Claes PR ex

PROGRAM BITEN 80-4



Innehåll

Dyre medlem!	2
Insänt	3
Utmaningen	4
Pgm-bitar	8
Kommentar till handboken	10
Avbetalning	13
Game of life	14
Bråkfinnaré 2	18
Spline	20
Fermats sats	22
Djupdykning i TI-59 (2)	23
Tröghetsmoment TI-57, 58/59	24
Komponentritningar	29
HIR - vackrare vardagsvara	30
EE-modulen och ett rotlokusprogram	34
Fakultetsprogram i fast mode	38
Egen modul	39
Recensioner	40
Tillbehör till TI-59	41
Programbiblioteket	42



Dyre medlem!

Här kommer detta års sista nummer. Efter en litet långsam start i våras har vi försökt komma in i regelbundna utgivningsbanor (i stort sett vid kvartalsskiftena med någon försening i okt och tidigareläggning vid jul). Vi uppnådde inte de 192 sidor vi satt som mål (det blev 164) men vi har åstadkommit vad som var möjligt och vi hoppas tidningen givit vad läsarna väntat. Vårt mål är att det skall vara trevligt att läsa den, så trevligt att läsarna gärna sänder in egna bidrag. Ropet efter granskare i förra numret gav ett trevligt svar men det finns plats för fler.

Och i detta sammanhang ett hjärtligt tack från föreningen till alla som bidragit till tidningen - författare, granskare och klistrare. Vi är kanske ibland dåliga med att skriva svar till dem som sänt in bidrag, men det beror inte på bristande uppskattning!

Uppslaget med den egna modulen fick kanske inte det gensvar vi trott. Kontakter med andra föreningar har dessvärre delvis varit negativa. Men desto större anledning för dem som tror på idén att komma med förslag!

Innan nästa nummer kommer ut skall föreningen ha årsmöte. Utnyttja Din möjlighet att komma och påverka vår verksamhet!

Trevlig helg och ett gott 1981!

Lars Hedlund, ordf

PS
Betala gärna in årsavgiften för 1981 (120:-)
DS

KALLELSE TILL

ÅRSMÖTE

lördagen den 28 februari 1981 kl. 14.00
på Djupdalsvägen 127, Bromma (500 m från Drottningholmsv./Åkeshovsv, strax S om Gustaf III:s väg)

DAGORDNING:

1. Mötets öppnande. Dess behöriga utlysande. Godkännande av dagordning.
2. Val av ordf och sekr för mötet.
3. Val av två justeringsmän.
4. Verksamhetsberättelse.
5. Revisorernas berättelse.
6. Fråga om ansvarsfrihet för styrelsen.
7. Val av styrelse, redaktion, valberedning, revisorer och övriga ombud.
8. Årsplan för verksamhetsåret 1981.
9. Årsavgift.
10. Till styrelsen inkomna förslag.
11. Övriga frågor.

Förslag som önskas behandlat på årsmötet skall vara styrelsen tillhanda senast den 6 febr 1981.

Vid mötet serveras smörgås till den som anmält deltagande senast tre dagar före mötet.

TILLBEHÖRSFÖRSÄLJNINGEN

Från föreningen kan Du köpa följande, bäst genom att sätta in köpesumman på vårt postgirokonto 430 01 59 - 3:

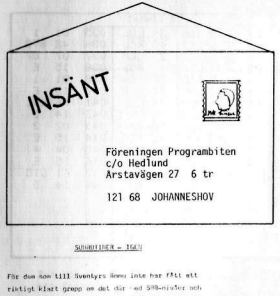
Sourcebook for Programmable Calculators	175,-
40 st tomma magnetkort och plånbok	108,-
Tom magnetkortsplånbok	10,-
Föreningens programmeringsblanketter, olika typer (se PB 79-3&4), block om 50 blanketter	11,-
Programbiten 1978-79 (4 nr + 1 dubbelnr)	100,-

Mallar till programbiblioteket sändes i rimlig mängd utan kostnad.

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

Mångfaldigande av innehållet i denna skrift, helt eller delvis, är enligt lag om upphovsrätt av den 30 december 1960 förbjudet utan medgivande av Föreningen Programbiten. Förbudet gäller varje form av mångfaldigande genom tryckning, duplicering, stencilering, bandinspelning, magnetkortsinspelning etc.

Tryck och distribution: Direct Marketing AB.



SUBMITTED - 1560

För dem som till Sventyrs ännu inte har fått ett riktigt klart grepp om det där med SUB-program och retur-registret kanske den här lilla övningen kan vara läsvärd.

Knappa först in det här undantliga pgm:

Lbl A	CLR 9 + Rtn	30 Lbl A' E 4 + Rtn
6 Lbl B	A B + Rtn	36 Lbl B' A' 3 + Rtn
12 Lbl C	B 7 + Rtn	42 Lbl C' C' 2 + Rtn
18 Lbl D	C 6 + Rtn	48 Lbl D' C' 1 + Rtn
24 Lbl E	D 5 + Rtn	54 Lbl E' D' Rtn$_{57}$

Tryck sedan 1 tur och ordning A, B, C ... E'i den ordning de står i pgm. Men innan du trycker en knapp försök räkna ut vad du svall få för resultat. I början är det enkelt, men fortsätt i alla fall till slutet, för det är där som de ev överraskningarna väntar!

blm

1.

Audun Hjortås skrev i Pgm-biten 3/80 om STO OG RCL MED SAMME TANGENT. Det han där föreslår är ytterst användbart och generellt. Kort formulerat föreslår han ju helt enkelt att man kan "fråga" vilken variabel man tidigare lagrat med en A-E'knapp genom att ha 0 på displayen och trycka på respektive knapp, varvid variabelns storlek visas.

I princip borde alla program skrivas så! Enda undantaget naturligtvis program där man ofta måste mata in just noll.

En liknande programmeringsteknik kan stoppa utförande av en beräkning om inte alla variabler matats in (minnesinnehåll skiljt från 0).Om man så vill.

2.

Om man använder sin räknares A-E'tangenter för omvandlingar mellan olika enheter, som i standardmodulens ML-24 och ML-25 med fot till meter, Celsius till Fahrenheit etc, så kan man tillämpa följande enkla teknik för att få plats med fler omvandlingar per kort/tio tangenter. Om man gör egna program.

Låt till exempel A omvandla fot till meter. I stället för att ta till en tangent extra för att få omvänd funktion, meter till fot, så använder vi E'som "tvärt-om-knapp". E'hissar en godtycklig flagga. När A skall utföras, sker hopp till omvänd funktion om flaggan är hissad. Sist av allt innan svaret presenteras nollställs flaggan.

På detta sätt får man in 18 st omvandlingsfunktioner i st f 10. Om man skippar en funktion plus dess baklängesfunktion så får man in hela jobbet som ML-24 och ML-25 (eller motsvarande) gör tillsammans på bara de tio användardefinierade tangenterna.

Tomas Dyfverman

Jag har ibland använt mig av Fix Ind kommandot för att styra displayens utseende beroende på metodens noggrannhet. Stoglängden påverkar ju t.ex. resultatet och jag tycker att 1.000 är snyggare än 3.9999. Men ibland händer det motsatta. Man vill på något sätt låta programmet "känna efter" vilken decimalplacering som är inställd. Då kan man använda följande sekvens: "(9 1/x - EE INV EE) 1/x Log Int" som direkt ger ett tal som motsvarar decimalplaceringen. Undantaget är Fix 9 som får utmatningen 10, men detta saknar praktisk betydelse, eftersom alla tal större än 9 ger flytande decimalkomma om man lagrar dem i något minne och senare använder dem till att ändra displayen med.

Om man har ett program där antalet minnen som behövs varierar med någon inmatad faktor, som t.ex. matrisprogrammen i modulen, kan det vara fördelaktigt att skriva (och spela in) det i normaluppdelning och sedan låta den inmatade faktorn styra uppdelningen med Op 17. På så sätt får man alltid största möjliga utrymme tillgängligt för andra program, samtidigt som man slipper en massa trassel med att ändra uppdelningen innan man läser programkortet.

P2Mk

I programbiten 80-3 sid 16 skrivs det om slumptal av olika slag. De vanligaste slumptalen är utan tvivel de s.k. pseudoslumptalen, olika metoder att få fram dessa beskrivs i artikeln. Men det finns faktiskt en ganska enkel metod att få fram äkta slumptal även i en räknedosa. Metoden som används på flera hemdatorer, fungerar så här:

När datorn inte "gör någonting", dvs väntar på instruktioner från tangentbordet, så kontrollerar den många gånger per sekund ifall någon knapp blivit nedtryckt. Detta görs av ett program i maskinen, det s.k. monitorprogrammet. Ofta fungerar monitorprogrammet så att för varje gång det kontrollerar om någon tangent blivit nedtryckt, räknas ett speciellt slumptalsregister upp med 1 enhet. Detta slumptalsregister kan innehålla heltal mellan 0 och en viss övre gräns, t.ex. 99, 255, 65535 eller något annat. När slumptalsregistret nått denna övre gräns räknas det nästa gång ner till 0, och så börjar allt om från början igen. När man trycker på en tangent, avbryts detta och en siffra matas in i maskinen, ett program utföres eller något annat. Så fort maskinen väntar på en ny tangentnedtryckning, räknas slumptalsregistret upp igen.

Slumptalsregistrets exakta innehåll kommer m.a.o. att bero på exakt hur lång tid det tar för oss att trycka ner tangenterna, eller rättare hur lång tid det tar innan vi trycker på dem. Och det är något vi knappast kan kontrollera, så förutsatt att det tar inte alltför lång tid att räkna upp slumptalsregistret från 0 till max (dvs högst ca 1 sekund, helst kortare tid ändå), blir det slumptal vi får ett äkta slumptal. Varje tal i serien blir praktiskt taget helt oberoende av det närmast föregående talet - och men kan aldrig, som med en pseudoslumptalsserie, exakt upprepa en serie av slumptal med denna metod.

För att metoden ska fungera, måste programmet göra en paus och vänta på en tangentnedtryckning för varje slumptal man behöver, och man kanske istället vill att programmet ska gå kontinuerligt. Den bästa lösningen är då att stoppa in en pseudoslumptalsgenerator i programmet, men att låta frröet till denna generator bestämmas av denna äkta slumptalsmetod.

Vissa räknedosor, t.ex. SR-51-I samt flera av Casios modeller, har en inbyggd slumptalsgenerator. Om de fungerar på det sätt jag beskrivit här känner jag inte till, men metoden används på Atminstone en hemdator (Apple II), trots att den är långsam även på flera andra hemdatorer. Och den skulle fungera bra även på en räknedosa, om man satte slumptalsregistrets max-värde måttligt högt, t.ex. 99 eller 999, detta för att räknedosor är mycket långsamma jämfört med datorer.

Paul Schlyter

UTMANINGEN!

Redaktörens adress:
Sven Östberg
Hammarvägen 39
863 00 SUNDSBRUK

Hej.

Alla som har varit undrande till Roland Månssons primtalsprogram i förra numret, kommer här att få förklarat hur det fungerar. Per Eric Holmberg, Katrineholm har i en klargörande uppsats "Primtal och primfaktorer" visat hur man kan finna serier, som garanterat genomlöper alla primtal. Han visar också att det finns ett oändligt antal sådana serier, men att bara ett fåtal av dem är lämpliga i primtalsprogram.

Samtliga uppgifter, utom programexemplen, i redogörelsen nedan, är hämtade från hans uppsats.

För att finna en talserie, som genomlöper alla primtal, börjar vi med att skriva upp (tillräckligt många) heltal

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

Vi kan välja att stryka alla tal, som är delbara med primtalen 2 och 3. Då får vi talserien (b):

1 5 7 11 13 17 19 (b)

Vi kan genast inse att denna serie genomlöper alla primtal utom 2 och 3.

En ytterligare talserie får vi, om vi ur (b) bildar "differensen till nästa tal". Då får vi talföljden (c):

4 2 4 2 4 2.., (c)

Låt oss kalla dessa differenser för tillskott. Tillskottsföljden (c) består av upprepningar av talföljden "4, 2". Vi kan därför säga att den är periodisk.

En ny period börjar vid 1, 7, 13, 19 osv. Periodlängden P kan därför sägas vara, i detta fall, 6 enheter.

Antalet tillskott per period, M, är 2 st (4 och 2).

Med hjälp av tillskottsföljden (c) kan vi nu skriva ett primtalsprogram, se program 1. Testar vi snabbheten hos detta program, finner vi att det konstaterar att talet 396 733 är ett primtal på 177 sek (vilket ska jämföras med primtalsprogrammet i MU-modulen: 214 sek och Roland Månssons program: 101 sek).

Program 1.

000	76	LBL	016	76	LBL	035	01	1
001	15	E	017	43	RCL	036	42	STO
002	44	SUM	018	25	CLR	037	07	07
003	07	07	019	43	RCL	038	15	E
004	43	RCL	020	06	06	039	01	1
005	06	06	021	91	R/S	040	15	E
006	55	÷	022	76	LBL	041	76	LBL
007	43	RCL	023	11	A	042	14	D
008	07	07	024	50	IXI	043	02	2
009	77	GE	025	42	STO	044	15	E
010	43	RCL	026	06	06	045	04	4
011	95	=	027	34	fX	046	15	E
012	22	INV	028	85	+	047	61	GTO
013	59	INT	029	35	1/X	048	14	D
014	35	1/X	030	95	=			
015	92	RTN	031	32	X:T			
			032	25	CLR			
			033	86	STF			
			034	08	08			

Serien (b) kallar vi 3-serien, eftersom alla tal delbara med primtal upp t o m 3 är strukna. Väljer vi att stryka alla tal delbara med 2, 3 och 5 får vi 5-serien (d):

1 7 11 13 17 19 23 29
31 37 41 43 47 49 53 59 (d)
61 67 71 73 77 79 83 89...

Om vi ur (d) bildar 5-seriens tillskottsföljd

6 4 2 4 2 4 6 2.... (e)

finner vi att också den är periodisk, på precis samma sätt som för 3-seriens tillskottsföljd. (P = 30 enheter och M = 8 st). Därmed kan vi skriva ett primtalsprogram byggt på 5-serien. Se program 2, programsteg 000 - 035 är samma som för program 1.

Program 2 är snabbare än det föregående (130 s), men det kräver mer minnesutrymme. (Obs även att tillskottsföljden i programmet inte har samma ordning som (e).)

Väljer vi att stryka alla tal delbara med 2, 3, 5 och 7 får vi 7-serien. Stryker vi alla tal delbara med 2, 3, 5, 7 och 11 får vi 11-serien. Vi kommer finna att tillskottsföljden för dessa serier, är periodiska.

Program 2

035	01	1	051	04	4			
036	42	STO	052	15	E			
037	07	07	053	02	2			
038	15	E	054	15	E			
039	01	1	055	04	4			
040	15	E	056	15	E			
041	02	2	057	06	6			
042	15	E	058	15	E			
043	02	2	059	02	2			
044	15	E	060	15	E			
045	76	LBL	061	06	6			
046	14	D	062	15	E			
047	04	4	063	61	GTO			
048	15	E	064	14	D			
049	02	2						
050	15	E						

Tabell 1

n-serie	M antal tillsk.	P period-längd	P/M snabbhet	anm.
1	1	1	1,000	
2	1	2	2,000	MU-program
3	2	6	3,000	program 1
5	8	30	3,750	program 2
7	48	210	4,375	
11	480	2 310	4,813	Roland Månssons pgm
13	5 760	30 030	5,214	
17	92 160	510 510	5,539	

För en godtycklig n-serie kommer vi alltid att finna att

- tillskottsföljden är periodisk och att periodlängden P kan beräknas som "produkten av de strukna primtalen".
- antalet tillskott per period, M, kan beräknas som "produkten av de strukna primtalen subtraherade med 1".

Speciellt för 5-serien gäller då att

$$P = 5 \cdot 3 \cdot 2 = 30$$

och att

$$M = (5-1) \cdot (3-1) \cdot (2-1) = 8$$

Ur dessa konstateranden kan vi, för en godtycklig n-serie, beräkna M och P. Se tab 1.

Konstruerar vi ett primtalsprogram med hjälp av en n-serie, kan vi säga att M är ett mått på programmets minnesbehov. Kvoten P/M, också tabellerad i tab 1, är ett mått på programmets snabbhet. (I sin uppsats har Per Eric använt begreppet "relativ volym", kvoten M/P, som ett mått på tidåtgången. Med hans tillåtelse har vi alltså ändrat på detta.)

Ur tabellen kan vi se, att då n stiger, ökar programmets snabbhet, men också dess programminnesbehov. Snabbheten ökar långsamt, men programminnesbehovet ökar mycket snabbt.

Roland Månsson valde 11-serien till sitt primtalsprogram. Ett program konstruerat med 7-serien blir 9% långsammare, men kräver bara tiondelen så mycket programminne.

Liknande jämförelser kan göras för de övriga n-serierna. Vilken den högsta praktiskt användbara n-serien är, varierar beroende på använd räknedosa (eller dator). Att använda någon serie långt ovanför 11-serien torde inte vara önskvärt eller ens möjligt.

Så långt Per Erics uppsats. Hur kan man ge en förklaring till (bevisa) att tillskottsföljden alltid är periodisk? Och att P och M kan beräknas enligt reglerna ovan? Kan man bevisa att kvoten P/M går mot något gränsvärde då n växer? Eller att det inte går mot något gränsvärde? Den teoretiskt sinnade analytikern har här möjlighet att komma med bidrag.

xxxxxx

I förra numret av Utmaningen efterlyste jag olika sätt att optimera Patrik Johanssons e-program. Då termen $1/n!$ blir liten (inleds med många nollor) använder programmet en stor del av sin arbetstid att arbeta med nollor. Det dividerar 0 med N i beräkningsrutinen och det summerar nollor i summationsrutinen. Genom att sätta in en sökrutin, som letar efter det första registret, vars innehåll inte är noll, fick Patrik sitt program att gå nästan dubbelt så fort.

För att öka antalet decimaler i resultatet kan man göra om beräkningsrutinen, så att decimalbråksutvecklingen görs med 13 siffror per register. Man kan också flytta över arbets- och pekregister till HIR-stacken. Med Patriks beräkningsmetod kan man då maximalt beräkna e med 650 siffror.

Björn Gustavsson, Smedjebacken, har tacklat problemet på ett litet annorlunda sätt, men fått fram ett helt annorlunda resultat. Så här skriver han:

Här är en lösning till e-beräkningen. Jag har lyckats göra ett program som beräknar e med 980 siffror på 6½ timme. Jag har skrivit om formeln på följande sätt:

$$e = 1 + 1/1! + 1/2! + 1/3! + 1/4! =$$
$$= (((1/4 + 1)/3 + 1)/2 + 1)/1 + 1 =$$

Nu tog jag bara med fem termer som exempel - men formeln gäller förstås för godtyckligt många termer.

Metoden blir alltså följande:

1. Sätt N = ett tillräckligt högt tal.
2. Sätt T = 1.
3. Låt T = T/N + 1
4. Minska N med 1 och gå till 3 om N är skilt från noll.
5. Skriv T och stoppa

I mitt program lagras N i register 99 och T lagras i registren 1 - 98. Divisionen av T utföres i programstegen 033 - 059, ungefär som i Patrik Johanssons program, men kortare och snabbare.

Beträffande stegen 033 - 037 sker multiplikation med 10^{10} där. Kanske ett underligt sätt att göra det på, men jag tror det är snabbare än x 1 EE 10 som är lika långt

Beträffande valet av N: Enligt beräkningar med ett fakultetsprogram har jag funnit att 458! har 979 siffror. När jag sedan skrev programmet drog jag till med 450.

Programmets 10 första programsteg används till att försätta räknaren i sk snabbhetstillstånd.

Bruksanvisningen för programmet blir:

1. Se till att programmet är inspelat på magnetkort i normal uppdelning, och ML-modulen är isatt.
2. Läs in programmet och tryck på RST R/S.
3. Läs in programmet igen. Programmet startar helt automatiskt. Det går inte att stoppa, varken med RST eller R/S

Vad felet i resultatet beträffar tror jag att felet i sista siffran maximalt är e enheter. Detta resultat kommer man fram till om man antar att felet i varje division är 1 enhet i sista siffran.

Björns program finns listat,se program 3, dess resultatutskrift tar jag inte med här, av skäl som strax framgår. De fem sista siffrorna blev ...69655.

Björns program kan optimeras på samma sätt som föreslogs till Patriks program. Genom att lagra 13 siffror i alla 100 registren, borde man kunna beräkna e med 1300 siffror. Mycket riktigt, det dröjde inte länge innan jag fick ett nytt program från Björn som klarade detta, se program 4.

Björn har löst det besvärliga problemet med decimalbråksutveckling med 13 siffror per register, genom att dela upp divisionen i två delar: först en division på vanligt sätt med 10 siffror, sedan en division med 3 siffror.

HIR 8 används som DSZ-register, pgm-steg 114-116 utför "decrement" och 122-128 utför "skip on zero". HIR 7 används för de 13 sista decimalerna, i stället för R0 som används till pekregister. När programmet är färdigkört kopieras innehållet i HIR 7 över till R0. Därefter kan hela datainnehållet överföras till magnetkort.

Programmet körs på samma sätt som det förra, med ML-modulen isatt. Efter avslutad körning, tryck på RST och spela in alla fyra blocken på magnetkort. Genom ett separat utskriftsprogram, som använt MU-modulen och skrivare, har sedan Björn listat hela datainnehållet, se tab 2.

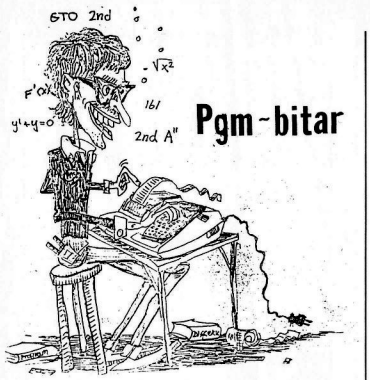
Utskriften är på 1300 siffror. Vi kan därför kontrollera tidigare beräkningar. Felanalysen till Patriks program har visat sig vara alldeles korrekt. Vi beräknade att de 5 sista siffrorna borde bli ...90602 med en rimlig variation uppåt och nedåt på 9 enheter. Det exakta värdet ska bli ..90599 (avrundat).

Program 3

000	00	0	033	22	INV	068	22	INV
001	00	0	034	52	EE	069	49	PRD
002	00	0	035	52	EE	070	98	98
003	00	0	036	01	1	071	25	CLR
004	00	0	037	00	0	072	09	9
005	36	PGM	038	85	+	073	08	8
006	02	02	039	73	RC*	074	42	STD
007	71	SBR	040	00	00	075	00	00
008	02	02	041	75	-	076	98	ADV
009	39	39	042	53	(	077	73	RC*
010	09	9	043	24	CE	078	00	00
011	00	0	044	55	÷	079	99	PRT
012	22	INV	045	43	RCL	080	97	DSZ
013	58	FIX	046	99	99	081	00	00
014	22	INV	047	54	)	082	00	00
015	57	ENG	048	59	INT	083	77	77
016	01	1	049	72	ST*	084	25	CLR
017	00	0	050	00	00	085	91	R/S
018	69	DP	051	65	×			
019	17	17	052	43	RCL			
020	47	CMS	053	99	99			
021	04	4	054	95	=			
022	05	5	055	97	DSZ			
023	00	0	056	00	00			
024	42	STD	057	00	00			
025	99	99	058	33	33			
026	61	GTD	059	01	1			
027	00	00	060	52	EE			
028	59	59	061	09	9			
029	09	9	062	44	SUM			
030	08	8	063	98	98			
031	48	EXC	064	97	DSZ			
032	00	00	065	99	99			
			066	00	00			
			067	29	29			

Tab 2

e = 2.718281828459045235360287471352662497757247093699959574966967627724076630353547594571382178525166427427466391932003059921817413596629047976108401948440634623572900334295260595634249684875602336248197381323266279434904717329621935561038272024120845404484280767006172135601663515105707310914837180652458810975665933446128475648233786783165271201909145648566923460348610454326648213393607260249141273724587006606315577488168400188702038807342607801690151579063306740813342667048601941809466072301552094457812864609948303290216510342081053032310128185233645312414062043439943763830265879569733318116606382690286205671431452356934616313305754071681809927990480989332153967111821082148642614841694676283379074860943296062094971759600694796511963523172678408819338779513491567089477252449171382470960864034418159813629774771309960518707211349999834406283637150104681695681430784286433593059957643002740840278137126856800919280424238430492283209391911377092847581847878746288596792596731492337131702161515946747069106489860340082239019232330221130689411279751731864823311949431995808198863818488384754606960424407738115187247808539348162059535662411003436147554460703194617362238824464564850263961822368947177072839981853677834699815628618255012070239053823094729054629406586538150375340568894527261048107510477528489366112157824272310883938105754576718057527912242730829420903963821178529282347808000769319468753743194173788596486774865463659483289552368362851371347974949538158420562230272408328425330586481974798612013491873983125397547891193946612280868483816248642277013609479310476240264464682178497065375774224599505239638878943531640998245251269493725596100408603406024678583601569431268150212480294431221035896954651614173176852573642318571660487232788866812383859793992486572683418338823481546853674366536722244781495396887597611535764874319722917821049865231162151160949432014506412384786310362473090708238759658332243559075091403931723278417007528557861489112984481135430927817601858283096840644502916924588894316229493069839430342121702985461573958001270914647564148007577421590109403954394368308951594031330243764332462641539180678614229038621335730834816515409399320754176516560668076002228406230136994093285745568502876187356539531449702483596206507807572205836561548434638810516283264825380675310096301631782908652501993761039795458150855068631936373883856986705435849699922576345962284254567882076479291198825142683501856907246170050355829781232786203173989484639106359495725719354856493662168862245686755849274168096664365034862628612832092190710186031480331870585307764657312752035361369297522653383421884572547014380694113676388311475938314139393674327650005684155030984677468916311366175595657993956382815488337189798332809100329392296389522578477502851145535603486649437076165738627809530033945677153855432864837492124306035392618297259549054906605323060076596116354838216304577348034550543982905618719003222656250000000000



DECIMALPUNKTSKNEPET

Decimalpunktsknepet är ett sätt att göra pgm användarvänliga. Det används flitigt i TI PPC NOTLS.

För att visa knepet har jag förbättrat "STO OG RCL MED SAMME TANGENT" från förra numret. Direkt efter varje label har jag placerat ".". Det får den effekten att ett tal på skärmen, som är resultatet av en beräkning, raderas. Däremot, om man nyss har matat in ett tal, blir det kvar.

I pgmet nedan kommer varje tal som användaren matar in innan han trycker på A eller B att lagras, men trycker han bara ned en av tangenterna återkallas det lagrade talet. Användaren slipper trycka ned CLR.

Björn Gustavsson, Smedjebacken

```
000  76 LBL      012  76 LBL
001  11  A       013  12  B
002  93  .       014  93  .
003  29 CP       015  29 CP
004  67  EQ      016  67  EQ
005  00  00      017  00  00
006  09  09      018  21  21
007  42 STD      019  42 STD
008  01  01      020  02  02
009  43 RCL      021  43 RCL
010  01  01      022  02  02
011  92 RTN      023  92 RTN
```

Manuell förlängning av ADV i pgm under pågående exekvering

När exekveringen av ett pgm kommer fram till en ADV-instruktion, utföres givetvis denna.

Men om ADV-knappen på PC 100 eller decimal-tangenten på TI 59 hålls nedtryckt under exekveringen, så fortsätter pappersframmatningen efter ADV-instruktionen, tills tangenten släpps, varefter exekveringen fortsätter med nästa pgm-steg.

Detta kan användas, om man någon gång vill öka avståndet mellan två printar på remsan från i programmet angivet till enligt egen önskan.

P23a

ATT VÄNDA ETT TAL

Om man vill vända ett tal, så att det skrivs baklänges, så kan man använda denna lilla pgm-snutt som subrutin. Den anropas genom att man, när talet är i display, "skriver" A.

P23a

```
000  76 LBL      014  54  )
001  11  A       015  29 CP
002  53  (       016  22 INV
003  24 CE       017  67  EQ
004  55  ÷       018  11  A
005  01  1       019  53  (
006  00  0       020  48 EXC
007  49 PRD      021  01  01
008  01  01      022  65  ×
009  75  -       023  01  1
010  22 INV      024  00  0
011  59 INT      025  54  )
012  44 SUM      026  92 RTN
013  01  01
```

BLINDOPERATIONER MED ANDRA TANGENTER ÄN CE

Som bekant har CE funktionen att upprepa talet i displayen i en beräkning - antingen från tangentbordet eller i ett program. "3 + =" är ju otillåtet och ger blink medan "3 + CE =" ger 6. Det viktigaste sättet att använda CE blir ju dock i program där man "flyttar in" talet i displayen innanför ett parentestecken.

Nu är det inte bara CE som har den här funktionen. Inte mindre än 20 andra tangenter åstadkommer samma sak, och ytterligare 12 andra kan användas utan att CE samtidigt behöver vara med.

De 20 som kan ersätta CE (utan att fördenskull förlora sina normala funktioner) är CP, CMs, EE, Eng, Deg, Rad, Grad, Adv, Prt, STO nn, SUM nn, Prd nn, Fix n, Op nn, x=t och x≥t (båda med adress), Lbl NN, St flg n, If flg n och Dsz n (båda med adress).

De 12 som kan användas utan CE är ln x, log x, x^2, $\sqrt{x}$, 1/x, sin, cos, tan, Int, |x|, +/- och Exc nn.

Åtminstone teoretiskt finns det möjlighet att spara programsteg på det här sättet. Om CP skulle ha kommit litet senare eller en flagga skall ställas så går det alltså bra att passa på och slopa CE.

LH

ETT ANVÄNDBART KNEP FÖR TI-57

De statistiska funktionerna $\bar{x}$ och σ^2, med inverser, kan vara mycket användbara. De placerar ett värde i X-registret som är beräknat enligt följande:

$\bar{x} \Rightarrow x \leftarrow \frac{R_1}{R_0}$

INV $\bar{x} \Rightarrow x \leftarrow \frac{R_2}{R_0}$

$\sigma^2 \Rightarrow x \leftarrow \frac{R_3}{R_0} - \left(\frac{R_1}{R_0}\right)^2$

INV $\sigma^2 \Rightarrow x \leftarrow \frac{R_4}{R_0} - \left(\frac{R_3}{R_0}\right)^2$

I ex $\bar{x}$ placerar i X-registret R1 delat med R0.

Hur det kan utnyttjas visas i pgmet nedan ur vSnöpll av TI PPC NOTLS (skrivet av Peter van Hoy). Ett annat exempel är bråkräkningspgmet på s 12 i PB 80-3.

Pgmet läser en andragradsekvation på formen $ax^2 + bx + c = 0$. Lösningen är som bekant:

$$x = -\frac{b}{2a} \pm \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}}$$

Programmeraren har skickligt utnyttjat $\bar{x}$ och σ^2 för att göra pgmet kort och snabbt. Jag tror inte att den intresserade läsaren skall ha några svårigheter att sätta sig in i hur pgmet fungerar.

Bruksanvisning: Mata in a, tryck ned R/S. Mata in b, tryck R/S. Mata in c, tryck R/S. Om skärmen blinkar är rötterna komplexa. Då finns realdelen i X-registret och imaginärdelen i T-registret. I annat fall finns den ena roten i X-registret och den andra i T-registret.

Björn Gustavsson, Smedjebacken

Ej önskvärd bieffekt av "op 00"

Det är omöjligt att använda "op 00" om man har ett decimalbråk i displayen som inte får förstöras! "Op 00" verkar nämligen som "Int", vilket man kanske inte tänker på. En helt normal sekvens är ju t ex "op 00 op 01 op 05" vilken ju rensar övriga skrivregister samtidigt som viss kod laddas i "op 01" och skrivs ut. Om man använder "HIR" är det ju inte ovanligt att skrivkoden är i form av decimalbråk - och om man då använder "op 00" innan man laddar den i t ex HIR 05 blir det kanske inte mycket kvar av den!

Höja bråk till närmast högre heltal

Vid beräkning av kvantiteter kan man behöva ge resultatet i form av ett heltal (åtgång av något som bara förekommer i hela enheter). Följande sekvens åstadkommer detta:

- INV Int CP EQ log + 1 Lbl log =

Avrundning av tal till önskat antal decimaler

Det här programmet avrundar ett tal till önskat antal decimaler, och genom att använda negativa värden på antalet decimaler så kan man även avrunda till tiotal, hundratal osv. Jag har lagt in en rutin som beräknar talets exponent, så att man även kan låta programmet avrunda talet till önskat antal siffrors noggrannhet.

Precis som i Torsten Dahlkvists avrundningsrutin (Pgm-biten 80-2) så lagrar man talet man vill avrunda i t-registret och önskat antal decimaler/värdesiffror i x-registret. Sedan trycker man på A eller B (avrundar till decimaler resp. värdesiffror).

```
Lbl B STO 02 x:t STO 01 CP EQ 060 Ixl log STO 00
GE 020 op 30 RCL 01 x:t RCL 00 ( Int - RCL 02 + 1 )
+/- Lbl A ( INV log EE INV EE x x:t STO 01 + . 5
x RCL 01 op 10 ) ( Int + x:t ) RTN
```

Patrik Johansson, Klemensnäs

KURSER TI-58/59

1 Grundkurs i kalkylatorprogrammering, GK

Ur kursinnehållet:

Enkel programmering
Att lösa problem med program
Formler
Slingor
Villkorliga hopp och subrutiner
Magnetkort och biblioteksmodul
Felsökning och rättning av program

Avgift GK 950:-

Tider GK 49 80.12.01-02
GK 6 81.02.02-03
GK 12 81.03.16-17
GK 18 81.04.27-28
GK 23 81.06.01.02

2 Avancerad kurs i kalkylatorprogrammering, AK

Ur kursinnehållet:

Flaggor, t-register, labels
Indirekta hopp och adresser
Lagring av data på magnetkort
Utnyttjande av de inre HIR-minnena
Programkonstruktion, -analys och dokumentation
Programmeringsfinesser och tips

Avgift AK 1 100:-

Tider AK 6 81.02.04-05
AK 12 81.03.18-19
AK 23 81.06.03-04

ANMÄLAN 08-753 31 40, LEXICON AB, Box 136, 182 12 DANDERYD

LEXICON utbildning på småtatorer

KOMMENTARER TILL :

TI PROGRAMMERBARA 58/59

Handbok I programmering

av GÖSTA BLUME

Det finns i Handboken en del oklarheter, ofullständigheter och felaktigheter som kan förorsaka svårigheter för den som vill sätta sig in i programmeringens mysterier. I den här artikeln skall vi försöka reda ut begreppen i en del fall och understryka vikten av ett par anvisningar som har kommit lite i skymundan. Hittar du andra exempel på formuleringar som behöver förtydligas för att bli begripliga, så låt höra av dig, vare sig du vet hur texten skall tolkas eller behöver hjälp med tolkningen.

Sida IV-47. Längst ner på sidan finns ett observandum (saknas i den amerikanska upplagan) som betonar skillnaden mellan A och GTO A (och motsvarande för andra användardefinierbara labels). Brott mot den regeln har troligen orsakat fler timmar av debuggningarbete än något annat programmeringsfel.

I många fall går det att spara ett pgm-steg genom att utelämna GTO vid programmering av ett ovillkorligt hopp, men eftersom det är svårt att veta exakt när så kan ske bör man undvika att göra det. Ett fall, där sparsamheten med nästan 100-%ig säkerhet kommer att medföra problem, är när pgm-körningen slutar i en felaktigt anropad rutin.

Sida IV-51. På tal om SBR-anropsreg och faran av att en returadress glöms kvar i det reg står det:

"Om nu inte kalkylatorn stängs av, RST eller CP trycks kan problem uppträda vid nästa exekvering"

Påståndet är endast delvis riktigt. Att SBR-reg nollställs när räknaren stängs av är väl rätt självklart. En viktig och riktig upplysning (som f ö förekommer på flera ställen, t ex i V-44) är att RST återställer SBR-reg. Det borde kanske ha tillagts, att detta sker vare sig RST utföres från tangentbordet eller genom en RST-instruktion i ett pgm.

Med CP förhåller det sig däremot så här: förekommer CP i ett pgm, sker endast nollställning av T-reg; tas däremot CP från tangentbordet, sker nollställning av hela pgm-minnet och samtidigt nollställs SBR-reg (se V-3 och V43). CP (eller avstängning av kalkylatorn) bör därför bara användas som förberedelse för inmatning av nytt pgm.för enbart nollställning av returreg använd RST och ingenting annat.

Sida IV-52:

"Observera, att om Pgm mm följs av någon annan instruktion än SBR eller en användardefinierad label, betraktas sekvensen som ogiltig ..."

Samma uttalande återfinns också längst ner på V-61.

Påståendet är felaktigt! Redan på sida V-62 nämns i översikten under rubriken Subrutinistruktioner att även anropet Pgm mm R/S kan förekomma. En pgm-bit i PB78-2 s22 visar hur detta anrop används vid samarbete mellan det egna pgm och ML-02. Det torde vara svårt att ge en generell regel för när denna anropsform skall användas. Ett studium av det använda modulpgm är nog enda sättet att komma underfund med saken.

Sida IV-62:

"Alla programflaggor kan nollställas samtidigt med RST eller CP."

Detta fall är fullständigt analogt med nollställningen av SBR-returregistret: använd RST från tangentbordet eller i pgm såvida du inte tänker mata in ett nytt pgm.

Sida IV-99 behandlar i första stycket skillnaden mellan hopp till en Lbl-adress och till en numerisk adress så:

"Om en absolut adress används, flyttas pgm-behandlingen omedelbart till denna adress. Används emellertid en labeladress måste denna label sökas upp. Labelsökningen börjar alltid vid steg 000 och fortsätter steg för steg genom pgm-minnet."

Det är naturligt att av denna formulering dra den slutsatsen, att hopp till en numerisk adress går direkt från utgångspunkten framåt eller bakåt till adressposition en. Så sker dock inte: i båda fallen går pgm-visaren först tillbaka till 000 och därefter framåt till adresspositionen. Den verkliga skillnaden består däri, att framstegningen till adresspositionen sker mycket snabbare vid numerisk adressering än vid sökning av en Lbl.

En detaljerad redogörelse för tillvägagångssättet och tidsåtgången vid hopp finns i PB79-2 sll.

Sida V-4. Att INV, tryckt av misstag, kan annuleras genom en ny tryckning av INV är väl känt av alla och är en bekväm anordning. Men utnyttja den inte vid inknappning av att pgm! Effekten blir visserligen den avsedda, men eftersom båda tryckningarna går in i pgm-minnet så belägger du två pgm-pos till ingen nytta. — 2.nd kan däremot annuleras genom ny tryckning av 2.nd utan förlust av utrymme i pgm-minnet.

Sida V-8. Under rubriken "Fast Decimalplacering" står det bl a:

"... utom vid DMS-GG (sexagesimala) omvandlingar som enbart använder talet på sifferindikatorn."

Ehuru rätt tänkt är det påståendet faktiskt felaktigt formulerat. Om du i Fix1 mode slår in talet 12,15 och trycker D.MS, så skulle du enligt regeln ovan förvandla $12^{0}15'$ (det är ju vad som står i sifferindikatorn) till decimaldelade grader och få 12,25 i x-registret. Men tittar du (med INV Fix) efter vad som verkligen står där, så finner du att där står 12,333..., dvs den decimala motsvarigheten till $12^{0}20'$. Det var alltså inte det på sifferindikatorn visade talet som omvandlades, utan det tal som, efter avrundning till Fix1, skulle ha stått där, men som aldrig kom upp.

Den här saken är lite lömsk eftersom du, såländge du står kvar i Fix1 aldrig får tillfälle att se att du har kommit fel: både 12,333.. och 12,25 visas ju som 12,3!

Sida V-20. Definitionen av Int är ordagrant riktig sådan den står där: vad som är kvar sedan decimaldelen huggits av.

Int ger således inte alltid vad vi vanligen kallar "heltalsdelen" av operanden, alltså det största heltal som är lika med eller mindre än operanden, och motsvarar sålunda inte i alla avseenden vårt Int eller, som det numera skall beteckna s, ent(x).

Int 2,4 = ent(2,4) = 2 men
Int(-2,4) = -2 medan ent(-2,4) = -3

-o0o-

Mitt på samma sida står ett stycke som börjar

"Lägg märke till att heltalsfunktionen opererar på talet i sifferindikatorns register, inte på det visade talet."

I många sammanhang, särskilt kanske inom heltalsteorin, förekommer det fall då man har att söka heltalsdelen av en rest som har framkommit som resultat av en lång serie beräkningar och denna rest kan vara ett heltal. Man kan då exvis komma att stå inför följande situation:

RCL1 y^{x} 3 = Int ...

som ju inte ser så märkvärdig ut. Men gör du beräkningen med t ex 3 eller 4 i R1 , så får du fel resultat. Skriver du istället

RCL1 y^{x} 3 = EE INV EE Int ...

så blir det rätt. Går man inte ända från början av pgmmeringen in för att genom förutseende hindra fel av det här slaget från att uppkomma,kan man, som otaliga pgmmerare fått erfara, få betala sin försummelse med timmar av felsökningsarbete.

Sida V-22 och V-29 beskriver användningen av Op 17 för att ändra reg-fördelningen. Ett intressant faktum som inte omnämns är, att Op 17 även kan stå i ett pgm för att under körning ändra fördelningen, ett förfarande som ofta är att rekommendera (se t ex PB78-2 s 22).

Sida V-37 har, närmast ovanför fig, den underliga satsen

"Kvadratroten av kvadraterna på de lodräta differenserna minimeras."

Det är inte svårt att lista ut att satsen skall börja

"Kvadratroten ur summan av kvadraterna ..."

Sida V-44. Beskrivningen under rubriken "Pause" skulle må väl av att kompletteras med det goda rådet:

"Se också Appendix D-8."

Sida V-54. I samband med diskussionen om ändring av mergade koder kan det vara värt att notera, att om man under inknappning av ett pgm har tryckt t ex STO2 och sedan kommer underfund med att man borde ha tryckt STO#2 därför att det kommer en siffra efteråt, så är det enklaste sättet att reparera fauten att trycka antingen BST SST eller LRN LRN och fortsätta med de efterföljande siffrorna.

Sida V-56 konstaterar bl a att ingen Lbl får användas för att definiera mer än ett pgm-segment. Ett påpekande, att det däremot inte möter något hinder att i det egna pgm använda samma Lbl som förekommer i ett anropat modulpgm, kan vara på sin plats.

Sida V-59. Här varnas för användning av = i SBR. Den varningen är i hög grad befogad. Men om man förstår bakgrunden till den, behöver man inte bli en slav den utan kan själv avgöra när man får och inte får använda =.

Antag att vi har att beräkna ett uttryck av formen a(b+c), där a, b och c själva är mer eller mindre komplicerade uttryck i diverse parametrar. Uttryck av samma form som b antas förekomma på flera ställen i pgm, och det är därför naturligt att göra beräkningen av b i en SBR. Hela räkningen kan då schematiskt se ut så här:

(a) ↠ [B + (c) = R/S
Lbl B (b) Rtn

(parenteserna kring a, b och c vill antyda, att dessa storheter är sammansatta uttryck som måste beräknas vart för sig).

Om nu Lbl B avslutas med = får vi som resultat av hela räkningen ab + c i stället för a(b+c), eftersom = i slutet av B utlöser även den föregående multiplikationen med a. Om däremot beräkningen i Lbl B innesluter i parenteser, blir resultatet rätt.

Om däremot a eller c hade legat i en SBR, hade ett = där inte gjort någon skada.

Men inte ens användningen av parenteser i SBR är helt riskfri. Den saken behandlas i en pgm-bit i PB80-3 s 10.

Sida V-63 behandlar Dsz-operationen applicerad på RØ - R9. Den första frågan blir då, hur den skall slås in. Den allmänna regeln (se IV-15) säger nu, att om en siffra följer närmast efter att reg-nummer 0 - 9, så måste hela numret slås in: RCLØ5 12 STO2. Vid Dsz är det inte så. Om du vill åstadkomma Dsz Ø5 12 STOØ2 och trycker tangenterna Dsz 0 5 1 2 STO och 2 så står det i pgm Dszf 512 STO2 och du får ett hopp till pos 512.

På motsvarande sätt ger Dsz 0 5 8 Rtn inte Dsz Ø5 8 Rtn utan Dszf ØØ5 8 Rtn.

Slutsatsen blir alltså denna: Dsz i RØØ - RØ9 måste skrivas in med utelämnande av första nolla i regnumret, vare sig en siffra följer eller inte.

(Ser man riktigt noga efter, så står det faktiskt i HB "2nd Dsz X" och inte "2nd Dsz Ox", men det är ju lätt att förbise, särskilt som den allmänna regeln, när den formulerades, inte gav plats för något undantag.)

I samband härmed kan det vara värt att påminna om att Dsz-operationen faktiskt kan knytas till vilket som helst reg utom R40 (se P878-2 s22). Dsz 40 ger Dsz Ind (se V-68).

Till sist skall det påpekas, att om ett icke-helt tal står i ett Dsz-reg, så tas talet ned med en enhet i sänder, tills endast decimalerna återstår. Det fordras sedan ett extra varv för att få ner det till noll. Detta faktum kan ibland med stor fördel utnyttjas. Om man inte vill att decimalerna skall ge ett extra varv i loopen, får man ta Int före inläggningen i Dsz-reg.

-o0o-

Samma sida innehåller också ett MÄRK som påminner om att man inte får någon returadress noterad vid villkorliga hopp. Men anvisningen för hur man kan åstadkomma en retur är så kryptisk, att den ger föga eller ingen hjälp. Problemet löses emellertid av pgm-biten "Hopp vid test med återhopp" i PB80-1 s 14.

blm

OP 01-04 och HIR 05-08

Ovanstående instruktioner lagrar parvis tal i HIR-registren 05-08 (OP01/HIR05 - OP02/HIR06 - - OP03/HIR07 - OP04/HIR08).

Men de gör det på olika sätt. Talet abodefghij lagras:

- av OP 04 som 0,00abodefghij i HIR 08
- av HIR08 som abodefghij i HIR 08

Om ett tal består av en heltalsdel och en decimaldel lagras:

- endast decimaldelen i HIR 08 med OP 04
- hela talet i HIR 08 med HIR 08

  Exempel: Talet 163200,421 ger:

  - med OP 04 HIR 08 = 0,0000001632
  - med HIR 08 HIR 08 = 163200,421

  (Se remsa I)

Om talen i HIR 04 - 08 skall printas, bortser maskinen från de tre första siffrorna. Detta kan användas till att ge olika text från samma tal.

Ex.: 413333,11716 ger:
- med OP 04 "UPP"
- med HIR 08 "NED"

(Se remsa II)

Ex.: 916012917,4437 ger:
- med OP 04 "DONE"
- med HIR 08 "NEXT"

(Se remsa III)

P23a

Remsa I

```
000  76 LBL
001  11  A
002  43 RCL
003  00  00
004  99 PRT
005  69 DP
006  04  04
007  82 HIR
008  18  18
009  99 PRT
010  98 ADV
011  98 ADV
012  91 R/S
013  76 LBL
014  12  B
015  43 RCL
016  00  00
017  99 PRT
018  82 HIR
019  08  08
020  82 HIR
021  18  18
022  99 PRT
023  98 ADV
024  98 ADV
025  91 R/S
026  00  0
```

```
163200.421      STO
                  0
163200.421

        0.      GTO
                  A

163200.421
.0000001632

163200.421      GTO
                  B
```

Remsa II

```
000  76 LBL
001  11  A
002  43 RCL
003  00  00
004  69 DP
005  04  04
006  69 DP
007  06  06
008  98 ADV
009  91 R/S
010  76 LBL
011  12  B
012  43 RCL
013  00  00
014  82 HIR
015  08  08
016  69 DP
017  05  05
018  98 ADV
019  91 R/S
020  00  0
```

```
413333.         STO
                  0
413333.
0.11716         SUM
                  0
0.11716
0.11716         RCL
                  0
413333.1172

        0.      GTO
                  A
413333.         UPP

        0.      GTO
                  B
413333.1172     NED
```

Remsa III

```
000  76 LBL
001  11  A
002  43 RCL
003  00  00
004  99 PRT
005  69 DP
006  04  04
007  69 DP
008  06  06
009  98 ADV
010  98 ADV
011  91 R/S
012  76 LBL
013  12  B
014  43 RCL
015  00  00
016  99 PRT
017  82 HIR
018  08  08
019  69 DP
020  06  06
021  98 ADV
022  98 ADV
023  91 R/S
024  00  0
```

```
916012917.      STO
                  0
916012917.
0.4437          SUM
                  0
0.4437
0.4437          RCL
                  0
916012917.4

        0.      GTO
                  A
916012917.4
916012917.      DONE

        0.      GTO
                  B
916012917.4
916012917.4     NEXT
```


AVBETALNING

Detta program kommer från Sven Tengsjö i Ulricehamn. Programmet vänder sig i första hand till en försäljare, som med hjälp av detta kan beräkna kundens avbetalningskostnad, månadsamortering samt den effektiva räntesatsen. Genom att mata in ett högsta och ett lägsta möjligt lånebelopp, ger räknaren flera möjliga värden inom detta intervall. Man kan därigenom, tillsammans med kunden, välja ut ett lämpligt lånebelopp.

Vid ränteberäkningarna används standardmodulens pgm 19. Effektivräntan är beräknad enligt konsumentverkets normer. Programmet är utformat för TI-59 med skrivare. Men med små förändringar kan programmet göras om för att användas utan skrivare. Exempelvis kan man stoppa in en R/S- och/eller en STO-funktion mellan steg:

36 - 37 Lånebelopp
61 - 62 Räntekostnad i kr
86 - 87 Kostnader utöver räntekostnad
99 - 100 Avbetalningskostnad
128 - 129 Amorteringarnas storlek (utom sista)
164 - 165 Den effektiva räntan

Tar man sedan bort alla onödiga textsteg i programmet går det att använda en normalupplelad TI-58. I programmet används dataregister 20-25 samt de register som ingår i standardmodulens pgm 19.

Bruksanvisning

Mata in:

önskad räntekostnad i %/månad	Tryck A
aviseringskostnad i kr/månad	Tryck B
uppläggningskostnad i kr/kontrakt	Tryck C
önskad amorteringstid i månader	Tryck D
önskat avbetalningsbelopp i kr	Tryck E

Vill man ha flera möjliga avbetalningsplaner inom ett visst intervall matar man in högsta tänkbara avbetalningsbelopp på E' och lägsta tänkbara avbetalningsbelopp på E. Svar fås då på lånebelopp i 200 kr-steg upp till 6 000 kr, därefter i 500 kr-steg upp till 12 000 kr och därefter i 1 000 kr-steg.

Exempel

Ett avbetalningslån om 5 000 kr amorteras på 12 månader. Försäljaren kräver 1,75% ränta per månad och vill ha 15 kr i betalning för uppläggningskostnaden av lånet samt 1,50 kr per månad i aviseringskostnad. Om dessa värden matas in enligt bruksanvisning ovan får man fram följande resultat:

```
      1.75      RTA

       1.5      AVIS

        15.     UPPL

        12.      MAN

      5000.     KAP.
     587.       RTA.
      33.       KOST
     620.       SUMMA
     468.       1.AM
      25.        %
SLUT MANADER ? KAP.?
```

KAP. = avbetalningsbelopp
RTA. = kundens räntekostnad i kr
KOST = kostnad utöver räntekostnad
SUMMA = summa avbetalningskostnad
1.AM = amorteringsbeloppens storlek (utom sista)
% = den effektiva räntan

PROGRAMLISTNING

```
000  36 PGM      045  69 DP       090  06  6
001  19  19      046  04  04      091  06  6
002  10 E'       047  25 CLR      092  02  2
003  36 PGM      048  36 PGM      093  01  1
004  19  19      049  19  19      094  03  3
005  18 C'       050  13  C       095  69 DP
006  58 FIX      051  65  ×       096  04  04
007  00  00      052  43 RCL      097  32 XIT
008  02  2       053  25  25      098  69 DP
009  06  6       054  75  -       099  06  06
010  01  1       055  43 RCL      100  85  +
011  03  3       056  24  24      101  43 RCL
012  03  3       057  95  =       102  24  24
013  03  3       058  59 FIX      103  95  =
014  04  4       059  00  00      104  55  ÷
015  00  0       060  69 DP      105  43 RCL
016  69 DP       061  06  06      106  25  25
017  04  04      062  32 XIT      107  95  =
018  43 RCL      063  02  2       108  36 PGM
019  25  25      064  06  6       109  19  19
020  36 PGM      065  03  3       110  13  C
021  19  19      066  02  2       111  58 FIX
022  11  A       067  03  3       112  00  00
023  43 RCL      068  06  6       113  52 EE
024  20  20      069  03  3       114  22 INV
025  36 PGM      070  07  7       115  52 EE
026  19  19      071  69 DP       116  32 XIT
027  12  B       072  04  04      117  02  2
028  43 RCL      073  32 XIT      118  04  4
029  24  24      074  85  +       119  00  0
030  36 PGM      075  53  (       120  01  1
031  19  19      076  43 RCL      121  03  3
032  14  D       077  22  22      122  03  3
033  58 FIX      078  85  +       123  00  0
034  00  00      079  43 RCL      124  69 DP
035  69 DP       080  25  25      125  04  04
036  06  06      081  65  ×       126  32 XIT
037  03  3       082  43 RCL      127  69 DP
038  05  5       083  21  21      128  06  06
039  03  3       084  54  )       129  06  6
040  07  7       085  69 DP       130  01  1
041  01  1       086  06  06      131  69 DP
042  03  3       087  95  =       132  04  04
043  04  4       088  32 XIT      133  25 CLR
044  00  0       089  03  3       134  36 PGM
```

LISTNING FORTSÄTTER SIDA 17

GAME of LIFE

av PER-ERIC HOLMBERG

Detta "spel" kan betecknas som ett "ensmans-luffarschack" av mycket hög klass. Det utvecklades ursprungligen av John Horton Conway vid University of Cambridge, England, och har sedermera varit föremål för många studier. Det presenterades oktober 1970 i Scientific American av Martin Gardner.

Spelet, som i princip endast kräver ett rutat papper och en penna, börjar med, att man sätter ut ett valfritt antal markeringar i lika många rutor. Dessa får vara placerade i helt valfria inbördes lägen. De bildar tillsammans en "population" av N st individer, och denna kallas "generation 0". Ur gen.0 utvecklas sedan gen.1 och ur denna gen.2 o.s.v. Utvecklingen från en generation till nästa styrs av tre mycket enkla regler, som kan beskrivas sålunda:

Varje ruta omges ju av en "ram", bestående av 8 rutor, och det är tillståndet i dessa 9 rutor, som bestämmer, vad som skall ske vid nästa generationskifte.

Regel nr 1 (födelsekriteriet)

Om en tom ruta omges av exakt 3 individer, så föds det en individ i den tomma rutan.

Regel nr 2 (överlevnadskriteriet)

Om en individ omges av 2 eller 3 individer, så överlever den.

Regel nr 3 (dödskriteriet)

Om en individ omges av mindre än 2 individer, så dör den "av ensamhet", och om den omges av mer än 3 individer, så dör den "av trångboddhet".

Det är viktigt att förstå, att alla individer i en generation deltar i skapandet och utvecklingen av nästa generation. Även de som kommer att dö vid nästa generationsskifte. Varje generation måste alltså gås igenom totalt, innan nästa generation kan skapas och sättas ut. Man kan således inte ta det "rad för rad" och stryka de dödsdömda, för de har sin inverkan även "på nästa rad".

En annan viktig sak i själva grundförutsättningen för spelet är tanken, att "spelplanen" skall anses ha "oändlig utsträckning" åt alla fyra hållen. D.v.s. att en population skall ha möjlighet att växa och/eller röra sig hur mycket som helst i alla riktningar utan att begränsas av något "slut på pappret".

Olika populationer

har olika växtkraft och även i övrigt mycket olika egenskaper. Det är oerhört svårt att i förväg gissa sig till, hur en viss population kommer att utvecklas. Och det finns exempel på:

- konstanta populationer, t.ex. ,  m.fl.
- snabbväxande " "  "
- vandrande " "  (glider) "
- periodiska " " xxx =  "
- kanoner som avfyrar glidera utan att förändras
- "eaters" som "äter gliders" " " "

Många tidningsartiklar har skrivits, som beskriver olika populationer och deras egenskaper. Men det verkligt roliga är ju att själv ge sig i kast med studiet av allt detta.

Datorprogram för LIFE

Den ursprungliga avsikten var kanske just, att man skulle sitta och plita detta spel med papper och penna. Och visst kan man göra det, om man har lust. Det är faktiskt riktigt roligt, tycker jag. Men tyvärr visar det sig, att man mycket lätt gör något fel ibland. Och ett enda fel en enda gång får katastrofala följder för den fortsatta utvecklingen, som blir en helt annan än den avsedda.

Använder man papper och penna, så kan ju kravet på obegränsad utsträckning av spelplanen lätt uppfyllas, eftersom man - vid behov - kan skarva till pappret så många gånger man vill.

Med hänsyn till dels risken för fel, dels tidsåtgången vid manuell bearbetning, har detta spel under åren kommit att locka många konstruktörer av program till datorer och programmerbara räknare att försöka göra bearbetningen både säkrare och snabbare än, vad en mänsklig hand och hjärna kan direkt prestera. Och en stor mängd dylika program har sett dagens ljus. En del av dessa är långsamma (upp till timmar per generation), och en del är hypersnabba (ner till bråkdelar av en sekund per generation). Gemensamt för dem alla är dock - såvitt jag har mig bekant - den svagheten, att spelplanen har en begränsad storlek. Detta innsbär alltid, att vissa populationer blir för stora för programmet. De växer ur spelplanen. Och för de flesta programmen får det den tråkiga konsekvensen, att man inte kan lita på, att den senaste generationen är riktig. För när man kommer ut till en kant av spelplanen, så uppfattar programmet det, som ligger utanför, som "tomma rutor", även om där rätteligen skulle ha funnits en individ. Man kan alltså befara, att man i ett tidigare skede av utvecklingen varit ute vid en kant och fått en "kantskada", varefter senare "krympning" har dolt detta förhållande.

(Här vill jag inskjuta, att jag för ABC 80 har gjort ett program, där populationen - om den kommer för nära en kant - studsar bort från den, innan den hinner bli skadad.)

När det gäller program för TI59/PC100 har jag - och många med mig - gjort sådana med varierande framgång. Man finner därvid, att en lämplig storlek på spelplanen är 20 x 20 rutor. Jag skulle här ha kunnat visa ett av mina egna program, men jag måste tyvärr konstatera, att inget av dem kan mäta sig med det, som finns nedan. Det är konstruerat i USA av Bill Skillman och hämtat ur "52-NOTES". Och det har många förtjänster. Det står helt enkelt i en klass för sig.

Bland dess goda egenskaper vill jag nämna följande:

- "tom ruta" markeras med -
- "nyfödd" " " Q
- "överlevande" " " O
- "nyss död" " " "blank"
- generationsnummer skrivs ut
- antalet individer " "

Beteckningarna -, Q, O och blank gör det lätt att se, vilka förändringar, som skett från en generation till nästa.

Tidsåtgången per generation

har jag klockat till exempelvis ca 5,5 min för 7 individer och ca 13,6 min för 67 individer. Det är dock inte bara antalet individer, som bestämmer tidsbehovet för en generation utan också, hur de är fördelade på spelplanen.

Inmatningen av generation 0

görs radvis med tal bestående av 1 för individ och 0 för tom ruta. Upp till max 10 siffror per rad. Om man mater in t.ex. 111, så läggs de tre individerna ungefär mitt på den 20 tecken långa raden. Vill man ha den mera exakt mitt på, så får man inmata 111000. Detsamma gäller i höjdled, där det kan vara lämpligt att först inmata några rader med 0 för att inte få gen.0 för högt upp på spelplanen.

Fullständig bruksanvisning

1. Mata in g,i
   där g = gen.nr (oftast = 0)
   och i = valfri identifikationskod för inmatad grundpopulation
2. Tryck E (display visar 1 = rad 1)
3. Mata in rad 1 enligt beskrivningen ovan
4. Tryck R/S
   Upprepa 3 och 4 för max 10 rader totalt

Om 10 rader inmatas, startar exekveringen automatiskt efter rad 10. Efter starten sker först en utskrift av gen.0, varefter bearbetning och utskrift sker successivt med följande generationer.

Om man vill starta exekveringen efter mindre än 10 rader, så trycker man tangenten A.

Exekveringen pågår sedan ända tills man stöppar med R/S.

Observera

Håll uppsikt på remsan, så att inga "kantskador" uppkommer i någon generation. Om sådan befaras, stoppa med R/S.

Stopp för senare fortsättning

kan göras när som helst, när ett generationsskifte är klart. Man trycker då R/S just när "N Σ" skrivits, gör LRN och anteckna steg-nr i programräknaren, gör LRN igen och spelar in registerinnehållen som block 3 och 4 på ett extrakort.

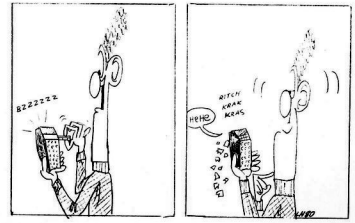
När man sedan vill göra omstart, läser man in alla fyra blocken igen samt gör GTO 222 R/S (undantag: om stegräknare enligt ovan stod på 216-220, så gör man i stället GTO 220 R/S).

Förslag till första körning

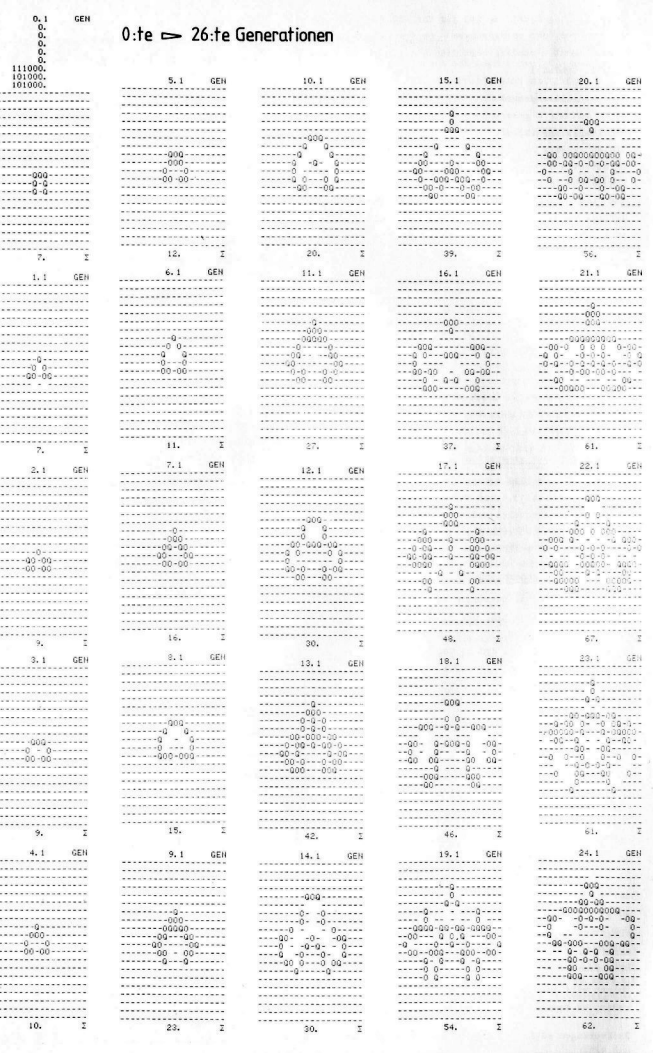
Välj den snabbväxande populationen:		x x x
		x x
genom att:		x x
- inmata (t.ex.)	0,1 (1 = godtyckl.)	och trycka E
- "	0	" " R/S
- "	0	" " "
- "	0	" " "
- "	0	" " "
- "	0	" " "
- "	111000	" " "
- "	101000	" " "
- "	101000	" " "
- trycka A		

Gen. 0 kommer då att hamna något under mitten på spelplanen, men det behövs, för den växer mest uppåt. Spelplanen räcker i detta fall för 26 generationer utan kantskador, men den 27:e generationen blir skadad. De 26 generationerna tar tillsammans ca 4 tim. 15 min. Start samt gen. 0 och gen. 1 visas i exemplet. Observera att man av markeringarna i gen. 1 kan utläsa: 4 "överlevande", 3 "nyfödda" och 3 "nydöda".

Uppdelningen skall vara 479,59
Registerbeläggningen beror på populationsstorleken
Hoppadresserna är numeriska (utom vissa Sbr)
T-registret berörs
Plagga 2 används



0:te ⊂ 26:te Generationen



Forts. Programlistning *AVBETALNING* sid.13

```
135  19  19          180  01  1          225  32  XRT          270  76  LBL          315  03  3
136  10  E'          181  09  9          226  69  DP           271  14  D            316  06  6
137  36  PGM         182  00  0          227  06  06           272  42  STO          317  01  1
138  19  19          183  00  0          228  98  ADV          273  03  3            318  03  3
139  18  C'          184  27  GE         229  91  R/S          274  32  XRT          319  03  3
140  36  PGM         185  34  FX         230  76  LBL          275  03  3            320  01  1
141  19  19          186  01  1          231  12  B            276  00  0            321  01  1
142  12  B           187  00  0          232  42  STO          277  01  1            322  03  3
143  05  5           188  00  0          233  21  21           278  03  3            323  01  1
144  01  1           189  00  0          234  32  XRT          279  03  3            324  06  6
145  00  0           190  44  SUM         235  01  1          280  98  ADV          325  69  DP
146  00  0           191  24  24          236  03  3          281  69  DP           326  02  02
147  65  ×           192  98  ADV         237  04  4          282  04  04           327  01  1
148  01  1           193  91  R/S         238  02  2          283  32  XRT          328  07  7
149  95  =           194  76  LBL         239  02  2          284  69  DP           329  03  3
150  45  YX          195  23  LN2         240  04  4          285  06  06           330  05  5
151  01  1           196  02  2           241  03  3          286  98  ADV          331  00  0
152  02  2           197  00  0           242  06  6          287  91  R/S          332  00  0
153  -   -           198  00  0           243  69  DP         288  76  LBL          333  07  7
154  01  1           199  44  SUM         244  04  04         289  10  E'           334  01  1
155  95  =           200  24  24          245  32  XRT         290  42  STO          335  00  0
156  65  ×           201  98  ADV         246  69  DP         291  23  LN2          336  00  0
157  01  1           202  81  RST         247  06  06         292  99  PRT          337  69  DP
158  00  0           203  76  LBL         248  98  ADV         293  98  ADV          338  03  03
159  00  0           204  34  FX          249  91  R/S         294  91  R/S          339  02  2
160  58  FIX         205  05  5           250  76  LBL         295  76  LBL          340  06  6
161  00  00          206  00  0           251  13  C           296  15  E            341  01  1
162  95  =           207  00  0           252  42  STO         297  42  STO          342  03  3
163  69  DP          208  44  SUM         253  22  22          298  24  24           343  03  3
164  06  06          209  24  24          254  32  XRT         299  25  CLR          344  03  3
165  43  RCL         210  98  ADV         255  04  4           300  61  GTO          345  04  4
166  24  24          211  81  RST         256  01  1           301  76  LBL          346  00  0
167  32  XRT         212  76  LBL         257  03  3           302  15  E            347  07  7
168  43  RCL         213  17  B           258  03  3           303  03  3            348  01  1
169  23  23          214  42  STO         259  03  3           304  06  6            349  69  DP
170  22  INV         215  20  20          260  03  3           305  02  2            350  04  04
171  77  GE          216  32  XRT         261  02  2           306  07  7            351  69  DP
172  35  1/X         217  03  3           262  07  7           307  04  4            352  06  06
173  65  ×           218  05  5           263  69  DP          308  01  1            353  98  ADV
174  09  9           219  02  2           264  04  04          309  03  3            354  25  CLR
175  00  0           220  07  7           265  32  XRT         310  07  7            355  22  INV
176  00  0           221  01  1           266  69  DP          311  00  0            356  58  FIX
177  77  GE          222  03  3           267  06  06          312  00  0            357  91  R/S
178  23  LN2         223  69  DP          268  98  ADV         313  69  DP
179  01  1           224  04  04          269  91  R/S         314  01  01
```

PROGRAMLISTNING *GAME of LIFE*

```
000  76  LBL         060  05  5           120  67  EQ          180  93  .            240  07  07          300  54  54          360  02  2           420  07  07
001  18  C'          061  94  +/-         121  01  01          181  43  RCL          241  29  CP          301  19  D'          361  00  0           421  02  2
002  95  =           062  49  PRD         122  34  34          182  08  08          242  11  A           302  63  DP          362  00  0           422  00  0
003  44  SUM         063  54  54          123  22  INV         183  71  SBR          243  03  03          303  05  05          363  00  0           423  42  STO
004  55  55          064  49  PRD         124  86  STF         184  69  DP          244  42  STO         304  42  STO         364  02  2           424  06  06
005  76  LBL         065  59  59          125  02  02          185  09  09          245  42  STO         305  20  20          365  04  4           425  04  04
006  10  E'          066  61  GTO         126  68  NOP         186  14  D+          246  54  54          306  55  ÷           366  02  2           426  42  STO
007  22  INV         067  00  00          127  43  RCL         187  52  EE          247  67  EQ          307  43  RCL         367  00  0           427  01  01
008  59  INT         068  72  72          128  54  54          188  02  2           248  02  02          308  97  DSZ         368  02  2           428  71  SBR
009  65  ×           069  71  SBR         129  22  INV         189  02  02          249  43  RCL         309  00  00          369  00  0           429  04  04
010  01  1           070  00  00          130  59  INT         190  01  1           250  98  ADV         310  02  02          370  07  7           430  04  04
011  52  EE          071  86  86          131  32  XRT         191  07  7           251  01  1           311  41  41          371  56  56          431  71  SBR
012  01  1           072  03  3           132  93  .           192  03  3           252  02  2           312  00  0           372  02  2           432  04  04
013  00  0           073  32  XRT         133  03  3           193  01  1           253  55  ÷           313  49  PRD         373  00  0           433  53  (-)
014  95  =           074  43  RCL         134  67  EQ          194  69  DP          254  43  RCL         314  03  03          374  69  DP          434  42  STO
015  22  INV         075  54  54          135  01  01          195  04  04          255  54  54          315  42  STO         375  01  01          435  05  05
016  52  EE          076  77  GE          136  54  54          196  43  RCL         256  01  1           316  48  48          376  01  1           436  87  BST
017  92  RTN         077  03  03          137  22  INV         197  32  XRT         257  52  EE          317  00  0           377  00  0           437  03  3
018  76  LBL         078  41  41          138  67  EQF         198  69  DP          258  08  8           318  48  EXC         378  42  STO         438  04  04
019  17  B'          079  17  B'          139  06  06          199  06  06          259  18  C'          319  57  E'          379  07  07          439  04  04
020  43  RCL         080  01  1           140  01  01          200  66  NOP         260  44  SUM         320  42  STO         380  01  1           440  71  SBR
021  56  56          081  52  EE          141  72  72          201  92  RTN         261  24  24          321  49  PRD         381  55  ÷           441  04  04
022  76  LBL         082  04  4           142  93  .           202  25  CLR         262  71  SBR         322  61  GTO         382  95  =           442  46  46
023  14  D           083  61  GTO         143  04  4           203  07  7           263  02  02          323  02  02          383  43  RCL         443  61  GTO
024  22  INV         084  03  03          144  67  EQ          204  03  03          264  04  04          324  00  0           384  04  04          444  02  02
025  52  EE          085  47  47          145  01  01          205  69  DP          265  67  EQ          325  69  DP          385  95  =           445  02  02
026  64  DP+         086  77  GE          146  59  59          206  04  04          266  02  02          326  25  25          386  91  R/S         446  05  5
027  01  01          087  56  56          147  06  6           207  00  0           267  84  84          327  69  DP          387  93  .           447  84  84
028  69  DP          088  42  STO         148  00  0           208  42  STO         268  42  STO         328  26  26          388  99  PRT         448  05  05
029  31  31          089  08  08          149  44  SUM         209  08  08          269  53  (-)         329  53  (-)         389  55  ÷           449  04  4
030  92  RTN         090  05  5           150  08  08          210  42  STO         270  44  SUM         330  02  2           390  01  1           450  43  RCL
031  76  LBL         091  42  STO         151  07  07          211  09  09          271  55  ÷           331  73  RC+         391  55  ÷           451  01  01
032  19  D'          092  09  09          152  01  01          212  48  EXC         272  65  ×           332  05  05          392  05  5           452  17  B'
033  11  A           093  29  CP          153  72  72          213  00  00          273  95  =           333  95  =           393  65  ×           453  14  D
034  22  22          094  43  RCL         154  22  INV         214  69  DP          274  00  0           334  73  RC+         394  01  1           454  08  8
035  73  RC+         095  54  54          155  87  IFF         215  06  06          275  55  ÷           335  06  06          395  95  =           455  14  D+
036  00  00          096  59  INT         156  02  02          216  98  ADV         276  44  SUM         336  75  -           396  72  ST+         456  69  DP
037  42  STD         097  65  ×           157  01  01          217  29  CP          277  55  ÷           337  73  RC+         397  01  01          457  85  ↑
038  93  .           098  93  .           158  62  62          218  01  1           278  07  7           338  07  7           398  32  XRT         458  95  =
039  00  0           099  01  1           159  04  4           219  91  R/S         279  00  0           339  95  =           399  63  EX+         459  05  05
040  42  STO         100  49  PRD         160  07  7           220  69  DP          280  00  0           340  92  RTN         400  01  01          460  04  04
041  52  EE          101  53  53          161  65  ×           221  22  22          281  18  C'          341  76  LBL         401  55  ÷           461  56  56
042  43  RCL         102  95  =           162  01  1           222  71  SBR         282  44  SUM         342  00  00          402  69  DP          462  92  RTN
043  54  54          103  42  STO         163  04  4           223  88  88          283  04  4           343  01  1           403  21  21          463  76  LBL
044  55  ÷           104  54  54          164  95  =           224  08  8           284  04  4           344  01  1           404  72  ST+         464  06  06
045  01  1           105  67  EQ          165  44  SUM         225  12  B           285  42  STO         345  56  EE          405  01  01          465  42  STD
046  50  EE+         106  01  01          166  08  08          226  00  0           286  01  01          346  09  9           406  69  DP          466  54  54
047  05  5           107  72  72          167  01  1           227  19  D'          287  45  ↑           347  65  ×           407  31  21          467  02  2
048  95  =           108  24  RCL         168  44  SUM         228  48  EXC         288  48  EXC         348  43  RCL         408  42  STD         468  42  STD
049  22  INV         109  59  59          169  53  53          229  02  7           289  58  58          349  53  53          409  07  07          469  04  04
050  54  INT         110  59  INT         170  69  DP          230  42  STO         290  69  DP          350  95  =           410  03  03          470  19  D'
051  32  XRT         111  65  ×           171  20  20          231  05  05          291  92  RTN         351  92  RTN         411  80  80          471  01  ST+
052  00  0           112  93  .           172  93  .           232  09  9           292  72  ST+         352  76  LBL         412  76  LBL         472  06  06
053  01  1           113  01  1           173  00  0           233  42  STO         293  05  05          353  15  E           413  11  A           473  69  DP
054  77  GE          114  95  =           174  01  1           234  04  04          294  69  DP          354  42  CMS         414  71  SBR         474  05  05
055  00  00          115  42  STO         175  49  PRD         235  42  STO         295  25  25          355  42  STO         415  04  04          475  92  RTN
056  69  69          116  59  59          176  08  08          236  06  06          296  00  0           356  02  02          416  46  46
057  17  B'          117  22  INV         177  97  DSZ         237  01  1           297  48  EXC         357  71  SBR         417  01  1
058  01  1           118  59  INT         178  09  09          238  01  1           298  55  ÷           358  01  01          418  00  0
059  52  EE          119  22  INV         179  00  00          239  42  STD         299  42  STD         359  88  88          419  42  STD
```

BRÅKFINNARE 2

av GÖSTA BLUME

I PB80-1 s30 visar Dan Altemyr ett pgm för uppsökning av ett allmänt bråk, som med föreskriven noggrannhet approximerar ett givet decimalbråk. Det är ett snyggt pgm, som emellertid lider av svagheten att vara långsamt när man kommer långt ner i det kedjebråk som är kärnan i beräkningsmetoden.

Långsamheten beror nästan helt på att konstruktören gör sig mödan att för varje närmevärde rulla upp hela kedjebråket ända nerifrån. Det behövs inte.

Antag att vi redan har funnit två på varandra följande närmevärden T_{r-1}/N_{r-1} och T_r/N_r och att nästa partialnämnare är n_{r+1}. Då blir nästa närmevärde helt enkelt T_{r+1}/N_{r+1} med $T_{r+1} = n_{r+1} \cdot T_r + T_{r-1}$ och $N_{r+1} = n_{r+1} \cdot N_r + N_{r-1}$.

Om Dan prövar den här kortare metoden på det ovanligt bökiga värdet $(1+\sqrt{5})/2$ skall han finna att den tar totalt ca 1/4 av tiden enligt hans pgm och alla närmevärden tar ungefär samma tid, knappt 4 s.

Vi andra gör klokt i att lägga på minnet den eleganta sekvensen Exc3 Prd3 Exc4 SUM3 i pos 46 - 53 av Dans pgm, som med a i x-reg, b i R3 och c i R4 ger (ab+c) i R3 och b i R4.

Ett sätt, troligen långt ifrån det bästa, att skriva pgm visas i bifogad listning. Talet slås in på A, maximalt relativt fel i promille på R/S. Felet 0 ‰ ger 13-siffrig noggrannhet. Jag har knyckt Dans snygga rutin och använt den på två ställen i pos 10 - 29.

blm

$(1+\sqrt{5})/2$ =	21.	4181.	46368.
1.618033989	13.	2584.	28657.
	-1.637404739	0.000051394	-0.000002328
1.	34.	6765.	75025.
1.	21.	4181.	46368.
-381.9660113	0.626457976	-0.000015811	0.000000128
2.	55.	10946.	121393.
1.	34.	6765.	75025.
236.0679775	-0.238135848	0.000006039	-0.000000049
3.	89.	17711.	196418.
2.	55.	10946.	121393.
-72.94901688	0.091963613	-0.000002307	0.000000019
5.	144.	28657.	317811.
3.	89.	17711.	196418.
30.05664792	-0.034894607	0.000000881	-0.000000007
8.	233.	987.	514229.
5.	144.	610.	317811.
-11.1456618	0.013329015	-0.000742793	0.000000003
13.	377.	1597.	832040.
8.	233.	987.	514229.
4.305231719	-0.005091165	0.000283722	0.000000001
	610.	2584.	
	377.	1597.	
	0.001944661	-0.000108372	

De närmevärden man successivt får fram med kedjebråksmetoden har flera speciella egenskaper. Den viktigaste är väl att varje värde är det garanterat noggrannaste som kan åstadkommas med (täljare och) nämnare som inte är större än vad bråket ifråga har. Härav följer bl a att bråken inte kan förkortas och att varje bråk är en bättre approximation än något föregående bråk.

Det är däremot ingenting som säger att dessa bråk är de enda som kan användas vid konstruktion av exvis en kuggväxel med given omsättning. Ett typiskt fall är ett utväxlingsförhållande som är en multipel av π. Pgm ger i tur och ordning bråken 3/1 (relativt fel i promille -45), 22/7 (0,4), 333/106 (-0,026) och 355/113 (0,00008) etc. Behöver jag nu en noggrannhet av ca 1 ‰, är jag mer än tillgodosedd om jag väljer kuggantalen 22/7 eller någon multipel därav. Men om jag t ex kräver ett fel av högst 0,2 ‰ så skulle jag tvingas upp i utväxlingen 333/106, vilket dels inte låter så roligt, dels ger en alldeles onödigt stor noggrannhet.

I det stora flertalet fall finns det bråk som ger noggrannheter som ligger mellan dem som kedjebråksmetoden ger. Det pgm som visas här bredvid tar fram även dessa bråk. Jag kan sålunda, utgående från 22/7, som antas ge för stort fel, kugg för kugg, så att säga, söka mig fram till successiva bråk, av vilka vart och ett ger större noggrannhet än det närmast föregående.

Pgm består av två delar: å ena sidan pos 0 - 58 och 113 - 129, å den andra pos 59 - 112 och 130 - 150. Den förra delen söker närmevärden med kedjebråksmetoden och bör alltid användas, eftersom den ger en god överblick över vad som kan åstadkommas och därvid anvisar lämpliga områden för den mycket långsammare steg-för-steg-sökningen efter mellanvärden. Vardera delen kan dock användas oberoende av den andra och man behöver inte knäppa in mer än vad man har användning för.

3.141592654	3.141592654	289.	223.
		92.	71.
		-0.091770575	-0.237963114
3.	1.	99.	72.
1.	1.	31.	23.
-45.07034145	179.	311.	245.
	57.	99.	72.
	-0.395269704	-0.058822191	-0.180485705
22.	22.	1.	1.
7.	7.	1.	1.
0.402499434	201.	333.	267.
	64.	106.	85.
	-0.308013704	-0.026489631	-0.132475164
333.			
106.			
-0.026489631			

A, Närmevärden ur kedjebråk

1. Slå in talet på A. Skärmen visar 0
2. Slå in önskat relativfel i pro mille och tryck R/S. För 13-siffrig noggrannhet tryck endast R/S

För varje närmevärde skrivs täljare, nämnare och relativa felet i ‰.
Exekveringstid: 3,5 à 4 s per värde.

B, Mellanvärden

1. Om A körts, tryck B. Annars slå in talet på B'
2. Slå in önskat relativfel (i ‰) på C
3. Slå in första nämnare som önskas provad på D. Utskrift som i A
4. För fortsatt körning till närmast mindre fel, tryck R/S

Exekveringstid: ca 30 nämnare provas per minut.

(I pgm ingår Dan Altemyrs snygga, ovan nämnda rutin två gånger i pos 10 - 29.)

Närmevärden/Bråkfinnare

```
        RCL2 - Int STO3 = 1/x STO2
10      RCL3 Exc7 Prd7 Exc6 SUM7
20      RCL3 Exc5 Prd5 Exc4 SUM5
30      RCL7 Adv Prt : RCL5 Prt : RCL1 - 1 )
44      * 1000 = Prt
51      /x/ x=t 58 x≥t 00
58      Rtn
59      RCL5 * RCL2 + .5 = Int STO7
71      : RCL5 : RCL2 - 1 ) * 1000 = STO8
80      /x/ INV x≥t 98 Op25 GTO 59
98      x/t RCL7 Adv Prt RCL5 Prt RCL3 Prt Rtn
110     GTO 59
113 Lbl A CMs Prt STO1 STO2
121     1 STO4 STO7 CLR Rtn
128     x/t RST
130 Lbl B RCL1
134 Lbl B'Prt STO2 Rtn
140 Lbl C x/t Rtn
144 Lbl D STO5 GTO 59                150
```

-oOo-

Rubriken "Bråkfinnaren" tycks ha misslett en och annan läsare av PB till att tro att Dan Altemyrs pgm hade till uppgift att "översätta" ett decimalbråk till allmänt bråk. Så är alltså inte fallet, utan det gäller att hitta närmevärden med varierande noggrannhetsgrad i form av allmänna bråk.

Problemet att översätta ett decimalbråk till allmänt bråk är mycket lättare att lösa. Varje avslutat (i motsats till oändligt) decimalbråk kan ju direkt skrivas som ett allmänt bråk med en täljare som är identisk med decimalbråkets sifferföljd och vars nämnare är en 10-potens med exponenten lika med antalet givna decimaler. Därmed är problemet i princip löst, även om man sedan kanske vill försöka förkorta bråket.

Eftersom vi från våra räknare bara kan få ut ett begränsat antal (12 å 13) siffror och alla våra decimalbråk sålunda är avslutade, skulle egentligen ingenting vara att tillägga i saken. Men nu är det ju så, att det finns en typ av oändliga decimalbråk, som vi i vissa fall kan hantera på ett mer effektivt sätt, nämligen periodiska bråk med så korta perioder, att dessa i sin helhet får plats i registren.

Ett periodiskt bråk kan vara renperiodiskt, dvs perioden börjar omedelbart efter decimalkommat (t ex 2,027.. = 75/37) eller icke-renperiodiskt, där så ej är fallet (t ex 2,12027... = 1569/740) och perioden sålunda föregås av ett antal icke-periodiska siffror. I båda fallen löses problemet med följande enkla algoritm (här illustrerad på de nämnda exemplen):

x = 2,027...	x = 2,12027...
10^3 x = 2027,027...	10^2 x = 212,027...
999 x = 2027 - 2 =	10^2+3 x = 212027,027...
= 2025	99900 x = 212027 - 212 =
x = 2025/999 =	= 211815
= 75/37	x = 211815/99900 =
	= 1569/740

Den visade algoritmen ligger i sin helhet i Lbl C, medan förkortningsrutinen upptar pos 0 - 50. Vid programmeringen av algoritmen var det nödvändigt att skära bort alla decimaler efter den första perioden och att göra detta på sådant sätt, att periodens sista siffra inte påverkas genom avrundning. Anledningen till denna komplikation i jämförelse med beräkningsexemplen ovan är att vi där kunde förutsätta en oändlig följd av perioder, medan vi vid arbete med räknaren får vara glada om vi kan få plats i registren med en enda hel och ostörd period.

Bruksanvisning

1. Antalet icke-periodiska decimaler slås in på A
2. Periodlängden på B (eller R/S)
3. Decimalbråket med minst en hel period, vars sista siffra inte får vara avrundad, på C (eller R/S)

Täljaren visas. R/S ger nämnaren. R/S utför divisionen som skall ge decimalbråket. Denna sista tryckning av R/S bör göras, eftersom den återställer tidigare Fix och utlöser den påbörjade divisionen.

Ex. 1,1202703 (sista siffran antas vara höjd)
2→A, 3→B, 1,1202703→C ger 829/740 = 1,1202702

Anm. A' har inte direkt med problemet att göra. Den är en inmatningsrutin om man vill använda förkortningsrutinen i annat sammanhang, varvid tiljaren slås in på A' och nämnaren på R/S. Den kan naturligtvis också användas för att direkt översätta ett avslutat decimalbråk till allmänt bråk, eftersom den operationen, efter vad som redan konstaterats, inte är något annat än en bråkförkortning. Decimalbråket (högst 10 siffror) kan därvid slås in på A' precis som det står, varefter 1 slås in på R/S.

blm

FORTS. SID. 40

SPLINE

av HENRIK och JAN TJERNBERG

Anpassning av ett polynom till ett antal punkter är en vanlig metod vid undersökning av mätdata. En variant på detta är att istället styckvis anpassa ett antal polynom till punkterna, med ett antal skarvningsvillkor.

Spline (eller Ri på svenska) är namnet på den metod som innebär att man mellan varje punktpar lägger tredjegradspolynom och som skarvningsvillkor kräver att första och andra derivatorna för de båda polynomen skall vara lika i skarvningspunkten.

Namnet Spline kommer av att sedan länge har konstruktörer löst problemet med kurvanpassning till mätdata genom att markera sina mätpunkter med stift på ritbrädet och sedan lägga en elastisk linjal (Spline på engelska) genom dessa. Kurvan som linjalen beskriver är med god noggrannhet sammansatt av tredjegradspolynom. En motsvarande funktion kallas kubisk splinefunktion och infördes 1946 av Schoenberg.

Följande program av Henrik Tjernberg beräknar splinefunktionen för upp till 15 punkter.

TEORI

Givet är de m+1 funktionsvärdena $f_0, f_1, \ldots f_m$

för x-värdena $x_0, x_1, \ldots x_m$.

Beteckna tredjegradspolynomen med $S_1(x)$, $S_2(x), \ldots S_m(x)$.

De kan skrivas

$$x_n \le x \le x_{n+1}$$

Sätt $x = x_n + ph_n$ där $h_n = x_{n+1} - x_n$

$$S_{n+1}(x_n+ph_n) = f_n + p(f_{n+1}-f_n) + \\ + h_n p(1-p)\left[(k_n - d_{n+1})(1-p) - (k_{n+1} - d_{n+1})p\right]$$

där $d_n = (f_n - f_{n-1})/h_n$

Parametern k_n betyder derivatan för splinefunktionen i x_n och den måste bestämmas med hjälp av villkoren för första och andra derivatorna i punkterna.

Man får: $k_{n-1} + 4k_n + k_{n+1} = 3(d_n + d_{n+1})$

för n= 1,2,....,m-1

Alltså m-1 ekvationer men m+1 obekanta.

Ytterligare 2 villkor krävs. Här har valts att sätta andra derivatan till noll utanför intervallet $x_0 \le x \le x_m$ (en elastisk linjal är ju rak där!).

Villkoren blir: $2k_0 + k_1 = 3d_1$

$$k_{m-1} + 2k_m = 3d_m$$

Vi får följande ekvationssystem att lösa:

$$\begin{bmatrix} 2/3 & 1/3 & 0 & \cdots & 0 \\ 1/3 & 4/3 & 1/3 & & \vdots \\ 0 & 1/3 & 4/3 & \ddots & \\ & & \ddots & 1/3 & 0 \\ & & 1/3 & 4/3 & 1/3 \\ 0 & \cdots & 0 & 1/3 & 2/3 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \\ \\ k_m \end{bmatrix} = \begin{bmatrix} d_2 \\ d_2+d_3 \\ \\ \\ d_{m-1}+d_m \\ d_m \end{bmatrix}$$

Dvs ett tridiagonalt ekvationssystem.

Eftersom "samma" koefficienter uppträder i matrisen oberoende av mätvärdena, endast storleken på matrisen varierar, så kan man "förbereda" den Gausselemination som behövs. För att få första komponenten på rad två lika med noll, behöver vi multiplicera rad 1 med -0.5 och addera till rad 2. Fortsätter man på samma sätt fås en rad koefficienter som i programmet lagrats i minne 53-68. Enklast fås dessa om man utnyttjar formeln

$R_{mm} = -1/(R_{mm-1} + 4)$

Ex $R_{54} = -1/(R_{53} + 4) = -1/(-0.5 + 4) = -.285\ldots$

där R_{54} alltså är minne 54.

Vid lösningen av det aktuella ekvationssystemet utnyttjar man ett antal av dessa koefficienter (beroende på antalet mätpunkter). Därefter elimineras den övre diagonalen genom att gå baklänges.

När nu ekvationssystemet lösts kan man för varje intervall ställa upp ekvationer för tredjegradsnolvnomen med fvra obekanta. När dessa lösts har man fått uttrycken för splinefunktionen.

Inmatning första gången.

1. Mata in Spline pgm 1 steg 000-384
2. Lagra konstanterna i minne 53-68
   Använd formeln ovan för att få 13-siffrig noggrannhet.
3. Spela in (normaluppdelning) block 1 2 3 på magnetkort.

OBS Provkör inte programmet innan Du spelat in det, eftersom programmet förstörs under exekveringen.

4. Mata in Spline pgm 2 steg 000-138
5. Ändra uppdelning 9 OP 17
6. Spela in pgm 2 (lämpligen på magnetkort 2's andra sida)

Tid

Beräkningen av splinefunktionens koefficienter (pgm 1) tar max 3 min 40 sek. dvs för 15 punkter (då löses en 15X15 tridiagonal matris och 14 stycken ekvationssystem med 4 obekanta).

Beräkningen av funktionsvärde (pgm 2) tar 5 - 10 sek.

Litteratur:

Pohl, Eriksson, Dahlquist: Numeriska metoder
Björck, Dahlquist: Numeriska metoder

STEP	PROCEDURE	ENTER	PRESS	DISPLAY	COMMENTS
1	Läs in pgm 1 (block 1 2 3)				
2	Ange antal mätpunkter	m	A	21.	m ≤ 15
3	Mata in de m x,y-värdena	x y	x⇄t R/S		När m stycken värden givits startar beräkningen direkt. Efter beräkningen visas -1.
4	Läs in pgm 2 (block 1)				
5	Beräkna splinefunktionsvärde	x	A	f(x)	Kan upprepas för önskat antal x.
6	För ny splinefunktion gå till Sten 1.				
	Anm. En splinefunktion vars koefficienter beräknats kan sparas genom att spela in block 2, 3, 4. (på blanka kort) vid ny beräkning läser man in dessa kort och pgm 2 (uppdelning 9 OP 17)				
	EXEMPEL				
1	Utför sten 1				
2	3 stycken mätvärden	3	A	21.	
3	Mata in värdena	1 3 4	x⇄t 7 R/S x⇄t 5 R/S x⇄t 8 R/S	1. 3. -1.	Efter ca 40 sek.
4	Läs in pgm 2				
5	Beräkna splinefunktionsvärde	2 1 3.5	A A A	5.25 7. 6.125	Om printer är ansluten: 2. X 5.25 Y 1. X 7. Y
	OBS att vid sten 3 måste x-värdena ges i ordning dvs i stigande följd!				3.5 X 6.125 Y

Minnen att lagra

53 -5.-01
54 2.857142857142-01
55 -2.692307692308-01
56 2.680412371133-01
57 -2.675955001105-01
58 2.679496669135-01
59 -2.679492264974-01
60 -2.679491940877-01
61 2.679491926067-01
62 -2.679491924437-01
63 2.679491924432-01
64 -2.679491924312-01
65 2.679491924312-01
66 -2.679491924312-01
67 2.679491924312-01
68 -2.679491924312-01
69 0. 00

Spline pgm 1

000	71	SBR
001	79	*
002	09	9
003	04	DP
004	17	17
005	43	RCL
006	03	03
007	75	-
008	02	02
009	95	=
010	44	SUM
011	00	00
012	44	SUM
013	01	01
014	44	SUM
015	02	02
016	85	+
017	03	3
018	59	÷
019	95	=
020	42	STO
021	85	85
022	69	DP
023	22	22
024	69	DP
025	20	20
026	73	RC+
027	02	02
028	42	STO
029	69	69
030	43	DP
031	32	32
032	73	RC+
033	85	85
034	42	STO
035	54	54
036	03	3
037	06	6
038	42	STO
039	86	86
040	05	5
041	02	2
042	42	STO
043	87	87
044	06	6
045	08	8
046	42	STO
047	88	88
048	08	8
049	04	4
050	42	STO
051	89	89
052	73	RC+
053	00	00
054	85	+
055	09	9
056	20	20
057	02	02
058	00	00
059	75	-
060	42	2
061	65	×
062	53	(
063	73	RC+
064	02	02
065	75	-
066	69	DP
067	20	20
068	73	RC+
069	02	02
070	54	)
071	55	÷
072	53	(
073	73	RC+
074	01	01
075	-	-
076	53	(
077	21	21
078	73	RC+
079	01	01
080	54	)
081	42	STO
082	85	85
083	95	=
084	55	÷
085	42	RCL
086	85	85
087	33	ΣΣ
088	95	=
089	72	ST+
090	86	86
091	73	RC+
092	01	01
093	75	-
094	69	DP
095	30	30
096	73	RC+
097	00	00
098	95	=
099	55	÷
100	43	RCL
101	85	85
102	94	+/-
103	75	-
104	73	RC+
105	86	86
106	65	×
107	01	01
108	73	RC+
109	01	01
110	85	+
111	69	DP
112	31	31
113	73	RC+
114	00	00
115	54	)
116	65	×
117	53	(
118	95	=
119	55	÷
120	02	2
121	95	=
122	72	ST+
123	87	87
124	01	01
125	00	00
126	03	3
127	03	3
128	55	÷
129	95	=
130	86	86
131	59	÷
132	73	RC+
133	01	01
134	22	INV
135	75	-
136	02	2
137	65	×
138	87	87
140	65	×
141	73	RC+
142	01	01
143	95	=
144	72	ST+
145	88	88
146	69	DP
147	32	32
148	73	RC+
149	02	02
150	75	-
151	73	RC+
152	86	86
153	55	÷
154	53	(
155	65	2
156	86	86
157	86	86
158	75	-
159	73	RC+
160	87	87
161	95	=
162	73	RC+
163	01	01
164	33	ΣΣ
165	75	-
166	73	RC+
167	88	88
168	65	×
169	73	RC+
170	01	01
171	95	=
172	72	ST+
173	86	86
174	01	01
175	94	+/-
176	59	÷
177	89	89
178	88	88
179	88	88
180	44	SUM
181	87	87
182	44	SUM
183	86	86
184	44	SUM
185	00	00
186	95	=
187	01	01
188	42	STO
189	03	03
190	00	00
191	59	÷
192	00	00
193	52	)
194	91	R/S
195	76	LBL
196	11	A
197	42	STO
198	04	04
199	42	STO
200	03	03
201	07	7
202	69	DP
203	17	17
204	65	×
205	42	STO
206	01	01
207	02	2
208	01	01
209	42	STO
210	02	02
211	76	LBL
212	12	B
213	91	R/S
214	72	ST+
215	02	02
216	02	02
217	72	ST+
218	03	03
219	01	01
220	69	DP
221	00	00
222	22	INV
223	29	CP
224	03	03
225	12	B
226	71	SBR
227	33	ΣΣ
228	02	2
229	02	2
230	42	STO
231	00	00
232	43	RCL
233	04	04
234	42	STO
235	01	01
236	69	DP
237	33	33
238	02	2
239	02	2
240	42	STO
241	02	02
242	06	6
243	42	STO
244	04	04
245	71	SBR
246	69	DP
247	00	00
248	00	00
249	92	RTN
250	76	LBL
251	78	Σ+
252	91	R/S
253	02	02
254	02	02
255	69	DP
256	32	32
257	52	)
258	02	02
259	95	=
260	55	÷
261	53	(
262	04	4
263	01	01
264	75	-
265	03	3
266	31	31
267	02	2
268	03	03
269	69	DP
270	72	ST+
271	00	00
272	02	2
273	44	SUM
274	01	01
275	44	SUM
276	02	02
277	69	DP
278	20	20
279	97	DSZ
280	03	03
281	70	Σ+
282	71	SBR
283	79	*
284	69	DP
285	33	33
286	73	RC+
287	00	00
288	76	LBL
289	68	NOP
290	69	DP
291	20	20
292	85	+
293	73	RC+
294	00	00
295	95	=
296	63	EX>
297	00	00
298	97	DSZ
299	00	00
300	68	NOP
301	69	DP
302	20	20
303	72	ST+
304	00	00
305	71	SBR
306	79	*
307	05	5
308	03	3
309	42	STO
310	02	02
311	69	DP
312	69	DP
313	73	RC+
314	00	00
315	65	×
316	53	(
317	02	02
318	85	+
319	03	3
320	20	20
321	74	SM+
322	00	00
323	85	+
324	22	22
325	97	DSZ
326	03	03
327	69	DP
328	03	3
329	55	÷
330	53	(
331	02	2
332	85	+
333	69	DP
334	32	32
335	73	RC+
336	02	02
337	95	=
338	54	)
339	00	00
340	43	RCL
341	04	04
342	42	STO
343	03	03
344	69	DP
345	33	33
346	71	SBR
347	60	DEG
348	69	DP
349	00	00
350	55	÷
351	03	3
352	94	+/-
353	95	=
354	64	FIX
355	00	00
356	69	DP
357	00	00
358	73	RC+
359	02	02
360	65	×
361	03	3
362	94	+/-
363	95	=
364	64	FIX
365	00	00
366	69	DP
367	32	32
368	97	DSZ
369	00	00
370	79	*
371	73	RC+
372	00	00
373	42	STO
374	00	00
375	95	=
376	22	INV
377	44	SUM
378	37	37
379	01	01
380	55	÷
381	05	5
382	49	PRD
383	37	37
384	81	RST

Spline pgm 2

000	76	LBL
001	11	A
002	42	STO
003	20	20
004	32	NXT
005	04	4
006	04	4
007	75	-
008	43	RCL
009	03	3
010	07	7
011	42	STO
012	01	01
013	42	RCL
014	04	04
015	42	STO
016	00	00
017	42	STO
018	05	5
019	43	RCL
020	03	03
021	04	4
022	02	02
023	69	DP
024	00	00
025	42	STO
026	03	03
027	69	DP
028	32	32
029	43	RCL
030	06	6
031	06	6
032	69	DP
033	00	00
034	32	NXT
035	54	)
036	95	=
037	42	STO
038	04	04
039	43	RCL
040	01	01
041	21	21
042	95	=
043	42	STO
044	06	6
045	43	RCL
046	06	06
047	69	DP
048	04	4
049	05	5
050	69	DP
051	04	04
052	43	RCL
053	05	05
054	95	=
055	69	DP
056	43	RCL
057	06	6
058	85	+
059	43	RCL
060	21	21
061	95	=
062	06	06
063	06	06
064	98	ADV
065	81	RST
066	69	DP
067	32	NXT
068	69	DP
069	04	4
070	75	-
071	04	4
072	43	RCL
073	69	DP
074	04	04
075	73	RC+
076	00	00
077	95	=
078	65	×
079	01	01
080	95	=
081	44	SUM
082	06	06
083	69	DP
084	20	20
085	43	RCL
086	06	06
087	69	DP
088	00	00
089	69	DP
090	43	RCL
091	06	06
092	01	01
093	69	DP
094	20	20
095	43	RCL
096	06	06
097	69	DP
098	00	00
099	69	DP
100	04	04
101	69	DP
102	32	32
103	20	20
104	43	RCL
105	01	01
106	01	01
107	95	=
108	01	01
109	00	00
110	06	6
111	44	SUM
112	69	DP
113	01	01
114	43	RCL
115	01	01
116	69	DP
117	01	01
118	69	DP
119	00	00
120	69	DP
121	04	04
122	69	DP
123	32	32
124	43	RCL
125	01	01
126	01	01
127	69	DP
128	06	06
129	06	6
130	44	SUM
131	69	DP
132	01	01
133	43	RCL
134	01	01
135	69	DP
136	01	01
137	69	DP
138	00	00

FERMAT's SATS

av GÖSTA BLUME

Den store talteoretikern Pierre de Fermat (1601 – 1665) har bl a formulerat följande sats: om p är ett primtal och a är ett heltal, vilket som helst som inte är en multipel av p, så ger a^{p-1} ett tal som vid division med p ger resten 1. Med kongruensräkningens symbolik skrives satsen vanligen så:

$$a^{p-1} \equiv 1 \pmod{p}$$

Att direkt verifiera satsens riktighet genom beräkning av a^{p-1} och division med p är för stora värden på p praktiskt omöjligt. Om mant ex väljer det relativt stora primtalet p = 6 324 551 (som vi får anledning återkomma till längre fram) och väljer a ungefär hälften så stort, får vi för a^{p-1} ett tal bestående av ca 41 000 000 siffror. Likafullt kan vi med våra räknare klara av beräkningen på mindre än en minut!

För att den följande beskrivningen av beräkningsmetoden skall bli begriplig även för dem som ännu inte har satt sig in i kongruensräkningens mysterier är det nödvändigt att först ge en kort sammanfattning av vad som menas med en kongruens och de viktigaste reglerna som gäller för dem.

Om jag skriver $a \equiv b \pmod{m}$, så utläses det som "a är kongruent med b modulo m" och innebär att a och b ger samma rest vid division med m. Så är t ex

$$8 \equiv 47 \equiv 489 \equiv -5 \pmod{13}$$

och raden kunde ha fortsatt i det oändliga. Bland alla dessa "rester" är det två som intar en viss särställning, nämligen å ena sidan 8, som är den minsta positiva resten, och -5, som är den numeriskt minsta resten, minsta "absolutresten".

Antag nu att vi, modulo m, har $a \equiv b$ och $c \equiv d$. Då gäller att $(a + c) \equiv (b + d)$, $(a - c) \equiv (b - d)$, $ka \equiv kb$ och $ac \equiv bd$. Av den sistnämnda kongruensen följer naturligtvis att $a^n \equiv b^n$.

Om vi nu vill beräkna x ur $a^n \equiv x \pmod{m}$ kan vi börja med a^2 och reducera kvadraten till minsta positiva eller, hellre, minsta absolutrest modulo m. Vi får då ett mindre tal att handskas med än vad a^2 är.

Den erhållna resten kan vi nu multiplicera med a, reducera den nya produkten osv, till dess att vi får ett värde på a^n reduceret modulo m. Vi har därvid aldrig behövt räkna med tal större än den minsta absolutresten multiplicerad med a. Den nämnda resten kan tydligen aldrig bli numeriskt större än (m+1)/2 medan minsta positivresten kan bli (m-1). Vid räkning med Fermats sats finns det inte heller någon anledning att räkna med a-värden större än (m+1)/2.

Låt oss nu återgå till Fermats sats. Om p är ett stort tal så är det uppenbart, att vi inte kan hålla på och multiplicera oss fram steg för steg. Vi måste finna ett sätt som är mycket snabbare. Och det kan vi få, om vi skriver (p-1) i binär form och använder oss av det binära talsystemets egenskaper.

I det binära talet anger sista siffran enheter, nästa siffra har värdet 2, nästa värdet 4 osv: varje gång vi tar ett steg åt vänster fördubblas värdet av en etta, och detta värde skall adderas till vad vi hade förut. Har vi t ex talet 1010010 så motsvarar det decimaltalet 2 + 16 + 64 = 82.

Men i vårt fall är (p-1) en exponent till vilken talet a skall höjas. Fördubbling av exponenten motsvarar emellertid en kvadrering av talet och addition av exponenter multiplikation av potenser. Om för korthets skull kvadrering betecknas med k och multiplikation med m får vi i fallet p-1=82 sekvensen

a – km – k – k – km – k – km

Vi börjar alltså från slutet!

Det återstår nu bara att påminna om hur man lättast översätter ett givet decimaltal till binär form. Vi utför successiva heltalsdivisioner med 2. Blir resten 0 skriver vi en nolla, blir den 1 skriver vi en etta, och får så binärformens siffror i omvänd ordning, dvs så som vi vill ha dem.

Som en illustration till de kanske alltför korta anvisningarna ovan lämnas här nedan en detaljerad plan över beräkningarnas gång för bestämning av resten modulo 83 av talet 18^{82}.

		Kvadrering	Multiplikation
82	0	18	------ (1)
41	1	$18^2 = 324 \equiv -8$	$* 1 = -8 \equiv -8$
20	0	$(-8)^2 = 64 \equiv -19$	------
10	0	$(-19)^2 = 361 \equiv 29$	------
5	1	$29^2 = 841 \equiv 11$	$* (-8) = -88 \equiv -5$
2	0	$11^2 = 121 \equiv 38$	------
1	1	$38^2 = 1444 \equiv 33$	$* (-5) = -165 \equiv 1$
(0)			

-oOo-

Vår flitige vän Björn Gustavsson har gjort att litet snyggt program för TI-58/59, som beräknar rester av heltalspotenser i godtycklig modul med tillämpning av här skildrad metod.

Bruksanvisningen är ytterst enkel:

Slå in a på A och p på B eller R/S. För Fermats sats, dvs exponenten = p-1, tryck R/S en gång till. För godtycklig exponent slå först in denna innan du trycker R/S sista gången. När en körning är avslutad kan du slå in både/alla tre parametrarna med användning enbart av R/S.

Framtagningen av de binära siffrorna sker i steg 34-43, testen för val mellan kvadrering + multiplikation och enbart kvadrering sker i 44-46, multiplikation + reducering i 47-54, kvadrering + reduktion i 55-60, test på om körningen skall avslutas i 63-67 och avslutande multiplikation + reduktion + återställning för ny körning i 68 – 7 . Reduktion till minsta absolutrest ligger i A'.

Det har redan förut påpekats, att varken resten eller a behöver överstiga (m+1)/2 och det bör vara möjligt att härav beräkna hur långt man med säkerhet kan gå utan att räknarens 13-siffriga kapacitet överskrids. Jag har funnit den vara $2\sqrt{10^{13}}$. Det tidigare nämnda talet 6 324 551 är i så fall det största primtal som kan provas med detta pgm. Ingen dålig prestation, eller hur?

Jörn

000	76	LBL	039	59	INT
001	16	A'	040	42	STO
002	75	-	041	02	02
003	53	(	042	95	=
004	24	CE	043	29	CP
005	55	÷	044	67	EQ
006	43	RCL	045	00	00
007	01	01	046	55	55
008	54	)	047	43	RCL
009	52	EE	048	00	00
010	22	INV	049	65	×
011	52	EE	050	43	RCL
012	65	×	051	03	03
013	43	RCL	052	16	A'
014	01	01	053	42	STO
015	95	=	054	03	03
016	92	RTN	055	43	RCL
017	76	LBL	056	00	00
018	11	A	057	33	X²
019	42	STO	058	16	A'
020	00	00	059	42	STO
021	92	RTN	060	00	00
022	76	LBL	061	02	2
023	12	B	062	32	X↑T
024	42	STO	063	43	RCL
025	01	01	064	02	02
026	75	-	065	77	GE
027	01	1	066	00	00
028	42	STO	067	36	36
029	03	03	068	43	RCL
030	54	)	069	00	00
031	92	RTN	070	65	×
032	76	LBL	071	43	RCL
033	13	C	072	03	03
034	58	FIX	073	16	A'
035	00	00	074	22	INV
036	55	÷	075	58	FIX
037	02	2	076	61	GTO
038	75	-	077	00	00
			078	16	16

DJUP-DYKNING I TI-59 (2)

I förra numrets genomgång av den fasta programvaran hade vi ingen förklaring till de underliga stegen 384 - 487. I ett PS nämnde vi emellertid att den nya danska klubbtidningen "Pgm" hade en förklaring till åtskilligt. Det gäller också de här stegen, som om de sammanförs i grupper om åtta och alltså görs om till minnen får följande betydelse:

391-384 2302585092994(012) där de sista tre siffrorna innehåller exponent och tecken, så att alltsammans betyder 2.302585092994, vilket är detsamma som ln 10! I "Pgm" påpekar man att de sista tre siffrorna här inte fungerar på samma sätt som vid vanlig lagring.

En sammanställning ger:

391-384	2.302585092994	ln 10
399-392	0.6931471805600	ln 2
407-400	0.0953101798043	ln 1.1
415-408	0.0099503308532	ln 1.01
423-416	0.0009995003331	ln 1.001
431-424	0.0000999950003	ln 1.0001
439-432	0.0000099999500	ln 1.00001
447-440	0.0000009999995	ln 1.000001
455-448	0.7853981633975	π ÷ 4
463-456	0.0996686524910	INV tan 0.1
471-464	0.0099966666860	" " 0.01
479-472	0.0009999996660	" " 0.001
487-480	0.0000999999990	" " 0.0001
495-488	1.570796326795	π ÷ 2
503-496	3.141592653590	π
511-504	57.29577951308	180

De tre sista raderna innehåller värden som inte fanns med i förra numrets listning. Man kommer åt dem genom att i den återgivna formeln trycka GTO 489 resp. GTO 505 - stegen 488 och 504 som påstås innehålla INV går att någon underlig anledning inte att lista. I "Pgm" anges för övrigt något varirade formler, nämligen "Pgm 24 R/S R/S R/S 0 Op 17 GTO xxx 10 Op 17 R/S P→R LRN" resp. "0 Op 17 GTO xxx 10 Op 17 Pgm 10 SBR 300 R/S Σ+ LRN". Och därmed slut med djupdykningen för den här gången!

TRÖGHETSMOMENT

TI 57 - 58 /59

av INGVAR MAGNUSSON

Vid hållfasthetsberäkningar är ett vanligt problem att beräkna sektioners yttröghetsmoment. Med en programmerbar räknare blir detta mycket lätt, det som tidigare för invecklade sektioner ledde till tidsödande beräkningar. Även en relativt påver räknare som TI-57 kan man ha stor nytta av vid sådana beräkningar.

Introduktion.

Definition av begreppet tröghetsmoment.
Vi lägger i den sektion vi skall behandla ett rätvinkligt axelkors "x/y-axlarna" genom sektionens tyngdpunkt "TP". Vi delar in sektionens yta i ett antal små element och arean för varje sådant delelement multiplicerar vi med dess avstånd i kvadrat till x-axeln och när vi summerar alla sådana delprodukter får vi sektionens tröghetsmoment med avseende på x-axeln.

$$I_x = \int y^2 dA$$

Om man sedan genomför samma summering av delytorna men nu räknar till y-axeln får vi trögthetsmomentet med avseende på denna.

$$I_y = \int x^2 dA$$

Sorten för tröghetsmoment blir ett längdmått upphöjt till 4 ex.vis mm^4.

För symmetriska sektioner som t.ex. en enkel rund eller rektangulär balk, en I-balk eller en U-balk räker det sagda oftast för behandlingen. De två tröghetsmoment vi får fram är i dessa fall de två huvudtröghetsmomenten, av vilka det ena har ett max- och det andra ett min-värde. Hur går det då om vi för vår symmetriska sektion vrider koordinatsystemet en aning som i fig 1? Jo, det tröghetsmoment som hade max-värde blir något mindre och det som hade min-värde blir något större. Summan av de två momenten förblir konstant.

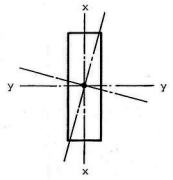


Fig. 1.

Låt oss förutom att se på tröghetsmoment även betrakta begreppet tröghetsprodukt. Vi lägger även här in i den sektion som vi skall behandla ett rätvinkligt axelkors genom dess TP.
Vi delar också denna gång upp sektionen i ett antal delytor men förfar litet annorlunda med dem. I stället för att multiplicera med avståndet i kvadrat till en av axlarna och sedan summera tar vi och multiplicerar varje delelements yta först med avståndet till den ena axeln och sedan med avståndet till den andra. Summan av alla delprodukterna blir tröghetsprodukten.

$$A_{xy} = \int xydA$$

Sorten blir samma som för tröghetsmoment, en längd upphöjd till fyra.

Om vi vrider vårt axelkors en liten vinkel i taget och för varje nytt läge genomför beräkningarna kommer vi att finna att för vissa lägen får produkten ett positivt värde, för andra ett negativt. Vi kommer att hitta två lägen, förskjutna 90° i förhållande till varandra, där produkten blir = 0. Om man tar en symmetrisk sektion, som i fig 1 och låter axelkorset inta olika lägen kommer man att finna att den ursprungliga placeringen med axlarna parallella med sektionens symmetrilinjer ger tröghetsprodukten = 0.

Låt oss nu se på en enkel men osymmetrisk sektion som i fig 2.
Att ta fram tröghetsmomenten med avseende på x- och y-axlarna är enkelt men dessvärre är resultatet rätt trivialt.

Intressantare är då det fall när tröghetsmomenten antar sina extremvärden - då tröghetsprodukten = 0.

Dessa huvudtröghetsmoment är orienterade kring ett axelkors som i hållfasthetstabeller för L-, Z- och andra osymmetriska profiler brukar beteckn as ξ/η.
En vanlig benämning är principalaxlarna.

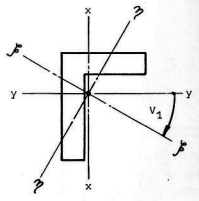


Fig. 2.

Metod.

Arean för en triangel i ett koordinatsystem där triangeln har ett hörn i origo och de två andra hörnen har koordinaterna X_1/Y_1 respektive X_2/Y_2 ges av formeln: $A_i = (Y_1X_2 - Y_2X_1)/2$.

Formeln ger negativ area för triangeln 0 - 1 - 2 och 0 - 2 - 3, se fig. 3, men triangeln 0 - 3 - 4 får positiv area osv .
Arean för polygonen får man genom att summera alla triangelareor med sina tecken, som bildas av respektive sidor i polygonen och origo.
Polygonens area blir positiv när man går medurs och negativ om man går moturs.

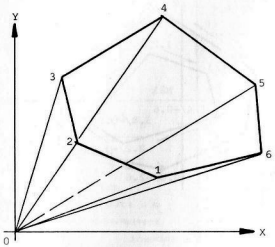


Fig.3.

Ekvimoment system.

En triangel, se fig. 4, har samma tyngdpunkt, statiska moment, tröghetsmoment och tröghetsprodukt som ett system av tre punkter, belägna i triangelsidornas mittpunkter och var och en svarande mot arean $A_i/3$.
Statiska momenten, tröghetsmomenten och tröghetsprodukten för polygonen avseende x- och y-axlarna får man på motsvarande sätt genom att summera storheterna var för sig för respektive triangel.

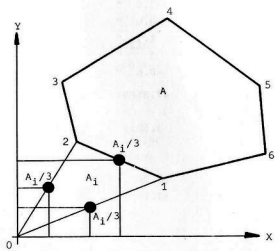


Fig. 4.

Det program för TI-58/59 som presenteras här, tar för en sektion som kan delas upp i en eller flera polygoner, fram tröghetsmomenten och tröghetsprodukten för ett axelkors vars axlar är parallella med primäraxlarna och som går genom tyngdpunkten.
Dessutom beräknas de båda huvudtröghetsmomenten och den vinkel V_1 som axelkorset vridits för att få fram principalaxlarna.
Programmet är uppgjort för att användas tillsammans med skrivare men kan också användas utan skrivare men då får man sätta in ett R/S på rad 140.
Origo för primäraxlarna kan placeras godtyckligt.
Börja varje polygon med att trycka ned "RST" och där efter "X_1R/S Y_1R/S, X_2R/S Y_2R/S, ..., X_nR/S Y_nR/S, X_1R/S Y_1R/S". Gå medurs.
Med hål förfar man på samma sätt, men går då moturs i stället.

Framtagning av resultatet för hela sektionen:

Tryck		Indikatorn/skrivaren visar
E	Sektionens TP-avstånd till x-axeln	X
R/S	" " " y-axeln	Y
R/S	" area	A
R/S	" tröghetsmoment*	IX
R/S	" "	IY
R/S	" tröghetsprodukten*	AXY
R/S	Den vinkelvridning (V_1) som ger huvudtröghetsmomenten. Positivt tecken på V_1 anger vridning moturs; negativt medurs.	V
R/S	Sektionens ena huvudtrögh.moment*	IX
R/S	" andra " "	IY

*) Avser x/y-axlarna parallellförflyttade till TP.

Pgm 1, för TI-57 tar fram statiska momentet och tröghetsmomentet avseende primäraxlarna för en axel åt gången. Dessutom beräknas sektionens area. Programmet får således användas två gånger om storheterna för både x- och y-axlarna skall beräknas.

Beräkning av storheterna avs x-axeln:

Börja varje polygon med att trycka ned "SBR 1 R/S" och där efter "X_1x⇄tY_1R/S, X_2x⇄tY_2R/S, ...X_nx⇄tY_nR/S, X_1x⇄tY_1R/S". Gå medurs (för hål gå moturs).

RCL 0 = tröghetsmomentet
RCL 1 = statiska momentet
RCL 2 = arean

Beräkning avs y-axeln:

Förfar på samma sätt som vid beräkningen av x-axeln, men börja i stället varje polygon med att trycka ned "SBR 2 R/S".

Obs! Radera register 0 - 2 före nya beräkningar.

Pgm 2, för TI-57 tar fram tröghetsprodukten avseende primäraxlarna. Dessutom beräknas sektionens area. Förfar på samma sätt som vid beräkningen avs x-axeln i pgm 1.

RCL 0 = tröghetsprodukten
RCL 2 = arean

Pgm 3, för TI-57 tar fram statiska momenteten avseende respektive primäraxlar i en och samma beräkning. Dessutom beräknas sektionens area. Förfar på samma sätt som vid beräkning avs x-axeln i pgm 1.

RCL 0 = statiska momentet avseende x-axeln
RCL 1 = " " " y-axeln
RCL 2 = arean

Formler.

1) $A_i = (Y_iX_{i+1} - Y_{i+1}X_i)/2$

2) $S_{yi} = (Y_i + Y_{i+1})A_i/3$

3) $S_{xi} = (X_i + X_{i+1})A_i/3$

4) $I_{xi} = ((Y_i + Y_{i+1})^2 - Y_iY_{i+1})A_i/6$

5) $I_{yi} = ((X_i + X_{i+1})^2 - X_iX_{i+1})A_i/6$

6) $A_{xyi} = ((2Y_i + Y_{i+1})X_i + (Y_i + 2Y_{i+1})X_{i+1})A_i/12$

$$A = \sum_{i=1}^{n} A_i \qquad S_y = \sum_{i=1}^{n} S_{yi} \qquad S_x = \sum_{i=1}^{n} S_{xi}$$

$$I_x = \sum_{i=1}^{n} I_{xi} \qquad I_y = \sum_{i=1}^{n} I_{yi} \qquad A_{xy} = \sum_{i=1}^{n} A_{xyi}$$

n = antal hörn i polygonen.

7) $X_o = S_x/A$ sektionens TP, avstånd till x-axeln.

8) $Y_o = S_y/A$ " " TP, avstånd till y-axeln.

9) $I_{\bar{x}} = I_x - Y_o^2A$ " " tröghetsmoment avs x-axeln parallellflyttad till TP.

10) $I_{\bar{y}} = I_y - X_o^2A$ sektionens tröghetsmoment avs y-axeln parallellflyttad till TP.

11) $A_{\bar{x}\bar{y}} = A_{xy} - X_oY_oA$ sektionens tröghetsprodukt avs x/y-axlarna parallellflyttad till TP.

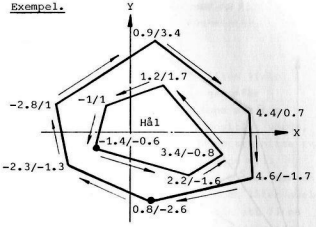
12) $V_1 = (\tan^{-1}(2A_{\bar{x}\bar{y}}/(I_{\bar{y}} - I_{\bar{x}})))/2$ den vinkelvridning av axelkorset i TP för vilken $I_{\bar{x}}$ och $I_{\bar{y}}$ antar sitt största respektive minsta värde. Positivt tecken på V_1 anger vridning moturs; negativt medurs.

$V = V_1$ ger det ena huvudtröghetsmomentet.
$V = V_1 + 90°$ ger det andra huvudtröghetsmomentet.

13) $I_V = I_{\bar{x}} + (I_{\bar{y}} - I_{\bar{x}})\sin^2V - A_{\bar{x}\bar{y}}\sin 2V$ tröghetsmometet efter vridning av axelkorset med vinkeln V.

14) $A_{\bar{x}\bar{y}V} = 1/2(I_{\bar{x}} - I_{\bar{y}})\sin 2V + A_{\bar{x}\bar{y}}\cos 2V$ tröghetsprodukten efter vridning av axelkorset med vinkeln V.

Exempel.



TI-57
Tröghetsmoment
avs x-axeln.
Mata in pgm 1.

```
SBR 1 R/S
 0.8 x⇄t -2.6 R/S
-2.3  "  -1.3  "
-2.8  "   1    "
 0.9  "   3.4  "
 4.4  "   0.7  "
 4.6  "  -1.7  "
 0.8  "  -2.6  "
     Hål
SBR 1 R/S
-1.4 x⇄t -0.6 R/S
 2.2  "  -1.6  "
 3.4  "  -0.8  "
 1.2  "   1.7  "
-1    "   1    "
-1.4  "  -0.6  "
RCL 0   47.858942   Ix
-
RCL 1    3.5215     Sy
x²
÷
RCL 2   20.065      A
=       47.240902   Ix̄
RCL 1
÷
RCL 2
=        0.17550461 Yo
```

Tröghetsmoment
avs y-axeln.
Upprepa tabellen men
tryck ned SBR 2 i stället
för SBR 1.
Obs! Tryck ned INV C.t
före ny beräkning.

```
SBR 2 R/S
.....................
.....................
RCL 0  103.85653    Iy
-
RCL 1  19.771167    Sx
x²
÷
RCL 2  20.065       A
=      84.374889    Iȳ
RCL 1
÷
RCL 2
=       0.98535593  Xo
```

Tröghetsprodukten
avs x/y-axlarna.
Mata in pgm 2 och
förfar på samma sätt som
vid ber. av trögh.moment
avs x-axeln.

```
SBR 1 R/S
.....................
.....................
RCL 0   -2.6198542  Axy
-
RCL 2   20.065      A
x
0.98535593          Xo
x
0.17550461          Yo
=       -6.0897851  Ax̄ȳ
```

Beräkning av V_1 och huvudtröghetsmomenten.

Sätt in $A_{\bar{x}\bar{y}}$, $I_{\bar{y}}$ och $I_{\bar{x}}$ i formel 12.

$V_1 = -9.0794823°$

$V = V_1$

V insatt i formel 13 ger det ena huvudtröghetsmomentet.

$I_{vx} = 46.267713$

$V = V_1 + 90°$ insatt i formel 13 ger det andra huvudtröghetsmomentet.

$I_{vy} = 85.348078$

TI-58/59
Mata in programmet efter listan, för TI-58 uppdelning 2 Op 17

```
RST
 0.8 R/S
-2.6  "
-2.3  "
-1.3  "
-2.8  "
 1    "
 0.9  "
 3.4  "
 4.4  "
 0.7  "
 4.6  "
-1.7  "
 0.8  "
-2.6  "
  Hål
RST
-1.4 R/S
-0.6  "
 2.2  "
-1.6  "
 3.4  "
-0.8  "
 1.2  "
 1.7  "
-1    "
 1    "
-1.4  "
-0.6  "
```

Framtagning av resultatet.

```
E     0.9853559266 Xo
R/S   0.17550461   Yo
R/S  20.065        A
R/S  47.24090218   Ix̄
R/S  84.37488875   Iȳ
R/S -6.089785062  Ax̄ȳ
R/S -9.079482311   V1
R/S  46.26771323   Ivx
R/S  85.3480777    Ivy
```

Registerinnehåll:

```
00  Ix     03  Sx
01  Sy     04  Axy
02  Iy ,   05  A
```

Utskrift från skrivare

```
                0.8
               -2.6

               -2.3
               -1.3

               -2.8
                1.

                0.9
                3.4

                4.4
                0.7

                4.6
               -1.7

                0.8
               -2.6

               -1.4
               -0.6

                2.2
               -1.6

                3.4
               -0.8

                1.2
                1.7

               -1.
                1.

               -1.4
               -0.6

 ,9853559266          X
 0.17550461           Y
  20.065              A
 47.24090218          IX
 84.37488875          IY
-6.089785062          AXY

-9.079482311          V
 46.26771323          IX
 85.3480777           IY

 Registerlistning
 47.85894167          00
  3.5215              01
 103.856525           02
 19.77116667          03
-2.619854167          04
  20.065              05
```

Pgm 1.
TI-57 program.
A, S_y, I_x S_x, I_y

Tangent	Rad	Kod
LBL 1	00	86 1
CLR	01	15
C.t	02	19
R/S	03	81
SBR 3	04	61 3
RST	05	71
LBL 3	06	86 3
+	07	75
STO 4	08	32 4
Exc 5	09	38 5
Prd 4	10	39 4
STO 6	11	32 6
=	12	85
x⇄t	13	22
Prd 6	14	39 6
Exc 3	15	38 3
x	16	55
RCL 5	17	33 5
=	18	85
INV SUM 6	19	-34 6
RCL 7	20	33 7
x^2	21	23
INV SUM 4	22	-34 4
3	23	03
INV Prd 7	24	-39 7
6	25	06
INV Prd 4	26	-39 4
2	27	02
INV Prd 6	28	-39 6
RCL 6	29	33 6
SUM 2	30	34 2
Prd 4	31	39 4
Prd 7	32	39 7
RCL 7	33	33 7
SUM 1	34	34 1
RCL 4	35	33 4
INV SUM 0	36	-34 0
LBL 2	37	86 2
CLR	38	15
C.t	39	19
INV SBR	40	-61
+/-	41	84
x⇄t	42	22
GTO 3	43	51 3

PROGRAMLISTNING FORTSÄTTER NÄSTA SIDA

Pgm 2.

TI-57 program.

A, A_{xy}

Tangent	Rad	Kod
LBL 1	00	86 1
CLR	01	15
C.t	02	19
R/S	03	81
STO 4	04	32 4
STO 6	05	32 6
Exc 5	06	38 5
SUM 4	07	34 4
SUM 4	08	34 4
x≠t	09	22
Prd 7	10	39 7
Exc 3	11	38 3
Prd 6	12	39 6
Prd 4	13	39 4
RCL 7	14	33 7
SUM 4	15	34 4
RCL 6	16	33 6
INV SUM 7	17	-34 7
2	18	02
INV Prd 7	19	-39 7
x	20	55
RCL 5	21	33 5
x	22	55
RCL 3	23	33 3
=	24	85
SUM 4	25	34 4
RCL 7	26	33 7
SUM 2	27	34 2
Prd 4	28	39 4
1	29	01
2	30	02
INV Prd 4	31	-39 4
RCL 4	32	33 4
SUM 0	33	34 0
RST	34	71

Pgm 3.

TI-57 program.

A, S_x, S_y

Tangent	Rad	Kod
LBL 1	00	86 1
CLR	01	15
C.t	02	19
R/S	03	81
+	04	75
STO 6	05	32 6
Exc 5	06	38 5
STO 4	07	32 4
=	08	85
x≠t	09	22
Prd 4	10	39 4
Exc 3	11	38 3
Prd 6	12	39 6
Exc 6	13	38 6
INV SUM 4	14	-34 4
RCL 3	15	33 3
SUM 6	16	34 6
6	17	06
INV Prd 4	18	-39 4
RCL 4	19	33 4
Prd 7	20	39 7
Prd 6	21	39 6
x	22	55
3	23	03
=	24	85
SUM 2	25	34 2
RCL 7	26	33 7
SUM 1	27	34 1
RCL 6	28	33 6
SUM 0	29	34 0
RST	30	71

Tröghetsmoment

000	98	ADV
001	42	STD
002	06	06
003	99	PRT
004	91	R/S
005	42	STD
006	08	08
007	99	PRT
008	76	LBL
009	16	A'
010	98	ADV
011	91	R/S
012	42	STD
013	07	07
014	99	PRT
015	42	STD
016	11	11
017	91	R/S
018	42	STD
019	09	09
020	99	PRT
021	42	STD
022	12	12
023	65	×
024	43	RCL
025	06	06
026	44	SUM
027	11	11
028	75	-
029	43	RCL
030	08	08
031	44	SUM
032	12	12
033	65	×
034	43	RCL
035	07	07
036	95	=
037	94	+/-
038	42	STD
039	10	10
040	55	÷
041	02	2
042	95	=
043	44	SUM
044	05	05
045	43	RCL
046	11	11
047	65	×
048	43	RCL
049	10	10
050	55	÷
051	06	6
052	95	=
053	44	SUM
054	03	03
055	43	RCL
056	12	12
057	65	×
058	43	RCL
059	10	10
060	55	÷
061	06	6
062	95	=
063	44	SUM
064	01	01
065	43	RCL
066	12	12
067	42	STD
068	13	13
069	33	X^2
070	75	-
071	43	RCL
072	09	09
073	44	SUM
074	13	13
075	65	×
076	48	EXC
077	08	08
078	44	SUM
079	12	12
080	95	=
081	65	×
082	43	RCL
083	10	10
084	55	÷
085	01	1
086	02	2
087	95	=
088	44	SUM
089	00	00
090	43	RCL
091	11	11
092	33	X^2
093	75	-
094	43	RCL
095	07	07
096	49	PRD
097	13	13
098	65	×
099	48	EXC
100	06	06
101	49	PRD
102	12	12
103	95	=
104	65	×
105	43	RCL
106	10	10
107	55	÷
108	01	1
109	02	2
110	95	=
111	44	SUM
112	02	02
113	43	RCL
114	12	12
115	85	+
116	43	RCL
117	13	13
118	95	=
119	65	×
120	43	RCL
121	10	10
122	55	÷
123	02	2
124	04	4
125	95	=
126	44	SUM
127	04	04
128	25	CLR
129	61	GTD
130	16	A'
131	76	LBL
132	10	E'
133	22	INV
134	68	FIX
135	69	OP
136	04	04
137	32	X:T
138	69	OP
139	06	06
140	68	NOP
141	92	RTN
142	76	LBL
143	19	D'
144	85	+
145	42	STD
146	13	13
147	95	=
148	38	SIN
149	65	×
150	43	RCL
151	10	10
152	94	+/-
153	85	+
154	43	RCL
155	13	13
156	38	SIN
157	33	X^2
158	65	×
159	43	RCL
160	14	14
161	85	+
162	43	RCL
163	11	11
164	95	=
165	92	RTN
166	76	LBL
167	15	E
168	98	ADV
169	43	RCL
170	03	03
171	55	÷
172	43	RCL
173	05	05
174	95	=
175	32	X:T
176	04	4
177	04	4
178	10	E'
179	43	RCL
180	01	01
181	55	÷
182	43	RCL
183	05	05
184	95	=
185	32	X:T
186	04	4
187	05	5
188	10	E'
189	43	RCL
190	05	05
191	32	X:T
192	01	1
193	03	3
194	10	E'
195	43	RCL
196	00	00
197	75	-
198	43	RCL
199	01	01
200	33	X^2
201	55	÷
202	43	RCL
203	05	05
204	95	=
205	42	STD
206	11	11
207	32	X:T
208	02	2
209	04	4
210	04	4
211	04	4
212	42	STD
213	07	07
214	10	E'
215	43	RCL
216	02	02
217	75	-
218	43	RCL
219	03	03
220	33	X^2
221	55	÷
222	43	RCL
223	05	05
224	95	=
225	42	STD
226	12	12
227	32	X:T
228	02	2
229	04	4
230	04	4
231	05	5
232	42	STD
233	09	09
234	10	E'
235	43	RCL
236	04	04
237	75	-
238	43	RCL
239	01	01
240	65	×
241	43	RCL
242	03	03
243	55	÷
244	43	RCL
245	05	05
246	95	=
247	42	STD
248	10	10
249	32	X:T
250	01	1
251	03	3
252	04	4
253	04	4
254	04	4
255	05	5
256	10	E'
257	98	ADV
258	43	RCL
259	12	12
260	42	STD
261	13	13
262	75	-
263	43	RCL
264	11	11
265	44	SUM
266	13	13
267	95	=
268	42	STD
269	14	14
270	35	1/X
271	65	×
272	43	RCL
273	10	10
274	65	×
275	02	2
276	65	×
277	53	(
278	43	RCL
279	10	10
280	65	×
281	03	3
282	22	INV
283	28	LOG
284	55	÷
285	43	RCL
286	13	13
287	54	)
288	50	I×I
289	59	INT
290	69	OP
291	10	10
292	95	=
293	24	CE
294	60	DEG
295	22	INV
296	30	TAN
297	55	÷
298	02	2
299	95	=
300	32	X:T
301	04	4
302	02	2
303	10	E'
304	19	D'
305	32	X:T
306	43	RCL
307	07	07
308	10	E'
309	43	RCL
310	13	13
311	85	+
312	09	9
313	00	0
314	19	D'
315	32	X:T
316	43	RCL
317	09	09
318	10	E'
319	91	R/S

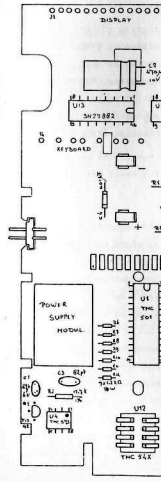
TI 58

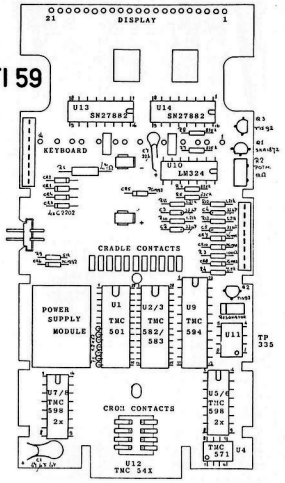
TI 58C

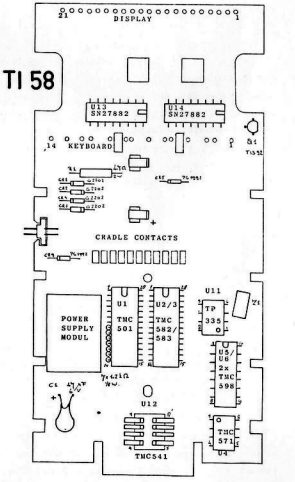
TI 59

Dessa ritningar över komponentplacering i 58,58C och-59 har vi erhållit från 'Texas'. Håll till godo.

KRETS	FUNKTION	
U1	Logik-krets	ALU
U2	Förprogrammerad krets	DSCROM
U3	-"-	-"-
U4	-"-	BROM
U5	Minnen / Programsteg	RAM
U6	-"-	-"-
U7	-"-	-"-
U8	-"-	-"-
U9	Omvandlarkrets för kortläsning	INTERFACE
U10	Förstärkare -"- -"-	OP-AMP
U11	Svängningskrets	Clock
U12	Modul	CROM
U13	Drivkrets för Display	DRIVER
U14	-"- -"- -"-	-"-







vackrare vardagsvara

av GÖSTA BLUME

Enligt Handboken har TI-58 60 och TI-59 120 register som vi kan använda oss av. I själva verket finns det ytterligare några, som vi emellertid inte får veta något om. AOS innebär som bekant, att vi kan programmera formler så som de står skrivna utan att bekymra oss om att tänka på vilka operationer som skall utföras först och vilka som skall komma senare. Detta innebär att räknaren ofta måste hålla inne med en eller flera operationer som inte skall utföras förrän långt senare. Dessa operationer och motsvarande operander (alltså de tal på vilka de skall utföras) måste uppenbarligen lagras någonstans tills de skall användas. Denna lagring sker i operationsstackens register, de s k HIR-registren (Hierarchical Internal Registers). De är till antalet 8 och används, förutom för den nämnda lagringen, även för en del andra saker som programmeraren inte har någon direkt kontroll över (exvis P/R, D.MS, statistikfunktionerna Op 11 - 15, skrivkoder).

Vetskapen att dessa register existerar har satt fantasin i rörelse hos en del individer och har resulterat i en forskning, bakom räknarnes kulisser, så att säga, i det vällovliga syftet att utröna, om det är möjligt att använda dessa reg även för ändamål som vi själva kan bestämma. Delresultat av den forskningen har från tid till annan, vartefter som de framkommit, publicerats här i PB, men något inbördes sammanhang mellan dessa fragment har det varit svårt för oss vanliga dödliga att upptäcka. Nu har man emellertid kommit så långt, att det är tid att försöka pussla ihop bitarna och ge en sammanfattning av de rön, som kan vara användbara för oss alla. Den här uppsatsen är avsedd att vara ett första försök i den riktningen.

-oOo-

Alla 8 stackregistren kan användas för samma operationer som de "vanliga" dataregistren, alltså för STO, RCL, SUM, Prd, INV SUM och INV Prd och de är åtkomliga både från tangentbordet och genom instruktioner i pgm.

Men medan de vanliga dataregistren inkallas genom att man först anger, vilken operation man vill ha utförd och därefter numret på det register som skall ta hand om uppgiften, är det ifråga om HIR-registren snarast tvärtom.

Först anropas gruppen av HIR-register. Det sker med kodnumret 82 (sammansmält = merged). Eftersom det inte finns någon tangent som har påskriften HIR och som direkt ger koden måste vi åstadkomma denna kod på annat sätt. Den kanske enklaste metoden att göra detta vid programmering är att skriva STO 82, som ju automatiskt ger den sammansmälta koden 82, och därefter (med BST och Del) ta bort STO. Detta går, märkligt nog, att göra även på TI-58, trots att inget register med det numret finns där. Vid användning av HIR-registren från tangentbordet finns det bättre sätt att åstadkomma koden och det återkommer vi till längre fram. Har man PC-100 ansluten, skrivs koden 82 ut, men dessutom motsvarande operationsbeteckning som är just HIR.

Sedan vi genom HIR har nått kontakt med operationsstacken har vi att ange dels vilket register vi vill nå, dels vilken operation vi önskar få utförd. Båda dessa saker klaras med en enda 2-siffrig kod, vars första siffra anger operationen och sista siffra anger numret på registret. Registren är numrerade i den ordning de tas i anspråk för AOS-funktionen, från 1 till 8. Operationskoden, alltså den första siffran, framgår av nedanstående tabell

0	betyder lagring ("STO")
1	" utläsning ("RCL")
(2)	(okänt, ännu så länge!)
3	betyder summering ("SUM")
4	" multiplikation ("Prd")
5	" subtraktion ("INV SUM")
6, 7, 8 och 9	" division ("INV Prd")

En fullständig instruktion kan alltså se ut så här: " 8 HIR 35" som innebär att 8 skall summeras in i HIR5.

Vad som hittills sagts om HIR-registren och deras potentiella möjligheter är i stort sett generellt giltigt, oberoende av för vilket ändamål de används. Ifråga om vissa detaljer finns det emellertid större eller mindre skillnader, varför den fortsatta framställningen lämpligen delas upp på användningsområden, inte minst för att ge varje läsare tillfälle att studera det som intresserar henne/honom utan att behöva förvillas av de komplikationer som kan möta vid användning i andra sammanhang. Vi har tre väsentligt skilda användningar för HIR-registren, nämligen som komplement till de vanliga dataregistren i program, motsvarande användning från tangentbordet samt användning i samband med skrivkoder för PC-100. Vi kommer att behandla dem i denne ordning.

A. HIR-registrens användning som dataregister i program

Alla, åtminstone de som arbetar med TI-58, har vi väl åtminstone någon gång varit utsatta för att vi finner, att vi skulle ha behövt ännu ett dataregister men inte har råd att skaffa oss ett sådant genom en omfördelning (re-partitioning) av registren, därför att det skulle ta bort för många (80) steg av pgm-minnet. I denna situation skall vi finna, att HIR-registren är räddningen ur en eljest penibel situation.

Vi konstaterade förut, att HIR-reg har samma möjligheter som de vanliga dataregistren. Här följer ett exempel på den saken. Till vänster ser vi en serie operationer utförda på vanligt sätt med registren R5 - R7 och till höger exakt samma beräkning utförd med enbart registren HIR5 - HIR7. (Att HIR-pgm därvid visar sig bli två steg kortare är knappast någon nackdel.) —(Om någon undrar över vad pgm gör, kan jag tala om, att det är en lite tillkrånglad version av den gamla salongsmagi-leken "Tänk på ett tal (x i R5 och R6), fördubbla det, lägg till e (i R7), ta hälften av summen, dra ifrån det tal du först tänkte på; har du räknat rätt så har du (e/2) kvar." för att köra pgm slår du in x på A och a på R/S.

Fig. 1

000	76	LBL	000	76	LBL
001	11	A	001	11	A
002	42	STO	002	82	HIR
003	05	05	003	05	05
004	42	STO	004	82	HIR
005	06	06	005	06	06
006	91	R/S	006	91	R/S
007	42	STO	007	82	HIR
008	07	07	008	07	07
009	02	2	009	02	2
010	49	PRD	010	82	HIR
011	05	05	011	45	45
012	43	RCL	012	82	HIR
013	07	07	013	17	17
014	44	SUM	014	82	HIR
015	05	05	015	35	35
016	02	2	016	02	2
017	22	INV	017	82	HIR
018	49	PRD	018	65	65
019	05	05	019	82	HIR
020	43	RCL	020	16	16
021	06	06	021	82	HIR
022	22	INV	022	55	55
023	44	SUM	023	82	HIR
024	05	05	024	15	15
025	43	RCL	025	91	R/S
026	05	05			
027	91	R/S			

Vid inslagningen av HIR-operationer i pgm är följande att observera (det mesta är redan sagt men upprepas här som en repetition):

operationsstacken anropas med HIR, som har den sammansmälta koden 82. Den åstadkoms enklast genom att man slår in STO 82 och deleter STO. - När det gäller ett pgm som är så späckat med HIR som det visade, är det troligen lättare att slå in hela pgm, inklusive alla STO och därefter, samtidigt med kontrollen av inslagningen, tar bort STO överallt;

operationskoden enligt tabellen ovan, följd av HIR-registrets nummer, slås in som en enda 2-siffrig kod. Denna kod kan oftast fås genom tryckning av en enda tangent enligt översikten på sidan W-50 i Handboken, som visar att de enda koder som inte kan fås på det sättet är 21, 31, 41 och 51 samt de motsvarande 26, 35, 46 och 56;

observera, att koden (0) för STO inte skall slås in. Det efterföljande reg-numret, säg 4, har ju i sig själv koden 04, som alltså är den sammansmälta kod vi skall ha;

tal mindre än 1 måste i alla operationer från HIR 33 till HIR 99, alltså alla operationer som innebär registeraritmetik, slås in i EE-format; 0,13 HIR35 ger sålunda helt felaktigt resultat, medan 1,3 EE 1 +/- HIR 35 ger rätt resultat; 0,13 HIR 5 (alltså STO som inte är någon aritmetisk operation) är också fullt korrekt; man måste alltså se upp med att värden mindre än 1 möjligen kan uppkomma under pgm-körningen;

val av HIR-reg: det är inte likgiltigt vilka reg man använder eftersom, såsom inledningsvis påpekades, HIR-registren används för diverse speciella funktioner och, naturligtvis, för lagring av vilande operationer; på sidan 13 i PC80-3 finns en detaljerad redogörelse för vilka reg som engageras i de olika operationerna. Av denna lista framgår det att vi vid val av reg är klämda mellan å ena sidan stackfunktionerna som belägger registren nerifrån och å andra sidan av vissa specialfunktioner. Här vi inte skrivare ansluten förefaller det som om HIR 3 - 5 i genomsnitt vore de säkraste att välja, men pgm i övrigt kan radikalt ändra detta förhållande. Statistikfunktionerna, Op 11 - 15, tycks nära nog utesluta användningen av HIR för något programmerarens ändamål under den tid de körs;

nollställning av HIR-reg kan göras med rutinen 0 HIR X, där X är registrets nummer. Är PC-100 ansluten, kan alla registren nollas på en gång med CLR D.MS HIR 3 HIR 4 Op 00, och även utan PC-100 kan HIR 1 - HIR 4 nollas med samma rutin utan det sista Op 00. (Dessa rutiner är hämtade ur TI PPC notes N4/5)

B. HIR-registrens användning som dataregister från tangentbordet

Liksom vid programmering kan man, åtminstone teoretiskt, hamna i den situationen, att man vid beräkningar direkt från tangentbordet finner att man inte har tillräckligt med datareg tillgängliga. Man kanske har ett långt pgm i pgm-minnet och flertalet datainmatningen upptagna av data som man måste spara där, samtidigt som man behöver göra en kalkyl vid sidan av pgm. Även i ett sådant, låt vara sällsynt fall kan HIR-reg rädda situationen.

Principerna för HIR-reg handhavande i detta fall följer i stort sett de allmänna linjerna. Men utgångsläget är i några avseenden olikt det som råder när man gör ett pgm och det är i första hand de därav betingade ändrade rutinerna som vi här skall fästa oss vid.

Problemet att framställa operationsstackens kodnummer, 82, kan nu inte lösas på samma sätt som förut beskrivits, utan en helt ny rutin fordras. Den nya rutinen går ut på att ordna så, att koden 82 finns lätt tillgänglig någonstans där vi vid behov kan hämta den, och detta skall kunna ske utan att vi påverkar den pågående räkningen.

Så här går det till. Lägg in kod 82 i position 0 och anropa denna position när du vill ha koden. Skulle du finna att platsen redan är upptagen av en pgm-instruktion, flytta då med INS ner pgm tre steg, slå in STO82 och gå tillbaka och ta bort STO. Tryckning av LÄN tar oss ur pgm och lämnar kvar HIR och en nolla.

Skall vi nu åstadkomma t ex operationen 2 82 04, så slår vi först in 2 och hämtar därefter ner 82 med sekvensen RST SST, där RST ställer pgm-visaren till 0 och SST flyttar fram den till pos 1 så att koden 82 går in på räknaren precis som om vi hade skrivit in den för hand.

Det återstår att slå in operationskoden och HIR-reg nummer. Vi stöter nu på en ny avvikelse från den vanliga rutinen: de båda koderna skall här inte sammansmältas, utan operationskoden slås in som en siffra (även om den är 0 för STO) och registernumret som en annan, separat siffra. Hela instruktionen slås alltså in så här: 2 RST SST 0 4, alltså fem tangenttryckningar. På motsvarande sätt förfar man med andra instruktioner: 0,13 HIR65 slås in som .13 RST SST 6 5 och ger 0,13 INV Prd HIR5. Observera, att vi i detta fall kan slå in tal som är mindre än 1 så som de står.

En sammanfattning (med några kompletteringar) av rutinerna för användning av HIR i detta fall skulle då kunna se ut så här:

operationsstacken anropas som HIR med koden 82. Denna kan inte framställas under pågående räkning utan måste hämtas ur ett litet pgm, som kan vara antingen helt enkelt 82 (sammansmält med hjälp av STO) och placerat i pos 0, eller ett Lbl-försett pgm t ex Lbl GTO 82. -- I intetdera fallet behöver pgm avslutas med Rtn eller R/S eftersom vi bara kommer att ta ett enda steg i det.
Däremot kan ett av följande pgm behöva skyddas med den nolla jag förut rekommenderade. Antag nämligen att pgm börjar med t ex Lbl C. Om jag anropar detta pgm, kan pgm-visaren inte hitta någon Lbl C, därför att kombinationen 82 76 i sig är en fullkomligt rimlig kombination (INV Prd HIR 6) som också noteras såsom sådan och inte som två oberoende instruktioner. Lbl C kommer därför inte att vara registrerad såsom Lbl, och räknaren svarar att det inte finns någon Lbl C;

HIR nås med rutinen "RST SST" resp "GTO GTO SST";

operationskoden enligt tidigare tabell slås in som en siffra för sig - även siffran 0 skall slås in;

registernumret slås in som en separat siffra;

valet av HIR-reg sker med hänsyn till den pågående beräkningens karaktär, så att de använda HIR-reg inte kommer i vägen för vare sig de register som fordras för ADS eller de register som används av speciella funktioner enligt P880-3, sidan 13;

HIR Ind kan utföras genom att önskad HIR-kod (operationsnr + HIR-regnr) läggs in i ett lämpligt datareg och detta registers nummer slås in efter HIR Ind. Men tyvärr går detta bara från tangentbordet, inte i program. Från tangentbordet ger emellertid (med 37 i R25) 4 HIR Ind 25 en summering av 4 till HIR-reg 7 (4 HIR 37). Koden för HIR måste som vanligt tas från ett litet hjälppgm.

C. HIR-registrens användning för lagring av skrivkoder för PC-100

Låt mig börja med en kort sammanfattning av några välkända fakta som jag i fortsättningen upprepade gånger måste åberopa.

En fullständig skrivrad i PC-100 omfattar 20 alfanumeriska tecken fördelade på fyra delar, i HB benämnda "yttersta vänstra fjärdedelen av utskriftsraden" osv. För korthets skull kommer jag att kalla dessa delar för "block", inte att förväxla med de programblock som är aktuella vid skrivning och läsning av magnetkort. Blocken tänks numrerade från vänster till höger 1 till 4. Varje block rymmer alltså 5 tecken, vart och ett representerat av 2 siffror. Inslagningen av textkoderna sker normalt så, att ett blocks koder slås in i x-registret och överförs därifrån till motsvarande skrivregister med en operation, Op 1 för block 1 osv till Op 4 för block 4. Skrivningen utlöses genom Op 5 eller Op 6. Skrivregistren, till vilka koderna överförs, är våra nya vänner HIR 5 - HIR 8, i nummerordning motsvarande block 1 - 4.

Eftersom varje block här ihop med sin Op och sitt HIR-reg, är det för sin funktion fullständigt oberoende av de andra blocken och vi behöver därför bara syssla med ett enda block för att få en fullständig bild av hur helheten fungerar. I flertalet fall kommer jag att ha block 4 i tankarna, inte därför att det är något särskilt med det blocket, utan därför att det vanligen är där som man har användning för ett enstaka, eller ett par, tecken för identifiering av utskrivna resultat via Op 06.

För att skriva bokstäverna ABCDE i block 4 programmerar man t ex

Lbl A 13 14 15 16 17 Op 4 CLR Op 5 Rtn

där CLR bara är till för att nolla x-reg efter fullbordad uppgift. Skriver jag i stället

Lbl A 13 Op 4 CLR Op 5 Rtn

får jag ett A i femte läget. Vill jag ha A i fjärde läget måste jag skriva

Lbl A 13 00 Op 4 CLR Op5 Rtn

och i första läget

Lbl A 13 00 00 00 00 Op 4 CLR Op 5 Rtn
-000-

Om jag, i stället för att gå omvägen över Op 4 vill lägga in skrivkoden direkt i HIR 8, blir rutinen en annan. Försöker jag skriva ABCDE med ett "normalt" pgm

Lbl A 13 14 15 16 17 HIR 8 CLR Op 5 Rtn

får jag Uwg[2], vilket motsvarar koderna 41 51 61 70. Även om det kanske inte syns direkt, så finns det ett mönster i detta beteende: de tre första kodsiffrorna är borttappade (och den överblivna 7:an har kompletterats med en nolla, såsom vanligt är). Detta bortfall kan vi kompensera genom att föregå koden med tre blindsiffror. Vill man då fortfarande ha plats för tio kodsiffror så måste man utnyttja registrens fulla 13-siffriga kapacitet. Det har nu visat sig, att det är mycket små krav på blindsiffrorna och att kombinationen 100 fungerar bra. Därigenom kommer man över de svårigheter som vanligen möter när man har att slå in 13-siffriga tal. Följande rutin brukar rekommenderas:

Lbl A 13 14 15 16 17 + 1 EE 12 = HIR 8 CLR Op 5 Rtn

vilken mycket riktigt ger ABCDE.

I detta fall kommer, som man lätt ser, blindsiffrorna att stå omedelbart framför kodsiffrorna. Så blir däremot inte fallet om man har färre tecken än 5. Pröva t ex med

Lbl A 13 14 + 1 EE 12 = HIR 8 CLR Op 5 Rtn

Bokstäverna AB kommer då på platserna 4-5, därför att det har kommit sex nollor (motsvarande 3 steg) mellan blindsiffrorna och skrivkoderna. Vill du flytta AB till platserna 1-2 så går det lätt genom att byta ut EE 12 mot EE 6 som ju har sex nollor mindre.

Genom att manövrera 10-potenserna kan man placera sina tecken där man vill ha dem. Följande lilla tabell visar vilken 10-potens som placerar sista tecknet i en teckengrupp på vilken plats. Tabellen gäller för 1 - 5 tecken om du i teknen inkluderar ev mellanslag (vart och ett representerat av två nollor).

Sista siffrans plats	1	2	3	4	5
+ 1 EE	4	6	8	10	12

Har man flera olika texter av ungefär samma längd att skriva i samma block (här block 4) kan man rationalisera skrivarbetet med följande rutiner

ett hjälppgm: Lbl A' + 1 EE X = HIR 8 CLR Rtn
huvudpgm: ... (skrivkoder) A'Op 5 ...

Har man ett enda tecken och vill ha det på plats 1, kan man, i stället för + 1 EE 4, använda det ekvivalenta huvudpgm

100 34 HIR 8 Op 5

och får då Q på plats 1 (10034 är ju = 34 + 1 EE 4).

Innan vi går vidare kan det vara på sin plats att konstatera, att genvägen till HIR, med förbigående av Op 4, obestridligen är besvärligare än omvägen över Op 4, och att fråga sig om det kan vara lönt att ta på sig det besväret. Svaret, eller rättare sagt svaren, ty de är två, är:

a) Op 4 fungerar bara om räknaren står i INV Fix (det är vad HB säger) eller i Fix 0, medan HIR-metoden är fullständigt okänslig för den saken och t o m själv kan hantera decimalbråk; med HIR slipper man således bl a att gå över till INV Fix eller Fix 0 varje gång man vill mata in skrivkoder, för att sedan gå tillbaka till den fix som pgm kräver;

b) vill man ändra i texten, t ex övergå från att identifiera ett resultat som X1 till att beteckna nästa resultat med X2 osv, så kan vi låta pgm göra dessa ändringar allteftersom beräkningen fortskrider. Skall ändringar göras via Op 4 krävs ett totalt uthyte av hela skrivkoden.

Vi skall nu, med utnyttjande av en möjlighet som antyddes under a), utöka programmeringsrepertoaren genom införande av decimalbråk i skrivkoden.

Vi behandlar det fallet, att vi vill ha två tecken, X (kod 44) och Y (45).
Skriver vi 100 44 45 HIR 8 eller 44 45 + EE 6 ..
så får vi XY i läge 1-2.
Med 44,45 EE 6 kommer de i läge 2-3.
,4445 EE 6 eller 44 45 00 EE12 ger läge 3-4.
Slutligen ger 44 45 EE 12 läge 4-5.
Alla dessa möjligheter är lätta att kontrollera genom att man tänker sig koderna + blindsiffrorna utskrivna.

Antal tecken	Sista teckenets plats: 1 Kod	+ EE	2 Kod	+ EE	3 Kod	+ EE	4 Kod	+ EE	5 Kod	+ EE
1	100A A00	-- 6	A	6	,A	6	,00A A00	6 12	A	12
2	----------		100A0 AB	+ 6	A,B	6	,AB AB00	6 12	AB	12
3	----------		----------		100A0C AB,C	- 6	A,0C AB00	6 12	,ABC AB0	6 12
4	----------		----------		----------		AB,CD AB00	6 12	A,BCD AB0	6 12
5	----------		----------		----------		----------		AB,CDE AB0	6 12

Några synpunkter på EE-MODULEN och ett eget ROTLOKUSPROGRAM

av: LARS-BERTIL OLSSON, GÖTEBORG

För cirka 1 1/2 år sedan köpte jag en TI-59 och i våras kompletterade jag den med EE-modulen (Electrical Engineering).

Eftersom jag själv inte har använt alla modulprogram utan bara tittat i manualen samt testat litet grann, så nöjer jag mig med en enkel presentation av programmens funktion.

Det program som jag själv utnyttjat mest är EE-09, "Roots of a polynomial". Programmet ger alla rötter, både reella och komplexa , till polynom med grad upp till 21:a ordningen. Ett "måste"-program även för den inte helt renodlade elektroingenjören!

Ett annat program är EE-08 "Signal Detection" som ger sannolikhet för falskt larm i brusig miljö. Programmet är ett kraftprov på numeriska metoder, och det sparar mycket arbete.

Just det villkoret brukar jag sätta främst när jag gör program. Programmet skall verkligen underlätta arbetet och inte bara bestå av en enkel formel (undantag för sortomvandlingar). Ett program av den typen tillverkar man snabbare på nytt än försöker sätta sig in i "hur var det nu det fungerade"-programmet.

Nåväl, tillbaks till modulen.

Där finns också den typen av program som jag inte gillar:

EE 11 "Reactance Chart" som använder formlerna

$$f_0 = \frac{1}{2\pi\sqrt{LC}}\ ;\quad C = \frac{1}{\omega_0^2 \cdot L}\ ;\quad L = \frac{1}{\omega^2 \cdot C}\ ;\quad \left(f_0 = \frac{\omega_0}{2\pi}\right)$$

och som ger den av de tre som är okänd.

EE-07 "Decibels, Nepers, Power, Voltage, Current Ratio Conversions" som gör det rubriken säger: omvandlingar.

EE-13 "Active Lowpass, Highpass and Bandpass Filters" skulle kunna vara arbetsbesparande om filtren var mer generella. Här är man låst till en viss uppbyggnad av återkopplingsnäten runt operationsförstärkaren, vilket gör användbarheten liten.

EE-15 "Convolution" är en faltnings-approximation mellan en insignal och ett filter. Programmet ger utsignalens utseende.

EE-17 "Discrete Fourier Transform" ger också ett antal punkter som svar. Programmet ger Fourier- eller inversa Fourier-transformen till en tids- eller frekvensfunktion.

EE-08 "Phase Locked Loop" dimensionerar en fastlåst slinga för både frekvenshöjande och -sänkande typ.

EE-03 "S⇄Y Parameter Conversions" är komplexa omvandlingar (omvandlar mellan reflexions- och admittanskoefficienter i en fyrpol).

EE-01 "Module Check" har precis som standardmodulen en initiering för linjär regression. Med samma label!
Det är ett bra försök till standardisering, och det fortsätter i programmen EE-04, EE-05 och EE-06. Programmen motsvarar standardmodulens ML-04, ML-05 resp. ML-06. Dessutom finns finessen med intern omvandling från polär till rektangulär form. Men, om standardiseringen hade tillåtits gå lite längre, så skulle förbistringen med att dessa program och funktionen P→R har olika inmatningssätt kunnat undvikas.

EE-10 multiplicerar ihop två polynom (koefficienterna)
EE-12 omvandlar serie- och parallellkopplade impedanser (och tvärtom)
EE-14 ger passiva lågpassfilter. Obs! LC-filter (inte RC)
EE-18 utför Smith-diagramsberäkningar

EE-19 ger frekvensanalys av ett nod-nät med max 5 noder + jord och 9 komponenter eller 4 noder + jord och 14 komponenter eller 3 eller 2 noder och 24 komponenter. Programmet kräver att TI-59 och skrivare används samt att man skriver ett inmatningskort och ett konstruktionskort från listning i handboken.

Det var alla program i modulen (få men långa), och jag skall nu försöka ge lite kött på benen för mitt eget program. Det utför rotlokusberäkningar, som till skillnad från EE-16 verkligen ger rätt svar (peppar, peppar...). (Red.anm.: Enligt uppgift skulle felet i EE-16 bestå i att poler på negativa realaxeln + origo ger fel i avgångsvinkeln.)

Att här dra hela den bakomliggande teorin för rotlokusmetoden är naturligtvis inte praktiskt genomförbart, utan jag nöjer mig med att redovisa använda formler och hur programmet arbetar.

$$\text{Tyngdpunkten} = \frac{\sum \alpha_{z_{n-1}} - \sum \alpha_{z_n}}{P - N}$$

P = antal poler
N = antal nollställen
$S = \alpha + j\omega$ $\quad S_1\ S_3\ S_5\ \ldots$ = poler
$S_2\ S_4\ S_6\ .$ = nollställen

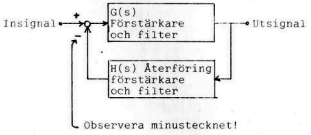
$$\text{Asymptoternas vinklar} = \pm \frac{(2m+1)}{P-N}\ 180^0$$

m = 0, 1, 2, ..., P-N-1.

Avgångsvinklar från pol:

$$\Theta_{z_{n-1}} = \sum_{k_p} \left\{ \frac{k_p}{M_p} \cdot \left[180^\circ + \arg\left(G(s) \cdot H(s) \Big|_{s \le s_{z_{n-1}}} \right) \right] \right\}$$

M_p = polens multiplicitet
k_p = 1, 2, 3, ..., M_p
G(s)·H(s) är den öppna överföringen
(G(s) är förstärkaren, H(s) är återföringen)



Ankomstvinklar till nollställe:

$$\Theta_{z_n} = \sum_{k_N} \left\{ \frac{k_N}{M_N} \left[180^\circ - \arg\left(G(s) \cdot H(s) \Big|_{s \le s_{z_n}} \right) \right] \right\}$$

M_N = nollställets multiplicitet
k_N = 1, 2, ... M_N

Ref: Modern control system theory and application, Stanley M. Shinners, Addison-Wesley Publ.Co. 1972.
Formelsamling i reglerteknik, Birger Qvarnström, Inst.f. reglerteknik, CTH 1970.

För att allt skall fungera, följ arbetsordningen nedan.

Inmatningar

A. 1. Initiera för inmatning av poler. Tryck A'.
2. Mata in realdel (måste alltid finnas). Tryck (siffror) A.
3* Om imaginärdel ≠ 0, mata in den. Tryck (siffror) B.
4** Om multiplicitet ≠ 1, mata in den. Tryck (siffror) C.
5. Fortsätt steg A.2 - A.4 tills alla polers data har matats in.

B. 1. Initiera för inmatning av nollställen. Tryck B'.
2. Samma som A.2.
3. Samma som A.3.
4. Samma som A.4.
5. Fortsätt steg B.2 - B.4 tills alla nollställens data har matats in.

Beräkningar

C. 1. Tryck C' så fås tyngdpunkten.
2. Tryck R/S tills du fått alla vinklar (vinklarna ges i grader).
3. Tryck polnummer (= ordningsnummer vid inmatningen), tryck D så fås lokus avgångsvinkel/avgångsvinklar (i grader) vid den polen. Stanna med R/S.
4. Om du glömt vilken pol som hade vilket ordningsnummer (de automatiskt inmatade komplexkonjugaten får eget nummer), så tryck polnummer följt av D' och du får poldata i ordningen: realdel, imaginärdel, multiplicitet och till sist polnummer.
5. Tryck nollställenummer (=ordningsnummer vid inmatningen), tryck E så fås lokus ankomstvinkel/ankomstvinklar (i grader) vid det nollstället. Stanna med R/S.
6. Samma förutsättningar som C.4. Tryck nollställenummer, tryck E' så får man nollställets data.

Inmatningsautomatik

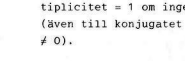
* P g a att komplexa poler och nollställen alltid förekommer som komplexkonjugerade par, så har jag byggt in en automatisk konjugatkomplettering i programmet. Konjugatet lagras som separat pol/nollställe och får ett eget ordningsnummer.
Alltså: Mata bara in ena halvan av paret (t ex polen/nollstället med positiv imaginärdel).

** Programmet lägger automatiskt in multiplicitet = 1 om inget annat matas in (även till konjugatet om imaginärdelen ≠ 0).

Allmänt om arbetsordningen

Del A och del B i inmatningen kan byta plats.

Användningsordningen av tangenterna A, B och C (med sifferinmatning) måste följa någon av pilarna i figuren och skall alltid börja med (siffror) följt av A.



Siffran som visas efter tryck på A, B eller C är antal inmatade poler/nollställen.

Antalet olika poler/nollställen = största ordningsnumret för resp. poler/nollställen.

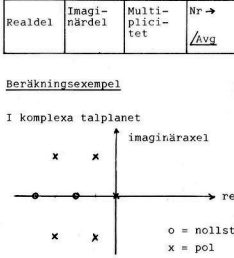
Maximalt antal olika poler = 8 (lika många olika nollställen).

Kortlayout

Initiering Polinmatning	Initiering Nollst. inmatn.	→Tyngdp. R/S → Asymptot	Nr → Visa poldata	Nr → visa nollst. data
Realdel	Imaginärdel	Multiplicitet	Nr → /Avg	Nr → /Ank

Beräkningsexempel

I komplexa talplanet



eller i form av överföringsfunktion G(s)·H(s)

$$\frac{(s+2)(s+4)}{s^2(s+1+j2)(s+1-j2)(s+3+j2)(s+3-j2)}$$

Kommentar	Tangenter	x-reg	Tid (s)
Initiera poler	A'	0	1
Dubbel pol s=0	0 A	1	2
	2 C	2	2
Pol s=-1+j2	1 +/- A	3	2
	2 B	4	3
Pol s=-3+j2	3 +/- A	5	2
	2 B	6	3
Initiera nollst.	B'	0	1
Nollställe s=-2	2 +/- A	1	2
Nollställe s=-4	4 +/- A	2	2
Tyngdpunkt	C'	-0,5	10
Asymptoter	R/S	45	1
	R/S	315	1
	R/S	225	1
	R/S	135	1
Visa pol nr 1	1 D'	0	pause
		0	pause
		2	pause
		3	6
Avgångsvinklar för pol nr 1	1 D	90	24+pau
		270	24+pau
		90	
		osv	
Stanna programmet	R/S	vad som helst	

Programbeskrivning

Programmet svällde ut till hela 451 steg men arbetar trots det ganska snabbt, eftersom relativt små delar körs åt gången. Det är bara rutinerna under D och E som är stora. Hela programmet är optimerat m a p exekveringstid och hanterbarhet (användarlogik). Det innebär att jag har fasta adresser i stället för labels vid olika sorters hopp.

Samtidigt med ovan nämnda optimering har jag försökt minska programutrymmet för att få så många register som möjligt. Det råkade bli normalinställning 479.59 vilket möjliggör 8 + 8 olika poler och nollställen. Minskningen av programmet har gjorts m h a indirekta minnesaccesser, flagganvändning och gemensamma utrymmen för olika delar av programmet (en sorts subrutiner).

I programlistningen är inhoppsadresser utmärkta med pilar och hoppadresser med klammer.

Rutinerna D, D', E och E' presenterar sina resultat under två pause-instruktioner (programsteg 251, 252 och 374, 375). De kan naturligtvis bytas ut mot Nop, R/S som jag visat i programlistningen. Tänk bara på att programmet arbetar med fasta adresser och undvik insert- och delete-funktionerna.

Om ni har några synpunkter eller frågor så heter jag Lars-Bertil Olsson och bor Fyradalersgatan 3, 413 19 GÖTEBORG.

PROGRAMLISTNING **ROTLOKUSPROGRAM**

```
000 76 LBL   057 87 IFF   114 69 DP   171  95 =    228 97 DSZ   285 14  D     342 03 03   399 03 03
001 16 A'    058 00  00   115 25  CL   172  69 DP   229 03  03   286 71 SBR   343 45 YX    400 95 =
002 86 STF   059 00  00   116 69 DP    173 24  24   230 02 02    287 02 02    344 22 INV   401 32 XIT
003 00  00   060 64 EE    117 25  CL   174 69 DP    231 00 00    288 65 EX    345 44 SUM   402 42 STO
004 03  3    061 42 STO   118 72 ST+   175 25  CL   232 61 GTO   289 73 RC+   346 05 05    403 23 23
005 04  4    062 02 02    119 05 05    176 44 SUM   233 01 01    290 05 05    347 43 RCL   404 43 RCL
→006 42 STO  063 92 RTN   120 86 STF   177 06 06    234 94 +/-   291 42 STO   348 02 02    405 08 08
007 03  03   →064 42 STO  121 01 01    178 97 DSZ   235 76 LBL   292 07 07    349 13 13    406 75 -
008 25 CLR   065 01 01    122 32 XIT   179 03 03    236 10 E'    293 69 DP    350 05 05    407 73 RC+
009 42 STO   066 92 RTN   123 02  2    180 01 01    237 86 STF   294 25 25    351 43 RCL   408 03 03
010 04  04   067 76 LBL   124 45  Y^   181 52 EE    238 00 00    295 69 DP    352 65 X     409 35 1/X
011 60 DEG   068 12  B     125 53  (    182 43 RCL   239 76 LBL   296 05 05    353 95 =     410 69 DP
012 86 STF   069 29 CP    126 32 XIT   183 01 01    240 19 D'    297 42 STO   354 32 INV   411 23 23
013 01 01    070 00  00   →127 76 LBL   184 42 STO   241 42 STO   298 04 04    355 44 SUM   412 22 INV
014 86 STF   071 00  00   128 01  1     185 43 RCL   242 00 00    299 69 DP    356 03 03    413 37 P/R
015 02 02    072 97 DSZ   129 95  =     186 02 02    243 71 SBR   300 07 07    357 42 STO   414 42 STO
016 92 RTN   073 72 ST+   130 44 SUM   187 95  =     244 02 02    301 73 RC+   358 05 05    415 73 RC+
017 76 LBL   074 03  03   131 04  04    188 22 INV   245 65 EX    302 05 05    →359 43 RCL  416 03 03
018 17 B'    075 32 XIT   132 61 GTO   189 42 STO   246 03 03    303 42 STO   360 05 05    417 22 INV
019 22 INV   076 69 DP    133 00 00    190 06 06    247 42 STO   304 09 09    361 42 STO   418 44 SUM
020 86 STF   077 07  33   134 55 DIV   191 42 STO   248 04 04    305 03 03    362 06 06    419 04 04
021 00  00   078 43 RCL   135 76 LBL   192 05 05    →249 73 RC+  306 05 05    363 43 RCL   420 95 =
022 09  9    079 03 03    136 28 LOG   193 43 RCL   250 05 05    307 42 STO   364 09 09    421 44 SUM
023 23  23   080 42 STO   137 48 EXC   194 06 06    251 66 PAU   308 03 03    365 42 STO   422 04 04
024 00  00   081 05 05    138 01  01    195 91 R/S   252 66 PAU   309 43 RCL   366 04 04    423 69 DP
025 76 LBL   082 08  8     139 42 STO   →196 42 STO  253 69 DP    310 01 01    367 35 1/X   424 23 23
026 76 LBL   083 44 SUM   140 03  03    197 05 05    254 25 25    311 71 SBR   368 71 SBR   →425 29 CP
027 11  A    084 03 03    141 03  3     198 42 STO   255 97 DSZ   312 02 02    369 02 02    426 76 LBL
028 87 IFF   085 73 RC+   142 05  5     199 03 03    256 04 04    313 87 IFF   370 22 INV   427 04 04
029 02 02    086 05 05    143 42 STO   →200 01  1    257 02 02    314 94 +/-   371 32 XIT   428 22 INV
030 00  00   087 72 ST+   144 04 04    201 08  8     258 49 PRD   315 42 STO   372 42 STO   429 44 SUM
031 34 FIX   088 03 03    145 01  1     202 00  0     259 22 INV   316 05 05    373 06 06    430 03 03
032 69 DP    089 02 02    146 00  0     203 95  =     260 86 STF   317 01 01    374 86 STF   431 95 =
033 23 23    090 94 +/-   147 42 STO   204 53  (     261 03 03    318 00 0     375 00 00    432 43 RCL
→034 69 DP   091 69 DP    148 05 05    205 02  2     262 43 RCL   319 42 STO   376 32 XIT   433 06 06
035 23 23    092 24 24    149 25 CL    206 65  X     263 03 03    320 07 07    377 44 SUM   434 00 00
036 69 DP    093 71 SBR   150 42 STO   207 43 RCL   264 38 RTN   321 43 RCL   378 06 06    435 76 LBL
037 24 24    094 00 00    151 06 06    208 03 03    265 76 LBL   322 08 8     379 32 XIT   436 14 D
038 72 ST+   095 44 44    →152 73 RC+  209 85 +     266 03 3     323 71 SBR   380 97 DSZ   437 22 INV
039 03 03    096 22 INV   153 04 04    210 01 1     267 42 STO   324 03 03    381 04 04    438 58 FIX
040 22 INV   →097 86 STF   154 65 X     211 54 )     268 07 7     325 87 IFF   382 03 03    439 08 8
041 86 STF   098 01 01    155 69 DP    212 55 ÷     269 87 IFF   326 95 =     383 73 RC+   440 00 0
042 02 02    099 43 RCL   156 24 24    213 43 RCL   270 07 07    327 01 1     384 07 07    441 00 0
043 25 CLR   100 04 04    157 69 DP    214 05 05    271 02 02    328 08 8     385 03 03    442 06 6
→044 69 DP   101 92 RTN   158 24 24    215 55 ÷     272 76 LBL   329 00 0     →386 43 RCL  443 02 2
045 23 23    102 76 LBL   159 73 RC+   216 03 3     273 85 +     330 95 =     387 32 XIT   444 05 5
046 72 ST+   103 13 C     160 04 04    217 06 6     274 02 2     331 44 SUM   388 04 04    445 93 .
047 03 03    104 86 STF   161 75 -     218 00 0     275 05 5     332 05 05    389 25 CL    446 07 7
048 69 DP    105 02 02    162 73 RC+   219 95 =     276 95 =     333 04 4     390 42 STO   447 03 3
049 23 23    106 69 DP    163 05 05    220 22 INV   277 42 STO   334 05 5     391 06 06    448 01 1
050 01 1     107 23 23    164 65 X     221 59 INT   278 05 05    335 65 X     392 61 GTO   449 08 8
051 72 ST+   108 72 ST+   165 69 DP    →222 65 X    279 32 XIT   336 43 RCL   393 04 04    450 92 RTN
052 03 03    109 03 03    166 25 25    223 03 3     280 76 LBL   337 09 09    394 25 CL
053 69 DP    110 87 IFF   167 69 DP    224 06 6     281 15 E     338 95 =     →395 43 RCL
054 33 33    111 01 01    168 25 25    225 00 0     282 86 STF   339 02 02    396 07 07
→055 43 RCL  112 01 01    169 73 RC+   226 95 =     283 02 02    340 87 IFF   397 75 -
056 04 04    113 27 27    170 05 05    227 92 RTN   284 76 LBL   341 03 03    398 73 RC+
```

FORTS. FR. SID. 33 HIR-...

Tabellen på s. 33, i vilken A - E representerar godtyckliga teckenkoder, ger exempel på vad som kan åstadkommas endast med 10-potenserna 6 och 12.

Det är nu lätt att inse, att om vi också tar övriga jämna 10-potenser till hjälp, så kan vi med vilken decimalplacering som helst åstadkomma önskad placering av skrivteknen. Och i så fall kan vi naturligtvis också med operationen 1 HIR 39 ("SUM") höja den närmast före decimalkommat stående koden med 1 och därmed också ändra tecknet som skall skrivas. När det gäller t ex tensorutskrift indexerade x-värden kan ändringskoden se ut så här

...1 HIR 38 Op 6 CLR Rtn

(Följande pgm förvandlar din PC-100 till något liknande en småll-press:

Lbl A Op 44 00 + 1 EE 12 = HIR 6

Lbl A' 1 HIR 38 Op5 GTO A')

Kör du det så finner du, att koderna 08 och 09 existerar och ger 7 resp 8.)

Sammanfattning. Vid inmatning av skrivkoder direkt i HIR 5 - HIR 8 gäller

skrivkoderna för de enskilda tecknen är desamma som vid inmatning via Op 1 - Op 4;

tre blindsiffror, jämpligen 100, måste föregå koderna;

blindsiffrorna kan tillfogas direkt före koden, t ex 100 13 35, eller

blindsiffrorna kan tillfogas i form av en lämplig 10-potens som adderas till det tal som representeras av följden av kodsiffror;

tecknens placering inom blocket bestäms av antalet par av nollor som finns mellan blindsiffrorna och de första egentliga (= från 00 skilda) kodsifferparen, varvid inget 0-par medför att första tecknet kommer på plats 1;

decimalkomma får införas mellan godtyckliga kodsifferpar. Ett sådant decimalkomma påverkar storleken av det inmatade talet och därmed antalet nollor före det första kodsifferparet, men ändrar i övrigt inte utskriften. Ex. 13 14 + 10^6 = = 100 13 14 placerar AB i lägena 1-2, medan 13,14 + 10^6 = 100 00 13,14 flyttar dem till lägena 2-3 och ,13 14 + 10^6 = 100 00 00,1314 flyttar den ännu ett steg åt höger;

Fix mode har ingen inverkan på utskriften;

ändring av en inmatad kod kan ske genom addition eller subtraktion med HIR 3X resp HIR 5X.

"Vackrare vardagsvara" är underrubriken på den här redogörelsen för HIR-registrens användning. Det ligger en liten tanke bakom den. Den vill antyda, att den beskrivna metoden att gå in i HIR-registren varken är något som skall komma i bruk varje gång vi gör ett program, eller något som vi har anledning att dra oss för att använda när vi har behov av det.

FAKULTETSPROGRAM i

FAST MODE

I PB 80-3 nämndes Fast Mode eller ett "snabbtillstånd". Det är faktiskt ingen kuriositet, utan kan praktiskt användas. Jag tänkte visa hur, med hjälp av ett pgm som beräknar fakultet exakt för tal ≤ 438. Ett sådant pgm finns redan i pgm-biblioteket. Enligt programmakaren tar det ca 24 timmar att köra hans pgm för 438!

Det här pgmet behöver ungefär 4 timmar och 11 minuter!

Programmet knäpps in på vanligt sätt och spelas in på magnetkort i normal uppdelning (lämpligen före provkörning).

För att köra det gör man på följande – litet ovanliga – sätt (som gäller för alla pgm som körs i Fast Mode):

1. Läs in pgmet.
2. Tryck RST R/S. (Standardmodulen måste vara isatt.)
3. Läs in pgmet igen.

Lägg märke till att pgmet börjar exekveras när man läser in kortet andra gången, utan nedtryckning av R/S. Detta är en av de underliga saker som händer i Fast Mode.

För det här pgmet är resterande instruktioner:

4. Mata in ett heltal och tryck ned R/S. Efter en stund skrivs resultatet ut. Upprepa detta steg om så önskas.

Några andra underligheter: Pgm som exekveras kan inte stoppas – varken med R/S eller med RST. Precis som med modulpgm fungerar inte trace-läget eller GTO-tangenten.

Nedtryckning av vissa tangenter startar pgm-körning, t ex, de aritmetiska tangenterna, funktionstangenter (t ex 1/x) och en del andra. Nedtryckning av siffertangenter, EE och CE har dock ingen sådan verkan. Därför bör man bara använda sistnämnda tangenter. (Från tangentbordet.)

I pgm bör också DMS, P/R, Σ+ m fl, som beräknas genom TI-58/59's fasta pgmvara (se "Djupdykning i TI-59"), undvikas. Pgm fastnar i en oändlig slinga, som dock kan brytas med RST.

Som stod i "Djupdykning i TI-59" får inte heller subrutinanrop användas, ej heller A – E' eller modulpgm användas. RTN får inte användas som stoppinstruktion. SBR non får dock starta ett program från tangentbordet.

Det verker också som om varken labels eller flaggor kan användas.

För att stanna pgmet i Fast Mode är inte alltid R/S tillräckligt. R/S måste **föregås** av en av följande: CLR, CE, en siffra, ., +/−, PRT, ADV, EE INV EE. (Fler möjliga instruktioner finns säkert.)

Ett pgm i Fast Mode kan inledas som mitt pgm (till steg 015). Därefter börjar man sitt eget pgm. Man bör tänka på att exekveringen av det kommer att startas när man läser in magnetkortet för andra gången. I mitt pgm satte jag då dit CLR R/S som stoppinstruktion direkt, eftersom programmet måste ha ett tal innan det startar. Gör man däremot ett pgm som inte behöver något startvärde är det naturligtvis onödigt att ha en stoppinstruktion.

Subrutinanrop måste undvikas. Det bästa är att skriva om pgmet och använda någon sorts snurra, som jag har gjort i fakultets-pgmet. I ursprungs-programmet, utan Fast Mode användes en "skrivbordsomvandlare", placerad först i pgmet under Lbl A', och anropades så här:

A' UP 01 A' UP 02 UP 30 A' UP 03 A' UP 04 ...

Detta är ett ganska bekvämt sätt, men kan inte användas i Fast Mode.

Om subrutiner inte kan ersättas med en snurra kan man simulera subrutinanrop och subrutinåterhopp (SBR och RTN) med GTO och GTO IND.

Varje gång man då skall använda en "subrutin" lagrar man en återhoppsadress i ett register och använder GTO nnn. I subrutinens slut används GTO IND nn, där nn är numret på registret.

Nedan visas hur man kan bära sig åt. Pgmet är endast tänkt som demonstration av simulering av subrutiner och gör inget vettigt. Som jämförelse finns ett pgm med normala subrutinanrop.

Björn Gustavsson, Smedjebacken

000	00	0	071	43	EXC	142	42	RCL	000	00	0
001	00	0	072	00	00	143	00	00	001	00	0
002	00	0	073	00	EXC	144	04	4	002	00	0
003	00	0	074	00	00	145	77	CE	003	00	0
004	00	0	075	42	STO	146	00	00	004	00	0
005	76	LBL	076	00	00	147	00	00	005	36	PGM
006	00	A'	077	97	DSZ	148	42	RCL	006	02	02
007	11	SBR	078	00	00	149	00	00	007	00	SBR
008	00	0	079	00	00	150	69	DP	008	02	02
009	00	0	080	39	SIN	151	05	05	009	39	SIN
010	00	0	081	93	.	152	01	1	010	09	9
011	00	0	082	93	.	153	67	EQ	011	00	0
012	00	0	083	91	R/S	154	84	+/-	012	22	INV
013	00	0	084	44	SUM	155	97	DSZ	013	56	FIX
014	22	INV	085	98	98	156	61	GTO	014	22	INV
015	56	FIX	086	76	-	157	00	00	015	END	
016	01	CLR	087	69	DP	158	12	12	016	02	2
017	00	0	088	04	4				017	04	4
018	69	DP	089	95	=				018	42	STO
019	00	00	090	05	5				019	00	00
020	98	RTN	091	00	0				020	05	5
021	94	+/-	092	93	.	016	02	2	021	03	3
022	97	DSZ	093	95	=	017	03	3	022	42	STO
023	01	1	094	42	STO	018	07	07	023	07	07
024	00	0	095	00	00	019	42	STO	024	43	RCL
025	69	DP	096	05	CLR	020	00	00	025	01	01
026	14	14	097	92	HIR	021	03	3	026	88	GD+
027	47	CMS	098	00	00	022	07	7	027	07	07
028	00	0	099	69	DP	023	97	DSZ	028	43	RCL
029	42	STO	100	00	00	024	43	RCL	029	02	02
030	07	STO	101	05	5	025	01	01	030	88	GD+
031	01	99	102	42	STO	026	16	A'	031	07	07
032	99	PRT	103	99	99	027	42	STO	032	43	RCL
033	00	0	104	73	RC*	028	02	02	033	03	03
034	58	FIX	105	58	FIX	029	03	3	034	88	GD+
035	25	CLR	106	65	×	030	43	RCL	035	07	07
036	35	1/x	107	01	1	031	03	03	036	04	4
037	42	STO	108	00	0	032	16	A'	037	03	3
038	51	51	109	92	HIR	033	04	4	038	42	STO
039	51	LDG	110	45	YX	034	42	STO	039	10	10
040	86	STF	111	92	HIR	035	04	04	040	61	GTO
041	86	INT	112	45	YX	036	01	1	041	00	00
042	69	DP	113	65	×	037	10	SBR	042	64	64
043	69	DP	114	01	1	038	00	00	043	42	STO
044	06	6	115	55	÷	039	00	00	044	06	06
045	69	PGM	116	28	LOG	040	42	STO	045	08	8
046	00	00	117	59	INT	041	08	08	046	44	SUM
047	52	EE	118	65	×	042	08	8	047	10	10
048	74	SM+	119	02	2	043	SUM		048	61	GTO
049	76	-	120	75	-	044	10	SBR	049	00	00
050	25	CLR	121	59	INT	045	71	SBR	050	64	64
051	00	RCL	122	92	HIR	046	00	00	051	25	CLR
052	00	00	123	33	X²	047	62	62	052	91	R/S
053	59	INT	124	95	=	048	25	CLR	053	34	FX
054	22	INV	125	97	DSZ	049	R/S		054	85	+
055	74	SM+	126	99	99	050	76	LBL	055	35	1/x
056	00	00	127	01	1	051	92	RTN	056	54	)
057	52	EE	128	06	06	052	34	FX	057	44	SUM
058	01	1	129	75	ST*	053	85	+	058	11	11
059	69	0	130	08	08	054	95	1/x	059	04	4
060	94	+/-	131	92	HIR	055	54	)	060	44	SUM
061	97	DSZ	132	13	13	056	44	SUM	061	00	00
062	96	96	133	84	DF+	057	11	11	062	89	GD+
063	00	00	134	00	00	058	04	4	063	00	00
064	42	42	135	92	HIR	059	45	SUM	064	43	RCL
065	29	CP	136	05	5	060	00	00	065	02	02
066	67	EQ	137	02	INV	061	92	RTN	066	95	1/x
067	00	00	138	44	SUM	062	43	RCL	067	89	GD+
068	69	69	139	98	98	063	02	02	068	10	10
069	29	DP	140	03	3	064	95	1/x			
070	20	20	141	75	-	065	92	RTN			

Egen modul!

Red: Björn Gustavsson
Nordlandervägen 12 A
??? OO SMEDJEBACKEN

De enda synpunkter på en egen modul som jag hittills har fått är följande brev från Tomas Dyfverman (som sändes till Stockholms-redaktionen och sedan vidarebefordrades till mig):

Låt oss grovt säga att det kostar 200.000 kr att göra en egen modul. Den rymmer 5.000 st programsteg. Det blir ca 40 kr/st. Låt vem som helst teckna sig för önskat antal steg och betala t ex 500 + 40/steg. Föreningar och andras "öppna" program slipper startavgiften på 500 kr.

Modulen skulle kunna innenålla:

- ★ kommersiella program från företag (listskyddat etc) som inte kan utnyttjas av andra än dom som respektive företag säljer modulen till.
- ★ Öppna publicerade subrutiner etc som finansieras genom att intresserade i förväg binder sig som delfinansiärer av dessa steg. Om vi är hundra som är intresserade av ett visst program, så blir det ju bara 40 öre per steg för att få det i modulform. I många fall kan då TI-58 användas i stället för TI-59, vilket spar en tusenlapp per apparat.
- ★ korta program från privatpersoner/föreningar/grupper, öppna eller skyddade.

Jag tror att det är 1.000 ex som ingår i de 200.000 kronorna. Således kan man nu rätt att få ett modulex per 5 tecknade rader. Och extraupplagor lär vara, relativt sett, ändrligt mycket billigare än grundpriset.

Huvudtanken är alltså att inte försöka fylla hela modulen med "öppna" subrutiner – vilket jag bedömer som orealistiskt – utan att istället acceptera att man bara kan besagna en del av den teoretiska kapaciteten (som man fyller med det viktigaste), och som beläsning får man ett väldigt mycket lägre per skalle. Och, icke minst, det hela kanske blir av!

Vad säger Du? Vad säger TI? Vad säger Föreningen om att sköta "programstegstecknigen"?

Ni som har synpunkter på detta – hör av er (till adressen ovan).

Också ni som har synpunkter på allt annat som har med "Egen modul" att göra – hör av er.

Målsättningen med denna sida är att:

★ Utveckla pgmbitar och hela pgm, som är lämpliga att ha med i en eventuell modul, och som förhoppningsvis kan vara lämpliga att använda utan modul, om det inte blir någon modul.

★ Diskutera hur finansiering av en modul skall gå till.

För att detta skall kunna uppfyllas är det givetsvis nödvändigt med läsarnas medverkan.

Vad gäller den första punkten så tar jag tacksamt emot förslag av alla sorter: Både helt färdiga pgm och endast förslag hur ett pgm skall fungera, och synpunkter på tidigare införde pgm.

För att få igång verksamheten har jag själv skrivit några förslag:

Att utnyttja skrivarens alfanumeriska möjligheter är tidsödande och minneskrävande. Jag tycker alltså att det behövs pgm i modulen som underlättar sådant utnyttjande. Vad tycker du? Skall några av modulens dyrbara pgmsteg användas till detta? Om ja, hur skall pgmen fungera?

I det här fallet är både pgm och synpunkter välkomna.

Pgmet nedan gör samma sak som ML-07 – "Polynomberäkningar". Är ett sådant pgm användbart i modulen?

Det är också ett inlägg i debatten som förts på pgmbits-sidorna om hur y^x skall beräknas på bästa sätt.

Bruksanvisningen är mycket lik den till ML-07: steg 1 & 2 i den bruksanvisningen överhoppas, medan steg 3 utföres med A i stället för B. Annars ingen skillnad.

På återhörande

Björn Gustavsson

000	76	LBL	011	76	LBL	022	82	HIR	033	73	RC*
001	11	A	012	12	B	023	18	18	034	00	00
002	42	STO	013	42	STO	024	07	07	035	54	)
003	00	00	014	00	00	025	00	00	036	97	DSZ
004	43	RCL	015	61	GTO	026	00	0	037	00	00
005	00	00	016	00	00	027	43	RCL	038	00	00
006	82	HIR	017	04	04	028	24	CE	039	87	IFF
007	08	08	018	16	A'	029	85	+	040	92	RTN
008	69	DP	019	42	STO	030	82	HIR			
009	20	20	020	82	HIR	031	17	17			
010	32	X↑I	021	07	07	032	85	+			

Det här är en förbättrad version av bråkräkningspgmet i PB 80-3. Minsta absolutrest används i Euklides' algoritm, som Lösta Blume rekommenderade. Andra förbättringar har också gjorts. Ett resultat lagras automatiskt som när A trycks ned, vilket möjliggör snabba kedjeberäkningar (jfr ML-04 – ML-06).

För både in- och utdata gäller: täljare xit nämnare.

A' är förkortningsrutin

A matar in första bråket i en beräkning

B adderar

C subtraherar

D multiplicerar

E dividerar

BRÅK-RÄKNING

Björn Gustavsson, Smedjebacken

000	49	PRD	021	53	(	044	49	PRD	066	33	X↑T
001	04	04	022	49	PRD	045	01	01	067	34	FX
002	92	RTN	023	00	00	046	02	02	068	75	-
003	32	X↑T	024	55	÷	047	49	PRD	069	76	LBL
004	65	×	025	00	00	048	03	03	070	12	B
005	43	RCL	026	43	RCL	049	43	RCL	071	49	PRD
006	03	03	027	00	00	050	02	02	072	04	04
007	55	÷	028	54	)	051	32	X↑T	073	32	X↑T
008	76	LBL	029	55	÷	052	43	RCL	074	65	×
009	14	A'	030	53	(	053	04	04	075	43	RCL
010	56	FIX	031	43	RCL	054	22	INV	076	03	03
011	00	00	032	82	HIR	055	58	FIX	077	85	+
012	42	STO	033	12	12	056	76	LBL	078	22	INV
013	01	01	034	95	=	057	11	A	079	76	LBL
014	32	X↑T	035	22	INV	058	42	STO	080	19	E'
015	42	STO	036	67	EQ	059	03	03	081	32	X↑T
016	02	02	037	00	00	060	32	X↑T	082	76	LBL
017	42	STO	038	95	=	061	42	STO	083	14	D
018	00	00	039	42	STO	062	04	04	084	32	X↑T
019	25	CLR	040	43	RCL	063	04	4	085	65	×
020	32	X↑T	041	00	00	064	76	LBL	086	91	R/S
021	65	×	042	50	IX[	065	13	C			
			043	22	INV	066	92	RTN			

RECENSIONER

Lennart Råde: Att programmera miniräknare
(capris 40 kr)
Att programmera Texas TI 58 och TI 59
(capris 40 kr)
Äventyr med programmerbar miniräknare
(capris 79 kr)
Studentlitteratur

-oOo-

Råde surfar vidare på miniräknarvågen och har nu landat med ytterligare tre böcker om programmering.

"Att programmera miniräknare" är en mycket elementär handledning, som i separata kapitel behandlar "raka" pgm, pgm med villkorliga hopp och pgm med subrutiner. De ämnesområden ur vilka problemen hämtats är huvudsakligen de vanliga: talteori, kombinatorik och sannolikhet, men även tillämpningar inom ränteberäkning, ackord, amortering och bilhyra förekommer.

Problemen är som sagt mycket enkla men samtidigt instruktiva. Boken beskriver pgm med hjälp av flödesdiagram och är därför oberoende av vilken typ av räknare som läsaren har. Tanken är att boken skall kompletteras med häftten som direkt tillämpar de genomgångna principerna på olika räknartyper.

"Att programmera Texas TI 58 och TI 59" är ett sådant pgmhäfte och naturligtvis det som är av särskilt intresse för oss PB-medlemmar. Pgm är ordentligt listade, emellanåt i ett par versioner, kompletterade med flödesscheman och vid behov försedda med kommentarer.

Rutinerade TI-programmerare har inte mycket att hämta ur dessa båda böcker, även om ett och annat visst knep kan vara värt att ta vara på. Men för nybörjare, och främst då sådana som har blivit skrämda av handbokens mängder av syntaktiska regler, öppnar de en inkörsport på vid gavel.

"Äventyr med programmerbar miniräknare" står, utan att vara direkt avancerad, på ett högre plan. Ämnesområdena för de ca 135 problemen är i stort sett desamma som i de nyss nämnda böckerna. Problemen inom flertalet av ämnesområdena föregås av korta presentationer av den matematiska eller kulturella (!) bakgrunden.

Kommentarer med anvisningar för programmeringen och programmens handhavande bildar en särskild del av boken.

Vad som nästan fullständigt saknas i "Äventyr" är problem som inte i och för sig är av omedelbart matematisk karaktär, men som kan lösas genom konstruktion av en matematisk modell, följd av en simulering. Om programmeringen av en given matematisk operationsföljd är den kanske allra enklaste pgmuppgiften, så är nog konstruktionen av matematiska modeller det

FORTS.FR.SID. 19

Periodiskt decimalbråk till allmänt bråk

```
      Fix 0 CP
  3   RCL1 - RCL2 x ( RCL1 : RCL2 ) CE
 17   = Exc2 STO1 CLR
 23   RCL2 INV x=t 0'3
 29   RCL3 : RCL1 = /x/ : R/S
 38   (RCL4 : RCL1 ) /x/ R/S
 47   = INV Fix Rtn
 51 Lbl A INV log EE ST05 CLR R/S
 60 Lbl B INV log EE ST06 CLR R/S
 69 LBL C x RCL5 x RCL6 = Int -
 80   (CE : RCL6 ) Int = STO1 STO'3
 92   RCL6 - 1 ) x RCL5 = ST02 STO'4
105   RST
106 Lbl A' STO1 ST03 R/S ST02 ST04 RST
                                    117
```

FÖR HIR-vänner

Detta program är absolut någonting för HIR-vänner! Det fyller nämligen hela programminnet med "HIR"! Men det är inte bara en ploj - det är arbetsbesparande om man gör ett program med många HIR (man skriver helt enkelt över med egna koder och för HIR gör man SST) och man kan ha det vid räkningar med HIR från tangentbordet - tryck bara SST och den tvåsiffriga koden!

Anvisning: Knappa in programmet. Kör det med E. Spela in block 4 med "4 Write". Tryck "1 +/-" och läs kortet (koderna från block 4 tvångslagras i block 1). Tryck "2 +/-" och läs samma sida. Nu är programminnet fullt med HIR från 0 till 479. Eftersom det väl är enklare att spara de färdiga blocken 1 och 2 (om du vill det) kan du ju nu spela in dem på samma kort med "1 Write" och "2 Write" - programmet behöver du ju inte längre!

(Bearbetning av danska "Pgm")

000	76	LBL	011	02	2	022	02	2	033	97	DSZ
001	15	E	012	08	8	023	08	8	034	00	00
002	02	2	013	02	2	024	54	)	035	00	00
003	09	9	014	94	+/-	025	65	x	036	31	31
004	42	STD	015	75	-	026	01	1	037	42	STD
005	00	00	016	93	.	027	52	EE	038	00	00
006	53	(	017	08	8	028	02	2	039	25	CLR
007	53	(	018	02	2	029	03	3	040	92	RTN
008	08	8	019	08	8	030	54	)			
009	02	2	020	02	2	031	72	ST+			
010	08	8	021	08	8	032	00	00			

MEDLEMS ANNONSER

TI-59 + SKRIVARE 1.750 kr inklusive 10 pappersrullar.
Lars Testad tel arb 08/82 04 00
bost 08/85 24 75.

som flertalet pgmbitna finner vara det svåraste.
Kan vi hoppas på att Lennart Rådes nästa bok behandler det ämnet?

blm

TILLBEHÖR TILL TI 59

MODULVÄLJARE

En firma i USA har börjat sälja anslutningsboxar för upp till fyra moduler till TI-59! De är av två sorter, en manuell för $ 99.95 och en som kan manövreras genom program (!) för $ 199.95. Från lådan för modulerna går en kabel som slutar i en kontakt utformad som en modul så att den passar i modulfacket. Ingen extra strömkälla erfordras. Om modulerna kallas O till 3 anropas nr 3 med koden 3 77 op 04 op 05 och därefter som vanligt Pgm xx. Ovanstående uppgifter kommer från TI PPC Notes V5N8, och tillverkaren lämnar $ 10 i rabatt om TI PPC Notes åberopas. Anordningen försäljes av Systems 7, 1020 Park Ave., Suite 101, CRANSTON RI, 02910, USA. Den tillverkas av en firma som tillverkar delar till Texas Instruments, nämligen American Micro-Products Inc i Richardson, Texas.

PRIVAT BYGGMODUL

En firma Gespro i Tyskland har på egen hand utarbetat underlaget för en modul med program i byggnadsstatik som säljs för 450 DM. Tyvärr har det dröjt att få hit den, men nu kan den erbjudas för 1.050 kr + moms. Den har ett program som på sitt sätt är smått fantastiskt men som å andra sidan tyvärr inte är så användarvänligt att vem som helst sätter sig ner och använder det vid behov. Det är ett program för kontinuerliga balkar som klarar jämnt fördelade laster, triangulära laster (även på delar av fackens spännvidd), punktlaster, punktmoment, stödsänkningar, elastiska stöd, elastiskt underlag och allt detta med variabelt tröghetsmoment! Teoretiskt finns det minnesutrymme för 35 fack (!) men räknenoggrannheten räcker "bara" till ca 15. Men även det brukar ju räcka. Programmet ger vidare vinkeländringar och nedböjningar i godtyckliga punkter. Dessutom går det att räkna med andra ordningens teori och få moment och utböjning i valfria punkter på en excentriskt belastad pelare. Vidare finns ett omfattande fackverksprogram och tröghetsmoment för polygoner och tunnväggiga profiler. Och till sist ett program för flervåningsramar! Beställes hos Lars Hedlund, Årstavägen 27, 121 68 Johanneshov, tel 08/91 38 97.

OM PROGRAM-SKRIVNING

Det skrivsätt för pgm som Gösta Blume föreslog tycker jag inte är lämplig för pgm som skall publiceras i PB. Metoden har följande nackdelar:

1. Det tar lång tid att skriva en sådan "listning", jämfört med att trycka på List.
2. Risken för fel är för stor.
3. Det tar längre tid att knäppa in ett pgm i dosan.

Beträffande läsligheten, så är nog pgmen lättlästa - om man vänjer sig vid metoden. Jag tycker faktiskt att en "riktig" listning är lättare att läsa. Men det är väl en fråga om vana.

Själva idén att skriva pgmet i fraser är bra. Därför bör man uppdela "riktiga" listningar i fraser (tycker jag). Det görs genom att dra streck mellan raderna i en listning (se listningen av TI-57-listningspgmet i förra numret eller Fermat-pgmet i det här).

En annan sak att tänka på när man skall skriva instruktioners namn i löpande text: Hur skall de skrivas?

Den instruktion som har flest varianter är väl x⇆t. Det bästa vore ju att skriva den som i raden ovanför, men det tar ju sådan tid, när man skriver skrivmeskin. Av de symboler som kan skrivas på skrivmaskin direkt utan penna tycker jag att X≥T eller x≥t är att föredra.

Den symbolen liknar mest den på en listning.

En annan "knepig" symbol är ÷. Den går att skriva på skrivmaskin: : (kolon och streck), men är tyvärr för likt ett plus-tecken. Den symbolen kan också likna ett minus-tecken i "riktiga" listningar, speciellt i en förminskad och blek kopia. Här är inte listningens symbol särskilt lämplig. I TI PPC NOTES används DIV. Skall den symbolen införas också i PB? Eller är : eller lämpligare?

Det var mina åsikter i frågan. Liksom Gösta Blume tycker jag att läsare som har åsikter om pgm-skrivning skall skriva till PB.

Björn Gustavsson, Smedjebacken

KOMMERSIELLA PROGRAM

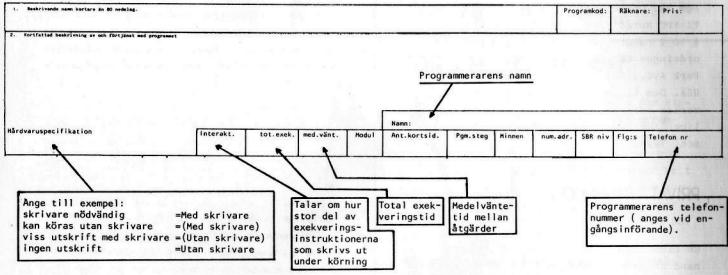
BYGGNADSTEKNIK

Kontinuerliga balkar, en- och flervåningsramar, takstolar och fackverk, dimensionering av betong, stål och trä, stödmurar, pålgrupper, sponter, plattor.

Carl-Adolf Granholm
Box 5030
421 05 V. FRÖLUNDA
Tel 031/29 20 75

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

PROGRAMBIBLIOTEKET



VÄGLEDNING VID MEDLEMSBESTÄLLNING

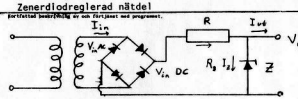
All medlemsbeställning skall ske på postgiroblankett med den aktuella summan inbetald. Program och övrig materiel kommer då direkt i brevlådan. I annat fall får programmet uthämtas mot postförskott. Kontant betalning kan ej accepteras.
De priser som anges i programbiblioteket är vad Du betalar för att få programmet hem i brevlådan.

Kom ihåg att:
-betalning sker till postgiro 430 01 59 - 3 och lämpligen på föreningens inbetalningskort.
-ange önskad vara med den referenskod som är knuten till varan i tidningen. I Programbiblioteket står denna kod på samma rad som programnamnet.
-leverans sker till den adress som finns angiven på postgiroinbetalningskortet.

Vid leverans av program ur programbiblioteket får Du:
-en kopia av ifylld mall för inskickade program, i förekommande fall.
-programbeskrivning och listning.
-hela programmet på inspelade magnetkort.

1. Beskrivande namn kortare än 80 omdning.
Zenerdiodreglerad nätdel
Programkod: 89aa
Räknare: TI-58/59
Pris: 40:-

2. Fortfattad beskrivning av och förtjänst med programmet.



Programmet beräknar motståndet (ohm och effekt), effekten på zenerdioden och strömmen från transformatorn. Givet: V_{in} AC el. DC, V_{ut}, I_{ut} och $I_{z\ min}$

Namn: Thomas Fänge

Hårdvaruspecifikation	Interakt.	tot.exek.	med.vänt.	Modul	Ant.kortsid.	Pgm.steg	Minnen	num.adr.	SBR niv	Flg:s	Telefon nr
ej skrivare	-	5 s.	-	-	1	97	8	-	-	-	

1. Beskrivande namn kortare än 80 omdning.
Golf
Programkod: 89ab
Räknare: TI-59
Pris: 40:-

2. Fortfattad beskrivning av och förtjänst med programmet.
Realistiskt golfprogram med variabla banor, bunkrar, greenspel, vind och studsande bollar.
För en eller två personer. Två banexempel medsändes.

Namn: Johan Westerlund

Hårdvaruspecifikation	Interakt.	tot.exek.	med.vänt.	Modul	Ant.kortsid.	Pgm.steg	Minnen	num.adr.	SBR niv	Flg:s	Telefon nr
Utan skrivare			30 s.	1	4	635	37			6	

1. Beskrivande namn kortare än 80 omdning.
TYNGDEPUNKT OG VEKT
Programkod: 89ac
Räknare: TI-59
Pris: 40:-

2. Fortfattad beskrivning av och förtjänst med programmet.
Program for beregning av total vekt og felles vektstyngdepunkt for ubegrenset antall dataelement. Programmet gjør overflødig tidkrevende mellomregning av statisk moment. Tyngdepunkt kan beregnes om 1,2 eller 3 akser. Med to akser blir et sett av inputdata av formen: VEKT, ARM 1, ARM 2 som mates inn fortløpende med A/S. Kontinuerlig registrering av alle inputdata for eventuell etterkontroll. Siste sett med inputdata kan slettes med tangenten E. Selvforklarende resultatutskrift.

Namn: Audun Hjortås

Hårdvaruspecifikation	Interakt.	tot.exek.	med.vänt.	Modul	Ant.kortsid.	Pgm.steg	Minnen	num.adr.	SBR niv	Flg:s	Telefon nr
Med skriver				—	1,2	446	10				

1. Beskrivande namn kortare än 80 omdning.
Sortering av längder och kast vid tävlingar.
Programkod: 89ad
Räknare: TI-59
Pris: 40:-

2. Fortfattad beskrivning av och förtjänst med programmet.
Programet sorterar talen störst till minst. Man har även med startnummer eller åtta siffrors textkod. PGMiet kontrollerar så att det bästa resultatet av varje startnummer är med, när en tävlandes resultat är sämre än sitt föregående resultat blinkar det sämre som då talar om att han har gjort bättre resultat. Man får fram avståndet till ettan med en paus samt placeringen. Man kan trycka fram en resultatlista manuellt, samt avstånd till ettan. Man kan även söka upp ett speciellt st.nr för kontroll av vilken placering den har. Är skrivare innkopplad fås viss utskrift, men får även en komplett resultatlista utskriven.

Namn: Claes-Göran Frankenberg

Hårdvaruspecifikation	Interakt.	tot.exek.	med.vänt.	Modul	Ant.kortsid.	Pgm.steg	Minnen	num.adr.	SBR niv	Flg:s	Telefon nr
(med skrivare)		25 sek			3 st.	398	30	ja	1	0	0920/32324

ENGÅNGSINFÖRANDE

Program införda under rubriken Engångsinförande förmedlas EJ av Föreningen Programbiten.
Kontakta istället programmeraren (förslagsvis per telefon) för vidare information.

1. Beskrivande namn kortare än 80 omdning.
Köteori
Programkod:
Räknare: TI-58/59
Pris: 35:-

2. Fortfattad beskrivning av och förtjänst med programmet.
Om det genomsnittliga antalet kunder/tidsenhet som kommer till ett antal parallella kassor och det antal kunder som en kassa i genomsnitt kan betjäna/tidsenhet är känt, så kan programmet för olika antal parallella kassor räkna ut bl.a. sannolikheten att en kund skall behöva vänta, det genomsnittliga antalet kunder i kön och den tid kunderna i genomsnitt får vänta i kön. Programmet förutsätter att kunderna anländer oberoende av varandra, att inga kunder lämnar systemet p.g.a. långa köer och att betjäningstiden per kund är Poissonfördelad.

Namn: Carl-Magnus Ekermann,Tunag. 7A, 753 37 Uppsala

Hårdvaruspecifikation	Interakt.	tot.exek.	med.vänt.	Modul	Ant.kortsid.	Pgm.steg	Minnen	num.adr.	SBR niv	Flg:s	Telefon nr
Utan printer		13 s min		-	1	221	16				018/138750

Rekommenderade fackhandlare

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

AB Harry Andersson
Lunahuset
151 21 SÖDERTÄLJE
Tel: 0755/325 20

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Claes PR ex

PROGRAM BITEN 80-4

Innehåll

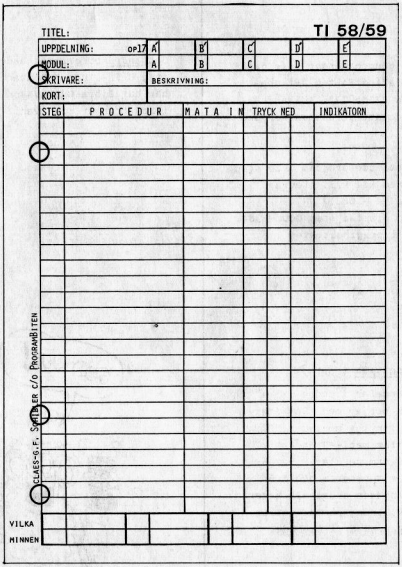
Dyre medlem!	2
Insänt	3
Utmaningen	4
Pgm-bitar	8
Kommentar till handboken	10
Avbetalning	13
Game of life	14
Bråkfinnaré 2	18
Spline	20
Fermats sats	22
Djupdykning i TI-59 (2)	23
Tröghetsmoment TI-57, 58/59	24
Komponentritningar	29
HIR - vackrare vardagsvara	30
EE-modulen och ett rotlokusprogram	34
Fakultetsprogram i fast mode	38
Egen modul	39
Recensioner	40
Tillbehör till TI-59	41
Programbiblioteket	42



DET KOM ETT BREV....

från Bertil Wallin Karlskoga med en efterlysning;
PÄRM för förminskad blankett TI-58/59 A och plastficka för tillhörande magnetkort,- för att enkelt ha kör-instruktionerna med tillsammans med räknaren,hemma och på jobbet.
Jag har gjort en speciell blankett vilken visas här bredvid i naturlig storlek :A6 ,och har i Farsta bok-handel funnit en passande pärm.Den har sk. Trio-meka-nism,så det vanliga standard hålslaget passar.

HETZEL Ringbok nr 74 46 94
International Registernr 77 46 94
och kostar kr 39:90 Blad nr 35 06 /4
Blankettens pris ?! 100 blad ev.i block skr 11:-
inklusive porto = som för de stora blocken om 50 blad A4.



"Press-stopp-nytt"

Egen modul!

Från Maurice Swinnens TI PPC Notes i USA meddelas nu att man dels tagit in anbud på modul från TI, dels börjat på allvar fundera över vad som skall ingå i den.

Först priset: Det var förvånansvärt förmånligt även för små upplagor och skulle betyda att vi i Sverige skulle behöva betala högst 250 - 300 kr. En större upplaga medför att priset sjunker, och det är därför att hoppas att så många som möjligt vågar binda sig på ett tidigt stadium. Att tillverka en ny upplaga om den första skulle ta slut är förmodligen oekonomiskt och därför osäkert.

Så till innehållet: Maurice Swinnen föreställer sig t ex dels skrivkodsomvandlaren i 80-3 (rättad på blad som följer med detta nummer), dels olika sorteringsrutiner och överhuvud sådant som skulle passa i M/U-modulen men inte finns där (alltså s k "utility", dvs hjälpprogram). Och här är alltså ännu möjligheterna fria att önska! Vi har velat gå ut med det här bladet nu så att vi kan få in svar i början av januari för vidare befordran till USA. TI behöver tre månader för tillverkning och om vi är snabba från vår sida kan vi alltså påverka innehållet i modulen och den skulle ändå kunna vara klar till sommaren. I nästa nummer av Programbiten skulle vi då summera resultatet - förslagsställandet skulle då vara avslutat.

Skriv nu till Björn Gustavsson, Nordlandervägen 12 A, 777 00 SMEDJEBACKEN, och tala om vad Du vill ha för innehåll för att vara med om att lägga max 300 kr på en modul!

LH

PROGRAM BITEN 80-4

Sid. 9

ETT ANVÄNDBART KNEP FÖR TI-57

Nedanstående listning tillhör texten.

```
00  32 0 STO 0
01  19   CT
02  15   CLR
03  81   R/S
04  32 1 STO 1
05  02   2
06 -39 1 INV PRD 1
07  15   CLR
08  81   R/S
09  32 2 STO 2

10  80   SI
11  84   +/-
12 -76   INV GE
13  32 3 STO 3
14  24   CE
15 -34 7 INV SUM 7
16  65   -
17  89   R
18 -34 7 INV SUM 7
19  85   =

20  81   R/S
21  71   RST
22  86 2 LBL 2
23  84   +/-
24  24   CE
25  22   INT
26  89   R
27  84   +/-
28  51 4 GTO 4
29  71   RST
```

Sid. 20

SPLINE

I "Inmatning första gången" anges inte det i och för sig självfallna faktum att uppdelningen skall vara "7 op 17". Följande anvisning är avsedd att göra inmatningen ännu lättare:

1. Tryck "7 op 17".
2. Knappa in nedanstående program i 29 steg.

2A. Kör programmet med RST R/S. (Lagrar värden i minnena 53 - 68.)

2B. Knappa in Spline pgm 1 i steg 000 - 384.

Fortsätt med steg 3 i originaltexten.

```
000  05 5      008  93 .      016  04 4      024  97 DSZ
001  04 4      009  05 5      017  54 )      025  00 00
002  42 STO    010  94 +/-     018  95 I/N    026  00 00
003  01 01     011  42 STO    019  94 +/-     027  13 13
004  01 1      012  53 53     020  72 ST*     028  91 R/S
005  06 6      013  53 (      021  01 01
006  42 STO    014  24 CE     022  69 DP
007  00 00     015  85 +      023  21 21
```

Sid. 32

HIR - VACKRARE VARDAGSVARA (forts)

I vänstra spalten, ungefär i undre 1/3-punkt, i stycket som börjar med "Det återstår att slå in --" strykes meningen "På motsvarande sätt förfar man --" ända till styckets slut "-- så som de står." Det är alltså omöjligt, både i program och från tangentbordet, att använda andra HIR-koder än lagring och återkallning på tal mellan -1 och 1 om inte räknaren är i EE-mode.

Sid. 38

FAKULTETSPROGRAM I FAST MODE

I vänstra spalten, undre 1/4-punkt står det: "Det verkar också som om varken labels eller flaggor kan användas." Flaggor kan dock användas - se t ex "Program 4" på s.7 som också är i fast mode.

PROGRAM BITEN 80-3

RÄTTELSE

Och så två rättelser och förbättringar till "ROBERT SNOWS BERÖMDA SKRIVKODSOMVANDLARE" i Programbiten 80-3, s. 8, vänstra spalten. Det första programmet kan enligt Björn Gustavsson förbättras på följande sätt:

```
000  76 LBL    011  69 DP     022  00 0.     033  01 1
001  99 PRT    012  28 28     023  00 0      034  00 0
002  29 CP     013  32 INV    024  49 PRD    035  97 DSZ
003  67 EQ     014  28 LOG    025  01 01     036  08 08
004  00 00     015  85 +      026  02 2      037  00 00
005  41 41     016  01 1      027  75 -      038  15 15
006  55 :      017  85 +      028  59 INT    039  25 CLR
007  28 LOG    018  28 LOG.   029  44 SUM    040  48 EXC
008  59 INT    019  59 INT    030  01 01     041  01 01
009  42 STO    020  65 x      031  95 =      042  92 RTN
010  08 08     021  01 1      032  65 x
```

Det andra programmet hade ett direkt fel i steg 017 - inte "+" utan "division". Även här har Björn Gustavsson gjort motsvarande förbättringar som ovan:

```
000  76 LBL    014  01 01     028  85 +      042  95 =
001  99 PRT    015  32 X!T    029  28 LOG    043  65 x
002  29 CP     016  50 I-J    030  59 INT    044  01 1
003  67 EQ     017  55 ÷      031  65 x      045  00 0
004  00 00     018  28 LOG    032  01 1      046  97 DSZ
005  52 EE     019  59 INT    033  00 0      047  08 08
006  32 X!T    020  42 STO    034  00 0      048  00 00
007  22 INV    021  08 08     035  49 PRD    049  24 24
008  77 GE     022  69 DP     036  01 01     050  25 CLR
009  00 00     023  28 28     037  02 2      051  43 RCL
010  12 12     024  22 INV    038  75 -      052  01 01
011  02 2      025  28 LOG    039  59 INT    053  92 RTN
012  00 0      026  85 +      040  44 SUM
013  42 STO    027  01 1      041  01 01
```

