellan x:t och 4E kan man få annat resultat.

1E (h24) är (enligt Patrick Acosta) den konsistigaste hexkoden av alla. Sätt den på steg 000 och sätt andra koder på steg 001 - 007 som vi kan kalla AB, CD, EF, GH, IJ, KL och MN. Tryck nu RST R/S. Nu har koderna på steg 000 - 007 ändrats till DA, FC, HE, JG, LI, NK, OL och OO. SBR 001 ger alltså ett program på 7 steg medan SBR 000 ger ett helt annat program! (Patrick rekommenderar en IQ på 160 och nervver av stål för den som vill försöka skriva ett sådant program.)

0B (h11) ger följande resultat som "ändraled" i sammansättningar:

RCL OB: Återkallar R00, kör SBR A.

Fix OB: Ställer in Fix O, kör SBR A.

Stf OB: Ställer flagga O, kör SBR A.

GTO OB: kör SBR A, återvänder till steg 001.

SBR OB: kör SBR A, kör sedan SBR 001 och återvänder till steget efter koden OB.

GTO Ox OB: kör SBR A, återvänder till steg (10x+1).

SBR Ox OB: kör SBR A, kör sedan SBR (10x+1) och återvänder till steget efter koden OB.

Test av TI 57 LCD

av Björn Gustavsson

TI har tidigare kommit med LCD-versioner av äldre räknare som t ex TI-30 och TI Programmer, vilket ju inneburit en förändring till det bättre. Jag har därför nyfiket väntat på TI-57 LCD. Med TI-57:s 50 kopplade programsteg kan förvåningsvärt komplicerade problem lösas. Vad skulle inte då Ingvar Magnusson ("a new giant in TI-57 programming") kunna Åstadkomma med en 57:a med t ex 100 steg?

När jag läste instruktionsboken till TI-57 LCD märkte jag snart att den är en unik TI-räknare -- den är TI:s enda LCD-räknare som är betydligt sämre än sin föregångare!

Till det yttre är förstas TI-57 LCD betydligt elegantare än TI-57. Tangenterna har delats in i olikfärgade fält efter funktion - programmeringsfunktionerna t ex har fått en ljusblå färg. Det enda som jag inte tycker om är att LBL är en förstafunktion, medan SBR är en andrafunktion. (Men det är helt logiskt - SBR kan inte användas från tangentbordet för att starta ett program som på TI-57. Sekvensen GTO n R/S måste användas.)

Förbättringarna som införts är dels TI:s standardförbättringar för LCD-räknare, dels några hämtade från HP-41C. Standardförbättringarna är LCD-display med indikatorer för 2nd, INV och vinkelmått, kontinuerligt minne och fakultet.

Följande HP-förbättringar har införts:

Man skriver inte längre SUM 1, utan i stället STO + 1. Man kan t o m skriva STO Y^x 1 och STO INV Y^x 1, vilket saknar motsvarighet på TI-57. Alla dessa instruktioner kopplas till ett programsteg.

Inmatning och redigering av program fungerar ungefär som på HP-41C. När man matar in program visas hela tiden senast inmatade programsteg. Någon Ins-tangent finns inte; i stället är insert automatisk vid all inmatning. Varje gång något matas in flyttas alla efterföljande programsteg ett steg mot slut och ger plats för instruktionen. Detta får till följd att man dels aldrig kan skriva över ett steg oavsiktligt, dels att för att ändra ett steg måste man radera tidigare innehåll med Del och sedan mata in den nya instruktionen.

Det var de väsentliga förbättringarna, nu till försämrningarna.

Den allvarligaste är att TI-57 LCD endast har 48 programsteg + 1 register och T-register. Dessa 48 programsteg kan göras om till register helt eller delvis med instruktionen 2nd Part m, där m är antal register som önskas. Varje register motsvarar 8 programsteg, så man kan få högst 7 register (och T-register), men man har då inga programsteg.

Detta kan jämföras med TI-57 som har 8 register (inklusive R₇ som samtidigt är T-register) och 50 programsteg. TI-57 kan alltså sägas ha 50 programsteg mer än TI-57 LCD, eftersom räknarna har lika många register.

Inte nog med att TI-57 LCD har mindre minne, dessutom slösar räknaren med det på ett makalöst sätt. Om man tittar på hur mycket varje programsteg på TI-57 LCD kan rymma (koder som -61.45.02 "STO INV Y^x 2" kan lagras) kan man sluta sig till att 4 programsteg räcker för att rymma innehållet i ett register. Om TI hade utnyttjat minnet på detta sätt hade 48 steg räckt för att ge 12 register, inte 6 som nu är fallet! (Ett register samt T-register finns alltid på TI-57 LCD.)

Andra försämringar:

- * TI-57:s statistiska funktioner saknas helt
- * Det finns bara en subrutinnivå (TI-57 har två), dvs en subrutin kan inte anropa en annan

Sammanfattning: Det som gör mig besviken är inte att TI släpper ut ännu en halvdan programmerbar LCD-räknare i lågprisklassen. Om de hade kallat den t ex TI-13 hade jag inte klagat. Men när TI kallar den för TI-57 LCD blir jag verkligen upprörd.

Med TI-57 kunde man göra saker som verkar otroliga med tanke på antal programsteg. Några exempel hämtade ur PB: Integralberäkning med trapezformeln, beräkning av veckodag för godtyckligt datum, tröghetsmoment, lösning av differentialekvationer med Runge Kutta metoden (Det är tveksamt om det ens går att få in ens Eulers metod på TI-57 LCD), tredjegradskvadrant och cirkel genom tre punkter. Dessa är

på grund av minnesbrist omöjliga att köra på TI-57 LCD.

Ett mästerverk inom TI-57 programmering. Mästermind av Peter van Roy, återges här för första gången i PB. Det kommer aldrig att kunna köras på TI-57 LCD, av tre orsaker:

1. Det består av 50 programsteg.
2. Det behövs åtta arbetsregister.
3. Statistiska funktioner saknas. Å används i programmet för att med ett steg beräkna kvoten mellan R₇ och R₀.

00	32 4 STO 4	25	89 8
01	32 2 STO 3	26	66 EQ
02	32 2 FCL 2	27	-61 INV SBR
03	32 1 STO 1	28	43
04	61 8 SBR 8	29	32 1 STO 1
05	32 3 STO 3	30	-49 INV INT
06	38 6 EXC 6	31	-34 1 INV SUM 1
07	32 5 STO 5	32	52 7 STO 7
08	43	33	52
09	33 4 RCL 4	34	33 0 RCL 0
10	32 1 STO 1	35	44
11	61 8 SBR 8	36	-18 INV LOG
12	38 6 EXC 6	37	34 6 SUM 6
13	32 1 STO 1	38	-89 INV 8
14	33 6 RCL 6	39	32 3 STO 3
15	32 3 STO 3	40	-49 INV INT
16	61 8 SBR 8	41	-34 1 INV SUM 3
17	32 6 STO 6	42	74 DE
18	44	43	32 XIT
19	33 5 RCL 5	44	34 5 SUM 5
20	85	45	-56 INV EQ
21	91 R/S	46	51 8 CTD 8
22	71 RST	47	01 1
23	86 8 LBL 8	48	75 8
24	19 CT	49	51 8 CTD 8

Mästermind för TI-57 (Peter van Roy ut TI PPC Notes v6n6/19). Bruksanvisning:

1. Initiera: 10 STO 0 0 STO 6 RST.
2. Be någon mata in ett hemligt tal bestående av siffrorna 1 - 9. (Eller gör det själv utan att titta. Men håll då fingrarna borta från "0"-tangenten.) Tryck STO 2 CLR.
3. Mata in glesning och tryck R/S. Resultatet visas på formen b.c, där b är antal svart pinnar, ds antal siffror i rätt position, och c är antal vita pinnar, ds antalet riktiga siffror utan hänsyn till position. (Även siffror som står i rätt position och alltså ger en svart pinne ger även en vit pinne.)
4. Upprepa steg 3 tills du har listat ut talet.
5. Gå till steg 2 för nytt spel.

Anmärkning: Instruktionsboken går helt i stil med räknaren och kommer inte att göra det lätt för nybörjaren som skall lära sig programmera. Den är för det första alldeles för kortfattad och innehåller för få exempel. För det andra är översättningen slarvigt gjord. Det finns alldeles för mycket engelska ord kvar i den, och även engelsk meningsbyggnad har trängt igenom till översättningen. Vissa ord är enkla att ersätta med bra svenska ord som t ex dekrementera (minsk), modifiera (ändra) och makulera (ta bort). (Dsz definieras som: "Dekrementera och 'skip on zero'".)

Jag vet mycket väl att vissa ord är svåra att

översätta. Men om ett främmande ord används (t ex exekvera) bör det åtminstone förklaras första gången det används.

Översättningen innehåller också korrekturfel och några sakfel. Det engelska "partition" har fått två olika översättningar: "avdelning" och "uppdelning". Ett ord räcker.

Priset är 345 kr, vilket även TI-57 kostar. Därför skulle jag rekommendera att köpa den gamla 57:an i stället. Passa på medan den finns kvar. (Det går nog att få tag på TI-57 billigare nu, eftersom handlarna vill tömma lagren.)

15 ► HEXKODER

Dsz 01 08: om R01 ≠ 0 körs SBR A och programmet

Återvänder till steg 001; om däremot R01 = 0 körs SBR A men programmet Återvänder till steget efter 08.

EQ 0B: om x=t körs SBR A och exekveringen fortsätter på steg 001; om däremot x≠t körs SBR A men exekveringen fortsätter på steget efter 08.

Till sist koderna för Fast and Graphics Mode än en gång:

OC (h12) för över räknaren i Fast Mode. IF (h25) avbryter utskriften med Op 05 och Åstadkommer därmed Graphics Mode.

Artikeln bygger på artiklar ur TI PPC Notes enligt Källförteckningen nedan, brev från Robert Prins (Robert har bl a kommit på hur STF ind 7 EE sekvensen kan utföras utan att Trace ställs in) samt egna undersökningar. Samtliga program har jag gjort själv.

KÄLLFÖRTECKNING

LRN v6n6/7p11: Dave Leising, Graphics Mode
LRN v6n9/10p14-15: Patrick Acosta, Hexadecimal Keycodes
LRN v7n7/8p18-19: Patrick Acosta, Creating Hex-keycodes
(LRN = TI PPC Notes)

VIDARE LÄSNING

Komplement till instruktionsboken
PB 81-3 s 29-32: Björn Gustavsson, Mer om Fast Mode -2
PB 82-1 s 18-20: Björn Gustavsson, Nyheter på TI-58C
PB 80-3 s 14-15: Lars Hedlund, Djupdykning i TI-59

SNABBTRITARE

Text: Lars Hedlund

Denna "snabbritare" har sänts till oss av Dejan Ristanović, Gvozdićeva 34, YU-11000 Belgrad. Det är ett intressant program bl a därför att det samtidigt utnyttjar Fast Mode och Graphics Mode. Vad det gör är att rita (i princip) oändligt stora teckningar av typ Musse Pigg på omslaget till PB 81-4.

Så här skriver Dejan i utdrag och fri översättning:

Programmet består av två delprogram och data. Allt sammans går in på två magnetkort. Inspelningsprogrammet och början av ritprogrammet (t o m steg 173) spelas in på ett kort i normalupplösning. Ställ sedan in på 17 upplösning och knappa in slutet på ritprogrammet och data i R30 - R61 samt R68 (4 E 12 + 2). Återgå till 6 op 17 och spela in sidorna 2 och 3 på det andra kortet.

För att en teckning skall kunna lagras måste den vara gjord på rutat papper. Markera kolumnar med 5 rutors bredd (var fjärde kolumn = 20 rutor markeras starkare), motsvarande skrivregister och hela remсор.

För att lagra teckningen matar man in grupper om fem ettor eller nollor där ettan motsvarar rutan med tecken och nollan tom ruta. Normalt trycker man därefter A (även för den sista 5-gruppen på raden) men om det inte finns fler teckenprogram på raden kan man trycka B och därmed direkt göra radskift - även för hel tomrad. Observera ordningsnumret för teckenprogram i displayen och att detta nummer inte duger som indata ens om det skulle vara 1 utan uppfattas som 0. Om man råkar slå in mer än fem tecken före A eller B används bara de fem sista.

Den sista raden måste avslutas med A eller B. När en remslängd är klar, tryck C. Displayen visar då 1 och det är klart att börja med nioasta 20-teckens kolumn. När hela teckningen är klar, tryck E. Räkaren arbetar då en stund men displayen förblir släckt, vilket betyder att ett kort skall sättas in för inspelning. Detta kan också inträffa (vid en större teckning) redan innan E har hunnit tryckas; om displayen är släckt länge skall alltså en kortsida spelas in (och lämpligen märkas!).

Upprättningen av figuren startar med att räkaren slås på varefter ritprogramms kortsidor 1, 2 och 3 läses. Tryck E. Hela initieringsproceduren för Fast Mode och Graphics Mode skrivs ut. Utöver den från tangentbordet (blinkningar ignoreras!). Här efter slöcknar displayen och det är dags att läsa första kortsidan med data. Upprättningen sätter i gång. Om den stannar och displayen slöcknar, läs nästa datakort.

När allt är upprättat, klipp vid märket "-" och (hyfsa remskanter och) klistra ihop remсорna.

Om man vill ha en titel under teckningen eller något sådant finns stegen 176 - 298 disponibla. Kom ihåg att stegnumreringen ändras två steg vid initieringen. Återgång till teckenritningsprogrammet görs med GTO 118. För inga subrutiner under inmatning av en rad.

Jag fick idén till det här programmet när jag analyserade Frederic de Mees' Musse Pigg-program (omnämnt men ej återgivet i PB 81-3 s 25). Delar av detta program kommer därför från. Peter Poloczek har redan tidigare gjort program med både Fast och Graphics Mode.

Jag tar gärna emot förslag till ändringar av programmet (adressen stöter i början!). Ha så trevligt med programmet och sänd gärna några bra teckningar!

Så skrev Dejan. Eftersom vi i PB 81-4 efterlyste program av den här typen gör jag här ett litet tillägg.

Dejans register 30 - 61 (32 st!) innehåller som synes skrivorden "11" (dvs 8) ordnad som de binda talen 0 - 11111, dvs 0 - 31. Om man analyserar datalagringen som görs i inspelningsprogrammet finner man att den sker som tolvstiffriga tal med fyra grupper om tre siffror där de två första är registernumret för resp. skrivord och den tredje "op"-numret eller "9" för en avslutad rad. Vad jag menade i PB 81-4 s 25 var att jag undrade om inte denna lagring kunde göras ännu kompaktare.

Allra mest kompakt kan man givetvis göra den om skrivorderna lagras som talen 0 - 31, där 0 är ett kort skrivregister och 31 det binda talet 11111. Eftersom 32⁸ rymmer i ett register ruckar detta för två rader, vilket jag skulle vilja påstå är mer ekonomiskt för en normalt teckning än Frederic de Mees' metod och dessutom mycket enklare (Op-numren bortfaller).

Ännu enklare inmatningsprogram får man om man förutsätter att operatören själv någorlunda snabbt kan översätta ett binärt tal under 11111 till decimalt vilket inte kräver så värst mycket! Det är ju 30 tal att läsa sig, där första siffran är 16, andra 8 osv. Inmatningen går då oerhört mycket snabbare - antingen med ett tal 0 - 31 och R/ eller fyra tal på en gång som ett åttastiffrigt tal och R/ vilket ger bättre överblick men kräver längre program och beräkningstid.

Det var den "utmaningsuppgiften" jag tänkte mig i PB 81-4. Kanske någon känner för den efter att ha sett Dejans program? På annat ställe i det här numret har vi dessutom en artikel om hexkoder som öppnar nya möjligheter för initieringen till Fast och Graphics Mode.

På julnumrets omslag brukade vi traditionellt ha något julmotiv. Hugade tecknare inbjöds härmed till en

TÄVLING
om bästa omslagsteckningen i Grafisk mode. (Tänk på hur det tillgängliga utrymmet ser ut - förmåkan kan vi göra!) Sänd in ditt bidrag före den 15 november 1982. Som första pris uppsätts medlemskap 1983 och ett paket magnetkort - möjligen utdelas även tröstpris!

000 76 LBL	060 73 39	120 01 01	180 25 +
000 12 5	061 65 +	121 37 17	181 01 +
002 99	062 01 1	122 01 1	182 22 EE
003 5BR	063 00 0	123 52 EE	183 05 +
004 01 01	064 65 +	124 09 9	184 95 +
005 80	065 02 2	125 32 17	185 22 HW
006 22 HW	066 45 +	126 73 RC	186 59 INT
007 06 STF	067 43 RCL	127 32 32	187 45 +
008 00 00	068 38 38	128 77 GE	188 01 1
009 42 STD	069 45	129 02 62	189 02 62
010 39 39	070 52 EE	130 07 07	190 05 5
011 05 5	071 44 SUM	131 01 01	191 02 62
012 44 SUM	072 37 37	132 00 0	192 22 HW
013 01 31	073 01 1	133 00 0	193 02 62
014 61 GTO	074 44 SUM	134 00 0	194 92 RTH
015 00 00	075 38 38	135 44 P9A	195 76 LBL
016 41 41	076 43 RCL	136 32 32	196 10 E*
017 74 LBL	077 39 39	137 25 LBL	197 01 1
018 11 8	078 22 HW	138 63 RCL	198 07 7
019 91 91	079 47 67	139 01 1	199 49 OP
020 71 8BR	080 00 00	140 92 RTH	200 17 17
021 01 01	081 49 49	141 00 0	201 01 1
022 80 80	082 03 3	142 13 C	202 42 STD
023 86 STF	083 00 0	143 25 LBL	203 31 31
024 00 00	084 32 17	144 01 1	204 94 +
025 42 STD	085 43 RCL	145 03 3	205 42 STD
026 39 39	086 37 37	146 01 1	206 32 32
027 04 4	087 45	147 74 LBL	207 01 1
028 32 17	088 22 HW	148 14 8	208 44 SUM
029 43 RCL	089 07 07	149 74 LBL	209 42 STD
030 31 31	090 00 00	150 32 32	210 25 LBL
031 10 HW	091 00 00	151 41 GTO	211 00 00
032 67 67	092 96 96	152 01 1	212 32 32
033 00 00	093 47 67	153 18 18	213 02 62
034 41 41	094 01 01	154 76 LBL	214 09 9
035 09 9	095 05 05	155 15 15	215 32 32
036 42 STD	096 01 1	156 25 LBL	216 43 RCL
037 31 31	097 00 0	157 01 1	217 32 32
038 22 HW	098 95 +	158 01 1	218 22 HW
039 86 STF	099 43 RCL	159 04 4	219 67 67
040 00 00	100 31 31	160 14 8	220 01 1
041 25 LBL	101 41 41	161 41 41	221 01 1
042 29 CF	102 74 SUM	162 32 32	222 09 9
043 42 STD	103 32 32	163 02 62	223 01 1
044 38 38	104 22 HW	164 25 LBL	224 09 9
045 03 3	105 42 STD	165 60 60	225 42 STD
046 00 0	106 01 01	166 02 62	226 29 29
047 42 STD	107 42 STD	167 25 LBL	227 42 STD
048 37 37	108 01 1	168 01 1	228 04 4
049 01 01	109 11 11	169 02 62	229 42 STD
050 00 00	110 31 31	170 99 9	230 01 1
051 22 HW	111 01 01	171 01 01	231 01 01
052 49 PRD	112 00 00	172 77 GE	232 02 62
053 39 39	113 01 01	173 01 01	233 04 4
054 43 RCL	114 18 18	174 22 22	234 01 01
055 22 HW	115 42 STD	175 01 01	235 01 01
056 29 INT	116 42 STD	176 14 14	236 01 01
057 29 INT	117 42 STD	177 01 01	237 01 01
058 22 HW	118 01 01	178 01 01	238 01 01
059 44 SUM	119 01 01	179 01 01	239 01 01

PS. Det finns ju andra varianter till utmaningsuppgiften ovan som kanske är intressanta, t ex att göra teckningar med flera grönor genom tre olika tecken med olika antal punkter vilket går att göra både i normal och Graphics Mode. Kompakt lagring uppnås med 0, 1, 2 och 3 i 20 positioner (en rad) i ett register (4 s 1, 10¹²). Rätt skrivtecken hämtas med RCLIND. Ett sådant program blir givetvis mycket längre och långsammare än den enkla snabbritaren jag nyss föreslog, men kanske grönorna ändå gör det intressant? Naturligtvis kan sådana teckningar också prisbelönas.

0.	30	1100110000.	50
11.	31	1100110011.	51
1100.	32	1100111000.	52
1111.	33	1100111111.	53
111000.	34	1110000000.	54
110011.	35	1111000011.	55
111111.	36	1111001100.	56
111111.	37	1111001111.	57
11100000.	38	1111100000.	58
11110001.	39	1111100011.	59
11001111.	40	1111110000.	60
11111111.	41	1111111001.	61
11100000.	42	1111000000.	62
11100011.	43	1111000011.	63
11111000.	44	1111000000.	64
11111111.	45	1111000000.	65
1100000000.	46	1100000000.	66
1100000011.	47	1100000000.	67
1100001000.	48	40000000000002.	68
1100011111.	49	1100011111.	69

Hed till t.v. inspelningsprogram, t.h. ritprogram
(delen 000-173 lagras på samma kort som inspelningsprogrammet, delen 334-389 och registren 30-69 på ett annat kort).

000 83 GD*	060 03 3	120 43 43	340 31 P-5
001 69 69	061 00 0	121 01 01	341 75 LBL
002 76 LBL	062 00 0	122 44 SUM	342 15 E
003 11 8	063 32 17	123 63 63	343 25 LBL
004 25 CLR	064 03 3	124 04 4	344 07 7
005 07 07	065 22 HW	125 42 STD	345 00 0
006 69 DP	066 28 LDB	126 62 62	346 17 17
007 17 17	067 95 +	127 61 GTO	347 49 DP
008 98 RDV	068 59 INT	128 00 00	348 00 00
009 19 19	069 42 STD	129 43 43	349 00 00
010 29 INT	070 00 00	130 62 62	350 03 3
011 96 INT*	071 00 00	131 00 00	351 00 00
012 04 4	072 42 STD	132 69 DP	352 05 5
013 42 STD	073 00 00	133 04 04	353 00 00
014 62 62	074 25 CLR	134 69 DP	354 07 7
015 41 GTO	075 00 00	135 00 00	355 00 00
016 00 00	076 25 +	136 69 DP	356 06 6
017 32 32	077 00 00	137 00 00	357 00 00
018 76 LBL	078 00 00	138 61 GTO	358 06 6
019 49 DP	079 00 00	139 00 00	359 00 00
020 25 CLR	080 42 STD	140 18 18	360 00 00
021 49 DP	081 00 00	141 97 STD	361 00 00
022 05 05	082 22 HW	142 69 69	362 05 05
023 68 NDB	083 00 00	143 00 00	363 00 00
024 74 SUM	084 65 +	144 58 FIF	364 90 LST
025 80 80	085 02 2	145 00 00	365 00 00
026 00 00	086 75 +	146 43 RCL	366 61 GTO
027 00 00	087 48 48	147 69 DP	367 00 00
028 00 00	088 82 RIR	148 66 BEG	368 24 24
029 00 00	089 84 STF	149 86 STF	369 00 00
030 00 00	090 95 +	150 40 INT	370 12 12
031 00 00	091 00 00	151 12 12	371 00 00
032 25 LBL	092 05 5	152 37 P/F	372 04 04
033 42 STD	093 00 00	153 82 EE	373 00 00
034 43 63	094 42 STD	154 30 TWH	374 91 P-8
035 00 00	095 42 STD	155 00 00	375 00 00
036 00 00	096 73 RC*	156 00 00	376 31 LBN
037 04 04	097 61 GTO	157 00 00	377 00 00
038 03 3	098 84 STF	158 00 00	378 31 LBN
039 41 GTO	099 65 +	159 00 00	379 00 00
040 01 01	100 29 CF	160 01 1	380 25 LBL
041 42 42	101 37 37	161 00 00	381 00 00
042 162 162	102 14 14	162 42 STD	382 61 GTO
043 92 EE	103 00 00	163 00 00	383 00 00
044 22 HW	104 01 01	164 16 R*	384 52 52
045 38 LDB	105 00 00	165 00 00	385 00 00
046 52 EE	106 01 1	166 00 00	386 12 12
047 25 LBL	107 00 00	167 00 00	387 14 R*
048 57 FNG	108 01 1	168 01 1	388 04 04
049 68 INT	109 00 00	169 00 00	389 00 00
050 37 37	110 69 69	170 09 09	390 91 P-8
051 85 +	111 02 02	171 00 00	391 00 00
052 39 39	112 81 RST	172 01 01	392 31 LBN
053 02 02	113 00 00	173 00 00	393 00 00
054 73 RC*	114 25 CLR	174 76 LBL	394 31 LBN
055 63 63	115 82 RTH	175 00 00	395 00 00
056 95 +	116 16 R*	176 25 CLR	396 23 INT
057 00 00	117 00 00	177 00 00	397 00 00
058 59 INT	118 62 62	178 69 DP	398 11 8
059 45 +	119 00 00	179 17 17	399 98 RD

En jämförelse mellan TI 59 och TI 88

av Maurice Swinnen, TI PPC Notes v7n7/8p3-7 i fri översättning och bearbetning av Björn Gustavsson

Skillnaden mellan de båda räknarna TI-59 och TI-88 är väl det som intresserar oss TI-58/59-användare mest: "Vilka nya trick måste vi lära oss för att kunna programmera 88:an med någon framgång?" Det finns en hel del skillnader, har jag upptäckt. Vissa av dem kommer som en angenäm överraskning och får en att utbrista: "Varför kunde de inte ha gjort det möjligt på 59:an också?" Tja, tekniken går framåt hela tiden. Kanske var det inte ekonomiskt möjligt att bygga in den finessen i TI-59, även om den var känd då.

Det första som slår en när man äntligen får tag på en TI-88 är den extrema lättheten med vilken man matar in alfanumeriska tecken. Man trycker bara på en tangent märkt ALPHA och helt plötsligt är räknaren förvandlad till en (kanske primitiv) skrivmaskin. Adjö till alla skrivkodstabeller för att mata in skrivkod. Om man nu trycker A, får man ett A i displayen och utskrivet på skrivaren om så önskas. Tryck ALPHA igen och den gamla räknaren med dess

många funktioner kommer tillbaka. Så enkelt är det. Displayen visar alfanumeriska tecken, siffror eller en blandning av båda. Prt trycker ut precis vad som visas i displayen.

Så fort man börjar programmera lägger man märke till fler skillnader, som t ex det utökade AOS (kallas EAOS eller EOS). Som vi alla vet baseras prioriteterna för beräkning på matematiska regler. Men dessa prioriteter har ändrats. Nedan visas en jämförelse mellan de vi är vana vid för TI-59 och de nya för TI-88.

Men sättet på vilket vissa funktioner måste matas in skiljer sig drastiskt från vad vi är vana vid. På 59:an matade vi in sinus för 30° som 30 sin och logaritmen för 2,34 som 2.34 log. På 88:an görs inmatningen som sin 30 = och log 2.34 =. Om man tänker kritiskt på det måste man hålla med om att inmatningen 30 sin på TI-59 egentligen var en sorts Omvänd Poisk notation. Fast det kommer TI aldrig att erkänna. Med den nya utökningen kan vi verkligen påstå att vi använder ett algebraiskt system.

Potenser och rötter måste också behandlas annorlunda, på TI-88 används pilar som pekar upp och ner. För att göra det lättare att skriva på skrivmaskin använder vi "up" för "+" och "dn" för "-". Det som var a^b b = på 59:an matas in som a up b dn, där den sista pilen ner avslutar operationen. Den här formen av exponent- och rotinmatning gör vissa knep möjliga vid invecklade exponenter av formen

$$3^{4+2}$$

som matas in som 3 up 4 up 5 + 2 dn dn. Rötter matas in som exponenter mindre än ett. Fjärde roten ur fem t ex måste skrivas om till $5^{1/4}$ och matas in som 5 up .25 dn. Man kan också mata in det som 5 up 4 ()⁻¹ dn, där ()⁻¹ har samma betydelse som TI-59:s 1/x. För beräkning av potenser med heltalsexponent från 0-9 kan tangenten ()^N användas. 5⁴ t ex beräknas med sekvensen ()^N 4; beräkningen startar omedelbart när man tryckt 4. I ett program lagras sekvensen som "5^{4N}".

Statistiksummeringen, Σ , görs nu till HIR-register i stället för till dataregister som i TI-59. Det frigör fler register, som kan användas som arbetsregister.

Att anropa moduler kräver lite mer knappnåpande än på TI-59, som ju bara har en modulhållare och ett anrop som helt enkelt består av Pgm nn. Eftersom TI-88 har två modulhållare måste man använda anropsformen Pgm mmm, där mm är modulnummer och nn numret på 88skat program. I motsats till TI-59 kan man på TI-88 själv placera program i en CRAM-modul och anropa dessa på samma sätt.

Nollställningsfunktionerna har också ändrats. På TI-59 finns ju både CLR och CE. På TI-88 finns endast tangenten CE/C. Trycks den ner en gång fungerar den som CLR - dvs endast en raderas - och trycks den ner två gånger fungerar den som CLR. (I program används därmed CE och CLR. CE åstadkommes genom att trycka CE/C en gång; CLR genom att trycka CE/C två gånger, vilket ger CE CLR i programmet varefter man raderar CE manuellt.)

Även registeradressering skiljer sig. Det finns fler än 100 register, vilket betyder att tre siffror måste användas. Vi skriver alltså RCL 126 eller STO 007. Men, för att göra det lättare, kan man adressera de 26 första registren med A till Z, t ex RCL H eller Exc C. (Det som vi kallade registernumren på TI-59 kallas här registers adress.) I ett program lagras RCL H som H. Vid indirekt adressering när man skriver t ex RCL Ind A lagras räknaren det som RCL A. För att spara ett steg kan man manuellt radera RCL och då endast a, som fungerar alldeles utmärkt.

Testerna är också lite annorlunda: På TI-59 måste varje test följas av en hoppadress (label eller absolutadress). Om villkoret är uppfyllt sker ett hopp till den adressen. Annars fortsätter bara programkörningen. På TI-88 är dessa test lite mer flexibla. Om villkoret är uppfyllt utförs instruktionsblocket (= en fullständig instruktion med adress, t ex RCL 007) som följer direkt efter testet. Om det ej är uppfyllt överhoppas instruktionsblocket och exekveringen fortsätter på nästföljande instruktion.

Man kan också koppla ihop flera test så att var och en fungerar som ingången till en ELLER-grind. Det betyder att om ett eller flera villkor är uppfyllda så utförs det första instruktionsblocket efter testerna. Om inget villkor är uppfyllt överhoppas instruktionsblocket.

Andra nyheter är att displayen inte endast kan testas mot ett T-register som på TI-59. Testen

kan utföras mot alla register, t om HIR-register! Dessutom finns alla sex test som kan utföras mellan två tal: $=, >, \geq, <, \leq, \neq$. På TI-59 kan endast de fyra första utföras.

Instruktionen Op finns kvar på TI-88 men alla koder är annorlunda och många fler. De gamla Op 20 till Op 39 är borta. I stället måste ST+ och ST-, vilket motsvarar TI-59:s SUM och INV SUM, användas. För minskning kan också Dst användas, t ex Dst A Kop (i vissa fall kan Kop tas bort). Men det finns 88 st andra Op-koder. En lista över Op-koderna finns i PB 82-2 s 37. Alla definitioner för Op-koderna kan ställas med Op 00. Definitionen för Op nn visas om man trycker INV Op nn.

Labl är en annan sak som är annorlunda. På TI-59 används de användardefinierade tangenterna A-E och A'-E'. På TI-88 används A-J. På TI-59 kan nästan alla tangenten användas som labl. Så inte på 88:an. I stället används lablarna A till Z och de numeriska lablarna 00 till 99. Vår särskilda instruktion används för att hoppa till en numerisk label: GTO Lbl och RSP Lbl som i program kopplas till Gtl resp Sbl.

Direkt (absolut) adressering kan också användas, men nu med en fyrsiffrig adress. Förutom detta finns också GÅ FRAMÅT RELATIVT (OFR) och GÅ BAKÅT RELATIVT (GBR). Det betyder att man kan hoppa framåt eller bakåt ett visst antal steg räknat från positionen för GFR eller GBR up till 99 steg. Våldigt bekämt när man vill ha snabba hopp och man planerar att göra många insättningar och raderingar i resten av programmet. Naturligtvis får man inte redigera mellan "nuvarande position" och målet för det relativa hoppet. Men man kan redigera hur mycket som helst i resten av programmet. HP-räknare har haft den finessen i flera år (men den saknas på HP-41C/CV).

Uppdelning görs nu så att man anger hur många programsteg man vill ha, i stället för antalet register.

Som tidigare sagt kan TI-88 trycka allt som finns i displayen. Detta fungerar tvärtom också. Den visar allting som man trycker ut med Prt, även utan skrivare ansluten. Texten blir kvar i displayen medan programmet fortsätter att exekveras ända tills den skrivs över med ny text. Den finessen kommer onkligt att spara många rullar skrivarpapper.

TI-59	TI-88
Funktionsnamn	Funktionsnamn
Inverterat värde	Inverterat värde
Kvadratrot	Kvadratrot
Kvadrat	1 Heltalspotens
Absolutbelopp	Absolutbelopp
Signumfunktion	Signumfunktion
Tiologarit	
1 Naturlig logarit	Kvadratrot
Dessas antilogaritmer	Tiologarit
Sin, cos, tan med	Naturlig logarit
inverser	Dessas antilogaritmer
Heltalsdel	2 Sin, cos, tan samt
Decimaldel	inverser
	Heltalsdel
	Decimaldel
2 Potenser och rötter	
3 Multiplikation	3 Multiplikation
Division	Division
4 Addition	4 Addition
Subtraktion	Subtraktion
5 Likamed	5 Likamed

Primtalsprogram på TI 88

av Björn Gustavsson

För att visa vad som kan göras med TI-88 har jag gjort om pgm 2 i Utmaningen 80-4 s 4 till TI-88. Programmet Återges här som pgm 1, medan pgm 2 är TI-88-programmet.

Pgm 1 kontrollerar om ett tal är primtal eller sammansatt genom att dela det med alla tal som är mindre än talets kvadratroten och ej delbara med 2, 3 eller 5. Programmet "hjärta" är stegen 000-015, där först testnämaren i R_0 ökas och sedan talet som skall testas (i R_6) delas med testnämaren.

Kvotens decimaldel testas sedan med sekvensen INV Int 1/x (steg 012-014). Om decimaldelen är noll skapas en felindikering och programmet stoppas eftersom flagga åtta är ställd. Mitt i divisionen testas också om testnämaren är större än kvadratroten ur talet. (T-registret innehåller roten ur talet ökad med ett litet tal. Detta är nödvändigt eftersom testet "större än eller lika med" vid steg 007 borde vara "större än" som saknas på TI-59.)

TI-88-programmet används på följande sätt:

1. Tryck CONT. Först visas "PRIMTALSPGM" i en paus; därefter visas "Tal:".
2. Mata in ett heltal och tryck ENT. Medan 88:an räknar visas "n=" och talet som matades in.
3. Om talet är ett primtal visas "Prim" och talet. Om talet är sammansatt visas "Sam" och talet.

21

EE INV EE avrundar inte längre värdet i X-registret, men OP 13 gör det i stället. T-registret är borta (vila i frid!) eftersom testerna kan göras mot alla register. Det finns därmed ett hjälpregister som nås på samma sätt som T-registret och som används för statistik, P/R- och R/P-omvandlingar, kassetthantering och hantering av in/ut-porten.

Slutligen finns trace-tangenten på räknaren i stället för på skrivaren. Detta gör det möjligt att använda trace utan skrivare.

Förutom alla dessa skillnader finns det en mängd nyheter som vi kommer att ta upp i kommande artiklar.

Vi skall nu titta på hur programmet arbetar. När man trycker CONT anropas Lbl E vid steg 0044 (E är placerad på samma tangent som CONT). Sto A lagrar X-registrets innehåll i R_0 . (Varför det görs framgår längre fram.) OP 1 ersätter följande sekvens: CE/C CE/C INV EE Inv Eng Inv Fix Deg OP 15 (redigerad visning av tal som på TI-59) OP 17 (decimalt tillstånd i motsats till hexadecimant) OP 22 (pauslängd är 1,5 s) OP 25 (utför ej underförstådd multiplikation) OP 49 (andra uppdelning till 480 steg) OP 52 (hård uppdelning - som på TI-59 - i motsats till mjuk uppdelning som på SR-52) Inv STP F (stannar ej vid fel vid exekvering) Inv HIR (koppla bort hir-tillstånd) Inv Indh (koppla bort indirekt hir-tillstånd). Det är alltid klokt att låta OP 1 ingå i inledningen för undvika problem orsakade av tidigare inställningar.

Sto A nollställer R_0 (displayen nollställs av OP 11). Därefter visas rubriken "PRIMTALS-PGM" i en paus. CE vid steg 0067 raderar alfa-registret innan "Tal:" matas in.

OP 6 är en av "ledtext"-funktionerna och definieras som "ENT/CONT CUE". När OP 6 påträffas visas alfaregistret och användaren kan mata in ett tal och trycka ENT eller CONT. Om ENT trycks fortsätter programkörningen på steget efter OP 6, i det här fallet 0075. Annars, om CONT trycks, överhoppas den första fullständiga ga instruktionen efter OP 6, i det här fallet Sto A.

Sekvensen OP 6 Sto A resulterar alltså i att en tryckning av ENT lagrar talet i X-registret i R_0 och en tryckning av CONT inte påverkar R_0 .

Alternativt kan det här programmet köras genom att mata in talet och trycka CONT två gånger. Den första tryckningen anropar Lbl E, medan räknaren "kommer ihåg" den andra och direkt fortsätter programkörningen vid OP 6.

0040 76 LBL 017 43 RCL 034 09 08 051 04 4
0041 18 E 018 25 CLP 035 01 1 052 15 E
0042 44 SUB 019 43 RCL 036 45 STO 053 02 2
0043 07 07 020 06 06 037 07 07 054 15 E
0044 43 RCL 021 41 LBL 038 15 E 075 04 +
0045 06 06 025 76 LBL 039 01 1 056 15 E
0046 22 + 042 15 H 043 12 E 057 06 +
0047 43 RCL 024 50 151 041 02 058 15 E
0048 07 07 035 45 STO 042 15 E 059 02 2
0049 77 CE 026 06 06 043 02 060 15 E
0050 43 RCL 012 44 72 046 12 E 061 06 +
0051 35 029 05 045 76 LBL 062 15 E
0052 22 INV 030 05 + 046 04 +
0053 22 2 031 32 INT 048 15 E 063 61 STO
0054 76 LBL 032 25 CLP 049 02 064 14 D
0055 43 RCL 038 48 STP 050 15 E

Pgm 1

PROGRAMBITEN 82-3

Vid steg 0077 återkallar instruktionen A innehållet i R_0 (denna instruktion matas in som Rcl A och räknaren tar bort Rcl). ALPH sätter räknaren i alfaläge och kopierar samtidigt talet i det numeriska displayregistret till alfaregistret. Upp till fyra tecken kan sedan matas in till vänster om talet. "n=" matas sålunda in och ger som resultat t ex "n= 41.".

Utän skrivare fungerar Prt så att displayens innehåll visas under programkörningen. De följande stegen lagrar kvadratroten ur talet i R_0 . Sedan följer samma serie av subrutin-anrop som i TI-59-programmet men till Lbl D i stället för till Lbl E. Anropet lagras som "D" för att skilja det från D (=Rcl D=Rcl 3). Vid steg 0113 åstadkommer GBR 19 ett hopp 19 steg bakåt till steg 0097. (Det relativa hoppet räknas från steg efter GBR 19, i det här fallet 0116.)

Nu är det dags att titta på subrutinen vid stegen 0000-0032. St+ B vid steg 0002 motsvarar 59:ans SUM 1. Därefter kommer ett uttryck som egentligen borde skrivas Frc(A DIV B) = (motsvarar RCL O DIV RCL 1 = INV Int på TI-59) men som förkortas och dessutom blandas testet If> D Gtl 0 in mitt i beräkningen. If> D testar om X-registret är större än R_0 och utför nästa fullständiga instruktion om så är fallet, annars överhoppas denna.

Nästa test If# C Rtn testar om X-registret är skilt från R_0 och utför om så är fallet ett återhopp till huvudprogrammet. I annat fall är talet sammansatt och sekvensen A ALPH SAM ALPH ger sedan texten "Sam" och till höger därom talet.

Sekvensen CLR #267 är lite knepigare. Den "lura" räknaren att inget subrutinanrop har gjorts genom att lagra en nolla direkt i siffrorna 7 i HIR-register 26 (subrutinstackens stackpekare). Vi kommer att ta upp HIR-instruktioner mer detaljerat i kommande nummer av PB. Instruktionsboken för TI-88 beskriver till skillnad från den för TI-59 vilka HIR-instruktioner som finns och hur vissa av dessa används internt, men det finns fortfarande mycket kvar att utforska.

Vid steg 0030 återkallar OP 9 alfaregistret till displayen - CLR tidigare i programmet raderade ju displayen. Vid Rtn stoppas exekveringen.

Vi har fått frågan: "Hur mycket snabbare är TI-88 jämfört med TI-59?" från flera medlemmar. Enkla test med slingor visar att TI-88 är åtminstone dubbelt så snabb som TI-59, och åtminstone 50 procent snabbare än HP-41C. Sådana enkla tester säger givetvis inte allt om hastigheten. Stort minne och avancerade instruktioner ökar hastigheten.

En jämförelse mellan pgm 1 och pgm 2 ger för printalet 14215469 ett hastighetsförhållande på 1:2,4. Pgm 1 exekveras på 13 min 13 s och pgm 2 på 5 min 25 s.

Om man tittar på hur många programsteg som programmet tar upp så ser man att TI-88-programmet är nästan dubbelt så långt. Men det ger också ledtexter i displayen som saknas i TI-59-programmet. Även några andra finesser finns. Om man plockar bort alla ledtexter och alla extra finesser blir TI-88-programmet lika långt som TI-59-programmet (med en felmarginall på några steg eftersom jag inte har provkört det programmet).

0000 Lbl D	0041 ALPH	0081 ALPH
0002 St+ B	0042 Gtl 2	0082 Prt
0004 Frc	0044 Lbl E	0083 -
0005 C	0046 Sto A	0084 -
0006 A	0048 OP 1	0085 Sto D
0007 DIV	0050 Sto C	0087 I
0008 B	0052 ALPH	0088 Sto B
0009 If> D	0053 F	0090 D'
0011 Gtl 0	0054 R	0091 I
0013 =	0055 I	0092 D'
0014 If# C	0056 M	0093 2
0016 Rtn	0057 T	0094 D'
0017 A	0058 A	0095 2
0018 ALPH	0059 L	0096 D'
0019 S	0060 S	0097 4
0020 a	0061 P	0098 D'
0021 m	0062 G	0099 2
0022 ALPH	0063 M	0100 D'
0023 Lbl 2	0064 ALPH	0101 4
0025 CLR	0065 Pau	0102 D'
0026 n	0066 ALPH	0103 2
0027 2	0067 CE	0104 D'
0028 6	0068 T	0105 4
0029 7	0069 a	0106 D'
0030 OP 9	0070 1	0107 6
0032 Rtn	0071 :	0108 D'
0033 Lbl 0	0072 ALPH	0109 2
0035 A	0073 OP 6	0110 D'
0036 ALPH	0075 Sto A	0111 6
0037 P	0077 A	0112 D'
0038 r	0078 ALPH	0113 GBR 19
0039 i	0079 n	
0040 m	0080 =	

Pgm 2

PROGRAMBITEN 82-3

Varför inte BASIC?

av Gösta Blume

I P882-2 skriver Björn Gustavsson och Lasse Hedlund om den nya TI-88 bl a "Språket bygger på 59:ans språk och är alltså inte BASIC (som tur är)." Det är lätt att förstå deras inställning till BASIC: båda tillhör eliten av dessa pgnbitna individer, som på jakt efter dolda skatter gör djupdykningar i räknares skrymman och försöker få räknares att göra saker som de (skänks, alltså) inte är gjorda för att göra (fast mode, direktgång i MIR, syntaktisk pgnering och annat som här den högre skolan till). För dem är pgnering en konst för konstans egen skull. Snabbare, kortare, elegantare, det är vad de vill lära oss andra. Vad pgn sedan handlar om är, något överdrivet, av sekundär betydelse.

Troligen finns det bland våra medlemmar en del som, samtidigt som de tacksat tar emot och försöker använda vad dessa skänks erbjuder oss, har intresse av att göra pgn för den nytta eller det nöje som de får. De pgn kan ge dem. För dem är huvudsaken att pgn kan tillverkas med ett minimum av möda och tidsutdräkt och sedan fungerar tillfredsställande. Och för dem borde ett enkelt högnivåspråk som BASIC kunna vara av intresse. Eftersom jag, mer av nyfikenhet än något annat, har skaffat mig en BASIC-språkig fickräknare skall jag, med REDs tillåtelse, här meddela några synpunkter på och vad jag ser för dem.

Det finns åtminstone ett par sådana räknare på marknaden. Jag kommer att nämna mig till minCASIO FX-702P (som jag i det följande kallar "702an"). Den är verkligen en fickräknare: 18x165 (bredd)x82 mm, vikt 176 g.

Displayen, LCD, rymmer 10 till 20 tecken. Inmatade pgn och data kvarstår efter avstängning. Batterierna är inte laddningsbara, ingen elminator.

702an är gjord främst för matematik: alla Tis funktioner finns med och dessutom hyperboliska funktioner (sinh, cosh, tanh) och deras inverser (arsinh etc.).

Kapaciteten är 26 fasta datareg, $A \rightarrow Z$, + 1680 pgn-steg + separata reg för statistikfunktionerna. Antalet datareg kan ökas i grupper om 10 upp till ett max av 226, varvid varje sådan grupp tar 80 steg ur pgnminnet. Räkner från 10^{-99} till $9,999...10^{99}$, visar upp till 10-siffrig mantissa + 2-siffrig exponent. 10 598-nivåer, 8 FÖR-NEXT-nivåer, 10 vilande operationer.

Den algebraiska organisationen motsvarar den som TI (enligt den ovan citerade artikeln) kallar "EOS", dvs operationföljden mellan funktion och argument är densamma som vid skrivning eller tryckning (på en rad): $\sin 30, \sqrt{(4+5^2)}$, nt osv. Men kan ton utan att räkna skriva $A=0$ utan parenteser kring =0.

En lustig(?) konsekvens av denna organisation är, att funktionerna x^2 och $1/x$ inte har någon plats i BASICs vokabulär. x^2 skrivs sålunda normalt i BASIC som A^2 , motsvarande $Tis A \times A^2$. Eftersom potentering är en notoriiskt långsam process vinner man dock avsevärd tid (ca 1/4 s) på att i stället skriva $A \times A$. Detsamma gäller i princip även högre heltalspotenser, även om tidskillnaden minskar (och pgnlängden ökar) med stigande potens. — Inom parentes må erinras om att ett analogt förhållande gäller TI: $A \times x^2, A \times x^2, A \times x^2, A \times x^2$ är snabbare än $A \times x^n$ med $n = 3, 4, 5, 6$ resp.

Språket är alltså en BASIC-dialekt. Varför "dialekt"? Jo, bl a därför att behandlingen av teckensträngar är torfig och därför att i andra sådan standard-BASIC inte känner ens tillnärmelsevis så många matematiska funktioner som 702an.

För att demonstrera några karakteristiska skillnader mellan Tis assemblerpråk och 702ans BASIC visar jag här ett och samma problem behandlat i de båda språken. Det gäller det ytterst enkla problemet att steg för steg beräkna det nte talet i den Fibonacciföljd som börjar med 0, 1. (Såsom torde vara bekant för de flesta, är varje tal i en sådan följd lika med summan av de båda närmast föregående talen i följden. Det tredje och de därefter närmast följande talen blir då 1, 2, 3, 5, 8, 13,...)

FX-702P

```
10 INP N:A=0:0=1
20 FOR I=3 TO N
30 C=A+B:A=B:B=C
40 NEXT I
50 PRINT C (40 steg)
```

TI-59/50

```
+ xit
0=0:Y=R/5
LBL A-2 = STOY
OP 1 RST
(10 steg)
```

I 702ans pgn visar rad 30 varje steg i beräkningens gång på samma sätt som om man skulle utföra räkningen för hand. Beräkningen utförs så många gånger som förordas av att med steg av längden 1 (underförstått) kommer $fr\ 0\ 3\ 0\ n\ 1$. I rad 10 visar displayen vad 7 som uppmärksam att så i det 8:ande Fibonacci-talet nummer, vilket placeras i reg N.

I jämförelse med 702-pgn är TI-pgn svårare att med ett ögonkast genomskåda. Sekvensen $x \rightarrow x$ kanske vid första påseende ser ut att hänga i luften, men det är den som gör nästan hela jobbet. Det är inte troligt, att en mindre tränad pgnare skulle ha hittat på det pgn utan en viss tankemöda. Pgn har möjligheten av den nästan totala frihet som assemblerspråket medger ifråga om hanteringen av räknares funktioner och operationer. — 702ans språkregler tolererar däremot inte att pgnseraren tar sig några friheter med syntaxen. Varje övertramp framkallar en omedelbar reaktion från räknares sida i form av en felkod, som ger anvisning om felats plats och art.

Jag förmodar, att det är denna högnivåspråkets tvångströja som känns våra pgnexperter för trång. För dem av oss, som inte är så sofistikerade är dessa klara regler däremot ett utomordentligt moraliskt stöd: det lönar sig sällan att öda tid på att förbereda vara smart. BASIC är, särskilt när det har så pass begränsade resurser utom det matematiska området som i 702an, lätt att lära sig och att använda.

Språkets omdömlighet, sådan den visar sig i exemplet, måste väl ändå resultera i långa körtider? Svaret är, att om vi söker det 50-de talet i följden, så tar TI-pgn ca 16,5 s, medan motsvarande siffra för 702-pgn är ca 7 s. Detta motsvarar ett snabbhetsförhållande ca 2,4 till 702ans fördel. Den siffran är emellertid inte typisk, eftersom problemet ligger ovanligt väl till för TI. Normala siffror torde vara ca 3,5 - 5, men kan, i fall som gynnar 702an, uppgå till 10 eller mera. (Om t ex första raden i TI-pgn ändras till det mindre effektiva men kanske närmare till hands liggande + EXEC STO1 och Lbl A anpassas därefter, ökas körtiden med 10 s och snabbhetsförhållandet till 3,7.)

702-pgn kräver 40 steg (ung. = antalet tryckningar av funktions-, tecken- och sifvertangenten), därav sammanlagt 15 för de 5 radnumren. Att det inte blir fler betor på än IMP, FOR, NEXT etc varusåradet med en enda tryckning på samma sätt som STO, EXC osv i TI. I själva verket är det få av 702-språkets instruktioner som kräver mer än ett steg. — Med hänsyn till att 702an i normala fall kräver upp-dubbel så många steg som TI för motsvarande pgn, bör 702ans minneskapacitet vara närmast jämförbar med TI-59ans.

Användningen av explicita radnummer är karakteristisk för högnivåspråken. Ekvationering går alltid från rad till rad efter stigande radnummer. I vilken ordning raderna såls in är likgiltigt, bara de ges rätt radnummer: räknares ombesörjer själv att de kommer i rätt ordning. Visar det inte man måste skjuta in eller ta bort en rad, kan detta göras när som helst utan att pgn i övrigt ändras.

Det totala pgnutrymmet om max 1680 steg kan, om man så önskar, delas upp mellan upp till 10 av varandra oberoende pgn. Eftersom hela pgnminnet ligger kvar vid avstängning av räknares kan man sålunda i regel ha ett eller ett par pgn liggande i räknares medan man arbetar med ett nytt pgn. —

Det finns sålunda en hel del som talar för användningen av BASIC, om man är intresserad inte bara av själva pgningsarbetet utan också av att få fram pgn för lösning av praktiska problem eller av att studera de resultat som de ger. Men får mer tid att konstruera bästa möjliga matematiska lösningsmetoder och kan relativt lätt och snabbt pröva olika alternativ. Vill man sedan översätta pgn till ett avancerat TI-språk, så brukar det gå lättare när man bara har pgningsmetodiken att tänka på.

Med träning i BASIC får man också lätt att övergå till en eller annan mikrodator för behandling av problem av icke-matematisk art, t ex till en TI-99/4. Personligen har jag dessutom en känsla av att en åtminstone elementär kännedom om något högnivåspråk och lite övning i att använda det bör höra till varje amatörgpners allmänbildning. Detta innebär naturligtvis inte att P8 skall övergå till att behandla BASIC. Vår tidskrift ser ut att ha tillräckligt att göra med TI-språket. Men vi medlemmar bör inte känna, att vi måste träffa valet mellan två gode ting när vi kan ha både.

Varför inte Även BASIC?

KLUBB FÖR TI-99-ANVÄNDARE

I broschyren till TI: "hemdator" TI-99/4A har Föreningen Programbiten kommit med som exempel på anknödföreningar för TI: s produkter. Det har gjort att ganska många skrivit eller ringt oss och frågat om en användarförening för TI-99.

Inom Programbiten börjar nu några personer träda fram som skulle kunna vara intresserade av att bilda och driva en avdelning för TI-99. Vi har också blivit erbjudna visst stöd med program från den belgiska föreningen Tisoft. Planer finns på en gemensam dansk-svensk klubbtidning med 6 nummer om året för ca 120 kr/år. TI utredor om de kan vara behjälpliga att förmåda prenumerationer på den amerikanska tidskriften "99ers".

Under hösten kommer alltså ganska säkert något att börja fungera. Om Du är intresserad av klubbarbete med TI-99/4A, hör av Dig, så kan Du komma med från början.

Gardering mot strömvabrott

av Anders Persson

Under Årsmötet i vintras framgick det att fler än Jag själv har lidit av utformningen av strömförsörjningen till räknaren från skrivaren. Om man använder laddaren till räknaren har man ju flera flermars batteriback-up genom att batterierna sitter kvar i räknaren även när man använder laddare. På skrivaren däremot tas batterierna ut före monteringen, varför även någon millicsekunds bortvaro av nätspänningen kan få förödande konsekvenser. Detta är speciellt förvargligt om t.ex. en simulering som tar kanske en vecka plötsligt nollställs efter fem dagar.

Eftersom man kan lägga in batterierna i skrivaren och få dem laddade därifrån ligger det närmast till hands att utnyttja dem för reservkraft. Problemet är alltså en automatisk omkopplare. Då TI inte har någon sådan beslut jag mig för att göra en sådan själv. Innan Jag nu berättar hur man kan gå tillväga vill jag tala om att TI utför eventuella reparationer även på en skrivare som är modifierad på det här sättet, men den vanligaste fasta prislåsten tillämpas inte utan de tar betalt efter arbetstiden. Dessutom garanteras inte att modifieringen kvarstår efter reparationen. Det är dock nån så enkelt att ta ur enheten igen om en reparation av någon anledning skulle bli aktuell. Modifieringen gör inga nya hål eller något annat i den vägen.

Funktionen har valts så att om nätspänningen försvinner kopplas batterierna in och håller minnet i räknaren intakt. Så snart nätet kommer tillbaka återgår skrivaren till det normala igen och laddar batterierna. Automatisk uppladdning mellan strömvabrotten alltså. Batteripacken måste naturligtvis finnas i facket på skrivaren.

Skrivarmöjligheterna försvinner givetvis med detta arrangemang när strömmen går. Dels ska skrivaren ha andra spänningar än de som räknaren använder och dels är strömförbrukningen i mesta laget för penlightbatterier i batteripacken.

Att strömvabrott är sällan förekommande är knappast något skäl till att inte montera in den här kretsen. Själv har jag sparat åtskilliga möda genom att den även räddar sådana situationer som då någon av misstag drar ut sladden eller då jag av någon anledning har velat flytta skrivaren till ett annat uttag under pågående beräkningar. Med en TI-59 kan man visserligen lösa detta genom inspelning av magnetkort, men med en TI-58 (utan C) är det omöjligt att flytta utrustningen utan att börja om igen.

Extrakortet inkopplas med hjälp av de befintliga kablarna mellan skrivarens två halvor. Från kortet får sedan fyra nya ledningar dras till den övre skrivarhalvan som ersättning för de som löds bort. Dessutom behövs två trådar till transformatorn för övervakning av nätspänningen.

Under själva omkopplingen hålls drivspänningen till räknaren uppe av kondensatorn C2. Tidskonstanten är så vald att den klarar av att hålla räknaren igång även under den strömkrävande programkörningen eller med många siffror tända. In- eller avspeling av magnetkort lyckas dock i regel inte om en omkoppling sker mitt under genomdragningen. Dessutom går det att provocera fram konstigheter om man upprepade gånger drar ut och sätter i kontakten så snabbt att C2 inte hinna uppladdas tillräckligt mellan omkopplingarna. Värmeutvecklingen i skrivaren ökar något, beroende på att reläets spole alstrar en del värme. Tänk på att reläet ska vara avsett för kontinuerlig inkoppling.

Kretslösningen framgår av kopplingsschemat. Samtliga komponenter är enkla att få tag i, t.ex. på ELFA. Min prototyp är byggd på Veroboard, men det går naturligtvis lika bra att rita ett eget kort. Var dock noga med layouten! Även om det ser luftigt ut i skrivaren är det millimeterkrav på precisionen vid monteringen, annars får den inte plats. Jag valde att placera kortet ovanför det ordinarie framför kylplåten till stabilisatorn för räknarens drivspänning. Det går bra, men en viss pulsräns kan rekommenderas.

Under provkörningen visade det sig att den stabilisatorn kan påverkas av kondensatorn C2 på så sätt att den lämnar för hög spänning. Efter byte till en annan liknande typ försvann det fenomenet. Utrustningen har nu fungerat utan problem i mer än sex månader.

Jag har själv en PC-100C, men rent principiellt bör det inte vara något hinder för att koppla in den här kretsen i någon av de tidigare modellerna. Det skulle i så fall vara om det finns mer utrustning från början i PC-100A, så att utrymmet är mindre. Det har jag inte undersökt.

Har du några frågor eller problem kan du ringa mig. Mitt nr är 046-11 69 19.



UTLÄNDSKA KLUBBAR

av Björn Gustavsson

TI PPC. Redaktör är Maurice E.T. Swinnen som varje år ger ut tio nummer av TI PPC Notes. Program och artiklar håller mycket hög kvalitet. TI PPC är också den mest internationella klubben med aktiva medlemmar från bl a Belgien, Tyskland, Italien, Jugoslavien och Sverige. Ett brev från Maurice är publicerat i PB 80-2 s 4. Observera dock att adressen är ändrad:

Maurice E.T. Swinnen
9213 Lanham Severn Road
Lanham MD 20706
USA

Helårsmedlemskap: \$30, tidigare årgångar \$24

SR-52 Users Club. Föregångare till TI PPC. Redaktör var Richard Vanderburgh som mellan juni 1976 och maj 1979 gav ut sammanlagt 34 nummer av 52-Notes. Tvärtom vad namnet på tidskriften kan antydja så handlade bara de 12 första numren om SR-52, SR-56 och PC-100. Övriga 22 nummer handlade mest om TI-58/59 och PC-100C. Richard Vanderburgh säljer fortfarande gamla nummer av 52-Notes och de är väl värda att köpa. För det första innehåller de många bra program och rutiner. ("Game of Life" i PB 80-4 s 14-17 är hämtat ur 52-Notes och idén till kalendertävlingen kommer också därifrån, för att bara nämna två exempel.) För det andra ger 52-Notes en intressant historisk återblick över bl a TI-59:s "ankomst", viktiga upptäckter som t ex HIR etc. För det tredje hänvisas det ofta i TI PPC Notes till 52-Notes. TI PPC Notes' årgångar är numrerade från 5 och uppåt, eftersom 52-Notes' sista årgång var den fjärde.

Även de första 12 numren som handlade utslutande om SR-52 och SR-56 är intressanta för en TI-58/59-ägare. Idéerna till Dynamisk Nim och Stjärnskytte i PB 81-3 har hämtats från SR-52-program i 52-Notes.

Richard Vanderburgh
9459 Taylorsville Road
Huber Heights OH 45424
USA

Gamla nummer kostar \$2.00 styck.

TI Software Exchange Club (TISOFT). En belgisk klubb, men medlemsbladet TI Software Exchange

Newsletter är skrivet helt på engelska. Redaktör är Thomas Coppens. Medlemsbladet utkommer med 4 stycken 16-sidiga nummer per år. Klubben har också Europas största programbibliotek.

TISOFT
Tennisstraat 16
B-2610 WILRIJK
Belgien

Helårsmedlemskap kostar 550 BF om det betalas till belgiskt postgiro 000 -1149689 - 45 eller med internationell postanvisning till TISOFT, Thomas Coppens, PB 63, B-2080 KAPELEN Belgien.

Programklubben. Redaktör för medlemsbladet Pgm som denna danska programklubb ger ut är Hans Peter Nielsen. Pgm utkommer med 6 stycken 48-sidiga nummer per år (sidorna är dock inte i A4-format). En av klubbens medlemmar håller på att framställa ett system med vilket var och en kan programmera en egen modul till TI-58/59.

Programklubben
Vestervold 16
DK-8060 Varde
Danmark

Helårsmedlemskap kostar 110 danska kronor.

PPX. Utan tvekan den största TI-användarklubben. Medlemsbladet PPX Exchange utges 6 gånger per år med 12 sidor per nummer. Som bekant är det TI själva som ligger bakom PPX, vilket gör att man håller en korrektare stil än TI PPC Notes och därmed blir tråkigare. Nyheter publiceras inte då de är nya; så presenterades Fast Mode och Graphics Mode ungefär ett år senare än i TI PPC Notes och då med en varningstext som slutade:

"Användning av 'X' kan ha oförutsedda följder och kan inte garanteras av TI. Inte heller ansvarar TI för skador på programvara som orsakas härav. Nedanstående artikel återges endast som något som kan vara av intresse för TI-59-ägare."

PPX
P.O. Box 53
Lubbock TX 79408
USA

Helårsmedlemskap kostar \$32.00.

OBST! Skicka inte pengar genom bank! Använd internationell postanvisning (eller postgiro om du har det) eller sänd bankcheck i landets mynt. ■

Å, Ä och Ö på PC 100

- på riktigt!

Ursprungligt program: Lars Nilsson, Landskrona. Bearbetning och text: Björn Gustavsson.

I Programbiten 82-1 på s 31 skrev jag att jag hade hittat ett sätt att skriva Å, Ä och Ö på PC-100. Eftersom det var första-aprilnumret blev emellertid resultatet av en körning bara att programmet skrev ut "APRIL, APRIL" och utplånade sig själv. Hexkoden som användes är känd under namnet "Super clear".

Lars Nilsson, Landskrona har emellertid gjort ett riktigt program som verkligen skriver ut Å, Ä och Ö. Det trycker först eventuellt ut en rad med "prickar" och sedan bokstäverna på normalt vis; allt som användaren behöver göra är att ge de nya "skrivkoderna" för Å, Ä och Ö samt ett minustecken som skrivregistret skulle innehålla något av dessa tecken.

Eftersom Lars' program närmast är ett demonstrationsprogram med manuell inmatning av skrivkoderna från tangentbordet, har jag gjort om programmet så att det hämtar skrivkoderna från ett antal register. Dessutom har jag använt en senare metod för initiering av graphics mode, som Patrick Acosta har upptäckt och redogjort för i TI PPC Notes V7 N7/8 P18-19. Man behöver då inte utföra så mycket manuellt "knapptryckande".

Metoden är som följer: prickarna på Å och Ö åstadkommes med tecknet "*" (kod 51) och Å med "" (apostrof; kod 65). För att detta ska fungera måste hexkoden för graphics mode finnas på steg 008, eftersom då endast två punktrader skrivs ut.

Programmet har utformats för att skriva ut text vars skrivkoderna finns lagrade i valfria register (R00-R07 och R90-R99 får dock ej användas). Å, Ä och Ö har fått skrivkoderna 80, 81 respektive 82. För att programmet skall behandla innehåll i ett skrivregister korrekt där en av koderna 80, 81 eller 82 ingår, måste dess tecken vara negativt. Detta för att räknenaren inte ska slösa tid på att specialbehandla vanliga skrivkod, som inte innehåller Å, Ä eller Ö.

NU GÅR DET ATT
SKRIVA Å, Ä OCH Ö
PÅ PC-100!

Bruksanvisning:

1. Mata eller läs in program 1. Tryck E SST SST 7 EE LRN INS LRN RST.
2. Mata in lägsta och högsta registernumren där skrivkoderna är lagrade. Denna inmatning sker på formen xx.yy. Tryck därefter A.

Exempel: om man efter att ha utfört steg 1 i bruksanvisningen trycker 2182351731 +/- STO 08 2431221731 STO 09 3353222 STO 10 351330200 STO 11 1424371731 STO 12 8.12 A skrivs
FÖRENINGEN PROGRAMBITEN
ut.

Man kan få tysk teckenuppställning om man gör följande ändring: GT0 124 LRN 41 SST 51 LRN. Koden 80 ger då ü.

Om programmet ska användas som subrutin bör man ändra GT0-adressen vid steg 004-005, så att hopp sker till huvudprogrammet vid nedtryckning av A. Från huvudprogrammet anropas sedan subrutinen med SBR 033.

```
000 32 ETH 050 00 0 100 01 01 150 01 01
001 76 LR 051 42 STD 101 25 5 151 75 52
002 11 R 052 00 00 102 08 8 152 25 CLR
003 41 GTD 053 42 STD 103 25 5 153 75 52
004 00 00 054 07 07 104 25 INV 154 00 00
005 22 33 055 49 DP 105 00 0 155 00 00
006 69 DP 056 27 27 106 01 01 156 49 DP
007 05 05 057 00 0 107 00 0 157 00 00
008 00 0 058 72 ST+ 108 03 3 158 25 CLR
009 12 R 059 07 7 109 00 0 159 00 00
010 00 0 060 82 HIR 110 32 KIT 160 00 00
011 00 0 061 04 04 111 05 5 161 00 00
012 00 0 062 73 RC+ 112 01 1 162 73 RC+
013 76 LR 063 05 05 113 41 GTD 163 07 7
014 15 E 064 49 DP 114 01 01 164 84 RC+
015 25 CLR 065 25 25 115 31 3 165 07 7
016 01 1 066 29 CF 116 01 1 166 97 BZ2
017 05 + 067 77 CE 117 01 1 167 07 7
018 01 1 068 01 01 118 32 KIT 168 01 01
019 09 9 069 34 34 119 05 5 169 43 3
020 52 EE 070 94 +/- 120 05 5 170 49 DP
021 01 1 071 25 - 121 41 GTD 171 00 00
022 01 1 072 05 - 122 01 1 172 49 DP
023 94 +/- 073 42 STD 123 31 3 173 00 00
024 22 INV 074 00 00 124 01 1 174 04 4
025 52 EE 075 01 1 125 03 3 175 00 00
026 35 + 076 52 EE 126 32 KIT 176 06 6
027 40 BZ2 077 01 1 127 05 5 177 00 00
028 31 E 078 00 0 128 01 1 178 24 56
029 69 DP 079 45 45 129 03 3 179 25 INV
030 40 INV 080 01 1 130 82 HIR 180 47 67
031 61 51 081 00 0 131 04 34 181 01 01
032 75 DP 082 00 0 132 32 KIT 182 53 53
033 49 DP 083 82 HIR 133 74 RC+ 183 00 00
034 00 00 084 44 44 134 07 07 184 92 ETH
035 42 STD 085 44 44 135 37 BZ2 185 00 00
036 05 05 086 07 07 136 00 00
037 95 + 087 45 45 137 00 00
038 65 + 088 59 INV 138 81 81
039 01 1 089 82 HIR 139 25 CLR 189 00 00
040 00 0 090 51 51 140 82 HIR
041 00 0 091 32 KIT 141 14 14
042 75 DP 092 08 8 142 84 RC+
043 43 RCL 093 00 0 143 07 07
044 05 05 094 67 EQ 144 04 4
045 35 + 095 01 01 145 22 KIT
046 01 1 096 17 17 146 43 RCL
047 95 + 097 08 8 147 07 07
048 42 STD 098 01 1 148 22 INV
049 06 06 099 47 EQ 149 07 EQ
```

Diofantiska ekvationen $ax + by = c$ -2

av Michael Hedberg

Reds. anm.: Gösta Blume lämnade en köll en förbittring till ett eget program för den diofantiska ekvationen $ax + by = c$, SE 81-4 s 16. Med posten morgon därefter kom ytterligare en förbittrad version från Michael Hedberg, Lund. Efter att ha tittat på Michaels program tyckte Gösta själv att det var det bästa programmet, och därför tar vi in det.

Gösta påpekar också att det tidigare programmet var ett primitivt program. Det bygger på en oeländig algoritmen vilket leder till att det ibland övergår från att lösa en lösbar ekvation till en olöslig. Formeln som angavs för ekvationens allmänna lösning är korrekt endast om a och b saknar gemensamma faktorer.

I denna är a, b och c heltalskonstanter och man söker heltalslösningar (X, Y). Problemet är initierat förknippat med Euklides algoritmen och stötar gemensamma delaren till a och b , $d = GCD(a, b)$ (Greatest Common Divisor). Med beteckningarna $s = a/d$ och $t = b/d$ gäller då:

1. Ekvationen är lösbar om och endast om d går jämnt upp i c .
2. Om (X_0, Y_0) är en lösning, så ges den allmänna lösningen (X, Y) av
 $X = X_0 - ks'$
 $Y = Y_0 + ks'$
där k är ett godtyckligt heltal.

Detta visar vilken roll telet d spelar, dels för lösningens existens (1.) och dels dess allmänhet (2.).

Euklides algoritmen är förstås viktig som ett medel för att beräkna d , men, som vi ska se, även för att finna en lösning till den närbesläktade ekvationen $ax + by = d$. En sådan lösning multipliceras med c/d (som ju är ett heltal enligt (1.)), om det finns någon lösning) för att ge en lösning till den ursprungliga ekvationen.

—000—
Exempel: $a = 116, b = 34$, Euklides algoritmen:
 $116 = 3 \cdot 34 + 14$
 $34 = 2 \cdot 14 + 6$
 $14 = 2 \cdot 6 + 2$
 $6 = 3 \cdot 2 + 0$ verur $d = 2$.
Skriv nu om identiteterna:
 $a = a - 3b$
 $b = b - 2(a - 3b) = 7b - 2a$
 $d = 14 - 2 \cdot 6 = (a - 3b) - 2(7b - 2a) = 5a - 17b$
Här är alltså (X, Y) = (5 -17) en lösning till ekv $ax + by = d$.

För en diskussion av det allmänna fallet, införas följande beteckningar för de storheter, som uppträder i Euklides algoritmen:

$$\begin{aligned} r_0 &= a \\ r_1 &= b \\ r_2 &= q_1 r_1 + r_2 \\ &\vdots \\ r_{k-1} &= q_k r_k + r_{k+1} \\ &\vdots \\ r_{n-1} &= q_n r_n + r_{n+1} \end{aligned}$$

Här är alltså q_k kvoten och r_{k+1} resten vid division av r_{k-1} med r_k . I sista steget ($k=n$) är $r_{n+1} = 0$ och $r_n = d$.

Ekvationerna ovan kan sammanfattas så här: Skvansen r_i är en lösning till differensekvationen
 $r_{k+1} + q_k r_k - r_{k-1} = 0, k=1, n$
och uppfyller begynnelsevillkoren
 $r_0 = a$
 $r_1 = b$.

Eftersom differensekvationen (efter en trivial omskrivning) anger hur ett element i skvansen kan beräknas ur de två närmast föregående, är lösningen en entydigt bestämd av sina begynnelsevärden. Låt nu X_1 och Y_1 vara lösningar med startvärdena
 $X_0 = 1, Y_0 = 0$
 $X_1 = 0, Y_1 = 1$.

Då är linjekompositionen $ax_1 + by_1$ också en lösning och med samma startvärden som r_1 . Det måste alltså gälla $r_1 = ax_1 + by_1, i=0, n-1$. Speciellt är då $d = r_n = ax_n + by_n$. I fullständighets namn bär det kanske påseks att X_n och Y_n verkligen är heltal, vilket de är eftersom differensekvationen är sådan att en lösning, som startar med heltal, också fortsätter med heltal.

Algoritmen för samtidig beräkning av $d = GCD(a, b)$ och en lösning till den diofantiska ekvationen
 $ax + by = d$

Indata: d, X, Y (heltal)
Utdata: d, X, Y
Variabler: $X0, X1, X2, Y0, Y1, Y2, R0, R1, R2, 0$

1. Sätt $X0=1, X1=0, Y0=0, Y1=1, R0=a, R1=b$.
2. Och vidare till steg 4.
3. Låt Q vara kvoten och $R2$ resten vid division av $R0$ med $R1$.
Sätt $X2 = X0 - QX1, Y2 = Y0 - QY1$.
Sätt $X0=X1, X1=X2, Y0=Y1, Y1=Y2, R0=R1$ och $R1=R2$.
4. Om $R1=0$, så gå till steg 3.
5. Sätt $d=R0, X=X0$ och $Y=Y0$.

31

KONTROLLSUMMA

av BJÖRN GUSTAVSSON

Mitt program för kontrollsumma som beskrivs här kommer troligen att bli TI PPC Notes' officiella program för kontrollsumma. Från och med nästa nummer kommer kontrollsumma att anges för alla program även här i Programbiten. Den som har negativa synpunkter uppmanas att meddela oss före pressläggningen av nästa nummer, dvs senast i mitten av november.

Tidigare versioner av kontrollsumma-program, t.ex. det i PB 81-4 sida 35, ignorerade tre siffror i varje register: exponenten och tecknen framför mantissa och exponent. Det här programmet betraktar innehållet i varje register som ett 16-siffrigt heltal; det

talet delas i två åtta-siffriga heltal som adderas. Om teckensiffran är en åtta eller nia är det som vanligt omöjligt att erhålla några upplysningar om innehållet, -9,999... 99 blinkar vid återkallning av det registrerat. Ingen har hittat någon lösning på detta problem, detta program ignorerar sådana register förutom att deras antal räknas. Antalet sådana register placeras efter den åtta-siffriga kontrollsumman: xxxxxxxxxx; den fullständiga kontrollsumman är alltså tio-siffrig. Programmet kan om så önskas, göra en lista över "osäkra" steg (dvs. steg som ingår i ett register där teckensiffran är en åtta eller nia).

```
000 00 0 070 55 + 140 01 1
000 00 0 072 00 0 142 00 0
000 76 LBL 073 79 00 142 00 0
004 11 R 074 02 - 144 85 RCL
005 35 PCH 075 25 + 146 85 RCL
006 02 02 076 44 SUM 146 85 RCL
007 01 01 077 00 0 148 85 RCL
008 02 02 078 23 21 148 85 RCL
009 26 29 079 41 01 01 148 85 RCL
010 09 9 080 11 11 150 53 (
011 00 0 081 42 EE 152 55 (
012 22 INV 082 55 EE 152 55 (
013 00 0 083 55 EE 152 55 (
014 22 INV 084 00 0 154 00 0
015 27 ENG 085 00 0 156 00 0
016 47 CHG 086 50 1:1 156 00 0
017 27 ENG 087 54 00 158 00 0
018 59 FTL 088 42 STD 158 59 INT
019 09 09 089 35 35 159 59 INT
020 59 PRT 090 65 + 160 01 1
021 01 0 091 01 0 162 00 0
022 25 + 092 52 EE 162 22 INV
023 25 + 093 52 EE 164 95 RCL
024 67 EE 094 75 - 164 95 RCL
025 02 02 095 00 0 166 95 RCL
026 04 04 096 44 SUM 166 95 RCL
027 77 GE 097 31 168 87 IFF
028 00 0 098 95 95 168 87 IFF
029 32 32 099 00 0 170 00 0
030 44 + 100 01 1 170 00 0
031 22 INV 101 01 1 172 99 PRT
032 65 FTL 102 08 8 172 99 PRT
033 00 0 103 00 0 174 00 0
034 02 STD 104 44 SUM 174 00 0
035 34 34 105 00 0 176 00 0
036 42 2 106 25 CLR 176 00 0
037 04 0 107 00 0 178 00 0
038 00 0 108 35 35 178 44 SUM
039 49 PCH 109 00 0 180 87 IFF
040 34 34 110 58 FTL 180 87 IFF
041 22 INV 111 00 0 182 00 0
042 44 SUM 112 52 EE 182 01 01
043 34 34 113 00 0 184 00 0
044 04 + 114 77 GE 184 84 RCL
045 04 04 115 00 0 186 85 RCL
046 22 INV 116 18 19 186 85 RCL
047 00 0 117 00 0 188 00 0
048 02 2 118 93 93 188 40 IND
049 09 9 119 44 SUM 188 40 IND
050 93 + 120 31 31 190 07 7
051 01 1 122 00 0 192 55 +
052 42 STD 124 00 0 194 22 INV
053 25 + 126 00 0 196 95 RCL
054 25 + 128 00 0 198 59 INT
055 42 STD 130 00 0 200 65 PRT
056 31 31 132 31 31 196 95 RCL
057 42 STD 134 00 0 202 00 0
058 32 32 136 44 SUM 202 00 0
059 38 38 138 23 21 204 00 0
060 32 32 140 25 CLR 206 00 0
061 38 38 142 00 0 208 00 0
062 30 30 144 00 0 210 00 0
063 01 01 146 00 0 212 00 0
064 01 01 148 00 0 214 00 0
065 01 01 150 00 0 216 00 0
066 01 01 152 00 0 218 00 0
067 01 01 154 00 0 220 00 0
068 01 01 156 00 0 222 00 0
069 01 01 158 00 0 224 00 0
070 01 01 160 00 0 226 00 0
071 01 01 162 00 0 228 00 0
072 01 01 164 00 0 230 00 0
073 01 01 166 00 0 232 00 0
074 01 01 168 00 0 234 00 0
075 01 01 170 00 0 236 00 0
076 01 01 172 00 0 238 00 0
077 01 01 174 00 0 240 00 0
078 01 01 176 00 0 242 00 0
079 01 01 178 00 0 244 00 0
080 01 01 180 00 0 246 00 0
081 01 01 182 00 0 248 00 0
082 01 01 184 00 0 250 00 0
083 01 01 186 00 0 252 00 0
084 01 01 188 00 0 254 00 0
085 01 01 190 00 0 256 00 0
086 01 01 192 00 0 258 00 0
087 01 01 194 00 0 260 00 0
088 01 01 196 00 0 262 00 0
089 01 01 198 00 0 264 00 0
090 01 01 200 00 0 266 00 0
091 01 01 202 00 0 268 00 0
092 01 01 204 00 0 270 00 0
093 01 01 206 00 0 272 00 0
094 01 01 208 00 0 274 00 0
095 01 01 210 00 0 276 00 0
096 01 01 212 00 0 278 00 0
097 01 01 214 00 0 280 00 0
098 01 01 216 00 0 282 00 0
099 01 01 218 00 0 284 00 0
100 01 01 220 00 0 286 00 0
101 01 01 222 00 0 288 00 0
102 01 01 224 00 0 290 00 0
103 01 01 226 00 0 292 00 0
104 01 01 228 00 0 294 00 0
105 01 01 230 00 0 296 00 0
106 01 01 232 00 0 298 00 0
107 01 01 234 00 0 300 00 0
108 01 01 236 00 0 302 00 0
109 01 01 238 00 0 304 00 0
110 01 01 240 00 0 306 00 0
111 01 01 242 00 0 308 00 0
112 01 01 244 00 0 310 00 0
113 01 01 246 00 0 312 00 0
114 01 01 248 00 0 314 00 0
115 01 01 250 00 0 316 00 0
116 01 01 252 00 0 318 00 0
117 01 01 254 00 0 320 00 0
118 01 01 256 00 0 322 00 0
119 01 01 258 00 0 324 00 0
120 01 01 260 00 0 326 00 0
121 01 01 262 00 0 328 00 0
122 01 01 264 00 0 330 00 0
123 01 01 266 00 0 332 00 0
124 01 01 268 00 0 334 00 0
125 01 01 270 00 0 336 00 0
126 01 01 272 00 0 338 00 0
127 01 01 274 00 0 340 00 0
128 01 01 276 00 0 342 00 0
129 01 01 278 00 0 344 00 0
130 01 01 280 00 0 346 00 0
131 01 01 282 00 0 348 00 0
132 01 01 284 00 0 350 00 0
133 01 01 286 00 0 352 00 0
134 01 01 288 00 0 354 00 0
135 01 01 290 00 0 356 00 0
136 01 01 292 00 0 358 00 0
137 01 01 294 00 0 360 00 0
138 01 01 296 00 0 362 00 0
139 01 01 298 00 0 364 00 0
140 01 01 300 00 0 366 00 0
141 01 01 302 00 0 368 00 0
142 01 01 304 00 0 370 00 0
143 01 01 306 00 0 372 00 0
144 01 01 308 00 0 374 00 0
145 01 01 310 00 0 376 00 0
146 01 01 312 00 0 378 00 0
147 01 01 314 00 0 380 00 0
148 01 01 316 00 0 382 00 0
149 01 01 318 00 0 384 00 0
150 01 01 320 00 0 386 00 0
151 01 01 322 00 0 388 00 0
152 01 01 324 00 0 390 00 0
153 01 01 326 00 0 392 00 0
154 01 01 328 00 0 394 00 0
155 01 01 330 00 0 396 00 0
156 01 01 332 00 0 398 00 0
157 01 01 334 00 0 400 00 0
158 01 01 336 00 0 402 00 0
159 01 01 338 00 0 404 00 0
160 01 01 340 00 0 406 00 0
161 01 01 342 00 0 408 00 0
162 01 01 344 00 0 410 00 0
163 01 01 346 00 0 412 00 0
164 01 01 348 00 0 414 00 0
165 01 01 350 00 0 416 00 0
166 01 01 352 00 0 418 00 0
167 01 01 354 00 0 420 00 0
168 01 01 356 00 0 422 00 0
169 01 01 358 00 0 424 00 0
170 01 01 360 00 0 426 00 0
171 01 01 362 00 0 428 00 0
172 01 01 364 00 0 430 00 0
173 01 01 366 00 0 432 00 0
174 01 01 368 00 0 434 00 0
175 01 01 370 00 0 436 00 0
176 01 01 372 00 0 438 00 0
177 01 01 374 00 0 440 00 0
178 01 01 376 00 0 442 00 0
179 01 01 378 00 0 444 00 0
180 01 01 380 00 0 446 00 0
181 01 01 382 00 0 448 00 0
182 01 01 384 00 0 450 00 0
183 01 01 386 00 0 452 00 0
184 01 01 388 00 0 454 00 0
185 01 01 390 00 0 456 00 0
186 01 01 392 00 0 458 00 0
187 01 01 394 00 0 460 00 0
188 01 01 396 00 0 462 00 0
189 01 01 398 00 0 464 00 0
190 01 01 400 00 0 466 00 0
191 01 01 402 00 0 468 00 0
192 01 01 404 00 0 470 00 0
193 01 01 406 00 0 472 00 0
194 01 01 408 00 0 474 00 0
195 01 01 410 00 0 476 00 0
196 01 01 412 00 0 478 00 0
197 01 01 414 00 0 480 00 0
198 01 01 416 00 0 482 00 0
199 01 01 418 00 0 484 00 0
200 01 01 420 00 0 486 00 0
201 01 01 422 00 0 488 00 0
202 01 01 424 00 0 490 00 0
203 01 01 426 00 0 492 00 0
204 01 01 428 00 0 494 00 0
205 01 01 430 00 0 496 00 0
206 01 01 432 00 0 498 00 0
207 01 01 434 00 0 500 00 0
208 01 01 436 00 0 502 00 0
209 01 01 438 00 0 504 00 0
210 01 01 440 00 0 506 00 0
211 01 01 442 00 0 508 00 0
212 01 01 444 00 0 510 00 0
213 01 01 446 00 0 512 00 0
214 01 01 448 00 0 514 00 0
215 01 01 450 00 0 516 00 0
216 01 01 452 00 0 518 00 0
217 01 01 454 00 0 520 00 0
218 01 01 456 00 0 522 00 0
219 01 01 458 00 0 524 00 0
220 01 01 460 00 0 526 00 0
221 01 01 462 00 0 528 00 0
222 01 01 464 00 0 530 00 0
223 01 01 466 00 0 532 00 0
224 01 01 468 00 0 534 00 0
225 01 01 470 00 0 536 00 0
226 01 01 472 00 0 538 00 0
227 01 01 474 00 0 540 00 0
228 01 01 476 00 0 542 00 0
229 01 01 478 00 0 544 00 0
230 01 01 480 00 0 546 00 0
231 01 01 482 00 0 548 00 0
232 01 01 484 00 0 550 00 0
233 01 01 486 00 0 552 00 0
234 01 01 488 00 0 554 00 0
235 01 01 490 00 0 556 00 0
236 01 01 492 00 0 558 00 0
237 01 01 494 00 0 560 00 0
238 01 01 496 00 0 562 00 0
239 01 01 498 00 0 564 00 0
240 01 01 500 00 0 566 00 0
241 01 01 502 00 0 568 00 0
242 01 01 504 00 0 570 00 0
243 01 01 506 00 0 572 00 0
244 01 01 508 00 0 574 00 0
245 01 01 510 00 0 576 00 0
246 01 01 512 00 0 578 00 0
247 01 01 514 00 0 580 00 0
248 01 01 516 00 0 582 00 0
249 01 01 518 00 0 584 00 0
250 01 01 520 00 0 586 00 0
251 01 01 522 00 0 588 00 0
252 01 01 524 00 0 590 00 0
253 01 01 526 00 0 592 00 0
254 01 01 528 00 0 594 00 0
255 01 01 530 00 0 596 00 0
256 01 01 532 00 0 598 00 0
257 01 01 534 00 0 600 00 0
258 01 01 536 00 0 602 00 0
259 01 01 538 00 0 604 00 0
260 01 01 540 00 0 606 00 0
261 01 01 542 00 0 608 00 0
262 01 01 544 00 0 610 00 0
263 01 01 546 00 0 612 00 0
264 01 01 548 00 0 614 00 0
265 01 01 550 00 0 616 00 0
266 01 01 552 00 0 618 00 0
267 01 01 554 00 0 620 00 0
268 01 01 556 00 0 622 00 0
269 01 01 558 00 0 624 00 0
270 01 01 560 00 0 626 00 0
271 01 01 562 00 0 628 00 0
272 01 01 564 00 0 630 00 0
273 01 01 566 00 0 632 00 0
274 01 01 568 00 0 634 00 0
275 01 01 570 00 0 636 00 0
276 01 01 572 00 0 638 00 0
277 01 01 574 00 0 640 00 0
278 01 01 576 00 0 642 00 0
279 01 01 578 00 0 644 00 0
280 01 01 580 00 0 646 00 0
281 01 01 582 00 0 648 00 0
282 01 01 584 00 0 650 00 0
283 01 01 586 00 0 652 00 0
284 01 01 588 00 0 654 00 0
285 01 01 590 00 0 656 00 0
286 01 01 592 00 0 658 00 0
287 01 01 594 00 0 660 00 0
288 01 01 596 00 0 662 00 0
289 01 01 598 00 0 664 00 0
290 01 01 600 00 0 666 00 0
291 01 01 602 00 0 668 00 0
292 01 01 604 00 0 670 00 0
293 01 01 606 00 0 672 00 0
294 01 01 608 00 0 674 00 0
295 01 01 610 00 0 676 00 0
296 01 01 612 00 0 678 00 0
297 01 01 614 00 0 680 00 0
298 01 01 616 00 0 682 00 0
299 01 01 618 00 0 684 00 0
300 01 01 620 00 0 686 00 0
301 01 01 622 00 0 688 00 0
302 01 01 624 00 0 690 00 0
303 01 01 626 00 0 692 00 0
304 01 01 628 00 0 694 00 0
305 01 01 630 00 0 696 00 0
306 01 01 632 00 0 698 00 0
307 01 01 634 00 0 700 00 0
308 01 01 636 00 0 702 00 0
309 01 01 638 00 0 704 00 0
310 01 01 640 00 0 706 00 0
311 01 01 642 00 0 708 00 0
312 01 01 644 00 0 710 00 0
313 01 01 646 00 0 712 00 0
314 01 01 648 00 0 714 00 0
315 01 01 650 00 0 716 00 0
316 01 01 652 00 0 718 00 0
317 01 01 654 00 0 720 00 0
318 01 01 656 00 0 722 00 0
319 01 01 658 00 0 724 00 0
320 01 01 660 00 0 726 00 0
321 01 01 662 00 0 728 00 0
322 01 01 664 00 0 730 00 0
323 01 01 666 00 0 732 00 0
324 01 01 668 00 0 734 00 0
325 01 01 670 00 0 736 00 0
326 01 01 672 00 0 738 00 0
327 01 01 674 00 0 740 00 0
328 01 01 676 00 0 742 00 0
329 01 01 678 00 0 744 00 0
330 01 01 680 00 0 746 00 0
331 01 01 682 00 0 748 00 0
332 01 01 684 00 0 750 00 0
333 01 01 686 00 0 752 00 0
334 01 01 688 00 0 754 00 0
335 01 01 690 00 0 756 00 0
336 01 01 692 00 0 758 00 0
337 01 01 694 00 0 760 00 0
338 01 01 696 00 0 762 00 0
339 01 01 698 00 0 764 00 0
340 01 01 700 00 0 766 00 0
341 01 01 702 00 0 768 00 0
342 01 01 704 00 0 770 00 0
343 01 01 706 00 0 772 00 0
344 01 01 708 00 0 774 00 0
345 01 01 710 00 0 776 00 0
346 01 01 712 00 0 778 00 0
347 01 01 714 00 0 780 00 0
348 01 01 716 00 0 782 00 0
349 01 01 718 00 0 784 00 0
350 01 01 720 00 0 786 00 0
351 01 01 722 00 0 788 00 0
352 01 01 724 00 0 790 00 0
353 01 01 726 00 0 792 00 0
354 01 01 728 00 0 794 00 0
355 01 01 730 00 0 796 00 0
356 01 01 732 00 0 798 00 0
357 01 01 734 00 0 800 00 0
358 01 01 736 00 0 802 00 0
359 01 01 738 00 0 804 00 0
360 01 01 740 00 0 806 00 0
361 01 01 742 00 0 808 00 0
362 01 01 744 00 0 810 00 0
363 01 01 746 00 0 812 00 0
364 01 01 748 00 0 814 00 0
365 01 01 750 00 0 816 00 0
366 01 01 752 00 0 818 00 0
367 01 01 754 00 0 820 00 0
368 01 01 756 00 0 822 00 0
369 01 01 758 00 0 824 00 0
370 01 01 760 00 0 826 00 0
371 01 01 762 00 0 828 00 0
372 01 01 764 00 0 830 00 0
373 01 01 766 00 0 832 00 0
374 01 01 768 00 0 834 00 0
375 01 01 770 00 0 836 00 0
376 01 01 772 00 0 838 00 0
377 01 01 774 00 0 840 00 0
378 01 01 776 00 0 842 00 0
379 01 01 778 00 0 844 00 0
380 01 01 780 00 0 846 00 0
381 01 01 782 00 0 848 00 0
382 01 01 784 00 0 850 00 0
383 01 01 786 00 0 852 00 0
384 01 01 788 00 0 854 00 0
385 01 01 790 00 0 856 00 0
386 01 01 792 00 0 858 00 0
387 01 01 794 00 0 860 00 0
388 01 01 796 00 0 862 00 0
389 01 01 798 00 0 864 00 0
390 01 01 800 00 0 866 00 0
391 01 01 802 00 0 868 00 0
392 01 01 804 00 0 870 00 0
393 01 01 806 00 0 872 00 0
394 01 01 808 00 0 874 00 0
395 01 01 810 00 0 876 00 0
396 01 01 812 00 0 878 00 0
397 01 01 814 00 0 880 00 0
398 01 01 816 00 0 882 00 0
399 01 01 818 00 0 884 00 0
400 01 01 820 00 0 886 00 0
401 01 01 822 00 0 888 00 0
402 01 01 824 00 0 890 00 0
403 01 01 826 00 0 892 00 0
404 01 01 828 00 0 894 00 0
405 01 01 830 00 0 896 00 0
406 01 01 832 00 0 898 00 0
407 01 01 834 00 0 900 00 0
408 01 01 836 00 0 902 00 0
409 01 01 838 00 0 904 00 0
410 01 01 840 00 0 906 00 0
411 01 01 842 00 0 908 00 0
412 01 01 844 00 0 910 00 0
413 01 01 846 00 0 912 00 0
414 01 01 848 00 0 914 00 0
415 01 01 850 00 0 916 00 0
416 01 01 852 00 0 918 00 0
417 01 01 854 00 0 920 00 0
418 01 01 856 00 0 922 00 0
419 01 01 858 00 0 924 00 0
420 01 01 860 00 0 926 00 0
421 01 01 862 00 0 928 00 0
422 01 01 864 00 0 930 00 0
423 01 01 866 00 0 932 00 0
424 01 01 868 00 0 934 00 0
425 01 01 870 00 0 936 00 0
```

33

Subrutinerna som sköter detta är A' och B'. A' plockar fram koden för pjäsen på rutan vars nummer finns i displayen vid anropet. B' kastar om pjäskoden i R₁₈ med koden i rutan som anges vid anropet.

Den viktigaste subrutinen är C'. Den utforskar systematiskt alla möjliga drag som en pjäs på en viss ruta kan utföra. Varje gång C' anropas ger den ett rutnummer och ett värde för pjäsen på rutan enligt följande:

- 0 Tom ruta
- 0,1 En bonde kan flytta två steg framåt till en tom ruta
- 1 Bonde kan tas
- 2 Springare kan tas
- 3 Löpare kan tas
- 4 Torn kan tas
- 5 Dam kan tas
- 5,1 En bonde kan flytta till sista raden och bli dam (tyvärr går bonden vid förflyttningen två steg framåt och förstår innehålllet i R₉.)
- 6 Kung kan tas

Vid värdering av draget används dessa värden upphöjda till 4. Detta värde jämförs med värdet för vad motspelaren kan ta. Om det egna värdet är större lagras draget som ett möjligt drag. Till att börja med har räknaren ett så stort handikapp att det första draget som hittas som inte medför att kungen tas accepteras. Sedan sänks handikappet så att nya drag endast kan ersätta det tidigare lagrade draget om de är bättre. På detta sätt hittas det drag som medför minsta pjäsförlusten.

HUVUDSAKLIG ANVÄNDNING AV REGISTREN

- 0 Använt
- 1-8 Spelman
- 9 Tecknet: räknarens färg (minus=svart, plus=vit); talet: rörelsemönster för springare
- 10 Rörelsemönster för löpare
- 11 Rörelsemönster för torn
- 12 Rörelsemönster för dam och kung
- 13 Sparar flagga 2 eller 3
- 14 Rörelsemönster för aktuell pjäs
- 15 Rutnummer för utgångsruta
- 16 Rutnummer för målruta
- 17 Värde för vad motspelaren kan ta
- 18 Används i subrutinen B'
- 19 Pjäskod för vad räknaren kan ta

BETYDELSE AV STÄLLD FLAGGA

- 0 Drag är möjligt
- 1 Det finns ingen ruta med sådant nummer
- 2 Springare eller kung
- 3 Bonde
- 4 Räknaren spelar mot sig själv
- 5 Räknaren har vita pjäser
- 6 En tom ruta har hittats vid sökning av pjäser att

- slå
- 7 Ingen skrivare

KOMMENTARER TILL PROGRAMMET

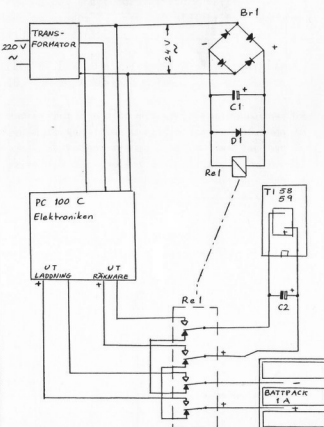
- 000-006 Avslutning av initiering.
- 007-044 Subrutin som återkallar kod för pjäs på en viss ruta.
- 045-066 Subrutin som skiftar pjäskoden i R₁₈ med pjäskoden i en ruta.
- 067-079 Förbereder ett EQ-test för att testa om pjäsen på en viss ruta är fientlig.
- 080-091 Beräknar målruta för en pjäs på rutan enligt R₁₅ genom att addera talet i displayen till talets absolutbelopp. (Målruta = ruta dit pjäs kan flytta.)
- 092-130 Kontrollera om ett rutnr. är giltigt (ligger på spelplanen). Flagga 1 ställs om rutan ligger utanför spelplanen och nollställs annars.
- 131-139 Initiering av program. Skriver rubrik och ställer in arbetsuppdatering (2 Op 17).
- 140-141 Här startar subrutinen C' som hittar alla möjliga drag för en pjäs.
- 142-145 Flagga 0 är ställd om C' tidigare har anropats för samma ruta.
- 146-163 Detta utförs första gången C' anropas för en viss ruta. Rutnr lagras med samma tecken som pjäsen på rutan har i R₁₅, flaggor initieras och pjäskod lagras i R₁₄.
- 164-179 Testa vilken pjäs som finns på rutan. Bonde behandlas i en särskild rutin. Kung och springare som båda endast får flytta en begränsad längd markeras genom att ställa flagga 2. I övrigt behandlas kungen som en dam.
- 180-188 Rörelsemönster för pjäsen plockas fram från något av R₉-R₁₂.
- 189-199 Ett rörelsemönster tas fram. Om slut på rörelsemönster utförs återhopp till huvudprogrammet med flagga 0 nollställd för att markera att inga fler drag kan göras med pjäsen.
- 200-206 Placerar utgångspunkt för förflyttning i R₁₆.
- 207-219 Adderar rörelsemönster till utgångsrutan för att få målruta. Kontrollera om målruta finns på brädet; upprepa i annat fall hela proceduren för nytt rörelsemönster.
- 220-235 Hämtar pjäskod från målruta. Om pjäsen är ens egen (ej kan tas) hämtas nytt rörelsemönster och nytt drag provas. Annars utförs återhopp till huvudprogrammet med pjäskod i displayen.
- 236-249 Hit sker hopp när C' har anropats tidigare. Bonde behandlas i en särskild rutin. Om målruta inte var tom eller pjäsen är kung eller springare kan den inte gå längre i samma riktning och nytt rörelsemönster plockas fram.
- 250-259 Nytt rörelsemönster.
- 260-266 Se steg 164-179.
- 267-290 Bondes förflyttning, initiering. Först provas förflyttning rakt fram ett steg och återhopp utförs om det går. Annars utförs hopp till

- steg 332 för att prova förflyttning diagonalt framåt. (Detta GTO kunde ha utesluts om rutinen vid 332 placerats direkt efter steg 287.)
- 291-331 Används ej första gången C' anropas. Provar att flytta en bonde två steg framåt. Om detta är möjligt returneras värdet 0,1 (detta värde används alltså som ett bättre drag än att flytta till en tom ruta). Om en bonde i nästa drag kan nå sista raden returneras 5,1. (Men här innehåller programmet ett fel: Om räknaren har den möjligheten kommer den att flytta utanför brädet och förstöra innehålllet i R₉.)
- 332-364 Bondeförflyttning diagonalt framåt provas.
- 365-390 Används ej första gången C' anropas. Kontrollerar vilka tidigare provförflyttningar av bonden som har gjorts och utför sedan ny provförflyttning, om möjligt.
- 391-417 Används av förflyttningsrutinen (735-769). Byter ut en ev. bonde på sista raden mot dam.
- 418-419 Start av huvudprogram.
- 420-446 Skriver ut drag, ställer flagga 5 om räknaren spelar vitt samt utför draget.
- 447-459 Initierar bl a var genomökningen av brädet skall starta.
- 460-471 Söker upp ruta med egen pjäs.
- 472-481 Ta reda på kod för pjäs som kan tas. (Sök upp ny ruta om inget drag kan göras.)
- 482-489 Beräkna dragets värde. Tag hänsyn till hur bra tidigare provdrag var.
- 490-534 Spara alla register och flaggor nödvändiga för att söka flera egna drag. Utför hittat drag. Förbered för att prova alla motdrag.
- 535-554 Sök upp en ruta med en av motspelarens pjäser.
- 548-555 Ta reda på vad motspelarens pjäs kan ta.
- 556-570 Gå vidare till nästa motspelardrag om tidigare motspelardrag var bättre än det nuvarande.
- 571-582 Om värde för det egna draget är sämre än motspelarens återställs register och flaggor för fortsatt sökning efter egna drag. I annat fall fortsätts genomökningen av motdrag.
- 583-605 Söker upp en ny ruta vid genomökning av motdrag. Hopp till 606 om ingen hittad.
- 606-627 Jämför skillnad mellan värde för vad räknaren kan ta och motspelaren kan ta. Om negativ skillnad förkastas draget och nästa drag provas. Annars sparas skillnaden för att drag som sedan provas ut endast skall kunna användas om de är bättre.
- 628-640 Lagra möjligt drag på formen xx.yy i H₄.
- 641-649 Kontrollerar om motspelarens pjäs kan flytta till tom ruta och ställer om så är fallet flagga 6.
- 650-659 Förbereder för att återställa eget provdrag genom att placera kod för ev pjäs som togs i R₁₈.
- 660-686 Återställer provdrag; återställer register och flaggor för att kunna fortsätta söka egna drag.
- 687-696 Se 490-534. (Sparar flagga 2 eller 3 i R₁₃.)

- 697-719 Hittar ny ruta vid genomökning av egna drag. Hopp till 720 om ingen hittad.
- 720-729 Alla egna drag provade. Återkalla hittat bästa drag. Om räknaren spelar mot sig själv sker hopp till en särskild rutin (se 789-794).
- 730-769 Skriver ut och utför drag i displayen på formen xx.yy. Spellet avbryts med hopp till 006 om antingen inmatat drag är 0 eller en kung tas. (DMS på steg 739 placerar yy i H₅.)
- 770-776 Byter färger genom att ändra tecknet på talet i R₉.
- 777-789 Initierar så att räknaren kan spela mot sig själv. Utan skrivare ställs flagga 7.
- 789-794 Används när räknaren spelar mot sig själv. Hopp hit sker när ett drag har hittats (se 720-729). Räknaren och motspelaren byter färger och det hittade draget matas in i huvudprogrammet med ett hopp till Lbl A. Utan skrivare visas först det hittade draget.
- 800-855 Skriver rubrik.

26 Gardering mot strömavbrott KOMPLEMENTFÖRTECKNING

- Br 1 8 125 C
- D 1 BAX 13
- C 1 4,7 µF/100 V
- C 2 2200 µF/10 V
- Re 1 4 pol vx1, subminiatur, flatpacktyp, 24 V, DC, 100 % inkoppl.



Adress: Bo Nordlin, Stora Björnens gata 70, 136 64 HANDEN

Den del av mallen som tas med i tidningen innehåller följande uppgifter: programbeskrivning, hävdarna nödvändig för utrytning av programmet, programmerarens namn och datum för insändandet. Dessutom ger det åtta-siffriga identifikationsnumret, t ex 75593007, ytterligare upplysningar. De två första siffrorna betecknar jämsområde enligt tabellen nedan. Nästa två siffror betecknar räkna som programmet är avsett för. Den femte siffran betecknar källan:

- De tre sista siffrorna är ett löpnummer som programförmedlaren tilldelar.

Skriv programdokumentation på skrivmaskin eller läsligt för hand med svart skrift. För att göra hälslagning möjlig skall marginalerna på samtliga papper vara minst 2 cm breda (undantag: monterade programlistningar). Dokumentation skall som minimumkrav bestå av: ifyllt mall, bruksanvisning för körning och monterad listning. Skicka med en inspelad version av programmet om du har möjlighet. Vi returnerar magnetkort eller kassetter.

För varje program som antas får du ur Programbiblioteket välja antingen: 4 st svenska eller engelska program

BUSINESS		STATISTICS & PROBABILITY	
01	Management Accounting	20	Regression/Curve Fit
02	Manufacturing Engineering	21	Analysis of Variance
03	Inventory Control	22	Statistical Testing
04	Marketing/Sales	23	Statistical Inference
05	Personnel	24	Stochastic Processes
06	Transportation	25	Probability Theory
07	Insurance	26	Probability Distributions
08	Rent Estate	27	Quality Assurance
09	Business (General)	28	Probability/Reliability
		29	Statistics (Probability & normal)
FINANCE		MATHEMATICS	
10	Accounting	30	Linear Algebra/Matrixes
11	Auditing	31	Complex Variables
12	Banking	32	Harmonic Analysis
13	Consumer Finance	33	Nonlinear Systems
14	Personal Finance	34	Nonlinear Systems
15	Economics	35	Nonlinear Integration
16	Lending	36	Differential Equations
17	Real Planning/Preparation	37	Number Systems
18	Securities	38	System Modeling
19	Finance (General)	39	Operations Research
			Mathematics (General)

40	Physics	60	Aeronautical Engineering
41	Chemistry	61	Chemical Engineering
42	Biology	62	Civil Engineering
43	Agriculture	63	Computer Science
44	Forestry	64	Electrical Engineering
45	Ecology	65	Electronic Engineering
46	Geology/Resources	66	Mechanical Engineering
47	Oceanography	67	Nuclear Engineering
48	Anthropology	68	System Engineering
49	Natural Sciences (Other)	69	Engineering (General)

LIFE SCIENCES		TECHNICAL	
50	Clinical / Diagnostic	70	Acoustics
51	Virology/Immunology	71	Architecture
52	Pathology	72	Ceramics
53	Biochemistry	73	Heating, Air Cond., Cool.
54	Genetics	74	Optics
55	Physiology	75	Programming
56	Pharmacology	76	Seismology
57	Ophthalmology/Optics	77	Surveying
58	Nutrition/Food Science	78	Astronomy
59	Life Sciences (General)		

SOCIAL & BEHAVIORAL SCIENCES

80	Political Science
81	Sociology
82	Psychology/Psychiatry
83	Law Enforcement
84	Social & Behavioral Sciences (Other)

NATURAL RESOURCES

85	Lumber/Forest Products
86	Oil/Gas/Energy
87	Food Resources
88	Water Resources
89	Natural Resources (Other)

GENERAL

90 Utility Programs
91 Demonstration/Games
92 Education
93 Air Navigation
94 Marine Navigation
95 Photography
96 Music
97 Astrology
98 Sports
99 Other

utan magnetkort eller 3 st svenska TI-59-program med magnetkort eller 3 st belgiska program utan magnetkort. I stället för program kan du få ett gratis medlemskap när du har fått sammanlagt 6 program antagna.

Om någon hittar ett fel i ditt program sedan det tagits in i biblioteket får du en upmaning att revidera det. Det reviderade programmet ersätter sedan det felaktiga programmet i biblioteket. Försök till revidering av ditt program av en annan programmerare betraktas som ett förslag och kan tas in i biblioteket som en revidering endast efter ditt godkännande.

Beställning av program sker genom inbetalning av aktuellt belopp till föreningens postgiro 430 01 59-3. För varje beställning uttas en beställningsavgift på 10 kr. Endast svenska och engelska program lagerförs. Övriga beställs en gång i månaden från den utländska klubben.

Blanketten "Användarsynpunkter" följer med varje program. Fyll i den och skicka den till programförmedlaren om du har synpunkter på programmet. Synpunkterna kommer att vidarebefordras till programmeraren.

Svenska och engelska (BTIUC) program utan magnetkort	10,-
Svenska TI-59-program med magnetkort	15,-
Belgiska program (TISOFT) utan magnetkort	15,-
Katalog med belgiska och engelska program	20,-

Programbeskrivning: Programmet utför regressionsanalys på inmatade datapar. Dataparen kan innan analysen transformeras på tio olika förprogrammerade sätt eller på något av dig fritt valt sätt. Inmatade datapar kan lätt ändras.

Ansatt steg: 213 Svarere: ☐ ja ☒ nei ☐ valgfritt CBOR nr.: mch CBOR antall: Bandspelare: ☐ ja ☐

Øvrige tilbehør: Gjenstand: Inventar dato: 200810

Programbeskrivning: Detta interpolationsprogram klarar på en TI-59 av upp till 48 punkter (TI-58 18 punkter). Inmatningen är mycket enkel. Eventuella fel vid inmatningen kan lätt ändras.

Antal steg: 150	Skivare: ☐ ja ☒ nej ☐ valfritt	CRM nr: och	CRM antal:	Bandspelare: ☐ ja ☐ nej
-----------------	--	-------------	------------	---

Programbeskrivning: Programmet kan lista samtliga register med 13 siffrors
noggranhet (begränsning: 10-exponent trycks aldrig ut - mycket små
ellen mycket stora tal kan därefter inte listas korrekt). Man kan v

mitt i en listning. Vid alfanumerisk listning "känner" programmet av om skrivkoden är avsedd för Op- eller NIR-laddning.

Antal steg: 385	Skrivare: ☒ Ja ☐ Nej ☐ Kalifritt	CRON nr: och	CRAN antal:	Bandspeltare: ☐ Ja ☐ Nej
-----------------	--	--------------	-------------	--

Programbeskrivning: Programmet testar och tränar din förmåga att räkna i huvudet, genom att ge slumpmässigt valda uppgifter i de fyra räknesätten. Du väljer själv räknesätt och svårighetsgrad. Programmet kontrollerar om du har fuskat, dvs. räknat utanför den på räknaren.

Antal steg: 134	Skrivare: ☐ ja ☒ nej ☐ halffritt	CDROM nr: och	CDROM antal:	Bandspelare: ☐ ja ☐ nej
Övrigt tillbehör:				
Namn: Rikard Gustafsson				Insat datum: 820825

Säljes: TI 58 C 500:-. Jan Björklund
tel. 0303/ 101 82 efter kl 15.30.

Säljes: TI 58 obetydligt begagnad 500:-
Ulf Sjöstrand, 08/ 86 52 43 efter 18.

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

Det har kommit brev till Programbiten från en f d verktygskonstruktör, numera sjukpensionerad. Intresserad av att konstruera elorgel, tonteori och Fourieranalys. Nybörjare i programmering. PB vidarebefordrar brev till honom från ev. intresserade.

FÖR AVANCFRADE HEMBYGGARE

Red, här fått se ett specialnummer av den tyska tidskriften "Chip" som måste vara en guldgruva för alla som vill bygga tillätsater till sina TI-58/59:or och skrivare. Tjugotalet artiklar om interface, alfanumeriskt tangentbord, fram- och bakckörning av skrivaremsan, XY-skrivare m.m. Priset blir med nya växelkursen ganska precis hundralappen (DM 33,50). Beställ från CHIP-Leserservice, Vogel-Verlag, Postfach 6740, D-8700 WÜRZBURG i Västtyskland, så skickar de räkning.

Den minsta hemdatorn du kan köpa från Texas Instruments har 16K minne.



Börja din "datakarriär" på vettigaste sättet – med en hemdator från Texas Instruments. Den ger dig en total minneskapacitet upp till 110K Bytes RAM/ROM. Dessutom ger den ett komplett utbyggnadsprogram, både vad det gäller programvara och kringutrustning. Så, när ditt intresse för databehandling ökar har du en hemdator som kan växa i samma takt.

Att det är en god investering, även på sikt, förstår du när du jämför det rimliga priset med hemdatorns möjligheter.



TI 99/4A är en högt utvecklad dator konstruerad inte bara för nybörjare, med sitt enkla sätt att arbeta, utan också för den professionella användaren tack vare sin kraftfulla 16 bits mikroprocessor och högt utvecklade programvara. Hemdatorn kan även anslutas till vilken vanlig TV som helst.

Den har färggrafik med hög upplösning (256 x 192 punkter) och kan återge 24 rader med 32 tecken per rad i 16 olika färger. Med tre tongeneratorer som spänner över fem oktaver kan du komponera musik och hitta på ljud effekter. Med ett specialtillbehör kan man t.o.m. åstadkomma syntetiskt tal!

Som standardspråk används BASIC, men du kan också få UCSD-PASCAL, TI-LOGO och ASSEMBLER.

TI 99/4A kan du köpa för 3.995:- (ca. pris basenheten). När du önskar lösa problem kan du använda någon av Texas Instruments omfattande urval av lättanvändbara programmoduler (Solid State Software). Över 600 mjukvaruprogram är redan tillgängliga. Som du ser, ett prisvärt system, vad man än jämför med.

Avancerad teknik och realistiska priser. Något som är naturligt att vänta sig från företaget som uppfann mikroprocessorn, den integrerade kretsen och mikrodatoren.



Vi hjälper dig att göra ditt bästa.

TEXAS INSTRUMENTS