

HEJ DU Programmeringsintresserade!

Det förutsätter vi nämligen att Du är om Du kommit så långt att Du börjat öppna den här lilla broschyren.

Vi är en förening av ägare till programmerbara räknare (mest TI-57, TI-58 och TI-59) och vi ger fyra gånger om året ut ett medlemsblad i den här storleken (men med ca 40 sidor i stället för dessa 12). Hur sidorna kan se ut syns på följande broschyrsidor, där vi dock förminskat de ursprungliga sidorna till hälften.

Utom den här föreningstidningen har vi också ett programbibliotek, där man kan köpa färdiga beräkningsprogram inom olika områden för 15 kr/st plus en avgift per beställning av 10 kr. Biblioteket har uppstått dels inom vår egen förening genom att den som sänder in ett användbart program får två andra i utbyte, dels genom samarbete med andra klubbar i Europa och USA. De svenska programmen går i regel också att köpa inspelade på magnetkort (för TI-59), då för 30 kr/st.

För medlemmar i Stockholmsområdet har vi börjat anordna träffar där vi kan diskutera programmering.

Medlemsavgiften har vi satt till 120 kr/kalenderår. Du blir medlem genom att sätta in detta belopp på vårt postgiro 430 01 59 - 3. Som medlem kan Du också beställa gamla årgångar av Programbiten och en del böcker samt magnetkort enligt prislista nedan.

Med bästa hälsningar
FÖRENINGEN PROGRAMBITEN

Lars Hedlund

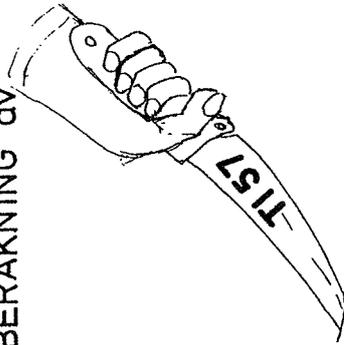
Lars Hedlund, ordf

FÖRENINGENS TILLBEHÖRSFÖRSÄLJNING

Som medlem kan Du köpa nedanstående från föreningen:

Sourcebook for Programmable Calculators (restupplaga)	80,-
Programbiten, senaste kompletta årgång	100,-
"- , näst "-	80,-
"- , tidigare årgångar (1. årg. 1978/79) pr st	60,-
"- , tre årgångar 1978/79 - 1981	200,-
Patenthandlingar TI-59	25,-
Astronomical Formulae for Calculators (viss leveranstid)	55,-
40 st tomma magnetkort och plånbok	108,-
Tom magnetkortsplånbok	10,-
Föreningens programmeringsblanketter, olika typer, block om 50 blanketter	11,-

BERÄKNING AV



för 2 CIRKLAR av INGVAR MAGNUSSON

Med det här programmet kan man beräkna skärningspunkterna för två cirklar.

Programmet använder alla data-register utom register 6.

Om man försöker att beräkna skärningspunkterna för cirklar som inte skär eller tangerar varandra, indikeras detta med att exekveringen stoppas och att sifferindikatorn blinkar.

Bruksanvisning:

Mata in programmet enligt listan.

Lägg origo för x/y-axlarna i centrum för R₁.

Mata in Tryck på Sifferindikatorn visar/

R ₁	STO 0	Sifferindikatorn visar/
R ₂	STO 1	
X	STO 2	
Y	STO 3	
	RST	
	R/S	X ₁
	X↵	Y ₁
	R/S	X ₂
	X↵	Y ₂

Exempel:

R ₁ = 40	STO 0		
R ₂ = 29	STO 1		
X = 20	STO 2		
Y = 10	STO 3		
	RST		
	R/S	12.472054	X ₁
	X↵	38.005893	Y ₁
	R/S	37.887946	X ₂
	X↵	-12.825893	Y ₂

DECIMALPUNKTTEKNIKET

Decimalpunkttekniken är ett sätt att göra pgn användbar-
vänliga. Det används flitigt i TI-57C NOTIS.

För att visa knopet när jag föreställt "STO 0C 1CL
Jag läser från förtä numret. Direkt efter var-
je label har jag placerat ". ". Det får den effekten att
ett tal på skärmen, som är resultatet av en beräkning,
redas. Övrigt, om man nyss har matat in ett tal,
blir det kvar.

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

Ejörn Gustavsson, Smedjebacken

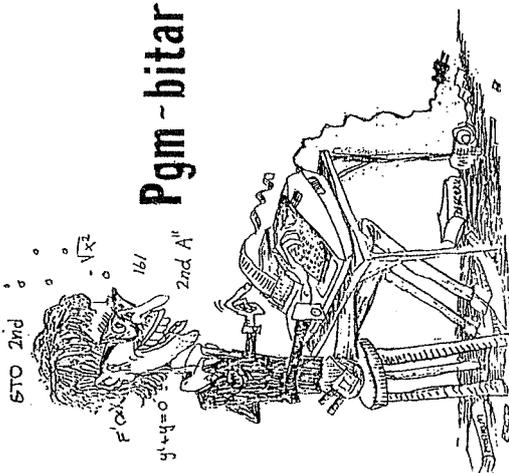
0000	76 LBL	012	76 LBL
0001	11 A	013	12 B
0002	93	014	93
0003	29 CP	015	29 CP
0004	67 EQ	016	67 EQ
0005	00 00	017	00 00
0006	09 09	018	21 21
0007	42 STO	019	42 STO
0008	01 01	020	02 02
0009	43 RCL	021	43 RCL
0100	01 01	022	02 02
0101	32 RTN	023	92 RTN

Sortera efter storlek

Att sortera siffervärden i storleksordning är ganska
tidsödande på en programmerbar räknare. I matematik-
modulen finns ett program MU-06 där själva sorterings-
rutinen upptar 100 programsteg. Här visas ett enklare
program på 40 programsteg som i gengäld är långsammare.
Att sortera 30 värden tar med detta drygt sex minuter
neden MU-06 gör det på ganska precis fyra, och det är
att vänta att skillnaden blir större med fler värden
båda.) Listningen nedan klarar 57 värden som skall
ligga i minnena 01 - 57. Om "59" ändras till "99" och
"58" till "98" går det med 97 värden (uppsanning "10
op 17"). Programmet startas med antal värden fyllt av A.

0000	76 LBL	021	77 GE
0001	11 H	022	00 00
0002	75	023	32 32
0003	01 I	024	59 DP
0004	95 =	025	63 EX*
0005	42 STO	026	00 00
0006	59 59	027	00 00
0007	01 I	028	59 DP
0008	42 STO	029	20 20
0009	00 00	030	72 ST*
0010	43 RCL	031	00 00
0011	59 59	032	97 DS2
0012	42 STO	033	58 58
0013	58 58	034	00 00
0014	73 RC*	035	16 16
0015	00 00	036	97 DS2
0016	32 X↵	037	59 59
0017	69 DP	038	00 00
0018	20 20	039	07 07
0019	73 RC*	040	91 R/S
0020	00 00		

Pgm-bitar



"ROBERT SHOWS BERÖMDA SKRIVKODSONVANDLARE"

Nedanstående program för att översätta upp
till femsiffriga tal till skrivkod är tydlig-
gen sedan länge känt bland programmerare i
USA. Den här versionen är desuout förkortad
med två steg men går trots detta på 6!
Observera direktadresseringen trots att
programmet startas med "SBR PRT". (Från TI
PFC NOTES V5 M1 1980)

0000	76 LBL	015	85 +	030	01 01
0001	99 PRT	016	01 I	031	95 =
0002	29 CP	017	85 +	032	65 x
0003	67 EQ	018	28 LOG	033	01 I
0004	00 00	019	59 INT	034	00 00
0005	41 41	020	65 x	035	97 DS2
0006	95 +	021	01 I	036	08 08
0007	28 LOG	022	00 00	037	00 00
0008	59 INT	023	00 00	038	15 15
0009	42 STO	024	49 PRD	039	25 CLR
0010	08 08	025	01 01	040	48 EXC
0011	69 DP	026	02 2	041	01 01
0012	28 28	027	75 -	042	92 RTN
0013	22 INV	028	59 INT		
0014	28 LOG	029	44 SUM		

Följande variant är 11 steg längre men skriver
även upp till fyrsiffriga tal med minustecken:

0000	76 LBL	018	28 LOG	036	01 01
0001	99 PRT	019	59 INT	037	02 2
0002	29 CP	020	42 STO	038	75
0003	67 EQ	021	08 08	039	59 INT
0004	00 00	022	69 DP	040	44 SUM
0005	52 52	023	28 28	041	01 01
0006	32 X↵	024	22 INV	042	95 =
0007	22 INV	025	28 LOG	043	65 x
0008	77 GE	026	85 +	044	01 I
0009	00 00	027	01 I	045	00 00
0010	12 12	028	85 +	046	97 DS2
0011	02 2	029	28 LOG	047	08 08
0012	00 00	030	59 INT	048	00 00
0013	42 STO	031	65 x	049	26 26
0014	01 01	032	01 I	050	25 CLR
0015	32 X↵	033	00 00	051	43 RCL
0016	50 I+	034	00 00	052	01 01
0017	55 +	035	49 PRD	053	92 RTN

3-D LUFFARSCHACK FAST MODE

I PB 78-2 publicerades ett program för tredimensionellt luffarschack. Programmet stöds också av exekverings- tigen, som kan vara upp till fem minuter. Genom att skriva om pöset så att det kan köras i fast mode kan exekverings- ringstiden nedringas till hälften.

Programbeskrivning: Några stora förändringar har inte gjorts. Originalprogrammet är så skickligt skrivet att jag inte gjort några försök att optimera det. Genom att flytta några steg och ta bort de onödiga L01 A och L01 E samt att RW fick jag plats med nödvändiga tillägg. Det blev tom två programsteg över.

För att även utnyttja dessa sätter jag in Adv Pst vid steg 006 för att användarens drag skall skrivas ut. Initieringen för Fast mode ligger förstås på en särskild kortslås (block 1), eftersom den rutinen har 48 steg. Den rutinen skrivs över när det riktiga programmet i block 1 läses in. Pgnatagen 040 - 047 (av initieringsrutinen) blir kvar efter inläsning av block 1 tills GT0 ind 7 har exekverats. (Se PB 78-2 s 21) Därefter ersätts även dessa steg med det "riktiga" programmet.

Inmatning av programmet: Fem kortslås skall matas in. Den som har matat in Luffarschackspillet i PB 78-2 slippar göra om den jobbigaste delen att ladda registren 04 - 89. Använd bara block 2, 3 & 4 av det programmet. Block 1 av det programmet måste bytas ut mot det nya "huvud- programmet", och ett tredje magnetkort måste spelas in med initieringsprogrammet.

1. Slå av och på alla på räknaren.
2. Mata in initieringspöset (48 steg) och spela in på magnetkort. Märk kortslåsen som inspolningen skedde på med "1" (block 1a).
3. Tryck CP 9 Op 17.

4. Mata in huvudprogrammet. (240 steg)
5. Spela in block 1 på magnetkort (den som köpt in originalversionen spelar lämpligen in det på originalversionens block 1, ova spelar över det gamla programmet.)
Den som har originalversionen hoppar över resterande inmatningsinstruktioner.

6. Det enklaste sättet att ladda R10 - R89 med de 12- siffriga talen (se registerlistningen) är att använda Inmatningsprogrammet. Så tryck CP och mata in det.
Tryck A. Mata in 9 av de första siffrorna som skall lagras i R10 och tryck R/S. (4. är redan inmatat av programmet.) Mata in resterande siffror och tryck R/S.
Gör nu likadant med övriga register. Ladda slutligen R4, R5, R6 och R8. Spela in block 2, 3 & 4 på magnetkort.

KORREKTUR AV PROGRAMMET:

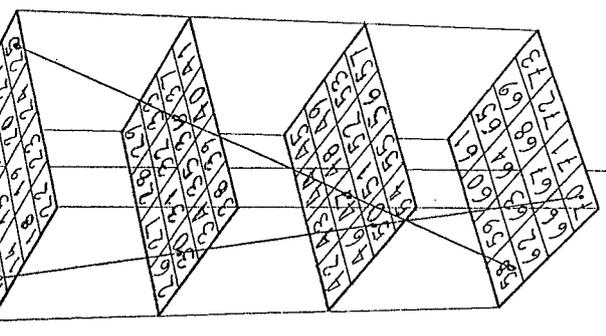
- Läs in block 1a. Tryck A.
- Läs in block 1a igen. 4 visas.
- Läs in block 3, 2 visas.
- Läs in block 2, 0 visas, är skrivare ansluten matas papperet fram ett steg.
- Läs in block 4, 3 visas.
- Om du vill att räknaren skall börja tryck R/S.
- Om du vill börja mata in ditt startdrag och tryck R/S.
- I båda fallen visas 1. Läs in block 1. Om du valde att börja skrivs ditt startdrag ut, annars skriv 0, ut.
- Efters att användarens drag och skrivs mataraget (eller start- draget om räknaren skulle börja) ut.
- Mata in ditt drag och tryck R/S. Är skrivare ansluten skrivs det ut. Därefter visas motdraget (skrivs ut om skrivare är ansluten).
- Upprepa steg 9 tills du får ett blinkande tal på skärmen. Då har du eller räknaren vunnit (främjare av tallet).
- Skulle du mata in numret på en upptagen ruta visas 0.5 (och trycks med skrivare ansluten). Du kan slå in ett nytt rut-nr.
- För ny omgång upprepa från steg 1.

0,252215	04	4,29356638842	48
1,	05	4,47433328253	49
0,3791262215	06	4,47041218171	50
0,1304933324	08	4,38258678383	51
4,38581807466	10	4,38578678383	52
4,48067291673	11	4,47591332488	53
4,48067291673	12	4,45047321865	54
4,3828767272	13	4,47343322364	55
4,47963272172	14	4,47238322262	56
4,4815321167	15	4,47851333259	57
4,4843211724	16	4,38476718139	58
4,48234262125	17	4,46839291471	59
4,47963251355	18	4,46639291471	60
4,45349161342	19	4,3837747874	61
4,48152171346	20	4,46259271063	62
4,46434241318	21	4,4615141066	63
4,3837723073	22	4,45446121067	64
4,4684231672	23	4,4634428107	65
4,466422177	24	4,4649231561	66
4,38481747967	25	4,45346151473	67
4,47463130663	26	4,4615215126	68
4,4773313963	27	4,46544241572	69
4,4763313861	28	4,38273707671	70
4,455331196	29	4,46960231459	71
4,47040307339	30	4,46760231274	72
4,3826723958	31	4,4797705762	73
4,4754261958	32	4,4757572751	74
4,4714300526	33	4,4615051444	75
4,3826723958	34	4,4603950459	76
4,3826723958	35	4,4765443282	77
4,3826723958	36	4,4564643282	78
4,3826723958	37	4,4730143282	79
4,4754261958	38	4,4718054254	80
4,4585301157	39	4,4613954396	81
4,4738221134	40	4,46259433571	82
4,4738221134	41	4,46259433571	83
4,4738221134	42	4,45874434174	84
4,4738221134	43	4,45874434174	85
4,4738221134	44	4,452484536	86
4,4738221134	45	4,452484536	87
4,4738221134	46	4,454494037	88
4,4738221134	47	4,454494037	89
4,4738221134	48	4,458544036	90

INITIERINGS-PROGRAM
000 08 4
001 00 0
002 00 0
003 00 0
004 00 0
005 00 0
006 00 0
007 00 0
008 00 0
009 00 0
010 00 0
011 00 0
012 00 0
013 00 0
014 00 0
015 00 0
016 00 0
017 00 0
018 00 0
019 00 0
020 00 0
021 00 0
022 00 0
023 00 0
024 00 0
025 00 0
026 00 0
027 00 0
028 00 0
029 00 0

HJULPROGRAM FÖR INMÄTNING I MINNEN
000 7h L01
001 11 A
002 34 CE
003 42 ST
004 42 ST
005 00 0
006 48 DP
007 48 DP
008 48 DP
009 00 0
010 66 RW
011 66 RW
012 04 4
013 93 3
014 34 R/S
015 91 P/S
016 52 EE
017 52 EE
018 52 EE
019 52 EE
020 01 1
021 00 0
022 94 4
023 94 4
024 32 11
025 00 0
026 00 0
027 00 0
028 00 0
029 00 0

HUVUDPROGRAM
080 01 01
081 00 00
082 00 00
083 00 00
084 00 00
085 00 00
086 00 00
087 00 00
088 00 00
089 00 00
090 00 00
091 00 00
092 00 00
093 00 00
094 00 00
095 00 00
096 00 00
097 00 00
098 00 00
099 00 00
100 00 00
101 00 00
102 00 00
103 00 00
104 00 00
105 00 00
106 00 00
107 00 00
108 00 00
109 00 00
110 00 00
111 00 00
112 00 00
113 00 00
114 00 00
115 00 00
116 00 00
117 00 00
118 00 00
119 00 00
120 00 00
121 00 00
122 00 00
123 00 00
124 00 00
125 00 00
126 00 00
127 00 00
128 00 00
129 00 00
130 00 00
131 00 00
132 00 00
133 00 00
134 00 00
135 00 00
136 00 00
137 00 00
138 00 00
139 00 00
140 00 00
141 00 00
142 00 00
143 00 00
144 00 00
145 00 00
146 00 00
147 00 00
148 00 00
149 00 00
150 00 00
151 00 00
152 00 00
153 00 00
154 00 00
155 00 00
156 00 00
157 00 00
158 00 00
159 00 00
160 00 00
161 00 00
162 00 00
163 00 00
164 00 00
165 00 00
166 00 00
167 00 00
168 00 00
169 00 00
170 00 00
171 00 00
172 00 00
173 00 00
174 00 00
175 00 00
176 00 00
177 00 00
178 00 00
179 00 00
180 00 00
181 00 00
182 00 00
183 00 00
184 00 00
185 00 00
186 00 00
187 00 00
188 00 00
189 00 00
190 00 00
191 00 00
192 00 00
193 00 00
194 00 00
195 00 00
196 00 00
197 00 00
198 00 00
199 00 00



Här nedan ett program i fast mode för en utskrift av en "riktig" spelplan (betr. "Fuskplaner", se följande sida). Knappa in programmet och ladda minnena 90 - 94 (använd hjälpprogrammet upptill t V, "10 p 17" måste förstås ha tryckts). Allt sammans rymns på en kortslås (återgå till normaluppläsning med "6 p 17" före inspelning). Bruksanvisning: Läs kortet, tryck A. Läs kortet igen så startar utskriften i fast mode. För jämförelse kan Du trycka RST R/S och se hur fort det går med "normal" fart.

10020202020	2
10000,2020202	91
10000510000,51	92
1000000510000	93
100510000,51	94

KALENDER T157

av Ingvar Magnusson

Med det här programmet kan man beräkna antal dagar mellan två datum, efter år 1582. Beräkningarna är baserade på den gregorianska kalendern.

Man kan också själv, lätt bestämma veckodagen för ett datum med hjälp av faktorn F för detta datum. Veckodag = F - (INT(F/7))17.

Veckodagen representeras av en siffra:
 0 = Lör 4 = Ons
 1 = Sön 5 = Tor
 2 = Mån 6 = Fre
 3 = Tis

Exempel:

Beräkna antal dagar mellan 17 mars 1889 och 3 maj 1985 och veckodag för det första datumet.

Tangent	Rad Kod	365	STO 5 (konstanter)
RCL 1	00 33 1	2.3	STO 6
X	01 55		
RCL 5	02 33 5	1889	STO 1 År
=	03 85	3	STO 2 Månad
SUM 3	04 34 3	17	STO 3 Dag
3	05 03	RST	
1	06 01	R/S	690019 F ₁
INV SUM 3	07 -34 3	STO 0	690019
X	08 55		
RCL 2	09 33 2	1985	STO 1 År
STO 7	10 32 7	5	STO 2 Månad
=	11 85	3	STO 3 Dag
SUM 3	12 34 3	RST	
1	13 01	R/S	725129 F ₂
INV SUM 1	14 -34 1	-	
2	15 03	RCL 0	690019
X	16 76	=	35110 Dagar
STO 1	17 51 1		
1	18 01	RCL 0	690019
SUM 1	19 34 1	+	
.	20 83	7	
4	21 04	=	98574.143
Prd 2	22 39 2	Int	98574
RCL 6	23 33 6	X	
SUM 2	24 34 2	7	
RCL 2	25 33 2	+/-	-7
INT	26 49	+	
INV SUM 3	27 -34 3	RCL 0	690019
LBL 1	28 86 1	=	1 (= söndag)
4	29 04		
INV Prd 1	30 -39 1		
RCL 1	31 33 1	X	
INT	32 49	.	41 55
SUM 3	33 34 3	7	42 83
2	34 02	5	43 07
5	35 05	7	44 05
INT	36 -39 1	=	45 85
1	37 01	INT	46 49
SUM 1	38 34 1	INV SUM 3	47 -34 3
RCL 1	39 33 1	RCL 3	48 33 3
INT	40 49	R/S	49 81

Markus har också skickat ett snabbare program (pgm 4) till problem 13. Metoden är den samma som i mitt pgm 6 i P8 81-3, men den är en loop, som genomlöps väldigt många gånger, är betydligt kortare.

Det kan kanske vara på plats att förklara metoden som används. Den kallas "inricks-metoden" i fri översättning och går till så att ett register i taget placeras in i den sorterade listan. Den listan består från början av ett register. Efter första insättningen består således listan av två register. Ett tredje register jämförs med listan och placeras in på rätt ställe. Så fortsätter det tills alla register finns på rätt ställe i den sorterade listan.

Här nedan visas ett exempel på hur sortering går till. Talen till vänster om kolonet bildar den sorterade listan. Talet precis till höger om kolonet är nästa tal som skall sättas in i listan. Pilens markerar var insättningen skall göras. Talen inom parentes är den helt osorterade listan.

503	087	(512	061	908	170	987)
087	503	(061	908	170	987)	
087	503	512	(061	(908	170	987)
061	087	503	512	908	(170	987)
061	087	503	512	908	(170	(987)
061	087	170	503	512	908	987
061	087	170	503	512	908	987

Programmen söker uppifrån i den sorterade listan tills de finner ett tal som är mindre än eller lika med tal som skall sorteras in och sätter in det. Samtidigt som rätt ställe söks, flyttas registren om, så när rätt ställe har hittats, finns det ett tomt register att placera talet i.

000	76	LEL	008	00	0	012	76	LEL	024	01	02
002	25	CLR	010	00	0	012	31	LE	027	11	FE
003	01	1	011	00	0	019	00	00	027	08	66
005	93	5	012	00	0	030	24	INT	028	24	CE
006	00	0	014	42	STO	032	00	00	030	63	PIH
007	00	0	015	00	00	023	36	PIH			

Pgm 3. (Red.) Problem 11: Exempel på anrop av pgm 2.

000	76	LEL	012	02	HIR	024	73	RC	036	00	00
002	81	HIR	013	17	011	027	01	01	037	31	01
003	04	04	015	01	11	030	31	01	038	31	01
004	03	3	016	05	0	038	30	30	040	21	11
005	34	734	018	00	00	039	00	00	041	21	11
006	34	734	019	46	46	031	21	01	043	21	01
007	85	7	020	02	STO	032	01	01	044	00	00
008	29	024	021	42	STO	034	31	01	047	31	01
010	42	002	023	01	01	035	11	01	047	31	01

UTMANINGEN!

Redaktionsadress:
 Björn Gustavsson
 Nordlandervägen 12 A
 777 00 SHEDJEBACKEN

Haj.
 Markus Sveinn Markusson har skickat in ett program för DSE-funktion (problem 11); se pgm 1. Det använder T-registret och två vilande operationer. Före användning skall R₀ laddas med ett tal på formen ttttt.sssssfff, där alltså ttttt är talet som räknas ned, sssss är stopptal och fff förändringen. ttttt kommer att minskas med fff ända tills ttttt ≤ sssss.

Gör så här för att prova rutinen: Mata in programmet. Gå sedan till steg 000. Skjut där in den rutin som skall ingå i DSE-loop-en. Som demonstration är RCL 00 INT PRI lämpligt. Lagra sedan ett tal i R₀ och tryck RST R/S.

Den här rutinen kan inte användas som sub-rutin i t ex. en modul. Därför skrev jag pgm 2 efter Markus' metod men gjorde om rutinen så att flagga 6 nollställs när loopen är färdig och ställer flaggan annars. Om vi antar att rutinen används i en modul och har pgm-nummer 02 kan den anropas som i pgm 03.

Den som kommer på något smartare sätt att ordna samordning mellan huvudprogram och modulprogram, var vänlig skriv till utmaningen och berätta.

000	55	1	008	01	5	017	55	INT	024	57	INT
002	00	00	010	23	LDC	018	03	3	026	00	00
003	15	1	011	33	INT	020	00	00	027	00	00
005	21	INT	013	00	0	021	54	7	028	92	PIH
006	55	INT	014	54	7	022	43	STO			
007	85	7	015	23	INT	023	00	00			

Pgm 1. (Markus S. Markusson.) Problem 11: DSE-funktion.

000	76	LEL	008	05	2	018	52	INT	027	77	DE
002	53	7	011	32	INT	020	01	3	028	31	51
003	43	FCL	012	28	LDC	021	23	INT	030	23	INT
004	25	INT	022	28	LDC	021	28	INT	031	86	PIH
005	25	INT	023	28	LDC	021	28	INT	033	92	PIH
006	33	1	015	00	0	024	43	STO	033	92	PIH
008	52	INT	017	23	INT	025	00	00	026	55	INT

REGULA FALSI

TI 57 RUNGE KUTTA

BINOMIALFÖRDELNING

Nedanstående tre TI-57 program är ett bidrag från Anders Lundberg som är lärare vid Kungl Militärhögskolan i Stockholm.

Regula Falsi-programmet, är förmodligen i alla avseenden bättre än det ekvationslösningsprogram av Per Eric Holmberg, som publicerades i Fgb 79:1. Det är ungefär lika långt (kort), man behöver inte välja något k-värde, det konvergerar för det flesta x_1, x_2 om $x_2 < x_1$, det konvergerar så snabbt att maskinens hela programmet är utnyttjad efter (vanligtvis) 5 a*10 iterationer, alla rötter kan fås genom lämpliga val av x_1, x_2 .

Ni hade skrivit om hans program till TI-57, och efter som Ni skrev RCL 0 X² X RCL 0 i st.f RCL 0 X X² trodde jag inte att Ni kände till denna möjlighet. Subrutinen kan totalt göras åtminstone 2 steg kortare.

TI-57 är roligare än mer avancerade maskiner, Just på grund av sina begränsningar. Det är en sport att försöka utnyttja dess möjligheter så effektivt som möjligt. När det gäller Regula Falsi och Runge Kutta programmet, känner jag mig faktiskt ganska nöjd med mitt sätt att utnyttja dosan på bästa möjliga sätt.

A.L.

Regula Falsi.

Om x_1 och x_2 är olika närmevärderna av en rot till ekvationen $f(x) = 0$, så beräknas ett bättre närmevärde x_3 medelst formeln

$$x_3 = x_2 - \frac{x_1 - x_2}{\frac{f(x_1)}{f(x_2)} - 1}$$

Före körning lagras:

x_1 i minne 6

x_2 i minne 7

(x_2 lämpligen det bättre närmevärdet)

Resultat av en körning:

x_2 i minne 6

x_3 i minne 7 och i fönstret

Exempel:

Minne 0-4 är lediga för eventuella subroutiner.

Subrutinen för beräkning av $f(x)$ måste börja på rad 17 med 2nd lbl 0 STO 5

d.v.s x lagras i minne 5.

$$f(x) = x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

Då blir forts. + RCL 4 = x RCL 5 + RCL 3 = x RCL 5 + RCL 2 = x RCL 5 + RCL 1 = x RCL 5 + RCL 0 = INV SBR.

a_k lagras i minne 1 före körning.

Dosan kan rymma även 6:e-grads polynom, men då måste en koefficient skrivas i själva programmet.

R/S på rad 15 kan ersättas med pause eller utgå. Då stannar exekveringen vid division med med noll p.g.a $x_1 = x_2$, som kan återkallas ur minne 5 och är då slutresultatet.

Regula Falsi

```
00 33 6 RCL 6
01 61 0 SBR 0
02 22 X↑T
03 -34 6 INV SUM 6
04 61 0 SBR 0
05 -39 7 INV PRD 7
06 01 1
07 -34 7 INV SUM 7
08 -33 7 RCL 7
09 -33 6 INV PRD 6
10 33 5 RCL 5
11 85 6 EXC 6
12 33 6 RCL 6
13 85 7 STD 7
14 32 7 STD 7
15 81 R/S
16 71 RST
17 33 0 LBL 0
18 32 5 STD 5
19 75 +
20 33 4 RCL 4
21 85 =
22 55 X
23 33 5 RCL 5
24 75 +
25 33 3 RCL 3
26 85 =
27 55 X
28 33 5 RCL 5
29 75 +
30 33 2 RCL 2
31 85 =
32 55 X
33 33 5 RCL 5
34 75 +
35 33 1 RCL 1
36 85 =
37 55 X
38 33 5 RCL 5
39 75 +
40 33 0 RCL 0
41 85 =
42 -61 INV SBR
```

Runge Kutta.

Approximativ lösning av $y' = f(x,y)$, då $y(x_0) = y_0$ y-värdena bestäms rekursivt med formeln:

$$y_{i+1} = y_i + h \cdot f(x_i, y_i), \quad i = 0, 1, 2, \dots, n-1, \text{ där}$$

$$h = 1/6(k_1 + 2k_2 + 2k_3 + k_4)$$

$$= 1/3(k_1 + k_2 + k_3 + k_4) / (2 + 2k_2/2 + k_4/2), \text{ och}$$

$$k_1/2 = h/2f(x_i, y_i)$$

$$k_2/2 = h/2f(x_i + h/2, y_i + k_1/2)$$

$$k_3 = 2h/2f(x_i + h/2, y_i + k_2/2)$$

$$k_4/2 = h/2f(x_i + h, y_i + k_3)$$

Minne	innehåller storheten	lagras före körning
0	x_1	x_0
1	y_1	y_0
2	$h/2$	$h/2$

4-6 lediga för SBR 1

7 x_n

$h/2$ måste väljas så att $(x_n - x_0)/h$ är ett heltal.

Om beräkningen av $f(x,y)$ kräver fler än 13 steg, kan rad 22-23 bytas ut mot t.ex RCL 1, 2nd pause. Då utnyttjas heller inte x_n , så minne 7 blir också ledigt.

Runge Kutta

```
00 00 0
01 61 1 SBR 1
02 34 0 SUM 0
03 85 =
04 32 3 STD 3
05 61 1 SBR 1
06 85 =
07 34 3 SUM 3
08 34 3 SUM 3
09 61 1 SBR 1
10 34 0 SUM 0
11 55 X
12 02 2
13 05 2
14 34 3 SUM 3
15 61 1 SBR 1
16 85 =
17 34 3 SUM 3
18 03 3
19 -39 3 INV PRD 3
20 33 3 RCL 3
21 34 1 SUM 1
22 33 0 RCL 0
23 66 ED
24 71 RST
25 32 1 RCL 1
26 86 2 LBL 2
27 33 1 RCL 1
28 81 P/S
29 71 RST
30 86 1 LBL 1
31 86 1 LBL 1
32 32 1 RCL 1
33 85 =
34 33 2 RCL 2
35 85 X
36 33 5 RCL 5
37 85 =
38 -61 INV SBR
```

Nu är x i minne 0 och y i fönstret. Beräkna $f(x,y)$ och sätt i fönstret. Remsen t.h. är en förbättring föreslagen av Björn Gustavsson. "13" på fjärde textraden nedifrån kan då ändras till

Binomialfördelning.

Programmet beräknar $\sum_{k=a}^b f(k)$, där

$$f(k) = \binom{n}{k} p^k q^{n-k}, \quad (q = 1-p).$$

I subrutin 4 utnyttjas rekursionsformeln:

$$f(k-1) = f(k) \cdot \frac{k}{n-k+1} \cdot \frac{q}{p}$$

I minne lagras insättes

0 k

1 n

2 p

3 q/p

4 f(k)

5 summan

6 a

7 b-1, a b

Efter körning står summan i fönstret. Innehållet i minne 7 ändras vid körningen. Om du vill köra programmet på nytt med samma b, måste b sålunda sättas in på nytt. n, p och a är dock kvar i resp. minne efter körning.

Binomialfördelning

```
00 01 1
01 32 3 STD 3
02 34 7 SUM 7
03 33 2 RCL 2
04 -34 3 INV SUM 3
05 -39 3 INV PRD 3
06 35 YX
07 33 1 RCL 1
08 32 0 STD 0
09 85 =
10 32 4 STD 4
11 00 0
12 32 5 STD 5
13 86 1 LBL 1
14 33 0 RCL 0
15 76 GE
16 51 2 STD 2
17 33 6 RCL 6
18 32 7 STD 7
19 34 3 LBL 3
20 33 4 RCL 4
21 34 5 SUM 5
22 31 0 RCL 0
23 53 0 RCL 0
24 76 GE
25 51 3 STD 3
26 33 5 RCL 5
27 81 R/S
28 71 RST
29 86 2 LBL 2
30 61 4 SBR 4
31 51 1 STD 1
32 86 4 LBL 4
33 33 0 RCL 0
34 39 4 PRD 4
35 33 3 RCL 3
36 39 4 PRD 4
37 01 1
38 -34 0 INV SUM 0
39 33 1 RCL 1
40 35 1 RCL 1
41 32 0 RCL 0
42 83 =
43 -33 4 INV PRD 4
44 -41 INV SBR
```

MR MAGIC

av Robert & Richard Snow
ur TI PPC Notes v5n3p10
översättning Björn Gustavsson

Som alla andra kortkonster, kräver även den här illusionen lite fingerfärdighet. 59:an själv står för fingerfärdigheten. Du talar om att din räknare kan lista ut alla kort. Till de åhörarnas förvåning utför räknaren den prestationen med ungefär sex gissningar. Vad det stackars "offret" inte inser är att det är du som sätter räknaren på rätt spår.

När du trycker B, för BÖJA, skriver skrivaren: READY? PICK A CARD - alltså en uppmaning att välja ett kort. "Offret" plockar upp ett ur en kortlek och lägger det på bordet med rättsidan upp. Titta nu på displayen och lägg märke till att den blinker omväxlande 9999 och 9999. (med decimalpunkt). När du trycker ned R/S och håller den nere i ungefär en sekund finns det alltid två möjliga utfall, beroende på om niorna följs av en decimalpunkt eller inte. Men det vet bara du!

Ditt offer, även någon som känner till 59:an, lägger inte alltid märke till den lilla skillnaden. Du kan enkelt förutsäga vad skrivaren kommer att skriva, genom att följa, och förhoppningsvis lära utantill, det binära trädet nedan. De stora svarta punkterna anger närvaron av decimalpunkt. Således får du BLACK första gången du trycker R/S med decimalpunkt. (Övers. anm.: På slutet kommer en översättning av engelska korttermerna.) Nästa gång blir resultatet SPADE.

Från och med nu betyder decimalpunkt höga kort. Med lite övning kommer du att märka att det är väldigt enkelt att erhålla rätt resultat. Men om du gör ett misstag - och det gör ju alla då och då - misströsa intel Detta djävulska program har en fantastisk förmåga att släta över fel: Om du gör en tavlå, och skriver fel färg eller kort, börja prata snabbt: "Konstigt, jag måste fortfarande kontrollera det här programmet. Ibland skriver det ut motsatsen. Kanske, den... låt se..." Och se, programmet skriver I CHANGED MY MIND ("Jag ändrade mig") och skriver sedan motsatsen till sin gissning!

Den här felbehandlingen äger rum var tionde sekund, och bör för trovärdighetens skull inte missbrukas. Det betyder att du bara har ungefär tio sekunder på dig varje gång för att ge rätt resultat, så öva och sätt fart!

Åhörarna är vanligen förbluffade över hur räknaren klarar av det. De misstänkte något sorts knep förstås, men få lyckas peka ut det. Några av mina offer höll ögonen på mina fötter: de kanske trodde att jag styrde räk-naren med fotpedaler! Andra bad mig till och med att inte upprepa högt valören och färgen på kortet. Frodde de verkligen att 59:an kunde de höra det?

1. Tryck B för att starta. READY? PICK A CARD skrivs.
2. Offret väljer ett kort.
3. Tryck ned R/S inom tio sekunder.
4. Tryck A för att fortsätta. Skrivaren skriver räknarens (eller är det ditt?) val.
5. Gå tillbaka till 3 tills hela kortet är avslöjat.

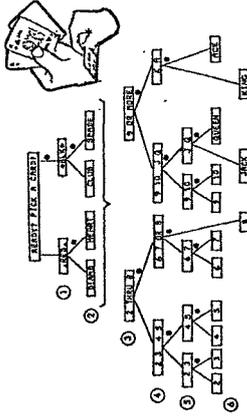
Översättarens not: Programmet har gjorts av de i TI PPC mycket berömda bröderna Snow. Som vanligt är programmet välprogrammerat, med en del knep. Observera t ex skrivkoden i R55. Om ett skrivregister laddas med innehållat med hjälp av OP nr blir resultatet "I", men laddas det med HIR nr blir resultatet "MIND"! Detta underlättar vid utskrift av I CHANGED MY MIND. Bara tre register behövs återkallas (se steg 012-027).

Det finns också en rutin i programmet så att kortet skall kunnas listas ut med färre antal gissningar. Men då måste man trycka A', och så ser ju åhörarna att man trycker på olika tangenter och förstår direkt att det är något fifvel på gång.

Lars Redlund föreslog en kanske bättre metod: Ett pekminne räknas upp samtidigt med displayväxlingen. Det pekar på t ex tre register som alla innehåller 9999. (med punkt). I display ser man då en gång 9999 utan punkt och tre gånger 9999 med punkt. Genom att räkna 9999. (med punkt) klarar man alltså fyra valmöjligheter per R/S. Man klarar då rätt färg direkt och valören i tre 'gissningar'.

Läarna inbjuds skicka in program som bygger på den principen. Det måste gå att ångra sig även i det programmet. (Skicka programmen till utmaningen.)

Förklaring av engelska korttermerna: Färgerna heter Diamond (förkortat DIAMD), Heart, Club och Spade, vilket alltså motsvarar ruter, hjärter, klöver och spader. Knekt heter Jack, dam heter Queen, kung heter King och ess heter Ace.



121.4571003324	30	11.00013003235	45
119.13248013	31	11.00089223517	46
3.007535171675	32	15.00013000201	47
-25.01624133016	33	15.000250034	48
-25.02341339337	34	30.000480013	50
3.0031143272651	35	25131326.	51
-25.00015274114	36	3441171931.	52
-25.02341339337	37	328293KING	53
11.000033003723	38	13151700.	54
11.03341000973	39	2400.032343116	55
15.000330004	40	152313312.	56
15.000050006	41	1714003045.	57
15.000070008	42	35171316.	58
15.032350009	43	1513351573.	59

000	76	LBL	049	11	09	54	DF	145	55	7
002	59	INT	090	59	DF	098	85	10	146	05
003	09	9	053	10	10	099	43	RCL	147	18
004	09	9	053	01	04	101	75	4	148	02
006	09	9	054	32	XIT	102	08	3	150	95
007	03	03	057	03	03	103	25	9	152	47
009	03	03	057	03	03	105	59	PR	153	05
010	00	00	059	03	PCL	106	75	LBL	154	44
012	43	RCL	060	32	STO	108	25	CLR	156	82
013	55	75	061	06	06	109	43	PCL	157	04
015	08	08	062	03	09	110	28	DF	158	73
016	69	DF	064	22	...	112	01	01	160	57
018	43	RCL	065	03	04	113	63	DF	161	71
019	56	56	066	76	LBL	115	64	DF	163	04
020	69	DF	068	13	CF	116	04	04	164	05
022	43	RCL	070	02	02	118	02	03	166	17
023	57	57	071	82	HIF	119	03	3	167	44
024	02	02	073	01	05	120	42	STO	169	01
026	69	DF	074	76	LBL	122	00	00	170	01
028	05	05	075	17	DF	123	42	STO	171	32
030	00	00	077	04	05	125	16	CF	173	05
032	43	RCL	078	73	DF	126	85	...	174	22
034	00	00	080	22	INT	128	87	DF	176	01
036	06	06	082	06	06	129	58	...	177	05
038	42	STO	083	00	00	130	05	55	178	04
040	42	STO	084	00	00	131	46	DF	180	44
042	00	00	085	32	XIT	132	08	8	181	05
044	42	STO	086	42	STO	133	18	CF	182	03
046	44	RCL	087	01	01	134	06	DF	183	05
048	75	...	088	24	DF	135	44	SUM	184	69
050	44	RCL	089	16	DF	136	09	DF	185	03
052	54	DF	090	16	DF	137	63	...	186	05
054	16	RCL	091	16	DF	138	41	SUM	187	05
056	76	LBL	092	31	DF	139	54	SUM	188	12
058	24	DF	093	24	DF	140	03	05	189	05
059	24	DF	094	09	09	141	31	DF	190	03

Anders Perssons idé från förra numret om ett program för att beräkna kostnad för telefon-samtal gjorde många intresserade. Det presenteras därför på annan plats i det här numret. Innan jag fick det gjorde jag ett eget som fördrar besked om markeringstervall eller minutkostnad. Bruksanvisning: Slå in endera av dessa och när abonnenten svarar, tryck A vid markeringstervall och B vid minutkostnad. Kostnaden uppdateras var 10:e sekund och visas i fönstret till nästa uppdatering. Då samtalet slutar trycker man R/S. Skulle kostnaden då ej stå i fönstret, tryck E.

IDÉBÖRSSEN

000	43	RCL	018	43	RCL	036	81	RST
001	01	01	019	01	01	037	76	LBL
002	66	PAU	020	71	R/S	038	12	B
003	97	DSZ	021	96	LBL	039	58	FIX
004	00	00	022	11	FX	040	02	02
005	00	00	023	58	FX	041	47	CMS
006	00	00	024	02	02	042	55	+
007	43	RCL	025	47	CMS	043	06	6
008	02	02	026	35	1/X	044	95	+
009	44	SUM	027	65	X	045	42	STO
010	01	01	028	02	2	046	02	01
011	01	01	029	95	=	047	02	02
012	03	3	030	42	STO	048	02	2
013	42	STO	031	02	02	049	42	STO
014	00	00	032	01	1	050	00	00
015	81	RST	033	02	2	051	81	RST
016	76	LBL	034	42	STO	052	00	00
017	15	E	035	00	00			

Att annat tips till program finns i TI PPC Notes V6N9/10P28. Jules Bell fann att han kunde spara Böver en halvtimma per vecka i det stora snabbköp där han handlar. Han skrev en lista på de 100 vanligaste artiklar han köpte och sorterade deras skrivkoder skrivna som decimalbråk (.1314151617 t ex) i bokstavsordning med M/U-modulen. Han spelade in denna lista på magnetkort. Sedan tog han den ordnade listan med sig till snabbköpet och antecknade en tvåsiffrig kod för låget i affären så att nummerordning motsvaras av lämplig gångväg (t ex monter, gång nr). Denna kod adderas sedan till decimalbräket enligt ovan. Han spelade in en omgång magnetkort även den här gången. När man skall gå och handla slår man in registernumret för de varor man vill ha så får man med programmets hjälp en lista med varorna i den ordning de skall plockas i butiken.

Kom gärna med nya idéer så att vi inte behöver lägga ner den här avdelningen p g a idébrist!

BRIDGE-givar

av MARKÚS S MARKÚSSON ISLAND

Från Markús S. Markússon i Garðabær i Island har vi fått det här bidraget. Han skrev det på engelska och det skulle väl ha förstås av majoriteten av våra läsare, men för "principsen" skall här vi ändå översatt det. Dessutom har vi gjort ett litet ingrepp i programmet - på Björn Gustavssons förslag har vi satt in den ändrade uppdelningen i programmet.

I Programbiten 80-2 finns det ett program med bridgegivar av Agne Swerin. När jag såg det blev jag förtjust och knappade omedelbart in det i min TI-59 och började använda det mycket. Många månader senare hade jag blivit ganska trött på utskriftsformatet och på att renons i klöver inte skrevs ut när den förekom. Jag bestämde mig därför att rätta till det hela. Under de följande dagarna lyckades jag skriva ett program som visar fördelningen av kort på nord, syd, öst och väst. Det visar också alltid renons när sådan upprädrar. En ytterligare finess är att giv och valka som är i riskzonen anges.

Uppenbarligen får med denna utskrift antalet kort i en färg inte överstiga tio. Det besvärar mig emellertid inte särskilt mycket, fastän jag en gång hade tio kort i en färg (med det enda saknade honnröskortet, spader, dam, hos partnern!). Jag väntar inte att det skall inträffa igen, ännu mindre att det inte tar fel är chansen att få fler än tio kort i samma färg bara $\binom{13}{10} \binom{2}{2} \binom{13}{10} = 1/26.093.5881$

Programmet använder samma metod som Agne Swerins program, dvs mod 13-metoden. Riskzonen och giv alternerar i en cykel på 16 givar så att första gången nord är giv och ingen i riskzonen. Så vitt jag vet är därför programmet i enlighet med internationella tävlingsnormer.

Jag skall nu ge en kort beskrivning av programmet. 000 - 023 bestämmer och trycker giv och och 112 - 162 riskzonen

PROGRAMBITEN 81-4

- 024 - 084 ger utskrift av händerna
 - 085 - 104 initiering
 - 163 - 200 giv
 - 201 - 257 sortering
 - 258 - 388 gör i ordning händerna för utskrift
 - 389 - 393 avslutning
- Registerinnehåll:
- 64 textkod (NS),(OV) 3136.3242
 - 65 " (ALIA) 1327271300
 - 66 0,1408418
 - 67 0,80140841
 - 68 textkod (N),(OSV) 31.323642
 - 69 " (9TJQKA)X1EE-32, får framställas enligt red. anm. som program), listas som 2.1372534-21.

som synes har vi behållit de engelska beteckningarna för korten, dvs A(ce), K(ing), Q(ueen) och J(ack). Den som vill ha något annat (vi kunde inte hitta något bättre) kan ju lätt ändra det med hjälp av ovanstående förklaringar. - Registren 66 och 67 reglerar vilka som är i riskzonen (steg 138 - 162) och där står O för OV i riskzonen, 1 för NS, 4 för alla och 8 för ingen. Register 68 bestämmer giv.

- Körinstruktion:
- 1. Läs kortsida 1 och 2.
- 2. Lägg ett slumpfärför mellan 0 och 1 i t-registret.
- 3. Ange önskat antal givar.
- 4. Tryck A.

Programbitens kommentar till inmatning av Programmet: Vi har valt att ge program och registerinnehåll i minnena 64 - 69 som en sammanhängande listning i standarduppdelning. Både i programmet (särskilt vid dsz-funktionerna) och i stegen 400-447 som ju motsvarar minnena kommer att fordras en hel del syntetisk programmering. En kod som inte kan programmeras direkt kan ju alltid kodas med STO nn BST BST del SST. Se också upp med en kod som "27 INV" dvs "2nd INV"!

.5072049537

1. N/ -

ROT74
0487
K653
K965
K6532
87
AKQ48754 62

J2
AT94
0442
T93

2. D/NS

J7
7
R82
JT8632

AKT62 953
R4 QJT98532
03 T7
AK04

084
K0
K9654
975

130 01 1 180 35 1/2X
131 00 0 182 55 INT
132 00 0 183 82 HIR
133 45 PFB 184 33 XIT
134 82 5 185 81 4
135 82 5 186 43 RCL
136 52 EE 187 57 37
137 08 8 188 57 37
138 01 1 189 42 87D
139 44 8UM 190 58 58
140 68 8 191 57 57
141 01 1 192 57 57
142 01 1 193 63 EX4
143 01 1 194 78 57
144 01 1 195 57 57
145 01 1 196 57 57
146 01 1 197 57 57
147 42 STD 200 77 77
148 01 1 201 43 STD
149 01 1 202 57 57
150 01 1 203 57 57
151 01 1 204 42 STD
152 01 1 205 57 57
153 01 1 206 01 1
154 01 1 207 02 2
155 01 1 208 57 57
156 01 1 209 57 57
157 01 1 210 43 RCL
158 01 1 211 57 57
159 01 1 212 01 1
160 01 1 213 01 1
161 40 HIR 214 03 3
162 40 HIR 215 42 STD
163 40 HIR 216 56 56
164 40 HIR 217 56 56
165 40 HIR 218 45 RCL
166 40 HIR 219 45 RCL
167 40 HIR 220 45 RCL
168 40 HIR 221 57 37
169 40 HIR 222 57 37
170 40 HIR 223 57 37
171 40 HIR 224 57 37
172 40 HIR 225 57 37
173 40 HIR 226 57 37
174 40 HIR 227 57 37
175 40 HIR 228 57 37
176 40 HIR 229 57 37
177 40 HIR 230 57 37
178 40 HIR 231 57 37
179 40 HIR 232 57 37

3. S/DV

K982
0743
QJT5
2
J7
AK52
K63
AK54

R6543
T96
987
97

4. V/RLLR

J92
T8632.
QJ4
97

K09 854
R4 QJT98532 9
03 T7 9
AK04 8642 R0T3

RT76
K0
AK63
KJ5

240 97 582
241 43 02 02
242 37 27
243 37 27
244 37 27
245 37 27
246 37 27
247 37 27
248 37 27
249 37 27
250 37 27
251 37 27
252 37 27
253 37 27
254 37 27
255 37 27
256 37 27
257 37 27
258 37 27
259 37 27
260 37 27
261 37 27
262 37 27
263 37 27
264 37 27
265 37 27
266 37 27
267 37 27
268 37 27
269 37 27
270 37 27
271 37 27
272 37 27
273 37 27
274 37 27
275 37 27
276 37 27
277 37 27
278 37 27
279 37 27
280 37 27
281 37 27
282 37 27
283 37 27
284 37 27
285 37 27
286 37 27
287 37 27
288 37 27
289 37 27
290 37 27
291 37 27
292 37 27
293 37 27
294 37 27
295 37 27
296 37 27
297 37 27
298 37 27
299 37 27
300 37 27

5. N/NS

K95
093
AK7
AKT9
RJT43
T862
J82
I
J8742

0
AKT
0,196432
65

6. D/OV

87
J532
T8
K9J73

96532 K0
T8 K9764
KJ973 642
T A62

RJT4
R0
R05
9854

300 77 77
301 77 77
302 77 77
303 77 77
304 77 77
305 77 77
306 77 77
307 77 77
308 77 77
309 77 77
310 77 77
311 77 77
312 77 77
313 77 77
314 77 77
315 77 77
316 77 77
317 77 77
318 77 77
319 77 77
320 77 77
321 77 77
322 77 77
323 77 77
324 77 77
325 77 77
326 77 77
327 77 77
328 77 77
329 77 77
330 77 77
331 77 77
332 77 77
333 77 77
334 77 77
335 77 77
336 77 77
337 77 77
338 77 77
339 77 77
340 77 77
341 77 77
342 77 77
343 77 77
344 77 77
345 77 77
346 77 77
347 77 77
348 77 77
349 77 77
350 77 77
351 77 77
352 77 77
353 77 77
354 77 77
355 77 77
356 77 77
357 77 77
358 77 77
359 77 77
360 77 77
361 77 77
362 77 77
363 77 77
364 77 77
365 77 77
366 77 77
367 77 77
368 77 77
369 77 77
370 77 77
371 77 77
372 77 77
373 77 77
374 77 77
375 77 77
376 77 77
377 77 77
378 77 77
379 77 77
380 77 77
381 77 77
382 77 77
383 77 77
384 77 77
385 77 77
386 77 77
387 77 77
388 77 77
389 77 77
390 77 77
391 77 77
392 77 77
393 77 77
394 77 77
395 77 77
396 77 77
397 77 77
398 77 77
399 77 77
400 77 77
401 77 77
402 77 77
403 77 77
404 77 77
405 77 77
406 77 77
407 77 77
408 77 77
409 77 77
410 77 77
411 77 77
412 77 77
413 77 77
414 77 77
415 77 77
416 77 77
417 77 77
418 77 77
419 77 77
420 77 77
421 77 77
422 77 77
423 77 77
424 77 77
425 77 77
426 77 77
427 77 77
428 77 77
429 77 77
430 77 77
431 77 77
432 77 77
433 77 77
434 77 77
435 77 77
436 77 77
437 77 77
438 77 77
439 77 77
440 77 77
441 77 77
442 77 77
443 77 77
444 77 77
445 77 77
446 77 77
447 77 77
448 77 77
449 77 77
450 77 77

GAME of LIFE

av PER-ERIC HOLMBERG

HÅRDVARAN I TI 58

FERMAT's SATS

av GÖSTA BLUME

FRÅN **LABELS**

Avancerade

programmeringsmetoder

av BJÖRN GUSTAVSSON

TILL

DIREKT-ADRESSER

Beräkningar med

KOMPLEXA TAL

med RPN och stack

Några Åsikter Om
SPELMODULEN

Dynamiskt NIM

förmedl

K

st

DEC

BINA

av CHR

S

av GÖSTA BLUME

BLI MEDLEM NU!

Sätt in 120 skr på PG 430 01 59-3

stest

CTER

i-08

**M
N**

Höjdkurveplotter

av Harald M. Oto

...ruenser

FAKULTET för STOR

SPLINE

av HENRIK och JAN
TJERNBERG

**Hur svår var
TIPS-omgången?**

Generaliserad
Binomialfördelning

av CAJ ERLÖ

En TI 59 användares

Regressionsanalys

av J. Worthington och E. Regelman USA

**Prov med
HP-41C**

och en HP-41C-användares kommentar

Skeppsspel

av ANDERS PERSSON

Beräkning av

KONTROLLSIFFRA

av Lars Kristiansson Mölnlycke

Stjärnskytte

av BJÖRN GUSTAVSSON

Twelve days of

CHRISTMAS

**Inkomst-
skatt**

av MIKAEL BENGTSO

TRÖGHETSMOMENT

av INGVAR MAGNUSSON

julramsa med TI 59

KOMMENTARER TILL:

BRÅKFINNARE 2

av GÖSTA BLUME

Diofantiska ekvationen

$$ax + by = c$$

av GÖSTA BLUME

**Snabba
primfaktor
finnare**

En ny sorts
RENGÖRINGSKORT
för kortläsaren i TI 59

**SUPER-
PLOTTER**